RM0475
参考手册

STM32MP13xx
基于 Arm® 内核的 32 位高级 MPU

引言

本参考手册面向应用开发人员，提供有关如何使用 STM32MP13xx 微处理器内存与外设的完整信息。

STM32MP13xx 构成一个微处理器系列，各产品具有不同封装和外设。

有关订购信息以及设备的机械与电气特性，请参见相应的数据手册。

有关 Arm® Cortex®-A7 内核的信息，请参见 Cortex®-A7 *技术参考手册*。

STM32MP13xx 微处理器搭载意法半导体最先进的专利技术。

相关文档

- Cortex®-A7 技术参考手册，可从 http://infocenter.arm.com 获取
- STM32MP13xx 数据手册
- STM32MP131x/3x/5x 设备勘误表 (ES0539)

目录

1 文档约定 109
1.1 概述 109
1.2 寄存器相关缩写词列表 109
1.3 词汇表 110
1.4 外设可用性 110

2 内存和总线架构 111
2.1 系统架构 111
2.1.1 总线架构 111
2.1.2 存储映射结构 113
2.2 AXI 互连矩阵 (AXIM) 116
2.2.1 AXIM 特性 116
2.2.2 AXIM 互连配置 116
2.2.3 主端口描述 117
2.2.4 主端口的主要特性 117
2.2.5 主端口安全特性 118
2.2.6 从端口描述 119
2.2.7 从端口主要特性 119
2.2.8 从端口安全特性 120
2.3 多层 AHB 互连 120
2.3.1 多层 AHB 特性 120
2.3.2 多层 AHB 互连配置 120
2.3.3 多层 AHB 主端口特性 121
2.3.4 多层 AHB 从端口特性 122
2.4 AXIMC 寄存器 122
2.4.1 AXIMC 外设 ID4 寄存器 (AXIMC_PERIPH_ID_4) 122
2.4.2 AXIMC 外设 ID5 寄存器 (AXIMC_PERIPH_ID_5) 123
2.4.3 AXIMC 外设 ID6 寄存器 (AXIMC_PERIPH_ID_6) 123
2.4.4 AXIMC 外设 ID7 寄存器 (AXIMC_PERIPH_ID_7) 124
2.4.5 AXIMC 外设 ID0 寄存器 (AXIMC_PERIPH_ID_0) 124
2.4.6 AXIMC 外设 ID1 寄存器 (AXIMC_PERIPH_ID_1) 125
2.4.7 AXIMC 外设 ID2 寄存器 (AXIMC_PERIPH_ID_2) 125
2.4.8 AXIMC 外设 ID3 寄存器 (AXIMC_PERIPH_ID_3) 126

2.4.9 AXIMC 组件 ID0 寄存器 (AXIMC_COMP_ID_0) 126
2.4.10 AXIMC 组件 ID1 寄存器 (AXIMC_COMP_ID_1) 127
2.4.11 AXIMC 组件 ID2 寄存器 (AXIMC_COMP_ID_2) 127
2.4.12 AXIMC 组件 ID3 寄存器 (AXIMC_COMP_ID_3) 128
2.4.13 AXIMC 主机 x 封装功能寄存器 (AXIMC_Mx_FN_MOD2) 128
2.4.14 AXIMC 主机 x AHB 转换覆盖功能寄存器 (AXIMC_Mx_FN_MOD_AHB) 129
2.4.15 AXIMC 主机 x 读取优先级寄存器 (AXIMC_Mx_READ_QOS) 129
2.4.16 AXIMC 主机 x 写入优先级寄存器 (AXIMC_Mx_WRITE_QOS) 130
2.4.17 AXIMC 主机 x 发出能力覆盖功能寄存器 (AXIMC_Mx_FN_MOD) 130
2.4.18 AXIMC 主机 x 封装功能寄存器 (AXIMC_Mx_FN_MOD2) 131
2.4.19 AXIMC 主机 x AHB 转换覆盖功能寄存器 (AXIMC_Mx_FN_MOD_AHB) 131
2.4.20 AXIMC 主机 x 读取优先级寄存器 (AXIMC_Mx_READ_QOS) 132
2.4.21 AXIMC 主机 x 写入优先级寄存器 (AXIMC_Mx_WRITE_QOS) 132
2.4.22 AXIMC 主机 x 发出能力覆盖功能寄存器 (AXIMC_Mx_FN_MOD) 133
2.4.23 AXIMC 主机 x 读取优先级寄存器 (AXIMC_Mx_READ_QOS) 133
2.4.24 AXIMC 主机 x 写入优先级寄存器 (AXIMC_Mx_WRITE_QOS) 134
2.4.25 AXIMC 主机 x 发出能力覆盖功能寄存器 (AXIMC_Mx_FN_MOD) 134
2.4.26 AXIMC 主机 x 读取优先级寄存器 (AXIMC_Mx_READ_QOS) 135
2.4.27 AXIMC 主机 x 写入优先级寄存器 (AXIMC_Mx_WRITE_QOS) 135
2.4.28 AXIMC 主机 x 发出能力覆盖功能寄存器 (AXIMC_Mx_FN_MOD) 136
2.4.29 AXIMC 长突发能力抑制寄存器 (AXIMC_M4_FN_MOD_LB) 136
2.4.30 AXIMC 主机 x 读取优先级寄存器 (AXIMC_Mx_READ_QOS) 137
2.4.31 AXIMC 主机 x 写入优先级寄存器 (AXIMC_Mx_WRITE_QOS) 137
2.4.32 AXIMC 主机 x 发出能力覆盖功能寄存器 (AXIMC_Mx_FN_MOD) 138
2.4.33 AXIMC 寄存器映射 138
2.5 存储结构 147
2.5.1 简介 147
2.5.2 内存映射和寄存器边界地址 148

3 启动、安全性和 OTP 控制 (BSEC) 159
3.1 简介 159
3.2 BSEC 主要特性 159
3.3 BSEC 功能描述 160
3.3.1 BSEC 框图 160
3.3.2 OTP 接口 161

3.3.3 OTP 安全模式 161
3.3.4 系统复位时自动加载 OTP 162
3.3.5 OTP 字状态 163
3.3.6 OTP 操作 164
3.3.7 JTAG 寄存器 165
3.3.8 BSEC 锁定寄存器 165
3.3.9 调试控制 166
3.3.10 粘性锁定寄存器 166
3.4 OTP 布局 166
3.4.1 DMA 请求 167
3.4.2 TrustZone 感知 167
3.4.3 BSEC Tamper 保护 167
3.4.4 BSEC 时钟和初始化 168
3.5 BSEC 中断 168
3.6 BSEC 寄存器 168
3.6.1 BSEC OTP 配置寄存器 (BSEC_OTP_CONFIG) 168
3.6.2 BSEC OTP 控制寄存器 (BSEC_OTP_CONTROL) 169
3.6.3 BSEC OTP 写入数据寄存器 (BSEC_OTP_WRDATA) 170
3.6.4 BSEC OTP 状态寄存器 (BSEC_OTP_STATUS) 170
3.6.5 BSEC OTP 锁定配置寄存器 (BSEC_OTP_LOCK) 171
3.6.6 BSEC 调试配置寄存器 (BSEC_DENABLE) 172
3.6.7 BSEC OTP 干扰状态寄存器 x (BSEC_OTP_DISTURBEDx) 173
3.6.8 BSEC OTP 错误状态寄存器 x (BSEC_OTP_ERRORx) 174
3.6.9 BSEC_OTP 锁定状态寄存器 x (BSEC_OTP_WRLOCKEDx) 174
3.6.10 BSEC OTP 粘性编程锁定寄存器 x (BSEC_OTP_SPLOCKx) 175
3.6.11 BSEC_OTP 影子写入粘性锁定寄存器 x (BSEC_OTP_SWLOCKx) . . 176
3.6.12 BSEC OTP 影子读取粘性锁定寄存器 x (BSEC_OTP_SRLOCKx) . . . 177
3.6.13 BSEC JTAG 输入寄存器 (BSEC_JTAGIN) 177
3.6.14 BSEC JTAG 输出寄存器 (BSEC_JTAGOUT) 178
3.6.15 BSEC 暂存寄存器 (BSEC_SCRATCH) 178
3.6.16 BSEC 影子寄存器 x (BSEC_OTP_DATAx) 178
3.6.17 BSEC 硬件配置寄存器 (BSEC_HWCFGR) 179
3.6.18 BSEC 版本寄存器 (BSEC_VERR) 179
3.6.19 BSEC 标识寄存器 (BSEC_IPIDR) 180
3.6.20 BSEC 大小标识寄存器 (BSEC_SIDR) 180
3.6.21 BSEC 寄存器映射 181

4 OTP 映射 (OTP) 184

5 DDR3/LPDDR2/LPDDR3 控制器 (DDRCTRL) 198

5.1 简介 198
5.2 DDRCTRL 特性 198
5.3 DDRCTRL 框图 199
5.4 DDRCTRL 架构概述 200
5.4.1 AXI 端口接口 (XPI) 200
5.4.2 端口仲裁器 (PA) 201
5.4.3 主机接口 (HIF) 201
5.4.4 DDR 调度器 (DDRC) 202
5.4.5 APB 接口 (APB) 202
5.4.6 DFI 接口 (DFI) 202
5.5 DDRCTRL 功能描述 203
5.5.1 事务服务控制 (TSC) 和服务质量 (QoS) 203
5.5.2 页策略选项 204
5.5.3 节能特性 204
5.5.4 地址映射器 205
5.5.5 DRAM 时序参数 205
5.5.6 SDRAM 初始化序列 205
5.5.7 刷新控制 206
5.5.8 ZQ 校准 206
5.6 DDRCTRL 配置 207
5.7 DDRCTRL 寄存器 207
5.7.1 DDRCTRL 主寄存器 0 (DDRCTRL_MSTR) 208
5.7.2 DDRCTRL 工作模式状态寄存器 (DDRCTRL_STAT) 210
5.7.3 DDRCTRL 模式寄存器读/写控制寄存器 0 (DDRCTRL_MRCTRL0) . . 211
5.7.4 DDRCTRL 模式寄存器读/写控制寄存器 1 (DDRCTRL_MRCTRL1) . . 212
5.7.5 DDRCTRL 模式寄存器读/写状态寄存器 (DDRCTRL_MRSTAT) 213
5.7.6 DDRCTRL 温度降额使能寄存器 (DDRCTRL_DERATEEN) 213
5.7.7 DDRCTRL 温度降额间隔寄存器 (DDRCTRL_DERATEINT) 214
5.7.8 DDRCTRL 低功耗控制寄存器 (DDRCTRL_PWRCTL) 215
5.7.9 DDRCTRL 低功耗时序寄存器 (DDRCTRL_PWRTMG) 216
5.7.10 DDRCTRL 硬件低功耗控制寄存器 (DDRCTRL_HWLPCTL) 217
5.7.11 DDRCTRL 刷新控制寄存器 0 (DDRCTRL_RFSHCTL0) 218
5.7.12 DDRCTRL 刷新控制寄存器 3 (DDRCTRL_RFSHCTL3) 220

5.7.13 DDRCTRL 刷新时序寄存器 (DDRCTRL_RFSHTMG) 221
5.7.14 DDRCTRL CRC 奇偶校验控制寄存器 0 (DDRCTRL_CRCPARCTL0) . . 222
5.7.15 DDRCTRL CRC 奇偶校验状态寄存器 (DDRCTRL_CRCPARSTAT) . . 223
5.7.16 DDRCTRL SDRAM 初始化寄存器 0 (DDRCTRL_INIT0) 224
5.7.17 DDRCTRL SDRAM 初始化寄存器 1 (DDRCTRL_INIT1) 225
5.7.18 DDRCTRL SDRAM 初始化寄存器 2 (DDRCTRL_INIT2) 225
5.7.19 DDRCTRL SDRAM 初始化寄存器 3 (DDRCTRL_INIT3) 226
5.7.20 DDRCTRL SDRAM 初始化寄存器 4 (DDRCTRL_INIT4) 227
5.7.21 DDRCTRL SDRAM 初始化寄存器 5 (DDRCTRL_INIT5) 227
5.7.22 DDRCTRL DIMM 控制寄存器 (DDRCTRL_DIMMCTL) 228
5.7.23 DDRCTRL SDRAM 时序寄存器 0 (DDRCTRL_DRAMTMG0) 229
5.7.24 DDRCTRL SDRAM 时序寄存器 1 (DDRCTRL_DRAMTMG1) 230
5.7.25 DDRCTRL SDRAM 时序寄存器 2 (DDRCTRL_DRAMTMG2) 231
5.7.26 DDRCTRL SDRAM 时序寄存器 3 (DDRCTRL_DRAMTMG3) 234
5.7.27 DDRCTRL SDRAM 时序寄存器 4 (DDRCTRL_DRAMTMG4) 235
5.7.28 DDRCTRL SDRAM 时序寄存器 5 (DDRCTRL_DRAMTMG5) 236
5.7.29 DDRCTRL SDRAM 时序寄存器 6 (DDRCTRL_DRAMTMG6) 238
5.7.30 DDRCTRL SDRAM 时序寄存器 7 (DDRCTRL_DRAMTMG7) 239
5.7.31 DDRCTRL SDRAM 时序寄存器 8 (DDRCTRL_DRAMTMG8) 240
5.7.32 DDRCTRL SDRAM 时序寄存器 14 (DDRCTRL_DRAMTMG14) 240
5.7.33 DDRCTRL SDRAM 时序寄存器 15 (DDRCTRL_DRAMTMG15) 241
5.7.34 DDRCTRL ZQ 控制寄存器 0 (DDRCTRL_ZQCTL0) 242
5.7.35 DDRCTRL ZQ 控制寄存器 1 (DDRCTRL_ZQCTL1) 243
5.7.36 DDRCTRL ZQ 控制寄存器 2 (DDRCTRL_ZQCTL2) 244
5.7.37 DDRCTRL ZQ 状态寄存器 (DDRCTRL_ZQSTAT) 244
5.7.38 DDRCTRL DFI 时序寄存器 0 (DDRCTRL_DFITMG0) 245
5.7.39 DDRCTRL DFI 时序寄存器 1 (DDRCTRL_DFITMG1) 246
5.7.40 DDRCTRL 低功耗配置寄存器 0 (DDRCTRL_DFILPCFG0) 247
5.7.41 DDRCTRL DFI 更新寄存器 0 (DDRCTRL_DFIUPD0) 250
5.7.42 DDRCTRL DFI 更新寄存器 1 (DDRCTRL_DFIUPD1) 251
5.7.43 DDRCTRL DFI 更新寄存器 2 (DDRCTRL_DFIUPD2) 252
5.7.44 DDRCTRL DFI 杂项控制寄存器 (DDRCTRL_DFIMISC) 252
5.7.45 DDRCTRL DFI 状态寄存器 (DDRCTRL_DFISTAT) 253
5.7.46 DDRCTRL DFI PHY 主寄存器 (DDRCTRL_DFIPHYMSTR) 254
5.7.47 DDRCTRL 地址映射寄存器 1 (DDRCTRL_ADDRMAP1) 254
5.7.48 DDRCTRL 地址映射寄存器 2 (DDRCTRL_ADDRMAP2) 255
5.7.49 DDRCTRL 地址映射寄存器 3 (DDRCTRL_ADDRMAP3) 257

5.7.50 DDRCTRL 地址映射寄存器 4 (DDRCTRL_ADDRMAP4) 260
5.7.51 DDRCTRL 地址映射寄存器 5 (DDRCTRL_ADDRMAP5) 261
5.7.52 DDRCTRL 地址寄存器 6 (DDRCTRL_ADDRMAP6) 262
5.7.53 DDRCTRL 地址映射寄存器 9 (DDRCTRL_ADDRMAP9) 264
5.7.54 DDRCTRL 地址映射寄存器 10 (DDRCTRL_ADDRMAP10) 265
5.7.55 DDRCTRL 地址映射寄存器 11 (DDRCTRL_ADDRMAP11) 266
5.7.56 DDRCTRL ODT 配置寄存器 (DDRCTRL_ODTCFG) 266
5.7.57 DDRCTRL ODT/Rank 映射寄存器 (DDRCTRL_ODTMAP) 268
5.7.58 DDRCTRL 调度器控制寄存器 (DDRCTRL_SCHED) 269
5.7.59 DDRCTRL 调度器控制寄存器 1 (DDRCTRL_SCHED1) 270
5.7.60 DDRCTRL 高优先级读取 CAM 寄存器 1 (DDRCTRL_PERFHPR1) . . 271
5.7.61 DDRCTRL 低优先级读取 CAM 寄存器 1 (DDRCTRL_PERFLPR1) . . 272
5.7.62 DDRCTRL 写入 CAM 寄存器 1 (DDRCTRL_PERFWR1) 272
5.7.63 DDRCTRL 调试寄存器 0 (DDRCTRL_DBG0) 273
5.7.64 DDRCTRL 调试寄存器 1 (DDRCTRL_DBG1) 274
5.7.65 DDRCTRL CAM 调试寄存器 (DDRCTRL_DBGCAM) 274
5.7.66 DDRCTRL 命令调试寄存器 (DDRCTRL_DBGCMD) 276
5.7.67 DDRCTRL 状态调试寄存器 (DDRCTRL_DBGSTAT) 277
5.7.68 DDRCTRL 软件寄存器编程控制使能 (DDRCTRL_SWCTL) 278
5.7.69 DDRCTRL 软件寄存器编程控制状态 (DDRCTRL_SWSTAT) 278
5.7.70 DDRCTRL AXI Poison配置寄存器 (DDRCTRL_POISONCFG) 279
5.7.71 DDRCTRL AXI Poison 状态寄存器 (DDRCTRL_POISONSTAT) 280
5.7.72 DDRCTRL 端口状态寄存器 (DDRCTRL_PSTAT) 281
5.7.73 DDRCTRL 端口通用配置寄存器 (DDRCTRL_PCCFG) 281
5.7.74 DDRCTRL 端口 x 配置读取寄存器 (DDRCTRL_PCFGR_0) 283
5.7.75 DDRCTRL 端口 x 配置写入寄存器 (DDRCTRL_PCFGW_0) 284
5.7.76 DDRCTRL 端口 x 控制寄存器 (DDRCTRL_PCTRL_0) 285
5.7.77 DDRCTRL 端口 x 读取 Q0S 配置寄存器 0 (DDRCTRL_PCFGQOS0_0) 286
5.7.78 DDRCTRL 端口 x 读取 Q0S 配置寄存器 1 (DDRCTRL_PCFGQOS1_0) 287
5.7.79 DDRCTRL 端口 x 写入 Q0S 配置寄存器 0 (DDRCTRL_PCFGWQOS0_0) 287
5.7.80 DDRCTRL 端口 x 写入 Q0S 配置寄存器 1 (DDRCTRL_PCFGWQOS1_0) 289
5.7.81 DDRCTRL 寄存器总结 290

6 用于 DDR 的 TrustZone® 地址空间控制器 (TZC) 305

6.1 简介 305
6.2 TZC 特性 305
6.3 TZC 框图 306
6.4 TZC 功能描述 307
6.5 TZC 应用编程示例 308
6.5.1 DMA 请求 312
6.5.2 中断 312
6.6 TZC 寄存器 313
6.6.1 TZC 配置寄存器 (TZC_BUILD_CONFIG) 313
6.6.2 TZC 操作寄存器 (TZC_ACTION) 313
6.6.3 TZC 门卫寄存器 (TZC_GATE_KEEPER) 314
6.6.4 TZC 推测控制寄存器 (TZC_SPECULATION_CTRL) 315
6.6.5 TZC 中断状态寄存器 (TZC_INT_STATUS) 315
6.6.6 TZC 中断清除寄存器 (TZC_INT_CLEAR) 316
6.6.7 TZC 失败地址低位寄存器 (TZC_FAIL_ADDRESS_LOW) 316
6.6.8 TZC 失败地址高位寄存器 (TZC_FAIL_ADDRESS_HIGH) 317
6.6.9 TZC 失败控制寄存器 (TZC_FAIL_CONTROL) 317
6.6.10 TZC 失败 ID 寄存器 (TZC_FAIL_ID) 318
6.6.11 TZC 区域 0 基址低位寄存器 (TZC_REGION_BASE_LOW0) 318
6.6.12 TZC 区域 x 基址高位寄存器 (TZC_REGION_BASE_HIGHx) 319
6.6.13 TZC 区域 0 顶地址低位寄存器 (TZC_REGION_TOP_LOW0) 319
6.6.14 TZC 区域 x 顶地址高位寄存器 (TZC_REGION_TOP_HIGHx) 320
6.6.15 TZC 区域 0 属性寄存器 (TZC_REGION_ATTRIBUTE0) 320
6.6.16 TZC 区域 x ID 访问寄存器 (TZC_REGION_ID_ACCESSx) 321
6.6.17 TZC 区域 x 基址低位寄存器 (TZC_REGION_BASE_LOWx) 321
6.6.18 TZC 区域 x 顶地址低位寄存器 (TZC_REGION_TOP_LOWx) 322
6.6.19 TZC 区域 x 属性寄存器 (TZC_REGION_ATTRIBUTEx) 322
6.6.20 TZC 外设 ID 4 寄存器 (TZC_PID4) 323
6.6.21 TZC 外设 ID 5 寄存器 (TZC_PID5) 323
6.6.22 TZC 外设 ID 6 寄存器 (TZC_PID6) 323
6.6.23 TZC 外设 ID 7 寄存器 (TZC_PID7) 324
6.6.24 TZC 外设 ID 0 寄存器 (TZC_PID0) 324
6.6.25 TZC 外设 ID 1 寄存器 (TZC_PID1) 325
6.6.26 TZC 外设 ID 2 寄存器 (TZC_PID2) 325
6.6.27 TZC 外设 ID 3 寄存器 (TZC_PID3) 325

6.6.28 TZC 组件 ID 0 寄存器 (TZC_CID0) 326
6.6.29 TZC 组件 ID 1 寄存器 (TZC_CID1) 326
6.6.30 TZC 组件 ID 2 寄存器 (TZC_CID2) 327
6.6.31 TZC 组件 ID 3 寄存器 (TZC_CID3) 327
6.6.32 TZC 寄存器映射 328

7 DDR 物理接口控制 (DDRPHYC) 334

7.1 DDRPHYC 特性 334
7.2 DDRPHYC 框图 334
7.3 DDRPHYC SDRAM 接口 335
7.4 DDRPHY（DDR 多 PHY IP 内核） 336
7.4.1 DDRPHY 概览 336
7.4.2 延迟锁相环 (DLL) 337
7.4.3 接口时序模块 (ITM) 339
7.4.4 SSTL IO 344
7.5 PUBL 345
7.5.1 PHY 和 SDRAM 初始化 346
7.5.2 DDR3 初始化序列 348
7.5.3 LPDDR2 初始化序列 349
7.5.4 初始化触发和旁路 349
7.5.5 数据训练 349
7.5.6 DQS 门控训练 (DQSTRN) 350
7.5.7 读取有效训练 (RVTRN) 354
7.6 PUBL 时序和 DFI 接口 354
7.6.1 简介 354
7.6.2 写时序 355
7.6.3 读时序 356
7.6.4 DDRPHY 读/写延迟 357
7.6.5 DFI 2.1 概述 358
7.7 PUBL 寄存器 360
7.7.1 DDRPHYC 版本 ID 寄存器 (DDRPHYC_RIDR) 360
7.7.2 DDRPHYC PHY 初始化寄存器 (DDRPHYC_PIR) 360
7.7.3 DDRPHYC PHY 全局控制寄存器 (DDRPHYC_PGCR) 363
7.7.4 DDRPHYC PHY 全局状态寄存器 (DDRPHYC_PGSR) 365
7.7.5 DDRPHYC DDR DLL 全局控制寄存器 (DDRPHYC_DLLGCR) 366
7.7.6 DDRPHYC AC DLL 控制寄存器 (DDRPHYC_ACDLLCR) 368

7.7.7 DDRPHYC PT 寄存器 0 (DDRPHYC_PTR0) 369
7.7.8 DDRPHYC PT 寄存器 1 (DDRPHYC_PTR1) 369
7.7.9 DDRPHYC PT 寄存器 2 (DDRPHYC_PTR2) 370
7.7.10 DDRPHYC ACIOC 寄存器 (DDRPHYC_ACIOCR) 371
7.7.11 DDRPHYC DXCC 寄存器 (DDRPHYC_DXCCR) 372
7.7.12 DDRPHYC DSGC 寄存器 (DDRPHYC_DSGCR) 373
7.7.13 DDRPHYC DC 寄存器 (DDRPHYC_DCR) 376
7.7.14 DDRPHYC DTP 寄存器 0 (DDRPHYC_DTPR0) 377
7.7.15 DDRPHYC DTP 寄存器 1 (DDRPHYC_DTPR1) 378
7.7.16 DDRPHYC DTP 寄存器 2 (DDRPHYC_DTPR2) 380
7.7.17 DDRPHYC DDR3 的 MR0 寄存器 (DDRPHYC_DDR3_MR0) 380
7.7.18 DDRPHYC 用于 DDR3 的 MR1 寄存器 (DDRPHYC_DDR3_MR1) . . 382
7.7.19 DDRPHYC 用于 LPDDR2 的 MR1 寄存器 (DDRPHYC_LPDDR2_MR1) 383
7.7.20 DDRPHYC 用于 LPDDR3 的 MR1 寄存器 (DDRPHYC_LPDDR3_MR1) 384
7.7.21 DDRPHYC 用于 DDR3 的 MR2 寄存器 (DDRPHYC_DDR3_MR2) . . 384
7.7.22 DDRPHYC 用于 LPDDR2 的 MR2 寄存器 (DDRPHYC_LPDDR2_MR2) 385
7.7.23 DDRPHYC 用于 LPDDR3 的 MR2 寄存器 (DDRPHYC_LPDDR3_MR2) 386
7.7.24 DDRPHYC 用于 DDR3 的 MR3 寄存器 (DDRPHYC_DDR3_MR3) . . 386
7.7.25 DDRPHYC ODTC 寄存器 (DDRPHYC_ODTCR) 387
7.7.26 DDRPHYC DTA 寄存器 (DDRPHYC_DTAR) 387
7.7.27 DDRPHYC DTD 寄存器 0 (DDRPHYC_DTDR0) 388
7.7.28 DDRPHYC DTD 寄存器 1 (DDRPHYC_DTDR1) 388
7.7.29 DDRPHYC 通用寄存器 0 (DDRPHYC_GPR0) 389
7.7.30 DDRPHYC 通用寄存器 1 (DDRPHYC_GPR1) 389
7.7.31 DDRPHYC ZQ0C 寄存器 0 (DDRPHYC_ZQ0CR0) 389
7.7.32 DDRPHYC ZQ0CR1 寄存器 (DDRPHYC_ZQ0CR1) 390
7.7.33 DDRPHYC ZQ0S 寄存器 0 (DDRPHYC_ZQ0SR0) 391
7.7.34 DDRPHYC ZQ0S 寄存器 1 (DDRPHYC_ZQ0SR1) 392
7.7.35 DDRPHYC 字节通道 n GC 寄存器 (DDRPHYC_DXnGCR) 393
7.7.36 DDRPHYC 字节通道 n GS 寄存器 0 (DDRPHYC_DXnGSR0) 395
7.7.37 DDRPHYC 数据通道 n GS 寄存器 1 (DDRPHYC_DXnGSR1) 395
7.7.38 DDRPHYC 字节通道 n DLLC 寄存器 (DDRPHYC_DXnDLLCR) 396
7.7.39 DDRPHYC 字节通道 n DQT 寄存器 (DDRPHYC_DXnDQTR) 398
7.7.40 DDRPHYC 字节通道 n DQST 寄存器 (DDRPHYC_DXnDQSTR) 399

7.7.41 DDRPHYC 寄存器映射 401

8 DDR 性能监控器 (DDRPERFM) 406

8.1 简介 406
8.2 特性 406
8.3 框图 406
8.4 功能描述 407
8.5 应用信息 408
8.5.1 信号组 0：用于监视 DDR 利用率 408
8.5.2 信号组 1：监视 DDR 命令存储 409
8.5.3 信号组 2：监视刷新和自刷新持续时间 409
8.5.4 信号组 3：监视 DDR 停顿 409
8.6 DDRPERFM 寄存器 410
8.6.1 DDRPERFM 控制寄存器 (DDRPERFM_CTL) 410
8.6.2 DDRPERFM 配置寄存器 (DDRPERFM_CFG) 410
8.6.3 DDRPERFM 状态寄存器 (DDRPERFM_STATUS) 411
8.6.4 DDRPERFM 计数器清零寄存器 (DDRPERFM_CCR) 411
8.6.5 DDRPERFM 中断使能寄存器 (DDRPERFM_IER) 412
8.6.6 DDRPERFM 中断状态寄存器 (DDRPERFM_ISR) 412
8.6.7 DDRPERFM 中断清除寄存器 (DDRPERFM_ICR) 413
8.6.8 DDRPERFM 时间计数器寄存器 (DDRPERFM_TCNT) 413
8.6.9 DDRPERFM 事件计数器 x 寄存器 (DDRPERFM_CNTx) 414
8.6.10 DDRPERFM 硬件配置寄存器 (DDRPERFM_HWCFG) 414
8.6.11 DDRPERFM 版本寄存器 (DDRPERFM_VER) 414
8.6.12 DDRPERFM ID 寄存器 (DDRPERFM_ID) 415
8.6.13 DDRPERFM 魔术 ID 寄存器 (DDRPERFM_SID) 415
8.6.14 DDRPERFM 寄存器映射 416

9 电源控制 (PWR) 417

9.1 PWR 主要特性 417
9.2 PWR 框图 418
9.2.1 引脚概述 419
9.3 PWR 电源 421
9.3.1 CORE 和 MPU 电源域 423
9.3.2 电源系统启动序列 423
9.3.3 VSW 域 424

9.3.4 VBAT 电池充电 427
9.3.5 模拟电源 427
9.3.6 1V8 稳压器 427
9.3.7 1V1 稳压器 427
9.4 PWR 电源监控 428
9.4.1 上电复位 (POR)/掉电复位 (PDR) 429
9.4.2 欠压复位 (BOR) 430
9.4.3 VDDCORE OK 和 VDDCPU OK 复位 431
9.4.4 可编程电压检测器 (PVD) 432
9.4.5 模拟电压检测器 (AVD) 433
9.4.6 电池电压阈值 434
9.4.7 温度阈值 435
9.5 PWR 电源管理 436
9.5.1 工作模式 436
9.5.2 系统工作模式和外部稳压器模式 442
9.5.3 系统唤醒 444
9.6 PWR 低功耗工作模式 444
9.6.1 降低 CPU 时钟和/或总线矩阵时钟速度 444
9.6.2 控制外设时钟 444
9.6.3 进入低功耗模式 444
9.6.4 使用 WFI 或 WFE 退出低功耗模式 445
9.6.5 MPU CSleep 模式 445
9.6.6 MPU CStop 模式 446
9.6.7 系统 Stop 模式 446
9.6.8 系统 LP-Stop 模式 448
9.6.9 系统 LPLV-Stop 模式 450
9.6.10 系统 LPLV-Stop2 模式 451
9.6.11 系统待机模式 453
9.6.12 系统 VBAT 模式 456
9.6.13 Stop 调试 457
9.6.14 待机调试 458
9.7 PWR WKUP 引脚 458
9.8 PWR_ON、PWR_CPU_ON 和 PWR_LP 引脚 458
9.9 PWR 中断 459
9.10 PWR TrustZone 安全性 460
9.11 PWR 寄存器 461

9.11.1 PWR 控制寄存器 1 (PWR_CR1) 461
9.11.2 PWR 控制状态寄存器 1 (PWR_CSR1) 463
9.11.3 PWR 控制寄存器 2 (PWR_CR2) 463
9.11.4 PWR 控制寄存器 3 (PWR_CR3) 465
9.11.5 PWR MPU 控制寄存器 (PWR_MPUCR) 467
9.11.6 PWR 唤醒控制寄存器 (PWR_WKUPCR) 468
9.11.7 PWR 唤醒标志寄存器 (PWR_WKUPFR) 470
9.11.8 PWR MPU 唤醒使能寄存器 (PWR_MPUWKUPENR) 470
9.11.9 PWR IP 版本寄存器 (PWR_VERR) 471
9.11.10 PWR IP 标识寄存器 (PWR_ID) 471
9.11.11 PWR 大小 ID 寄存器 (PWR_SID) 472
9.11.12 PWR 寄存器映射 472

10 复位和时钟控制 (RCC) 474

10.1 RCC 主要特性 474
10.2 RCC 框图 475
10.2.1 RCC 引脚概述 476
10.3 RCC 功能描述——复位部分 477
10.3.1 复位概述 477
10.3.2 电源复位 480
10.3.3 看门狗复位 481
10.3.4 备份域复位 481
10.3.5 Coresight 调试复位 481
10.3.6 选项字节加载 481
10.3.7 按外设复位 481
10.3.8 复位脉冲控制 (RPCTL) 482
10.3.9 复位延迟控制 (RDCTL) 483
10.3.10 复位覆盖范围汇总 485
10.3.11 复位源标识 486
10.3.12 启动、冷/热复位和唤醒序列 488
10.4 RCC 功能描述——时钟部分 490
10.4.1 时钟命名约定 492
10.4.2 振荡器说明 492
10.4.3 时钟安全系统 (CSS) 499
10.4.4 时钟输出生成（MCO1 和 MCO2） 501
10.4.5 PLL 描述 501
10.4.6 子系统时钟 508

10.4.7 Stop 模式和待机模式下的时钟生成 512
10.4.8 外设时钟分配 519
10.4.9 常规时钟概念概述 557
10.4.10 外设分配 557
10.4.11 外设时钟门控控制 558
10.4.12 处理器和互连时钟门控 561
10.4.13 低功耗仿真模式 565
10.5 RCC 安全性 566
10.6 RCC 中断 570
10.7 RCC 应用信息 572
10.7.1 处理动态时钟切换 572
10.7.2 PLL 编程 577
10.7.3 配置子系统时钟 581
10.7.4 使用 TIMx 进行时钟校准 583
10.7.5 时钟恢复序列示例 584
10.8 RCC 寄存器 586
10.8.1 RCC 安全配置寄存器 (RCC_SECCFGR) 586
10.8.2 RCC 停止请求设置寄存器 (RCC_MP_SREQSETR) 588
10.8.3 RCC 停止请求清除寄存器 (RCC_MP_SREQCLRR) 589
10.8.4 RCC 应用复位控制寄存器 (RCC_MP_APRSTCR) 590
10.8.5 RCC 应用复位状态寄存器 (RCC_MP_APRSTSR) 591
10.8.6 RCC 低功耗 Stop 模式延迟控制寄存器 (RCC_PWRLPDLYCR) 592
10.8.7 RCC 全局复位控制设置寄存器 (RCC_MP_GRSTCSETR) 593
10.8.8 RCC Boot ROM 复位状态清除寄存器 (RCC_BR_RSTSCLRR) 594
10.8.9 RCC 复位状态设置寄存器 (RCC_MP_RSTSSETR) 596
10.8.10 RCC 复位状态清除寄存器 (RCC_MP_RSTSCLRR) 598
10.8.11 RCC IWDG 时钟冻结设置寄存器 (RCC_MP_IWDGFZSETR) 600
10.8.12 RCC IWDG 时钟冻结清除寄存器 (RCC_MP_IWDGFZCLRR) 601
10.8.13 RCC 时钟源中断使能寄存器 (RCC_MP_CIER) 602
10.8.14 RCC 时钟源中断标志寄存器 (RCC_MP_CIFR) 605
10.8.15 RCC 备份域控制寄存器 (RCC_BDCR) 608
10.8.16 RCC 复位持续时间和 LSI 控制寄存器 (RCC_RDLSICR) 610
10.8.17 RCC 振荡器时钟使能设置寄存器 (RCC_OCENSETR) 612
10.8.18 RCC 振荡器时钟使能清零寄存器 (RCC_OCENCLRR) 614
10.8.19 RCC 振荡器和 MPU 与 AXI 时钟就绪寄存器 (RCC_OCRDYR) 616
10.8.20 RCC HSI 配置寄存器 (RCC_HSICFGR) 618
10.8.21 RCC CSI 配置寄存器 (RCC_CSICFGR) 619

10.8.22 RCC MCO1 配置寄存器 (RCC_MCO1CFGR) 620
10.8.23 RCC MCO2 配置寄存器 (RCC_MCO2CFGR) 621
10.8.24 RCC 调试配置寄存器 (RCC_DBGCFGR) 622
10.8.25 RCC PLL1 和 PLL2 参考时钟选择寄存器 (RCC_RCK12SELR) 623
10.8.26 RCC PLL3 参考时钟选择寄存器 (RCC_RCK3SELR) 624
10.8.27 RCC PLL4 参考时钟选择寄存器 (RCC_RCK4SELR) 625
10.8.28 RCC PLL1 控制寄存器 (RCC_PLL1CR) 626
10.8.29 RCC PLL1 配置寄存器 1 (RCC_PLL1CFGR1) 628
10.8.30 RCC PLL1 配置寄存器 2 (RCC_PLL1CFGR2) 629
10.8.31 RCC PLL1 小数寄存器 (RCC_PLL1FRACR) 630
10.8.32 RCC PLL1 时钟展频发生器寄存器 (RCC_PLL1CSGR) 631
10.8.33 RCC PLL2 控制寄存器 (RCC_PLL2CR) 632
10.8.34 RCC PLL2 配置寄存器 1 (RCC_PLL2CFGR1) 634
10.8.35 RCC PLL2 配置寄存器 2 (RCC_PLL2CFGR2) 635
10.8.36 RCC PLL2 小数寄存器 (RCC_PLL2FRACR) 636
10.8.37 RCC PLL2 时钟展频发生器寄存器 (RCC_PLL2CSGR) 637
10.8.38 RCC PLL3 控制寄存器 (RCC_PLL3CR) 638
10.8.39 RCC PLL3 配置寄存器 1 (RCC_PLL3CFGR1) 640
10.8.40 RCC PLL3 配置寄存器 2 (RCC_PLL3CFGR2) 642
10.8.41 RCC PLL3 小数寄存器 (RCC_PLL3FRACR) 643
10.8.42 RCC PLL3 时钟展频发生器寄存器 (RCC_PLL3CSGR) 644
10.8.43 RCC PLL4 控制寄存器 (RCC_PLL4CR) 645
10.8.44 RCC PLL4 配置寄存器 1 (RCC_PLL4CFGR1) 647
10.8.45 RCC PLL4 配置寄存器 2 (RCC_PLL4CFGR2) 649
10.8.46 RCC PLL4 小数寄存器 (RCC_PLL4FRACR) 650
10.8.47 RCC PLL4 时钟展频发生器寄存器 (RCC_PLL4CSGR) 651
10.8.48 RCC MPU 时钟选择寄存器 (RCC_MPCKSELR) 652
10.8.49 RCC AXI 子系统时钟选择寄存器 (RCC_ASSCKSELR) 653
10.8.50 RCC MLAHB 子系统时钟选择寄存器 (RCC_MSSCKSELR) 654
10.8.51 RCC 通用外设内核时钟选择寄存器 (RCC_CPERCKSELR) 655
10.8.52 RCC RTC 时钟分频寄存器 (RCC_RTCDIVR) 656
10.8.53 RCC MPU 时钟分频器寄存器 (RCC_MPCKDIVR) 657
10.8.54 RCC AXI 时钟分频器寄存器 (RCC_AXIDIVR) 658
10.8.55 RCC MLAHB 时钟分频器寄存器 (RCC_MLAHBDIVR) 659
10.8.56 RCC APB1 时钟分频器寄存器 (RCC_APB1DIVR) 660
10.8.57 RCC APB2 时钟分频器寄存器 (RCC_APB2DIVR) 661
10.8.58 RCC APB3 时钟分频器寄存器 (RCC_APB3DIVR) 662

10.8.59 RCC APB4 时钟分频寄存器 (RCC_APB4DIVR) 663
10.8.60 RCC APB5 时钟分频器寄存器 (RCC_APB5DIVR) 664
10.8.61 RCC APB6 时钟分频器寄存器 (RCC_APB6DIVR) 665
10.8.62 RCC APB1 定时器组 1 预分频器寄存器 (RCC_TIMG1PRER) 666
10.8.63 RCC APB2 定时器组 2 预分频器寄存器 (RCC_TIMG2PRER) 667
10.8.64 RCC APB6 定时器组 3 预分频器寄存器 (RCC_TIMG3PRER) 668
10.8.65 RCC DDR 接口控制寄存器 (RCC_DDRITFCR) 669
10.8.66 RCC I2C1 和 I2C2 内核时钟选择寄存器 (RCC_I2C12CKSELR) 674
10.8.67 RCC I2C3、I2C4 和 I2C5 内核时钟选择寄存器 (RCC_I2C345CKSELR) 675
10.8.68 RCC SPI/I2S1 内核时钟选择寄存器 (RCC_SPI2S1CKSELR) 676
10.8.69 RCC SPI/I2S2 和 SPI/I2S3 内核时钟选择寄存器 (RCC_SPI2S23CKSELR) 677
10.8.70 RCC SPI/I2S4 和 SPI5 内核时钟选择寄存器 (RCC_SPI45CKSELR) . 678
10.8.71 RCC USART1 和 USART2 内核时钟选择寄存器 (RCC_UART12CKSELR) 679
10.8.72 RCC USART3 和 UART5 内核时钟选择寄存器 (RCC_UART35CKSELR) 680
10.8.73 RCC UART4 内核时钟选择寄存器 (RCC_UART4CKSELR) 681
10.8.74 RCC USART6 内核时钟选择寄存器 (RCC_UART6CKSELR) 682
10.8.75 RCC UART7 和 UART8 内核时钟选择寄存器 (RCC_UART78CKSELR) 683
10.8.76 RCC LPTIM1 内核时钟选择寄存器 (RCC_LPTIM1CKSELR) 684
10.8.77 RCC LPTIM2 和 LPTIM3 内核时钟选择寄存器 (RCC_LPTIM23CKSELR) 685
10.8.78 RCC LPTIM4 和 LPTIM5 内核时钟选择寄存器 (RCC_LPTIM45CKSELR) 686
10.8.79 RCC SAI1 内核时钟选择寄存器 (RCC_SAI1CKSELR) 687
10.8.80 RCC SAI2 内核时钟选择寄存器 (RCC_SAI2CKSELR) 688
10.8.81 RCC FDCAN 内核时钟选择寄存器 (RCC_FDCANCKSELR) 689
10.8.82 RCC SPDIFRX 内核时钟选择寄存器 (RCC_SPDIFCKSELR) 690
10.8.83 RCC ADC1 和 ADC2 内核时钟选择寄存器 (RCC_ADC12CKSELR) . . 691
10.8.84 RCC SDMMC1 和 SDMMC2 内核时钟选择寄存器 (RCC_SDMMC12CKSELR) 692
10.8.85 RCC ETH1 和 ETH2 内核时钟选择寄存器 (RCC_ETH12CKSELR) . . 693
10.8.86 RCC USB 内核时钟选择寄存器 (RCC_USBCKSELR) 695
10.8.87 RCC QSPI 内核时钟选择寄存器 (RCC_QSPICKSELR) 696
10.8.88 RCC FMC 内核时钟选择寄存器 (RCC_FMCCKSELR) 697
10.8.89 RCC RNG1 内核时钟选择寄存器 (RCC_RNG1CKSELR) 698

10.8.90 RCC STGEN 时钟选择寄存器 (RCC_STGENCKSELR) 699
10.8.91 RCC DCMIPP 内核时钟选择寄存器 (RCC_DCMIPPCKSELR) 700
10.8.92 RCC SAES 内核时钟选择寄存器 (RCC_SAESCKSELR) 701
10.8.93 RCC APB1 外设复位设置寄存器 (RCC_APB1RSTSETR) 701
10.8.94 RCC APB1 外设复位清除寄存器 (RCC_APB1RSTCLRR) 704
10.8.95 RCC APB2 外设复位设置寄存器 (RCC_APB2RSTSETR) 707
10.8.96 RCC APB2 外设复位清除寄存器 (RCC_APB2RSTCLRR) 708
10.8.97 RCC APB3 外设复位设置寄存器 (RCC_APB3RSTSETR) 710
10.8.98 RCC APB3 外设复位清除寄存器 (RCC_APB3RSTCLRR) 712
10.8.99 RCC APB4 外设复位设置寄存器 (RCC_APB4RSTSETR) 714
10.8.100 RCC APB4 外设复位清除寄存器 (RCC_APB4RSTCLRR) 715
10.8.101 RCC APB5 外设复位设置寄存器 (RCC_APB5RSTSETR) 717
10.8.102 RCC APB5 外设复位清除寄存器 (RCC_APB5RSTCLRR) 718
10.8.103 RCC APB6 外设复位设置寄存器 (RCC_APB6RSTSETR) 719
10.8.104 RCC APB6 外设复位清除寄存器 (RCC_APB6RSTCLRR) 722
10.8.105 RCC AHB2 外设复位设置寄存器 (RCC_AHB2RSTSETR) 725
10.8.106 RCC AHB2 外设复位清除寄存器 (RCC_AHB2RSTCLRR) 727
10.8.107 RCC AHB4 外设复位设置寄存器 (RCC_AHB4RSTSETR) 729
10.8.108 RCC AHB4 外设复位清除寄存器 (RCC_AHB4RSTCLRR) 731
10.8.109 RCC AHB5 外设复位设置寄存器 (RCC_AHB5RSTSETR) 733
10.8.110 RCC AHB5 外设复位清除寄存器 (RCC_AHB5RSTCLRR) 735
10.8.111 RCC AHB6 外设复位设置寄存器 (RCC_AHB6RSTSETR) 737
10.8.112 RCC AHB6 外设复位清除寄存器 (RCC_AHB6RSTCLRR) 739
10.8.113 RCC APB1 外设使能设置寄存器 (RCC_MP_APB1ENSETR) 741
10.8.114 RCC APB1 外设使能清零寄存器 (RCC_MP_APB1ENCLRR) 744
10.8.115 RCC APB2 外设使能设置寄存器 (RCC_MP_APB2ENSETR) 747
10.8.116 RCC APB2 外设使能清零寄存器 (RCC_MP_APB2ENCLRR) 749
10.8.117 RCC APB3 外设使能设置寄存器 (RCC_MP_APB3ENSETR) 751
10.8.118 RCC APB3 外设使能清零寄存器 (RCC_MP_APB3ENCLRR) 752
10.8.119 RCC 安全 APB3 外设使能设置寄存器 (RCC_MP_S_APB3ENSETR) 754
10.8.120 RCC 安全 APB3 外设使能清零寄存器 (RCC_MP_S_APB3ENCLRR) 755
10.8.121 RCC 非安全 APB3 外设使能设置寄存器 (RCC_MP_NS_APB3ENSETR) 756
10.8.122 RCC 非安全 APB3 外设使能清零寄存器 (RCC_MP_NS_APB3ENCLRR) 757
10.8.123 RCC APB4 外设使能设置寄存器 (RCC_MP_APB4ENSETR) 758

10.8.124 RCC APB4 外设使能清零寄存器 (RCC_MP_APB4ENCLRR) 760
10.8.125 RCC 安全 APB4 外设使能设置寄存器 (RCC_MP_S_APB4ENSETR) 762
10.8.126 RCC 安全 APB4 外设使能清零寄存器 (RCC_MP_S_APB4ENCLRR) 763
10.8.127 RCC 非安全 APB4 外设使能设置寄存器 (RCC_MP_NS_APB4ENSETR) 764
10.8.128 RCC 非安全 APB4 外设使能清零寄存器 (RCC_MP_NS_APB4ENCLRR) 765
10.8.129 RCC APB5 外设使能设置寄存器 (RCC_MP_APB5ENSETR) 766
10.8.130 RCC APB5 外设使能清零寄存器 (RCC_MP_APB5ENCLRR) 768
10.8.131 RCC APB6 外设使能设置寄存器 (RCC_MP_APB6ENSETR) 770
10.8.132 RCC APB6 外设使能清零寄存器 (RCC_MP_APB6ENCLRR) 773
10.8.133 RCC AHB2 外设使能设置寄存器 (RCC_MP_AHB2ENSETR) 776
10.8.134 RCC AHB2 外设使能清零寄存器 (RCC_MP_AHB2ENCLRR) 778
10.8.135 RCC 安全 AHB4 外设使能设置寄存器 (RCC_MP_S_AHB4ENSETR) 780
10.8.136 RCC 安全 AHB4 外设使能清零寄存器 (RCC_MP_S_AHB4ENCLRR) 782
10.8.137 RCC 非安全 AHB4 外设使能设置寄存器 (RCC_MP_NS_AHB4ENSETR) 784
10.8.138 RCC 非安全 AHB4 外设使能清零寄存器 (RCC_MP_NS_AHB4ENCLRR) 786
10.8.139 RCC AHB5 外设使能设置寄存器 (RCC_MP_AHB5ENSETR) 788
10.8.140 RCC AHB5 外设使能清零寄存器 (RCC_MP_AHB5ENCLRR) 790
10.8.141 RCC AHB6 外设使能设置寄存器 (RCC_MP_AHB6ENSETR) 792
10.8.142 RCC AHB6 外设使能清零寄存器 (RCC_MP_AHB6ENCLRR) 795
10.8.143 RCC 安全 AHB6 外设使能设置寄存器 (RCC_MP_S_AHB6ENSETR) 798
10.8.144 RCC 安全 AHB6 外设使能清零寄存器 (RCC_MP_S_AHB6ENCLRR) 799
10.8.145 RCC 非安全 AHB6 外设使能设置寄存器 (RCC_MP_NS_AHB6ENSETR) 800
10.8.146 RCC 非安全 AHB6 外设使能清零寄存器 (RCC_MP_NS_AHB6ENCLRR) 801
10.8.147 RCC APB1 睡眠时钟使能设置寄存器 (RCC_MP_APB1LPENSETR) 802
10.8.148 RCC APB1 睡眠时钟使能清零寄存器 (RCC_MP_APB1LPENCLRR) 805
10.8.149 RCC APB2 睡眠时钟使能设置寄存器 (RCC_MP_APB2LPENSETR) 808

10.8.150 RCC APB2 睡眠时钟使能清零寄存器 (RCC_MP_APB2LPENCLRR) 810
10.8.151 RCC APB3 睡眠时钟使能设置寄存器 (RCC_MP_APB3LPENSETR) 812
10.8.152 RCC APB3 睡眠时钟使能清零寄存器 (RCC_MP_APB3LPENCLRR) 813
10.8.153 RCC 安全 APB3 睡眠时钟使能设置寄存器 (RCC_MP_S_APB3LPENSETR) 815
10.8.154 RCC 安全 APB3 睡眠时钟使能清零寄存器 (RCC_MP_S_APB3LPENCLRR) 816
10.8.155 RCC 非安全 APB3 睡眠时钟使能设置寄存器 (RCC_MP_NS_APB3LPENSETR) 817
10.8.156 RCC 非安全 APB3 睡眠时钟使能清零寄存器 (RCC_MP_NS_APB3LPENCLRR) 818
10.8.157 RCC APB4 睡眠时钟使能设置寄存器 (RCC_MP_APB4LPENSETR) 819
10.8.158 RCC APB4 睡眠时钟使能清零寄存器 (RCC_MP_APB4LPENCLRR) 821
10.8.159 RCC 安全 APB4 睡眠外设使能设置寄存器 (RCC_MP_S_APB4LPENSETR) 823
10.8.160 RCC 安全 APB4 睡眠外设使能清零寄存器 (RCC_MP_S_APB4LPENCLRR) 824
10.8.161 RCC 非安全 APB4 睡眠外设使能设置寄存器 (RCC_MP_NS_APB4LPENSETR) 825
10.8.162 RCC 非安全 APB4 睡眠外设使能清零寄存器 (RCC_MP_NS_APB4LPENCLRR) 826
10.8.163 RCC APB5 睡眠时钟使能设置寄存器 (RCC_MP_APB5LPENSETR) 827
10.8.164 RCC APB5 睡眠时钟使能清零寄存器 (RCC_MP_APB5LPENCLRR) 829
10.8.165 RCC APB6 睡眠时钟使能设置寄存器 (RCC_MP_APB6LPENSETR) 831
10.8.166 RCC APB6 睡眠时钟使能清零寄存器 (RCC_MP_APB6LPENCLRR) 834
10.8.167 RCC AHB2 睡眠时钟使能设置寄存器 (RCC_MP_AHB2LPENSETR) 837
10.8.168 RCC AHB2 睡眠时钟使能清零寄存器 (RCC_MP_AHB2LPENCLRR) 839
10.8.169 RCC 安全 AHB4 睡眠时钟使能设置寄存器 (RCC_MP_S_AHB4LPENSETR) 841
10.8.170 RCC 安全 AHB4 睡眠时钟使能清零寄存器 (RCC_MP_S_AHB4LPENCLRR) 843

10.8.171 RCC 非安全 AHB4 睡眠时钟使能设置寄存器 (RCC_MP_NS_AHB4LPENSETR) 845
10.8.172 RCC 非安全 AHB4 睡眠时钟使能清零寄存器 (RCC_MP_NS_AHB4LPENCLRR) 847
10.8.173 RCC AHB5 睡眠时钟使能设置寄存器 (RCC_MP_AHB5LPENSETR) 849
10.8.174 RCC AHB5 睡眠时钟使能清零寄存器 (RCC_MP_AHB5LPENCLRR) 851
10.8.175 RCC AHB6 睡眠时钟使能设置寄存器 (RCC_MP_AHB6LPENSETR) 853
10.8.176 RCC AHB6 睡眠时钟使能清零寄存器 (RCC_MP_AHB6LPENCLRR) 857
10.8.177 RCC 安全 AHB6 睡眠时钟使能设置寄存器 (RCC_MP_S_AHB6LPENSETR) 861
10.8.178 RCC 安全 AHB6 睡眠时钟使能清零寄存器 (RCC_MP_S_AHB6LPENCLRR) 862
10.8.179 RCC 非安全 AHB6 睡眠时钟使能设置寄存器 (RCC_MP_NS_AHB6LPENSETR) 863
10.8.180 RCC 非安全 AHB6 睡眠时钟使能清零寄存器 (RCC_MP_NS_AHB6LPENCLRR) 864
10.8.181 RCC 安全 AXI 睡眠时钟使能设置寄存器 (RCC_MP_S_AXIMLPENSETR) 865
10.8.182 RCC 安全 AXI 睡眠时钟使能清零寄存器 (RCC_MP_S_AXIMLPENCLRR) 866
10.8.183 RCC 非安全 AXI 睡眠时钟使能设置寄存器 (RCC_MP_NS_AXIMLPENSETR) 867
10.8.184 RCC 非安全 AXI 睡眠时钟使能清零寄存器 (RCC_MP_NS_AXIMLPENCLRR) 868
10.8.185 RCC MLAHB 睡眠时钟使能设置寄存器 (RCC_MP_MLAHBLPENSETR) 869
10.8.186 RCC MLAHB 睡眠时钟使能清零寄存器 (RCC_MP_MLAHBLPENCLRR) 870
10.8.187 RCC APB3 外设安全状态寄存器 (RCC_APB3SECSR) 871
10.8.188 RCC APB4 外设安全状态寄存器 (RCC_APB4SECSR) 871
10.8.189 RCC APB5 外设安全状态寄存器 (RCC_APB5SECSR) 872
10.8.190 RCC APB6 外设安全状态寄存器 (RCC_APB6SECSR) 874
10.8.191 RCC AHB2 外设安全状态寄存器 (RCC_AHB2SECSR) 876
10.8.192 RCC AHB5 外设安全状态寄存器 (RCC_AHB5SECSR) 877
10.8.193 RCC AHB6 外设安全状态寄存器 (RCC_AHB6SECSR) 879
10.8.194 RCC 版本寄存器 (RCC_VERR) 880
10.8.195 RCC ID 寄存器 (RCC_IDR) 881
10.8.196 RCC 大小 ID 寄存器 (RCC_SIDR) 881

10.9 RCC 寄存器映射 882

11 通用 I/O (GPIO) 909

11.1 简介 909
11.2 GPIO 主要特性 909
11.3 GPIO 功能描述 909
11.3.1 通用 I/O (GPIO) 911
11.3.2 I/O 引脚复用功能复用器和映射 912
11.3.3 I/O 端口控制寄存器 912
11.3.4 I/O 端口数据寄存器 912
11.3.5 I/O 数据位操作 913
11.3.6 GPIO 锁定机制 913
11.3.7 I/O 复用功能输入/输出 913
11.3.8 外部中断线/唤醒线 913
11.3.9 输入配置 914
11.3.10 输出配置 914
11.3.11 I/O 补偿单元 915
11.3.12 高速低电压模式 915
11.3.13 复用功能配置 916
11.3.14 模拟配置 916
11.3.15 将 HSE 或 LSE 振荡器引脚用作 GPIO 917
11.3.16 在备份电源域中使用 GPIO 引脚 917
11.3.17 BOOT/GPIO 引脚共享 917
11.3.18 TrustZone 安全性 918
11.4 GPIO寄存器 919
11.4.1 GPIO 端口模式寄存器 (GPIOx_MODER)（x = A 到 I） 919
11.4.2 GPIO 端口输出类型寄存器 (GPIOx_OTYPER)（x = A 到 I） 919
11.4.3 GPIO 端口输出速度寄存器 (GPIOx_OSPEEDR)（x = A 到 I） 920
11.4.4 GPIO 端口上拉/下拉寄存器 (GPIOx_PUPDR)（x = A 到 I） 920
11.4.5 GPIO 端口输入数据寄存器 (GPIOx_IDR)（x = A 到 I） 921
11.4.6 GPIO 端口输出数据寄存器 (GPIOx_ODR)（x = A 到 I） 921
11.4.7 GPIO 端口位置 1/复位寄存器 (GPIOx_BSRR)（x = A 到 I） 922
11.4.8 GPIO 端口配置锁定寄存器 (GPIOx_LCKR)（x = A 到 I） 922
11.4.9 GPIO 复用功能低位寄存器 (GPIOx_AFRL)（x = A 到 I） 923
11.4.10 GPIO 复用功能高位寄存器 (GPIOx_AFRH)（x = A 到 I） 924
11.4.11 GPIO 端口位复位寄存器 (GPIOx_BRR)（x = A 到 I） 925
11.4.12 GPIO 安全配置寄存器 (GPIOx_SECCFGR)（x = A 到 I） 925

11.4.13 GPIO 硬件配置寄存器 10 (GPIOx_HWCFGR10)（x = A 到 I） 926
11.4.14 GPIO 硬件配置寄存器 9 (GPIOx_HWCFGR9)（x = A 到 I） 926
11.4.15 GPIO 硬件配置寄存器 8 (GPIOx_HWCFGR8)（x = A 到 I） 927
11.4.16 GPIO 硬件配置寄存器 7 (GPIOx_HWCFGR7)（x = A 到 I） 928
11.4.17 GPIO 硬件配置寄存器 6 (GPIOx_HWCFGR6)（x = A 到 I） 928
11.4.18 GPIO 硬件配置寄存器 5 (GPIOx_HWCFGR5)（x = A 到 I） 929
11.4.19 GPIO 硬件配置寄存器 4 (GPIOx_HWCFGR4)（x = A 到 I） 929
11.4.20 GPIO 硬件配置寄存器 3 (GPIOx_HWCFGR3)（x = A 到 I） 930
11.4.21 GPIO 硬件配置寄存器 2 (GPIOx_HWCFGR2)（x = A 到 I） 930
11.4.22 GPIO 硬件配置寄存器 1 (GPIOx_HWCFGR1)（x = A 到 I） 930
11.4.23 GPIO 硬件配置寄存器 0 (GPIOx_HWCFGR0)（x = A 到 I） 931
11.4.24 GPIO 版本寄存器 (GPIOx_VERR)（x = A 到 I） 931
11.4.25 GPIO 标识寄存器 (GPIOx_IPIDR)（x = A 到 I） 932
11.4.26 GPIO 大小标识寄存器 (GPIOx_SIDR)（x = A 到 I） 932
11.4.27 GPIO 寄存器映射 933

12 系统配置控制器 (SYSCFG) 936

12.1 TrustZone 安全性 936
12.2 I/O 补偿单元 936
12.3 以太网 PHY 类型和时钟源选择 937
12.4 模拟开关电源 937
12.5 高速低压 (HSLV) 输出缓冲器 938
12.6 SYSCFG 寄存器 939
12.6.1 SYSCFG 启动引脚控制寄存器 (SYSCFG_BOOTR) 939
12.6.2 SYSCFG 外设模式配置设置寄存器 (SYSCFG_PMCSETR) 940
12.6.3 SYSCFG 外设模式配置清除寄存器 (SYSCFG_PMCCLRR) 942
12.6.4 SYSCFG 启动控制寄存器 (SYSCFG_BOOTCR) 945
12.6.5 SYSCFG SRAM3 擦除寄存器 (SYSCFG_SRAM3ERASER) 945
12.6.6 SYSCFG SRAM3 擦除密钥寄存器 (SYSCFG_SRAM3KR) 946
12.6.7 SYSCFG 主补偿单元控制寄存器 (SYSCFG_CMPCR) 947
12.6.8 SYSCFG 主补偿单元使能设置寄存器 (SYSCFG_CMPENSETR) 948
12.6.9 SYSCFG 主补偿单元使能清除寄存器 (SYSCFG_CMPENCLRR) 949
12.6.10 SYSCFG 控制定时器刹车寄存器 (SYSCFG_CBR) 949
12.6.11 SYSCFG VDDSD1 补偿单元控制寄存器 (SYSCFG_CMPSD1CR) . . . 950
12.6.12 SYSCFG VDDSD1 补偿单元使能设置寄存器 (SYSCFG_CMPSD1ENSETR) 951

12.6.13 SYSCFG VDDSD1 补偿单元使能清除寄存器 (SYSCFG_CMPSD1ENCLRR) 952
12.6.14 SYSCFG VDDSD2 补偿单元控制寄存器 (SYSCFG_CMPSD2CR) . . . 952
12.6.15 SYSCFG VDDSD2 补偿单元使能设置寄存器 (SYSCFG_CMPSD2ENSETR) 954
12.6.16 SYSCFG VDDSD2 补偿单元使能清除寄存器 (SYSCFG_CMPSD2ENCLRR) 955
12.6.17 SYSCFG 高速低压使能寄存器 x (SYSCFG_HSLVENxR) 956
12.6.18 SYSCFG 标识代码寄存器 (SYSCFG_IDC) 956
12.6.19 SYSCFG 版本寄存器 (SYSCFG_VERR) 957
12.6.20 SYSCFG 标识寄存器 (SYSCFG_IPIDR) 957
12.6.21 SYSCFG 大小标识寄存器 (SYSCFG_SIDR) 958
12.6.22 SYSCFG 接口寄存器映射 959

13 扩展 TrustZone® 保护控制器 (ETZPC) 963
13.1 ETZPC 简介 963
13.2 ETZPC 特性 963
13.3 扩展 TrustZone 架构和 ETZPC 963
13.4 ETZPC 功能描述 964
13.5 STM32MP13xx 安全架构 964
13.6 ETZPC 寄存器 969
13.6.1 ETZPC ROM 安全大小定义 (ETZPC_TZMA0_SIZE) 970
13.6.2 ETZPC RAM 安全大小定义 (ETZPC_TZMA1_SIZE) 971
13.6.3 ETZPC 受控安全外设控制寄存器 x (ETZPC_DECPROTx) 972
13.6.4 ETZPC decprot 锁定 x 寄存器 (ETZPC_DECPROT_LOCKx) 973
13.6.5 ETZPC IP HW 配置寄存器 (ETZPC_HWCFGR) 976
13.6.6 ETZPC IP 版本寄存器 (ETZPC_VERR) 976
13.6.7 ETZPC IP 版本寄存器 (ETZPC_IDR) 977
13.6.8 ETZPC IP 版本寄存器 (ETZPC_SIDR) 977
13.6.9 ETZPC 寄存器映射 978

14 外设互联矩阵 980
14.1 外设互连 980
14.1.1 简介 980
14.1.2 连接概述 980
14.2 MDMA 992

15 MDMA 控制器 (MDMA) 994
15.1 MDMA 简介 994
15.2 MDMA 主要特性 994
15.3 MDMA 功能描述 995
15.3.1 MDMA 框图 995
15.3.2 MDMA 内部信号 995
15.3.3 MDMA 概述 996
15.3.4 MDMA通道 997
15.3.5 源、目标和传输模式 997
15.3.6 指针更新 997
15.3.7 MDMA 缓冲区传输 998
15.3.8 请求仲裁 998
15.3.9 FIFO 999
15.3.10 块传输 999
15.3.11 块重复模式 999
15.3.12 链表模式 999
15.3.13 MDMA 传输完成 1000
15.3.14 MDMA 传输暂停 1000
15.3.15 错误管理 1000
15.4 MDMA 中断 1001
15.5 MDMA 寄存器 1002
15.5.1 MDMA 全局中断状态寄存器 (MDMA_GISR0) 1002
15.5.2 MDMA 安全全局中断状态寄存器 (MDMA_SGISR0) 1002
15.5.3 MDMA 通道 x 中断状态寄存器 (MDMA_CxISR) 1003
15.5.4 MDMA 通道 x 中断标志清零寄存器 (MDMA_CxIFCR) 1004
15.5.5 MDMA 通道 x 错误状态寄存器 (MDMA_CxESR) 1005
15.5.6 MDMA 通道 x 控制寄存器 (MDMA_CxCR) 1006
15.5.7 MDMA 通道 x 传输配置寄存器 (MDMA_CxTCR) 1008
15.5.8 MDMA 通道 x 块数据数寄存器 (MDMA_CxBNDTR) 1011
15.5.9 MDMA 通道 x 源地址寄存器 (MDMA_CxSAR) 1012
15.5.10 MDMA 通道 x 目标地址寄存器 (MDMA_CxDAR) 1013
15.5.11 MDMA 通道 x 块重复地址更新寄存器 (MDMA_CxBRUR) 1013
15.5.12 MDMA 通道 x 链路地址寄存器 (MDMA_CxLAR) 1014
15.5.13 MDMA 通道 x 触发和总线选择寄存器 (MDMA_CxTBR) 1015
15.5.14 MDMA 通道 x 掩码地址寄存器 (MDMA_CxMAR) 1015
15.5.15 MDMA 通道 x 掩码数据寄存器 (MDMA_CxMDR) 1016

15.5.16 MDMA 寄存器映射 1016

16 直接内存访问控制器 (DMA) 1018

16.1 DMA 简介 1018
16.2 DMA 主要特性 1018
16.3 DMA 功能描述 1019
16.3.1 DMA 框图 1019
16.3.2 DMA 概述 1020
16.3.3 DMA 事务 1020
16.3.4 DMA 请求映射 1020
16.3.5 仲裁器 1020
16.3.6 DMA 数据流 1021
16.3.7 源、目标和传输模式 1021
16.3.8 指针递增 1024
16.3.9 循环模式 (Circular mode) 1025
16.3.10 双缓冲区模式 1025
16.3.11 可编程数据宽度、封装/解封、字节序 1026
16.3.12 单次传输和突发传输 1028
16.3.13 FIFO 1028
16.3.14 DMA 传输完成 1031
16.3.15 DMA 传输暂停 1032
16.3.16 流控制器 1033
16.3.17 可能的 DMA 配置汇总 1034
16.3.18 流配置过程 1034
16.3.19 错误管理 1035
16.4 DMA 中断 1036
16.5 DMA 寄存器 1037
16.5.1 DMA 低中断状态寄存器 (DMA_LISR) 1037
16.5.2 DMA 高中断状态寄存器 (DMA_HISR) 1038
16.5.3 DMA 低中断标志清零寄存器 (DMA_LIFCR) 1039
16.5.4 DMA 高中断标志清零寄存器 (DMA_HIFCR) 1039
16.5.5 DMA 数据流 x 配置寄存器 (DMA_SxCR) 1040
16.5.6 DMA 数据流 x 数据项数寄存器 (DMA_SxNDTR) 1043
16.5.7 DMA 数据流 x 外设地址寄存器 (DMA_SxPAR) 1044
16.5.8 DMA 数据流 x 内存 0 地址寄存器 (DMA_SxM0AR) 1044
16.5.9 DMA 数据流 x 内存 1 地址寄存器 (DMA_SxM1AR) 1045
16.5.10 DMA 数据流 x FIFO 控制寄存器 (DMA_SxFCR) 1045

16.5.11 DMA 硬件配置 2 寄存器 (DMA_HWCFGR2) 1047
16.5.12 DMA 硬件配置 1 寄存器 (DMA_HWCFGR1) 1048
16.5.13 DMA 版本寄存器 (DMA_VERR) 1049
16.5.14 DMA 标识寄存器 (DMA_IPDR) 1050
16.5.15 DMA 大小标识寄存器 (DMA_SIDR) 1050
16.5.16 DMA 寄存器映射 1050

17 DMA 请求复用器 (DMAMUX) 1055
17.1 简介 1055
17.2 DMAMUX 主要特性 1055
17.3 DMAMUX 实现 1056
17.3.1 DMAMUX 实例化 1056
17.3.2 DMAMUX1 映射 1056
17.3.3 DMAMUX2 映射 1058
17.4 DMAMUX 功能描述 1060
17.4.1 DMAMUX 框图 1060
17.4.2 DMAMUX 信号 1061
17.4.3 DMAMUX 通道 1061
17.4.4 DMAMUX 请求线复用器 1061
17.4.5 DMAMUX 请求发生器 1064
17.5 DMAMUX 中断 1065
17.6 DMAMUX 寄存器 1066
17.6.1 DMAMUX 请求线复用器通道 x 配置寄存器 (DMAMUX_CxCR) 1066
17.6.2 DMAMUX 请求线复用器通道 x 配置寄存器 [备用] (DMAMUX_CxCR) 1067
17.6.3 DMAMUX 请求线复用器中断通道状态寄存器 (DMAMUX_CSR) 1068
17.6.4 DMAMUX 请求线复用器中断清除标志寄存器 (DMAMUX_CFR) 1069
17.6.5 DMAMUX 请求发生器通道 x 配置寄存器 (DMAMUX_RGxCR) 1069
17.6.6 DMAMUX 请求发生器中断状态寄存器 (DMAMUX_RGSR) 1070
17.6.7 DMAMUX 请求发生器中断清除标志寄存器 (DMAMUX_RGCFR) . . . 1070
17.6.8 DMAMUX 大小标识寄存器 (DMAMUX_SIDR) 1071
17.6.9 DMAMUX IP 标识寄存器 (DMAMUX_IPIDR) 1071
17.6.10 DMAMUX 版本寄存器 (DMAMUX_VERR) 1072
17.6.11 DMAMUX 硬件配置 1 寄存器 (DMAMUX_HWCFGR1) 1072
17.6.12 DMAMUX 硬件配置 2 寄存器 (DMAMUX_HWCFGR2) 1073
17.6.13 DMAMUX 寄存器映射 1073

18 中断列表 1076
18.1 GIC 中断 1076
18.2 EXTI 事件 1081

19 全局中断控制器 (GIC) 1085
19.1 简介 1085
19.2 GIC 主要特性 1085
19.3 GIC 功能描述 1085
19.4 中断源 1086
19.4.1 软件生成中断 (SGI) 1086
19.4.2 私有外设中断 (PPI) 1087
19.4.3 共享外设中断 (SPI) 1087
19.4.4 中断优先级格式 1088
19.5 GIC 分配器 (GICD) 1088
19.5.1 GICD 控制寄存器 (GICD_CTLR) 1088
19.5.2 GICD 控制非安全访问寄存器 (GICD_CTLRNS) 1089
19.5.3 GICD 中断控制器类型寄存器 (GICD_TYPER) 1089
19.5.4 GICD 实现器标识寄存器 (GICD_IIDR) 1090
19.5.5 GICD 中断组寄存器 x (GICD_IGROUPRx) 1090
19.5.6 GICD 中断设置使能寄存器 (GICD_ISENABLER0) 1090
19.5.7 GICD 中断设置使能寄存器 x (GICD_ISENABLERx) 1091
19.5.8 GICD 中断清除使能寄存器 (GICD_ICENABLER0) 1091
19.5.9 GICD 中断清除使能寄存器 x (GICD_ICENABLERx) 1092
19.5.10 GICD 中断设置挂起寄存器 x (GICD_ISPENDRx) 1092
19.5.11 GICD 中断清除挂起寄存器 x (GICD_ICPENDRx) 1092
19.5.12 GICD 中断设置有效寄存器 x (GICD_ISACTIVERx) 1093
19.5.13 GICD 中断清除有效寄存器 x (GICD_ICACTIVERx) 1093
19.5.14 GICD 中断优先级寄存器 x (GICD_IPRIORITYRx) 1094
19.5.15 GICD 中断处理器目标寄存器 x (GICD_ITARGETSRx) 1095
19.5.16 GICD 中断处理器目标寄存器 x (GICD_ITARGETSRx) 1096
19.5.17 GICD 中断配置寄存器 (GICD_ICFGR0) 1097
19.5.18 GICD 中断配置寄存器 (GICD_ICFGR1) 1098
19.5.19 GICD 中断配置寄存器 x (GICD_ICFGRx) 1099
19.5.20 GICD 私有外设中断状态寄存器 (GICD_PPISR) 1100
19.5.21 GICD 共享外设中断寄存器 x (GICD_SPISRx) 1100
19.5.22 GICD 软件生成中断寄存器 (GICD_SGIR) 1101

19.5.23 GICD SGI 清除挂起寄存器 x (GICD_CPENDSGIRx) 1102
19.5.24 GICD SGI 设置挂起寄存器 x (GICD_SPENDSGIRx) 1103
19.5.25 GICD 外设 ID4 寄存器 (GICD_PIDR4) 1104
19.5.26 GICD 外设 ID5 到 ID7 寄存器 x (GICD_PIDRx) 1104
19.5.27 GICD 外设 ID0 寄存器 (GICD_PIDR0) 1105
19.5.28 GICD 外设 ID1 寄存器 (GICD_PIDR1) 1105
19.5.29 GICD 外设 ID2 寄存器 (GICD_PIDR2) 1105
19.5.30 GICD 外设 ID3 寄存器 (GICD_PIDR3) 1106
19.5.31 GICD 组件 ID0 寄存器 (GICD_CIDR0) 1106
19.5.32 GICD 组件 ID1 寄存器 (GICD_CIDR1) 1106
19.5.33 GICD 组件 ID2 寄存器 (GICD_CIDR2) 1107
19.5.34 GICD 组件 ID3 寄存器 (GICD_CIDR3) 1107
19.5.35 GICD 寄存器映射 1107
19.6 GIC CPU 接口 (GICC) 1111
19.6.1 GICC 控制寄存器 (GICC_CTLR) 1111
19.6.2 GICC 控制非安全访问寄存器 (GICC_CTLRNS) 1112
19.6.3 GICC 输入优先级屏蔽寄存器 (GICC_PMR) 1113
19.6.4 GICC 二进制点寄存器 (GICC_BPR) 1113
19.6.5 GICC 二进制点非安全访问寄存器 (GICC_BPRNS) 1113
19.6.6 GICC 中断确认寄存器 (GICC_IAR) 1114
19.6.7 GICC 中断结束寄存器 (GICC_EOIR) 1114
19.6.8 GICC 运行优先级寄存器 (GICC_RPR) 1115
19.6.9 GICC 最高优先级挂起中断寄存器 (GICC_HPPIR) 1115
19.6.10 GICC 别名二进制点寄存器 (GICC_ABPR) 1116
19.6.11 GICC 别名中断确认寄存器 (GICC_AIAR) 1116
19.6.12 GICC 别名中断结束寄存器 (GICC_AEOIR) 1117
19.6.13 GICC 别名最高优先级挂起中断寄存器 (GICC_AHPPIR) 1117
19.6.14 GICC 有效优先级寄存器 (GICC_APR0) 1118
19.6.15 GICC 非安全有效优先级寄存器 (GICC_NSAPR0) 1118
19.6.16 GICC 接口标识寄存器 (GICC_IIDR) 1118
19.6.17 GICC 禁用中断寄存器 (GICC_DIR) 1119
19.6.18 GICC 寄存器映射 1119
19.7 GIC 虚拟接口控制，通用 (GICH) 1121
19.7.1 GICH 管理程序控制寄存器 (GICH_HCR) 1122
19.7.2 GICH VGIC 类型寄存器 (GICH_VTR) 1123
19.7.3 GICH 虚拟机控制寄存器 (GICH_VMCR) 1123
19.7.4 GICH 维护中断状态寄存器 (GICH_MISR) 1124

19.7.5 GICH 中断结束状态寄存器 (GICH_EISR0) 1125
19.7.6 GICH 空列表状态寄存器 (GICH_ELSR0) 1125
19.7.7 GICH 有效优先级寄存器 (GICH_APR0) 1125
19.7.8 GICH 列表寄存器 x (GICH_LRx) 1126
19.7.9 GICH 寄存器映射 1126
19.8 GIC 虚拟 CPU 接口 (GICV) 1127
19.8.1 GICV 虚拟机控制寄存器 (GICV_CTLR) 1128
19.8.2 GICV VM 优先级屏蔽寄存器 (GICV_PMR) 1128
19.8.3 GICV VM 二进制点寄存器 (GICV_BPR) 1129
19.8.4 GICV VM 中断确认寄存器 (GICV_IAR) 1129
19.8.5 GICV VM 中断结束寄存器 (GICV_EOIR) 1130
19.8.6 GICV VM 运行优先级寄存器 (GICV_RPR) 1130
19.8.7 GICV VM 最高优先级挂起中断寄存器 (GICV_HPPIR) 1131
19.8.8 GICV VM 别名二进制点寄存器 (GICV_ABPR) 1131
19.8.9 GICV VM 别名中断寄存器 (GICV_AIAR) 1132
19.8.10 GICV VM 别名中断结束寄存器 (GICV_AEOIR) 1132
19.8.11 GICV VM 别名最高优先级挂起中断寄存器 (GICV_AHPPIR) 1133
19.8.12 GICV VM 有效优先级寄存器 (GICV_APR0) 1133
19.8.13 GICV VM CPU 接口标识寄存器 (GICV_IIDR) 1134
19.8.14 GICV VM 禁用中断寄存器 (GICV_DIR) 1134
19.8.15 GICV 寄存器映射 1134

20 扩展中断和事件控制器 (EXTI) 1137
20.1 EXTI 主要特性 1137
20.2 EXTI 框图 1138
20.2.1 外设和 CPU 之间的 EXTI 连接 1139
20.3 EXTI 功能描述 1140
20.3.1 EXTI 可配置事件输入唤醒 1141
20.3.2 EXTI 直接事件输入唤醒 1142
20.3.3 EXTI 复用器选择 1142
20.4 EXTI 功能行为 1143
20.5 EXTI TrustZone 安全性 1144
20.6 EXTI 寄存器 1145
20.6.1 EXTI 上升沿触发选择寄存器 (EXTI_RTSR1) 1145
20.6.2 EXTI 下降沿触发选择寄存器 (EXTI_FTSR1) 1146
20.6.3 EXTI 软件中断事件寄存器 (EXTI_SWIER1) 1146

20.6.4 EXTI 上升沿挂起寄存器 (EXTI_RPR1) 1147
20.6.5 EXTI 下降沿挂起寄存器 (EXTI_FPR1) 1147
20.6.6 EXTI TrustZone 使能寄存器 (EXTI_TZENR1) 1148
20.6.7 EXTI 上升沿触发选择寄存器 (EXTI_RTSR2) 1149
20.6.8 EXTI 下降沿触发选择寄存器 (EXTI_FTSR2) 1149
20.6.9 EXTI 软件中断事件寄存器 (EXTI_SWIER2) 1149
20.6.10 EXTI 上升沿挂起寄存器 (EXTI_RPR2) 1150
20.6.11 EXTI 下降沿挂起寄存器 (EXTI_FPR2) 1150
20.6.12 EXTI TrustZone 使能寄存器 (EXTI_TZENR2) 1151
20.6.13 EXTI 上升沿触发选择寄存器 (EXTI_RTSR3) 1152
20.6.14 EXTI 下降沿触发选择寄存器 (EXTI_FTSR3) 1152
20.6.15 EXTI 软件中断事件寄存器 (EXTI_SWIER3) 1153
20.6.16 EXTI 上升沿挂起寄存器 (EXTI_RPR3) 1153
20.6.17 EXTI 下降沿挂起寄存器 (EXTI_FPR3) 1153
20.6.18 EXTI TrustZone 使能寄存器 (EXTI_TZENR3) 1154
20.6.19 EXTI 外部中断选择寄存器 1 (EXTI_EXTICR1) 1154
20.6.20 EXTI 外部中断选择寄存器 2 (EXTI_EXTICR2) 1156
20.6.21 EXTI 外部中断选择寄存器 3 (EXTI_EXTICR3) 1158
20.6.22 EXTI 外部中断选择寄存器 4 (EXTI_EXTICR4) 1159
20.6.23 EXTI CPU 唤醒中断屏蔽寄存器 (EXTI_IMR1) 1161
20.6.24 EXTI CPU 唤醒事件屏蔽寄存器 (EXTI_EMR1) 1162
20.6.25 EXTI CPU 唤醒中断屏蔽寄存器 (EXTI_IMR2) 1162
20.6.26 EXTI CPU 唤醒事件屏蔽寄存器 (EXTI_EMR2) 1163
20.6.27 EXTI CPU 唤醒中断屏蔽寄存器 (EXTI_IMR3) 1163
20.6.28 EXTI CPU 唤醒事件屏蔽寄存器 (EXTI_EMR3) 1164
20.6.29 EXTI 硬件配置寄存器 x (EXTI_HWCFGRx) 1164
20.6.30 EXTI 硬件配置寄存器 x (EXTI_HWCFGRx) 1165
20.6.31 EXTI 硬件配置寄存器 x (EXTI_HWCFGRx) 1165
20.6.32 EXTI 硬件配置寄存器 x (EXTI_HWCFGRx) 1166
20.6.33 EXTI 硬件配置寄存器 1 (EXTI_HWCFGR1) 1166
20.6.34 EXTI IP 版本寄存器 (EXTI_VERR) 1167
20.6.35 EXTI 标识寄存器 (EXTI_IPIDR) 1167
20.6.36 EXTI 大小 ID 寄存器 (EXTI_SIDR) 1167
20.6.37 EXTI 寄存器映射 1168

21 循环冗余校验计算单元 (CRC) 1172

21.1 简介 1172

21.2 CRC 主要特性 1172
21.3 CRC 功能描述 1173
21.3.1 CRC 框图 1173
21.3.2 CRC 内部信号 1173
21.3.3 CRC 运算 1173
21.4 CRC 寄存器 1175
21.4.1 CRC 数据寄存器 (CRC_DR) 1175
21.4.2 CRC 独立数据寄存器 (CRC_IDR) 1175
21.4.3 CRC 控制寄存器 (CRC_CR) 1176
21.4.4 CRC 初始值 (CRC_INIT) 1177
21.4.5 CRC 多项式 (CRC_POL) 1177
21.4.6 CRC 硬件配置寄存器 (CRC_HWCFGR) 1178
21.4.7 CRC 版本寄存器 (CRC_VERR) 1178
21.4.8 CRC 标识寄存器 (CRC_IPIDR) 1179
21.4.9 CRC 大小标识寄存器 (CRC_SIDR) 1179
21.4.10 CRC 寄存器映射 1180

22 可变存储控制器 (FMC) 1181
22.1 FMC 主要特性 1181
22.2 FMC 实现 1182
22.3 FMC 框图 1182
22.4 FMC 内部信号 1183
22.5 AHB 接口 1183
22.6 AXI 接口 1184
22.6.1 支持的内存和事务 1184
22.7 外部设备地址映射 1185
22.7.1 NOR/PSRAM 地址映射 1185
22.7.2 NAND Flash 地址映射 1186
22.8 NOR Flash/PSRAM 控制器 1187
22.8.1 外部内存接口信号 1188
22.8.2 支持的内存和事务 1190
22.8.3 通用时序规则 1192
22.8.4 NOR Flash/PSRAM 控制器异步事务 1192
22.8.5 同步事务 1211
22.8.6 NOR/PSRAM 控制寄存器 1217
22.9 NAND Flash 控制器 1227

22.9.1 外部内存接口信号 1228
22.9.2 NAND Flash 支持的内存和事务 1229
22.9.3 NAND Flash 的时序图 1230
22.9.4 NAND Flash 操作 1231
22.9.5 NAND Flash 预等待功能 1232
22.9.6 NAND ECC 控制器 1233
22.9.7 FMC 命令定序器 1235
22.9.8 NAND Flash 控制器中断 1240
22.9.9 NAND Flash 控制器寄存器 1241
22.10 FMC 寄存器映射 1265

23 四线 SPI 接口 (QUADSPI) 1269
23.1 简介 1269
23.2 QUADSPI 主要特性 1269
23.3 QUADSPI 功能描述 1269
23.3.1 QUADSPI 框图 1269
23.3.2 QUADSPI 引脚和内部信号 1270
23.3.3 QUADSPI 命令序列 1271
23.3.4 QUADSPI 信号接口协议模式 1273
23.3.5 QUADSPI 间接模式 1275
23.3.6 QUADSPI 自动状态轮询模式 1276
23.3.7 QUADSPI 内存映射模式 1276
23.3.8 QUADSPI 自由运行时钟模式 1277
23.3.9 QUADSPI Flash 配置 1277
23.3.10 QUADSPI 延迟数据采样 1277
23.3.11 QUADSPI 配置 1278
23.3.12 QUADSPI 使用 1278
23.3.13 指令仅发送一次 1280
23.3.14 QUADSPI 错误管理 1280
23.3.15 QUADSPI 的 BUSY 位和中止功能 1280
23.3.16 NCS 行为 1281
23.4 QUADSPI 中断 1282
23.5 QUADSPI 寄存器 1283
23.5.1 QUADSPI 控制寄存器 (QUADSPI_CR) 1283
23.5.2 QUADSPI 设备配置寄存器 (QUADSPI_DCR) 1286
23.5.3 QUADSPI 状态寄存器 (QUADSPI_SR) 1287

23.5.4 QUADSPI 标志清零寄存器 (QUADSPI_FCR) 1288
23.5.5 QUADSPI 数据长度寄存器 (QUADSPI_DLR) 1288
23.5.6 QUADSPI 通信配置寄存器 (QUADSPI_CCR) 1289
23.5.7 QUADSPI 地址寄存器 (QUADSPI_AR) 1291
23.5.8 QUADSPI 交替字节寄存器 (QUADSPI_ABR) 1292
23.5.9 QUADSPI 数据寄存器 (QUADSPI_DR) 1292
23.5.10 QUADSPI 轮询状态屏蔽寄存器 (QUADSPI _PSMKR) 1293
23.5.11 QUADSPI 轮询状态匹配寄存器 (QUADSPI _PSMAR) 1293
23.5.12 QUADSPI 轮询间隔寄存器 (QUADSPI_PIR) 1294
23.5.13 QUADSPI 低功耗超时寄存器 (QUADSPI_LPTR) 1294
23.5.14 QUADSPI 硬件配置寄存器 (QUADSPI_HWCFGR) 1295
23.5.15 QUADSPI 版本寄存器 (QUADSPI_VERR) 1295
23.5.16 QUADSPI 标识寄存器 (QUADSPI_IPIDR) 1296
23.5.17 QUADSPI 大小标识寄存器 (QUADSPI_SIDR) 1296
23.5.18 QUADSPI 寄存器映射 1296

24 安全数字输入/输出多媒体卡接口 (SDMMC) 1298

24.1 SDMMC 主要特性 1298
24.2 SDMMC 实现 1298
24.3 SDMMC 总线拓扑 1299
24.4 SDMMC 工作模式 1301
24.5 SDMMC 功能描述 1302
24.5.1 SDMMC 框图 1302
24.5.2 SDMMC 引脚和内部信号 1302
24.5.3 一般说明 1303
24.5.4 SDMMC 适配器 1305
24.5.5 SDMMC AHB 从接口 1323
24.5.6 SDMMC AHB 主接口 1324
24.5.7 AHB 和 SDMMC_CK 时钟的关系 1327
24.6 卡功能描述 1327
24.6.1 SD I/O 模式 1327
24.6.2 CMD12 发送时序 1335
24.6.3 睡眠 (CMD5) 1338
24.6.4 中断模式 (Wait-IRQ) 1338
24.6.5 启动操作 1339
24.6.6 响应 R1b 的处理 1342

24.6.7 复位和卡掉电再上电 1342
24.7 硬件流控制 1344
24.8 超高速 I 相 (UHS-I) 电压切换 1345
24.9 SDMMC 中断 1348
24.10 SDMMC 寄存器 1349
24.10.1 SDMMC 电源控制寄存器 (SDMMC_POWER) 1349
24.10.2 SDMMC 时钟控制寄存器 (SDMMC_CLKCR) 1350
24.10.3 SDMMC 参数寄存器 (SDMMC_ARGR) 1352
24.10.4 SDMMC 命令寄存器 (SDMMC_CMDR) 1352
24.10.5 SDMMC 命令响应寄存器 (SDMMC_RESPCMDR) 1354
24.10.6 SDMMC 响应 x 寄存器 (SDMMC_RESPxR) 1354
24.10.7 SDMMC 数据定时器寄存器 (SDMMC_DTIMER) 1355
24.10.8 SDMMC 数据长度寄存器 (SDMMC_DLENR) 1355
24.10.9 SDMMC 数据控制寄存器 (SDMMC_DCTRL) 1356
24.10.10 SDMMC 数据计数器寄存器 (SDMMC_DCNTR) 1358
24.10.11 SDMMC 状态寄存器 (SDMMC_STAR) 1358
24.10.12 SDMMC 中断清零寄存器 (SDMMC_ICR) 1361
24.10.13 SDMMC 屏蔽寄存器 (SDMMC_MASKR) 1363
24.10.14 SDMMC 确认定时器寄存器 (SDMMC_ACKTIMER) 1366
24.10.15 SDMMC 数据 FIFO 寄存器 x (SDMMC_FIFORx) 1366
24.10.16 SDMMC DMA 控制寄存器 (SDMMC_IDMACTRLR) 1367
24.10.17 SDMMC IDMA 缓冲区大小寄存器 (SDMMC_IDMABSIZER) 1367
24.10.18 SDMMC IDMA 缓冲区基址寄存器 (SDMMC_IDMABASER) 1368
24.10.19 SDMMC IDMA 链表地址寄存器 (SDMMC_IDMALAR) 1368
24.10.20 SDMMC IDMA 链表内存基址寄存器 (SDMMC_IDMABAR) 1369
24.10.21 SDMMC 版本寄存器 (SDMMC_VERR) 1370
24.10.22 SDMMC 标识寄存器 (SDMMC_IPIDR) 1370
24.10.23 SDMMC 大小标识寄存器 (SDMMC_SIDR) 1370
24.10.24 SDMMC 寄存器映射 1371

25 延迟模块 (DLYB) 1374
25.1 简介 1374
25.2 DLYB 主要特性 1374
25.3 DLYB 功能描述 1374
25.3.1 DLYB 框图 1374
25.3.2 DLYB 引脚和内部信号 1375

25.3.3 一般说明 1375
25.3.4 延迟线长度配置程序 1376
25.3.5 输出时钟相位配置程序 1376
25.4 DLYB 寄存器 1377
25.4.1 DLYB 控制寄存器 (DLYB_CR) 1377
25.4.2 DLYB 配置寄存器 (DLYB_CFGR) 1377
25.4.3 DLYB 寄存器映射 1378

26 模数转换器 (ADC1/2) 1379
26.1 简介 1379
26.2 ADC 主要特性 1379
26.3 ADC 实现 1381
26.4 ADC 功能描述 1382
26.4.1 ADC 框图 1382
26.4.2 ADC 引脚和内部信号 1383
26.4.3 ADC 时钟 1386
26.4.4 ADC 连接 1388
26.4.5 从设备 AHB 接口 1390
26.4.6 ADC 深度掉电模式 (DEEPPWD) 和 ADC 稳压器 (ADVREGEN) . . . 1390
26.4.7 单端输入通道和差分输入通道 1390
26.4.8 校准（ADCAL、ADCALDIF、ADC_CALFACT） 1391
26.4.9 ADC 开关控制（ADEN、ADDIS、ADRDY） 1393
26.4.10 写入 ADC 控制位时的限制 1394
26.4.11 通道选择（SQRx、JSQRx） 1395
26.4.12 可独立设置各通道采样时间（SMPR1、SMPR2） 1396
26.4.13 单次转换模式 (CONT = 0) 1398
26.4.14 连续转换模式 (CONT = 1) 1398
26.4.15 开始转换（ADSTART、JADSTART） 1399
26.4.16 ADC 时序 1400
26.4.17 停止正在进行的转换（ADSTP、JADSTP） 1400
26.4.18 外部触发转换和触发极性（EXTSEL、EXTEN、JEXTSEL、JEXTEN） 1402
26.4.19 注入通道管理 1403
26.4.20 不连续模式（DISCEN、DISCNUM、JDISCEN） 1404
26.4.21 注入转换的上下文队列 1405
26.4.22 可编程分辨率 (RES)——快速转换模式 1413
26.4.23 转换结束、采样阶段结束（EOC、JEOC、EOSMP） 1414

26.4.24 转换序列结束（EOS、JEOS） 1414
26.4.25 时序图示例（单次模式/连续模式，硬件/软件触发） 1415
26.4.26 数据管理 1417
26.4.27 使用 DFSDM 管理转换 1423
26.4.28 动态低功耗特性 1423
26.4.29 模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx） 1428
26.4.30 过采样器 1432
26.4.31 温度传感器 1437
26.4.32 VBAT 电源监测 1439
26.4.33 监测内部参考电压 1439
26.4.34 监视电源电压 1441
26.5 ADC 中断 1441
26.6 ADC 寄存器 1442
26.6.1 ADC 中断和状态寄存器 (ADC_ISR) 1442
26.6.2 ADC 中断使能寄存器 (ADC_IER) 1444
26.6.3 ADC 控制寄存器 (ADC_CR) 1446
26.6.4 ADC 配置寄存器 (ADC_CFGR) 1449
26.6.5 ADC 配置寄存器 2 (ADC_CFGR2) 1453
26.6.6 ADC 采样时间寄存器 1 (ADC_SMPR1) 1455
26.6.7 ADC 采样时间寄存器 2 (ADC_SMPR2) 1455
26.6.8 ADC 看门狗阈值寄存器 1 (ADC_TR1) 1456
26.6.9 ADC 看门狗阈值寄存器 2 (ADC_TR2) 1457
26.6.10 ADC 看门狗阈值寄存器 3 (ADC_TR3) 1458
26.6.11 ADC 常规序列寄存器 1 (ADC_SQR1) 1458
26.6.12 ADC 常规序列寄存器 2 (ADC_SQR2) 1459
26.6.13 ADC 常规序列寄存器 3 (ADC_SQR3) 1460
26.6.14 ADC 常规序列寄存器 4 (ADC_SQR4) 1461
26.6.15 ADC 常规数据寄存器 (ADC_DR) 1462
26.6.16 ADC 注入序列寄存器 (ADC_JSQR) 1462
26.6.17 ADC 偏移 y 寄存器 (ADC_OFRy) 1464
26.6.18 ADC 注入通道 y 数据寄存器 (ADC_JDRy) 1465
26.6.19 ADC 模拟看门狗 2 配置寄存器 (ADC_AWD2CR) 1465
26.6.20 ADC 模拟看门狗 3 配置寄存器 (ADC_AWD3CR) 1466
26.6.21 ADC 差分模式选择寄存器 (ADC_DIFSEL) 1466
26.6.22 ADC 校准系数 (ADC_CALFACT) 1467
26.6.23 ADC 选项寄存器 (ADC_OR) 1467

26.7 ADC 通用寄存器 1468
26.7.1 ADC 通用控制寄存器 (ADC_CCR) 1468
26.7.2 ADC 硬件配置寄存器 (ADC_HWCFGR0) 1469
26.7.3 ADC 版本触发器 (ADC_VERR) 1470
26.7.4 ADC 标识寄存器 (ADC_IPDR) 1471
26.7.5 ADC 大小标识寄存器 (ADC_SIDR) 1471
26.8 ADC 寄存器映射 1472

27 数字温度传感器 (DTS) 1475
27.1 简介 1475
27.2 DTS 主要特性 1475
27.3 DTS 功能描述 1476
27.3.1 DTS 框图 1476
27.3.2 DTS 内部信号 1476
27.3.3 DTS 模块操作 1476
27.3.4 操作模式 1477
27.3.5 校准 1477
27.3.6 预分频器 1477
27.3.7 温度测量原理 1477
27.3.8 采样时间 1478
27.3.9 快速测量模式 1479
27.3.10 触发输入 1479
27.3.11 开关控制和就绪标志 1480
27.3.12 温度测量序列 1480
27.4 DTS 低功耗模式 1481
27.5 DTS 中断 1481
27.5.1 温度窗口比较器 1481
27.5.2 同步中断 1481
27.5.3 异步唤醒 1481
27.6 DTS 寄存器 1482
27.6.1 温度传感器配置寄存器 1 (DTS_CFGR1) 1482
27.6.2 温度传感器 T0 值寄存器 1 (DTS_T0VALR1) 1484
27.6.3 温度传感器斜坡值寄存器 (DTS_RAMPVALR) 1484
27.6.4 温度传感器中断阈值寄存器 1 (DTS_ITR1) 1485
27.6.5 温度传感器数据寄存器 (DTS_DR) 1485
27.6.6 温度传感器状态寄存器 (DTS_SR) 1486

27.6.7 温度传感器中断使能寄存器 (DTS_ITENR) 1487
27.6.8 温度传感器清除中断标志寄存器 (DTS_ICIFR) 1488
27.6.9 温度传感器选项寄存器 (DTS_OR) 1489
27.6.10 DTS 寄存器映射 1490

28 电压参考缓冲器 (VREFBUF) 1491
28.1 简介 1491
28.2 VREFBUF 实现 1491
28.3 VREFBUF 功能描述 1491
28.4 VREFBUF 校准 1492
28.5 VREFBUF 寄存器 1492
28.5.1 VREFBUF 控制和状态寄存器 (VREFBUF_CSR) 1492
28.5.2 VREFBUF 校准控制寄存器 (VREFBUF_CCR) 1493
28.5.3 VREFBUF 寄存器映射 1494

29 ∑Δ 调制器数字滤波器 (DFSDM) 1495
29.1 简介 1495
29.2 DFSDM主要特性 1496
29.3 DFSDM 实现 1497
29.4 DFSDM 功能描述 1498
29.4.1 DFSDM 框图 1498
29.4.2 DFSDM 引脚和内部信号 1499
29.4.3 DFSDM 复位和时钟 1500
29.4.4 串行通道收发器 1501
29.4.5 配置输入串行接口 1510
29.4.6 并行数据输入 1510
29.4.7 通道选择 1512
29.4.8 数字滤波器配置 1513
29.4.9 积分器单元 1514
29.4.10 模拟看门狗 1514
29.4.11 短路检测器 1516
29.4.12 极值检测器 1517
29.4.13 数据单元模块 1517
29.4.14 有符号数据格式 1518
29.4.15 启动转换 1518
29.4.16 连续和快速连续模式 1519

29.4.17 请求优先级 1520
29.4.18 在运行模式下的功耗优化 1520
29.5 DFSDM 中断 1520
29.6 DFSDM DMA 传输 1522
29.7 DFSDM 通道 y 寄存器 (y=0..3) 1522
29.7.1 DFSDM 通道 y 配置寄存器 (DFSDM_CHyCFGR1) 1522
29.7.2 DFSDM 通道 y 配置寄存器 (DFSDM_CHyCFGR2) 1525
29.7.3 DFSDM 通道 y 模拟看门狗和短路检测器寄存器 (DFSDM_CHyAWSCDR) 1525
29.7.4 DFSDM 通道 y 看门狗滤波器数据寄存器 (DFSDM_CHyWDATR) . . 1526
29.7.5 DFSDM 通道 y 数据输入寄存器 (DFSDM_CHyDATINR) 1527
29.7.6 DFSDM 通道 y 延迟寄存器 (DFSDM_CHyDLYR) 1528
29.8 DFSDM 滤波器 x 模块寄存器 (x=0..1) 1528
29.8.1 DFSDM 滤波器 x 控制寄存器 1 (DFSDM_FLTxCR1) 1528
29.8.2 DFSDM 滤波器 x 控制寄存器 2 (DFSDM_FLTxCR2) 1531
29.8.3 DFSDM 滤波器 x 中断和状态寄存器 (DFSDM_FLTxISR) 1532
29.8.4 DFSDM 滤波器 x 中断标志清零寄存器 (DFSDM_FLTxICR) 1534
29.8.5 DFSDM 滤波器 x 注入通道组选择寄存器 (DFSDM_FLTxJCHGR) . . 1535
29.8.6 DFSDM 滤波器 x 控制寄存器 (DFSDM_FLTxFCR) 1535
29.8.7 注入组的 DFSDM 滤波器 x 数据寄存器 (DFSDM_FLTxJDATAR) . . . 1536
29.8.8 常规通道的 DFSDM 滤波器 x 数据寄存器 (DFSDM_FLTxRDATAR) . 1537
29.8.9 DFSDM 滤波器 x 模拟看门狗阈值上限寄存器 (DFSDM_FLTxAWHTR) 1538
29.8.10 DFSDM 滤波器 x 模拟看门狗阈值下限寄存器 (DFSDM_FLTxAWLTR) 1538
29.8.11 DFSDM 滤波器 x 模拟看门狗状态寄存器 (DFSDM_FLTxAWSR) . . . 1539
29.8.12 DFSDM 滤波器 x 模拟看门狗清零标志寄存器 (DFSDM_FLTxAWCFR) 1539
29.8.13 DFSDM 滤波器 x 极值检测器最大值寄存器 (DFSDM_FLTxEXMAX) 1540
29.8.14 DFSDM 滤波器 x 极值检测器最小值寄存器 (DFSDM_FLTxEXMIN) . . 1540
29.8.15 DFSDM 滤波器 x 转换定时器寄存器 (DFSDM_FLTxCNVTIMR) 1541
29.9 DFSDM 版本寄存器 1542
29.9.1 DFSDM 硬件配置寄存器 (DFSDM_HWCFGR) 1542
29.9.2 DFSDM 版本寄存器 (DFSDM_VERR) 1542
29.9.3 DFSDM 标识寄存器 (DFSDM_IPIDR) 1543
29.9.4 DFSDM 大小标识寄存器 (DFSDM_SIDR) 1543
29.9.5 DFSDM 寄存器映射 1544

30 数字摄像头接口像素流水线 (DCMIPP) 1549
30.1 简介 1549
30.2 DCMIPP 主要特性 1550
30.3 DCMIPP 功能描述 1551
30.3.1 DCMIPP 框图 1551
30.3.2 DCMIPP 引脚和内部信号 1551
30.3.3 DCMIPP 复位和时钟 1552
30.3.4 DCMIPP 最大分辨率 1554
30.3.5 DCMIPP 对帧结构的最低要求 1554
30.3.6 DCMIPP 像素格式支持的说明 1554
30.4 DCMIPP 输入和流控制 1555
30.4.1 并行输入接口 1555
30.4.2 帧计数器 1559
30.4.3 帧控制 1559
30.4.4 流水线禁用 1562
30.5 Pipe0（转储流水线） 1563
30.5.1 概述 1563
30.5.2 抽取 1563
30.5.3 裁剪/统计数据选择/抑制 1564
30.5.4 转储计数器 1564
30.5.5 双缓冲区模式 1565
30.5.6 像素封装 1566
30.5.7 上溢检测 1566
30.6 像素格式说明 1567
30.6.1 并行接口格式 1567
30.6.2 转储流水线格式 1569
30.6.3 AXI IP-Plug 1570
30.7 影子寄存器 1572
30.8 DCMIPP 低功耗模式 1572
30.9 DCMIPP 中断 1573
30.9.1 自由运行 DCMIPP 1573
30.9.2 中断 1573
30.9.3 事件引脚 1575
30.10 DCMIPP 寄存器 1575
30.10.1 DCMIPP IP-Plug 全局寄存器 1 (DCMIPP_IPGR1) 1576
30.10.2 DCMIPP IP-Plug 全局寄存器 2 (DCMIPP_IPGR2) 1576

30.10.3 DCMIPP IP-Plug 全局寄存器 3 (DCMIPP_IPGR3) 1577
30.10.4 DCMIPP IP-Plug 标识寄存器 (DCMIPP_IPGR8) 1577
30.10.5 DCMIPP IP-Plug 客户端 x 寄存器 1 (DCMIPP_IPCxR1) 1578
30.10.6 DCMIPP IP-Plug 客户端 x 寄存器 2 (DCMIPP_IPCxR2) 1578
30.10.7 DCMIPP IP-Plug 客户 x 寄存器 3 (DCMIPP_IPCxR3) 1579
30.10.8 DCMIPP 并行接口控制寄存器 (DCMIPP_PRCR) 1579
30.10.9 DCMIPP 并行接口内嵌同步码寄存器 (DCMIPP_PRESCR) 1581
30.10.10 DCMIPP 并行接口内嵌码同步取消屏蔽寄存器 (DCMIPP_PRESUR) . . 1582
30.10.11 DCMIPP 并行接口中断使能寄存器 (DCMIPP_PRIER) 1582
30.10.12 DCMIPP 并行接口状态寄存器 (DCMIPP_PRSR) 1583
30.10.13 DCMIPP 并行接口中断清零寄存器 (DCMIPP_PRFCR) 1584
30.10.14 DCMIPP 通用 IP 硬件配置寄存器 (DCMIPP_CMHWCFGR) 1584
30.10.15 DCMIPP 通用配置寄存器 (DCMIPP_CMCR) 1585
30.10.16 DCMIPP 通用帧计数器寄存器 (DCMIPP_CMFRCR) 1585
30.10.17 DCMIPP 通用中断使能寄存器 (DCMIPP_CMIER) 1586
30.10.18 DCMIPP 通用状态寄存器 1 (DCMIPP_CMSR1) 1587
30.10.19 DCMIPP 通用状态寄存器 2 (DCMIPP_CMSR2) 1588
30.10.20 DCMIPP 通用中断清零寄存器 (DCMIPP_CMFCR) 1589
30.10.21 DCMIPP Pipe0 硬件配置寄存器 (DCMIPP_P0HWCFGR) 1590
30.10.22 DCMIPP Pipe0 流选择配置寄存器 (DCMIPP_P0FSCR) 1591
30.10.23 DCMIPP Pipe0 流控制配置寄存器 (DCMIPP_P0FCTCR) 1592
30.10.24 DCMIPP Pipe0 统计数据/裁剪开始寄存器 (DCMIPP_P0SCSTR) . . 1592
30.10.25 DCMIPP Pipe0 统计数据/裁剪大小寄存器 (DCMIPP_P0SCSZR) . . 1593
30.10.26 DCMIPP Pipe0 转储计数器寄存器 (DCMIPP_P0DCCNTR) 1594
30.10.27 DCMIPP Pipe0 转储限制寄存器 (DCMIPP_P0DCLMTR) 1594
30.10.28 DCMIPP Pipe0 像素打包器配置寄存器 (DCMIPP_P0PPCR) 1595
30.10.29 DCMIPP Pipe0 像素打包器内存 0 地址寄存器 1 (DCMIPP_P0PPM0AR1) 1596
30.10.30 DCMIPP Pipe0 像素打包器内存 0 地址寄存器 2 (DCMIPP_P0PPM0AR2) 1596
30.10.31 DCMIPP Pipe0 中断使能寄存器 (DCMIPP_P0IER) 1597
30.10.32 DCMIPP Pipe0 状态寄存器 (DCMIPP_P0SR) 1597
30.10.33 DCMIPP Pipe0 中断清零寄存器 (DCMIPP_P0FCR) 1598
30.10.34 DCMIPP Pipe0 当前流控制配置寄存器 (DCMIPP_P0CFCTCR) 1599
30.10.35 DCMIPP Pipe0 当前统计数据/剪裁开始寄存器 (DCMIPP_P0CSCSTR) 1600
30.10.36 DCMIPP Pipe0 当前统计数据/剪裁大小寄存器 (DCMIPP_P0CSCSZR) 1600

30.10.37 DCMIPP Pipe0 当前像素打包器配置寄存器 (DCMIPP_P0CPPCR) . 1601
30.10.38 DCMIPP Pipe0 当前像素打包器内存 0 地址寄存器 1 (DCMIPP_P0CPPM0AR1) 1602
30.10.39 DCMIPP Pipe0 当前像素打包器内存 0 地址寄存器 2 (DCMIPP_P0CPPM0AR2) 1603
30.10.40 DCMIPP 硬件配置寄存器 2 (DCMIPP_HWCFGR2) 1603
30.10.41 DCMIPP 硬件配置寄存器 1 (DCMIPP_HWCFGR1) 1604
30.10.42 DCMIPP 版本寄存器 (DCMIPP_VERR) 1605
30.10.43 DCMIPP 标识寄存器 (DCMIPP_IPIDR) 1606
30.10.44 DCMIPP 大小标识寄存器 (DCMIPP_SIDR) 1606
30.11 DCMIPP 寄存器映射 1607

31 LCD-TFT 显示控制器 (LTDC) 1611

31.1 LTDC 简介 1611
31.2 LTDC 主要特性 1611
31.3 LTDC 功能描述 1612
31.3.1 LTDC 框图 1612
31.3.2 LTDC 引脚和内部信号 1612
31.3.3 LTDC 复位和时钟 1613
31.4 LTDC 配置参数 1614
31.4.1 AXI 主设备 1614
31.4.2 输入图层定义和裁剪 1615
31.4.3 输入像素格式 1616
31.4.4 YUV 平面 1618
31.4.5 YUV 到 RGB 颜色转换 1618
31.4.6 水平或垂直镜像 1620
31.4.7 默认图层颜色 1620
31.4.8 查色表 (CLUT) 1621
31.4.9 透明色键 1621
31.4.10 显示合成 - 窗口化 1622
31.4.11 显示组成——混合 1623
31.4.12 伽马校正 1624
31.4.13 YUV 输出转换 1625
31.4.14 抖动 1625
31.4.15 CRC 哈希 1626
31.4.16 显示时序 1626
31.4.17 输出接口极性 1628

31.4.18 影子寄存器 1629
31.4.19 常规控制 1630
31.5 LTDC 中断 1630
31.6 LTDC 编程步骤 1631
31.7 LTDC 寄存器 1632
31.7.1 LTDC 标识寄存器 (LTDC_IDR) 1632
31.7.2 LDTC 图层计数寄存器 (LTDC_LCR) 1632
31.7.3 LTDC 同步大小配置寄存器 (LTDC_SSCR) 1633
31.7.4 LTDC 后沿配置寄存器 (LTDC_BPCR) 1633
31.7.5 LTDC 有效宽度配置寄存器 (LTDC_AWCR) 1634
31.7.6 LTDC 总宽度配置寄存器 (LTDC_TWCR) 1634
31.7.7 LTDC 全局控制寄存器 (LTDC_GCR) 1635
31.7.8 LTDC 全局配置 1 寄存器 (LTDC_GC1R) 1637
31.7.9 LTDC 全局配置 2 寄存器 (LTDC_GC2R) 1638
31.7.10 LTDC 影子重载配置寄存器 (LTDC_SRCR) 1639
31.7.11 LTDC 伽马校正配置寄存器 (LTDC_GCCR) 1639
31.7.12 LTDC 背景色配置寄存器 (LTDC_BCCR) 1640
31.7.13 LTDC 中断使能寄存器 (LTDC_IER) 1640
31.7.14 LTDC 中断状态寄存器 (LTDC_ISR) 1641
31.7.15 LTDC 中断清零寄存器 (LTDC_ICR) 1642
31.7.16 LTDC 行中断位置配置寄存器 (LTDC_LIPCR) 1643
31.7.17 LTDC 当前位置状态寄存器 (LTDC_CPSR) 1643
31.7.18 LTDC 当前显示状态寄存器 (LTDC_CDSR) 1644
31.7.19 LTDC 外部显示控制寄存器 (LTDC_EDCR) 1644
31.7.20 LTDC 中断使能寄存器 2 (LTDC_IER2) 1645
31.7.21 LTDC 中断状态寄存器 2 (LTDC_ISR2) 1646
31.7.22 LTDC 中断清零寄存器 2 (LTDC_ICR2) 1647
31.7.23 LTDC 行中断位置配置寄存器 2 (LTDC_LIPCR2) 1648
31.7.24 LTDC 预期 CRC 寄存器 (LTDC_ECRCR) 1648
31.7.25 LTDC 计算 CRC 寄存器 (LTDC_CCRCR) 1648
31.7.26 LTDC FIFO 下溢阈值寄存器 (LTDC_FUTR) 1649
31.7.27 LTDC 第 x 层配置 0 寄存器 (LTDC_LxC0R) 1649
31.7.28 LTDC 第 x 层配置 1 寄存器 (LTDC_LxC1R) 1650
31.7.29 LTDC 第 x 层重载控制寄存器 (LTDC_LxRCR) 1651
31.7.30 LTDC 第 x 层控制寄存器 (LTDC_LxCR) 1652
31.7.31 LTDC 第 x 层窗口水平位置配置寄存器 (LTDC_LxWHPCR) 1653
31.7.32 LTDC 第 x 层窗口垂直位置配置寄存器 (LTDC_LxWVPCR) 1653

31.7.33 LTDC 第 x 层色键配置寄存器 (LTDC_LxCKCR) 1654
31.7.34 LTDC 第 x 层像素格式配置寄存器 (LTDC_LxPFCR) 1654
31.7.35 LTDC 第 x 层恒定 alpha 配置寄存器 (LTDC_LxCACR) 1655
31.7.36 LTDC 第 x 层默认颜色配置寄存器 (LTDC_LxDCCR) 1655
31.7.37 LTDC 第 x 层混合系数配置寄存器 (LTDC_LxBFCR) 1656
31.7.38 LTDC 第 x 层突发传输长度配置寄存器 (LTDC_LxBLCR) 1657
31.7.39 LTDC 第 x 层平面配置寄存器 (LTDC_LxPCR) 1657
31.7.40 LTDC 第 x 层颜色帧缓冲区地址寄存器 (LTDC_LxCFBAR) 1658
31.7.41 LTDC 第 x 层颜色帧缓冲区长度寄存器 (LTDC_LxCFBLR) 1659
31.7.42 LTDC 第 x 层颜色帧缓冲区行数寄存器 (LTDC_LxCFBLNR) 1659
31.7.43 LTDC 第 1 层辅助帧缓冲区地址 0 寄存器 (LTDC_L1AFBA0R) 1660
31.7.44 LTDC 第 1 层辅助帧缓冲区地址 1 寄存器 (LTDC_L1AFBA1R) 1660
31.7.45 LTDC 第 1 层辅助帧缓冲区长度寄存器 (LTDC_L1AFBLR) 1661
31.7.46 LTDC 第 1 层辅助帧缓冲区行数寄存器 (LTDC_L1AFBLNR) 1661
31.7.47 LTDC 第 x 层 CLUT 写寄存器 (LTDC_LxCLUTWR) 1662
31.7.48 LTDC 第 x 层转换 YCbCr RGB 0 寄存器 (LTDC_LxCYR0R) 1662
31.7.49 LTDC 第 x 层转换 YCbCr RGB 1 寄存器 (LTDC_LxCYR1R) 1663
31.7.50 LTDC 第 x 层灵活像素格式 0 寄存器 (LTDC_LxFPF0R) 1663
31.7.51 LTDC 第 x 层灵活像素格式 1 寄存器 (LTDC_LxFPF1R) 1664
31.7.52 LTDC 寄存器映射 1664

32 真随机数发生器 (RNG) 1670

32.1 简介 1670
32.2 RNG 主要特性 1670
32.3 RNG 功能描述 1671
32.3.1 RNG 框图 1671
32.3.2 RNG 内部信号 1671
32.3.3 随机数生成 1672
32.3.4 RNG 初始化 1674
32.3.5 RNG 操作 1675
32.3.6 RNG 时钟 1676
32.3.7 错误管理 1676
32.3.8 RNG 低功耗使用 1677
32.4 RNG 中断 1678
32.5 RNG 处理时间 1678
32.6 RNG 熵源验证 1678

32.6.1 简介 . . . 1678
32.6.2 验证条件 . . . 1679
32.6.3 数据采集 . . . 1679
32.7 RNG 寄存器 . . . 1680
32.7.1 RNG 控制寄存器 (RNG_CR) . . . 1680
32.7.2 RNG 状态寄存器 (RNG_SR) . . . 1682
32.7.3 RNG 数据寄存器 (RNG_DR) . . . 1683
32.7.4 RNG 噪声源控制寄存器 (RNG_NSCR) . . . 1683
32.7.5 RNG 运行状态测试控制寄存器 (RNG_HTCR) . . . 1684
32.7.6 RNG 硬件配置寄存器 (RNG_HWCFGR) . . . 1684
32.7.7 RNG 版本寄存器 (RNG_VERR) . . . 1685
32.7.8 RNG 标识寄存器 (RNG_IPIDR) . . . 1685
32.7.9 RNG 大小标识寄存器 (RNG_SIDR) . . . 1686
32.7.10 RNG 寄存器映射 . . . 1687

33 安全 AES 协处理器 (SAES) . . . 1688
33.1 简介 . . . 1688
33.2 SAES 主要特性 . . . 1688
33.3 SAES 实现 . . . 1689
33.4 SAES 功能描述 . . . 1689
33.4.1 SAES 框图 . . . 1689
33.4.2 SAES 内部信号 . . . 1690
33.4.3 SAES 密码内核 . . . 1690
33.4.4 执行密码操作的 SAES 过程 . . . 1696
33.4.5 SAES 解密轮密钥准备 . . . 1699
33.4.6 SAES 密文窃取和数据填充 . . . 1699
33.4.7 SAES 任务挂起和恢复 . . . 1700
33.4.8 SAES 基本链接模式（ECB 和 CBC） . . . 1701
33.4.9 SAES 计数器 (CTR) 模式 . . . 1705
33.4.10 SAES Galois/计数器模式 (GCM) . . . 1707
33.4.11 SAES Galois 消息认证码 (GMAC) . . . 1712
33.4.12 SAES CBC-MAC 计数器模式 (CCM) . . . 1713
33.4.13 SAES 使用封装密钥进行操作 . . . 1718
33.4.14 使用共享密钥进行 SAES 操作 . . . 1721
33.4.15 SAES 数据寄存器和数据交换 . . . 1723
33.4.16 SAES 密钥寄存器 . . . 1725
33.4.17 SAES 初始化向量寄存器 . . . 1726

33.4.18 SAES DMA 接口 1726
33.4.19 SAES 错误管理 1728
33.5 SAES 中断 1729
33.6 SAES 处理延迟 1730
33.7 SAES 寄存器 1731
33.7.1 SAES 控制寄存器 (SAES_CR) 1731
33.7.2 SAES 状态寄存器 (SAES_SR) 1734
33.7.3 SAES 数据输入寄存器 (SAES_DINR) 1736
33.7.4 SAES 数据输出寄存器 (SAES_DOUTR) 1736
33.7.5 SAES 密钥寄存器 0 (SAES_KEYR0) 1737
33.7.6 SAES 密钥寄存器 1 (SAES_KEYR1) 1737
33.7.7 SAES 密钥寄存器 2 (SAES_KEYR2) 1738
33.7.8 SAES 密钥寄存器 3 (SAES_KEYR3) 1738
33.7.9 SAES 初始化向量寄存器 0 (SAES_IVR0) 1738
33.7.10 SAES 初始化向量寄存器 1 (SAES_IVR1) 1739
33.7.11 SAES 初始化向量寄存器 2 (SAES_IVR2) 1739
33.7.12 SAES 初始化向量寄存器 3 (SAES_IVR3) 1740
33.7.13 SAES 密钥寄存器 4 (SAES_KEYR4) 1740
33.7.14 SAES 密钥寄存器 5 (SAES_KEYR5) 1740
33.7.15 SAES 密钥寄存器 6 (SAES_KEYR6) 1741
33.7.16 SAES 密钥寄存器 7 (SAES_KEYR7) 1741
33.7.17 SAES 挂起寄存器 (SAES_SUSPxR) 1741
33.7.18 SAES 中断使能寄存器 (SAES_IER) 1742
33.7.19 SAES 中断状态寄存器 (SAES_ISR) 1743
33.7.20 SAES 中断清零寄存器 (SAES_ICR) 1744
33.7.21 SAES 硬件配置寄存器 (SAES_HWCFGR) 1745
33.7.22 SAES 版本寄存器 (SAES_VERR) 1745
33.7.23 SAES 标识寄存器 (SAES_IPIDR) 1746
33.7.24 SAES 尺寸 ID 寄存器 (SAES_SIDR) 1746
33.7.25 SAES 寄存器映射 1746

34 密码处理器 (CRYP) 1749
34.1 简介 1749
34.2 CRYP 主要特性 1749
34.3 CRYP 实现 1750
34.4 CRYP 功能描述 1751

34.4.1 CRYP 框图 1751
34.4.2 CRYP 内部信号 1752
34.4.3 CRYP DES/TDES 密码内核 1752
34.4.4 CRYP AES 密码内核 1753
34.4.5 执行加密操作的 CRYP 过程 1759
34.4.6 CRYP 繁忙状态 1761
34.4.7 为解密准备 CRYP AES 密钥 1762
34.4.8 CRYP 窃取和数据填充 1762
34.4.9 CRYP 挂起/恢复操作 1763
34.4.10 CRYP DES/TDES 基本链接模式（ECB 和 CBC） 1764
34.4.11 CRYP AES 基本链接模式（ECB 和 CBC） 1769
34.4.12 CRYP AES 计数器模式 (AES-CTR) 1774
34.4.13 CRYP AES Galois/计数器模式 (GCM) 1778
34.4.14 CRYP AES Galois 消息认证码 (GMAC) 1782
34.4.15 CRYP AES CBC-MAC 计数器模式 (CCM) 1783
34.4.16 与安全 AES 协处理器共享 AES 密钥 1787
34.4.17 CRYP 数据寄存器和数据交换 1787
34.4.18 CRYP 密钥寄存器 1791
34.4.19 CRYP 初始化向量寄存器 1792
34.4.20 CRYP DMA 接口 1793
34.4.21 CRYP 错误管理 1795
34.5 CRYP 中断 1795
34.6 CRYP 处理时间 1796
34.7 CRYP 寄存器 1797
34.7.1 CRYP 控制寄存器 (CRYP_CR) 1797
34.7.2 CRYP 状态寄存器 (CRYP_SR) 1800
34.7.3 CRYP 数据输入寄存器 (CRYP_DIN) 1801
34.7.4 CRYP 数据输出寄存器 (CRYP_DOUT) 1802
34.7.5 CRYP DMA 控制寄存器 (CRYP_DMACR) 1802
34.7.6 CRYP 中断屏蔽置 1/清零寄存器 (CRYP_IMSCR) 1803
34.7.7 CRYP 原始中断状态寄存器 (CRYP_RISR) 1803
34.7.8 CRYP 屏蔽中断状态寄存器 (CRYP_MISR) 1804
34.7.9 CRYP 密钥寄存器 0L (CRYP_K0LR) 1805
34.7.10 CRYP 密钥寄存器 0R (CRYP_K0RR) 1805
34.7.11 CRYP 密钥寄存器 1L (CRYP_K1LR) 1806
34.7.12 CRYP 密钥寄存器 1R (CRYP_K1RR) 1806
34.7.13 CRYP 密钥寄存器 2L (CRYP_K2LR) 1807

34.7.14 CRYP 密钥寄存器 2R (CRYP_K2RR) 1807
34.7.15 CRYP 密钥寄存器 3L (CRYP_K3LR) 1808
34.7.16 CRYP 密钥寄存器 3R (CRYP_K3RR) 1808
34.7.17 CRYP 初始化向量寄存器 0L (CRYP_IV0LR) 1809
34.7.18 CRYP 初始化向量寄存器 0R (CRYP_IV0RR) 1809
34.7.19 CRYP 初始化向量寄存器 1L (CRYP_IV1LR) 1810
34.7.20 CRYP 初始化向量寄存器 1R (CRYP_IV1RR) 1810
34.7.21 CRYP 上下文交换 GCM-CCM 寄存器 (CRYP_CSGCMCCMxR) . . . 1811
34.7.22 CRYP 上下文交换 GCM 寄存器 (CRYP_CSGCMxR) 1811
34.7.23 CRYP 硬件配置寄存器 (CRYP_HWCFGR) 1812
34.7.24 CRYP 版本寄存器 (CRYP_VERR) 1812
34.7.25 CRYP 标识 (CRYP_IPIDR) 1813
34.7.26 CRYP 大小标识寄存器 (CRYP_SIDR) 1813
34.7.27 CRYP 寄存器映射 1814

35 哈希处理器 (HASH) 1817

35.1 简介 1817
35.2 HASH 主要特性 1817
35.3 HASH 实现 1818
35.4 HASH 功能描述 1818
35.4.1 HASH 框图 1818
35.4.2 HASH 内部信号 1819
35.4.3 关于安全哈希算法 1819
35.4.4 消息数据输送 1820
35.4.5 哈希值计算 1821
35.4.6 消息填充 1823
35.4.7 HMAC 运算 1825
35.4.8 HASH 挂起/恢复操作 1827
35.4.9 HASH DMA 接口 1829
35.4.10 HASH 错误管理 1829
35.5 HASH 中断 1830
35.6 HASH 处理时间 1830
35.7 HASH 寄存器 1831
35.7.1 HASH 控制寄存器 (HASH_CR) 1831
35.7.2 HASH 数据输入寄存器 (HASH_DIN) 1833
35.7.3 HASH 启动寄存器 (HASH_STR) 1834

35.7.4 哈希结果寄存器 1835
35.7.5 HASH 中断使能寄存器 (HASH_IMR) 1837
35.7.6 HASH 状态寄存器 (HASH_SR) 1837
35.7.7 HASH 上下文交换寄存器 1838
35.7.8 HASH 硬件配置寄存器 (HASH_HWCFGR) 1839
35.7.9 HASH 版本寄存器 (HASH_VERR) 1840
35.7.10 HASH 标识寄存器 (HASH_IPIDR) 1840
35.7.11 HASH 大小识别寄存器 (HASH_SIDR) 1840
35.7.12 HASH 寄存器映射 1841

36 公钥加速器 (PKA) 1843

36.1 简介 1843
36.2 PKA 主要特性 1843
36.3 PKA 功能描述 1844
36.3.1 PKA 框图 1844
36.3.2 PKA 内部信号 1844
36.3.3 PKA 复位和时钟 1844
36.3.4 PKA 公钥加速 1845
36.3.5 PKA 的典型应用 1846
36.3.6 PKA 执行过程 1849
36.3.7 PKA 错误管理 1850
36.4 PKA 运算模式 1850
36.4.1 简介 1850
36.4.2 Montgomery 参数计算 1851
36.4.3 模加 1852
36.4.4 模减 1852
36.4.5 模数和 Montgomery 乘法 1852
36.4.6 模幂 1854
36.4.7 模逆 1855
36.4.8 模约简 1856
36.4.9 算术加法 1856
36.4.10 算术减法 1856
36.4.11 算术乘法 1857
36.4.12 算术比较 1857
36.4.13 RSA CRT 求幂 1858
36.4.14 检查点是否在椭圆曲线 Fp 上 1858
36.4.15 ECC Fp 标量乘法 1859

36.4.16 ECDSA 签名 1860
36.4.17 ECDSA 验证 1862
36.4.18 ECC 完整加法 1863
36.4.19 ECC 双基阶梯运算 1863
36.4.20 ECC 仿射投影运算 1864
36.5 配置和处理时间示例 1865
36.5.1 支持的椭圆曲线 1865
36.5.2 计算时间 1867
36.6 PKA 中断 1869
36.7 PKA 寄存器 1870
36.7.1 PKA 控制寄存器 (PKA_CR) 1870
36.7.2 PKA 状态寄存器 (PKA_SR) 1872
36.7.3 PKA 清零标志寄存器 (PKA_CLRFR) 1873
36.7.4 PKA RAM 1873
36.7.5 PKA 硬件配置寄存器 (PKA_HWCFGR) 1874
36.7.6 PKA 版本寄存器 (PKA_VERR) 1874
36.7.7 PKA 标识寄存器 (PKA_IPIDR) 1875
36.7.8 PKA 大小标识寄存器 (PKA_SIDR) 1875
36.7.9 PKA 寄存器映射 1875

37 DDR 内存加密引擎 (DDRMCE) 1877
37.1 简介 1877
37.2 MCE 主要特性 1877
37.3 MCE 实现 1877
37.4 MCE 功能描述 1878
37.4.1 MCE 框图 1878
37.4.2 MCE 内部信号 1878
37.4.3 MCE 编程 1879
37.4.4 MCE 复位和时钟 1879
37.4.5 MCE 块密码加密模式 1879
37.4.6 MCE AXI 通信管理 1880
37.4.7 MCE 加密禁止选项 1880
37.4.8 MCE 错误管理 1880
37.5 MCE 中断 1880
37.6 MCE 寄存器 1881
37.6.1 MCE 配置寄存器 (MCE_CR) 1881

37.6.2 MCE 状态寄存器 (MCE_SR) 1881
37.6.3 MCE 非法访问状态寄存器 (MCE_IASR) 1882
37.6.4 MCE 非法访问清零寄存器 (MCE_IACR) 1882
37.6.5 MCE 非法访问中断使能寄存器 (MCE_IAIER) 1883
37.6.6 MCE 特权配置寄存器 (MCE_PRIVCFGR) 1883
37.6.7 MCE 区域配置寄存器 (MCE_REGCR) 1884
37.6.8 MCE 起始地址寄存器 (MCE_SADDR) 1884
37.6.9 MCE 结束地址寄存器 (MCE_EADDR) 1885
37.6.10 MCE 主密钥 x (MCE_MKEYRx) 1885
37.6.11 MCE 版本寄存器 (MCE_HWCFGR3) 1886
37.6.12 MCE 版本寄存器 (MCE_HWCFGR2) 1886
37.6.13 MCE 版本寄存器 (MCE_HWCFGR1) 1887
37.6.14 MCE 版本寄存器 (MCE_VERR) 1888
37.6.15 MCE 标识寄存器 (MCE_IPIDR) 1888
37.6.16 MCE 大小 ID 寄存器 (MCE_SIDR) 1889
37.6.17 MCE 寄存器映射 1889

38 高级控制定时器 (TIM1/TIM8) 1891

38.1 TIM1/TIM8 简介 1891
38.2 TIM1/TIM8 主要特性 1891
38.3 TIM1/TIM8 功能描述 1893
38.3.1 时基单元 1893
38.3.2 计数器模式 1895
38.3.3 重复计数器 1905
38.3.4 外部触发输入 1907
38.3.5 时钟选择 1908
38.3.6 捕获/比较通道 1912
38.3.7 输入捕获模式 1914
38.3.8 PWM 输入模式 1915
38.3.9 强制输出模式 1916
38.3.10 输出比较模式 1916
38.3.11 PWM 模式 1918
38.3.12 不对称 PWM 模式 1921
38.3.13 组合 PWM 模式 1921
38.3.14 组合三相 PWM 模式 1922
38.3.15 互补输出和死区插入 1923
38.3.16 使用刹车功能 1925

38.3.17 双向刹车输入 1930
38.3.18 发生外部事件时清除 OCxREF 信号 1932
38.3.19 生成 6 步 PWM 1933
38.3.20 单脉冲模式 1934
38.3.21 可再触发单脉冲模式 1935
38.3.22 编码器接口模式 1936
38.3.23 UIF 位重映射 1938
38.3.24 定时器输入异或功能 1939
38.3.25 连接霍尔传感器 1939
38.3.26 定时器同步 1941
38.3.27 ADC 同步 1944
38.3.28 DMA 分组传送模式 1944
38.3.29 调试模式 1945
38.4 TIM1/TIM8 寄存器 1946
38.4.1 TIMx 控制寄存器 1 (TIMx_CR1)（x = 1、8） 1946
38.4.2 TIMx 控制寄存器 2 (TIMx_CR2)（x = 1、8） 1947
38.4.3 TIMx 从模式控制寄存器 (TIMx_SMCR)（x = 1、8） 1950
38.4.4 TIMx DMA/中断使能寄存器 (TIMx_DIER)（x = 1、8） 1952
38.4.5 TIMx 状态寄存器 (TIMx_SR)（x = 1、8） 1954
38.4.6 TIMx 事件生成寄存器 (TIMx_EGR)（x = 1、8） 1956
38.4.7 TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1)（x = 1、8） 1957
38.4.8 TIMx 捕获/比较模式寄存器 1 [复用] (TIMx_CCMR1)（x = 1、8） . . 1959
38.4.9 TIMx 捕获/比较模式寄存器 2 (TIMx_CCMR2)（x = 1、8） 1961
38.4.10 TIMx 捕获/比较模式寄存器 2 [复用] (TIMx_CCMR2)（x = 1、8） . . 1962
38.4.11 TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 1、8） 1964
38.4.12 TIMx 计数器 (TIMx_CNT)（x = 1、8） 1967
38.4.13 TIMx 预分频器 (TIMx_PSC)（x = 1、8） 1967
38.4.14 TIMx 自动重载寄存器 (TIMx_ARR)（x = 1、8） 1967
38.4.15 TIMx 重复计数器寄存器 (TIMx_RCR)（x = 1、8） 1968
38.4.16 TIMx 捕获/比较寄存器 1 (TIMx_CCR1)（x = 1、8） 1968
38.4.17 TIMx 捕获/比较寄存器 2 (TIMx_CCR2)（x = 1、8） 1969
38.4.18 TIMx 捕获/比较模式寄存器 3 (TIMx_CCR3)（x = 1、8） 1969
38.4.19 TIMx 捕获/比较寄存器 4 (TIMx_CCR4)（x = 1、8） 1970
38.4.20 TIMx 刹车和死区寄存器 (TIMx_BDTR)（x = 1、8） 1970
38.4.21 TIMxDMA 控制寄存器 (TIMx_DCR)（x = 1、8） 1974
38.4.22 TIMx 全传输 DMA 地址 (TIMx_DMAR)（x = 1、8） 1975
38.4.23 TIMx 捕获/比较模式寄存器 3 (TIMx_CCMR3)（x = 1、8） 1975

38.4.24 TIMx 捕获/比较寄存器 5 (TIMx_CCR5)（x = 1、8） 1976
38.4.25 TIMx 捕获/比较寄存器 6 (TIMx_CCR6)（x = 1、8） 1977
38.4.26 TIM1 复用功能选项寄存器 1 (TIM1_AF1) 1977
38.4.27 TIM1 复用功能寄存器 2 (TIM1_AF2) 1978
38.4.28 TIM8 复用功能选项寄存器 1 (TIM8_AF1) 1979
38.4.29 TIM8 复用功能选项寄存器 2 (TIM8_AF2) 1980
38.4.30 TIM1 定时器输入选择寄存器 (TIM1_TISEL) 1981
38.4.31 TIM8 定时器输入选择寄存器 (TIM8_TISEL) 1982
38.4.32 TIM1 寄存器映射 1983
38.4.33 TIM8 寄存器映射 1985

39 通用定时器 (TIM2/TIM3/TIM4/TIM5) 1988

39.1 TIM2/TIM3/TIM4/TIM5 简介 1988
39.2 TIM2/TIM3/TIM4/TIM5 主要特性 1988
39.3 TIM2/TIM3/TIM4/TIM5 功能描述 1990
39.3.1 时基单元 1990
39.3.2 计数器模式 1992
39.3.3 时钟选择 2002
39.3.4 捕获/比较通道 2005
39.3.5 输入捕获模式 2007
39.3.6 PWM 输入模式 2008
39.3.7 强制输出模式 2009
39.3.8 输出比较模式 2009
39.3.9 PWM 模式 2010
39.3.10 不对称 PWM 模式 2014
39.3.11 组合 PWM 模式 2015
39.3.12 发生外部事件时清除 OCxREF 信号 2016
39.3.13 单脉冲模式 2017
39.3.14 可再触发单脉冲模式 2018
39.3.15 编码器接口模式 2019
39.3.16 UIF 位重映射 2021
39.3.17 定时器输入异或功能 2021
39.3.18 定时器与外部触发同步 2022
39.3.19 定时器同步 2025
39.3.20 DMA 分组传送模式 2030
39.3.21 调试模式 2031
39.4 TIM2/TIM3/TIM4/TIM5 寄存器 2032

39.4.1 TIMx 控制寄存器 1 (TIMx_CR1)（x = 2 到 5） 2032
39.4.2 TIMx 控制寄存器 2 (TIMx_CR2)（x = 2 到 5） 2033
39.4.3 TIMx 从模式控制寄存器 (TIMx_SMCR)（x = 2 到 5） 2034
39.4.4 TIMx DMA/中断使能寄存器 (TIMx_DIER)（x = 2 到 5） 2037
39.4.5 TIMx 状态寄存器 (TIMx_SR)（x = 2 到 5） 2038
39.4.6 TIMx 事件生成寄存器 (TIMx_EGR)（x = 2 到 5） 2039
39.4.7 TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1)（x = 2 到 5） 2040
39.4.8 TIMx 捕获/比较模式寄存器 1 [复用] (TIMx_CCMR1)（x = 2 到 5） . . . 2042
39.4.9 TIMx 捕获/比较模式寄存器 2 (TIMx_CCMR2)（x = 2 到 5） 2044
39.4.10 TIMx 捕获/比较模式寄存器 2 [复用] (TIMx_CCMR2)（x = 2 到 5） . . . 2045
39.4.11 TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 2 到 5） 2046
39.4.12 TIMx 计数器 (TIMx_CNT)（x = 2 到 5） 2047
39.4.13 TIMx 计数器 [复用] (TIMx_CNT)（x = 2 到 5） 2048
39.4.14 TIMx 预分频器 (TIMx_PSC)（x = 2 到 5） 2048
39.4.15 TIMx 自动重载寄存器 (TIMx_ARR)（x = 2 到 5） 2048
39.4.16 TIMx 捕获/比较寄存器 1 (TIMx_CCR1)（x = 2 到 5） 2049
39.4.17 TIMx 捕获/比较寄存器 2 (TIMx_CCR2)（x = 2 到 5） 2049
39.4.18 TIMx 捕获/比较寄存器 3 (TIMx_CCR3)（x = 2 到 5） 2050
39.4.19 TIMx 捕获/比较寄存器 4 (TIMx_CCR4)（x = 2 到 5） 2050
39.4.20 TIMx DMA 控制寄存器 (TIMx_DCR)（x = 2 到 5） 2051
39.4.21 TIMx 全传输 DMA 地址 (TIMx_DMAR)（x = 2 到 5） 2052
39.4.22 TIM2 复用功能选项寄存器 1 (TIM2_AF1) 2052
39.4.23 TIM3 复用功能选项寄存器 1 (TIM3_AF1) 2053
39.4.24 TIM4 复用功能选项寄存器 1 (TIM4_AF1) 2053
39.4.25 TIM5 复用功能选项寄存器 1 (TIM5_AF1) 2054
39.4.26 TIM2 定时器输入选择寄存器 (TIM2_TISEL) 2054
39.4.27 TIM3 定时器输入选择寄存器 (TIM3_TISEL) 2055
39.4.28 TIM4 定时器输入选择寄存器 (TIM4_TISEL) 2056
39.4.29 TIM5 定时器输入选择寄存器 (TIM5_TISEL) 2056
39.4.30 TIMx 寄存器映射 2058

40 通用定时器 (TIM12/TIM13/TIM14) 2061

40.1 TIM12/TIM13/TIM14 简介 2061
40.2 TIM12/TIM13/TIM14 主要特性 2061
40.2.1 TIM12 主要特性 2061
40.2.2 TIM13/TIM14 主要特性 2062
40.3 TIM12/TIM13/TIM14 功能描述 2064

40.3.1 时基单元 2064
40.3.2 计数器模式 2066
40.3.3 时钟选择 2069
40.3.4 捕获/比较通道 2071
40.3.5 输入捕获模式 2073
40.3.6 PWM 输入模式（仅适用于 TIM12） 2073
40.3.7 强制输出模式 2074
40.3.8 输出比较模式 2075
40.3.9 PWM 模式 2076
40.3.10 组合 PWM 模式（仅适用于 TIM12） 2077
40.3.11 单脉冲模式 2078
40.3.12 可再触发单脉冲模式（仅限 TIM12） 2080
40.3.13 UIF 位重映射 2080
40.3.14 定时器输入异或功能 2081
40.3.15 TIM12 外部触发同步 2081
40.3.16 从模式——组合复位 + 触发模式 2084
40.3.17 定时器同步 (TIM12) 2084
40.3.18 使用定时器输出作为其他定时器的触发 (TIM13/TIM14) 2084
40.3.19 调试模式 2084
40.4 TIM12 寄存器 2084
40.4.1 TIM12 控制寄存器 1 (TIM12_CR1) 2084
40.4.2 TIM12 控制寄存器 2 (TIM12_CR2) 2086
40.4.3 TIM12 从模式控制寄存器 (TIM12_SMCR) 2086
40.4.4 中断使能寄存器 (TIM12_DIER) 2088
40.4.5 TIM12 状态寄存器 (TIM12_SR) 2088
40.4.6 TIM12 事件生成寄存器 (TIM12_EGR) 2089
40.4.7 TIM12 捕获/比较模式寄存器 1 (TIM12_CCMR1) 2090
40.4.8 TIM12 捕获/比较模式寄存器 1 [复用] (TIM12_CCMR1) 2092
40.4.9 TIM12 捕获/比较使能寄存器 (TIM12_CCER) 2094
40.4.10 TIM12 计数器 (TIM12_CNT) 2095
40.4.11 TIM12 预分频器 (TIM12_PSC) 2096
40.4.12 TIM12 自动重载寄存器 (TIM12_ARR) 2096
40.4.13 TIM12 捕获/比较寄存器 1 (TIM12_CCR1) 2096
40.4.14 TIM12 捕获/比较寄存器 2 (TIM12_CCR2) 2097
40.4.15 TIM12 定时器输入选择寄存器 (TIM12_TISEL) 2097
40.4.16 TIM12 寄存器映射 2098
40.5 TIM13/TIM14 寄存器 2100

40.5.1 TIMx 控制寄存器 1 (TIMx_CR1)（x = 13 到 14） 2100
40.5.2 TIMx 中断使能寄存器 (TIMx_DIER)（x = 13 到 14） 2101
40.5.3 TIMx 状态寄存器 (TIMx_SR)（x = 13 到 14） 2102
40.5.4 TIMx 事件生成寄存器 (TIMx_EGR)（x = 13 到 14） 2102
40.5.5 TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1)（x = 13 到 14） 2103
40.5.6 TIMx 捕获/比较模式寄存器 1 [复用] (TIMx_CCMR1)（x = 13 到 14） 2104
40.5.7 TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 13 到 14） 2106
40.5.8 TIMx 计数器 (TIMx_CNT)（x = 13 到 14） 2107
40.5.9 TIMx 预分频器 (TIMx_PSC)（x = 13 到 14） 2107
40.5.10 TIMx 自动重载寄存器 (TIMx_ARR)（x = 13 到 14） 2107
40.5.11 TIMx 捕获/比较寄存器 1 (TIMx_CCR1)（x = 13 到 14） 2108
40.5.12 TIM13 定时器输入选择寄存器 (TIM13_TISEL) 2108
40.5.13 TIM14 定时器输入选择寄存器 (TIM14_TISEL) 2109
40.5.14 TIM13/TIM14 寄存器映射 2109

41 通用定时器 (TIM15/TIM16/TIM17) 2111

41.1 TIM15/TIM16/TIM17 简介 2111
41.2 TIM15 主要特性 2111
41.3 TIM16/TIM17 主要特性 2112
41.4 TIM15/TIM16/TIM17 功能描述 2115
41.4.1 时基单元 2115
41.4.2 计数器模式 2117
41.4.3 重复计数器 2120
41.4.4 时钟选择 2121
41.4.5 捕获/比较通道 2123
41.4.6 输入捕获模式 2126
41.4.7 PWM 输入模式（仅适用于 TIM15） 2126
41.4.8 强制输出模式 2127
41.4.9 输出比较模式 2128
41.4.10 PWM 模式 2129
41.4.11 组合 PWM 模式（仅适用于 TIM15） 2130
41.4.12 互补输出和死区插入 2131
41.4.13 使用刹车功能 2133
41.4.14 双向刹车输入 2137
41.4.15 生成 6 步 PWM 2138
41.4.16 单脉冲模式 2140

41.4.17 可再触发单脉冲模式（仅限 TIM15） 2141
41.4.18 UIF 位重映射 2142
41.4.19 定时器输入异或功能（仅适用于 TIM15） 2142
41.4.20 外部触发同步（仅适用于 TIM15） 2142
41.4.21 从模式——组合复位 + 触发模式 2145
41.4.22 DMA 分组传送模式 2145
41.4.23 定时器同步 (TIM15) 2146
41.4.24 使用定时器输出作为其他定时器的触发 (TIM16/TIM17) 2146
41.4.25 调试模式 2146
41.5 TIM15 寄存器 2147
41.5.1 TIM15 控制寄存器 1 (TIM15_CR1) 2147
41.5.2 TIM15 控制寄存器 2 (TIM15_CR2) 2148
41.5.3 TIM15 从模式控制寄存器 (TIM15_SMCR) 2150
41.5.4 TIM15 DMA/中断使能寄存器 (TIM15_DIER) 2151
41.5.5 TIM15 状态寄存器 (TIM15_SR) 2152
41.5.6 TIM15 事件产生寄存器 (TIM15_EGR) 2154
41.5.7 TIM15 捕获/比较模式寄存器 1 (TIM15_CCMR1) 2155
41.5.8 TIM15 捕获/比较模式寄存器 1 [复用] (TIM15_CCMR1) 2156
41.5.9 TIM15 捕获/比较使能寄存器 (TIM15_CCER) 2158
41.5.10 TIM15 计数器 (TIM15_CNT) 2161
41.5.11 TIM15 预分频器 (TIM15_PSC) 2161
41.5.12 TIM15 自动重载寄存器 (TIM15_ARR) 2161
41.5.13 TIM15 重复计数器寄存器 (TIM15_RCR) 2162
41.5.14 TIM15 捕获/比较寄存器 1 (TIM15_CCR1) 2162
41.5.15 TIM15 捕获/比较寄存器 2 (TIM15_CCR2) 2163
41.5.16 TIM15 刹车和死区寄存器 (TIM15_BDTR) 2163
41.5.17 TIM15 DMA 控制寄存器 (TIM15_DCR) 2166
41.5.18 TIM15 全传输 DMA 地址 (TIM15_DMAR) 2166
41.5.19 TIM15 复用寄存器 1 (TIM15_AF1) 2167
41.5.20 TIM15 输入选择寄存器 (TIM15_TISEL) 2167
41.5.21 TIM15 寄存器映射 2168
41.6 TIM16/TIM17 寄存器 2171
41.6.1 TIMx 控制寄存器 1 (TIMx_CR1)（x = 16 到 17） 2171
41.6.2 TIMx 控制寄存器 2 (TIMx_CR2)（x = 16 到 17） 2172
41.6.3 TIMx DMA/中断使能寄存器 (TIMx_DIER)（x = 16 到 17） 2173
41.6.4 TIMx 状态寄存器 (TIMx_SR)（x = 16 到 17） 2174
41.6.5 TIMx 事件生成寄存器 (TIMx_EGR)（x = 16 到 17） 2175

41.6.6 TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1)（x = 16 到 17） 2175
41.6.7 TIMx 捕获/比较模式寄存器 1 [复用] (TIMx_CCMR1)（x = 16 到 17） 2176
41.6.8 TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 16 到 17） 2178
41.6.9 TIMx 计数器 (TIMx_CNT)（x = 16 到 17） 2180
41.6.10 TIMx 预分频器 (TIMx_PSC)（x = 16 到 17） 2181
41.6.11 TIMx 自动重载寄存器 (TIMx_ARR)（x = 16 到 17） 2181
41.6.12 TIMx 重复计数器寄存器 (TIMx_RCR)（x = 16 到 17） 2181
41.6.13 TIMx 捕获/比较寄存器 1 (TIMx_CCR1)（x = 16 到 17） 2182
41.6.14 TIMx 刹车和死区寄存器 (TIMx_BDTR)（x = 16 到 17） 2182
41.6.15 TIMx DMA 控制寄存器 (TIMx_DCR)（x = 16 到 17） 2185
41.6.16 TIMx 全传输 DMA 地址 (TIMx_DMAR)（x = 16 到 17） 2185
41.6.17 TIM16 复用功能寄存器 1 (TIM16_AF1) 2186
41.6.18 TIM16 输入选择寄存器 (TIM16_TISEL) 2187
41.6.19 TIM17 复用功能寄存器 1 (TIM17_AF1) 2187
41.6.20 TIM17 输入选择寄存器 (TIM17_TISEL) 2188
41.6.21 TIM16/TIM17 寄存器映射 2189

42 基本定时器 (TIM6/TIM7) 2191

42.1 TIM6/TIM7 简介 2191
42.2 TIM6/TIM7 主要特性 2191
42.3 TIM6/TIM7 功能描述 2192
42.3.1 时基单元 2192
42.3.2 计数模式 2194
42.3.3 UIF 位重映射 2197
42.3.4 时钟源 2197
42.3.5 调试模式 2198
42.4 TIM6/TIM7 寄存器 2198
42.4.1 TIMx 控制寄存器 1 (TIMx_CR1)（x = 6 到 7） 2198
42.4.2 TIMx 控制寄存器 2 (TIMx_CR2)（x = 6 到 7） 2200
42.4.3 TIMx DMA/中断使能寄存器 (TIMx_DIER)（x = 6 到 7） 2200
42.4.4 TIMx 状态寄存器 (TIMx_SR)（x = 6 到 7） 2201
42.4.5 TIMx 事件生成寄存器 (TIMx_EGR)（x = 6 到 7） 2201
42.4.6 TIMx 计数器 (TIMx_CNT)（x = 6 到 7） 2201
42.4.7 TIMx 预分频器 (TIMx_PSC)（x = 6 到 7） 2202
42.4.8 TIMx 自动重载寄存器 (TIMx_ARR)（x = 6 到 7） 2202
42.4.9 TIMx 寄存器映射 2203

43 低功耗定时器 (LPTIM) 2204
43.1 简介 2204
43.2 LPTIM 主要特性 2204
43.3 LPTIM 实现 2205
43.4 LPTIM 功能描述 2205
43.4.1 LPTIM 框图 2205
43.4.2 LPTIM 引脚和内部信号 2207
43.4.3 LPTIM 输入和触发映射 2207
43.4.4 LPTIM 复位和时钟 2210
43.4.5 干扰滤波器 2210
43.4.6 预分频器 2211
43.4.7 触发多路复用器 2212
43.4.8 工作模式 2212
43.4.9 超时功能 2214
43.4.10 生成波形 2214
43.4.11 寄存器更新 2215
43.4.12 计数器模式 2216
43.4.13 定时器使能 2216
43.4.14 定时器计数器复位 2216
43.4.15 编码器模式 2217
43.4.16 调试模式 2218
43.5 LPTIM 低功耗模式 2218
43.6 LPTIM 中断 2219
43.7 LPTIM 寄存器 2219
43.7.1 LPTIM 中断和状态寄存器 (LPTIM_ISR) 2220
43.7.2 LPTIM 中断清零寄存器 (LPTIM_ICR) 2221
43.7.3 LPTIM 中断使能寄存器 (LPTIM_IER) 2221
43.7.4 LPTIM 配置寄存器 (LPTIM_CFGR) 2222
43.7.5 LPTIM 控制寄存器 (LPTIM_CR) 2225
43.7.6 LPTIM 比较寄存器 (LPTIM_CMP) 2226
43.7.7 LPTIM 自动重载寄存器 (LPTIM_ARR) 2226
43.7.8 LPTIM 计数器寄存器 (LPTIM_CNT) 2227
43.7.9 LPTIM3 选项寄存器 (LPTIM3_OR) 2227
43.7.10 LPTIM 配置寄存器 2 (LPTIM_CFGR2) 2228
43.7.11 LPTIM x 外设硬件配置寄存器 (LPTIMx_HWCFGR)（x = 1 到 5） . . 2229
43.7.12 LPTIM 外设版本标识寄存器 (LPTIM_VERR) 2229

43.7.13 LPTIM 外设类型标识寄存器 (LPTIM_PIDR) 2230
43.7.14 LPTIM 寄存器映射大小标识寄存器 (LPTIM_SIDR) 2230
43.7.15 LPTIM 寄存器映射 2231

44 系统定时器生成器 (STGEN) 2233
44.1 STGEN 简介 2233
44.2 STGEN 主要特性 2233
44.3 STGEN 功能描述 2233
44.3.1 STGEN 框图 2234
44.4 STGEN 寄存器 2235
44.4.1 STGENC 控制寄存器 (STGENC_CNTCR) 2235
44.4.2 STGENC 状态寄存器 (STGENC_CNTSR) 2235
44.4.3 STGENC 值低位寄存器 (STGENC_CNTCVL) 2236
44.4.4 STGENC 值高位寄存器 (STGENC_CNTCVU) 2236
44.4.5 STGENC 基频寄存器 (STGENC_CNTFID0) 2237
44.4.6 STGENC 外设 ID4 寄存器 (STGENC_PIDR4) 2237
44.4.7 STGENC 外设 ID5 寄存器 (STGENC_PIDR5) 2238
44.4.8 STGENC 外设 ID6 寄存器 (STGENC_PIDR6) 2238
44.4.9 STGENC 外设 ID7 寄存器 (STGENC_PIDR7) 2238
44.4.10 STGENC 外设 ID0 寄存器 (STGENC_PIDR0) 2239
44.4.11 STGENC 外设 ID1 寄存器 (STGENC_PIDR1) 2239
44.4.12 STGENC 外设 ID2 寄存器 (STGENC_PIDR2) 2240
44.4.13 STGENC 外设 ID3 寄存器 (STGENC_PIDR3) 2240
44.4.14 STGENC 组件 ID0 寄存器 (STGENC_CIDR0) 2241
44.4.15 STGENC 组件 ID1 寄存器 (STGENC_CIDR1) 2241
44.4.16 STGENC 组件 ID2 寄存器 (STGENC_CIDR2) 2242
44.4.17 STGENC 组件 ID3 寄存器 (STGENC_CIDR3) 2242
44.4.18 STGENR 值低位寄存器 (STGENR_CNTCVL) 2243
44.4.19 STGENR 值高位寄存器 (STGENR_CNTCVU) 2243
44.4.20 STGENR 外设 ID4 寄存器 (STGENR_PIDR4) 2244
44.4.21 STGENR 外设 ID5 寄存器 (STGENR_PIDR5) 2244
44.4.22 STGENR 外设 ID6 寄存器 (STGENR_PIDR6) 2245
44.4.23 STGENR 外设 ID7 寄存器 (STGENR_PIDR7) 2245
44.4.24 STGENR 外设 ID0 寄存器 (STGENR_PIDR0) 2245
44.4.25 STGENR 外设 ID1 寄存器 (STGENR_PIDR1) 2246
44.4.26 STGENR 外设 ID2 寄存器 (STGENR_PIDR2) 2246
44.4.27 STGENR 外设 ID3 寄存器 (STGENR_PIDR3) 2247

44.4.28 STGENR 组件 ID0 寄存器 (STGENR_CIDR0) 2247
44.4.29 STGENR 组件 ID1 寄存器 (STGENR_CIDR1) 2248
44.4.30 STGENR 组件 ID2 寄存器 (STGENR_CIDR2) 2248
44.4.31 STGENR 组件 ID3 寄存器 (STGENR_CIDR3) 2249
44.4.32 STGEN 寄存器映射 2249

45 独立看门狗 (IWDG) 2252
45.1 简介 2252
45.2 IWDG 主要特性 2252
45.3 IWDG 实现 2252
45.4 IWDG 功能描述 2253
45.4.1 IWDG 框图 2253
45.4.2 IWDG 内部信号 2253
45.4.3 软件和硬件看门狗模式 2254
45.4.4 窗口选项 2255
45.4.5 调试 2256
45.4.6 寄存器访问保护 2256
45.5 IWDG 低功耗模式 2256
45.6 IWDG 中断 2257
45.7 IWDG 寄存器 2259
45.7.1 IWDG 密钥寄存器 (IWDG_KR) 2259
45.7.2 IWDG 预分频器寄存器 (IWDG_PR) 2260
45.7.3 IWDG 重载寄存器 (IWDG_RLR) 2260
45.7.4 IWDG 状态寄存器 (IWDG_SR) 2261
45.7.5 IWDG 窗口寄存器 (IWDG_WINR) 2262
45.7.6 IWDG 提前唤醒中断寄存器 (IWDG_EWCR) 2263
45.7.7 IWDG 硬件配置寄存器 (IWDG_HWCFGR) 2263
45.7.8 IWDG 版本寄存器 (IWDG_VERR) 2264
45.7.9 IWDG 标识寄存器 (IWDG_IDR) 2265
45.7.10 IWDG 尺寸标识寄存器 (IWDG_SIDR) 2265
45.7.11 IWDG 寄存器映射 2266

46 实时时钟 (RTC) 2267
46.1 简介 2267
46.2 RTC 主要特性 2267
46.3 RTC 功能描述 2267

46.3.1 RTC 框图 2267
46.3.2 RTC 引脚和内部信号 2269
46.3.3 RTC 和 TAMP 控制的 GPIO 2270
46.3.4 RTC 安全保护模式 2274
46.3.5 RTC 特权保护模式 2275
46.3.6 时钟和预分频器 2276
46.3.7 实时时钟和日历 2277
46.3.8 日历超低功耗模式 2278
46.3.9 可编程闹钟 2278
46.3.10 周期性自动唤醒 2278
46.3.11 RTC 初始化和配置 2279
46.3.12 读取日历 2281
46.3.13 复位 RTC 2282
46.3.14 RTC 同步 2282
46.3.15 RTC 参考时钟检测 2283
46.3.16 RTC 平滑数字校准 2283
46.3.17 时间戳功能 2285
46.3.18 校准时钟输出 2286
46.3.19 Tamper 和闹钟输出 2286
46.4 RTC 低功耗模式 2286
46.5 RTC 中断 2287
46.6 RTC 寄存器 2288
46.6.1 RTC 时间寄存器 (RTC_TR) 2288
46.6.2 RTC 日期寄存器 (RTC_DR) 2289
46.6.3 RTC 亚秒寄存器 (RTC_SSR) 2290
46.6.4 RTC 初始化控制和状态寄存器 (RTC_ICSR) 2291
46.6.5 RTC 预分频器寄存器 (RTC_PRER) 2293
46.6.6 RTC 唤醒定时器寄存器 (RTC_WUTR) 2294
46.6.7 RTC 控制寄存器 (RTC_CR) 2294
46.6.8 RTC 特权模式控制寄存器 (RTC_PRIVCFGR) 2298
46.6.9 RTC 安全配置寄存器 (RTC_SECCFGR) 2299
46.6.10 RTC 写保护寄存器 (RTC_WPR) 2300
46.6.11 RTC 校准寄存器 (RTC_CALR) 2301
46.6.12 RTC 移位控制寄存器 (RTC_SHIFTR) 2302
46.6.13 RTC 时间戳时间寄存器 (RTC_TSTR) 2303
46.6.14 RTC 时间戳日期寄存器 (RTC_TSDR) 2304
46.6.15 RTC 时间戳亚秒寄存器 (RTC_TSSSR) 2304

46.6.16 RTC 闹钟 A 寄存器 (RTC_ALRMAR) 2305
46.6.17 RTC 闹钟 A 亚秒寄存器 (RTC_ALRMASSR) 2306
46.6.18 RTC 闹钟 B 寄存器 (RTC_ALRMBR) 2307
46.6.19 RTC 闹钟 B 亚秒寄存器 (RTC_ALRMBSSR) 2308
46.6.20 RTC 状态寄存器 (RTC_SR) 2309
46.6.21 RTC 非安全屏蔽中断状态寄存器 (RTC_MISR) 2310
46.6.22 RTC 安全屏蔽中断状态寄存器 (RTC_SMISR) 2311
46.6.23 RTC 状态清零寄存器 (RTC_SCR) 2312
46.6.24 RTC 选项寄存器 (RTC_OR) 2313
46.6.25 RTC 闹钟 A 二进制模式寄存器 (RTC_ALRABINR) 2314
46.6.26 RTC 闹钟 B 二进制模式寄存器 (RTC_ALRBBINR) 2314
46.6.27 RTC 硬件配置寄存器 2 (RTC_HWCFGR2) 2315
46.6.28 RTC 硬件配置寄存器 (RTC_HWCFGR1) 2315
46.6.29 RTC 版本寄存器 (RTC_VERR) 2316
46.6.30 RTC 标识寄存器（RTC_IPIDR） 2317
46.6.31 RTC 大小标识寄存器 (RTC_SIDR) 2317
46.6.32 RTC 寄存器映射 2318

47 Tamper 和备份寄存器 (TAMP) 2321

47.1 简介 2321
47.2 TAMP 主要特性 2321
47.3 TAMP 功能描述 2322
47.3.1 TAMP 框图 2322
47.3.2 TAMP 引脚和内部信号 2323
47.3.3 RTC 和 TAMP 控制的 GPIO 2325
47.3.4 TAMP 寄存器写保护 2325
47.3.5 TAMP 安全保护模式 2325
47.3.6 备份寄存器保护区 2326
47.3.7 TAMP 特权保护模式 2326
47.3.8 启动硬件密钥 (BHK) 2327
47.3.9 Tamper 检测 2327
47.3.10 TAMP 备份寄存器和其他设备机密信息擦除 2327
47.3.11 Tamper 检测配置和初始化 2329
47.4 TAMP 低功耗模式 2334
47.5 TAMP 中断 2335
47.6 TAMP 寄存器 2336

47.6.1 TAMP 控制寄存器 1 (TAMP_CR1) 2336
47.6.2 TAMP 控制寄存器 2 (TAMP_CR2) 2338
47.6.3 TAMP 控制寄存器 3 (TAMP_CR3) 2341
47.6.4 TAMP 滤波器控制寄存器 (TAMP_FLTCR) 2342
47.6.5 TAMP 主动 Tamper 控制寄存器 1 (TAMP_ATCR1) 2343
47.6.6 TAMP 主动 Tamper 种子寄存器 (TAMP_ATSEEDR) 2346
47.6.7 TAMP 主动 Tamper 输出寄存器 (TAMP_ATOR) 2346
47.6.8 TAMP 主动 Tamper 控制寄存器 2 (TAMP_ATCR2) 2347
47.6.9 TAMP 安全配置寄存器（TAMP_SECCFGR） 2350
47.6.10 TAMP 特权配置寄存器 (TAMP_PRIVCFGR) 2351
47.6.11 TAMP 中断使能寄存器 (TAMP_IER) 2352
47.6.12 TAMP 状态寄存器 (TAMP_SR) 2354
47.6.13 TAMP 非安全屏蔽中断状态寄存器 (TAMP_MISR) 2356
47.6.14 TAMP 安全屏蔽中断状态寄存器 (TAMP_SMISR) 2358
47.6.15 TAMP 状态清零寄存器 (TAMP_SCR) 2360
47.6.16 TAMP 单调计数器 1 寄存器 (TAMP_COUNT1R) 2361
47.6.17 TAMP 单调计数器 2 寄存器 (TAMP_COUNT2R) 2362
47.6.18 TAMP 选项寄存器 (TAMP_OR) 2362
47.6.19 TAMP 擦除配置寄存器 (TAMP_ERCFGR) 2363
47.6.20 TAMP 备份 x 寄存器 (TAMP_BKPxR) 2364
47.6.21 TAMP 硬件配置寄存器 2 (TAMP_HWCFGR2) 2364
47.6.22 TAMP 硬件配置寄存器 1 (TAMP_HWCFGR1) 2365
47.6.23 TAMP 版本寄存器 (TAMP_VERR) 2366
47.6.24 TAMP 标识寄存器 (TAMP_IPIDR) 2366
47.6.25 TAMP 大小标识寄存器 (TAMP_SIDR) 2367
47.6.26 TAMP 寄存器映射 2368

48 内部集成电路 (I2C) 接口 2371

48.1 简介 2371
48.2 I2C 主要特性 2371
48.3 I2C 特性实现 2372
48.4 I2C 功能描述 2372
48.4.1 I2C 框图 2373
48.4.2 I2C 引脚和内部信号 2374
48.4.3 I2C 时钟要求 2374
48.4.4 模式选择 2374
48.4.5 I2C 初始化 2375

48.4.6 软件复位 2379
48.4.7 数据传输 2380
48.4.8 I2C 从模式 2382
48.4.9 I2C 主模式 2390
48.4.10 I2C_TIMINGR 寄存器配置示例 2400
48.4.11 SMBus 特性 2402
48.4.12 SMBus 初始化 2405
48.4.13 SMBus：I2C_TIMEOUTR 寄存器配置示例 2406
48.4.14 SMBus 从模式 2407
48.4.15 地址匹配时从 Stop 模式唤醒 2414
48.4.16 错误条件 2414
48.4.17 DMA 请求 2416
48.4.18 调试模式 2416
48.5 I2C 低功耗模式 2417
48.6 I2C 中断 2417
48.7 I2C 寄存器 2418
48.7.1 I2C 控制寄存器 1 (I2C_CR1) 2418
48.7.2 I2C 控制寄存器 2 (I2C_CR2) 2420
48.7.3 I2C 自身地址 1 寄存器 (I2C_OAR1) 2422
48.7.4 I2C 自身地址 2 寄存器 (I2C_OAR2) 2423
48.7.5 I2C 时序寄存器 (I2C_TIMINGR) 2424
48.7.6 I2C 超时寄存器 (I2C_TIMEOUTR) 2425
48.7.7 I2C 中断和状态寄存器 (I2C_ISR) 2426
48.7.8 I2C 中断清零寄存器 (I2C_ICR) 2428
48.7.9 I2C PEC 寄存器 (I2C_PECR) 2429
48.7.10 I2C 接收数据寄存器 (I2C_RXDR) 2430
48.7.11 I2C 发送数据寄存器 (I2C_TXDR) 2430
48.7.12 I2C 硬件配置寄存器 (I2C_HWCFGR) 2431
48.7.13 I2C 版本寄存器 (I2C_VERR) 2431
48.7.14 I2C 标识寄存器 (I2C_IPIDR) 2432
48.7.15 I2C 大小标识寄存器 (I2C_SIDR) 2432
48.7.16 I2C 寄存器映射 2433

49 通用同步/异步收发器 (USART/UART) 2435

49.1 USART 简介 2435
49.2 USART 主要特性 2435

49.3 USART 扩展特性 2436
49.4 USART 实现 2436
49.5 USART 功能描述 2438
49.5.1 USART 框图 2438
49.5.2 USART 信号 2439
49.5.3 USART 字符说明 2440
49.5.4 USART FIFO 和阈值 2442
49.5.5 USART 发送器 2442
49.5.6 USART 接收器 2445
49.5.7 USART 波特率生成 2452
49.5.8 USART 接收器对时钟偏差的容差 2453
49.5.9 USART 自动波特率检测 2454
49.5.10 USART 多处理器通信 2456
49.5.11 USART Modbus 通信 2457
49.5.12 USART 极性控制 2458
49.5.13 USART LIN（局域互连网络）模式 2459
49.5.14 USART 同步模式 2461
49.5.15 USART 单线半双工通信 2465
49.5.16 USART 接收器超时 2465
49.5.17 USART 智能卡模式 2466
49.5.18 USART IrDA SIR ENDEC 模块 2469
49.5.19 使用 USART 和 DMA 进行连续通信 2471
49.5.20 RS232 硬件流控制和 RS485 驱动器使能 2473
49.5.21 USART 低功耗管理 2476
49.6 处于低功耗模式的 USART 2479
49.7 USART 中断 2479
49.8 USART 寄存器 2480
49.8.1 USART 控制寄存器 1 (USART_CR1) 2480
49.8.2 USART 控制寄存器 1 [复用] (USART_CR1) 2484
49.8.3 USART 控制寄存器 2 (USART_CR2) 2487
49.8.4 USART 控制寄存器 3 (USART_CR3) 2491
49.8.5 USART 波特率寄存器 (USART_BRR) 2495
49.8.6 USART 保护时间和预分频器寄存器 (USART_GTPR) 2495
49.8.7 USART 接收器超时寄存器 (USART_RTOR) 2496
49.8.8 USART 请求寄存器 (USART_RQR) 2497
49.8.9 USART 中断和状态寄存器 (USART_ISR) 2498

49.8.10 USART 中断和状态寄存器 [复用] (USART_ISR) 2503
49.8.11 USART 中断标志清零寄存器 (USART_ICR) 2507
49.8.12 USART 接收数据寄存器 (USART_RDR) 2509
49.8.13 USART 发送数据寄存器 (USART_TDR) 2509
49.8.14 USART 预分频器寄存器 (USART_PRESC) 2510
49.8.15 USART 硬件配置寄存器 2 (USART_HWCFGR2) 2510
49.8.16 USART 硬件配置寄存器 1 (USART_HWCFGR1) 2511
49.8.17 USART 版本寄存器 (USART_VERR) 2512
49.8.18 USART 标识寄存器 (USART_IPIDR) 2512
49.8.19 USART 大小标识寄存器 (USART_SIDR) 2513
49.8.20 USART 寄存器映射 2514

50 串行外设接口 (SPI) 2516

50.1 简介 2516
50.2 SPI 主要特性 2516
50.3 SPI 实现 2517
50.4 SPI 功能描述 2517
50.4.1 SPI 框图 2517
50.4.2 SPI 信号 2518
50.4.3 SPI 通信一般情况 2519
50.4.4 一个主设备和一个从设备之间的通信 2519
50.4.5 标准多从设备通信 2521
50.4.6 多主设备通信 2524
50.4.7 从设备选择 (SS) 引脚管理 2524
50.4.8 通信格式 2528
50.4.9 SPI 配置 2530
50.4.10 使能 SPI 的步骤 2531
50.4.11 SPI 数据发送和接收过程 2531
50.4.12 禁止 SPI 的步骤 2535
50.4.13 数据封装 2536
50.4.14 使用 DMA（直接内存寻址）进行通信 2537
50.5 SPI 特定模式和控制 2539
50.5.1 TI 模式 2539
50.5.2 SPI 错误标志 2539
50.5.3 CRC 计算 2543
50.6 低功耗模式管理 2544

50.7 SPI 唤醒和中断 2546
50.8 I2S 主要特性 2547
50.9 I2S 功能描述 2548
50.9.1 I2S 一般描述 2548
50.9.2 与 SPI 功能共享的引脚 2548
50.9.3 I2S/PCM 模式下可用的位字段 2549
50.9.4 从模式和主模式 2550
50.9.5 支持的音频协议 2550
50.9.6 更灵活的串行接口 2556
50.9.7 启动序列 2558
50.9.8 停止序列 2560
50.9.9 时钟发生器 2560
50.9.10 内部 FIFO 2563
50.9.11 FIFO 状态标志 2564
50.9.12 下溢情况的处理 2564
50.9.13 上溢情况的处理 2565
50.9.14 帧错误检测 2566
50.9.15 DMA 接口 2568
50.9.16 编程示例 2568
50.10 I2S 唤醒和中断 2570
50.11 SPI/I2S 寄存器 2571
50.11.1 SPI/I2S 控制寄存器 1 (SPI_CR1) 2571
50.11.2 SPI 控制寄存器 2 (SPI_CR2) 2573
50.11.3 SPI 配置寄存器 1 (SPI_CFG1) 2573
50.11.4 SPI 配置寄存器 2 (SPI_CFG2) 2576
50.11.5 SPI/I2S 中断使能寄存器 (SPI_IER) 2578
50.11.6 SPI/I2S 状态寄存器 (SPI_SR) 2579
50.11.7 SPI/I2S 中断/状态标志清零寄存器 (SPI_IFCR) 2581
50.11.8 SPI/I2S 发送数据寄存器 (SPI_TXDR) 2582
50.11.9 SPI/I2S 接收数据寄存器 (SPI_RXDR) 2583
50.11.10 SPI 多项式寄存器 (SPI_CRCPOLY) 2583
50.11.11 SPI 发送器 CRC 寄存器 (SPI_TXCRC) 2584
50.11.12 SPI 接收器 CRC 寄存器 (SPI_RXCRC) 2584
50.11.13 SPI 下溢数据寄存器 (SPI_UDRDR) 2585
50.11.14 SPI/I2S 配置寄存器 (SPI_I2SCFGR) 2585
50.11.15 SPI/I2S 硬件配置寄存器 (SPI_SPI_I2S_HWCFGR) 2587

50.11.16 SPI/I2S 版本寄存器 (SPI_VERR) 2588
50.11.17 SPI/I2S 标识寄存器 (SPI_IPIDR) 2589
50.11.18 SPI/I2S 大小标识寄存器 (SPI_SIDR) 2589
50.12 SPI 寄存器映射和复位值 2590

51 串行音频接口 (SAI) 2592
51.1 简介 2592
51.2 SAI 主要特性 2592
51.3 SAI 实现 2593
51.4 SAI 功能描述 2593
51.4.1 SAI 框图 2593
51.4.2 SAI 引脚和内部信号 2595
51.4.3 SAI 的主要模式 2595
51.4.4 SAI 同步模式 2596
51.4.5 音频数据大小 2597
51.4.6 帧同步 2597
51.4.7 Slot 配置 2600
51.4.8 SAI 时钟发生器 2602
51.4.9 内部 FIFO 2605
51.4.10 PDM 接口 2607
51.4.11 AC’97 链路控制器 2615
51.4.12 SPDIF 输出 2617
51.4.13 特性 2619
51.4.14 错误标志 2623
51.4.15 禁止 SAI 2626
51.4.16 SAI DMA 接口 2626
51.5 SAI 中断 2627
51.6 SAI 寄存器 2628
51.6.1 全局配置寄存器 (SAI_GCR) 2628
51.6.2 SAI 配置寄存器 1 (SAI_ACR1) 2629
51.6.3 SAI 配置寄存器 1 (SAI_BCR1) 2631
51.6.4 SAI 配置寄存器 2 (SAI_ACR2) 2634
51.6.5 SAI 配置寄存器 2 (SAI_BCR2) 2635
51.6.6 SAI 帧配置寄存器 (SAI_AFRCR) 2637
51.6.7 帧配置寄存器 (SAI_BFRCR) 2638
51.6.8 SAI Slot 寄存器 (SAI_ASLOTR) 2639

51.6.9 SAI Slot 寄存器 (SAI_BSLOTR) 2640
51.6.10 SAI 中断屏蔽寄存器 (SAI_AIM) 2641
51.6.11 SAI 中断屏蔽寄存器 (SAI_BIM) 2643
51.6.12 SAI 状态寄存器 (SAI_ASR) 2644
51.6.13 SAI 状态寄存器 (SAI_BSR) 2646
51.6.14 SAI 清零标志寄存器 (SAI_ACLRFR) 2647
51.6.15 SAI 清零标志寄存器 (SAI_BCLRFR) 2648
51.6.16 SAI 数据寄存器 (SAI_ADR) 2649
51.6.17 SAI 数据寄存器 (SAI_BDR) 2650
51.6.18 SAI PDM 控制寄存器 (SAI_PDMCR) 2650
51.6.19 SAI PDM 延迟寄存器 (SAI_PDMDLY) 2652
51.6.20 SAI 硬件配置寄存器 (SAI_HWCFGR) 2654
51.6.21 SAI 版本寄存器 (SAI_VERR) 2655
51.6.22 SAI 标识寄存器 (SAI_IPIDR) 2655
51.6.23 SAI 存储空间标识寄存器 (SAI_SIDR) 2656
51.6.24 SAI 寄存器映射 2656

52 SPDIF 接收器接口 (SPDIFRX) 2658

52.1 SPDIFRX 接口简介 2658
52.2 SPDIFRX 主要特性 2658
52.3 SPDIFRX 功能描述 2658
52.3.1 SPDIFRX 引脚和内部信号 2659
52.3.2 S/PDIF 协议 (IEC-60958) 2660
52.3.3 SPDIFRX 解码器 (SPDIFRX_DC) 2662
52.3.4 SPDIFRX 对时钟偏差的容差 2665
52.3.5 SPDIFRX 同步 2665
52.3.6 SPDIFRX 处理 2667
52.3.7 数据接收管理 2669
52.3.8 专用控制流 2671
52.3.9 接收错误 2672
52.3.10 时钟策略 2674
52.3.11 符号时钟生成 2674
52.3.12 DMA 接口 2676
52.3.13 中断生成 2676
52.3.14 寄存器保护 2677
52.4 编程步骤 2678
52.4.1 初始化阶段 2678

52.4.2 处理来自 SPDIFRX 的中断 2679
52.4.3 处理来自 DMA 的中断 2679
52.5 SPDIFRX 接口寄存器 2680
52.5.1 SPDIFRX 控制寄存器 (SPDIFRX_CR) 2680
52.5.2 SPDIFRX 中断屏蔽寄存器 (SPDIFRX_IMR) 2682
52.5.3 SPDIFRX 状态寄存器 (SPDIFRX_SR) 2683
52.5.4 SPDIFRX 中断标志清零寄存器 (SPDIFRX_IFCR) 2685
52.5.5 SPDIFRX 数据输入寄存器 (SPDIFRX_FMT0_DR) 2685
52.5.6 SPDIFRX 数据输入寄存器 (SPDIFRX_FMT1_DR) 2686
52.5.7 SPDIFRX 数据输入寄存器 (SPDIFRX_FMT2_DR) 2687
52.5.8 SPDIFRX 通道状态寄存器 (SPDIFRX_CSR) 2687
52.5.9 SPDIFRX 调试信息寄存器 (SPDIFRX_DIR) 2688
52.5.10 SPDIFRX 版本寄存器 (SPDIFRX_VERR) 2689
52.5.11 SPDIFRX 标识寄存器 (SPDIFRX_IPIDR) 2689
52.5.12 SPDIFRX 大小标识寄存器 (SPDIFRX_SIDR) 2690
52.5.13 SPDIFRX 接口寄存器映射 2690

53 具有灵活数据速率的控制器局域网络 (FDCAN) 2692
53.1 简介 2692
53.2 FDCAN 主要特性 2695
53.3 FDCAN 实现 2695
53.4 FDCAN 功能描述 2696
53.4.1 工作模式 2697
53.4.2 消息 RAM 2705
53.4.3 FIFO 确认处理 2716
53.4.4 位时序 2716
53.4.5 CAN 时钟校准 2717
53.4.6 应用 2720
53.4.7 TTCAN 操作（仅限 FDCAN1） 2721
53.4.8 TTCAN 配置 2723
53.4.9 消息调度 2724
53.4.10 TTCAN 间隙控制 2730
53.4.11 秒表 2731
53.4.12 本地时间、周期时间、全局时间和外部时钟同步 2731
53.4.13 TTCAN 错误级别 2734
53.4.14 TTCAN 消息处理 2735
53.4.15 TTCAN 中断和错误处理 2737

53.4.16 0 级 2738
53.4.17 与外部时间调度同步 2740
53.4.18 FDCAN 接收缓冲区和 FIFO 元素 2741
53.4.19 FDCAN 发送缓冲区元素 2743
53.4.20 FDCAN 发送事件 FIFO 元素 2745
53.4.21 FDCAN 标准消息 ID 过滤器元素 2746
53.4.22 FDCAN 扩展消息 ID 过滤器元素 2748
53.4.23 FDCAN 触发内存元素 2749
53.5 FDCAN 寄存器 2751
53.5.1 FDCAN 核心版本寄存器 (FDCAN_CREL) 2751
53.5.2 FDCAN 字节序寄存器 (FDCAN_ENDN) 2751
53.5.3 FDCAN 数据位定时和预分频器寄存器 (FDCAN_DBTP) 2752
53.5.4 FDCAN 测试寄存器 (FDCAN_TEST) 2753
53.5.5 FDCAN RAM 看门狗寄存器 (FDCAN_RWD) 2753
53.5.6 FDCAN CC 控制寄存器 (FDCAN_CCCR) 2754
53.5.7 FDCAN 标称位定时和预分频器寄存器 (FDCAN_NBTP) 2756
53.5.8 FDCAN 时间戳计数器配置寄存器 (FDCAN_TSCC) 2757
53.5.9 FDCAN 时间戳计数器值寄存器 (FDCAN_TSCV) 2758
53.5.10 FDCAN 超时计数器配置寄存器 (FDCAN_TOCC) 2758
53.5.11 FDCAN 超时计数器值寄存器 (FDCAN_TOCV) 2759
53.5.12 FDCAN 错误计数器寄存器 (FDCAN_ECR) 2760
53.5.13 FDCAN 协议状态寄存器 (FDCAN_PSR) 2760
53.5.14 FDCAN 发送延迟补偿寄存器 (FDCAN_TDCR) 2762
53.5.15 FDCAN 中断寄存器 (FDCAN_IR) 2763
53.5.16 FDCAN 中断使能寄存器 (FDCAN_IE) 2766
53.5.17 FDCAN 中断线选择寄存器 (FDCAN_ILS) 2768
53.5.18 FDCAN 中断线使能寄存器 (FDCAN_ILE) 2769
53.5.19 FDCAN 全局过滤器配置寄存器 (FDCAN_GFC) 2770
53.5.20 FDCAN 标准 ID 过滤器配置寄存器 (FDCAN_SIDFC) 2771
53.5.21 FDCAN 扩展 ID 过滤器配置寄存器 (FDCAN_XIDFC) 2771
53.5.22 FDCAN 扩展 ID 和掩码寄存器 (FDCAN_XIDAM) 2772
53.5.23 FDCAN 高优先级消息状态寄存器 (FDCAN_HPMS) 2773
53.5.24 FDCAN 新数据 1 寄存器 (FDCAN_NDAT1) 2773
53.5.25 FDCAN 新数据 2 寄存器 (FDCAN_NDAT2) 2774
53.5.26 FDCAN 接收 FIFO 0 配置寄存器 (FDCAN_RXF0C) 2774
53.5.27 FDCAN 接收 FIFO 0 状态寄存器 (FDCAN_RXF0S) 2775
53.5.28 FDCAN 接收 FIFO 0 确认寄存器 (FDCAN_RXF0A) 2776

53.5.29 FDCAN 接收缓冲区配置寄存器 (FDCAN_RXBC) 2776
53.5.30 FDCAN 接收 FIFO 1 配置寄存器 (FDCAN_RXF1C) 2777
53.5.31 FDCAN 接收 FIFO 1 状态寄存器 (FDCAN_RXF1S) 2778
53.5.32 FDCAN 接收 FIFO 1 确认寄存器 (FDCAN_RXF1A) 2779
53.5.33 FDCAN 接收缓冲区元素大小配置寄存器 (FDCAN_RXESC) 2779
53.5.34 FDCAN 发送缓冲区配置寄存器 (FDCAN_TXBC) 2780
53.5.35 FDCAN 发送 FIFO/队列状态寄存器 (FDCAN_TXFQS) 2781
53.5.36 FDCAN 发送缓冲区元素大小配置寄存器 (FDCAN_TXESC) 2782
53.5.37 FDCAN 发送缓冲区请求挂起寄存器 (FDCAN_TXBRP) 2782
53.5.38 FDCAN 发送缓冲区添加请求寄存器 (FDCAN_TXBAR) 2783
53.5.39 FDCAN 发送缓冲区取消请求寄存器 (FDCAN_TXBCR) 2784
53.5.40 FDCAN 发送缓冲区发送已发生寄存器 (FDCAN_TXBTO) 2784
53.5.41 FDCAN 发送缓冲区取消完成寄存器 (FDCAN_TXBCF) 2785
53.5.42 FDCAN 发送缓冲区发送中断使能寄存器 (FDCAN_TXBTIE) 2785
53.5.43 FDCAN 发送缓冲区取消完成中断使能寄存器 (FDCAN_TXBCIE) . . . 2786
53.5.44 FDCAN 发送事件 FIFO 配置寄存器 (FDCAN_TXEFC) 2786
53.5.45 FDCAN 发送事件 FIFO 状态寄存器 (FDCAN_TXEFS) 2787
53.5.46 FDCAN 发送事件 FIFO 确认寄存器 (FDCAN_TXEFA) 2788
53.5.47 FDCAN 寄存器映射 2788
53.6 TTCAN 寄存器 2792
53.6.1 FDCAN TT 触发内存配置寄存器 (FDCAN_TTTMC) 2792
53.6.2 FDCAN TT 参考消息配置寄存器 (FDCAN_TTRMC) 2792
53.6.3 FDCAN TT 操作配置寄存器 (FDCAN_TTOCF) 2793
53.6.4 FDCAN TT 矩阵限值寄存器 (FDCAN_TTMLM) 2795
53.6.5 FDCAN TUR 配置寄存器 (FDCAN_TURCF) 2796
53.6.6 FDCAN TT 操作控制寄存器 (FDCAN_TTOCN) 2797
53.6.7 FDCAN TT 全局时间预设寄存器 (FDCAN_TTGTP) 2799
53.6.8 FDCAN TT 时间标记寄存器 (FDCAN_TTTMK) 2799
53.6.9 FDCAN TT 中断寄存器 (FDCAN_TTIR) 2800
53.6.10 FDCAN TT 中断使能寄存器 (FDCAN_TTIE) 2802
53.6.11 FDCAN TT 中断线选择寄存器 (FDCAN_TTILS) 2804
53.6.12 FDCAN TT 工作状态寄存器 (FDCAN_TTOST) 2805
53.6.13 FDCAN TUR 分子实际寄存器 (FDCAN_TURNA) 2807
53.6.14 FDCAN TT 本地和全局时间寄存器 (FDCAN_TTLGT) 2808
53.6.15 FDCAN TT 周期时间和计数寄存器 (FDCAN_TTCTC) 2808
53.6.16 FDCAN TT 捕获时间寄存器 (FDCAN_TTCPT) 2809
53.6.17 FDCAN TT 周期同步标记寄存器 (FDCAN_TTCSM) 2809

53.6.18 FDCAN TT 触发选择寄存器 (FDCAN_TTTS) 2810
53.6.19 FDCAN TT 寄存器映射 2810
53.7 CCU 寄存器 2812
53.7.1 时钟校准单元核心版本寄存器 (FDCAN_CCU_CREL) 2812
53.7.2 校准配置寄存器 (FDCAN_CCU_CCFG) 2812
53.7.3 校准状态寄存器 (FDCAN_CCU_CSTAT) 2814
53.7.4 校准看门狗寄存器 (FDCAN_CCU_CWD) 2814
53.7.5 时钟校准单元中断寄存器 (CCU_IR) 2815
53.7.6 时钟校准单元中断使能寄存器 (FDCAN_CCU_IE) 2816
53.7.7 CCU 寄存器映射 2816

54 USB on-the-go 高速 (OTG) 2818
54.1 简介 2818
54.2 OTG 主要特性 2819
54.2.1 通用特性 2819
54.2.2 主机模式特性 2820
54.2.3 从机模式特性 2820
54.3 OTG 实现 2820
54.4 OTG 功能描述 2821
54.4.1 OTG 框图 2821
54.4.2 OTG 引脚和内部信号 2821
54.4.3 OTG 内核 2822
54.4.4 OTG 检测 2822
54.4.5 连接到 OTG_HS 的高速 OTG PHY 2822
54.5 OTG 双角色设备 (DRD) 2823
54.5.1 ID 线检测 2823
54.5.2 HNP 双角色设备 2823
54.5.3 SRP 双角色设备 2824
54.6 OTG 作为 USB 设备 2824
54.6.1 支持 SRP 功能的 USB 设备 2825
54.6.2 USB 设备状态 2825
54.6.3 USB 设备端点 2826
54.7 OTG 作为 USB 主机 2827
54.7.1 支持 SRP 功能的 USB 主机 2828
54.7.2 USB 主机状态 2829
54.7.3 主机通道 2830

54.7.4 主机调度器 2831
54.8 OTG SOF 触发 2832
54.8.1 主机 SOF 2832
54.8.2 设备 SOF 2832
54.9 OTG 低功耗模式 2833
54.10 OTG 动态更新 OTG_HFIR 寄存器 2834
54.11 OTG 数据 FIFO 2834
54.11.1 设备 FIFO 架构 2835
54.11.2 主机 FIFO 架构 2836
54.11.3 FIFO RAM 分配 2837
54.12 **OTG** 中断 2838
54.13 **OTG** 控制和状态寄存器 2840
54.13.1 CSR 内存映射 2840
54.14 **OTG** 寄存器 2845
54.14.1 OTG 控制和状态寄存器 (OTG_GOTGCTL) 2845
54.14.2 OTG 中断寄存器 (OTG_GOTGINT) 2848
54.14.3 OTG AHB 配置寄存器 (OTG_GAHBCFG) 2849
54.14.4 OTG USB 配置寄存器 (OTG_GUSBCFG) 2850
54.14.5 OTG 复位寄存器 (OTG_GRSTCTL) 2852
54.14.6 OTG 模块中断寄存器 (OTG_GINTSTS) 2854
54.14.7 OTG 中断屏蔽寄存器 (OTG_GINTMSK) 2858
54.14.8 OTG 接收状态调试读取寄存器 (OTG_GRXSTSR) 2861
54.14.9 OTG 接收状态调试读取寄存器 [复用] (OTG_GRXSTSR) 2862
54.14.10 OTG 状态读取和弹出寄存器 (OTG_GRXSTSP) 2863
54.14.11 OTG 状态读取和弹出寄存器 [复用] (OTG_GRXSTSP) 2864
54.14.12 OTG 接收 FIFO 大小寄存器 (OTG_GRXFSIZ) 2865
54.14.13 OTG 主机非周期性发送 FIFO 大小寄存器 (OTG_HNPTXFSIZ)/ 端点 0 发送 FIFO 大小 (OTG_DIEPTXF0) 2866
54.14.14 OTG 非周期性发送 FIFO/队列状态寄存器 (OTG_HNPTXSTS) 2867
54.14.15 OTG 通用模块配置寄存器 (OTG_GCCFG) 2868
54.14.16 OTG 模块 ID 寄存器 (OTG_CID) 2869
54.14.17 OTG 模块 LPM 配置寄存器 (OTG_GLPMCFG) 2869
54.14.18 OTG 主机周期性发送 FIFO 大小寄存器 (OTG_HPTXFSIZ) 2873
54.14.19 OTG 设备 IN 端点发送 FIFO x 大小寄存器 (OTG_DIEPTXFx) 2873
54.14.20 主机模式寄存器 2874
54.14.21 OTG 主机配置寄存器 (OTG_HCFG) 2874

54.14.22 OTG 主机帧间隔寄存器 (OTG_HFIR) 2875
54.14.23 OTG 主机帧编号/帧剩余时间寄存器 (OTG_HFNUM) 2876
54.14.24 OTG_Host 周期性发送 FIFO/队列状态寄存器 (OTG_HPTXSTS) . . . 2876
54.14.25 OTG 主机全体通道中断寄存器 (OTG_HAINT) 2877
54.14.26 OTG 主机全体通道中断屏蔽寄存器 (OTG_HAINTMSK) 2878
54.14.27 OTG 主机帧列表基址寄存器 (OTG_HFLBADDR) 2878
54.14.28 OTG 主机端口控制和状态寄存器 (OTG_HPRT) 2879
54.14.29 OTG 主机通道 x 特性寄存器 (OTG_HCCHARx) 2881
54.14.30 OTG 主机通道 x 分离控制寄存器 (OTG_HCSPLTx) 2882
54.14.31 OTG 主机通道 x 中断寄存器 (OTG_HCINTx) 2883
54.14.32 OTG 主机通道 x 中断屏蔽寄存器 (OTG_HCINTMSKx) 2884
54.14.33 OTG 主机通道 x 传输大小寄存器 (OTG_HCTSIZx) 2886
54.14.34 OTG 主机通道 x 传输大小寄存器 (OTG_HCTSIZSGx) 2886
54.14.35 OTG 主机通道 x DMA 地址寄存器（缓冲 DMA 模式）[复用] (OTG_HCDMAx) 2888
54.14.36 OTG 主机通道 x DMA 地址寄存器（分散/收集 DMA）[复用] (OTG_HCDMASGx) 2888
54.14.37 OTG 主机通道 n DMA 地址缓冲区寄存器 (OTG_HCDMABx) 2889
54.14.38 从机模式寄存器 2890
54.14.39 OTG 设备配置寄存器 (OTG_DCFG) 2890
54.14.40 OTG 设备控制寄存器 (OTG_DCTL) 2891
54.14.41 OTG 设备状态寄存器 (OTG_DSTS) 2894
54.14.42 OTG 设备 IN 端点通用中断屏蔽寄存器 (OTG_DIEPMSK) 2895
54.14.43 OTG 设备 OUT 端点通用中断屏蔽寄存器 (OTG_DOEPMSK) 2896
54.14.44 OTG 设备全体端点中断寄存器 (OTG_DAINT) 2897
54.14.45 OTG 全体端点中断屏蔽寄存器 (OTG_DAINTMSK) 2898
54.14.46 OTG 设备 V_{BUS} 放电时间寄存器 (OTG_DVBUSDIS) 2899
54.14.47 OTG 设备 V_{BUS} 脉冲时间寄存器 (OTG_DVBUSPULSE) 2899
54.14.48 OTG 设备阈值控制寄存器 (OTG_DTHRCTL) 2900
54.14.49 OTG 设备 IN 端点 FIFO 空中断屏蔽寄存器 (OTG_DIEPEMPMSK) . . . 2901
54.14.50 OTG 设备 IN 端点 x 控制寄存器 (OTG_DIEPCTLx) 2901
54.14.51 OTG 设备 IN 端点 x 中断寄存器 (OTG_DIEPINTx) 2903
54.14.52 OTG 设备 IN 端点 0 传输大小寄存器 (OTG_DIEPTSIZ0) 2905
54.14.53 OTG 设备 IN 端点 x DMA 地址寄存器 (OTG_DIEPDMAx) 2905
54.14.54 OTG 设备 IN 端点发送 FIFO 状态寄存器 (OTG_DTXFSTSx) 2906
54.14.55 OTG 设备 IN 端点 x 传输大小寄存器 (OTG_DIEPTSIZx) 2906
54.14.56 OTG 设备控制 OUT 端点 0 控制寄存器 (OTG_DOEPCTL0) 2907
54.14.57 OTG 设备 OUT 端点 x 中断寄存器 (OTG_DOEPINTx) 2908

54.14.58 OTG 设备 OUT 端点 0 传输大小寄存器 (OTG_DOEPTSIZ0) 2910
54.14.59 OTG 设备 OUT 端点 x DMA 地址寄存器 (OTG_DOEPDMAx) 2911
54.14.60 OTG 设备 OUT 端点 x 控制寄存器 (OTG_DOEPCTLx) 2911
54.14.61 OTG 设备 OUT 端点 x 传输大小寄存器 (OTG_DOEPTSIZx) 2913
54.14.62 OTG 电源和时钟门控控制寄存器 (OTG_PCGCCTL) 2914
54.14.63 **OTG** 寄存器映射 2915
54.15 **OTG** 编程模型 2926
54.15.1 模块初始化 2926
54.15.2 主机初始化 2926
54.15.3 设备初始化 2927
54.15.4 DMA 模式 2927
54.15.5 主机编程模型 2928
54.15.6 设备编程模型 2957
54.15.7 最坏情况下的响应时间 2975
54.15.8 OTG 编程模型 2977

55 USB HS PHY 控制器 (USBPHYC) 2982
55.1 USBPHYC 简介 2982
55.2 USBPHYC 主要特性 2982
55.3 USBPHYC 功能描述 2982
55.3.1 USBPHYC 框图 2982
55.3.2 USBPHYC 复位和时钟 2983
55.3.3 PLL 控制 2984
55.4 USBPHYC 寄存器 2985
55.4.1 USBPHYC PLL 控制寄存器 (USBPHYC_PLL) 2985
55.4.2 USBPHYC 杂项控制寄存器 (USBPHYC_MISC) 2986
55.4.3 USBPHYC PHY x 监视寄存器 (USBPHYC_MONITORx) 2987
55.4.4 USBPHYC PHY x 调整寄存器 (USBPHYC_TUNEx) 2988
55.4.5 USBPHYC 版本寄存器 (USBPHYC_VERR) 2990
55.4.6 USBPHYC 寄存器映射 2991

56 USB_EHCI/USB_OHCI 高速/全速链路控制器 (USBH) 2993
56.1 简介 2993
56.2 USBH 主要特性 2993
56.3 USBH 功能描述 2994
56.3.1 USBH 框图 2994

56.3.2 USBH 复位和时钟 2994
56.3.3 USBH 数据包缓冲区描述 2995
56.3.4 USBH 根集线器的描述 2995
56.4 USBH 中断 2995
56.5 USBH 寄存器 2995
56.5.1 USBH OHCI 标准寄存器 2996
56.5.2 USBH OHCI 实现寄存器 #6 (USBH_OHCI_INSNREG06) 2997
56.5.3 USBH OHCI 实现寄存器 #7 (USBH_OHCI_INSNREG07) 2997
56.5.4 USBH EHCI 标准寄存器 2998
56.5.5 USBH EHCI 实现寄存器 #1 (USBH_EHCI_INSNREG01) 2998
56.5.6 USBH EHCI 实现寄存器 #2 (USBH_EHCI_INSNREG02) 2999
56.5.7 USBH EHCI 实现寄存器 #3 (USBH_EHCI_INSNREG03) 3000
56.5.8 USBH EHCI 实现寄存器 #4 (USBH_EHCI_INSNREG04) 3001
56.5.9 USBH EHCI 实现寄存器 #5 (USBH_EHCI_INSNREG05) 3002
56.5.10 USBH EHCI 实现寄存器 #6 (USBH_EHCI_INSNREG06) 3003
56.5.11 USBH EHCI 实现寄存器 #7 (USBH_EHCI_INSNREG07) 3003
56.5.12 USBH 寄存器映射 3004

57 以太网 (ETH)：带有 DMA 控制器的千兆介质访问控制 (GMAC) 3009
57.1 以太网简介 3009
57.2 以太网主要特性 3009
57.2.1 标准合规 3009
57.2.2 MAC 特性 3009
57.2.3 事务层 (MTL) 特性 3011
57.2.4 DMA 模块特性 3012
57.2.5 总线接口功能 3012
57.2.6 通用排队特性 3013
57.3 以太网引脚和内部信号 3013
57.4 以太网架构 3014
57.4.1 DMA 控制器 3015
57.4.2 MTL 3022
57.4.3 MAC 3022
57.4.4 多通道和多队列支持 3026
57.5 以太网功能描述：MAC 3029
57.5.1 双 VLAN 处理 3029
57.5.2 源地址和 VLAN 插入、替换或删除 3030

57.5.3 数据包过滤 . . . 3031
57.5.4 IEEE 1588 时间戳支持 . . . 3038
57.5.5 校验和减荷引擎 . . . 3059
57.5.6 TCP 分段减荷 . . . 3064
57.5.7 IPv4 ARP 减荷 . . . 3070
57.5.8 回环 . . . 3071
57.5.9 流控 . . . 3072
57.5.10 MAC 管理计数器 . . . 3074
57.5.11 MAC 生成的中断 . . . 3075
57.5.12 MAC 和 MMC 寄存器说明 . . . 3075
57.6 以太网功能描述：PHY 接口 . . . 3076
57.6.1 站管理代理 (SMA) . . . 3076
57.6.2 介质独立接口 (MII) . . . 3082
57.6.3 精简介质独立接口 (RMII) . . . 3083
57.6.4 精简千兆介质独立接口 (RGMII) . . . 3086
57.7 以太网低功耗模式 . . . 3089
57.7.1 低功耗管理 . . . 3089
57.7.2 节能型以太网 (EEE) . . . 3095
57.8 以太网中断 . . . 3100
57.8.1 DMA 中断 . . . 3100
57.8.2 MTL 中断 . . . 3101
57.8.3 MAC 中断 . . . 3101
57.9 以太网编程模型 . . . 3102
57.9.1 DMA 初始化 . . . 3102
57.9.2 MTL 初始化 . . . 3103
57.9.3 MAC 初始化 . . . 3104
57.9.4 执行正常接收和发送操作 . . . 3105
57.9.5 停止和开始发送 . . . 3105
57.9.6 关于在 RxDMA 中切换至新描述符列表的编程指南 . . . 3105
57.9.7 关于切换 AXI 时钟频率的编程指南 . . . 3106
57.9.8 关于多通道多队列操作的编程指南 . . . 3106
57.9.9 关于 GMII 链路状态转换的编程指南 . . . 3107
57.9.10 关于 IEEE 1588 时间戳的编程指南 . . . 3108
57.9.11 PTP 减荷功能的编程指南 . . . 3109
57.9.12 关于节能型以太网 (EEE) 的编程指南 . . . 3113
57.9.13 关于每秒脉冲数 (PPS) 灵活的输出的编程指南 . . . 3114
57.9.14 关于 TSO 的编程指南 . . . 3116

57.9.15 关于在接收端执行 VLAN 过滤的编程指南 3117
57.10 描述符 3117
57.10.1 描述符概述 3117
57.10.2 描述符结构 3118
57.10.3 发送描述符 3120
57.10.4 接收描述符 3130
57.11 以太网寄存器 3140
57.11.1 以太网寄存器映射 3140
57.11.2 以太网 DMA 寄存器 3141
57.11.3 以太网 MTL 寄存器 3168
57.11.4 以太网 MAC 和 MMC 寄存器 3182

58 硬件调试端口 (HDP) 3279
58.1 特性 3279
58.2 框图 3279
58.3 功能描述 3280
58.4 HDP 寄存器 3283
58.4.1 HDP 控制寄存器 (HDP_CTRL) 3283
58.4.2 HDP 复用器控制寄存器 (HDP_MUX) 3283
58.4.3 HDP 回读值寄存器 (HDP_VAL) 3284
58.4.4 HDP 通用输出设置寄存器 (HDP_GPOSET) 3284
58.4.5 HDP 通用输出清零寄存器 (HDP_GPOCLR) 3284
58.4.6 HDP 通用输出值寄存器 (HDP_GPOVAL) 3285
58.4.7 HDP 版本寄存器 (HDP_VERR) 3285
58.4.8 HDP IP 标识寄存器 (HDP_IPIDR) 3286
58.4.9 HDP 大小标识寄存器 (HDP_SIDR) 3286
58.4.10 HDP 寄存器映射 3287

59 调试支持 (DBG) 3288
59.1 简介 3288
59.2 调试用例 3289
59.3 DBG 框图 3290
59.4 DBG 引脚 3290
59.5 DBG 电源和时钟 3291
59.5.1 DBG 电源域 3291
59.5.2 DBG 时钟 3291

59.5.3 调试和低功耗模式 . . . 3292
59.5.4 DBG 复位 . . . 3292
59.6 安全性 . . . 3292
59.6.1 认证信号 . . . 3292
59.6.2 JTAG/SWD Tamper 检测 . . . 3293
59.7 芯片级 TAP 控制器 (CLTAPC) . . . 3294
59.8 串行线和 JTAG 调试端口 (SWJ-DP) . . . 3294
59.8.1 JTAG 调试端口 . . . 3295
59.8.2 串行线调试端口 . . . 3297
59.8.3 调试端口寄存器 . . . 3298
59.9 访问端口 . . . 3307
59.9.1 认证 . . . 3308
59.9.2 MEM-AP 寄存器 . . . 3308
59.10 系统调试功能 . . . 3317
59.10.1 系统 ROM 表 . . . 3317
59.10.2 CoreSight™ 全局时间戳发生器 (TSGEN) . . . 3326
59.10.3 交叉触发接口 (CTI) 和矩阵 (CTM) . . . 3334
59.10.4 嵌入式跟踪 FIFO (ETF) . . . 3354
59.10.5 跟踪端口接口单元 (TPIU) . . . 3375
59.10.6 微处理器调试单元 (DBGMCU) . . . 3393
59.11 Cortex®-A7 调试功能 . . . 3405
59.11.1 Cortex®-A7 调试单元 . . . 3405
59.11.2 Cortex®-A7 性能监视单元 (PMU) . . . 3448
59.11.3 嵌入式跟踪宏单元 (ETM) . . . 3469
59.11.4 交叉触发接口 (CTI) . . . 3521
59.12 参考资料 . . . 3522

60 设备电子签名 . . . 3523
60.1 唯一设备 ID 寄存器（96 位）(UID) . . . 3523
60.2 设备产品编号 (RPN) . . . 3524
60.3 设备版本 . . . 3525

61 重要安全说明 . . . 3526

62 版本历史 . . . 3527

表格索引

表 1. 内存映射概述STM32MP13xx . . . 115
表 2. 主端口的主要特性 . . . 117
表 3. 主端口安全特性 . . . 118
表 4. 从端口主要特性 . . . 119
表 5. 从端口安全特性 . . . 120
表 6. 多层 AHB 主端口的主要特性 . . . 121
表 7. 多层 AHB 从端口主要特性 . . . 122
表 8. AXIMC 寄存器映射和复位值 . . . 138
表 9. 寄存器边界地址 . . . 149
表 10. 调试器寄存器边界地址 . . . 157
表 11. OTP 模式定义 . . . 162
表 12. OTP 永久写入锁定 . . . 165
表 13. 复位后的 BSEC_DENABLE 默认值 . . . 166
表 14. 写入访问权限 . . . 167
表 15. 读取访问权限 . . . 167
表 16. BSEC 寄存器映射和复位值 . . . 181
表 17. OTP 列表字 0 到 31 . . . 185
表 18. OTP 列表字 32 到 95 . . . 194
表 19. DDRCTRL 寄存器映射和复位值 . . . 290
表 20. NSAID 和 AXI _ID 映射 . . . 307
表 21. TZC 寄存器映射和复位值 . . . 328
表 22. DDRPHYC I/O 列表 . . . 335
表 23. MSDLL 控制位 . . . 338
表 24. PHY 延迟 . . . 357
表 25. DFI 时序参数 . . . 359
表 26. DDRPHYC 寄存器映射和复位值 . . . 401
表 27. DDRPERFM 信号组 . . . 407
表 28. 电源控制寄存器映射和复位值 . . . 416
表 29. PWR 引脚概述 . . . 419
表 30. 存在 VDD 时 VSW 域的功能 . . . 425
表 31. 存在 VBAT 时 VSW 域的功能 . . . 425
表 32. MPU 低功耗模式汇总 . . . 438
表 33. PWR 系统模式汇总 . . . 439
表 34. 系统低功耗模式汇总 . . . 439
表 35. 功能取决于系统工作模式 . . . 440
表 36. PDDS 低功耗模式控制 . . . 443
表 37. 低功耗模式退出标志 . . . 443
表 38. MPU CSleep . . . 445
表 39. MPU CStop . . . 446
表 40. 系统停止 . . . 447
表 41. 系统 LP-Stop . . . 448
表 42. 系统 LPLV-Stop . . . 450
表 43. 系统 LPLV-Stop2 . . . 451
表 44. 待机 . . . 454
表 45. FIFO 状态标志 . . . 459
表 46. 寄存器安全性概述 . . . 460
表 47. 电源控制寄存器映射和复位值 . . . 472
表 48. 引脚概述 . . . 476

表 49. 复位覆盖范围汇总 . . . 485
表 50. 具有特定复位和特定电压的 RCC 寄存器 . . . 486
表 51. 复位源标识（RCC_MP/BR_RSTSCLRR 和 PWR） . . . 487
表 52. 振荡器状态与系统模式 . . . 492
表 53. HSI、CSI 和 HSE 状态与系统模式 . . . 513
表 54. 在时钟恢复序列期间可能重新配置的资源 . . . 518
表 55. 外设时钟分配概述 . . . 520
表 56. 内核时钟分配与外设 . . . 526
表 57. 音频应用的内核时钟设置示例 . . . 527
表 58. 时钟定时器与 pclk 之比 . . . 533
表 59. aclk_1 时钟的门控控制 . . . 547
表 60. DDRCKMOD 控制描述 . . . 548
表 61. DDRC 低功耗 (DDRC-LP) 状态描述 . . . 549
表 62. 推荐配置 . . . 554
表 63. MPU 的外设时钟使能详细信息 . . . 559
表 64. MPU 模式详细信息 . . . 561
表 65. AXIM 的总线时钟使能详细信息 . . . 562
表 66. AHB5 和 AHB6 的总线时钟使能详细信息 . . . 563
表 67. MLAHB 和 AHB4 的总线时钟使能详细信息 . . . 563
表 68. AHB2 和 AHB1 的总线时钟使能详细信息 . . . 564
表 69. APB4 和 APB[2:1] 的总线时钟使能详细信息 . . . 564
表 70. RCC 安全性和外设/内存 . . . 567
表 71. 中断源和控制 . . . 571
表 72. MPU 的时钟设置示例 . . . 579
表 73. AXI、DDR 和 DCMIPP 的时钟设置示例 . . . 580
表 74. RCC 寄存器映射和复位值 . . . 882
表 75. 端口位配置表 . . . 911
表 76. GPIO 安全位 . . . 918
表 77. GPIO 寄存器映射和复位值 . . . 933
表 78. 模拟开关供电电压设置 . . . 937
表 79. 高速低电压接口 . . . 938
表 80. SYSCFG 接口寄存器映射和复位值 . . . 959
表 81. STM32MP13xx 安全和受控安全外设 . . . 965
表 82. STM32MP13xx TrustZone 感知外设 . . . 966
表 83. STM32MP13xx 内存防火墙 . . . 967
表 84. STM32MP13xx 支持 TrustZone 的总线主设备 . . . 967
表 85. DECPROT 分配 . . . 967
表 86. ETZPC 寄存器映射和复位值 . . . 978
表 87. 外设互连矩阵 . . . 981
表 88. 外设互连矩阵详细信息 . . . 982
表 89. MDMA 连接 . . . 992
表 90. MDMA 内部输入/输出信号 . . . 995
表 91. MDMA 中断请求 . . . 1001
表 92. MDMA 寄存器映射和复位值 . . . 1016
表 93. 源和目标地址 . . . 1021
表 94. 双缓冲区模式下的源和目标地址寄存器 (DBM=1) . . . 1026
表 95. 封装/解封和字节序行为（位 PINC = MINC = 1） . . . 1027
表 96. PSIZE 与 MSIZE 确定时对 NDT 的限制条件 . . . 1027
表 97. FIFO 阈值配置 . . . 1030
表 98. 可能的 DMA 配置 . . . 1034
表 99. DMA 中断请求 . . . 1036
表 100. DMA 寄存器映射和复位值 . . . 1050

表 101. DMAMUX1 和 DMAMUX2 实例化 1056
表 102. DMAMUX1：复用器输入到资源的分配 1056
表 103. DMAMUX1：触发输入到资源的分配 1057
表 104. DMAMUX1：同步输入到资源的分配 1058
表 105. DMAMUX2：复用器输入到资源的分配 1058
表 106. DMAMUX2：触发输入到资源的分配 1059
表 107. DMAMUX2：同步输入到资源的分配 1059
表 108. DMAMUX 信号 1061
表 109. DMAMUX 中断 1065
表 110. DMAMUX 寄存器映射和复位值 1073
表 111. 基于 STM32MP13xx 上 Cortex®-A7 的 GIC 中断映射 1076
表 112. STM32MP13xx EXTI 事件 1081
表 113. GICD 寄存器映射和复位值 1107
表 114. GICC 寄存器映射和复位值 1119
表 115. GICH 寄存器映射和复位值 1126
表 116. GICV 寄存器映射和复位值 1134
表 117. EXTI 引脚概述 1138
表 118. EVG 引脚概述 1139
表 119. EXTI 事件输入配置和寄存器控制 1140
表 120. 屏蔽功能 1143
表 121. 寄存器安全性概述 1144
表 122. EXTI 寄存器映射的各个部分 1145
表 123. EXTI 寄存器映射和复位值 1168
表 124. CRC 内部输入/输出信号 1173
表 125. CRC 寄存器映射和复位值 1180
表 126. FMC 特性 1182
表 127. FMC 内部信号 1183
表 128. FMC 存储区域（默认映射） 1185
表 129. NOR/PSRAM 子区域选择 1185
表 130. NOR/PSRAM 外部内存地址 1185
表 131. NAND 存储映射和时序寄存器 1186
表 132. NAND Flash 区域选择 1186
表 133. NAND 芯片使能信号选择 1186
表 134. NOR/PSRAM 的可编程访问参数 1188
表 135. 非复用 I/O NOR Flash 1188
表 136. 16 位复用 I/O NOR Flash 1189
表 137. 非复用 I/O PSRAM/SRAM 1189
表 138. 16 位复用 I/O PSRAM 1189
表 139. NOR Flash/PSRAM：支持的内存和事务示例 1190
表 140. FMC_BCRx 位字段（模式 1） 1194
表 141. FMC_BTRx 位字段（模式 1） 1194
表 142. FMC_BCRx 位字段（模式 A） 1196
表 143. FMC_BTRx 位字段（模式 A） 1197
表 144. FMC_BWTRx 位字段（模式 A） 1197
表 145. FMC_BCRx 位字段（模式 2/B） 1199
表 146. FMC_BTRx 位字段（模式 2/B） 1200
表 147. FMC_BWTRx 位字段（模式 2/B） 1200
表 148. FMC_BCRx 位字段（模式 C） 1202
表 149. FMC_BTRx 位字段（模式 C） 1202
表 150. FMC_BWTRx 位字段（模式 C） 1203
表 151. FMC_BCRx 位字段（模式 D） 1204
表 152. FMC_BTRx 位字段（模式 D） 1205

表 153. FMC_BWTRx 位字段（模式 D） 1205
表 154. FMC_BCRx 位字段（复用模式） 1207
表 155. FMC_BTRx 位字段（复用模式） 1208
表 156. FMC_BCRx 位字段（同步复用读取模式） 1213
表 157. FMC_BTRx 位字段（同步复用读取模式） 1214
表 158. FMC_BCRx 位字段（同步复用写入模式） 1215
表 159. FMC_BTRx 位字段（同步复用写入模式） 1216
表 160. 可编程的 NAND Flash 访问参数 1228
表 161. 8 位 NAND Flash 1228
表 162. 16 位 NAND Flash 1229
表 163. 支持的内存和事务 1229
表 164. 每个扇区的 ECC 奇偶校验字节数 1234
表 165. FMC NAND 控制器中断请求 1240
表 166. HPR 相关位 1246
表 167. ECC 结果相关位 1247
表 168. FMC 寄存器映射 1265
表 169. QUADSPI 内部信号 1270
表 170. QUADSPI 引脚 1270
表 171. QUADSPI 中断请求 1282
表 172. QUADSPI 寄存器映射和复位值 1296
表 173. SDMMC 特性 1298
表 174. SDMMC 工作模式 SD 和 SDIO 1301
表 175. SDMMC 工作模式 e•MMC 1301
表 176. SDMMC 内部输入/输出信号 1302
表 177. SDMMC 引脚 1303
表 178. SDMMC 命令和数据相位选择 1304
表 179. 命令令牌格式 1309
表 180. 带 CRC 的短响应令牌格式 1310
表 181. 不带 CRC 的短响应令牌格式 1310
表 182. 带 CRC 的长响应令牌格式 1310
表 183. 特定命令概述 1311
表 184. 命令路径状态标志 1311
表 185. 命令路径错误处理 1312
表 186. 数据令牌格式 1318
表 187. 数据路径状态标志和清零位 1319
表 188. 数据路径错误处理 1320
表 189. 数据 FIFO 访问 1321
表 190. 发送 FIFO 状态标志 1321
表 191. 接收 FIFO 状态标志 1322
表 192. AHB 和 SDMMC_CK 时钟频率的关系 1327
表 193. SDIO 特殊操作控制 1327
表 194. 4 位模式启动、中断和 CRC 状态信号检测 1331
表 195. CMD12 用例 1335
表 196. SDMMC 中断 1348
表 197. 响应类型和 SDMMC_RESPxR 寄存器 1355
表 198. SDMMC 寄存器映射 1371
表 199. DLYB 内部输入/输出信号 1375
表 200. DLYB 互连 1375
表 201. 延迟模块控制 1375
表 202. DLYB 寄存器映射和复位值 1378
表 203. ADC 特性 1381
表 204. ADC 输入/输出引脚 1383

表 205. ADC 内部输入/输出信号 . . . 1383
表 206. ADC 互连 . . . 1383
表 207. 为常规外部触发配置触发极性 . . . 1402
表 208. 为注入外部触发配置触发极性 . . . 1402
表 209. TSAR 与分辨率的对应关系 . . . 1414
表 210. 偏移计算与数据分辨率 . . . 1417
表 211. 模拟看门狗通道选择 . . . 1428
表 212. 模拟看门狗 1 比较 . . . 1429
表 213. 模拟看门狗 2 和 3 比较 . . . 1429
表 214. 最大输出结果与 N 和 M 的对应关系（灰色单元格表示截断的部分） . . . 1433
表 215. ADC 中断 . . . 1441
表 216. ADC 全局寄存器映射 . . . 1472
表 217. ADC 寄存器映射和复位值 . . . 1472
表 218. ADC 寄存器映射和复位值（主 ADC 和从 ADC 通用寄存器） . . . 1474
表 219. DTS 内部输入/输出信号 . . . 1476
表 220. 采样时间配置 . . . 1478
表 221. 触发配置 . . . 1479
表 222. 低功耗模式下的温度传感器行为 . . . 1481
表 223. 中断控制位 . . . 1482
表 224. DTS 寄存器映射和复位值 . . . 1490
表 225. VREFBUF 典型值 . . . 1491
表 226. VREF 缓冲器模式 . . . 1492
表 227. VREFBUF 寄存器映射和复位值 . . . 1494
表 228. DFSDM1 实现 . . . 1497
表 229. DFSDM 外部引脚 . . . 1499
表 230. DFSDM 内部信号 . . . 1499
表 231. DFSDM 触发器连接 . . . 1499
表 232. DFSDM 断路连接 . . . 1500
表 233. 滤波器最大输出分辨率（来自滤波器输出的峰值数据值），基于某些 FOSR 值 . . . 1514
表 234. 积分器最大输出分辨率（来自积分器输出的峰值数据值），基于某些 IOSR 值，FOSR = 256 以及 Sinc3 滤波器类型（最大数据） . . . 1514
表 235. DFSDM 中断请求 . . . 1521
表 236. DFSDM 寄存器映射和复位值 . . . 1544
表 237. 可用流水线 . . . 1549
表 238. 词汇表 . . . 1549
表 239. DCMIPP 输入/输出引脚 . . . 1551
表 240. DCMIPP 输入/输出引脚 . . . 1552
表 241. DCMIPP 时钟 . . . 1552
表 242. DCMIPP 复位 . . . 1552
表 243. 并行接口最大分辨率 (80 MHz) . . . 1553
表 244. 支持的像素格式 . . . 1554
表 245. DCMIPP_PRCR 位功能 . . . 1558
表 246. DCMIPP_PRESCR 和 DCMIPP_PRESUR 位功能 . . . 1558
表 247. DCMIPP_PxFCTCR 位功能 . . . 1561
表 248. DCMIPP_P0PPCR 位功能 . . . 1563
表 249. DCMIPP_P0DCCNTR 和 DCMIPP_P0DCLMTR 位功能 . . . 1565
表 250. DCMIPP_PxPPCR 位功能 . . . 1566
表 251. 并行接口输入像素格式 . . . 1567
表 252. 索引与 DCMIPP_PRCR 寄存器值之间的对应关系 . . . 1568
表 253. 并行接口输入像素格式 . . . 1569
表 254. 转储流水线输出像素格式 . . . 1569
表 255. 影子和物理寄存器 . . . 1572

表 256. DCMIPP 低功耗模式 . . . 1572
表 257. DCMIPP 中断 . . . 1574
表 258. 事件连接 . . . 1575
表 259. DCMIPP 寄存器结构 . . . 1575
表 260. DCMIPP 寄存器映射和复位值 . . . 1607
表 261. LTDC 引脚 . . . 1612
表 262. 各个寄存器的时钟域 . . . 1613
表 263. LTDC 寄存器访问和更新持续时间 . . . 1614
表 264. 像素数据映射与颜色格式的关系 . . . 1617
表 265. LTDC 中断请求 . . . 1631
表 266. LTDC 寄存器映射和复位值 . . . 1664
表 267. RNG 内部输入/输出信号 . . . 1671
表 268. RNG 中断请求 . . . 1678
表 269. RNG 配置 . . . 1679
表 270. 配置选择 . . . 1679
表 271. RNG 寄存器映射和复位值 . . . 1687
表 272. SAES 内部输入/输出信号 . . . 1690
表 273. CTR 模式初始化向量定义 . . . 1706
表 274. GCM 最后一个块定义 . . . 1708
表 275. 在 GCM 模式下初始化 SAES_IVRx 寄存器 . . . 1709
表 276. 在 CCM 模式下初始化 SAES_IVRx 寄存器 . . . 1715
表 277. SAES_KEYRx 寄存器中的密钥字节序（128 位或 256 位密钥长度） . . . 1725
表 278. SAES 中断请求 . . . 1730
表 279. ECB、CBC 和 CTR 模式下的处理延迟 . . . 1730
表 280. GCM 和 CCM 模式下的处理延迟（SAES 内核时钟周期数） . . . 1730
表 281. SAES 寄存器映射和复位值 . . . 1746
表 282. CRYP 内部输入/输出信号 . . . 1752
表 283. 计数器模式初始化向量 . . . 1776
表 284. GCM 最后一个块定义 . . . 1778
表 285. GCM 模式 IV 寄存器初始化 . . . 1779
表 286. CCM 模式 IV 寄存器初始化 . . . 1784
表 287. DES/TDES 数据交换示例 . . . 1788
表 288. AES 数据交换示例 . . . 1789
表 289. CRYP_KxR/LR 寄存器中的密钥字节序（AES 128/192/256 位密钥） . . . 1791
表 290. CRYP_KxR/LR 寄存器中的密钥字节序（DES K1 和 TDES K1/2/3） . . . 1791
表 291. CRYP_IVx(L/R)R 寄存器中的初始化向量字节序 (AES) . . . 1792
表 292. CRYP_IVx(L/R)R 寄存器中的初始化向量字节序 (DES/TDES) . . . 1792
表 293. 针对内存到外设的 DMA 传输的密码处理器配置 . . . 1793
表 294. 针对外设到内存的 DMA 传输的密码处理器配置 . . . 1794
表 295. CRYP 中断请求 . . . 1796
表 296. ECB、CBC 和 CTR 模式下的处理延迟 . . . 1796
表 297. GCM 和 CCM 模式下每个 128 位块的处理时间（以时钟周期计） . . . 1797
表 298. CRYP 寄存器映射和复位值 . . . 1814
表 299. HASH 内部输入/输出信号 . . . 1819
表 300. 所支持哈希算法的相关信息 . . . 1819
表 301. 哈希处理器输出 . . . 1823
表 302. HASH 中断请求 . . . 1830
表 303. 处理时间（时钟周期数） . . . 1830
表 304. HASH1 寄存器映射和复位值 . . . 1841
表 305. 内部输入/输出信号 . . . 1844
表 306. PKA 整数算术函数列表 . . . 1845
表 307. PKA 素域 (Fp) 椭圆曲线函数列表 . . . 1846

表 308. Montgomery 参数计算 1851
表 309. 模加 1852
表 310. 模减 1852
表 311. Montgomery 乘法 1853
表 312. 模幂运算（正常模式） 1854
表 313. 模幂运算（快速模式） 1854
表 314. 模幂运算（受保护模式） 1855
表 315. 模逆 1855
表 316. 模约简 1856
表 317. 算术加法 1856
表 318. 算术减法 1857
表 319. 算术乘法 1857
表 320. 算术比较 1857
表 321. CRT 求幂 1858
表 322. 检查点是否在椭圆曲线 Fp 上 1859
表 323. ECC Fp 标量乘法 1859
表 324. ECDSA 签名——输入 1861
表 325. ECDSA 签名——输出 1861
表 326. 扩展 ECDSA 签名——额外输出 1861
表 327. ECDSA 验证——输入 1862
表 328. ECDSA 验证——输出 1862
表 329. ECC 完整加法 1863
表 330. ECC 双基阶梯运算 1864
表 331. ECC 仿射投影运算 1865
表 332. 针对 ECC 运算而支持的曲线族 1866
表 333. 模幂 1867
表 334. ECC 标量乘法 1867
表 335. ECDSA 签名平均计算时间 1868
表 336. ECDSA 验证平均计算时间 1868
表 337. ECC 双基阶梯运算平均计算时间 1868
表 338. ECC 仿射投影平均计算时间 1868
表 339. ECC 完整加法平均计算时间 1868
表 340. 检查点是否在椭圆曲线 Fp 上平均计算时间 1868
表 341. Montgomery 参数平均计算时间 1869
表 342. PKA 中断请求 1869
表 343. PKA 寄存器映射和复位值 1875
表 344. MCE 内部输入/输出信号 1878
表 345. MCE 中断请求 1880
表 346. MCE 寄存器映射和复位值 1889
表 347. 定时器输出行为与 BRK/BRK2 输入 1929
表 348. 刹车保护解除条件 1931
表 349. 计数方向与编码器信号的关系 1937
表 350. TIMx 内部触发连接 1952
表 351. 具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位 1966
表 352. TIM1 寄存器映射和复位值 1983
表 353. TIM8 寄存器映射和复位值 1985
表 354. 计数方向与编码器信号的关系 2020
表 355. TIMx 内部触发连接 2036
表 356. 标准 OCx 通道的输出控制位 2047
表 357. TIM2/TIM3/TIM4/TIM5 寄存器映射和复位值 2058
表 358. TIMx 内部触发连接 2087
表 359. 标准 OCx 通道的输出控制位 2095

表 360. TIM12 寄存器映射和复位值 2098
表 361. 标准 OCx 通道的输出控制位 2106
表 362. TIM13/TIM14 寄存器映射和复位值 2109
表 363. 刹车保护解除条件 2137
表 364. TIMx 内部触发连接 2151
表 365. 具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM15) 2160
表 366. TIM15 寄存器映射和复位值 2168
表 367. 具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM16/17) 2179
表 368. TIM16/TIM17 寄存器映射和复位值 2189
表 369. TIMx 寄存器映射和复位值 2203
表 370. STM32MP13xx LPTIM 特性 2205
表 371. LPTIM 输入/输出引脚 2207
表 372. LPTIM 内部信号 2207
表 373. LPTIM1 外部触发连接 2207
表 374. LPTIM2 外部触发连接 2208
表 375. LPTIM3 外部触发连接 2208
表 376. LPTIM4 外部触发连接 2208
表 377. LPTIM5 外部触发连接 2209
表 378. LPTIM1 输入 1 连接 2209
表 379. LPTIM1 输入 2 连接 2209
表 380. LPTIM2 输入 1 连接 2209
表 381. LPTIM2 输入 2 连接 2210
表 382. LPTIM3 输入 1 连接 2210
表 383. 预分频器的分频比 2211
表 384. 编码器计数方案 2217
表 385. 低功耗模式对 LPTIM 的影响 2218
表 386. 中断事件 2219
表 387. LPTIM 寄存器映射和复位值 2231
表 388. STGEN 寄存器映射和复位值 2249
表 389. STGENR 寄存器映射和复位值 2250
表 390. STM32MP13xx IWDG 特性 2252
表 391. IWDG 内部输入/输出信号 2253
表 392. 低功耗模式对 IWDG 的影响 2257
表 393. IWDG 中断请求 2258
表 394. IWDG 寄存器映射和复位值 2266
表 395. RTC 输入/输出引脚 2269
表 396. RTC 内部输入/输出信号 2269
表 397. RTC 互连 2270
表 398. RTC 引脚 PC13 配置 2270
表 399. PI1 配置 2273
表 400. RTC_OUT 映射 2274
表 401. 低功耗模式对 RTC 的影响 2286
表 402. 各个模式下的 RTC 引脚功能 2287
表 403. 非安全中断请求 2287
表 404. 安全中断请求 2288
表 405. RTC 寄存器映射和复位值 2318
表 406. TAMP 输入/输出引脚 2323
表 407. TAMP 内部输入/输出信号 2323
表 408. TAMP 互连 2324
表 409. 设备资源 x Tamper 保护 2328
表 410. 主动 Tamper 输出变化周期 2330
表 411. 最小 ATPER 值 2331

表 412. 主动 Tamper 滤波脉冲的持续时间 2332
表 413. 低功耗模式对 TAMP 的影响 2334
表 414. 各个模式下的 TAMP 引脚功能 2334
表 415. 各模式下的内部 Tamper 功能 2335
表 416. 非安全中断请求 2335
表 417. 安全中断请求 2336
表 418. TAMP 寄存器映射和复位值 2368
表 419. STM32MP13xx I2C 特性实现 2372
表 420. I2C 输入/输出引脚 2374
表 421. I2C 内部输入/输出信号 2374
表 422. 模拟滤波器与数字滤波器对比 2376
表 423. I2C-SMBus 规范数据建立和保持时间 2378
表 424. I2C 配置 2381
表 425. I2C-SMBus 规范时钟时序 2392
表 426. fI2CCLK = 8 MHz 时的时序设置示例 2401
表 427. fI2CCLK = 16 MHz 时的时序设置示例 2401
表 428. fI2CCLK = 48 MHz 时的时序设置示例 2402
表 429. SMBus 超时规范 2404
表 430. 带 PEC 的 SMBus 配置 2405
表 431. TIMEOUTA 设置示例（最大 $t_{TIMEOUT}$ = 25 ms） 2406
表 432. TIMEOUTB 设置示例 2407
表 433. TIMEOUTA 设置示例（最大 t_{IDLE} = 50 µs） 2407
表 434. 低功耗模式对 I2C 的影响 2417
表 435. I2C 中断请求 2417
表 436. I2C 寄存器映射和复位值 2433
表 437. STM32MP13xx 特性 2436
表 438. USART 特性 2437
表 439. 通过采样数据进行噪声检测 2451
表 440. BRR [3:0] = 0000 时的 USART 接收器容差 2454
表 441. BRR[3:0] 不等于 0000 时的 USART 接收器容差 2454
表 442. USART 帧格式 2458
表 443. 低功耗模式对 USART 的影响 2479
表 444. USART 中断请求 2479
表 445. USART 寄存器映射和复位值 2514
表 446. STM32MP13xxSPI 特性 2517
表 447. SPI 唤醒和中断请求 2546
表 448. PCM/I2S 模式下可用的位字段 2549
表 449. AFCNTR = 1 时在使能 SPI/I2S 之前的 WS 和 CK 电平 2557
表 450. 串行数据线交换 2557
表 451. 适合常见 I2S 频率的 CLKGEN 编程示例 2562
表 452. I2S 中断请求 2570
表 453. SPI 寄存器映射和复位值 2590
表 454. STM32MP13xx SAI 特性 2593
表 455. SAI 内部输入/输出信号 2595
表 456. SAI 输入/输出引脚 2595
表 457. 外部同步选择 2597
表 458. MCLK_x 激活条件 2602
表 459. 时钟发生器编程示例 2605
表 460. TDM 设置 2612
表 461. TDM 帧配置示例 2614
表 462. SOPD 模式 2617
表 463. 奇偶校验位计算 2618

表 464. 音频采样频率与符号率 2619
表 465. SAI 中断源 2627
表 466. SAI 寄存器映射和复位值 2656
表 467. SPDIFRX 内部输入/输出信号 2659
表 468. SPDIFRX 引脚 2659
表 469. 前导码的转换序列 2665
表 470. spdifrx_ker_ck 的最小频率与音频采样率 2674
表 471. spdifrx_symb_ck 的生成条件 2675
表 472. 位字段属性与 SPDIFRX 状态 2677
表 473. SPDIFRX 接口寄存器映射和复位值 2690
表 474. CAN 子系统 I/O 信号 2692
表 475. CAN 子系统 I/O 引脚 2693
表 476. CAN 触发器 2693
表 477. 主要特性 2695
表 478. FDCAN 中的 DLC 编码 2699
表 479. 接收缓冲区过滤器配置示例 2711
表 480. 调试消息过滤器配置示例 2711
表 481. 帧发送的可能配置 2712
表 482. 发送缓冲区/FIFO - 队列元素大小 2713
表 483. 1 级参考消息的第一个字节 2722
表 484. 2 级参考消息的前四个字节 2722
表 485. 0 级参考消息的前四个字节 2722
表 486. TUR 配置示例 2724
表 487. 系统矩阵，节点 A 2727
表 488. 触发列表，节点 A 2728
表 489. 随参考消息一起发送的数据字节数 2735
表 490. 接收缓冲区和 FIFO 元素 2741
表 491. 接收缓冲区和 FIFO 元素说明 2741
表 492. 发送缓冲区和 FIFO 元素 2743
表 493. 发送缓冲区元素说明 2743
表 494. 发送事件 FIFO 元素 2745
表 495. 发送事件 FIFO 元素说明 2745
表 496. 标准消息 ID 过滤器元素 2746
表 497. 标准消息 ID 过滤器元素位字段说明 2747
表 498. 扩展消息 ID 过滤器元素 2748
表 499. 扩展消息 ID 过滤器元素位字段说明 2748
表 500. 触发内存元素 2749
表 501. 触发内存元素说明 2749
表 502. FDCAN 寄存器映射和复位值 2788
表 503. FDCAN TT 寄存器映射和复位值 2810
表 504. CCU 寄存器映射和复位值 2816
表 505. 支持的 OTG 速度 2818
表 506. OTG 实现 2820
表 507. OTG输入/输出引脚 2821
表 508. OTG 输入/输出信号 2822
表 509. STM32 低功耗模式与 OTG 的兼容性 2833
表 510. 模块全局控制和状态寄存器 (CSR) 2840
表 511. 主机模式控制和状态寄存器 (CSR) 2841
表 512. 从机模式控制和状态寄存器 2843
表 513. 数据 FIFO (DFIFO) 访问寄存器地址映射 2845
表 514. 电源和时钟门控控制和状态寄存器 2845
表 515. TRDT 值 2852

表 516. 软断开的最小时间 . . . 2893
表 517. OTG 寄存器映射和复位值 . . . 2915
表 518. USBPHYC 寄存器映射和复位值 . . . 2991
表 519. USBH 寄存器映射和复位值 . . . 3004
表 520. 以太网外设引脚 . . . 3013
表 521. 以太网内部输入/输出信号 . . . 3014
表 522. DMA 通道的固定优先级方案 . . . 3026
表 523. Tx 路径中的双 VLAN 处理功能 . . . 3029
表 524. Rx 路径中的双 VLAN 处理功能 . . . 3030
表 525. 基于 VLTI 位进行 VLAN 插入或替换 . . . 3031
表 526. 目标地址过滤 . . . 3034
表 527. 源地址过滤 . . . 3035
表 528. VLAN 匹配状态 . . . 3036
表 529. 普通时钟：用于快照的 PTP 消息 . . . 3038
表 530. 端对端透明时钟：用于快照的 PTP 消息 . . . 3039
表 531. 点对点透明时钟：用于快照的 PTP 消息 . . . 3039
表 532. PHY 接口的出站和入站延迟 . . . 3043
表 533. 最小 PTP 时钟频率示例 . . . 3043
表 534. IEEE 1588-2008 中定义的消息格式 . . . 3044
表 535. IEEE 1588-2008 中定义的消息格式 . . . 3045
表 536. 控制和状态所需的 IPv6-UDP PTP 数据包位字段 . . . 3046
表 537. 控制和状态所需的以太网 PTP 数据包位字段 . . . 3047
表 538. 时间戳快照对 ETH_MACTSCR 位的相依性 . . . 3048
表 539. PTP 消息生成条件 . . . 3053
表 540. 通用 PTP 消息报头位字段 . . . 3055
表 541. MAC 发送 PTP 模式和单步时间戳操作 . . . 3057
表 542. 针对不同数据包类型的发送校验和减荷引擎功能 . . . 3061
表 543. 针对不同数据包类型的接收校验和减荷引擎功能 . . . 3063
表 544. TSO：TCP 和 IP 报头位字段 . . . 3067
表 545. 暂停数据包位字段 . . . 3072
表 546. Tx MAC 流控制 . . . 3072
表 547. Rx MAC 流控制 . . . 3073
表 548. 最大接收数据包的大小 . . . 3075
表 549. MCD 时钟选择 . . . 3077
表 550. MDIO 条款 45 帧结构 . . . 3078
表 551. MDIO 条款 22 帧结构 . . . 3079
表 552. 远程唤醒数据包过滤寄存器 . . . 3091
表 553. 远程唤醒过滤器位字段的描述 . . . 3091
表 554. 远程唤醒数据包和 PMT 中断生成 . . . 3093
表 555. 传送完成中断行为 . . . 3101
表 556. TDES0 正常描述符（读取格式） . . . 3120
表 557. TDES1 正常描述符（读取格式） . . . 3121
表 558. TDES2 正常描述符（读取格式） . . . 3121
表 559. TDES3 正常描述符（读取格式） . . . 3122
表 560. TDES0 正常描述符（回写格式） . . . 3124
表 561. TDES1 正常描述符（回写格式） . . . 3124
表 562. TDES2 正常描述符（回写格式） . . . 3125
表 563. TDES3 正常描述符（回写格式） . . . 3125
表 564. TDES0 上下文描述符 . . . 3128
表 565. TDES1 上下文描述符 . . . 3128
表 566. TDES2 上下文描述符 . . . 3128
表 567. TDES3 上下文描述符 . . . 3129

表 568. RDES0 正常描述符（读取格式）. 3131
表 569. RDES1 正常描述符（读取格式）. 3131
表 570. RDES2 正常描述符（读取格式）. 3132
表 571. RDES3 正常描述符（读取格式）. 3132
表 572. RDES0 正常描述符（回写格式）. 3133
表 573. RDES1 正常描述符（回写格式）. 3134
表 574. RDES2 正常描述符（回写格式）. 3136
表 575. RDES3 正常描述符（回写格式）. 3137
表 576. RDES0 上下文描述符. 3139
表 577. RDES1 上下文描述符. 3140
表 578. RDES2 上下文描述符. 3140
表 579. RDES3 上下文描述符. 3140
表 580. ETH_DMA 通用寄存器映射和复位值 3164
表 581. ETH_DMA_CH0 寄存器映射和复位值 3165
表 582. ETH_DMA_CH1 寄存器映射和复位值 3166
表 583. ETH_MTL 寄存器映射和复位值 3179
表 584. 基于 S2KP 和 JE 位的大型数据包状态. 3186
表 585. 基于 CST 和 ACS 位的数据包长度. 3186
表 586. 以太网 MAC 寄存器映射和复位值. 3268
表 587. HDP 信号复用. 3281
表 588. HDP 寄存器映射和复位值. 3287
表 589. JTAG/串行线调试端口引脚. 3290
表 590. 跟踪端口引脚 3290
表 591. 触发引脚. 3290
表 592. 认证信号连接 3293
表 593. 认证信号初始状态. 3293
表 594. CLTAPC 寄存器 3294
表 595. JTAG-DP 数据寄存器. 3296
表 596. 数据包请求 3297
表 597. ACK 响应 3297
表 598. 数据传输. 3298
表 599. 调试端口寄存器. 3299
表 600. AP0（系统互连）认证行为. 3308
表 601. AP1（调试 APB）认证行为 3308
表 602. MEM-AP 寄存器 3308
表 603. 调试子系统 ROM 表 3317
表 604. 跟踪子系统 ROM 表 3318
表 605. Cortex-A7 子系统 ROM 表 3318
表 606. ROM 表寄存器映射和复位值 3325
表 607. TSGEN 寄存器映射和复位值 3332
表 608. 跟踪子系统 CTI 输入. 3335
表 609. 跟踪子系统 CTI 输出. 3335
表 610. Cortex®-A7 CTI 输入. 3335
表 611. Cortex®-A7 CTI 输出. 3336
表 612. CTI 寄存器映射和复位值. 3350
表 613. ETF 寄存器映射和复位值 3372
表 614. TPIU 寄存器映射和复位值 3390
表 615. DBGMCU 寄存器映射和复位值 3403
表 616. 调试单元寄存器映射和复位值. 3443
表 617. 性能监视单元寄存器映射和复位值 3466
表 618. ETM 寄存器映射和复位值. 3515
表 619. 文档版本历史 3527

图片索引

图 1. STM32MP13xx 的总线互连 . . . 112
图 2. STM32MP13xx 的 NIC-400 互连 AXIM . . . 116
图 3. STM32MP13xx 的多层 AHB 互连 . . . 121
图 4. 内存映射 . . . 148
图 5. BSEC 框图 . . . 160
图 6. 上电和初始读取时序图 . . . 163
图 7. DDRCTRL 框图 . . . 199
图 8. TZC 框图 . . . 306
图 9. TZC 示例 1 . . . 308
图 10. TZC 示例 2 . . . 309
图 11. TZC 示例 3 . . . 310
图 12. DDRPHYC 框图 . . . 335
图 13. DDRPHY 框图 . . . 337
图 14. MDLL 框图 . . . 338
图 15. MSDLL 框图 . . . 338
图 16. ITM 框图 . . . 340
图 17. 用于 LPDDR2/3 的 ITMC_D2 . . . 341
图 18. 用于 DDR3/3L 的 ITMC_D2 . . . 342
图 19. 用于 DQ/DQS 通道的 ITMS 和 ITMD . . . 343
图 20. PUBL 框图 . . . 345
图 21. DDRPHY 初始化序列 . . . 347
图 22. 内置 DQS 门控训练流程 . . . 350
图 23. 软件 DQS 门控训练流程 . . . 352
图 24. 写时序，WL = 5，BL = 8 . . . 356
图 25. 读时序，WL = 5，BL = 8 . . . 357
图 26. DDRPERFM 框图 . . . 406
图 27. 电源控制框图 . . . 418
图 28. 电源概述 . . . 422
图 29. 设备启动供电序列 . . . 424
图 30. 备份域 . . . 426
图 31. 上电复位/掉电复位波形 . . . 429
图 32. BOR 阈值 . . . 430
图 33. 待机模式下自动域复位 . . . 431
图 34. PVD 阈值 . . . 432
图 35. AVD 阈值 . . . 433
图 36. VBAT 阈值 . . . 434
图 37. 温度阈值 . . . 435
图 38. 电源状态和外部稳压器控制 . . . 442
图 39. 设备低功耗停止供电时序 . . . 448
图 40. 设备低功耗 LP-Stop 供电时序 . . . 449
图 41. 设备低功耗 LPLV-Stop 供电时序 . . . 451
图 42. 设备 LPLV-Stop2 模式电源控制 . . . 453
图 43. 设备待机模式电源控制 . . . 456
图 44. 设备 VBAT 模式电源控制 . . . 457
图 45. PWR_ON 和 PWR_LP 引脚复用 . . . 458
图 46. RCC 框图 . . . 475
图 47. RCC 复位图 . . . 478
图 48. 复位电路概述 . . . 480

图 49. 复位脉冲控制 482
图 50. 复位延迟控制 483
图 51. 复位源寄存器 486
图 52. 启动、冷/热复位以及从任何 Stop 模式唤醒的序列 489
图 53. 顶级时钟树 491
图 54. HSE 时钟源 493
图 55. HSE 时钟生成 493
图 56. HSI 校准流程 496
图 57. CSI 校准流程 497
图 58. LSE 时钟生成 497
图 59. LSE 和 HSE CSS 功能 500
图 60. PLL1 框图 502
图 61. PLL2 框图 503
图 62. PLL3 和 PLL4 框图 503
图 63. 展频调制 506
图 64. 三角波形发生器 508
图 65. 动态时钟切换 509
图 66. 内核与总线时钟生成 511
图 67. 控制低功耗模式的关键信号 513
图 68. 时钟恢复激活与系统模式 515
图 69. 时钟恢复过程在任何系统 Stop 模式后激活 517
图 70. 从 CStop 过程唤醒 518
图 71. SAI 和 DFSDM 的外设时钟分配 528
图 72. SPI/I2S 和 SPDIFRX 的外设时钟分配 529
图 73. SPI 的外设时钟分配 530
图 74. I2C 的外设时钟分配 531
图 75. UART 和 USART 的外设时钟分配 532
图 76. TIM 和 LPTIM 的外设时钟分配 534
图 77. LTDC 的外设时钟分配 535
图 78. QUADSPI 和 FMC 的外设时钟分配 535
图 79. SDMMC 的外设时钟分配 536
图 80. USB 的外设时钟分配 537
图 81. 以太网的外设时钟分配 538
图 82. RNG1 的内核时钟分配 539
图 83. FDCAN 的内核时钟分配 540
图 84. ADC 和 STGEN 的内核时钟分配 541
图 85. DCMIPP 的内核时钟分配 542
图 86. DTS 的内核时钟分配 542
图 87. RTC 的时钟分配 543
图 88. IWDG[2:1] 的时钟分配 544
图 89. TZC 的时钟分配 545
图 90. DDRC、DDRPHYC 和 DDRPERFM 的时钟分配 546
图 91. 在 ASR1 或 ASR2 模式下进入和退出自刷新 551
图 92. 在 HSR1 或 HSR2 模式下进入和退出自刷新 552
图 93. 允许在 SSR、ASR 与 HSR 模式之间进行的跳变 553
图 94. DBG 和跟踪的时钟说明 556
图 95. IO 补偿单元的时钟说明 556
图 96. 外设内核时钟使能逻辑详细信息 559
图 97. MPU 低功耗接口 561
图 98. MPU 时钟路径 572
图 99. PLL2 时钟路径 573
图 100. MLAHB 时钟路径 575

图 101. 因 MPU 事件退出 LP-Stop . . . 584
图 102. 快速时钟恢复 . . . 585
图 103. I/O 端口位的基本结构 . . . 910
图 104. 5 V 容限 I/O 端口位的基本结构 . . . 910
图 105. 输入浮空/上拉/下拉配置 . . . 914
图 106. 输出配置 . . . 915
图 107. 复用功能配置 . . . 916
图 108. 高阻态模拟配置 . . . 917
图 109. I/O 补偿控制概述 . . . 936
图 110. STM32MP13xx 安全架构 . . . 965
图 111. MDMA 框图 . . . 995
图 112. DMA 框图 . . . 1019
图 113. 外设到内存模式 . . . 1022
图 114. 内存到外设模式 . . . 1023
图 115. 内存到内存模式 . . . 1024
图 116. FIFO 结构 . . . 1029
图 117. DMAMUX 框图 . . . 1060
图 118. DMAMUX 请求线复用器通道的同步模式 . . . 1063
图 119. DMA 请求线复用器通道的事件生成 . . . 1063
图 120. GIC 框图 . . . 1086
图 121. EXTI 框图 . . . 1138
图 122. 可配置事件触发逻辑 CPU 唤醒 . . . 1141
图 123. 直接事件触发逻辑 CPU 唤醒 . . . 1142
图 124. EXTI 复用器 GPIO 选择 . . . 1142
图 125. CRC 计算单元框图 . . . 1173
图 126. FMC 框图 . . . 1183
图 127. 模式 1 读取访问波形 . . . 1193
图 128. 模式 1 写入访问波形 . . . 1193
图 129. 模式 A 读取访问波形 . . . 1195
图 130. 模式 A 写入访问波形 . . . 1196
图 131. 模式 2 和模式 B 读取访问波形 . . . 1198
图 132. 模式 2 写入访问波形 . . . 1198
图 133. 模式 B 写入访问波形 . . . 1199
图 134. 模式 C 读取访问波形 . . . 1201
图 135. 模式 C 写入访问波形 . . . 1201
图 136. 模式 D 读取访问波形 . . . 1203
图 137. 模式 D 写入访问波形 . . . 1204
图 138. 复用读取访问波形 . . . 1206
图 139. 复用写入访问波形 . . . 1207
图 140. 读取访问波形期间的异步等待 . . . 1210
图 141. 写入访问波形期间的异步等待 . . . 1210
图 142. 等待配置波形 . . . 1212
图 143. 同步复用读取模式波形 - NOR、PSRAM (CRAM) . . . 1213
图 144. 同步复用写入模式波形 - PSRAM (CRAM) . . . 1215
图 145. NAND Flash 控制器的通用内存访问波形 . . . 1230
图 146. TCLR 和 TAR 时序的 NAND Flash 控制器波形 . . . 1230
图 147. 访问 NAND Flash . . . 1232
图 148. 带 BCH 的 NAND Flash 页 (2 KB) 布局 . . . 1236
图 149. 带 4 位 BCH 的 2 KB 页编程序列示例 . . . 1237
图 150. 链式页读命令时序 . . . 1239
图 151. 带 4 位 BCH 的 2 KB 页读取序列示例 . . . 1239
图 152. QUADSPI 框图 . . . 1269

图 153. QUADSPI 功能框图（双闪存模式使能） 1270
图 154. 四线 SPI 模式下的读命令示例 1271
图 155. 四线模式下 DDR 命令示例 1274
图 156. CKMODE = 0 时的 NCS（T = CLK 周期） 1281
图 157. SDR 模式下 CKMODE = 1 时的 NCS（T = CLK 周期） 1281
图 158. DDR 模式下 CKMODE = 1 时的 NCS（T = CLK 周期） 1281
图 159. CKMODE = 1 且发生中止时的 NCS（T = CLK 周期） 1282
图 160. SDMMC“无响应”和“无数据”操作 1299
图 161. SDMMC（多个）块读取操作 1299
图 162. SDMMC（多个）块写入操作 1300
图 163. SDMMC（连续）流读取操作 1300
图 164. SDMMC（连续）流写入操作 1300
图 165. SDMMC 框图 1302
图 166. SDMMC 命令和数据相位关系 1304
图 167. 控制单元 1305
图 168. 命令/响应路径 1306
图 169. 命令路径状态机 (CPSM) 1307
图 170. 数据路径 1312
图 171. DDR 模式数据包时钟驱动 1313
图 172. DDR 模式 CRC 状态/启动确认时钟驱动 1313
图 173. 数据路径状态机 (DPSM) 1314
图 174. CLKMUX 单元 1323
图 175. 链表结构 1325
图 176. 异步中断生成 1328
图 177. 同步中断周期数据读取 1329
图 178. 同步中断周期数据写入 1329
图 179. 异步中断周期数据读取 1330
图 180. 异步中断周期数据写入 1331
图 181. 在 DS、HS、SDR12 和 SDR25 模式下通过 SDMMC_CK 停止时钟 1333
图 182. 在 DDR50、SDR50 和 SDR104 模式下通过 SDMMC_CK 停止时钟 1334
图 183. SDMMC_CK < 50 MHz 时的读取等待 1334
图 184. SDMMC_CK > 50 MHz 时的读取等待 1335
图 185. CMD12 流时序 1337
图 186. CMD5 睡眠唤醒过程 1338
图 187. 正常启动模式操作 1340
图 188. 备用启动模式操作 1341
图 189. 命令响应 R1b 繁忙信号 1342
图 190. SDMMC 状态控制 1343
图 191. 卡掉电再上电/上电图 1344
图 192. CMD11 信号电压切换序列 1345
图 193. 电压切换收发器的典型应用 1347
图 194. DLYB 框图 1374
图 195. ADC 框图 1382
图 196. ADC 时钟方案 1387
图 197. ADC1 连接功能 1388
图 198. ADC2 连接功能 1389
图 199. ADC 校准 1392
图 200. 更新 ADC 校准系数 1392
图 201. 混合单端通道和差分通道 1393
图 202. 使能/禁止 ADC 1394
图 203. Bulb 模式时序图 1397
图 204. 模数转换时间 1400

图 205. 停止正在进行的常规转换 . . . 1401
图 206. 停止正在进行的常规转换和注入转换 . . . 1401
图 207. 触发选择 . . . 1403
图 208. 注入转换延迟 . . . 1404
图 209. JSQR 上下文队列示例（队列更改） . . . 1407
图 210. JSQR 上下文队列示例（触发更改） . . . 1407
图 211. 转换前发生溢出的 JSQR 上下文队列示例 . . . 1408
图 212. 转换期间发生溢出的 JSQR 上下文队列示例 . . . 1408
图 213. 队列为空时的 JSQR 上下文队列示例（JQM = 0 的情况） . . . 1409
图 214. 队列为空时的 JSQR 上下文队列示例（JQM = 1 的情况） . . . 1410
图 215. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM = 0) —— 正在进行转换时发生 JADSTP . . . 1410
图 216. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM = 0) —— 正在进行转换时发生 JADSTP 并出现新触发 . . . 1411
图 217. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM = 0) —— 在正在进行转换的范围之外发生 JADSTP . . . 1411
图 218. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM = 1) . . . 1412
图 219. 通过将 ADDIS 置 1 的方式清空 JSQR 上下文队列 (JQM = 0) . . . 1412
图 220. 通过将 ADDIS 置 1 的方式清空 JSQR 上下文队列 (JQM = 1) . . . 1413
图 221. 单次序列转换，软件触发 . . . 1415
图 222. 连续序列转换，软件触发 . . . 1415
图 223. 单次序列转换，硬件触发 . . . 1416
图 224. 连续序列转换，硬件触发 . . . 1416
图 225. 右对齐（偏移禁止，无符号值） . . . 1418
图 226. 右对齐（偏移使能，有符号值） . . . 1419
图 227. 左对齐（偏移禁止，无符号值） . . . 1419
图 228. 左对齐（偏移使能，有符号值） . . . 1420
图 229. 溢出示例 (OVRMOD = 0) . . . 1421
图 230. 溢出示例 (OVRMOD = 1) . . . 1421
图 231. AUTODLY = 1，连续模式下的常规转换，软件触发 . . . 1424
图 232. AUTODLY = 1，被注入转换中断的常规硬件转换（DISCEN = 0；JDISCEN = 0） . . . 1425
图 233. AUTODLY = 1，被注入转换中断的常规硬件转换（DISCEN = 1，JDISCEN = 1） . . . 1426
图 234. AUTODLY = 1，被注入转换中断的常规连续转换 . . . 1427
图 235. AUTODLY = 1，自动注入模式 (JAUTO = 1) . . . 1427
图 236. 模拟看门狗的保护区域 . . . 1428
图 237. ADCy_AWDx_OUT 信号生成（所有常规通道上） . . . 1430
图 238. ADCy_AWDx_OUT 信号生成（AWDx 标志未通过软件清零） . . . 1430
图 239. ADCy_AWDx_OUT 信号生成（单条常规通道上） . . . 1431
图 240. ADCy_AWDx_OUT 信号生成（所有注入通道上） . . . 1431
图 241. 20 位到 16 位结果的截断过程 . . . 1432
图 242. 移 5 位并进行舍入的数值示例 . . . 1432
图 243. 已触发的常规过采样模式（TROVS 位 = 1） . . . 1434
图 244. 常规过采样模式（4x 过采样率） . . . 1435
图 245. 同时使用常规和注入过采样模式 . . . 1436
图 246. 已触发常规过采样支持注入 . . . 1436
图 247. 在自动注入模式下进行过采样 . . . 1437
图 248. 温度传感器通道框图 . . . 1438
图 249. VBAT 通道框图 . . . 1439
图 250. VREFINT 通道框图 . . . 1440
图 251. 温度传感器功能框图 . . . 1476
图 252. 低 REF_CLK 频率时的方法 . . . 1477
图 253. 高 REF_CLK 频率时的方法 . . . 1478

图 254. 温度传感器序列 1480
图 255. VREFBUF 框图 1491
图 256. 单个 DFSDM 框图 1498
图 257. 输入通道引脚重定向 1502
图 258. 通道收发器时序图 1504
图 259. SPI 时钟缺失时序图 1505
图 260. 曼彻斯特编码时钟缺失时序图 1506
图 261. 曼彻斯特编码首次转换（曼彻斯特同步） 1508
图 262. DFSDM_CHyDATINR 寄存器操作模式和分配 1512
图 263. 示例：Sinc3 滤波器类型 1513
图 264. DCMIPP 概述 1549
图 265. DCMIPP 框图 1551
图 266. 快照 (CPTMODE = 1) 和连续 (CPTMODE = 0) 捕获模式 1561
图 267. Pipe0（转储）架构概览 1563
图 268. LTDC 框图 1612
图 269. 图层窗口可编程参数 1622
图 270. 两层混合 1623
图 271. LTDC 同步时序 1627
图 272. RNG 框图 1671
图 273. NIST SP800-90B 熵源模型 1672
图 274. RNG 初始化概述 1674
图 275. SAES 框图 1689
图 276. ECB 加密和解密原理 1692
图 277. CBC 加密和解密原理 1693
图 278. CTR 加密和解密原理 1694
图 279. GCM 加密和认证原理 1695
图 280. GMAC 认证原理 1695
图 281. CCM 加密和认证原理 1696
图 282. 挂起模式管理示例 1700
图 283. ECB 加密 1701
图 284. ECB 解密 1701
图 285. CBC 加密 1702
图 286. CBC 解密 1702
图 287. ECB/CBC 加密（模式 1） 1703
图 288. ECB/CBC 解密（模式 3） 1704
图 289. CTR 模式下的消息结构 1705
图 290. CTR 加密 1706
图 291. CTR 解密 1706
图 292. GCM 中的消息结构 1708
图 293. GCM 已验证加密 1709
图 294. GMAC 模式下的消息结构 1712
图 295. GMAC 认证模式 1712
图 296. CCM 模式下的消息结构 1713
图 297. CCM 模式认证加密 1715
图 298. ECB 和 CBC 模式下的封装密钥操作 1718
图 299. CTR 模式下的封装密钥操作 1720
图 300. 共享密钥模式的使用 1721
图 301. 根据数据交换构建 128 位块 1724
图 302. 密钥保护机制 1726
图 303. 输入阶段 128 位数据块的 DMA 传输 1727
图 304. 输出阶段 128 位数据块的 DMA 传输 1727
图 305. CRYP 框图 1751

图 306. AES-ECB 模式概览 . . . 1754
图 307. AES-CBC 模式概览 . . . 1755
图 308. AES-CTR 模式概览 . . . 1756
图 309. AES-GCM 模式概览 . . . 1757
图 310. AES-GMAC 模式概览 . . . 1757
图 311. AES-CCM 模式概览 . . . 1758
图 312. 挂起模式管理示例 . . . 1763
图 313. DES/TDES-ECB 模式加密 . . . 1764
图 314. DES/TDES-ECB 模式解密 . . . 1765
图 315. DES/TDES-CBC 模式加密 . . . 1766
图 316. DES/TDES-CBC 模式解密 . . . 1767
图 317. AES-ECB 模式加密 . . . 1769
图 318. AES-ECB 模式解密 . . . 1770
图 319. AES-CBC 模式加密 . . . 1771
图 320. AES-CBC 模式解密 . . . 1772
图 321. 计数器模式的消息结构 . . . 1774
图 322. AES-CTR 模式加密 . . . 1775
图 323. AES-CTR 模式解密 . . . 1776
图 324. Galois/计数器模式的消息结构 . . . 1778
图 325. Galois 消息认证码模式的消息结构 . . . 1782
图 326. CBC-MAC 计数器模式的消息结构 . . . 1783
图 327. 根据数据类型构建 64 位块 (IN FIFO) . . . 1789
图 328. 根据数据类型构建 128 位块 . . . 1790
图 329. HASH 框图 . . . 1818
图 330. 消息数据交换功能 . . . 1821
图 331. HASH 挂起/恢复机制 . . . 1827
图 332. PKA 框图 . . . 1844
图 333. MCE 框图 . . . 1878
图 334. MCE 块密码实现 . . . 1879
图 335. 高级控制定时器框图 . . . 1892
图 336. 预分频器分频由 1 变为 2 时的计数器时序图 . . . 1894
图 337. 预分频器分频由 1 变为 4 时的计数器时序图 . . . 1894
图 338. 计数器时序图，1 分频内部时钟 . . . 1895
图 339. 计数器时序图，2 分频内部时钟 . . . 1896
图 340. 计数器时序图，4 分频内部时钟 . . . 1896
图 341. 计数器时序图，N 分频内部时钟 . . . 1897
图 342. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载） . . . 1897
图 343. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载） . . . 1898
图 344. 计数器时序图，1 分频内部时钟 . . . 1899
图 345. 计数器时序图，2 分频内部时钟 . . . 1899
图 346. 计数器时序图，4 分频内部时钟 . . . 1900
图 347. 计数器时序图，N 分频内部时钟 . . . 1900
图 348. 计数器时序图，未使用重复计数器时更新事件 . . . 1901
图 349. 计数器时序图，1 分频内部时钟，TIMx_ARR = 0x6 . . . 1902
图 350. 计数器时序图，2 分频内部时钟 . . . 1903
图 351. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36 . . . 1903
图 352. 计数器时序图，N 分频内部时钟 . . . 1904
图 353. 计数器时序图，ARPE=1 时的更新事件（计数器下溢） . . . 1904
图 354. 计数器时序图，ARPE=1 时的更新事件（计数器上溢） . . . 1905
图 355. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例 . . . 1906
图 356. 外部触发输入模块 . . . 1907
图 357. TIM1/TIM8 ETR 输入电路 . . . 1907

图 358. 正常模式下的控制电路，1 分频内部时钟 1908
图 359. TI2 外部时钟连接示例 1909
图 360. 外部时钟模式 1 下的控制电路 1910
图 361. 外部触发输入模块 1910
图 362. 外部时钟模式 2 下的控制电路 1911
图 363. 捕获/比较通道（例如：通道 1 输入阶段） 1912
图 364. 捕获/比较通道 1 主电路 1912
图 365. 捕获/比较通道的输出阶段（通道 1、通道 2 和通道 3） 1913
图 366. 捕获/比较通道的输出阶段（通道 4） 1913
图 367. 捕获/比较通道的输出阶段（通道 5 和通道 6） 1914
图 368. PWM 输入模式时序 1916
图 369. 输出比较模式，翻转 OC1 1917
图 370. 边沿对齐模式的 PWM 波形 (ARR=8) 1919
图 371. 中心对齐模式 PWM 波形 (ARR=8) 1920
图 372. 50% 占空比时产生的 2 个相移 PWM 信号 1921
图 373. 通道 1 和通道 3 上的组合 PWM 模式 1922
图 374. 三相组合 PWM 信号（每个周期多个触发脉冲） 1923
图 375. 带死区插入的互补输出 1924
图 376. 延迟时间大于负脉冲宽度的死区波形 1924
图 377. 延迟时间大于正脉冲宽度的死区波形 1925
图 378. 刹车事件和刹车 2 电路概述 1926
图 379. 响应 BRK 上的刹车事件的不同输出行为 (OSSI = 1) 1928
图 380. BRK 和 BRK2 引脚使能后的 PWM 输出状态 (OSSI=1) 1929
图 381. BRK 使能后的 PWM 输出状态 (OSSI=0) 1930
图 382. 输出重定向（图中未显示 BRK2 请求） 1931
图 383. 清除 TIMx 的 OCxREF 1932
图 384. COM 事件生成 6 步 PWM 的示例 (OSSR=1) 1933
图 385. 单脉冲模式示例 1934
图 386. 可再触发单脉冲模式 1936
图 387. 编码器接口模式下的计数器工作示例 1937
图 388. TI1FP1 极性反相时的编码器接口模式示例 1938
图 389. 测量 3 个信号上边沿之间的时间间隔 1939
图 390. 霍尔传感器接口的示例 1940
图 391. 复位模式下的控制电路 1941
图 392. 门控模式下的控制电路 1942
图 393. 触发模式下的控制电路 1943
图 394. 外部时钟模式 2 + 触发模式下的控制电路 1944
图 395. 通用定时器框图 1989
图 396. 预分频器分频由 1 变为 2 时的计数器时序图 1991
图 397. 预分频器分频由 1 变为 4 时的计数器时序图 1991
图 398. 计数器时序图，1 分频内部时钟 1992
图 399. 计数器时序图，2 分频内部时钟 1993
图 400. 计数器时序图，4 分频内部时钟 1993
图 401. 计数器时序图，N 分频内部时钟 1994
图 402. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载） 1994
图 403. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载） 1995
图 404. 计数器时序图，1 分频内部时钟 1996
图 405. 计数器时序图，2 分频内部时钟 1996
图 406. 计数器时序图，4 分频内部时钟 1997
图 407. 计数器时序图，N 分频内部时钟 1997
图 408. 计数器时序图，更新事件 1998
图 409. 计数器时序图，1 分频内部时钟，TIMx_ARR=0x6 1999

图 410. 计数器时序图，2 分频内部时钟 2000
图 411. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36 2000
图 412. 计数器时序图，N 分频内部时钟 2001
图 413. 计数器时序图，ARPE=1 时的更新事件（计数器下溢） 2001
图 414. 计数器时序图，ARPE=1 时的更新事件（计数器上溢） 2002
图 415. 正常模式下的控制电路，1 分频内部时钟 2003
图 416. TI2 外部时钟连接示例 2003
图 417. 外部时钟模式 1 下的控制电路 2004
图 418. 外部触发输入模块 2004
图 419. 外部时钟模式 2 下的控制电路 2005
图 420. 捕获/比较通道（例如：通道 1 输入阶段） 2006
图 421. 捕获/比较通道 1 主电路 2006
图 422. 捕获/比较通道的输出阶段（通道 1） 2007
图 423. PWM 输入模式时序 2009
图 424. 输出比较模式，翻转 OC1 2010
图 425. 边沿对齐模式的 PWM 波形 (ARR=8) 2011
图 426. 中心对齐模式 PWM 波形 (ARR=8) 2013
图 427. 50% 占空比时产生的 2 个相移 PWM 信号 2014
图 428. 通道 1 和通道 3 上的组合 PWM 模式 2015
图 429. 清除 TIMx 的 OCxREF 2016
图 430. 单脉冲模式示例 2017
图 431. 可再触发单脉冲模式 2019
图 432. 编码器接口模式下的计数器工作示例 2020
图 433. TI1FP1 极性反相时的编码器接口模式示例 2021
图 434. 复位模式下的控制电路 2022
图 435. 门控模式下的控制电路 2023
图 436. 触发模式下的控制电路 2024
图 437. 外部时钟模式 2 + 触发模式下的控制电路 2025
图 438. 主/从定时器示例 2025
图 439. 单通道定时器的主/从连接示例 2026
图 440. 使用 TIM3 的 OC1REF 对 TIM2 实施门控控制 2027
图 441. 使用 TIM3 的使能信号对 TIM2 实施门控控制 2028
图 442. 使用 TIM3 的更新事件触发 TIM2 2028
图 443. 使用 TIM3 的使能信号触发 TIM2 2029
图 444. 使用 TIM3 的 TI1 输入触发 TIM3 和 TIM2 2030
图 445. 通用定时器框图 (TIM12) 2062
图 446. 通用定时器框图 (TIM13/TIM14) 2063
图 447. 预分频器分频由 1 变为 2 时的计数器时序图 2065
图 448. 预分频器分频由 1 变为 4 时的计数器时序图 2065
图 449. 计数器时序图，1 分频内部时钟 2066
图 450. 计数器时序图，2 分频内部时钟 2067
图 451. 计数器时序图，4 分频内部时钟 2067
图 452. 计数器时序图，N 分频内部时钟 2068
图 453. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载） 2068
图 454. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载） 2069
图 455. 正常模式下的控制电路，1 分频内部时钟 2070
图 456. TI2 外部时钟连接示例 2070
图 457. 外部时钟模式 1 下的控制电路 2071
图 458. 捕获/比较通道（例如：通道 1 输入阶段） 2071
图 459. 捕获/比较通道 1 主电路 2072
图 460. 捕获/比较通道的输出阶段（通道 1） 2072
图 461. PWM 输入模式时序 2074

图 462. 输出比较模式，翻转 OC1 2076
图 463. 边沿对齐模式的 PWM 波形 (ARR=8) 2077
图 464. 通道 1 和通道 2 上的组合 PWM 模式 2078
图 465. 单脉冲模式示例 2079
图 466. 可再触发单脉冲模式 2080
图 467. 测量 2 个信号上边沿之间的时间间隔 2081
图 468. 复位模式下的控制电路 2082
图 469. 门控模式下的控制电路 2083
图 470. 触发模式下的控制电路 2083
图 471. TIM15 框图 2113
图 472. TIM16/TIM17 框图 2114
图 473. 预分频器分频由 1 变为 2 时的计数器时序图 2116
图 474. 预分频器分频由 1 变为 4 时的计数器时序图 2116
图 475. 计数器时序图，1 分频内部时钟 2117
图 476. 计数器时序图，2 分频内部时钟 2118
图 477. 计数器时序图，4 分频内部时钟 2118
图 478. 计数器时序图，N 分频内部时钟 2119
图 479. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载） 2119
图 480. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载） 2120
图 481. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例 2121
图 482. 正常模式下的控制电路，1 分频内部时钟 2122
图 483. TI2 外部时钟连接示例 2122
图 484. 外部时钟模式 1 下的控制电路 2123
图 485. 捕获/比较通道（例如：通道 1 输入阶段） 2124
图 486. 捕获/比较通道 1 主电路 2124
图 487. 捕获/比较通道的输出阶段（通道 1） 2125
图 488. 捕获/比较通道的输出阶段（TIM15 的通道 2） 2125
图 489. PWM 输入模式时序 2127
图 490. 输出比较模式，翻转 OC1 2129
图 491. 边沿对齐模式的 PWM 波形 (ARR=8) 2130
图 492. 通道 1 和通道 2 上的组合 PWM 模式 2131
图 493. 带死区插入的互补输出 2132
图 494. 延迟时间大于负脉冲宽度的死区波形 2132
图 495. 延迟时间大于正脉冲宽度的死区波形。 2132
图 496. 刹车电路概述 2134
图 497. 输出的刹车响应行为 2136
图 498. 输出重定向 2138
图 499. COM 事件生成 6 步 PWM 的示例 (OSSR=1) 2139
图 500. 单脉冲模式示例 2140
图 501. 可再触发单脉冲模式 2141
图 502. 测量 2 个信号上边沿之间的时间间隔 2142
图 503. 复位模式下的控制电路 2143
图 504. 门控模式下的控制电路 2144
图 505. 触发模式下的控制电路 2144
图 506. 基本定时器框图 2191
图 507. 预分频器分频由 1 变为 2 时的计数器时序图 2193
图 508. 预分频器分频由 1 变为 4 时的计数器时序图 2193
图 509. 计数器时序图，1 分频内部时钟 2194
图 510. 计数器时序图，2 分频内部时钟 2195
图 511. 计数器时序图，4 分频内部时钟 2195
图 512. 计数器时序图，N 分频内部时钟 2196
图 513. 计数器时序图，ARPE = 0 时更新事件（TIMx_ARR 未预装载） 2196

图 514. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载） 2197
图 515. 正常模式下的控制电路，1 分频内部时钟 2198
图 516. 低功耗定时器框图（LPTIM1 和 LPTIM2） 2205
图 517. 低功耗定时器框图 (LPTIM3) 2206
图 518. 低功耗定时器框图（LPTIM4 和 LPTIM5） 2206
图 519. 干扰滤波器时序图 2211
图 520. LPTIM 输出波形，单次计数模式配置 2212
图 521. LPTIM 输出波形，单次计数模式配置且激活单次设置模式（WAVE 位置 1） 2213
图 522. LPTIM 输出波形、连续计数模式配置 2213
图 523. 生成波形 2215
图 524. 编码器模式计数序列 2218
图 525. STGEN 框图 2234
图 526. 独立看门狗框图 2253
图 527. 由超时引起的复位时序 2254
图 528. 由于在不允许刷新的区域中执行刷新而引起的复位时序 2255
图 529. 独立看门狗中断时序图 2258
图 530. RTC 框图 2268
图 531. TAMP 框图 2322
图 532. 备份寄存器保护区 2326
图 533. 主动 Tamper 滤波 2332
图 534. I2C 框图 2373
图 535. I2C 总线协议 2375
图 536. 建立和保持时序 2376
图 537. I2C 初始化流程 2379
图 538. 数据接收 2380
图 539. 数据发送 2381
图 540. 从设备初始化流程 2384
图 541. I2C 从发送器的传输序列流程 (NOSTRETCH = 0) 2385
图 542. I2C 从设备发送器的传输序列流程 (NOSTRETCH = 1) 2386
图 543. I2C 从设备发送器的传输总线图（仅限强制性事件） 2387
图 544. 从接收器的传输序列流程 (NOSTRETCH = 0) 2388
图 545. 从接收器的传输序列流程 (NOSTRETCH = 1) 2389
图 546. I2C 从接收器的传输总线图（仅限强制性事件） 2389
图 547. 主时钟生成 2391
图 548. 主设备初始化流程 2393
图 549. 10 位地址读访问 (HEAD10R = 0) 2393
图 550. 10 位地址读访问 (HEAD10R = 1) 2394
图 551. I2C 主发送器的传输序列流程（N ≤ 255 字节） 2395
图 552. I2C 主发送器的传输序列流程（N > 255 字节） 2396
图 553. I2C 主发送器的传输总线图（仅限强制性事件） 2397
图 554. I2C 主接收器的传输序列流程（N ≤ 255 字节） 2398
图 555. I2C 主接收器的传输序列流程（N > 255 字节） 2399
图 556. I2C 主接收器的传输总线图（仅限强制性事件） 2400
图 557. $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的超时间隔 2404
图 558. SMBus 从发送器的传输序列流程（N 字节 + PEC） 2408
图 559. SMBus 从发送器的传输总线图 (SBC = 1) 2408
图 560. SMBus 从接收器的传输序列流程（N 字节 + PEC） 2410
图 561. SMBus 从接收器的总线传输图 (SBC = 1) 2411
图 562. SMBus 主发送器的总线传输图 2412
图 563. SMBus 主接收器的总线传输图 2413
图 564. USART 框图 2438
图 565. 字长编程 2441

图 566. 可配置的停止位 . . . 2443
图 567. 发送时的 TC/TXE 行为 . . . 2445
图 568. 16 倍或 8 倍过采样时的起始位检测 . . . 2446
图 569. usart_ker_ck 时钟分频器框图 . . . 2449
图 570. 16 倍过采样时的数据采样 . . . 2450
图 571. 8 倍过采样时的数据采样 . . . 2451
图 572. 使用空闲线路检测时的静默模式 . . . 2456
图 573. 使用地址标记检测时的静默模式 . . . 2457
图 574. LIN 模式下的刹车检测（11 位刹车长度——LBDL 位置 1） . . . 2460
图 575. LIN 模式下的刹车检测与帧错误检测 . . . 2461
图 576. USART 同步主发送示例 . . . 2462
图 577. USART 在同步主模式下的数据时钟时序图（M 位 = 00） . . . 2462
图 578. USART 在同步主模式下的数据时钟时序图（M 位 = 01） . . . 2463
图 579. USART 在同步从模式下的数据时钟时序图（M 位 = 00） . . . 2464
图 580. ISO 7816-3 异步协议 . . . 2466
图 581. 使用 1.5 个停止位检测奇偶校验错误 . . . 2467
图 582. IrDA SIR ENDEC 框图 . . . 2470
图 583. IrDA 数据调制 (3/16)——正常模式 . . . 2471
图 584. 使用 DMA 进行发送 . . . 2472
图 585. 使用 DMA 进行接收 . . . 2473
图 586. 2 个 USART 间的硬件流控制 . . . 2473
图 587. RS232 RTS 流控制 . . . 2474
图 588. RS232 CTS 流控制 . . . 2475
图 589. 唤醒事件通过验证（唤醒事件 = 地址匹配，禁止 FIFO） . . . 2477
图 590. 唤醒事件未通过验证（唤醒事件 = 地址匹配，禁止 FIFO） . . . 2478
图 591. SPI2S 框图 . . . 2517
图 592. 全双工单个主设备/单个从设备应用 . . . 2519
图 593. 半双工单个主设备/单个从设备应用 . . . 2520
图 594. 单工单个主设备/单个从设备应用（主设备为仅发送模式/从设备为仅接收模式） . . . 2521
图 595. 使用星形拓扑的主设备和三个独立从设备 . . . 2522
图 596. 一主三从环形（菊花链）拓扑 . . . 2523
图 597. 多主设备应用 . . . 2524
图 598. SS 控制逻辑原理图 . . . 2526
图 599. 数据流时序控制（SSOE = 1，SSOM = 0，SSM = 0） . . . 2526
图 600. 数据之间的 SS 交错脉冲（SSOE = 1，SSOM = 1，SSM = 0） . . . 2527
图 601. 数据时钟时序图 . . . 2529
图 602. 数据大小不等于 8 位、16 位或 32 位时的数据对齐 . . . 2530
图 603. 将数据封装到 FIFO 中以进行发送和接收 . . . 2537
图 604. TI 模式传输 . . . 2539
图 605. 检测到下溢情况时从设备行为的可选配置 . . . 2541
图 606. 低功耗模式应用示例 . . . 2545
图 607. 波形示例 . . . 2551
图 608. 主模式 I2S Philips 协议波形（16/32 位全精度） . . . 2552
图 609. I2S Philips 标准波形 . . . 2552
图 610. 主模式 MSB 对齐的 16 位或 32 位全精度长度 . . . 2553
图 611. 主模式 MSB 对齐的 16 位或 24 位数据长度 . . . 2553
图 612. 从模式 MSB 对齐 . . . 2554
图 613. LSB 对齐的 16 位或 24 位数据长度 . . . 2554
图 614. 帧长度等于数据长度时的主模式 PCM . . . 2555
图 615. 主模式 PCM 标准波形（16 位或 24 位数据长度） . . . 2555
图 616. 从模式 PCM 波形 . . . 2556
图 617. 启动序列，I2S Philips 标准，主模式 . . . 2558

图 618. 启动序列，I2S Philips 标准，从模式 2559
图 619. 停止序列，I2S Philips 标准，主模式 2560
图 620. I^2S 时钟发生器架构 2560
图 621. 数据格式 2563
图 622. 下溢情况的处理 2565
图 623. 上溢情况的处理 2566
图 624. 帧错误检测，FIXCH = 0 2567
图 625. 帧错误检测，FIXCH = 1 2567
图 626. SAI 功能框图 2594
图 627. 音频帧 2597
图 628. FS 的作用是 SOF 信号 + 通道识别信号 (FSDEF = TRIS = 1) 2599
图 629. FS 的作用是 SOF 信号 (FSDEF = 0) 2600
图 630. SAI_xSLOTR 中的 FBOFF = 0 时的 Slot 大小配置 2601
图 631. 第一位偏移 2601
图 632. 音频模块时钟发生器概览 2603
图 633. PDM 典型连接和时序 2607
图 634. 详细的 PDM 接口模块框图 2608
图 635. 启动序列 2609
图 636. TDM 中的 SAI_ADR 格式（32 位 Slot 宽度） 2610
图 637. TDM 中的 SAI_ADR 格式（16 位 Slot 宽度） 2611
图 638. TDM 中的 SAI_ADR 格式（8 位 Slot 宽度） 2612
图 639. AC’97 音频帧 2615
图 640. 至少具有 2 个嵌入式 SAI 的设备的典型 AC’97 配置示例（三个外部 AC’97 解码器） . . 2616
图 641. SPDIF 格式 2617
图 642. SAI_xDR 寄存器定序 2618
图 643. SAI 的音频模块中的数据压扩硬件 2621
图 644. 发送无效 Slot 时 SD 输出线上的三态策略 2622
图 645. 采用 I2S 等协议时输出数据线上的三态策略 2623
图 646. 上溢错误检测 2624
图 647. FIFO 下溢事件 2624
图 648. SPDIFRX 框图 2659
图 649. S/PDIF 子帧格式 2660
图 650. S/PDIF 块格式 2660
图 651. S/PDIF 前导码 2661
图 652. 通道编码示例 2662
图 653. SPDIFRX 解码器 2663
图 654. 噪声滤波和边沿检测 2663
图 655. 阈值 2664
图 656. 同步流程图 2666
图 657. 同步过程调度 2667
图 658. SPDIFRX 状态 2668
图 659. SPDIFRX_FMTx_DR 寄存器格式 2670
图 660. 通道/用户数据格式 2671
图 661. RXSTEO = 0 时的 S/PDIF 上溢错误 2673
图 662. RXSTEO = 1 时的 S/PDIF 上溢错误 2674
图 663. SPDIFRX 接口中断映射图 2676
图 664. CAN 子系统 2694
图 665. FDCAN 框图 2696
图 666. 收发器延迟测量 2701
图 667. 总线监控模式下的引脚控制 2702
图 668. 环回模式下的引脚控制 2704
图 669. 消息 RAM 配置 2705

图 670. 标准消息 ID 过滤器路径 2708
图 671. 扩展消息 ID 过滤器路径 2709
图 672. 混合配置专用发送缓冲区/发送 FIFO 示例 2714
图 673. 混合配置专用发送缓冲区/发送队列示例 2715
图 674. 位时序 2716
图 675. 绕过操作 2717
图 676. FSM 校准 2719
图 677. 周期时间和全局时间同步 2732
图 678. TTCAN 0 级和 2 级偏移补偿 2733
图 679. 0 级调度同步状态机 2739
图 680. 0 级主控节点与被控节点的关系 2740
图 681. 高速 OTG 模块框图 2821
图 682. OTG A-B 设备连接 2823
图 683. OTG 纯设备的连接 2824
图 684. OTG 仅做主机的连接 2828
图 685. SOF 连接（SOF 触发输出与 TIM 和 ITR1 的连接） 2832
图 686. 动态更新 OTG_HFIR (RLDCTRL = 1) 2834
图 687. 从机模式下的 FIFO 地址映射和 AHB FIFO 访问映射 2835
图 688. 主机模式下的 FIFO 地址映射和 AHB FIFO 访问映射 2836
图 689. 中断层级 2839
图 690. 发送 FIFO 写任务 2930
图 691. 接收 FIFO 读任务 2931
图 692. 正常批量/控制 OUT/SETUP 2932
图 693. 批量/控制 IN 事务 2936
图 694. 正常中断 OUT 2938
图 695. 正常中断 IN 2942
图 696. 同步 OUT 事务 2944
图 697. 同步 IN 事务 2947
图 698. 正常批量/控制 OUT/SETUP 事务 - DMA 2949
图 699. 正常批量/控制 IN 事务 - DMA 2951
图 700. 正常中断 OUT 事务 - DMA 模式 2952
图 701. 正常中断 IN 事务 - DMA 模式 2953
图 702. 正常同步 OUT 事务 - DMA 模式 2954
图 703. 正常同步 IN 事务 - DMA 模式 2955
图 704. 接收 FIFO 数据包读取 2960
图 705. 处理 SETUP 数据包 2962
图 706. 批量 OUT 事务 2968
图 707. TRDT 最大时序情况 2976
图 708. A 设备 SRP 2977
图 709. B 设备 SRP 2978
图 710. A 设备 HNP 2979
图 711. B 设备 HNP 2980
图 712. USBPHYC 框图 2983
图 713. USBH 框图 2994
图 714. 以太网高级框图 3015
图 715. DMA 发送流程（标准模式） 3017
图 716. DMA 发送流程（OSP 模式） 3019
图 717. 接收 DMA 流程 3021
图 718. MAC 发送流程概览 3023
图 719. MAC 接收流程 3025
图 720. 数据包过滤序列 3032
图 721. 网络时间同步 3040

图 722. 支持点对点路径校准的时钟的传播延迟计算 3041
图 723. 使用精密校准方法更新系统时间 3050
图 724. TCP 分段减荷概览 3064
图 725. TCP 分段减荷流程 3065
图 726. 分段数据包的报头和有效负载位字段 3068
图 727. 支持的 PHY 接口 3076
图 728. SMA 接口模块 3076
图 729. MDIO 数据包结构（条款 45） 3077
图 730. MDIO 数据包结构（条款 22） 3078
图 731. SMA 写入操作流程 3080
图 732. 写数据包 3081
图 733. 读取数据包 3081
图 734. 介质独立接口 (MII) 信号 3082
图 735. RMII 框图 3084
图 736. 发送位序 3085
图 737. 接收位序 3086
图 738. RGMII 框图及接口信号 3087
图 739. LPI 转换（发送，100 Mbps） 3096
图 740. LPI Tx 时钟门控（LPITCSE = 1 时） 3096
图 741. LPI 转换（接收，100 Mbps） 3098
图 742. 描述符环结构 3118
图 743. DMA 描述符环 3119
图 744. 发送描述符（读取格式） 3120
图 745. 发送描述符回写格式 3124
图 746. 发送上下文描述符格式 3127
图 747. 接收正常描述符（读取格式） 3131
图 748. 接收正常描述符（回写格式） 3133
图 749. 接收上下文描述符 3139
图 750. ETH_DMAISR 标志的生成 3157
图 751. 硬件调试端口框图 3279
图 752. 调试支持基础结构框图 3290
图 753. 调试时钟域 3291
图 754. JTAG TAP 状态机 3295
图 755. SWD 成功的数据传输过程 3298
图 756. 调试和访问端口连接 3307
图 757. 调试 APB CoreSight™ 拓扑 3320
图 758. CoreSight™ 时间戳分布 3326
图 759. 嵌入式交叉触发 3334
图 760. 将触发输入映射到输出 3336

1 文档约定

1.1 概述

STM32MP13xx 设备内置一个具有 FPU 内核的 Arm® Cortex®-A7。

arm

1.2 寄存器相关缩写词列表

寄存器说明中使用以下缩写词[a]：

读/写 (rw)	软件可以读写该位。
只读 (r)	软件只能读取该位。
只写 (w)	软件只能写入该位。读取该位时将返回复位值。
读取/写入 0 清零 (rc_w0)	软件可以通过读取该位或通过写入 0 将该位清零。写入 1 对该位的值无影响。
读取/写入 1 清零 (rc_w1)	软件可以通过读取该位或通过写入 1 将该位清零。写入 0 对该位的值无影响。
读取/写入清零 (rc_w)	软件可以通过读取该位或通过写入寄存器将该位清零。写入该位的值并不重要。
读取/读取清零 (rc_r)	软件可以读取该位。读取该位时，将自动清零。写入该位对其值无影响。
读取/读取置 1 (rs_r)	软件可以读取该位。读取该位时，将自动置 1。写入该位对其值无影响。
读取/置位 (rs)	软件可以读取该位，也可将其置 1。写入 0 对该位的值无影响。
读/仅可写入一次 (rwo)	软件只能对该位进行一次写操作，并且可以随时读取它。只能通过复位操作将该位恢复到复位值。
切换 (t)	软件可以通过写入 1 来切换该位。写入 0 无影响。
只读，写触发 (rt_w1)	软件可以读取该位。写入 1 时，将触发事件，但不会影响该位的值。
保留 (Res.)	保留位，必须保持复位值。

a. 这是一个涵盖适用于意法半导体微控制器的所有缩写词的详尽列表，其中一些缩写词在当前文档中可能并未使用。

1.3 词汇表

本节简要介绍本文档中所用首字母缩略词和缩写词的定义：

- 产品集成了两个调试端口：
  - JTAG 调试端口 (**JTAG-DP**) 提供基于联合测试工作组 (JTAG) 协议的 5 引脚标准接口。
  - SWD 调试端口 (**SWD-DP**) 提供基于串行线调试 (SWD) 协议的 2 引脚（时钟和数据）接口。
- **字**：32 位数据。
- **半字**：16 位数据。
- **字节**：8 位数据。
- **双字**：64 位数据。
- **选项字节**：存储于内部熔丝中的产品配置位。
- **OTP**：一次性编程（熔丝）。
- **AHB**：高级高性能总线。
- **APB**：高级外设总线。
- **AXI**：高级可扩展接口协议。
- **ECC**：误码校正。
- **DMA**：直接内存访问。
- **AHBS**: AHB 从接口。
- **AXIM**：AXI 互连矩阵。
- **RMA**：退料分析

1.4 外设可用性

有关各型号产品所支持的外设类型及数量，请参见相关设备数据手册。

2 内存和总线架构

2.1 系统架构

2.1.1 总线架构

总线架构基于两个

频率域不同的互连矩阵，如*图 1* 中所述。

- 一个基于 AXI 的高速 **Arm® CoreLink™ NIC-400 网络互连**：

  该互连专用于 Cortex-A7 CPU 子系统和高带宽主机：USBH、ETH1/2、SDMMC1/2、MDMA、DCMIPP、LTDC。

  它管理以下设备的访问权限：

  – 外部内存 DDR（通过 DDRCTRL）、NAND/NOR（通过 FMC 或 QUADSPI）
  – 主系统内部内存 SYSRAM 和 ROM
  – 主要安全外设集群（通过 AHB5 总线）
  – 控制和配置包括摄像头和显示接口在内的高带宽主机（通过 AHB6 总线）

- 一个**多层 AHB 互连**

  此互连专用于低带宽主机：OTG、DMA1-3。

  它管理以下设备的访问权限：

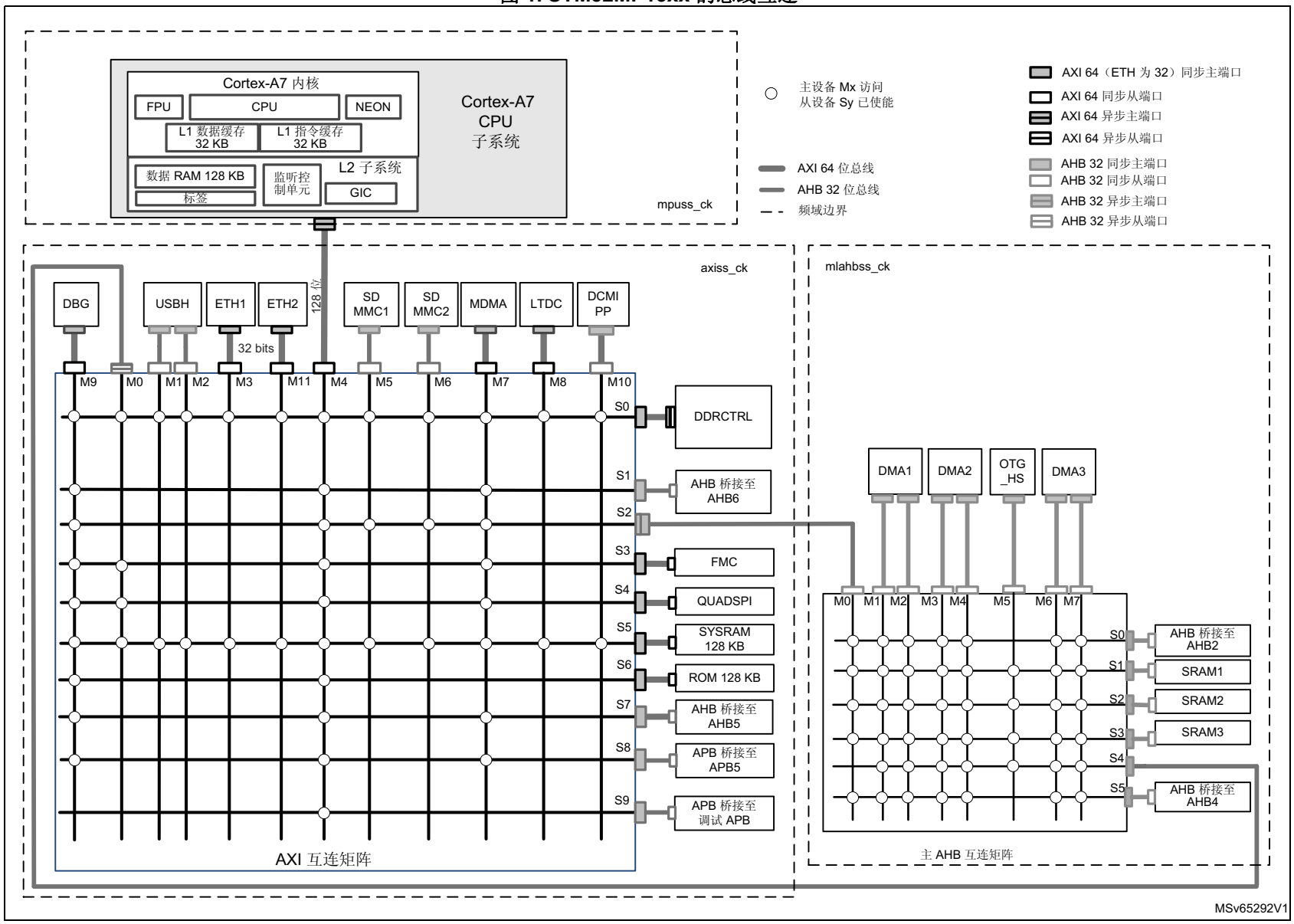
  – SRAM1、2、3
  – 大多数定时器和低速通信外设（通过 AHB2 总线）
  – 主要基础设施块，如时钟生成、电源控制、系统配置和杂项外设（通过 AHB4 总线）

AXI 和多层 AHB 矩阵通过一个专用的双向数据路径相互连接：

- AXIM 主机能访问所有多层 AHB 从设备。
- 多层 AHB 主机能访问 SYSRAM 和外部内存（通过 DDRCTRL、FMC、QUADSPI）

图 1 中的总线架构图描述了 STM32MP13xx 系列的互连。

图 1. STM32MP13xx 的总线互连



2.1.2 存储映射结构

简介

有关可用内存和寄存器区域的详细映射，请参见*第 2.5 节：存储结构*。

以下段落描述了系统中的各种资源。

嵌入式内存

- **ROM**

  128 KB 嵌入式 ROM 是一种快速访问内存（无等待周期）。它在 Cortex-A7 CPU 子系统中专用于启动代码执行。该内存及其相关存储控制器以 aclk 频率运行。

- **Cortex-A7 CPU 子系统缓存**

  Cortex-A7 包含 32 KB 指令和数据缓存。请参见 Cortex-A7 技术参考手册以了解缓存特性。

- **SYSRAM**

  128 KB 嵌入式 SYSRAM 是一种快速访问的静态 RAM，在 Cortex-A7 CPU 子系统中专用于代码执行和低延迟数据管理。SRAM 本身是 0 个等待周期。通过 AXI 互连传播数据时存在延迟。这个内存及其关联的存储控制器以 aclk 频率运行。

- **SRAM1、2、3**

  SRAM1、SRAM2、SRAM3 是三个快速访问的静态 RAM，分别为 16 KB、8 KB 和 8 KB，用于存储与 Cortex-A7 CPU 子系统交换的数据。这些内存及其关联的存储控制器以 mlhclk 频率运行。

- **Backup RAM**

  Backup RAM 是一种 8 KB 的静态 RAM，支持双轨供电，可以对整个系统进行数据备份，在各种功耗模式下，均支持标准延迟。该内存及其关联的存储控制器以 hclk5 频率运行。

- **OTP**

  OTP 是一种安全的一次性编程非易失性内存，总共有 3072 个有效位。它包含系统配置信息。这个内存及其关联的存储控制器以 pclk5 频率运行。

  复位后 OTP 会自动加载。位存储在影子寄存器中，支持 pclk5 访问频率。

外部内存支持

- **DDR3/LPDDR2 控制器 (DDRCTRL)**

  DDR 控制器支持 LPDDR2、LPDDR3、DDR3/3L，并具有 16 位数据接口。在内部，DDRCTRL 通过 64 位 AXI 端口连接到总线互连。支持的最大外部内存大小为 1 GB。

  DDRCTRL 可通过两个具有相同特性的 AXI 数据端口访问。系统配置寄存器允许指定的主设备选择要使用的端口，以平衡两个端口之间的访问量。

  有关内存特性和控制器参数化的更多信息，请参见*第 5 章：DDR3/LPDDR2/LPDDR3 控制器 (DDRCTRL)* 和*第 7 章：DDR 物理接口控制 (DDRPHYC)*。

  DDRCTRL 以 aclk 频率运行。外部内存的工作频率可达 pll2_r。

- **可变存储控制器 (FMC)/QUADSPI**

  SoC 中没有内部 Flash。系统初始启动阶段和自我初始化通过 Boot ROM 和 BSEC 实现。

  外部 Flash 可以通过可配置的 FMC NAND 接口或通过 QUADSPI 进行访问。

  如有需要，FMC 还允许并行连接 SRAM/PSRAM 以取代 Flash。

- **SD/SDIO/MMC 卡主机接口 (SDMMCx)**

  SoC 中没有内部 Flash。系统初始启动阶段和自我初始化通过 Boot ROM 和 BSEC 实现。

  通过 SDMMCx 可以访问 SD 卡或 *e*•MMC 等外部 Flash。

  SD、SDIO、eMMC 设备通过两个连接到 AXIM 互连的 SDMMC 控制器进行接口连接。这些控制器是互连的主设备，并发起与 SD/SDIO/MMC 内存之间的数据传输。

外设集群

下文简要描述了系统提供的七个外设集群。

注： *AHBx/APBy 表示资源通过 AHBx 总线访问，其中一些资源需要结合 APBy 总线和 AHBx 总线共同访问。*

- **AHB4/APB3**

  该集群包含重要的系统功能，如时钟和电源控制、系统配置、大部分 GPIO 控制，以及一些额外的定时器和低速通信接口。

- **AHB2/APB1-APB2-APB6**

  该集群包含定时器和低速通信接口：SPI、I2C、U(S)ART、SAI、ADC、FDCAN 和 DFSDM。

- **AHB5**

  该集群包含 Cortex-A7 和 Backup RAM 专用的安全模块。

- **APB5**

  该集群包括 BSEC 和设备配置控制资源。实时时钟生成和防 Tamper 功能也属于这个集群。

- **AHB6/APB4**

  这个集群包含大部分图形/显示资源控制，以及快速主设备的配置寄存器访问权限：MDMA、ETHernet、SDMMC、USBH。

- **调试**

  此集群包含所有调试功能，包括 Arm® CoreSight™ 基础设施。

内存映射概述

表 1 汇总了主机（参见*图 1*）与内存映射资源的访问关系。

表 1. 内存映射概述STM32MP13xx

资源	Cortex-A7	DMA1 / DMA2 / DMA3	调试器	MDMA	SDMMC1,2	OTG_HS	USBH / ETH1,2 / DCMIPP / LTDC
DDR	x	x	x	x	x	x	x
Cortex-A7 调试	x(1)	-	x	-	-	-	-
Cortex-A7 GIC	x	-	-	-	-	-	-
Ext FMC 接口	x	x	x	x	-	-	-
Ext QUADSPI	x	x	x	x	-	-	-
AXIM 寄存器	x	-	x	-	-	-	-
APB5 外设	x	-	x	x	-	-	-
AHB6/APB4 外设	x	-	x	x	-	-	-
AHB5 外设	x	-	x	x	-	-	-
APB 调试外设	x	-	x	-	-	-	-
AHB4/APB3 外设	x	x	x	x	x	-	-
AHB2/APB1,2,6 外设	x	x	x	x	x	-	-
SRAM 1,2,3	x	x	x	x	x	x	-
SYSRAM	x	x	x	x	x	x	x
ROM	x	-	x	x	-	-	-

1. Cortex-A7 通过 CoreSight 基础设施访问其自身的调试资源。

2.2 AXI 互连矩阵 (AXIM)

2.2.1 AXIM 特性

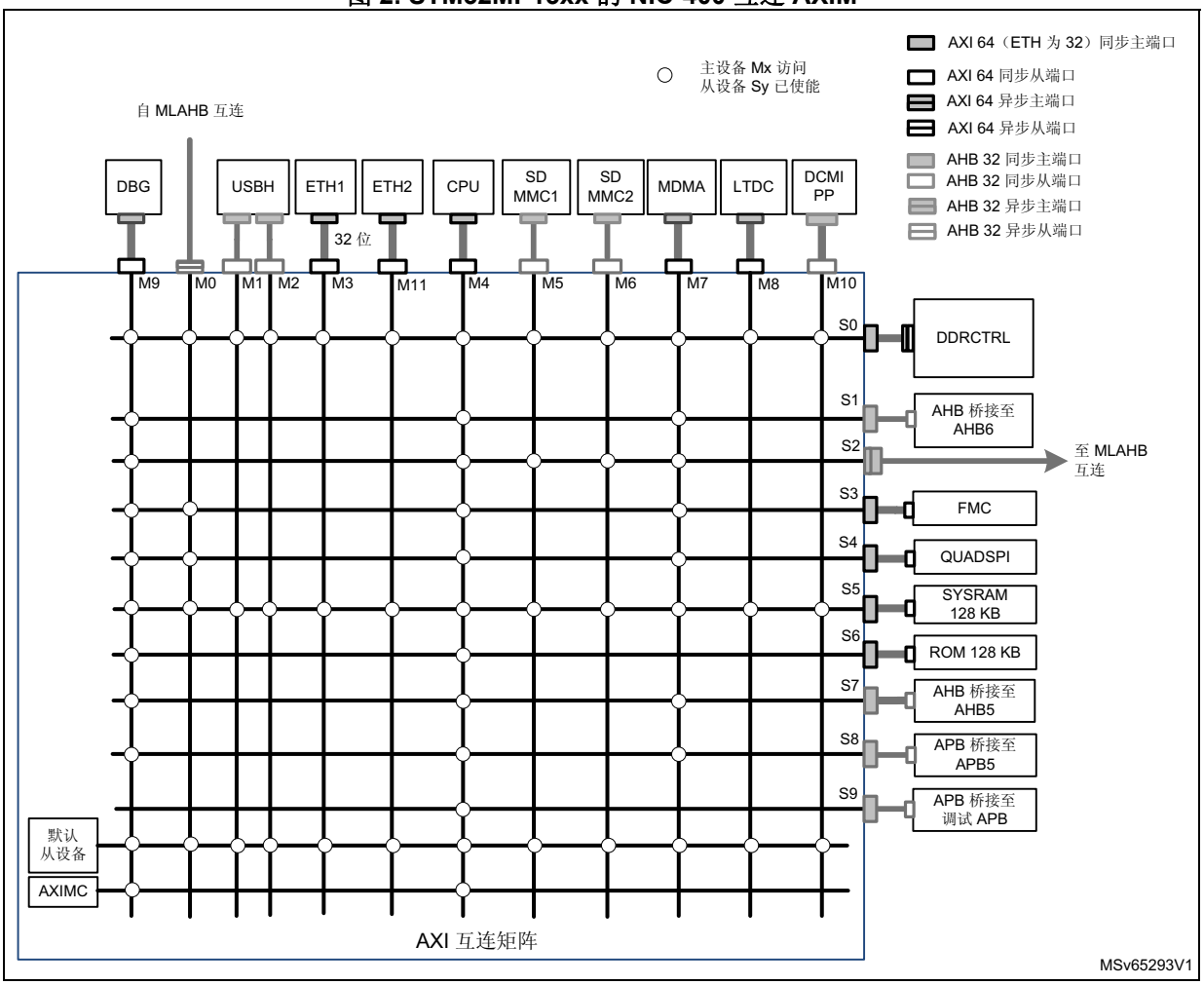
AXIM 采用 NIC-400 网络互连实现。其主要特点为：

- 12 个主机和 10 个从设备层（另加 2 个内部从设备），从主机到从设备的优化互连拓扑（非全互连）
- 支持从不同主机到连接到分离层的从设备的同时访问
- 协议转换（AXI、AHB）支持
- 内部数据和命令缓冲，以提高访问量效率
- QoS 编程支持
- 异步管理
- 通过用户可访问的寄存器对内部参数（QOS、旁路...）进行软件配置。

2.2.2 AXIM 互连配置

NIC-400 互连配置参见*图 2*。

图 2. STM32MP13xx 的 NIC-400 互连 AXIM



2.2.3 主端口描述

- **M0**：连接到 MLAHB 互连主端口的异步端口
  允许来自 MLAHB 子系统的主机访问系统中的所有块，除了 ROM 和 AXIMC 内部从设备（也称为 GPV，全局编程器视图），其中包含所有 NIC-400 编程寄存器。
- **M1/M2**：两个连接到 USBHost 主端口的 AHB-Lite 端口，具有相同的功能
  仅限于 DDR 和 SYSRAM 内存访问。
- **M3/M11**：连接到 ETH1 和 ETH2 以太网千兆接口的 AXI4 端口
  仅限于 DDR 和 SYSRAM 内存访问。
- **M4**：连接到 Cortex-A7 CPU 子系统（128 位宽）的 AXI4 端口
  在 CPU 子系统内部进行异步管理。在 NIC-400 内部进行 64 位转换。
- **M5/M6**：两个连接到 SDMMC1 和 SDMMC2 主端口的 AHB-Lite 端口
  仅限于 DDR 和 SYSRAM 内存访问，以及通过 MLAHB 实现的 SRAM3 内存访问。
- **M7**：连接到 MDMA 的 AXI4 端口
  MDMA 能够访问系统中除 ROM 和调试 APB 之外的所有从设备，包括主机 IP 的配置端口。借助其链表支持，它可以用于执行这些块的完整编程。这个主机具有重要的读取发行能力，并能生成接近层理论最大吞吐量的峰值访问量。
- **M8**：连接到 LTDC 显示驱动器的 AXI4 端口
  仅限于 DDR 和 SYSRAM 内存访问。
- **M9**：连接到调试访问端口 (DAP) AXI 的 AXI4 端口
  这个主端口可以访问所有从设备，除了返回到 DAP 的 S11，以避免系统拓扑中出现双重数据路径。M9 也访问包含所有互连编程寄存器的 AXIMC 内部从设备。
- **M10**：连接到 DCMIPP 摄像头接口的 AXI4 端口
  仅限于 DDR 和 SYSRAM 内存访问。

2.2.4 主端口的主要特性

表 2 总结了主端口的主要特性。

表 2. 主端口的主要特性

主端口	Nb	协议	数据总线宽度	同步性	ID 宽度	默认 QOS 值(1)	读/写发出能力
MLAHB interco	M0	AHB_Lite	32	异步	1	6	1/1
USBH p1	M1	AHB_Lite	32	同步	1	5	1/1
USBH p2	M2	AHB_Lite	32	同步	1	5	1/1
ETH1	M3	AXI4	32	同步	4	7	4/4
CPU	M4	AXI4	128	同步	6	12	8/8
SDMMC1	M5	AHB_Lite	32	同步	1	4	1/1
SDMMC2	M6	AHB_Lite	32	同步	1	4	1/1
MDMA	M7	AXI4	64	同步	1	8	8/1
LTDC	M8	AXI4	64	同步	4	10	2/1
DEBUG / DAP	M9	AXI4	64	同步	1	2	1/1
DCMIPP	M10	AXI4	64	同步	4	11	1/2
ETH2	M11	AXI4	32	同步	4	7	4/4

1. 主机优先级可以通过软件修改。

2.2.5 主端口安全特性

表 3 列出了主端口的安全特性。

表 3. 主端口安全特性

主端口	Nb	用于安全传播的 NIC 输入信号	TrustZone 配置	NSAID(1)
MLAHB interco	M0	HAuser(5:0)	非安全	0b0110 对应 DMA1 / DMA2 / DMA3 0b1000 对应 OTG
USBH p1	M1	HAuser(5:0)	非安全	0b0111
USBH p2	M2	HAuser(5:0)	非安全	0b0111
ETH1	M3	ARuser(5:0) AWuser(5:0)	安全/非安全	0b1010
CPU	M4	ARuser(5:0) AWuser(5:0)	Per-access	0b0000
SDMMC1	M5	HAuser(5:0)	非安全	0b1001
SDMMC2	M6	HAuser(5:0)	非安全	0b1001
MDMA	M7	ARuser(5:0) AWuser(5:0)	Per-access	0b0101
LTDC	M8	ARuser(5:0) AWuser(5:0)	Per-access	0b0011
DEBUG / DAP	M9	ARuser(5:0) AWuser(5:0)	Per-access	0b1111
DCMIPP	M10	ARuser(5:0) AWuser(5:0	非安全	0b1011
ETH2	M11	ARuser(5:0) AWUser(5:0	安全/非安全	0b1010

1. 主机 M0 发送多个对应于多层 AHB 主机的 NSAID。相同类型的不同主机（SDMMC1/2 和 USBH1/2）生成相同的 NSAID 值。

2.2.6 从端口描述

- **S0**：DDR 控制器的 AXI3 同步端口 0 和 1。在 DDR 控制器内部进行异步管理。
- **S1**：连接到 AHB6 的 AHB-Lite 同步端口，它是连接到所有主机的配置端口的总线桥。
- **S2**：连接到主多层 AHB 互连的 AHB-Lite 异步端口，Cortex-A7 通过该端口访问外设。
- **S3**：NAND/NOR 控制器的 AXI3 同步端口
- **S4**：QUADSPI 控制器的 AXI3 同步端口
- **S5**：SYSRAM 控制器的 AXI3 同步端口
- **S6**：ROM 控制器的 AXI3 同步端口
- **S7**：AHB5 的 AHB-Lite 同步端口。该总线桥控制加密资源、安全 GPI0 和 Backup RAM 的访问。
- **S8**：APB5 协议转换器的 AHB-Lite 同步端口。APB5 是支持安全信号传播的 APB3 总线。该总线控制安全管理模块和一组安全外设的访问。
- **S9**：位于 DAP 中实现 APB_DEBUG 访问的协议转换器的 AHB-Lite 同步端口。

内部从设备：

- **默认从设备**：定义了在非有效区域进行访问时的互连行为。
- **AXIMC**（也称为 GPV）：包含所有 NIC-400 互连配置寄存器。互连配置寄存器集组只能被两个主机访问：Cortex-A7 CPU（应用早期阶段的软件配置）和 DAP（调试，主要用于检查）。

2.2.7 从端口主要特性

表 4 列出了从端口的主要特性。

表 4. 从端口主要特性

从端口	Nb	协议	数据总线宽度	同步性	读/写接受能力
DDR p0	S0	AXI4	64	同步	20/20(1)
AHB6	S1	AHB_Lite	32	同步	1/1
多层 AHB	S2	AXI4	32	异步	1/1
FMC	S3	AXI4	64	同步	1/1
QUADSPI	S4	AXI4	64	同步	1/1
SYSRAM	S5	AXI3	64	同步	1/1
ROM	S6	AXI3	64	同步	1/1
AHB5	S7	AXI4	32	同步	1/1
APB5	S8	AHB_Lite	32	同步	1/1
DEBUG/DAP	S9	AHB_Lite	32	同步	1/1

1. DDR 读/写接受能力 = CAM 大小 (16) + 内部 AXI 端口接口 FIFO (4)

2.2.8 从端口安全特性

表 5 列出了从端口的安全特性。

表 5. 从端口安全特性

从端口	Nb	用于安全传播的 NIC 输入信号	TrustZone 配置
DDR p0	S0	ARuser(5:0) AWuser(5:0)	安全和非安全
AHB6	S1	HAuser(5:0)	安全和非安全
多层 AHB	S2	HAuser(5:0)	安全和非安全
FMC	S3	ARuser(5:0) AWuser(5:0)	安全和非安全
QUADSPI	S4	HAuser(5:0)	安全和非安全
SYSRAM	S5	ARuser(5:0) AWuser(5:0)	安全和非安全
ROM	S6	ARuser(5:0) AWuser(5:0)	安全和非安全
AHB5	S7	ARuser(5:0) AWuser(5:0)	安全和非安全
APB5	S8	ARuser(5:0) AWuser(5:0)	安全和非安全
DEBUG/DAP	S9	HAuser(5:0)	Per-access

2.3 多层 AHB 互连

2.3.1 多层 AHB 特性

多层 AHB 互连是经过优化的 Arm BP010 互连。

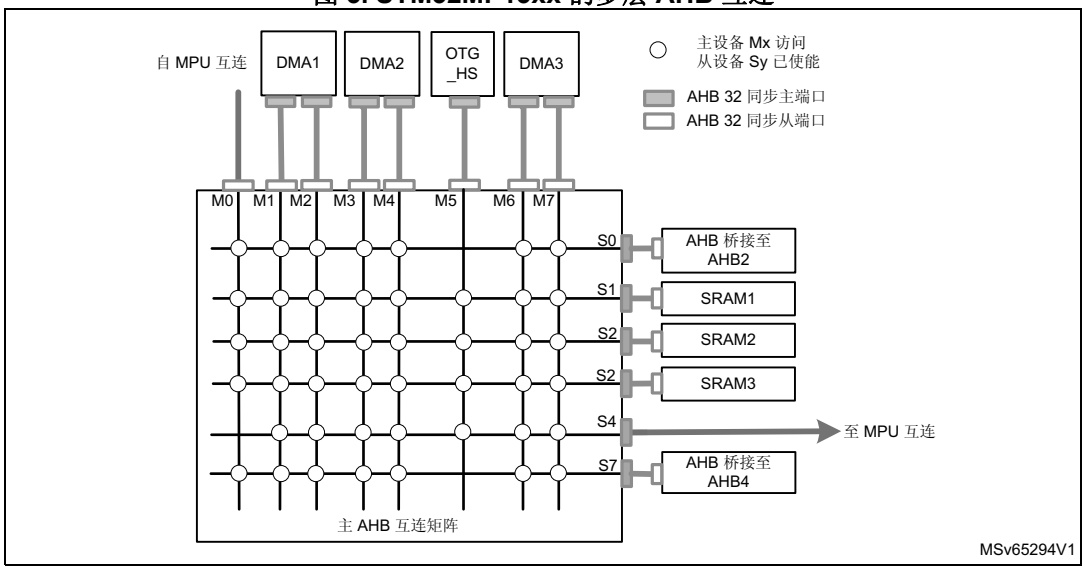
其主要特点为：

- 8 个主机和 6 个从设备层，全部是 32 位数据
- 支持从不同的主设备到连接到分离层的从设备的同时访问
- 内部数据和命令缓冲，以提高访问量效率

2.3.2 多层 AHB 互连配置

多层 AHB 互连配置参见*图 3*。

图 3. STM32MP13xx 的多层 AHB 互连



2.3.3 多层 AHB 主端口特性

- **M0**：连接到 MPU 互连主端口的 AHB-Lite

  它允许 MPU 子系统的主机访问系统中的所有从设备，除了指向 MPU 的路径，以避免回送。
- **M1、M2、M3、M4**：连接到 DMA1 和 DMA2 双主机模块的四个 AHB-Lite 端口，具有相同的功能。

  授权访问子系统中的所有从设备：SRAM 和所有外设的总线桥。
- **M5**：连接到 OTG 接口的 AHB-Lite 端口。

  仅限于 SRAM 和 MPU 子系统（SYSRAM，DDR）中的内存（通过主 NIC-400 AXI 互连）访问。
- **M6、M7**：连接到 DMA3 双主机的两个 AHB-Lite 端口。

  授权访问子系统中的所有从设备：SRAM 和所有外设的总线桥

表 6 列出了多层 AHB 主端口的主要特性。

表 6. 多层 AHB 主端口的主要特性

主端口	Nb	协议	数据总线宽度	同步性
AHB_MPU	M0	AHB-Lite	32	同步
DMA1 p1	M1	AHB-Lite	32	同步
DMA1 p2	M2	AHB-Lite	32	同步
DMA2 p1	M3	AHB-Lite	32	同步
DMA2 p2	M4	AHB-Lite	32	同步
OTG	M5	AHB-Lite	32	同步
DMA3 p1	M6	AHB-Lite	32	同步
DMA3 p2	M7	AHB-Lite	32	同步

2.3.4 多层 AHB 从端口特性

- **S0**：连接到 AHB2 的 AHB-Lite 同步端口
  它是连接到多层 AHB 的所有主机的配置端口的总线桥。该端口还允许访问位于 APB1 和 APB2 集群中的所有资源，以及 ADC。
- **S1、S2、S3**：SRAM1、SRAM2、SRAM3 的 AHB-Lite 同步端口。
- **S4**：连接到主 NIC-400 互连的 AHB-Lite 同步端口
  它允许多层 AHB 主机访问大多数 SoC 外设。
- **S5**：连接到 AHB4 的 AHB-Lite 同步端口
  该总线桥控制对包括主系统管理工具（时钟，电源）、所有非安全 GPIO 以及一组补充外设的集群的访问，仅允许在简单应用中激活这个集群。

下表列出了多层 AHB 从端口的主要特性。

表 7. 多层 AHB 从端口主要特性

从端口	Nb	协议	数据总线宽度	同步性	时钟分频比
AHB2	S0	AHB-Lite	32	同步	1/1
SRAM1	S1	AHB-Lite	32	同步	1/1
SRAM2	S2	AHB-Lite	32	同步	1/1
SRAM3	S3	AHB-Lite	32	同步	1/1
MPU_NIC400	S4	AHB-Lite	32	同步	1/1
AHB4	S5	AHB-Lite	32	同步	1/1

2.4 AXIMC 寄存器

以下寄存器描述仅与 AXIM 互连相关，因为多层 AHB 互连没有配置寄存器。

x 是主机的编号（Mx 中的 x=10 时表示 M10）。主机的参数随它们的能力而变化，因此寄存器的数量可能会有所不同。

2.4.1 AXIMC 外设 ID4 寄存器 (AXIMC_PERIPH_ID_4)

AXIMC peripheral ID4 register

偏移地址：0x1FD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	K4COUNT[3:0]				JEP106CON[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **K4COUNT[3:0]**：寄存器文件大小

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码

2.4.2 AXIMC 外设 ID5 寄存器 (AXIMC_PERIPH_ID_5)

AXIMC peripheral ID5 register

偏移地址：0x1FD4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERIPH_ID_5[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PERIPH_ID_5[7:0]**：保留，未使用。

2.4.3 AXIMC 外设 ID6 寄存器 (AXIMC_PERIPH_ID_6)

AXIMC peripheral ID6 register

偏移地址：0x1FD8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERIPH_ID_6[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PERIPH_ID_6[7:0]**：保留，未使用。

2.4.4 AXIMC 外设 ID7 寄存器 (AXIMC_PERIPH_ID_7)

AXIMC peripheral ID7 register

偏移地址：0x1FDC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERIPH_ID_7[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PERIPH_ID_7[7:0]**：保留，未使用。

2.4.5 AXIMC 外设 ID0 寄存器 (AXIMC_PERIPH_ID_0)

AXIMC peripheral ID0 register

偏移地址：0x1FE0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERIPH_ID_0[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PERIPH_ID_0[7:0]**：产品编号 [7:0]

2.4.6 AXIMC 外设 ID1 寄存器 (AXIMC_PERIPH_ID_1)

AXIMC peripheral ID1 register

偏移地址：0x1FE4

复位值：0x0000 00B4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERIPH_ID_1[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PERIPH_ID_1[7:0]**：JEP106 标识 [3:0]，产品编号 [11:8]

2.4.7 AXIMC 外设 ID2 寄存器 (AXIMC_PERIPH_ID_2)

AXIMC peripheral ID2 register

偏移地址：0x1FE8

复位值：0x0000 003B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERIPH_ID_2[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PERIPH_ID_2[7:0]**：产品版本，JEP106 代码标志，JEP106 标识 [6:4]

2.4.8 AXIMC 外设 ID3 寄存器 (AXIMC_PERIPH_ID_3)

AXIMC peripheral ID3 register

偏移地址：0x1FEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REV_AND[3:0]				CUST_MOD_NUM[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REV_AND[3:0]**：客户版本
0：无

位 3:0 **CUST_MOD_NUM[3:0]**：客户修改
0：无

2.4.9 AXIMC 组件 ID0 寄存器 (AXIMC_COMP_ID_0)

AXIMC component ID0 register

偏移地址：0x1FF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：报头位 [7:0]

2.4.10 AXIMC 组件 ID1 寄存器 (AXIMC_COMP_ID_1)

AXIMC component ID1 register

偏移地址：0x1FF4

复位值：0x0000 00F0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件类
0xF：通用 IP 组件类

位 3:0 **PREAMBLE[11:8]**：报头位 [11:8]
0x0：公共 ID 值

2.4.11 AXIMC 组件 ID2 寄存器 (AXIMC_COMP_ID_2)

AXIMC component ID2 register

偏移地址：0x1FF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：报头位 [19:12]

2.4.12 AXIMC 组件 ID3 寄存器 (AXIMC_COMP_ID_3)

AXIMC component ID3 register

偏移地址：0x1FFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：报头位 [27:20]

2.4.13 AXIMC 主机 x 封装功能寄存器 (AXIMC_Mx_FN_MOD2)

AXIMC master x packing functionality register

偏移地址：0x42024 + 0x1000 * x（x = 0 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BYPASS_MERGE
															rw

位 31:1 保留，必须保持复位值。

位 0 **BYPASS_MERGE**：禁止节拍封装，以匹配输出数据宽度
未对齐事务不会与输入数据字边界重新对齐。
0：正常工作
1：禁止封装

2.4.14 AXIMC 主机 x AHB 转换覆盖功能寄存器 (AXIMC_Mx_FN_MOD_AHB)

(AXIMC_Mx_FN_MOD_AHB)

偏移地址：0x42028 + 0x1000 * x（x = 0 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR_INC_OVERRIDE	RD_INC_OVERRIDE
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **WR_INC_OVERRIDE**：将所有 AHB-Lite 写入事务转换为一系列单节拍 AXI 事务，每个 AHB-Lite 写入节拍由 AXI 缓存的写入响应确认。

0：禁止覆盖

1：使能覆盖

位 0 **RD_INC_OVERRIDE**：将所有 AHB-Lite 读取事务转换为一系列单节拍 AXI 事务。

0：禁止覆盖

1：使能覆盖

2.4.15 AXIMC 主机 x 读取优先级寄存器 (AXIMC_Mx_READ_QOS)

AXIMC master x read priority register

偏移地址：0x42100 + 0x1000 * x（x = 0 到 2）

复位值：0x0000 0006、0x0000 0005、0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **AR_QOS[3:0]**：读取通道 QoS 设置

0x0：最低优先级

0xF：最高优先级

2.4.16 AXIMC 主机 x 写入优先级寄存器 (AXIMC_Mx_WRITE_QOS)

AXIMC master x write priority register

偏移地址：0x42104 + 0x1000 * x（x = 0 到 2）

复位值：0x0000 0006、0x0000 0005、0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **AW_QOS[3:0]**：写入通道 QoS 设置
0x0：最低优先级
0xF：最高优先级

2.4.17 AXIMC 主机 x 发出能力覆盖功能寄存器 (AXIMC_Mx_FN_MOD)

AXIMC master x issuing capability override functionality register

偏移地址：0x42108 + 0x1000 * x（x = 0 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **WRITE_ISS_OVERRIDE**：覆盖 AMIB 写入发出能力
0：正常发出能力
1：将发出能力强制置 1

位 0 **READ_ISS_OVERRIDE**：覆盖 AMIB 读取发出能力
0：正常发出能力
1：将发出能力强制置 1

2.4.18 AXIMC 主机 x 封装功能寄存器 (AXIMC_Mx_FN_MOD2)

AXIMC master x packing functionality register

偏移地址：0x42024 + 0x1000 * x（x = 5 到 6）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BYPASS_MERGE
															rw

位 31:1 保留，必须保持复位值。

位 0 **BYPASS_MERGE**：禁止节拍封装，以匹配输出数据宽度
未对齐事务不会与输入数据字边界重新对齐。
0：正常工作
1：禁止封装

2.4.19 AXIMC 主机 x AHB 转换覆盖功能寄存器 (AXIMC_Mx_FN_MOD_AHB)

AXIMC master x AHB conversion override functionality register

偏移地址：0x42028 + 0x1000 * x（x = 5 到 6）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR_INC_OVERRIDE	RD_INC_OVERRIDE
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **WR_INC_OVERRIDE**：将所有 AHB-Lite 写入事务转换为一系列单节拍 AXI 事务，每个 AHB-Lite 写入节拍由 AXI 缓存的写入响应确认。
0：禁止覆盖
1：使能覆盖

位 0 **RD_INC_OVERRIDE**：将所有 AHB-Lite 读取事务转换为一系列单节拍 AXI 事务。
0：禁止覆盖
1：使能覆盖

2.4.20 AXIMC 主机 x 读取优先级寄存器 (AXIMC_Mx_READ_QOS)

AXIMC master x read priority register

偏移地址：0x42100 + 0x1000 * x（x = 5 到 6）

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **AR_QOS[3:0]**：读取通道 QoS 设置
0x0：最低优先级
0xF：最高优先级

2.4.21 AXIMC 主机 x 写入优先级寄存器 (AXIMC_Mx_WRITE_QOS)

AXIMC master x write priority register

偏移地址：0x42104 + 0x1000 * x（x = 5 到 6）

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **AW_QOS[3:0]**：写入通道 QoS 设置
0x0：最低优先级
0xF：最高优先级

2.4.22 AXIMC 主机 x 发出能力覆盖功能寄存器 (AXIMC_Mx_FN_MOD)

AXIMC master x issuing capability override functionality register

偏移地址：0x42108 + 0x1000 * x（x = 5 到 6）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **WRITE_ISS_OVERRIDE**：覆盖 AMIB 写入发出能力
0：正常发出能力
1：将发出能力强制置 1

位 0 **READ_ISS_OVERRIDE**：覆盖 AMIB 读取发出能力
0：正常发出能力
1：将发出能力强制置 1

2.4.23 AXIMC 主机 x 读取优先级寄存器 (AXIMC_Mx_READ_QOS)

AXIMC master x read priority register

偏移地址：0x42100 + 0x1000 * x（x = 3 到 4）

复位值：0x0000 0007，0x0000 000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **AR_QOS[3:0]**：读取通道 QoS 设置
0x0：最低优先级
0xF：最高优先级

2.4.24 AXIMC 主机 x 写入优先级寄存器 (AXIMC_Mx_WRITE_QOS)

AXIMC master x write priority register

偏移地址：0x42104 + 0x1000 * x（x = 3 到 4）

复位值：0x0000 0007，0x0000 000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **AW_QOS[3:0]**：写入通道 QoS 设置
0x0：最低优先级
0xF：最高优先级

2.4.25 AXIMC 主机 x 发出能力覆盖功能寄存器 (AXIMC_Mx_FN_MOD)

AXIMC master x issuing capability override functionality register

偏移地址：0x42108 + 0x1000 * x（x = 3 到 4）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **WRITE_ISS_OVERRIDE**：覆盖 AMIB 写入发出能力
0：正常发出能力
1：将发出能力强制置 1

位 0 **READ_ISS_OVERRIDE**：覆盖 AMIB 读取发出能力
0：正常发出能力
1：将发出能力强制置 1

2.4.26 AXIMC 主机 x 读取优先级寄存器 (AXIMC_Mx_READ_QOS)

AXIMC master x read priority register

偏移地址：0x42100 + 0x1000 * x（x = 7 到 8）

复位值：0x0000 0008，0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **AR_QOS[3:0]**：读取通道 QoS 设置
0x0：最低优先级
0xF：最高优先级

2.4.27 AXIMC 主机 x 写入优先级寄存器 (AXIMC_Mx_WRITE_QOS)

AXIMC master x write priority register

偏移地址：0x42104 + 0x1000 * x（x = 7 到 8）

复位值：0x0000 0008，0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **AW_QOS[3:0]**：写入通道 QoS 设置
0x0：最低优先级
0xF：最高优先级

2.4.28 AXIMC 主机 x 发出能力覆盖功能寄存器 (AXIMC_Mx_FN_MOD)

AXIMC master x issuing capability override functionality register

偏移地址：0x42108 + 0x1000 * x（x = 7 到 8）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **WRITE_ISS_OVERRIDE**：覆盖 AMIB 写入发出能力
- 0：正常发出能力
- 1：将发出能力强制置 1

位 0 **READ_ISS_OVERRIDE**：覆盖 AMIB 读取发出能力
- 0：正常发出能力
- 1：将发出能力强制置 1

2.4.29 AXIMC 长突发能力抑制寄存器 (AXIMC_M4_FN_MOD_LB)

AXIMC long burst capability inhibition register

偏移地址：0x4602C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FN_MOD_LB
															rw

位 31:1 保留，必须保持复位值。

位 0 **FN_MOD_LB**：控制长突发的突发中断
- 0：ASIB 输出不生成长突发
- 1：ASIB 输出可生成长突发

2.4.30 AXIMC 主机 x 读取优先级寄存器 (AXIMC_Mx_READ_QOS)

AXIMC master x read priority register

偏移地址：0x42100 + 0x1000 * x（x = 9 到 11）

复位值：0x0000 0002，0x0000 000B，0x0000 0007

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **AR_QOS[3:0]**：读取通道 QoS 设置
0x0：最低优先级
0xF：最高优先级

2.4.31 AXIMC 主机 x 写入优先级寄存器 (AXIMC_Mx_WRITE_QOS)

AXIMC master x write priority register

偏移地址：0x42104 + 0x1000 * x（x = 9 到 11）

复位值：0x0000 0002，0x0000 000B，0x0000 0007

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **AW_QOS[3:0]**：写入通道 QoS 设置
0x0：最低优先级
0xF：最高优先级

2.4.32 AXIMC 主机 x 发出能力覆盖功能寄存器 (AXIMC_Mx_FN_MOD)

AXIMC master x issuing capability override functionality register

偏移地址：0x42108 + 0x1000 * x（x = 9 到 11）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **WRITE_ISS_OVERRIDE**：覆盖 AMIB 写入发出能力
- 0：正常发出能力
- 1：将发出能力强制置 1

位 0 **READ_ISS_OVERRIDE**：覆盖 AMIB 读取发出能力
- 0：正常发出能力
- 1：将发出能力强制置 1

2.4.33 AXIMC 寄存器映射

表 8. AXIMC 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1FD0	**AXIMC_PERIPH_ID_4**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	K4COUNT[3:0]				JEP106CON[3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0x1FD4	**AXIMC_PERIPH_ID_5**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PERIPH_ID_5[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x1FD8	**AXIMC_PERIPH_ID_6**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PERIPH_ID_6[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x1FDC	**AXIMC_PERIPH_ID_7**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PERIPH_ID_7[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

表 8. AXIMC 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1FE0	**AXIMC_PERIPH_ID_0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERIPH_ID_0[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x1FE4	**AXIMC_PERIPH_ID_1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERIPH_ID_1[7:0]							
	Reset value																									1	0	1	1	0	1	0	0
0x1FE8	**AXIMC_PERIPH_ID_2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERIPH_ID_2[7:0]							
	Reset value																									0	0	1	1	1	0	1	1
0x1FEC	**AXIMC_PERIPH_ID_3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REV_AND[3:0]				CUST_MOD_NUM[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0x1FF0	**AXIMC_COMP_ID_0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0x1FF4	**AXIMC_COMP_ID_1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
	Reset value																									1	1	1	1	0	0	0	0
0x1FF8	**AXIMC_COMP_ID_2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0x1FFC	**AXIMC_COMP_ID_3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1
0x2000 - 0x42020	Reserved	Reserved																															
0x42024	**AXIMC_M0_FN_MOD2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BYPASS_MERGE
	Reset value																																0
0x42028	**AXIMC_M0_FN_ MOD_AHB**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR_INC_OVERRIDE	RD_INC_OVERRIDE
	Reset value																															0	0
0x4202C - 0x420FC	Reserved	Reserved																															
0x42100	**AXIMC_M0_ READ_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
	Reset value																													0	1	1	0

表 8. AXIMC 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x42104	**AXIMC_M0_ WRITE_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]			
	Reset value																													0	1	1	0
0x42108	**AXIMC_M0_FN_MOD**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRTIE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x4210C - 0x43020	Reserved	Reserved																															
0x43024	**AXIMC_M1_FN_MOD2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BYPASS_MERGE
	Reset value																																0
0x43028	**AXIMC_M1_FN_ MOD_AHB**	Res.	Res.	Res.	Res	Res	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR_INC_OVERRIDE	RD_INC_OVERRIDE
	Reset value																															0	0
0x4302C - 0x430FC	Reserved	Reserved																															
0x43100	**AXIMC_M1_ READ_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
	Reset value																													0	1	0	1
0x43104	**AXIMC_M1_ WRITE_QOS**	Res.	Res.	Res.	Res	Res	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]		
	Reset value																													0	1	0	1
0x43108	**AXIMC_M1_FN_MOD**	Res.	Res.	Res.	Res	Res	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRTIE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x4310C - 0x44020	Reserved	Reserved																															

表 8. AXIMC 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x44024	**AXIMC_M2_FN_MOD2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BYPASS_MERGE
	Reset value																																0
0x44028	**AXIMC_M2_FN_ MOD_AHB**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR_INC_OVERRIDE	RD_INC_OVERRIDE
	Reset value																															0	0
0x4402C - 0x440FC	Reserved	Reserved																															
0x44100	**AXIMC_M2_ READ_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
	Reset value																													0	1	0	1
0x44104	**AXIMC_M2_ WRITE_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]			
	Reset value																													0	1	0	1
0x44108	**AXIMC_M2_FN_MOD**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRTIE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x4410C - 0x450FC	Reserved	Reserved																															
0x45100	**AXIMC_M3_ READ_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
	Reset value																													0	1	1	1
0x45104	**AXIMC_M3_ WRITE_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]			
	Reset value																													0	1	1	1

表 8. AXIMC 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x45108	**AXIMC_M3_FN_MOD**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRTIE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x4510C - 0x46028	Reserved	Reserved																															
0x4602C	**AXIMC_M4_FN_MOD_LB**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FN_MOD_LB
	Reset value																																0
0x46030 - 0x460FC	Reserved	Reserved																															
0x46100	**AXIMC_M4_ READ_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
	Reset value																													1	1	0	0
0x46104	**AXIMC_M4_ WRITE_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]			
	Reset value																													1	1	0	0
0x46108	**AXIMC_M4_FN_MOD**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRTIE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x4C10C - 0x47020	Reserved	Reserved																															
0x47024	**AXIMC_M5_FN_MOD2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BYPASS_MERGE
	Reset value																																0
0x47028	**AXIMC_M5_FN_ MOD_AHB**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR_INC_OVERRIDE	RD_INC_OVERRIDE
	Reset value																															0	0
0x4702C - 0x470FC	Reserved	Reserved																															

表 8. AXIMC 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x47100	**AXIMC_ M5_ READ_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
	Reset value																													0	1	0	0
0x47104	**AXIMC_M5_ WRITE_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]			
	Reset value																													0	1	0	0
0x47108	**AXIMC_M5_FN_MOD**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRTIE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x4710C - 0x48020	Reserved	Reserved																															
0x48024	**AXIMC_M6_FN_MOD2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BYPASS_MERGE
	Reset value																																0
0x48028	**AXIMC_M6_FN_ MOD_AHB**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR_INC_OVERRIDE	RD_INC_OVERRIDE
	Reset value																															0	0
0x4802C - 0x480FC	Reserved	Reserved																															
0x48100	**AXIMC_M6_ READ_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
	Reset value																													0	1	0	0
0x48104	**AXIMC_M6_ WRITE_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]			
	Reset value																													0	1	0	0

表 8. AXIMC 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x48108	**AXIMC_M6_FN_MOD**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRTIE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x4810C - 0x490FC	Reserved	Reserved																															
0x49100	**AXIMC_M7_ READ_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
	Reset value																													1	0	0	0
0x49104	**AXIMC_M7_ WRITE_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]			
	Reset value																													1	0	0	0
0x49108	**AXIMC_M7_FN_MOD**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FN_MOD[1:0]	
	Reset value																															0	0
0x4910C - 0x4A0FC	Reserved	Reserved																															
0x4A100	**AXIMC_M8_ READ_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
	Reset value																													1	0	1	0
0x4A104	**AXIMC_M8_ WRITE_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]			
	Reset value																													1	0	1	0
0x4A108	**AXIMC_M8_FN_MOD**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FN_MOD[1:0]	
	Reset value																															0	0
0x4A10C - 0x4B0FC	Reserved	Reserved																															
0x4B100	**AXIMC_M9_ READ_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
	Reset value																													0	0	1	0

表 8. AXIMC 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x4B104	**AXIMC_M9_ WRITE_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]			
	Reset value																													0	0	1	0
0x4B108	**AXIMC_M9_FN_MOD**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRTIE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x4B10C - 0x4C0FC	Reserved	Reserved																															
0x4C100	**AXIMC_M10_ READ_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
	Reset value																													1	0	1	1
0x4C104	**AXIMC_M10_ WRITE_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]			
	Reset value																													1	0	1	1
0x4C108	**AXIMC_M10_FN_MOD**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRTIE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x4C10C - 0x4D0FC	Reserved	Reserved																															
0x4D100	**AXIMC_M11_ READ_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS[3:0]			
	Reset value																													0	1	1	1
0x4D104	**AXIMC_M11_ WRITE_QOS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS[3:0]			
	Reset value																													0	1	1	1

表 8. AXIMC 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x4D108	**AXIMC_M11_FN_MOD**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRTIE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0

有关寄存器边界地址的信息，请参见*第147 页的第 2.5 节*。

2.5 存储结构

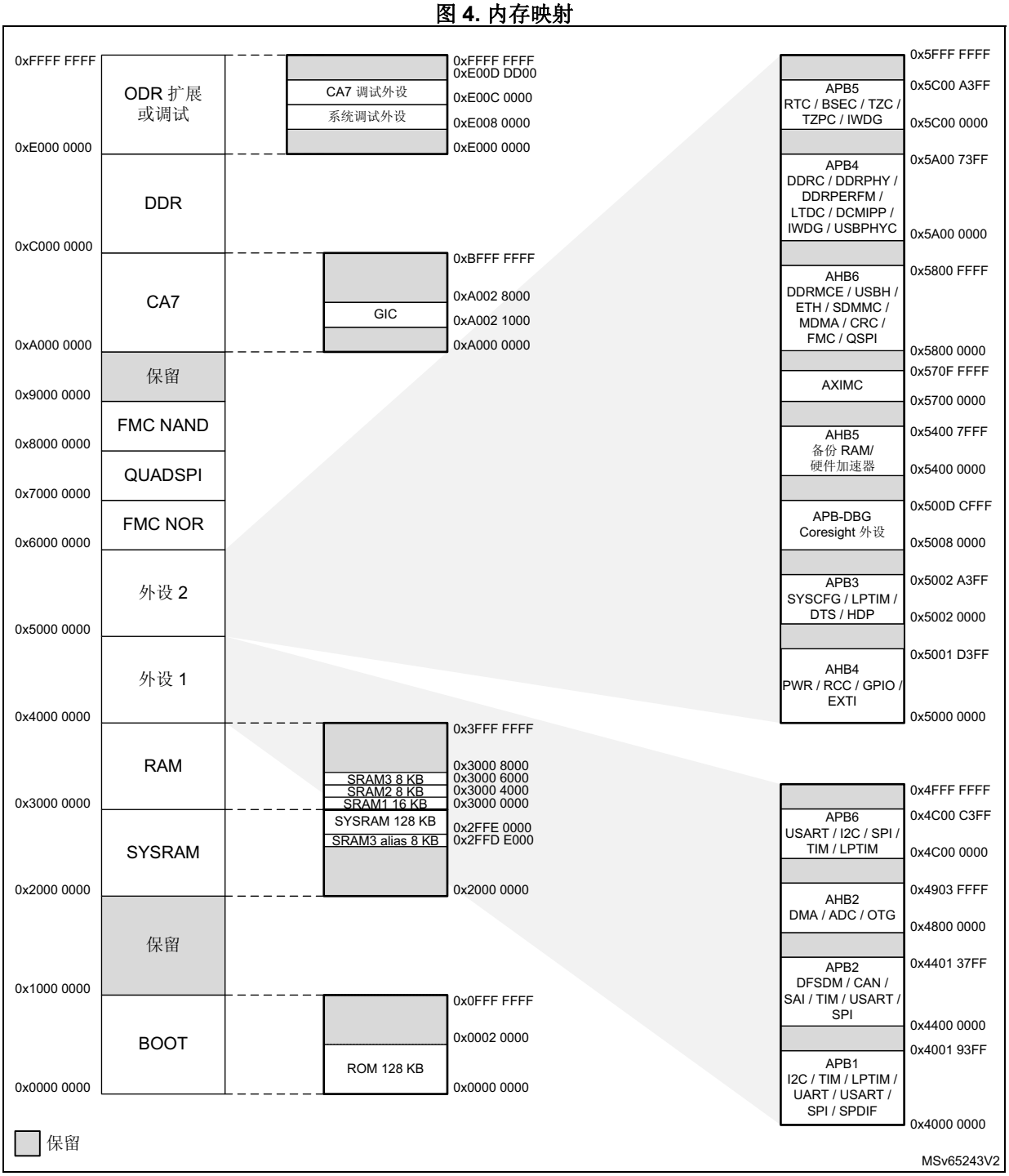
2.5.1 简介

程序内存、数据内存、寄存器和 I/O 端口排列在同一个线性（即地址连续）的 4 GB 地址空间内。

各字节按小端格式在内存中编码。一个存储字单元中编号最低的字节被视为该字的最低有效字节，而编号最高的字节被视为最高有效字节。

2.5.2 内存映射和寄存器边界地址

图 4. 内存映射



未分配给片上内存和外设的所有内存映射区域均视为“保留区”。有关可用内存和寄存器区域的详细映射，请参见下表。

下表给出了设备中可用外设的边界地址。

外设映射

表 9. 寄存器边界地址

总线	边界地址	大小（字节）	外设	外设寄存器映射
Cortex-A7 内部	0xA0026000 - 0xA0027FFF	8 KB	GICV	*GIC 虚拟 CPU 接口 (GICV)*
	0xA0024000 - 0xA0025FFF	8 KB	GICH	*GIC 虚拟接口控制，通用 (GICH)*
	0xA0022000 - 0xA0023FFF	8 KB	GICC	*GIC CPU 接口 (GICC)*
	0xA0021000 - 0xA0021FFF	4 KB	GICD	*GIC 分配器 (GICD)*
	0xA0000000 - 0xA0020FFF	132 KB	保留	-
APB5	0x5C00A400 - 0x5FFFFFFF	65495 KB	保留	-
	0x5C00A000 - 0x5C00A3FF	1 KB	TAMP	*TAMP 寄存器*
	0x5C009000 - 0x5C009FFF	4 KB	保留	-
	0x5C008000 - 0x5C008FFF	4 KB	STGENC	*STGEN 寄存器*
	0x5C007400 - 0x5C007FFF	3 KB	保留	-
	0x5C007000 - 0x5C0073FF	1 KB	ETZPC	*ETZPC 寄存器*
	0x5C006000 - 0x5C006FFF	4 KB	TZC	*TZC 寄存器*
	0x5C005000 - 0x5C005FFF	4 KB	BSEC	*BSEC 寄存器*
	0x5C004400 - 0x5C004FFF	3 KB	保留	-
	0x5C004000 - 0x5C0043FF	1 KB	RTC	*RTC 寄存器*
	0x5C003400 - 0x5C003FFF	3 KB	保留	-
	0x5C003000 - 0x5C0033FF	1 KB	IWDG1	*IWDG 寄存器*
	0x5C000000 - 0x5C002FFF	12 KB	保留	-
APB4	0x5A007400 - 0x5BFFFFFF	32739 KB	保留	-
	0x5A007000 - 0x5A0073FF	1 KB	DDRPERFM	*DDRPERFM 寄存器*
	0x5A006000 - 0x5A006FFF	4 KB	USBPHYC	*USBPHYC 寄存器*
	0x5A005000 - 0x5A005FFF	4 KB	STGENR	*STGEN 寄存器*
	0x5A004000 - 0x5A004FFF	4 KB	DDRPHYC	*PUBL 寄存器*
	0x5A003000 - 0x5A003FFF	4 KB	DDRCTRL	*DDRCTRL 寄存器*
	0x5A002400 - 0x5A002FFF	3 KB	保留	-
	0x5A002000 - 0x5A0023FF	1 KB	IWDG2	*IWDG 寄存器*
	0x5A001400 - 0x5A001FFF	3 KB	保留	-
	0x5A001000 - 0x5A0013FF	1 KB	LTDC	*LTDC 寄存器*
	0x5A000000 - 0x5A000FFF	4 KB	DCMIPP	DCMIPP 寄存器

表 9. 寄存器边界地址（续）

总线	边界地址	大小（字节）	外设	外设寄存器映射
AHB6	0x58010000 - 0x59FFFFFF	32704 KB	保留	-
	0x5800E000 - 0x5800FFFF	8 KB	ETH2	*以太网寄存器*
	0x5800D000 - 0x5800DFFF	4 KB	USBH_EHCI	*USBH 寄存器*
	0x5800C000 - 0x5800CFFF	4 KB	USBH_OHCI	*USBH 寄存器*
	0x5800A000 - 0x5800BFFF	8 KB	ETH1	*以太网寄存器*
	0x58009000 - 0x58009FFF	4 KB	CRC1	*CRC 寄存器*
	0x58008000 - 0x58008FFF	4 KB	DLYBSD2	*DLYB 寄存器*
	0x58007000 - 0x58007FFF	4 KB	SDMMC2	*SDMMC 寄存器*
	0x58006000 - 0x58006FFF	4 KB	DLYBSD1	*DLYB 寄存器*
	0x58005000 - 0x58005FFF	4 KB	SDMMC1	*SDMMC 寄存器*
	0x58004000 - 0x58004FFF	4 KB	DLYBQS	*DLYB 寄存器*
	0x58003000 - 0x58003FFF	4 KB	QUADSPI	*QUADSPI 寄存器*
	0x58002000 - 0x58002FFF	4 KB	FMC	*FMC 寄存器映射*
	0x58001400 - 0x58001FFF	3 KB	保留	-
	0x58001000 - 0x580003FF	1 KB	DDRMCE	*MCE 寄存器*
	0x58000000 - 0x58000FFF	4 KB	MDMA	*MDMA 寄存器*
AXIM	0x57100000 - 0x57FFFFFF	15360 KB	保留	-
	0x57000000 - 0x570FFFFF	1024 KB	AXIMC	*AXIMC 寄存器*
AHB5	0x54008000 - 0x56FFFFFF	49120 KB	保留	-
	0x54006000 - 0x54007FFF	8 KB	PKA	*PKA 寄存器*
	0x54005400 - 0x54005FFF	3 KB	保留	-
	0x54005000 - 0x540053FF	1 KB	SAES	*SAES 寄存器*
	0x54004400 - 0x54004FFF	3 KB	保留	-
	0x54004000 - 0x540043FF	1 KB	RNG	*RNG 寄存器*
	0x54003400 - 0x54003FFF	3 KB	保留	-
	0x54003000 - 0x540033FF	1 KB	HASH	*HASH 寄存器*
	0x54002400 - 0x54002FFF	3 KB	保留	-
	0x54002000 - 0x540023FF	1 KB	CRYP(1)	*CRYP 寄存器*
	0x54000000 - 0x54001FFF	8 KB	BKPSRAM	-

表 9. 寄存器边界地址（续）

总线	边界地址	大小（字节）	外设	外设寄存器映射
APB3	0x5002A400 - 0x5007FFFF	343 KB	保留	-
	0x5002A000 - 0x5002A3FF	1 KB	HDP	*HDP 寄存器*
	0x50028400 - 0x50029FFF	7 KB	保留	-
	0x50028000 - 0x500283FF	1 KB	DTS	*DTS 寄存器*
	0x50025400 - 0x50027FFF	11 KB	保留	-
	0x50025000 - 0x500253FF	1 KB	VREFBUF	*VREFBUF 寄存器*
	0x50024400 - 0x50024FFF	3 KB	保留	-
	0x50024000 - 0x500243FF	1 KB	LPTIM5	*LPTIM 寄存器*
	0x50023400 - 0x50023FFF	3 KB	保留	-
	0x50023000 - 0x500233FF	1 KB	LPTIM4	*LPTIM 寄存器*
	0x50022400 - 0x50022FFF	3 KB	保留	-
	0x50022000 - 0x500223FF	1 KB	LPTIM3	*LPTIM 寄存器*
	0x50021400 - 0x50021FFF	3 KB	保留	-
	0x50021000 - 0x500213FF	1 KB	LPTIM2	*LPTIM 寄存器*
	0x50020400 - 0x50020FFF	3 KB	保留	-
	0x50020000 - 0x500203FF	1 KB	SYSCFG	*SYSCFG 寄存器*

表 9. 寄存器边界地址（续）

总线	边界地址	大小（字节）	外设	外设寄存器映射
AHB4	0x5000D400 - 0x5001FFFF	75 KB	保留	-
	0x5000D000 - 0x5000D3FF	1 KB	EXTI	*EXTI 寄存器*
	0x5000A400 - 0x5000CFFF	11 KB	保留	-
	0x5000A000 - 0x5000A3FF	1 KB	GPIOI	*GPIO 寄存器*
	0x50009400 - 0x50009FFF	3 KB	保留	-
	0x50009000 - 0x500093FF	1 KB	GPIOH	*GPIO 寄存器*
	0x50008400 - 0x50008FFF	3 KB	保留	-
	0x50008000 - 0x500083FF	1 KB	GPIOG	*GPIO 寄存器*
	0x50007400 - 0x50007FFF	3 KB	保留	-
	0x50007000 - 0x500073FF	1 KB	GPIOF	*GPIO 寄存器*
	0x50006400 - 0x50006FFF	3 KB	保留	-
	0x50006000 - 0x500063FF	1 KB	GPIOE	*GPIO 寄存器*
	0x50005400 - 0x50005FFF	3 KB	保留	-
	0x50005000 - 0x500053FF	1 KB	GPIOD	*GPIO 寄存器*
	0x50004400 - 0x50004FFF	3 KB	保留	-
	0x50004000 - 0x500043FF	1 KB	GPIOC	*GPIO 寄存器*
	0x50003400 - 0x50003FFF	3 KB	保留	-
	0x50003000 - 0x500033FF	1 KB	GPIOB	*GPIO 寄存器*
	0x50002400 - 0x50002FFF	3 KB	保留	-
	0x50002000 - 0x500023FF	1 KB	GPIOA	*GPIO 寄存器*
	0x50001400 - 0x50001FFF	3 KB	保留	-
	0x50001000 - 0x500013FF	1 KB	PWR	*PWR 寄存器*
	0x50000000 - 0x50000FFF	4 KB	RCC	*RCC 寄存器*

表 9. 寄存器边界地址（续）

总线	边界地址	大小（字节）	外设	外设寄存器映射
APB6	0x4C00C400 - 0x4FFFFFFF	65487 KB	保留	-
	0x4C00C000 - 0x4C00C3FF	1 KB	TIM17	*TIM16/TIM17 寄存器*
	0x4C00B400 - 0x4C00BFFF	3 KB	保留	-
	0x4C00B000 - 0x4C00B3FF	1 KB	TIM16	*TIM16/TIM17 寄存器*
	0x4C00A400 - 0x4C00AFFF	3 KB	保留	-
	0x4C00A000 - 0x4C00A3FF	1 KB	TIM15	*TIM15 寄存器*
	0x4C009400 - 0x4C009FFF	3 KB	保留	-
	0x4C009000 - 0x4C0093FF	1 KB	TIM14	*TIM13/TIM14 寄存器*
	0x4C008400 - 0x4C008FFF	3 KB	保留	-
	0x4C008000 - 0x4C0083FF	1 KB	TIM13	*TIM13/TIM14 寄存器*
	0x4C007400 - 0x4C007FFF	3 KB	保留	-
	0x4C007000 - 0x4C0073FF	1 KB	TIM12	*TIM12 寄存器*
	0x4C006400 - 0x4C006FFF	3 KB	保留	-
	0x4C006000 - 0x4C0063FF	1 KB	I2C5	*I2C 寄存器*
	0x4C005400 - 0x4C005FFF	3 KB	保留	-
	0x4C005000 - 0x4C0053FF	1 KB	I2C4	*I2C 寄存器*
	0x4C004400 - 0x4C004FFF	3 KB	保留	-
	0x4C004000 - 0x4C0043FF	1 KB	I2C3	*I2C 寄存器*
	0x4C003400 - 0x4C003FFF	3 KB	保留	-
	0x4C003000 - 0x4C0033FF	1 KB	SPI5	*SPI/I2S 寄存器*
	0x4C002400 - 0x4C002FFF	3 KB	保留	-
	0x4C002000 - 0x4C0023FF	1 KB	SPI4	*SPI/I2S 寄存器*
	0x4C001400 - 0x4C001FFF	3 KB	保留	-
	0x4C001000 - 0x4C0013FF	1 KB	USART2	*USART 寄存器*
	0x4C000400 - 0x4C000FFF	3 KB	保留	-
	0x4C000000 - 0x4C0003FF	1 KB	USART1	*USART 寄存器*

表 9. 寄存器边界地址（续）

总线	边界地址	大小（字节）	外设	外设寄存器映射
AHB2	0x49040000 - 0x4BFFFFFF	48896 KB	保留	-
	0x49000000 - 0x4903FFFF	256 KB	OTG	**OTG *寄存器***
	0x48006400 - 0x48FFFFFF	16359 KB	保留	-
	0x48006000 - 0x480063FF	1 KB	DMAMUX2	*DMAMUX 寄存器*
	0x48004400 - 0x48004FFF	3 KB	保留	-
	0x48005000 - 0x480053FF	1 KB	DMA3	*DMA 寄存器*
	0x48004400 - 0x48004FFF	3 KB	保留	-
	0x48004000 - 0x480043FF	1 KB	ADC2	*ADC 寄存器*
	0x48003400 - 0x48003FFF	3 KB	保留	-
	0x48003000 - 0x480033FF	1 KB	ADC1	*ADC 寄存器*
	0x48002400 - 0x48002FFF	3 KB	保留	-
	0x48002000 - 0x480023FF	1 KB	DMAMUX1	*DMAMUX 寄存器*
	0x48001400 - 0x48001FFF	3 KB	保留	-
	0x48001000 - 0x480013FF	1 KB	DMA2	*DMA 寄存器*
	0x48000400 - 0x48000FFF	3 KB	保留	-
	0x48000000 - 0x480003FF	1 KB	DMA1	*DMA 寄存器*

表 9. 寄存器边界地址（续）

总线	边界地址	大小（字节）	外设	外设寄存器映射
APB2	0x44013800 - 0x47FFFFFF	65458 KB	保留	-
	0x44011000 - 0x440137FF	10 KB	CANSRAM	-
	0x44010400 - 0x44010FFF	3 KB	保留	-
	0x44010000 - 0x440103FF	1 KB	CCU	*CCU 寄存器*
	0x4400F400 - 0x4400FFFF	3 KB	保留	-
	0x4400F000 - 0x4400F3FF	1 KB	FDCAN2	*FDCAN 寄存器*
	0x4400E400 - 0x4400EFFF	3 KB	保留	-
	0x4400E000 - 0x4400E3FF	1 KB	FDCAN1	*FDCAN 寄存器*
	0x4400D800 - 0x4400DFFF	2 KB	保留	-
	0x4400D000 - 0x4400D7FF	2 KB	DFSDM1	*DFSDM 通道 y 寄存器 (y=0..3)* *DFSDM 滤波器 x 模块寄存器 (x=0..1)*
	0x4400B400 - 0x4400CFFF	7 KB	保留	-
	0x4400B000 - 0x4400B3FF	1 KB	SAI2	*SAI 寄存器*
	0x4400A400 - 0x4400AFFF	3 KB	保留	-
	0x4400A000 - 0x4400A3FF	1 KB	SAI1	*SAI 寄存器*
	0x44004400 - 0x44009FFF	23 KB	保留	-
	0x44004000 - 0x440043FF	1 KB	SPI1	*SPI/I2S 寄存器*
	0x44003400 - 0x44003FFF	3 KB	保留	-
	0x44003000 - 0x440033FF	1 KB	USART6	*USART 寄存器*
	0x44001400 - 0x44002FFF	7 KB	保留	-
	0x44001000 - 0x440013FF	1 KB	TIM 8	*TIM1/TIM8 寄存器*
	0x44000400 - 0x44000FFF	3 KB	保留	-
	0x44000000 - 0x440003FF	1 KB	TIM 1	*TIM1/TIM8 寄存器*

表 9. 寄存器边界地址（续）

总线	边界地址	大小（字节）	外设	外设寄存器映射
APB1	0x40019400 - 0x43FFFFFF	65435 KB	保留	-
	0x40019000 - 0x400193FF	1 KB	UART8	*USART 寄存器*
	0x40018400 - 0x40018FFF	3 KB	保留	-
	0x40018000 - 0x400183FF	1 KB	UART7	*USART 寄存器*
	0x40013400 - 0x40017FFF	19 KB	保留	-
	0x40013000 - 0x400133FF	1 KB	I2C2	*I2C 寄存器*
	0x40012400 - 0x40012FFF	3 KB	保留	-
	0x40012000 - 0x400123FF	1 KB	I2C1	*I2C 寄存器*
	0x40011400 - 0x40011FFF	3 KB	保留	-
	0x40011000 - 0x400113FF	1 KB	UART5	*USART 寄存器*
	0x40010400 - 0x40010FFF	3 KB	保留	-
	0x40010000 - 0x400103FF	1 KB	UART4	*USART 寄存器*
	0x4000F400 - 0x4000FFFF	3 KB	保留	-
	0x4000F000 - 0x4000F3FF	1 KB	USART3	*USART 寄存器*
	0x4000D400 - 0x4000EFFF	7 KB	保留	-
	0x4000D000 - 0x4000D3FF	1 KB	SPDIFRX	*SPDIFRX 接口寄存器*
	0x4000C400 - 0x4000CFFF	3 KB	保留	-
	0x4000C000 - 0x4000C3FF	1 KB	SPI3	*SPI/I2S 寄存器*
	0x4000B400 - 0x4000BFFF	3 KB	保留	-
	0x4000B000 - 0x4000B3FF	1 KB	SPI2	*SPI/I2S 寄存器*
	0x40009400 - 0x4000AFFF	7 KB	保留	-
	0x40009000 - 0x400093FF	1 KB	LPTIM1	*LPTIM 寄存器*
	0x40005400 - 0x40008FFF	15 KB	保留	-
	0x40005000 - 0x400053FF	1 KB	TIM7	*TIM6/TIM7 寄存器*
	0x40004400 - 0x40004FFF	3 KB	保留	-
	0x40004000 - 0x400043FF	1 KB	TIM6	*TIM6/TIM7 寄存器*
	0x40003400 - 0x40003FFF	3 KB	保留	-
	0x40003000 - 0x400033FF	1 KB	TIM5	*TIM2/TIM3/TIM4/TIM5 寄存器*
	0x40002400 - 0x40002FFF	3 KB	保留	-
	0x40002000 - 0x400023FF	1 KB	TIM4	*TIM2/TIM3/TIM4/TIM5 寄存器*
	0x40001400 - 0x40001FFF	3 KB	保留	-
	0x40001000 - 0x400013FF	1 KB	TIM3	*TIM2/TIM3/TIM4/TIM5 寄存器*
	0x40000400 - 0x40000FFF	3 KB	保留	-
	0x40000000 - 0x400003FF	1 KB	TIM2	*TIM2/TIM3/TIM4/TIM5 寄存器*

1. 请检查设备数据表以确认可用性。

调试器映射

表 10. 调试器寄存器边界地址

总线	边界地址	大小（字节）	外设
DAPBUS(1)	0xE0100000 - 0xFFFFFFFF	523264 KB	保留
	0xE00DD000 - 0xE00FFFFF	140 KB	保留
	0xE00DC000 - 0xE00DCFFF	4 KB	CA7_ETM0
	0xE00D9000 - 0xE00DBFFF	12 KB	保留
	0xE00D8000 - 0xE00D8FFF	4 KB	CA7_CTI0
	0xE00D2000 - 0xE00D7FFF	24 KB	保留
	0xE00D1000 - 0xE00D1FFF	4 KB	CA7_PMU0
	0xE00D0000 - 0xE00D0FFF	4 KB	CA7_DBG0
	0xE00C1000 - 0xE00CFFFF	60 KB	保留
	0xE00C0000 - 0xE00C0FFF	4 KB	CA7_ROM
	0xE0095000 - 0xE00BFFFF	172 KB	保留
	0xE0094000 - 0xE0094FFF	4 KB	CTI
	0xE0093000 - 0xE0093FFF	4 KB	TPIU
	0xE0092000 - 0xE0092FFF	4 KB	ETF
	0xE0091000 - 0xE0091FFF	4 KB	保留
	0xE0090000 - 0xE0090FFF	4 KB	跟踪子系统 ROM 表
	0xE0083000 - 0xE008FFFF	52 KB	保留
	0xE0082000 - 0xE0082FFF	4 KB	TSGEN1
	0xE0081000 - 0xE0081FFF	4 KB	DBG_MCU
	0xE0080000 - 0xE0080FFF	4 KB	DAP ROM Table
	0xE0000000 - 0xE007FFFF	512 KB	保留
调试 APB(2)	0x500DD000 - 0x53FFFFFF	64652 KB	保留
	0x500DC000 - 0x500DCFFF	4 KB	CA7_ETM0
	0x500D9000 - 0x500DBFFF	12 KB	保留
	0x500D8000 - 0x500D8FFF	4 KB	CA7_CTI0
	0x500D2000 - 0x500D7FFF	24 KB	保留
	0x500D1000 - 0x500D1FFF	4 KB	CA7_PMU0
	0x500D0000 - 0x500D0FFF	4 KB	CA7_DBG0
	0x500C1000 - 0x500CFFFF	60 KB	保留
	0x500C0000 - 0x500C0FFF	4 KB	CA7_ROM
	0x50095000 - 0x500BFFFF	172 KB	保留
	0x50094000 - 0x50094FFF	4 KB	CTI
	0x50093000 - 0x50093FFF	4 KB	TPIU

表 10. 调试器寄存器边界地址（续）

总线	边界地址	大小（字节）	外设
调试 APB(2)	0x50092000 - 0x50092FFF	4 KB	ETF
	0x50091000 - 0x50091FFF	4 KB	保留
	0x50090000 - 0x50090FFF	4 KB	跟踪子系统 ROM 表
	0x50083000 - 0x5008FFFF	52 KB	保留
	0x50082000 - 0x50082FFF	4 KB	TSGEN1
	0x50081000 - 0x50081FFF	4 KB	DBGMCU
	0x50080000 - 0x50080FFF	4 KB	DAP ROM 表

1. 只能通过调试器访问。
2. 可通过 Cortex®-A7 访问

3 启动、安全性和 OTP 控制 (BSEC)

3.1 简介

BSEC（启动、安全性和 OTP 控制）旨在控制 OTP（一次性编程）保险丝盒，用于设备配置和安全参数的片上非易失性存储。

3.2 BSEC 主要特性

- 32 位 APB4 接口
- 4096 个 OTP 位（含 3072 个有效位）
- 有效的电源 **pwrok** 输入，以保证 OTP 只在 VDD/VDD1 有效范围内操作
- 用于与外部代理通信和存储启动参数的暂存寄存器
- 特性锁定机制：BSEC 提供 15 个硬件信号，通过 OTP 进行控制，禁止特定 SoC 功能以实现产品差异化
- 通过 BSEC_JTAGIN 和 BSEC_JTAGOUT 寄存器实现的 JTAG SOC 接口，作为 JTAG TAP 控制器的通信通道。
- 两个 OTP 区域被指定为低位 OTP (1 Kb) 和高位 OTP (2 Kb)，具有不同的属性
- **低位 OTP 位** (1 Kb)，支持 **2:1 冗余**，旨在存储：
  - 设备配置：BSEC OTP 模式和设备特性
  - 启动参数
  - 工程数据和模拟微调
  - 公共根密钥（256 位）

  所有基本安全功能，包括采用代码认证的安全启动，仅支持使用低位 OTP 区域。
- **高位 OTP 位** (2 Kb)，每个 32 位字 ECC 增加 6 位，用于存储安全敏感信息
  - ST 密钥
  - OEM 机密

  高位 OTP 应仅按 32 位字处理
- 改进 OTP 读取时通过时钟和电源噪声发起的硬件攻击的防护能力
- OTP 位始终通过 disturbcheck 读取。

  读取的每个 OTP 字都会与内部参考电平进行内部比较。disturbcheck 信息被提供给 BSEC，用于确认 OTP 字值。
- 通过黏着位实现全局编程锁定
- 每个字的永久 OTP 程序锁定
- 在启动过程中通过黏着位对 OTP 字进行编程锁定
- 在启动过程中，影子 OTP 寄存器可以通过黏着位单独写锁定
- 在启动过程中，影子 OTP 寄存器可以通过黏着位单独读锁定（以防止重载）
- Arm® TrustZone® 感知适用于：
  - BSEC 控制寄存器
  - 来自低位 OTP 区域的 BSEC 影子寄存器（字 0 到 31）
  - 来自高位 OTP 区域的 BSEC 影子寄存器（字 32 到 95）

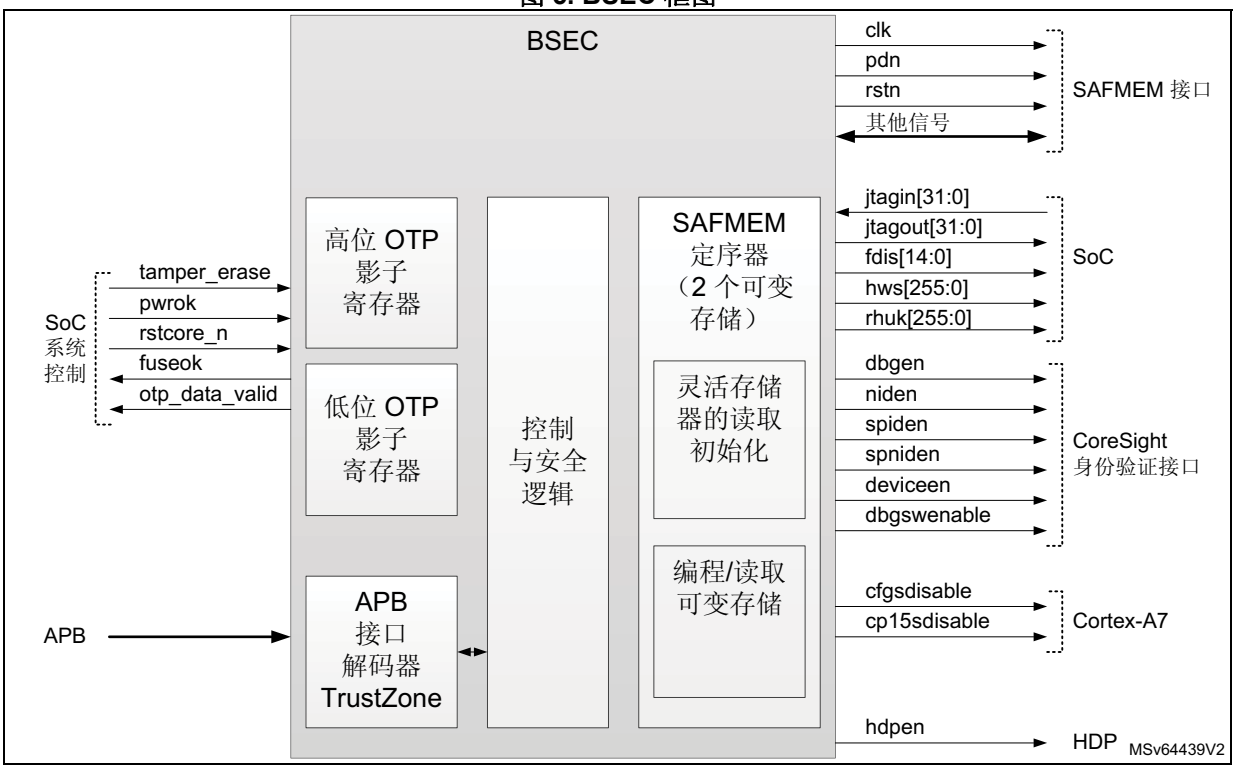
- 含六个信号的 Arm® CoreSight™ 调试认证接口：
  - **dbgen**（调试使能）
  - **devicen**（外部调试使能）
  - **niden**（非侵入式调试使能）
  - **spiden**（安全特权侵入式调试使能）
  - **spniden**（安全特权非侵入式调试使能）
  - **dbgswenable**（自托管调试使能）
- 多达 256 个硬件信号 **hws[255:0]**，用于模拟微调和工厂设置
- 非易失性随机生成的 256 位根硬件唯一密钥 (RHUK) 直接传送到 SAES 外设

3.3 BSEC 功能描述

BSEC 提供了一个 OTP 接口（也称为反熔丝宏），其中包含 3072 个有效位。BSEC 还提供了几个由启动过程使用的寄存器，以及防止访问敏感安全信息的机制。

3.3.1 BSEC 框图

图 5. BSEC 框图



3.3.2 OTP 接口

OTP 位按每组 32 位字分组，并分为两个区域：

- **低位区域**：32 个 32 位字（低位 2 Kb），使用 2:1 冗余。SoC 和启动配置参数、工程数据以及公共根密钥哈希存储在低位区域。
- **高位区域**：64 个 32 位字（高位 2 Kb），使用 ECC 方案（32 位编码为 38 位）。敏感的安全参数，如密钥，存储在高位区域。

注： *OTP 位初始化为 0，并且可以编程一次设置为 1。*
使用 2:1 冗余的低位 OTP 字，可以在多个操作中进行逐位编程
使用内部 ECC 的高位 OTP 字，必须一次性编程为批量 32 位数据。

OTP 位不能通过 CPU 直接访问。CPU 必须访问包含 OTP 有效位副本的影子寄存器。

提供 96 个**影子寄存器：BSEC_OTP_DATA0 到 BSEC_OTP_DATA95**。

这些 32 位寄存器是实际 OTP 位的副本。对用户来说，冗余和 ECC 是透明的。
BSEC_OTP_DATAx 在本文档中也被称为 OTP 字 #x。

可通过两种方式来更新影子寄存器：

- 系统复位时，BSEC 会自动将所有 OTP 有效位复制到影子寄存器。
- CPU 请求的 OTP 读操作，在 OTP 编程操作之后使用，以更新相应的影子寄存器。

以下 OTP 有效位对 BSEC 有特殊意义：

- **字 #0：位 0 到 6** 保留，用于确定 OTP 安全模式
- **字 #1：位 0 到 15** 用于控制功能禁用
- 字 #16 到字 #23 连接到 SoC 信号 hws[255:0]。
  这些信号可以：
  - 连接到 SoC 外设（例如：用于模拟外设工厂微调）。
  - 如果 OTP 位只由软件使用，则保持未连接（参见*第 4 章：OTP 映射 (OTP)*）。

3.3.3 OTP 安全模式

OTP 字 #0 位 0 到 9（及其潜在的干扰状态）用于决定 OTP 安全模式。

以下 OTP 安全模式可用：

- OTP-SECURED，包括子模式，根据 OTP 字#0 如*表 11* 所述：
  - open_device
  - closed_device
  - 禁止边界扫描的 closed device
  - 禁止 JTAG 的 closed device
- OTP-INVALID
  - 在这种状态下，高位 OTP 影子寄存器被清除且不可访问，低位 OTP 影子寄存器不可访问，BSEC 控制接口仅限安全读/写。

OTP 模式通过 BSEC_OTP_STATUS 寄存器对软件可见。

表 11. OTP 模式定义

BSEC_OTP_DATA0 [9:0]	OTP 模式	BSEC_OTP_STATUS				
		SECURE	INVALID	CLOSED	BSCANDIS	JTAGDIS
0000010111	OTP_SECURED open_device	1	0	0	0	0
0000111111	OTP_SECURED closed_device	1	0	1	0	0
0101111111	OTP_SECURED 禁止边界扫描的 closed device	1	0	1	1	0
1111111111	OTP_SECURED 禁止 JTAG 的 closed device	1	0	1	1	1
所有其他值	OTP-INVALID	x	1	x	x	x

注： *仅当 BSEC_OTP_DISTURBED[0] 等于零时，才会根据 BSEC_OTP_DATA0[9:0] 计算 OTP 模式。否则 OTP 模式被强制设置为 OTP-INVALID。*

3.3.4 系统复位时自动加载 OTP

在系统复位时，BSEC 会自动更新所有影子寄存器。在这个阶段确定 OTP 模式。

对于 OTP-**SECURED**模式，读取所有 OTP 字，并更新 BSEC_OTP_DATA**0** 到 BSEC_OTP_DATA**95**。

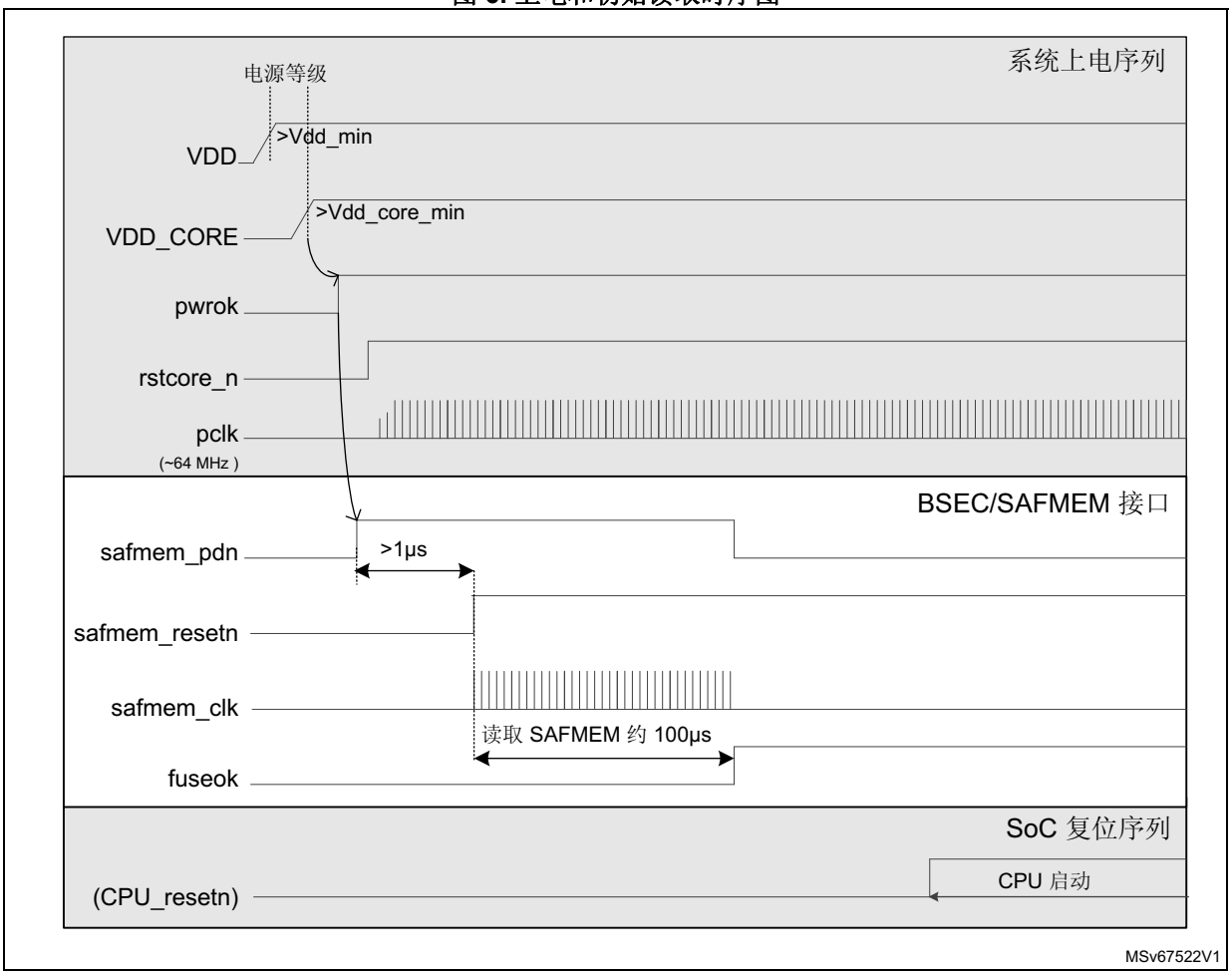
对于 OTP-**INVALID**模式，仅更新 BSEC_OTP_DATA**0** 到 BSEC_OTP_DATA**31**。BSEC_OTP_DATA**32** 到 BSEC_OTP_DATA**95** 被设置为 0。

BSEC_OTP_STATUS、BSEC_OTP_DISTURBEDx 和 BSEC_OTP_ERRORx 寄存器也会更新。

fuseok 信号在这个阶段末尾生效。此信号用于释放 SoC 的复位。

注： *OTP 字的读取从字 0 开始，以便 BSEC 从一开始就确定 OTP 模式。*

图 6. 上电和初始读取时序图



3.3.5 OTP 字状态

- **BSEC_OTP_DISTURBEDx** 寄存器 (x = 0...2) 提供了每个 OTP 字的**干扰状态**。任何被设置为 1 的位表示相应字的上一次读取受到干扰。已检测到电路解码和电压读取的异常读取条件。
- **BSEC_OTP_ERRORx** 寄存器 (x = 0...2) 提供每个 OTP 字的**错误状态**。任何被设置为 1 的位表示有关字的上一次读操作出现冗余或 ECC 检查错误。
- **BSEC_OTP_WRLOCKEDx** 寄存器 (x = 0...2) 提供了每个 OTP 字的**永久写入锁定**状态。任何被设置为 1 的位表示相应的字永久锁定，无法写入。

注： *所有状态寄存器在初始读取时会更新，但是干扰和错误状态寄存器也会在每次 OTP 读取操作后更新。*
前 32 个 OTP 字实施了一个方案，在读取时提高对硬件攻击的 OTP 鲁棒性，具体如下：备用位 32 到 37 旨在支持 ECC，不适用于 2:1 冗余。因此，它们采用一个简单的模式写入（例如 OTP 字地址占据 5 个位，第 6 位为偶校验）。每次 OTP 读取时都会检查此模式。如果读取的 32 到 37 位的值与预期值不匹配，则会向 BSEC_OTP_DISTURBED0 寄存器报告一个干扰信息。

3.3.6 OTP 操作

BSEC 时钟由 PCLK 时钟源提供（功能寄存器和接口寄存器使用一个时钟）。OTP 字的读取、写入（编程）或锁定称为 OTP 操作。

通过 BSEC_OTP_CONTROL 寄存器执行一次 OTP 操作。当软件向 BSEC_OTP_CONTROL 寄存器写入有效请求且无其它操作正在进行时（一个 OTP 操作必须在上一个操作完成后才能开始）就会启动该操作。

具体的读取、编程或永久写入锁定操作详见下文。

OTP 读取

读取操作包括更新 BSEC_OTP_DATA0 到 BSEC_OTP_DATA95 寄存器中的一个与 OTP 字对应的字。

要触发读取操作，软件必须通过 ADDR 字段中给出的字编号并且在 PROG 位设置为 0 的条件下设置 BSEC_OTP_CONTROL 寄存器。

软件可以检查 BSEC_OTP_STATUS 寄存器中的 BUSY 位：一旦清除，这个 BUSY 位表示读取操作已完成。

当读取操作完成时，BSEC 状态机更新“干扰”和“错误”状态寄存器。

在读取相应的 OTP 字时，与 OTP 内容相关的 BSEC 参数也会更新。当对字 0 执行读取操作时，OTP 模式就会更新。

注： *在 OTP-INVALID 模式下，或者如果 BSEC_OTP_LOCK 寄存器中的 OTP 位置 1，就无法读取 32 到 95 的字。相应的影子寄存器不会被修改。*

OTP 编程

编程操作包括将一个 OTP 字的一个或多个位编程为一个位。

OTP 字位可以以递增方式编程为一，但已经编程为一的位不能被写入为零。

低位 OTP 字支持逐位可编程，可以以递增方式编程为一，由于 ECC 编码，高位 OTP 字必须在单个操作中编程。

触发编程操作需要以下两个步骤：

- 在 BSEC_OTP_WRDATA 寄存器中设置要写入的字值
- 设置 BSEC_OTP_CONTROL 寄存器：
  - ADDR 字段中给出的字编号
  - 将 PROG 位设置为 1
  - 将 LOCK 位设置为 0。

然后软件检查 BSEC_OTP_STATUS 寄存器中的 BUSY 位：一旦清除，这个 BUSY 位指示写入操作已完成。

在同一寄存器中，如果写入操作失败，则 progfail 位置 1。

注： *OTP 字的编程操作不会更新相应的影子寄存器。如有需要，软件可以通过读取操作来更新影子寄存器。*
在 OTP-INVALID 模式下，高位 OTP（32 到 95 字）无法编程，不受 BSEC_OTP_LOCK 寄存器中的 GPLOCK 位，以及 BSEC_OTP_SPLOCK2/3 寄存器的影响。

OTP 永久写入锁定

永久写入锁定操作包括强制一个 OTP 字永久写入锁定且无法编程。

要实现 OTP 字的永久写入锁定，软件必须执行与正常编程操作相同的序列：
在 BSEC_OTP_CONTROL 寄存器中设置 LOCK 位，设置 BSEC_OTP_CONTROL 寄存器中的 ADDR 字段和 BSEC_OTP_WRDATA 寄存器（参见*表 12*）。永久锁定编程后，需要用 OTP 内容更新 BSEC_OTP_WRLOCKEDx 寄存器。为此，软件必须在 BSEC_OTP_CONTROL 寄存器中设置 LOCK 位并提供字地址的条件下执行字读取操作。

表 12. OTP 永久写入锁定

要锁定的 OTP 字	BSEC 控制寄存器值		备注
	ADDR[6:0]	WRDATA[31:0]	
0	0x00	0x0000 0003	对于低位 OTP，使用每字 2 位
1	0x00	0x0000 000C	
...	...	...	
7	0x00	0x0000 C000	
8	0x01	0x0000 0003	
...	...	...	
15	0x01	0x0000 C000	
16	0x02	0x0000 0003	
...	...	...	
31	0x03	0x0000 C000	
32	0x04	0x0000 0001	对于高位 OTP，使用每字 1 位
33	0x04	0x0000 0002	
...	...	...	
47	0x04	0x0000 8000	
48	0x05	0x0000 0001	
...	...	...	
63	0x05	0x0000 8000	
...	...	...	
95	0x07	0x0000 8000	

3.3.7 JTAG 寄存器

BSEC_JTAGIN 和 BSEC_JTAGOUT 寄存器是两个 32 位寄存器，用于软件与 JTAG SoC 接口之间的通信。该接口在启动期间用作外部 JTAG TAP 控制器的通信通道。

3.3.8 BSEC 锁定寄存器

BSEC_OTP_LOCK 寄存器用于使用黏着位锁定对安全敏感区域的访问。
这些锁定将持续有效，直到下一次系统复位（参见*第 3.6.5 节：BSEC OTP 锁定配置寄存器 (BSEC_OTP_LOCK)*）。

3.3.9 调试控制

软件通过 BSEC_DENABLE 寄存器控制 Debug 的范围。可以通过 BSEC_OTP_LOCK 寄存器中的 DENREG 位，将对该寄存器的访问锁定，直到下一次系统复位。

BSEC_DENABLE 正在向 SoC 驱动硬件信号（例如 CoreSight 身份验证接口和*表 13* 中列出的其他特定控制信号）。

有关更多信息，请参见*第 59 章：调试支持 (DBG)*。

BSEC_DENABLE 的默认值取决于 BSEC OTP 模式，如*表 13* 所示。

表 13. 复位后的 BSEC_DENABLE 默认值

信号	OTP-SECURED open_device	OTP-SECURED closed_device	OTP-INVALID	备注
dbgen	1	0	0	CoreSight 身份验证
niden	1	0	0	CoreSight 身份验证
spiden	0	0	0	CoreSight 身份验证
spniden	0	0	0	CoreSight 身份验证
deviceen	1	0	0	CoreSight 身份验证
dbgswenable	0	0	0	CoreSight 身份验证
hdpen	0	0	0	硬件调试端口跟踪
cfgsdisable	0	0	0	禁止部分 Cortex®-A7 GIC 安全访问
cp15sdisable	0b0	0b0	0b0	禁止部分 Cortex®-A7 CP15 安全访问

3.3.10 粘性锁定寄存器

三个粘性锁寄存器用于控制影子寄存器的重新加载，按 OTP 字粒度：

- BSEC_SRLOCKx (x = 0..2) 用于防止重载所选影子寄存器，直到下一次系统复位。
- BSEC_SWLOCKx (x = 0..2) 用于锁定来自 APB 对所选影子寄存器的写操作，直到下一次系统复位。
- BSEC_SPLOCKx (x = 0..2) 用于防止所选 OTP 字编程，直到下一次系统复位。

3.4 OTP 布局

低位 OTP 字 (1 Kb)

低位 OTP 位存储在影子寄存器 BSEC_OTP_DATA0 到 BSEC_OTP_DATA31 中，用于常规用途和非机密数据。

这些 OTP 通过连续的偶数位/奇数位实现 2:1 冗余保护。

影响硬件的特定位，如 OTP 模式、工程和微调位于：CFG0、CFG1 和其他 8 个字（HW0 到 HW7）。

高位 OTP 字 (2 Kb)

高位 OTP 字存储在影子寄存器 BSEC_OTP_DATA32 到 BSEC_OTP_DATA95 中，专门用于存储安全敏感信息（密钥、证书、RMA 密钥）。这些字也可以存储公共的非易失性信息。

在启动过程中可以使用这些高位 OTP 位实现高级安全服务，如安全机密供应 (SSP)、加密启动或安全存储。

高位 OTP 字受 ECC 保护，必须与 32 位字粒度结合使用。

3.4.1 DMA 请求

BSEC 不支持 DMA 硬件接口。

3.4.2 TrustZone 感知

BSEC 支持 TrustZone 感知。所有 APB 寄存器的读写访问都要进行权限检查。*表 14* 和*表 15* 描述了根据 OTP 模式和寄存器类型划分的权限规则。

如果权限检查失败：

- 写入访问被忽略，并在 APB 总线上发出 SLVERR 信号。
- 读取访问返回零值，并在 APB 总线上发出 SLVERR 信号。

表 14. 写入访问权限

OTP 模式	控制寄存器(1)		低位 OTP 影子寄存器		高位 OTP 影子寄存器	
	APB 安全	APB 非安全	APB 安全	APB 非安全	APB 安全	APB 非安全
OTP-SECURED	是	否	是	否	是	否
OTP-INVALID	否	否	否	否	否	否

1. BSEC 控制寄存器是除 BSEC_OTP_DATA0 到 BSEC_OTP_DATA95 之外的任何 BSEC 寄存器。

表 15. 读取访问权限

OTP 模式	控制寄存器(1)		低位 OTP 影子寄存器		高位 OTP 影子寄存器	
	APB 安全	APB 非安全	APB 安全	APB 非安全	APB 安全	APB 非安全
OTP-SECURED	是	是	是	是	是	否
OTP-INVALID	是	否	否	否	否	否

1. BSEC 控制寄存器是除 BSEC_OTP_DATA0 到 BSEC_OTP_DATA95 之外的任何 BSEC 寄存器。

3.4.3 BSEC Tamper 保护

SOC Tamper 检测通过信号 tamper_erase 通知给 BSEC。

当 tamper_erase 生效时，无论 Tamper 事件何时发生，BSEC_OTP_DATA32 到 BSEC_OTP_DATA95 都被强制设置为“零”，BSEC 被强制设置为 OTP-INVALID 模式。转换到 OTP-INVALID 之后，BSEC_OTP_STATUS 和 BSEC_DENABLE 寄存器也根据*第 166 页的表 13* 进行复位。

复位之后，如果所有确认的 Tamper 事件已被清除，并且初始 OTP 读取完成，未受干扰且无错误，BSEC OTP 模式将恢复到 OTP-SECURED 模式。

3.4.4 BSEC 时钟和初始化

BSEC 使用单一时钟域 (PCLK)。

- 在读取和编程 OTP 时，PCLK 频率必须 ≤ 67 MHz。
- 在写入影子寄存器时，PCLK 频率必须 ≤ 67 MHz。
- 在所有其他操作中，PCLK 频率没有限制，但建议所有操作都保持 PCLK ≤ 67 MHz（以避免频率变化）。
- BSEC 正在执行写信号的正确序列：上电和掉电序列期间的 CLK、PDN 和 RESETN。

3.5 BSEC 中断

BSEC **不支持**任何中断。

3.6 BSEC 寄存器

外设寄存器按字（32 位）进行访问。

对只读寄存器的任何写入访问都会被静默忽略。

任何尝试修改锁定寄存器（通过粘性锁定或锁定位锁定）的行为都会被静默忽略。

3.6.1 BSEC OTP 配置寄存器 (BSEC_OTP_CONFIG)

BSEC OTP configuration register

偏移地址：0x000

复位值：0x0000 000E

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TREAD[1:0]		PRGWIDTH[3:0]				FRC[1:0]		PWRUP
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:9 保留，必须保持复位值。

位 8:7 **TREAD[1:0]**：设置 OTP 读取电流水平（默认值 = 0b00）

位 6:3 **PRGWIDTH[3:0]**：OTP 编程脉冲宽度（默认值 = 0b0001）

位 2:1 **FRC[1:0]**：OTP 时钟频率范围选择

00: 10 MHz ≤ 频率 ≤ 20 MHz
01: 20 MHz ≤ 频率 ≤ 30 MHz
10: 30 MHz ≤ 频率 ≤ 45 MHz
11: 45 MHz ≤ 频率 ≤ 67 MHz

*注： 默认使用 0b11。*OTP *初始读取始终使用最大频率为 64 MHz 的内部 OSC。可以以比 FRC[1:0] 编程范围更低的频率读取* OTP。

位 0 **PWRUP**：OTP 上电控制
0: OTP 掉电
1: OTP 上电
注：　在上电初次读取 OTP 之后，BSEC 会让 OTP 掉电。随后 PWRUP 位被清除。
在进行任何读取或编程操作之前，必须通过设置 PWRUP 位再次对 OTP 上电。
*OTP 读取由 **pwrok** 输入信号校验，该信号表明 VDD 和 VDD1 电源处于有效范围内。*

3.6.2 BSEC OTP 控制寄存器 (BSEC_OTP_CONTROL)

BSEC OTP control register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	LOCK	PROG	Res.	ADDR[6:0]						
						rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:10 保留，必须保持复位值。

位 9 **LOCK**：OTP 永久字锁定控制
0: OTP 常规字编程/读取
1: OTP 永久写入锁定字编程/读取

位 8 **可编程**：OTP 操作控制
0: OTP 读取操作
1: OTP 编程操作

位 7 保留，必须保持复位值。

位 6:0 **ADDR[6:0]**：OTP 字地址
实际的 OTP 字地址。
BSEC 的字地址范围是 0 到 95。
注：　实际的 OTP 字地址根据冗余或 ECC 方案进行调整。

3.6.3 BSEC OTP 写入数据寄存器 (BSEC_OTP_WRDATA)

BSEC OTP write data register

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRDATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRDATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **WRDATA[31:0]**：OTP 写入数据

3.6.4 BSEC OTP 状态寄存器 (BSEC_OTP_STATUS)

BSEC OTP status register

偏移地址：0x00C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	JTAG DIS	BSCAN DIS	CLO SED	Res.	Res.	PWR 开启	PROG FAIL	BUSY	IN VALID	Res.	SEC URE
					r	r	r			r	r	r	r		r

位 31:11 保留，必须保持复位值。

位 10 **JTAGDIS**：JTAG 端口状态

0: JTAG 端口未禁止

1: JTAG 端口除了访问 REV_ID 信息（参见“设备电子签名”部分）和 BSEC_JTAGIN 寄存器外，均已禁止。

位 9 **BSCANDIS**：边界扫描状态

0: 边界扫描未禁止

1: 边界扫描被禁止

位 8 **CLOSED**：OTP_SECURED 模式

0: OTP-SECURED 开放设备

1: OTP-SECURED 封闭设备

位 7:6 保留，必须保持复位值。

位 5 **PWRON**：OTP 电源状态

0: OTP 处于掉电状态

1: OTP 处于上电状态

注： 用于轮询 pwrok 信号值

位 4 **PROGFAIL**：上一次编程状态
0: OTP 上一次编程成功
1: OTP 上一次编程失败

位 3 **BUSY**：OTP 操作状态
0: OTP 空闲
1: 正在进行 OTP 操作
注： 使用位轮询来判断操作完成度。

位 2 **INVALID**：OTP 无效模式
0: OTP 模式不是 OTP-INVALID
1: OTP 模式是 OTP-INVALID

位 1 保留，必须保持复位值。

位 0 **SECURE**：OTP 安全模式
0: OTP 模式不是 OTP-SECURED
1: OTP 模式是 OTP-SECURED

注： SECURE 和 INVALID 位的复位值取决于 OTP 模式。

注： 最低位 SECURE 和 INVALID 必须用来识别 OTP 模式，而不是读取影子寄存器 BSEC_OTP_DATA0。优先级为 INVALID > SECURE。

3.6.5 BSEC OTP 锁定配置寄存器 (BSEC_OTP_LOCK)

BSEC OTP lock configuration register

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GP LOCK	Res.	DEN REG	ROM LOCK	OTP
											rs		rs	rs	rs

位 31:5 保留，必须保持复位值。

位 4 **GPLOCK**：编程粘性锁定
0: 允许编程
1: 编程功能被禁止，直到下一次系统复位

位 3 保留，必须保持复位值。

位 2 **DENREG**：调试使能寄存器粘性锁定
0: BSEC_DENABLE 寄存器未锁定
1: BSEC_DENABLE 寄存器锁定，直到下一次系统复位

位 1 **ROMLOCK**：高位 ROM 区域读取锁定
0: 高位 ROM 区域可访问
1: 高位 ROM 区域不可访问，直到下一次系统复位
高位 ROM 区域只有在 OTP-SECURED 开放模式和 OTP-SECURED 封闭模式下可访问，ROMLOCK 位仅在这些模式下相关。

位 0 **OTP**：高位 OTP 读取锁定
0: 高位 OTP 未锁定
1: 高位 OTP 无法重载，直到下一次系统复位

3.6.6 BSEC 调试配置寄存器 (BSEC_DENABLE)

BSEC debug configuration register

偏移地址：0x014

复位值：0x0000 0000

注： *依据第 166 页的表 13：复位后的 BSEC_DENABLE 默认值，复位值取决于 OTP 安全模式。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	DBG SWEN ABLE	CFGS DIS ABLE	Res.	CP15S DIS ABLE	SPNID EN	SPID EN	HDPEN	DE VICE EN	NIDEN	DBG EN	Res.
					rw	rw		rw	rw	rw	rw	rw	rw	rw	

位 31:11 保留，必须保持复位值。

位 10 **DBGSWENABLE**：通过信号 **dbgswenable** 控制自托管调试使能
0: 禁止软件访问所有调试组件
1: 启用软件访问所有调试组件

位 9 **CFGSDISABLE**：通过信号 **cfgsdisable** 禁止对安全 GIC 寄存器的写入访问
0: 无影响，所有 GIC 寄存器均可访问
1: 禁止对部分安全 GIC 寄存器的写入访问

位 8 保留，必须保持复位值。

位 7 **CP15SDISABLE**：禁止对部分安全 Cortex®-A7 CP15 寄存器的写入访问
0: 所有 CP15 寄存器均可访问
1: 禁止对 Cortex®-A7 对应 CPU 中部分安全 CP15 寄存器的写入访问

位 6 **SPNIDEN**：通过信号 **spiden** 使能安全特权非侵入式调试
0: 禁止安全特权非侵入式调试
1: 使能安全特权非侵入式调试

位 5 **SPIDEN**：通过信号 **spniden** 使能安全特权侵入式调试
0: 禁止安全特权侵入式调试
1: 使能安全特权侵入式调试

位 4 **HDPEN**：通过信号 **hdpen** 使能硬件调试端口
0: 禁止硬件调试端口
1: 使能硬件调试端口

位 3 **DEVICEEN**：通过信号 **deviceen** 限制通过外部调试端口访问调试组件
0: 禁止
1: 使能

位 2 **NIDEN**：通过信号 **niden** 使能非侵入调试
0: 禁止非侵入式调试
1: 使能非侵入式调试

位 1 **DBGEN**：通过信号 **dbgen** 使能调试
0: 禁止
1: 使能

位 0 保留，必须保持复位值。

3.6.7 BSEC OTP 干扰状态寄存器 x (BSEC_OTP_DISTURBEDx)

BSEC OTP disturbed status register x

BSEC_OTP_DISTURBED**0** 用于报告 BSEC_OTP_DATA0 到 BSEC_OTP_DATA31（低位 1 Kb OTP）的干扰状态。

BSEC_OTP_DISTURBED**1** 用于报告 BSEC_OTP_DATA32 到 BSEC_OTP_DATA63 的干扰状态。

BSEC_OTP_DISTURBED**2** 用于报告 BSEC_OTP_DATA64 到 BSEC_OTP_DATA95 的干扰状态。

偏移地址：0x01C + 0x004 * x（x = 0 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIS[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **DIS[31:0]**：相应 OTP 字的干扰状态
0: OTP 字已正确读取
1: OTP 字干扰，可能是由于带有 disturbcheck 的读取失败或者是因为第 0 到 31 个字的备用 ECC 位 [37:32] 与预期值不匹配。这可能是由潜在的硬件攻击引起的。

注： ECC 位对硬件攻击的抵抗力仅应用于低位 OTP 区域。

3.6.8 BSEC OTP 错误状态寄存器 x (BSEC_OTP_ERRORx)

BSEC OTP error status register x

BSEC_OTP_ERROR**0** 用于报告 BSEC_OTP_DATA0 到 BSEC_OTP_DATA31（受 2:1 冗余保护的低位 1 Kb OTP）的错误状态。

BSEC_OTP_ERROR**1** 用于报告 BSEC_OTP_DATA32 到 BSEC_OTP_DATA63 的错误状态，由 6 位 ECC 保护。

BSEC_OTP_ERROR**2** 用于报告 BSEC_OTP_DATA64 到 BSEC_OTP_DATA95 的错误状态，由 6 位 ECC 保护。

偏移地址：0x034 + 0x004 * x（x = 0 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ERR[31:0]**：对应 OTP 字的错误状态

0: OTP 字已正确读取

1: OTP 字出错，可能是因为第 0 到 31 个字的 2:1 冗余失败或第 32 到 95 个字的 ECC 检查失败。

3.6.9 BSEC_OTP 锁定状态寄存器 x (BSEC_OTP_WRLOCKEDx)

BSEC OTP lock status register x

BSEC_OTP_WRLOCKED**0** 用于报告 BSEC_OTP_DATA0 到 BSEC_OTP_DATA31 的永久写入锁定。

BSEC_OTP_WRLOCKED**1** 用于报告 BSEC_OTP_DATA32 到 BSEC_OTP_DATA63 的永久写入锁定。

BSEC_OTP_WRLOCKED**2** 用于报告 BSEC_OTP_DATA64 到 BSEC_OTP_DATA95 的永久写入锁定。

永久写入锁定需要一个编程序列来锁定一个字（参见节：*第 164 页的 第 3.3.6 节：OTP 操作*）。

偏移地址：0x04C + 0x004 * x（x = 0 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRLOCK[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRLOCK[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **WRLOCK[31:0]**：对应 OTP 字的永久字锁定状态
0: OTP 字未锁定
1: OTP 字已永久锁定

3.6.10 BSEC OTP 粘性编程锁定寄存器 x (BSEC_OTP_SPLOCKx)

BSEC OTP sticky programming lock register x

BSEC_OTP_SPLOCK**0** 用于锁定 BSEC_OTP_DATA0 到 BSEC_OTP_DATA31 的编程，直到下一次系统复位

BSEC_OTP_SPLOCK**1** 用于锁定 BSEC_OTP_DATA32 到 BSEC_OTP_DATA63 的编程，直到下一次系统复位

BSEC_OTP_SPLOCK**2** 用于锁定 BSEC_OTP_DATA64 到 BSEC_OTP_DATA95 的编程，直到下一次系统复位

尝试对已粘性编程锁定的 OTP 字的操作将被静默忽略。

偏移地址：0x064 + 0x004 * x（x = 0 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPLOCK[31:16]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPLOCK[15:0]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:0 **SPLOCK[31:0]**：锁定 OTP 字的编程，直到下一次上电复位
0: OTP 字未锁定
1: OTP 字被锁定，无法编程

3.6.11 BSEC_OTP 影子写入粘性锁定寄存器 x (BSEC_OTP_SWLOCKx)

BSEC OTP shadow write sticky lock register x

BSEC_OTP_SWLOCK**0** 用于防止对 BSEC_OTP_DATA0 到 BSEC_OTP_DATA31 进行写入，直到下一次系统复位。

BSEC_OTP_SWLOCK**1** 用于防止对 BSEC_OTP_DATA32 到 BSEC_OTP_DATA63 进行写入，直到下一次系统复位。

BSEC_OTP_SWLOCK**2** 用于防止对 BSEC_OTP_DATA64 到 BSEC_OTP_DATA95 进行写入，直到下一次系统复位。

对影子写入锁定的 BSEC_OTP_DATA 字的写入操作将被静默忽略。

注： *始终阻止 OTP 字 0 影子的写入。*

偏移地址：0x07C + 0x004 * x（x = 0 到 2）

复位值：0x0000 0001、0x0000 0000、0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWLOCK[31:16]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWLOCK[15:0]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:0 **SWLOCK[31:0]**：锁定 OTP 影子字写入，直到下一次上电复位。

0：影子字未锁定

1：影子字已锁定，不能由软件写入

注： *BSEC_OTP_SWLOCK0 位 0 被硬件强制设置为 1，写入此位无效。*

3.6.12 BSEC OTP 影子读取粘性锁定寄存器 x (BSEC_OTP_SRLOCKx)

BSEC OTP shadow read sticky lock register x

BSEC_OTP_SRLOCK**0** 用于防止重载 BSEC_OTP_DATA0 到 BSEC_OTP_DATA31，直到下一次系统复位。

BSEC_OTP_SRLOCK**1** 用于防止重载 BSEC_OTP_DATA32 到 BSEC_OTP_DATA63，直到下一次系统复位。

BSEC_OTP_SRLOCK**2** 用于防止重载 BSEC_OTP_DATA64 到 BSEC_OTP_DATA95，直到下一次系统复位。

设置 SRLOCK 位或尝试重载已锁定的 OTP **不会**清除相应的 BSEC_OTP_DATAx 影子寄存器。

注： *BSEC_OTP_SRLOCK0 位 0 根据 fuse_ok 由硬件控制，写入此位无效。*

偏移地址：0x094 + 0x004 * x（x = 0 到 2）

复位值：0x0000 000X，0x0000 0000，0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRLOCK[31:16]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRLOCK[15:0]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:0 **SRLOCK[31:0]**：防止重载来自 OTP 的影子字，直到下一次上电复位。

0: OTP 字未锁定

1: OTP 字已锁定，不能被重载

注： 系统复位时，首次读取后，防止重载 OTP 字 0。

3.6.13 BSEC JTAG 输入寄存器 (BSEC_JTAGIN)

BSEC JTAG input register

偏移地址：0x0AC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **DATA[31:0]**：JTAG 输入数据

从 JTAG SoC 寄存器复制的 32 位

3.6.14 BSEC JTAG 输出寄存器 (BSEC_JTAGOUT)

BSEC JTAG output register

偏移地址：0x0B0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DATA[31:0]**：JTAG 输出数据
复制到 JTAG SoC 寄存器的 32 位

3.6.15 BSEC 暂存寄存器 (BSEC_SCRATCH)

BSEC scratch register

偏移地址：0x0B4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DATA[31:0]**：暂存数据
该寄存器是通用寄存器。

3.6.16 BSEC 影子寄存器 x (BSEC_OTP_DATAx)

BSEC shadow register x

有关 BSEC_OTP_DATAx 寄存器值和属性的完整描述，请参见*第 4 章：OTP 映射 (OTP)*。

部分 OTP 直接影响 BSEC 行为，例如：

- BSEC_OTP_DATA0[9:0]（参见*第 162 页的表 11：OTP 模式定义*）

偏移地址：0x200 + 0x004 * x（x = 0 到 95）

复位值：0xXXXX XXXX

复位值取决于实际编程的 OTP 值和 OTP 模式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DATA[31:0]**：从 OTP 读取或由软件写入的影子寄存器（OTP 模拟模式）
影子寄存器的更新由其他寄存器和 OTP 安全模式控制。

3.6.17 BSEC 硬件配置寄存器 (BSEC_HWCFGR)

BSEC hardware configuration register

偏移地址：0xFF0

复位值：0x0000 0014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECC_USE[3:0]				SIZE[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **ECC_USE[3:0]**：使用的保护/冗余方案
0x0：否
0x1：用于高位 OTP 位的 ECC
其他值：保留

位 3:0 **SIZE[3:0]**：OTP 块大小
0x2：2 Kb
0x4：4 Kb
0x8：8 Kb
其他值：保留

3.6.18 BSEC 版本寄存器 (BSEC_VERR)

BSEC version register

偏移地址：0xFF4

复位值：0x0000 0020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本信息

位 3:0 **MINREV[3:0]**：次版本信息

3.6.19 BSEC 标识寄存器 (BSEC_IPIDR)

BSEC identification register

偏移地址：0xFF8

复位值：0x0010 0032

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：BSEC 标识

3.6.20 BSEC 大小标识寄存器 (BSEC_SIDR)

BSEC size identification register

偏移地址：0xFFC

复位值：0xA3C5 DD04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：BSEC 大小标识

3.6.21 BSEC 寄存器映射

表 16 描述 BSEC 寄存器映射和复位值。

表 16. BSEC 寄存器映射和复位值

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**BSEC_OTP_CONFIG**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TREAD		PRGWDTH				FRC		PWRUP
	Reset value																								0	0	0	0	0	1	1	1	0
0x004	**BSEC_OTP_CONTROL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCK	PROG	Res.	ADDR[6:0]						
	Reset value																							0	0		0	0	0	0	0	0	0
0x008	**BSEC_OTP_WRDATA**	WRDATA[31:0]																															
	Reset value																																
0x00C	**BSEC_OTP_STATUS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JTAGDIS	BSCANDIS	CLOSED	Res.	Res.	PWRON	PROGFAIL	BUSY	INVALID	Res.	SECURE
	Reset value																						0	0	0			0	0	0	0		0
0x010	**BSEC_OTP_LOCK**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPLOCK	Res.	DENREG	ROMLOCK	OTP
	Reset value																												0		0	0	0
0x014	**BSEC_OTP_DENABLE**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFGSWENABLE	CFGSDISABLE	Res.	CP15SDISABLE	SPNIDEN	SPIDEN	HDPEN	DEVICEEN	NIDEN	\DBGEN	Res.
	Reset value																						0	0		0	0	0	0	0	0	0	
0x01C	**BSEC_OTP_DISTRUBED0**	DIS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	**BSEC_OTP_DISTURBED1**	DIS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	**BSEC_OTP_DISTURBED2**	DIS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x034	**BSEC_OTP_ERROR0**	ERR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 16. BSEC 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x038	**BSEC_OTP_ERROR1**	ERR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03C	**BSEC_OTP_ERROR2**	ERR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04C	**BSEC_OTP_WRLOCKED0**	WRLOCK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x050	**BSEC_OTP_WRLOCKED1**	WRLOCK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x054	**BSEC_OTP_WRLOCKED2**	WRLOCK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x064	**BSEC_OTP_SPLOCK0**	SPLOCK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x068	**BSEC_OTP_SPLOCK1**	SPLOCK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x06C	**BSEC_OTP_SPLOCK2**	SPLOCK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07C	**BSEC_OTP_SWLOCK0**	SWLOCK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x080	**BSEC_OTP_SWLOCK1**	SWLOCK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	**BSEC_OTP_SWLOCK2**	SWLOCK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x094	**BSEC_OTP_SRLOCK0**	SRLOCK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X
0x098	**BSEC_OTP_SRLOCK1**	SRLOCK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 16. BSEC 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x09C	**BSEC_OTP_SRLOCK2**	SRLOCK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0AC	**BSEC_JTAGIN**	DATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B0	**BSEC_JTAGOUT**	DATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B4	**BSEC_SCRATCH**	DATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x200	**BSEC_OTP_DATA0**	DATA[31:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x204	**BSEC_OTP_DATA1**	DATA[31:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
. . .	. . .	. . .																															
0x37C	**BSEC_OTP_DATA95**	DATA[31:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0xFF0	**BSEC_IPHW_CFGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECC_USE [3:0]				SIZE[3:0]			
	Reset value																									0	0	0	1	0	1	0	0
0xFF4	**BSEC_VERR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV [3:0]				MINREV [3:0]			
	Reset value																									0	0	1	0	0	0	0	0
0xFF8	**BSEC_IPIDR**	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0
0xFFC	**BSEC_SIDR**	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	1	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

4 OTP 映射 (OTP)

OTP 位由启动、安全和 OTP 控制 (BSEC) 读取或编程。在编程之前，OTP 位的默认值是“0”。

STM32MP13xx 设备有 3072 个 OTP（一次性编程）位，可以在 96 个字中读取访问：BSEC_OTP_DATAx（x = 0 到 95）：

- 低位 OTP 参见*表 17：OTP 列表字 0 到 31*。
- 高位 OTP 参见*表 18：OTP 列表字 32 到 95*。

表格列出了 OTP 字和以下位字段相关属性：

名称：OTP 字同义词（例如 OTP word0 又称为 CFG0）。

说明：OTP 字或位字段描述。

启动：系统复位和 Boot ROM 执行之后。

Prog：编程选项：

s：由意法半导体在工厂编程，无论是产品差异化还是密钥。

u：可供 Boot ROM 后执行的任何其他软件使用。

wplock：

y：通过工厂编程进行永久编程锁定。

swprlockx：启动时设置的粘性锁定，表示为 s[w][p][r][c]，其中字符 w、p、r、c 表示：

w：表示影子写入已粘性锁定，
p：表示 OTP 编程已粘性锁定，
r：表示 OTP 重载已粘性锁定，
c：表示启动时 OTP 影子寄存器被清除。

swprlock1：在 secured-open 模式下启动后设置粘性锁定。

swprlock2：在 secured-closed 模式下启动后设置粘性锁定（仅可用于具有安全启动能力的设备）。

swprlock3：启动后设置 RMA 的粘性锁定。

表 17. OTP 列表字 0 到 31

字	名称	位	说明	prog	wplock	swprlock1	swprlock2	swprlock3	详细说明
0	CFG0	**OTP 模式编码**							
		9:0	OTP 模式编码(1)	s	n	sw	sw	sw	-
		31:10	意法半导体预留						
1	CFG1	**RPN 编码**							
		11:0	RPN 编码	s	y	sw	sw	sw	*第 60.2 节：设备产品编号 (RPN)*
		31:12	意法半导体预留						
2	CFG2	**意法半导体预留**							
		31:0	意法半导体预留	s	y	swp	swp	swp	-
3	CFG3	**启动/来源定义**							
		0	qspi_not_default_af	u	n	-	sw	-	0: QSPI 使用默认硬编码的 AFmux 1: QSPI 非默认 AFmux（使用在 OTP 字 5、6 和 7 中定义的非默认 AFmux）
		2:1	emmc_if_id						0: 来源是默认的：具有默认 AFMux 的 SDMMC2 1: SDMMC1 2: SDMMC2（使用在 OTP 字 5、6 和 7 中定义的非默认 AFmux）
		4:3	sd_if_id						0: 来源是默认的：具有默认 AFMux 的 SDMMC1 1: SDMMC1（使用在 OTP 字 5、6 和 7 中定义的非默认 AFmux） 2: SDMMC2
		5	no_cpu_pll						0: CPU/AXI 的 PLL 在冷启动时启用 1: CPU/AXI 的 PLL 在冷启动时未启用
		6	no_usb_dp_pullup						0: 已设置 USB DP 上拉。1: 未设置 USB DP 上拉
		14:7	uart_instances_disable						0b00000001：保留 0b00000010：保留 0b00000100：禁止 USART3 0b00001000：禁止 UART4 0b00010000：禁止 UART5 0b00100000：禁止 UART6 0b01000000：禁止 UART7 0b10000000：禁止 USART8

表 17. OTP 列表字 0 到 31（续）

字	名称	位	说明	prog	wplock	swprlock1	swprlock2	swprlock3	详细说明
3	CFG3	15	no_data_cache	u	n	-	sw	-	0: 数据缓存被 Boot ROM 使用 1: 数据缓存未被 Boot ROM 使用
		22:16	boot_source_ disable						0b0000001：禁止 FMC 启动源 0b0000010：禁止 QSPI NOR 启动源 0b0000100：禁止 eMMC 启动源 0b0001000：禁止 SD 启动源 0b0010000：禁止 UART 启动源 0b0100000：禁止 USB 启动源 0b1000000：禁止 QSPI NAND 启动源
		23	意法半导体预留						-
		26:24	secondary_boot_ source						0: 未定义次要启动源 1: FMC NAND 2: QSPI NOR 3: eMMC 4: SD 5: QSPI NAND
		29:27	primary_boot_ source						0: 未定义主要启动源 1: FMC NAND 2: QSPI NOR 3: *e*•MMC 4: SD 5: QSPI NAND
		31:30	HSE 值						自动检测 0b00 HSE 0b01 HSE 为 24 MHz 0b10 HSE 为 25 MHz 0b11 HSE 为 26 MHz
4	CFG4	启动/单调计数器							
		31:0	启动/单调计数器	u	n	-	-	-	单调计数器的值为 X，其中 X 为 1 的最高有效位的位置。

表 17. OTP 列表字 0 到 31（续）

字	名称	位	说明	prog	wplock	swprlock1	swprlock2	swprlock3	详细说明
5	CFG5	启动/AFmux 配置 1							
		3:0	mode0	u	n	-	sw	-	0: AF；无上下拉；低速 1: AF；无上下拉；中速 2: AF；无上下拉；高速 3: AF；上拉；低速 4: AF；上拉；中速 5: AF；上拉；高速 6: AF；下拉；低速 7: AF；下拉；中速 8: AF；下拉；高速 9: GPIO 输出高电平 10: GPIO 输出低电平 11: GPIO 输入 12: GPIO 开漏；无上下拉 13: GPIO 开漏；上拉 14: GPIO 开漏；下拉 15: GPIO 模拟模式
		7:4	afmux0						AF 复用值在 0 到 15 之间
		11:8	pin0						GPIOA 到 GPIOH 的 pin id 在 0 到 15 之间，GPIOI 的 pin id 在 0 到 7 之间
		15:12	port0						0: 未使用 1: 存储区 A 2: 存储区 B 3: 存储区 C 4: 存储区 D 5: 存储区 E 6: 存储区 F 7: 存储区 G 8: 存储区 H 9: 存储区 I 10-15: 未使用
		19:16	mode1						idem CFG5.mode0
		23:20	afmux1						idem CFG5.afmux0
		27:24	pin1						idem CFG5.pin0
		31:28	port1						idem CFG5.port0

表 17. OTP 列表字 0 到 31（续）

字	名称	位	说明	prog	wplock	swprlock1	swprlock2	swprlock3	详细说明
6	CFG6	启动/AFmux 配置 2							
		3:0	mode2	u	n	-	sw	-	idem CFG5.mode0
		7:4	afmux2						idem CFG5.afmux0
		11:8	pin2						idem CFG5.pin0
		15:12	port2						idem CFG5.port0
		19:16	mode3						idem CFG5.mode0
		23:20	afmux3						idem CFG5.afmux0
		27:24	pin3						idem CFG5.pin0
		31:28	port3						idem CFG5.port0
7	CFG7	启动/AFmux 配置 3							
		3:0	mode4	u	n	-	sw	-	idem CFG5.mode0
		7:4	afmux4						idem CFG5.afmux0
		11:8	pin4						idem CFG5.pin0
		15:12	port4						idem CFG5.port0
		19:16	mode5						idem CFG5.mode0
		23:20	afmux5						idem CFG5.afmux0
		27:24	pin5						idem CFG5.pin0
		31:28	port5						idem CFG5.port0
8	CFG8	意法半导体预留							
		31:0	意法半导体预留	s	n	swp	swp	swp	-
9	CFG9	Boot ROM/配置							
		0	Boot_traces_ disable	u	n	-	-	-	0: 否 1: 是
		1	hse_frequency_ autodetection_ disable	u					0: 否 1: 是
		2	hse_bypass_ disable	u					0: 否 1: 是
		3	意法半导体预留	s					-
		4	emmc_128k_boot_ partition						支持具有 128 KB 启动分区的 *e*•MMC： 0: 否 1: 是
		5	ssp_req						设置为 1 请求 SSP 过程

表 17. OTP 列表字 0 到 31（续）

字	名称	位	说明	prog	wplock	swprlock1	swprlock2	swprlock3	详细说明
9	CFG9	6	ssp_success	s	n	-	-	-	1: 成功运行 SSP 过程
		7	fsbl_decrypt_prio						FSBL 解密优先级： 0: 使用 CRYP 1: 使用 SAES
		13:8	尚未使用						-
		14	snand_need_ plane_select_1	u					0：无需选择 sNAND 层 1: 需要选择 sNAND 层
		17:15	pnand_number_of _ecc_bits_1	u					0: ECC 未设置。 1: ECC 1bit (Hamming) 2: ECC 4bit (BCH4) 3: ECC 8bit (BCH8) 4: 片上 ECC
		18	pnand_bus_ width_1	u					0: 8 位 1: 16 位
		26:19	nand_nb_of_ blocks_1	u					以 256 个块为单位的块数
		28:27	nand_block_size_1	u					块大小，以页数计： 0: 每块 64 页 1: 每块 128 页 2: 每块 256 页
		30:29	nand_page_size_1	u					0: 2 KB 1: 4 KB 2: 8 KB
		31	pnand_param_ stored_in_otp_1	u					0: 否 1: 使用 Nand 参数位 15 到 30
10	CFG10	**Boot ROM/配置**							
		0	nand_config_ distribution	u	n	-	-	-	0: 在 nand*_2 字段中配置 pNAND，在 nand*_1 字段中配置 sNAND 1: 在 nand*_2 字段中配置 sNAND，在 nand*_1 字段中配置 pNAND
		1	snand_need_plane _select_2	u					0: 无需选择 sNAND 层 1: 需要选择 sNAND 层
		4:2	pnand_number_of _ecc_bits_2	u					0: ECC 未设置。 1: ECC 1bit (Hamming) 2: ECC 4bit (BCH4) 3: ECC 8bit (BCH8) 4: 片上 ECC
		5	pnand_bus_ width_2	u					0: 8 位 1: 16 位

表 17. OTP 列表字 0 到 31（续）

字	名称	位	说明	prog	wplock	swprlock1	swprlock2	swprlock3	详细说明
10	CFG10	13:6	nand_nb_of_ blocks_2	u	n	-	-	-	以 256 个块为单位的块数
		15:14	nand_block_size_2	u					块大小，以页数计： 0: 每块 64 页 1: 每块 128 页 2: 每块 256 页
		17:16	nand_page_size_2	u					0: 2 KB 1: 4 KB 2: 8 KB
		20:18	rng_htcr_value	u					0: 默认 RNG 值，RNG_HTCR 未修改。 1: 0xA2B3 (unsigned long) 2: 0xAA74 (unsigned long) 3: 0xA6BA (unsigned long) 4: 0x9AAE (unsigned long) 5: 0x72AC (unsigned long) 6: 0xAAC7 (unsigned long) 其他值：默认 RNG 值，RNG_HTCR 未修改。
		31:21	意法半导体预留	s					-
11	CFG11	**设备配置**							
		31:0	用于 OEM 设备配置	u	n	-	-	-	-
12	CFG12	**工程用途**							
		31:0	意法半导体预留	s	y	-	-	-	-
13	ID0	**唯一设备 ID（96 位）**							
		31:0	UID[31:0]	s	y	-	-	-	-
14	ID1	**唯一设备 ID（96 位）**							
		31:0	UID[63:32]	s	y	-	-	-	-
15	ID2	**唯一设备 ID（96 位）**							
		31:0	UID[95:64]	s	y	-	-	-	-

表 17. OTP 列表字 0 到 31（续）

字	名称	位	说明	prog	wplock	swprlock1	swprlock2	swprlock3	详细说明
16	HW0	硬件微调/模拟 1							
		5:0	意法半导体预留	s	y	-	-	-	-
		9:6	I/O 补偿微调						用于 I/O 校准
		11:10	保留						-
		15:12	1V8 参考微调						1V8 调压器微调
		21:16	LSI 微调						-
		26:22	1V1 参考微调						1V1 调压器微调
		31:27	意法半导体预留						-
17	HW1	硬件微调/模拟 2							
		7:0	CSITRIM[7:0]	s	y	-	-	-	CSI 微调
		19:8	HSITRIM[11:0]						HSI 微调
		23:20	TEMPHTRIM[3:0]						高温阈值微调
		27:24	TEMPLTRIM[3:0]						低温阈值微调
		31:28	VBATHTRIM[3:0]						高 V_{BAT} 阈值微调

表 17. OTP 列表字 0 到 31（续）

字	名称	位	说明	prog	wplock	swprlock1	swprlock2	swprlock3	详细说明
18	HW2	硬件微调/PKG/WDG 配置							
		2:0	意法半导体预留	s	n	-	-	-	-
		3	IWDG1_HW	u					IWDG1 启动 0: 通过软件启动 1: 自动启动
		4	IWDG2_HW	u					IWDG2 启动 0: 通过软件启动 1: 自动启动
		5	IWDG1_FZ_STOP	u					在停止模式下冻结 IWDG1 0: 否 1: 在停止模式下冻结
		6	IWDG2_FZ_STOP	u					在停止模式下冻结 IWDG2 0: 否 1: 在停止模式下冻结
		7	IWDG1_FZ_STANDBY	u					在待机模式下冻结 IWDG1 0: 否 1: 在待机模式下冻结
		8	IWDG2_FZ_STANDBY	u					在待机模式下冻结 IWDG2 0: 否 1: 在待机模式下冻结
		9	RNG_BYPASS_DISABLE	s					RNG 噪声观察 0: 使能 1: 禁止
		10	意法半导体预留	s					-
		12:11	SELINBORH[1:0]	u					00: BOR 禁止 01: BOR = 2.1 V 10: BOR = 2.4 V 11: BOR = 2.7 V
18	HW2	13	PRODUCT_BELOW_2V5	u	n	-	-	-	1: 当 V_{DD} 低于 2.5 V 时需要，以便采用 SYSCFG 高速低压使能寄存器 HSLVEN 位。 有关更多信息，请参见*第 12.6.17 节：SYSCFG 高速低压使能寄存器 x (SYSCFG_HSLVENxR)* 和*表 79：高速低电压接口*。
		31:14	意法半导体预留	s					-

表 17. OTP 列表字 0 到 31（续）

字	名称	位	说明	prog	wplock	swprlock1	swprlock2	swprlock3	详细说明
19	HW3	**DTS 的硬件微调/校准**							
		15:0	TS1_FMT0[15:0]	s	y	-	-	-	在 T0 测量的 DTS 频率
		31:16	TS1_RAMP_COEF [15:0]						DTS 斜坡系数
20	HW4	**V_{REFINT}，DTS 的硬件微调/校准**							
		6:0	TRIM_DTP_R[6:0]	s	y	-	-	-	DTS R 微调
		8:7	TS1_T0[1:0]						DTS T0 值
		15:9	意法半导体预留						-
		31:16	VREFINT_CAL [15:0]						ADC VBG 测量以校正 VREF
21	HW5	**VREFBUF，SDMMC I/O 的硬件微调/校准**							
		23:0	VREFBUFTRIM [23:0]	s	y	-	-	-	每 4 个 VREFBUF 设置对应 6 个微调位
		27:24	SD1COMPCELL [3:0]						V_{DDSD1} I/O 校准
		31:28	SD2COMPCELL [3:0]						V_{DDSD2} I/O 校准
22	HW6	31:0	意法半导体预留		n	swp	swp	swp	-
23	HW7	**模拟温度传感器的硬件微调/校准**							
		15:0	TS_CAL1[15:0]	s	y	-	-	-	T0 中的 ADC TEMP 测量
		31:16	TS_CAL2[15:0]						T1 中的 ADC TEMP 测量
24	PKH0	-							
		31:0	公钥哈希(2)	u/s	n/y(3)	sw	sw	sw	-
25	PKH1	-							
		31:0	公钥哈希(2)	u/s	n/y(3)	sw	sw	sw	-
26	PKH2	-							
		31:0	公钥哈希(2)	u/s	n/y(3)	sw	sw	sw	-
27	PKH3	-							
		31:0	公钥哈希(2)	u/s	n/y(3)	sw	sw	sw	-
28	PKH4	-							
		31:0	公钥哈希(2)	u/s	n/y(3)	sw	sw	sw	-
29	PKH5	-							
		31:0	公钥哈希(2)	u/s	n/y(3)	sw	sw	sw	-
30	PKH6	-							
		31:0	公钥哈希(2)	u/s	n/y(3)	sw	sw	sw	-

表 17. OTP 列表字 0 到 31（续）

字	名称	位	说明	prog	wplock	swprlock1	swprlock2	swprlock3	详细说明
31	PKH7	-							
		31:0	公钥哈希(2)	u/s	n/y(3)	sw	sw	sw	-

1. 这些位禁止在没有安全启动选项（参见产品数据表订购信息中的“安全选项”）的产品上编程为 **OTP-SECURED closed_device**（参见 BSEC 中的*表 11：OTP 模式定义*）。
   例如，在 STM32MP13xA 和 STM32MP13xD 上将这些位设置为的 **OTP-SECURED closed_device** 将导致这些产品在启动时永久失败。为避免这种情况，建议在这类设备上永久写入锁定 OTP 字 0（参见 BSEC 中的*OTP 永久写入锁定*一节）。
2. 公钥哈希由 32 字节组成：哈希 = byte0 byte1 byte2 … byte31。这些字节以每 4 字节 8 个字的形式存储在 OTP 中。哈希表的前四个字节存储在 PKH0 中，接下来四个存储在 PKH1 中，依此类推。在每个字中，编号最低的字节存储在最高有效位，编号最高的字节存储在最低有效位。
   例如：如果哈希 = 00 01 02 03 04 05 06 07 08… 那么 PKH0 = 0x00010203，PKH1 = 0x04050607，等等，…
3. 这个字在没有安全启动选项的设备上永久写锁定（参见产品数据表订购信息中的“安全选项”）。

表 18. OTP 列表字 32 到 95

字	名称	位	说明	prog	wplock	swprlock1	swprlock2	swprlock3
32	XK0	-						
		31:0	意法半导体预留	s	y	swrc	swrc	swrc
33	XK1	-						
		31:0	意法半导体预留	s	y	swrc	swrc	swrc
34	XK2	-						
		31:0	意法半导体预留	s	y	swrc	swrc	swrc
35	XK3	-						
		31:0	意法半导体预留	s	y	swrc	swrc	swrc
36	XK4	-						
		31:0	意法半导体预留	s	y	swrc	swrc	swrc
37	XK5	-						
		31:0	意法半导体预留	s	y	swrc	swrc	swrc
38	XK6	-						
		31:0	意法半导体预留	s	y	swrc	swrc	swrc
39	XK7	-						
		31:0	意法半导体预留	s	y	swrc	swrc	swrc
40	XK8	**SSP 的 ST ECDSA 公钥芯片证书**						
		31:0	证书位(1)	s	y	swp	swp	swp
41	XK9	**SSP 的 ST ECDSA 公钥芯片证书**						
		31:0	证书位(1)	s	y	swp	swp	swp

表 18. OTP 列表字 32 到 95（续）

字	名称	位	说明	prog	wplock	swpr lock1	swpr lock2	swpr lock3
42	XK10	SSP 的 ST ECDSA 公钥芯片证书						
		31:0	证书位(1)	s	y	swp	swp	swp
43	XK11	SSP 的 ST ECDSA 公钥芯片证书						
		31:0	证书位(1)	s	y	swp	swp	swp
44	XK12	SSP 的 ST ECDSA 公钥芯片证书						
		31:0	证书位(1)	s	y	swp	swp	swp
45	XK13	SSP 的 ST ECDSA 公钥芯片证书						
		31:0	证书位(1)	s	y	swp	swp	swp
46	XK14	SSP 的 ST ECDSA 公钥芯片证书						
		31:0	证书位(1)	s	y	swp	swp	swp
47	XK15	SSP 的 ST ECDSA 公钥芯片证书						
		31:0	证书位(1)	s	y	swp	swp	swp
48	XK16	SSP 的 ST ECDSA 公钥芯片证书						
		31:0	证书位(1)	s	y	swp	swp	swp
49	XK17	SSP 的 ST ECDSA 公钥芯片证书						
		31:0	证书位(1)	s	y	swp	swp	swp
50	XK18	SSP 的 ST ECDSA 公钥芯片证书						
		31:0	证书位(1)	s	y	swp	swp	swp
51	XK19	SSP 的 ST ECDSA 公钥芯片证书						
		31:0	证书位(1)	s	y	swp	swp	swp
52	XK20	SSP 的 ST ECDSA 公钥芯片证书						
		31:0	证书位(1)	s	y	swp	swp	swp
53	XK21	SSP 的 ST ECDSA 公钥芯片证书						
		31:0	证书位(1)	s	y	swp	swp	swp
54	XK22	SSP 的 ST ECDSA 公钥芯片证书						
		31:0	证书位(1)	s	y	swp	swp	swp
55	XK23	SSP 的 ST ECDSA 公钥芯片证书						
		31:0	证书位(1)	s	y	swp	swp	swp
56	XK24	RMA 密码						
		31:0	RMA 密码	u	n(2)	swrc	swrc	swrc
57	XK25	-	用户可自由使用	u	n	c	c	swprc
58	XK26	-	用户可自由使用	u	n	c	c	swprc
59	XK27	-	用户可自由使用	u	n	c	c	swprc
60	XK28	-	用户可自由使用	u	n	c	c	swprc

表 18. OTP 列表字 32 到 95（续）

字	名称	位	说明	prog	wplock	swpr lock1	swpr lock2	swpr lock3
61	XK29	-	用户可自由使用	u	n	c	c	swprc
62	XK30	-	用户可自由使用	u	n	c	c	swprc
63	XK31	-	用户可自由使用	u	n	c	c	swprc
64	XK32	-	用户可自由使用	u	n	c	c	swprc
65	XK33	-	用户可自由使用	u	n	c	c	swprc
66	XK34	-	用户可自由使用	u	n	c	c	swprc
67	XK35	-	用户可自由使用	u	n	c	c	swprc
68	XK36	-	用户可自由使用	u	n	c	c	swprc
69	XK37	-	用户可自由使用	u	n	c	c	swprc
70	XK38	-	用户可自由使用	u	n	c	c	swprc
71	XK39	-	用户可自由使用	u	n	c	c	swprc
72	XK40	-	用户可自由使用	u	n	c	c	swprc
73	XK41	-	用户可自由使用	u	n	c	c	swprc
74	XK42	-	用户可自由使用	u	n	c	c	swprc
75	XK43	-	用户可自由使用	u	n	c	c	swprc
76	XK44	-	用户可自由使用	u	n	c	c	swprc
77	XK45	-	用户可自由使用	u	n	c	c	swprc
78	XK46	-	用户可自由使用	u	n	c	c	swprc
79	XK47	-	用户可自由使用	u	n	c	c	swprc
80	XK48	-	用户可自由使用	u	n	c	c	swprc
81	XK49	-	用户可自由使用	u	n	c	c	swprc
82	XK50	-	用户可自由使用	u	n	c	c	swprc
83	XK51	-	用户可自由使用	u	n	c	c	swprc
84	XK52	-	用户可自由使用	u	n	c	c	swprc
85	XK53	-	用户可自由使用	u	n	c	c	swprc
86	XK54	-	用户可自由使用	u	n	c	c	swprc
87	XK55	-	用户可自由使用	u	n	c	c	swprc
88	XK56	-	用户可自由使用	u	n	c	c	swprc
89	XK57	-	用户可自由使用	u	n	c	c	swprc
90	XK58	-	用户可自由使用	u	n	c	c	swprc
91	XK59	-	用户可自由使用	u	n	c	c	swprc
92	XK60	-	加密启动的 OEM 机密(3)	u	n	c	c	swprc
93	XK61	-	加密启动的 OEM 机密	u	n	c	c	swprc

表 18. OTP 列表字 32 到 95（续）

字	名称	位	说明	prog	wplock	swpr lock1	swpr lock2	swpr lock3
94	XK62	-	加密启动的 OEM 机密	u	n	c	c	swprc
95	XK63	-	加密启动的 OEM 机密	u	n	c	c	swprc

1. SSP 的 ST ECDSA 公钥芯片证书由 64 字节组成：密钥 = byte0 byte1 byte2 ... byte63。这些字节以每 4 字节 16 个字的形式存储在 OTP 中。密钥表的前四个字节存储在 XK8 中，接下来四个存储在 XK9 中，依此类推。在每个字中，编号最低的字节存储在最高有效位，编号最高的字节存储在最低有效位。
   例如：如果密钥 = 00 01 02 03 04 05 06 07 08… 那么 XK8 = 0x00010203，XK9 = 0x04050607，等等，…
2. 重要提示：编程后必须永久锁定。
3. 加密启动的 OEM 机密由 16 字节组成：密钥 = byte0 byte1 byte2 … byte15。这些字节以每 4 字节 4 个字的形式存储在 OTP 中。密钥表的前四个字节存储在 XK60 中，接下来四个存储在 XK61 中，依此类推。在每个字中，编号最低的字节存储在最高有效位，编号最高的字节存储在最低有效位。
   例如：如果密钥 = 00 01 02 03 04 05 06 07 08… 那么 XK60 = 0x00010203，XK61 = 0x04050607，等等，…

5 DDR3/LPDDR2/LPDDR3 控制器 (DDRCTRL)

5.1 简介

Portions Copyright (c) Synopsys, Inc. 保留所有权利。使用须经许可。

DDRCTRL 是一款 DDR-SDRAM 控制器，也是 Synopsys uMCTL2 DDR 控制器 v3.00a 的特定配置。

将 DDRCTRL 与 DDRPHYC（带有物理实用工具模块的 DDR PHY）结合使用，可打造出一个面向 DDR 内存子系统的完整内存接口解决方案。

5.2 DDRCTRL 特性

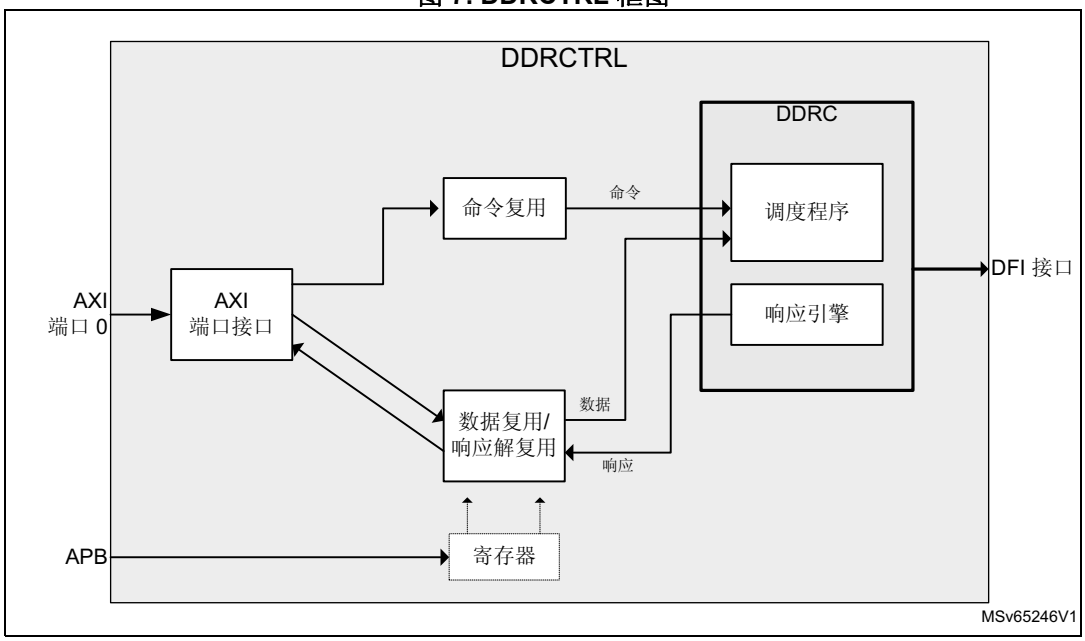
- 一个 64 位 AMBA® 4 AXI4 端口接口 (XPI)
- 与控制器异步的 AXI 时钟
- 所支持的标准：
  - JEDEC DDR3 SDRAM 规范 JESD79-3E，适用于带有 16 位接口的 DDR3/3L
  - JEDEC LPDDR2 SDRAM 规范 JESD209-2E，适用于带有 16 位接口的 LPDDR2
  - JEDEC LPDDR3 SDRAM 规范 JESD209-3B，适用于带有 16 位接口的 LPDDR3
- 通过符合 DFI 2.1 标准的 64 位接口连接到 DDRPHYC
- 高级调度器和 SDRAM 命令生成器
- DDRC 时钟与 DDR PHY/外部内存时钟（又称 SDR 模式）的频率比为 1:1
- AXI 端口与 DDRC 之间采用异步时钟（时钟交叉存在于 XPI 内）
- 高级 QoS 支持，包括 3 个用于读取的数据传输类和 2 个用于写入的数据传输类
- 可通过多个选项避免出现低优先级数据传输饥饿
- 可保证 AXI 端口上读后写 (WAR) 与写后读 (RAW) 的一致性
- 为突发长度选项（4、8、16）提供可编程支持
- 支持写入合并，可将多次写入同一地址合并为单次写入操作
- 单 rank 配置
- 支持在可编程时间内没有事务到达的情况下，SDRAM 自动进入和退出掉电模式
- 支持在没有事务到达的情况下，系统自动进入和退出时钟停止模式 (LPDDR2/3)
- 支持在可编程时间内没有事务到达的情况下，系统自动通过硬件低功耗接口在低功耗模式下运行
- 可编程页策略，用户可选择以下任意一种：
  - 访问页后使其保持打开（打开页策略）
  - 当控制器无法再访问某页时，将该页关闭
  - 在每次访问时自动进行预充电，并对页关闭模式进行如下优化：在为解决读写冲突和写读冲突而执行清空操作后，使页保持打开
- 支持进入和退出自刷新模式，具体如下：
  - 在可编程时间内没有事务到达的情况下，系统自动进入和退出自刷新模式
  - 在软件控制下进入和退出自刷新模式
- 支持在软件控制下进入和退出深度掉电模式 (LPDDR2)
- 支持在软件控制下显式更新 SDRAM 模式寄存器

- 灵活的地址映射器逻辑，允许对行、列、bank 位进行特定应用映射
- 为 1T 或 2T 时序提供可编程支持
- 用户可选择以下刷新控制选项：
  - 由控制器以可编程的平均间隔生成自动刷新
  - 至多可将 8 个控制器生成的刷新分组到一起，以便连续发出（这样可以降低页关闭频率，从而提高总体效率）
  - 一旦控制器生成的刷新完成分组，系统即可在控制器空闲了可编程的一段时间后，推测性地发出部分刷新
  - 能够禁止控制器生成的自动刷新
  - 能够通过直接软件请求发出刷新
  - 用户可选择对每个 bank 单独进行刷新，而非对 LPDDR2/3 的所有 bank 进行刷新

5.3 DDRCTRL 框图

简化的 DDRCTRL 框图如*图 7* 所示。

图 7. DDRCTRL 框图



5.4 DDRCTRL 架构概述

请参见*图 7* 中的 DDRCTRL 框图。

- AXI 端口接口 (XPI) 模块为连接 AXI 端口的接口提供了 AXI 总线协议处理、读取数据的数据缓冲和重新排序、数据总线大小/时钟域转换以及内存突发地址对齐功能。
- 端口仲裁器 (PA) 模块可面向 XPI 发出的地址，提供对延迟敏感的、基于优先级的仲裁。
- DDR 控制器 (DDRC) 模块中包含了一个逻辑 CAM（内容可寻址内存）。其中包含了一些命令相关信息，以供调度算法根据优先级、bank/rank 状态和 DDR 时序限制情况，以最优方式调度要发送到 PHY 的命令。
- APB 寄存器模块中包含了可供软件访问的寄存器。
- 系统将存储写入数据，直到将其相关命令发送到 PHY 为止。写入数据缓冲区位于 DDRC 内部。
- 读取数据将由 DDRC 中的响应引擎进行处理，并按照在 HIF 上调度的读取命令顺序返回。系统会将读取数据存储在重排序读取缓冲区（位于 XPI 内部）内，并将其按顺序返回到 AXI 端口。

5.4.1 AXI 端口接口 (XPI)

XPI 将处理地址、突发大小、突发长度和突发类型等 AXI 信息，并将 AXI 突发转换为需要转发到 PA 的读写请求。

在相反的方向上，XPI 会将来自 DDRC 的响应转换为适当的 AXI 响应。

读取地址通道特性：

- 支持任何长度的增量突发（<=256 字节）以及长度 <=16 的 WRAP 突发。
- 起始地址可不与 AXI 宽度对齐。
- 突发大小可小于 AXI 宽度。
- 读取地址队列 (RAQ) 配置为单个队列（名为“蓝色”队列）
- 在 RAQ 中进行时钟交叉操作
- RAQ 的深度设置为 4。
- 读取请求的生成基于 AXI 地址和突发信息，且被划分为与内存突发长度（BL2、BL4、BL8、BL16）相等的数据包
- 如果是未对齐突发，则第一个读取请求将是未对齐请求，而剩余的读取请求将是对齐请求。

  一般来说，重新对齐至内存突发边界可能会导致部分数据节拍遭到丢弃（影响带宽），并可能对读取数据和响应通道造成额外延迟。

写入地址通道特性：

- 写入地址队列 (WAQ) 用于存储来自给定端口的所有写入请求地址。
- 系统针对来自给定端口的所有 AXI ID 提供了一个专门的队列。
- 写入地址通道的行为与读取地址通道相似。
- 在 WAQ 中进行时钟交叉操作
- WAQ 的深度设置为 4。

读取数据和响应通道：

- 读取数据和响应通道具有一个单独的数据存储队列 (RDQ)
- 在 RDQ 中进行时钟交叉操作
- RDQ 的深度设置为 10
- 不同 ID 的数据存储在同一队列中，并会按照接受读取地址的顺序返回。
- 系统将为每个端口实现一个重排序读取缓冲区，可将该端口的读取数据重新排序为与 AXI 读取命令的顺序相同。
- 虚拟通道和动态映射（可对读取数据进行最优的重新排序）

5.4.2 端口仲裁器 (PA)

PA 会对从 AXI 端口发往 DDR 控制器 HIF 的命令请求进行仲裁。

PA 包含了多个层次的仲裁阶段，其中包括：

a) 读取/写入仲裁（单 HIF 选项）
b) 基于到期 VPR/到期 VPW 命令和端口老化情况的 2 优先级仲裁（超时 - 优先级 0）
c) 根据 DDRC 读取优先级 (HPR/LPR-VPR) 进行的读取请求 2 优先级仲裁
d) 基于内部端口老化的 32 优先级仲裁，或基于 AXI QoS 输入的 16 优先级仲裁（在 AXI 互连中编程）
e) 循环调度仲裁，用于解决端口在通过所有仲裁阶段后具有相同优先级的问题。

注： *端口老化指的是端口处于有请求但未获得 DDRC 访问权限这一状态的倒计时时长。*

端口超时指的是端口老化时间变为 0，我们也称之为最高优先级—— 优先级 0。

到期 VPR 指的是计数器已到期的可变优先级读取命令。读取/写入额度指的是 HIF 接口所使用的 DDRC“额度机制”。

5.4.3 主机接口 (HIF)

HIF 是一个用于连接 DDRCTRL 的内部接口（由 AXI 端口接口和端口仲裁器来处理此接口上的所有信号）。介绍该接口的目的只是为了帮助您理解 QoS 和多层仲裁。

系统将通过 HIF 接口将读取请求、写入请求以及写入数据从 PA 传输到 DDRC。
可通过以下方式对请求进行限制：

a) 一种额度机制，可确保在发出任何面向 DDRC 的请求之前，为其提供缓冲区空间。
b) 一种独立的停止机制，可在发生地址冲突时或是通过软件自刷新进入自刷新模式时，对请求进行限制。

系统将通过一个通用响应接口将读取数据发送回去

额度机制：

DDRC 采用额度机制来确保缓冲区不会溢出。向 DDRC 发出请求的接口，只能请求已获得发出额度的命令。

系统将分别按照以下三种命令类型对额度进行跟踪：

- 高优先级读取 (HPR)
- 低优先级读取 (LPR)
- 写入 (WR)

接口逻辑包括用于跟踪额度数的计数器。每次 DDRC 发放额度时，额度计数都会递增。

VPR 命令将计入 LPR 额度池，而 VPW 命令将计入 WR 额度池。

读取请求：

当 LPR 或 HPR 额度计数大于零时，接口可以向 DDRC 发出读取请求。

写入请求：

当 WR 额度计数大于零时，接口可以向 DDRC 发出写入请求。

5.4.4 DDR 调度器 (DDRC)

DDRC 可为命令生成队列，以便进行调度或将其用于 SDRAM 命令生成单元。DDRC 中包含了以下 3 种存储：

- 通用读取 CAM 中的 HPR 和 LPR 存储
- 写入 CAM 中的 WR 存储

两个 CAM 都有 16 个条目，且 HPR/LPR 拆分可通过软件编程的方式来完成。

DDRC 支持用户根据 QoS 和超时情况来管理事务服务控制 (TSC)。

5.4.5 APB 接口 (APB)

DDRCTRL 中有以下三类寄存器：

- 动态寄存器
- 准动态寄存器
- 静态寄存器

动态寄存器可在运行过程中随时写入。

静态寄存器只能在控制器处于复位状态时写入。
当 DDRCTRL 处于复位状态或是复位完成后的特定条件下时，可以写入准动态寄存器。要写入准动态寄存器，必须在编程序列开始时将 DDRCTRL_SWCTL.SW_DONE 位设置为 0，并在编程序列结束时将其重新置 1。之后必须轮询 DDRCTRL_SWSTAT.SW_DONE_ACK，以确认写入操作已完成。

有关动态和准动态寄存器的更多信息，请参见 uMCTL2 数据手册 [1]

5.4.6 DFI 接口 (DFI)

DDRCTRL 包含了一个 DFI MC（存储控制器）接口，可用于连接到符合 DFI 标准的 DDRPHYC，以便将地址、控制信号和数据传输到 PHY。

可将 DFI 接口细分为以下接口组：

- 控制器接口
- 写入数据接口
- 读取数据接口
- 更新接口
- 状态接口
- 训练接口（受 uMCTL2 支持的，独立于 PHY 之外的模式）
- 低功耗控制接口

DDRCTRL 及其 DFI 接口将以单倍数据速率 (SDR) 时钟运行（DFI 规范中称之为 1:1 频率比模式），因此所有 DFI 命令和地址信号的宽度都与等效 DDR SDRAM 信号的宽度相等。DFI 数据信号的宽度是等效 DDR SDRAM 信号宽度的两倍。

DDRCTRL 将以 SDR 时钟运行，而 DDRPHYC 将在数据总线上执行 SDR 到 DDR 的速率转换。在 1:1 频率比模式下，core_ddrc_core_clk 和 CK/CK# 将以相同的 SDR 时钟频率运行。

5.5 DDRCTRL 功能描述

5.5.1 事务服务控制 (TSC) 和服务质量 (QoS)

事务服务控制可帮助用户仔细管理以下事务：

- 消耗资源和时间较多的读/写总线周转
- 读取请求的优先级通常倾向于高优先级数据传输，同时也会防止出现低优先级数据传输饥饿

该功能可在面向每种数据传输类型的简单 2 状态机中，通过完全可配置控制来实现。2 状态机的状态将决定读/写操作的处理时机，以及任何给定时刻的相对优先级（高优先级读取与低优先级读取）。

可将事务分为五个数据传输类，包括 3 个读取类和 2 个写入类：

- 低优先级读取 (LPR)
- 可变优先级读取 (VPR)
- 高优先级读取 (HPR)
- 正常优先级写入 (NPW)
- 可变优先级写入 (VPW)

数据传输类需根据 SOC AXI 互连 (AXIM) 设置的 AXI QoS 值 (AxQOS[3:0]) 和 QoS 区域设置进行确定。

HPR 和 VPR/LPR 存储在一个通用读取 CAM 中；该 CAM 可根据 DDRCTRL_SCHED.lpr_num 进行拆分。

在发生由 DDRCTRL_PERFVPR1.vpr_timeout 设置的超时后，VPR 命令将变为到期 VPR。到期 VPR 将转为最高优先级（优先级 0）。

VPW 和 NPW 存储在一个通用写入 CAM 中；在发生由 DDRC_PERFVPW1.vpw_timeout 设置的超时后，VPW 命令将转为到期 VPR。到期 VPW 命令将转为最高优先级（优先级 0）。

以下是与事务服务和 QoS 控制相关的寄存器列表：

- DDRCTRL_SCHED：主调度器控制
- DDRCTRL_SCHED1：页关闭定时器
- DDRCTRL_PERFHPR1：控制 HPR 饥饿和 run_length
- DDRCTRL_PERFHPR1：控制 HPR 饥饿和 run_length
- DDRCTRL_PERFLPR1：控制 LPR 饥饿和 run_length
- DDRCTRL_PCFGR_0/1：面向端口 0/1 的 AXI 端口读取控制
- DDRCTRL_PCFGW_0/1：面向端口 0/1 的 AXI 端口写入控制
- DDRCTRL_PCFRQOS0_0/1 和 DDRCTRL_PCFRQOS1_0/1：面向端口 0/1 和端口超时的 AXI 端口读取 QoS 区域映射。
- DDRCTRL_PCFWQOS0_0/1 和 DDRCTRL_PCFWQOS1_0/1：面向端口 0/1 和端口超时的 AXI 端口写入 QoS 区域映射。

5.5.2 页策略选项

打开页策略将由 DDRC_SCHED.pageclose = 0 进行设置。

任何 bank 都将保持打开状态，直到因页丢失或刷新超时导致需要将其关闭为止。

智能预充电是打开页策略与关闭页策略之间的折中方案，可通过 DDRC_SCHED.pageclose = 1 和 DDRC_SCHED.pageclose_timer 值进行设置，从而使用户在安排自动预充电或显式预充电之前，能够对 CAM 中的 bank 没有页命中时的等待时间进行控制。

如果 CAM 中没有面向某个 bank 的挂起命令，则此定时器支持在可配置的时钟周期数内使页保持在打开状态。如果数据流的上游（例如 XPI/PA）存在面向相同 bank/ 页的挂起命令，则 DDRCTRL 有机会将其作为打开页命令进行调度。

5.5.3 节能特性

DDRCTRL 支持从 SDRAM、DDRPHYC 和 SOC 入手，通过多种方法来降低系统内部的能耗。

预充电掉电：(PDN)

DDRCTRL 可通过控制以下参数，将 SDRAM 自动置于预充电掉电模式：

- DDRCTRL_PWRCTRL.powerdown_en：在经过一定数量的空闲周期后进入 PDN (PDE) 模式
- DDRCTRL_PWRTMG.powerdown_to_x32：以 32 的倍数计算的空闲时钟周期数，用以进入 PDN 模式
- DDRCTRL_PWRTMG1.txp：从 PDN 退出 (PDX) 到任何命令后的最短等待时间（PDN 模式期间 DLL 保持锁定）

注： *PDN 模式期间将执行刷新命令，且必须保持时钟；DDRCTRL 在预充电掉电退出 (PDE) 之前会对所有 bank 进行预充电。*
系统不支持（通过设置 MR0[12] =1）实现 DDR3 慢速掉电。

自刷新：(SR)

SR 可为 SDRAM 带来重要的节能机会。此外，在 SR 过程中，系统可以绕过 DDRPHY C 中的 DLL，且可以停止 DDRCTRL 时钟。

在以下情况下，DDRCTRL 会将 SDRAM 置于自刷新 (SR) 模式：

- 预定义时间内没有读/写操作时，自动进入 SR 模式。该模式可通过以下参数进行控制：
  - DDRCTRL_PWRCTRL.selfref_en：在经过一定数量的空闲周期后进入 SR (SRE) 模式
  - DDRCTRL_PWRTMG.selfref_to_x32：以 32 的倍数计算的空闲时钟周期数，用以进入 SR 模式。
- 由软件控制的 SR：该模式可通过以下参数进行控制：
  - DDRCTRL_PWRCTRL.selfref_sw：设置此位的操作被称为 self_refresh 软件入口点。
- 由硬件控制的 SR：用户可通过以下方式为该模式提供支持：
  - 位于 DDRCTRL 外部且受 RCC 控制的 FSM（有关该模式的详细信息，请参见*第 10 章：复位和时钟控制 (RCC)*）。当 SoC 进入停止模式时，DDRCTRL 可将 SDRAM 置于自刷新模式。

深度掉电：(DPD)

注： *DPD 仅适用于 LPDDR2/3。*
尽管我们可以采用这种节能方式，但在 DPD 期间 SDRAM 内容会丢失，因此 DPD 与关闭 DDR 实际上并无区别，也不存在任何明显的优势。

DDR3 DLL 关闭模式：

系统可在低频下支持 DDR3 DLL 关闭模式，但需要设置 MRS (MR1[0] =1)，以禁止 DLL（DLL 默认处于开启状态）。应使用软件时钟移除序列：

```
Write 0 to DDRC_PCTRLn.port_en //to block AXI port n
Poll DDRC_PSTAT.rd_port_busy = 0
Poll DDRC_PSTAT.wr_port_busy = 0
Write 1 to DDRC_PWRCTL.selref_en // move to self-refresh
Poll DDRC_PSTAT.selfref_type = 2b10
... change AXI and DDRC clocks to lower value used w/ DLL OFF
...MRS to disable DLL
Write 0 to DDRC_PWRCTL.selref_en // exit self-refresh
Poll DDRC_PSTAT.selfref_type = 2b00
Write 1 to DDRC_PCTRLn.port_en // to unlock AXI ports
```

DDRPHY DLL 旁路模式：

该模式可禁止 DDRPHYC DLL，从而在低频下节省大量电力；该模式必须由软件进行控制。软件序列可将 SDRAM 转换到自刷新模式，且用户可在此期间更改时钟频率。

5.5.4 地址映射器

DDRCTRL 地址映射器可在 XPI 层面将 AXI 端口读写请求转换为行、bank 和列 SDRAM 地址。

地址映射非常灵活，可根据 DDR_ADDRMAP1 到 DDR_ADDRMAP11 寄存器设置进行定义。

用户可根据 SDRAM 架构（行/bank/列）以及全宽或半宽接口模式对常规的非交错行/bank/列 (RBC) 以及 bank/行/列 (BRC) 进行简单的编程。

5.5.5 DRAM 时序参数

DDRCTRL 会根据在寄存器 DDRCTRL_DRAMTMGn 中编程的多项时序限制向 SDRAM 调度命令；这些时序限制将根据 SDRAM JEDEC 时序参数与频率进行确定。

以下寄存器与时序限制有关

- DDRCTRL_DRAMTMG0 到 DDRCTRL_DRAMTMG14
- DDRCTRL_ODTTMG
- DDRCTRL_DFITMG0 和 DDRCTRL_DFITMG1

5.5.6 SDRAM 初始化序列

初始化由 DDRPHYC 进行控制，可支持面向每种 DDR 类型的 PUBL 内置初始化序列，并由 PIR 寄存器负责触发。有关更多信息，请参见*第 7 章：DDR 物理接口控制 (DDRPHYC)*。

5.5.7 刷新控制

可以使用 DDRCTRL 中的自动刷新功能，或是通过直接软件请求刷新命令来发出刷新。DDRCTRL_RFSHCTL3.dis_auto_refresh 可用于选择刷新方法。当此位设置为 1 时，uMCTL2 将使用直接软件请求刷新命令。当其设置为 0 时，系统将使用内部自动刷新功能。

使用直接软件请求刷新命令进行刷新

请按照以下步骤将 DDRCTRL 置于直接软件请求刷新命令模式：

1. 将 DDRCTRL_RFSHCTL3.dis_auto_refresh 位置为 1。寄存器位设置完成后，DDRCTRL 将检查是否存在挂起的刷新。一旦存在任何挂起的刷新，系统将立即使用 DDRCTRL 内部的“关键刷新”功能将其发出。发出这些刷新后，DDRCTRL 内部的所有刷新定时器都将复位为 0。这些定时器只有在系统使能自动刷新功能时才会重新激活。
2. SoC 内核必须持续跟踪 SDRAM 的刷新要求。
3. 可通过将寄存器位 DDRCTRL_DBGCMD.rank0_refresh 设置为 1 来发出刷新命令。一旦存储刷新请求，相应的寄存器位将自动清零。仅当 DDRCTRL_DBGSTAT.rank0_refresh_busy 为低电平时，SoC 内核才能启动刷新操作。最早会由 DDRCTRL 向 SDRAM 发出刷新。
4. 由软件驱动的刷新命令会加载到一个 9 条目缓冲区中，并由 DFI 上的 DDRCTRL 尽快合法地发出。如果缓冲区饱和，则 DBGSTAT.rank0_refresh_busy 将保持在生效状态，以防止软件启动进一步的刷新。

使用 DDRCTRL 内部的自动刷新功能进行刷新

DDRCTRL 可提供高级刷新控制。除了完全可配置的刷新限制（tRFC(min) 和 tREFI），DDRCTRL 还支持通过编程将针对 SDRAM 上每一 rank 的刷新汇集到一起，从而减少刷新消耗的带宽并增加在空闲周期处理刷新的可能性。

以下寄存器与刷新控制有关：

- DDRCTRL_RFSHCTL0
- DDRCTRL_RFSHCTL3
- DDRCTRL_RFSHTMG

进行刷新控制的目的包括：

- 降低刷新周期对带宽的影响
- 增加在空闲期间处理刷新的可能性
- 支持通过细粒度控制，在获得汇集刷新带来的各项好处与承受由此带来的巨大延迟之间找到完美的平衡点

有关更多信息，请参见寄存器说明；除了根据 tRFC(min) 和 tREFI 定期发出刷新之外，还有以下几种可行的优化选项：突发刷新、推测性刷新和逐 bank 刷新（适用于 LPDDR2/3）。

5.5.8 ZQ 校准

此功能适用于所有 DDR 类型。DDRCTRL 可以使用 ZQ 校准命令，根据 PVT（制程、电压、温度）来校准 SDRAM RON（电阻开启）和 ODT（片上端接）值。

DDRPHYC 支持在初始化时进行 RON 和 ODT 校准；有关更多信息，请参见*第 7 章：DDR 物理接口控制 (DDRPHYC)*。

DDRCTRL 可以发出周期校准，从而减少常规校准间隔所需的时间。(ZQCS)。

DDRCTRL 在退出自刷新模式后也可发出长校准 (ZQCL)

以下寄存器与 ZQ 校准控制有关：

- DDRCTRL_ZQCTL0
- DDRCTRL_ZQCTL1
- DDRCTRL_ZQCTL2
- DDRCTRL_ZQSTAT

有关 ZQ 校准的更多信息，请参见相应 SDRAM 的 JEDEC 规范。

5.6 DDRCTRL 配置

DDRCTRL 是 uMTCL2 的一项配置，其中包含了以下主要硬件配置选项：

- DDR 模式支持：DDR3/LPDDR2/LPDDR3
- 频率比：1:1
- 仲裁器：包括端口仲裁器 (PA)
- 部分写入支持
- CAM 深度：16
- 不支持双 HIF
- 可变优先级读取和可变优先级写入
- 逐端口配置单个读取地址队列

5.7 DDRCTRL 寄存器

DDRCTRL 已配置为支持 DDR3/LPDDR2/LPDDR3。在接下来的寄存器说明部分，请忽略所有有关其他标准（mDDR、DDR2、DDR4、LPDDR4）的参考信息，因为这些标准不适用。

根据以下内容，多个寄存器值取决于 DDRCTRL 的配置：

- 包括仲裁器 (UMCTL2_INCL_ARB = 1)
- 部分写入支持 (UMTCL2__PARTIAL_WR = 1)
- 可变优先级读取 (UMCTL2_VPR_EN = 1)
- 可变优先级写入 (UMCTL2_VPW_EN = 1)

DDRCTRL 寄存器只能按字（32 位）进行访问。

对未映射寄存器进行读写访问会引起 APB SLVERR。

针对只读寄存器的写入访问将被忽略，不会引起 APB SLVERR。

位字段中的编程模式标记描述了该位字段可以采用的编程方式：

- **静态**：位字段仅会在 DDRCTRL 复位释放之前写入。
- **动态**：位字段可在运行期间写入。
- **准动态**：位字段可在运行期间按照特定条件和程序要求写入（*第 5.4.5 节：APB 接口 (APB)* 中对此进行了说明）。

系统不支持四分之一总线宽度 (MSTR.data_bus_width = 10)。

系统不支持半总线宽度 (MSTR.data_bus_width = 01)。

有关寄存器总结，请参见：*第 290 页的表 19*.

5.7.1 DDRCTRL 主寄存器 0 (DDRCTRL_MSTR)

DDRCTRL master register 0

偏移地址：0x000

复位值：0x0004 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BURST_RDWR[3:0]			
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DLL_OFF_MODE	Res.	DATA_BUS_WIDTH[1:0]		Res.	EN_2T_TIMING_MODE	BURSTCHOP	Res.	Res.	Res.	Res.	Res.	LPDDR3	LPDDR2	Res.	DDR3
rw		rw	rw		rw	rw						rw	rw		rw

位 31:20 保留，必须保持复位值。

位 19:16 **BURST_RDWR[3:0]**：所使用的 SDRAM 突发长度：
- 0001 - 突发长度 2（仅适用于 mDDR）
- 0010 - 突发长度 4
- 0100 - 突发长度 8
- 1000 - 突发长度 16（仅适用于 mDDR、LPDDR2 和 LPDDR4）

所有其他值均保留。

此参数将控制访问 SDRAM 时使用的突发大小。此参数必须与 SDRAM 中的突发长度模式寄存器设置相匹配。（对于 DDR3 和 DDR4 的 BC4/8 实时模式，请将此位字段设置为 0x0100）当 MEMC_BURST_LENGTH 为 8 时，系统不支持将 AXI 端口的突发长度设为 2。

仅当控制器在 1:1 频率模式下运行时，系统才支持将突发长度设为 2。

对于 DDR3、DDR4 和 LPDDR3，必须将其设置为 0x0100 (BL8)。

对于 LPDDR4，必须将其设置为 0x1000 (BL16)。

编程模式：静态

位 15 **DLL_OFF_MODE**：当 DDRCTRL 和 DRAM 必须设置为 DLL 关闭模式以实现低频运行时，将其设置为 1。

当需要将 DDRCTRL 和 DRAM 置于 DLL 开启模式以实现正常频率运行时，将其设置为 0。

如果使能了 DDR4 CRC/奇偶校验重试 (CRCPARCTL1.crc_parity_retry_enable = 1)，则系统将不支持 dll_off_mode，并且必须将此位设置为“0”。

编程模式：准动态组 2

位 14 保留，必须保持复位值。

位 13:12 **DATA_BUS_WIDTH[1:0]**：选择由 SDRAM 使用的 DQ 总线宽度的比例
- 00 - 将全 DQ 总线宽度用于 SDRAM
- 01 - 保留
- 10 - 保留
- 11 - 保留

编程模式：静态

位 11 保留，必须保持复位值。

位 10 **EN_2T_TIMING_MODE**：如果为 1，则 DDRCTRL 将使用 2T 时序。否则，将使用 1T 时序。在 2T 时序中，SDRAM 总线上的所有命令信号（片选信号除外）都将保持 2 个时钟周期。片选信号将在命令的第二个周期内生效。

注意：LPDDR2/LPDDR3/LPDDR4 模式下不支持 2T 时序

注意：如果设置了配置参数 MEMC_CMD_RTN2IDLE，则系统将不支持 2T 时序

注意：DDR4 减速模式下不支持 2T 时序。

注意：共享 AC 双通道模式下不支持 2T 时序，且寄存器值为无关值。

编程模式：静态

位 9 **BURSTCHOP**：如果置 1，请在 DDR3/DDR4 中使能突发突变（BC4 或 8 实时模式）。用于读取的突发突变仅会在 HIF 配置（未设置 UMCTL2_INCL_ARB）中执行，其前提是必须处于全总线宽度模式下 (MSTR.data_bus_width = 00)，且 MEMC_BURST_LENGTH 为 8 或 16。用于写入的突发突变仅会在使能了部分写入 (UMCTL2_PARTIAL_WR=1) 且禁止了 CRC (CRCPARCTL1.crc_enable = 0) 的情况下执行。

不支持 BC4（固定）模式。

编程模式：静态

位 8:4 保留，必须保持复位值。

位 3 **LPDDR3**：选择 LPDDR3 SDRAM

- 1 - 正在使用 LPDDR3 SDRAM 设备。
- 0 - 正在使用非 LPDDR3 设备

仅存在于配置为支持 LPDDR3 的设计中。

编程模式：静态

位 2 **LPDDR2**：选择 LPDDR2 SDRAM

- 1 - 正在使用 LPDDR2 SDRAM 设备。
- 0 - 正在使用非 LPDDR2 设备

仅存在于配置为支持 LPDDR2 的设计中。

编程模式：静态

位 1 保留，必须保持复位值。

位 0 **DDR3**：选择 DDR3 SDRAM

- 1 - 正在使用 DDR3 SDRAM 设备
- 0 - 正在使用非 DDR3 SDRAM 设备

仅存在于支持 DDR3 的设计中。

编程模式：静态

5.7.2 DDRCTRL 工作模式状态寄存器 (DDRCTRL_STAT)

DDRCTRL operating mode status register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	SELFREF_CAM_NOT_EMPTY	Res.	Res.	Res.	Res.	Res.	Res.	SELFREF_TYPE[1:0]		Res.	OPERATING_MODE[2:0]		
			r							r	r		r	r	r

位 31:13 保留，必须保持复位值。

位 12 **SELFREF_CAM_NOT_EMPTY**：在 CAM 不为空时进行自刷新。在已进入自刷新模式但 CAM 尚未清空时，设置为 1。在退出自刷新模式后清零。

编程模式：动态

位 11:6 保留，必须保持复位值。

位 5:4 **SELFREF_TYPE[1:0]**：指示是否已进入自刷新模式（LPDDR4 除外）或 SR 掉电模式 (LPDDR4)，并指示其是否仅受自动自刷新控制。

- 00 - SDRAM 未处于自刷新模式（LPDDR4 除外）或 SR 掉电模式 (LPDDR4)。如果通过 CRCPARCTRL1.crc_parity_retry_enable 使能了重试，则这也表示 SRE 命令仍在奇偶校验错误窗口中，或重试正在进行中。
- 11 - SDRAM 处于自刷新模式（LPDDR4 除外）或 SR 掉电模式 (LPDDR4)，该模式仅可由自动自刷新引发。如果使能了重试，则这可保证 SRE 命令能够正确执行，而不会出现奇偶校验错误。
- 10 - SDRAM 处于自刷新模式（LPDDR4 除外）或 SR 掉电模式 (LPDDR4)，该模式并不仅可由自动自刷新控制引发。该模式可能由硬件低功耗接口和/或软件 (PWRCTL.selfref_sw) 引发。如果使能了重试，则这可保证 SRE 命令能够正确执行，而不会出现奇偶校验错误。
- 01 - SDRAM 处于自刷新模式，该模式由 PHY 主设备请求引发。

编程模式：动态

位 3 保留，必须保持复位值。

位 2:0 **OPERATING_MODE[2:0]**：工作模式。在支持 mDDR/LPDDR2/LPDDR3/LPDDR4/DDR4 的配置中，该模式的宽度为 3 位，而在所有其他配置中均为 2 位。

条件非 mDDR/LPDDR2/LPDDR3/LPDDR4 以及非 DDR4 设计：

- 000 - 初始化
- 001 - 正常
- 010 - 掉电
- 011 - 自刷新

条件 mDDR/LPDDR2/LPDDR3 或 DDR4 设计：

- 000 - 初始化
- 001 - 正常
- 010 - 掉电
- 011 - 自刷新
- 1XX - 深度掉电 / 最大程度节能模式

条件 LPDDR4 设计：

- 000 - 初始化
- 001 - 正常
- 010 - 掉电
- 011 - 自刷新 / 自刷新掉电

编程模式：动态

5.7.3 DDRCTRL 模式寄存器读/写控制寄存器 0 (DDRCTRL_MRCTRL0)

DDRCTRL mode register read/write control register 0

偏移地址：0x010

复位值：0x0000 0010

模式寄存器读/写控制寄存器 0。

注： *请勿同时使能多个以下位字段：*

- *sw_init_int*
- *pda_en*
- *mpr_en*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MR_WR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR_ADDR[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR_TYPE
rw	rw	rw	rw												rw

位 31 **MR_WR**：将此寄存器位设置为 1 将触发模式寄存器的读取或写入操作。MR 操作完成后，DDRCTRL 会自动将该位清零。设置此 mr_wr 位之前，必须在单独的 APB 事务中写入此寄存器的其他寄存器位字段。建议在初始化、深度掉电或 MPSM 工作模式下不要设置此信号。

编程模式：动态

位 30:16 保留，必须保持复位值。

位 15:12 **MR_ADDR[3:0]**：要写入的模式寄存器的地址。
- 0000 - MR0
- 0001 - MR1
- 0010 - MR2
- 0011 - MR3
- 0100 - MR4
- 0101 - MR5
- 0110 - MR6
- 0111 - MR7

该地址与 LPDDR2/LPDDR3/LPDDR4 无关（请参见 MRCTRL1.mr_data，以了解 LPDDR2/LPDDR3/LPDDR4 的模式寄存器地址）

此信号也可用于写入 RDIMM/LRDIMM 上的寄存器芯片的控制字。在这种情况下，该信号与发送到 RDIMM/LRDIMM 的 bank 地址位相对应。

对于 DDR4，位[3:2] 将与 bank 组位相对应。因此，必须将位[3] 以及位[2:0] 设置为适当的值；该值需要同时考虑到 UDIMM/RDIMM/LRDIMM 的地址镜像和 RDIMM/LRDIMM 的输出反转。

编程模式：动态

位 11:5 保留，必须保持复位值。

位 4:1 保留，必须保持复位值。

位 0 **MR_TYPE**：指示模式寄存器操作是读取还是写入。仅用于 LPDDR2/LPDDR3/LPDDR4/DDR4。
- 0 - 写入，MRCTRL 寄存器不支持面向 LPDDR2/3 的 MR 读取
- 1 - 保留

编程模式：动态

5.7.4 DDRCTRL 模式寄存器读/写控制寄存器 1 (DDRCTRL_MRCTRL1)

DDRCTRL mode register read/write control register 1

偏移地址：0x014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR_DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **MR_DATA[15:0]**：面向所有非 LPDDR2/非 LPDDR3/非 LPDDR4 模式的模式寄存器写入数据。对于 LPDDR2/LPDDR3/LPDDR4，可将 MRCTRL1[15:0] 视为 [15:8] MR 地址 [7:0] 写入的 MR 数据（与读取数据无关）。在支持 DDR4 的配置中，该数据的宽度为 18 位，而在所有其他配置中均为 16 位。

编程模式：动态

5.7.5 DDRCTRL 模式寄存器读/写状态寄存器 (DDRCTRL_MRSTAT)

DDRCTRL mode register read/write status register

偏移地址：0x018

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR_WR_BUSY
															r

位 31:1 保留，必须保持复位值。

位 0 **MR_WR_BUSY**：仅当此信号为低电平时，SoC 内核才能启动 MR 写入操作。在 DDRCTRL 接受 MRW/MRR 请求后的那个时钟周期，此信号会变为高电平。当 MRW/MRR 命令发送到 SDRAM 时，该信号会变为低电平。建议在“MRSTAT.mr_wr_busy”为高电平时不要执行 MRW/MRR 命令。

- 0 - 表示 SoC 内核可以启动模式寄存器写入操作
- 1 - 表示正在进行模式寄存器写入操作

编程模式：动态

5.7.6 DDRCTRL 温度降额使能寄存器 (DDRCTRL_DERATEEN)

DDRCTRL temperature derate enable register

偏移地址：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DERATE_BYTE[3:0]				Res.	DERATE_VALUE[1:0]		DERATE_ENABLE
								rw	rw	rw	rw		rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:4 **DERATE_BYTE[3:0]**：对字节降额 (Derate byte)
仅存在于配置为支持 LPDDR2/LPDDR3/LPDDR4 的设计中
指示将 MRR 数据的哪个字节用于降额。最大有效值取决于 MEMC_DRAM_TOTAL_DATA_WIDTH。
编程模式：静态

位 3 保留，必须保持复位值。

位 2:1 **DERATE_VALUE[1:0]**：对值降额 (Derate value)
- 0 - 降额使用 +1。
- 1 - 降额使用 +2。
仅存在于配置为支持 LPDDR2/LPDDR3/LPDDR4 的设计中
针对所有 LPDDR2 速度等级设置为 0，原因是降额值 +1.875 ns 小于一个 core_ddrc_core_clk 周期。
对于 LPDDR3/4，如果 core_ddrc_core_clk 的周期小于 1.875 ns，则应将此寄存器位字段设置为 1，否则应将其设置为 0。
编程模式：准动态组 2 和组 4

位 0 **DERATE_ENABLE**：使能降额 (Enables derating)
- 0 - 禁止时序参数降额
- 1 - 使用 MR4 读取值使能时序参数降额。
仅存在于配置为支持 LPDDR2/LPDDR3/LPDDR4 的设计中
对于非 LPDDR2/LPDDR3/LPDDR4 模式，此位字段必须设置为“0”。
编程模式：动态

5.7.7 DDRCTRL 温度降额间隔寄存器 (DDRCTRL_DERATEINT)

DDRCTRL temperature derate interval register

偏移地址：0x024

复位值：0x0080 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MR4_READ_INTERVAL[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR4_READ_INTERVAL[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **MR4_READ_INTERVAL[31:0]**：两次 MR4 读取之间的间隔，用于对时序参数进行降额。
仅存在于配置为支持 LPDDR2/LPDDR3/LPDDR4 的设计中。此寄存器不能设置为零。
单位：DFI 时钟周期。
编程模式：静态

5.7.8 DDRCTRL 低功耗控制寄存器 (DDRCTRL_PWRCTL)

DDRCTRL low power control register

偏移地址：0x030

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIS_CAM_DRAIN_SELFREF	Res.	SELFREF_SW	Res.	EN_DFI_DRAM_CLK_DISABLE	DEEPPOWERDOWN_EN	POWERDOWN_EN	SELFREF_EN
								rw		rw		rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7 **DIS_CAM_DRAIN_SELFREF**：指示在进入自刷新模式时是否允许跳过 CAM 清空。
在 PWRCTL.selfref_sw==1 时，不能修改此寄存器位字段。
- 0 - 进入 SR 模式前 CAM 必须为空
- 1 - 进入 SR 模式前不清空 CAM（不支持）

请注意，此版本不支持 PWRCTL.dis_cam_drain_selfref=1。
必须设置 PWRCTL.dis_cam_drain_selfref=0。
编程模式：动态

位 6 保留，必须保持复位值。

位 5 **SELFREF_SW**：此寄存器的值为 1 会立即导致系统变为自刷新状态，前提是系统不处于 INIT 或 DPD/MPSM operating_mode。可将其称为软件进入/退出自刷新。
- 1 - 软件进入自刷新
- 0 - 软件退出自刷新

编程模式：动态

位 4 保留，必须保持复位值。

位 3 **EN_DFI_DRAM_CLK_DISABLE**：当 SDRAM 不需要某时钟时，将 dfi_dram_clk_disable 置为有效。

如果设置为 0，则 dfi_dram_clk_disable 不会生效。

dfi_dram_clk_disable 的生效情况如下所示：

在 DDR2/DDR3 中，只能在自刷新模式下生效。

在 DDR4 中，可在以下情况下生效：

- 在自刷新模式下。
- 在最大程度节能模式下

在 mDDR/LPDDR2/LPDDR3 中，可在以下情况下生效：

- 在自刷新模式下
- 在掉电模式下
- 在深度掉电模式下
- 在正常操作期间（时钟停止）

在 LPDDR4 中，可在以下情况下生效：

- 在自刷新掉电模式下
- 在掉电模式下
- 在正常操作期间（时钟停止）

编程模式：动态

位 2 **DEEPPOWERDOWN_EN**：当此值为 1 时，DDRCTRL 会在事务存储为空时将 SDRAM 置于深度掉电模式。

必须将此寄存器复位为“0”，方可使 DDRCTRL 退出深度掉电模式。控制器会在退出深度掉电模式时自动进行 SDRAM 初始化。

仅存在于配置为支持 mDDR、LPDDR2 或 LPDDR3 的设计中。对于非 mDDR/ 非 LPDDR2/ 非 LPDDR3，不应将此寄存器设置为 1。

仅限性能用途

编程模式：动态

位 1 **POWERDOWN_EN**：如果该位为真，则 DDRCTRL 将在经过可编程周期数“掉电前的最大空闲时钟周期数”(PWRTMG.powerdown_to_x32) 之后进入掉电模式。

在正常操作期间，可能会对该寄存器位进行重新编程。

编程模式：动态

位 0 **SELFREF_EN**：如果该位为真，则 DDRCTRL 将在经过可编程周期数“自刷新前的最大空闲时钟周期数 (PWRTMG.selfref_to_x32)”之后，将 SDRAM 置于自刷新模式。在正常操作期间，可能会对该寄存器位进行重新编程。

编程模式：动态

5.7.9 DDRCTRL 低功耗时序寄存器 (DDRCTRL_PWRTMG)

DDRCTRL low power timing register

偏移地址：0x034

复位值：0x0040 2010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELFREF_TO_X32[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T_DPD_X4096[7:0]								Res.	Res.	Res.	POWERDOWN_TO_X32[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **SELFREF_TO_X32[7:0]**：在 DDRC 命令通道处于空闲状态若干时钟周期数后，DDRCTRL 会自动将 SDRAM 置于自刷新模式。一旦没有未完成的 HIF 命令，即可认为 DDRC 命令通道处于空闲状态。必须在 PWRCTL.selfref_en 中使能此位。

单位：32 个 DFI 时钟周期的倍数。

仅限性能用途。

编程模式：准动态组 4

位 15:8 **T_DPD_X4096[7:0]**：最短深度掉电时间。

对于 mDDR，JEDEC 规范中的值为 0，原因是在 PWRCTL.deeppowerdown_en 清零后，mDDR 会立即退出深度掉电模式。

对于 LPDDR2/LPDDR3，JEDEC 规范中的值为 500 us。

单位：4096 个 DFI 时钟周期的倍数。

仅存在于配置为支持 mDDR、LPDDR2 或 LPDDR3 的设计中。

仅限性能用途。

编程模式：准动态组 4

位 7:5 保留，必须保持复位值。

位 4:0 **POWERDOWN_TO_X32[4:0]**：在 DDRC 命令通道处于空闲状态若干时钟周期数后，DDRCTRL 会自动将 SDRAM 置于掉电模式。一旦没有未完成的 HIF 命令，即可认为 DDRC 命令通道处于空闲状态。必须在 PWRCTL.powerdown_en 中使能此位。

单位：32 个 DFI 时钟周期的倍数

仅限性能用途。

编程模式：准动态组 4

5.7.10 DDRCTRL 硬件低功耗控制寄存器 (DDRCTRL_HWLPCTL)

DDRCTRL hardware low power control register

偏移地址：0x038

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	HW_LP_IDLE_X32[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HW_LP_EXIT_IDLE_EN	HW_LP_EN
														rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **HW_LP_IDLE_X32[11:0]**：硬件空闲周期。如果 DDRC 命令通道在 hw_lp_idle * 32 个周期内处于空闲状态，且系统未处于 INIT 或 DPD/MPSM operating_mode，系统会将 cactive_ddrc 输出驱动为低电平。一旦没有未完成的 HIF 命令，即可认为 DDRC 命令通道处于空闲状态。当 hw_lp_idle_x32=0 时，将禁止硬件空闲功能。

单位：32 个 DFI 时钟周期的倍数。

仅限性能用途。

编程模式：静态

位 15:2 保留，必须保持复位值。

位 1 **HW_LP_EXIT_IDLE_EN**：当此位设置为 1 时，可以使用 DDRC 的 cactive_in_ddrc 引脚来退出时钟自动停止、自动掉电或自动自刷新模式。请注意，此位不会导致系统退出由硬件低功耗接口和/或软件 (PWRCTL.selfref_sw) 引发的自刷新模式。

编程模式：静态

位 0 **HW_LP_EN**：使能硬件低功耗接口。

编程模式：准动态组 3

5.7.11 DDRCTRL 刷新控制寄存器 0 (DDRCTRL_RFSHCTL0)

DDRCTRL refresh control register 0

偏移地址：0x050

复位值：0x0021 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REFRESH_MARGIN[3:0]				Res.	Res.	Res.	REFRESH_TO_X32[4]
								rw	rw	rw	rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REFRESH_TO_X32[3:0]				Res.	Res.	Res.	REFRESH_BURST[4:0]					Res.	PER_BANK_REFRESH	Res.	Res.
rw	rw	rw	rw				rw	rw	rw	rw	rw		rw		

位 31:24 保留，必须保持复位值。

位 23:20 **REFRESH_MARGIN[3:0]**：关键刷新或页定时器到期前，以 DFI 时钟周期数表示的阈值。必须在达到该阈值之前发出关键刷新。建议不要更改该阈值的默认值；当前该值显示为 0x2。该值必须始终小于内部使用的 t_rfc_nom/32。请注意，如果 RFSHTMG.t_rfc_nom_x1_sel=0，则内部使用的 t_rfc_nom 将等于 RFSHTMG.t_rfc_nom_x1_x32 * 32。如果 RFSHTMG.t_rfc_nom_x1_sel=1（仅限 LPDDR2/LPDDR3/LPDDR4 逐 bank 刷新），则内部使用的 t_rfc_nom 将等于 RFSHTMG.t_rfc_nom_x1_x32。请注意，在 LPDDR2/LPDDR3/LPDDR4 中，如果使能了降额 (DERATEEN.derate_enable=1)，则内部使用的 t_rfc_nom 可能需要除以 4。

单位：32 个 DFI 时钟周期的倍数。

编程模式：动态 - 刷新相关

位 19:17 保留，必须保持复位值。

位 16:12 **REFRESH_TO_X32[4:0]**：如果刷新定时器（tRFCnom，也称为 tREFI）至少到期了一次，但尚未到期 (RFSHCTL0.refresh_burst+1) 次，则可以执行推测性刷新。推测性刷新指的是在刷新有作用，但尚未成为绝对必要事项的时候执行的刷新。如果 SDRAM 总线在一段时间内处于空闲状态（具体时长由 RFSHCTL0.refresh_to_x32 决定），且自上次刷新以来刷新定时器至少已到期了一次，则系统将执行推测性刷新。推测性刷新将持续进行，直到不存在挂起的刷新，或直到系统向 DDRCTRL 发出新的读取或写入请求为止。

仅限性能用途。

单位：32 个 DFI 时钟周期的倍数。

编程模式：动态 - 刷新相关

位 11:9 保留，必须保持复位值。

位 8:4 **REFRESH_BURST[4:0]**：编程值 + 1 指的是在系统阻塞数据传输并强制执行刷新之前，允许累积的刷新超时次数。执行刷新时需要将页关闭，这是执行每组刷新时都必须承担的一次性代价。因此，以突发方式执行刷新可以有效减少每次刷新时关闭页的代价。较高的 RFSHCTL.refresh_burst 数值能够略微提升利用率；较低的数值则能够减少与刷新相关的巨大延迟。

- 0 - 单次刷新
- 1 - 2 次突发刷新
- 7 - 8 次突发刷新

有关突发刷新功能的信息，请参见 DDR2 JEDEC 规范 - JESD79-2F.pdf 中的第 3.9 节。

对于 DDR2/3，刷新始终会逐 rank 而非逐 bank 进行。使用突发刷新功能，用户可以在 8*tREFI 周期内累积 rank 刷新。

在 DDR4 模式下，用户可以依据精细粒度特性，在 1X 模式下延迟 8 次刷新，在 2X 模式下延迟 16 次刷新，并在 4X 模式下延迟 32 次刷新。如果要使用 PHY 发起的更新，则在设置 RFSHCTL0.refresh_burst 时必须谨慎，以确保不会因为在刷新突发即将到期时执行 PHY 发起的更新而违反 tRFCmax。一旦出现这种情况，系统将延迟刷新突发，直到 PHY 发起的更新完成为止。

编程模式：动态 - 刷新相关

位 3 保留，必须保持复位值。

位 2 **PER_BANK_REFRESH**：

- 1 - 逐 bank 刷新；
- 0 - 全 bank 刷新。

逐 bank 刷新允许传输数据流向其他 bank。并非所有 LPDDR2 设备都支持逐 bank 刷新，但所有 LPDDR3/LPDDR4 设备应该都支持该刷新方式。仅存在于配置为支持 LPDDR2/LPDDR3/LPDDR4 的设计中

编程模式：静态

位 1:0 保留，必须保持复位值。

5.7.12 DDRCTRL 刷新控制寄存器 3 (DDRCTRL_RFSHCTL3)

DDRCTRL refresh control register 3

偏移地址：0x060

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REFRESH_UPDATE_LEVEL	DIS_AUTO_REFRESH
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **REFRESH_UPDATE_LEVEL**：翻转此信号（从 0 翻转到 1 或从 1 翻转到 0），以指示刷新寄存器已更新。

当 DDRC 处于复位状态时 (core_ddrc_rstn = 0)，不得翻转 refresh_update_level。

退出复位时，刷新寄存器将自动更新。

编程模式：动态

位 0 **DIS_AUTO_REFRESH**：当设置为“1”时，将禁止由 DDRCTRL 生成的自动刷新。当禁止自动刷新时，SoC 内核必须使用寄存器 DBGCMD.rankn_refresh 生成刷新。

当 dis_auto_refresh 从 0 转换到 1 时，任何挂起的刷新都将立即由 DDRCTRL 进行调度。

如果使能了 DDR4 CRC/奇偶校验重试 (CRCPARCTL1.crc_parity_retry_enable = 1)，则系统将不支持禁止自动刷新，并且必须将此位设置为“0”。

（仅限 DDR4）如果使能了 FGR 模式 (RFSHCTL3.refresh_mode > 0)，则系统将不支持禁止自动刷新，并且必须将此位设置为“0”。

此寄存器位字段可实时更改。

编程模式：动态 - 刷新相关

5.7.13 DDRCTRL 刷新时序寄存器 (DDRCTRL_RFSHTMG)

DDRCTRL refresh timing register

偏移地址：0x064

复位值：0x0062 008C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T_RFC_NOM_X1_SEL	Res.	Res.	Res.	T_RFC_NOM_X1_X32[11:0]											
rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPDDR3_TREFBW_EN	Res.	Res.	Res.	Res.	Res.	T_RFC_MIN[9:0]									
rw						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **T_RFC_NOM_X1_SEL**：指定 t_rfc_nom_x1_x32 寄存器值是 x1 还是 x32。
编程模式：动态 - 刷新相关

位 30:28 保留，必须保持复位值。

位 27:16 **T_RFC_NOM_X1_X32[11:0]**：tREFI：每个 rank 的平均刷新时间间隔（规范：DDR2、DDR3 和 DDR4 为 7.8 us。请参见 mDDR、LPDDR2、LPDDR3 和 LPDDR4 的 JEDEC 规范）。
对于 LPDDR2/LPDDR3/LPDDR4：
- 如果使用全 bank 刷新 (RFSHCTL0.per_bank_refresh = 0)，则应将此寄存器设置为 tREFIab
- 如果使用逐 bank 刷新 (RFSHCTL0.per_bank_refresh = 1)，则应将此寄存器设置为 tREFIpb

当控制器在 1:2 频率比模式下运行时，请将其设置为 (tREFI/2)，且不进行向上舍入。
在 DDR4 模式下，tREFI 值会根据具体的刷新模式而有所出入。用户应根据刷新模式寄存器中编程的值，对规范中的适当值进行编程。
请注意，如果 RFSHTMG.t_rfc_nom_x1_sel == 1，则 RFSHTMG.t_rfc_nom_x1_x32 必须大于 RFSHTMG.t_rfc_min；如果 RFSHTMG.t_rfc_nom_x1_sel == 0，则 RFSHTMG.t_rfc_nom_x1_x32 * 32 必须大于 RFSHTMG.t_rfc_min；RFSHTMG.t_rfc_nom_x1_x32 必须大于 0x1。
- 非 DDR4 或 DDR4 固定 1x 模式：RFSHTMG.t_rfc_nom_x1_x32 必须小于或等于 0xFFE。
- DDR4 固定 2x 模式：RFSHTMG.t_rfc_nom_x1_x32 必须小于或等于 0x7FF。
- DDR4 固定 4x 模式：RFSHTMG.t_rfc_nom_x1_x32 必须小于或等于 0x3FF。

单位：时钟周期数或 32 个时钟周期的倍数，具体取决于 RFSHTMG.t_rfc_nom_x1_sel。
编程模式：动态 - 刷新相关

位 15 **LPDDR3_TREFBW_EN**：仅在连接了 LPDDR3 内存类型时使用。仅在 DDRCTRL 处于复位状态时才应更改。指定是否使用 tREFBW 参数（以满足部分符合较早版本 LPDDR3 JEDEC 规范要求的 LPDDR3 设备所需）：

- 0 - 不使用 tREFBW 参数
- 1 - 使用 tREFBW 参数

编程模式：静态

位 14:10 保留，必须保持复位值。

位 9:0 **T_RFC_MIN[9:0]**：tRFC(min)：从刷新到刷新或激活的最短时间。

当控制器在 1:1 模式下运行时，应将 t_rfc_min 设置为 RoundUp(tRFCmin/tCK)。

当控制器在 1:2 模式下运行时，应将 t_rfc_min 设置为 RoundUp(RoundUp(tRFCmin/tCK)/2)。

在 LPDDR2/LPDDR3/LPDDR4 模式下：

- 如果使用全 bank 刷新，上述公式中的 tRFCmin 值将等于 tRFCab
- 如果使用逐 bank 刷新，上述公式中的 tRFCmin 值将等于 tRFCpb

在 DDR4 模式下，上述公式中的 tRFCmin 值将根据具体的刷新模式（固定 1X、2X、4X）和设备密度而有所出入。用户应根据所使用的“refresh_mode”和设备密度，对规范中的适当值进行编程。

单位：时钟。

编程模式：动态 - 刷新相关

5.7.14 DDRCTRL CRC 奇偶校验控制寄存器 0 (DDRCTRL_CRCPARCTL0)

DDRCTRL CRC parity control register 0

偏移地址：0x0C0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFI_ALERT_ERR_CNT_CLR	DFI_ALERT_ERR_INT_CLR	DFI_ALERT_ERR_INT_EN
													rc_w1	rc_w1	rw

位 31:3 保留，必须保持复位值。

位 2 **DFI_ALERT_ERR_CNT_CLR**：DFI 报警错误计数清零。DFI 报警错误计数器的清零位。将此位置为有效会将 DFI 报警错误计数器 CRCPARSTAT.dfi_alert_err_cnt 清零。清零操作完成后，DDRCTRL 会自动将该位清零。

编程模式：动态

位 1 **DFI_ALERT_ERR_INT_CLR**：DFI 报警错误的中断清零位。如果将该位置 1，则 CRCPARSTAT.dfi_alert_err_int 上的报警错误中断将清零。清零操作完成后，DDRCTRL 会自动将该位清零。

编程模式：动态

位 0 **DFI_ALERT_ERR_INT_EN**：DFI 报警错误的中断使能位。如果将该位置 1，则在 dfi_alert_n 输入上检测到的任何奇偶校验/CRC 错误都会导致系统在 CRCPARSTAT.dfi_alert_err_int 上设置中断。

编程模式：动态

5.7.15 DDRCTRL CRC 奇偶校验状态寄存器 (DDRCTRL_CRCPARSTAT)

DDRCTRL CRC parity status register

偏移地址：0x0CC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFI_ALERT_ERR_INT
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFI_ALERT_ERR_CNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:17 保留，必须保持复位值。

位 16 **DFI_ALERT_ERR_INT**：DFI 报警错误中断。

如果在 dfi_alert_n 上检测到奇偶校验/CRC 错误，并且已由 CRCPARCTL0.dfi_alert_err_int_en 使能了中断，则系统将设置此中断位。该位将保持在已设置状态，直到由 CRCPARCTL0.dfi_alert_err_int_clr 将其清零为止

编程模式：静态

位 15:0 **DFI_ALERT_ERR_CNT[15:0]**：DFI 报警错误计数。

如果在 dfi_alert_n 上检测到奇偶校验/CRC 错误，则此计数器将递增。这与 CRCPARCTL0.dfi_alert_err_int_en 的设置无关。此位将在 0xFFFF 下饱和，且可以通过使 CRCPARCTL0.dfi_alert_err_cnt_clr 生效来进行清零。

编程模式：静态

5.7.16 DDRCTRL SDRAM 初始化寄存器 0 (DDRCTRL_INIT0)

DDRCTRL SDRAM initialization register 0

偏移地址：0x0D0

复位值：0x0002 004E

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SKIP_DRAM_INIT[1:0]		Res.	Res.	Res.	Res.	POST_CKE_X1024[9:0]									
rw	rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PRE_CKE_X1024[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 **SKIP_DRAM_INIT[1:0]**：如果使能了低位，则系统将跳过 SDRAM 初始化程序。高位将决定控制器在移除复位时的启动状态

- 00 - 上电后运行 SDRAM 初始化程序
- 01 - 上电后跳过 SDRAM 初始化程序。控制器在正常模式下启动
- 11 - 上电后跳过 SDRAM 初始化程序。控制器在自刷新模式下启动
- 10 - 保留。

编程模式：准动态组 2

位 29:26 保留，必须保持复位值。

位 25:16 **POST_CKE_X1024[9:0]**：为了启动 SDRAM 初始化序列，将 CKE 驱动为高电平后需要等待的周期数。

单位：1024 个 DFI 时钟周期。

DDR2 通常需要 400 ns 的延迟，因此需要在所有时钟速度下将此值设置为 2。

LPDDR2/LPDDR3 通常需要对此值进行编程，以实现 200 us 的延迟。

LPDDR4 通常需要对此值进行编程，以实现 2 us 的延迟。

当控制器在 1:2 频率比模式下运行时，请将此参数设置为用 JEDEC 规范值来除以 2，并将结果向上舍入到下一个整数值。

编程模式：静态

位 15:12 保留，必须保持复位值。

位 11:0 **PRE_CKE_X1024[11:0]**：为了启动 SDRAM 初始化序列，复位后将 CKE 驱动为高电平之前需要等待的周期数。

单位：1024 个 DFI 时钟周期。

DDR2 规范通常要求对此值进行编程，以实现 >= 200 us 的延迟。

LPDDR2/LPDDR3：tINIT1 为 100 ns（最小值）

LPDDR4：tINIT3 为 2 ms（最小值）

当控制器在 1:2 频率比模式下运行时，请将此参数设置为用 JEDEC 规范值来除以 2，并将结果向上舍入到下一个整数值。

对于 DDR3/DDR4 RDIMM，此值应包含满足 tSTAB 所需的时间

编程模式：静态

5.7.17 DDRCTRL SDRAM 初始化寄存器 1 (DDRCTRL_INIT1)

DDRCTRL SDRAM initialization register 1

偏移地址：0x0D4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DRAM_RSTN_X1024[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRE_OCD_X32[3:0]			
												rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24:16 **DRAM_RSTN_X1024[8:0]**：初始化序列期间，将 SDRAM 复位信号置为有效所需的周期数。
此位仅存在于支持 DDR3、DDR4 或 LPDDR4 设备的设计中。如需配合 DDR PHY 使用，至少应将此位设置为 1。
当控制器在 1:2 频率比模式下运行时，请将此参数设置为用 JEDEC 规范值来除以 2，并将结果向上舍入到下一个整数值。
单位：1024 个 DFI 时钟周期。
编程模式：静态

位 15:4 保留，必须保持复位值。

位 3:0 **PRE_OCD_X32[3:0]**：将 OCD 完成命令驱动至 SDRAM 之前的等待周期。
单位：全局定时器（每隔 32 个 DFI 时钟周期脉冲一次）的计数。
目前并无已知相关具体要求；可将其设置为零。
编程模式：静态

5.7.18 DDRCTRL SDRAM 初始化寄存器 2 (DDRCTRL_INIT2)

DDRCTRL SDRAM initialization register 2

偏移地址：0x0D8

复位值：0x0000 0D05

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDLE_AFTER_RESET_X32[7:0]								Res.	Res.	Res.	Res.	MIN_STABLE_CLOCK_X1[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:8 **IDLE_AFTER_RESET_X32[7:0]**：复位命令后的空闲时间，tINIT4。仅存在于配置为支持 LPDDR2 的设计中。

当控制器在 1:2 频率比模式下运行时，请将此参数设置为用 JEDEC 规范值来除以 2，并将结果向上舍入到下一个整数值。

单位：32 个 DFI 时钟周期。

编程模式：静态

位 7:4 保留，必须保持复位值。

位 3:0 **MIN_STABLE_CLOCK_X1[3:0]**：第一次将 CKE 置于高电平后的等待时间，tINIT2。仅存在于配置为支持 LPDDR2/LPDDR3 的设计中。

LPDDR2/LPDDR3 通常需要 5 x tCK 的延迟。

当控制器在 1:2 频率比模式下运行时，请将此参数设置为用 JEDEC 规范值来除以 2，并将结果向上舍入到下一个整数值。

单位：DFI 时钟周期。

编程模式：静态

5.7.19 DDRCTRL SDRAM 初始化寄存器 3 (DDRCTRL_INIT3)

DDRCTRL SDRAM initialization register 3

偏移地址：0x0DC

复位值：0x0000 0510

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **MR[15:0]**：DDR2：要写入 MR 寄存器的值。位 8 适用于 DLL，此处的设置将被忽略。DDRCTRL 会对此位进行适当设置。

DDR3/DDR4：加载到 MR0 寄存器中的值。

mDDR：要写入 MR 寄存器的值。

LPDDR2/LPDDR3/LPDDR4 - 要写入 MR1 寄存器的值

编程模式：准动态组 1 和组 4

位 15:0 **EMR[15:0]**：DDR2：要写入 EMR 寄存器的值。位 9:7 适用于 OCD，此寄存器中的设置将被忽略。DDRCTRL 会对这些位进行适当设置。

DDR3/DDR4：要写入 MR1 寄存器的值。将位 7 设置为 0。如果使能了 PHY 评估模式训练，则 DDRCTRL 会在写入均衡期间对此位进行适当设置。

mDDR：要写入 EMR 寄存器的值。

LPDDR2/LPDDR3/LPDDR4 - 要写入 MR2 寄存器的值

编程模式：准动态组 4

5.7.20 DDRCTRL SDRAM 初始化寄存器 4 (DDRCTRL_INIT4)

DDRCTRL SDRAM initialization register 4

偏移地址：0x0E0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EMR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **EMR2[15:0]**：DDR2：要写入 EMR2 寄存器的值。
DDR3/DDR4：要写入 MR2 寄存器的值
LPDDR2/LPDDR3/LPDDR4：要写入 MR3 寄存器的值
mDDR：未使用
编程模式：准动态组 4

位 15:0 **EMR3[15:0]**：DDR2：要写入 EMR3 寄存器的值。
DDR3/DDR4：要写入 MR3 寄存器的值
mDDR/LPDDR2/LPDDR3：未使用
LPDDR4：要写入 MR13 寄存器的值
编程模式：准动态组 2 和组 4

5.7.21 DDRCTRL SDRAM 初始化寄存器 5 (DDRCTRL_INIT5)

DDRCTRL SDRAM initialization register 5

偏移地址：0x0E4

复位值：0x0010 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEV_ZQINIT_X32[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	MAX_AUTO_INIT_X1024[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **DEV_ZQINIT_X32[7:0]**：ZQ 初始校准，tZQINIT。仅存在于配置为支持 DDR3、DDR4 或 LPDDR2/LPDDR3 的设计中。
DDR3 通常需要 512 个 SDRAM 时钟周期。
DDR4 需要 1024 个 SDRAM 时钟周期。
LPDDR2/LPDDR3 需要 1 us。
当控制器在 1:2 频率比模式下运行时，请将此参数设置为用 JEDEC 规范值来除以 2，并将结果向上舍入到下一个整数值。
单位：32 个 DFI 时钟周期。
编程模式：静态

位 15:10 保留，必须保持复位值。

位 9:0 **MAX_AUTO_INIT_X1024[9:0]**：自动初始化的最长持续时间，tINIT5。仅存在于配置为支持 LPDDR2/LPDDR3 的设计中。

LPDDR2/LPDDR3 通常需要 10 us。

单位：1024 个 DFI 时钟周期。

编程模式：静态

5.7.22 DDRCTRL DIMM 控制寄存器 (DDRCTRL_DIMMCTL)

DDRCTRL DIMM control register

偏移地址：0x0F0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIMM_ADDR_MIRR_EN	DIMM_STAGGER_CS_EN
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **DIMM_ADDR_MIRR_EN**：地址镜像使能（适用于多 rank UDIMM 实现和多 rank DDR4 RDIMM/LRDIMM 实现）。

一些 UDIMM 和 DDR4 RDIMM/LRDIMM 将为奇数 rank 实现地址镜像，这意味着系统将交换以下地址、bank 地址和 bank 组位：(A3, A4)、(A5, A6)、(A7, A8)、(BA0, BA1) 以及适用于 DDR4 的 (A11, A13) 和 (BG0, BG1)。设置此位可确保在自动初始化程序期间执行模式寄存器访问时，系统将在 DDRCTRL 内部交换这些位，以补偿此 UDIMM/RDIMM/LRDIMM 交换。除了自动初始化程序之外，对于 DDR4 UDIMM/RDIMM/LRDIMM，系统将在自动 MRS 访问期间交换这些位，以使能/禁止特定的 DDR4 功能。

注意：此位对任何其他内存访问的地址，或是由软件驱动的模式寄存器访问的地址没有影响。

此位不适用于 mDDR、LPDDR2、LPDDR3 或 LPDDR4 SDRAM。

注意：对于 x16 DDR4 DIMM，奇数 rank MRS 的 BG1 输出与 BG0 相同，原因是 BG1 无效，所以必须将 dimm_dis_bg_mirroring 寄存器设置为 1。

- 1 - 对于奇数 rank，请在初始化期间针对任何自动 DDR4 MRS 命令实现 MRS 命令的地址镜像（在 UDIMM/RDIMM/LRDIMM 实现了地址镜像的情况下使用）

- 0 - 不实现地址镜像

编程模式：静态

位 0 **DIMM_STAGGER_CS_EN**：交错使能多 rank 访问（仅适用于多 rank UDIMM、RDIMM 和 LRDIMM 实现）。此位不适用于 mDDR、LPDDR2、LPDDR3 或 LPDDR4 SDRAM。

注意：即使将此位置 1，系统也不会处理由软件（通过 MRCTRL0/MRCTRL1）驱动的 MR 命令，而是由软件负责视情况将其发送到单独的 rank。

- 1 - (DDR4) 将 MRS 命令分别发送到每个 rank
- 1 -（非 DDR4）将所有命令分别发送到偶数 rank 和奇数 rank
- 0 - 不进行交错访问

编程模式：静态。

5.7.23 DDRCTRL SDRAM 时序寄存器 0 (DDRCTRL_DRAMTMG0)

DDRCTRL SDRAM timing register 0

偏移地址：0x100

复位值：0x0F10 1B0F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	WR2PRE[6:0]							Res.	Res.	T_FAW[5:0]					
	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	T_RAS_MAX[6:0]							Res.	Res.	T_RAS_MIN[5:0]					
	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:24 **WR2PRE[6:0]**：

写入 bank 与对同一 bank 进行预充电之间的最短时间。

单位：时钟

规格：WL + BL/2 + tWR = 大约 8 个周期 + 15 ns = 以 400 MHz 的频率运行 14 个时钟周期（频率越低，时钟周期数越少）

其中：

- WL = 写入延迟
- BL = 突发长度。此参数必须与 SDRAM 模式寄存器的 BL 位中的编程值相匹配。目前不支持 BST（突发终止）。
- tWR = 写恢复时间。此参数直接来自 SDRAM 规范。

对于此参数，请为 LPDDR2/LPDDR3/LPDDR4 增加一个额外的周期。

当控制器在 1:2 频率比模式或 1T 模式下运行时，请将上述值除以 2。不进行向上舍入。

当控制器在 1:2 频率比模式、2T 模式或 LPDDR4 模式下运行时，请将上述值除以 2，并将结果向上舍入到下一个整数值。

请注意，如果使用 LRDIMM，则可能需要调整此参数的值，以通过 LRDIMM 补偿额外的延迟周期，具体取决于 PHY 的情况。

编程模式：准动态组 1、组 2 和组 4。

位 23:22 保留，必须保持复位值。

位 21:16 **T_FAW[5:0]**：只有当存在 8 个或更多 bank（或 bank x bank 组）时，tFAW 才有效。
在 8 bank 设计中，系统必须在 tFAW 周期的滚动窗口中激活最多 4 个 bank。
当控制器在 1:2 频率比模式下运行时，请将此参数设置为 (tFAW/2)，并向上舍入到下一个整数值。
在 4 bank 设计中，请将此寄存器设置为 0x1；此项设置与 1:1/1:2 频率模式无关。
单位：时钟
编程模式：准动态组 2 和组 4。

位 15 保留，必须保持复位值。

位 14:8 **T_RAS_MAX[6:0]**：tRAS(max)：激活 bank 与对同一 bank 进行预充电之间的最长时间。此参数是页可以保持开启状态的最长时间
此寄存器的最小值为 1。零为无效值。
当控制器在 1:2 频率比模式下运行时，请将其设置为 (tRAS(max)-1)/2。不进行向上舍入。
单位：1024 个时钟周期的倍数。
编程模式：准动态组 2 和组 4

位 7:6 保留，必须保持复位值。

位 5:0 **T_RAS_MIN[5:0]**：tRAS(min)：激活 bank 与对同一 bank 进行预充电之间的最短时间。
当控制器在 1:2 频率模式或 1T 模式下运行时，请将其设置为 tRAS(min)/2。不进行向上舍入。
当控制器在 1:2 频率比模式、2T 模式或 LPDDR4 模式下运行时，请将其设置为 (tRAS(min)/2)，并将结果向上舍入到下一个整数值。
单位：时钟。
编程模式：准动态组 2 和组 4。

5.7.24 DDRCTRL SDRAM 时序寄存器 1 (DDRCTRL_DRAMTMG1)

DDRCTRL SDRAM timing register 1

偏移地址：0x104

复位值：0x0008 0414

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T_XP[4:0]				
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	RD2PRE[5:0]						Res.	T_RC[6:0]						
		rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值。

位 20:16 **T_XP[4:0]**：tXP：从退出掉电模式到执行任何操作的最短时间。对于 DDR3，如果在 MR0[12] 中选择了退出慢速掉电模式，则应将其设置为 tXPDLL。
如果使用面向 DDR4 的 C/A 奇偶校验，则将其设置为 (tXP+PL)。
如果选择了 LPDDR4 且其规范中含有 tCKELPD 参数，则将其设置为 tXP 与 tCKELPD 中的较大值。
当控制器在 1:2 频率比模式下运行时，请将此参数设置为 (tXP/2)，并将结果向上舍入到下一个整数值。
单位：时钟。
编程模式：准动态组 2 和组 4。

位 15:14 保留，必须保持复位值。

位 13:8 **RD2PRE[5:0]**：tRTP：从读取 bank 到对同一 bank 进行预充电的最短时间。

- DDR2：tAL + BL/2 + max(tRTP,2) - 2
- DDR3：tAL + max(tRTP,4)
- DDR4：以下两个公式的最大值：
  tAL + max(tRTP,4) 或
  RL + BL/2 - tRP (*)。
- mDDR：BL/2
- LPDDR2：具体取决于是 LPDDR2-S2 还是 LPDDR2-S4：
  LPDDR2-S2：BL/2 + tRTP - 1。
  LPDDR2-S4：BL/2 + max(tRTP,2) - 2。
- LPDDR3：BL/2 + max(tRTP,4) - 4
- LPDDR4：BL/2 + max(tRTP,8) - 8

(*) 当同时使用 DDR4 SDRAM 和 ST-MRAM 时，请使用 SDRAM 的 tRP 值进行计算。

当控制器在 1:2 模式或 1T 模式下运行时，请将上述值除以 2。不进行向上舍入。

当控制器在 1:2 模式、2T 模式或 LPDDR4 模式下运行时，请将上述值除以 2，并将结果向上舍入到下一个整数值。

单位：时钟。

编程模式：准动态组 1、组 2 和 组 4

位 7 保留，必须保持复位值。

位 6:0 **T_RC[6:0]**：tRC：两次激活同一 bank 之间的最短时间。

当控制器在 1:2 频率比模式下运行时，请将此参数设置为 (tRC/2)，并向上舍入到下一个整数值。

单位：时钟。

编程模式：准动态组 2 和组 4。

5.7.25 DDRCTRL SDRAM 时序寄存器 2 (DDRCTRL_DRAMTMG2)

DDRCTRL SDRAM timing register 2

偏移地址：0x108

复位值：0x0305 060D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	WRITE_LATENCY[5:0]						Res.	Res.	READ_LATENCY[5:0]					
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	RD2WR[5:0]						Res.	Res.	WR2RD[5:0]					
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29:24 **WRITE_LATENCY[5:0]**：设置为 WL

从写入命令到通过 SDRAM 接口写入数据的时间。必须将其设置为 WL。

对于 mDDR，通常应将其设置为 1。

请注意，如果使用 RDIMM/LRDIMM，则可能需要调整 WL 的值，以通过 RDIMM/LRDIMM 补偿额外的延迟周期，具体取决于 PHY 的情况。

当控制器在 1:2 频率比模式下运行时，请将使用上述公式计算得出的值除以 2，并将结果向上舍入到下一个整数值。

对 DDR2 和 DDR3 而言，此寄存器位字段并非必需位字段（已设置 MEMC_TRAINING 的情况除外），原因是在 DFITMG0 和 DFITMG1 中定义的 DFI 读写延迟已足够满足这些协议的需求。

单位：时钟周期。

编程模式：准动态组 1、组 2 和组 4。

位 23:22 保留，必须保持复位值。

位 21:16 **READ_LATENCY[5:0]**：设置为 RL

从读取命令到通过 SDRAM 接口读取数据的时间。必须将其设置为 RL。

请注意，如果使用 RDIMM/LRDIMM，则可能需要调整 RL 的值，以通过 RDIMM/LRDIMM 补偿额外的延迟周期，具体取决于 PHY 的情况。

当控制器在 1:2 频率比模式下运行时，请将使用上述公式计算得出的值除以 2，并将结果向上舍入到下一个整数值。

对 DDR2 和 DDR3 而言，此寄存器位字段并非必需位字段（已设置 MEMC_TRAINING 的情况除外），原因是在 DFITMG0 和 DFITMG1 中定义的 DFI 读写延迟已足够满足这些协议的需求。

单位：时钟周期。

编程模式：准动态组 1、组 2 和组 4。

位 15:14 保留，必须保持复位值。

位 13:8 **RD2WR[5:0]**：

DDR2/3/mDDR：RL + BL/2 + 2 - WL

DDR4：RL + BL/2 + 1 + WR_PREAMBLE - WL

LPDDR2/LPDDR3：RL + BL/2 + RU(tDQSCKmax/tCK) + 1 - WL

LPDDR4（禁止 DQ ODT 时）：RL + BL/2 + RU(tDQSCKmax/tCK) + WR_PREAMBLE + RD_POSTAMBLE - WL

LPDDR4（使能 DQ ODT 时）：RL + BL/2 + RU(tDQSCKmax/tCK) + RD_POSTAMBLE - ODTLon - RU(tODTon(min)/tCK)

从读取命令到写入命令的最短时间。其中包括所有逐 bank、逐 rank 和全局限制以及总线周转的时间。有关此处应包括的详细内容，请参见相关 PHY 数据手册。

单位：时钟。

其中：

- WL = 写入延迟
- BL = 突发长度。此参数必须与 SDRAM 模式寄存器的 BL 位中的编程值相匹配
- RL = 读取延迟 = CAS 延迟
- WR_PREAMBLE = 写入预编码。此为 DDR4 和 LPDDR4 所独有的参数。
- RD_POSTAMBLE = 读取后编码。此为 LPDDR4 独有的参数。

对于 LPDDR2/LPDDR3/LPDDR4，如果使能了降额 (DERATEEN.derate_enable=1)，则应使用降额后的 tDQSCKmax。

当控制器在 1:2 频率比模式下运行时，请将使用上述公式计算得出的值除以 2，并将结果向上舍入到下一个整数值。

请注意，如果使用 LRDIMM，则可能需要调整此参数的值，以通过 LRDIMM 补偿额外的延迟周

位 7:6 保留，必须保持复位值。

位 5:0 **WR2RD[5:0]**：

DDR4：CWL + PL + BL/2 + tWTR_L

LPDDR2/3/4：WL + BL/2 + tWTR + 1

其他值：CWL + BL/2 + tWTR

DDR4 中，对同一 bank 组执行写入命令与执行读取命令之间的最短时间。在其他情况下，为写入命令到读取命令的最短时间。其中包括所有逐 bank、逐 rank 和全局限制以及总线周转时间和恢复时间。

单位：时钟。

其中：

- CWL = CAS 写入延迟
- WL = 写入延迟
- PL = 奇偶校验延迟
- BL = 突发长度。此参数必须与 SDRAM 模式寄存器的 BL 位中的编程值相匹配
- tWTR_L = 面向同一 bank 组的内部写入与读取命令间的延迟。此参数直接来自 SDRAM 规范。
- tWTR = 内部写入与读取命令间的延迟。此参数直接来自 SDRAM 规范。

请为 LPDDR2/LPDDR3/LPDDR4 操作增加一个额外的周期。

当控制器在 1:2 模式下运行时，请将使用上述公式计算得出的值除以 2，并将结果向上舍入到下一个整数值。

编程模式：准动态组 1、组 2 和 组 4

5.7.26 DDRCTRL SDRAM 时序寄存器 3 (DDRCTRL_DRAMTMG3)

DDRCTRL SDRAM timing register 3

偏移地址：0x10C

复位值：0x0050 400C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	T_MRW[9:0]										Res.	Res.	T_MRD[5:4]	
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T_MRD[3:0]				Res.	Res.	T_MOD[9:0]									
rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29:20 **T_MRW[9:0]**：模式寄存器写入或读取（MRW 或 MRR）后的等待时间。
仅存在于配置为支持 LPDDR2、LPDDR3 或 LPDDR4 的设计中。
LPDDR2 通常要求值为 5。
LPDDR3 通常要求值为 10。
LPDDR4：请将其设置为 tMRW 与 tMRWCKEL 中的较大值。
对于 LPDDR2，系统会将此寄存器用于从 MRW/MRR 到所有其他命令的时间。
当控制器在 1:2 频率比模式下运行时，请将此参数设置为用上述值来除以 2，并将结果向上舍入到下一个整数值。
对于 LPDDR3，系统会将此寄存器用于从 MRW/MRR 到 MRW/MRR 的时间。
编程模式：准动态组 2 和组 4

位 19:18 保留，必须保持复位值。

位 17:12 **T_MRD[5:0]**：tMRD：模式寄存器写入或读取后需要等待的周期数。根据所连接的 SDRAM，tMRD 可表示：
DDR2/mDDR：从 MRS 到任何命令的时间
DDR3/4：从 MRS 到 MRS 命令的时间
LPDDR2：未使用
LPDDR3/4：从 MRS 到非 MRS 命令的时间。
当控制器在 1:2 频率比模式下运行时，请将此参数设置为 (tMRD/2)，并将结果向上舍入到下一个整数值。
如果使用面向 DDR4 的 C/A 奇偶校验，则将其设置为 tMRD_PAR(tMOD+PL)。
编程模式：准动态组 2 和组 4

位 11:10 保留，必须保持复位值。

位 9:0 **T_MOD[9:0]**：tMOD：仅用于 DDR3 和 DDR4 的参数。加载模式命令与后续的非加载模式命令之间的周期。

如果使用面向 DDR4 的 C/A 奇偶校验，则将其设置为 tMOD_PAR(tMOD+PL)。

如果使用面向 DDR4 的 MPR 写入，则应将其设置为 tMOD + AL（如果同时使用了 C/A 奇偶校验，则将其设置为 tMPD_PAR + AL）。

如果控制器在 1:1 频率比模式下运行，则将其设置为 tMOD；如果控制器在 1:2 频率比模式下运行，则将其设置为 tMOD/2（向上舍入到下一个整数）。请注意，如果使用 RDIMM/LRDIMM，则可能需要调整此参数的值，以补偿 RDIMM/LRDIMM 芯片对模式寄存器写入所应用的额外延迟周期，具体取决于 PHY 的情况。

另外请注意，如果使用 LRDIMM，则当控制器在 1:1 频率比模式下运行时，此寄存器的最小值为 tMRD_L2；当控制器在 1:2 频率比模式下运行时，该值为 tMRD_L2/2（向上舍入到下一个整数）。

编程模式：准动态组 2 和组 4

5.7.27 DDRCTRL SDRAM 时序寄存器 4 (DDRCTRL_DRAMTMG4)

DDRCTRL SDRAM timing register 4

偏移地址：0x110

复位值：0x0504 0405

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	T_RCD[4:0]					Res.	Res.	Res.	Res.	T_CCD[3:0]			
			rw	rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	T_RRD[3:0]				Res.	Res.	Res.	T_RP[4:0]				
				rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **T_RCD[4:0]**：tRCD - tAL：从激活 bank 到对同一 bank 发出读或写命令的最短时间。

当控制器在 1:2 频率比模式下运行时，请将此参数设置为 ((tRCD - tAL)/2)，并将结果向上舍入到下一个整数值。

该寄存器所允许的最小值为 1，这意味着当控制器在 1:2 频率比模式下运行时，(tRCD - tAL) 的最小值为 2。

单位：时钟。

编程模式：准动态组 1、组 2 和 组 4

位 23:20 保留，必须保持复位值。

位 19:16 **T_CCD[3:0]**：DDR4：tCCD_L：此参数为面向同一 bank 组执行两次读取或两次写入之间的最短时间。

其他值：tCCD：此参数为两次读取或两次写入之间的最短时间。

当控制器在 1:2 频率比模式下运行时，请将此参数设置为（tCCD_L/2 或 tCCD/2），并将结果向上舍入到下一个整数值。

单位：时钟周期。

编程模式：准动态组 2 和组 4。

位 15:12 保留，必须保持复位值。

位 11:8 **T_RRD[3:0]**：DDR4：tRRD_L：面向同一 bank 组，从激活 bank“a”到激活 bank“b”的最短时间。

其他值：tRRD：从激活 bank“a”到激活 bank“b”的最短时间

当控制器在 1:2 频率比模式下运行时，请将此参数设置为（tRRD_L/2 或 tRRD/2），并将结果向上舍入到下一个整数值。

单位：时钟。

编程模式：准动态组 2 和组 4。

位 7:5 保留，必须保持复位值。

位 4:0 **T_RP[4:0]**：tRP：从对 bank 进行预充电到激活同一 bank 的最短时间。

当控制器在 1:1 频率比模式下运行时，应将 t_rp 设置为 RoundUp(tRP/tCK)。

当控制器在 1:2 频率比模式下运行时，应将 t_rp 设置为 RoundDown(RoundUp(tRP/tCK)/2) + 1。

当控制器在 LPDDR4 中以 1:2 频率比模式运行时，应将 t_rp 设置为 RoundUp(RoundUp(tRP/tCK)/2)。

单位：时钟。

编程模式：准动态组 2 和组 4。

5.7.28 DDRCTRL SDRAM 时序寄存器 5 (DDRCTRL_DRAMTMG5)

DDRCTRL SDRAM timing register 5

偏移地址：0x114

复位值：0x0505 0403

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	T_CKSRX[3:0]				Res.	Res.	Res.	Res.	T_CKSRE[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	T_CKESR[5:0]						Res.	Res.	Res.	T_CKE[4:0]				
		rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **T_CKSRX[3:0]**：此参数是在发出 SRX 前，退出自刷新模式之前 CK 保持为有效时钟的时间。指定 SRX 前的时钟稳定时间。

推荐设置：

- mDDR：1
- LPDDR2：2
- LPDDR3：2
- LPDDR4：tCKCKEH
- DDR2：1
- DDR3：tCKSRX
- DDR4：tCKSRX

当控制器在 1:2 频率比模式下运行时，请将此参数设置为用推荐值来除以 2，并将结果向上舍入到下一个整数。

编程模式：准动态组 2 和组 4

位 23:20 保留，必须保持复位值。

位 19:16 **T_CKSRE[3:0]**：此参数为进入自刷新掉电模式后，CK 保持为有效时钟的时间。指定 SRE 之后的时钟禁止延迟。

推荐设置：

- mDDR：0
- LPDDR2：2
- LPDDR3：2
- LPDDR4：tCKELCK
- DDR2：1
- DDR3：max(10 ns,5 tCK)
- DDR4：max(10 ns,5 tCK)（+ PL（奇偶校验延迟）(*)）

(*)仅当 CRCPARCTL1.caparity_disable_before_sr=0 时，此寄存器才应增加 PL。

当控制器在 1:2 频率比模式下运行时，请将此参数设置为用推荐值来除以 2，并将结果向上舍入到下一个整数。

编程模式：准动态组 2 和组 4

位 15:14 保留，必须保持复位值。

位 13:8 **T_CKESR[5:0]**：内存时钟周期中，自刷新或自刷新掉电模式的进入到退出时序的最小 CKE 低电平宽度。

推荐设置：

- mDDR：tRFC
- LPDDR2：tCKESR
- LPDDR3：tCKESR
- LPDDR4：max(tCKE,tSR)
- DDR2：tCKE
- DDR3：tCKE + 1
- DDR4：tCKE + 1（+ PL（奇偶校验延迟）(*)）

(*)仅当 CRCPARCTL1.caparity_disable_before_sr=0 时，此寄存器才应增加 PL。

当控制器在 1:2 频率比模式下运行时，请将此参数设置为用推荐值来除以 2，并将结果向上舍入到下一个整数。

编程模式：准动态组 2 和组 4

位 7:5 保留，必须保持复位值。

位 4:0 **T_CKE[4:0]**：掉电和自刷新期间，CKE 高电平/低电平的最小周期数。

- LPDDR2/LPDDR3 模式：请将此参数设置为 tCKE 与 tCKESR 中的较大值。
- LPDDR4 模式：请将此参数设置为 tCKE 与 tSR 中的较大值。
- 非 LPDDR2/非 LPDDR3/非 LPDDR4 设计：请将此参数设置为 tCKE 值。

当控制器在 1:2 频率比模式下运行时，请将此参数设置为（上述值）/2，并将结果向上舍入到下一个整数值。

单位：时钟。

编程模式：准动态组 2 和组 4

5.7.29 DDRCTRL SDRAM 时序寄存器 6 (DDRCTRL_DRAMTMG6)

DDRCTRL SDRAM timing register 6

偏移地址：0x118

复位值：0x0202 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	T_CKDPDE[3:0]				Res.	Res.	Res.	Res.	T_CKDPDX[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T_CKCSX[3:0]			
												rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **T_CKDPDE[3:0]**：此参数为进入深度掉电模式后，CK 保持为有效时钟的时间。指定 DPDE 之后的时钟禁止延迟。

推荐设置：

- mDDR：0
- LPDDR2：2
- LPDDR3：2

当控制器在 1:2 频率比模式下运行时，请将此参数设置为用推荐值来除以 2，并将结果向上舍入到下一个整数。

此位仅存在于支持 mDDR 或 LPDDR2/LPDDR3 设备的设计中。

编程模式：准动态组 2 和组 4

位 23:20 保留，必须保持复位值。

位 19:16 **T_CKDPDX[3:0]**：此参数是在发出 DPDX 前，退出深度掉电模式之前 CK 保持为有效时钟的时间。指定 DPDX 前的时钟稳定时间。

推荐设置：

- mDDR：1
- LPDDR2：2
- LPDDR3：2

当控制器在 1:2 频率比模式下运行时，请将此参数设置为用推荐值来除以 2，并将结果向上舍入到下一个整数。

此位仅存在于支持 mDDR 或 LPDDR2 设备的设计中。

编程模式：准动态组 2 和组 4

位 15:4 保留，必须保持复位值。

位 3:0 **T_CKCSX[3:0]**：此参数是在发出时钟停止退出命令前，退出时钟停止模式之前 CK 保持为有效时钟的时间。指定发出时钟停止退出命令后的下一条命令之前的时钟稳定时间。

推荐设置：

- mDDR：1
- LPDDR2：tXP + 2
- LPDDR3：tXP + 2
- LPDDR4：tXP + 2

当控制器在 1:2 频率比模式下运行时，请将此参数设置为用推荐值来除以 2，并将结果向上舍入到下一个整数。

此位仅存在于支持 mDDR 或 LPDDR2/LPDDR3/LPDDR4 设备的设计中。

编程模式：准动态组 2 和组 4

5.7.30 DDRCTRL SDRAM 时序寄存器 7 (DDRCTRL_DRAMTMG7)

DDRCTRL SDRAM timing register 7

偏移地址：0x11C

复位值：0x0000 0202

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	T_CKPDE[3:0]				Res.	Res.	Res.	Res.	T_CKPDX[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:8 **T_CKPDE[3:0]**：此参数为进入掉电模式后，CK 保持为有效时钟的时间。指定 PDE 之后的时钟禁止延迟。

推荐设置：

- mDDR：0
- LPDDR2：2
- LPDDR3：2
- LPDDR4：tCKELCK

使用 DDR2/3/4 SDRAM 时，应将此寄存器设置为与 DRAMTMG5.t_cksre 相同的值。

当控制器在 1:2 频率比模式下运行时，请将此参数设置为用推荐值来除以 2，并将结果向上舍入到下一个整数。

此位仅存在于支持 mDDR 或 LPDDR2/LPDDR3/LPDDR4 设备的设计中。

编程模式：准动态组 2 和组 4

位 7:4 保留，必须保持复位值。

位 3:0 **T_CKPDX[3:0]**：此参数是在发出 PDX 前，退出掉电模式之前 CK 保持为有效时钟的时间。指定 PDX 前的时钟稳定时间。

推荐设置：

- mDDR：0
- LPDDR2：2
- LPDDR3：2
- LPDDR4：2

使用 DDR2/3/4 SDRAM 时，应将此寄存器设置为与 DRAMTMG5.t_cksrx 相同的值。

当控制器在 1:2 频率比模式下运行时，请将此参数设置为用推荐值来除以 2，并将结果向上舍入到下一个整数。

此位仅存在于支持 mDDR 或 LPDDR2/LPDDR3/LPDDR4 设备的设计中。

编程模式：准动态组 2 和组 4

5.7.31 DDRCTRL SDRAM 时序寄存器 8 (DDRCTRL_DRAMTMG8)

DDRCTRL SDRAM timing register 8

偏移地址：0x120

复位值：0x0000 4405

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	T_XS_DLL_X32[6:0]							Res.	T_XS_X32[6:0]						
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:15 保留，必须保持复位值。

位 14:8 **T_XS_DLL_X32[6:0]**：tXSDLL：退出自刷新模式以执行需要锁定 DLL 的命令。
当控制器在 1:2 频率比模式下运行时，请将此参数设置为用上述值来除以 2，并向上舍入到下一个整数值。
单位：32 个时钟周期的倍数。
注意：仅用于 DDR2、DDR3 和 DDR4 SDRAM。
编程模式：准动态组 2 和组 4

位 7 保留，必须保持复位值。

位 6:0 **T_XS_X32[6:0]**：tXS：退出自刷新模式以执行不需要锁定 DLL 的命令。
当控制器在 1:2 频率比模式下运行时，请将此参数设置为用上述值来除以 2，并向上舍入到下一个整数值。
单位：32 个时钟周期的倍数。
注意：仅用于 DDR2、DDR3 和 DDR4 SDRAM。
编程模式：准动态组 2 和组 4

5.7.32 DDRCTRL SDRAM 时序寄存器 14 (DDRCTRL_DRAMTMG14)

DDRCTRL SDRAM timing register 14

偏移地址：0x138

复位值：0x0000 00A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	T_XSR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:0 **T_XSR[11:0]**：tXSR：退出自刷新模式以执行任何命令。
当控制器在 1:2 频率比模式下运行时，请将此参数设置为用上述值来除以 2，并向上舍入到下一个整数值。
注：仅用于 mDDR/LPDDR2/LPDDR3/LPDDR4 模式。
编程模式：准动态组 2 和组 4

5.7.33 DDRCTRL SDRAM 时序寄存器 15 (DDRCTRL_DRAMTMG15)

DDRCTRL SDRAM timing register 15

偏移地址：0x13C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN_DFI_LP_T_STAB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T_STAB_X32[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31 **EN_DFI_LP_T_STAB**：

- 1 - 退出 DFI LP 时使能使用 tSTAB。当 PHY 在 DFI LP 期间停止时钟以最大限度降低功耗时，需要设置此参数。

- 0 - 退出 DFI LP 时禁止使用 tSTAB。

编程模式：准动态组 2 和组 4。

位 30:8 保留，必须保持复位值。

位 7:0 **T_STAB_X32[7:0]**：tSTAB：稳定时间。

在以下两种情况下，需要将其用于 DDR3/DDR4 RDIMM：

- 当退出节能模式时，如果时钟已经停止，则在重新使能时钟后，必须在由 tSTAB 指定的一段时间内使其保持稳定

- 在输入时钟频率变化的情况下 (DDR4)

- 在发出涉及时钟时序的控制字之后（规范：DDR3 为 6 us，DDR4 为 5 us）。

当控制器在 1:2 频率比模式下运行时，请将此参数设置为用推荐值来除以 2，并将结果向上舍入到下一个整数。

单位：32 个时钟周期的倍数。

编程模式：准动态组 2 和组 4。

5.7.34 DDRCTRL ZQ 控制寄存器 0 (DDRCTRL_ZQCTL0)

DDRCTRL ZQ control register 0

偏移地址：0x180

复位值：0x0200 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIS_AUTO_ZQ	DIS_SRX_ZQCL	ZQ_RESISTOR_SHARED	Res.	Res.	T_ZQ_LONG_NOP[10:0]										
rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	T_ZQ_SHORT_NOP[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **DIS_AUTO_ZQ**：

- 1 - 禁止 DDRCTRL 生成 ZQCS/MPC（ZQ 校准）命令。可以改为使用寄存器 DBGCMD.zq_calib_short 从 APB 模块发出 ZQ 校准请求。

- 0 - 根据 ZQCTL1.t_zq_short_interval_x1024 在内部生成 ZQCS/MPC（ZQ 校准）命令。

此位仅存在于支持 DDR3/DDR4 或 LPDDR2/LPDDR3/LPDDR4 设备的设计中。

编程模式：动态

位 30 **DIS_SRX_ZQCL**：

- 1 - 禁止在退出自刷新/SR 掉电模式时发出 ZQCL/MPC（ZQ 校准）命令。仅在运行 DDR3、DDR4、LPDDR2、LPDDR3 或 LPDDR4 模式时适用。

- 0 - 使能在退出自刷新/SR 掉电模式时发出 ZQCL/MPC（ZQ 校准）命令。仅在运行 DDR3、DDR4、LPDDR2、LPDDR3 或 LPDDR4 模式时适用。

此位仅存在于支持 DDR3/DDR4 或 LPDDR2/LPDDR3/LPDDR4 设备的设计中。

编程模式：准动态组 2 和组 4。

位 29 **ZQ_RESISTOR_SHARED**：

- 1 - 表示 ZQ 电阻在 rank 之间共享。意味着 ZQinit/ZQCL/ZQCS/MPC（ZQ 校准）命令每次发送到一个 rank，并且这些命令之间符合 tZQinit/tZQCL/tZQCS/tZQCAL/tZQLAT 时序的要求，从而使发送到不同 rank 的命令不会重叠。

- 0 - ZQ 电阻不共享。

此位仅存在于支持 DDR3/DDR4 或 LPDDR2/LPDDR3/LPDDR4 设备的设计中。

编程模式：静态。

位 28:27 保留，必须保持复位值。

位 26:16 **T_ZQ_LONG_NOP[10:0]**：DDR3/DDR4 的 tZQoper、LPDDR2/LPDDR3 的 tZQCL、LPDDR4 的 tZQCAL：向 SDRAM 发出 ZQCL（ZQ 长校准）/MPC（ZQ 启动）命令后所需的 NOP DFI 时钟周期数。

当控制器在 1:2 频率比模式下运行时：

DDR3/DDR4：请将此参数设置为 tZQoper/2，并将结果向上舍入到下一个整数值。

LPDDR2/LPDDR3：请将此参数设置为 tZQCL/2，并将结果向上舍入到下一个整数值。

LPDDR4：将其设置为 tZQCAL/2，并将结果向上舍入到下一个整数值。

此位仅存在于支持 DDR3/DDR4 或 LPDDR2/LPDDR3/LPDDR4 设备的设计中。

编程模式：静态。

位 15:10 保留，必须保持复位值。

位 9:0 **T_ZQ_SHORT_NOP[9:0]**：DDR3/DD4/LPDDR2/LPDDR3 的 tZQCS，LPDDR4 的 tZQLAT：向 SDRAM 发出 ZQCS（ZQ 短校准）/MPC（ZQ 锁存）命令后所需的 NOP DFI 时钟周期数。

当控制器在 1:2 频率比模式下运行时，请将此参数设置为 tZQCS/2，并将结果向上舍入到下一个整数值。

此位仅存在于支持 DDR3/DDR4 或 LPDDR2/LPDDR3/LPDDR4 设备的设计中。

编程模式：静态。

5.7.35 DDRCTRL ZQ 控制寄存器 1 (DDRCTRL_ZQCTL1)

DDRCTRL ZQ control register 1

偏移地址：0x184

复位值：0x0200 0100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	T_ZQ_RESET_NOP[9:0]										T_ZQ_SHORT_INTERVAL_X1024[19:16]			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T_ZQ_SHORT_INTERVAL_X1024[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29:20 **T_ZQ_RESET_NOP[9:0]**：tZQReset：向 SDRAM 发出 ZQReset（ZQ 校准复位）命令后所需的 NOP DFI 时钟周期数。

当控制器在 1:2 频率比模式下运行时，请将此参数编程为 tZQReset/2，并将结果向上舍入到下一个整数值。

此位仅存在于支持 LPDDR2/LPDDR3/LPDDR4 设备的设计中。

编程模式：静态。

位 19:0 **T_ZQ_SHORT_INTERVAL_X1024[19:0]**：系统自动向 DDR3/DDR4/LPDDR2/LPDDR3/LPDDR4 设备发出 ZQCS（ZQ 短校准）/MPC（ZQ 校准）命令之间的平均等待间隔。

如果 ZQCTL0.dis_auto_zq=1，则此参数无意义。

单位：1024 个 DFI 时钟周期。

此位仅存在于支持 DDR3/DDR4 或 LPDDR2/LPDDR3/LPDDR4 设备的设计中。

编程模式：静态。

5.7.36 DDRCTRL ZQ 控制寄存器 2 (DDRCTRL_ZQCTL2)

DDRCTRL ZQ control register 2

偏移地址：0x188

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ZQ_RESET
															rs

位 31:1 保留，必须保持复位值。

位 0 **ZQ_RESET**：

将此寄存器位设置为 1 将触发 ZQ 复位操作。ZQ 复位操作完成后，DDRCTRL 会自动将该位清零。如果处于初始化、自刷新（LPDDR4 除外）、SR 掉电 (LPDDR4) 或深度掉电工作模式下，建议不要设置此寄存器位。

对于自刷新（LPDDR4 除外）或 SR 掉电 (LPDDR4)，系统会安排在退出 SR（LPDDR4 除外）或 SPRD (LPDDR4) 模式后进行该操作。

对于深度掉电，尽管系统会将 ZQSTAT.zq_reset_busy 清零，但不会安排进行该操作。

此位仅存在于支持 LPDDR2/LPDDR3/LPDDR4 设备的设计中。

编程模式：动态

5.7.37 DDRCTRL ZQ 状态寄存器 (DDRCTRL_ZQSTAT)

DDRCTRL ZQ status register

偏移地址：0x18C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ZQ_RESET_BUSY
															r

位 31:1 保留，必须保持复位值。

位 0 **ZQ_RESET_BUSY**：仅当此信号为低电平时，SoC 内核才能启动 ZQ 复位操作。在 DDRCTRL 接受 ZQ 复位请求后的那个时钟周期，此信号会变为高电平。当向 SDRAM 发出 ZQ 复位命令且相关 NOP 周期结束时，此信号会变为低电平。建议在此信号为高电平时不要执行 ZQ 复位命令。

- 0 - 表示 SoC 内核可以启动 ZQ 复位操作
- 1 - 表示正在进行 ZQ 复位操作

编程模式：动态

5.7.38 DDRCTRL DFI 时序寄存器 0 (DDRCTRL_DFITMG0)

DDRCTRL DFI timing register 0

偏移地址：0x190

复位值：0x0702 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	DFI_T_CTRL_DELAY[4:0]					Res.	DFI_T_RDDATA_EN[6:0]						
			rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DFI_TPHY_WRDATA[5:0]						Res.	Res.	DFI_TPHY_WRLAT[5:0]					
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **DFI_T_CTRL_DELAY[4:0]**：指定 DFI 控制信号生效或失效之后的 DIF 时钟周期数，PHY-DRAM 接口处的控制信号可反映出生效或失效。如果 DFI 时钟与内存时钟并非相位对齐，则应将此时序参数向上舍入到下一个整数值。请注意，如果使用 RDIMM/LRDIMM，则必须在 DFI 时钟上将此参数递增 RDIMM/LRDIMM 的额外延迟周期。

编程模式：准动态组 4

位 23 保留，必须保持复位值。

位 22:16 **DFI_T_RDDATA_EN[6:0]**：从读取命令在 DFI 接口上生效到 dfi_rddata_en 信号生效所间隔的时间。

请参考 PHY 规范以得到正确的值。

该参数与 DFI 参数 trddata_en 相对应。请注意，如果使用 RDIMM/LRDIMM，则可能需要在计算 trddata_en 时使用调整后的 CL 值，具体取决于 PHY 的情况。这是为了通过 RDIMM/LRDIMM 补偿额外的延迟周期。

单位：DFI 时钟周期或 DFI PHY 时钟周期，具体取决于 DFITMG0.dfi_rddata_use_dfi_phy_clk。

编程模式：准动态组 1 和组 4

位 15:14 保留，必须保持复位值。

位 13:8 **DFI_TPHY_WRDATA[5:0]**：指定 dfi_wrdata_en 生效时与相关写入数据在 dfi_wrdata 信号上被驱动时所间隔的时钟周期数。该参数与 DFI 时序参数 tphy_wrdata 相对应。请参考 PHY 规范以得到正确的值。请注意，支持的最大值为 8。

单位：DFI 时钟周期或 DFI PHY 时钟周期，具体取决于 DFITMG0.dfi_wrdata_use_dfi_phy_clk。

编程模式：准动态组 4

位 7:6 保留，必须保持复位值。

位 5:0 **DFI_TPHY_WRLAT[5:0]**：写入延迟

从写入命令到写入数据使能 (dfi_wrdata_en) 的时钟周期数。该参数与 DFI 时序参数 tphy_wrlat 相对应。

请参考 PHY 规范以得到正确的值。请注意，如果使用 RDIMM/LRDIMM，则可能需要在计算 tphy_wrlat 时使用调整后的 CL 值，具体取决于 PHY 的情况。这是为了通过 RDIMM/LRDIMM 补偿额外的延迟周期。

对于 LPDDR4，不支持 dfi_tphy_wrlat>60。

单位：DFI 时钟周期或 DFI PHY 时钟周期，具体取决于 DFITMG0.dfi_wrdata_use_dfi_phy_clk。

编程模式：准动态组 1 和组 4

5.7.39 DDRCTRL DFI 时序寄存器 1 (DDRCTRL_DFITMG1)

DDRCTRL DFI timing register 1

偏移地址：0x194

复位值：0x0000 0404

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFI_T_WRDATA_DELAY[4:0]				
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DFI_T_DRAM_CLK_DISABLE[4:0]					Res.	Res.	Res.	DFI_T_DRAM_CLK_ENABLE[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值。

位 20:16 **DFI_T_WRDATA_DELAY[4:0]**：指定 dfi_wrdata_en 信号生效时与 DRAM 总线上完成对应的写入数据传输时所间隔的 DFI 时钟周期数。

该参数与 DFI 时序参数 twrdata_delay 相对应。请参考 PHY 规范以得到正确的值。

对于 DFI 3.0 PHY，请将其设置为 twrdata_delay；这是 DFI 3.0 中引入的一个新时序参数。

对于 DFI 2.1 PHY，请将其设置为 tphy_wrdata +（DFI 写入数据到 DRAM 的延迟）。

要设置的值将以 DFI 时钟周期为依据，而非以 PHY 时钟周期为依据。

在 FREQ_RATIO=2 时，请将 PHY 的值除以 2，并向上舍入到下一个整数。

如果使用 DFITMG0.dfi_wrdata_use_dfi_phy_clk=1，则在该值上加 1。

单位：时钟

编程模式：准动态组 4

位 15:13 保留，必须保持复位值。

位 12:8 **DFI_T_DRAM_CLK_DISABLE[4:0]**：指定从 DFI 上 dfi_dram_clk_disable 信号生效到 PHY-DRAM 边界的 DRAM 内存设备时钟保持低值所经过的 DIF 时钟周期数。如果 DFI 时钟与内存时钟并非相位对齐，则应将此时序参数向上舍入到下一个整数值。

编程模式：准动态组 4

位 7:5 保留，必须保持复位值。

位 4:0 **DFI_T_DRAM_CLK_ENABLE[4:0]**：指定从 DFI 上的 dfi_dram_clk_disable 信号失效到 PHY-DRAM 边界处 DRAM 内存设备的时钟的第一个有效上升沿所间隔的 DIF 时钟周期数。如果 DFI 时钟与内存时钟并非相位对齐，则应将此时序参数向上舍入到下一个整数值。
编程模式：准动态组 4

5.7.40 DDRCTRL 低功耗配置寄存器 0 (DDRCTRL_DFILPCFG0)

DDRCTRL low power configuration register 0

偏移地址：0x198

复位值：0x0700 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	DFI_TLP_RESP[4:0]					DFI_LP_WAKEUP_DPD[3:0]				Res.	Res.	Res.	DFI_LP_EN_DPD
			rw	rw	rw	rw	rw	rw	rw	rw	rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFI_LP_WAKEUP_SR[3:0]				Res.	Res.	Res.	DFI_LP_EN_SR	DFI_LP_WAKEUP_PD[3:0]				Res.	Res.	Res.	DFI_LP_EN_PD
rw	rw	rw	rw				rw	rw	rw	rw	rw				rw

位 31:29 保留，必须保持复位值。

位 28:24 **DFI_TLP_RESP[4:0]**：在 DFI tlp_resp 时间的 DFI 时钟周期中设置。
相同的值将用于掉电、自刷新、深度掉电和最大程度节能模式。
从 DFI 2.1 规范开始，始终建议使用固定值 7。
编程模式：静态

位 23:20 **DFI_LP_WAKEUP_DPD[3:0]**：进入深度掉电模式时，用于驱动 dfi_lp_wakeup 信号的 DFI 时钟周期值。

确定 DFI 的 tlp_wakeup 时间：

- 0x0 - 16 个周期
- 0x1 - 32 个周期
- 0x2 - 64 个周期
- 0x3 - 128 个周期
- 0x4 - 256 个周期
- 0x5 - 512 个周期
- 0x6 - 1024 个周期
- 0x7 - 2048 个周期
- 0x8 - 4096 个周期
- 0x9 - 8192 个周期
- 0xA - 16384 个周期
- 0xB - 32768 个周期
- 0xC - 65536 个周期
- 0xD - 131072 个周期
- 0xE - 262144 个周期
- 0xF - 无限制

此位仅存在于支持 mDDR 或 LPDDR2/LPDDR3 设备的设计中。

编程模式：静态。

位 19:17 保留，必须保持复位值。

位 16 **DFI_LP_EN_DPD**：在进入/退出深度掉电模式期间，使能 DFI 低功耗接口握手。

- 0 - 禁止
- 1 - 使能

此位仅存在于支持 mDDR 或 LPDDR2/LPDDR3 设备的设计中。

编程模式：静态。

位 15:12 **DFI_LP_WAKEUP_SR[3:0]**：进入自刷新模式时，用于驱动 dfi_lp_wakeup 信号的 DFI 时钟周期值。

确定 DFI 的 tlp_wakeup 时间：

- 0x0 - 16 个周期
- 0x1 - 32 个周期
- 0x2 - 64 个周期
- 0x3 - 128 个周期
- 0x4 - 256 个周期
- 0x5 - 512 个周期
- 0x6 - 1024 个周期
- 0x7 - 2048 个周期
- 0x8 - 4096 个周期
- 0x9 - 8192 个周期
- 0xA - 16384 个周期
- 0xB - 32768 个周期
- 0xC - 65536 个周期
- 0xD - 131072 个周期
- 0xE - 262144 个周期
- 0xF - 无限制

编程模式：静态

位 11:9 保留，必须保持复位值。

位 8 **DFI_LP_EN_SR**：在进入/退出自刷新模式期间，使能 DFI 低功耗接口握手。

- 0 - 禁止
- 1 - 使能

编程模式：静态

位 7:4 **DFI_LP_WAKEUP_PD[3:0]**：进入掉电模式时，用于驱动 dfi_lp_wakeup 信号的 DFI 时钟周期值。

确定 DFI 的 tlp_wakeup 时间：

- 0x0 - 16 个周期
- 0x1 - 32 个周期
- 0x2 - 64 个周期
- 0x3 - 128 个周期
- 0x4 - 256 个周期
- 0x5 - 512 个周期
- 0x6 - 1024 个周期
- 0x7 - 2048 个周期
- 0x8 - 4096 个周期
- 0x9 - 8192 个周期
- 0xA - 16384 个周期
- 0xB - 32768 个周期
- 0xC - 65536 个周期
- 0xD - 131072 个周期
- 0xE - 262144 个周期
- 0xF - 无限制

编程模式：静态

位 3:1 保留，必须保持复位值。

位 0 **DFI_LP_EN_PD**：在进入/退出掉电模式期间，使能 DFI 低功耗接口握手。
- 0 - 禁止
- 1 - 使能

编程模式：静态

5.7.41 DDRCTRL DFI 更新寄存器 0 (DDRCTRL_DFIUPD0)

DDRCTRL DFI update register 0

偏移地址：0x1A0

复位值：0x0040 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIS_AUTO_CTRLUPD	DIS_AUTO_CTRLUPD_SRX	CTRLUPD_PRE_SRX	Res.	Res.	Res.	DFI_T_CTRLUP_MAX[9:0]									
rw	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	DFI_T_CTRLUP_MIN[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **DIS_AUTO_CTRLUPD**：当设置为“1”时，将禁止 DDRCTRL 自动生成 dfi_ctrlupd_req。内核必须使用寄存器 DBGCMD.ctrlupd 发出 dfi_ctrlupd_req 信号。
当设置为“0”时，DDRCTRL 会定期发出 dfi_ctrlupd_req。
编程模式：准动态组 3

位 30 **DIS_AUTO_CTRLUPD_SRX**：当设置为“1”时，系统将在退出自刷新模式时禁止 DDRCTRL 自动生成 dfi_ctrlupd_req。
当设置为“0”时，DDRCTRL 会在退出自刷新模式之前或之后发出 dfi_ctrlupd_req，具体取决于 DFIUPD0.ctrlupd_pre_srx。
编程模式：静态

位 29 **CTRLUPD_PRE_SRX**：选择 SRX 时的 dfi_ctrlupd_req 要求：
- 0：在 SRX 后发送 ctrlupd
- 1：在 SRX 前发送 ctrlupd

如果 DFIUPD0.dis_auto_ctrlupd_srx=1，则此寄存器将无影响，原因是 SRX 时不会发出 dfi_ctrlupd_req。
编程模式：静态

位 28:26 保留，必须保持复位值。

位 25:16 **DFI_T_CTRLUP_MAX[9:0]**：指定 dfi_ctrlupd_req 信号可以生效的最大 DFI 时钟周期数。分配到此变量的最低值为 0x40。
编程模式：静态

位 15:10 保留，必须保持复位值。

位 9:0 **DFI_T_CTRLUP_MIN[9:0]**：指定 dfi_ctrlupd_req 信号必须生效的最小 DFI 时钟周期数。DDRCTRL 预期 PHY 会在此时间内进行响应。如果 PHY 没有响应，则 DDRCTRL 会在 dfi_t_ctrlup_min + 2 个周期后将 dfi_ctrlupd_req 清零。分配到此变量的最低值为 0x3。

编程模式：静态

5.7.42 DDRCTRL DFI 更新寄存器 1 (DDRCTRL_DFIUPD1)

DDRCTRL DFI update register 1

偏移地址：0x1A4

复位值：0x0001 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFI_T_CTRLUPD_INTERVAL_MIN_X1024[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFI_T_CTRLUPD_INTERVAL_MAX_X1024[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **DFI_T_CTRLUPD_INTERVAL_MIN_X1024[7:0]**：此参数为 DDRCTRL 启动 DFI 更新请求（当 DDRCTRL 空闲时执行）的最短间隔时间。将其设置为较高数值可以降低更新请求的频率；这可能会轻微影响到 DDRCTRL 空闲时首次读取请求的延迟情况。此位字段所允许的最小值是 1。

单位：1024 个 DFI 时钟周期。

编程模式：静态。

位 15:8 保留，必须保持复位值。

位 7:0 **DFI_T_CTRLUPD_INTERVAL_MAX_X1024[7:0]**：此参数为 DDRCTRL 启动 DFI 更新请求的最长间隔时间。该定时器会在每次更新请求时复位；当定时器到期时，系统将发送 dfi_ctrlupd_req 并阻塞数据传输，直至接收到 dfi_ctrlupd_ackx 为止。PHY 可利用这段空闲时间将延迟线重新校准至 DLL。一旦出现数据捕获错误，也可使用 DFI 控制器更新来复位 PHY FIFO 指针。系统需要通过更新来保持对 PVT 的校准，但频繁的更新可能会影响性能。此位字段所允许的最小值是 1。

注意：为 DFIUPD1.dfi_t_ctrlupd_interval_max_x1024 设置的值必须大于 DFIUPD1.dfi_t_ctrlupd_interval_min_x1024。

单位：1024 个 DFI 时钟周期。

编程模式：静态。

5.7.43 DDRCTRL DFI 更新寄存器 2 (DDRCTRL_DFIUPD2)

DDRCTRL DFI update register 2

偏移地址：0x1A8

复位值：0x8000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DFI_PHYUPD_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31 **DFI_PHYUPD_EN**：使能对确认 PHY 发起的更新的支持：
- 0 - 禁止
- 1 - 使能

编程模式：静态

位 30:0 保留，必须保持复位值。

5.7.44 DDRCTRL DFI 杂项控制寄存器 (DDRCTRL_DFIMISC)

DDRCTRL DFI miscellaneous control register

偏移地址：0x1B0

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DFI_FREQUENCY[4:0]					Res.	Res.	DFI_INIT_START	CTL_IDLE_EN	Res.	Res.	Res.	DFI_INIT_COMPLETE_EN
			rw	rw	rw	rw	rw			rw	rw				rw

位 31:13 保留，必须保持复位值。

位 12:8 **DFI_FREQUENCY[4:0]**：指示系统的工作频率。所支持频率的数量以及信号值对时钟频率的映射将由 PHY 进行定义。

编程模式：准动态组 1

位 7:6 保留，必须保持复位值。

位 5 **DFI_INIT_START**：PHY 初始化启动请求信号。一旦生效，该信号将触发 PHY 初始化启动请求
编程模式：准动态组 3

位 4 **CTL_IDLE_EN**：使能对 ctl_idle 信号的支持
编程模式：静态

位 3:1 保留，必须保持复位值。

位 0 **DFI_INIT_COMPLETE_EN**：PHY 初始化完成使能信号。
一旦生效，可使用 dfi_init_complete 信号来触发 SDRAM 初始化
编程模式：准动态组 3

5.7.45 DDRCTRL DFI 状态寄存器 (DDRCTRL_DFISTAT)

DDRCTRL DFI status register

偏移地址：0x1BC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFI_LP_ACK	DFI_INIT_COMPLETE
														r	r

位 31:2 保留，必须保持复位值。

位 1 **DFI_LP_ACK**：将 dfi_lp_ack 输入的值存储至控制器。
编程模式：动态

位 0 **DFI_INIT_COMPLETE**：用于宣布 DFI 初始化完成的状态标志寄存器。系统将使用 dfi_init_start 信号触发 DFI 初始化，然后轮询 dfi_init_complete 标志以获知初始化的完成时间。
编程模式：动态

5.7.46 DDRCTRL DFI PHY 主寄存器 (DDRCTRL_DFIPHYMSTR)

DDRCTRL DFI PHY master register

偏移地址：0x1C4

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFI_PHYMSTR_EN
															rw

位 31:1 保留，必须保持复位值。

位 0 **DFI_PHYMSTR_EN**：使能 PHY 主接口：
- 0 - 禁止
- 1 - 使能

编程模式：动态

5.7.47 DDRCTRL 地址映射寄存器 1 (DDRCTRL_ADDRMAP1)

DDRCTRL address map register 1

偏移地址：0x204

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRMAP_BANK_B2[5:0]					
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ADDRMAP_BANK_B1[5:0]						Res.	Res.	ADDRMAP_BANK_B0[5:0]					
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:16 **ADDRMAP_BANK_B2[5:0]**：选择用作 bank 地址位 2 的 HIF 地址位。
有效范围：0 到 31 以及 63
内部基数：4
所选的 HIF 地址位需通过用此位字段的值加上内部基数进行确定。
如果未使用该地址位，请将其设置为 63，然后将 bank 地址位 2 设置为 0。
编程模式：静态。

位 15:14 保留，必须保持复位值。

位 13:8 **ADDRMAP_BANK_B1[5:0]**：选择用作 bank 地址位 1 的 HIF 地址位。
有效范围：0 到 32 以及 63
内部基数：3
每个 bank 地址位的所选 HIF 地址位均需通过用此位字段的值加上内部基数进行确定。
如果未使用该地址位，请将其设置为 63，然后将 bank 地址位 1 设置为 0。
编程模式：静态。

位 7:6 保留，必须保持复位值。

位 5:0 **ADDRMAP_BANK_B0[5:0]**：选择用作 bank 地址位 0 的 HIF 地址位。
有效范围：0 到 32 以及 63
内部基数：2
每个 bank 地址位的所选 HIF 地址位均需通过用此位字段的值加上内部基数进行确定。
如果未使用该地址位，请将其设置为 63，然后将 bank 地址位 0 设置为 0。
编程模式：静态。

5.7.48 DDRCTRL 地址映射寄存器 2 (DDRCTRL_ADDRMAP2)

DDRCTRL address map register 2

偏移地址：0x208

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	ADDRMAP_COL_B5[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_COL_B4[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	ADDRMAP_COL_B3[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_COL_B2[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **ADDRMAP_COL_B5[3:0]**：
- 全总线宽度模式：选择用作列地址位 5 的 HIF 地址位。
- 半总线宽度模式：选择用作列地址位 6 的 HIF 地址位。
- 四分之一总线宽度模式：选择用作列地址位 7 的 HIF 地址位。

有效范围：0 到 7 以及 15
内部基数：5
所选的 HIF 地址位需通过用此位字段的值加上内部基数进行确定。
如果未使用该地址位，请将其设置为 15，然后将此列地址位设置为 0。
编程模式：静态

位 23:20 保留，必须保持复位值。

位 19:16 **ADDRMAP_COL_B4[3:0]**：
- 全总线宽度模式：选择用作列地址位 4 的 HIF 地址位。
- 半总线宽度模式：选择用作列地址位 5 的 HIF 地址位。
- 四分之一总线宽度模式：选择用作列地址位 6 的 HIF 地址位。

有效范围：0 到 7 以及 15
内部基数：4
所选的 HIF 地址位需通过用此位字段的值加上内部基数进行确定。

如果未使用该地址位，请将其设置为 15，然后将此列地址位设置为 0。
编程模式：静态

位 15:12 保留，必须保持复位值。

位 11:8 **ADDRMAP_COL_B3[3:0]**：
- 全总线宽度模式：选择用作列地址位 3 的 HIF 地址位。
- 半总线宽度模式：选择用作列地址位 4 的 HIF 地址位。
- 四分之一总线宽度模式：选择用作列地址位 5 的 HIF 地址位。

有效范围：0 到 7
内部基数：3
所选的 HIF 地址位需通过用此位字段的值加上内部基数进行确定。

请注意，如果 UMCTL2_INCL_ARB=1，MEMC_BURST_LENGTH=16，采用全总线宽度模式 (MSTR.data_bus_width=00) 且突发长度为 BL16 (MSTR.burst_rdwr=1000)，则建议将其设置为 0。

编程模式：静态

位 7:4 保留，必须保持复位值。

位 3:0 **ADDRMAP_COL_B2[3:0]**：

- 全总线宽度模式：选择用作列地址位 2 的 HIF 地址位。
- 半总线宽度模式：选择用作列地址位 3 的 HIF 地址位。
- 四分之一总线宽度模式：选择用作列地址位 4 的 HIF 地址位。

有效范围：0 到 7

内部基数：2

所选的 HIF 地址位需通过用此位字段的值加上内部基数进行确定。

请注意，如果 UMCTL2_INCL_ARB=1 并且 MEMC_BURST_LENGTH=8，则除非出现以下情况，否则必须将其设置为 0：

- 在半总线宽度模式或四分之一总线宽度模式下 (MSTR.data_bus_width!=00) 并且
- PCCFG.bl_exp_mode==1 并且
- 在 DDR4 模式下且 ADDRMAP8.addrmap_bg_b0==0 或
- 在 LPDDR4 模式下且 ADDRMAP1.addrmap_bank_b0==0

如果 UMCTL2_INCL_ARB=1 并且 MEMC_BURST_LENGTH=16，则除非出现以下情况，否则必须将其设置为 0：

- 在半总线宽度模式或四分之一总线宽度模式下 (MSTR.data_bus_width!=00) 并且
- PCCFG.bl_exp_mode==1 并且
- 在 DDR4 模式下且 ADDRMAP8.addrmap_bg_b0==0

否则，如果 MEMC_BURST_LENGTH=8 且采用全总线宽度模式 (MSTR.data_bus_width==00)，则建议将其设置为 0，以便将 HIF[2] 映射到列地址位 2。

如果 MEMC_BURST_LENGTH=16 且采用全总线宽度模式 (MSTR.data_bus_width==00)，则建议将其设置为 0，以便将 HIF[2] 映射到列地址位 2。

如果 MEMC_BURST_LENGTH=16 且采用半总线宽度模式 (MSTR.data_bus_width==01)，则建议将其设置为 0，以便将 HIF[2] 映射到列地址位 3。

编程模式：静态

5.7.49 DDRCTRL 地址映射寄存器 3 (DDRCTRL_ADDRMAP3)

DDRCTRL address map register 3

偏移地址：0x20C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	ADDRMAP_COL_B9[4:0]					Res.	Res.	Res.	ADDRMAP_COL_B8[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	ADDRMAP_COL_B7[4:0]					Res.	Res.	Res.	Res.	ADDRMAP_COL_B6[3:0]			
			rw	rw	rw	rw	rw					rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **ADDRMAP_COL_B9[4:0]**：

- 全总线宽度模式：选择用作列地址位 9 的 HIF 地址位。
- 半总线宽度模式：选择用作列地址位 11（LPDDR2/LPDDR3 模式下为 10）的 HIF 地址位。
- 四分之一总线宽度模式：选择用作列地址位 13（LPDDR2/LPDDR3 模式下为 11）的 HIF 地址位。

有效范围：0 到 7、x 以及 31。x 表示内联 ECC 配置中的有效值。

内部基数：9

所选的 HIF 地址位需通过用此位字段的值加上内部基数进行确定。

注意：根据 JEDEC DDR2/3/mDDR 规范，系统将保留列地址位 10，用于指示自动预充电，因此没有源地址位可以映射到列地址位 10。

在 LPDDR2/LPDDR3 中，CA 总线中有一个专用于自动预充电的位，因此将使用列位 10。

在内联 ECC 配置 (MEMC_INLINE_ECC=1) 中且使能了 ECC (ECCCFG0.ecc_mode>0) 的情况下，必须将最高的 3 个列地址位映射到最高的 3 个有效 HIF 地址位。

如果列位 9 是最高的列地址位，则必须将其映射到最高的有效 HIF 地址位。（x = 最高有效 HIF 地址位 - 内部基数）

如果列位 9 是第二高的列地址位，则必须将其映射到第二高的有效 HIF 地址位。（x = 最高有效 HIF 地址位 - 1 - 内部基数）

如果列位 9 是第三高的列地址位，则必须将其映射到第三高的有效 HIF 地址位。（x = 最高有效 HIF 地址位 - 2 - 内部基数）

如果未使用该地址位，请将其设置为 31，然后将此列地址位设置为 0。

编程模式：静态。

位 23:21 保留，必须保持复位值。

位 20:16 **ADDRMAP_COL_B8[4:0]**：

- 全总线宽度模式：选择用作列地址位 8 的 HIF 地址位。
- 半总线宽度模式：选择用作列地址位 9 的 HIF 地址位。
- 四分之一总线宽度模式：选择用作列地址位 11（LPDDR2/LPDDR3 模式下为 10）的 HIF 地址位。

有效范围：0 到 7、x 以及 31。x 表示内联 ECC 配置中的有效值。

内部基数：8

所选的 HIF 地址位需通过用此位字段的值加上内部基数进行确定。

注意：根据 JEDEC DDR2/3/mDDR 规范，系统将保留列地址位 10，用于指示自动预充电，因此没有源地址位可以映射到列地址位 10。

在 LPDDR2/LPDDR3 中，CA 总线中有一个专用于自动预充电的位，因此将使用列位 10。

在内联 ECC 配置 (MEMC_INLINE_ECC=1) 中且使能了 ECC (ECCCFG0.ecc_mode>0) 的情况下，必须将最高的 3 个列地址位映射到最高的 3 个有效 HIF 地址位。

如果列位 8 是第二高的列地址位，则必须将其映射到第二高的有效 HIF 地址位。（x = 最高有效 HIF 地址位 - 1 - 内部基数）。

如果列位 8 是第三高的列地址位，则必须将其映射到第三高的有效 HIF 地址位。（x = 最高有效 HIF 地址位 - 2 - 内部基数）。

如果未使用该地址位，请将其设置为 31，然后将此列地址位设置为 0。

编程模式：静态。

位 15:13 保留，必须保持复位值。

位 12:8 **ADDRMAP_COL_B7[4:0]**：

- 全总线宽度模式：选择用作列地址位 7 的 HIF 地址位。
- 半总线宽度模式：选择用作列地址位 8 的 HIF 地址位。
- 四分之一总线宽度模式：选择用作列地址位 9 的 HIF 地址位。

有效范围：0 到 7、x 以及 31。x 表示内联 ECC 配置中的有效值。

内部基数：7

所选的 HIF 地址位需通过用此位字段的值加上内部基数进行确定。

在内联 ECC 配置 (MEMC_INLINE_ECC=1) 中且使能了 ECC (ECCCFG0.ecc_mode>0) 的情况下，必须将最高的 3 个列地址位映射到最高的 3 个有效 HIF 地址位。

如果列位 7 是第三高的列地址位，则必须将其映射到第三高的有效 HIF 地址位。（x = 最高有效 HIF 地址位 - 2 - 内部基数）。

如果未使用该地址位，请将其设置为 31，然后将此列地址位设置为 0。

编程模式：静态。

位 7:4 保留，必须保持复位值。

位 3:0 **ADDRMAP_COL_B6[3:0]**：

- 全总线宽度模式：选择用作列地址位 6 的 HIF 地址位。
- 半总线宽度模式：选择用作列地址位 7 的 HIF 地址位。
- 四分之一总线宽度模式：选择用作列地址位 8 的 HIF 地址位。

有效范围：0 到 7 以及 15

内部基数：6

所选的 HIF 地址位需通过用此位字段的值加上内部基数进行确定。

如果未使用该地址位，请将其设置为 15，然后将此列地址位设置为 0。

编程模式：静态

5.7.50 DDRCTRL 地址映射寄存器 4 (DDRCTRL_ADDRMAP4)

DDRCTRL address map register 4

偏移地址：0x210

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	ADDRMAP_COL_B11[4:0]					Res.	Res.	Res.	ADDRMAP_COL_B10[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31:13 保留，必须保持复位值。

位 12:8 **ADDRMAP_COL_B11[4:0]**：

- 全总线宽度模式：选择用作列地址位 13（LPDDR2/LPDDR3 模式下为 11）的 HIF 地址位。
- 半总线宽度模式：未使用。请参见本说明后面的部分，以了解使其成为未使用状态所需设置的值。
- 四分之一总线宽度模式：未使用。请参见本说明后面的部分，以了解使其成为未使用状态所需设置的值。

有效范围：0 到 7、x 以及 31。x 表示内联 ECC 配置中的有效值。

内部基数：11

所选的 HIF 地址位需通过用此位字段的值加上内部基数进行确定。

注意：根据 JEDEC DDR2/3/mDDR 规范，系统将保留列地址位 10，用于指示自动预充电，因此没有源地址位可以映射到列地址位 10。

在 LPDDR2/LPDDR3 中，CA 总线中有一个专用于自动预充电的位，因此将使用列位 10。

在内联 ECC 配置 (MEMC_INLINE_ECC=1) 中且使能了 ECC (ECCCFG0.ecc_mode>0) 的情况下，必须将最高的 3 个列地址位映射到最高的 3 个有效 HIF 地址位。

如果列位 11 是最高的列地址位，则必须将其映射到最高的有效 HIF 地址位。（x = 最高有效 HIF 地址位 - 内部基数）

如果列位 11 是第二高的列地址位，则必须将其映射到第二高的有效 HIF 地址位。（x = 最高有效 HIF 地址位 - 1 - 内部基数）

如果列位 11 是第三高的列地址位，则必须将其映射到第三高的有效 HIF 地址位。（x = 最高有效 HIF 地址位 - 2 - 内部基数）

如果未使用该地址位，请将其设置为 31，然后将此列地址位设置为 0。

编程模式：静态

位 7:5 保留，必须保持复位值。

位 4:0 **ADDRMAP_COL_B10[4:0]**：

- 全总线宽度模式：选择用作列地址位 11（LPDDR2/LPDDR3 模式下为 10）的 HIF 地址位。
- 半总线宽度模式：选择用作列地址位 13（LPDDR2/LPDDR3 模式下为 11）的 HIF 地址位。
- 四分之一总线宽度模式：未使用。请参见本说明后面的部分，以了解使其成为未使用状态所需设置的值。

有效范围：0 到 7、x 以及 31。x 表示内联 ECC 配置中的有效值。

内部基数：10

所选的 HIF 地址位需通过用此位字段的值加上内部基数进行确定。

注意：根据 JEDEC DDR2/3/mDDR 规范，系统将保留列地址位 10，用于指示自动预充电，因此没有源地址位可以映射到列地址位 10。

在 LPDDR2/LPDDR3 中，CA 总线中有一个专用于自动预充电的位，因此将使用列位 10。

在内联 ECC 配置 (MEMC_INLINE_ECC=1) 中且使能了 ECC (ECCCFG0.ecc_mode>0) 的情况下，必须将最高的 3 个列地址位映射到最高的 3 个有效 HIF 地址位。

如果列位 10 是最高的列地址位，则必须将其映射到最高的有效 HIF 地址位。（x = 最高有效 HIF 地址位 - 内部基数）

如果列位 10 是第二高的列地址位，则必须将其映射到第二高的有效 HIF 地址位。（x = 最高有效 HIF 地址位 - 1 - 内部基数）

如果列位 10 是第三高的列地址位，则必须将其映射到第三高的有效 HIF 地址位。（x = 最高有效 HIF 地址位 - 2 - 内部基数）

如果未使用该地址位，请将其设置为 31，然后将此列地址位设置为 0。

编程模式：静态

5.7.51 DDRCTRL 地址映射寄存器 5 (DDRCTRL_ADDRMAP5)

DDRCTRL address map register 5

偏移地址：0x214

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	ADDRMAP_ROW_B11[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B2_10[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	ADDRMAP_ROW_B1[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B0[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **ADDRMAP_ROW_B11[3:0]**：选择用作行地址位 11 的 HIF 地址位。
有效范围：0 到 11 以及 15
内部基数：17
所选的 HIF 地址位需通过用此位字段的值加上内部基数进行确定。
如果未使用该地址位，请将其设置为 15，然后将行地址位 11 设置为 0。
编程模式：静态

位 23:20 保留，必须保持复位值。

位 19:16 **ADDRMAP_ROW_B2_10[3:0]**：选择用作行地址位 2 到 10 的 HIF 地址位。
有效范围：0 到 11 以及 15
内部基数：8（用于行地址位 2）、9（用于行地址位 3）、10（用于行地址位 4），依此递增到 16（用于行地址位 10）
每个行地址位的所选 HIF 地址位均需通过用此位字段的值加上内部基数进行确定。
当设置为 15 时，行地址位 2 到 10 的值将由寄存器 ADDRMAP9、ADDRMAP10 和 ADDRMAP11 进行定义。
编程模式：静态

位 15:12 保留，必须保持复位值。

位 11:8 **ADDRMAP_ROW_B1[3:0]**：选择用作行地址位 1 的 HIF 地址位。
有效范围：0 到 11
内部基数：7
每个行地址位的所选 HIF 地址位均需通过用此位字段的值加上内部基数进行确定。
编程模式：静态

位 7:4 保留，必须保持复位值。

位 3:0 **ADDRMAP_ROW_B0[3:0]**：选择用作行地址位 0 的 HIF 地址位。
有效范围：0 到 11
内部基数：6
每个行地址位的所选 HIF 地址位均需通过用此位字段的值加上内部基数进行确定。
编程模式：静态

5.7.52 DDRCTRL 地址寄存器 6 (DDRCTRL_ADDRMAP6)

DDRCTRL address register 6

偏移地址：0x218

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPDDR3_6GB_12GB	Res.	Res.	Res.	ADDRMAP_ROW_B15[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B14[3:0]			
rw				rw	rw	rw	rw					rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	ADDRMAP_ROW_B13[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B12[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31 **LPDDR3_6GB_12GB**：
如果正在使用 LPDDR3 SDRAM 6 Gb 或 12 Gb 设备，请将此参数设置为 1。
- 1 - 正在使用 LPDDR3 SDRAM 6 Gb/12 Gb 设备。所有包含 row[14:13]==2'b11 的地址均会被视为无效
- 0 - 正在使用非 LPDDR3 6 Gb/12 Gb 设备。所有地址均有效
仅存在于配置为支持 LPDDR3 的设计中。
编程模式：静态

位 30:28 保留，必须保持复位值。

位 27:24 **ADDRMAP_ROW_B15[3:0]**：选择用作行地址位 15 的 HIF 地址位。
有效范围：0 到 11 以及 15
内部基数：21
所选的 HIF 地址位需通过用此位字段的值加上内部基数进行确定。
如果未使用该地址位，请将其设置为 15，然后将行地址位 15 设置为 0。
编程模式：静态

位 23:20 保留，必须保持复位值。

位 19:16 **ADDRMAP_ROW_B14[3:0]**：选择用作行地址位 14 的 HIF 地址位。
有效范围：0 到 11 以及 15
内部基数：20
所选的 HIF 地址位需通过用此位字段的值加上内部基数进行确定。
如果未使用该地址位，请将其设置为 15，然后将行地址位 14 设置为 0。
编程模式：静态

位 15:12 保留，必须保持复位值。

位 11:8 **ADDRMAP_ROW_B13[3:0]**：选择用作行地址位 13 的 HIF 地址位。
有效范围：0 到 11 以及 15
内部基数：19
所选的 HIF 地址位需通过用此位字段的值加上内部基数进行确定。
如果未使用该地址位，请将其设置为 15，然后将行地址位 13 设置为 0。
编程模式：静态

位 7:4 保留，必须保持复位值。

位 3:0 **ADDRMAP_ROW_B12[3:0]**：选择用作行地址位 12 的 HIF 地址位。
有效范围：0 到 11 以及 15
内部基数：18
所选的 HIF 地址位需通过用此位字段的值加上内部基数进行确定。
如果未使用该地址位，请将其设置为 15，然后将行地址位 12 设置为 0。
编程模式：静态

5.7.53 DDRCTRL 地址映射寄存器 9 (DDRCTRL_ADDRMAP9)

DDRCTRL address map register 9

偏移地址：0x224

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	ADDRMAP_ROW_B5[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B4[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	ADDRMAP_ROW_B3[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B2[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **ADDRMAP_ROW_B5[3:0]**：选择用作行地址位 5 的 HIF 地址位。
有效范围：0 到 11
内部基数：11
每个行地址位的所选 HIF 地址位均需通过用此位字段的值加上内部基数进行确定。只有当 ADDRMAP5.addrmap_row_b2_10 设置为值 15 时，才会使用此寄存器位字段。
编程模式：静态

位 23:20 保留，必须保持复位值。

位 19:16 **ADDRMAP_ROW_B4[3:0]**：选择用作行地址位 4 的 HIF 地址位。
有效范围：0 到 11
内部基数：10
每个行地址位的所选 HIF 地址位均需通过用此位字段的值加上内部基数进行确定。只有当 ADDRMAP5.addrmap_row_b2_10 设置为值 15 时，才会使用此寄存器位字段。
编程模式：静态

位 15:12 保留，必须保持复位值。

位 11:8 **ADDRMAP_ROW_B3[3:0]**：选择用作行地址位 3 的 HIF 地址位。
有效范围：0 到 11
内部基数：9
每个行地址位的所选 HIF 地址位均需通过用此位字段的值加上内部基数进行确定。只有当 ADDRMAP5.addrmap_row_b2_10 设置为值 15 时，才会使用此寄存器位字段。
编程模式：静态

位 7:4 保留，必须保持复位值。

位 3:0 **ADDRMAP_ROW_B2[3:0]**：选择用作行地址位 2 的 HIF 地址位。
有效范围：0 到 11
内部基数：8
每个行地址位的所选 HIF 地址位均需通过用此位字段的值加上内部基数进行确定。只有当 ADDRMAP5.addrmap_row_b2_10 设置为值 15 时，才会使用此寄存器位字段。
编程模式：静态

5.7.54 DDRCTRL 地址映射寄存器 10 (DDRCTRL_ADDRMAP10)

DDRCTRL address map register 10

偏移地址：0x228

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	ADDRMAP_ROW_B9[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B8[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	ADDRMAP_ROW_B7[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B6[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **ADDRMAP_ROW_B9[3:0]**：选择用作行地址位 9 的 HIF 地址位。
有效范围：0 到 11
内部基数：15
每个行地址位的所选 HIF 地址位均需通过用此位字段的值加上内部基数进行确定。只有当 ADDRMAP5.addrmap_row_b2_10 设置为值 15 时，才会使用此寄存器位字段。
编程模式：静态

位 23:20 保留，必须保持复位值。

位 19:16 **ADDRMAP_ROW_B8[3:0]**：选择用作行地址位 8 的 HIF 地址位。
有效范围：0 到 11
内部基数：14
每个行地址位的所选 HIF 地址位均需通过用此位字段的值加上内部基数进行确定。只有当 ADDRMAP5.addrmap_row_b2_10 设置为值 15 时，才会使用此寄存器位字段。
编程模式：静态

位 15:12 保留，必须保持复位值。

位 11:8 **ADDRMAP_ROW_B7[3:0]**：选择用作行地址位 7 的 HIF 地址位。
有效范围：0 到 11
内部基数：13
每个行地址位的所选 HIF 地址位均需通过用此位字段的值加上内部基数进行确定。只有当 ADDRMAP5.addrmap_row_b2_10 设置为值 15 时，才会使用此寄存器位字段。
编程模式：静态

位 7:4 保留，必须保持复位值。

位 3:0 **ADDRMAP_ROW_B6[3:0]**：选择用作行地址位 6 的 HIF 地址位。
有效范围：0 到 11
内部基数：12
每个行地址位的所选 HIF 地址位均需通过用此位字段的值加上内部基数进行确定。只有当 ADDRMAP5.addrmap_row_b2_10 设置为值 15 时，才会使用此寄存器位字段。
编程模式：静态

5.7.55 DDRCTRL 地址映射寄存器 11 (DDRCTRL_ADDRMAP11)

DDRCTRL address map register 11

偏移地址：0x22C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRMAP_ROW_B10[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **ADDRMAP_ROW_B10[3:0]**：选择用作行地址位 10 的 HIF 地址位。

有效范围：0 到 11

内部基数：16

每个行地址位的所选 HIF 地址位均需通过用此位字段的值加上内部基数进行确定。只有当 ADDRMAP5.addrmap_row_b2_10 设置为值 15 时，才会使用此寄存器位字段。

编程模式：静态

5.7.56 DDRCTRL ODT 配置寄存器 (DDRCTRL_ODTCFG)

DDRCTRL ODT configuration register

偏移地址：0x240

复位值：0x0400 0400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	WR_ODT_HOLD[3:0]				Res.	Res.	Res.	WR_ODT_DELAY[4:0]				
				rw	rw	rw	rw				rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	RD_ODT_HOLD[3:0]				Res.	RD_ODT_DELAY[4:0]					Res.	Res.
				rw	rw	rw	rw		rw	rw	rw	rw	rw		

位 31:28 保留，必须保持复位值。

位 27:24 **WR_ODT_HOLD[3:0]**：
保持写入命令 ODT 的 DFI PHY 时钟周期。支持的最小值为 2。
推荐值：
DDR2：
- BL8：0x5 (DDR2-400/533/667)、0x6 (DDR2-800)、0x7 (DDR2-1066)
- BL4：0x3 (DDR2-400/533/667)、0x4 (DDR2-800)、0x5 (DDR2-1066)
DDR3：
- BL8：0x6
DDR4：
- BL8：5 + WR_PREAMBLE + CRC_MODE
WR_PREAMBLE = 1（1tCK 写入预编码）、2（2tCK 写入预编码）
CRC_MODE = 0（非 CRC 模式）、1（CRC 模式）
LPDDR3：
- BL8：7 + RU(tODTon(max)/tCK)
编程模式：准动态组 1 和组 4

位 23:21 保留，必须保持复位值。

位 20:16 **WR_ODT_DELAY[4:0]**：
发出写入命令到设置与该命令相关 ODT 值之间的延迟，以 DFI PHY 时钟周期为单位。在由 DDRCTRL 驱动 DQS 的整个时间内，ODT 设置必须保持不变。
推荐值：
DDR2：
- CWL + AL - 3 (DDR2-400/533/667)、CWL + AL - 4 (DDR2-800)、CWL + AL - 5 (DDR2-1066)
如果 (CWL + AL - 3 < 0)，则 DDRCTRL 不支持写入操作的 ODT。
DDR3：
- 0x0
DDR4：
- DFITMG1.dfi_t_cmd_lat（用于调整 CAL 模式）
LPDDR3：
- WL - 1 - RU(tODTon(max)/tCK))
编程模式：准动态组 1 和组 4

位 15:12 保留，必须保持复位值。

位 11:8 **RD_ODT_HOLD[3:0]**：
保持读取命令 ODT 的 DFI PHY 时钟周期。支持的最小值为 2。
推荐值：
DDR2：
- BL8：0x6（非 DDR2-1066）、0x7 (DDR2-1066)
- BL4：0x4（非 DDR2-1066）、0x5 (DDR2-1066)
DDR3：
- BL8 - 0x6
DDR4：
- BL8：5 + RD_PREAMBLE
RD_PREAMBLE = 1（1tCK 写入预编码）、2（2tCK 写入预编码）
LPDDR3：
- BL8：5 + RU(tDQSCK(max)/tCK) - RD(tDQSCK(min)/tCK) + RU(tODTon(max)/tCK)
编程模式：准动态组 1 和组 4

位 7 保留，必须保持复位值。

位 6:2 **RD_ODT_DELAY[4:0]**：
发出读取命令到设置与该命令相关 ODT 值之间的延迟，以 DFI PHY 时钟周期为单位。在由 DDRCTRL 驱动 DQS 的整个时间内，ODT 设置必须保持不变。
推荐值：
DDR2：
- CL + AL - 4（非 DDR2-1066）、CL + AL - 5 (DDR2-1066)
如果 (CL + AL - 4 < 0)，则 DDRCTRL 不支持读取操作的 ODT。
DDR3：
- CL - CWL
DDR4：
- CL - CWL - RD_PREAMBLE + WR_PREAMBLE + DFITMG1.dfi_t_cmd_lat（用于调整 CAL 模式）
WR_PREAMBLE = 1（1tCK 写入预编码）、2（2tCK 写入预编码）
RD_PREAMBLE = 1（1tCK 写入预编码）、2（2tCK 写入预编码）
如果 (CL - CWL - RD_PREAMBLE + WR_PREAMBLE) < 0，则 DDRCTRL 不支持读取操作的 ODT。
LPDDR3：
- RL + RD(tDQSCK(min)/tCK) - 1 - RU(tODTon(max)/tCK)
编程模式：准动态组 1 和组 4。

位 1:0 保留，必须保持复位值。

5.7.57 DDRCTRL ODT/Rank 映射寄存器 (DDRCTRL_ODTMAP)

DDRCTRL ODT/Rank map register

偏移地址：0x244

复位值：0x0000 0011

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RANK0_RD_ODT	Res.	Res.	Res.	RANK0_WR_ODT
											rw				rw

位 31:5 保留，必须保持复位值。

位 4 **RANK0_RD_ODT**：指示从 rank 0 读取期间必须打开哪些远程 ODT。
每个 rank 都有一个远程 ODT（位于 SDRAM 中），可通过在此处设置相应的位来将其打开。
Rank 0 由 LSB 控制；rank 1 由 LSB 旁边的位控制，依此类推。
对于每个 rank，请将其位设置为 1 以使能其 ODT。
编程模式：静态

位 3:1 保留，必须保持复位值。

位 0 **RANK0_WR_ODT**：指示对 rank 0 写入期间必须打开哪些远程 ODT。
每个 rank 都有一个远程 ODT（位于 SDRAM 中），可通过在此处设置相应的位来将其打开。
Rank 0 由 LSB 控制；rank 1 由 LSB 旁边的位控制，依此类推。
对于每个 rank，请将其位设置为 1 以使能其 ODT。
编程模式：静态

5.7.58 DDRCTRL 调度器控制寄存器 (DDRCTRL_SCHED)

DDRCTRL scheduler control register

偏移地址：0x250

复位值：0x0000 0805

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	RDWR_IDLE_GAP[6:0]							GO2CRITICAL_HYSTERESIS[7:0]							
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	LPR_NUM_ENTRIES[3:0]				Res.	Res.	Res.	Res.	Res.	PAGECLOSE	PREFER_WRITE	FORCE_LOW_PRI_N
				rw	rw	rw	rw						rw	rw	rw

位 31 保留，必须保持复位值。

位 30:24 **RDWR_IDLE_GAP[6:0]**：当首选事务存储在此数量的时钟周期内为空时，如果可选事务存储为非空状态，则切换到该存储。
读取事务存储（包括高优先级和低优先级）为默认首选事务存储，而写入事务存储为可选存储。
一旦将写入的优先级设置为高于读取，系统将反转上述设置。
0x0 为此寄存器的合法值。当设置为 0x0 时，一旦满足切换条件，事务存储切换就会立即发生。
仅限性能用途
编程模式：静态

位 23:16 **GO2CRITICAL_HYSTERESIS[7:0]**：未使用
编程模式：静态

位 15:12 保留，必须保持复位值。

位 11:8 **LPR_NUM_ENTRIES[3:0]**：低优先级事务存储中的条目数为此值 + 1。
(MEMC_NO_OF_ENTRY - (SCHED.lpr_num_entries + 1)) 为高优先级事务存储的可用条目数。
将此参数设置为最大值后，系统会将所有条目分配至低优先级事务存储。
将此参数设置为 0 后，系统会将 1 个条目分配至低优先级事务存储，并将其余条目分配至高优先级事务存储。
注意：在 ECC 配置中，发出的写入和低优先级读取额度数要比采用非 ECC 配置的情况下少 1。写入和低优先级读取 CAM 中将各保留一个条目，用于存储因单独位错误校正 RMW 操作而产生的 RMW 请求。
编程模式：静态

位 7:3 保留，必须保持复位值。

位 2 **PAGECLOSE**：如果该位为 1，则只有当 CAM 中存在适用于某 bank 的页命中事务时，该 bank 才会保持打开状态。如果 SCHED1.pageclose_timer=0，则 CAM 中具有 bank 和页命中的最后一个读取或写入命令将在进行自动预充电的情况下执行。即使此寄存器设置为 1 且 SCHED1.pageclose_timer 设置为 0，当某些情况下存在写入与读取之间或 LPR 与 HPR 之间的模式切换时，系统也可能发出显式预充电（而非自动预充电）。作为 ECC 擦洗请求的一部分执行的读取和写入命令也会在没有自动预充电的情况下执行。

如果该位为 0，则 bank 将保持打开状态，直到（为了打开不同的页，或是因页超时或刷新超时而导致）需要将其关闭为止——我们也可称之为打开页策略。用户可以在 HIF 接口上设置逐命令自动预充电位 (hif_cmd_autopre)，以覆盖打开页策略。

页关闭功能提供了一种介于打开页策略与关闭页策略之间的折中方案。

仅限性能用途。

编程模式：准动态组 3

位 1 **PREFER_WRITE**：如果置 1，则 bank 选择器会将写入的优先级视为高于读取。

仅限调试用途。

编程模式：静态

位 0 **FORCE_LOW_PRI_N**：低电平有效信号。当设置为生效（“0”）时，所有传入的事务都会被强制设置为低优先级。这意味着所有高优先级读取 (HPR) 和可变优先级读取命令 (VPR) 均会被视为低优先级读取 (LPR) 命令。在写入方面，所有可变优先级写入 (VPW) 命令均会被视为正常优先级写入 (NPW) 命令。将传入的事务强制设置为低优先级将隐式关闭读取命令的旁路路径。

仅限性能用途。

编程模式：静态

5.7.59 DDRCTRL 调度器控制寄存器 1 (DDRCTRL_SCHED1)

DDRCTRL scheduler control register 1

偏移地址：0x254

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PAGECLOSE_TIMER[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **PAGECLOSE_TIMER[7:0]**：此位字段与 SCHED.pageclose 结合使用。

仅当 SCHED.pageclose==1 时，此位字段才有意义。

如果 SCHED.pageclose==1 且 pageclose_timer==0，则可以为 CAM 中具有 bank 和页命中的最后一个读取或写入命令安排自动预充电。

请注意，系统有时会安排显式预充电而非自动预充电。有关可能出现这一状况的具体时机，请参见 SCHED.pageclose。

如果 SCHED.pageclose==1 且 pageclose_timer>0，则不会为 CAM 中具有 bank 和页命中的最后一个读取或写入命令安排自动预充电。

相反，系统会启动一个定时器，并将 pageclose_timer 作为初始值。

每个 bank 都有一个定时器。

除非 CAM 中面向 bank 的下一次读取或写入为页命中，否则定时器的值会递减。

如果 CAM 中面向 bank 的下一次读取或写入为页命中，则该值会复位为 pageclose_timer 值。

一旦定时器达到零，系统将尝试安排显式预充电。

编程模式：静态。

5.7.60 DDRCTRL 高优先级读取 CAM 寄存器 1 (DDRCTRL_PERFHPR1)

DDRCTRL high priority read CAM register 1

偏移地址：0x25C

复位值：0x0F00 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HPR_XACT_RUN_LENGTH[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HPR_MAX_STARVE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **HPR_XACT_RUN_LENGTH[7:0]**：

当 HPR 队列进入关键状态时，所处理的事务数为以下各项中的较小值：

- (a) 此数量
- (b) 可用事务数。

单位：事务。

仅限性能用途。

编程模式：准动态组 3

位 23:16 保留，必须保持复位值。

位 15:0 **HPR_MAX_STARVE[15:0]**：HPR 队列在进入关键状态前，可承受饥饿的 DFI 时钟周期数。此寄存器的最小有效功能值为 0x1。将其设置为 0x0 将禁止饥饿功能；在正常操作期间不得禁止此功能，原因是这会导致延迟过大。

仅限性能用途。

编程模式：准动态组 3

5.7.61 DDRCTRL 低优先级读取 CAM 寄存器 1 (DDRCTRL_PERFLPR1)

DDRCTRL low priority read CAM register 1

偏移地址：0x264

复位值：0x0F00 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPR_XACT_RUN_LENGTH[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPR_MAX_STARVE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **LPR_XACT_RUN_LENGTH[7:0]**：

当 LPR 队列进入关键状态时，所处理的事务数为以下各项中的较小值：

- (a) 此数量
- (b) 可用事务数。

单位：事务。

仅限性能用途

编程模式：准动态组 3

位 23:16 保留，必须保持复位值。

位 15:0 **LPR_MAX_STARVE[15:0]**：LPR 队列在进入关键状态前，可承受饥饿的 DFI 时钟周期数。此寄存器的最小有效功能值为 0x1。将其设置为 0x0 将禁止饥饿功能；在正常操作期间不得禁止此功能，原因是这会导致延迟过大。

仅限性能用途

编程模式：准动态组 3

5.7.62 DDRCTRL 写入 CAM 寄存器 1 (DDRCTRL_PERFWR1)

DDRCTRL write CAM register 1

偏移地址：0x26C

复位值：0x0F00 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
W_XACT_RUN_LENGTH[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W_MAX_STARVE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **W_XACT_RUN_LENGTH[7:0]**：
当 WR 队列进入关键状态时，所处理的事务数为以下各项中的较小值：
- (a) 此数量
- (b) 可用事务数。
单位：事务。
仅限性能用途。
编程模式：准动态组 3。

位 23:16 保留，必须保持复位值。

位 15:0 **W_MAX_STARVE[15:0]**：WR 队列在进入关键状态前，可承受饥饿的 DFI 时钟周期数。此寄存器的最小有效功能值为 0x1。将其设置为 0x0 将禁止饥饿功能；在正常操作期间不得禁止此功能，原因是这会导致延迟过大。
仅限性能用途。
编程模式：准动态组 3。

5.7.63 DDRCTRL 调试寄存器 0 (DDRCTRL_DBG0)

DDRCTRL debug register 0

偏移地址：0x300

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIS_COLLISION_PAGE_OPT	Res.	Res.	Res.	DIS_WC
											rw				rw

位 31:5 保留，必须保持复位值。

位 4 **DIS_COLLISION_PAGE_OPT**：当此参数设置为“0”时，系统将针对冲突情况下的清空命令禁止自动预充电。冲突情况包括写入后跟对同一地址的读取、读取后跟对同一地址的写入，或是写入后跟对同一地址的写入，且 DBG0.dis_wc 位 = 1（同一地址比较不包括表示关键字的两个地址位）。
仅限调试用途。
编程模式：静态

位 3:1 保留，必须保持复位值。

位 0 **DIS_WC**：当设置为 1 时，将禁止写入合并。
仅限调试用途
编程模式：静态

5.7.64 DDRCTRL 调试寄存器 1 (DDRCTRL_DBG1)

DDRCTRL debug register 1

偏移地址：0x304

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIS_HIF	DIS_DQ
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **DIS_HIF**：当设置为 1 时，DDRCTRL 会发出 HIF 命令信号 hif_cmd_stall。DDRCTRL 将忽略 hif_cmd_valid 以及所有其他相关的请求信号。
此位应支持实时切换。
编程模式：动态

位 0 **DIS_DQ**：当设置为 1 时，DDRCTRL 不会使 CAM 中的任何事务出队。旁路也将受到禁止。所有事务都会在 CAM 中排队。只要此设置生效，系统就不会向 SDRAM 发出任何读取或写入请求。
此位可用于防止 DDRCTRL 发出读取或写入请求，从而能够安全地修改某些与读取和写入相关的寄存器位字段（详细信息请参见用户指南）。设置此位后，强烈建议在对任何影响读取和写入的寄存器执行更改之前，轮询 DBGCAM.wr_data_pipeline_empty 和 DBGCAM.rd_data_pipeline_empty。这样可确保 DDRC 中的相关逻辑处于空闲状态。
此位应支持实时切换。
编程模式：动态

5.7.65 DDRCTRL CAM 调试寄存器 (DDRCTRL_DBGCAM)

DDRCTRL CAM debug register

偏移地址：0x308

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	WR_DATA_PIPELINE_EMPTY	RD_DATA_PIPELINE_EMPTY	Res.	DBG_WR_Q_EMPTY	DBG_RD_Q_EMPTY	DBG_STALL	Res.	Res.	Res.	DBG_W_Q_DEPTH[4:0]				
		r	r		r	r	r				r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBG_LPR_Q_DEPTH[4:0]					Res.	Res.	Res.	DBG_HPR_Q_DEPTH[4:0]				
			r	r	r	r	r				r	r	r	r	r

位 31:30 保留，必须保持复位值。

位 29 **WR_DATA_PIPELINE_EMPTY**：此位指示 DFI 接口上的写入数据管道为空。设置 DBG1.dis_dq 后，应至少对此寄存器进行两次轮询，以确保所有剩余命令/数据均已完成。
编程模式：动态。

位 28 **RD_DATA_PIPELINE_EMPTY**：此位指示 DFI 接口上的读取数据管道为空。设置 DBG1.dis_dq 后，应至少对此寄存器进行两次轮询，以确保所有剩余命令/数据均已完成。
编程模式：动态。

位 27 保留，必须保持复位值。

位 26 **DBG_WR_Q_EMPTY**：当设置为 1 时，DDRC 内部的所有写入命令队列和写入数据缓冲区均为空。该寄存器将用于调试目的。
以下为用例场景示例：当控制器使用低功耗进入序列进入自刷新模式时，预计控制器已执行了队列中的所有命令，并且写入和读取数据已被清空。因此，此时该寄存器的值应为 1。
仅限调试用途。
编程模式：动态。

位 25 **DBG_RD_Q_EMPTY**：当设置为 1 时，DDRC 内部的所有读取命令队列和读取数据缓冲区均为空。该寄存器将用于调试目的。
以下为用例场景示例：当控制器使用低功耗进入序列进入自刷新模式时，预计控制器已执行了队列中的所有命令，并且写入和读取数据已被清空。因此，此时该寄存器的值应为 1。
仅限调试用途。
编程模式：动态

位 24 **DBG_STALL**：Stall
仅限调试用途。
编程模式：动态

位 23:21 保留，必须保持复位值。

位 20:16 **DBG_W_Q_DEPTH[4:0]**：写入队列深度
WR 队列的最后一个条目将保留给 ECC 擦洗操作。此条目不会包含在队列深度计算中。
仅限调试用途。
编程模式：动态

位 15:13 保留，必须保持复位值。

位 12:8 **DBG_LPR_Q_DEPTH[4:0]**：低优先级读取队列深度
Lpr 队列的最后一个条目将保留给 ECC 擦洗操作。此条目不会包含在队列深度计算中。
仅限调试用途。
编程模式：动态

位 7:5 保留，必须保持复位值。

位 4:0 **DBG_HPR_Q_DEPTH[4:0]**：高优先级读取队列深度
仅限调试用途。
编程模式：动态

5.7.66 DDRCTRL 命令调试寄存器 (DDRCTRL_DBGCMD)

DDRCTRL command debug register

偏移地址：0x30C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTRLUPD	ZQ_CALIB_SHORT	Res.	Res.	Res.	RANK0_REFRESH
										rs	rs				rs

位 31:6 保留，必须保持复位值。

位 5 **CTRLUPD**：

将此寄存器位设置为 1 将指示 DDRCTRL 向 PHY 发出 dfi_ctrlupd_req。当此请求存储在 DDRCTRL 中时，该位将自动清零。此操作只能在 DFIUPD0.dis_auto_ctrlupd=1 时执行。

编程模式：动态。

位 4 **ZQ_CALIB_SHORT**：

将此寄存器位设置为 1 将指示 DDRCTRL 向 SDRAM 发出 ZQCS（ZQ 短校准）/MPC（ZQ 校准）命令。当此请求存储在 DDRCTRL 中时，该位将自动清零。此操作只能在 ZQCTL0.dis_auto_zq=1 时执行。如果处于初始化、自刷新（LPDDR4 除外）、SR 掉电 (LPDDR4)、深度掉电工作模式或最大程度节能模式下，建议不要设置此寄存器位。

对于自刷新（LPDDR4 除外）或 SR 掉电 (LPDDR4)，系统会安排在退出 SR（LPDDR4 除外）或 SPRD (LPDDR4) 模式后进行该操作。

对于深度掉电和最大程度节能模式，尽管系统会将 DBGSTAT.zq_calib_short_busy 清零，但不会安排进行该操作。

编程模式：动态。

位 3:1 保留，必须保持复位值。

位 0 **RANK0_REFRESH**：

将此寄存器位设置为 1 将指示 DDRCTRL 向 rank 0 发出刷新。向此位写入会导致系统设置 DBGSTAT.rank0_refresh_busy。当 DBGSTAT.rank0_refresh_busy 清零时，该命令已存储在 DDRCTRL 中。

对于 3DS 配置，系统会将刷新发送到 rank 索引 0。

此操作只能在 RFSHCTL3.dis_auto_refresh=1 时执行。如果处于初始化、深度掉电工作模式或最大程度节能模式下，建议不要设置此寄存器位。

编程模式：动态。

5.7.67 DDRCTRL 状态调试寄存器 (DDRCTRL_DBGSTAT)

DDRCTRL status debug register

偏移地址：0x310

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTRLUPD_BUSY	ZQ_CALIB_SHORT_BUSY	Res.	Res.	Res.	RANK0_REFRESH_BUSY
										r	r				r

位 31:6 保留，必须保持复位值。

位 5 **CTRLUPD_BUSY**：仅当此信号为低电平时，SoC 内核才能启动 ctrlupd 操作。在 DDRCTRL 接受 ctrlupd 请求后的那个时钟周期，此信号会变为高电平。当在 DDRCTRL 中启动 ctrlupd 操作时，该信号会变为低电平。建议在此信号为高电平时不要执行 ctrlupd 操作。

- 0 - 表示 SoC 内核可以启动 ctrlupd 操作
- 1 - 表示 DDRCTRL 中尚未启动 ctrlupd 操作

编程模式：动态

位 4 **ZQ_CALIB_SHORT_BUSY**：仅当此信号为低电平时，SoC 内核才能启动 ZQCS（ZQ 短校准）操作。在 DDRCTRL 接受 ZQCS 请求后的那个时钟周期，此信号会变为高电平。当在 DDRCTRL 中启动 ZQCS 操作时，该信号会变为低电平。建议在此信号为高电平时不要执行 ZQCS 操作。

- 0 - 表示 SoC 内核可以启动 ZQCS 操作
- 1 - 表示 DDRCTRL 中尚未启动 ZQCS 操作

编程模式：动态

位 3:1 保留，必须保持复位值。

位 0 **RANK0_REFRESH_BUSY**：仅当此信号为低电平时，SoC 内核才能启动 rank0_refresh 操作（面向 rank 0 的刷新操作）。在将 DBGCMD.rank0_refresh 设置为 1 后的那个时钟周期，此信号会变为高电平。当 rank0_refresh 操作存储在 DDRCTRL 中时，该信号会变为低电平。建议在此信号为高电平时不要执行 rank0_refresh 操作。

- 0 - 表示 SoC 内核可以启动 rank0_refresh 操作
- 1 - 表示 DDRCTRL 中尚未存储 rank0_refresh 操作

编程模式：动态

5.7.68 DDRCTRL 软件寄存器编程控制使能 (DDRCTRL_SWCTL)

DDRCTRL software register programming control enable

偏移地址：0x320

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SW_DONE
															rw

位 31:1 保留，必须保持复位值。

位 0 **SW_DONE**：在复位之外使能准动态寄存器编程。将寄存器设置为 0 以使能准动态编程。一旦编程完成，将寄存器重新置 1。
编程模式：动态

5.7.69 DDRCTRL 软件寄存器编程控制状态 (DDRCTRL_SWSTAT)

DDRCTRL software register programming control status

偏移地址：0x324

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SW_DONE_ACK
															r

位 31:1 保留，必须保持复位值。

位 0 **SW_DONE_ACK**：寄存器编程完成。此寄存器是 SWCTL.sw_done 的回响。请等待 sw_done 值 1 在编程序列结束时传播到 sw_done_ack，以确保将正确的寄存器值传播到目标时钟域。
编程模式：静态

5.7.70 DDRCTRL AXI Poison配置寄存器 (DDRCTRL_POISONCFG)

DDRCTRL AXI Poison configuration register

偏移地址：0x36C

复位值：0x0011 0011

AXI Poison 配置寄存器适用于所有 AXI 端口。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RD_POISON_INTR_CLR	Res.	Res.	Res.	RD_POISON_INTR_EN	Res.	Res.	Res.	RD_POISON_SLVERR_EN
							rc_w1				rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR_POISON_INTR_CLR	Res.	Res.	Res.	WR_POISON_INTR_EN	Res.	Res.	Res.	WR_POISON_SLVERR_EN
							rc_w1				rw				rw

位 31:25 保留，必须保持复位值。

位 24 **RD_POISON_INTR_CLR**：读取事务 Poison 的中断清零。允许利用 2/3 个时钟周期将正确值传播至核心逻辑，并将中断清零。
编程模式：动态

位 23:21 保留，必须保持复位值。

位 20 **RD_POISON_INTR_EN**：如果设置为 1，则会使能读取事务 Poison 的中断
编程模式：动态

位 19:17 保留，必须保持复位值。

位 16 **RD_POISON_SLVERR_EN**：如果设置为 1，则会使能读取事务 Poison 的 SLVERR 响应
编程模式：动态

位 15:9 保留，必须保持复位值。

位 8 **WR_POISON_INTR_CLR**：写入事务 Poison 的中断清零。允许利用 2/3 个时钟周期将正确值传播至核心逻辑，并将中断清零。
编程模式：动态

位 7:5 保留，必须保持复位值。

位 4 **WR_POISON_INTR_EN**：如果设置为 1，则会使能写入事务 Poison 的中断
编程模式：动态

位 3:1 保留，必须保持复位值。

位 0 **WR_POISON_SLVERR_EN**：如果设置为 1，则会使能写入事务 Poison 的 SLVERR 响应
编程模式：动态

5.7.71 DDRCTRL AXI Poison 状态寄存器 (DDRCTRL_POISONSTAT)

DDRCTRL AXI Poison status register

偏移地址：0x370

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RD_POISON_INTR_0
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR_POISON_INTR_0
															r

位 31:17 保留，必须保持复位值。

位 16 **RD_POISON_INTR_0**：端口 0 的读取事务 Poison 错误中断。此寄存器是当事务在相应 AXI 端口的读取地址通道上被 Poison 时生效的中断的 APB 时钟副本（双寄存器同步器）。位 0 与端口 0 相对应，依此类推。中断将由寄存器 rd_poison_intr_clr 进行清零，然后将值传播到 APB 时钟。
编程模式：动态

位 15:1 保留，必须保持复位值。

位 0 **WR_POISON_INTR_0**：端口 0 的写入事务 Poison 错误中断。此寄存器是当事务在相应 AXI 端口的写入地址通道上被 Poison 时生效的中断的 APB 时钟副本（双寄存器同步器）。位 0 与端口 0 相对应，依此类推。中断将由寄存器 wr_poison_intr_clr 进行清零，然后将值传播到 APB 时钟。
编程模式：动态

5.7.72 DDRCTRL 端口状态寄存器 (DDRCTRL_PSTAT)

DDRCTRL port status register

偏移地址：0x3FC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR_PORT_BUSY_0
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RD_PORT_BUSY_0
															r

位 31:17 保留，必须保持复位值。

位 16 **WR_PORT_BUSY_0**：指示 AXI 端口 0 是否有未完成的写入。
编程模式：动态

位 15:1 保留，必须保持复位值。

位 0 **RD_PORT_BUSY_0**：指示 AXI 端口 0 是否有未完成的读取。
编程模式：动态

5.7.73 DDRCTRL 端口通用配置寄存器 (DDRCTRL_PCCFG)

DDRCTRL port common configuration register

偏移地址：0x400

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	BL_EXP_MODE	Res.	Res.	Res.	PAGEMATCH_LIMIT	Res.	Res.	Res.	GO2CRITICAL_EN
							rw				rw				rw

位 31:9 保留，必须保持复位值。

位 8 **BL_EXP_MODE**：突发长度扩展模式。默认情况下（即 bl_exp_mode==0），XPI 会将每个 AXI 突发扩展为多个 HIF 命令，并使用内存突发长度作为单位。如果设置为 1，则 XPI 将使用内存突发长度的一半作为单位。

该设置同时适用于读取和写入。当 MSTR.data_bus_width==00 时，将 bl_exp_mode 设置为 1 不会产生影响。

可将其用于使能了部分写入 (UMCTL2_PARTIAL_WR=1) 的情况，以避免或最大限度减少在 DDR4 中使用 t_ccd_l 的代价以及在 LPDDR4 中使用 t_ccd_mw 的代价。因此，仅推荐在 DDR4 或 LPDDR4 模式下使用 bl_exp_mode=1。

请注意，如果 DBICTL.dm_en=0，则以下情况下将不支持该功能：

- UMCTL2_PARTIAL_WR=0
- UMCTL2_PARTIAL_WR=1，MSTR.data_bus_width=01，MEMC_BURST_LENGTH=8 且 MSTR.burst_rdwr=1000（仅限 LPDDR4）
- UMCTL2_PARTIAL_WR=1，MSTR.data_bus_width=01，MEMC_BURST_LENGTH=4 且 MSTR.burst_rdwr=0100（仅限 DDR4），同时 MSTR.burstchop=0 或 CRCPARCTL1.crc_enable=1

如果使能了数据通道交错，则系统也不会支持该功能。

编程模式：静态。

位 7:5 保留，必须保持复位值。

位 4 **PAGEMATCH_LIMIT**：页匹配存在四项限制。如果设置为 1，则当使能了页匹配功能时，系统会将端口仲裁器可授予的连续同一页 DDRC 事务数限制为 4。如果设置为 0，则系统将不对连续同一页 DDRC 事务数施加限制。

编程模式：静态。

位 3:1 保留，必须保持复位值。

位 0 **GO2CRITICAL_EN**：如果设置为 1（使能），系统将根据来自 AXI 主设备的紧急输入（awurgent、arurgent），设置进入 DDRC 的 co_gs_go2critical_wr 和 co_gs_go2critical_lpr/co_gs_go2critical_hpr 信号。如果设置为 0（禁止），系统会将 DDRC 上的 co_gs_go2critical_wr 和 co_gs_go2critical_lpr/co_gs_go2critical_hpr 信号驱动至 1b'0。

编程模式：静态。

5.7.74 DDRCTRL 端口 x 配置读取寄存器 (DDRCTRL_PCFGR_0)

DDRCTRL port x configuration read register

偏移地址：0x404

复位值：0x0000 4000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDWR_ORDERED_EN
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RD_PORT_PAGEMATCH_EN	RD_PORT_URGENT_EN	RD_PORT_AGING_EN	Res.	Res.	RD_PORT_PRIORITY[9:0]									
	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **RDWR_ORDERED_EN**：使能有序读取/写入。如果设置为 1，系统将保留在特定端口对同一地址发出的读取事务与写入事务之间的顺序。换句话说，控制器将确保来自应用端口接口的所有同一地址的读取和写入命令都按照接受的顺序传输到 DFI 接口。当主设备具有软件一致性要求，以便在不等待写入/读取响应的情况下发出连续读/写事务时，此功能非常有用。请注意，此寄存器仅在必要逻辑通过 UMCTL2_RDWR_ORDERED_n 参数进行了实例化时才有效。

编程模式：静态。

位 15 保留，必须保持复位值。

位 14 **RD_PORT_PAGEMATCH_EN**：如果设置为 1，则会使能页匹配功能。使能后，一旦请求端口获得了授权，如果下一个立即命令所面向的是同一内存页（相同的 bank 和行），则该端口将继续获得授权。另请参见相关的 PCCFG.pagematch_limit 寄存器。

编程模式：静态

位 13 **RD_PORT_URGENT_EN**：如果设置为 1，则会使能 AXI 紧急边带信号 (arurgent)。将其使能后，一旦主设备将 arurgent 置为有效，该端口将变为最高优先级；如果在 PCCFG.go2critical_en 寄存器中将其使能，则系统会将发送给 DDRC 的 co_gs_go2critical_lpr/co_gs_go2critical_hpr 信号置为有效。请注意，系统可以随时根据需要，将 arurgent 信号置为有效，且该操作不受地址握手情况的影响（与任何特定命令均无关联）。

编程模式：静态

位 12 **RD_PORT_AGING_EN**：如果设置为 1，则会为端口的读取通道使能老化功能。

编程模式：静态

位 11:10 保留，必须保持复位值。

位 9:0 **RD_PORT_PRIORITY[9:0]**：确定读取老化计数器的初始加载值。这些计数器会在复位后或每次授权至相应端口后并行加载。在端口正在请求授权但未获授权的每个时钟周期内，老化计数器会进行倒数。读取老化计数器的 5 个较高有效位将决定给定端口读取通道的优先级。随着计数器的 5 个较高有效位开始递减，端口的优先级将提升。当老化计数器变为 0 时，相应的端口通道将具有最高优先级（超时条件 - 优先级 0）。对于多端口配置，在使能外部动态优先级输入 (arqos) 时，无法使用老化计数器来设置端口优先级（超时仍适用）。对于单端口配置，系统仅会在老化计数器超时（变为 0）时使用该计数器，以强制进行读取-写入方向切换。在这种情况下，仍可使用外部动态优先级输入 arqos（仅限读取）逐个命令设置 DDRC 读取优先级（2 个优先级：低优先级读取 - LPR、高优先级读取 - HPR）。

注意：此寄存器位字段的两个 LSB 在内部连接到 2'b00。

编程模式：静态

5.7.75 DDRCTRL 端口 x 配置写入寄存器 (DDRCTRL_PCFGW_0)

DDRCTRL port x configuration write register

偏移地址：0x408

复位值：0x0000 4000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	WR_PORT_PAGEMATCH_EN	WR_PORT_URGENT_EN	WR_PORT_AGING_EN	Res.	Res.	WR_PORT_PRIORITY[9:0]									
	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:15 保留，必须保持复位值。

位 14 **WR_PORT_PAGEMATCH_EN**：如果设置为 1，则会使能页匹配功能。使能后，一旦请求端口获得了授权，如果下一个立即命令所面向的是同一内存页（相同的 bank 和行），则该端口将继续获得授权。另请参见相关的 PCCFG.pagematch_limit 寄存器。

编程模式：静态

位 13 **WR_PORT_URGENT_EN**：如果设置为 1，系统将使能 AXI 紧急边带信号 (awurgent)。将其使能后，一旦主设备将 awurgent 置为有效，该端口将变为最高优先级；如果在 PCCFG.go2critical_en 寄存器中将其使能，则系统会将发送给 DDRC 的 co_gs_go2critical_wr 信号置为有效。

请注意，系统可以随时根据需要，将 awurgent 信号置为有效，且该操作不受地址握手情况的影响（与任何特定命令均无关联）。

编程模式：静态

位 12 **WR_PORT_AGING_EN**：如果设置为 1，则会为端口的写入通道使能老化功能。

编程模式：静态

位 11:10 保留，必须保持复位值。

位 9:0 **WR_PORT_PRIORITY[9:0]**：确定写入老化计数器的初始加载值。这些计数器会在复位后或每次授权至相应端口后并行加载。在端口正在请求授权但未获授权的每个时钟周期内，老化计数器会进行倒数。写入老化计数器的 5 个较高有效位将决定给定端口写入通道的初始优先级。随着计数器的 5 个较高有效位开始递减，端口的优先级将提升。当老化计数器变为 0 时，相应的端口通道将具有最高优先级。

对于多端口配置，在使能外部动态优先级输入 (awqos) 时，无法使用老化计数器来设置端口优先级（超时仍适用）。

对于单端口配置，系统仅会在老化计数器超时（变为 0）时使用该计数器，以强制进行读取-写入方向切换。

注意：此寄存器位字段的两个 LSB 在内部连接到 2'b00。

编程模式：静态

5.7.76 DDRCTRL 端口 x 控制寄存器 (DDRCTRL_PCTRL_0)

DDRCTRL port x control register

偏移地址：0x490

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PORT_EN
															rw

位 31:1 保留，必须保持复位值。

位 0 **PORT_EN**：使能 AXI 端口 n。

编程模式：动态

5.7.77 DDRCTRL 端口 x 读取 Q0S 配置寄存器 0 (DDRCTRL_PCFGQOS0_0)

DDRCTRL port x read Q0S configuration register 0

偏移地址：0x494

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RQOS_MAP_REGION1[1:0]		Res.	Res.	RQOS_MAP_REGION0[1:0]	
										rw	rw			rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RQOS_MAP_LEVEL1[3:0]			
												rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:20 **RQOS_MAP_REGION1[1:0]**：该位字段指示区域 1 的数据传输类。
有效值为：
- 0 : LPR，
- 1: VPR，
- 2: HPR。

对于双地址队列配置，区域 1 将映射到蓝色地址队列。
在这种情况下，有效值为
- 0: LPR 和
- 1: 仅 VPR。

当禁止 VPR 支持 (UMCTL2_VPR_EN = 0) 并将区域 1 的数据传输类设置为 1 (VPR) 时，系统会将 VPR 数据传输另外命名为 LPR 数据传输。
编程模式：准动态组 3

位 19:18 保留，必须保持复位值。

位 17:16 **RQOS_MAP_REGION0[1:0]**：该位字段指示区域 0 的数据传输类。
有效值为：
- 0: LPR，1：VPR，2：HPR。

对于双地址队列配置，区域 0 将映射到蓝色地址队列。
在这种情况下，有效值为：
- 0: LPR 和 1：仅 VPR。

当禁止 VPR 支持 (UMCTL2_VPR_EN = 0) 并将区域 0 的数据传输类设置为 1 (VPR) 时，系统会将 VPR 数据传输另外命名为 LPR 数据传输。
编程模式：准动态组 3

位 15:4 保留，必须保持复位值。

位 3:0 **RQOS_MAP_LEVEL1[3:0]**：分隔级别 1 用于指示区域 0 映射的结束；区域 0 的开始为 0。级别 1 的可能值范围是 0 到 13（适用于双 RAQ）或 0 到 14（适用于单 RAQ），与 arqos 相对应。
请注意，对于 PA，arqos 值直接用作端口优先级；其数值越高，端口优先级越高。
必须将所有 map_level* 寄存器设置为不同的值。
编程模式：准动态组 3

5.7.78 DDRCTRL 端口 x 读取 Q0S 配置寄存器 1 (DDRCTRL_PCFGQOS1_0)

DDRCTRL port x read Q0S configuration register 1

偏移地址：0x498

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	RQOS_MAP_TIMEOUTR[10:0]										
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	RQOS_MAP_TIMEOUTB[10:0]										
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:27 保留，必须保持复位值。

位 26:16 **RQOS_MAP_TIMEOUTR[10:0]**：指定映射到红色地址队列的事务的超时值。在单个读取地址队列中，不存在红色队列，且此值无效。
编程模式：准动态组 3

位 15:11 保留，必须保持复位值。

位 10:0 **RQOS_MAP_TIMEOUTB[10:0]**：指定映射到蓝色地址队列的事务的超时值。
编程模式：准动态组 3。

5.7.79 DDRCTRL 端口 x 写入 Q0S 配置寄存器 0 (DDRCTRL_PCFGWQOS0_0)

DDRCTRL port x write Q0S configuration register 0

偏移地址：0x49C

复位值：0x0000 0E00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	WQOS_MAP_REGION2[1:0]		Res.	Res.	WQOS_MAP_REGION1[1:0]		Res.	Res.	WQOS_MAP_REGION0[1:0]	
						rw	rw			rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	WQOS_MAP_LEVEL2[3:0]				Res.	Res.	Res.	Res.	WQOS_MAP_LEVEL1[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25:24 **WQOS_MAP_REGION2[1:0]**：该位字段指示区域 2 的数据传输类。
有效值为：
0: NPW，1：VPW。
当禁止 VPW 支持 (UMCTL2_VPW_EN = 0) 并将区域 2 的数据传输类设置为 1 (VPW) 时，系统会将 VPW 数据传输另外命名为 NPW 数据传输。
编程模式：准动态组 3

位 23:22 保留，必须保持复位值。

位 21:20 **WQOS_MAP_REGION1[1:0]**：该位字段指示区域 1 的数据传输类。
有效值为：
0: NPW，1：VPW。
当禁止 VPW 支持 (UMCTL2_VPW_EN = 0) 并将区域 1 的数据传输类设置为 1 (VPW) 时，系统会将 VPW 数据传输另外命名为 NPW 数据传输。
编程模式：准动态组 3

位 19:18 保留，必须保持复位值。

位 17:16 **WQOS_MAP_REGION0[1:0]**：该位字段指示区域 0 的数据传输类。
有效值为：
0: NPW，1：VPW。
当禁止 VPW 支持 (UMCTL2_VPW_EN = 0) 并将区域 0 的数据传输类设置为 1 (VPW) 时，系统会将 VPW 数据传输另外命名为 NPW 数据传输。
编程模式：准动态组 3

位 15:12 保留，必须保持复位值。

位 11:8 **WQOS_MAP_LEVEL2[3:0]**：分隔级别 2 用于指示区域 1 映射的结束；区域 1 的开始为（级别 1 + 1）。级别 2 的可能值范围是（级别 1 + 1）到 14，与 awqos 相对应。
区域 2 从（级别 2 + 1）开始，到 15 结束。
请注意，对于 PA，awqos 值直接用作端口优先级；其数值越高，端口优先级越高。
必须将所有 map_level* 寄存器设置为不同的值。
编程模式：准动态组 3

位 7:4 保留，必须保持复位值。

位 3:0 **WQOS_MAP_LEVEL1[3:0]**：分隔级别用于指示区域 0 映射的结束；区域 0 的开始为 0。级别 1 的可能值范围是 0 到 13，与 awqos 相对应。
请注意，对于 PA，awqos 值直接用作端口优先级；其数值越高，端口优先级越高。
必须将所有 map_level* 寄存器设置为不同的值。
编程模式：准动态组 3

5.7.80 DDRCTRL 端口 x 写入 Q0S 配置寄存器 1 (DDRCTRL_PCFGWQOS1_0)

DDRCTRL port x write Q0S configuration register 1

偏移地址：0x4A0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	WQOS_MAP_TIMEOUT2[10:0]										
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	WQOS_MAP_TIMEOUT1[10:0]										
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:27 保留，必须保持复位值。

位 26:16 **WQOS_MAP_TIMEOUT2[10:0]**：指定区域 2 中写入事务的超时值。
编程模式：准动态组 3。

位 15:11 保留，必须保持复位值。

位 10:0 **WQOS_MAP_TIMEOUT1[10:0]**：指定区域 0 和 1 中写入事务的超时值。
编程模式：准动态组 3。

5.7.81 DDRCTRL 寄存器总结

表 19. DDRCTRL 寄存器映射和复位值

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	DDRCTRL_MSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BURST_RDWR[3:0]				DLL_OFF_MODE	Res.	DATA_BUS_WIDTH[1:0]		Res.	EN_2T_TIMING_MODE	BURSTCHOP	Res.	Res.	Res.	Res.	Res.	LPDDR3	LPDDR2	Res.	DDR3
	Reset value													0	1	0	0	0		0	0		0	0						0	0		1
0x004	DDRCTRL_STAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELFREF_CAM_NOT_EMPTY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELFREF_TYPE[1:0]		Res.	OPERATING_MODE[2:0]		
	Reset value																			0								0	0		0	0	0
0x008 - 0x00C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x010	DDRCTRL_MRCTRL0	MR_WR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR_ADDR[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR_TYPE
	Reset value	0																0	0	0	0												0
0x014	DDRCTRL_MRCTRL1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR_DATA[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	DDRCTRL_MRSTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR_WR_BUSY
	Reset value																																0
0x01C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x020	DDRCTRL_DERATEEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DERATE_BYTE[3:0]				Res.	DERATE_VALUE[1:0]		DERATE_ENABLE
	Reset value																									0	0	0	0		0	0	0
0x024	DDRCTRL_DERATEINT	MR4_READ_INTERVAL[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 19. DDRCTRL 寄存器映射和复位值（续）

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x028 - 0x02C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x030	DDRCTRL_PWRCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIS_CAM_DRAIN_SELFREF	Res.	SELFREF_SW	Res.	EN_DFI_DRAM_CLK_DISABLE	DEEPPOWERDOWN_EN	POWERDOWN_EN	SELFREF_EN
	Reset value																									0		0		0	0	0	0
0x034	DDRCTRL_PWRTMG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELFREF_TO_X32[7:0]								T_DPD_X4096[7:0]								Res.	Res.	Res.	POWERDOWN_TO_X32[4:0]				
	Reset value									0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0				1	0	0	0	0
0x038	DDRCTRL_HWLPCTL	Res.	Res.	Res.	Res.	HW_LP_IDLE_X32[11:0]												Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HW_LP_EXIT_IDLE_EN	HW_LP_EN
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0															1	1
0x03C - 0x04C	Res.																																
0x050	DDRCTRL_RFSHCTL0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REFRESH_MARGIN[3:0]				Res.	Res.	Res.	REFRESH_TO_X32[4:0]					Res.	Res.	Res.	REFRESH_BURST[4:0]					Res.	PER_BANK_REFRESH	Res.	Res.
	Reset value									0	0	1	0				1	0	0	0	0				0	0	0	0	0		0		
0x054 - 0x05C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 19. DDRCTRL 寄存器映射和复位值（续）

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x060	DDRCTRL_RFSHCTL3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REFRESH_UPDATE_LEVEL	DIS_AUTO_REFRESH
	Reset value																															0	0
0x064	DDRCTRL_RFSHTMG	T_RFC_NOM_X1_SEL	Res.	Res.	Res.	T_RFC_NOM_X1_X32[11:0]												LPDDR3_TREFBW_EN	Res.	Res.	Res.	Res.	Res.	T_RFC_MIN[9:0]									
	Reset value	0				0	0	0	0	0	1	1	0	0	0	1	0	0						0	0	1	0	0	0	1	1	0	0
0x068 - 0x0BC	Res.																																
0x0C0	DDRCTRL_CRCPARCTL0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFI_ALERT_ERR_CNT_CLR	DFI_ALERT_ERR_INT_CLR	DFI_ALERT_ERR_INT_EN
	Reset value																														0	0	0
0x0C4 - 0x0C8	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0CC	DDRCTRL_CRCPARSTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFI_ALERT_ERR_INT	DFI_ALERT_ERR_CNT[15:0]															
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0D0	DDRCTRL_INIT0	SKIP_DRAM_INIT[1:0]		Res.	Res.	Res.	Res.	POST_CKE_X1024[9:0]										Res.	Res.	Res.	Res.	PRE_CKE_X1024[11:0]											
	Reset value	0	0					0	0	0	0	0	0	0	0	1	0					0	0	0	0	0	1	0	0	1	1	1	0

表 19. DDRCTRL 寄存器映射和复位值（续）

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0D4	DDRCTRL_INIT1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DRAM_RSTN_X1024[8:0]									Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRE_OCD_X32[3:0]			
	Reset value								0	0	0	0	0	0	0	0	0													0	0	0	0
0x0D8	DDRCTRL_INIT2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDLE_AFTER_RESET_X32[7:0]								Res.	Res.	Res.	Res.	MIN_STABLE_CLOCK_X1[3:0]			
	Reset value																	0	0	0	0	1	1	0	1					0	1	0	1
0x0DC	DDRCTRL_INIT3	MR[15:0]																EMR[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0
		0																0x0510															
0x0E0	DDRCTRL_INIT4	EMR2[15:0]																EMR3[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0E4	DDRCTRL_INIT5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEV_ZQINIT_X32[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	MAX_AUTO_INIT_X1024[9:0]									
	Reset value									0	0	0	1	0	0	0	0							0	0	0	0	0	0	0	1	0	0
0x0E8 - 0x0EC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0F0	DDRCTRL_DIMMCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIMM_ADDR_MIRR_EN	DIMM_STAGGER_CS_EN
	Reset value																															0	0
0x0F4 - 0x0FC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 19. DDRCTRL 寄存器映射和复位值（续）

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x100	DDRCTRL_DRAMTMG0	Res.	WR2PRE[6:0]							Res.	Res.	T_FAW[5:0]						Res.	T_RAS_MAX[6:0]							Res.	Res.	T_RAS_MIN[5:0]					
	Reset value		0	0	0	1	1	1	1			0	1	0	0	0	0		0	0	1	1	0	1	1			0	0	1	1	1	1
0x104	DDRCTRL_DRAMTMG1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T_XP[4:0]					Res.	Res.	RD2PRE[5:0]						Res.	T_RC[6:0]						
	Reset value												0	1	0	0	0			0	0	0	1	0	0		0	0	1	0	1	0	0
0x108	DDRCTRL_DRAMTMG2	Res.	Res.	WRITE_LATENCY[5:0]						Res.	Res.	READ_LATENCY[5:0]						Res.	Res.	RD2WR[5:0]						Res.	Res.	WR2RD[5:0]					
	Reset value			0	0	0	0	1	1			0	0	0	1	0	1			0	0	0	1	1	0			0	0	1	1	0	1
0x10C	DDRCTRL_DRAMTMG3	Res.	Res.	T_MRW[9:0]										Res.	Res.	T_MRD[5:0]				Res.	Res.	T_MOD[9:0]											
	Reset value			0	0	0	0	0	0	0	1	0	1			0	0	0	1	0	0			0	0	0	0	0	0	1	1	0	0
0x110	DDRCTRL_DRAMTMG4	Res.	Res.	Res.	T_RCD[4:0]					Res.	Res.	Res.	Res.	T_CCD[3:0]				Res.	Res.	Res.	Res.	T_RRD[3:0]				Res.	Res.	Res.	T_RP[4:0]				
	Reset value				0	0	1	0	1					0	1	0	0					0	1	0	0				0	0	1	0	1
0x114	DDRCTRL_DRAMTMG5	Res.	Res.	Res.	Res.	T_CKSRX[3:0]				Res.	Res.	Res.	Res.	T_CKSRE[3:0]				Res.	Res.	T_CKESR[5:0]						Res.	Res.	Res.	T_CKE[4:0]				
	Reset value					0	1	0	1					0	1	0	1			0	0	0	1	0	0				0	0	0	1	1
0x118	DDRCTRL_DRAMTMG6	Res.	Res.	Res.	Res.	T_CKDPDE[3:0]				Res.	Res.	Res.	Res.	T_CKDPDX[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T_CKCSX[3:0]			
	Reset value					0	0	1	0					0	0	1	0													0	1	0	1
0x11C	DDRCTRL_DRAMTMG7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T_CKPDE[3:0]				Res.	Res.	Res.	Res.	T_CKPDX[3:0]			
	Reset value																					0	0	1	0					0	0	1	0
0x120	DDRCTRL_DRAMTMG8	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T_XS_DLL_X32[6:0]							Res.	T_XS_X32[6:0]						
	Reset value																		1	0	0	0	1	0	0		0	0	0	0	1	0	1
0x124 - 0x134	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x138	DDRCTRL_DRAMTMG14	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T_XSR[11:0]											
	Reset value																					0	0	0	0	1	0	1	0	0	0	0	0

表 19. DDRCTRL 寄存器映射和复位值（续）

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x13C	DDRCTRL_DRAMTMG15	EN_DFI_LP_T_STAB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T_STAB_X32[7:0]							
	Reset value	0																								0	0	0	0	0	0	0	0
0x140 - 0x17C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x180	DDRCTRL_ZQCTL0	DIS_AUTO_ZQ	DIS_SRX_ZQCL	ZQ_RESISTOR_SHARED	Res.	Res.	T_ZQ_LONG_NOP[10:0]											Res.	Res.	Res.	Res.	Res.	Res.	T_ZQ_SHORT_NOP[9:0]								
	Reset value	0	0	0			0	1	0	0	0	0	0	0	0	0	0							0	0	0	1	0	0	0	0	0	0
0x184	DDRCTRL_ZQCTL1	Res.	Res.	T_ZQ_RESET_NOP[9:0]										T_ZQ_SHORT_INTERVAL_X1024[19:0]																				
	Reset value			0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0x188	DDRCTRL_ZQCTL2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ZQ_RESET
	Reset value																																0
0x18C	DDRCTRL_ZQSTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ZQ_RESET_BUSY
	Reset value																																0
0x190	DDRCTRL_DFITMG0	Res.	Res.	Res.	DFI_T_CTRL_DELAY[4:0]					Res.	DFI_T_RDDATA_EN[6:0]							Res.	Res.	DFI_TPHY_WRDATA[5:0]				Res.	Res.	DFI_TPHY_WRLAT[5:0]					
	Reset value				0	0	1	1	1		0	0	0	0	0	1	0			0	0	0	0	0	0			0	0	0	0	1	0

表 19. DDRCTRL 寄存器映射和复位值（续）

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x194	DDRCTRL_DFITMG1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFI_T_WRDATA_DELAY[4:0]					Res.	Res.	Res.	DFI_T_DRAM_CLK_DISABLE[4:0]					Res.	Res.	Res.	DFI_T_DRAM_CLK_ENABLE[4:0]				
	Reset value												0	0	0	0	0				0	0	1	0	0				0	0	1	0	0
0x198	DDRCTRL_DFILPCFG0	Res.	Res.	Res.	DFI_TLP_RESP[4:0]					DFI_LP_WAKEUP_DPD[3:0]				Res.	Res.	Res.	DFI_LP_EN_DPD	DFI_LP_WAKEUP_SR[3:0]			Res.	Res.	Res.	DFI_LP_EN_SR	DFI_LP_WAKEUP_PD[3:0]				Res.	Res.	Res.	DFI_LP_EN_PD
	Reset value				0	0	1	1	1	0	0	0	0				0	0	0	0	0				0	0	0	0	0					0
0x19C	Res.																																
0x1A0	DDRCTRL_DFIUPD0	DIS_AUTO_CTRLUPD	DIS_AUTO_CTRLUPD_SRX	CTRLUPD_PRE_SRX	Res.	Res.	Res.	DFI_T_CTRLUP_MAX[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	DFI_T_CTRLUP_MIN[9:0]									
	Reset value	0	0	0				0	0	0	1	0	0	0	0	0	0							0	0	0	0	0	0	0	0	1	1
0x1A4	DDRCTRL_DFIUPD1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFI_T_CTRLUPD_INTERVAL_MIN_X1024[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFI_T_CTRLUPD_INTERVAL_MAX_X1024[7:0]									
	Reset value									0	0	0	0	0	0	0	1									0	0	0	0	0	0	0	1

表 19. DDRCTRL 寄存器映射和复位值（续）

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1A8	DDRCTRL_DFIUPD2	DFI_PHYUPD_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	1																															
0x1AC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x1B0	DDRCTRL_DFIMISC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFI_FREQUENCY[4:0]					Res.		DFI_INIT_START	CTL_IDLE_EN	Res.			DFI_INIT_COMPLETE_EN
	Reset value																				0	0	0	0	0			0	0				1
0x1B4 - 0x1B8	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x1BC	DDRCTRL_DFISTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFI_LP_ACK		DFI_INIT_COMPLETE
	Reset value																														0		0
0x1C0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x1C4	DDRCTRL_DFIPHYMSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFI_PHYMSTR_EN
	Reset value																																1
0x1C8 - 0x200	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x204	DDRCTRL_ADDRMAP1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRMAP_BANK_B2[5:0]						Res.	Res.	ADDRMAP_BANK_B1[5:0]						Res.	Res.	ADDRMAP_BANK_B0[5:0]					
	Reset value											0	0	0	0	0	0			0	0	0	0	0	0			0	0	0	0	0	0

表 19. DDRCTRL 寄存器映射和复位值（续）

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x208	DDRCTRL_ADDRMAP2	Res.	Res.	Res.	Res.	ADDRMAP_COL_B5[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_COL_B4[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_COL_B3[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_COL_B2[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0
0x20C	DDRCTRL_ADDRMAP3	Res.	Res.	Res.	ADDRMAP_COL_B9[4:0]					Res.	Res.	Res.	ADDRMAP_COL_B8[4:0]					Res.	Res.	Res.	ADDRMAP_COL_B7[4:0]					Res.	Res.	Res.	Res.	ADDRMAP_COL_B6[3:0]			
	Reset value				0	0	0	0	0				0	0	0	0	0				0	0	0	0	0					0	0	0	0
0x210	DDRCTRL_ADDRMAP4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRMAP_COL_B11[4:0]					Res.	Res.	Res.	ADDRMAP_COL_B10[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x214	DDRCTRL_ADDRMAP5	Res.	Res.	Res.	Res.	ADDRMAP_ROW_B11[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B2_10[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B1[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B0[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0
0x218	DDRCTRL_ADDRMAP6	LPDDR3_6GB_12GB	Res.	Res.	Res.	ADDRMAP_ROW_B15[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B14[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B13[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B12[3:0]			
	Reset value	0				0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0
0x21C - 0x220	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 19. DDRCTRL 寄存器映射和复位值（续）

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x224	DDRCTRL_ADDRMAP9	Res.	Res.	Res.	Res.	ADDRMAP_ROW_B5[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B4[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B3[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B2[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0
0x228	DDRCTRL_ADDRMAP10	Res.	Res.	Res.	Res.	ADDRMAP_ROW_B9[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B8[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B7[3:0]				Res.	Res.	Res.	Res.	ADDRMAP_ROW_B6[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0
0x22C	DDRCTRL_ADDRMAP11	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRMAP_ROW_B10[3:0]			
	Reset value																													0	0	0	0
0x230 - 0x23C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x240	DDRCTRL_ODTCFG	Res.	Res.	Res.	Res.	WR_ODT_HOLD[3:0]				Res.	Res.	Res.	WR_ODT_DELAY[4:0]					Res.	Res.	Res.	Res.	RD_ODT_HOLD[3:0]				Res.	RD_ODT_DELAY[4:0]			Res.	Res.
	Reset value					0	1	0	0				0	0	0	0	0					0	1	0	0		0	0	0	0	0		
0x244	DDRCTRL_ODTMAP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RANK0_RD_ODT	Res.	Res.	Res.	RANK0_WR_ODT
	Reset value																												1				1
0x248 - 0x24C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 19. DDRCTRL 寄存器映射和复位值（续）

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x250	DDRCTRL_SCHED	Res.	RDWR_IDLE_GAP[6:0]							GO2CRITICAL_HYSTERESIS[7:0]								Res.	Res.	Res.	Res.	LPR_NUM_ENTRIES[3:0]				Res.	Res.	Res.	Res.	Res.	PAGECLOSE	PREFER_WRITE	FORCE_LOW_PRI_N
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					1	0	0	0						1	0	1
0x254	DDRCTRL_SCHED1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PAGECLOSE_TIMER[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x258	Res.																																
0x25C	DDRCTRL_PERFHPR1	HPR_XACT_RUN_LENGTH[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HPR_MAX_STARVE[15:0]															
	Reset value	0	0	0	0	1	1	1	1									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x260	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x264	DDRCTRL_PERFLPR1	LPR_XACT_RUN_LENGTH[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPR_MAX_STARVE[15:0]															
	Reset value	0	0	0	0	1	1	1	1									0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
0x268	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 19. DDRCTRL 寄存器映射和复位值（续）

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x26C	DDRCTRL_PERFWR1	W_XACT_RUN_LENGTH[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	W_MAX_STARVE[15:0]															
	Reset value	0	0	0	0	1	1	1	1									0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
0x270 - 0x2FC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x300	DDRCTRL_DBG0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIS_COLLISION_PAGE_OPT	Res.	Res.	Res.	DIS_WC
	Reset value																												0				0
0x304	DDRCTRL_DBG1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIS_HIF	DIS_DQ
	Reset value																															0	0
0x308	DDRCTRL_DBGCAM	Res.	Res.	WR_DATA_PIPELINE_EMPTY	RD_DATA_PIPELINE_EMPTY	Res.	DBG_WR_Q_EMPTY	DBG_RD_Q_EMPTY	DBG_STALL	Res.	Res.	Res.	DBG_W_Q_DEPTH[4:0]					Res.	Res.	Res.	DBG_LPR_Q_DEPTH[4:0]					Res.	Res.	Res.	DBG_HPR_Q_DEPTH[4:0]				
	Reset value			0	0		0	0	0				0	0	0	0	0				0	0	0	0	0				0	0	0	0	0
0x30C	DDRCTRL_DBGCMD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTRLUPD	ZQ_CALIB_SHORT	Res.	Res.	Res.	RANK0_REFRESH
	Reset value																											0	0				0

表 19. DDRCTRL 寄存器映射和复位值（续）

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x310	DDRCTRL_DBGSTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTRLUPD_BUSY	ZQ_CALIB_SHORT_BUSY	Res.	Res.	Res.	RANK0_REFRESH_BUSY
	Reset value																											0	0				0
0x314 - 0x31C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x320	DDRCTRL_SWCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SW_DONE
	Reset value																																1
0x324	DDRCTRL_SWSTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SW_DONE_ACK
	Reset value																																1
0x328 - 0x368	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x36C	DDRCTRL_POISONCFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RD_POISON_INTR_CLR	Res.	Res.	Res.	RD_POISON_INTR_EN	Res.	Res.	Res.	RD_POISON_SLVERR_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR_POISON_INTR_CLR	Res.	Res.	Res.	WR_POISON_INTR_EN	Res.	Res.	Res.	WR_POISON_SLVERR_EN
	Reset value								0				1				1								0				1				1
0x370	DDRCTRL_POISONSTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RD_POISON_INTR_0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR_POISON_INTR_0
	Reset value																0																0
0x374 - 0x3F8	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 19. DDRCTRL 寄存器映射和复位值（续）

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x3FC	DDRCTRL_PSTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR_PORT_BUSY_0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RD_PORT_BUSY_0
	Reset value																0																0
0x400	DDRCTRL_PCCFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BL_EXP_MODE	Res.	Res.	Res.	PAGEMATCH_LIMIT	Res.	Res.	Res.	GO2CRITICAL_EN
	Reset value																								0				0				0
0x404	DDRCTRL_PCFGR_0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDWR_ORDERED_EN	Res.	RD_PORT_PAGEMATCH_EN	RD_PORT_URGENT_EN	RD_PORT_AGING_EN	Res.	Res.	RD_PORT_PRIORITY[9:0]									
	Reset value																0		1	0	0			0	0	0	0	0	0	0	0	0	0
0x408	DDRCTRL_PCFGW_0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR_PORT_PAGEMATCH_EN	WR_PORT_URGENT_EN	WR_PORT_AGING_EN	Res.	Res.	WR_PORT_PRIORITY[9:0]									
	Reset value																		1	0	0			0	0	0	0	0	0	0	0	0	0
0x40C - 0x48C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x490	DDRCTRL_PCTRL_0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PORT_EN
	Reset value																																0

表 19. DDRCTRL 寄存器映射和复位值（续）

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x494	DDRCTRL_PCFGQOS0_0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RQOS_MAP_REGION1[1:0]		Res.	Res.	RQOS_MAP_REGION0[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RQOS_MAP_LEVEL1[3:0]			
	Reset value											0	0			0	0													0	0	0	0
0x498	DDRCTRL_PCFGQOS1_0	Res.	Res.	Res.	Res.	Res.	RQOS_MAP_TIMEOUTR[10:0]											Res.	Res.	Res.	Res.	Res.	RQOS_MAP_TIMEOUTB[10:0]										
	Reset value						0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x49C	DDRCTRL_PCFGWQOS0_0	Res.	Res.	Res.	Res.	Res.	Res.	WQOS_MAP_REGION2[1:0]		Res.	Res.	WQOS_MAP_REGION1[1:0]		Res.	Res.	WQOS_MAP_REGION0[1:0]		Res.	Res.	Res.	Res.	WQOS_MAP_LEVEL2[3:0]				Res.	Res.	Res.	Res.	WQOS_MAP_LEVEL1[3:0]			
	Reset value							0	0			0	0			0	0					1	1	1	0					0	0	0	0
0x4A0	DDRCTRL_PCFGWQOS1_0	Res.	Res.	Res.	Res.	Res.	WQOS_MAP_TIMEOUT2[10:0]											Res.	Res.	Res.	Res.	Res.	WQOS_MAP_TIMEOUT1[10:0]										
	Reset value						0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

6 用于 DDR 的 TrustZone® 地址空间控制器 (TZC)

6.1 简介

TZC 根据 TrustZone 访问权限以及多达 9 个可编程区域的非安全主设备地址 ID (NSAID) 对 DDR 控制器的读 / 写访问进行过滤。

注： *有关更多信息，请参见 Arm TZC-400 技术参考手册。*

6.2 TZC 特性

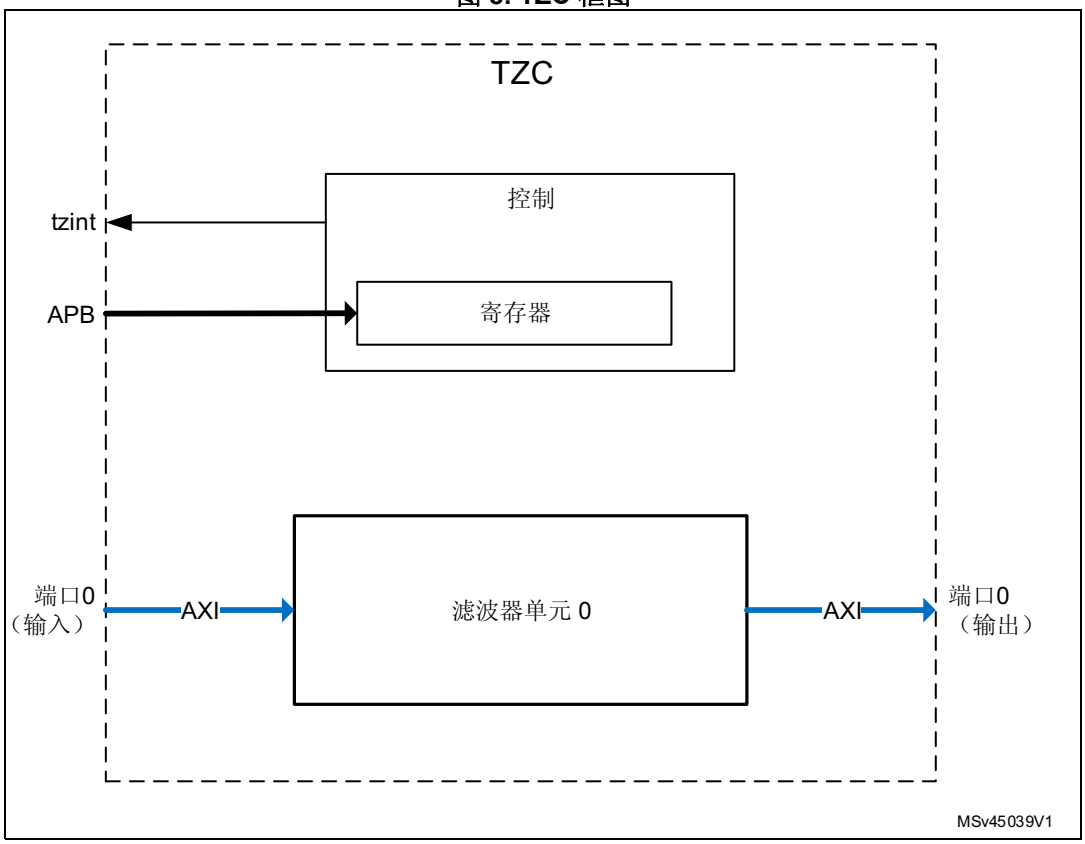
- 32 位 APB4 接口
- TZC 配置只能由安全主设备支持
- 1 个过滤单元
- 9 个区域：
  - 区域 0 始终使能，并覆盖整个地址范围
  - 区域 1 到 8 具有可编程的基址/结束地址，并且可以分配给过滤器。
- 每个区域都编程了安全和非安全访问权限
- 根据 NSAID 过滤非安全访问
- 未通过的权限检查会通过 AXI 总线错误和/或中断来指示
- 双重读取访问路径：
  - 支持 256 个未完成事务的正常路径
  - 使用 FPID 信号实现具有有限未完成能力的快速路径
- 可使能和禁止过滤器的门卫逻辑
- 推测性访问

注： *STM32MP13xx 中不使用快速路径和 FPID 信号。FPID 的配置接受设置为最小值 (8)。*

6.3 TZC 框图

TZC 框图如*图 8：TZC 框图*所示。

图 8. TZC 框图



6.4 TZC 功能描述

TZC 对多个区域强制执行访问权限：

- 区域 0：始终使能，并覆盖整个 32 位地址空间。
- 区域 1 到 8：完全可编程。

根据事务安全性和事务源检查访问权限：

- 事务安全性由 AXI TrustZone 保护信号确定：awprot[1] 和 arprot[1]。
- 事务源由主设备 NSAID 确定。如果权限失败，失败 ID 会由位于 AXI 端口的 AXI ID 报告。根据*表 20：NSAID 和 AXI _ID 映射*定义了 NSAID 和 AXI ID。

如果需要，必须授予区域安全访问权限，无论非安全读取和非安全写入的设置如何。例如，当权限设置为允许非安全读取或非安全写入访问时，这并不意味着相应的安全读取或安全写入访问权限也会自动被授予。

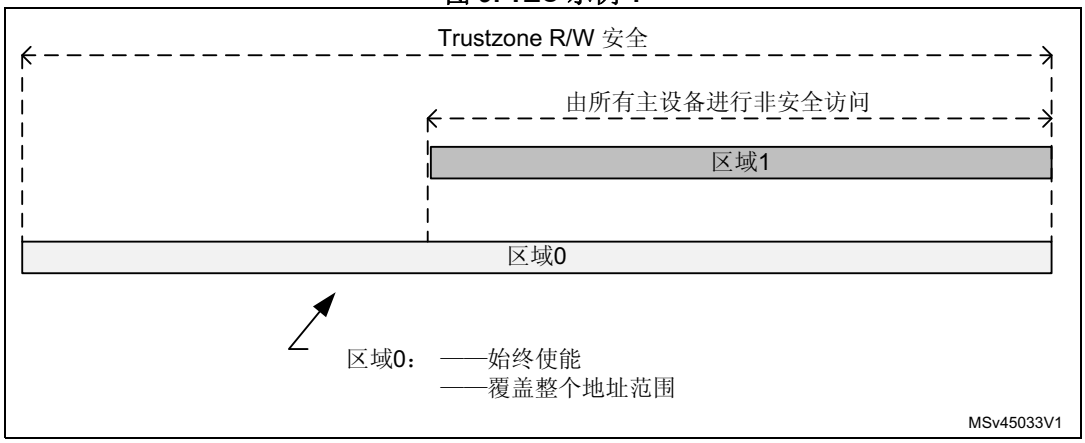
表 20. NSAID 和 AXI _ID 映射

主设备	NSAID[3:0]	ID_WIDTH	AXI_ID[10:0]
CA7	0b0000	AR:6/AW:5	11’b10XXXXXX000
LTDC	0b0011	4	11’b1000XXXX011
DCMIPP	0b1011	4	11’b0100XXXX100
MDMA	0b0101	1	11’b0000000X010
DMA1/2/3	0b0110	0	11’b00000000000
USBH (EHCI)	0b0111	0	11’b01000000000
USBH (OHCI)	0b0111	0	11’b01000000001
OTG	0b1000	0	11’b00000000000
SDMMC1	0b1001	0	11'b00000000011
SDMMC2	0b1001	0	11'b00000000100
ETH1	0b1010	4	11'b0100xxxx010
ETH2	0b1010	4	11'b0100xxxx011
DAP	0b1111	0	11'b00000000001

6.5 TZC 应用编程示例

图 9：TZC 示例 1 给出了一个简单的应用示例：

图 9. TZC 示例 1



在此示例中，安全区域可以访问整个 DDR 范围，所有非安全主设备都有权限访问 ≥ base1 的地址。

它的编程方式如下：

- 区域 0
  - 默认始终使能
  - 默认覆盖完整地址范围
  - 设置区域 0 的安全 r/w 访问权限
  - 设置区域 0 NSAID 的 r/w 访问权限
  - 使能区域 0 上的过滤器

```
REGION0_ATTRIBUTE.s_wr_en = 1
REGION0_ATTRIBUTE.s_rd_en = 1
REGION0_ID_ACCESS.nsaid_we_en = 0b00000000000
REGION0_ID_ACCESS.nsaid_re_en = 0b00000000000
REGION0_ATTRIBUTE.filter_en = 0b1
```

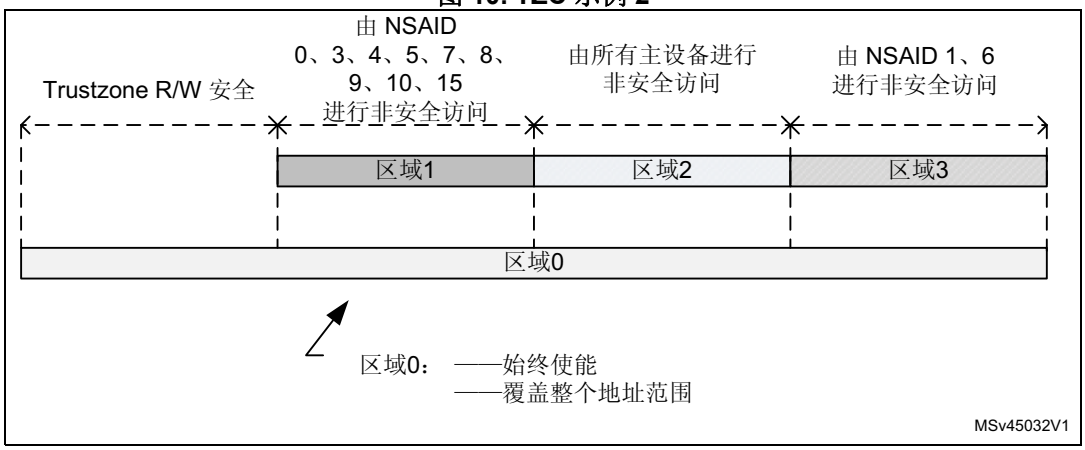
随后安全区域定义其他区域（1 到 8）并向所有非安全 NSAID 开放访问权限。

- 区域 1
  - 定义区域 1
  - 区域 1 可通过安全 r/w 访问
  - 为所有 NSAID 设置区域 1 NSAID 的 r/w 权限
  - 使能区域 1 上的过滤器

```
TZC.REGION_BASE_LOW1.base_address_low = `base1
REGION1_TOP_LOW.top_address_low = `base1 + `size1
REGION1_ATTRIBUTE.s_wr_en = 1
REGION1_ATTRIBUTE.s_rd_en = 1
REGION1_ID_ACCESS.nsaid_we_en = 0xFFFF
REGION1_ID_ACCESS.nsaid_re_en = 0xFFFF
REGION1_ATTRIBUTE.filter_en = 0b1
```

图 10：TZC 示例 2 给出了 TZC 的另一个应用示例：

图 10. TZC 示例 2



在此示例中，安全区域仅可访问 ≤ base1 的地址，非安全主设备对区域 1 和 3 具有选择性权限；区域 2 被所有非安全主设备共享。

它的编程方式如下：

- 区域 0
  - 默认始终使能
  - 默认覆盖完整地址范围
  - 设置区域 0 的安全 r/w 访问权限
  - 设置区域 0 NSAID 的 r/w 访问权限
  - 使能区域 0 上的过滤器

```
REGION0_ATTRIBUTE.s_wr_en = 1
REGION0_ATTRIBUTE.s_rd_en = 1
REGION0_ID_ACCESS.nsaid_we_en = 0x0000
REGION0_ID_ACCESS.nsaid_re_en = 0x0000
REGION0_ATTRIBUTE.filter_en = 0b1
```

随后，安全区域定义了几个区域（1 到 3）并开放了对非安全 NSAID 的访问。

- 区域 1
  - 定义区域 1
  - 区域 1 不可通过安全 r/w 访问
  - 为区域 1 设置 NSAID r/w 权限：0,3,4,5,7,8,9,10,15
  - 使能区域 1 上的过滤器
    TZC.REGION_BASE_LOW1.base_address_low = ‘base1

```
REGION1_TOP_LOW.top_address_low = `base1 + `size1
REGION1_ATTRIBUTE.s_wr_en = 0
REGION1_ATTRIBUTE.s_rd_en = 0
REGION1_ID_ACCESS.nsaid_we_en = 0x87B9
REGION1_ID_ACCESS.nsaid_re_en = 0x87B9
REGION1_ATTRIBUTE.filter_en = 0b1
```

- 区域 2
  - 定义区域 2
  - 区域 2 不可通过安全 r/w 访问
  - 为区域 2 设置 NSAID r/w 权限：0,1,3,4,5,6,7,8,9,10,15
  - 使能区域 2 上的过滤器

```
REGION2_BASE_LOW.base_address_low = 'base2
REGION2_TOP_LOW.top_address_low = 'base2 + 'size2
REGION2_ATTRIBUTE.s_wr_en = 0
REGION2_ATTRIBUTE.s_rd_en = 0
REGION2_ID_ACCESS.nsaid_we_en = 0x87FB
REGION2_ID_ACCESS.nsaid_re_en = 0x87FB
REGION2_ATTRIBUTE.filter_en = 0b1
```

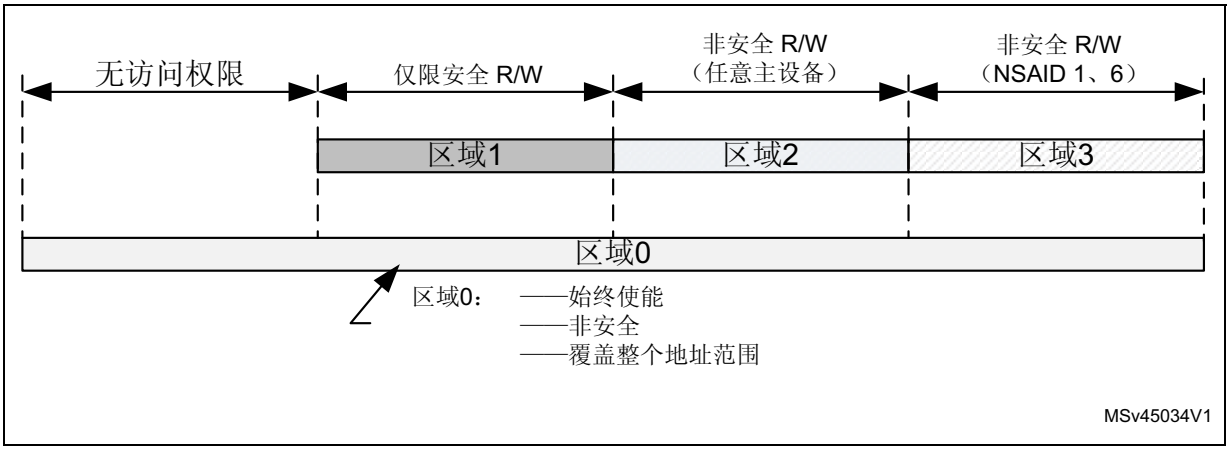
- 区域 3
  - 定义区域 3
  - 区域 3 不可通过安全 r/w 访问
  - 为区域 3 设置 NSAID r/w 权限：1,6
  - 使能区域 3 上的过滤器

```
REGION3_BASE_LOW.base_address_low = 'base3
REGION3_TOP_LOW.top_address_low = 'base3 + 'size3
REGION3_ATTRIBUTE.s_wr_en = 0
REGION3_ATTRIBUTE.s_rd_en = 0
REGION3_ID_ACCESS.nsaid_we_en = 0x0042
REGION3_ID_ACCESS.nsaid_re_en = 0x0042
REGION3_ATTRIBUTE.filter_en = 0b1
```

注： *所有共享区域必须是不同的，因为区域间不能重叠。*

图 11：TZC 示例 3 给出了 TZC 的第三个应用示例：此示例中使用区域 0 作为覆盖区，安全或非安全区域设置为空缺区。这种编程模型更便于捕捉编程错误。

图 11. TZC 示例 3



在此示例中，安全区域仅可访问 ≤ base1 的地址，非安全主设备对区域 2 和区域 3 有选择性访问权限。

它的编程方式如下：

- 区域 0
  - 默认始终使能
  - 默认覆盖完整地址范围
  - 设置区域 0 无安全访问权限（默认）
  - 设置区域 0 无 NSAID r/w 访问权限（默认）
  - 使能区域 0 上的过滤器

```
REGION0_ATTRIBUTE.s_wr_en = 0（默认）
REGION0_ATTRIBUTE.s_rd_en = 0（默认）
REGION0_ID_ACCESS.nsaid_we_en = 0x0000（默认）
REGION0_ID_ACCESS.nsaid_re_en = 0x0000（默认）
REGION0_ATTRIBUTE.filter_en = 0b1
```

随后，安全区域定义了几个区域（1 到 3）并为安全和非安全访问开放了空缺区

- 区域 1
  - 定义区域 1
  - 区域 1 可通过安全 r/w 访问
  - 设置区域 1 无 NSAID r/w 权限
  - 使能区域 1 上的过滤器

```
TZC.REGION_BASE_LOW1.base_address_low = `base1
REGION1_TOP_LOW.top_address_low = `base1 + `size1
REGION1_ATTRIBUTE.s_wr_en = 1
REGION1_ATTRIBUTE.s_rd_en = 1
REGION1_ID_ACCESS.nsaid_we_en = 0x0000
REGION1_ID_ACCESS.nsaid_re_en = 0x0000
REGION1_ATTRIBUTE.filter_en = 0b1
```

- 区域 2
  - 定义区域 2
  - 区域 2 不可通过安全 r/w 访问
  - 为所有 NSAID 设置区域 2 r/w 权限
  - 使能区域 2 上的过滤器

```
REGION2_BASE_LOW.base_address_low = `base2
REGION2_TOP_LOW.top_address_low = `base2 + `size2
REGION2_ATTRIBUTE.s_wr_en = 0
REGION2_ATTRIBUTE.s_rd_en = 0
REGION2_ID_ACCESS.nsaid_we_en = 0xFFFF
REGION2_ID_ACCESS.nsaid_re_en = 0xFFFF
REGION2_ATTRIBUTE.filter_en = 0b1
```

- 区域 3
  - 定义区域 3
  - 区域 3 不可通过安全 r/w 访问
  - 为 NSAID 1、6 设置区域 3 r/w 权限
  - 使能区域 3 上的过滤器

```
REGION3_BASE_LOW.base_address_low = 'base3
REGION3_TOP_LOW.top_address_low = 'base3 + 'size3
REGION3_ATTRIBUTE.s_wr_en = 0
REGION3_ATTRIBUTE.s_rd_en = 0
REGION3_ID_ACCESS.nsaid_we_en = 0x0042
REGION3_ID_ACCESS.nsaid_re_en = 0x0042
REGION3_ATTRIBUTE.filter_en = 0b1
```

6.5.1 DMA 请求

注： *TZC 不支持 DMA 接口。*

6.5.2 中断

注： *当某些安全检查未通过时，TZC 会将中断 **tzcint** 置为有效。*

6.6 TZC 寄存器

TZC 寄存器只能按字（32 位）进行访问。

6.6.1 TZC 配置寄存器 (TZC_BUILD_CONFIG)

TZC configuration register

偏移地址：0x000

复位值：0x0000 1F08

提供有关 TZC 配置的信息。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	NO_OF_FILTERS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							r								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ADDRESS_WIDTH[5:0]						Res.	Res.	Res.	NO_OF_REGIONS[4:0]				
		r	r	r	r	r	r				r	r	r	r	r

位 31:26 保留，必须保持复位值。

位 25 保留，必须保持复位值。

位 24 **NO_OF_FILTERS**：过滤器数

0x0：1 个过滤器

位 23:14 保留，必须保持复位值。

位 13:8 **ADDRESS_WIDTH[5:0]**：地址宽度

0x1F：32 位地址宽度

其他值：Reserved

位 7:5 保留，必须保持复位值。

位 4:0 **NO_OF_REGIONS[4:0]**：区域数

0x8：9 个区域

其他值：Reserved

6.6.2 TZC 操作寄存器 (TZC_ACTION)

TZC action register

偏移地址：0x004

复位值：0x0000 0000

当区域权限失败发生时，控制中断和总线错误响应行为。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REACTION_VALUE[1:0]	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **REACTION_VALUE[1:0]**：权限失败反应
控制 TZC 如何指示区域权限失败。
0：将 tzcint 设为低电平，并在总线上发出 OKAY
1：将 tzcint 设为低电平，并在总线上发出 DECERR
2：将 tzcint 设为高电平，并在总线上发出 OKAY
3：将 tzcint 设为高电平，并在总线上发出 DECERR

6.6.3 TZC 门卫寄存器 (TZC_GATE_KEEPER)

TZC gate keeper register

偏移地址：0x008

复位值：0x0000 0000

为实现的过滤单元中的门卫提供控制和状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPENSTAT
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPENREQ
															rw

位 31:18 保留，必须保持复位值。

位 17 保留，必须保持复位值。

位 16 **OPENSTAT**：过滤器的门卫状态
0：过滤器已开启
1：过滤器已关闭

位 15:2 保留，必须保持复位值。

位 1 保留，必须保持复位值。

位 0 **OPENREQ**：过滤器的门卫开启请求
0：请求过滤器开启
1：请求过滤器关闭

6.6.4 TZC 推测控制寄存器 (TZC_SPECULATION_CTRL)

TZC speculation control register

偏移地址：0x00C

复位值：0x0000 0000

控制读写访问推测。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITESPEC_DISABLE	READSPEC_DISABLE
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **WRITESPEC_DISABLE**：写访问推测禁止
- 0：使能写访问推测
- 1：禁止写访问推测

位 0 **READSPEC_DISABLE**：读访问推测禁止
- 0：使能读访问推测
- 1：禁止读访问推测

6.6.5 TZC 中断状态寄存器 (TZC_INT_STATUS)

TZC interrupt status register

偏移地址：0x010

复位值：0x0000 0000

包含中断信号 TZCINT 的状态，该信号报告访问安全违例或区域重叠错误。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVERLAP[1:0]	
														r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVERRUN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STATUS
							r								r

位 31:18 保留，必须保持复位值。

位 17:16 **OVERLAP[1:0]**：每个过滤器的重叠违例
位 17 用于过滤器 1，位 16 用于过滤器 0。
- 0：中断未置为有效
- 1：中断已置为有效并等待清除

位 15:10 保留，必须保持复位值。

位 9 保留，必须保持复位值。

位 8 **OVERRUN**：权限失败上溢
过滤器有两个或多个区域权限失败。
0：中断未置为有效
1：中断已置为有效并等待清除

位 7:2 保留，必须保持复位值。

位 1 保留，必须保持复位值。

位 0 **STATUS**：过滤器的中断状态。
0：中断未置为有效
1：中断已置为有效并等待清除

6.6.6 TZC 中断清除寄存器 (TZC_INT_CLEAR)

TZC interrupt clear register

偏移地址：0x014

复位值：0x0000 0000

过滤器的中断清除。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLEAR
															w

位 31:1 保留，必须保持复位值。

位 0 **CLEAR**：过滤器中断清除
写入“1”以清除每个过滤器的中断。

6.6.7 TZC 失败地址低位寄存器 (TZC_FAIL_ADDRESS_LOW)

TZC fail address low register

偏移地址：0x020

复位值：0x0000 0000

过滤器中第一个失败访问的低位地址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR_STATUS_LOW[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR_STATUS_LOW[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ADDR_STATUS_LOW[31:0]**：失败地址低位
过滤器中第一个未通过访问权限检查的低 32 位地址。

6.6.8 TZC 失败地址高位寄存器 (TZC_FAIL_ADDRESS_HIGH)

TZC fail address high register

偏移地址：0x024

复位值：0x0000 0000

过滤器中第一个失败访问的高位地址。

注： *不用于 32 位地址。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

6.6.9 TZC 失败控制寄存器 (TZC_FAIL_CONTROL)

TZC fail control register

偏移地址：0x028

复位值：0x0000 0000

过滤器中未通过区域权限检查的第一个访问的状态信息。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIRECTION	Res.	Res.	NON_SECURE	PRIVILEGE	Res.	Res.	Res.	Res.
							r			r	r				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:25 保留，必须保持复位值。

位 24 **DIRECTION**：访问失败方向
0: 读访问失败
1: 写访问失败

位 23:22 保留，必须保持复位值。

位 21 **NON_SECURE**：非安全访问失败
0: 安全访问失败
1: 非安全访问失败

位 20 **PRIVILEGE**：特权访问失败
0: 非特权访问失败
1: 特权访问失败

位 19:0 保留，必须保持复位值。

6.6.10 TZC 失败 ID 寄存器 (TZC_FAIL_ID)

TZC fail ID register

偏移地址：0x02C

复位值：0x0000 0000

包含在过滤器单元中未通过区域权限检查的第一个访问的主设备 AXI ARID 或 AWID。即使 ACTION 寄存器设置为不驱动中断信号，这也会发生。

注： *表 20：NSAID 和 AXI _ID 映射*中描述了 AXI ID 映射。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	ID[10:0]										
					r	r	r	r	r	r	r	r	r	r	r

位 31:11 保留，必须保持复位值。

位 10:0 **ID[10:0]**：AXI 失败 ID
返回第一次失败的 AXI ID。

6.6.11 TZC 区域 0 基址低位寄存器 (TZC_REGION_BASE_LOW0)

TZC region 0 base address low register

偏移地址：0x100

复位值：0x0000 0000

区域 0 的基址低位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BASE_ADDRESS_LOW[19:4]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_ADDRESS_LOW[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r												

位 31:12 **BASE_ADDRESS_LOW[19:0]**：区域 0 的基址位 [31:12]
0x00000：区域 0 的固定值
其他值：Reserved

位 11:0 保留，必须保持复位值。

6.6.12 TZC 区域 x 基址高位寄存器 (TZC_REGION_BASE_HIGHx)

TZC region x base address high register

偏移地址：0x104 + 0x20 * x（x = 0 到 8）

复位值：0x0000 0000

基址高位不与 32 位地址一起使用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

6.6.13 TZC 区域 0 顶地址低位寄存器 (TZC_REGION_TOP_LOW0)

TZC region 0 top address low register

偏移地址：0x108

复位值：0xFFFF FFFF

区域 0 的顶地址位 [31:12]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TOP_ADDRESS_LOW[19:4]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOP_ADDRESS_LOW[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r												

位 31:12 **TOP_ADDRESS_LOW[19:0]**：区域 0 的顶地址位 [31:12]
0xFFFFF：区域 0 的固定值
其他值：Reserved

位 11:0 保留，必须保持复位值。

6.6.14 TZC 区域 x 顶地址高位寄存器 (TZC_REGION_TOP_HIGHx)

TZC region x top address high register

偏移地址：0x10C + 0x20 * x（x = 0 到 8）

复位值：0x0000 0000

区域的顶地址高位不与 32 位地址一起使用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

6.6.15 TZC 区域 0 属性寄存器 (TZC_REGION_ATTRIBUTE0)

TZC region 0 attribute register

偏移地址：0x110

复位值：0x0000 0003

区域 0 属性。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
S_WR_EN	S_RD_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FILTER_EN
															r

位 31 **S_WR_EN**：安全全局写入使能
0：不允许对该区域进行安全写入
1：允许对该区域进行安全写入

位 30 **S_RD_EN**：安全全局读取使能
0：不允许对该区域进行安全读取
1：允许对该区域进行安全读取

位 29:2 保留，必须保持复位值。

位 1 保留，必须保持复位值。

位 0 **FILTER_EN**：过滤器的区域使能
0: Reserved
1：使能该区域的过滤器

注： *对于区域 0，FILTER_EN 位为“1”且不能修改。*

6.6.16 TZC 区域 x ID 访问寄存器 (TZC_REGION_ID_ACCESSx)

TZC region x ID access register

偏移地址：0x114 + 0x20 * x（x = 0 到 8）

复位值：0x0000 0000

基于 NSAID 的区域非安全访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NSAID_WR_EN[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSAID_RD_EN[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **NSAID_WR_EN[15:0]**：每个 NSAID 的区域写入使能
位 16 关联到 NSAID = 0 .. 位 31 关联到 NSAID = 15
0：禁止对此 NSAID 的区域进行非安全写入
1：允许对此 NSAID 的区域进行非安全写入

位 15:0 **NSAID_RD_EN[15:0]**：每个 NSAID 的区域读取使能
位 0 关联到 NSAID = 0 ... 位 15 关联到 NSAID = 15。
0：禁止对此 NSAID 的区域进行非安全读取
1：允许对此 NSAID 的区域进行非安全读取

6.6.17 TZC 区域 x 基址低位寄存器 (TZC_REGION_BASE_LOWx)

TZC region x base address low register

偏移地址：0x120 + 0x20 * (x - 1)（x = 1 到 8）

复位值：0x0000 0000

区域 1 到 8 的基址低位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BASE_ADDRESS_LOW[19:4]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_ADDRESS_LOW[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw												

位 31:12 **BASE_ADDRESS_LOW[19:0]**：区域 x 的基址位 [31:12]

位 11:0 保留，必须保持复位值。

6.6.18 TZC 区域 x 顶地址低位寄存器 (TZC_REGION_TOP_LOWx)

TZC region x top address low register

偏移地址：0x128 + 0x20 * (x - 1)（x = 1 到 8）

复位值：0x0000 0FFF

区域 x 的顶地址位 [31:12]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TOP_ADDRESS_LOW[19:4]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOP_ADDRESS_LOW[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw												

位 31:12 **TOP_ADDRESS_LOW[19:0]**：区域 x 的顶地址位 [31:12]

位 11:0 保留，必须保持复位值。

6.6.19 TZC 区域 x 属性寄存器 (TZC_REGION_ATTRIBUTEx)

TZC region x attribute register

偏移地址：0x130 + 0x20 * (x - 1)（x = 1 到 8）

复位值：0x0000 0000

区域 x 属性。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
S_WR_EN	S_RD_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FILTER_EN
															rw

位 31 **S_WR_EN**：安全全局写入使能
- 0：不允许对该区域进行安全写入
- 1：允许对该区域进行安全写入

位 30 **S_RD_EN**：安全全局读取使能
- 0：不允许对该区域进行安全读取
- 1：允许对该区域进行安全读取

位 29:2 保留，必须保持复位值。

位 1 保留，必须保持复位值。

位 0 **FILTER_EN**：过滤器的区域使能
- 0：禁止该区域的过滤器
- 1：使能该区域的过滤器

6.6.20 TZC 外设 ID 4 寄存器 (TZC_PID4)

TZC peripheral ID 4 register

偏移地址：0xFD0

复位值：0x0000 0004

外设 ID 4。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PER_ID_4[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PER_ID_4[7:0]**：外设 ID 4

6.6.21 TZC 外设 ID 5 寄存器 (TZC_PID5)

TZC peripheral ID 5 register

偏移地址：0xFD4

复位值：0x0000 0000

外设 ID 5。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PER_ID_5[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PER_ID_5[7:0]**：外设 ID 5

6.6.22 TZC 外设 ID 6 寄存器 (TZC_PID6)

TZC peripheral ID 6 register

偏移地址：0xFD8

复位值：0x0000 0000

外设 ID 6。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PER_ID_6[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PER_ID_6[7:0]**：外设 ID 6

6.6.23 TZC 外设 ID 7 寄存器 (TZC_PID7)

TZC peripheral ID 7 register

偏移地址：0xFDC

复位值：0x0000 0000

外设 ID 7。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PER_ID_7[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PER_ID_7[7:0]**：外设 ID 7

6.6.24 TZC 外设 ID 0 寄存器 (TZC_PID0)

TZC peripheral ID 0 register

偏移地址：0xFE0

复位值：0x0000 0060

外设 ID 0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PER_ID_0[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PER_ID_0[7:0]**：外设 ID 0

6.6.25 TZC 外设 ID 1 寄存器 (TZC_PID1)

TZC peripheral ID 1 register

偏移地址：0xFE4

复位值：0x0000 00B4

外设 ID 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PER_ID_1[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PER_ID_1[7:0]**：外设 ID 1

6.6.26 TZC 外设 ID 2 寄存器 (TZC_PID2)

TZC peripheral ID 2 register

偏移地址：0xFE8

复位值：0x0000 002B

外设 ID 2。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PER_ID_2[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PER_ID_2[7:0]**：外设 ID 2

6.6.27 TZC 外设 ID 3 寄存器 (TZC_PID3)

TZC peripheral ID 3 register

偏移地址：0xFEC

复位值：0x0000 0000

外设 ID 3。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PER_ID_3[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PER_ID_3[7:0]**：外设 ID 3

6.6.28 TZC 组件 ID 0 寄存器 (TZC_CID0)

TZC component ID 0 register

偏移地址：0xFF0

复位值：0x0000 000D

组件 ID 0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP_ID_0[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **COMP_ID_0[7:0]**：组件 ID 0

6.6.29 TZC 组件 ID 1 寄存器 (TZC_CID1)

TZC component ID 1 register

偏移地址：0xFF4

复位值：0x0000 00F0

组件 ID 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP_ID_1[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **COMP_ID_1[7:0]**：组件 ID 0

6.6.30 TZC 组件 ID 2 寄存器 (TZC_CID2)

TZC component ID 2 register

偏移地址：0xFF8

复位值：0x0000 0005

组件 ID 2。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP_ID_2[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **COMP_ID_2[7:0]**：组件 ID 2

6.6.31 TZC 组件 ID 3 寄存器 (TZC_CID3)

TZC component ID 3 register

偏移地址：0xFFC

复位值：0x0000 00B1

组件 ID 3。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP_ID_3[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **COMP_ID_3[7:0]**：组件 ID 3

6.6.32 TZC 寄存器映射

表 21. TZC 寄存器映射和复位值

偏移	寄存器名称和复位值	寄存器大小																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	TZC_BUILD_CONFIG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NO_OF_FILTERS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRESS_WIDTH[5:0]						Res.	Res.	Res.	NO_OF_REGIONS[4:0]				
	Reset value								0											0	1	1	1	1	1				0	1	0	0	0
0x004	TZC_ACTION	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REACTION_VALUE[1:0]	
	Reset value																															0	0
0x008	TZC_GATE_KEEPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPENSTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		OPENREQ
	Reset value																0																0
0x00C	TZC_SPECULATION_CTRL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITESPEC_DISABLE	READSPEC_DISABLE
	Reset value																															0	0
0x010	TZC_INT_STATUS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVERLAP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVERRUN	Res.	Res.	Res.	Res.	Res.	Res.		STATUS
	Reset value																0								0								0
0x014	TZC_INT_CLEAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLEAR
	Reset value																																0
0x018 - 0x01C	Reserved	Res.																															
0x020	TZC_FAIL_ADDRESS_LOW	ADDR_STATUS_LOW[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	TZC_FAIL_ADDRESS_HIGH	FAIL_ADDRESS_HIGH0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 21. TZC 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	寄存器大小																														
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x028	**TZC_FAIL_CONTROL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIRECTION	Res.	Res.	NON_SECURE	PRIVILEGE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value								0			0	0																				
0x02C	**TZC_FAIL_ID**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ID[10:0]										
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x030 - 0x0FC	Reserved	Res.																															
0x100	**TZC_REGION_BASE_LOW0**	BASE_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x104	**TZC_REGION_BASE_HIGH0**	REGION_BASE_HIGH0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x108	**TZC_REGION_TOP_LOW0**	TOP_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1												
0x10C	**TZC_REGION_TOP_HIGH0**	REGION_TOP_HIGH0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x110	**TZC_TREGION_ATTRIBUTE0**	S_WR_EN	S_RD_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FILTER_EN
	Reset value	0	0	0																													1
0x114	**TZC_REGION_ID_ACCESS0**	NSAID_WR_EN[15:0]																NSAID_RD_EN[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x118 - 0x11C	Reserved	Res.																															
0x120	**TZC_REGION_BASE_LOW1**	BASE_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x124	**TZC_REGION_BASE_HIGH1**	REGION_BASE_HIGH1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x128	**TZC_REGION_TOP_LOW1**	TOP_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x12C	**TZC_REGION_TOP_HIGH1**	REGION_TOP_HIGH1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x130	**TZC_REGION_ATTRIBUTE1**	S_WR_EN	S_RD_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FILTER_EN
	Reset value	0	0																														0
0x134	**TZC_REGION_ID_ACCESS1**	NSAID_WR_EN[15:0]																NSAID_RD_EN[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x138 - 0x13C	Reserved	Res.																															

表 21. TZC 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	寄存器大小																														
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x140	TZC_REGION_BASE_LOW2	BASE_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x144	TZC_REGION_BASE_HIGH2	REGION_BASE_HIGH2[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x148	TZC_REGION_TOP_LOW2	TOP_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x14C	TZC_REGION_TOP_HIGH2	REGION_TOP_HIGH2[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x150	TZC_REGION_ATTRIBUTE2	S_WR_EN	S_RD_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FILTER_EN
	Reset value	0	0																														0
0x154	TZC_REGION_ID_ACCESS2	NSAID_WR_EN[15:0]																NSAID_RD_EN[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x158 - 0x15C	Reserved	Res.																															
0x160	TZC_REGION_BASE_LOW3	BASE_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x164	TZC_REGION_BASE_HIGH3	REGION_BASE_HIGH3[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x168	TZC_REGION_TOP_LOW3	TOP_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x16C	TZC_REGION_TOP_HIGH3	REGION_TOP_HIGH3[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x170	TZC_REGION_ATTRIBUTE3	S_WR_EN	S_RD_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FILTER_EN
	Reset value	0	0																														0
0x174	TZC_REGION_ID_ACCESS3	NSAID_WR_EN[15:0]																NSAID_RD_EN[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x178 - 0x17C	Reserved	Res.																															
0x180	TZC_REGION_BASE_LOW4	BASE_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x184	TZC_REGION_BASE_HIGH4	REGION_BASE_HIGH4[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x188	TZC_REGION_TOP_LOW4	TOP_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												

表 21. TZC 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	寄存器大小																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x18C	**TZC_REGION_TOP_HIGH4**	REGION_TOP_HIGH4[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x190	**TZC_REGION_ATTRIBUTE4**	S_WR_EN	S_RD_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FILTER_EN
	Reset value	0	0																														0
0x194	**TZC_REGION_ID_ACCESS4**	NSAID_WR_EN[15:0]																NSAID_RD_EN[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x198 - 0x19C	Reserved	Res.																															
0x1A0	**TZC_REGION_BASE_LOW5**	BASE_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x1A4	**TZC_REGION_BASE_HIGH5**	REGION_BASE_HIGH5[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1A8	**TZC_REGION_TOP_LOW5**	TOP_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x1AC	**TZC_REGION_TOP_HIGH5**	REGION_TOP_HIGH5[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1B0	**TZC_REGION_ATTRIBUTE5**	S_WR_EN	S_RD_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FILTER_EN
	Reset value	0	0																														0
0x1B4	**TZC_REGION_ID_ACCESS5**	NSAID_WR_EN[15:0]																NSAID_RD_EN[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1B8 - 0x1BC	Reserved	Res.																															
0x1C0	**TZC_REGION_BASE_LOW6**	BASE_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x1C4	**TZC_REGION_BASE_HIGH6**	REGION_BASE_HIGH6[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C8	**TZC_REGION_TOP_LOW6**	TOP_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x1CC	**TZC_REGION_TOP_HIGH6**	REGION_TOP_HIGH6[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1D0	**TZC_REGION_ATTRIBUTE6**	S_WR_EN	S_RD_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FILTER_EN
	Reset value	0	0																														0
0x1D4	**TZC_REGION_ID_ACCESS6**	NSAID_WR_EN[15:0]																NSAID_RD_EN[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 21. TZC 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	寄存器大小																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1D8 - 0x1DC	Reserved	Res.																															
0x1E0	**TZC_REGION_BASE_LOW7**	BASE_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x1E4	**TZC_REGION_BASE_HIGH7**	REGION_BASE_HIGH7[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1E8	**TZC_REGION_TOP_LOW7**	TOP_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x1EC	**TZC_REGION_TOP_HIGH7**	REGION_TOP_HIGH7[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1F0	**TZC_REGION_ATTRIBUTE7**	S_WR_EN	S_RD_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FILTER_EN
	Reset value	0	0																														0
0x1F4	**TZC_REGION_ID_ACCESS7**	NSAID_WR_EN[15:0]																NSAID_RD_EN[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1F8 - 0x1FC	Reserved	Res.																															
0x200	**TZC_REGION_BASE_LOW8**	BASE_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x204	**TZC_REGION_BASE_HIGH8**	REGION_BASE_HIGH8[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x208	**TZC_REGION_TOP_LOW8**	TOP_ADDRESS_LOW[19:0]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x20C	**TZC_REGION_TOP_HIGH8**	REGION_TOP_HIGH8[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x210	**TZC_REGION_ATTRIBUTE8**	S_WR_EN	S_RD_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FILTER_EN
	Reset value	0	0																														0
0x214	**TZC_REGION_ID_ACCESS8**	NSAID_WR_EN[15:0]																NSAID_RD_EN[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x218 - 0xFCC	Reserved	Res.																															
0xFD0	**TZC_PID4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PER_ID_4[7:0]							
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	**TZC_PID5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PER_ID_5[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

表 21. TZC 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	寄存器大小																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFD8	**TZC_DRASC_PID6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PER_ID_6[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0xFDC	**TZC_PID7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PER_ID_7[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0xFE0	**TZC_PID0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PER_ID_0[7:0]							
	Reset value																									0	1	1	0	0	0	0	0
0xFE4	**TZC_PID1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PER_ID_1[7:0]							
	Reset value																									1	0	1	1	0	1	0	0
0xFE8	**TZC_DRASC_PID2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PER_ID_2[7:0]							
	Reset value																									0	0	1	0	1	0	1	1
0xFEC	**TZC_PID3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PER_ID_3[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	**TZC_CID0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP_ID_0[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	**TZC_CID1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP_ID_1[7:0]							
	Reset value																									1	1	1	1	0	0	0	0
0xFF8	**TZC_CID2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP_ID_2[7:0]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	**TZC_CID3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP_ID_3[7:0]							
	Reset value																									1	0	1	1	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

7 DDR 物理接口控制 (DDRPHYC)

Portions Copyright (c) Synopsys, Inc. 保留所有权利。使用须经许可。

7.1 DDRPHYC 特性

- 16 位 DDR PHY 兼容 JEDEC 标准的 DDR3/3L、LPDDR2 和 LPDDR3 SDRAM。
- 16 位（2 字节通道）接口只允许以 16 位运行 SDRAM（全填充接口）
- 采用 DDR3/3L 时工作频率最高可达 533 MHz (1066 MT/s)。
- 采用 LPDDR2/3 时工作频率最高可达 533 Mhz (1066 Mt/s)。
- 提供 PHY Utility Block (PUBL) 以支持 PHY 初始化、控制和校准。
- 符合 DFI 2.1 标准的接口连接 DDRCTRL 控制器。
- 具有动态 PVT 补偿的可编程输出和 ODT 阻抗。
- 在 PHY 中嵌入动态漂移检测，以便控制器进行动态漂移补偿。
- 利用 Master-Slave DLL 实现精确时序管理。
- 基于通道的架构，具有 2 个字节通道 + 1 个命令/地址通道。
- 支持 IDDq 和 DLL 特性的测试模式。

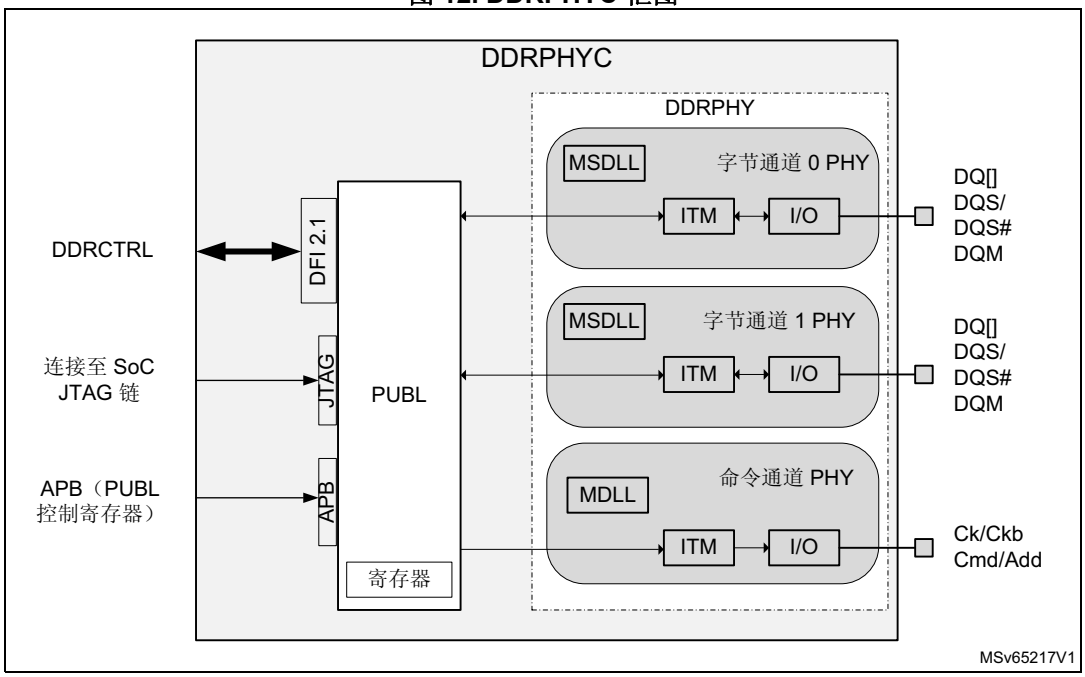
7.2 DDRPHYC 框图

DDRPHYC 框图如*图 12* 所示。

DDRPHYC 包括两个部分：

- DDRPHY 物理接口，由一个命令通道和 2 字节通道组成，用于在 SDRAM 接口生成具有准确时序的信号。
- PUBL（PHY Utility Block 逻辑）用于处理 PHY，通过 APB 和 JTAG 接口控制。

图 12. DDRPHYC 框图



7.3 DDRPHYC SDRAM 接口

DDRPHYC SDRAM 接口在*表 22* 中列出。

表 22. DDRPHYC I/O 列表

引脚名称	DDR3/3L	LPDDR2/3	说明
DDR_DQ15 到 DDR_DQ8	是	是	字节通道 1 DQ
DDR_DQS1P	是	是	字节通道 1 DQS
DDR_DQS1N	是	是	字节通道 1 DQS#
DDR_DQM1	是	是	字节通道 1 DQM
DDR_DQ7 到 DDR_DQ0	是	是	字节通道 0 DQ
DDR_DQS0P	是	是	字节通道 0 DQS
DDR_DQS0N	是	是	字节通道 0 DQS#
DDR_DQM0	是	是	字节通道 0 DQM
DDR_A15 到 DDR_A0	是	是(1)	LPDDR2/3 的行/列地址/CA[] 引脚
DDR_BA2 到 DDR_BA0	是	否	Bank 地址
DDR_CKE	是	是	CKE
DDR_CLKP	是	是	CK
DDR_CLKN	是	是	CK#
DDR_RASN	是	否	RAS#
DDR_CASN	是	否	CAS#

表 22. DDRPHYC I/O 列表（续）

引脚名称	DDR3/3L	LPDDR2/3	说明
DDR_WEN	是	否	WE#
DDR_CSN	是	是	CS#
DDR_ODT	是	否	ODT
DDR_RESETN	是	否	RESET#
DDR_VREF1 到 DDR_VREF3	是	是(2)	连接到 Vref=VDDQ/2
DDR_ZQ	是	是	连接到外部 R =240 Ω ±1%
DDR_DTO0	测试	测试	保留用于 DLL 测试
DDR_DTO1	测试	测试	保留用于 DLL 测试
DDR_ATO	测试	测试	保留用于 DLL 测试

1. 对于 LPDDR2/3：DDR_A[9:0] 用作 Ca[9:0]，不使用 DDR_A[15:10]。
2. DDR_VREF1：命令通道参考，DDR_VREF2：字节通道 3 和 1 公共参考，DDR_VREF3：字节通道 2 和 0 公共参考。

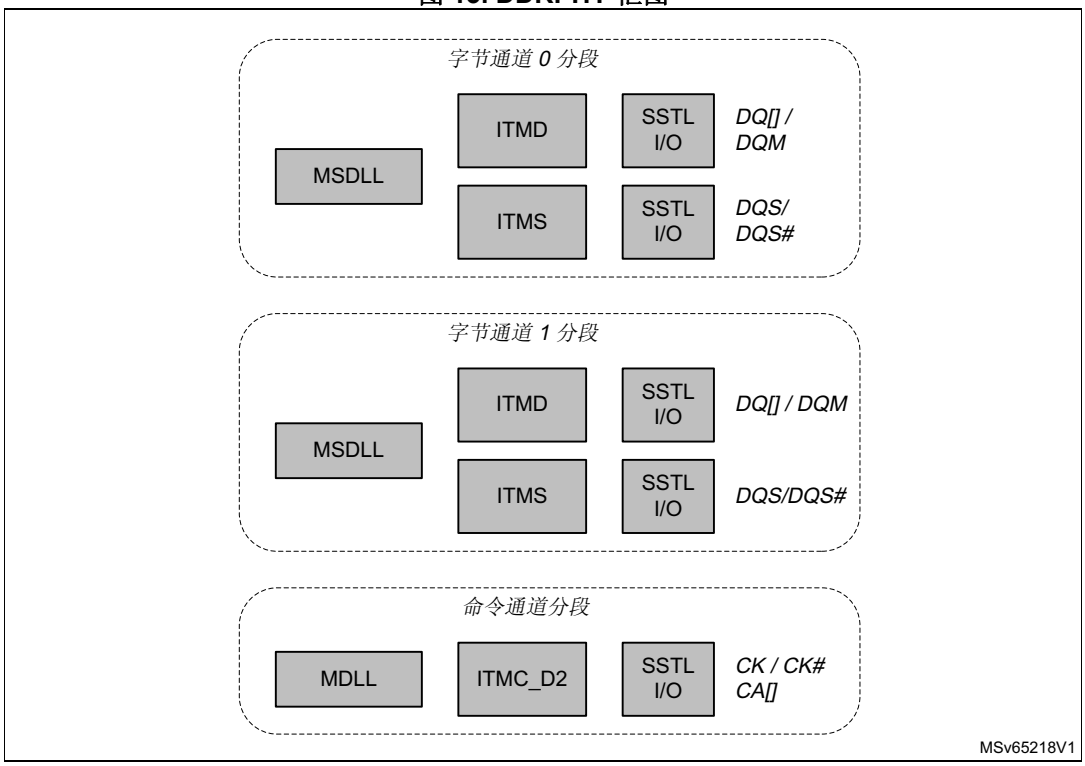
7.4 DDRPHY（DDR 多 PHY IP 内核）

7.4.1 DDRPHY 概览

DDRPHY 框图如*图 13* 所示。它采用结构化布局，实现了 2 字节数据通道和一个地址/命令通道，并具有：

- MDLL 和 MSDLL，提供具有 90 度相移的精确时序
- ITM：接口时序模块，类型如下：
  - ITMD，用于 DQ/DM 信号，
  - ITMS，用于 DQS/DQS# 信号，
  - ITMC_D2，用于地址/命令和 CK/CK# 信号。
- SSTL I/O。

图 13. DDRPHY 框图



7.4.2 延迟锁相环 (DLL)

DLL 如*图 14* 所示。

Master DLL (MDLL) 使用时钟输入 (clk_in) 生成四个时钟输出，每个输出以四分之一个时钟周期 (90º) 增量延迟。这四个时钟相位（clk_0、clk_90、clk_180 和 clk_270）在广泛的频率范围内以非常高的准确性和低抖动生成。

Master-Slave DLL (MSDLL) 由一个 MDLL 和一个从设备延迟对组成。

从设备延迟对使用 MDLL 的时序参考为 DQS/DQS# 输入提供高精确度的 90º 延迟，并分别生成 dqs_90 和 dqsb_90。

DLL 主要特性：

- Master-Slave DLL 架构可用于复杂的时序，并通过以下组成部分降低总功耗：
  - Master DLL，用于 SDRAM 命令/地址接口和通用主设备时序。
  - Master-Slave DLL，用于 SDRAM 数据通道 (DQ/DQS)，进行写入和读取数据捕获。
- 工作频率范围为 100 MHz 到 533 MHz。
- 可选：LPDDR 的 DLL 旁路模式，适用频率范围 10 MHz 到 200 MHz，用于低速功能和 IDDQ 测试。
- 多相输出：0 度、90 度、180 度和 270 度。
- 全面的 DLL 控制，用于微调系统时序。
- 模拟和数字 DLL 测试输出，用于设备特性分析。

图 14. MDLL 框图

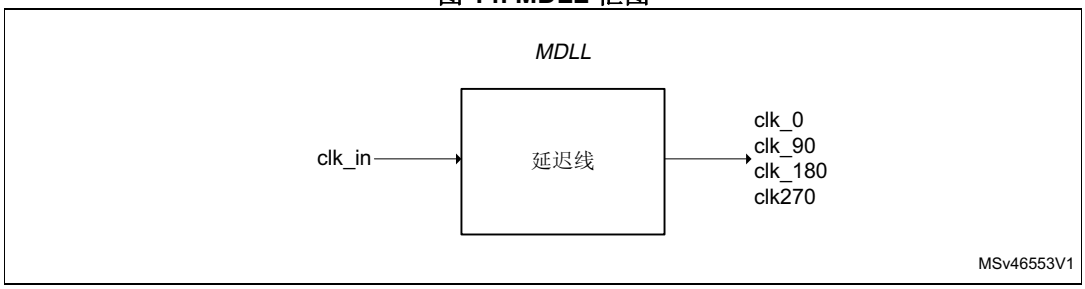
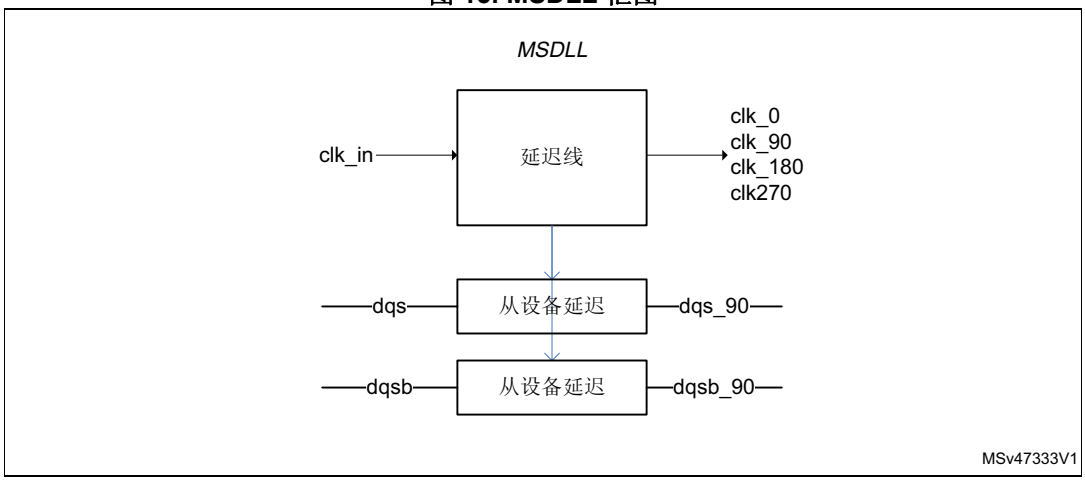


图 15. MSDLL 框图



DLL 设置由控制总线 dllctrl[51:0] 控制，如*表 23：MSDLL 控制位*所述。

表 23. MSDLL 控制位

控制总线	信号名称	MSDLL PUBL 寄存器	MDLL PUBL 寄存器	默认	说明
dllctrl[51]	lltm	DXnDLLCR[19]	N/A	0b0	锁定回送测试模式选择：0：正常，1：回送
dllctrl[50]	test_hizb_a	DXnDLLCR[18]	ACDLLCR[18]	0b0	模拟测试输出三态控制 0：高阻态，1：模拟低阻态
dllctrl[49:46]	sl_phase_trm[3:0]	DXnDLLCR[17:14]	N/A	0b0000	从设备锁相微调（4 位编码，从 36 到 144 度，每步 18 度）
dllctrl[45:44]	sl_bypass_start_up[1:0]	DXnDLLCR[13:12]	N/A	0b00	从设备自动启动旁路
dllctrl[43:38]	fb_trm[5:0]	DXnDLLCR[11:6]	ACDLLCR[11:6]	0b000000	主设备反馈延迟调整。fb_trm[5:3]：前馈延迟，fb_trm[2:0]：反馈延迟
dllctrl[37:32]	sl_fb_trm[5:0]	DXnDLLCR[5:0]	N/A	0b000000	从设备反馈延迟调整（fb_trm[5:3]：ffd，fb_trm[2:0]：fbk）
dllctrl[29]	lock_det_en	DLLGCR[29]	N/A	0b0	锁相检测器使能
dllctrl[28:27]	fdtrm_sl[1:0]	DLLGCR[28:27]	DLLGCR[28:27]	0b00	从设备旁路固定延迟微调（-10%、0、+10%、+20%）
dllctrl[26:24]	sl_bias_trm[6:4]	DLLGCR[26:24]	DLLGCR[26:24]	0b011	从设备偏置生成器控制电压微调：VC

表 23. MSDLL 控制位（续）

控制总线	信号名称	MSDLL PUBL 寄存器	MDLL PUBL 寄存器	默认	说明
dllctrl[23]	bps200	DLLGCR[23]	DLLGCR[23]	0b0	旁路频率选择 （0：0-100 Mhz，1：0-200 MHz）
dllctrl[22:20]	sl_bias_trm[2:0]	DLLGCR[22:20]	DLLGCR[22:20]	0b111	从设备偏置生成器频率微调：Fmax
dllctrl[18:16]	bias_trm[6:4]	DLLGCR[18:16]	DLLGCR[18:16]	0b011	主设备偏置生成器控制电压微调：VC
{dllctrl[19],dllctrl[15]}	fdtrm[1:0]	DLLGCR[19],DLLGCR[15]	DLLGCR[19],DLLGCR[15]	0b00	主设备旁路固定延迟微调 （-10%、0、+10%、+20%）
dllctrl[14:12]	bias_trm[2:0]	DLLGCR[14:12]	DLLGCR[14:12]	0b111	主设备偏置生成器频率微调：Fmax
dllctrl[11]	test_ctrl_switch	DLLGCR[11]	N/A	0b0	测试控制：模数切换。0：主设备，1：从设备
dllctrl[10:9]	test_ctrl_a[1:0]	DLLGCR[10:9]	DLLGCR[10:9]	0b00	测试控制： 在 ana 测试时选择模拟信号
dllctrl[8:6]	test_ctrl_d[2:0]	DLLGCR[8:6]	DLLGCR[8:6]	0b000	测试控制： 在 dig 测试时选择数字信号
dllctrl[5]	test_ctrl_en	DLLGCR[5]	DLLGCR[5]	0b0	为模拟和数字测试使能测试控制
dllctrl[4:2]	ipump_trm[2:0]	DLLGCR[4:2]	DLLGCR[4:2]	0b000	电荷泵电流微调 （0：最大电流）

7.4.3 接口时序模块 (ITM)

ITM 用于生成具有精准时序的信号：

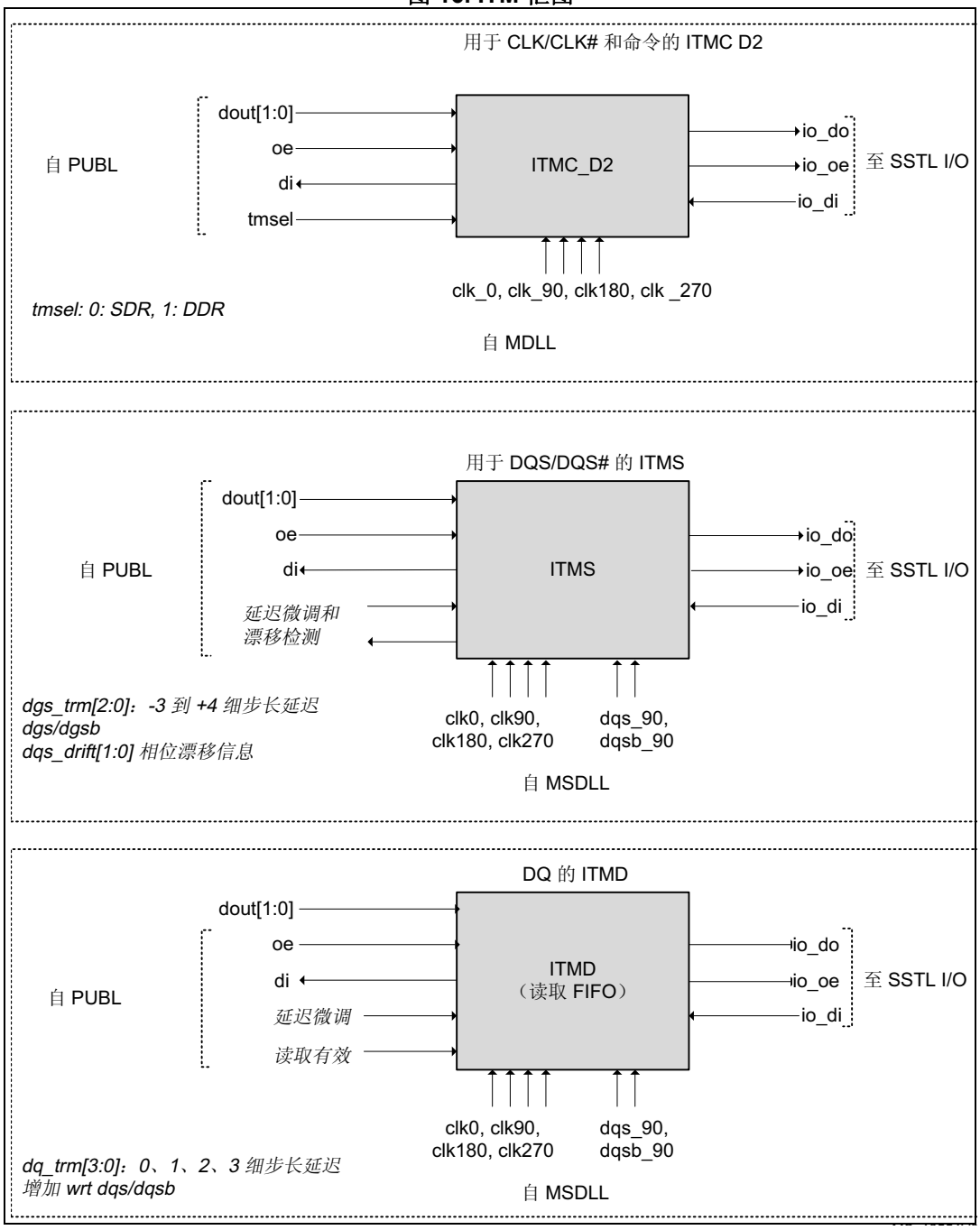
- 命令/地址总线 CK/CK# 的 90 度相位延迟
- 写入 DQ/DQM 字节通道输出的 DQS/DQS# 的 90 度相位延迟
- 以传入 DQS/DQS# 的 90 度相位延迟对读取的 DQ 输入采样

有三种类型的 ITM：

- ITMC_D2，用于时钟、地址和命令通道。
- ITMS，用于 DQS/ DQS# 差分选通。
- ITMD，用于 DQ、DQM 字节通道。

ITM 如*图 16* 所示。

图 16. ITM 框图



ITMC-D2 为 LPDDR2/3 执行 SDR 到 DDR 的信号转换 (tmsel =1)。

图 17 和*图 18* 所示为根据 SDRAM 类型设置时序的 ITMC_D2 连接。

图 17. 用于 LPDDR2/3 的 ITMC_D2

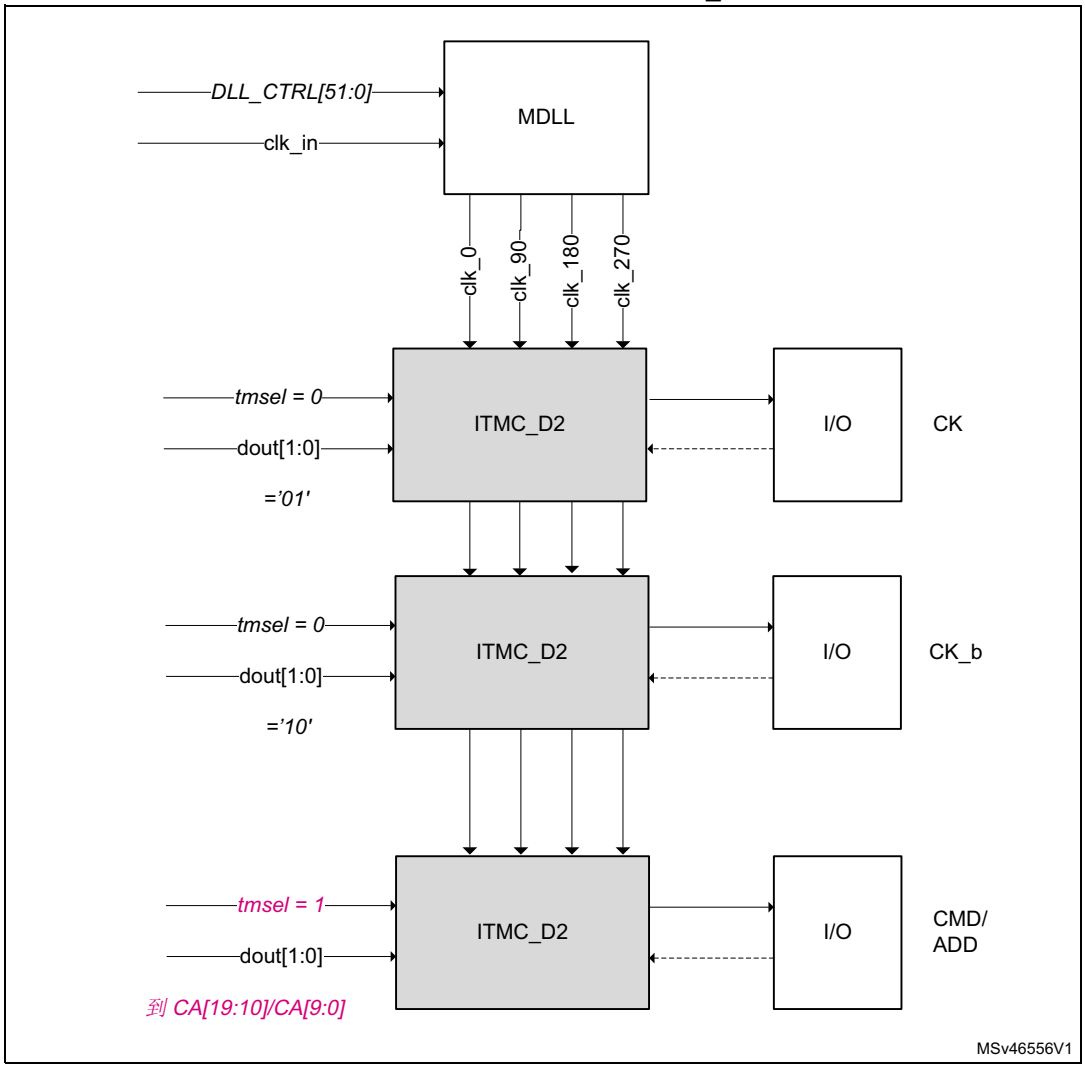
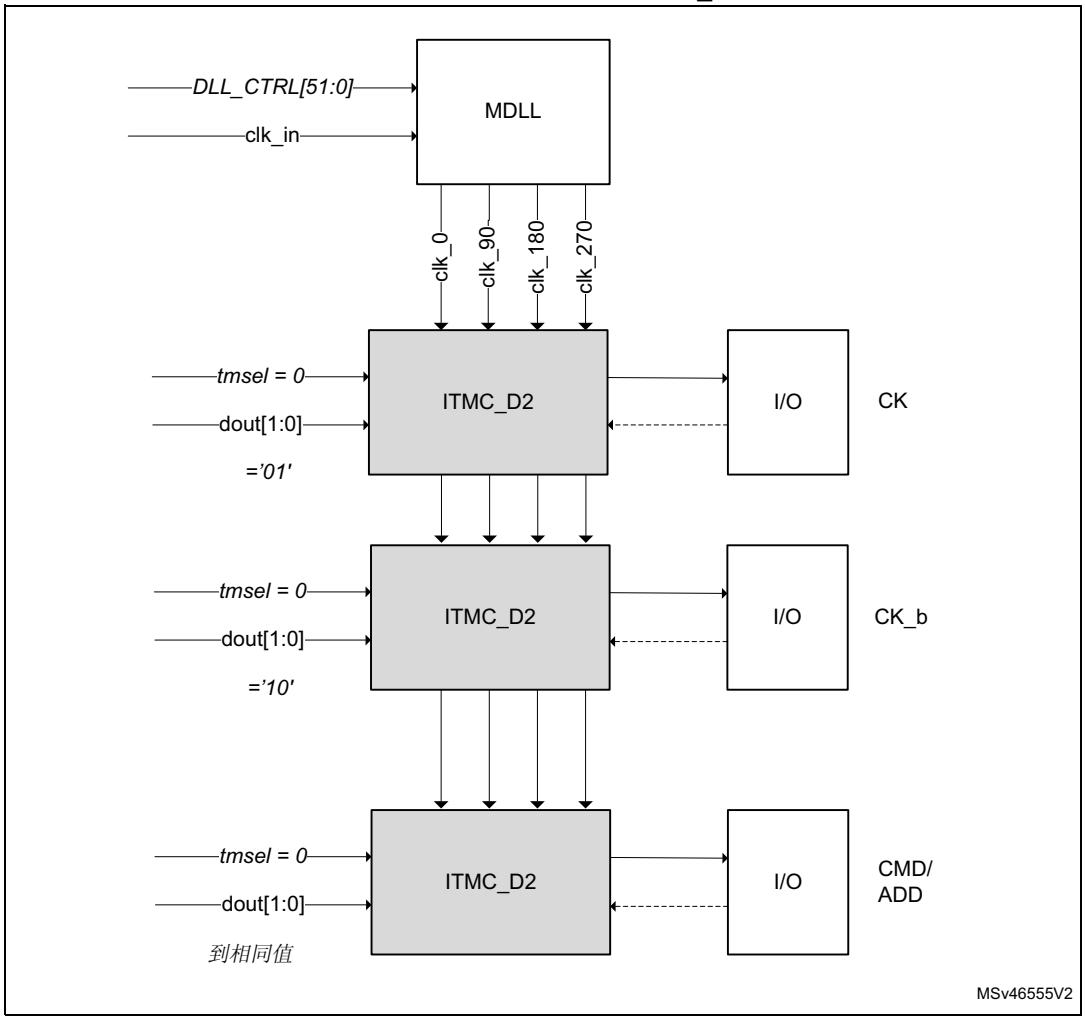


图 18. 用于 DDR3/3L 的 ITMC_D2



ITMS 用于具有以下特性的 DQ 选通 (DQS/DQS#)：

- 输出方向上的 SDR 到 DDR 信号转换。
- 相位选择定位
- 细步长延迟
- 漂移监视支持

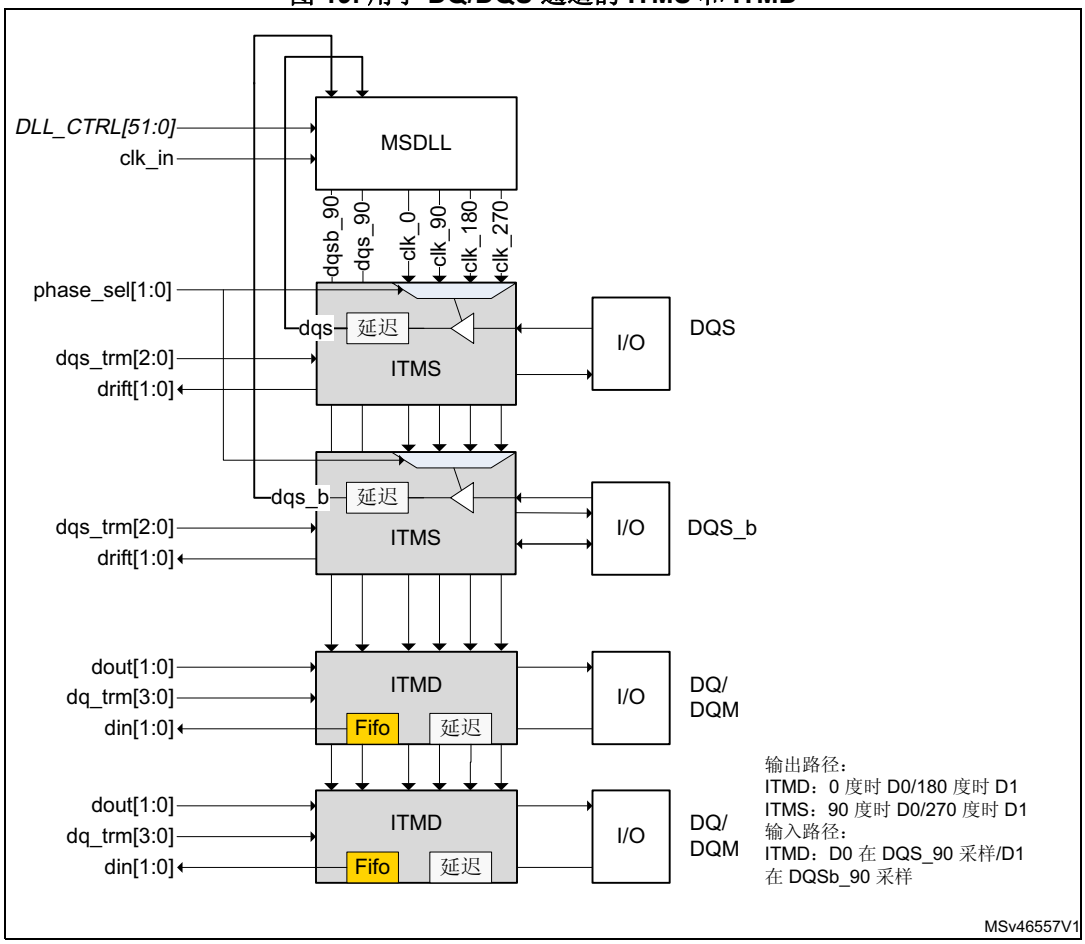
ITMS 连接和时序如*图 19* 所示。

ITMD 用于具有以下特性的数据通道（DQ[] 和 DQM）：

- 输入和输出方向上的 SDR 到 DDR 信号转换。
- 以 DQS 和 DQS# 90 度相位在读取的 FIFO 中对传入 dq 输入进行采样。
- 细步长采样延迟。

ITMD 连接和时序如*图 19* 所示。

图 19. 用于 DQ/DQS 通道的 ITMS 和 ITMD



7.4.4 SSTL IO

SSTL IO 主要特性：

- DDR3/DDR3L/LVCMOS 工作模式
- DDR3/DDR3L SDRAM 与 JEDEC 标准兼容
- 可编程 DDR3/3L 端接 (ODT)
- 可编程输出阻抗
- PVT 补偿 ODT 和输出阻抗
- 驱动器和接收器掉电控制
- 嵌入式边界扫描支持逻辑
- PAD 和内部回送模式
- 支持内核和 I/O 掉电选项
- 保持特性在 VDD 掉电期间维持 I/O 单元状态

ZQ 校准单元：

ZQ 校准单元为上拉和下拉 ODT 端接及 SSTL I/O 的输出阻抗提供独立的校准功能。

有四个感测模块，分别为各个阻抗元件提供独立的感测功能：

- 上拉端接阻抗。
- 下拉端接阻抗。
- 上拉输出阻抗。
- 下拉输出阻抗。

这四个元件使用校准选择输入 ZCAL[1:0] 连续校准。校准序列为：

1. 输出阻抗下拉。
2. 输出阻抗上拉。
3. 片内终结 (ODT) 下拉。
4. ODT 上拉。

7.5 PUBL

PUBL 旨在简化 PHY 特性的控制。它支持 PHY 初始化、DQS 门控训练以及可编程配置控制。

PUBL 支持：

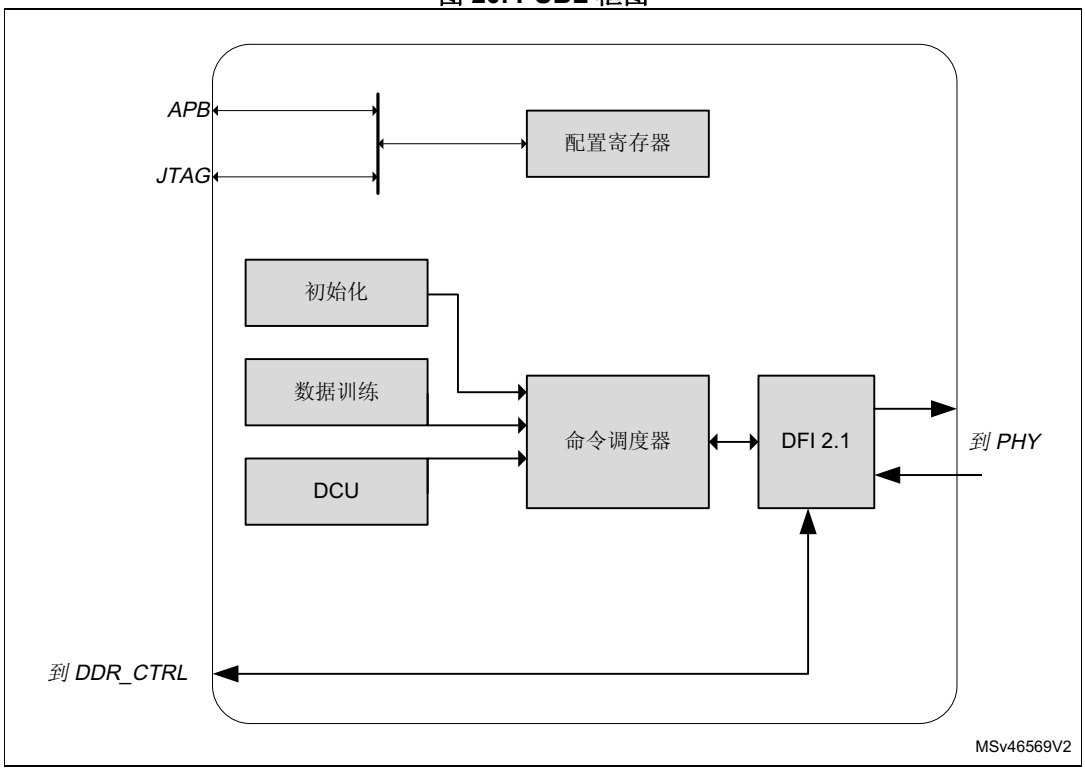
1. DDRPHY 和 SDRAM 初始化序列，适用于所有类型内存：DDR3/3L、LPDDR2 和 LPDDR3。
2. 内置 DQS 门控训练 (DQSTRN)，帮助对读取的 DQS 门控时序编程。
3. DRAM 命令单元 (DCU)，帮助编程 DRAM 以进行时序校准。

PUBL 支持 DFI 2.1 接口以连接 DDRCTRL。

PUBL 包含可通过 APB 接口访问的配置寄存器。它还提供了测试时使用的 JTAG 接口访问。

图 20 显示了 PUBL 的简化框图。

图 20. PUBL 框图



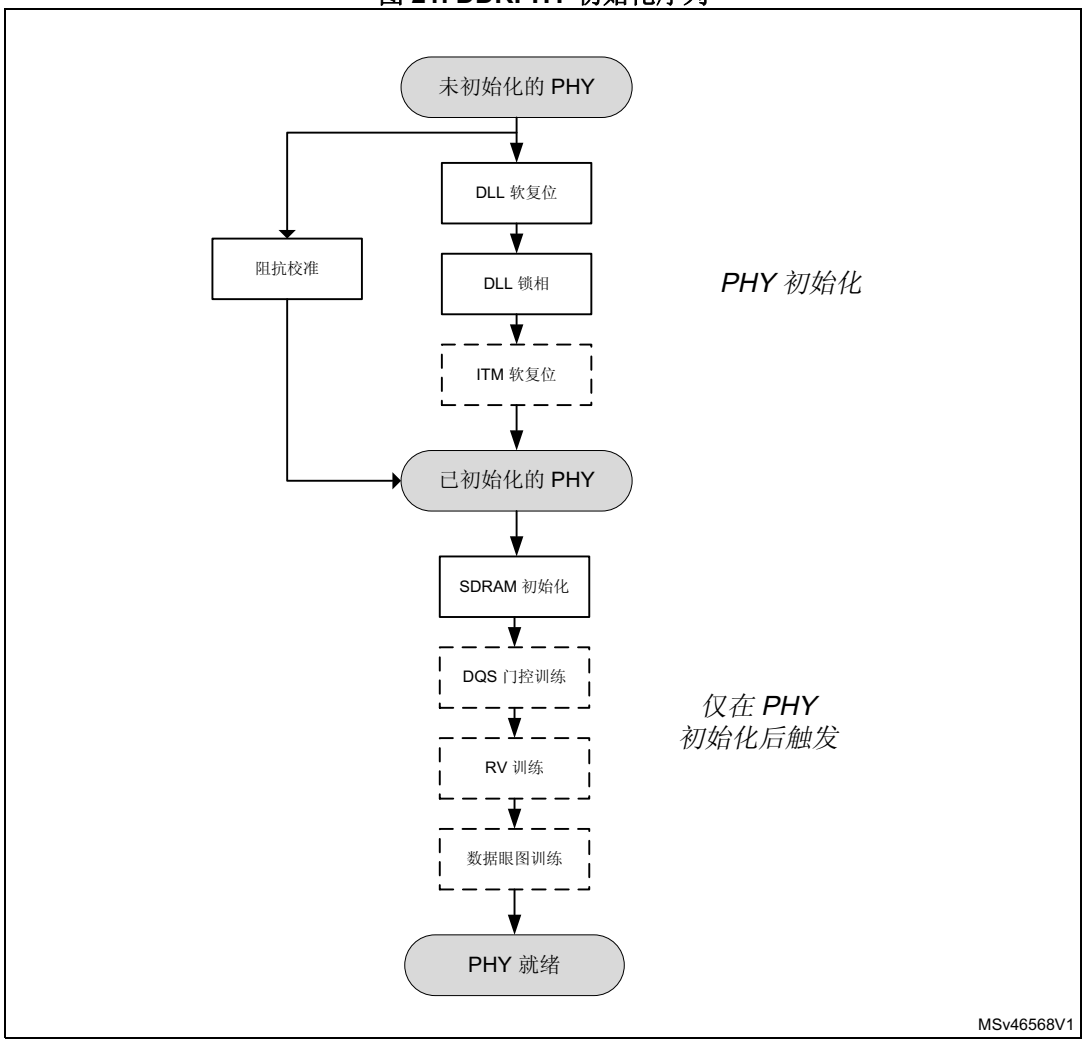
以下是 PUBL 的不同设计模块和功能的简要概述：

- **配置寄存器**：包括配置、控制、训练和自测 PHY 所需的所有寄存器。配置模块还包含初始化 DRAM 所需的寄存器。
- **初始化**：用于自动初始化 PHY，包括 DLL 锁相和 I/O 阻抗的校准。初始化模块还实现了所有支持的 DRAM 的内置初始化。
- **数据训练**：用于训练 PHY 以优化时序裕量。这包括在读取、读取有效生成期间找到 DQS 门控窗口的最佳位置及优化读取的数据眼图。
- **DRAM 命令单元 (DCU)**：允许通过配置端口发出 DRAM 命令。
  这可用来实现软件 DRAM 初始化。也可用于芯片设计调试以及 PHY 和 DDR 接口的特性分析。
- **命令调度器**：复用来自内部 PUBL 模块和外部存储控制器的命令，并以 DFI 2.1 格式生成正确的 DRAM 事务。来自外部控制器的命令原样传递给 DFI 2.1 模块。
- **DFI 2.1 接口**：在控制器 DDRCTRL 与 PHY 之间实现了一个符合 DFI 2.1 标准的接口。

7.5.1 PHY 和 SDRAM 初始化

图 21 显示了 PHY 初始化序列的高级框图。

图 21. DDRPHY 初始化序列



初始化序列有两个阶段。第一阶段（*图 21* 中以实线显示）在复位时自动发生，用途如下：

1. 在配置复位之前和期间，PHY 为未初始化状态，保持此状态直到复位清零。
2. 复位清零时，PHY 进入 DLL 初始化阶段。
   通过向 DDR PHY 初始化寄存器 (DDRPHYC_PIR) 的 LOCKBYP 位写入“1”，可以随时旁路该阶段。
3. 在 DLL 初始化的同时，阻抗校准阶段也会在复位清零时开始。
   也可以通过向 DDR PHY 初始化寄存器 (DDRPHYC_PIR) 的 ZCALBYP 位写入“1”旁路该阶段。
4. 如果用户触发了 PHY 初始化序列，可以选择向 ITM 发出软复位。
   复位时自动触发的初始化不会向 ITM 发出软复位，因为这些组件已经通过主复位而进行了复位。
5. 在 DLL 初始化和阻抗校准阶段完成，并且 ITM 复位之后，PHY 即完成了初始化。

如果旁路了这些阶段，则需要在后期由用户通过软件执行这些阶段或触发这些阶段，之后才能使用 PHY。

PHY 初始化完成后，开始第二阶段的初始化。这一阶段的每一步都由用户 (APB 接口) 或 DDR_CTRL 触发：

1. 用户或存储控制器执行 SDRAM 初始化序列（参见“SDRAM 初始化”）。必须先将 DRAM 模式寄存器和必要的时序参数编程到 PUBL 中，然后再触发 SDRAM 初始化。
2. SDRAM 初始化后，用户或存储控制器执行或触发 PUBL，以执行 DQS 门控训练（参见 *第 7.5.6 节：DQS 门控训练 (DQSTRN)* 中的“内置 DQS 门控训练”）。必须先初始化 SDRAM，然后才能触发 DQS 门控训练。
3. 用户或存储控制器执行（或触发 PUBL 执行）读数据有效训练。PHY 初始化可以由复位触发或是 PIR 触发
4. 用户或存储控制器执行（或触发 PUBL 执行）读数据眼图训练。
   注意：当前版本的 PUBL 并不包含读数据眼图训练设计。
5. PHY 现在已准备就绪，可进行 SDRAM 读/写访问。

PUBL 有一个内置的 SDRAM 初始化程序，可以通过软件或 DDR_CTRL 向 PHY 初始化寄存器 (DDRPHYC_PIR) 执行写入来触发该程序。

内置在 PUBL 中的初始化程序是通用的，无需知道外部 SDRAM 的类型或配置即可正确执行该程序。该程序根据 PUBL 支持的最快和最慢 SDRAM 的相关 JEDEC 规范设计，可生成通用初始化序列。这一通用序列适用于 DDR3、LPDDR2 和 DDR SDRAM。

建议使用此内置 PUBL 程序来初始化 SDRAM。然而，由于系统调试等原因，可能会出现内置 PUBL DRAM 初始化未被触发，DRAM 初始化由软件或控制器执行的情况。

在这些情况下，系统必须先等待 PHY 进行初始化，即 DLL 已锁相且阻抗校准完成，然后必须向 DDRPHYC_PIR 的 INIT 位写入“1”，并将 DDRPHYC_PIR 的 CTLDINIT 位设置为“1”（控制器初始化）或“0”（软件或 PUBL 初始化），以通知 PUBL，DRAM 初始化在稍后通过软件、控制器或重新触发 PUBL 完成。
软件或控制器随后通过向 DRAM 发送相关命令，同时遵守初始化序列的各种时序要求，来执行初始化序列。

必须先将必要的 SDRAM 配置编程到 PUBL 中，然后再触发 SDRAM 初始化。

只能在 PHY 已经初始化之后才能触发 SDRAM 初始化。

下述所有 SDRAM 初始化时序参数在 PTR1 和 PTR2 寄存器中控制器时钟周期内均可编程。

以下各节描述了不同类型 SDRAM 的内置 SDRAM 初始化程序。

7.5.2 DDR3 初始化序列

DDR3 SDRAM 的初始化步骤如下：

1. 保持 RESET# 低电平至少 200 us（上电初始化）或 100 ns（通电初始化）（可选）。PUBL 从复位置 1 开始将 RESET# 驱动为低电平，因此如果 RESET# 保持低电平的时间充足，可以跳过此步骤。
2. RESET# 清零后，在 CKE 保持低电平的情况下等待至少 500 us。
3. 执行 NOP 并将 CKE 驱动为高电平。
4. 等待至少 tXPR 的时间
   小心：PUBL 寄存器、SDRAM 模式寄存器以及存储控制器内的等效寄存器位字段的编程必须一致。寄存器位字段之间不匹配可能导致事务失败。在开始任何 SDRAM 事务之前，先验证所有寄存器位字段编程是否一致。

5. 发出加载模式寄存器 2 (MR2) 命令。
6. 发出加载模式寄存器 3 (MR3) 命令。
7. 发出加载模式寄存器 (MR1) 命令（以设置参数及使能 DLL）。
8. 发出加载模式寄存器 (MR0) 命令以设置参数并复位 DLL。
9. 发出 ZQ 校准命令。
10. 等待 512 个 SDRAM 时钟周期以完成 DLL 锁相 (tDLLK) 和 ZQ 校准 (tZQinit)。
此等待时间与步骤 8 相关，即与 SDRAM 命令总线上发出 DLL 复位命令的时间相关。

7.5.3 LPDDR2 初始化序列

LPDDR2 SDRAM 的初始化步骤如下：

1. 在 CKE 驱动为低电平的情况下等待至少 100 ns (tINIT1)。
2. 执行 NOP 并将 CKE 设置为高电平。
3. 等待至少 200 us (tINIT3)。
4. 发出 RESET 命令。
5. 等待至少 1 us + 10 us (tINIT4 + tINIT5)。
6. 发出 ZQ 校准命令。
7. 等待至少 1 us (tZQINIT)。
8. 向 MR1 发出写入模式寄存器命令。
9. 向 MR2 发出写入模式寄存器命令
10. 向 MR3 发出写入模式寄存器命令

7.5.4 初始化触发和旁路

图 21 中显示的所有初始化步骤都可以使用 PHY 初始化寄存器 (DDRPHYC_PIR) 来触发。向 DDRPHYC_PIR 寄存器的 INIT 位写入“1”即开始初始化过程，并由 DDRPHYC_PIR 寄存器位选择要运行的程序。如果选择了多个程序，则它们会按照*图 21* 中显示的顺序运行。

程序完成状态通过 PHY 通用状态寄存器 (DDRPHYC_PGSR) 中相应的完成状态位来指示。DDRPHYC_PGSR.IDONE 位指示初始化序列是否总体完成。
重新触发相应程序时，状态寄存器位清零（复位为“0”）。

复位清零将自动触发 PUBL 执行 **DLL 初始化**，包括 DLL 锁定和阻抗校准。一旦 DLL 锁定且阻抗校准完成，SDRAM 初始化和 DQS 门控即可由软件或存储控制器触发或执行。

由于 PUBL 允许在使用 DDRPHYC_PIR 寄存器触发初始化时选择要运行的单独程序，因此在复位清零时自动触发的程序能够单独进行旁路。
这意味着可以通过在 DDRPHYC_PIR 寄存器中相应的旁路寄存器位中写入“1”随时旁路 DLL 锁相和/或阻抗校准。一旦旁路某个程序，它会在内部注册为已完成，并且 PGSR 寄存器中相应的完成状态寄存器位会置 1。

用户应先决定在稍后是重新触发还是执行被旁路的程序，然后才能使用 PHY。

DDRPHYC_PIR 寄存器的 INITBYP 位提供了旁路整个初始化序列的选项

7.5.5 数据训练

成功初始化 DDRPHY 和外部 SDRAM 后，可以训练 DDRPHY 以优化操作时序裕量。
包括：

- 在读取期间进行 DQS 门控训练 (DQSTRN)
- 优化读取有效位置设定训练 (RVTRN)

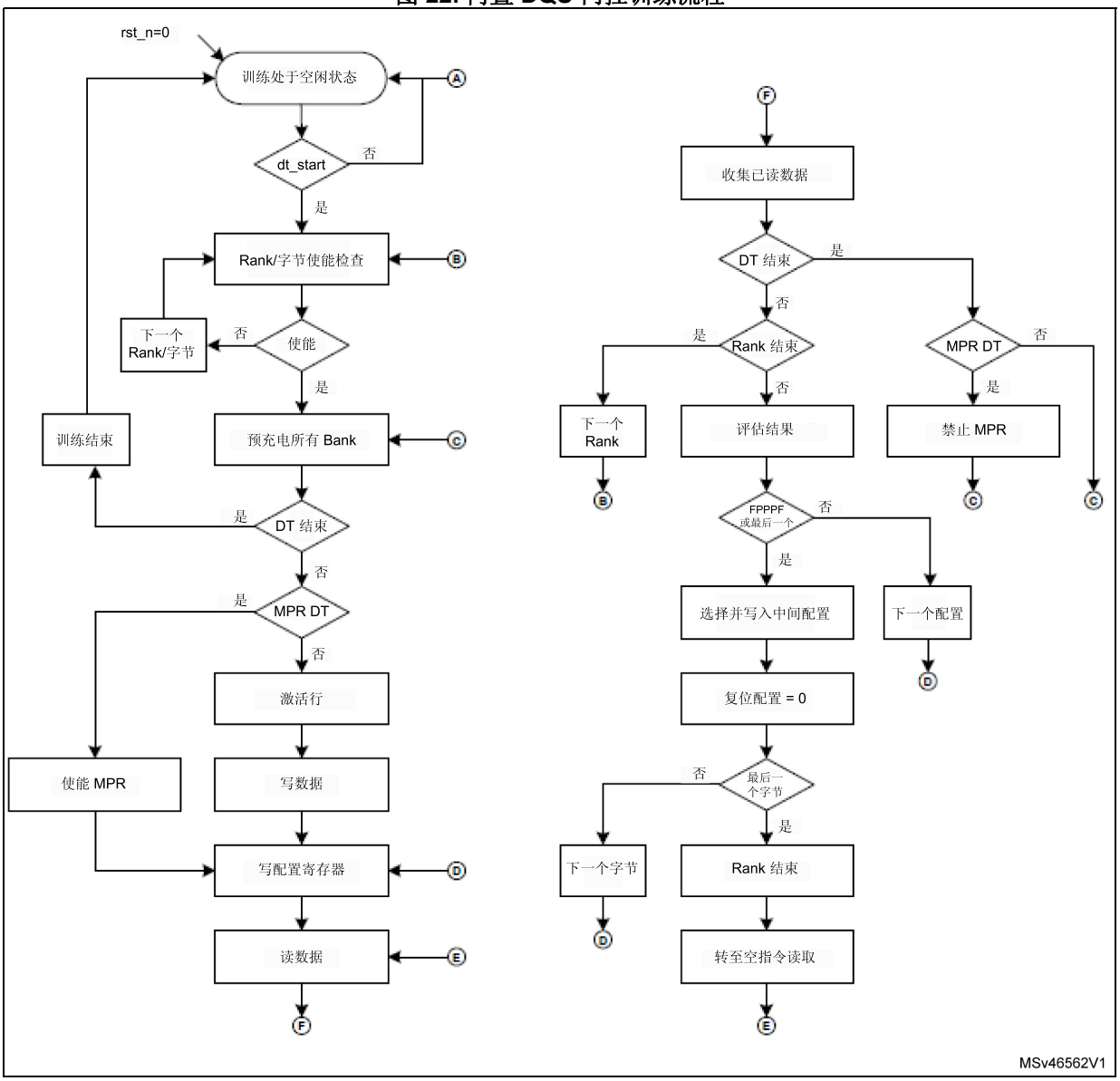
7.5.6 DQS 门控训练 (DQSTRN)

内置 DQS 门控训练：

注： *内置 DQSTRN 是强制项，仅适用于 DDR3/3L，并可使用 MPR 寄存器。*

PUBL 具有一个内置的 DQS 门控训练程序，可以由软件或由存储控制器使用 DDRPHYC_PIR 寄存器触发。*图 22* 显示了在 PUB 中实现的 DQS 门控训练算法的详细流程。图下面是主要训练阶段的介绍。

图 22. 内置 DQS 门控训练流程



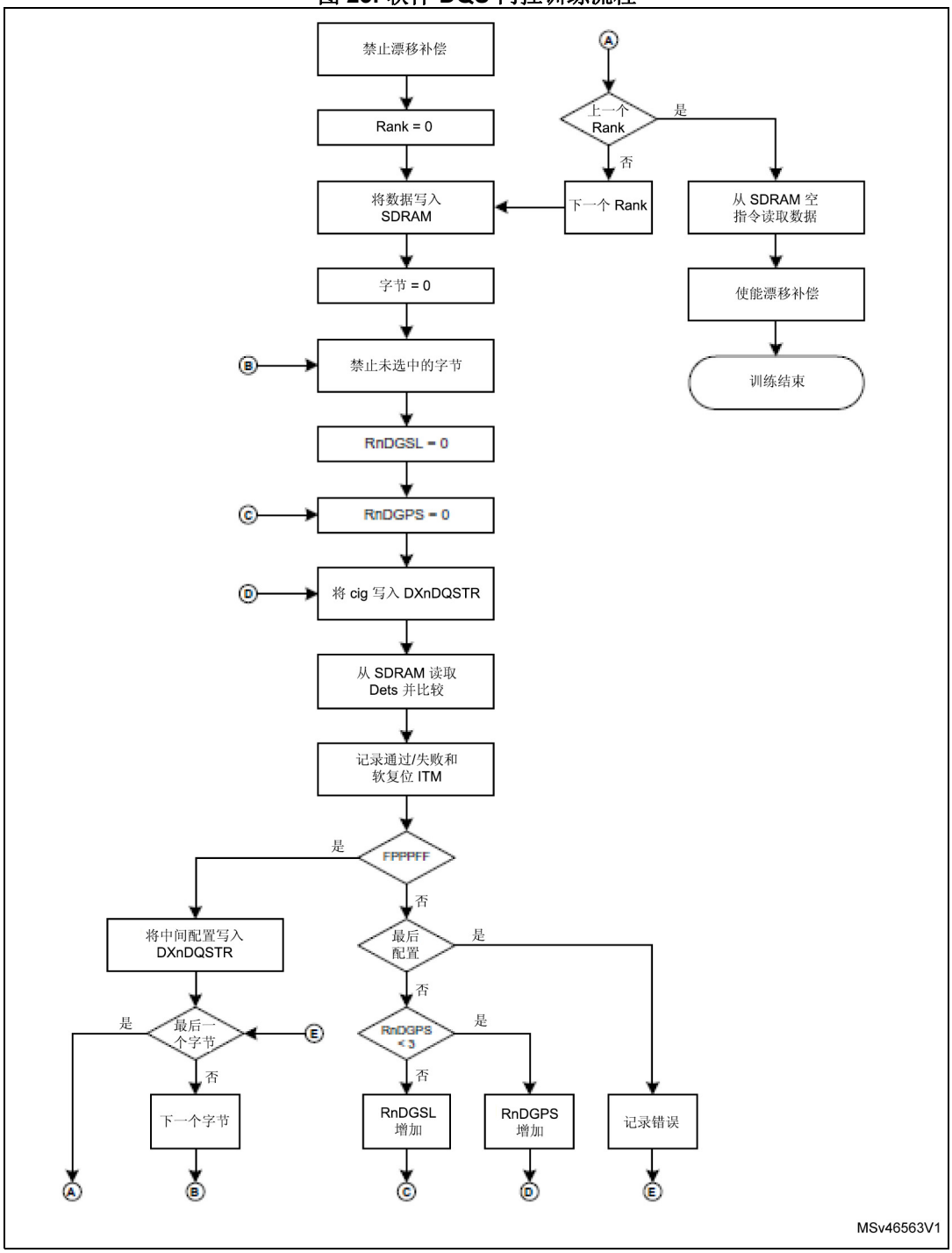
DQS 门控训练阶段如下所述：

1. 数据训练空闲：在复位前和复位期间，数据训练状态机处于空闲状态，并保持该状态到复位清零。用于 DQS 门控的下一个配置寄存器值也将复位为零。这些值是在每次迭代 DQS 门控训练之前写入 DDRPHYC_DXnDQSTR 寄存器的 DQS 门控系统延迟 (R0DGSL) 和 DQS 门控相位选择 (R0DGPS) 位字段中的值。一旦复位清零，FSM 就会继续保持空闲状态，直到数据训练被触发。如果数据训练被触发但未使能 rank 或字节，则 FSM 也将保持空闲状态。否则，FSM 将检查下一个 rank 或字节是否使能。
2. Rank/字节使能检查：这一阶段检查是否使能了要训练的下一个 rank/字节，否则它将跳过该 rank/字节并转向下一个 rank/字节。字节是使用 DDRPHYC_DXnGCR 寄存器中的 DXEN 位使能/禁止的，而 rank 则是使用 DDRPHYC_DXnGCR 寄存器中的 RANKEN 位字段来使能/禁止的。
3. 对所有 bank 进行预充电：在每个 rank 的训练开始前和整个训练结束时，所有 bank 都会进行预充电。如果预充电在训练的最后阶段进行，则 FSM 将进入空闲状态，表示训练结束。否则它会准备 DRAM 以供访问，例如如果选择了使用 MPR 的数据训练，则使能 MPR 寄存器，或激活 DRAM 行以用于读/写访问。
4. 使能 MPR：向 DRAM 发出加载模式寄存器命令以使能 MPR 寄存器。这仅适用于 DDR3 DRAM，并且仅当 DTAR 寄存器中的 MPR 位置 1 时适用。
5. 激活：向 DRAM 发出激活命令以在进行训练数据写入之前打开 bank 行。用于训练的 DRAM 地址在 DTAR 寄存器中可以由用户编程。
6. 写入数据：向 DRAM 发出 8 节拍写入命令以加载训练数据。用于训练的 DRAM 地址和 DRAM 数据在 DTAR、DTDTR0、DTDTR1 寄存器中可由用户编程。
7. 写配置寄存器：一旦数据被写入 DRAM（或者使能了 MPR 寄存器），FSM 将下一个字节的配置设置写入相应的 DQS 门控寄存器位字段（DDRPHYC_R0DGSL 和 DDRPHYC_R0DGPS）。
8. 读取/收集数据：这一阶段在 rank 训练结束或整个训练结束时作为空指令读取执行（以将 PHY ITM 中的漂移状态寄存器复位），或者作为正常读取执行，以评估 DQS 门控配置是否有效。在空指令读取之后，FSM 继续训练下一个 rank，或者，如果整个训练结束，则发出最后一条预充电命令或禁止 MPR。向 DRAM 发出读取命令后，FSM 等待所有读取数据返回。如果读取数据未返回，将会产生超时。对于非空指令读取，FSM 继续评估读取数据的通过/失败状态。
9. 评估结果：读取数据与预期数据进行比较，生成通过/失败状态并保存下来。如果情况是，发现了一个失败-通过-失败区域 (FPPPPF)，或者这是最后一个可能配置（即该字节没有通过的区域），则将选择通过区域的中点并将其写入 DDRPHYC_DXnDQSTR 区域。这表示所选 rank 上的字节训练结束。如果这是该 rank 的最后一个字节，FSM 继续训练下一个 rank，否则它将转到该 rank 的下一个字节。在 rank 的所有字节训练结束时，总会执行一次空指令读取。

软件 DQS 门控训练

DQS 门控训练也可以使用 DDRCTRL 和/或 PUB DCU 由软件执行。*图 23* 所示为 DQS 门控训练软件算法。图下面是主要训练阶段的介绍。

图 23. 软件 DQS 门控训练流程



软件 DQS 门控训练阶段如下所述：

1. 通过向 DDRPHYC_DXnGCR 寄存器中的 DFTCMP 位写入“0”来禁止漂移补偿。
2. 从 rank 0 开始（DDRPHYC 仅支持单个 rank）。
3. 对 SDRAM 执行至少两次写入。任何类型的数据和任何 SDRAM 地址均可用于 DQS 门控训练。但不建议使用全零数据，因为这可能会屏蔽读取数据比较。数据掩码必须设置

为 0 以使能所有字节写入。选择的写入次数必须能确保至少在 SDRAM 上产生八个数据节拍。这意味着，在使用 SDRAM 突发传输长度 4 时，至少需要两个写命令。

4. 从字节 0 开始（即选择字节 0 进行训练）。
5. 禁止所有未选择进行训练的其他字节。这些字节通过在 DDRPHYC_DXnGCR 寄存器中将 DXEN 位写入 1/0 来使能/禁止。为了避免在稍后重新使能已禁止的字节通道时增加编程和延迟 DLL 重锁，建议在禁止任何字节通道之前，将 DDRPHYC_DXnGCR 寄存器中的 PDDISDX 位也设置为 0。这可以防止字节通道因 DDRPHYC_DXnGCR 寄存器中的 DXEN 位为 0 而被禁止时，字节通道的 DLL 进入旁路模式。
6. 从选择的字节 DQS 门控系统延迟开始（DXnDQSTR 寄存器中 RnDGSL=0）。
7. 从选择的字节 DQS 门控相位选择开始（DXnDQSTR 寄存器中 RnDGPS=0）。
8. 写入所选的 DQS 门控配置（所选字节的 DXnDQSTR 寄存器中的 R0DGSL 和 R0DGPS 位）。
9. 从之前写入的 SDRAM 位置执行读取。读取次数必须等于第 3 步中使用的写入次数。将读取的数据与预期（写入）的数据进行比较，并将每个经过训练的 RnDGSL/RnDGPS 配置的通过/失败状态记录为标志序列或历史（例如，FFPPPPFF）。失败是指数据不匹配或返回的数据少于预期。

注： *一种被设计为总是等待正确数量读取数据的控制器可能需要一个超时机制，以防训练后的配置所产生的数据比预期的要少。当使用 PUB DCU 时，这不是问题，因为它不等待预期的读取次数；而是由读取计数状态指示返回的读取数是否少于预期。*

10. 在比较读取的数据并记录了通过/失败状态后，立即发出 ITM 软复位命令以清除 PHY 中读取数据逻辑的状态。这很重要，因为在导致错误 DQS 门控窗口的 RnDGSL/RnDGPS 配置训练结束时，ITM 读取数据 FIFO 指针可能处于错误状态。
11. 如果出现连续两次失败以及部分通过的情况，则表示该 rank 类型的训练结束。在这种情况下，执行以下操作：
    – 选择通过的中间值，并将这些值写入 DDRPHYC_DXnDQSTR 寄存器的相应位字段。
      - 如果这不是最后一个字节，则选择下一个字节并转到步骤 5。
      - 如果这是最后一个字节但不是最后一个 rank，则选择下一个 rank 并转到步骤 3。
      - 如果这是最后一个字节并且是最后一个 rank，则转到步骤 12，在 DQS 门控训练结束之前进行最终清理。
    – 如果不存在连续两次失败和部分通过的情况，则表明需要为此 rank 字节训练更多 RnDGSL/RnDGPS 配置。在这种情况下，执行以下操作：
      - 如果 R0DGPS 小于 3，则递增 R0DGPS 并转到步骤 8
      - 如果 R0DGPS 等于 3 但 R0DGSL 小于 7，则递增 R0DGSL 并转到步骤 7
      - 如果 R0DGPS 等于 3 并且 R0DGSL 等于 7，则记录一个错误，因为这表示系统出现了很大的问题，导致此 rank 字节可能没有通过的配置。出现这样的错误时，用户可以终止整个训练以对系统进行调研，也可以继续训练下一个字节。
12. 完成所有 rank 和所有字节的训练后，立即发出对同一 SDRAM 位置进行一次或多次空指令读取的命令。这样可以清除 PHY 中的 DQS 漂移补偿逻辑，从而可以避免报告由之前的 DQS 门控设置所导致的任何假漂移事件。
13. 空指令读取完成后，立即通过向 DXnDQSTR 寄存器的 DFTCMP 位写入 1 来重新使能漂移补偿。这标志着 DQS 门控训练结束。现在可以开始对内存进行常规操作。

有关如何使用 PUBL DCU 访问 SDRAM 的详细信息，请参见*第 7.5 节：PUBL*。

注： *如果 SDRAM 中除了 DQS 门控训练数据之外没有其他数据，则在软件 DQS 门控训练期间无需向 SDRAM 发出自动刷新命令。这是因为在相对较短的时间（通常应小于刷新时间要求）内，同一位置被反复读取。然而，如果 SDRAM 中有其他数据，并且训练时间超过了所需的刷新时间，则必须在所要求的间隔内相互穿插地发出刷新命令和训练命令。*

低频 LPDDR 的读取 DQS 门控训练

使用 LPDDR SDRAM 时，内置的读取 DQS 门控训练对于标准时钟频率而言是准确的。

但是，随着 LPDDR tAC 时序参数占时钟周期的百分比变小，有效的读取 DQS 门控设置也变少。使用 LPDDR SDRAM 时，读取 DQS 门控在低频下工作，而本节提供的是关于如何在使用 LPDDR SDRAM 的情况下正确设置读取 DQS 门控以低频工作的额外指导。

对于所有 tAC >= (0.25*tCK) 的值，可以使用正常训练。在 tAC 小于 1 tCK 的较慢速度下，数据训练通过的数量可能小于理想值（低至两次），但门控位置仍然符合要求，即使与 DQS 的上升沿对齐也符合。这是因为在开始出现数据失败的情况之前，有足够的裕量（90 度的裕量在低速条件下是非常大的数字）。

下面列出了这些设置中每个设置的通过配置数量和门控位置：

tAC ≥ 1.0*tCK

– 五次通过，门控开口正确定位于 DQS 上升沿前 90 度

tAC ≥ 0.75*tCK < 1.0*tCK

– 四次通过，门控开口正确定位为与 DQS 的上升沿对齐，或定位于 DQS 上升沿前 90 度

tAC ≥ 0.5*tCK < 0.75*tCK

– 三次通过，门控开口正确定位为与 DQS 的上升沿对齐

tAC ≥ 0.25*tCK < 0.5*tCK

– 两次通过，门控开口正确定位为与 DQS 的上升沿对齐

*注意：该指导的例外情况是 tAC < 0.25*tCK 时。在这种情况下，不应执行任何训练，应使用被动加窗。另外，确保 DGSL 和 DGPS 位设置为零（默认情况下 DGPS 为 1。）*

7.5.7 读取有效训练 (RVTRN)

PUBL 内置读取有效训练程序 (RVTRN)，可由软件或存储控制器使用 DDRPHYC_PIR 寄存器触发。

DDR3 不需要 RVTRN

对于 LPDDR2/3，RVTRN 通常与 DQSTRN 同时运行；它在极端条件下使用 tDQSCK 时序值调整有效读取系统延迟。

7.6 PUBL 时序和 DFI 接口

注： *本节旨在介绍 DDRPHYC 接口的信号时序以及 DFI 与 SDRAM 接口信号之间的关系，并提供编程到 DDRCTRL 中的物理延迟的相关信息。*

7.6.1 简介

本节介绍控制器与 PHY 之间接口的时序。

读/写时序波形包括四组信号：

1. DFI 信号 (dfi_*)
2. PHY 时钟（ctl_clk、clk_0、clk_90、clk_180 和 clk_270）
3. 控制器到 PHY 接口信号（ctl_cmd、ctl_ds_oe、ctl_d_oe、ctl_d、ctl_qs_en、ctl_qs_dis 和 ctl_qs_phase）
4. SDRAM 信号（ck/ck_n、cmd、dqs/dqs_n 和 dq）。

仅显示需要控制时序的重要信号，即 dfi_* 和 SDRAM 信号 (DDR_*)。

PHY 时钟组未显示，且 ctl_clk 与 dfi_clk 相同。

在控制器到 PHY 接口信号组中，仅显示带有 PHY 被动 DQS 门控时序的 ctl_qs_en 信号。

提供了典型 PHY 配置和 SDRAM 延迟的读/写波形；显示的波形为 DDR3 模式的波形。

主动和被动 DQS 门控的波形大致相同，对于 ctl_qs_en 信号，仅显示被动 DQS 门控时序。

所示时序波形是针对写延迟 (WL) 5 和读延迟 (RL) 5。任何延迟的时序 RWL 都可以由 RWL 控制器时钟简单地将（ctl_ds_oe、ctl_d_oe、ctl_d、ctl_qs_en、ctl_qs_dis 和 ctl_qs_phase）的时序相对于 ctl_cmd 移位来获得。

对于 DDR3，写延迟 (WL) 等于 CAS 写延迟 (CWL) 加上附加延迟 (AI)，读延迟 (RL) 等于 CAS 延迟 (CL) 加上附加延迟 (AL)。

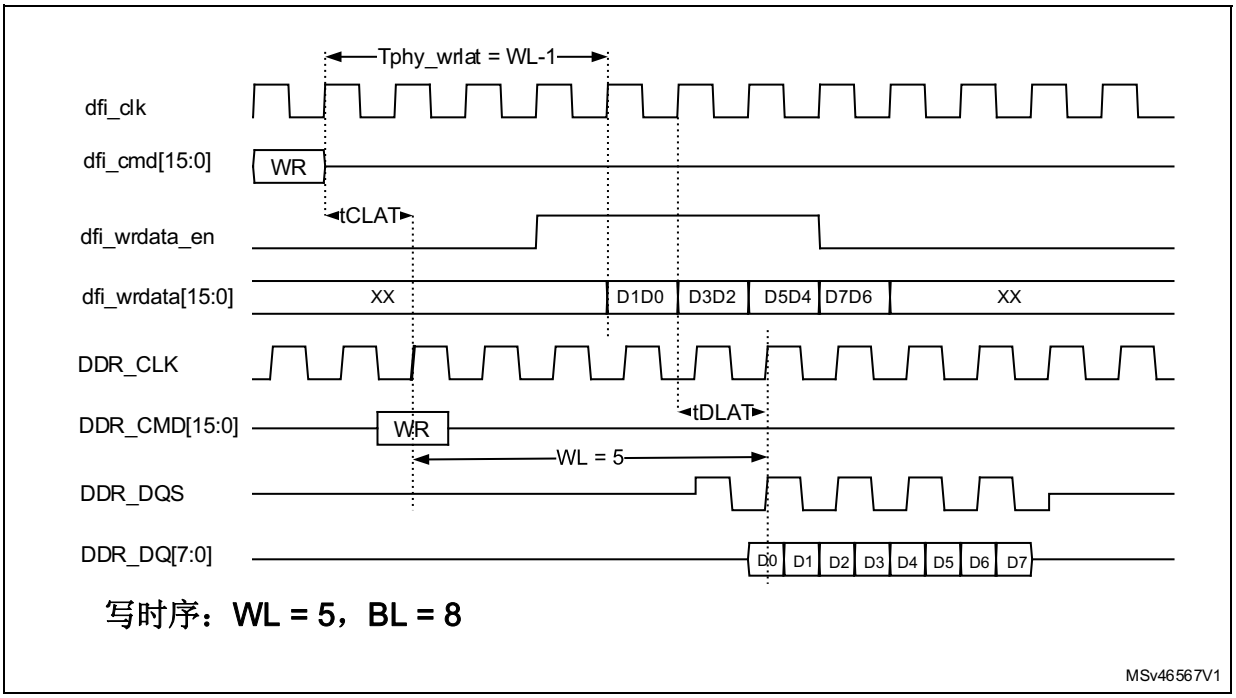
7.6.2 写时序

图 24 显示了 WL = 5 和 BL = 8 时 DDR3 模式的写时序
请注意，dfi_clk 和 ctl-clk 是相同的时钟，只显示 dfi_clk。DFI 时序中唯一的变量是 tphy_wrlat，即 DFI 写命令与 DFI 写数据使能 (dfi_wrdata_en) 信号之间的间隔。

如*表 25* 所示，此间隔等于 WL-1。DFI 控制器总是在驱动写数据使能信号后的一个时钟周期驱动写数据 (dfi_wrdata)。
DFI 写数据掩码 (dfl_wrdata_mask) 的时序与 DFI 写数据 (dfi_wrdata) 的时序相同，因此未在图中显示。DFI 写命令 (dfi_cmd) 只是缓冲到 PHY 控制器命令 (ctl_cmd)，因而与该命令对齐。

图 24 还显示了从控制器到 SDRAM 的命令和数据的两个 PHY 延迟参数。测量的命令延迟 (tCLAT) 是命令对 PHY 可用时的控制器时钟 (ctl_clk) 上升沿与命令输出时的 SDRAM 时钟 (DDR_CLK) 上升沿之间的间隔时间。同样，测量的写数据延迟 (tDLAT) 是可从控制器获得数据时的控制器时钟上升沿与 SDRAM 数据可用时的数据选通 (DDR_DQS) 上升沿之间的间隔时间。

图 24. 写时序，WL = 5，BL = 8



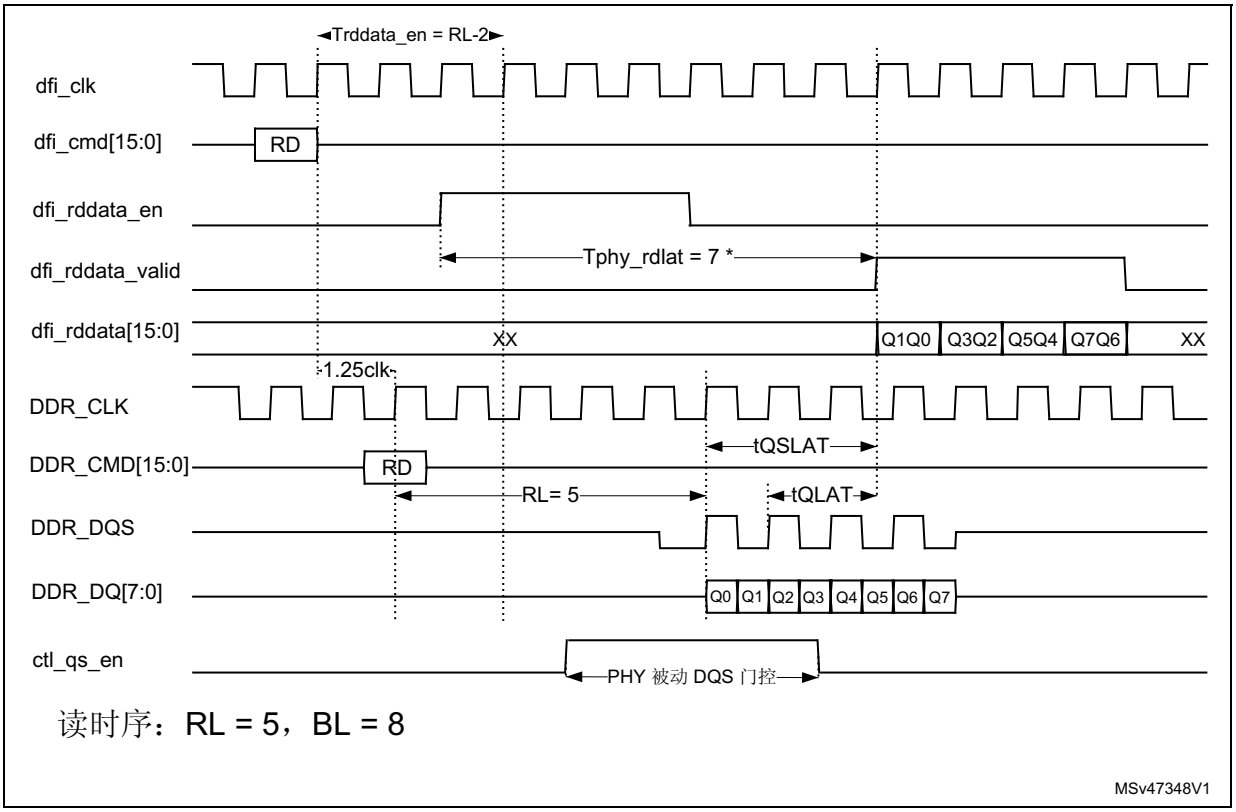
7.6.3 读时序

图 25 显示了 DDR3 模式下，WL = 5 且 BL = 8 时的读时序。

请注意，dfi_clk 和 ctl-clk 是相同的时钟，只显示 dfi_clk。
DFI 时序中的唯一变量是 trddata_en，即 DFI 读命令与 DFI 读数据使能信号 (dfi_rddata_en) 之间的间隔时间。如*图 25* 所示，此间隔等于 RL-2。
DFI 读命令 (dfi_cmd) 只是缓冲到 PHY 控制器命令 (ctl_cmd)，因而与该命令对齐。

图 25 显示的时序包括通过 PHY 读 FIFO 返回控制器的读时序。它还显示从 SDRAM 到控制器的数据的两个 PHY 延迟参数。
测量的数据选通延迟 (tQSLAT) 是与第一个数据节拍同步的数据选通 (DDR_DQS) 的上升沿，与控制器采样数据时的控制器时钟 (ctl_clk 与 dfi_clk 相同) 的上升沿之间的间隔时间。
测量的读数据延迟 (tQLAT) 是两个数据节拍可用后的数据选通 (DDR_DQS) 的上升沿与控制器采样数据时的控制器时钟 (ctl_clk 与 dfi_clk 相同) 的上升沿之间的间隔时间。
这些参数值在*表 24* 和*表 25* 中进行了描述。
请注意，视读数据选通和控制器时钟的对齐情况而定，读延迟最多可能会有一个控制器时钟周期的变化。这是因为读数据选通与控制器时钟之间存在时钟域交叉，而这被视为彼此完全异步。

图 25. 读时序，WL = 5，BL = 8



7.6.4 DDRPHY 读/写延迟

*表 24：PHY 延迟*显示了通过 PHY 时命令和数据的延迟。

有关这些参数的定义和说明，请参见*第 355 页的写时序*和*第 356 页的读时序*。

表 24. PHY 延迟

符号	说明	延迟	单位	注释
t_{CLAT}	命令延迟	1.25	tCLK	-
t_{DLAT}	写数据延迟	1.25	tCLK	-
t_{QLAT}	读数据延迟	1-2	tCLK	1
t_{QSLAT}	读数据选通延迟	2-3	tCLK	1

注： *PHY 延迟取决于读数据选通与控制器时钟的对齐情况。*

PHY 读延迟参数 tphy_rdlat（即，从 dfi_rddata_en 信号置 1 到 dfi_rddata_valid 置 1 的 DFI 时钟周期数）定义为：

tphy_rdlat = 3 + tQSLAT + 1 + x + y

如果使能了无气泡功能，则延迟为：

tphy_rdlat = 3 + tQSLAT + 1 + x + y + 2

如果使能了固定延迟功能，则延迟为：

tphy_rdlat = 12

其中：

x = 从 PHY 宏输出到 SDRAM 的命令的传播延迟

y = 从 SDRAM 到 PHY 宏输入的读数据传播延迟

gdqs_rsl = 所有 rank 和所有字节通道的 DQS 门控系统延迟值集合

上述等式中的 (+3) 项和 (+1) 项是内部 DFI 逻辑延迟。
x 和 y 项包括，PHY 与 I/O 之间的路由、I/O 传播延迟、封装延迟、板级延迟，以及其他未在 SDRAM 读延迟 (RL) 参数中考虑的与 SDRAM 相关的延迟。

从 DFI 读命令到 DFI 读数据有效的总的读延迟是

trddata_en + tphy_rdlat.

其中 trddata_en 在*表 25* 中进行了描述。

7.6.5 DFI 2.1 概述

DFI 是一个接口协议，定义了 DDR 内存设备（本文中为 DDR3 和 LPDDR2）的 DDR 存储控制器 (DDRCTRL) 与 DDR 物理接口 (DDRPHY) 之间的连接。

该协议定义了通过 DFI 向 DRAM 设备传输控制信息和数据所需的信号、信号关系和时序参数。

DDRPHYC 支持协议版本 2.1：

接口信号组

DFI 进一步分为以下接口组：

- 控制器接口
- 写入数据接口
- 读取数据接口
- 更新接口
- 状态接口
- 训练接口
- 低功耗控制接口

DFI 时序参数

DFI 定义了多个具有固定值及指定为 DFI 时钟周期数的时序参数。*表 25* 描述了适用于 DDRPHYC 的 DFI 参数。

DFI 时序参数与需要相应编程的 DDRCTL 相关。

表 25. DFI 时序参数

名称	说明	SDR 模式	注释
t_{ctrl_delay}	指定 DFI 控制信号生效或失效之后的 DIF 时钟周期数，PHY-DRAM 接口处的控制信号可反映出生效或失效。如果 DFI 时钟与内存时钟并非相位对齐，则应将此时序参数向上舍入到下一个整数值。	2	(1)
t_{phy_wrdata}	指定当 dfi_wrdata_en 信号生效时与相关写数据在 dfi_wrdata 信号上被驱动时所间隔的 DFI 时钟周期数。	1	-
t_{phy_wrlat}	指定当在 DFI 控制接口上发送写命令时与 dfi_wrdata_en 信号生效时所间隔的 DFI 时钟周期数 注意：在 LPDDR2 模式下，必须使用以下设置： WL = MR2[3:0].WL + 1（考虑 tDQSS）	WL-1	(2)(3) (4)
t_{rddata_en}	指定从读命令在 DFI 上生效到 dfi_rddata_en 信号生效所间隔的时间。 注意：在 LPDDR2 模式下，必须使用以下设置： RL = MR2[3:0].RL + tDQSCK	RL-2	(2)(3)
t_{phy_rdlat}	指定从 dfi_rddata_en 信号生效到 dfi_rddata_valid 信号信号生效所允许间隔的最大周期数。	15	(5)
$t_{dram_ck_disable}$	指定从 DFI 上 dfi_dram_clk_disable 信号生效到 PHY-DRAM 边界的 DRAM 内存设备时钟保持低值所经过的时钟数。	2	-
$t_{dram_ck_enable}$	指定从 DFI 上的 dfi_dram_clk_disable 信号失效到 PHYDRAM 边界处 DRAM 设备的时钟的第一个有效上升沿所间隔的时钟数。	2	-
t_{phyupd_type0}	在 dfi_phyupd_type = 0 的情况下，指定 dfi_phyupd_ack 信号生效之后，dfi_phyupd_req signal 可保持生效的最大 DFI 时钟周期数。	1	(6)
t_{phyupd_resp}	指定从 dfi_phyupd_req 信号生效到 dfi_phyupd_ack 信号生效所间隔的最大 DFI 时钟周期数。	64	
$t_{phy_paritylat}$	指定从 dfi_parity_in 信号被驱动，到返回关联的 dfi_parity_error，所间隔的最大 DFI 时钟周期数。	PL+6	(7)

1. 实际延迟是 1.25，但根据 DFI 规范四舍五入为下一个整数。
2. 使用 LPDDR2 设备时，WL 是 SDRAM 写延迟（CAS 写延迟 + 附加延迟 + 1），以确保将 LPDDR2 tDQSS 考虑在内。使用所有其他 DDR 设备时，WL 是 SDRAM 写延迟（CAS 写延迟 + 附加延迟）。如果外部控制器希望在回送模式（即 LBMODE 置 1）下运行 PHY，则必须将 tphy_wrlat 设置为 WL-2。
3. 在 LPDDR2 模式下，WL = MR2.WL + 1，RL = MR2.RL + tDQSCK。
4. RL 是 SDRAM 读延迟（CAS 延迟 + 附加延迟）。
5. 该数字包括通过 PHY 的延迟（即命令延迟加上读延迟），加上可由 PUB 训练的最大可能系统延迟。
6. 这与更新 ITM 所需的最长时间相对应。
7. PL 是 DIMM 奇偶校验延迟。它在当前配置下不适用。

7.7 PUBL 寄存器

7.7.1 DDRPHYC 版本 ID 寄存器 (DDRPHYC_RIDR)

DDRPHYC revision ID register

偏移地址：0x000

复位值：0x0020 1210

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UDRID[7:0]								PHYMJR[3:0]				PHYMDR[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHYMNR[3:0]				PUBMJR[3:0]				PUBMDR[3:0]				PUBMNR[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 **UDRID[7:0]**：用户自定义版本 ID (User-defined rev ID)

位 23:20 **PHYMJR[3:0]**：PHY 主版本 (PHY maj rev)

位 19:16 **PHYMDR[3:0]**：PHY 中版本 (PHY moderate rev)

位 15:12 **PHYMNR[3:0]**：PHY 次版本 (PHY minor rev)

位 11:8 **PUBMJR[3:0]**：PUB 主版本 (PUB maj rev)

位 7:4 **PUBMDR[3:0]**：PUB 中版本 (PUB moderate rev)

位 3:0 **PUBMNR[3:0]**：PUB 次版本修订 (PUB minor rev)

7.7.2 DDRPHYC PHY 初始化寄存器 (DDRPHYC_PIR)

DDRPHYC PHY initialization register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INITBYP	ZCALBYP	LOCKBYP	CLRSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTLDINIT	DLLBYP	ICPC
w	w	w	w										w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RVTRN	QSTRN	DRAMINIT	DRAMRST	ITMSRST	ZCAL	DLLLOCK	DLLSRST	INIT
							w	w	w	w	w	w	w	w	w

位 31 **INITBYP**：初始化旁路 (Initialization bypass)
如果置 1，将旁路或停止所有当前正在运行的初始化程序，包括 PHY 初始化、DRAM 初始化和 PHY 训练。
可以使用 INIT 和 DDRPHYC_PIR 寄存器的其他相关位手动触发初始化。该位为自清零位。

位 30 **ZCALBYP**：zcal 旁路 (zcal bypass)
如果置 1，将旁路或停止所有 ZQ 控制模块的阻抗校准，该校准在复位后会自动触发。
可以使用 DDRPHYC_PIR 寄存器的 INIT 和 ZCAL 位手动触发阻抗校准。该位为自清零位。

位 29 **LOCKBYP**：DLL 锁相旁路 (DLL lock bypass)
如果置 1，将旁路或停止 DLL 锁相等待。DLL 锁相等待会在复位后自动触发。
可以使用 DDRPHYC_PIR 寄存器的 INIT 和 DLLLOCK 位手动触发 DLL 锁相等待。该位为自清零位。

位 28 **CLRSR**：清空状态寄存器 (clear status register)
向该位写入 1 会执行下操作：
– 自动自清零。这表示之后读取 CLRSR 位将返回 0
– 将 DXnGSR0 寄存器中的 DTDONE、DTERR、DTIERR 位清零
– 将 DDRPHYC_PGSR 寄存器中的 DFTERR 位和 DDRPHYC_DXnGSR1 寄存器中的 DFTERR 位清零
该位主要用于调试，在正常运行过程中通常不需要。
当 PGSR 寄存器中的 IDONE=1 时，可以使用该位手动清零 PGSR 状态位，但 PGSR 状态位（DFTRR 和 TQ 位除外）会通过启动新的初始化过程自动清零。
该位也可用来手动清零 DXnGSR 状态位，但启动新的数据训练过程会自动清零 DXnGSR 状态位。

位 27:19 保留，必须保持复位值。

位 18 **CTLDINIT**：控制器 DRAM 初始化 (Controller DRAM initialization)
如果置 1，则表示 DRAM 初始化由控制器执行。如果未置 1，表示 DRAM 初始化是使用内置的初始化序列或通过配置端口使用软件执行的。

位 17 **DLLBYP**：DLL 旁路 (DLL bypass)
如果该位置 1，会将所有 PHY DLL 置于旁路模式。已旁路的 DLL 还会掉电（禁止）。

位 16 **ICPC**：初始化完成引脚配置 (Initialization complete pin configuration)
指定应如何使用 DFI 2.1 初始化完成输出引脚来指示初始化状态。
0: 在 PHY 初始化（DLL 锁相和阻抗校准）完成后生效。
1: 在 PHY 初始化完成以及触发的 PUBL 初始化（DRAM 初始化、数据训练或未选择初始化的初始化触发）完成后生效。

位 15:9 保留，必须保持复位值。

位 8 **RVTRN**：
使用 LPDDR2/3 时 RVTRN 和 DQSTRN 应运行；DDR3 不需要 RVTRN
如果 RVTRN=1 且位 QSTRN=0，则当位 INIT=1（触发初始化过程）时，读取 DQS 门控训练算法仍然运行（就像 DDRPHYC_PIR 寄存器中的位 QSTRN 实际上置 1 一样）。
如果需要单独运行 RV 训练，而不运行读取 DQS 门控训练，可以通过将 DDRPHYC_PGCR 寄存器中的位 LBGDQS 置 1 来阻止读取 DQS 门控训练运行。
注： RV 训练不能使用多用途寄存器 (MPR)，必须使用在 DTDR0-1 中编程的用户数据。

位 7 **QSTRN**：读取 DQS 训练 (Read DQS training)
执行 PUBL 训练程序以确定读取数据 DQS 选通信号的最佳位置，以获得最大的系统时序裕度。
注： 读取 DQS 训练不能使用多用途寄存器 (MPR)，必须使用在 DTDR0-1 中编程的用户数据。

位 6 **DRAMINIT**：DRAM 初始化 (DRAM initialization)
执行 DRAM 初始化序列

位 5 **DRAMRST**：DRAM 复位 (DRAM reset)（仅限 DDR3）

向 DRAM 发出复位信号（通过将 DRAM 复位引脚驱动为低电平）并等待 200 us。
可以独立触发，也可以在 DRAM 完全初始化 (DRAMINIT) 后触发。对于后一种情况，应先发出复位信号并等待 200 us，然后再开始完全初始化序列。

位 4 **ITMSRST**：ITM 复位 (ITM reset)

软复位数据和数据选通的接口时序模块，即，它将 ITM 软复位 (srstb) 信号置为有效。

位 3 **ZCAL**：阻抗校准 (Impedance calibration)（驱动器和 ODT）

执行 PHY 阻抗校准。

位 2 **DLLLOCK**：DLL 锁相 (DLL lock)

等待 PHY DLL 锁相。

位 1 **DLLSRST**：DLL 软复位 (DLL soft reset)

通过驱动 DLL 软复位引脚来软复位所有 PHY DLL。

注： *- 要求 ctl_clk 翻转，以便 DLL 软复位信号从 DLL 输出。如果无法保证翻转 ctl_clk（AC DLL 的 cclk_0），建议使用 DDRPHYC_ACDLLCR 的手动 DLL 软复位位 DLLSRST，而不是 DDRPHYC_PIR 的 DLLSRST 位。*

- 在置位此位时，要确保遵守 DLL 旁路和 DLL 复位的最低要求。

位 0 **INIT**：初始化触发 (Initialization trigger)

向该位写入“1”会触发 DDR 系统初始化，包括 PHY 初始化、DRAM 初始化和 PHY 训练。
要执行的确切初始化步骤在该寄存器的位 **1 到 6** 中指定。位
置 1 表示该步骤作为初始化序列的一部分执行，而设置为“0”表示该步骤被旁路。
初始化触发位是自清零位。

7.7.3 DDRPHYC PHY 全局控制寄存器 (DDRPHYC_PGCR)

DDRPHYC PHY global control register

偏移地址：0x008

复位值：0x0184 2E04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LBMODE	LBGDQS	LBDQSS	RFSHDT[3:0]				PDDISDX	ZKSEL[1:0]		RANKEN[3:0]				IODDRM[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IOLB	CKINV	CKDV[1:0]		CKEN[2:0]			DTOSEL[3:0]				DFTLMT[1:0]		DFTCMP	DQSCFG	ITMDMD
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **LBMODE**：回送模式 (Loop back mode)
如果置 1，则表示 PHY/PUB 处于回送模式

位 30 **LBGDQS**：回送 DQS 门控 (Loop back DQS gating)
选择 PHY 处于回送模式时应使用的 DQS 门控模式。有效值为：
0: 在 PUB 上触发了 DQS 门控训练
1: DQS 门控使用软件手动设置

位 29 **LBDQSS**：回送 DQS 移位 (Loop back DQS shift)
选择在回送期间读取 DQS 如何移位，以确保读取 DQS 在读取数据眼图中居中。有效值为：
0: PUB 将读取 DQS 延迟设置为 0；DQS 已由写路径移位 90 度
1: 读取 DQS 移位通过软件手动设置。

位 28:25 **RFSHDT[3:0]**：训练期间刷新 (Refresh during training)
非零值，指定在训练每个 rank（最后一个 rank 除外）后，应向 SDRAM 连续发送一定数量的刷新，具体数量为该位字段所指定的数值。

位 24 **PDDISDX**：掉电禁止字节 (Power down disabled byte)
如果置 1，表示应对已禁止字节的 DLL 和 I/O 进行掉电处理。

位 23:22 **ZKSEL[1:0]**：阻抗时钟分频比选择 (Impedance clock divider selection)
选择阻抗控制逻辑使用的时钟的分频比。分频比的源时钟
是配置端口时钟信号 (pclk)，具体取决于使用的配置端口类型（请参见 tbc 页的“阻抗校准”）。
有效值为：
00: 2 分频
01: 8 分频
10: 32 分频
11: 64 分频

位 21:18 **RANKEN[3:0]**：Rank 使能 (Rank enable)
指定为数据训练使能的 rank。
仅使用位 0 并控制 rank 0。
将该位置 1 将使能 rank，设置为“0”将禁止 rank。

位 17:16 **IODDRM[1:0]**：I/O DDR 模式 (I/O DDR mode)
为 I/O 选择 DDR 模式。

位 15 **IOLB**：I/O 回送选择 (I/O loop back select)
选择 I/O 内发生信号回送的位置。不适用于 D3A I/O。
有效值为：
0: 回送在输出缓冲器之后；必须将输出使能置为有效
1: 回送在输出缓冲器之前；与输出使能无关

位 14 **CKINV**：CK 反转 (CK invert)
如果置 1，则指定应反转 CK/CK#。否则，CK/CK# 在正常极性下翻转。

位 13:12 **CKDV[1:0]**：CK 禁止值 (CK disable value)
指定在禁止 CK/CK# 对时应驱动的静态值。CKDV[0] 指定 CK 的值，CKDV[1] 指定 CK# 的值。

位 11:9 **CKEN[2:0]**：CK 使能 (CK enable)
控制去往 SDRAM 的 CK 是使能（翻转）还是
禁止（由 CKDV 定义的静态值）。三对 CK 每对各一位。

位 8:5 **DTOSEL[3:0]**：数字测试输出选择 (Digital test output select)
数字测试输出选择：选择应被驱动到
PHY 数字测试输出 (DDR_DTO0) 引脚上的 PHY 数字测试输出：有效值为：
0000: DATX8 0 DLL 数字测试输出
0001: DATX8 1 DLL 数字测试输出
0010: DATX8 2 DLL 数字测试输出
0011: DATX8 3 DLL 数字测试输出
0100: DATX8 4 DLL 数字测试输出
0101: DATX8 5 DLL 数字测试输出
0110: DATX8 6 DLL 数字测试输出
0111: DATX8 7 DLL 数字测试输出
1000: DATX8 8 DLL 数字测试输出
1001: AC DLL 数字测试输出
其他：Reserved

位 4:3 **DFTLMT[1:0]**：DQS 漂移限制 (DQS drift limit)
指定读取数据选通的预期漂移限制。此值或更大漂移值将通过主机端口错误标志报告为漂移错误。有效值为：
00: 无限制（不报告错误）
01: 90° 漂移
10: 180° 漂移
11: 270° 或更大漂移
注： 虽然通过错误标志进行了报告，但该错误不需要采取任何措施。仅表示漂移超出了预期。

位 2 **DFTCMP**：DQS 漂移补偿 (DQS drift compensation)
使能或禁止 DQS 漂移补偿。有效值为：
0: 禁止数据选通漂移补偿
1: 使能数据选通漂移补偿
默认情况下，使能漂移补偿。
注： 在以下任何情况下都不支持漂移补偿：
- LPDDR2/3（DDRPHYC_DCR 寄存器中的 DDRMD 位设置为 LPDDR2）
- 突发传输长度 2（DDRPHYC_MR0 寄存器中的 BL 位设置为突发传输长度 2）
- 使用被动加窗的读 DQS 门控（DDRPHYC_PGCR 寄存器中的 DQSCFG 位设置为被动加窗）
如果进行了以上任一设置，则必须禁止漂移补偿。

位 1 **DQSCFG**：DQS 门控配置 (DQS gating configuration)
选择以下两种 DQS 门控方案之一：
0: 使用 DQS_b 的上升沿关闭 DQS 门控（主动加窗模式）
1: DQS 门控覆盖整个突发传输（被动加窗模式）。
注意：LPDDR2/3 必须使用被动加窗模式。

位 0 **ITMDMD**：ITM DDR 模式 (ITM DDR mode)
选择 ITMS 是使用 DQS 和 DQS# 还是仅使用 DQS。
有效值为：
0: ITMS 使用 DQS 和 DQS#
1: ITMS 仅使用 DQS
注： 正确的设置始终是 DQS 和 DQS#。

7.7.4 DDRPHYC PHY 全局状态寄存器 (DDRPHYC_PGSR)

DDRPHYC PHY global status register

偏移地址：0x00C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TQ	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	RVEIRR	RVERR	DFTERR	DTIERR	DTERR	DTDONE	DIDONE	ZCDDONE	DLDONE	IDONE
						r	r	r	r	r	r	r	r	r	r

位 31 **TQ**：温度输出 (Temperature output)（仅限 LPDDR）不适用

位 30:10 保留，必须保持复位值。

位 9 **RVEIRR**：读取有效训练间歇性出错 (Read valid training intermittent error)
如果置 1，表示在读取有效训练期间间歇性出错，例如一次通过后下一次失败，然后又通过。

位 8 **RVERR**：读取有效训练错误 (Read valid training error)
如果置 1，表示在读取有效训练期间无法找到有效的读取有效位置。

位 7 **DFTERR**：DQS 漂移错误 (DQS drift error)
如果置 1，表示至少有一个读取数据选通的漂移已达到或超出 PHY 通用配置寄存器 (PGCR) 中设置的漂移限制。

位 6 **DTIERR**：DQS 门控训练间歇性出错 (DQS gate training intermittent error)
如果置 1，表示在 DQS 门控训练期间间歇性出错，例如一次通过后下一次失败，然后又通过。

位 5 **DTERR**：DQS 门控训练错误 (DQS gate training error)
如果置 1，表示在 DQS 门控训练期间无法找到有效的 DQS 门控窗口。

位 4 **DTDONE**：数据训练完成 (Data training done)
如果置 1，表示 PHY 已完成数据训练。

位 3 **DIDONE**：DRAM 初始化完成 (DRAM initialization done)
如果置 1，表示 DRAM 初始化已完成。

位 2 **ZCDDONE**：阻抗校准完成 (zcal done)
如果置 1，表示阻抗校准已完成。

位 1 **DLDONE**：DLL 锁相完成 (DLL lock done)
如果置 1，表示 DLL 锁相已完成。

位 0 **IDONE**：初始化完成 (Initialization done)
如果置 1，表示 DDR 系统初始化已完成。
该位在 DDRPHYC_PIR 寄存器中的所有选定初始化程序完成后置 1。

7.7.5 DDRPHYC DDR DLL 全局控制寄存器 (DDRPHYC_DLLGCR)

DDRPHYC DDR DLL global control register

偏移地址：0x010

复位值：0x0373 7000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DLLRSVD2[1:0]		LOCKDET	FDTRMSL[1:0]		SBIAS5_3[2:0]			BPS200	SBIAS2_0[2:0]			MBIAS[7:4]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MBIAS[3:0]				TESTSW	ATC[1:0]		DTC[2:0]			TESTEN	IPUMP[2:0]			DRES[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 **DLLRSVD2[1:0]**：
这些位连接到 DLL 控制总线，保留供将来使用。

位 29 **LOCKDET**：主时钟检测使能 (Master clock detect enable)
注意：该位字段仅对少数几个过程有效。对于所有其他过程，该位字段保留。

位 28:27 **FDTRMSL[1:0]**：从设备旁路固定延迟微调
00: 标称延迟
01: 标称延迟 - 10%
10: 标称延迟 + 10%
11: 标称延迟 + 20%

位 26:24 **SBIAS5_3[2:0]**：从偏置微调 (Slave bias trim)
用于微调 Slave DLL 的偏置。

位 23 **BPS200**：旁路模式频率范围 (Bypass mode frequency range)
0: 0 到 100 MHz
1: 0 到 200 MHz

位 22:20 **SBIAS2_0[2:0]**：从偏置微调 (Slave bias trim)
用于微调 Slave DLL 的偏置。

位 19:12 **MBIAS[7:0]**：主偏置微调 (Master bias trim)
用于微调 Master DLL 的偏置。

位 11 **TESTSW**：测试开关 (Test switch)
选择 Master DLL（设为 0）或 Slave DLL（设为 1）的测试信号。

位 10:9 **ATC[1:0]**：模拟测试控制 (Analog test control)

当 TESTEN 为高电平时，选择要在 DLL 模拟测试输出 (DDR_ATO) 上输出的模拟信号（TESTEN 为低电平时输出为 Vss）。

测试输出来自 Master DLL 或 Slave DLL，具体取决于测试开关 (TESTSW) 的设置。

Master DLL 和 Slave DLL 均输出相似的模拟测试信号。模拟测试控制的有效设置包括：

00: PMOS (Vbp) 的复制偏置输出

01: NMOS (Vbn) 的复制偏置输出

10: 滤波器输出 (Vc)

11: V_{DDCORE}

位 8:6 **DTC[2:0]**：数字测试控制 (Digital test control)

当 TESTEN 为高电平时，选择要在 DLL 数字测试输出（DTO1 引脚）上输出的数字信号（当 TESTEN 低电平时输出为“0”）。DTO0 引脚输出控制 DLL 参考时钟 (clk_0)。

Master DLL 的有效设置（例如，当 TESTSW =“0”时）包括：

- 000: 0 输出时钟 (clk_0)
- 001: 90 输出时钟 (clk_90)
- 010: 180 输出时钟 (clk_180)
- 011: 270 输出时钟 (clk_270)
- 100: 360 内部时钟 (clk_360_int)
- 101: 加速脉冲 (spdup)
- 110: 减速脉冲 (slwdn)
- 111: 0 MCTL 逻辑时钟 (cclk_0)

Slave DLL 的有效设置（例如，当 TESTSW =“1”时）包括：

- 000: 输入 DQS 选通 (dqs)
- 001: 输入时钟参考 (clk_90_in)
- 010: 内部反馈时钟 (clk_0_out)
- 011: 90 输出 DQS# 选通 (dqsb_90)
- 100: 90 输出 DQS 选通 (dqs_90)
- 101: 加速脉冲 (spdup)
- 110: 减速脉冲 (slwdn)
- 111: 自动锁相使能信号

位 5 **TESTEN**：测试使能 (Test enable)

分别使能 DTC 和 ATC 选择的数字和模拟测试输出。

位 4:2 **IPUMP[2:0]**：电荷泵电流微调

000：最大电流

111：最小电流

位 1:0 **DRES[1:0]**：微调参考电流相对于电阻值的变化

00: Rnom

01: Rnom - 20%

1x：Rnom + 20%

注： 对于少数几个过程，该位字段保留，不供 DLL 使用。

7.7.6 DDRPHYC AC DLL 控制寄存器 (DDRPHYC_ACDLLCR)

DDRPHYC AC DLL control register

偏移地址：0x014

复位值：0x4000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DLLDIS	DLLSRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ATESTEN	Res.	Res.
rw	rw												rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	MFWDLY[2:0]			MFBDLY[2:0]			Res.	Res.	Res.	Res.	Res.	Res.
				rw	rw	rw	rw	rw	rw						

位 31 **DLLDIS**：DLL 禁止 (DLL disable)
已禁止的 DLL 会被旁路。默认情况下使能 DLL。

位 30 **DLLSRST**：DLL 软复位 (DLL soft reset)
通过驱动 DLL 软复位引脚来软复位 AC DLL。

位 29:19 保留，必须保持复位值。

位 18 **ATESTEN**：模拟测试使能 (Analog test enable)
使能模拟测试信号在 DLL 模拟测试输出（ATO 引脚）上输出。当该位为“0”时，DLL 模拟测试输出为三态。

位 17:12 保留，必须保持复位值。

位 11:9 **MFWDLY[2:0]**：Master DLL 前馈延迟微调 (Master DLL feed-forward delay trim)
用于微调 Master DLL 前馈路径中的延迟：
000：最小延迟
111：最大延迟

位 8:6 **MFBDLY[2:0]**：Master DLL 反馈延迟微调 (Master DLL feed-back delay trim)
用于微调 Master DLL 反馈路径中的延迟：
000：最小延迟
111：最大延迟

位 5:0 保留，必须保持复位值。

7.7.7 DDRPHYC PT 寄存器 0 (DDRPHYC_PTR0)

DDRPHYC PT register 0

偏移地址：0x018

复位值：0x0022 AF9B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TITMSRST[3:0]				TDLLLOCK [11:10]	
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDLLLOCK[9:0]										TDLLSRST[5:0]					
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:18 **TITMSRST[3:0]**：ITM 软复位 (ITM soft reset)

在对 ITM 应用软复位时，ITM 软复位引脚必须保持有效的配置时钟周期数。
该值必须等于或大于 8 个控制器时钟周期。
默认值为 8 个控制器时钟周期。

位 17:6 **TDLLLOCK[11:0]**：DLL 锁相时间 (DLL lock time)

DLL 稳定并锁相所需的配置时钟周期数，即从 DLL 复位引脚失效到 DLL 锁相并准备好使用的时钟周期数。
默认值为 533 Mhz 下 5.12 us。

位 5:0 **TDLLSRST[5:0]**：DLL 软复位 (DLL soft reset)

通过 PHY 初始化寄存器 (DDRPHYC_PIR) 触发软复位时，DLL 软复位引脚必须保持有效的配置时钟周期数。
该值必须大于或等于 50 ns 或 8 个控制器时钟周期（以较大者为准）。
默认值为 533 MHz 下 50 ns。

7.7.8 DDRPHYC PT 寄存器 1 (DDRPHYC_PTR1)

DDRPHYC PT register 1

偏移地址：0x01C

复位值：0x0604 111D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	TDINIT1[7:0]								TDINIT0[18:16]		
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDINIT0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:27 保留，必须保持复位值。

位 26:19 **TDINIT1[7:0]**：tDINIT1

DRAM 初始化时间对应于以下值：
DDR3 = 第一条命令前 CKE 高电平时间（tRFC + 10 ns 或 5 tCK，取二者中的较大者）
LPDDR2 = 电源和时钟稳定情况下 CKE 低电平时间 (100 ns)
LPDDR3 = 电源和时钟稳定情况下 CKE 低电平时间 (100 ns)
默认值为 DDR3 在 533 MHz 下 360 ns。

位 18:0 **TDINIT0[18:0]**：tDINIT0

DRAM 初始化时间对应于以下值：
DDR3 = 电源和时钟稳定情况下 CKE 低电平时间 (500 us)
LPDDR2 = 第一条命令前 CKE 高电平时间 (200 us)
LPDDR3 = 第一条命令前 CKE 高电平时间 (200 us)
默认值为 DDR3 在 533 MHz 下 500 us。

7.7.9 DDRPHYC PT 寄存器 2 (DDRPHYC_PTR2)

DDRPHYC PT register 2

偏移地址：0x020

复位值：0x042D A072

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	TDINIT3[9:0]										TDINIT 2[16]
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDINIT2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:27 保留，必须保持复位值。

位 26:17 **TDINIT3[9:0]**：tDINIT3

DRAM 初始化时间对应于以下值：
LPDDR2 = ZQ 初始化命令到第一条命令的时间 (1 us)
LPDDR3 = ZQ 初始化命令到第一条命令的时间 (1 us)
默认值为 LPDDR2/3 在 533 MHz 下 1 us。

位 16:0 **TDINIT2[16:0]**：tDINIT2

DRAM 初始化时间对应于以下值：
DDR3 = 复位低电平时间（上电时 200 us 或上电后 100 ns）
LPDDR2 = 从复位命令到自动初始化结束的时间（1 us + 10 us = 11 us）
LPDDR3 = 从复位命令到自动初始化结束的时间 (11 us)
默认值为 DDR3 在 533 MHz 下200 us

7.7.10 DDRPHYC ACIOC 寄存器 (DDRPHYC_ACIOCR)

DDRPHYC ACIOC register

偏移地址：0x024

复位值：0x3040 0812

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ACSR[1:0]		RSTIOM	RSTPDR	RSTPDD	RSTODT	Res.	Res.	Res.	RANKPDR	Res.	Res.	Res.	CSPDD	Res.	Res.
rw	rw	rw	rw	rw	rw				rw				rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RANKODT	CKPDR[2:0]			CKPDD[2:0]			CKODT[2:0]			ACPDR	ACPDD	ACODT	ACOE	ACIOM
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 **ACSR[1:0]**：AC 压摆率 (AC slew rate)
选择所有地址和命令引脚的 I/O 的压摆率。

位 29 **RSTIOM**：复位 I/O 模式 (Reset I/O mode)
选择 SDRAM 复位的 I/O 的 SSTL 模式（设置为 0 时）或 CMOS 模式（设置为 1 时）。

位 28 **RSTPDR**：RST 引脚接收器掉电 (RST pin power down receiver)
置 1 时关闭 SDRAM RST# 引脚 I/O 上的输入接收器的电源。

位 27 **RSTPDD**：RST 引脚驱动器掉电 (RST pin power down driver)
置 1 时关闭 SDRAM RST# 引脚 I/O 上的输出驱动器的电源。

位 26 **RSTODT**：RST 引脚 ODT (RST pin ODT)
置 1 时使能 SDRAM RST# 引脚 I/O 上的片内终结。

位 25:23 保留，必须保持复位值。

位 22 **RANKPDR**：Rank 接收器掉电 (Rank power down receiver)
置 1 时关闭 I/O CKE、ODT 和 CS# 引脚上的输入接收器的电源。
单 rank 仅使用 RANKPDR[0]

位 21:19 保留，必须保持复位值。

位 18 **CSPDD**：CS 驱动器掉电 (CS power down driver)
置 1 时关闭 CS# 引脚 I/O 上的输出驱动器的电源。
单 rank 仅使用 PDD[0]。
CKE 和 ODT 驱动器掉电由 DSGCR 寄存器控制。

位 17:15 保留，必须保持复位值。

位 14 **RANKODT**：Rank ODT
置 1 时使能 CKE、ODT 和 CS# 引脚 I/O 上的片内终结。

位 13:11 **CKPDR[2:0]**：CK 引脚接收器掉电 (CK pin power down receiver)
置 1 时分别关闭 CK[0]、CK[1] 和 CK[2] 引脚 I/O 上的输入接收器电源。

位 10:8 **CKPDD[2:0]**：CK 引脚驱动器掉电 (CK pin power down driver)
置 1 时分别关闭 CK[0]、CK[1] 和 CK[2] I/O 引脚上的输出驱动器的电源。

位 7:5 **CKODT[2:0]**：CK 引脚 ODT (CK pin ODT)
置 1 时分别使能 CK[0]、CK[1] 和 CK[2] 引脚 I/O 上的片内终结。

位 4 **ACPDR**：AC 引脚接收器掉电 (AC pins power down receivers)
置 1 时关闭 RAS#、CAS#、WE#、BA[2:0] 和 A[15:0] 引脚上 I/O 的输入接收器电源。

位 3 **ACPDD**：AC 引脚驱动器掉电 (AC pins power down drivers)
置 1 时关闭 RAS#、CAS#、WE#、BA[2:0] 和 A[15:0] 引脚上 I/O 的输出驱动器电源

位 2 **ACODT**：AC 引脚 ODT (AC pins ODT)
置 1 时使能 RAS#、CAS#、WE#、BA[2:0] 和 A[15:0] 引脚 I/O 上的片内终结

位 1 **ACOE**：AC 引脚输出使能 (AC pins output enable)
置 1 时使能所有地址和命令的引脚 I/O 上的输出驱动器

位 0 **ACIOM**：AC 引脚 I/O 模式 (AC pins I/O mode)
选择所有地址和命令引脚 I/O 的 SSTL 模式（设置为 0 时）或 CMOS 模式（设置为 1 时）

7.7.11 DDRPHYC DXCC 寄存器 (DDRPHYC_DXCCR)

DDRPHYC DXCC register

偏移地址：0x028

复位值：0x0000 0800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDT
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RVSEL	DQSNRST	Res.	Res.	DQSNRES[3:0]				DQSRES[3:0]				DXPDR	DXPDD	DXIOM	DXODT
rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **AWDT**：主动窗口数据训练 (Active window data train)
如果置 1，则表示应通过主动 DQS 门控窗口进行数据训练（DQS 门控训练和读取有效训练）。仅用于调试。
默认情况下，通过被动加窗模式执行训练。

位 15 **RVSEL**：ITMD 读取有效选择 (ITMD read valid select)
选择用于 ITMD 读取有效的方案。有效值为：
0: ITMD 读取有效信号通过延迟的 DFI 读取使能信号生成。
1: ITMD 读取有效信号通过 ITMD 自身使用异步交叉生成。

位 14 **DQSNRST**：DQS 复位 (DQS reset)
如果置 1，则表示 DQS# 的 ITMS 应始终被复位，以使其输出使能始终为“1”，其数据输出始终为“0”。
这通过将此 ITMS 的 oe_set_b 和 do_rst_b 引脚驱动至“0”完成，目的是为了
在不使用 DQS# 的应用中，将未使用的 DQS# PAD 强制设为已知状态“0”。

位 13:12 保留，必须保持复位值。

位 11:8 **DQSNRES[3:0]**：DQS# 电阻 (DQS# resistor)

选择 DQS# 引脚的片上上拉/下拉电阻。与 DQSRES 编码相同。

注： 对于 LPDDR2，必须连接 DQS# 电阻。

位 7:4 **DQSRES[3:0]**：DQS 电阻 (DQS resistor)

选择 DQS 引脚的片上下拉/上拉电阻。

DQSRES[3] 用于选择下拉（设为 0 时）或上拉（设为 1 时）。

DQSRES[2:0] 按如下方式选择电阻值：

000: 断开：片上电阻断开连接

001: 688 Ω

010: 611 Ω

011: 550 Ω

100: 500 Ω

101: 458 Ω

110: 393 Ω

111: 344 Ω

注： 对于 LPDDR2，必须连接 DQS 电阻

位 3 **DXPDR**：数据接收器掉电 (Data power down receiver)

置 1 时关闭所有 DATX8 宏的 DQ、DM 和 DQS/DQS# 引脚 I/O 上的输入接收器的电源。它还会关闭所有 DAXT8 宏的 PDQSR 单元的电源。该位与各个 DATX8 的 PDR 和 DQSRPD 配置位进行 OR 运算

位 2 **DXPDD**：数据驱动器掉电 (Data power down driver)

置 1 时关闭所有 DATX8 宏的 DQ、DM 和 DQS/DQS# 引脚上 I/O 的输出驱动器的电源。该位与各个 DATX8 的 PDD 配置位进行 OR 运算。

位 1 **DXIOM**：数据 I/O 模式 (Data I/O mode)

选择所有 DATX8 宏的 DQ、DM 和 DQS/DQS# 引脚 I/O 的 SSTL 模式（设为 0 时）或 CMOS 模式（设为 1 时）。该位与各个 DATX8 的 IOM 配置位进行 OR 运算。

位 0 **DXODT**：数据片内终结 (Data on die termination)

置位时，此位用于使能所有 DATX8 宏中 DQ、DM 及 DQS/DQS# 引脚 I/O 上的片内终结 (on-die termination) 功能。该位会与各个 DATX8 的 ODT 配置位进行逻辑或运算。

7.7.12 DDRPHYC DSGC 寄存器 (DDRPHYC_DSGCR)

DDRPHYC DSGC register

偏移地址：0x02C

复位值：0xFA00 001F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CKEOE	RSTOE	ODTOE	CKOE	TPDOE	TPDPD	NL2OE	NL2PD	Res.	Res.	Res.	ODTPDD	Res.	Res.	Res.	CKEPDD
rw	rw	rw	rw	rw	rw	rw	rw				rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	FXDLAT	NOBUB	DQSGE[2:0]			DQSGX[2:0]			LPDLLPD	LPIOPD	ZUEN	BDISEN	PUREN
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **CKEOE**：CKE 输出使能 (CKE output enable)
设置 SDRAM CKE 引脚 I/O 上的输出驱动器。

位 30 **RSTOE**：RST 输出使能 (RST output enable)
置 1 时使能 SDRAM RST# 引脚 I/O 上的输出驱动器

位 29 **ODTOE**：ODT 输出使能 (ODT output enable)
置 1 时使能 SDRAM ODT 引脚 I/O 上的输出驱动器。

位 28 **CKOE**：CK 输出使能 (CK output enable)
置 1 时使能 SDRAM CK/CK# 引脚 I/O 上的输出驱动器。

位 27 **TPDOE**：TPD 输出使能 (TPD output enable)（不适用，仅限 LPDDR）

位 26 **TPDPD**：TPD 驱动器掉电 (TPD power down driver)（不适用，仅限 LPDDR）

位 25 **NL2OE**：非 LPDDR2 输出使能 (Non LPDDR2 output enable)
置 1 时使能非 LPDDR2/LPDDR3（ODT、RAS#、CAS#、WE# 和 BA）引脚 I/O 上的输出驱动器。
当设计用于 LPDDR2/LPDDR3 和其他 DDR 模式的芯片在 LPDDR2/LPDDR3 模式下使用时，可以使用此功能。
对于这些引脚，I/O 输出使能信号 (OE) 是该位与 ACIOCR 或 DSGCR 寄存器中的相应输出使能位的与 (AND) 操作。

位 24 **NL2PD**：非 LPDDR2 掉电 (Non LPDDR2 power down)
置 1 时关闭非 LPDDR2/LPDDR3（ODT、RAS#、CAS#、WE# 和 BA）引脚 I/O 上的输出驱动器和输入接收器。
当设计用于 LPDDR2/LPDDR3 和其他 DDR 模式的芯片在 LPDDR2/LPDDR3 模式下使用时，可以使用此功能。
对于这些引脚，I/O 掉电信号（PDD 或 PDR）是该位与 ACIOCR 寄存器中相应掉电位的 OR 操作。

位 23:21 保留，必须保持复位值。

位 20 **ODTPDD**：ODT 驱动器掉电 (ODT power down driver)
置 1 时关闭 ODT 引脚 I/O 上的输出驱动器的电源。

位 19:17 保留，必须保持复位值。

位 16 **CKEPDD**：CKE 驱动器掉电 (CKE power down driver)
置 1 时关闭 CKE 引脚 I/O 上的输出驱动器的电源。

位 15:13 保留，必须保持复位值。

位 12 **FXDLAT**：固定延迟 (Fixed latency)
指定是否应以固定的读延迟将所有读取内容返回给控制器。使能固定读延迟会延长读延迟。有效值为：
0: 禁止固定读延迟
1: 使能固定读延迟

位 11 **NOBUB**：无气泡 (No bubble)
指定是否应无气泡地将读取内容返回给控制器。使能无气泡读取会延长读延迟。有效值为：
0: 读取时允许有气泡
1: 读取时不允许有气泡

位 10:8 **DQSGE[2:0]**：DQS 门控提前 (DQS gate early)

指定使能 DQS 门控时必须比正常使能位置提前的时钟周期数。仅适用于使用 PDQSR I/O 单元、被动 DQS 门控且无漂移补偿的情况。

对于非 LPDDR2/LPDDR3 的所有 DDR 类型，建议将此位字段设置为零。

对于 LPDDR2/LPDDR3，应将其值设置为 (tDQSCKmax - tDQSCKmin) 除以时钟周期数并向上取整。tDQSCKmax 和 tDQSCKmin 在 LPDDR/LPDDR2 供应商数据手册中提供。

位 7:5 **DQSGX[2:0]**：DQS 门控延时 (DQS gate extension)

指定 DQS 门控必须延时超出正常突发传输长度宽度的时钟周期数。仅适用于使用 PDQSR I/O 单元、被动 DQS 门控且无漂移补偿的情况。

对于非 LPDDR2/LPDDR3 的所有 DDR 类型，建议将此位字段设置为零。

对于 LPDDR2/LPDDR3，应将其值设置为 (tDQSCKmax - tDQSCKmin) 除以时钟周期数并向上取整。tDQSCKmax 和 tDQSCKmin 在 LPDDR2 供应商数据手册中提供。

位 4 **LPDLLPD**：低功耗 DLL 掉电 (Low power DLL power down)

如果置 1，则指定 PHY 应响应 DFI 低功耗请求，并在唤醒时间请求满足 DLL 锁相时间的情况下关闭 PHY 的 DLL 电源。

有关使能和禁止 DLL 时的要求的更多信息，请参见“旁路模式寄存器控制”。

位 3 **LPIOPD**：低功耗 I/O 掉电 (Low power I/O power down)

如果置 1，则指定 PHY 应响应 DFI 低功耗请求，并关闭 PHY 的 I/O 电源。

位 2 **ZUEN**：DFI 更新请求时进行阻抗校准 (zcal on DFI update request)

如果置 1，则指定只要有控制器发起的 DFI 更新请求，PHY 就应执行阻抗校准（更新）。否则，PHY 将忽略来自控制器的更新请求。

位 1 **BDISEN**：字节禁止使能 (Byte disable enable)

如果置 1 则指定 PHY 应响应 DFI 字节禁止请求。否则将忽略来自 DFI 的字节禁止，这种情况下，只能使用 DXnGCR 寄存器禁止字节

位 0 **PUREN**：PHY 更新请求使能 (PHY update request enable)

若该位被置位，表示当在一个连续（背对背）的读突发操作中，DQS 信号漂移超过 ¾ 个时钟周期时，PHY 应发出 PHY 启动的 DFI 更新请求。

默认情况下，PHY 发出 PHY 发起的更新请求，控制器应给予响应，否则 PHY 可能返回错误值。

提供禁止该位的选项，仅用于硅片评估和测试。

7.7.13 DDRPHYC DC 寄存器 (DDRPHYC_DCR)

DDRPHYC DC register

偏移地址：0x030

复位值：0x0000 000B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TPD	RDIMM	UDIMM	DDR2T	NOSRA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	DDRTYPE[1:0]		MPRDQ	PDQ[2:0]			DDR8BNK	DDRMD[2:0]		
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **TPD**：测试掉电 (Test power down)（不适用，仅限 LPDDR）

如果置 1，则将 DRAM 置于深度掉电模式。

位 30 **RDIMM**：已注册 DIMM (Registered DIMM)

若该位被置位，则表明使用的是寄存式双列直插式内存模块 (Registered DIMM)。在此情况下，PUBL（可能是特定的组件或模块）会将同步动态随机存取内存 (SDRAM) 的写入和读取延迟 (WL/RL) 增加 1。

这仅适用于 PUBL 内部 SDRAM 事务。使用已注册 DIMM 时，控制器生成的事务必须自行调整 WL/RL。

位 29 **UDIMM**：无缓冲 DIMM (Unbuffered DIMM)

若该位被置位，则表示在无缓冲双列直插式内存模块 (UDIMM) 的第二通道（连接到 CS#[1] 的通道）上存在地址镜像。

在此情况下，PUBL 在向第二通道发送模式寄存器命令时，会重新整理 bank 和地址。

这仅适用于 PUBL 内部的 SDRAM 事务。当使用无缓冲双列直插式内存模块时，控制器生成的事务必须自行进行调整。若启用地址镜像功能，则必须设置 DCR [NOSRA]。

位 28 **DDR2T**：2T 时序 (2T timing)

如果置 1，则表示由 PUB 内部生成的 SDRAM 事务应使用 2T 时序。

位 27 **NOSRA**：不允许同时 rank 访问 (No simultaneous rank access)

如果置 1，则表示不允许在同一个时钟周期内同时进行 rank 访问。这意味着不应同时将多个片选信号置为有效。但某些 DIMM 系统可能需要如此。

位 26:10 保留，必须保持复位值。

位 9:8 **DDRTYPE[1:0]**：DDR 类型 (DDR type) (LPDDR2 S4)

选择指定 LPDDR 模式的 DDR 类型。

LPDDR2 的有效值包括：

00: LPDDR2-S4

01: LPDDR2-S2

其他值：Reserved

位 7 **MPRDQ**：MPR DQ

指定在 MPR 读取期间在非主 DQ 引脚上驱动的值。

有效值为：

0: 主 DQ 从 MPR 驱动出数据 (0-1-0-1)；非主 DQ 驱动“0”

1: 主 DQ 和非主 DQ 均从 MPR 驱动相同的数据 (0-1-0-1)

位 6:4 **PDQ[2:0]**：主 DQ (Primary DQ)（仅 DDR3）

指定字节中被指定为用于多用途寄存器 (MPR) 读的主引脚的 DQ 引脚。DQ[0] 到 DQ[7] 的有效值分别为 0 至 7。

位 3 **DDR8BNK**：DDR 8 bank

如果置 1，则表示所用的 SDRAM 有 8 个 bank。tRPA = tRP+1 和 tFAW 用于 8 bank 的 DRAM，其他情况 tRPA = tRP 且不使用 tFAW。

注： 对于少于 8 个 bank 的 DRAM，设置为 1 仍然能保持正确运行，但此处所述参数的 DRAM 命令间隔不那么紧密。

位 2:0 **DDRMD[2:0]**：SDRAM DDR 模式 (SDRAM DDR mode)

011: DDR3

100: LPDDR2（移动 DDR2）

101: LPDDR3（移动 DDR3）

其他值：Reserved

7.7.14 DDRPHYC DTP 寄存器 0 (DDRPHYC_DTPR0)

DDRPHYC DTP register 0

偏移地址：0x034

复位值：0x3092 666E

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCCD	TRC[5:0]							TRRD[3:0]				TRAS[4:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRCD[3:0]				TRP[3:0]				TWTR[2:0]			TRTP[2:0]			TMRD[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **TCCD**：tCCD 读命令间延迟和写命令间延迟

0: DDR2 为 BL/2，DDR3 为 4

1: DDR2 为 BL/2 + 1，DDR3 为 5

位 30:25 **TRC[5:0]**：tRC

激活与激活间的命令延迟（同一 bank）。有效值为 2 到 42。

位 24:21 **TRRD[3:0]**：tRRD

激活与激活间的命令延迟（不同 bank）。有效值为 1 到 8。

位 20:16 **TRAS[4:0]**：tRAS

激活与预充电间的命令延迟。有效值为 2 到 31。

位 15:12 **TRCD[3:0]**：tRCD

激活与读或写之间的延迟。从发出激活命令到可以对被激活行发出读或写命令的最短时间。有效值为 2 到 11。

位 11:8 **TRP[3:0]**：tRP

预充电命令周期：预充电命令与任何其他命令之间的最小时间间隔。

注： 注意，控制器会通过将 tRP 加 1 自动得出 8 bank DDR2 设备的 tRPA。有效值为 2 到 11。

位 7:5 **TWTR[2:0]**：tWTR
内部写与读命令间的延迟。有效值为 1 到 6。

位 4:2 **TRTP[2:0]**：tRTP
内部读取到预充电命令的延迟。有效值范围是 2 到 6。请注意，尽管读取到预充电延迟 (RTP) 不适用于 JEDEC（固态技术协会）标准的 DDR 设备，但对于 DDR 而言，此参数仍必须设置为最小值 2，因为控制器始终使用 DDR2 的计算公式“AL + BL/2 + max (RTP, 2) - 2”来计算读取到预充电的时序（对于 JEDEC 标准的 DDR，该时序为 BL/2）。

位 1:0 **TMRD[1:0]**：tMRD
加载模式循环时间 (Load mode cycle time)：加载模式寄存器命令与任何其他命令之间的最短时间。
对于 DDR3，是两个加载模式寄存器命令之间的最短时间。
对于 DDR3，用于 tMRD 的值是这些位中编程的值加 4，即 DDR3 的 tMRD 值范围是 4 到 7。
对于 LPDDR3，用于 tMRD 的值是这些位中编程的值加 8。

7.7.15 DDRPHYC DTP 寄存器 1 (DDRPHYC_DTPR1)

DDRPHYC DTP register 1

偏移地址：0x038

复位值：0x0983 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	TDQSCKMAX[2:0]			TDQSCKMIN[2:0]			TRFC[7:0]							
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TRTODT	TMOD[1:0]			TFAW[5:0]					TRTW	TAOND[1:0]	
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29:27 **TDQSCKMAX[2:0]**：tDQSCKmax
从 CK/CK# 进行的最长 DQS 输出访问时间（仅限 LPDDR2）。
该值用于实现读到写间隔。有效值为 1 到 7。

位 26:24 **TDQSCKMIN[2:0]**：tDQSCKmin
从 CK/CK# 进行的 DQS 输出访问时间（仅限 LPDDR2/3）。
该值用于计算读延迟。有效值为 0 到 7。
该值的获得方法是将 SDRAM 数据手册中的相应参数除以时钟周期时间，不进行四舍五入。
小数余数会通过四分之一时钟周期单位的数据训练自动调整。
如果未进行数据训练，则必须将这个小数余数转换为四分之一时钟周期单位，并相应地调整门控寄存器 (DXnDQSTR)。

位 23:16 **TRFC[7:0]**：tRFC
刷新到刷新：表示两个刷新命令或刷新与激活命令之间的最小时钟周期数。这是由数据手册中的最小刷新间隔 tRFC(min) 除以时钟周期时间得出的。
默认的时钟周期数适用于支持的最大 JEDEC tRFC(min) 参数值。

位 15:12 保留，必须保持复位值。

位 11 **TRTODT**：tRTODT

读到 ODT 延迟（仅限 DDR3）。指定是否可以在读取后置码后立即使能 ODT，或者是否需要增加一个时钟延迟。

有效值为：

0: 可以在读取后置码后立即打开 ODT

1: 读取后置码之后要等待一个时钟周期才能打开 ODT

如果 tRTODT 设置为 1，则若使能了 ODT，读到写延迟会增加 1。

位 10:9 **TMOD[1:0]**：tMOD

加载模式更新延迟 (Load mode update delay)（仅限 DDR3）。加载模式寄存器命令与非加载模式寄存器命令之间的最短时间。

有效值为：

00: 12

01: 13

10: 14

11: 15

位 8:3 **TFAW[5:0]**：tFAW

4-bank 激活周期。在给定的 tFAW 周期内，发出的 4-bank 激活命令不得超过此数量。仅适用于 8 bank 的设备。

有效值为 2 到 31。

位 2 **TRTW**：tRTW

读到写命令延迟 (Read to Write command delay)。

有效值为：

0：标准总线周转延迟

1：将标准总线周转延迟加上 1 个时钟周期

该参数允许用户增加在读命令之后向 SDRAM 发出写命令之间的延迟。这为高频系统提供了增加总线周转裕量的选项。

位 1:0 **TAOND[1:0]**：tAOND/tAOFD

ODT 开启/关闭延迟 (ODT turn-on/turn-off delays)（仅限 DDR2）。延迟以时钟周期数为单位。

有效值为：

00: 2/2.5

01: 3/3.5

10: 4/4.5

11: 5/5.5

大多数 DDR2 设备使用固定值 2/2.5。对于非标准 SDRAM，用户必须确保操作写延迟始终大于或等于 ODT 开启延迟。例如，CAS 延迟设置为 3 且 CAS 附加延迟设置为 0 的 DDR2 SDRAM 的写延迟为 2。因此可以使用 2/2.5，但不能使用 3/3.5 或更高。

7.7.16 DDRPHYC DTP 寄存器 2 (DDRPHYC_DTPR2)

DDRPHYC DTP register 2

偏移地址：0x03C

复位值：0x1001 A0C8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	TDLLK[9:0]										TCKE[3:1]		
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCKE [0]	TXP[4:0]					TXS[9:0]									
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:19 **TDLLK[9:0]**：tDLLK

PLL 锁相时间 (DLL locking time)。有效值为 2 到 1023。

位 18:15 **TCKE[3:0]**：tCKE

CKE 最小脉宽 (CKE minimum pulse width)。还指定 SDRAM 必须保持在掉电或自刷新模式下的最短时间。对于 DDR3，该参数必须设置为 tCKESR 的值，通常大于 tCKE 的值。有效值为 2 到 15。

位 14:10 **TXP[4:0]**：tXP

掉电退出延迟 (Power down exit delay)。掉电退出命令与任何其他命令之间的最短时间。该参数必须设置为 SDRAM 数据手册中指定的不同最小掉电退出延迟参数的最大值，即 DDR2 的 max(tXP, tXARD, tXARDS)，DDR3 的 max(tXP, tXPDLL)。有效值为 2 到 31。

位 9:0 **TXS[9:0]**：tXS

自刷新退出延迟 (Self refresh exit delay)。自刷新退出命令与任何其他命令之间的最短时间。该参数必须设置为 SDRAM 数据手册中指定的各种最小自刷新退出延迟参数的最大值，即 DDR2 的 max(tXSNR, tXSRD)，DDR3 的 max(tXS, tXSDLL)。有效值为 2 到 1023。

7.7.17 DDRPHYC DDR3 的 MR0 寄存器 (DDRPHYC_DDR3_MR0)

DDRPHYC MR0 register for DDR3

偏移地址：0x040

复位值：0x0A52

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD[2:0]			PD	WR[2:0]			DR	TM	CL[3:1]			BT	CL0	BL[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:13 **RSVD[2:0]**：JEDEC 保留。

位 12 **PD**：掉电控制 (Power-down control)

控制掉电模式的退出时间。有关掉电模式的详细信息，请参见 SDRAM 数据手册。有效值为：

0: 慢速退出（DLL 关闭）

1: 快速退出（DLL 开启）

位 11:9 **WR[2:0]**：写恢复 (Write recovery)

写恢复值以时钟周期数表示。该值的计算方法是，将数据手册的写恢复时间 tWR (ns) 除以数据手册提供的时钟周期时间 tCK (ns)，并将非整数值向上舍入到下一个整数。

有效值为：

001: 5

010: 6

011: 7

100: 8

101: 10

110: 12

所有其他设置均保留，不应使用。

注： tWR (ns) 是从写命令的最后一个数据输入对之后的第一个 SDRAM 正时钟沿，到可以发出同一个 bank 的预充电命令的时间。

位 8 **DR**：DLL 复位 (DLL reset)（自动清零）

向该位写入“1”会将 SDRAM DLL 复位。该位是自清零位，

即在发出 DLL 复位后它会恢复为“0”。

位 7 **TM**：工作模式 (Operating mode)

选择正常工作模式 (0) 或测试模式 (1)。测试模式为制造商保留模式，不应使用。

位 3 **BT**：突发传输类型 (Burst type)

指示突发传输是顺序进行 (0) 还是交错进行 (1)。

位 6、5、4、2 **CL[3:0]**：CAS 延迟 (CAS latency)

从 SDRAM 注册了一个读命令到数据可用之间的延迟（以时钟周期数表示）。

CL[3:0] 的有效值包括：

0010: 5

0100: 6

0110: 7

1000: 8

1010: 9

1100: 10

1110: 11

其他值：Reserved

位 1:0 **BL[1:0]**：突发传输长度 (Burst length)

确定在指定的读或写命令期间可以访问的列位置的最大数量。

DDR3 的有效值包括：

00: 8（固定）

01: 4 或 8（实时）

10: 4（固定）

11: Reserved

7.7.18 DDRPHYC 用于 DDR3 的 MR1 寄存器 (DDRPHYC_DDR3_MR1)

DDRPHYC MR1 register for DDR3

偏移地址：0x044

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	QOFF	TDQS	Res.	RTT2	Res.	LEVEL	RTT1	DIC1	AL[1:0]		RTT0	DIC0	DE
			rw	rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

位 15:13 保留，必须保持复位值。

位 12 **QOFF**：输出使能/禁止 (Output enable/disable)
0：所有输出正常工作
1：所有 SDRAM 输出被禁止，移除输出缓冲电流。
该功能用于读取电流的 IDD 特性分析，不应在正常操作中使用。

位 11 **TDQS**：终端数据选通 (Termination data strobe)
如果使能（“1”），TDQS 将提供额外的终端电阻输出，这在某些系统配置中可能有用。（不适用）

位 10 保留，必须保持复位值。

位 8 保留，必须保持复位值。

位 7 **LEVEL**：写入均衡使能 (Write leveling enable)（不适用）

位 4:3 **AL[1:0]**：发出的 CAS 附加延迟 (Posted CAS Additive Latency)：
设置允许比正常时间提前向 SDRAM 发出读和写命令的附加延迟（详细信息请参见 SDRAM 数据手册）。有效值为：
00: 0（禁止 AL）
01: CL - 1
10: CL - 2
11: Reserved

位 9、6 和 2 **RTT[2:0]**：片上端接 (On die termination)
为片上端接的 SDRAM 选择有效电阻。
{RTT2,RTT1,RTT0} 的有效值包括：
000: 禁止 ODT
001: RZQ/4
010: RZQ/2
011: RZQ/6
100: RZQ/12
101: RZQ/8
其他值：Reserved

位 5 和 1 **DIC[1:0]**：输出驱动器阻抗控制 (Output driver impedance control)
控制输出驱动强度。
{DIC1,DIC0} 的有效值包括：
00: RZQ/6
01: RZQ/7
10: Reserved
11: Reserved

位 0 **DE**：DLL 使能/禁止 (DLL enable/disable)
使能 (0) 或禁止 (1) DLL。
正常操作时必须使能 DLL。

7.7.19 DDRPHYC 用于 LPDDR2 的 MR1 寄存器 (DDRPHYC_LPDDR2_MR1)

DDRPHYC MR1 register for LPDDR2

偏移地址：0x044

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NWR[2:0]			WC	BT	BL[2:0]		
								rw	rw	rw	rw	rw	rw	rw	rw

位 15:8 保留，必须保持复位值。

位 7:5 **NWR[2:0]**：写恢复 (Write recovery)
001：nWR=3
100：nWR=6
110：nWR=8
111：nWR=9
其他值：Reserved

位 4 **WC**：封装控制 (Wrap control)
0：封装
1：不封装

位 3 **BT**：突发传输类型 (Burst Type)：指示突发传输是顺序进行 (0) 还是交错进行 (1)。

位 2:0 **BL[2:0]**：突发传输长度 (Burst length)
确定在指定的读或写命令期间可以访问的列位置的最大数量。有效值为：
010: 4
011: 8
100: 16
其他值：Reserved

7.7.20 DDRPHYC 用于 LPDDR3 的 MR1 寄存器 (DDRPHYC_LPDDR3_MR1)

DDRPHYC MR1 register for LPDDR3

偏移地址：0x044

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NWR[2:0]			Res.	Res.	BL[2:0]		
								rw	rw	rw			rw	rw	rw

位 15:8 保留，必须保持复位值。

位 7:5 **NWR[2:0]**：写恢复 (Write recovery)

如果 nWRE (MR2 [4]) = 0：

——001：nWR=3

——100：nWR=6

——110：nWR=8

——111：nWR=9

如果 nWRE (MR2[4]) = 1：

——000：nWR=10

——001：nWR=11

——010：nWR=12

其他值：Reserved

位 4:3 保留，必须保持复位值。

位 2:0 **BL[2:0]**：突发传输长度 (Burst length)

确定在指定的读或写命令期间可以访问的列位置的最大数量。

011: 8

其他值：Reserved

7.7.21 DDRPHYC 用于 DDR3 的 MR2 寄存器 (DDRPHYC_DDR3_MR2)

DDRPHYC MR2 register for DDR3

偏移地址：0x048

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	RTTWR[1:0]		Res.	SRT	ASR	CWL[2:0]			PASR[2:0]		
					rw	rw		rw	rw	rw	rw	rw	rw	rw	rw

位 15:11 保留，必须保持复位值。

位 10:9 **RTTWR[1:0]**：动态 ODT

选择用于动态 ODT 的 RTT。

00: 动态 ODT 关闭

01: RZQ/4

10: RZQ/2

11: Reserved

位 8 保留，必须保持复位值。

位 7 **SRT**：自刷新温度范围 (Self-refresh temperature range)
在自刷新期间选择正常（“0”）或扩展（“1”）工作温度范围。

位 6 **ASR**：自动自刷新 (Auto self-refresh)
如果使能（“1”），SDRAM 为所有支持的操作温度值自动提供自刷新电源管理功能。
否则必须编程 SRT 位以指示温度范围。

位 5:3 **CWL[2:0]**：CAS 写延迟 (CAS write latency)
从 SDRAM 注册写命令到写数据可用之间的延迟（以时钟周期数表示）。
000: 5 (tCK = 2.5ns)
001: 6 (2.5ns > tCK = 1.875ns)
010: 7 (1.875ns > tCK = 1.5ns)
011: 8 (1.5ns > tCK = 1.25ns)
其他值：Reserved

位 2:0 **PASR[2:0]**：部分阵列自刷新 (Partial array self-refresh)
指定如果进入自刷新模式，则指定位置之外的阵列区域中的数据将丢失。
8 bank 的有效设置包括：
000: 完整阵列
001: 半阵列（BA[2:0] = 000、001、010 和 011）
010: 四分之一阵列（BA[2:0] = 000、001）
011: 1/8 阵列（BA[2:0] = 000）
100: 3/4 阵列（BA[2:0] = 010、011、100、101、110 和 111）
101: 半阵列（BA[2:0] = 100、101、110 和 111）
110: 四分之一阵列（BA[2:0] = 110 和 111）
111: 1/8 阵列（BA[2:0] 111）

7.7.22 DDRPHYC 用于 LPDDR2 的 MR2 寄存器 (DDRPHYC_LPDDR2_MR2)

DDRPHYC MR2 register for LPDDR2

偏移地址：0x048

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RLWL[2:0]		
													rw	rw	rw

位 15:3 保留，必须保持复位值。

位 2:0 **RLWL[2:0]**：读和写延迟 (Read and write latency)
001: RL = 3 /WL = 1
010: RL = 4 / WL = 2
011: RL = 5 / WL = 2
100: RL = 6 / WL = 3
101: RL = 7 / WL = 4
110: RL = 8 / WL = 4
其他值：Reserved

7.7.23 DDRPHYC 用于 LPDDR3 的 MR2 寄存器 (DDRPHYC_LPDDR3_MR2)

DDRPHYC MR2 register for LPDDR3

偏移地址：0x048

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR	WL	Res.	NWRE	Res.	RLWL[2:0]		
								rw	rw		rw		rw	rw	rw

位 15:8 保留，必须保持复位值。

位 7 **WR**：LPDDR3 的新位（该 PHY 未使用此位，保留为零）

位 6 **WL**：LPDDR3 的新位（该 PHY 未使用此位，保留为零）

位 5 保留，必须保持复位值。

位 4 **NWRE**：LPDDR3 的新位（该 PHY 未使用此位，保留为零）

位 3 保留，必须保持复位值。

位 2:0 **RLWL[2:0]**：读和写延迟 (Read and write latency)

001: RL = 3 /WL = 1
010: RL = 4 / WL = 2
011: RL = 5 / WL = 2
100: RL = 6 / WL = 3
101: RL = 7 / WL = 4
110: RL = 8 / WL = 4
其他值：Reserved

7.7.24 DDRPHYC 用于 DDR3 的 MR3 寄存器 (DDRPHYC_DDR3_MR3)

DDRPHYC MR3 register for DDR3

偏移地址：0x04C

复位值：0x00

7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	MPR	MPRLOC[1:0]	
					rw	rw	rw

位 7:3 保留，必须保持复位值。

位 2 **MPR**：多用途寄存器使能 (Multi-purpose register enable)

如果置 1 则使能，允许读取数据来自多用途寄存器。否则读取数据来自 DRAM 数组。

位 1:0 **MPRLOC[1:0]**：多用途寄存器 (MPR) 位置 (Multi-purpose register (MPR) location)

选择 MPR 数据位置。
00: 系统校准的预定义模式
其他值：Reserved

7.7.25 DDRPHYC ODTC 寄存器 (DDRPHYC_ODTCR)

DDRPHYC ODTC register

偏移地址：0x050

复位值：0x0001 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRODT
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDODT
															rw

位 31:17 保留，必须保持复位值。

位 16 **WRODT**：
指定写操作时应使能（“1”）还是禁止（“0”）ODT

位 15:1 保留，必须保持复位值。

位 0 **RDODT**：
指定读操作时应使能（“1”）还是禁止（“0”）ODT

7.7.26 DDRPHYC DTA 寄存器 (DDRPHYC_DTAR)

DDRPHYC DTA register

偏移地址：0x054

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTMPR	DTBANK[2:0]			DTROW[15:4]											
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTROW[3:0]				DTCOL[11:0]											
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **DTMPR**：使用 MPR 进行数据训练 (Data training using MPR)（仅限 DDR3）：
如果置 1，则指定数据训练应使用 SDRAM 多用途寄存器 (MPR)。否则，数据训练通过先写入 SDRAM 中的某些位置，然后再读回来进行。

位 30:28 **DTBANK[2:0]**：数据训练 bank 地址 (Data training bank address)：
选择数据训练期间要使用的 SDRAM bank 地址。

位 27:12 **DTROW[15:0]**: 数据训练行地址 (Data training row address)：
选择数据训练期间要使用的 SDRAM 行地址。

位 11:0 **DTCOL[11:0]**：数据训练列地址 (Data training column address)：
选择数据训练期间要使用的 SDRAM 列地址。
该地址的低四位必须始终为“0000”。

7.7.27 DDRPHYC DTD 寄存器 0 (DDRPHYC_DTDR0)

DDRPHYC DTD register 0

偏移地址：0x058

复位值：0xDD22 EE11

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTBYTE3[7:0]								DTBYTE2[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTBYTE1[7:0]								DTBYTE0[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **DTBYTE3[7:0]**：数据训练数据

数据训练期间所用数据的前 4 个字节（例如，作为字节 0、1、2 和 3 移位）。每个字节通道使用相同的数据字节。

默认序列是在每个数据周期翻转数据时递增 1。

位 23:16 **DTBYTE2[7:0]**：数据训练数据

位 15:8 **DTBYTE1[7:0]**：数据训练数据

位 7:0 **DTBYTE0[7:0]**：数据训练数据

7.7.28 DDRPHYC DTD 寄存器 1 (DDRPHYC_DTDR1)

DDRPHYC DTD register 1

偏移地址：0x05C

复位值：0x7788 BB44

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTBYTE7[7:0]								DTBYTE6[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTBYTE5[7:0]								DTBYTE4[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **DTBYTE7[7:0]**：数据训练数据：

数据训练期间所用数据的第二个 4 字节（例如，作为字节 4、5、6 和 7 移位）。每个字节通道使用相同的数据字节。

默认序列是在每个数据周期翻转数据时递增 1。

位 23:16 **DTBYTE6[7:0]**：数据训练数据

位 15:8 **DTBYTE5[7:0]**：数据训练数据

位 7:0 **DTBYTE4[7:0]**：数据训练数据

7.7.29 DDRPHYC 通用寄存器 0 (DDRPHYC_GPR0)

DDRPHYC general purpose register 0

偏移地址：0x178

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **GPR0[31:0]**：通用寄存器 0 (General purpose register 0) 位

7.7.30 DDRPHYC 通用寄存器 1 (DDRPHYC_GPR1)

DDRPHYC general purpose register 1

偏移地址：0x17C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR1[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **GPR1[31:0]**：通用寄存器 1 (General purpose register 1) 位

7.7.31 DDRPHYC ZQ0C 寄存器 0 (DDRPHYC_ZQ0CR0)

DDRPHYC ZQ0C register 0

偏移地址：0x180

复位值：0x0000 014A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ZQPD	ZCAL	ZCALBYP	ZDEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ZDATA[19:16]			
rw	rw	rw	rw									rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ZDATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **ZQPD**：阻抗校准掉电 (ZCAL power down)
如果置 1，则关闭 PZQ 单元的电源。

位 30 **ZCAL**：ZCAL 触发
阻抗校准触发：向该位写入“1”将触发阻抗控制逻辑执行阻抗校准。
阻抗校准触发位是自清零位，校准完成后会自动回到“0”状态。
注： 如果 ZDEN 置 1，则应将 ZCAL 位设置为 0。

位 29 **ZCALBYP**：阻抗校准旁路 (Impedance calibration bypass)
阻抗校准旁路：如果置 1，则当使用 DDRPHYC_PIR 的 ZCAL 位全局触发阻抗校准时，将禁止此 ZQ 控制模块的阻抗校准。
可以使用 ZCAL 手动触发此 ZQ 模块的阻抗校准。

位 28 **ZDEN**：阻抗覆盖使能 (Impedance override enable)
阻抗覆盖使能：如果该位置 1，则允许用户使用在 ZQDATA 位字段中编程的数据直接驱动阻抗控制。否则，控制将由阻抗控制逻辑自动生成。
注： 如果 ZDEN 置 1，则应将 ZCAL 位设置为 0。

位 27:20 保留，必须保持复位值。

位 19:0 **ZDATA[19:0]**：阻抗覆盖
阻抗覆盖数据：用于直接驱动阻抗控制的数据。
针对 **D3R I/O** 的 ZDATA 位字段映射如下：
ZDATA[27:20] 保留并在读取时返回零
ZDATA[19:15] 用于选择上拉片上端接阻抗
ZDATA[14:10] 用于选择下拉片上端接阻抗
ZDATA[9:5] 用于选择上拉输出阻抗
ZDATA[4:0] 用于选择下拉输出阻抗
注： 有关 ZDATA 编码的信息，请参见 mPHY ST40c25 数据手册，默认值为：ODT 上拉/下拉设置为高阻抗，对于 DDR3 设置为约 44.5 Ω，DDR3L 设置为 48.8 Ω，LPDDR2 典型拐点温度 25C° 设置为 55.5 Ω

7.7.32 DDRPHYC ZQ0CR1 寄存器 (DDRPHYC_ZQ0CR1)

DDRPHYC ZQ0CR1 register

偏移地址：0x184

复位值：0x7B

7	6	5	4	3	2	1	0
ZPROG[7:4]				ZPROG[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw

位 7:4 **ZPROG[7:4]**：外部电阻的阻抗分压比 (Impedance divide ratio to ext R)

阻抗分压比：选择用于设置输出阻抗和片上端接的外部电阻分压比，如下所述：

ZPROG[7:4] = 片上端接分压选择 (RZQ = 240 Ω ±1%)

- 0x0：Reserved
- 0x1：120 Ω
- 0x2: 96 Ω
- 0x3：80 Ω
- 0x4：69 Ω
- 0x5：60 Ω
- 0x6：52Ω
- 0x7：46 Ω
- 0x8：40 Ω
- 0x9：37 Ω
- 0xA：34 Ω
- 0xB：32 Ω
- 0xC：30 Ω
- 0xD：28 Ω
- 0xE：26.5 Ω
- 0xF：25 Ω

位 3:0 **ZPROG[3:0]**：外部电阻的阻抗分压比 (Impedance divide ratio to ext R)

阻抗分压比：选择用于设置输出阻抗和片上端接的外部电阻分压比，如下所述：

ZPROG[3:0] = 输出阻抗分压选择 (RZQ = 240 Ω ±1%)

- 0x0 到 0x4：Reserved
- 0x5：80 Ω
- 0x6：69 Ω
- 0x7：60 Ω
- 0x8：53 Ω
- 0x9：48 Ω
- 0xA：44 Ω
- 0xB：40 Ω
- 0xC：37 Ω
- 0xD：34 Ω
- 0xE：32 Ω
- 0xF：30 Ω

7.7.33 DDRPHYC ZQ0S 寄存器 0 (DDRPHYC_ZQ0SR0)

DDRPHYC ZQ0S register 0

偏移地址：0x188

复位值：0x0000 014A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ZDONE	ZERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ZCTRL[19:16]			
r	r											r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ZCTRL[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31 **ZDONE**：阻抗校准完成 (Impedance calibration done)
指出阻抗校准已完成。

位 30 **ZERR**：阻抗校准错误 (Impedance calibration error)
如果置 1，表示在阻抗校准期间出现错误。

位 29:20 保留，必须保持复位值。

位 19:0 **ZCTRL[19:0]**：阻抗控制 (Impedance control)
阻抗控制：阻抗控制的当前值。
D3R I/O 的 ZZCTRL 位字段映射如下：
ZCTRL[27:20] 保留并在读取时返回零
ZCTRL[19:15] 用于选择上拉片上端接阻抗
ZCTRL[14:10] 用于选择下拉片上端接阻抗
ZCTRL[9:5] 用于选择上拉输出阻抗
ZCTRL[4:0] 用于选择下拉输出阻抗

7.7.34 DDRPHYC ZQ0S 寄存器 1 (DDRPHYC_ZQ0SR1)

DDRPHYC ZQ0S register 1

偏移地址：0x18C

复位值：0x00

7	6	5	4	3	2	1	0
OPU[1:0]		OPD[1:0]		ZPU[1:0]		ZPD[1:0]	
r	r	r	r	r	r	r	r

位 7:6 **OPU[1:0]**：opu 校准状态
片上端接 (ODT) 上拉校准状态。
00: 完成且无错
01: 上溢错误
10: 下溢错误
11: 正在进行校准

位 5:4 **OPD[1:0]**：opd 校准状态
片上端接 (ODT) 下拉校准状态。
00: 完成且无错
01: 上溢错误
10: 下溢错误
11: 正在进行校准

位 3:2 **ZPU[1:0]**：zpu 校准状态
输出阻抗上拉校准状态。有效状态编码包括：
00: 完成且无错
01: 上溢错误
10: 下溢错误
11: 正在进行校准

位 1:0 **ZPD[1:0]**：zpd 校准状态
输出阻抗下拉校准状态。有效状态编码包括：
00: 完成且无错
01: 上溢错误
10: 下溢错误
11: 正在进行校准

7.7.35 DDRPHYC 字节通道 n GC 寄存器 (DDRPHYC_DXnGCR)

DDRPHYC byte lane n GC register

偏移地址：0x1C0 + 0x40 * n，（n = 0 到 1）

复位值：0x0001 0E81

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	R0RVSL[2]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0RVSL[1:0]		RTTOAL	RTTOH[1:0]		DQRTT	DQSRTT	DSEN[1:0]		DQSRPD	DXPDR	DXPDD	DXIOM	DQODT	DQSODT	DXEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:14 **R0RVSL[2:0]**：分步读取有效系统延迟

用于指定相对于 ITMD 读取有效信号理想位置的读取有效系统延迟（DDRPHYC_DXCCR 寄存器中的 RVSEL 位设置为 0 时）。

上电默认为 011（即读取有效信号的理想位置）。

RVSL 位字段最初由 PUB 在自动读有效训练期间设置，但这些值可以通过直接写入此寄存器来覆盖。

有效值为：

000：读取有效系统延迟 = 理想位置 - 3

001：读取有效系统延迟 = 理想位置 - 2

010：读取有效系统延迟 = 理想位置 - 1

011：读取有效系统延迟 = 理想位置

100：读取有效系统延迟 = 理想位置 + 1

101：读取有效系统延迟 = 理想位置 + 2

110：读取有效系统延迟 = 理想位置 + 3

其他值：Reserved

位 13 **RTTOAL**：RTT ON 额外延迟

指示在读取周期过程中，DQ/DQS SSTL I/O 的 ODT 控制何时设置为 DQODT/DQSODT 中的值。

有效值为：

0：在读取数据前导码前约两个周期，将片内终结 (ODT) 控制设置为 DQSODT/DQODT。

1：在读取数据前导码前约一个周期，将片内终结 (ODT) 控制设置为 DQSODT/DQODT。

位 12:11 **RTTOH[1:0]**：RTT 输出暂停 (RTT output hold)

指示当使用动态 ODT 控制时，读取数据后编码之后的时钟周期数（0 到 3），在此期间，ODT 控制应保持设置为 DQSODT（对于 DQS）或 DQODT（对于 DQ/DM），直到禁止 ODT 控制（设置为“0”）为止。

读取后编码之后差不多 RTTOH 个时钟周期后，ODT 被禁止

位 10 **DQRTT**：DQ 动态 RTT 控制 (DQ dynamic RTT control)

如果置 1，则指示可以对 DQ/DM SSTL I/O 的 ODT 控制进行动态控制，具体方法是在读取期间将其设为 DQODT 中的值，在任何其他周期将其禁止（将其设置为“0”）。

如果该位未置 1，则 DQ SSTL I/O 的 ODT 控制始终设置为 DQODT 中的值。

由于动态 ODT 默认开启，使用 LPDDR2/LPDDR3 时，该位必须设置为 0，

因为 LPDDR2/LPDDR3 不需要 ODT 开启。

位 9 **DQSRTT**：DQS 动态 RTT 控制 (DQS dynamic RTT control)

如果置 1，则指示可以对 DQS SSTL I/O 的 ODT 控制进行动态控制，具体方法是在读取期间将其设为 DQSODT 中的值，在任何其他周期将其禁止（将其设置为“0”）。

如果该位未置 1，则 DQS SSTL I/O 的 ODT 控制始终设置为 DQSODT 位字段中的值。

由于动态 ODT 默认开启，使用 LPDDR2/LPDDR3 时，该位必须设置为 0，因为 LPDDR2/LPDDR3 不需要 ODT 开启。

位 8:7 **DSEN[1:0]**：写入 DQS 使能 (Write DQS enable)

控制使能（翻转）还是禁止（静态值）向 SDRAM 的 DQS 写入以及 DQS 是否反转。

DQS# 始终是 DQS 的反转。

仅当 DQS/DQS# 输出使能开启时，这些值才有效，否则 DQS/DQS# 是三态的。

有效设置包括：

00: DQS 禁止（驱动到恒定的 0）

01: DQS 以正常极性翻转（应为默认设置）

10: DQS 以反转后的极性翻转

11: DQS 禁止（驱动到恒定的 1）

位 6 **DQSRPD**：DQSR 掉电 (DQSR power-down)

如果置 1，则关闭 PDQSR 单元的电源。

该位与公共 PDR 配置位进行 OR 运算

位 5 **DXPDR**：数据接收器掉电

如果置 1，则将字节的 DQ、DM 和 DQS/DQS# 引脚上 I/O 的输入接收器电源关闭。该位与公共 PDR 配置位进行 OR 运算

位 4 **DXPDD**：数据驱动器掉电 (Data power-down driver)

如果置 1，则将字节的 DQ、DM 和 DQS/DQS# 引脚上 I/O 的输出驱动器电源关闭。该位与公共 PDD 配置位进行 OR 运算

位 3 **DXIOM**：数据 I/O 模式 (Data I/O mode)

选择字节的 DQ、DM 和 DQS/DQS# 引脚的 I/O 的 SSTL 模式（设为 0 时）或 CMOS 模式（设为 1 时）。

该位与各个 DATX8 的 IOM 配置位进行 OR 运算

位 2 **DQODT**：DQ ODT 使能 (DQ ODT enable)

置位时，启用该字节 DQ 和 DM 引脚 I/O 上的片内终结 (ODT) 功能。

此位会与通用的 DATX8 ODT 配置位进行“或”运算。

位 1 **DQSODT**：DQS ODT 使能 (DQS ODT enable)

置位时，启用该字节 DQS/DQS# 引脚 I/O 上的片内终结功能。

此位与通用的 DATX8 片内终结配置位进行“或”运算。

位 0 **DXEN**：数据字节使能 (DATA byte enable)

如果置 1，则使能用于数据字节的 DATX8 和 SSTL I/O。

如果将该位设置为“0”，将禁止字节，即，字节 SSTL I/O 被置于掉电模式，DATX8 中的 DLL 被置于旁路模式。

将字节通道从禁止改为使能后，必须复位该字节通道的 DLL 并重新锁相。

软件可以使用 DDRPHYC_PIR 寄存器的 DLLSRST 和 DLLLOCK 位来完成此操作（将 DDR PHY 中的所有 DLL 复位并重新锁相）。

7.7.36 DDRPHYC 字节通道 n GS 寄存器 0 (DDRPHYC_DXnGSR0)

DDRPHYC byte lane n GS register 0

偏移地址：0x1C4 + 0x40 *n，（n = 0 到 1）

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTPASS[2:0]			Res.	Res.	Res.	Res.	DTIERR	Res.	Res.	Res.	DTERR	Res.	Res.	Res.	DTDONE
r	r	r					r				r				r

位 15:13 **DTPASS[2:0]**：DQS 训练通过计数 (DQS training pass count)
DQS 门控训练期间的通过配置数量。

位 12:9 保留，必须保持复位值。

位 8 **DTIERR**：DQS 门控训练间歇性出错 (DQS gate training intermittent error)
如果置 1，表示在字节的 DQS 门控训练期间间歇性出错，例如一次通过后下一次失败，然后又通过。

位 7:5 保留，必须保持复位值。

位 4 **DTERR**：DQS 门控训练错误 (DQS gate training error)
如果置 1，表示在字节的 DQS 门控训练期间无法找到有效的 DQS 门控窗口。

位 3:1 保留，必须保持复位值。

位 0 **DTDONE**：数据训练完成 (Data training done)
如果置 1，表示字节已完成数据训练。

7.7.37 DDRPHYC 数据通道 n GS 寄存器 1 (DDRPHYC_DXnGSR1)

DDRPHYC byte lane n GS register 1

偏移地址：0x1C8 + 0x40 * n，（n = 0 到 1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RVPASS[2:0]			Res.	Res.	Res.	RVIERR
									r	r	r				r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	RVERR	Res.	Res.	Res.	Res.	Res.	Res.	DQSDFT[1:0]		Res.	Res.	Res.	DFTERR
			r							r	r				r

位 31:23 保留，必须保持复位值。

位 22:20 **RVPASS[2:0]**：读取有效训练通过计数 (Read valid training pass count)
读取有效训练期间的通过配置数量。

位 19:17 保留，必须保持复位值。

位 16 **RVIERR**：Rank 的读取有效间歇性错误 (RV intermittent error for rank)
如果置 1，表示在字节的读取有效训练期间间歇性出错，例如一次通过后下一次失败，然后又通过。

位 15:13 保留，必须保持复位值。

位 12 **RVERR**：RV 训练错误 (RV training error)
如果置 1，表示在字节的读取有效训练期间无法找到有效的读取有效位置。

位 11:6 保留，必须保持复位值。

位 5:4 **DQSDFT[1:0]**：DQS 漂移值 (DQS drift value)
用于报告数据字节的读取数据选通的漂移。
有效设置包括：
00: 无漂移
01: 90° 漂移
10: 180° 漂移
11: 270° 或更多的漂移

位 3:1 保留，必须保持复位值。

位 0 **DFTERR**：DQS 漂移错误 (DQS drift error)
如果置 1，表示字节读取数据选通的漂移已达到或超出 PHY 通用配置寄存器 (PGCR) 中设置的漂移限制。

7.7.38 DDRPHYC 字节通道 n DLLC 寄存器 (DDRPHYC_DXnDLLCR)

DDRPHYC byte lane n DLLC register

偏移地址：0x1CC + 0x40 * n，（n = 0 到 1）

复位值：0x4000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DLLDIS	DLLSRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDLBMODE	ATESTEN	SDPHASE[3:2]	
rw	rw											rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDPHASE[1:0]		SSTART[1:0]		MFWDLY[2:0]			MFBDLY[2:0]			SFWDLY[2:0]			SFBDLY[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **DLLDIS**：DLL 旁路 (DLL bypass)
已禁止的 DLL 会被旁路。默认（“0”）为使能 DLL

位 30 **DLLSRST**：DLL 复位 (DLL reset)
通过驱动 DLL 软复位引脚软复位字节 DLL

位 29:20 保留，必须保持复位值。

位 19 **SDLBMODE**：回送期间旁路 Slave DLL
如果该位置 1，则 Slave DLL 将被置入回送模式，在该模式下，读取 DQS/DQS# 时没有 90 度的相位偏移。在回送模式下操作字节 PHY 时必须将该位置 1。

位 18 **ATESTEN**：使能引脚“ATO”的路径 (Enable path to pin 'ATO')
使能模拟测试信号在 DLL 模拟测试输出 (DDR_ATO) 上输出。
当该位为“0”时，DLL 模拟测试输出为三态。

位 17:14 **SDPHASE[3:0]**：Slave DLL 相位 (Slave DLL phase)
选择从延迟锁定回路 (DLL) 的输入时钟与相应输出时钟之间的相位差。有效设置：
0000: 90
0001: 72
0010: 54
0011: 36
0100: 108
0101: 90
0110: 72
0111: 54
1000: 126
1001: 108
1010: 90
1011: 72
1100: 144
1101: 126
1110: 108
1111: 90

位 13:12 **SSTART[1:0]**：Slave DLL 自动启动 (Slave DLL autostart)
用于控制 Slave DLL 相对于
Master DLL 锁相的启动方式：
0X：Master DLL 实现锁相后，Slave DLL 将立即自动启动。
10: 禁止 Slave DLL 自动启动；相位检测器被禁止。
11: 禁止 Slave DLL 自动启动；相位检测器被使能。

位 11:9 **MFWDLY[2:0]**：Master DLL 前馈微调 (Master DLL feed-forward trim)
用于微调 Master DLL 前馈路径中的延迟：
000：最小延迟
...
111：最大延迟

位 8:6 **MFBDLY[2:0]**：Master DLL 反馈微调 (Master DLL feed-back trim)
用于微调 Master DLL 反馈路径中的延迟：
000：最小延迟
...
111：最大延迟

位 5:3 **SFWDLY[2:0]**：Slave DLL 前馈微调 (Slave DLL feed-forward trim)
Slave DLL 前馈延迟微调：用于微调 Slave DLL 前馈路径中的延迟：
000：最小延迟
...
111：最大延迟

位 2:0 **SFBDLY[2:0]**：Slave DLL 反馈微调 (Slave DLL feed-back trim)
Slave DLL 反馈延迟微调：用于微调 Slave DLL 反馈路径中的延迟：
000：最小延迟
...
111：最大延迟

7.7.39 DDRPHYC 字节通道 n DQT 寄存器 (DDRPHYC_DXnDQTR)

DDRPHYC byte lane n DQT register

偏移地址：0x1D0 + 0x40 * n，（n = 0 到 1）

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DQDLY7[3:0]				DQDLY6[3:0]				DQDLY5[3:0]				DQDLY4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DQDLY3[3:0]				DQDLY2[3:0]				DQDLY1[3:0]				DQDLY0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 **DQDLY7[3:0]**：位 7 的 DQ 延迟 (DQ delay for bit 7)

用于调整数据延迟相对于标称延迟 (nominal delay)，该标称延迟是通过从延迟锁定回路 (slave DLL) 和时钟树与数据选通脉冲 (data strobes) 的延迟相匹配的。

对于每个 DQ 位的 DQDLY，低两位对由 DQS 提供时钟的数据的延迟进行控制，高两位对由 DQS_b 提供时钟的数据的延迟进行控制。

每个 2 位控制字段的有效设置包括：

00：标称延迟

01：标称延迟 + 1 步

10：标称延迟 + 2 步

11：标称延迟 + 3 步

位 27:24 **DQDLY6[3:0]**：位 6 的 DQ 延迟 (DQ delay for bit 6)

同上

位 23:20 **DQDLY5[3:0]**：位 5 的 DQ 延迟 (DQ delay for bit 5)

同上

位 19:16 **DQDLY4[3:0]**：位 4 的 DQ 延迟 (DQ delay for bit 4)

同上

位 15:12 **DQDLY3[3:0]**：位 3 的 DQ 延迟 (DQ delay for bit 3)

同上

位 11:8 **DQDLY2[3:0]**：位 2 的 DQ 延迟 (DQ delay for bit 2)

同上

位 7:4 **DQDLY1[3:0]**：位 1 的 DQ 延迟 (DQ delay for bit 1)

同上

位 3:0 **DQDLY0[3:0]**：位 0 的 DQ 延迟 (DQ delay for bit 0)

同上

7.7.40 DDRPHYC 字节通道 n DQST 寄存器 (DDRPHYC_DXnDQSTR)

DDRPHYC byte lane n DQST register

偏移地址：0x1D4 + 0x40 * n，（n = 0 到 1）

复位值：0x3DB0 2000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	DMDLY[3:0]				DQSNDLY[2:0]			DQSDLY[2:0]			Res.	Res.	Res.	Res.
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	R0DGPS[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	R0DGSL[2:0]		
		rw	rw										rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29:26 **DMDLY[3:0]**：DM 延迟 (DM delay)

用于调整数据延迟相对于标称延迟 (nominal delay)，该标称延迟是通过从延迟锁定回路 (slave DLL) 和时钟树与数据选通脉冲 (data strobes) 的延迟相匹配的。

对于每个 DQ 位的 DQDLY，低两位对由 DQS 提供时钟的数据的延迟进行控制，高两位对由 DQS_b 提供时钟的数据的延迟进行控制。

每个 2 位控制字段的有效设置包括：

00：标称延迟

01：标称延迟 + 1 步

10：标称延迟 + 2 步

11：标称延迟 + 3 步

位 25:23 **DQSNDLY[2:0]**：DQS# 延迟 (DQS# delay)

用于调整数据延迟相对于标称延迟 (nominal delay)，该标称延迟是通过从延迟锁定回路 (slave DLL) 和时钟树与数据选通脉冲 (data strobes) 的延迟相匹配的。

000：标称延迟 - 3 步

001：标称延迟 - 2 步

010：标称延迟 - 1 步

011：标称延迟

100：标称延迟 + 1 步

101：标称延迟 + 2 步

110：标称延迟 + 3 步

111：标称延迟 + 4 步

注： - 更改此值后，必须发出 ITM 软复位：DDRPHYC_PIR 寄存器中的位 ITMSRST=1，位 INIT=1。

位 22:20 **DQSDLY[2:0]**：DQS 延迟 (DQS delay)

用于调整数据延迟相对于标称延迟 (nominal delay)，该标称延迟是通过从延迟锁定回路 (slave DLL) 和时钟树与数据选通脉冲 (data strobes) 的延迟相匹配的。

000：标称延迟 - 3 步
001：标称延迟 - 2 步
010：标称延迟 - 1 步
011：标称延迟
100：标称延迟 + 1 步
101：标称延迟 + 2 步
110：标称延迟 + 3 步
111：标称延迟 + 4 步

注： - 更改此值后，必须发出 ITM 软复位（DDRPHYC_PIR 寄存器中的位 ITMSRST=1，位 INIT=1）

位 19:14 保留，必须保持复位值。

位 13:12 **R0DGPS[1:0]**：Rank 0 DQS 门控相位选择 (Rank 0 DQS gating phase select)

选择在读取期间用于使能数据选通的时钟，以便筛选出预编码/后编码前后的数据选通值。
R0DGPS 最初由 PUBL 在自动 DQS 数据训练期间设置，并在数据选通漂移补偿期间进行更新。然而，这些值可以通过直接写入此寄存器来覆盖，并且可以使用 PHY 通用配置寄存器 (PGCR) 禁止在 DQS 漂移补偿期间的自动更新。
有效值为：
00: 180 时钟 (clk180)
01: 270 时钟 (clk270)
10: 360 时钟 (clk0)
11: 450 时钟（下一个 clk90）

位 11:3 保留，必须保持复位值。

位 2:0 **R0DGSL[2:0]**：Rank 0 DQS 门控系统延迟 (Rank 0 DQS gating system latency)

用于增加预期有效 DDR 读取数据所需的时钟周期数，最多可额外增加五个时钟周期。
这用于补偿板级延迟和其他系统延迟。
上电默认值为 000（即不需要额外的时钟周期）。
SL 位字段最初由 PUBL 在自动 DQS 数据训练期间设置，但这些值可以通过直接写入此寄存器来覆盖。
有效值为：
000: 无额外的时钟周期
001: 1 个额外时钟周期
010: 2 个额外时钟周期
011: 3 个额外时钟周期
100: 4 个额外时钟周期
101: 5 个额外时钟周期
其他值：Reserved

7.7.41 DDRPHYC 寄存器映射

表 26. DDRPHYC 寄存器映射和复位值

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**DDRPHYC_RIDR**	UDRID[7:0]								PHYMJR [3:0]				PHYMDR [3:0]				PHYMNR [3:0]				PUBMJR [3:0]				PUBMDR [3:0]				PUBMNR [3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
0x004	**DDRPHYC_PIR**	INITBYP	ZCALBYP	LCOKBYP	CLRSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DLLBYP	ICPC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RVTRN	QSTRN	DRAMINIT	DRAMRST	ITMSRST	ZCAL	DLLLOCK	DLLSRST	INIT
	Reset value	0	0	0	0											0	0								0	0	0	0	0	0	0	0	0
0x008	**DDRPHYC_PGCR**	LBMODE	LBGDQS	LBDQSS	RFSHDT [3:0]				PDDISDX	ZKSEL[1:0]		RANKEN [3:0]				IODDRM[1:0]		IOLB	CKINV	CKDV[1:0]		CKEN [2:0]			DTOSEL [3:0]				DFTLMT[1:0]		DFTCMP	DQSCFG	ITMDMD
	Reset value	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	1	0	0
0x00C	**DDRPHYC_PGSR**	TQ	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RVEIRR	RVERR	DFTERR	DTIERR	DTERR	DTDONE	DIDONE	ZCDDONE	DLDONE	IDONE
	Reset value	0																						0	0	0	0	0	0	0	0	0	0
0x010	**DDRPHYC_DLLGCR**	DLLRSVD2[1:0]		LOCKDET	FDTRMSL[1:0]		SBIAS5_3[2:0]			BPS200	SBIAS2_0[2:0]			MBIAS[7:0]								TESTSW	ATC [1:0]		DTC [2:0]			TESTEN	IPUMP[2:0]			DRES[1:0]	
	Reset value	0	0	0	0	0	0	1	1	0	1	1	1	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0x014	**DDRPHYC_ACDLLCR**	DLLDIS	DLLSRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ATESTEN	Res.	Res.	Res.	Res.	Res.	Res.	MFWDLY[2:0]			MFBDLY[2:0]			Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	1												0							0	0	0	0	0	0						
0x018	**DDRPHYC_PTR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TITMSRST[3:0]					TDLLLOCK[11:0]											TDLLSRST[5:0]				
	Reset value											1	0	0	0	1	0	1	0	1	0	1	1	1	1	1	0	0	1	1	0	1	1
0x01C	**DDRPHYC_PTR1**	Res.	Res.	Res.	Res.	Res.	TDINIT1[7:0]							TDINIT0[18:0]																			
	Reset value						1	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	1
0x020	**DDRPHYC_PTR2**	Res.	Res.	Res.	Res.	Res.	TDINIT3[9:0]									TDINIT2[16:0]																	
	Reset value						1	0	0	0	0	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	1	1	0	0	1	0
0x024	**DDRPHYC_ACIOCR**	ACSR[1:0]		RSTIOM	RSTPDR	RSTPDD	RSTODT	RANKPDR[3:0]				CSPDD [3:0]				Res.	Res.	Res.	RANKODT	CKPDR[2:0]		CKPDD[2:0]			CKODT[2:0]				ACPDR	ACPDD	ACODT	ACOE	ACIOM
	Reset value	0	0	1	1	0	0	0	0	0	1	0	0	0	0				0	0	0	1	0	0	0	0	0	0	1	0	0	1	0

表 26. DDRPHYC 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x028	**DDRPHYC_DXCCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDT	RVSEL	DQSNRST	Res.	Res.	DQSNRES[3:0]				DQSRES[3:0]				DXPDR	DXPDD	DXIOM	DXODT
	Reset value																0	0	0			1	0	0	0	0	0	0	0	0	0	0	0
0x02C	**DDRPHYC_DSGCR**	CKOE	RSTOE	ODTOE	CKOE	TPDOE	TPDPD	NL2OE	NL2PD	ODTPDD [3:0]				CKEPDD [3:0]				Res.	Res.	Res.	FXDLAT	NOBUB	DQSGE[2:0]			DQSGX[2:0]			LPDLLPD	LPIOPD	ZUEN	BDISEN	PUREN
	Reset value	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	1	1	1	1	1
0x030	**DDRPHYC_DCR**	TPD	RDIMM	UDIMM	DDR2T	NOSRA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DDRTYPE[1:0]		MPRDQ	PDQ[2:0]			DDR8BNK	DDRMD[2:0]		
	Reset value	0	0	0	0	0																		0	0	0	0	0	0	1	0	1	1
0x034	**DDRPHYC_DTPR0**	TCCD	TRC[5:0]						TRRD[3:0]				TRAS[4:0]					TRCD[3:0]				TRP[3:0]				TWTR [2:0]		TRTP [2:0]			TMRD[1:0]	
	Reset value	0	0	1	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1	1	1	0
0x038	**DDRPHYC_DTPR1**	Res.	Res.	TDQSCKMAX[2:0]			TDQSCKMIN[2:0]			TRFC[7:0]								Res.	Res.	Res.	Res.	TRTODT	TMOD[1:0]			TFAW[5:0]					TRTW	TAOND[1:0]	
	Reset value			0	0	1	0	0	1	1	0	0	0	0	0	1	1					0	0	0	0	1	0	0	1	0	0	0	0
0x03C	**DDRPHYC_DTPR2**	Res.	Res.	Res.	TDLLK[9:0]										TCKE[3:0]			TXP[4:0]				TXS[9:0]											
	Reset value				1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	1	1	0	0	1	0	0	0
0x040	**DDRPHYC_DDR3_MR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RSVD [2:0]			PD	WR[2:0]		DR	TM	CL1 [2:0]			BT	CL0	BL [1:0]	
	Reset value																	0	0	0	0	1	0	1	0	0	1	0	1	0	0	1	0
0x042 - 0x042	Reserved	Res.																															
0x044	**DDRPHYC_DDR3_MR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	QOFF	TDQS	Res.	RTT2	Res.	LEVEL	RTT1	DIC1	AL [1:0]		RTT0	DIC0	DE
	Reset value																				0	0		0		0	0	0	0	0	0	0	0
0x044	**DDRPHYC_LPDDR2_ MR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NWR [2:0]			WC	BT	BL[2:0]		
	Reset value																									0	0	0	0	0	0	0	0
0x044	**DDRPHYC_LPDDR3_ MR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NWR [2:0]			Res.	Res.	BL[2:0]		
	Reset value																									0	0	0			0	0	0
0x046 - 0x046	Reserved	Res.																															

表 26. DDRPHYC 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x048	**DDRPHYC_DDR3_MR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTTWR[1:0]		Res.	SRT	ASR	CWL [2:0]			PASR [2:0]		
	Reset value																						0	0		0	0	0	0	0	0	0	0
0x048	**DDRPHYC_LPDDR2_ MR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RLWL [2:0]		
	Reset value																														0	0	0
0x048	**DDRPHYC_LPDDR3_ MR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR	WL	Res.	NWRE	Res.	RLWL [2:0]		
	Reset value																									0	0		0		0	0	0
0x04A - 0x04B	Reserved	Res.																															
0x04C	**DDRPHYC_DDR3_MR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPR	MPRLOC[1:0]	
	Reset value																														0	0	0
0x04D - 0x04E	Reserved	Res.																															
0x050	**DDRPHYC_ODTCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRODT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDODT
	Reset value																1																0
0x052 - 0x050	Reserved	Res.																															
0x054	**DDRPHYC_DTAR**	DTMPR	DTBANK[2:0]			DTROW[15:0]																	DTCOL[11:0]										
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x058	**DDRPHYC_DTDR0**	DTBYTE3[7:0]								DTBYTE2[7:0]								DTBYTE1[7:0]								DTBYTE0[7:0]						
	Reset value	1	1	0	1	1	1	0	1	0	0	1	0	0	0	1	0	1	1	1	0	1	1	1	0	0	0	0	1	0	0	0	1
0x05C	**DDRPHYC_DTDR1**	DTBYTE7[7:0]								DTBYTE6[7:0]								DTBYTE5[7:0]								DTBYTE4[7:0]						
	Reset value	0	1	1	1	0	1	1	1	1	0	0	0	1	0	0	0	1	0	1	1	1	0	1	1	0	1	0	0	0	1	0	0
0x060 - 0x174	Reserved	Res.																															
0x178	**DDRPHYC_GPR0**	GPR0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x17C	**DDRPHYC_GPR1**	GPR1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x180	**DDRPHYC_ZQ0CR0**	ZQPD	ZCAL	ZCALBYP	ZDEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ZDATA[19:0]																			
	Reset value	0	0	0	0									0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0
0x184	**DDRPHYC_ZQ0CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ZPROG[7:4]					ZPROG[3:0]		
	Reset value																									0	1	1	1	1	0	1	1

表 26. DDRPHYC 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x185 - 0x184	Reserved	Res.																															
0x188	**DDRPHYC_ZQ0SR0**	ZDONE	ZERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ZCTRL[19:0]																			
	Reset value	0	0											0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0
0x18C	**DDRPHYC_ZQ0SR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPU [1:0]		OPD [1:0]		ZPU [1:0]		ZPD [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0x18D - 0x1BC	Reserved	Res.																															
0x1C0	**DDRPHYC_DX0GCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	R0RVSL[2:0]			RTTOAL	RTTOH[1:0]		DQRTT	DQSRTT	DSEN[1:0]		DQSRPD	DXPDR	DXPDD	DXIOM	DQODT	DQSODT	DXEN
	Reset value																1	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	1
0x1C4	**DDRPHYC_DX0GSR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTPASS[2:0]		Res.	Res.	Res.	Res.	Res.	DTIERR	Res.	Res.	Res.	DTERR	Res.	Res.	Res.	DTDONE
	Reset value																	0	0	0					0				0				0
0x1C8	**DDRPHYC_DX0GSR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RVPASS[2:0]			Res.	Res.	Res.	RVIERR	Res.	Res.	Res.	RVERR	Res.	Res.	Res.	Res.	Res.	Res.	DQSDFT[1:0]		Res.	Res.	Res.	DFTERR
	Reset value										0	0	0				0				0							0	0				0
0x1CC	**DDRPHYC_DX0DLLCR**	DLLDIS	DLLSRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDLBMODE	ATESTEN	SDPHASE[3:0]				SSTART[1:0]		MFWDLY[2:0]			MFBDLY[2:0]			SFWDLY[2:0]			SFBDLY[2:0]		
	Reset value	0	1											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1D0	**DDRPHYC_DX0DQTR**	DQDLY7 [3:0]				DQDLY6 [3:0]				DQDLY5 [3:0]				DQDLY4 [3:0]				DQDLY3 [3:0]				DQDLY2 [3:0]				DQDLY1 [3:0]				DQDLY0 [3:0]			
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x1D4	**DDRPHYC_DX0DQSTR**	Res.	Res.	DMDLY [3:0]				DQSNDLY[2:0]			DQSDLY[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	R0DGPS[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	R0DGSL[2:0]		
	Reset value			1	1	1	1	0	1	1	0	1	1							1	0									0	0	0
0x1D8 - 0x1FC	Reserved	Res.																															
0x200	**DDRPHYC_DX1GCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	R0RVSL[2:0]			RTTOAL	RTTOH[1:0]		DQRTT	DQSRTT	DSEN[1:0]		DQSRPD	DXPDR	DXPDD	DXIOM	DQODT	DQSODT	DXEN
	Reset value																1	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	1

表 26. DDRPHYC 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x204	**DDRPHYC_DX1GSR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RVPASS[2:0]			Res.	Res.	Res.	RVIERR	Res.	Res.	Res.	RVERR	Res.	Res.	Res.	Res.	Res.	Res.	DQSDFT[1:0]		Res.	Res.	Res.	DFTERR
	Reset value										0	0	0				0				0							0	0				0
0x208	**DDRPHYC_DX1GSR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RVPASS[2:0]			Res.	Res.	Res.	RVIERR	Res.	Res.	Res.	RVERR	Res.	Res.	Res.	Res.	Res.	Res.	DQSDFT[1:0]		Res.	Res.	Res.	DFTERR
	Reset value										0	0	0				0				0							0	0				0
0x20C	**DDRPHYC_DX1DLLCR**	DLLDIS	DLLSRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDLBMODE	ATESTEN	SDPHASE[3:0]				SSTART[1:0]		MFWDLY[2:0]			MFBDLY[2:0]			SFWDLY[2:0]			SFBDLY[2:0]		
	Reset value	0	1											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x210	**DDRPHYC_DX1DQTR**	DQDLY7 [3:0]				DQDLY6 [3:0]				DQDLY5 [3:0]				DQDLY4 [3:0]				DQDLY3 [3:0]				DQDLY2 [3:0]				DQDLY1 [3:0]				DQDLY0 [3:0]			
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x214	**DDRPHYC_DX1DQSTR**	Res.	Res.	DMDLY [3:0]				DQSNDLY[2:0]			DQSDLY[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	R0DGPS[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	R0DGSL[2:0]		
	Reset value			1	1	1	1	0	1	1	0	1	1							1	0										0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

8 DDR 性能监控器 (DDRPERFM)

8.1 简介

DDRPERFM 旨在通过计数一组信号的事件并在软件控制下收集结果来监控 DDR 控制器性能。

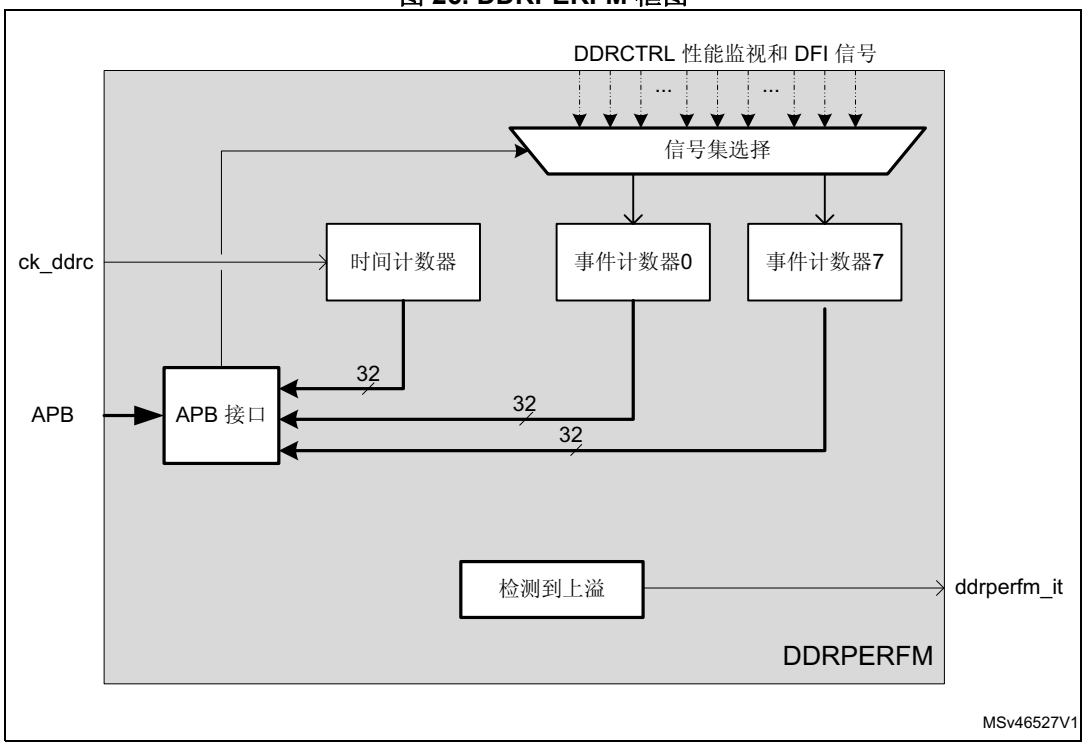
DDRPERFM 可用来反馈重要统计数据，如 DDR 利用率、读/写带宽、bank 激活、闲置状态持续时间，从而可以对 DDR 系统性能进行一些离线分析、优化参数设置以提高 DDR 利用率和进行功耗优化。

8.2 特性

- 一个 32 位时间计数器 (TCNT)
- 四个 32 位事件计数器（CNT0 到 CNT3）
- 可从 4 组要监视的事件中选择
- 全局启动/停止控制
- 单个计数器使能和清零
- 由软件触发全局计数器启动和停止
- 检测到计数器溢出时冻结所有计数器并生成中断事件
- APB 接口

8.3 框图

图 26. DDRPERFM 框图



8.4 功能描述

DDRPERFM 通过根据选择的信号集合报告统计数据，提供 DDR 控制器工作效率的信息。

最多可以监控来自 4 组信号的 4 个信号以及一个永久时钟计数器。

计数器的启动和停止由软件控制。

当一个计数器达到饱和时，所有计数器都会停止。（在 533 MHz DDR 下最快可在 8.05 秒内饱和）

计数器可以在不重置的情况下暂停，以中止计数。

从计数器值中可以导出有用的复合统计数据，这些数据可以帮助开发人员优化 DDR 控制器的设置。例如，优化 DDR 利用率或改进低功耗模式设置。

表 27. DDRPERFM 信号组

组	CNT0	CNT1	CNT2	CNT3
0	perf_op_is_rd	perf_op_is_wr	perf_op_is_activate	ctl_idle
1	Reserved	perf_lpr_req_with_nocredit	Reserved	cactive_ddrc
2	perf_op_is_enter_powerdown	perf_op_is_refresh	perf_selfref_mode	dfi_lp_req
3	Reserved	perf_lpr_xact_when_critical	perf_wr_xact_when_critical	dfi_lp_req

以下是信号描述：

perf_op_is_rd	对通过 CAM 的命令发出的每个读取均置为有效
perf_op_is_wr	对通过 CAM 的命令发出的每个写入均置为有效。
perf_op_is_activate	对通过 CAM 的命令发出的每个激活均置为有效。
ctl_idle	当控制器长时间空闲且仅当 DFI 接口上没有命令或数据时置为有效。
perf_lpr_req_with_nocredit	当因没有可用额度而无法处理低优先级读取 (LPR) 请求（从 XPI 到 PA）时置为有效。
cactive_ddrc	DDRC 硬件低功耗时钟活动。表示 DDRC 需要其时钟信号。
perf_op_is_enter_powerdown	每次进入掉电模式时均置为有效
perf_op_is_refresh	对于控制器发出的每个刷新命令均置为有效。
perf_selfref_mode	在控制器保持在自刷新模式的整个时间内均置为有效。
perf_lpr_xact_when_critical	对于在低优先级队列处于关键状态时调度的每个低优先级读取事务均置为有效。
perf_wr_xact_when_critical	对于在写队列处于关键状态时调度的每个写入事务均置为有效。
dfi_lp_req	DFI 低功耗机会请求。控制器使用此信号通知 PHY 可以转换为低功耗模式。

8.5 应用信息

8.5.1 信号组 0：用于监视 DDR 利用率

通过在时间窗口内计数发送到 DDR 的读写命令，可以监控：

- 读和写访问量（近似）带宽
- DDR 利用率
- DDR 调度效率

注： *DDR 效率通过读写次数与 DDR 非空闲时间之比来衡量。*

示例：

```
read_bandwidth (Mbyte/sec)  ~= 2 * BL * freq * CNT0/TCNT
write_bandwidth (Mbyte/sec) ~= 2 * BL * freq * CNT1/TCNT
DDR_utilization = (BL/2) *(CNT0 + CNT1)/TCNT
with
   BL = 8
   freq = 533 MHz
```

注： *在计算带宽时，假设每次数据传输都是一个完整的突发传输（即一个完整的突发长度，BL）。然而，在某些情况下，写操作可能会使用掩码写突发 (masked write burst)，这意味着并不是所有的数据传输都是完整的突发长度。掩码写突发可能会导致实际传输的数据量少于完整的突发长度。*
因此，如果我们按照每次传输都是完整突发长度来计算带宽，那么在存在掩码写突发的情况下，这样的计算结果可能会高估实际的带宽。

激活命令与读写命令总数之比提供了关于 bank 冲突效率的信息，这些信息可用于在顺序访问 bank 或交错访问 bank 之间选择最佳存储体访问模式。

8.5.2 信号组 1：监视 DDR 命令存储

通过计算 DDRC 存储超出额度的时间来微调 DDRCTRL 调度和性能控制：

- 更好地平衡 DDR 控制器命令存储之间的存储划分。
- 优化饥饿超时。
- 当存储处于关键状态时优化运行长度
- 微调页面打开/关闭策略。

这些设置由 DDRCTRL_SCHED、DDRCTRL_SCHED1、DDRCTRL_LPR1 和 DDRCTRL_WR1 寄存器控制。

8.5.3 信号组 2：监视刷新和自刷新持续时间

刷新命令在 DDR 处于非自刷新状态时以固定间隔发送。
监控在 DDR 空闲时发送的掉电命令，可以用来微调由 DDRCTRL_PWRCTRL 和 DDRCTRL_PWRTMG 控制的低功耗设置。

8.5.4 信号组 3：监视 DDR 停顿

通过计算任何 DDRC 存储处于关键状态的持续时间，可以微调 DDRCTRL 调度和性能控制。如果饥饿计数器超时，队列即进入关键状态。

这些设置由各端口的 DDRCTRL_PCFGR/W、DDRCTRL_PCFGQOS 和 DDRCTRL_CFGWQOS 寄存器控制。

8.6 DDRPERFM 寄存器

8.6.1 DDRPERFM 控制寄存器 (DDRPERFM_CTL)

DDRPERFM control register

偏移地址：0x000

复位值：0x0000 0000

注： *只写寄存器。读请求将返回全零。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STOP	START
														w	w

位 31:2 保留，必须保持复位值。

位 1 **停止**：停止所有计数器。

0：写入“0”无影响。

1：写入“1”停止所有计数器

当计数器停止时（DDRPERFM_STATUS.BUSY = 0），写入“1”将被忽略

位 0 **START**：启动已使能的计数器，时间计数器 (TCNT) 始终处于使能状态。

0：写入“0”无影响。

1：写入“1”会启动已使能的计数器

当计数器正在运行时 (DDRPERFM_STATUS.BUSY = 1)，写入“1”将被忽略

小心： 同时向 START 和 STOP 位写入“1”会产生不可预测的效果。

8.6.2 DDRPERFM 配置寄存器 (DDRPERFM_CFG)

DDRPERFM configurationl register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEL[1:0]	
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN[3:0]			
												rw	rw	rw	rw

位 31:18 保留，必须保持复位值。

位 17:16 **SEL[1:0]**：选择要监视的信号组（从 0 到 3）（参见*表 27：DDRPERFM 信号组*中的信号组描述）和将要使能的计数器

位 15:4 保留，必须保持复位值。

位 3:0 **EN[3:0]**：使能计数器 x（从 0 到 3）

0：对应的计数器被禁止且不会递增计数

1：对应的计数器被使能，并在发生每个时钟事件时递增计数

注： *时间计数器 (TCNT) 默认为使能状态。*

小心： **只能**在 DDRPERFM 停止时修改 DDRPERFM_CFG
如果 DDRPERFM_STATUS.BUSY = 1，将忽略对该寄存器的写操作。

8.6.3 DDRPERFM 状态寄存器 (DDRPERFM_STATUS)

DDRPERFM status register

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TOVF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY
r															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COVF[3:0]			
												r	r	r	r

位 31 **TOVF**：总计数器溢出 (total counter overflow)
0：未上溢
1：计数器上溢

位 30:17 保留，必须保持复位值。

位 16 **BUSY**：忙状态
0: DDRPERFM 已停止或某个计数器溢出
1: DDRPERFM 正在计数（至少 TCNT 在计数）

位 15:4 保留，必须保持复位值。

位 3:0 **COVF[3:0]**：计数器 x 溢出 (Counter x Overflow)（x 为 0 到 3）
0：未上溢
1：计数器上溢

注： *当任何计数器溢出时，DDRPERFM 会停止，所有计数器保持其值。在清除溢出的计数器后，DDRPERFM 需要重新启动。*

计数器溢出也可以通过读取 DDRPERFM_CNTx = 0xFFFF_FFFF 来观察。

8.6.4 DDRPERFM 计数器清零寄存器 (DDRPERFM_CCR)

DDRPERFM counter clear register

偏移地址：0x00C

注： *只写寄存器。读请求将返回全零*

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCLR[3:0]			
												w	w	w	w

位 31 **TCLR**：时间计数器清零 (time counter clear)
0：写入“0”无影响
1：写入“1”清零时间计数器

位 30:4 保留，必须保持复位值。

位 3:0 **CCLR[3:0]**：计数器 x 清零 (counter x Clear)（x 为 0 到 3）
0：写入“0”无影响
1：写入“1”清零相应的事件计数器 x（x 为 0 到 3）

注：　如果 DDRPERFM_STATUS.BUSY = 1，将忽略对该寄存器的写操作。

8.6.5 DDRPERFM 中断使能寄存器 (DDRPERFM_IER)

DDRPERFM interrupt enable register

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVFIE
															rw

位 31:1 保留，必须保持复位值。

位 0 **OVFIE**：溢出中断使能 (overflow interrupt enable)
0：禁止溢出中断
1：使能溢出中断

8.6.6 DDRPERFM 中断状态寄存器 (DDRPERFM_ISR)

DDRPERFM interrupt status register

偏移地址：0x014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVFF
															r

位 31:1 保留，必须保持复位值。

位 0 **OVFF**：溢出标志 (overflow flag)
当一个计数器溢出时，该标志置 1

8.6.7 DDRPERFM 中断清除寄存器 (DDRPERFM_ICR)

DDRPERFM interrupt clear register

偏移地址：0x018

注： *只写寄存器。读请求将返回全零*

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVF
															w

位 31:1 保留，必须保持复位值。

位 0 **OVF**：溢出标志 (overflow flag)
0：不执行任何操作
1：清除溢出标志

注： *计数器溢出后，编程序列必须是：*
1) 通过向 DDRPERFM_CCR 相应位写入“1”清零计数器
2) 通过向 DDRPERFM_ICR.OVF 写入“1”清除中断

8.6.8 DDRPERFM 时间计数器寄存器 (DDRPERFM_TCNT)

DDRPERFM time counter register

偏移地址：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CNT[31:0]**：总时间，即 DDRPERFM 运行期间经过的 DDR 控制器时钟数。

小心： 只能在 DDRPERFM 停止时读取时间计数器。
DDRPERFM_STATUS.BUSY=1 时，该寄存器读为零。

8.6.9 DDRPERFM 事件计数器 x 寄存器 (DDRPERFM_CNTx)

DDRPERFM event counter x register

偏移地址：0x030 + x * 0x008（x = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CNT[31:0]**：事件计数器值 (event counter value)

每个计数器在所选信号组中的相应信号置为有效时递增计数。请参见*表 27：DDRPERFM 信号组*。

小心： 只有在 DDRPERFM 停止时才能读取计数器。
当 DDRPERFM_STATUS.BUSY=1 时，这些寄存器读为零。

8.6.10 DDRPERFM 硬件配置寄存器 (DDRPERFM_HWCFG)

DDRPERFM hardware configuration register

偏移地址：0x3F0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NCNT[3:0]			
												r	r	r	r

位 31:4 保留，必须保持复位值。

位 3:0 **NCNT[3:0]**：此配置的计数器数量 (number of counters for this configuration) (4)

8.6.11 DDRPERFM 版本寄存器 (DDRPERFM_VER)

DDRPERFM version register

偏移地址：0x3F4

复位值：0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本号 (Major revision number)。

位 3:0 **MINREV[3:0]**：次版本号 (Minor revision number)。

8.6.12 DDRPERFM ID 寄存器 (DDRPERFM_ID)

DDRPERFM ID register

偏移地址：0x3F8

复位值：0x0014 0061

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：DDRPERFM 唯一标识。

8.6.13 DDRPERFM 魔术 ID 寄存器 (DDRPERFM_SID)

DDRPERFM magic ID register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：用于自动 IP 发现的魔术 ID。

8.6.14 DDRPERFM 寄存器映射

表 28. 电源控制寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	DDRPERFM_CTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STOP	START
	Reset value																															0	0
0x004	DDRPERFM_CFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEL[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN[3:0]			
	Reset value															0	0													0	0	0	0
0x008	DDRPERFM_STATUS	TOVF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COVF[3:0]			
	Reset value	0															0													0	0	0	0
0x00C	DDRPERFM_CCR	TCLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCLR[3:0]			
	Reset value	0																												0	0	0	0
0x010	DDRPERFM_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVFIE
	Reset value																																0
0x014	DDRPERFM_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVFF
	Reset value																																0
0x018	DDRPERFM_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVF
	Reset value																																0
0x01C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x020	DDRPERFM_TCNT	CNT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024 - 0x02C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x030	DDRPERFM_CNT0	CNT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x034	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x038	DDRPERFM_CNT1	CNT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x040	DDRPERFM_CNT2	CNT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x044	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x048	DDRPERFM_CNT3	CNT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04C - 0x3EC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x3F0	DDRPERFM_HWCFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NCNT[3:0]			
	Reset value																													0	1	0	0
0x3F4	DDRPERFM_VER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	0	0	1	0	0	0	0
0x3F8	DDRPERFM_ID	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1
0x3FC	DDRPERFM_SID	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

9 电源控制 (PWR)

电源控制 (PWR) 一节概述了不同电源域以及电源配置。

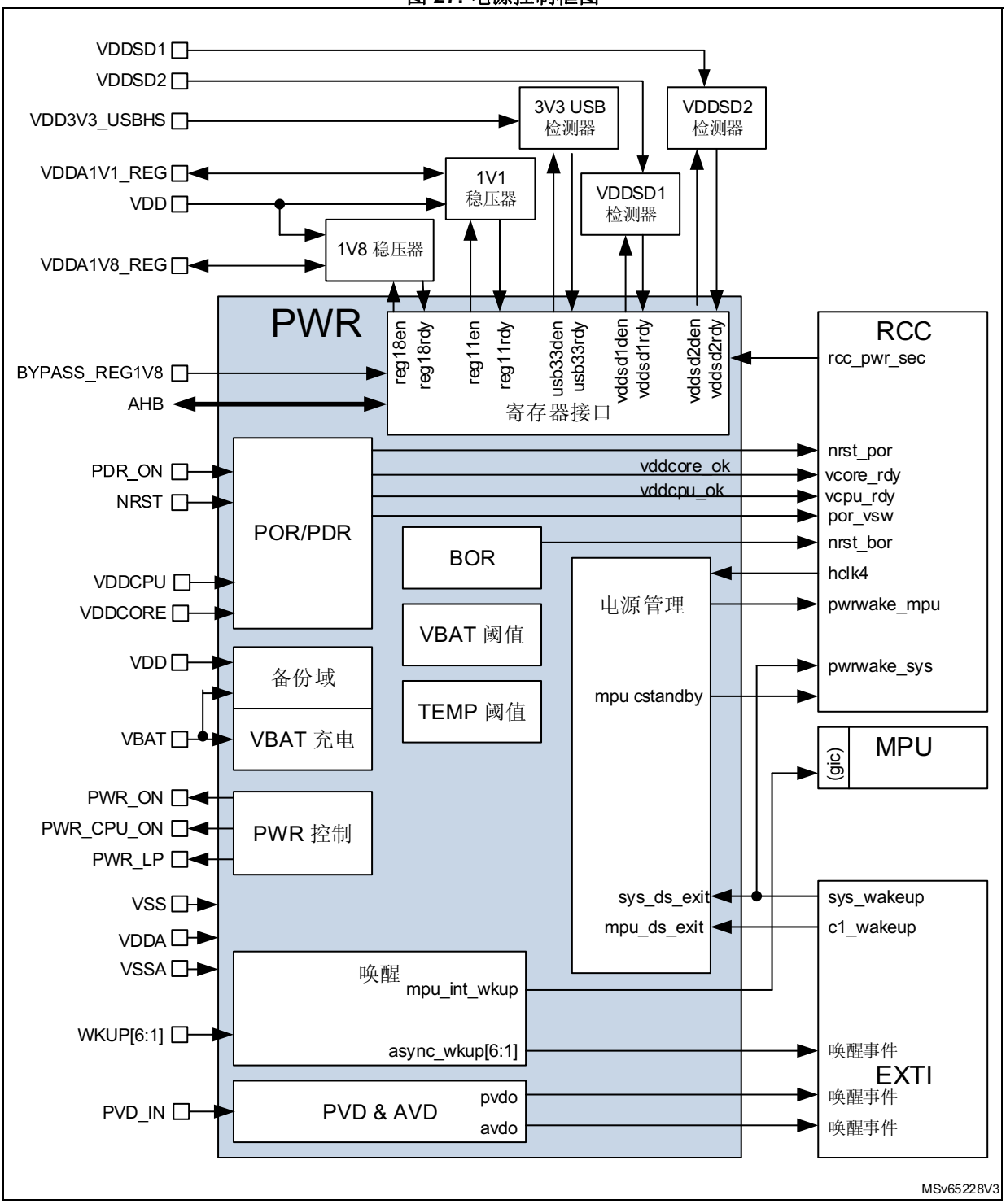
本节描述了电源监控器特性以及根据工作模式对 V_{DDCORE} 和 V_{DDCPU} 电源域进行的控制。在 PWR 一节中，Cortex®-A7 称为 MPU。

9.1 PWR 主要特性

- 系统电源外部稳压控制
  - V_{DDCORE} 和 V_{DDCPU} 供电请求信号 PWR_ON、PWR_CPU_ON
  - V_{DDCORE} 和 V_{DDCPU} 低功耗模式控制信号 PWR_LP
- 外设电源稳压
  - 1.8 V 稳压器（不属于 PWR IP）
  - 1.1 V 稳压器（不属于 PWR IP）
- 电源监控
  - POR/PDR 监视器
  - BOR 监视器
  - PVD 监视器
  - AVD 监视器
  - VBAT 阈值
  - 温度阈值
  - V_{DDCORE} 和 V_{DDCPU} 监视器
- VBAT 充电
- 电源管理
  - 工作模式
  - 低功耗模式
- 寄存器访问支持 TrustZone® 保护

9.2 PWR 框图

图 27. 电源控制框图



9.2.1 引脚概述

表 29. PWR 引脚概述

引脚名称	I/O	说明
VDD	P	主 IO 和 VDD 域电源输入
VDD_PLL	P	PLL 电源输入，必须连接到 V_{DD}（未被 PWR 使用）
VBAT	P	备份电池供电输入
VDDCORE	P	内核域电源
VDDCPU	P	MPU 域电源
VDDA	P	模拟域电源
VDDA1V8_REG	P	1V8 稳压器电源输出
VDDA1V1_REG	P	1V1 稳压器电源输出
VDD3V3_USBHS	P	USB HS PHY 3V3 供电输入，必须连接到 V_{DD3V3_USBHS}（未被 PWR 使用）
VDDSD1	P	SDMMC1 SD 卡 GPIO 电源（可连接至 V_{DD} 或由专用电源供电，以避免在超高速 I (UHS-I) 模式下需要外部驱动器(1)
VDDSD2	P	SDMMC2 SD 卡 GPIO 电源（可连接至 V_{DD} 或由专用电源供电，以避免在超高速 I (UHS-I) 模式下需要外部驱动器(1)
VDDQ_DDR	P	DDR PHY 电源输入（未被 PWR 使用）
VSS	P	主接地
VSSA	P	模拟接地
VSS_USBHS	P	USB HS PHY 接地（未被 PWR 使用）
AHB	I/O	AHB 寄存器接口
PWR_ON(2)	O	内核与 MPU 电源使能输出（使用 STPMICx 稳压器） 内核电源使能输出（未使用 STPMICx 稳压器）
PWR_CPU_ON	O	MPU 电源使能输出（未使用 STPMICx 稳压器）
PWR_LP	O	内核与 MPU 电源低功耗 STOP 模式选择输出（未使用 STPMICx 稳压器）
PDR_ON	I	V_{DD} 上电复位使能
NRST	IO	应用引脚复位输入/输出
BYPASS_REG1V8	I	1V8 稳压器旁路（当 V_{DD} = 1.8 V 时，必须置 1）
WKUP[6:1]	I	唤醒输入
PVD_IN	I	PVD 外部输入
hclk4	I	来自 RCC 的 PWR AHB 总线时钟输入
pvdo	O	可编程电压检测器输出
avdo	O	模拟电压检测器输出
nrst_por	O	V_{DD} 上电复位
nrst_bor	O	V_{DD} 欠压复位
vddcore_ok	O	V_{DDCORE} 就绪

表 29. PWR 引脚概述（续）

引脚名称	I/O	说明
vddcpu_ok	O	V_{DDCPU} 就绪（仅当 vddcore_ok 置为有效时才可置为有效）
sys_ds_exit	I	来自 EXTI 的系统唤醒请求 (sys_wakeup)
mpu_ds_exit	I	来自 EXTI 的 MPU 唤醒请求 (c1_wakeup)
mpu_cstandby	O	MPU CStandby 模式指示
pwrwake_mpu	O	MPU 的时钟唤醒请求
mpu_int_wkup	O	MPU 的唤醒中断
async_wkup[6:1]	O	EXTI 的异步唤醒
rcc_pwr_sec	I	PWR 模块安全状态（来自 RCC 中 PWRSEC 寄存器位）

1. 位于 V_{DDSD1} 和 V_{DDSD2} 电源域中的 GPIO 可用于 SDMMC 以外的其他功能（更多详细信息，请参见数据手册中的“引脚排列、引脚描述和复用功能”部分。）
2. 当使用 STPMICx 稳压器时，仅需 PWR_ON 信号来切换 STPMICx 稳压器的电源模式。

9.3 PWR 电源

电源 V_{DD}、V_{DDCORE}、V_{DDCPU}、V_{DDA}、V_{BAT}、V_{DD1V8_REG}、V_{DD1V1_REG}、V_{DD3V3_USBHS}、V_{DDSD1}、V_{DDSD2} 和 V_{DDQ_DDR} 均为独立电源。设备包含集成稳压器，用于特定功能：

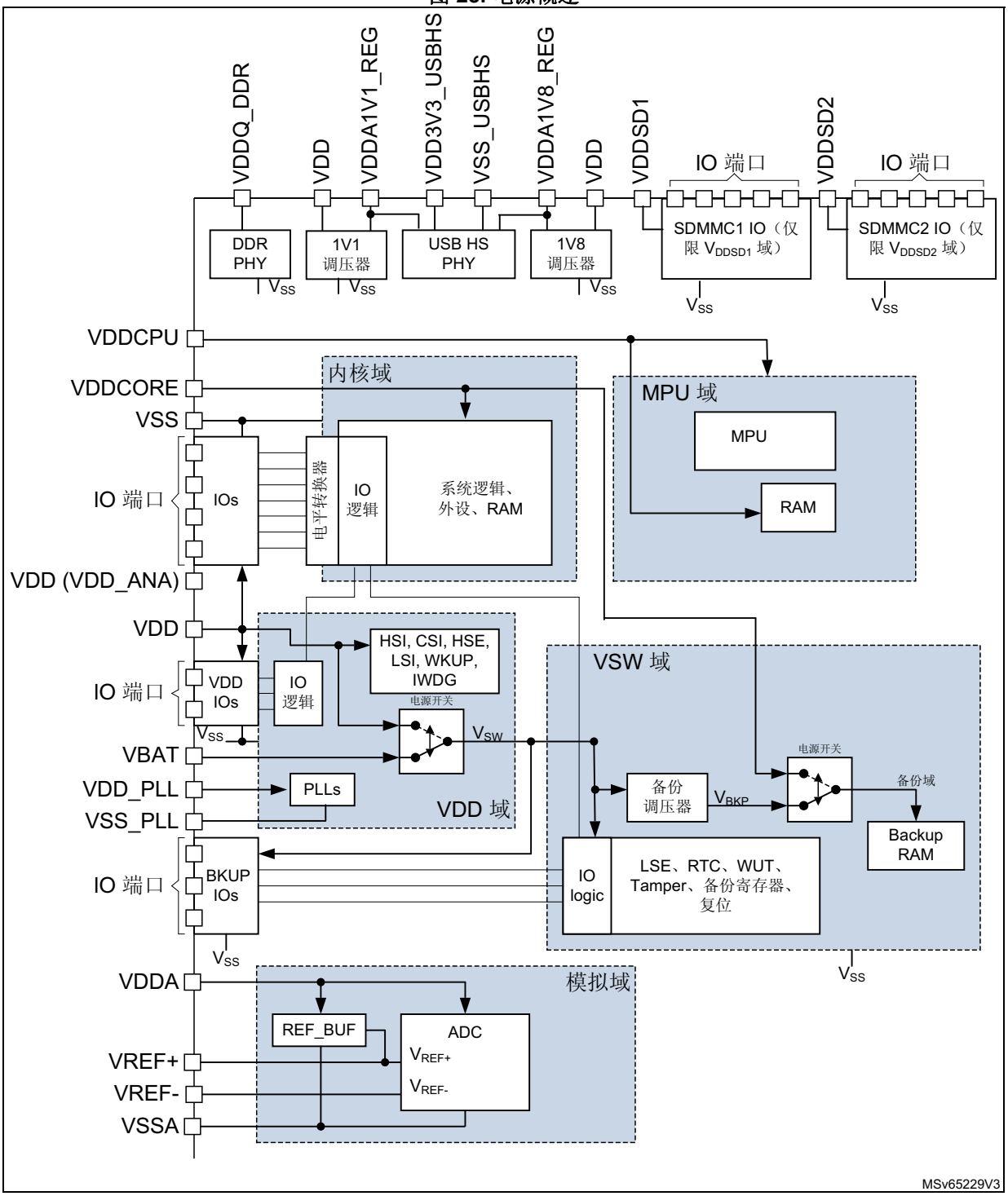
- 电源输入 V_{DD}，用于 I/O（V_{DDSD1} 和 V_{DDSD2} 域除外）及系统模拟功能，如复位、电源管理以及振荡器和 PLLs。
- 电源输入 V_{DDSD1}，为 V_{DDSD1} 电源域（独立于 V_{DD} 和 V_{DDSD2}）中的 I/O 供电[a]。
- 电源输入 V_{DDSD2}，为 V_{DDSD2} 电源域（独立于 V_{DD} 和 V_{DDSD1}）中的 I/O 供电[a]。
- 电源输入 V_{DD_PLL}，为 PLL 供电，连接至与 V_{DD} 相同的电源。
- 电源输入 V_{BAT}（可选），当 V_{DD} 不存在时用于备份域供电（VBAT 模式）
- 数字内核域电源 V_{DDCORE}，与 V_{DD} 电源相关。V_{DD} 必须在 V_{DDCORE} 之前就存在。
- MPU 域电源输入 V_{DDCPU}，与 V_{DD} 电源相关。V_{DD} 必须在 V_{DDCPU} 之前就存在。
- 模拟电源输入 V_{DDA}，为 ADC 和电压参考缓冲器供电，独立于任何其他电源。
- 外部参考电压 V_{REF+}，用于 ADC，独立于任何其他电源。
  - 使能电压参考缓冲器时，用作电源输出。
  - 禁止电压参考缓冲器时，用作独立外部参考电源输入。
- V_{SSA}，独立的模拟和参考电压地。
- 1V8 稳压器电源 V_{DDA1V8_REG}，独立于任何其他电源。
  - 使能 1V8 稳压器时，用作电源输出。
  - 禁止 1V8 稳压器时，用作独立外部电源输入。
- 1V1 稳压器电源 V_{DDA1V1_REG}，独立于任何其他电源。
- 电源输入 V_{DD3V3_USBHS}，为 USB HS PHY 供电。
- V_{SS_USBHS} 独立 USB HS PHY 接地。
- 电源输入 V_{DDQ_DDR}，为 DDR PHY 供电，独立于任何其他电源。
- 所有电源（模拟电源除外）的公共地电压 V_{SS}。

注：根据工作期间供电电压的不同，某些外设可能只提供有限的功能和性能。有关详细信息，请参见数据手册中“General operating conditions”一节。

注：外部 DDR 内存需由不受 PWR 控制的外部稳压器供电。当系统进入低功耗模式时，由固件通过外部稳压器直接控制 DDR 内存电源。当系统进入待机模式时，PWR_ON 引脚可由外部 DDR 内存稳压器使用，可以关断 DDR 内存供电。

a. 位于 V_{DDSD1} 和 V_{DDSD2} 电源域中的 GPIO 可用于 SDMMC 以外的其他功能（更多详细信息，请参见数据手册中的“引脚排列、引脚描述和复用功能”部分。）

图 28. 电源概述


VDDQ_DDR
VDD
VDDA1V1_REG
VDD3V3_USBHS
VSS_USBHS
VDDA1V8_REG
VDD
VDDSD1
IO 端口
VDDSD2
IO 端口
DDR PHY
1V1 调压器
USB HS PHY
1V8 调压器
SDMMC1 IO（仅限 VDDSD1 域）
SDMMC2 IO（仅限 VDDSD2 域）
VSS
VDDCPU
VDDCORE
VSS
内核域
MPU 域
MPU
RAM
IO 端口
IOs
电平转换器
IO 逻辑
系统逻辑、外设、RAM
VDD (VDD_ANA)
VDD
HSI, CSI, HSE, LSI, WKUP, IWDG
VSW 域
IO 端口
VDD IOs
IO 逻辑
电源开关
VSS
VSW
VBAT
VDD_PLL
PLLs
VDD 域
VSS_PLL
电源开关
备份域
备份调压器
VBKP
Backup RAM
IO 端口
BKUP IOs
IO logic
LSE、RTC、WUT、Tamper、备份寄存器、复位
VSS
VSS
VDDA
模拟域
REF_BUF
ADC
VREF+
VREF-
VSSA
MSv65229V3

9.3.1 CORE 和 MPU 电源域

V_{DDCORE} 和 V_{DDCPU} 由外部稳压器提供。

V_{DDCORE} 为所有数字电路供电，但以下除外：

- MPU 域
- 备份域和待机电路（待机模式时）。

V_{DDCPU} 为 MPU 域供电。

在 V_{DD} 上电复位后，通过 PWR_ON 和 PWR_CPU_ON 信号使能提供 V_{DDCORE} 和 V_{DDCPU} 的外部稳压器。

外部稳压器可在不同模式（主模式、低功耗模式和掉电模式）下运行，具体取决于系统工作模式（Run、Stop、LP-Stop、LPLV-Stop、LPLV-Stop2 和 Standby），参见*第 9.5.1 节：工作模式*。

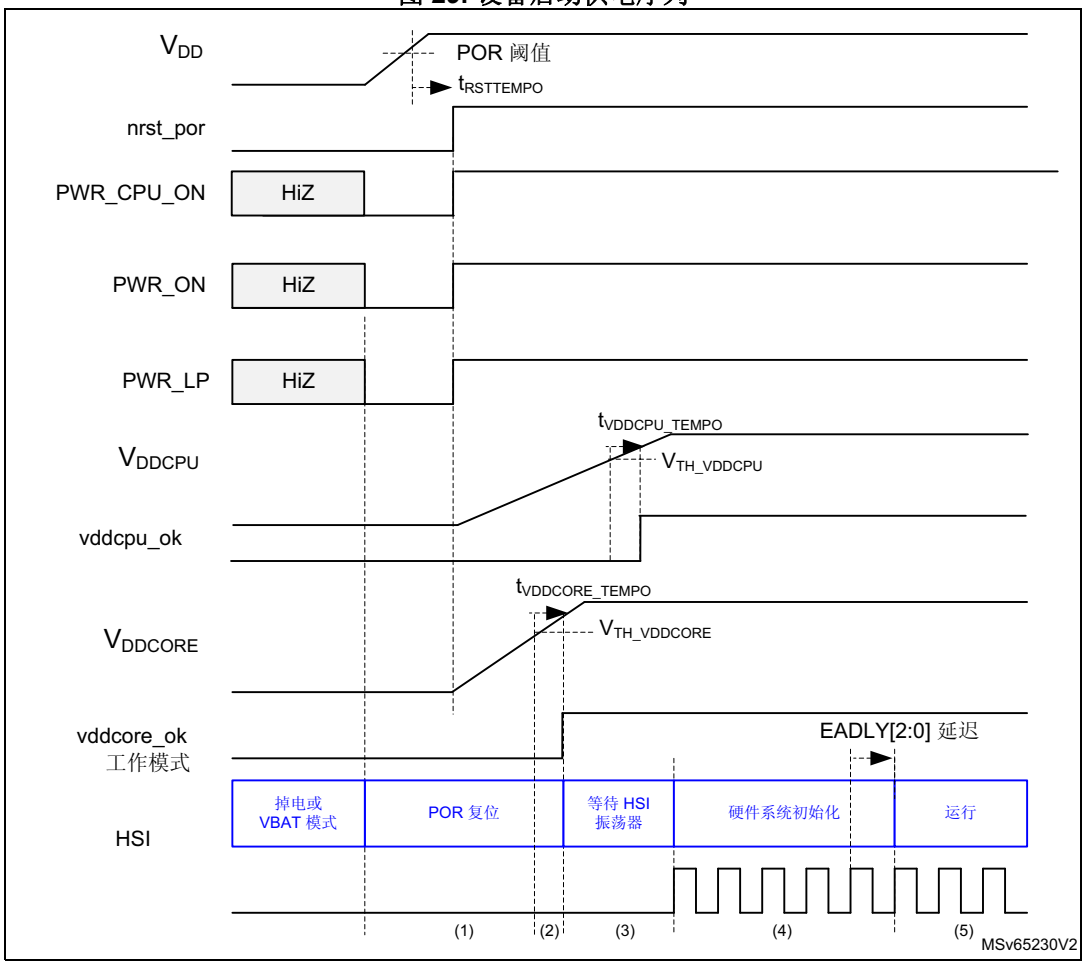
9.3.2 电源系统启动序列

系统启动序列如*图 29* 所示。

1. 从上电开始，POR 会监视 V_{DD} 电源。一旦电源电压超过 POR 阈值，在经过一个内部 $t_{RSTTEMPO}$ 延迟后，通过 PWR_ON 和 PWR_CPU_ON 信号使能提供 V_{DDCORE} 和 V_{DDCPU} 的外部稳压器。只要 V_{DDCORE} 低于 $V_{TH_VDDCORE}$ 阈值，系统就会保持在复位状态（有关 $t_{RSTTEMPO}$ 和 $V_{TH_VDDCORE}$ 的值，参见数据手册）。
2. 一旦 V_{DDCORE} 超过 $V_{TH_VDDCORE}$ 阈值，就会插入一个 $t_{VDDCORE_TEMPO}$ 延迟，以等待 V_{DDCORE} 达到最小工作值（有关 $t_{VDDCORE_TEMPO}$ 和 V_{DDCORE} 的最小工作值，请参见数据手册）。
3. 系统解除复位后，HSI 振荡器将被使能。

   一旦 V_{DDCPU} 超过 V_{TH_VDDCPU} 阈值，就会插入一个 t_{VDDCPU_TEMPO} 延迟，以等待 V_{DDCPU} 达到最小工作值（有关 t_{VDDCPU_TEMPO} 和 V_{DDCPU} 的最小工作值，请参见数据手册）。
4. HSI 振荡器稳定后，硬件便立即开始初始化系统。使用可编程 EADLY[2:0] 延迟确保所有可引导的外部接口在 Boot ROM 开始访问这些接口之前获得正确供电。有关 EADLY[2:0] 编程的详细信息，请参见*第 10 章：复位和时钟控制 (RCC)*。
5. 发出 vddcpu_ok 信号且硬件初始化完成后，便会立即进入运行模式。

图 29. 设备启动供电序列



9.3.3 VSW 域

要在 V_{DD} 关断后保留 VSW 域（RTC、备份寄存器和 Backup RAM）的内容，可将 VBAT 引脚连接到由电池或其他电源提供的可选备用电压。

切换到 V_{BAT} 电源（VBAT 模式）的过程由复位模块中内置的掉电复位 (PDR) 控制，以监视 V_{DD} 电源。

注： *为了在 V_{DD} 关断时保留 VSW 域的内容，必须通过 PDR_ON 引脚（高电平）使能掉电复位 (PDR)。请参见第 9.4.1 节：上电复位 (POR)/掉电复位 (PDR)。*

警告： **在 $t_{RSTTEMPO}$（V_{DD} 启动后的一段延迟）期间或检测到 PDR 后，V_{BAT} 与 V_{DD} 之间的电源开关保持连接到 V_{BAT}。**
在启动阶段，如果 V_{DD} 的建立时间小于 $t_{RSTTEMPO}$（有关 $t_{RSTTEMPO}$ 的值，请参见数据手册）且 $V_{DD} > V_{BAT} + 0.6$ V，会有电流经由 V_{DD} 与电源开关 (V_{BAT}) 之间连接的内部二极管注入 V_{BAT} 引脚。
如果连接到 VBAT 引脚的电源/电池无法承受此注入电流，则强烈建议在该电源与 VBAT 引脚之间连接一个低压降二极管。

如果存在 V_{DD} 电源，则 VSW 域由 V_{DD} 供电。这有助于延长 V_{BAT} 电源电池的使用寿命。

若应用中没有使用外部备份电池，则建议将 V_{BAT} 从外部连至 V_{DD}。

当 V_{DD} 电源存在且供电电压高于 PDR 阈值时，备份域 IO 由 V_{DD} 供电，并且可使用部分功能（根据*表 30*）：

表 30. 存在 V_{DD} 时 VSW 域的功能

引脚名称	GPIO(1)	LSE	RTC	TAMPER_OUT(2)	TAMPER_IN (2)	WAKEUP
PC14、PC15	x	x	-	-	-	-
PC13	x	-	x(3)	x	x	x (WKUP3)
PI2 PI3	x	-	-	x	x	x (WKUP5) x (WKUP2)
PI0	x	-	-	x	x	-
PI1	x	-	x(4)	x	x	x (WKUP4)
PA6、PB2、PC0、PC3、PE14、PF10 和 PG8	x	-	-	-	x	-

1. 如果存在 V_{DDCORE}。
2. 当由 RTC 配置为 Tamper 引脚时。
3. 请参见*表 398：RTC 引脚 PC13 配置*。
4. 请参见*表 399：PI1 配置*。

注： *由于 VSW 开关仅支持有限的电流量，所以使用 PC13 到 PC15 以及 PI0 到 PI3 作为 GPIO 会受到限制。详细信息请参见数据手册。*

当不存在 V_{DD} 电源并且 VBAT 上有电源时，备份域 IO 由 V_{BAT} 供电，以下功能在 VBAT 模式下可用：

表 31. 存在 VBAT 时 VSW 域的功能

引脚名称	GPIO	LSE	RTC	TAMPER_OUT(1)	TAMPER_IN (2)	WAKEUP
PC14、PC15	-	x	-	-	-	-
PC13	-	-	x(2)	x	x	-
PI0、PI2 和 PI3	-	-	-	x	x	-
PI1	-	-	x(3)	x	x	-
PA6、PB2、PC0、PC3、PE14、PF10 和 PG8	-	-	-	-	x	-

1. 当由 RTC 配置为 Tamper 引脚时。
2. 请参见*表 398：RTC 引脚 PC13 配置*。
3. 请参见*表 399：PI1 配置*。

VSW 域访问

系统复位后，RTC 寄存器和 RTC 备份寄存器将受到保护，以防止意外的写访问。要使能对 VSW 域的访问，请按以下步骤操作：

将 *PWR 控制寄存器 1 (PWR_CR1)* 中的 DBP 位置 1，使能对 VSW 域的访问

有关 RTC 和备份寄存器访问的更多详细信息，请参见*第 10 章：复位和时钟控制 (RCC)*。

Backup RAM

备份域包括可在 32 位、16 位 或 8 位模式下访问的 8 KB Backup RAM。

- 在 Run、Stop、LP-Stop、LPLV-Stop 和 LPLV-Stop2 模式下，Backup RAM 由 V_{DDCORE} 供电。
- 在 Standby 和 VBAT 模式下，Backup RAM 通过 VSW 域中的备份稳压器供电。

当备份稳压器通过 *PWR 控制寄存器 2 (PWR_CR2)* 中的 BREN 寄存器位使能时，即使处于待机和/或 VBAT 模式，也会保存 Backup RAM 的内容（如果 V_{BAT} 始终存在，则可将其视作内部 EEPROM。）备份稳压器既可以处于开启状态，也可以处于关闭状态，具体取决于应用在待机模式或 VBAT 模式下是否需要 Backup RAM 功能。

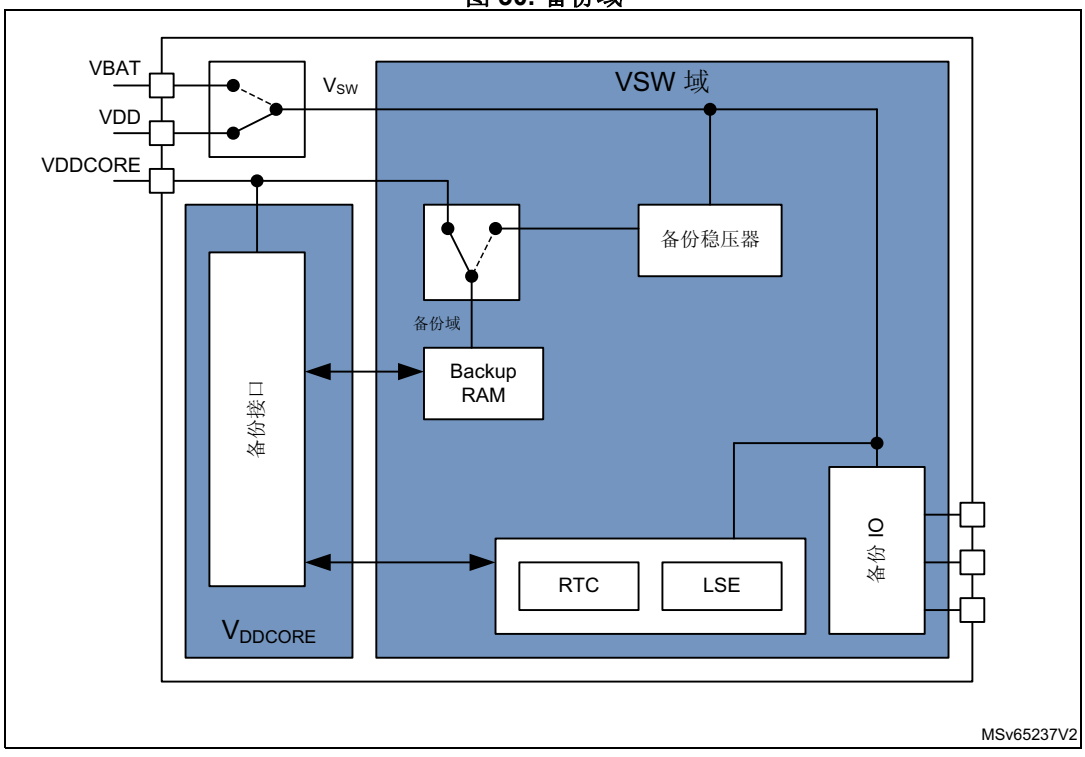
使能备份稳压器时，必须借助 *PWR 控制寄存器 2 (PWR_CR2)* 中的 BRRDY 寄存器位检查备份电池的准备情况，然后再进入待机或 VBAT 模式。

发生 Tamper 事件时，视*第 46 章：实时时钟 (RTC)* TAMP 擦除配置寄存器 (TAMP_ERCFGR) 中 ERCFG0 寄存器位而定，Backup RAM 的内容可能被擦除。

如果 ERCFG0 位被置为无效，则 Backup RAM 不会被擦除，并且在检测到 Tamper 时不会阻止访问。

否则，访问将被阻止，并且在首次对 Backup RAM 进行写访问时其内容将被擦除（如果系统处于 VBAT 模式，则在下一个 POR 之后和首次对 RAM 进行写访问时也如此）。

图 30. 备份域



9.3.4 VBAT 电池充电

当 V_{DD} 存在时，可通过两个电源之间的内部电阻为 VBAT 上的外部电池充电。

VBAT 充电可通过一个 5 kΩ 的电阻或 1.5 kΩ 的电阻来实现，具体取决于 *PWR 控制寄存器 3 (PWR_CR3)* 中 VBRS 位的值。

电池充电通过 *PWR 控制寄存器 3 (PWR_CR3)* 中的 VBE 位来使能/禁止。进入 VBAT 模式后会被自动禁止。

9.3.5 模拟电源

独立的 V_{DDA} 模拟电源

模拟电源域有独立的 V_{DDA} 和 V_{SSA} 引脚，可为电源滤除和屏蔽 PCB 上的噪声，从而可以提高 ADC 的转换精度。

模拟参考电压 V_{REF}

为了提高低电压信号的精度，ADC 有一个单独的参考电压 VREF+ 引脚。用户可在 VREF+ 上连接外部参考电压。V_{REF+} 决定了最高输入电压，由满量程数字值表示。较低的参考电压 V_{REF-} 连接到 V_{SSA}。

当通过*第 28.5.1 节：VREFBUF 控制和状态寄存器 (VREFBUF_CSR)* 中的电压参考缓冲器 ENVR 位使能时，V_{REF+} 由内部电压参考缓冲器提供。内部电压参考缓冲器还可通过同一个 VREF+ 引脚为外部元件提供参考电压，参见*第 28 章：电压参考缓冲器 (VREFBUF)*。

9.3.6 1V8 稳压器

1V8 稳压器由 V_{DD} 电源供电，并产生 V_{DDA1V8_REG} 电源。USB HS PHY 直接连接到此电源。

集成 1V8 稳压器的使用必须通过 BYPASS_REG1V8 引脚（低电平）和 *PWR 控制寄存器 3 (PWR_CR3)* 中的 REG18EN 寄存器位来使能，V_{DDA1V8_REG} 由 1V8 稳压器提供。在使用 V_{DDA1V8_REG} 之前，必须通过 *PWR 控制寄存器 3 (PWR_CR3)* 中的 REG18RDY 寄存器位检查其可用性。

如果通过 BYPASS_REG1V8 引脚（高电平）或 REG18EN 寄存器位禁止（旁路）了 1V8 稳压器，则 V_{DDA1V8_REG} 可以由外部电源提供。

使用 V_{DD} = 1.8 V 时，必须旁路 1V8 稳压器 (BYPASS_REG1V8 = 1)。

有关 1V8 稳压器的更多信息，请参见*第 9.3 节：PWR 电源*。

9.3.7 1V1 稳压器

1V1 稳压器由 V_{DD} 供电，并产生 V_{DDA1V1_REG} 电源。USB HS PHY 直接连接到此电源。

1V1 稳压器由 *PWR 控制寄存器 3 (PWR_CR3)* 中的 REG11EN 寄存器位使能。在使用 1V1 稳压器供电之前，必须通过 *PWR 控制寄存器 3 (PWR_CR3)* 中的 REG11RDY 寄存器位检查其可用性。

有关 1V1 稳压器的更多信息，请参见*第 9.3 节：PWR 电源*。

9.4 PWR 电源监控

电源电平监控可用于以下电源：

- V_{DD}，通过 POR/PDR（*第 9.4.1 节*）、BOR（*第 9.4.2 节*）和 PVD 监视器（*第 9.4.3 节*）监视
  - POR/PDR 监视标志是 RCC 中的 PORRSTF 寄存器位。
  - BOR 监视标志是 RCC 中的 BORRSTF 寄存器位。
  - PVD 监视标志是 *PWR 控制状态寄存器 1 (PWR_CSR1)* 中的 PVDO 寄存器位。此外，可通过 EXTI 生成中断和唤醒。
- V_{DDSD1}，通过 VDDSD1RDY *PWR 控制寄存器 3 (PWR_CR3)* 监视。
- V_{DDSD2}，通过 VDDSD2RDY *PWR 控制寄存器 3 (PWR_CR3)* 监视。
- V_{DDA}，通过 AVD 监视器监视（*第 9.4.5 节*）
  - AVD 监视标志是 *PWR 控制状态寄存器 1 (PWR_CSR1)* 中的 AVDO 寄存器位。此外，可通过 EXTI 生成中断和唤醒。
- V_{BAT}，通过 VBAT 阈值监视（*第 9.4.6 节*）。
  - VBAT 监视标志是 *PWR 控制寄存器 2 (PWR_CR2)* 中的 VBATH 和 VBATL 寄存器位。此外，可通过 EXTI 生成中断和唤醒。
- V_{DDCORE}，通过 vddcore_ok 监视，只要电压不正常，就会使 V_{DDCORE} 域保持复位状态（*第 9.5 节*）。
  - V_{DDCORE} OK 监视标志是 *PWR MPU 控制寄存器 (PWR_MPUCR)* 中的 SBF 存器位和 RCC 中的 VCORERSTF 寄存器位。
- V_{DDCPU}，通过 vddcpu_ok 监视，只要电压不正常，就会使 V_{DDCPU} 域保持复位状态（*第 9.5 节*）。
  - V_{DDCPU} OK 监视标志是 *PWR MPU 控制寄存器 (PWR_MPUCR)* 中的 SBFMPU 寄存器位和 RCC 中的 VCPURSTF 寄存器位。
- V_{SW}，通过 rst_vsw 监视，只要电压不正常，就会使 VSW 域保持复位状态。
  - V_{SW} 监视没有标志。可以使用 VSW 寄存器复位值。
- V_{BKP}，通过 *PWR 控制寄存器 2 (PWR_CR2)* 中的 BRRDY 监视。
- V_{DD3V3_USBHS}，通过 *PWR 控制寄存器 3 (PWR_CR3)* 中的 USB33RDY 监视。
- V_{DDA1V8_REG}，通过 *PWR 控制寄存器 3 (PWR_CR3)* 中的 REG18RDY 监视。
- V_{DDA1V1_REG}，通过 *PWR 控制寄存器 3 (PWR_CR3)* 中的 REG11RDY 监视

9.4.1 上电复位 (POR)/掉电复位 (PDR)

系统内部集成有 POR/PDR 电路，可实现正常的启动操作。

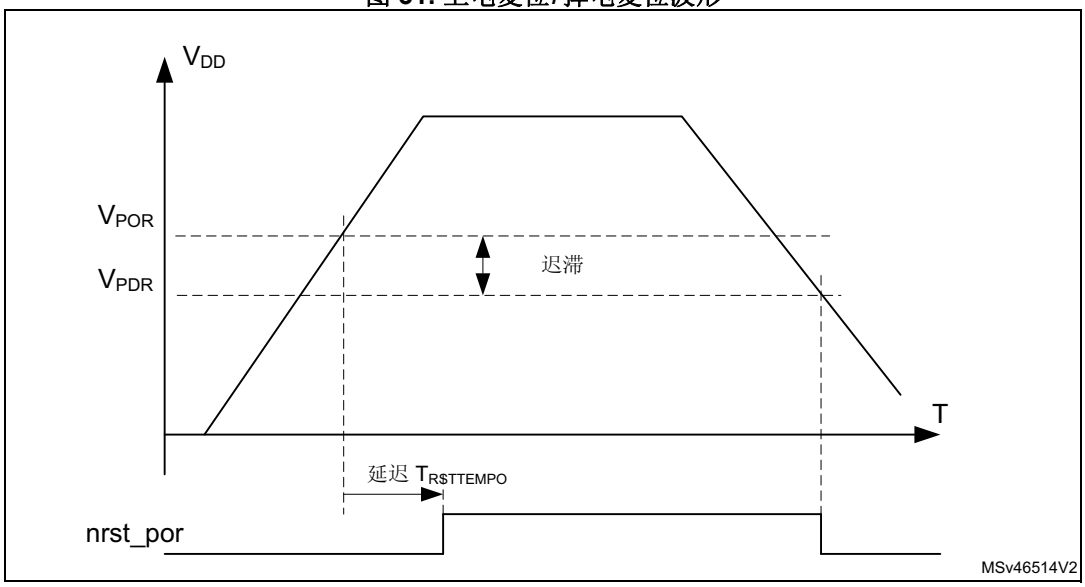
当 V_{DD} 低于指定的 V_{POR} 阈值时，系统无需外部复位电路便会保持复位模式。一旦 V_{DD} 电源电压高于 V_{POR} 阈值，系统便会退出复位状态。有关上电/掉电复位阈值的详细信息，请参见数据手册的电气特性部分。

POR/PDR 也会产生应用复位 (NRST)。

POR/PDR 可通过设备输入引脚 PDR_ON 使能/禁止。禁止时，NRST 设备输入引脚上必须提供 V_{DD} 上电复位。

使用低功耗模式 LPLV-Stop 和 LPLV-Stop2 时，应保持 PDR_ON 高电平。

图 31. 上电复位/掉电复位波形



1. 有关阈值和迟滞值的信息，请参见数据手册。

9.4.2 欠压复位 (BOR)

上电期间，欠压复位 (BOR) 将使系统保持复位状态，直到 V_{DD} 电源电压达到指定的 V_{BOR} 阈值。

V_{BOR} 阈值通过系统 OTP 位进行配置。BOR 默认为关闭。可以编程多个可编程的 V_{BOR} 阈值。

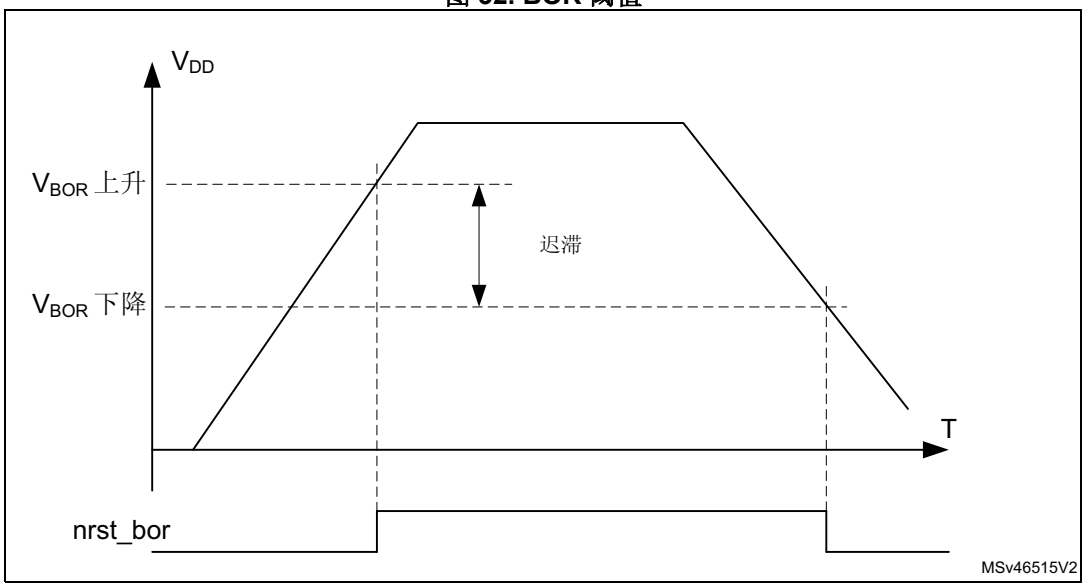
BOR 会产生一个 NRST 复位和一个应用复位 (app_rst)。

当 BOR 使能并且 V_{DD} 电源电压降至低于所选 V_{BOR} 阈值时，会产生一个应用复位 (app_rst)。

通过对系统 OTP 位编程可禁止 BOR。这样掉电就将由 PDR 监视（参见*第 9.4.1 节*）

有关 BOR 阈值和迟滞电压的更多信息，请参见数据手册。

图 32. BOR 阈值



1. 有关阈值和迟滞值的信息，请参见数据手册。

9.4.3 V_{DDCORE} OK 和 V_{DDCPU} OK 复位

系统具有集成电路，可实现 V_{DDCORE} 域的正常启动操作。

当 V_{DDCORE} 低于工作阈值 vddcore_ok 时，V_{DDCORE} 域保持在复位模式。一旦 V_{DDCORE} 电源电压高于工作阈值 vddcore_ok，V_{DDCORE} 域便会退出复位状态。有关 vddcore_ok 复位阈值的更多详细信息，请参见数据手册的电气特性部分。

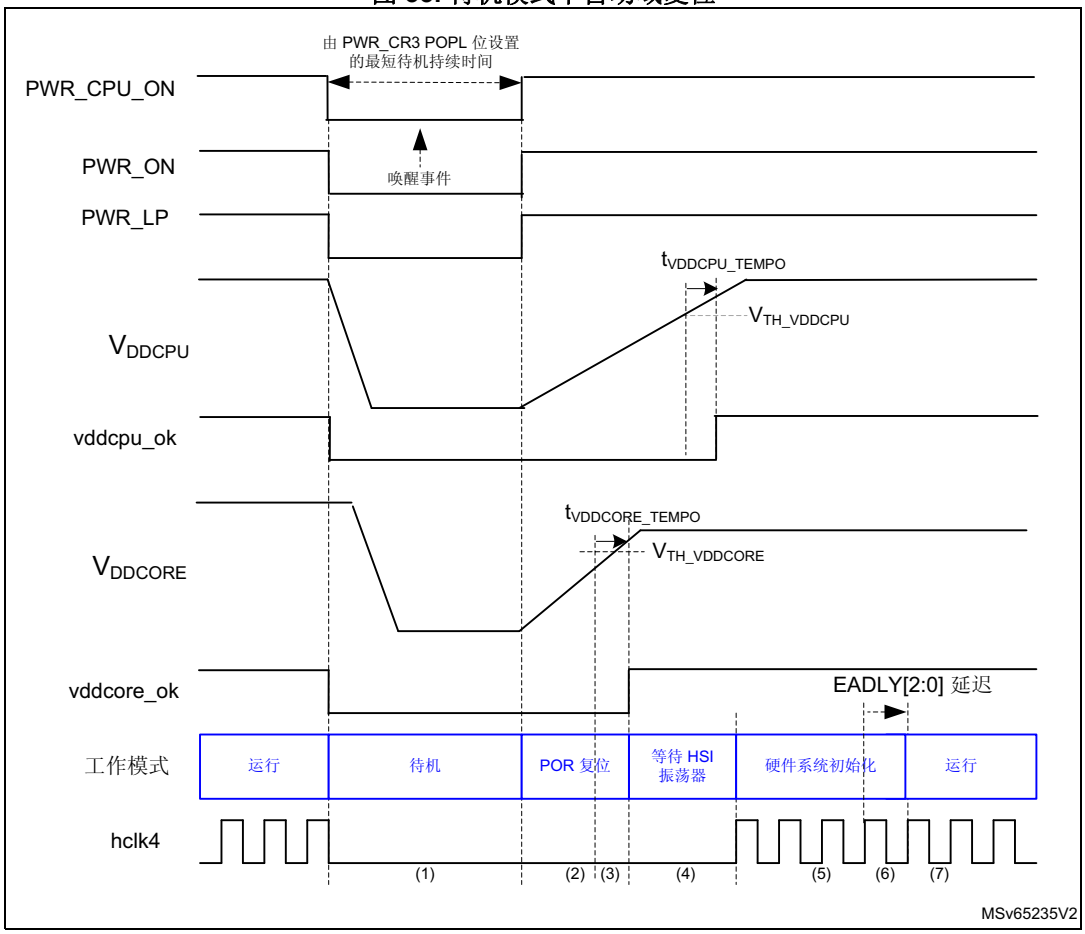
当 V_{DDCPU} 低于工作阈值 vddcpu_ok 时，系统便无法进入 RUN 模式。

vddcore_ok 和 vddcpu_ok 检测器可以通过设备输入引脚 PDR_ON 使能/禁止。

当检测器使能时，如果 V_{DDCORE} 电源电压低于工作阈值，在待机模式下 V_{DDCORE} 会被自动置于复位状态。仅当 V_{DDCORE} 和 V_{DDCPU} 的电源电压高出它们的工作阈值，才会进入运行模式。请参见*图 33*。有关退出序列的详细说明，请参见*第 9.6.11 节：系统待机模式*。

使用低功耗模式 LPLV-Stop 和 LPLV-Stop2 时，应保持 PDR_ON 高电平。

图 33. 待机模式下自动域复位



9.4.4 可编程电压检测器 (PVD)

可以使用 PVD 监视 V_{DD} 电源，方法是将其与 *PWR 控制寄存器 1 (PWR_CR1)* 中的 PLS[2:0] 位所选的阈值进行比较。

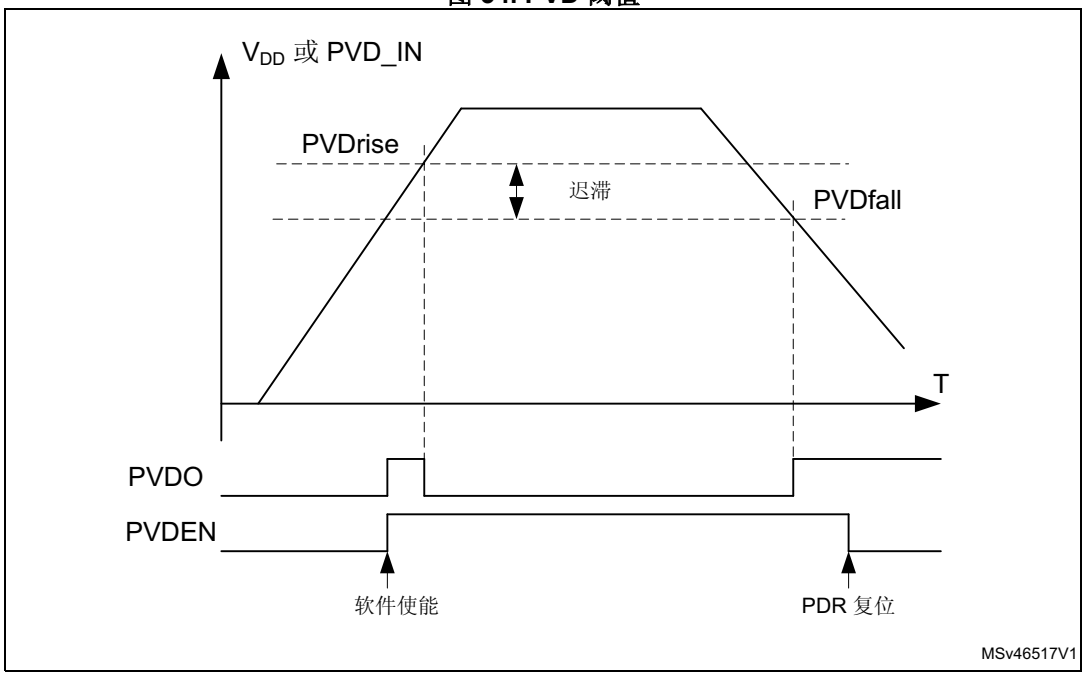
也可以使用 PVD 来监视 PVD_IN 引脚上的电压。在这种情况下，会将 PVD_IN 电压与内部 VREFINT 值进行比较。

通过将 *PWR 控制寄存器 1 (PWR_CR1)* 中的 PVDEN 位置 1 来使能 PVD。

在待机模式下 PVD 不可用。

PWR 控制状态寄存器 1 (PWR_CSR1) 中提供了 PVDO 标志，用于指示 V_{DD} 或 PVD_IN 电压是高于还是低于 PVD 阈值。此事件内部连接到 EXTI，如果通过 EXTI 寄存器使能了中断，则可以产生中断。当 V_{DD} 或 PVD_IN 电压降至 PVD 阈值以下以及/或者当 V_{DD} 或 PVD_IN 电压升至 PVD 阈值以上时，可以产生 PVDO 输出中断，具体取决于 EXTI 上升沿/下降沿的配置。该功能的用处之一就是可以在中断服务程序中执行紧急关闭系统的任务。

图 34. PVD 阈值



1. 有关阈值和迟滞值的信息，请参见数据手册。

9.4.5 模拟电压检测器 (AVD)

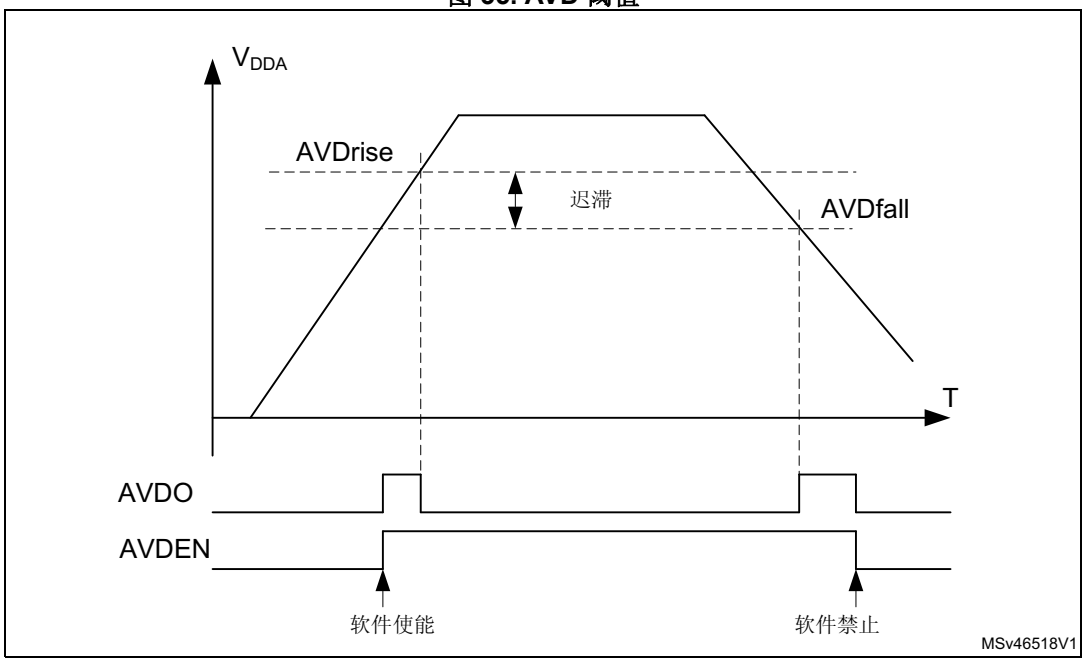
可以使用 AVD 监视 V_{DDA} 电源，方法是将其与 *PWR 控制寄存器 1 (PWR_CR1)* 中的 ALS[1:0] 位所选的阈值进行比较。

通过将 *PWR 控制寄存器 1 (PWR_CR1)* 中的 AVDEN 位置 1 来使能 AVD。

在待机模式下 AVD 不可用。

PWR 控制状态寄存器 1 (PWR_CSR1) 中提供了 AVDO 标志，用于指示 V_{DDA} 是高于还是低于 AVD 阈值。此事件内部连接到 EXTI，如果通过 EXTI 寄存器使能了中断，则可以产生中断。当 V_{DDA} 降至 AVD 阈值以下以及/或者当 V_{DDA} 升至 AVD 阈值以上时，可以产生 AVDO 中断，具体取决于 EXTI 上升沿/下降沿配置。例如，服务程序可以指示 V_{DDA} 电源何时降至最小值以下。

图 35. AVD 阈值



1. 有关阈值和迟滞值的信息，请参见数据手册。

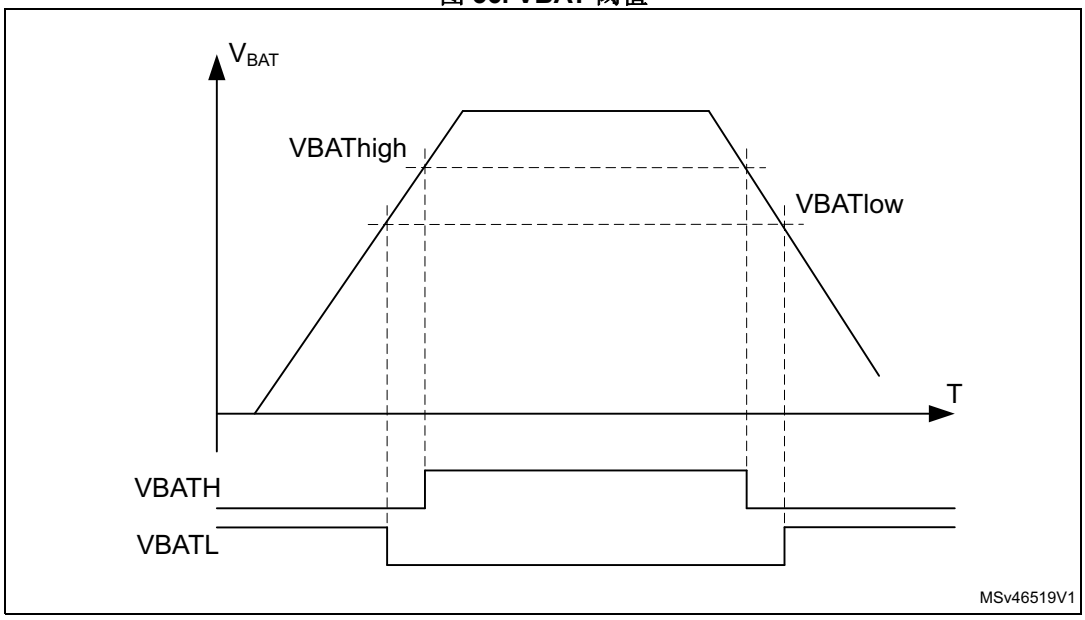
9.4.6 电池电压阈值

电池电压 V_{BAT} 可以通过将其与两个阈值进行比较来监视。*PWR 控制寄存器 2 (PWR_CR2)* 中提供的 VBATH 和 VBATL 标志用于指示 V_{BAT} 是高于还是低于相应阈值。可通过 *PWR 控制寄存器 2 (PWR_CR2)* 中的 MONEN 位使能/禁止 VBAT 电源监视。使能时，电池电压阈值会增加功耗。例如，电压值可用来触发执行紧急保存任务的程序。

如果使能电池电压阈值，则在待机模式和 VBAT 模式下也可使用这些阈值。

VBATL 和 VBATH 标志进行逻辑或 (OR)，并且在 Tamper 信号上可用（参见*第 46 章：实时时钟 (RTC)*）。

图 36. VBAT 阈值



1. 有关阈值和迟滞值的信息，请参见数据手册。

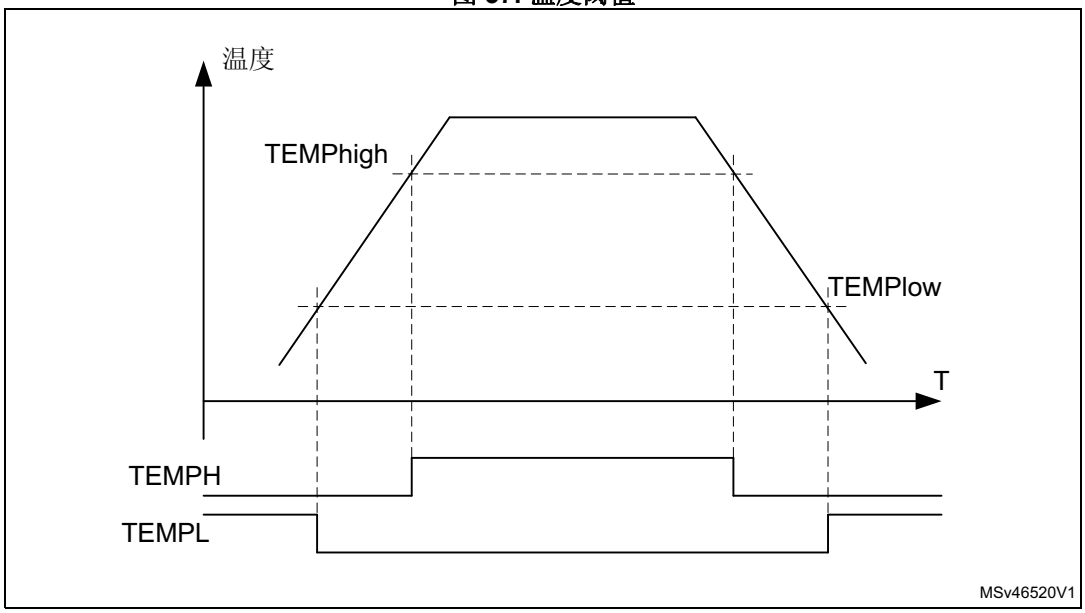
9.4.7 温度阈值

可以监视温度，方法是将其与两个阈值进行比较。*PWR 控制寄存器 2 (PWR_CR2)* 中提供的 TEMPH 标志和 TEMPL 标志用于指示设备温度是高于还是低于阈值。可通过 *PWR 控制寄存器 2 (PWR_CR2)* 中的 MONEN 位使能/禁止温度监视。当使能时，温度阈值会增加功耗。例如，温度值可用来触发执行温度控制任务的程序。

如果使能温度阈值，则在待机模式和 VBAT 模式下也可使用这些阈值。

TEMPL 和 TEMPH 标志进行逻辑或 (OR)，并且在 Tamper 信号上可用（参见*第 46 节：实时时钟 (RTC)*）。

图 37. 温度阈值



1. 有关阈值和迟滞值的信息，请参见数据手册。

9.5 PWR 电源管理

电源管理模块根据系统工作模式来控制 V_{DDCORE} 和 V_{DDCPU} 电源（请参见*第 9.5.1 节：工作模式*）。

9.5.1 工作模式

提供了多种系统工作模式，从而可以根据所需性能（比如当 MPU 不需要执行代码并且正在等待外部事件时）对系统进行调整。由用户选择具体的工作模式，以在低功耗、短启动时间和可用唤醒源之间寻求最佳平衡。

工作模式控制功耗以及时钟如何分配给系统的不同部分。针对系统的不同部分定义了以下工作模式：

- MPU 子系统模式：
  - **CRun**
    将为根据 RCC PERxEN 寄存器配置为 MPU 和外设提供时钟。V_{DDCORE} 和 V_{DDCPU} 处于开启状态。
  - **CSleep**：
    MPU 时钟停顿，外设时钟根据 RCC PERxLEN 状态决定。当 MPU 通过 WFE 进入 CSleep 模式时，外设时钟会像在 MPU CRun 模式下一样工作，与 RCC PERxLPEN 无关。V_{DDCORE} 和 V_{DDCPU} 处于开启状态。
  - **CStop**：
    MPU 和外设时钟停顿。系统处于 Stop、LP-Stop 或 LPLV-Stop 模式。V_{DDCORE} 和 V_{DDCPU} 处于开启状态。
  - **CStandby**
    MPU 和外设时钟停顿，系统处于 LPLV-Stop2 或待机模式，V_{DDCORE} 处于开启状态，V_{DDCPU} 处于关闭状态。
- 系统模式
  - **运行**
    各子系统时钟根据 RCC 配置运行。
    MPU 子系统处于 CRun 或 CSleep 状态。
    在主模式下，稳压器为 V_{DDCORE} 和 V_{DDCPU} 域提供全功率，PWR_ON 和 PWR_CPU_ON 均设为高电平，PWR_LP 也设为高电平。
  - **Stop**
    PWR 控制寄存器 1 (PWR_CR1) 中的 LPDS 位选择 Stop。
    各子系统时钟已停顿。
    MPU 子系统处于 CStop 状态，并且 PDDS[a] 位选择 Stop。
    稳压器为 V_{DDCORE} 和 V_{DDCPU} 域供电，以保存寄存器和内部内存的内容。外部稳压器保持在主模式，以便快速退出。通过 *PWR 控制寄存器 1 (PWR_CR1)* 中的 LPDS 寄存器位禁止低功耗模式信号。PWR_ON 和 PWR_CPU_ON 均保持高电平设置，PWR_LP 保持高电平。

a. *PWR MPU 控制寄存器 (PWR_MPUCR)* 中的 PPDS 位。

– **LP-Stop**

PWR 控制寄存器 1 (PWR_CR1) 中的 LPDS 位选择 LP-Stop。*PWR 控制寄存器 1 (PWR_CR1)* 中的 LVDS 位选择正常电压。
各子系统时钟已停顿。
MPU 子系统处于 CStop 状态，并且 PDDS(a) 位选择 Stop。

稳压器为 V_{DDCORE} 和 V_{DDCPU} 域供电，以保存寄存器和内部内存的内容。外部稳压器可进入低功耗模式。PWR_LP 上的低功耗模式信号通过 *PWR 控制寄存器 1 (PWR_CR1)* 中的 LPDS 寄存器位使能。根据 *PWR 控制寄存器 1 (PWR_CR1)* 中 LPCFG 的选择，在 PWR_LP 翻转后，PWR_ON 保持高电平或设为低电平，PWR_CPU_ON 保持高电平，PWR_LP 为低电平。

– **LPLV-Stop**

PWR 控制寄存器 1 (PWR_CR1) 中的 LPDS 和 LVDS 位选择 LPLV-Stop。

各子系统时钟已停顿。

MPU 子系统处于 CStop 状态，并且 PDDS(a) 位选择 Stop。

稳压器为 V_{DDCORE} 和 V_{DDCPU} 域供电，以保存寄存器和内部内存的内容。外部稳压器可进入低功耗低电压模式。PWR_LP 上的低功耗模式信号通过 *PWR 控制寄存器 1 (PWR_CR1)* 中的 LPDS 寄存器位使能。根据 *PWR 控制寄存器 1 (PWR_CR1)* 中 LPCFG 的选择，在 PWR_LP 翻转后，PWR_ON 保持高电平或设为低电平，PWR_CPU_ON 保持高电平，PWR_LP 为低电平。低功耗低电压模式是通过 *PWR 控制寄存器 1 (PWR_CR1)* 中的 LVDS 位使能的。在 LPLV-Stop 模式期间，V_{DDCORE} 和 V_{DDCPU} 的电源电压可能会降低。

– **LPLV-Stop2**

PWR 控制寄存器 1 (PWR_CR1) 中的 LPDS 和 LVDS 位选择 LPLV-Stop。
MPU 电源 V_{DDCPU} 已掉电。
MPU 子系统处于 CStandby 状态，STOP2 = 1，PDDS(a) 位选择 Stop。

稳压器为 V_{DDCORE} 域供电，以保存寄存器和内部内存的内容（V_{DDCPU} 域中的除外），关闭 V_{DDCPU} 域。V_{DDCORE} 外部稳压器可进入低功耗低电压模式，而 V_{DDCPU} 外部稳压器则被关闭。PWR_LP 上的低功耗模式信号通过 *PWR 控制寄存器 1 (PWR_CR1)* 中的 LPDS 寄存器位使能。根据 *PWR 控制寄存器 1 (PWR_CR1)* 中 LPCFG 的选择，在 PWR_LP 翻转后，PWR_ON 保持高电平或设为低电平，PWR_CPU_ON 设为低电平，PWR_LP 设为低电平。低功耗低电压模式是通过 *PWR 控制寄存器 1 (PWR_CR1)* 中的 LVDS 位使能的。在 LPLV-Stop2 模式期间，V_{DDCORE} 电源电压降低，V_{DDCPU} 电源被关闭。

– **待机**

系统掉电。
MPU 子系统处于 CStandby 状态且 STOP2 = x，PDDS(a) 位选择待机。

PWR_ON、PWR_CPU_ON 信号设为低电平，PWR_LP 设为低电平，外部稳压器被关闭，V_{DDCORE} 和 V_{DDCPU} 域掉电。除备份域和待机电路外，寄存器和内存的内容都将丢失。

在运行模式下，可以通过以下方法降低功耗：

- 不使用 APBx 和 AHBx 外设时，使用 PERxEN 将对应的外设时钟关闭（参见*第 10 章：复位和时钟控制 (RCC)*）。

MPU 超频运行模式

通过将 V_{DDCPU} 升至常规模式下允许的最大电压以上，可将 MPU 设置为超频运行模式，从而允许以更高频率工作（有关过载运行模式下的最大 MPU 频率的信息，请参见数据手册）。

当需要在超频模式下运行时，在 V_{DDCPU} 电源达到超频模式电压范围并在将 MPU 频率提高到超频频率范围之前，必须将 PWR_CR1 寄存器中的 MPU_RAM_LOWSPEED 位复位。

在将 MPU 频率降低到标准频率范围并在将 MPU 电压降低到超频模式电压范围以下之前，必须将其置 1。

警告： **当 V_{DDCPU} < 最小 VDDCPU（超频运行模式）且 MPU_RAM_LOWSPEED = 0 时，运行功能无法得到保证。**

MPU 低功耗模式汇总

MPU 子系统支持以下低功耗模式：

表 32. MPU 低功耗模式汇总

<table>
<tr><th>MPU</th><th>L2 缓存</th><th>入口(1)</th><th>唤醒</th><th>MPU 时钟</th><th>总线矩阵时钟</th></tr>
<tr><td>CRun</td><td rowspan="2">L2Run</td><td>-</td><td rowspan="6">请参见表 35</td><td>开启</td><td rowspan="2">开启</td></tr>
<tr><td rowspan="2">CSleep</td><td>STPREQ_P0 = 0
WFE</td><td rowspan="5">关闭</td></tr>
<tr><td rowspan="4">L2Stop</td><td>STPREQ_P0 = 0
WFI</td><td>开启 / 关闭(2)</td></tr>
<tr><td>CStop</td><td>STPREQ_P0 = 1
PDDS = 0
STOP2 = 0
LPDS = 0 或 1
WFI</td><td rowspan="3">关闭 / 开启</td></tr>
<tr><td rowspan="2">CStandby</td><td>STPREQ_P0= 1
PDDS = 0
STOP2 = 1
LPDS = 1
WFI</td></tr>
<tr><td>STPREQ_P0 = 1
PDDS = 1
STOP2 = x
WFI</td></tr>
</table>

1. 有关 STPREQ_P0 寄存器位功能的信息，请参见 RCC（来自 MPU 处理器的停止请求）。
2. 具有 PERxLPEN 位的外设相应地运行。

系统低功耗模式汇总

系统低功耗模式来自 MPU 低功耗模式。

表 33. PWR 系统模式汇总

MPU CRun	**MPU CSleep** STPREQ_P0 = 0	**MPU CStop** STPREQ_P0 = 1 STOP2 = 0 PDDS = 0	**MPU CStandby** STPREQ_P0 = 1 STOP2 = 1 PDDS = 0	**MPU CStandby** STPREQ_P0 = 1 STOP2 = x PDDS = 1
运行	运行	Stop LP-Stop LPLV-Stop	LPLV-Stop2	待机
时钟 MPU 开启 电源 MPU 开启 电源内核开启	时钟 MPU 关闭(1) 电源 MPU 开启 电源内核开启	时钟 MPU 关闭 电源 MPU 开启(2) 电源内核开启	时钟 MPU 关闭 电源 MPU 关闭 电源内核开启(3)	时钟 MPU 关闭 电源 MPU 关闭 电源内核关闭

1. 矩阵总线时钟开或关
2. V_{DDCORE} 和 V_{DDCPU} 在 LPLV-Stop 下可降低
3. V_{DDCORE} 在 LPLV-Stop2 下降低

表 34. 系统低功耗模式汇总

系统	**MPU**	**DDR(1)**	**唤醒**	**系统振荡器(2)**	**hclk4**	**PWR_LP**	**PWR_ON**		**PWR_CPU_ON**
							LPCFG = 0	**LPCFG = 1**	
运行	CRun 或 CSleep	主动/自动刷新	请参见*表 35*	开启	开启	1	1	1	1
(3) Stop (LPDS = 0)	CStop (STPREQ_P0 = 1, STOP2 = 0, PDDS = 0)	自刷新		开启/关闭(4)	关闭				
(3) LP-Stop, LPLV-Stop (LPDS = 1)						0(5)		0 (5)	
LPLV-Stop2 (LPDS = 1)	CStandby (STPREQ_P0 = 1, STOP2 = 1, PDDS = 0)	关闭/自刷新		关闭					0
待机 (LPDS = x)	CStandby (STPREQ_P0 = 1, STOP2 = x, PDDS = 1)	关闭/自刷新				0(6)	0(6)	0(6)	0

1. DDR 工作状态仅供参考。例如，如果 MPU 处于 CRun 状态，则当从 Boot ROM 执行时，DDR 可能关闭，或者当系统处于任意工作模式时，DDR 可能处于自刷新状态。
2. 系统振荡器：RCC 中配置为开启的 HSI/HSE/CSI 振荡器。
3. PDDS 位选择 Stop 模式（*PWR MPU 控制寄存器 (PWR_MPUCR)* 寄存器中的 PPDS 位）。
4. 当使用系统振荡器 HSI 或 CSI 时，状态通过相应的 xxxKERON 控制，否则系统振荡器关闭（参见*第 10 章：复位和时钟控制 (RCC)*）。
5. *PWR 控制寄存器 3 (PWR_CR3)* 寄存器位 POPL 的设置对 LP-Stop 和 LPLV-Stop 模式 PWR_ON、PWR_CPU_ON 和 PWR_LP 脉冲时间没有影响。
6. 可以在 *PWR 控制寄存器 3 (PWR_CR3)* 的位 POPL 中定义一个最小 PWR_ON、PWR_CPU_ON 和 PWR_LP 脉冲时间。

表 35. 功能取决于系统工作模式(1)

外设	运行	Stop 和 LP-Stop		LPLV-Stop LPLV-Stop2		待机		VBAT
		-	唤醒能力	-	唤醒能力	-	唤醒能力	
MPU	Y(2)	R	-	R(3)	-	-	-	-
DBG	O	O	O	R	-	-	-	-
ROM 内存	Y	R	-	R	-	R	-	R
SYSRAM	Y(2)	R	-	R	-	-	-	-
SRAMx (x = 1:3)	Y(4)	R	-	R	-	-	-	-
BKPSRAM	O(2)	R	-	R	-	O(5)	-	R
DDRCTRL	O	R	-	R	-	-	-	-
QUADSPI	O	R	-	R	-	-	-	-
FMC	O	R	-	R	-	-	-	-
备份寄存器	Y	R	-	R	-	R	-	R
欠压复位 (BOR)	Y	Y	Y	Y	Y	Y	Y	-
可编程电压检测器 (PVD)	O	O	O	O	O	-	-	-
模拟电压监视器 (AVD)	O	O	O	O	O	-	-	-
VBATH/VBATL 监视	O	O	O	O	O	O	O	O
TEMPH/TEMPL 监视	O	O	O	O	O	O	O	O
MDMA	O	R	-	R	-	-	-	-
DMAx (x = 1,2,3)	O	R	-	R	-	-	-	-
DMAMUX	O	R	-	R	-	-	-	-
高速内部时钟 (HSI)	O	O	-	O	-	-	-	-
高速外部时钟 (HSE)	O	O	-	O	-	-	-	-
低速内部时钟 (LSI)	O	O	-	O	-	O	-	-
低速外部时钟 (LSE)	O	O	-	O	-	O	-	O
多速内部时钟 (CSI)	O	O	-	-	-	-	-	-
HSE CSS 时钟安全系统	O	O	O	-	-	-	-	-
LSE CSS 时钟安全系统	O	O	O	O	O	O	O	O
RTC/自动唤醒	O	O	O	O	O	O	O	O
Tamper 引脚的数量	12	12	O	12	O	12	O	12
USBH、OTG	O	R	O	R	-	-	-	-
SDMMCx (x=1:2)	O	R	-	R	-	-	-	-

表 35. 功能取决于系统工作模式(1)（续）

外设	运行	Stop 和 LP-Stop		LPLV-Stop LPLV-Stop2		待机		VBAT
		-	唤醒能力	-	唤醒能力	-	唤醒能力	
FDCANx (x=1:2)	O	R	-	R	-	-	-	-
ETHx (x= 1:2)	O	R	O	R	-	-	-	-
USARTx (x = 1:8)	O	O(6)	O(6)	R	O(6)	-	-	-
I2Cx (x = 1:5)	O	O(7)	O(7)	R	O(7)	-	-	-
SPIx (x=1:5)	O	O(8)	O(8)	R	O(8)	-	-	-
SAIx (x = 1:2)	O	R	-	R	-	-	-	-
DFSDM	O	R	-	R	-	-	-	-
DCMIPP	O	R	-	R	-	-	-	-
HDP	O	R	-	R	-	-	-	-
LTDC	O	R	-	R	-	-	-	-
SPDIFRX	O	R	-	R	-	-	-	-
ADC1-2	O	R	-	R	-	-	-	-
VREFBUF	O	O	-	-	-	-	-	-
DTS	O	R	O	R	O	-	-	-
STGEN	O	R	-	R	-	-	-	-
TIMx (x = 1:17)	O	R	-	R	-	-	-	-
LPTIMx (x = 1:5)	O	O	O	R	O(9)	-	-	-
IWDGx (x=1,2)	O	O	O	O	O	O	O	-
SysTick 定时器	O	R	-	R(3)	-	-	-	-
RNGx（x = 1）	O	R	-	R	-	-	-	-
CRYPx(10) (x = 1)	O	R	-	R	-	-	-	-
HASHx (x = 1)	O	R	-	R	-	-	-	-
CRCx (x = 1)	O	R	-	R	-	-	-	-
GPIO	O	R	O(11)	R	O(11)	-	6 个引脚 (12)	-

1. 图注：Y = 支持（使能）。O = 可选（默认情况下禁止。可通过软件使能）。R = 保留数据/状态。- = 不可用；唤醒模式以灰色高亮显示。
2. SRAM 时钟可门控打开或关闭。当 MPU 处于 CSleep 模式时，不会为 MPU 提供时钟，默认情况下会禁止 SRAM，但可以通过软件使能。
3. 不处于 LPLV-Stop2 模式。

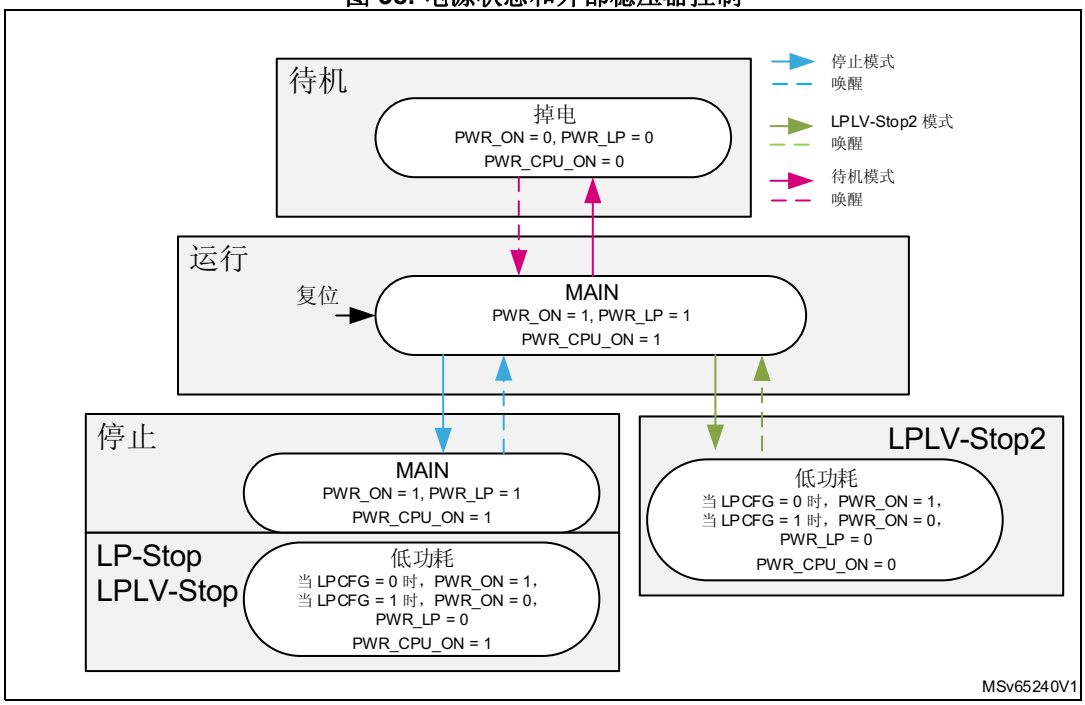
4. 当 MPU 处于 CSleep 模式时，可门控开启或关闭 SRAM 时钟。默认情况下禁止，但可由软件使能。
5. 如果使能，BKPSRAM 内容可以保留（可选）。
6. UART 接收在 Stop 模式下有效，并能在开始、地址匹配或收到帧事件时产生一个唤醒中断。
7. I2C 地址检测在 Stop 模式下有效，并能在地址匹配时产生一个唤醒中断。
8. SPI 发送和接收在 Stop 模式下有效，并且在达到 FIFO 阈值、传输结束、出错等情况下产生一个唤醒中断。
9. 仅 LSE 或 LSI 作为内核时钟源。
10. 并非所有产品编号的部件上都提供，更多信息请参见相关数据手册。
11. 所有 GPIO 均可通过 EXTI 接口唤醒系统。
12. 具有从待机模式唤醒能力的 I/O 仅在有限数量的引脚（PA3、PC13、PI1、PI2、PI3 和 PF8）上可用。

9.5.2 系统工作模式和外部稳压器模式

电源控制处理 V_{DDCORE} 和 V_{DDCPU} 电源电压，用于系统 Run、Stop、LP-Stop、LPLV-Stop、LPLV-Stop2 和待机模式。系统的工作模式取决于 MPU 子系统模式（CRun、CSleep、CStop、CStandby）。

- 在运行模式下，提供 V_{DDCORE} 和 V_{DDCPU} 电源。
- 在 Stop 和 LP-Stop 模式下，提供 V_{DDCORE} 和 V_{DDCPU} 电源。
- 在 LPLV-Stop 模式下，可提供较低电压的 V_{DDCORE} 和 V_{DDCPU} 电源。（两种供电的电压相同）
- 在 LPLV-Stop2 模式下，V_{DDCPU} 可能被关闭，而 V_{DDCORE} 可能会像在 LPLV-Stop 模式下那样以低电压提供电源。
- 在 Standby 模式下，V_{DDCORE} 和 V_{DDCPU} 电源被关闭。

图 38. 电源状态和外部稳压器控制



在 Stop、LP-Stop、LPLV-Stop 或 LPLV-Stop2 或待机之间的系统模式选择由 *PWR MPU 控制寄存器 (PWR_MPUCR)* 中的 PDDS 位和 STOP2 位控制。

表 36. PDDS 低功耗模式控制

STOP2	PDDS	系统工作模式
0	0	Stop、LP-Stop、LPLV-Stop（MPU 处于 Cstop 模式）
1	0	LPLV-Stop2（MPU 处于 Cstandby 模式）
x	1	Standby（MPU 处于 Cstandby 模式）

退出低功耗模式时的标志设置：

- EXTI 唤醒事件。
  - *图 38* 蓝色虚线跳变：系统从 Stop、LP-Stop 或 LPLV-Stop 唤醒，MPU 从 CStop 唤醒。STOPF 标志置 1。
  - *图 38* 绿色虚线跳变：系统从 LPLV-Stop2 唤醒，MPU 从 CStandby 唤醒（STOP2 = 1，PDDS = 0）。STOPF 和 SBFMPU 标志置 1。
- 唤醒事件
  - *图 38* 红色虚线跳变：系统从 Standby 唤醒，MPU 从 CStandby 唤醒（STOP2 = x，PDDS = 1）。SBF 和 SBFMPU 标志置 1。

表 35 列出了提供的用于确定系统从哪种模式退出的标志。MPU 标志可以从 *PWR MPU 控制寄存器 (PWR_MPUCR)* 读取。MPU 有一个复位位 CSSF，可用来清除其唤醒标志。MPU CSSF 寄存器位与 MPU 标志位于相同的寄存器中。

注： *从选择了 PDDS 寄存器位并且 MPU 发出 WFI 或 WFE 指令以请求系统待机模式开始，到实际进入待机模式 (PWR_ON = 0)，大约需要 50 个 HSI 时钟周期。如果在此期间发生唤醒事件，则在退出待机模式时 SBF 和 SBFMPU 标志不会置 1。*

表 37. 低功耗模式退出标志

系统模式	MPU 模式	SBF	SBFMPU	STOPF	备注
运行	CRun/CSleep	0	0	0	系统内容保留
Stop LP-Stop LPLV-Stop	CStop	0	0	1	系统内容保留，系统时钟停止。
LPLV-Stop2	CStandby	0	1	1	MPU 子系统内容丢失
待机	CStandby	1	1	0	系统内容丢失
VBAT(1)	n.a.	0	0	0	系统内容丢失

1. 从 VBAT 模式退出可能源于 VSW 域中保留的寄存器未复位。

9.5.3 系统唤醒

系统将通过应用复位（NRST、IWDG 等）或从待机模式唤醒来唤醒 MPU。从此时开始，由 Boot ROM 来管理启动。

9.6 PWR 低功耗工作模式

系统提供了多个低功耗工作模式，可在 MPU 不需要执行代码时（比如等待外部事件时）节省功耗。由用户根据应用选择具体的低功耗模式，以在低功耗、短启动时间和可用唤醒源之间寻求最佳平衡。

- 降低 CPU 时钟和/或总线矩阵时钟速度（参见*第 10 章：复位和时钟控制 (RCC)*）
- 控制各个外设时钟（请参见*第 10 章：复位和时钟控制 (RCC)*）
- 低功耗模式
  - CSleep（MPU 时钟停止）
  - CStop（MPU 子系统时钟停止）
  - Stop（总线矩阵时钟停顿，振荡器可停止，有关 Stop 模式下振荡器状态的详细信息，请参见*第 10.4.2 节：振荡器说明*和*第 10.4.7 节：Stop 模式和待机模式下的时钟生成*，向外部稳压器发出正常模式信号）。
  - LP-Stop 和 LPLV-Stop（总线矩阵时钟停顿，振荡器可被停止，有关 Stop 模式下振荡器状态的详细信息，请参见*第 10.4.2 节：振荡器说明*和*第 10.4.7 节：Stop 模式和待机模式下的时钟生成*，向外部稳压器发出低功耗模式信号）。
  - LPLV-Stop2（MPU 子系统掉电）
  - 待机（系统掉电，V_{DD} 开启）
  - VBAT（系统掉电，V_{DD} 关闭且 VSW 域由 VBAT 引脚供电）

9.6.1 降低 CPU 时钟和/或总线矩阵时钟速度

在运行模式下，可以降低 MPU 和/或总线矩阵时钟的速度，有关详细信息，请参见*第 10 章：复位和时钟控制 (RCC)*。

9.6.2 控制外设时钟

在运行模式下，可以随时通过 RCC PERxEN 寄存器位停止各个外设的总线时钟，以降低功耗。

为了在 CSleep 模式下降低功耗，可以通过 RCC PERxLPEN 寄存器位禁止各个外设的时钟。对于在 CSleep 模式下仍然接收时钟的外设，可以在进入 CSleep 模式之前降低它们的时钟速度。

9.6.3 进入低功耗模式

MPU 子系统低功耗模式是 MPU 在执行 WFI（等待中断）指令（以进入 CSleep、CStop 和 CStandby 模式）或 WFE（等待事件）指令（以进入 CSleep 模式）时进入的。

当所有 EXTI 唤醒源被清除并且 MPU 处于 CStop 或 CStandby 时，系统可进入 Stop、LP-Stop、LPLV-Stop、LPLV-Stop2 或待机模式。

9.6.4 使用 WFI 或 WFE 退出低功耗模式

如果曾使用 WFI 指令进入低功耗模式，则通过 GIC 确认的任何外设中断都能唤醒系统。

如果使用 WFE 指令进入低功耗模式，MPU 将在有事件发生时立即退出低功耗模式。唤醒事件可通过以下方式产生：

- GIC IRQ 中断，通过在外设控制寄存器和 GIC 中使能中断。
  只有具备足够优先权的 GIC 中断才能唤醒和中断 MCU。
- 事件，通过将 EXTI 线配置为事件模式。当 MPU 从 WFE 恢复时，不必清零 EXTI 外设中断挂起位。可能需要清零相应外设中的中断标志。

当从 LPLV-Stop2 或待机模式唤醒时，GIC 会被复位，必须重新使能中断以便软件对其进行处理。

9.6.5 MPU CSleep 模式

在 CSleep 模式下，MPU 时钟停止。外设时钟根据 PERxLPEN 位工作。

进入 MPU CSleep 模式

应根据*第 9.6.3 节：进入低功耗模式*进入 CSleep 模式。

- 当 STPREQ_P0 寄存器位（参见*第 10 章：复位和时钟控制 (RCC)*）清零时。

有关如何进入 CSleep 模式的详细信息，请参见*表 38*。

退出 MPU CSleep 模式

应根据*第 9.6.4 节：使用 WFI 或 WFE 退出低功耗模式*退出 CSleep 模式。

有关如何退出 CSleep 模式的详细信息，请参见*表 38*。

表 38. MPU CSleep

MPU CSleep 模式	说明
进入模式	WFI（等待中断），且： – STPREQ_P0 = 0（参见*第 10 章：复位和时钟控制 (RCC)*）。 – MPU GIC 中断和事件清除。
	WFE（等待事件），并且： – MPU GIC 中断和事件清除。
退出模式	如果使用 WFI 进入： – GIC 中使能的任何中断：请参见*第 18.1 节：GIC 中断* 如果使用 WFE 进入： – 任何中断（即使在 GIC 中禁止，请参见*第 18.1 节：GIC 中断*）或任何事件（请参见*第 9.6.4 节：使用 WFI 或 WFE 退出低功耗模式*）
唤醒延迟	无

9.6.6 MPU CStop 模式

在 CStop 模式下，MPU 时钟和外设时钟停止。

在 CStop 模式下，具有内核时钟请求的外设仍能通过选择 RCC 中的时钟（能够在 CStop 模式下运行）来请求它们的内核时钟。

进入 MPU CStop 模式

应根据*第 9.6.3 节：进入低功耗模式*进入 CStop 模式。

- 当*第 10 章：复位和时钟控制 (RCC)* 中的 STPREQ_P0 寄存器位置 1
- 并且 *PWR 控制寄存器 1 (PWR_CR1)* 中的 STOP2 寄存器位清零
- 并且 *PWR MPU 控制寄存器 (PWR_MPUCR)* 中的 PDDS 寄存器位清零。

有关如何进入 CStop 模式的详细信息，请参见*表 39*。

退出 MPU CStop 模式

应根据*第 9.6.4 节：使用 WFI 或 WFE 退出低功耗模式*退出 CStop 模式。

有关如何退出 CStop 模式的详细信息，请参见*表 39*。

表 39. MPU CStop

MPU CStop 模式	说明
进入模式	WFI（等待中断） – STPREQ_P0 = 1 且 PDDS = 0 且 STOP2 = 0 – MPU GIC 中断和事件清除。 – 所有 MPU EXTI 唤醒源都将被清除。
退出模式	GIC 中使能的 EXTI 中断：有关未停止的外设，请参见*第 18.1 节：GIC 中断*。
唤醒延迟	EXTI 和 RCC 唤醒同步（请参见*第 10 章：复位和时钟控制 (RCC)*）。

9.6.7 系统 Stop 模式

当 MPU 处于 CStop 模式、EXTI 唤醒源无效并且 *PWR MPU 控制寄存器 (PWR_MPUCR)* 中的 MPU PDDS 位请求停止时，系统进入 Stop 模式。在 Stop 模式下，总线矩阵时钟停顿。振荡器可以停止，有关 Stop 模式下振荡器状态的详细信息，请参见*第 10.4.2 节：振荡器说明*和*第 10.4.7 节：Stop 模式和待机模式下的时钟生成*。有关其他时钟的更多信息，请参见*第 10 章：复位和时钟控制 (RCC)*。

在 Stop 模式下，使用 LSI 或 LSE 时钟的外设以及具有内核时钟请求并通过 RCC（能够在 Stop 模式下运行）选择时钟的外设，仍能运行。

在系统 Stop 模式下，可以通过对各控制位进行编程来选择使以下功能保持有效：

- 独立的看门狗 (IWDG)：IWDG 通过写入其密钥寄存器启动或使用硬件选项自动启动。一旦启动便无法停止，除非通过应用复位。请参见*第 45 章：独立看门狗 (IWDG)*。
- 实时时钟 (RTC)：通过 *RCC 低功耗 Stop 模式延迟控制寄存器 (RCC_PWRLPDLYCR)* 中的 RTCEN 位进行配置。
- 内部 RC 振荡器 (LSI RC)：通过 *RCC 复位持续时间和 LSI 控制寄存器 (RCC_RDLSICR)* 中的 LSION 位进行配置。
- 外部 32.768 kHz 振荡器 (LSE OSC)：通过 *RCC 低功耗 Stop 模式延迟控制寄存器 (RCC_PWRLPDLYCR)* 中的 LSEON 位进行配置。

- 能够通过 LSI 或 LSE 时钟运行的外设。
- 具有内核时钟请求的外设。
- 内部 RC 振荡器（HSI、CSI）和 HSE 振荡器：通过 *RCC 停止请求设置寄存器 (RCC_MP_SREQSETR)* 中相应的 xxxKERON 位配置。
- 在 Stop 模式下，ADC 也会产生功耗，除非在进入该模式前将其禁止。要禁止，必须向 ADC_CR2 寄存器中的 ADON 位写入 0。

进入系统 Stop 模式

当 *PWR MPU 控制寄存器 (PWR_MPUCR)* 中的 PDDS 位请求停止时，应根据*第 9.6.3 节：进入低功耗模式*进入 Stop 模式。

有关如何进入 Stop 模式的详细信息，请参见*表 40*。

如果正在访问总线矩阵，则会延迟到总线矩阵访问结束再进入 Stop 模式。

退出系统 Stop 模式

应根据*第 9.6.4 节：使用 WFI 或 WFE 退出低功耗模式*退出 Stop 模式。

有关如何退出 Stop 模式的详细信息，请参见*表 40*。

退出 Stop 模式时，RCC 恢复 MPU 和 AXI 总线矩阵时钟。请参见 RCC 一节中的*退出 Stop、LP-Stop、LPLV-Stop 或 LPLV-Stop2 模式*。

PWR MPU 控制寄存器 (PWR_MPUCR) 寄存器中的 STOPF 状态标志表明系统已退出 Stop 模式

表 40. 系统停止

Stop 模式	说明
进入模式	– MPU 处于 CStop 模式并且没有有效的 EXTI 唤醒源时。 – PDDS 位选择 Stop 并且 STOP2 = 0 – LPDS 选择 Stop
退出模式	– EXTI 唤醒时。
唤醒延迟	请参见*图 39：设备低功耗停止供电时序*。

系统 Stop 模式下的 I/O 状态

在 Stop 模式下，所有 I/O 引脚均保留其配置。

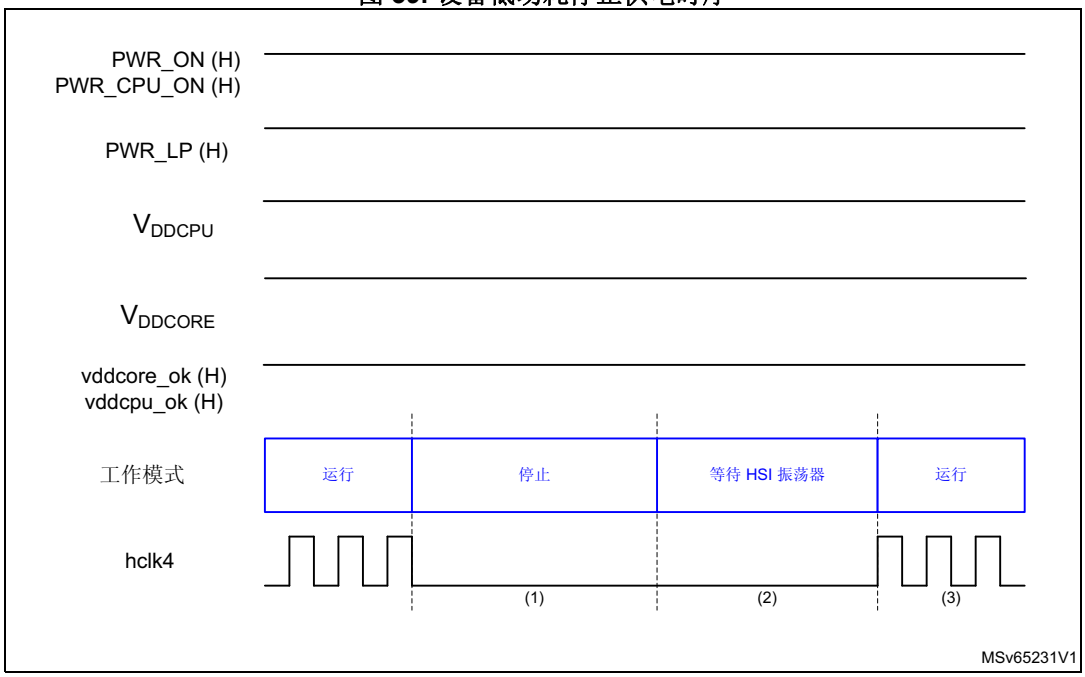
电源停止时序

图 39 所示为系统 Stop 模式时序。如果通过 *PWR 控制寄存器 1 (PWR_CR1)* 中的 LPDS 寄存器位进行了相应选择，将进入 Stop 模式。

1. 进入 Stop 模式时，hclk4 会停止。可以停止振荡器以降低功耗。有关 Stop 模式下振荡器状态的详细状态，请参见*第 10.4.2 节：振荡器说明*和*第 10.4.7 节：Stop 模式和待机模式下的时钟生成*。

   在 Stop 模式下，PWR_ON 和 PWR_CPU_ON 均保持高电平，PWR_LP 也保持高电平。外部稳压器保持主模式。
2. 在唤醒事件发生时，将执行时钟恢复过程。有关详细信息，请参见 RCC：*时钟恢复序列描述*。
3. 一旦振荡器稳定，系统便进入运行模式。

图 39. 设备低功耗停止供电时序



9.6.8 系统 LP-Stop 模式

LP-Stop 模式的处理方式与 Stop 模式的处理方式相同（当通过 *PWR 控制寄存器 1 (PWR_CR1)* 中的 LPDS 寄存器位选择了 LP-Stop 模式时）。请参见*第 9.6.7 节：系统 Stop 模式*。

有关如何进入和退出 LP-Stop 模式的详细信息，请参见*表 41*。

表 41. 系统 LP-Stop

LP-Stop 模式	说明
进入模式	– MPU 处于 CStop 模式并且没有有效的 EXTI 唤醒源时。 – PDDS 位选择 Stop 并且 STOP2 = 0 – LPDS 选择 LP-Stop
退出模式	– EXTI 唤醒时。
唤醒延迟	请参见*图 40：设备低功耗 LP-Stop 供电时序*。

PWR 控制寄存器 1 (PWR_CR1) 中的 LVDS 位选择在 LP-Stop 模式期间 V_{DDCORE} 的电源电压是保持在 Run 模式下的电压还是降低（低电压模式下的 LP-Stop）。

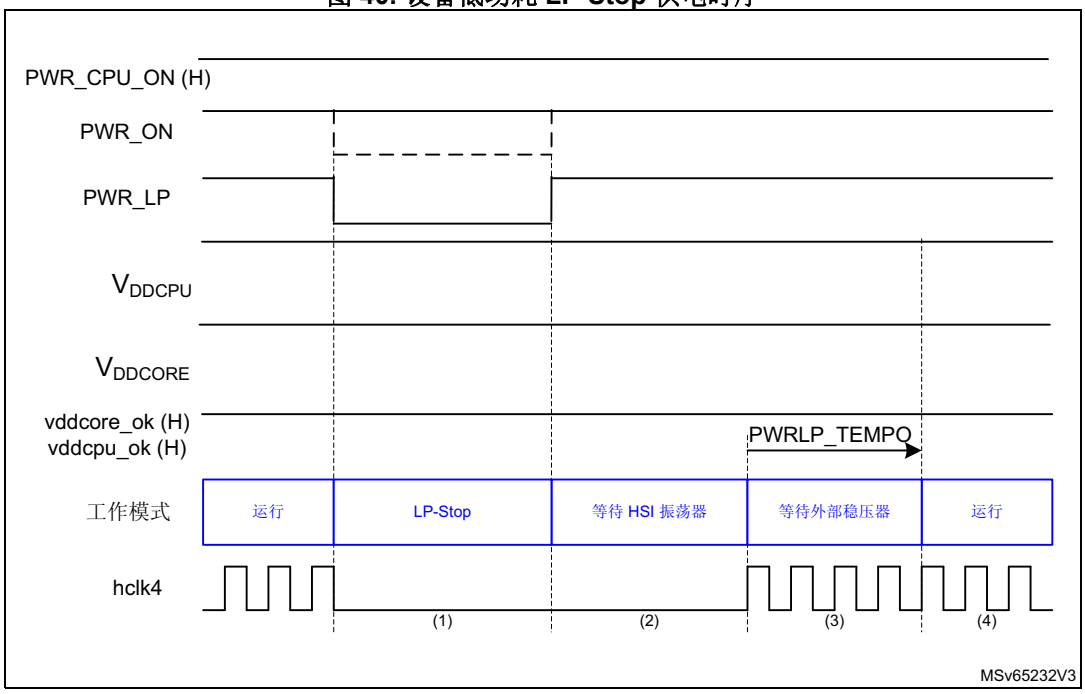
电源 LP-Stop 时序

图 40 所示为系统 LP-Stop 模式时序。如果通过 *PWR 控制寄存器 1 (PWR_CR1)* 中的 LPDS 寄存器位进行了相应选择，将进入 LP-Stop 模式。

1. 进入 LP-Stop 模式时，hclk4 会停止。可以停止振荡器以降低功耗。有关 Stop 模式下振荡器状态的详细状态，请参见*第 10.4.2 节：振荡器说明*和*第 10.4.7 节：Stop 模式和待机模式下的时钟生成*。

   在 LP-Stop 模式下，在 PWR_LP 翻转后，PWR_ON 保持高电平或清零为低电平，具体取决于 *PWR 控制寄存器 1 (PWR_CR1)* 中的 LPCFG 寄存器位。PWR_CPU_ON 保持高电平，PWR_LP 清零为低电平，表示处于 LP-Stop 模式。之后，外部稳压器可进入低功耗模式。
2. 在唤醒事件发生时，外部稳压器通过 PWR_LP 退出低功耗模式，并将执行时钟恢复过程。有关详细信息，请参见 RCC：*时钟恢复序列描述*。
3. 一旦振荡器稳定，PWRLP_TEMPO 延迟（可选）就会超时，以便让外部稳压器切换到主模式。默认情况下，复位后该延迟时间最短。（参见*第 10.8.6 节：RCC 低功耗 Stop 模式延迟控制寄存器 (RCC_PWRLPDLYCR)*）。
4. 延迟超时过后，系统进入运行模式。

图 40. 设备低功耗 LP-Stop 供电时序



9.6.9 系统 LPLV-Stop 模式

有关如何进入和退出 LPLV-Stop 模式的详细信息，请参见*表 42*。

表 42. 系统 LPLV-Stop

LPLV-Stop 模式	说明
进入模式	– MPU 处于 CStop 模式并且没有有效的 EXTI 唤醒源时。 – PDDS 位选择 Stop 并且 STOP2 = 0。 – LPDS 选择低功耗停止。 – LVDS 选择 LPLV-Stop。
退出模式	– EXTI 唤醒时。
唤醒延迟	请参见*图 41：设备低功耗 LPLV-Stop 供电时序*。

在 LPLV-Stop 模式下，DDR 保持必须通过 *PWR 控制寄存器 3 (PWR_CR3)* 中的 DDRRETEN 来使能。

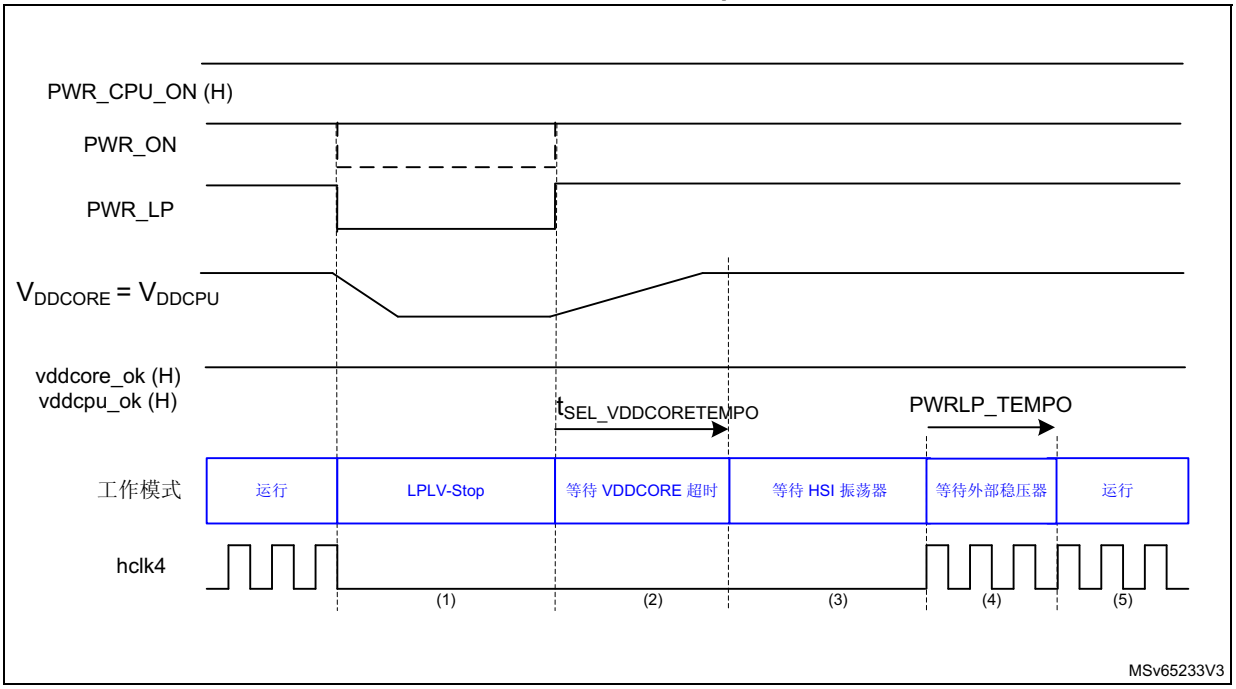
电源 LPLV-Stop 时序

图 41 所示为系统 LPLV-Stop 模式时序。如果通过 *PWR 控制寄存器 1 (PWR_CR1)* 中的 LPDS 和 LVDS 位进行了相应选择，将进入 LPLV-Stop 模式。

1. 进入 LPLV-Stop 模式时，hclk4 会停止。可以停止振荡器以降低功耗。有关 Stop 模式下振荡器状态的详细状态，请参见*第 10.4.2 节：振荡器说明*和*第 10.4.7 节：Stop 模式和待机模式下的时钟生成*。

   在 LPLV-Stop 模式下，在 PWR_LP 翻转后，PWR_ON 保持高电平或清零为低电平，具体取决于 *PWR 控制寄存器 1 (PWR_CR1)* 中的 LPCFG 寄存器位。PWR_CPU_ON 保持高电平，PWR_LP 清零为低电平，表示处于 LPLV-Stop 模式。之后，外部稳压器可进入低功耗低电压模式，并降低 V_{DDCORE} 和 V_{DDCPU} 电源电压。
2. 在唤醒事件发生时，外部稳压器通过 PWR_LP 上升退出低功耗低电压模式，并将开始一个 $t_{SEL_VDDCORETEMPO}$ 的延迟，以等待 V_{DDCORE} 和 V_{DDCPU} 达到运行模式工作电压。
3. 经过这段延迟之后，将使能 HSI 并执行时钟恢复过程。有关详细信息，请参见 RCC：*时钟恢复序列描述*。
4. 一旦振荡器稳定，PWRLP_TEMPO 延迟（可选）就会超时，以便让外部稳压器切换到主模式。默认情况下，复位后该延迟时间最短。（参见*第 10.8.6 节：RCC 低功耗 Stop 模式延迟控制寄存器 (RCC_PWRLPDLYCR)*）。
5. 延迟超时过后，系统进入运行模式。

图 41. 设备低功耗 LPLV-Stop 供电时序



9.6.10 系统 LPLV-Stop2 模式

LPLV-Stop2 模式的处理方式与 LPLV-Stop 模式相同。

有关如何进入和退出 LPLV-Stop2 模式的详细信息，请参见*表 43*。

表 43. 系统 LPLV-Stop2

LPLV-Stop2 模式	说明
进入模式	– MPU 处于 CStop 模式并且没有有效的 EXTI 唤醒源时。 – PDDS 位选择 Stop 并且 STOP2 = 1。 – LPDS 选择低功耗停止。 – LVDS 选择 LPLV-Stop。
退出模式	– EXTI 唤醒时。
唤醒延迟	请参见*图 42：设备 LPLV-Stop2 模式电源控制*。

在 LPLV-Stop2 模式下，DDR 保持必须通过 *PWR 控制寄存器 3 (PWR_CR3)* 中的 DDRRETEN 来使能。

从 LPLV-Stop2 唤醒会产生 MPU 复位。GIC 会复位，中断需要重新使能（如果软件需要处理该中断）

电源 LPLV-Stop2 时序

图 42 所示为系统 LPLV-Stop2 模式时序。如果通过 *PWR MPU 控制寄存器 (PWR_MPUCR)* 寄存器中的 PDDS 位和 *PWR 控制寄存器 1 (PWR_CR1)* 寄存器中的 STOP2 位进行了相应选择，将进入 LPLV-Stop2 模式。

1. 进入 LPLV-Stop2 模式时，hclk4 会停止。振荡器也会停止。

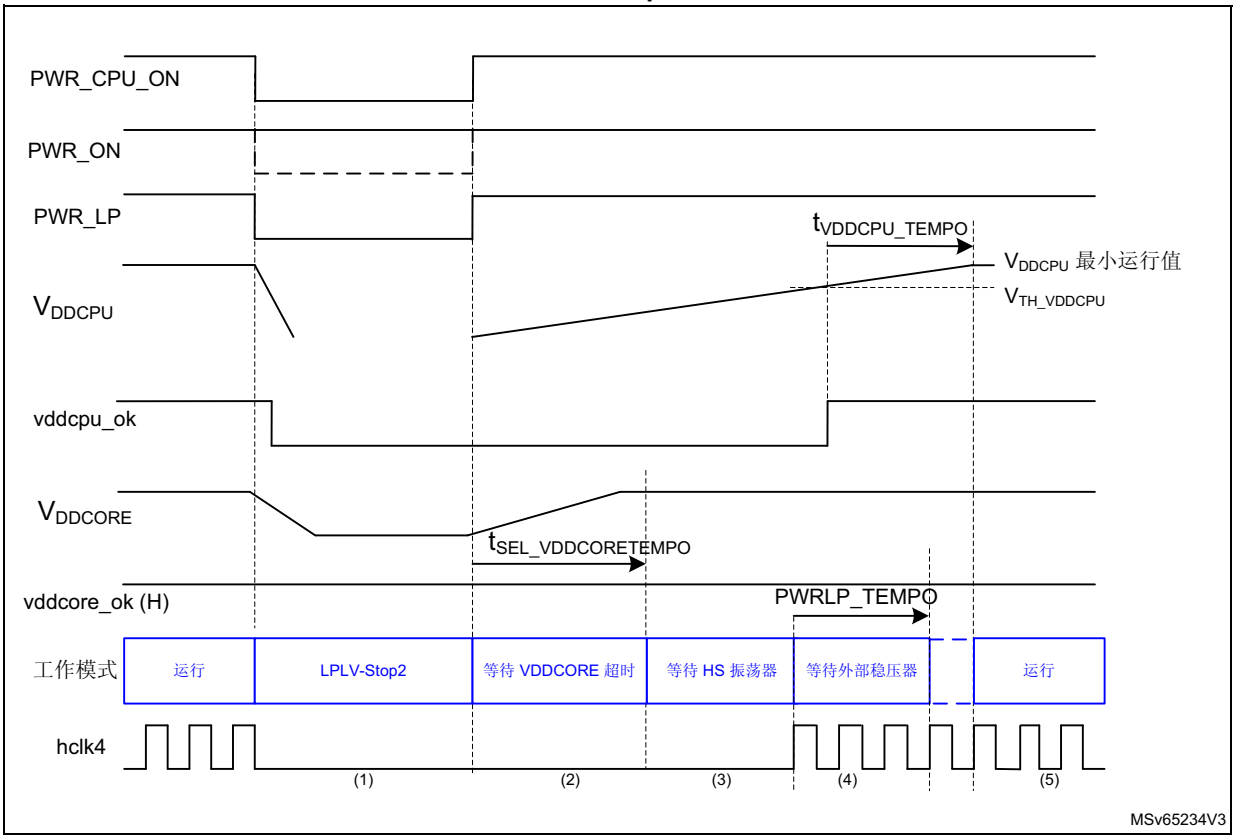
   在 LPLV-Stop2 模式下，在 PWR_LP 翻转后，PWR_ON 保持高电平或设为低电平，具体取决于 *PWR 控制寄存器 1 (PWR_CR1)* 中的 LPCFG 寄存器位。PWR_CPU_ON 清零，PWR_LP 清零为低电平，表示处于 LPLV-Stop2 模式。在此之后，V_{DDCPU} 外部稳压器可进入掉电模式，并将关闭 V_{DDCPU} 电源，而 V_{DDCORE} 外部稳压器可进入低功耗低电压模式，并会降低 V_{DDCORE} 电源电压。
2. 在唤醒事件发生时，将通过 PWR_ON 和 PWR_CPU_ON 信号上升请求外部稳压器，以开启 V_{DDCPU} 电源并升高 V_{DDCORE} 供电电压。将开始一段 $t_{SEL_VDDCORETEMPO}$ 延迟，以等待 V_{DDCORE} 达到运行模式工作电压。
3. 经过这段延迟之后，将使能 HSI 并执行时钟恢复过程。有关详细信息，请参见 RCC：*时钟恢复序列描述*。
4. 一旦振荡器稳定，PWRLP_TEMPO 延迟（可选）就会超时，以便让外部稳压器切换到主模式。默认情况下，复位后该延迟时间最短。（参见*第 10.8.6 节：RCC 低功耗 Stop 模式延迟控制寄存器 (RCC_PWRLPDLYCR)*）。
5. 一旦 V_{DDCPU} 供电电压超出 vddcpu_ok 阈值，就会开始 t_{VDDCPU_TEMPO}，以便 V_{DDCPU} 有足够时间达到其在运行模式下的最小值。
   在经过 PWRLP_TEMPO 和 t_{VDDCPU_TEMPO} 这两段时间后，将开始进入运行模式。

注： *在 LPLV-Stop2 期间，SYSRAM 的内容保持不变，因此无需重新加载其内容。从 LPLV-Stop2 唤醒时，程序使用 BSEC_SCRATCH 寄存器值作为 SYSRAM 的分支地址。因此，必须在进入 LPLV-Stop2 模式之前配置该寄存器。*

注： *如果在 LPLV-Stop2 期间（LPLV-Stop2 持续时间非常短）V_{DDCPU} 始终高于 V_{TH_VDDCPU}，则建议延长 PWRLP_TEMPO（通常为 2 ms，PWRLP_DLY[21:0] = 0x20000）。*

这样稳压器便会有足够的时间完成设置，并在再次开始运行模式之前提供足够的电流驱动。

图 42. 设备 LPLV-Stop2 模式电源控制



9.6.11 系统待机模式

待机模式可实现最低功耗。它基于 MPU 子系统 CStandby 模式，就像系统 LPLV-Stop2 模式一样，但是 V_{DDCORE} 和 V_{DDCPU} 电源稳压器关闭。

仅当 MPU 处于 CStandby 模式并且 *PWR MPU 控制寄存器 (PWR_MPUCR)* 中的 PDDS 位请求待机时，系统才进入待机模式。当系统进入待机模式时，PWR_ON 和 PWR_CPU_ON 引脚设为低电平，PWR_LP 引脚清零为低电平。应禁止为 V_{DDCORE} 和 V_{DDCPU} 提供电压的外部稳压器，因此 V_{DDCORE} 和 V_{DDCPU} 域也将掉电。PLL、HSI、CSI 振荡器和 HSE 振荡器也会被关闭，SRAM 和寄存器内容将丢失，但 V_{DD}、VSW、备份域（即 RTC 寄存器、RTC 备份寄存器、LSE、Backup RAM）和待机电路除外（参见*第 9.3.2 节：电源系统启动序列*）。

在系统待机模式下，可以通过对各控制位进行编程来选择以下功能：

- 独立的看门狗 (IWDG)：IWDG 通过写入其密钥寄存器启动或使用硬件选项来启动。一旦启动便无法停止，除非通过应用复位。请参见*第 45 章：独立看门狗 (IWDG)*。
- 实时时钟 (RTC)：通过备份域控制寄存器 (RCC_BDCR) 中的 RTCEN 位进行配置。
- 内部 RC 振荡器 (LSI RC)：通过控制/状态寄存器 (RCC_CSR) 中的 LSION 位进行配置。
- 外部 32.768 kHz 振荡器 (LSE OSC)：通过备份域控制寄存器 (RCC_BDCR) 中的 LSEON 位进行配置。

进入待机模式

当 *PWR MPU 控制寄存器 (PWR_MPUCR)* 中的 PDDS 位请求待机时，应根据*第 9.6.3 节：进入低功耗模式*进入待机模式。

有关如何进入待机模式的详细信息，请参见*表 44*。

退出待机模式

应根据*第 9.6.4 节：使用 WFI 或 WFE 退出低功耗模式*退出待机模式。

有关如何退出待机模式的详细信息，请参见*表 44*。

检测到应用复位（NRST、IWDG 等）、WKUP 引脚事件、RTC 闹钟、Tamper 事件或时间戳事件时，系统退出待机模式。除电源控制和状态寄存器（*PWR 控制寄存器 2 (PWR_CR2)* 和 *PWR 控制寄存器 3 (PWR_CR3)*）以及 *PWR MPU 控制寄存器 (PWR_MPUCR)*、*PWR 唤醒标志寄存器 (PWR_WKUPFR)* 和 *PWR MPU 唤醒使能寄存器 (PWR_MPUWKUPENR)* 的 SBF 位外，所有寄存器都将在从待机模式唤醒后复位。

从待机模式唤醒后，程序将按照上电复位（启动选项采样、启动向量复位获取等）后的方式重新执行。MPU 开始启动，Boot ROM 管理启动过程。

可以通过 *RCC 复位状态设置寄存器 (RCC_MP_RSTSSETR)* 中的 STDBYRSTF 和 CSTDBYRSTF 位检索退出待机/CStandby 状态标志。这些位由 Boot ROM 通过 *PWR MPU 控制寄存器 (PWR_MPUCR)* 中的 SBF/SBFMPU 位设置。SBF/SBFMPU 位专用于 Boot ROM 的使用，不应在其他场景使用。

表 44. 待机

待机模式	说明
进入模式	– MPU 子系统处于 CStandby 模式，并且没有有效的 EXTI 唤醒源。 – PDDS 位选择待机。 – 电源控制/状态寄存器 (PWR_WKUPFR) 中的所有 WKUPF 位均清零。 – 所有 RTC 唤醒事件均被清除。
退出模式	– WKUP 引脚、RTC 闹钟（闹钟 A 和闹钟 B）、RTC 唤醒、Tamper 事件、时间戳事件、应用复位（例如 NRST 和 IWDG(1)）或系统复位。
唤醒延迟	请参见*图 43：设备待机模式电源控制*。

1. 当 IWDG1 和 IWDG2 配置为生成应用复位以及配置为生成系统调试唤醒事件时，均会复位系统。

待机模式下的 I/O 状态

在待机模式下，除以下引脚外，其他所有 I/O 引脚都处于高阻态且无上下拉：

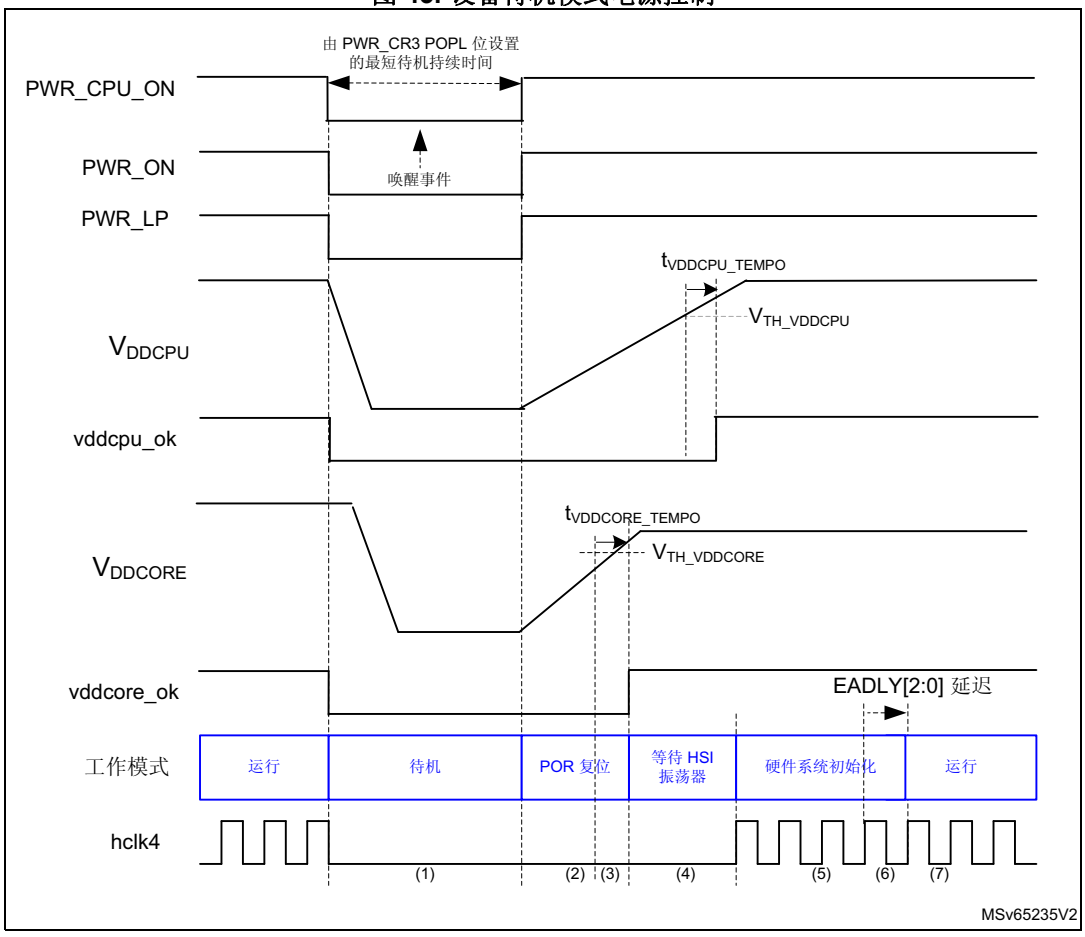
- NRST、PWR_ON、PWR_CPU_ON 和 PWR_LP 引脚（仍然可用）
- RTC 引脚（如果配置为 Tamper、时间戳、RTC 闹钟输出或 RTC 时钟校准输出）
- WKUP 引脚，如果使能，则可以通过 *PWR 唤醒控制寄存器 (PWR_WKUPCR)* 中的 WKUPPUPD 寄存器位配置 WKUP 引脚的上下拉。

电源待机序列

图 43 所示为系统待机模式序列。如果通过 *PWR MPU 控制寄存器 (PWR_MPUCR)* 中的 PDDS 寄存器位进行了相应选择，将进入待机模式。

1. 进入待机模式时，PWR_ON 和 PWR_CPU_ON 信号请求关闭 V_{DDCORE} 和 V_{DDCPU} 电源，hclk4 将停止。振荡器也会停止。PWR_LP 信号清零为低电平。
   如果需要，可以在 *PWR 控制寄存器 3 (PWR_CR3)* 的 POPL 位中定义一个最短 PWR_ON 和 PWR_CPU_ON 待机脉冲低电平时间。
2. 在唤醒事件发生时，通过 PWR_ON 和 PWR_CPU_ON 信号请求外部稳压器，以打开 V_{DDCORE} 和 V_{DDCPU} 电源。
   如果唤醒事件在经过最短 PWR_ON 脉冲低电平时间之前发生，则只有在最短待机 PWR_ON 和 PWR_CPU_ON 脉冲低电平时间过后，才会请求外部稳压器打开 V_{DDCORE} 和 V_{DDCPU} 电源。
   只要 V_{DDCORE} 电源电压低于 $V_{TH_VDDCORE}$ 阈值，系统就保持复位状态
3. 一旦 V_{DDCORE} 超过 $V_{TH_VDDCORE}$ 阈值，就会插入一个 $t_{VDDCORE_TEMPO}$ 延迟，以等待 V_{DDCORE} 达到最小工作值（有关 $t_{VDDCORE_TEMPO}$ 和 V_{DDCORE} 的最小工作值，请参见数据手册）。
4. 系统解除复位后，HSI 振荡器将被使能。
5. HSI 振荡器稳定后，硬件便立即开始初始化系统。
6. 一旦 V_{DDCPU} 超过 V_{TH_VDDCPU} 阈值，就会插入一个 t_{VDDCPU_TEMPO} 延迟，以等待 V_{DDCPU} 达到最小工作值（有关 t_{VDDCPU_TEMPO} 和 V_{DDCPU} 最小工作值的信息，请参见数据手册）。
   当 HSI 振荡器稳定且 V_{DDCPU} 高于 vddcpu_ok 阈值时，Boot ROM 将使用可编程的 EADLY[2:0] 延迟，以确保所有可引导的外部接口在 Boot ROM 开始访问它们之前，都得到正确供电。有关 EADLY[2:0] 编程的详细信息，请参见*第 10 章：复位和时钟控制 (RCC)*。
7. 在 EADLY[2:0] 延迟结束时进入运行模式。

图 43. 设备待机模式电源控制



9.6.12 系统 VBAT 模式

在 VBAT 模式下，V_{DD}、V_{DDCORE} 和 V_{DDCPU} 稳压器被关闭，VSW 备份域（即 RTC 寄存器、RTC 备份寄存器、LSE 和 Backup RAM）则通过 VBAT 引脚供电。

VBAT 模式下的 I/O 状态

在 VBAT 模式下，除了 RTC 和 Tamper 引脚外，所有 I/O 引脚都会掉电

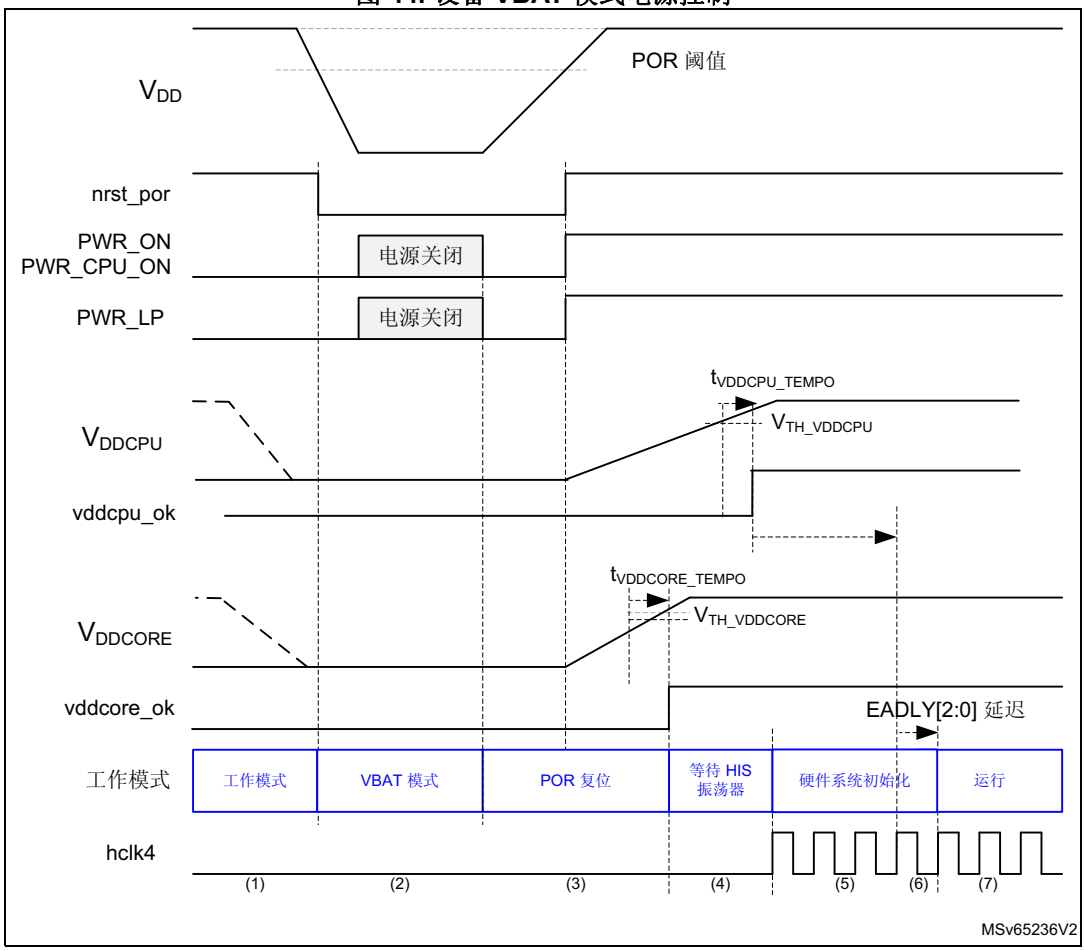
电源 VBAT 序列。

图 44 所示为系统 VBAT 模式序列。

1. 通过关闭 V_{DD} 电源，可将设备置于 VBAT 模式。必须先关闭 V_{DDCORE} 和 V_{DDCPU} 电源，再关闭 V_{DD} 电源（PWR_ON 和 PWR_CPU_ON 低电平）。当系统处于待机模式时，就是这种情况。
2. 当 V_{DD} 掉电时，设备进入 VBAT 模式。PWR_ON、PWR_CPU_ON 和 PWR_LP 信号变为高阻态 (HiZ)。
3. 要退出 VBAT 模式，必须重新建立 V_{DD} 电源，POR 监视 V_{DD} 电源电压。一旦电源电压超过 POR 阈值电压，就要通过 PWR_ON 和 PWR_CPU_ON 引脚使能提供 V_{DDCORE} 和 V_{DDCPU} 供电电压的外部稳压器：
   只要 V_{DDCORE} 低于 vddcore_ok 阈值，系统就保持复位状态。

4. 一旦 V_{DDCORE} 电源电压高于 vddcore_ok 阈值，系统就会在 HSI 振荡器使能后退出复位状态。
5. HSI 振荡器稳定后，硬件便立即开始初始化系统。
6. 当 HSI 振荡器稳定且 V_{DDCPU} 高于 vddcpu_ok 阈值时，Boot ROM 将使用可编程的 EADLY[2:0] 延迟，以确保所有可引导的外部接口在 Boot ROM 开始访问它们之前，都得到正确供电。有关 EADLY[2:0] 编程的详细信息，请参见*第 10 章：复位和时钟控制 (RCC)*。
7. 在 EADLY[2:0] 延迟结束时进入运行模式。

图 44. 设备 VBAT 模式电源控制



9.6.13 Stop 调试

当系统进入 Stop 低功耗模式时，将根据 DBGMCU 模块中 DBG_STOP 的配置来工作。

如果使能了在 Stop 模式下调试，则当进入 Stop、LP-Stop、LPLV-Stop 和 LPLV-Stop2 模式时，PWR 不会通过 PWR_ON、PWR_CPU_ON 和 PWR_LP 信号指示低功耗模式。这样可保持 V_{DDCORE} 和 V_{DDCPU} 域从主模式下的稳压器获得供电。

当退出 Stop、LP-Stop、LPLV-Stop 或 LPLV-Stop2 模式时，会像从正常 Stop 模式退出一样使能 MPU 和系统逻辑，STOPF 标志置 1，并且当 MPU 进入 CStandby 模式时 SBFMPU 标志也置 1。

9.6.14 待机调试

当系统进入待机低功耗模式时，将根据 DBG 模块中 DBG_STBY 的配置工作。

如果使能了在待机模式下调试，则当进入待机模式时，PWR 会使 PWR_ON 和 PWR_CPU_ON 保持工作状态，以便 V_{DDCORE} 和 V_{DDCPU} 域继续供电。在待机模式下，V_{DDCORE} 和 V_{DDCPU} 域不会复位，从而允许 CPU 和总线时钟保持工作状态。

退出待机模式时会产生复位，从而复位 V_{DDCORE} 和 V_{DDCPU} 逻辑（调试逻辑除外）。从调式待机模式唤醒时，MPU 和系统逻辑像从正常待机模式退出一样复位，并且 SBF 和 SBFMPU 标志置 1。

9.7 PWR WKUP 引脚

仅当唤醒中断在 WKUPEN 中被禁止时，WKUPP 中 WKUP 的极性才能改变。

为避免错误中断，使用 WKUPEN 使能唤醒中断之前，必须通过 WKUPC 将 WKUPF 标志清零。

参见：

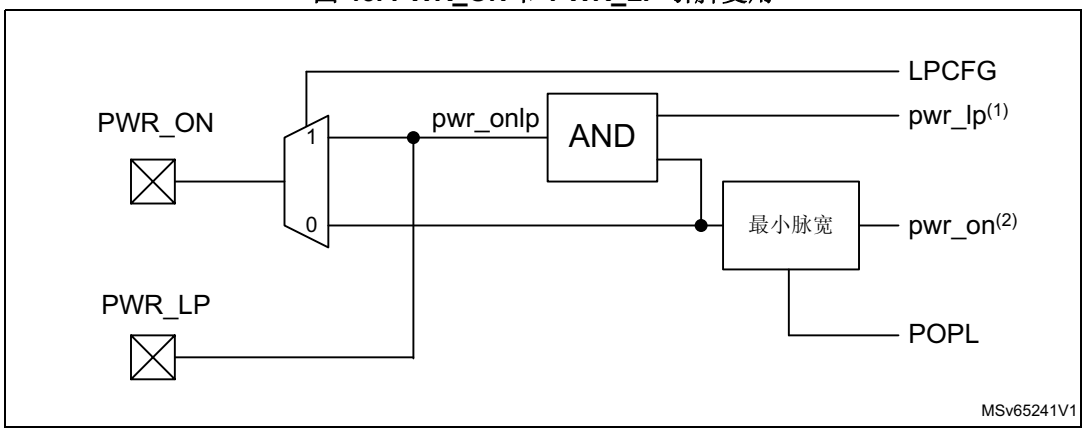
- *PWR MPU 唤醒使能寄存器 (PWR_MPUWKUPENR)* 中的 WKUPEN 寄存器位。
- *PWR 唤醒控制寄存器 (PWR_WKUPCR)* 中的 WKUPP 寄存器位
- *PWR 唤醒标志寄存器 (PWR_WKUPFR)* 中的 WKUPF 寄存器位。
- *PWR 唤醒控制寄存器 (PWR_WKUPCR)* 中的 WKUPC 寄存器位

9.8 PWR_ON、PWR_CPU_ON 和 PWR_LP 引脚

PWR_ON 引脚可以根据 *PWR 控制寄存器 1 (PWR_CR1)* 中 LPCFG 寄存器位的选择，发出 pwr_on 或 pwr_onlp 信号。

PWR_LP 引脚发出 pwr_onlp 信号

图 45. PWR_ON 和 PWR_LP 引脚复用



1. 当系统进入低功耗 Stop 模式时，pwr_lp 信号被清零为低电平。请参见 *PWR 控制寄存器 1 (PWR_CR1)* 中的 LPDS 位。
2. 当系统请求外部 V_{DDCORE} 稳压器关闭 V_{DDCORE} 时，pwr_on 信号被清零为低电平。

PWR_ON 引脚

当 LPCFG 选择 pwr_on 信号时，PWR_ON 引脚仅在待机模式下为低电平。

当 LPCFG 选择 PWR_ON 上的 pwr_onlp 信号时，必须在进入低功耗模式之前，将信号为低电平的定义（待机、LP-Stop、LPLV-Stop 或 LPLV-Stop2）通知给外部电源管理设备。PWR_ON 引脚在待机、LP-Stop、LPLV-Stop 和 LPLV-Stop2 模式下为低电平。

PWR_CPU_ON 引脚

当系统进入 LPLV-Stop2 或待机模式时，PWR_CPU_ON 引脚清零为低电平，请求关断外部 V_{DDCPU} 稳压器。

9.9 PWR 中断

PWR 提供以下中断：

- MPU 的唤醒中断。
  - 通过 *PWR MPU 唤醒使能寄存器 (PWR_MPUWKUPENR)* 中的 WKUPEN[6:1] 寄存器位使能。
  - *PWR 唤醒标志寄存器 (PWR_WKUPFR)* 提供中断状态标志 WKUPF[6:1]。

还提供了与 Stop 和待机模式以及监视功能相关的其他状态标志，参见*表 45*：

表 45. FIFO 状态标志

状态标志	说明	中断	清除方式
STOPF	退出系统 Stop、LP-Stop、LPLV-Stop 或 LPLV-Stop2 模式	从 Stop 模式唤醒事件	通过 CSSF
SBF(1)	退出系统待机模式	从待机模式唤醒事件	通过 CSSF
SBFMPU(1)	MPU 退出 CStandby 模式（系统 LPLV-Stop2 或系统待机模式）	从 Cstandby 模式唤醒事件。	通过 CSSF
WKUPF[6:1]	唤醒标志	MPU 从待机和 Stop 唤醒事件。	WKUPC[6:1]
PVDO	可编程电压检测器	通过 EXTI 实现的中断	n.a.
AVDO	模拟电压检测器	通过 EXTI 实现的中断	n.a.
VBATL、VBATH	VBAT 电源监视	通过 RTC Tamper 中断从待机和 Stop 唤醒	n.a.
TEMPL、TEMPH	温度监视	通过 RTC Tamper 中断从待机和 Stop 唤醒	n.a.
BRRDY	备份稳压器就绪	无相关中断	n.a.
USB33RDY	USB HS 3V3 供电就绪	无相关中断	n.a.
REG18RDY	稳压器 1V8 供电就绪	无相关中断	n.a.
REG11RDY	稳压器 1V1 供电就绪	无相关中断	n.a.

1. SBF 和 SBFMPU 专用于 Boot ROM 的使用，不应在其他场景使用。*RCC 复位状态设置寄存器 (RCC_MP_RSTSSETR)* 中的 STDBYRSTF 和 CSTDBYRSTF 位应由应用代码作为复位源标识过程的一部分使用。请参见*第 10.3.11 节：复位源标识*。

9.10 PWR TrustZone 安全性

PWR 能够防止寄存器被非安全访问修改。TrustZone 安全性通过*第 10.8.1 节：RCC 安全配置寄存器 (RCC_SECCFGR)* 中的 RCC PWRSEC 寄存器位激活。从 PWR 层面上看，安全性在于防止非安全写访问对以下内容进行更改：

- VBAT 和 TEMP 监视器、PVD 和 AVD 的设置。
- 低功耗深度睡眠和 RAM 低功耗设置。
- 备份域写保护。
- 备份稳压器、1V8 稳压器、1V1 稳压器和 USB 3.3V 电压检测器的设置。
- 备份电池充电设置。
- MPU 电源控制寄存器设置。
- 待机和 LPLV-Stop2 模式唤醒设置和标志。

PWR TrustZone 安全性同时为所有功能使能写保护。

表 46. 寄存器安全性概述

寄存器名称	访问类型	安全性(1)
PWR_CR1	RW	写安全
PWR_CSR1	R	非安全
PWR_CR2	RW	写安全
PWR_CR3	RW	写安全
PWR_MPUCR	RW	写安全
PWR_WKUPCR	RW	写安全(2)
PWR_WKUPFR	R	非安全
PWR_MPUWKUPENR	RW	写安全
PWR_VERR	R	非安全
PWR_ID	R	非安全
PWR_SID	R	非安全

1. 安全性通过*第 10.8.1 节：RCC 安全配置寄存器 (RCC_SECCFGR)* 中的 RCC PWRSEC 寄存器位使能。
2. 安全性还可通过 *PWR MPU 唤醒使能寄存器 (PWR_MPUWKUPENR)* 寄存器中的 WKUPEN 位按通道使能。

9.11 PWR 寄存器

除非另有明确规定，否则可按字、半字和字节格式访问 PWR 寄存器。

9.11.1 PWR 控制寄存器 1 (PWR_CR1)

PWR control register 1

偏移地址：0x000

复位值：0x0000 0200

只要系统复位就会复位。

当通过*第 10.8.1 节：RCC 安全配置寄存器 (RCC_SECCFGR)* 中的 PWRSEC 寄存器位使能时，该寄存器可提供写访问安全性。使能安全性时，非安全写访问会产生总线错误。安全和非安全读访问被允许并返回寄存器值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALS[1:0]		AVDEN
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	MPU_RAM_LOW SPEED	DBP	PLS[2:0]			PVDEN	STOP2	LVDS	LPCFG	LPDS
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:19 保留，必须保持复位值。

位 18:17 **ALS[1:0]**：模拟电压检测器电平选择 (Analog Voltage Detector level selection)

这些位用于选择通过 AVD 检测的电平阈值。

00: 1.7V

01: 2.1V

10: 2.5V

11: 2.8V

位 16 **AVDEN**：V_{DDA} 上的外设电压监视器使能 (Peripheral Voltage Monitor on V_{DDA} enable)

0: 禁止 V_{DDA} 上的外设电压监视器

1: 使能 V_{DDA} 上的外设电压监视器

位 15:10 保留，必须保持复位值。

位 9 **MPU_RAM_LOWSPEED**：MPU 内存低速选择 (Low speed selection for MPU memories)

当 MPU 需要在超频（高速）模式下运行时，在 V_{DDCPU} 电源达到超频模式电压范围并在将 MPU 频率提高到超频频率范围之前，必须通过软件复位此位。

在将 MPU 频率降低到标准频率范围并在将 MPU 电压降低到超频模式电压范围以下之前，必须设置此位以保证在运行模式下工作。

0: MPU 可以在超频运行模式下工作（V_{DDCPU} 在超频运行模式电压范围内）

1: MPU 只能在运行模式下工作（V_{DDCPU} 低于超频运行模式电压范围）

位 8 **DBP**：禁止备份域写保护 (Disable backup domain write protection)

在复位状态下，RCC_BDCR、PWR_CR2、RTC 和备份寄存器均受到保护，可防止寄生写访问。必须将该位置 1 才能使能对这些寄存器的写访问。

0: 禁止对 RTC 和备份域寄存器进行写访问。

1: 使能对 RTC 和备份域寄存器进行写访问。

位 7:5 **PLS[2:0]**：可编程电压检测器电压选择 (Programmable Voltage Detector level selection)

当 SYSCFG 寄存器位 PVDL 置 1 时，这些位为只读位（PVDL 置 1 时，写入此寄存器不会生成总线错误）。

这些位用于选择通过 PVD 检测到的电压阈值。

000: 1.95 V

001: 2.1 V

010: 2.25 V

011: 2.4 V

100: 2.55 V

101: 2.7 V

110: 2.85 V

111: PVD_IN 上的外部电压（与内部 VREFINT 比较）

位 4 **PVDEN**：可编程电压检测器使能 (Programmable voltage detector enable)

当 SYSCFG 寄存器位 PVDL 置 1 时，该位为只读位（PVDL 置 1 时，写入此寄存器不会生成总线错误）。

0: 禁止可编程电压检测器。

1: 使能可编程电压检测器

位 3 **STOP2**：系统 LPLV-Stop2 模式选择

0: 当 PDDS=0 且 MPU 进入 CStop 时，使系统保持 Stop 模式

1: 当 PDDS = 0 时，允许系统进入 LPLV-Stop2 模式

位 2 **LVDS**：低电压深度睡眠 LPLV-Stop 模式选择 (Low Voltage Deepsleep LPLV-Stop mode selection)。

仅当在 LPDS 中选择低功耗 Stop 模式，并更改了 V_{DDCORE} 和 V_{DDCPU} 域供电复位电压时，该位才有效。

0: LP-Stop 模式 V_{DDCORE} 和 V_{DDCPU} 的供电复位电压与运行模式相同。LP-Stop 模式下 V_{DDCORE} 和 V_{DDCPU} 域的供电电压必须与运行模式下的供电电压保持相同。

1: LPLV-Stop 模式 V_{DDCORE} 和 V_{DDCPU} 域的供电复位电压低于运行模式下。允许在 LPLV-Stop 模式下降低 V_{DDCORE} 和 V_{DDCPU} 域的供电电压（具体电压值请参见数据手册）。

位 1 **LPCFG**：PWR_ON 引脚配置 (PWR_ON pin configuration)

0: PWR_ON 引脚指示待机模式（在 Run、Stop、LP-Stop、LPLV-Stop、LPLV-Stop2 模式下 PWR_ON = 1，在待机模式下 PWR_ON = 0）。

1: PWR_ON 引脚指示待机、LP-Stop、LPLV-Stop 和 LPLV-Stop2 模式（在 Run 和 Stop 模式下 PWR_ON = 1，在 LP-Stop、LPLV-Stop、LPLV-Stop2 和待机模式下 PWR_ON = 0）。

位 0 **LPDS**：低功耗深度睡眠 Stop 模式选择 (Low Power Deepsleep Stop mode selection)。

0: 选择了 Stop 模式，外部稳压器保持在主电源模式 (pwr_lp = 1)。

1: 选择了低功耗 Stop 模式，外部稳压器可进入低功耗模式 (pwr_lp = 0)。LVDS 提供了更多的低功耗模式选择。

9.11.2 PWR 控制状态寄存器 1 (PWR_CSR1)

PWR control status register 1

偏移地址：0x004

复位值：0x0000 0000

只要系统复位就会复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AVDO
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PVDO	Res.	Res.	Res.	Res.
											r				

位 31:17 保留，必须保持复位值。

位 16 **AVDO**：V_{DDA} 上的模拟电压检测器输出 (Analog Voltage detector Output on V_{DDA})
该位由硬件置 1 和清零。仅当通过 AVDEN 位使能 V_{DDA} 上的 AVD 时该位才有效。
0: V_{DDA} 等于或高于通过 ALS[2:0] 位选择的 AVD 阈值。
1: V_{DDA} 低于通过 ALS[2:0] 位选择的 AVD 阈值。
注意：AVD 在待机模式下以及系统复位后会被禁止。因此，该位在待机和复位后等于 0。

位 15:5 保留，必须保持复位值。

位 4 **PVDO**：可编程电压检测器输出 (Programmable Voltage Detect Output)
该位由硬件置 1 和清零。仅当通过 PVDE 位使能了 PVD 时该位才有效。
0: V_{DD} 或 PVD_IN 上的电压等于或高于通过 PLS[2:0] 位选择的 PVD 阈值。
1: V_{DD} 或 PVD_IN 上的电压低于通过 PLS[2:0] 位选择的 PVD 阈值。
注意：PVD 在待机模式下以及系统复位后会被禁止。因此，该位在待机和系统复位后等于 0。

位 3:0 保留，必须保持复位值。

9.11.3 PWR 控制寄存器 2 (PWR_CR2)

PWR control register 2

偏移地址：0x008

复位值：0x0000 0000

不会因从待机模式唤醒、应用复位（NRST、IWDG、...）和 V_{DD} POR 而复位，只能通过 V_{SW} POR 和 VSWRST 复位。

写入该寄存器时，需要访问 6 个等待状态。

复位后，该寄存器受写保护，必须将 *PWR 控制寄存器 1 (PWR_CR1)* 中的 DBP 位置 1 才能向该寄存器写入。当 DBP 清零时，写入该寄存器不会产生总线错误。

当*第 10.8.6 节：RCC 低功耗 Stop 模式延迟控制寄存器 (RCC_PWRLPDLYCR)* 中的 RCC VSWRST 寄存器位将 VSW 域复位时，不得访问该寄存器。

当通过*第 10.8.1 节：RCC 安全配置寄存器 (RCC_SECCFGR)* 中的 PWRSEC 寄存器位使能时，该寄存器可提供写访问安全性。使能安全性时，非安全写访问会产生总线错误。安全和非安全读访问被允许并返回寄存器值。

当寄存器写周期内发生系统复位时，写入的数据无法得到保证。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEMPH	TEMPL	VBATH	VBATL	Res.	Res.	Res.	BRRDY
								r	r	r	r				r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MON EN	Res.	Res.	Res.	BREN
											rw				rw

位 31:24 保留，必须保持复位值。

位 23 **TEMPH**：监视的温度值高于阈值上限 (Monitored temperature level above high threshold)
0: 温度值低于阈值上限，或监视功能被禁止。
1: 温度值等于或高于阈值上限。

位 22 **TEMPL**：监视的温度值高于阈值下限 (Monitored temperature level above low threshold)
0: 温度值高于阈值下限，或监视功能被禁止。
1: 温度值等于或低于阈值下限。

位 21 **VBATH**：监视的 V_{BAT} 值高于阈值上限 (Monitored V_{BAT} level above high threshold)
0: V_{BAT} 值低于阈值上限，或监视功能被禁止
1: V_{BAT} 值等于或高于阈值上限

位 20 **VBATL**：监视的 V_{BAT} 值高于阈值下限 (Monitored V_{BAT} level above low threshold)
0: V_{BAT} 值高于阈值下限，或监视功能被禁止。
1: V_{BAT} 值等于或低于阈值下限

位 19:18 保留，必须保持复位值。

位 17 保留，必须保持复位值。

位 16 **BRRDY**：备份稳压器就绪 (Backup Regulator ready)
由硬件置 1，用以指示备份稳压器已就绪。在备份稳压器就绪之前进入待机或 VBAT 模式可能会导致 Backup RAM 的内容丢失。
0: 备份稳压器未就绪。
1: 备份稳压器就绪。

位 15:5 保留，必须保持复位值。

位 4 **MONEN**：使能 VBAT 和温度监视
将该位置 1 时，将使能 V_{BAT} 电源和温度监视。
0: 禁止 V_{BAT} 和温度监视。
1: 使能 V_{BAT} 和温度监视。

位 3:2 保留，必须保持复位值。

位 1 保留，必须保持复位值。

位 0 **BREN**：使能备份稳压器 (Backup regulator enable)
将该位置 1 时，将使能备份稳压器（用于在待机模式和 VBAT 模式下保持 Backup RAM 的内容）。
如果 BREN 复位，备份稳压器关闭。在运行模式和 Stop 模式下仍可使用 Backup RAM，但在待机模式和 VBAT 模式下其内容会丢失。将该位置 1 后，应用必须等待备份稳压器就绪标志 (BRRDY) 置 1，指示在待机模式和 VBAT 模式下会保持写入到 SRAM 中的数据。
0: 禁止备份稳压器
1: 使能备份稳压器

9.11.4 PWR 控制寄存器 3 (PWR_CR3)

PWR control register 3

偏移地址：0x00C

复位值：0x5000 A000

不会因从待机模式唤醒和应用复位（如 NRST、IWDG）而复位，只能通过 V_{DD} POR 复位。

写入该寄存器时，需要访问 6 个等待状态。

当通过*第 10.8.1 节：RCC 安全配置寄存器 (RCC_SECCFGR)* 中的 PWRSEC 寄存器位使能时，该寄存器可提供写访问安全性。使能安全性时，非安全写访问会产生总线错误。安全和非安全读访问被允许并返回寄存器值。

当寄存器写周期中发生系统复位时，写入的数据无法得到保证。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REG11 RDY	REG11 EN	REG18 RDY	REG18 EN	Res.	USB33 RDY	Res.	USB33 DEN	VDDSD 2VALID	VDDSD 1VALID	POPL[4:0]					VDDSD 2RDY
r	rw	r	rw		r		rw	rw	rw	rw	rw	rw	rw	rw	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDDSD 2DEN	VDDSD 1RDY	VDDSD 1DEN	DDRR ETEN	DDRS RDIS	DDRS REN	VBRS	VBE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	r	rw	rw	rw	rw	rw	rw								

位 31 **REG11RDY**：1V1 稳压器供电就绪 (1V1 regulator supply ready)
0: 1V1 供电未就绪
1: 1V1 供电就绪

位 30 **REG11EN**：1V1 稳压器使能 (1V1 regulator enable)
0: 禁止 1V1 稳压器
1: 使能 1V1 稳压器（在待机模式下 1V1 稳压器会被禁止）

位 29 **REG18RDY**：1V8 稳压器供电就绪 (1V8 regulator supply ready)
0: 1V8 供电未就绪
1: 1V8 供电就绪

位 28 **REG18EN**：1V8 稳压器使能 (1V8 regulator enable)
0: 禁止 1V8 稳压器
1: 使能 1V8 稳压器，且 BYPASS_REG1V8 引脚为低电平（在待机模式下 1V8 稳压器被禁止）。

位 27 保留，必须保持复位值。

位 26 **USB33RDY**：USB 3.3V 供电就绪 (USB 3.3V supply ready)
0: USB 3.3V 供电未就绪
1: USB 3.3V 供电就绪

位 25 保留，必须保持复位值。

位 24 **USB33DEN**：USB 3.3V 电压检测器使能 (USB 3.3V voltage level detector enable)
0: USB 3.3V 电压检测器被禁止
1: USB 3.3V 电压检测器使能

位 23 **VDDSD2VALID**：覆盖 V_{DDSD2} 电压检测器

无论 VDDSD2RDY 的状态如何，该位都使能 V_{DDSD2} 域中的 I/O。一旦 V_{DDSD2} 高于 V_{DDSDx_TH} 并且稳定下来（位 VDDSD2RDY = 1），就可以通过软件将该位置 1。随后可以将 VDDSD2EN 复位为 0 以禁止电压检测器并降低功耗（有关 V_{DDSDx_TH} 的值，请参见数据手册）。

0: I/O 使能/禁止由 V_{DDSD2} 电压检测器控制

1: 强制激活 I/O 使能

位 22 **VDDSD1VALID**：覆盖 V_{DDSD1} 电压检测器 (Override V_{DDSD1} voltage detector)

无论 VDDSD1RDY 的状态如何，该位都使能 V_{DDSD1} 域中的 I/O。一旦 V_{DDSD1} 高于 V_{DDSDx_TH} 并且稳定下来（位 VDDSD1RDY = 1），就可以通过软件将该位置 1。随后可以将 VDDSD1EN 复位为 0 以禁止电压检测器并降低功耗（有关 V_{DDSDx_TH} 的值，请参见数据手册）。

0: I/O 使能/禁止由 V_{DDSD1} 电压检测器控制

1: 强制激活 I/O 使能

位 21:17 **POPL[4:0]**：PWR_ON、PWR_CPU_ON 待机脉冲低电平配置 (PWR_ON, PWR_CPU_ON Standby pulse low configuration)

这些位由软件置 1 和清零，它们定义了待机模式下 PWR_ON 和 PWR_CPU_ON 脉冲低电平的最短持续时间（对 LP-Stop 和 LPLV-Stop 模式无影响）。

如果 POPL 脉冲低电平配置需要，LSI 振荡器会自动使能。

00000: 最短低电平时间约为 62.5 us（2 个 LSI 周期）

00001: 最短低电平时间约为 1 ms（1 x 32 个 LSI 周期）

00010: 最短低电平时间约为 2 ms（2 x 32 个 LSI 周期）

...

11111: 最短低电平时间约为 31 ms（31 x 32 个 LSI 周期）

位 16 **VDDSD2RDY**：V_{DDSD2} 供电就绪 (V_{DDSD2} supply ready)

当电压检测器处于工作状态 (VDDSD2DEN = 1) 时，该位指示 V_{DDSD2} 电源上是否存在高于 V_{DDSDx_TH} 的电压（有关 V_{DDSDx_TH} 值，请参见数据手册）。

0: V_{DDSD2} 电源不存在，或者电压检测器已关闭 (VDDSD2EN = 0)

1: V_{DDSD2} 电源存在

位 15 **VDDSD2DEN**：V_{DDSD2} 电压检测器使能 (V_{DDSD2} voltage level detector enable)

电压检测器默认处于工作状态。如果 V_{DDSD2} 供电电压高于 V_{DDSDx_TH}，它会使能 V_{DDSD2} 域中的 I/O。一旦电源电压稳定在此阈值以上 (VDDSD2RDY = 1)，就可以通过先设置 VDDSD2VALID = 1 然后 VDDSD2EN = 0 来关闭电压检测器以降低功耗，而不禁止 I/O（有关 V_{DDSDx_TH} 值，请参见数据手册）。

0: V_{DDSD2} 电压检测器已关闭。如果 VDDSD2VALID = 0，V_{DDSD2} 域中的 I/O 也会被禁止

1: V_{DDSD2} 电压检测器处于工作状态。如果 VDDSD2RDY = 1，则 V_{DDSD2} 域中的 I/O 将处于工作状态

位 14 **VDDSD1RDY**：V_{DDSD1} 供电就绪 (V_{DDSD1} supply ready)

当电压检测器处于工作状态 (VDDSD1DEN = 1) 时，该位指示 V_{DDSD1} 电源上是否存在高于 V_{DDSDx_TH} 的电压（有关 V_{DDSDx_TH} 值，请参见数据手册）。

0: V_{DDSD1} 电源不存在，或者电压检测器已关闭 (VDDSD1EN = 0)

1: V_{DDSD1} 电源存在

位 13 **VDDSD1DEN**：V_{DDSD1} 电压检测器使能 (V_{DDSD1} voltage level detector enable)

电压检测器默认处于工作状态。如果 V_{DDSD1} 供电电压高于 V_{DDSDx_TH}，它会使能 V_{DDSD1} 域中的 I/O。一旦电源电压稳定在此阈值以上 (VDDSD1RDY = 1)，就可以通过先设置 VDDSD1VALID = 1 然后 VDDSD1EN = 0 来关闭电压检测器以降低功耗，而不禁止 I/O（请参考数据手册了解 V_{DDSDx_TH} 的值）。

0: V_{DDSD1} 电压检测器已关闭。如果 VDDSD1VALID = 0，V_{DDSD1} 域中的 I/O 也会被禁止

1: V_{DDSD1} 电压检测器处于工作状态。如果 VDDSD1RDY = 1，则 V_{DDSD1} 域中的 I/O 将处于工作状态

位 12 **DDRRETEN**：DDR 保持使能 (DDR retention enable)

在 NRST 引脚复位时由硬件置 1，当通过 RCC_MP_APRSTCR.RDCTLEN 寄存器位使能了复位延迟控制时，或者当软件写入 'b1 时，DDR PHY 将其引脚置于保持状态。

要禁止保持，软件必须在该位中写入 'b0。

0: 禁止 DDR 自刷新保持。

1: 使能 DDR 自刷新保持。

位 11 **DDRSRDIS**：待机后 DDR 自刷新保持禁止 (DDR self-refresh retention after standby disable)

当写入 'b1 时，因 DDRSREN 而使能的 DDR PHY 引脚保持将被禁止。

0: 无操作

1: 当 DDRSREN 为 'b1 时：禁止待机模式后 DDR 自刷新保持。一旦 DDR 自刷新保持被禁止，该位就将由硬件复位。

当 DRRSREN 为 'b0 时无操作。

位 10 **DDRSREN**：待机模式下 DDR 自刷新使能 (DDR self-refresh in standby mode enable)

如果该位置 1，DDR PHY 在进入待机模式时会将其引脚置于保持状态。

0: 禁止进入待机模式时 DDR 自刷新保持。

1: 使能进入待机模式时 DDR 自刷新保持（待机后 DDR 自刷新保持必须通过 DDRSRDIS 禁止）。

位 9 **VBRS**：VBAT 充电电阻选择 (VBAT charging resistor selection)

0: 通过 5 kOhm 电阻器为 VBAT 充电。

1: 通过 1.5 kOhm 电阻器为 VBAT 充电。

位 8 **VBE**：VBAT 充电使能 (VBAT charging enable)

0: 禁止 VBAT 电池充电。

1: 使能 VBAT 电池充电。

位 7:0 保留，必须保持复位值。

9.11.5 PWR MPU 控制寄存器 (PWR_MPUCR)

PWR MPU control register

偏移地址：0x010

复位值：0x0000 0000

有关复位条件，请参见各个位。

写入该寄存器时，需要访问 6 个等待状态。

当通过*第 10.8.1 节：RCC 安全配置寄存器 (RCC_SECCFGR)* 中的 PWRSEC 寄存器位使能时，该寄存器可提供写访问安全性。使能安全性时，非安全写访问会产生总线错误。安全和非安全读访问被允许并返回寄存器值。

当寄存器写周期中发生系统复位时，写入的数据无法得到保证。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STANDBYWFIL2	Res.	Res.	Res.	Res.	Res.	通过 CSSF	Res.	SBFMPU	SBF	STOPF	Res.	Res.	Res.	Res.	PDDS
r						rw		r	r	r					rw

位 31:16 保留，必须保持复位值。

位 15 **STANDBYWFIL2**：MPU 系统空闲指示
该位由硬件根据 MPU 工作模式置 1 和复位。
0: MPU 系统 CRun
1: MPU 系统 CSleep 或 CStop 或 CStandby

位 14:10 保留，必须保持复位值。

位 9 **CSSF**：清除 MPU 待机、Stop 标志 (Clear MPU Standby, Stop flags)。（始终读为 0）
只要系统复位，该位就会复位。
0: 无影响
1: 写入时，清除 MPU 标志（STOPF、SBF 和 SBFMPU）。该寄存器位通过硬件清零。

位 8 保留，必须保持复位值。

位 7 **SBFMPU**：MPU 待机标志 (MPU Standby flag)
该位由硬件置 1，只能通过 V_{DD} POR 复位或将 MPU CSSF 位置 1 来清零（从待机模式退出时不复位）。该位由 MPU Boot ROM 清零。有关更多信息，请参见 Boot ROM 应用笔记。
0: MPU 未进入 CStandby 模式
1: MPU 已进入 CStandby 模式

位 6 **SBF**：系统待机标志 (System Standby flag)
该位由硬件置 1，只能通过 V_{DD} POR 复位或将 MPU CSSF 位置 1 来清零（从待机模式退出时不复位）。该位由 MPU Boot ROM 清零。有关更多信息，请参见 Boot ROM 应用笔记。
0: 系统未处于待机模式
1: 系统已处于待机模式，系统内容已丢失。

位 5 **STOPF**：停止标志 (Stop flag)
该位由硬件置 1，并且只能通过系统复位或将 MPU CSSF 位置 1 来清零。该位由 MPU Boot ROM 清零。有关更多信息，请参见 Boot ROM 应用笔记。
0: 系统未处于 Stop 模式
1: 系统已处于 Stop 模式，时钟系统已停止。

位 4:1 保留，必须保持复位值。

位 0 **PDDS**：系统掉电深度睡眠选择 (System Power Down Deepsleep selection)
该位仅通过 V_{DD} POR 复位来复位（从待机模式退出时不复位）。
允许为系统定义深度睡眠模式。
0: 当 MPU 进入 CStop 模式时，使系统保持在 Stop 模式，或者当 MPU 进入 CStandby 模式时，使系统保持在 LPLV-Stop2 模式。
1: 允许在 MPU 进入 CStandby 时系统进入待机模式。

9.11.6 PWR 唤醒控制寄存器 (PWR_WKUPCR)

PWR wakeup control register

偏移地址：0x020

复位值：0x0000 0000

不通过从待机模式唤醒复位，而是通过任何应用复位（如 NRST、IWDG）来复位。

写入此寄存器时需要访问 6 个等待状态（清零 WKUPF 时，AHB 写访问在 WKUPF 清零后完成）。

当通过*第 10.8.1 节：RCC 安全配置寄存器 (RCC_SECCFGR)* 中的 PWRSEC 寄存器位使能时，该寄存器可提供写访问安全性。如果使能安全性，则当设置了 *PWR MPU 唤醒使能寄存器 (PWR_MPUWKUPENR)* 中的相应 WKUPEN[6:1] 位时，对各个 WKUPC[6:1]、WKUPP[6:1] 位和 WKUPPUPD[6:1] 位进行的非安全写访问将被丢弃。不会生成总线错误。安全和非安全读访问被允许并返回寄存器值。

当寄存器写周期中发生系统复位时，写入的数据无法得到保证。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	WKUPPUPD6 [1:0]		WKUPPUPD5 [1:0]		WKUPPUPD4 [1:0]		WKUPPUPD3 [1:0]		WKUPPUPD2 [1:0]		WKUPPUPD1 [1:0]	
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	WKUP P6	WKUP P5	WKUP P4	WKUP P3	WKUP P2	WKUP P1	Res.	Res.	WKUP C6	WKUP C5	WKUP C4	WKUP C3	WKUP C2	WKUP C1
		rw	rw	rw	rw	rw	rw			rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:28 保留，必须保持复位值。

位 27:16 **WKUPPUPDn[1:0]**：WKUPn 引脚（n = 6 到 1）的唤醒上下拉配置

当 WKUPENn = 1 时，这些位定义使用的 IO 引脚上下拉配置（请注意，关联的 GPIO 端口上下拉配置必须设置为相同的值或 00）。唤醒引脚上拉/下拉配置会待机模式下保持不变。

当 RCC 中使能了 PWRSEC 并且 *PWR MPU 唤醒使能寄存器 (PWR_MPUWKUPENR)* 中的 WKUPENn 置 1 时，位对 n 只能由安全访问写入，对此位对的非安全写操作将被丢弃。

当 RCC 中禁止了 PWRSEC 或 *PWR MPU 唤醒使能寄存器 (PWR_MPUWKUPENR)* 中的 WKUPENn 清零时，位对 n 可以由安全访问和非安全访问写入。

00: 无上拉/下拉

01: 上拉

10: 下拉

11: Reserved

位 15:14 保留，必须保持复位值。

位 13:8 **WKUPP[6:1]**：WKUPn 引脚的唤醒极性位，n 的范围为 [6:1]

这些位定义在外部唤醒引脚 WKUPn 上进行事件检测所用的极性。

当 RCC 中使能了 PWRSEC 并且 *PWR MPU 唤醒使能寄存器 (PWR_MPUWKUPENR)* 中的 WKUPENn 置 1 时，位 n 只能由安全访问写入，对此位的非安全写操作将被丢弃。

当 RCC 中禁止了 PWRSEC 或 *PWR MPU 唤醒使能寄存器 (PWR_MPUWKUPENR)* 中的 WKUPENn 清零时，位 n 可以由安全和非安全访问写入。

0: 检测高电平（上升沿）

1: 检测低电平（下降沿）

位 7:6 保留，必须保持复位值。

位 5:0 **WKUPC[6:1]**：清除 WKUPn 引脚的唤醒标志，n 的范围为 [6:1]

当 RCC 中使能了 PWRSEC 并且 *PWR MPU 唤醒使能寄存器 (PWR_MPUWKUPENR)* 中的 WKUPENn 置 1 时，位 n 只能由安全访问写入，对此位的非安全写操作将被丢弃。

当 RCC 中禁止了 PWRSEC 或 *PWR MPU 唤醒使能寄存器 (PWR_MPUWKUPENR)* 中的 WKUPENn 清零时，位 n 可以由安全和非安全访问写入。

这些位始终读为 0

0: 无影响

1: 写入 1 会将 WKUPFn 唤醒标志清零（该位由硬件清零）

9.11.7 PWR 唤醒标志寄存器 (PWR_WKUPFR)

PWR wakeup flag register

偏移地址：0x024

复位值：0x0000 0000

不通过从待机模式唤醒复位，而是通过任何应用复位（如 NRST、IWDG 等）来复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WKUPF6	WKUPF5	WKUPF4	WKUPF3	WKUPF2	WKUPF1
										r	r	r	r	r	r

位 31:6 保留，必须保持复位值。

位 5:0 **WKUPF[6:1]**：使能前 WKUPn 引脚的唤醒标志，n 的范围为 [6:1]。
该位由硬件置 1，只能通过 NRST 复位或通过将 *PWR 唤醒控制寄存器 (PWR_WKUPCR)* 中的 WKUPCn 位置 1 清零。
0: 未发生唤醒事件
1: 从 WKUPCn 引脚接收到唤醒事件

9.11.8 PWR MPU 唤醒使能寄存器 (PWR_MPUWKUPENR)

PWR MPU wakeup enable register

偏移地址：0x028

复位值：0x0000 0000

不通过从待机模式唤醒复位，而是通过任何应用复位（如 NRST、IWDG 等）来复位。

写入该寄存器时，需要访问 6 个等待状态。

当通过*第 10.8.1 节：RCC 安全配置寄存器 (RCC_SECCFGR)* 中的 PWRSEC 寄存器位使能时，该寄存器可提供写访问安全性。使能安全性时，非安全写访问将被丢弃，并产生总线错误。安全和非安全读访问被允许并返回寄存器值。

当寄存器写周期中发生系统复位时，写入的数据无法得到保证。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WKUPEN6	WKUPEN5	WKUPEN4	WKUPEN3	WKUPEN2	WKUPEN1
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持复位值。

位 5:0 **WKUPEN[6:1]**：使能唤醒 WKUPn 引脚和 MPU 的中断，n 的范围为 [6:1]
各个位由软件置 1 和清零。
当 RCC 中使能了 PWRSEC 时，这些位只能由安全访问写入，非安全写入将被丢弃。
当 RCC 中禁止了 PWRSEC 时，这些位可以由安全和非安全访问写入。
0: WKUPn 引脚上的事件不会将系统从待机和 Stop 模式唤醒，也不会产生 MPU 中断。
1: WKUPn 引脚上的上升沿或下降沿将系统从待机和 Stop 模式唤醒，并产生 MPU 中断。

9.11.9 PWR IP 版本寄存器 (PWR_VERR)

PWR IP version register

偏移地址：0x3F4

复位值：0x0000 0020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本号 (Major revision number)

位 3:0 **MINREV[3:0]**：次版本号 (Minor revision number)

9.11.10 PWR IP 标识寄存器 (PWR_ID)

PWR IP identification register

偏移地址：0x3F8

复位值：0x0001 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **IPID[31:0]**：IP 标识 (IP identification)

9.11.11 PWR 大小 ID 寄存器 (PWR_SID)

PWR size ID register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识 (Size identification)

9.11.12 PWR 寄存器映射

表 47. 电源控制寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**PWR_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALS[1:0]		AVDEN	Res.	Res.	Res.	Res.	Res.	Res.	MPU_RAM_LOWSPEED	DBP	PLS[2:0]			PVDEN	STOP2	LVDS	LPCFG	LPDS
	Reset value														0	0	0							1	0	0	0	0	0	0	0	0	0
0x004	**PWR_CSR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AVDO	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PVDO	Res.	Res.	Res.	Res.
	Reset value																0												0				
0x008	**PWR_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEMPH	TEMPL	VBATH	VBATL	Res.	Res.	Res.	BRRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MONEN	Res.	Res.	Res.	BREN
	Reset value									0	0	0	0				0												0				0
0x00C	**PWR_CR3**	REG11RDY	REG11EN	REG18RDY	REG18EN	Res.	USB33RDY	Res.	USB33DEN	VDDSD2VALID	VDDSD1VALID	POPL[4:0]					VDDSD2RDY	VDDSD2DEN	VDDSD1RDY	VDDSD1DEN	DDRRETEN	DDRSRDIS	DDRSREN	VBRS	VBE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	1	0	1		0		0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0								
0x010	**PWR_MPUCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STANDBYWFIL2	Res.	Res.	Res.	Res.	Res.	CSSF	Res.	SBFMPU	SBF	STOPF	Res.	Res.	Res.	Res.	PDDS
	Reset value																	0						0		0	0	0					0
0x014-0x01C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 47. 电源控制寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x020	**PWR_WKUPCR**	Res.	Res.	Res.	Res.	WKUPPUPD6[1:0]		WKUPPUPD5[1:0]		WKUPPUPD4[1:0]		WKUPPUPD3[1:0]		WKUPPUPD2[1:0]		WKUPPUPD1[1:0]		Res.	Res.	WKUPP6	WKUPP5	WKUPP4	WKUPP3	WKUPP2	WKUPP1	Res.	Res.	WKUPC6	WKUPC5	WKUPC4	WKUPC3	WKUPC2	WKUPC1
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0			0	0	0	0	0	0
0x024	**PWR_WKUPFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WKUPF6	WKUPF5	WKUPF4	WKUPF3	WKUPF2	WKUPF1
	Reset value																											0	0	0	0	0	0
0x028	**PWR_ MPUWKUPENR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WKUPEN6	WKUPEN5	WKUPEN4	WKUPEN3	WKUPEN2	WKUPEN1
	Reset value																											0	0	0	0	0	0
0x02C-0x3F0	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x3F4	**PWR_VERR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	0	1	0	0	0	0	0
0x3F8	**PWR_ID**	IPID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x3FC	**PWR_SID**	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的 第 2.5 节*。

10 复位和时钟控制 (RCC)

RCC 模块负责管理整个电路的时钟和复位信号的生成。

本节经常会涉及*第 9.5.1 节：工作模式*中定义的工作模式。

10.1 RCC 主要特性

- 复位部分：
  - 生成本地复位和系统复位信号
  - 双向引脚复位 (NRST)，允许对外部设备或电路进行复位
- 时钟生成部分：
  - 为整个系统生成并分配时钟
  - 4 个独立的 PLL：
    - 整数或小数模式
    - 展频功能，可减少 EMI 峰值数量
    - 可实时更改 PLL 小数分频比
  - 智能时钟门控，可降低功耗
  - 2 个外部振荡器：
    - HSE，支持多种晶振：8 到 48 MHz
    - LSE，用于 32.768 kHz 晶振
  - 3 个内部振荡器：HSI、CSI 和 LSI
  - 针对外部设备的缓冲时钟输出
  - 生成 3 条中断线：
    - 从 CStop 唤醒 MPU 的唤醒中断线
    - 用于其他安全和相应非安全事件（振荡器就绪、PLL 就绪、检测到 LSE 时钟故障）的全局安全和非安全中断线
- 支持 TrustZone：
  - 可编程的安全和非安全 RCC 子功能
  - TrustZone 可感知 AHB 从设备端口，防止任何 RCC 安全子功能控制（寄存器、寄存器字段）受到非安全写访问。
  - 根据相应外设的安全状态，对每个外设的复位和时钟控制进行保护

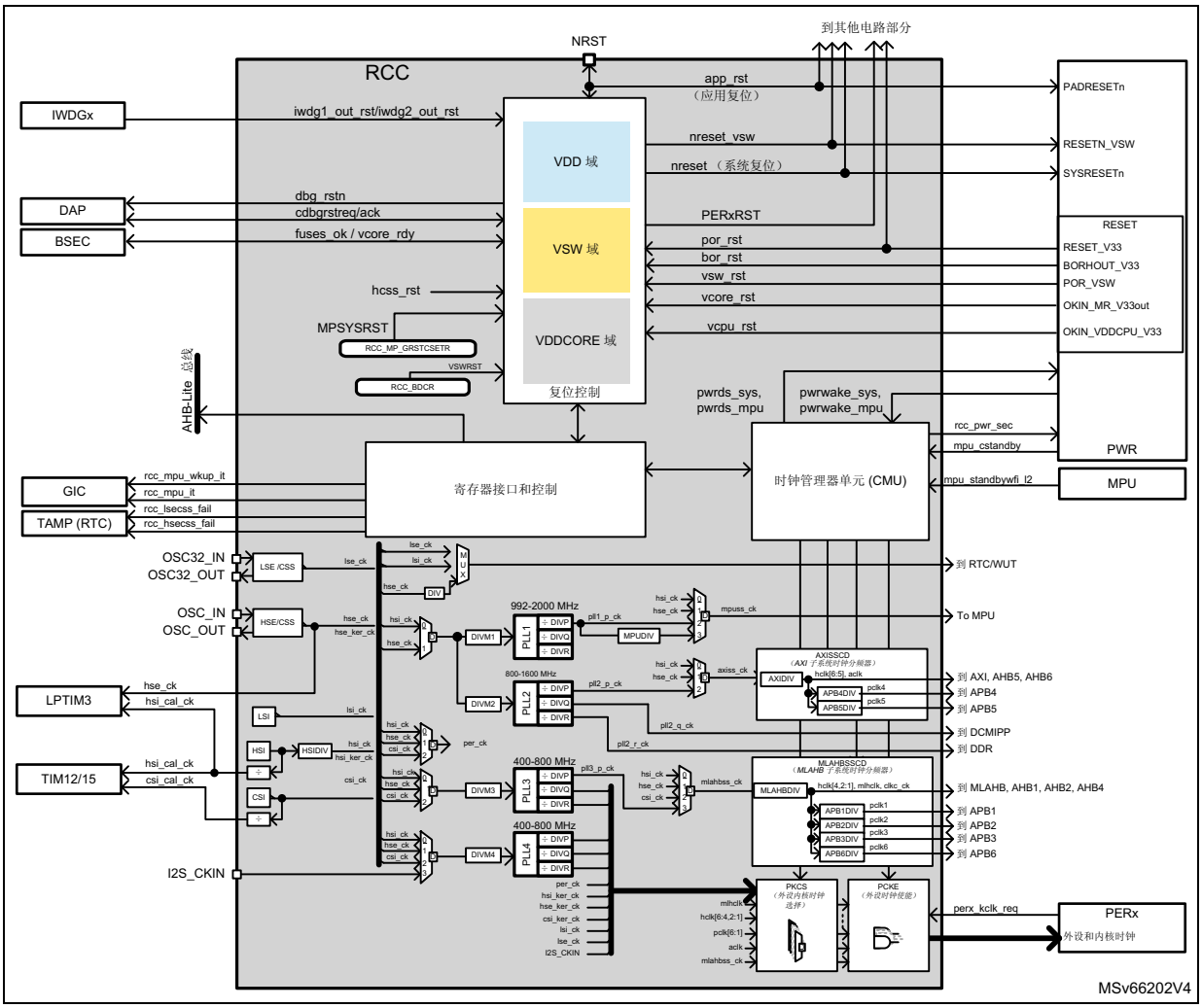
10.2 RCC 框图

图 46 所示为 RCC 的简图，同时显示了与其他模块交换的关键信号。RCC 中有三个电压域：V_{DD} 域（蓝色）、V_{SW} 域（黄色）和 V_{DDCORE} 域（灰色）。

对于复位部分，PWR 模块中的模拟比较器提供所有与电源相关的复位信号：**por_rst**、**bor_rst**、**vcore_rst**、**vcpu_rst** 和 **vsw_rst**。一个复位源 NRST 来自一个引脚。RCC 还接收来自看门狗和寄存器等各种其他来源的复位信号。所有这些复位信号均由 RCC 处理，以正确复位系统的所有部分。

对于时钟生成部分，RCC 根据寄存器设置和处理器模式处理所有时钟的生成和门控。请注意，RCC 由 hclk4 时钟驱动。

图 46. RCC 框图



RCC 还与 PWR 交换一些信号，以对系统进入和退出 Stop 模式和待机模式加以控制。

10.2.1 RCC 引脚概述

表 48 列出了连接到引脚或焊球的 RCC 信号。

表 48. 引脚概述

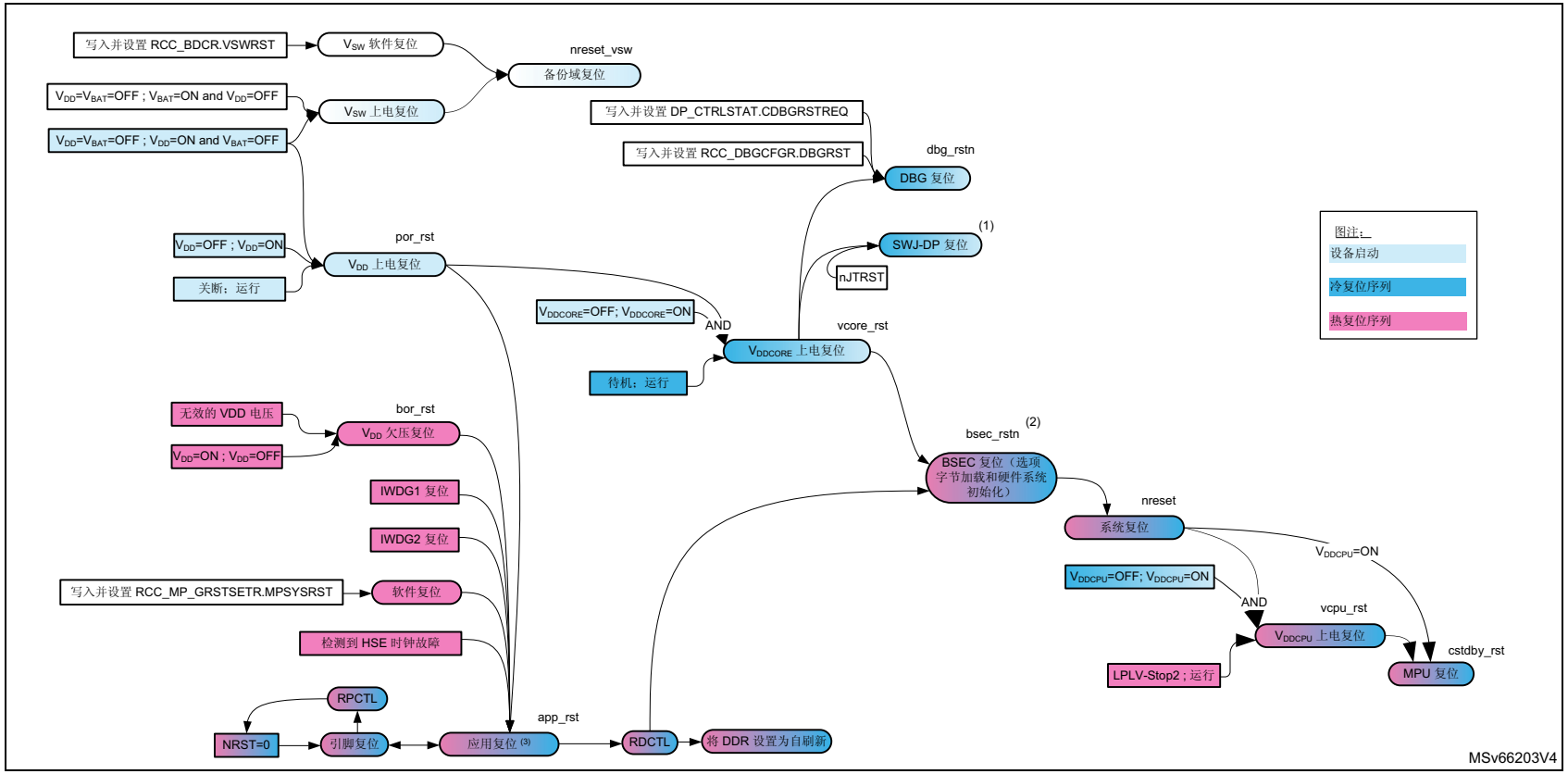
引脚名称	I/O	说明
NRST	I/O	外部（应用）复位，可用于为外部设备提供复位信号
OSC32_IN	I	32.768 kHz 振荡器输入或外部时钟输入
OSC32_OUT	O	32.768 KHz 振荡器输出
OSC_IN	I	HSE 振荡器输入或外部时钟输入
OSC_OUT	O	HSE 振荡器输出
MCO1	O	可提供给外部设备的输出时钟 1
MCO2	O	可提供给外部设备的输出时钟 2
I2S_CKIN	I	数字音频接口的外部内核时钟输入：SPI/I2Sx (x=1,2,3,4)、SAIx (x=1,2) 和 DFSDM
ETH[2:1]_TX_CLK	I	以太网 PHY 提供的用于 MII 的外部时钟 TX
ETH[2:1]_RX_CLK/ ETH[2:1]_REF_CLK	I	以太网 PHY 提供的用于 MII、RMII 和 RGMII 的外部时钟 RX
ETH[2:1]_CLK	O	输出到外部以太网 PHY 的时钟
ETH[2:1]_CLK125	I	以太网 PHY 提供的用于 RGMII 的外部时钟 RX
ETH[2:1]_RGMII_GTX_CLK	O	用于 RGMII 的外部以太网 PHY 的时钟输出

10.3 RCC 功能描述——复位部分

10.3.1 复位概述

本节使用的复位一词在下一页的*图 47* 进行了总结。该复位图总结并描述了不同的复位及复位源，包括复位之间的相关性，复位与设备电源模式和电压跳变的相关性。

图 47. RCC 复位图



1. 包括 JTAG 寄存器 BSEC_JTAGIN、BSEC_JTAGOUT、BSEC_DENABLE 的复位。
2. 不包括 JTAG 寄存器 BSEC_JTAGIN、BSEC_JTAGOUT、BSEC_DENABLE 的复位。

*图 47：RCC 复位图*所示的系统复位 (**nreset**) 信号在热复位或软复位之后生成。

- 热复位发生于以下情况之一：
  - 引脚复位 (**NRST**)
  - 应用复位 (**app_rst**)，并非由 V_{DD} 上电复位 (**por_rst**) 所致。
- 冷复位发生在 V_{DDCORE} 上电复位 (**vcore_rst**) 和 BSEC 复位 (**bsec_rstn**) 之后（例如选项字节加载和硬件系统初始化）；系统复位之后是 V_{DDCPU} 上电复位 (**vcpu_rst**)。

当冷复位序列由主 V_{DD} 上电复位 (**por_rst**) 发起时，该复位序列也称为设备启动，或者关断到运行的电源模式跳变。

应用复位通过以下源之一产生：

- V_{DD} 上电复位 (**por_rst**)
- V_{DD} 欠压复位 (**bor_rst**)
- 通过独立看门狗 1 (**iwdg1_out_rst**) 复位
- 通过独立看门狗 2 (**iwdg2_out_rst**) 复位
- 当 Cortex-A7 (MPU) 通过写操作将 RCC_MP_GRSTECSETR.MPSYSRST 寄存器控制位置 1 时发生的软件复位
- 在 HSE (**hcss_rst**) 上检测到时钟故障（前提是时钟安全系统功能已激活）
- 引脚复位 (**NRST**)。

注： *可编程复位脉冲控制 (RPCTL) 允许应用定义 NRST 引脚生效的最小脉宽。如果禁止 RPCTL，复位 PAD 会嵌入一个脉冲延展器，可保证每个内部复位源的最小复位脉冲持续时间为 20 µs。应用复位和系统复位也受该功能的影响。有关 RPCTL 的详细信息，请参见第 10.3.8 节：复位脉冲控制 (RPCTL)。*

系统复位 (**nreset**) 是 MPU 复位 (**cstdby_rst**) 的来源。

V_{DDCORE} 上电复位还会使 DBG 模块 (**dbg_rstn**) 和 SWJ-DP JTAG 端口复位（可能通过 nJTRST 引脚在外部复位）。

当系统处于待机模式时，V_{DDCORE} 关闭，但 V_{DD} 仍然存在。当系统退出待机模式进入运行模式时，会发生冷启动序列：将产生 V_{DDCORE} 上电复位 (**vcore_rst**)、DBG 复位 (**dbg_rst**)、BSEC 复位 (**bsec_rstn**) 和系统复位 (**nreset**)，以及 V_{DDCPU} 上电复位 (**vcpu_rst**) 和 MPU 复位 (**cstdby_rst**)。系统通过 WKUPx 引脚或 RTC 事件（时间戳、警报或 Tamper）退出待机模式。由于 V_{DD} 欠压复位、IWDG1/2 复位或引脚复位而产生的应用复位，也会使系统退出待机模式。

注： *不建议使 NRST 引脚处于未连接状态。如果未使用该引脚，则可通过一个容值介于* 10 nF 到 100 nF *之间的电容将其接地。*

当系统处于 LPLV-Stop2 模式时，V_{DDCPU} 关闭，但 V_{DD} 和 V_{DDCORE} 仍然存在。当系统退出 LPLV-Stop2 模式时，仅产生 V_{DDCPU} 上电复位 (**vcpu_rst**) 和 MPU 复位 (**cstdby_rst**)。系统通过使能的 EXTI 中断退出 LPLV-Stop2 模式。

注： *MPU 复位时，将产生 MPU 处理器复位。*

图 47 的最后介绍了备份域复位，这部分内容在*第 10.3.4 节*中也进行了阐述。

10.3.2 电源复位

PWR 模块为 RCC 提供了一些复位信号：

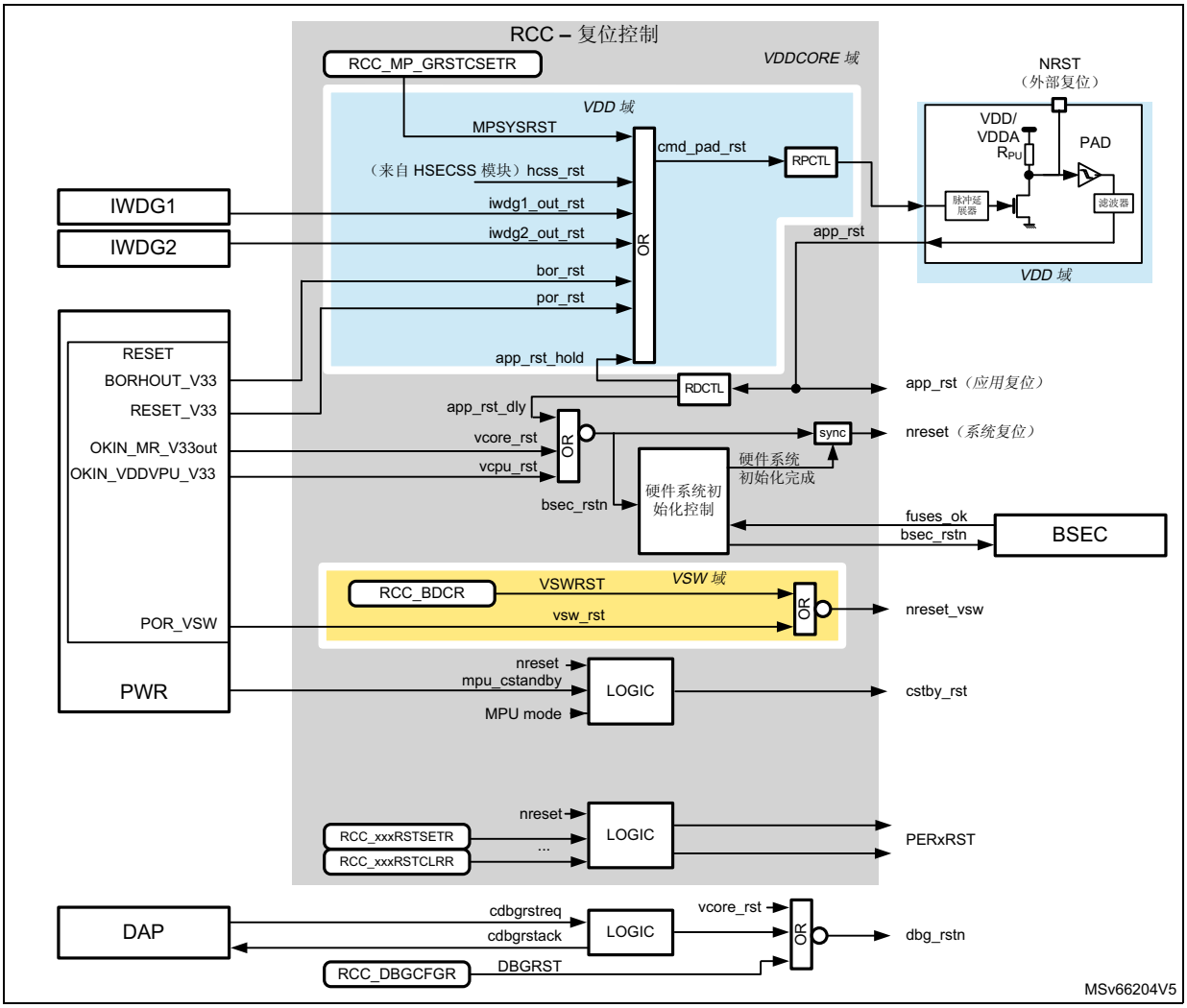
- **por_rst** 信号，当 V_{DD} 供电电压低于 V_{POR} 阈值时生效。
- **bor_rst** 信号，当 V_{DD} 供电电压低于 V_{BOR} 阈值时生效。
- **vcore_rst** 信号，当 V_{DDCORE} 供电电压低于 V_{DDCORE} 阈值时生效。
- **vsw_rst** 信号，当 V_{SW} 供电电压低于预期阈值时生效。
- **vcpu_rst** 信号，当 V_{DDCPU} 供电电压低于 V_{DDCPU} 阈值时生效。

如*图 48：复位电路概述*所示，PWR 模块提供的信号可指示 V_{DDCORE} 有效 (OKIN_MR_V33out) 以及 V_{DDCPU} 有效 (OKIN_VDDCPU_V33)，RCC 使用这些信号来触发硬件系统初始化序列。有关详细信息，请参见*第 3 章：启动、安全性和 OTP 控制 (BSEC)*。

有关各个复位的范围的详细信息，请参见*表 49：复位覆盖范围汇总*。

有关产生开/关机复位的详细信息，请参见*第 9 章：电源控制 (PWR)*。

图 48. 复位电路概述



10.3.3 看门狗复位

两个看门狗可供应用使用，即 IWDG1 和 IWDG2。发生超时时，IWDG1 和 IWDG2 会产生系统复位。

10.3.4 备份域复位

如*图 47：RCC 复位图*所示，在发生以下情况后，RCC 会产生备份域复位：

- V_{SW} 域上电复位：
  - 该复位由 V_{SW} 域中 PWR 控制器的子部分驱动。
  - 如果 V_{DD} 或 V_{BAT} 已断电，则在它们上电时会发生 V_{SW} 上电复位。
  - 这会将 V_{SW} 域复位。就 RCC 而言，这意味着它会复位从 LSE 源时钟到 RTC 内核时钟的时钟数据路径，并会复位 V_{SW} 域中的 RCC 寄存器（即 RCC_BDCR 寄存器）。BKPSRAM 不受影响。
- 备份域软件复位：通过写操作将 RCC_BDCR.VSWRST 寄存器控制位置 1 时。
  - 这会将 V_{SW} 域复位。就 RCC 而言，这意味着它会复位从 LSE 源时钟到 RTC 内核时钟的时钟数据路径，并会复位 RCC_BDCR 寄存器（VSWRST 除外）。在这种情况下，BKPSRAM 的内容不再有效。

请参见*第 1.3.3 节：*其他相关信息，请参见 PWR 模块的*备份域*一节。

10.3.5 Coresight 调试复位

Coresight 调试组件（包括 MPU 调试部分）可以通过三种不同的方式进行复位：

- 将调试端口的控制/状态寄存器的 CDBGRSTREQ 位置“1”，以通过 DAP 进行复位。将 CDBGRSTREQ 位置“1”可将连接到 RCC 的调试复位请求信号 CDBGRSTREQ 置为有效。这样，RCC 就会激活 **dbg_rstn**，握手信号 **cdbgrstack** 即被提供给 DAP 请求。只要 **cdbgrstreq** 为“1”，**dbg_rstn** 就处于激活状态。
- 通过应用将 RCC_DBGCFGR.DBGRST 控制位置“1”。当 DBGRST 置“1”时，**dbg_rstn** 复位激活，当 DBGRST = “0”时，将释放 **dbg_rstn** 复位信号。
- 发生 V_{DDCORE} 上电复位时 (**vcore_rst**)。该复位在 V_{DD} 上电复位后或产品退出待机状态时激活。

有关详细信息，请参见*第 10.3.5 节：Coresight 调试复位*。

10.3.6 选项字节加载

每当应用复位 (**app_rst**)或 V_{DDCORE} 上电复位 (**vcore_rst**) 激活时，即会触发选项字节配置序列。

10.3.7 按外设复位

只要有外设发出复位请求，应用就可以单独将其复位。实现方法是分别将寄存器 RCC_xxxxRSTSETR 和 RCC_xxxxRSTCLRR 置 1 和清零，其中“xxxx”是 AHBx/APBx 总线名称，具体取决于总线架构和外设映射。

请注意，当指定外设为安全外设或配置为安全外设时，只有安全访问可以控制该外设的软件复位（即通过向 RCC_xxxxRSTSETR 寄存器相应位写入“1”将复位信号置为有效，通过向相应的 RCC_xxxxRSTSETR 寄存器写入“1”来释放复位信号）。

当外设需要复位时，需要先将相应的外设使能位置“1”来使能其外设时钟。有关详细信息，请参见*第 10.4.11 节：外设时钟门控控制*。

10.3.8 复位脉冲控制 (RPCTL)

RPCTL 允许应用控制 NRST 引脚的最短激活时间。当外部设备请求特定的 NRST 激活持续时间时，这一功能特别有用。IWDG1/2 产生的复位脉冲对外部设备而言可能过短。

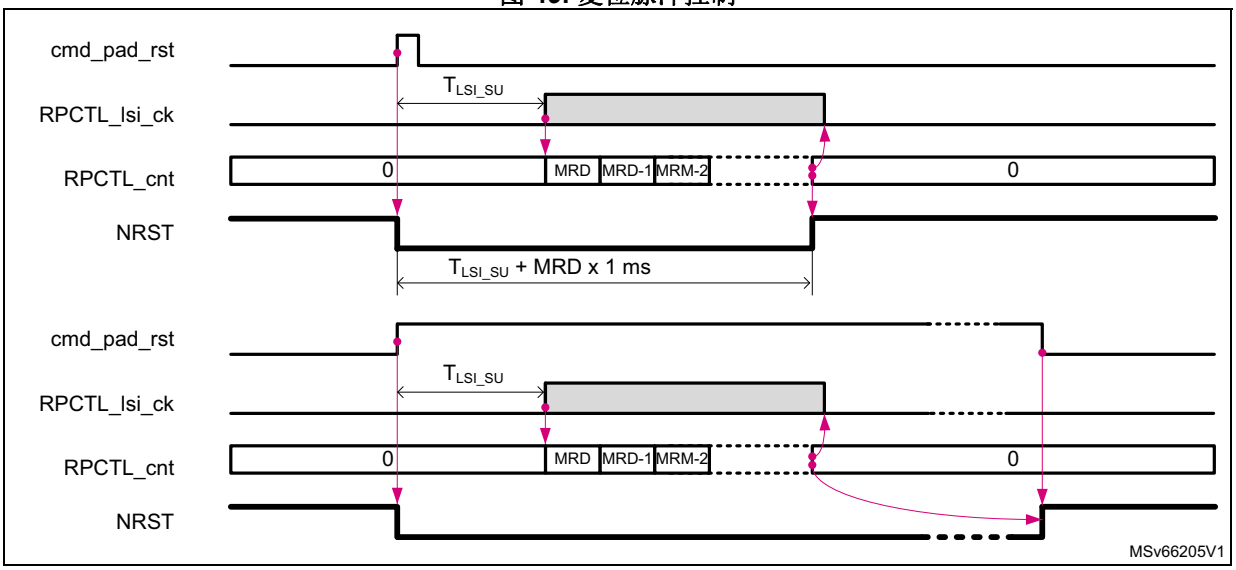
RPCTL 在 V_{DD} 域内，仅在上电复位后才会复位。

RPCTL 由 *RCC 复位持续时间和 LSI 控制寄存器 (RCC_RDLSICR)* 中的 MRD[4:0] 位控制。

- 如果 MRD = “0”，则旁路 RPCTL，**cmd_pad_rst** 信号直接驱动复位引脚。这种情况下，最短激活时间由嵌入到复位引脚中的脉冲延展器指定（通常为 20 µs）。
- 如果 MRD 不为 “0”，则当复位激活时（**cmd_pad_rst** 的上升沿），NRST 立即被驱动至低电平并且持续时间至少为 MRD[4:0] 指定的值。如果 **cmd_pad_rst** 信号的激活时间比 MRD[4:0] 编程设定值更长，则复位持续时间不受 RPCTL 影响。通过 MRD[4:0] 位可在 1 到 31 ms 之间调整最短激活时间。

  请注意，RPCTL 使用 LSI 时钟来控制复位宽度。当 **cmd_pad_rst** 变为高电平时，RPCTL 请求 LSI 时钟以控制复位持续时间。如果 LSI 未由其他功能使能，可能需要几微秒的时间等待 LSI 就绪 (T_{LSI_SU})。*图 49* 所示为简化的时序图。

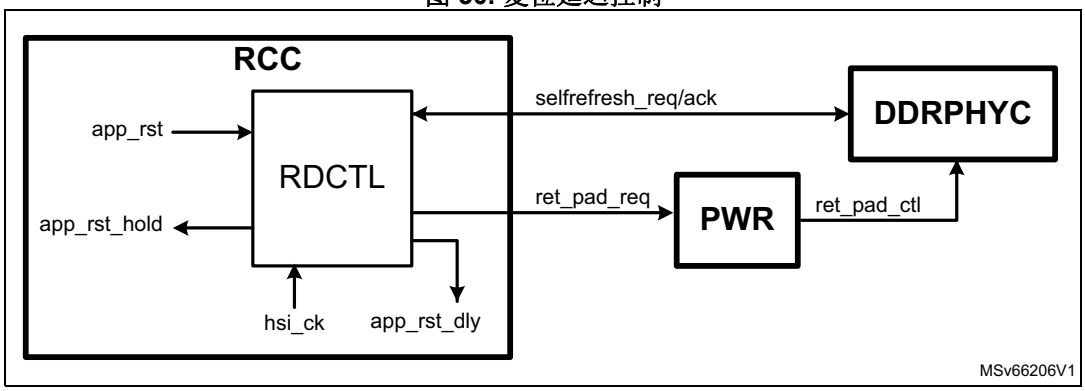
图 49. 复位脉冲控制



10.3.9 复位延迟控制 (RDCTL)

如果外部 DDR 设备需要在产生电路系统复位 (**nreset**) 之前设置为自刷新，则必须使用 RDCTL 模块。

图 50. 复位延迟控制



如果使能，RDCTL 执行以下操作：

如果出现 **app_rst**：

- RDCTL 将 **app_rst_dly** 保持在非工作状态，并将 **app_rst_hold** 信号置为有效以使 **cmd_pad_rst** 保持工作状态，
- RDCTL 激活 HSI 振荡器，并在需要时等待，直到 **hsi_ck** 时钟可用，
- 如果系统处于 Stop、LP-Stop、LPLV-Stop 或 LPLV-Stop2 模式，则 **app_rst_dly** 信号将被激活，**app_rst_hold** 信号被释放。
- 否则，如果系统处于运行模式，则：
  - 超时计数器开始倒计时，
  - RDCTL 向 DDRPHYC 发出将外部 DDR 设备设置为自刷新模式的请求。
  - 一旦 DDRPHYC 确认外部 DDR 设备处于自刷新模式，RDCTL 便会执行以下操作：
    ——冻结超时计数器，
    ——向 PWR 发出将 DDR 引脚设置为保持模式的请求 (**ret_pad_req**)，
    ——激活 **app_rst_dly** 信号，
    ——释放 **app_rst_hold** 信号。
  - 如果在 RDCTL 收到外部 DDR 设备设置为自刷新确认之前超时计数器到期，**app_rst_dly** 信号将被激活，**app_rst_hold** 信号将被释放。

应用可以通过 *RCC 应用复位控制寄存器 (RCC_MP_APRSTCR)* 的 RDCTLEN 位使能或禁止 RDCTL。当 RDCTL 被禁止（RDCTLEN = “0”）时，信号 **app_rst_dly** 不会延迟，而是遵循 **app_rst** 的值，另外，不会生成 DDR 引脚保持请求。

小心： **app_rst** 置为有效时，RDCTLEN 位不得置“1”。最好在初始化阶段使能该功能。

应用还可以通过 *RCC 应用复位控制寄存器 (RCC_MP_APRSTCR)* 的 RSTTO[6:0] 位字段编程超时值。

可以使用以下公式来编程该超时值：

$$RSTTO \geq round\left(\frac{8192 \times T_{ck_ker_dphy}}{10^{-6}}\right)$$

其中 $T_{ck_ker_dphy}$ 表示提供给 DDRPHYC 的内核时钟周期。

小心： 如果 RDCTLEN =“1”，则不允许更改该位字段的值。
不建议将 RSTTO[6:0] 设置为“0”。

系统复位 (**nreset**) 后，应用可以通过检查 RSTTO[6:0] 来确认外部 DDR 设备是否已正确设置为自刷新。RSTTO[6:0] 给出了超时计数器的当前值。如果该值等于“0”，则意味着最后一次 RDCTL 请求外部 DDR 设备自刷新时发生了超时。

注： ***hsi_ck*** *的频率取决于 HSIDIV 的值。*

10.3.10 复位覆盖范围汇总

表 49 详细说明了复位源和复位域（以灰色突出显示）。每个列代表一个复位源。

注： *假设当 V_{DD} 无效时，V_{DDCORE} 和 V_{DDCPU} 也无效。请参见图 47：RCC 复位图。*

表 49. 复位覆盖范围汇总 (1)

复位影响的功能		复位源									
		por_rst (2)	vcore_rst (3)	app_rst (4)	nreset	vcpu_rst	cstby_rst	dbg_rstn	bsec_rstn	PERxRST	nreset_vsw
V_{DD} 域		x	-	-	-	-	-	-	-	-	-
V_{DDCORE} 域（特别是包括 AXI/AHB 互连）		x	x	-	-	-	-	-	-	-	-
DDR 接口		x	x	x	x	-	-	-	-	-	-
V_{DDCPU} 域		x	x	x	x	x	x	-	-	-	-
调试组件，包括 DBGMCU 和 MPU 内核中的调试（SWJ-DP 除外）		x	x	-	-	-	-	x	-	-	-
SWJ-DP		x	x	-	-	-	-	x	-	-	-
IWDG[2:1]		x	-	x	-	-	-	-	-	-	-
硬件系统初始化，包括选项字节重载。		x	x	x	-	-	-	-	x	-	-
RCC 寄存器	RCC_BR_RSTSCLRR	x	-	-	-	-	-	-	-	-	-
	RCC_MP_IWDGFZSETR	x	x	x	x	-	-	-	-	-	-
	RCC_RDLSICR	x	-	-	-	-	-	-	-	-	-
	RCC_BDCR（VSWRST 除外）	-	-	-	-	-	-	-	-	-	x
	RCC_BDCR.VSWRST	-	-	-	-	-	-	-	-	-	-
	其他 RCC 寄存器	x	x	x	x	-	-	-	-	-	-
PWR 寄存器	PWR_CR2	-	-	-	-	-	-	-	-	-	x
	PWR_CR3	x	-	-	-	-	-	-	-	-	-
	PWR_MPUCR 这些寄存器的各个位的复位条件不同， 有关详细信息，请参见 PWR 部分。	x	x	x	x	-	-	-	-	-	-
	PWR_WKUPCR、PWR_WKUPFR、PWR_MPUWKUPENR	x	-	x	-	-	-	-	-	-	-
	其他寄存器	x	x	x	x	-	-	-	-	-	-
RTC		-	-	-	-	-	-	-	-	-	x
BKPSRAM V_{SW} 域复位后，BKPSRAM 的备份稳压器被禁止，该功能通过 PWR 模块 BREN 和 RREN 位控制。 如果 **nreset_vsw** 复位的原因是 V_{SW} 电压过低，BKPSRAM 的内容将会丢失。		-	-	-	-	-	-	-	-	-	x
其他外设		x	x	x	x	-	-	-	-	x	-

1. “x”表示该功能由对应的复位源复位，“-”表示该功能不会被对应的复位源复位。
2. 当 **por_rst** 置为有效时，**app_rst**、**vcore_rst**、**bsec_rst**、**nreset**、**vcpu_rst**、**cstby_rst** 和 **dbg_rst** 也将置为有效。
3. 当 **vcore_rst** 置为有效时，**bsec_rst**、**nreset**、**vcpu_rst**、**cstby_rst** 和 **dbg_rst** 也将置为有效。
4. 当 **app_rst** 置为有效时，**bsec_rst**、**nreset**、**vcpu_rst** 和 **cstby_rst** 也将置为有效。

RCC 寄存器复位

表 50 列出了不在 V_{DDCORE} 电压域中从而具有特定复位条件的特定寄存器及其位字段。其他 RCC 寄存器都是通过系统复位 (**nreset**) 来复位。

表 50. 具有特定复位和特定电压的 RCC 寄存器

RCC 寄存器（和位字段）名称	电压域	复位条件	备注
RCC_RDLSICR	V_{DD}	por_rst	当 V_{DD} 打开时复位。 V_{DD} 上电复位会发起设备启动序列，其中包括系统复位。
RCC_BR_RSTSCLRR			
RCC_BDCR（VSWRST 除外）	V_{SW}	nreset_vsw	通过备份域复位来复位，通过 V_{SW} 上电复位或通过写操作将 RCC_BDCR.VSWRST 置 1 启动
RCC_BDCR.VSWRST	V_{SW}	RCC_BDCR.VSWRST=0	通过明确写操作将 RCC_BDCR.VSWRST 清零来复位
其他	V_{DDCORE}	nreset	系统复位

10.3.11 复位源标识

RCC 提供三个寄存器来标识复位的根本原因：

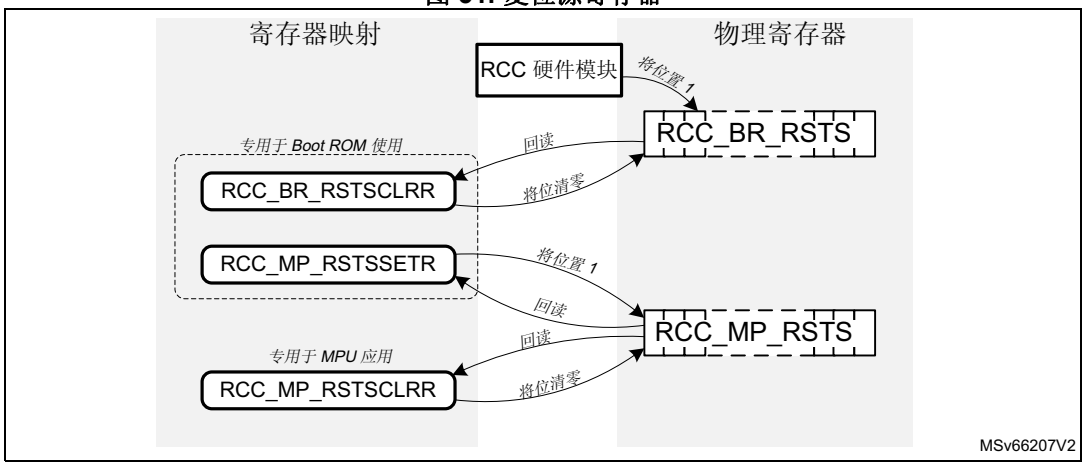
- *RCC Boot ROM 复位状态清除寄存器 (RCC_BR_RSTSCLRR)*，
- *RCC 复位状态设置寄存器 (RCC_MP_RSTSSETR)*，
- *RCC 复位状态清除寄存器 (RCC_MP_RSTSCLRR)*

寄存器 RCC_BR_RSTSCLRR 和 RCC_MP_RSTSSETR 专供 Boot ROM 使用，不得由应用使用。

寄存器 RCC_MP_RSTSCLRR 专用于 MPU 应用。

下图详细说明了复位源寄存器的工作方式。

图 51. 复位源寄存器



Boot ROM 使用 RCC_BR_RSTSCLRR 寄存器来标识复位的根本原因，并定义必须执行的启动序列。此外，每次系统复位、上电复位、退出待机模式或 LPLV-Stop2 模式后，Boot ROM 代码都会更新 RCC_MP_RSTSCLRR 寄存器。更准确地说，Boot ROM 使用 RCC_MP_RSTSSETR 寄存器将 RCC_BR_RSTSCLRR 的内容以及 SBF 的值和 SBF_MPU 标志复制到 RCC_MP_RSTSCLRR 寄存器中。

随后 MPU 可以通过读取 RCC_MP_RSTSCLRR 寄存器来检查并清除复位源。

表 51 按照产生复位的条件列出了 RCC_BR_RSTSCLRR 和 RCC_MP_RSTSCLRR 寄存器的状态位的行为。

表 51. 复位源标识（RCC_MP/BR_RSTSCLRR 和 PWR）(1)

#	产生复位的条件	HCSSRSTF	IWDG2RSTF	IWDG1RSTF	CSTDBYRSTF (2)	STDBYRSTF (2)	STP2RSTF (2)	VCPURSTF	MPSYSRSTF	PORRSTF	BORRSTF	VCORERSTF	PADRSTF
1	*V_{DD} 上电复位 (**por_rst**)*	0	0	0	0	0	0	**1**	0	**1**	0	**1**	**1**
2	*欠压复位 (**bor_rst**)*	0	0	0	0	0	0	0	0	0	**1**	0	**1**
3	*来自 NRST 的引脚复位*	0	0	0	0	0	0	0	0	0	0	0	**1**
4	*MPU 产生的系统复位 (MPSYSRST)*	0	0	0	0	0	0	0	**1**	0	0	0	**1**
5	*系统退出待机模式*	0	0	0	**1**	**1**	0	0	0	0	0	0	0
6	*系统退出 LPLV-Stop2 模式*	0	0	0	**1**	0	**1**	0	0	0	0	0	0
8	*MPU 产生的 MPU 处理器 0 复位*	0	0	0	0	0	0	0	0	0	0	0	0
9	*IWDG1 复位 (**iwdg1_out_rst**)*	0	0	**1**	0	0	0	0	0	0	0	0	**1**
10	*IWDG2 复位 (**iwdg2_out_rst**)*	0	**1**	0	0	0	0	0	0	0	0	0	**1**
11	*V_{DDCORE} 故障导致的复位*	0	0	0	0	0	0	0	0	0	0	**1**	0
12	*HSE 时钟故障导致的复位*	**1**	0	0	0	0	0	0	0	0	0	0	**1**
13	*V_{DDCPU} 上电复位*	0	0	0	0	0	0	**1**	0	0	0	0	0

1. 位状态“1”以灰色突出显示。
2. 这些位在 RCC_MP_RSTSCLRR 和 RCC_MP_RSTSSETR 寄存器中。Boot ROM 将 SBF 位复制到 STDBYRSTF 中，将 SBF_MPU 位复制到 CSTDBYRSTF 中。有关 SBF 和 SBF_MPU 的介绍，请参见 *第 1.9.5 节：PWR MPU 控制寄存器 (PWR_MPUCR)*。

10.3.12 启动、冷/热复位和唤醒序列

*图 47：RCC 复位图*对设备启动序列、冷复位序列（从待机模式的启动序列）、热复位（因 V_{DD} 上电复位而引起的应用复位的启动序列）以及从任何 Stop 模式唤醒进行了说明。

有关详细的时序图，请参见*第 9.6 节：PWR 低功耗工作模式*。

从产品退出低功耗模式到 MPU 能够执行代码的时间间隔取决于系统状态及其配置。*图 52* 所示为最常见的示例。

设备启动序列（上电启动序列）

图 52 所示的上电启动序列给出了上电序列的几个最重要阶段。由于电路未通电，因此这是最长的序列。请注意，无论是否存在 VBAT，该序列都保持不变。

冷复位序列（系统待机模式的启动序列）

从系统待机模式唤醒时，由于 V_{DD} 未关闭，稳定时间会更短，因为参考电压已经稳定。由于 V_{DDCORE} 已关闭，无法跳过 HSI 的重启延迟和选项字节重载。

热复位（因 V_{DD} 上电复位引起的应用复位的启动序列）

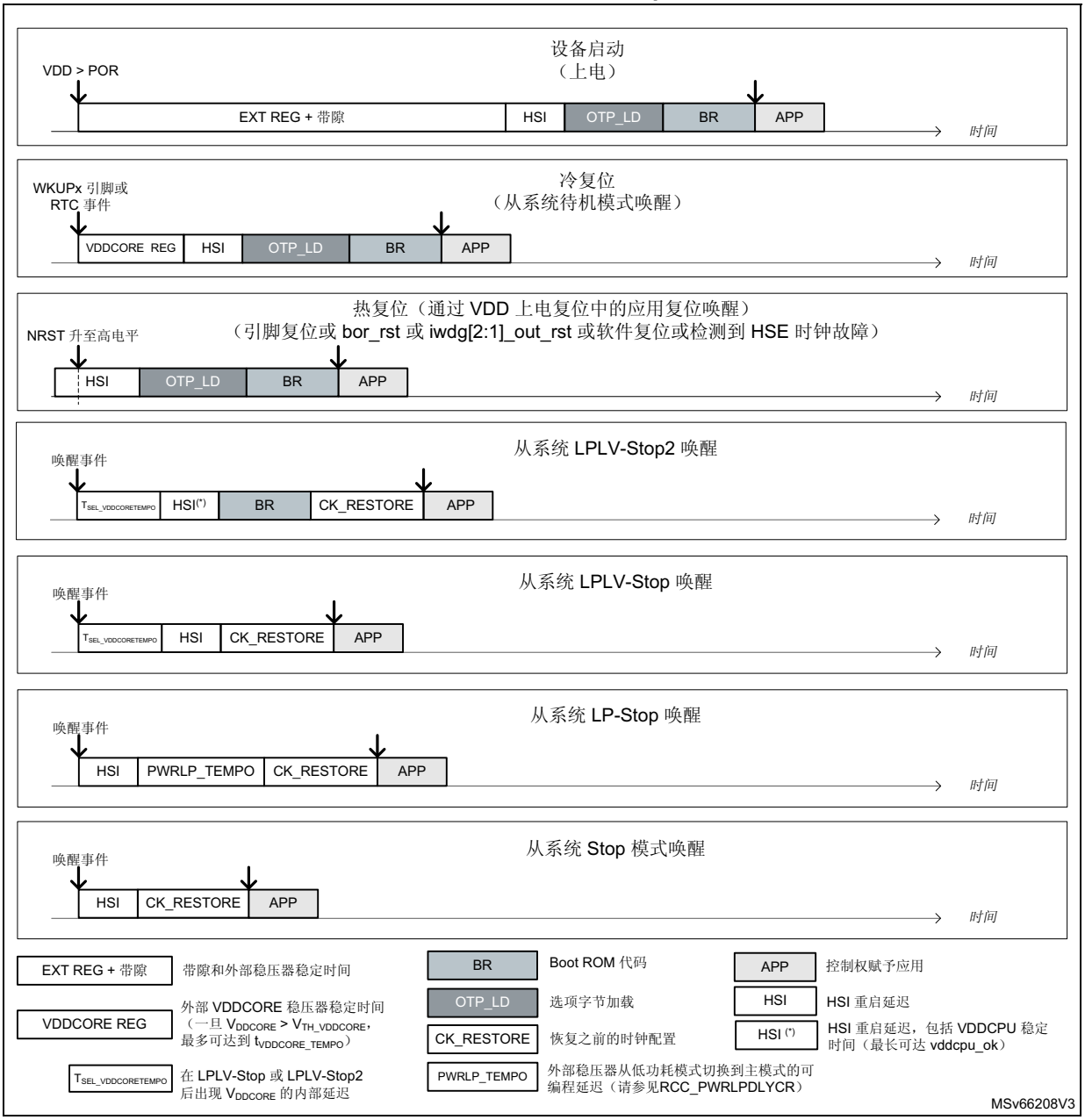
图 52 所示的热复位序列给出了该启动序列的一些最重要步骤。请注意，在这种情况下，V_{DD}/V_{DDA} 及 V_{DDCORE} 并未关闭。由于参考电压已稳定，稳定时间变短。如果在发生应用复位时未使能 HSI，则可能需要 HSI 重启延迟，否则会跳过此重启延迟阶段。将执行选项字节重载。

系统 Stop 模式的唤醒序列：Stop、LP-Stop、LPLV-Stop 和 LPLV-Stop2

从系统的一种 Stop 模式重新启动时，将执行自动时钟恢复序列，以恢复时钟（因为是针对运行模式定义的时钟），然后再将控制权交还给应用。具有以下特点：

- 与 Stop 模式相比，从 LP-Stop 模式唤醒会增加一个编程延迟（请参见下图中的 PWRLP_TEMPO 和相关的 RCC_PWRLPDLYCR 寄存器），以使外部稳压器从低功耗模式切换到主模式；
- 与从 LPLV-Stop 唤醒相反，在 LPLV-Stop2 之后，由于 V_{DDCPU} 已关闭，增加了一段 V_{DDCPU} 稳定时间和 Boot ROM 代码延迟。

图 52. 启动、冷/热复位以及从任何 Stop 模式唤醒的序列



10.4 RCC 功能描述——时钟部分

在时钟发生器的选择上，RCC 为应用提供了高度的灵活性：

- HSI 的时钟：高速内部振荡器（约 8、16、32 和 64 MHz）
- HSE 的时钟：高速外部振荡器（8 到 48 MHz）
- LSE 的时钟：低速外部振荡器 (32.768 kHz)
- LSI 的时钟：低速内部振荡器（约 32 kHz）
- CSI 的时钟：低功耗内部振荡器（约 4 MHz）

可帮助应用灵活地选择适合的 MPU 和外设时钟。

每个时钟源都可以单独打开或关闭，以优化功耗。

主要有 4 个时钟路径：

- MPU 时钟
- AXI 子系统 (AXI-SS) 时钟，包括 DDR 接口
- MLAHB（多层 AHB）总线矩阵时钟
- 外设内核时钟

RCC 提供多达 4 个 PLL，每个 PLL 均可采用整数或小数分频比配置。

- PLL1 专用于 MPU 时钟
- PLL2 提供：
  - 用于 AXI-SS 的时钟（包括 APB4、APB5、AHB5 和 AHB6 总线桥）
  - 用于 DCMIPP 的时钟
  - 用于 DDR 接口的时钟
- PLL3 提供：
  - 用于 MLAHB 及其子系统的时钟（包括 APB1、APB2、APB3、APB6、AHB1、AHB2 和 AHB4）
  - 用于外设的内核时钟
- PLL4 专用于生成不同外设的内核时钟

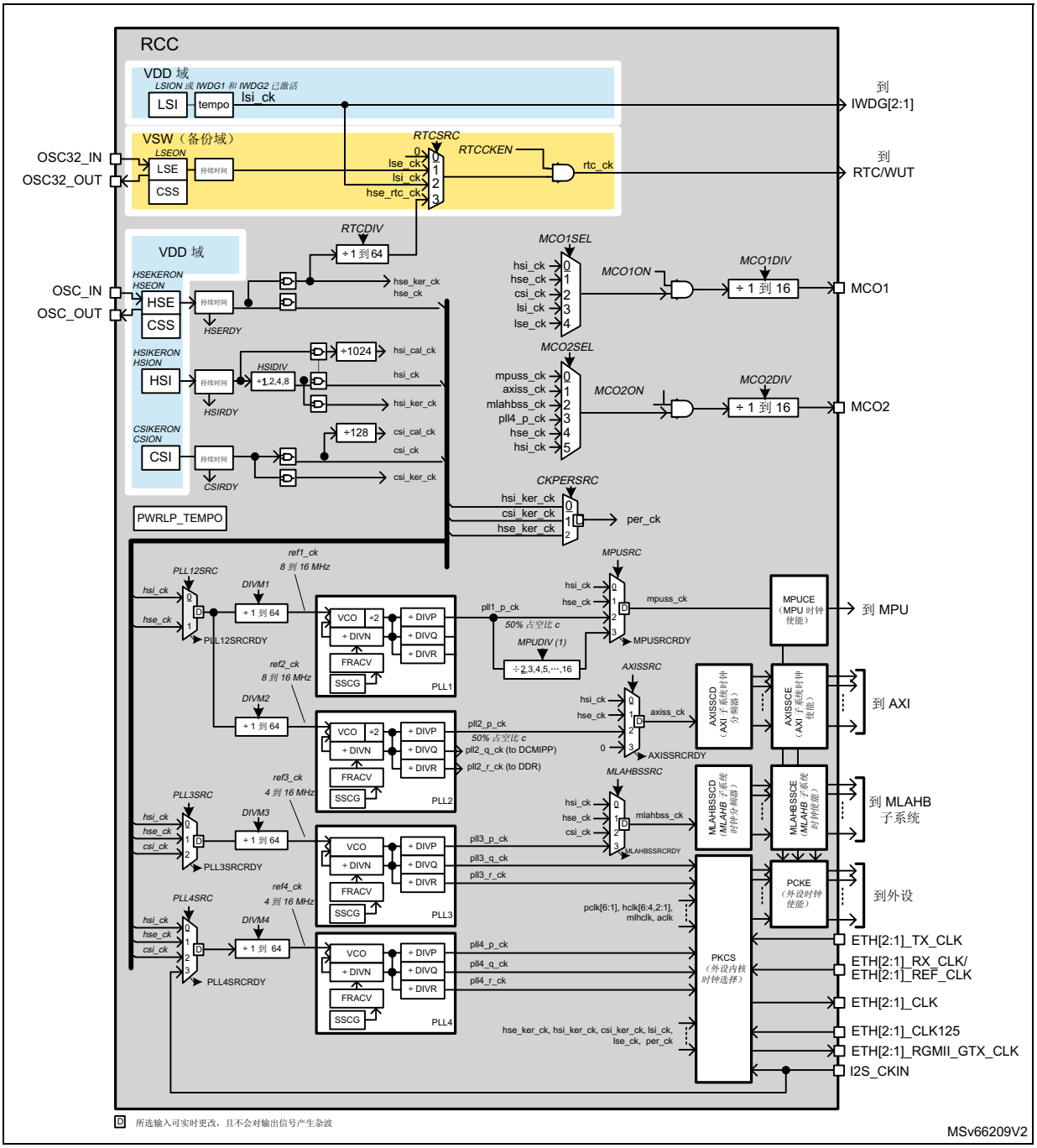
AXISSCD 和 MLAHBSSCD 模块包含多个预分频器，用于配置处理器和总线矩阵时钟频率。

AXISSCE、MLAHBSSCE 和 PCKE 模块负责对外设内核时钟、总线接口、内核与总线矩阵时钟进行门控。

PKCS（外设内核时钟选择）模块提供多个动态开关，支持丰富的外设内核时钟分配选项。

如*图 53* 所示，RCC 具有 2 个时钟输出（MCO1 和 MCO2），可非常灵活地进行时钟选择和频率调节。

图 53. 顶级时钟树



3. **X** 表示系统复位后所选择的复用器输入。

10.4.1 时钟命名约定

RCC 为整个电路提供时钟。为避免误解，本文档使用以下术语：

- **外设时钟：**
  外设时钟是指 RCC 为外设提供的时钟。可以为一个外设提供两种类型的时钟：
  – 总线接口时钟
  – 内核时钟
  外设从 RCC 接收一个或多个总线接口时钟。该时钟通常为 AHB、APB 或 AXI 时钟，具体取决于外设所连接的总线。有些外设只需要一个总线接口时钟（例如 GPIOx 和 DMAx）。
  而有些外设还需要一个专用时钟来处理接口功能。该时钟称为内核时钟。例如，SAI 等外设需要生成特定并且精确的主时钟频率，这需要专用的内核时钟频率。将总线接口时钟与特定接口需求相分离的另一个优势是，不需要重新编程外设即可更改总线时钟。
- **CPU 时钟**
  CPU 时钟是指提供给 CPU 的时钟 (**mpuss_ck**)。
- **总线矩阵时钟**
  总线矩阵时钟是指提供给不同矩阵和总线桥（APB、AHB 或 AXI）的时钟。

10.4.2 振荡器说明

下表显示了通过寄存器使能了振荡器时，振荡器状态与系统模式的对应关系。“可用”一词表示如果通过寄存器激活，则资源可供使用。

表 52. 振荡器状态与系统模式

系统模式	V_{DD} 域				V_{SW} 域
	HSE	HSI	CSI	LSI	LSE
退出系统复位	关闭	开启	关闭	可用	可用
处于运行模式	可用	可用	可用	可用	可用
处于 Stop、LP-Stop、LPLV-Stop 和 LPLV-Stop2 模式	可用 (1)	可用 (2)	可用 (3)	可用	可用
处于待机模式	关闭	关闭	关闭	可用	可用
处于 VBAT 模式	关闭	关闭	关闭	关闭	可用

1. 如果 HSEKERON =“1”，或者外设生成内核时钟请求，则 HSE 在任何 Stop 模式下都可以保持激活状态。
2. 如果 HSIKERON =“1”，或者外设生成内核时钟请求，则 HSI 在任何 Stop 模式下都可以保持激活状态。
3. 如果 CSIKERON =“1”，或者外设生成内核时钟请求，则 CSI 在任何 Stop 模式下都可以保持激活状态。

HSE 振荡器

高速外部模块可通过两种可能的时钟源来生成时钟：

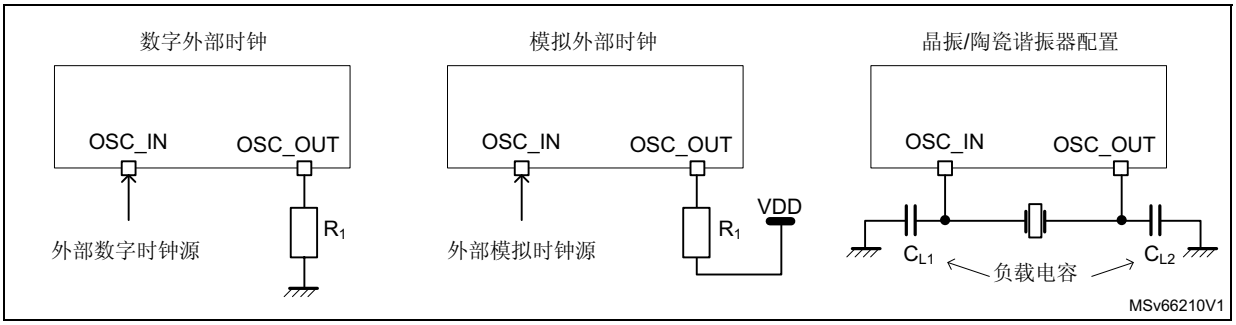
- 外部模拟时钟源
- 外部数字时钟源
- 外部晶振/陶瓷谐振器

为了使 Boot ROM 能够检测 HSE 使用的配置，必须将一个电阻 (R1) 连接到 GND 或 V_{DD}。

HSE 使用外部数字时钟时，必须将电阻连接到 GND。HSE 使用外部模拟时钟时，必须将电阻连接到 V_{DD}，如果使用了晶振或陶瓷谐振器，则应移除电阻。

有关 C_{L1}、C_{L2} 和 R1 的值，请参见数据手册。

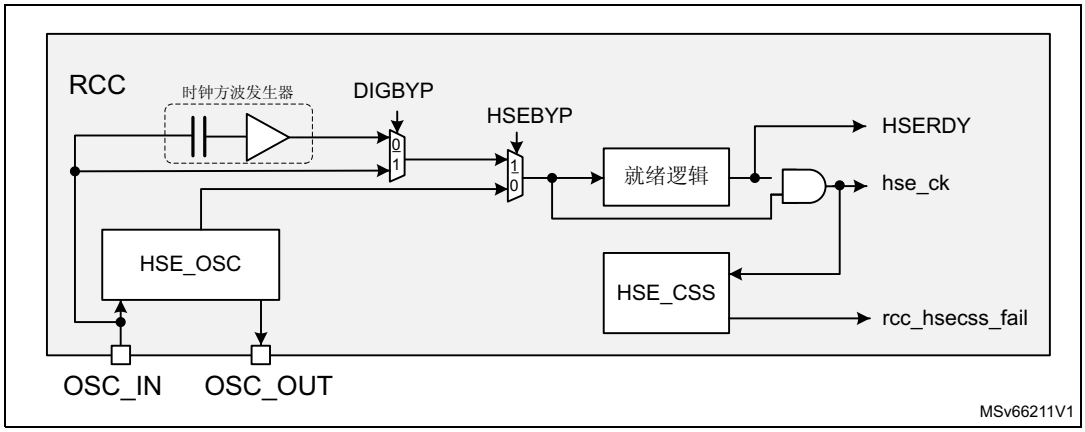
图 54. HSE 时钟源



RCC 振荡器和 MPU 与 AXI 时钟就绪寄存器 (RCC_OCRDYR) 的 HSERDY 标志指示 HSE 振荡器是否稳定。如果使能了 HSE（HSEON 置“1”），则经过 512 个 HSE 时钟周期后，HSERDY 标志将变为“1”。硬件将 HSERDY 位置 1 后，才可以使用 **hse_ck** 时钟。如在 *RCC 时钟源中断使能寄存器 (RCC_MP_CIER)* 中使能中断，则可产生中断。

图 55 所示为 HSE 时钟生成简图。

图 55. HSE 时钟生成



另请注意，系统处于任何 Stop 模式时，都可以使 HSE 保持使能状态，其他相关信息，请参见*第 10.4.7 节：Stop 模式和待机模式下的时钟生成*。

此外，可将 HSE 时钟驱动到 MCO1 和 MCO2 输出，以及用作其他应用组件的时钟源。

外部时钟配置（HSE 旁路）

当选择旁路模式（HSEBYP =“1”）时，HSE 可接受 OSC_IN 上的外部时钟源。

连接到 OSC_IN 引脚的外部时钟信号可以是数字信号或低摆幅（模拟）信号。由于有内部时钟方波器，可支持低摆幅时钟信号。输入信号的占空比必须接近 50%。

为了将 HSE 切换到外部时钟模式，用户必须执行以下操作：

- 向 *RCC 振荡器时钟使能清零寄存器 (RCC_OCENCLRR)* 中的 HSEON 位写入“1”以禁止 HSE。
- 验证 *RCC 振荡器和 MPU 与 AXI 时钟就绪寄存器 (RCC_OCRDYR)* 中的 HSERDY 位是否设为“0”来确认是否禁止 HSE。
- 如果提供给 OSC_IN 的时钟是全摆幅数字信号，则将 DIGBYP 位置“1”，如果是低摆幅信号，则设为“0”。可以通过 *RCC 停止请求设置寄存器 (RCC_MP_SREQSETR)* 和 *RCC 振荡器时钟使能清零寄存器 (RCC_OCENCLRR)* 来控制 DIGBYP 位。
- 向 *RCC 停止请求设置寄存器 (RCC_MP_SREQSETR)* 的 HSEBYP 位写入“1”将其置 1。
- 向 *RCC 停止请求设置寄存器 (RCC_MP_SREQSETR)* 的 HSEON 位写入“1”再次使能 HSE，以激活 hse_ck。
- 确认 *RCC 振荡器和 MPU 与 AXI 时钟就绪寄存器 (RCC_OCRDYR)* 的 HSERDY 位已置“1”，说明 HSE 已就绪可供使用。

有关详细信息，请参见*图 54* 和*图 55*。其他相关信息，请参见数据手册。

外部晶振/陶瓷谐振器配置

通过将 HSEBYP 位设为“0”并将 HSEON 位置“1”来使能该振荡器。

如果产品需要非常精确的高速时钟，可使用 HSE。

要选择外部晶振/陶瓷谐振器，应用必须遵循以下序列：

- 向 *RCC 振荡器时钟使能清零寄存器 (RCC_OCENCLRR)* 中的 HSEON 位写入“1”以禁止 HSE。
- 验证 HSERDY 位是否设为“0”，以确认是否禁止 HSE。
- 向 *RCC 振荡器时钟使能清零寄存器 (RCC_OCENCLRR)* 的 HSEBYP 位写入“1”，从而将 HSEBYP 位设为“0”。
- 向 *RCC 停止请求设置寄存器 (RCC_MP_SREQSETR)* 的 HSEON 位写入“1”再次使能 HSE。
- 确认 HSERDY 位已置“1”，说明 HSE 已就绪可供使用。

相关的硬件配置如*图 54* 所示。谐振器和负载电容必须尽可能地靠近振荡器的引脚，以尽量减小输出失真和起振稳定时间。负载电容值必须根据所选晶振或陶瓷谐振器的不同做适当调整。有关详细信息，请参见数据手册的电气特性部分。

HSI 振荡器

HSI 为产品提供默认时钟。

HSI 是高速内部 RC 振荡器，也可直接用作处理器和互连时钟、外设时钟的源，或用作 PLL 输入。

用户可使用预分频器来选择 8 MHz、16 MHz、32 MHz 或 64 MHz 的 HSI 输出频率。该预分频器由 *RCC HSI 配置寄存器 (RCC_HSICFGR)* 中的 HSIDIV 控制。*RCC 振荡器和 MPU 与 AXI 时钟就绪寄存器 (RCC_OCRDYR)* 中还有一个标志 (HSIDIVRDY)，用于确认硬件是否将新的分频比考虑在内。

请注意，HSIDIV 的值可以实时更改，而且 HSI 模块会将新的分频比考虑在内，而不会在 HSI 时钟上产生任何噪声。

但必须考虑以下几点：

- 如果 HSI 当前用作 PLL 的参考时钟，则**不允许**更改 HSIDIV。
- 如果 HSI 时钟当前用作一些外设的内核时钟，应用必须确保 HSI 频率的变化不会干扰这些外设。

HSI 的优点如下：

- 无需外部晶振；是一种低成本时钟源
- 启动时间比 HSE 短：几微秒
- 低功耗

即使经过频率校准后，HSI 频率精度也不如外部晶振或陶瓷谐振器的频率精度高。

将 HSI 用作通信外设的内核时钟时必须谨慎：应用必须考虑以下参数：

- 从外设生成内核时钟请求到时钟实际可用的时间间隔
- 频率精度

另请注意，系统处于任何 Stop 模式时，都可以使 HSI 保持使能状态，其他相关信息请参见 *第 10.4.7 节：Stop 模式和待机模式下的时钟生成*。

RCC_OCRDYR 寄存器的 HSIRDY 标志指示 HSI 是否稳定。在启动时，硬件将 HSIRDY 位置 1 后，才可以使用 HSI 输出时钟。HSI 的激活也可以由应用通过 *RCC 停止请求设置寄存器 (RCC_MP_SREQSETR)* 和 *RCC 振荡器时钟使能清零寄存器 (RCC_OCENCLRR)* 中的 HSION 位来控制。

此外，可将 HSI 时钟驱动到 MCO1 和 MCO2 输出，以及用作其他应用组件的时钟源。

校准

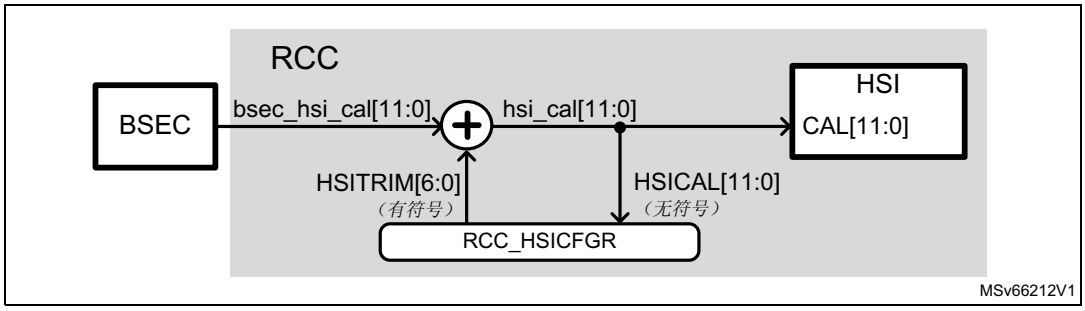
由于生产工艺不同，不同芯片的 RC 振荡器频率也不同，因此 ST 会对每个设备进行出厂校准，以达到 ACC_{HSI} 的精度。更多信息，请参见产品数据手册。

当发生 **app_rst** 或 **vcore_rst** 时，工厂校准值将重载到 HSI 中，并通过 HSICAL[11:0] 位显示。

如果应用受到电压或温度变化影响，则可能会影响到 RC 振荡器的频率。用户可使用 HSITRIM[6:0] 位对 HSI 频率进行微调。

HSICAL[11:0] 和 HSITRIM[6:0] 位在 *RCC HSI 配置寄存器 (RCC_HSICFGR)* 中。

图 56. HSI 校准流程



RCC 另外提供一个名为 **hsi_cal_ck** 的信号，专用于 HSI 校准。该信号连接到 TIM15 和 TIM12，其他相关信息，请参见*第 10.7.4 节：使用 TIMx 进行时钟校准*。

另请注意，当系统进入 Stop 模式时，信号 **hsi_cal_ck** 会被门控。

CSI 振荡器

CSI 是低功耗 RC 振荡器，也可直接用作 MLAHBSS 时钟、外设时钟的时钟源，或作为 PLL3 和 PLL4 的 PLL 输入。

CSI 的优点如下：

- 无需外部晶振：是一种低成本时钟源
- 启动时间比 HSE 短：几微秒
- 极低功耗

请注意，CSI 提供的时钟频率为 4 MHz 左右，而 HSI 能提供高达 64 MHz 的时钟。

即使经过频率校准后，CSI 频率精度也不如外部晶振或陶瓷谐振器的频率精度高。

虽然 CSI 的稳定时间要比 HSI 的稳定时间短，但是将 CSI 用作通信外设的内核时钟时必须谨慎，应用必须考虑以下参数：

- 从外设生成内核时钟请求到时钟实际可用的时间间隔
- 频率精度

另请注意，系统处于任何 Stop 模式时，都可以使 CSI 保持使能状态，其他相关信息请参见*第 10.4.7 节：Stop 模式和待机模式下的时钟生成*。

此外，可将 CSI 时钟驱动到 MCO1 输出，以及用作其他应用组件的时钟源。

CSI 也可以由应用通过 *RCC 停止请求设置寄存器 (RCC_MP_SREQSETR)* 和 *RCC 振荡器时钟使能清零寄存器 (RCC_OCENCLRR)* 中的 CSION 位来控制。

校准

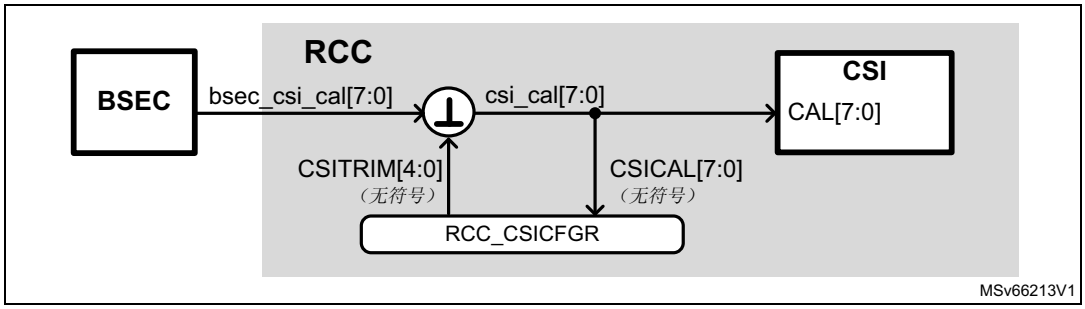
由于生产工艺不同，不同芯片的 RC 振荡器频率也不同，因此 ST 会对每个设备进行出厂校准，以达到 ACC_{CSI} 的精度。更多信息，请参见产品数据手册。

当发生 **app_rst** 或 **vcore_rst** 时，工厂校准值将重载到 CSI 中，并通过 CSICAL[7:0] 位显示。

如果应用受到电压或温度变化影响，则可能会影响到 RC 振荡器的频率。用户可使用 CSITRIM[4:0] 位对 CSI 频率进行微调。

CSICAL[7:0] 和 CSITRIM[4:0] 位在 *RCC CSI 配置寄存器 (RCC_CSICFGR)* 中。

图 57. CSI 校准流程



RCC 另外提供一个名为 **csi_cal_ck** 的信号，专用于 CSI 校准。该信号连接到 TIM15 和 TIM12，其他相关信息，请参见*第 10.7.4 节：使用 TIMx 进行时钟校准*。

另请注意，当系统进入 Stop 模式时，信号 **csi_cal_ck** 会被门控。

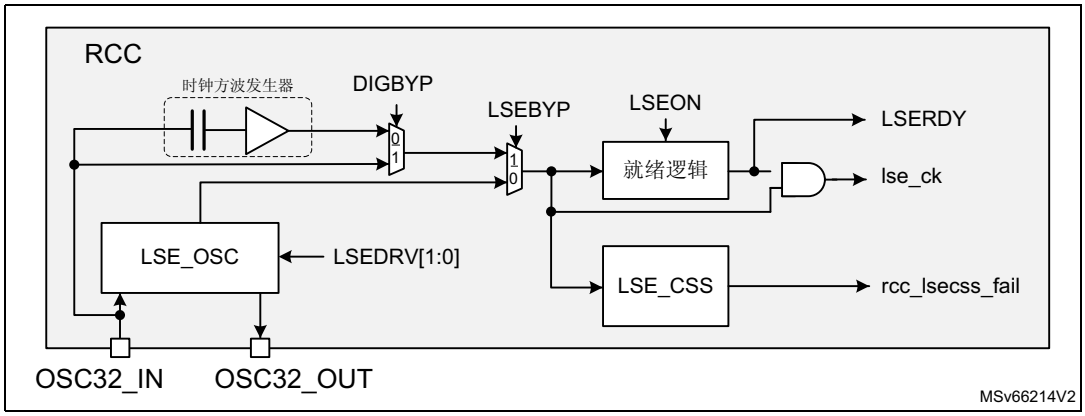
LSE 振荡器

低速外部模块可通过两种可能的时钟源来生成时钟：

- 外部晶振/陶瓷谐振器
- 外部用户时钟

LSE 功能提供一个 LSERDY 标志，用于指示 LSE 时钟是否可用。

图 58. LSE 时钟生成



LSE 振荡器通过 LSEON 位开启和关闭。当系统进入 Stop、LP-Stop、LPLV-Stop、LPLV-Stop2、待机或 VBAT 模式时，LSE 保持使能状态。

此外，可将 LSE 时钟驱动到 MCO1 输出，以及用作其他应用组件的时钟源。

外部时钟源（LSE 旁路）

在此模式下，必须为 OSC32_IN 引脚提供外部时钟源。输入时钟的频率最高可达 32.768 kHz，并且是低摆幅。通过将 LSEBYP 和 LSEON 位置“1”来选择此模式。在旁路模式下，使能 LSE 时（LSEON 置“1”），LSERDY 标志立即变为“1”。

外部时钟信号可以是数字或模拟信号（方波、正弦波或三角波）。由于有内部时钟方波器，可支持幅度减小的模拟时钟信号。输入信号的占空比必须接近 50%。OSC32_OUT 引脚应保持高阻态。其他相关信息，请参见数据手册。

为了选择外部时钟源，应用必须遵循以下操作顺序：

- 将 PWR_CR1 的 DBP 位置“1”，以允许对 RCC_BDCR 寄存器进行写访问。
- 向 LSEON 位写入“0”来禁止 LSE。
- 验证 LSERDY 位是否设为“0”，以确认是否禁止 LSE。
- 如果提供给 OSC32_IN 的时钟是全摆幅数字信号，则将 DIGBYP 位置“1”，如果是低摆幅信号，则设为“0”。
- 将 LSEBYP 位置“1”。
- 向 LSEON 位写入“1”再次使能 LSE。
- 确认 LSERDY 位已置“1”，则说明 LSE 已就绪可供使用。
- 将 PWR_CR1 的 DBP 位设为“0”，以使 RCC_BDCR 寄存器具有写保护权限。

请注意，除了 DBP 外，这些位均位于 *RCC 备份域控制寄存器 (RCC_BDCR)* 中。

外部晶振/陶瓷谐振器（LSE 晶振）

LSE 时钟由 32.768 kHz 晶振或陶瓷谐振器生成。可作为实时时钟 (RTC) 的时钟源来提供时钟/日历或其它定时功能，具有功耗低且精度高的优点。

要选择外部晶振/陶瓷谐振器，应用必须遵循以下序列：

- 将 PWR_CR1 的 DBP 位置“1”，以允许对 RCC_BDCR 寄存器进行写访问。
- 向 LSEON 位写入“0”来禁止 LSE。
- 验证 LSERDY 位是否设为“0”，以确认是否禁止 LSE。
- 将位 LSEBYP 设为“0”。
- 如果需要，配置 LSEDRV[1:0]。
- 向 LSEON 位写入“1”再次使能 LSE。
- 确认 LSERDY 位已置“1”，则说明 LSE 已就绪可供使用。
- 将 PWR_CR1 的 DBP 位设为“0”，以使 RCC_BDCR 寄存器具有写保护权限。

请注意，这些位均位于 *RCC 备份域控制寄存器 (RCC_BDCR)* 中。

如果使能了 LSE（LSEON 置“1”），则检测到 4096 个 LSE 时钟周期后，LSERDY 标志将变为“1”。该机制对外部时钟源（LSEBYP =“1”）也有效。当 LSE 就绪时，如果在 *RCC 时钟源中断使能寄存器 (RCC_MP_CIER)* 中使能，则可以产生中断。

注： *MCO1 输出上的 LSE 仅在运行模式和任何 Stop 模式下可用。在待机模式和 VBAT 模式下，无法通过 MCO1 输出获得 LSE。进入待机模式或 VBAT 模式时，MCO1 上可能会出现尖峰。*

LSE 提供了一种可编程的振荡器放大器驱动能力 (LSEDRV[1:0])。应用可以利用这种能力来调整振荡器，使之适应外部组件的特性。必须在使能 LSE 振荡器之前更改驱动能力。

警告： **不允许在 LSE 使能时更改驱动能力，否则 LSE 的行为得不到保证。**

LSI 振荡器

LSI 用作低功耗时钟源，可在系统处于 Stop、LP-Stop、LPLV-Stop、LPLV-Stop2 或待机模式时保持运行，供独立看门狗 (IWDG) 和唤醒定时器 (WUT) 使用。时钟频率在 32 kHz 左右。有关详细信息，请参见数据手册的电气特性部分。

可以使用 LSION 位开启或关闭 LSI，而 LSIRDY 标志则指示 LSI 振荡器是否稳定。如果独立看门狗已通过硬件选项或软件访问的方式启动，则 LSI 将强制打开且不可禁止。

LSI 启动时，硬件将 LSIRDY 位置 1 后，才会提供相应时钟。如在 *RCC 时钟源中断使能寄存器 (RCC_MP_CIER)* 中使能中断，则可产生中断。

此外，可将 LSI 时钟驱动到 MCO1 输出，以及用作其他应用组件的时钟源。

注： *请注意，MCO1 输出上的 LSI 只在运行模式和任何 Stop 模式下可用。在待机模式下，无法通过 MCO1 输出获得 LSI。进入待机模式时，MCO1 上可能会出现尖峰。*

LSION 和 LSIRDY 位在 *RCC 复位持续时间和 LSI 控制寄存器 (RCC_RDLSICR)* 中。

10.4.3 时钟安全系统 (CSS)

HSE 上的 CSS

RCC 具有时钟安全系统，允许应用检测 HSE 时钟是否因意外原因停止翻转。例如，硬件故障或遭受攻击等原因。如果检测到故障，CSS 会产生系统复位，并保护 BKPSRAM 中的敏感数据。

时钟安全系统可由应用软件通过 HSECSSON 位激活。即使将 HSEON 设为“0”，也仍可激活 HSECSSON 位。在 HSE 已使能并就绪且 HSECSSON 置“1”时，HSE 上的 CSS 将由硬件使能。

当 HSE 时钟由硬件自动禁止时（例如当系统进入 Stop 或待机模式时），HSE 上的 CSS 也会由硬件自动禁止。

不能由软件将 HSECSSON 位清零。

发生系统复位或系统进入待机模式时，由硬件将 HSECSSON 位清零。HSEON 和 HSECSSON 位在 *RCC 振荡器时钟使能设置寄存器 (RCC_OCENSETR)* 中。

如果检测到 HSE 上发生故障：

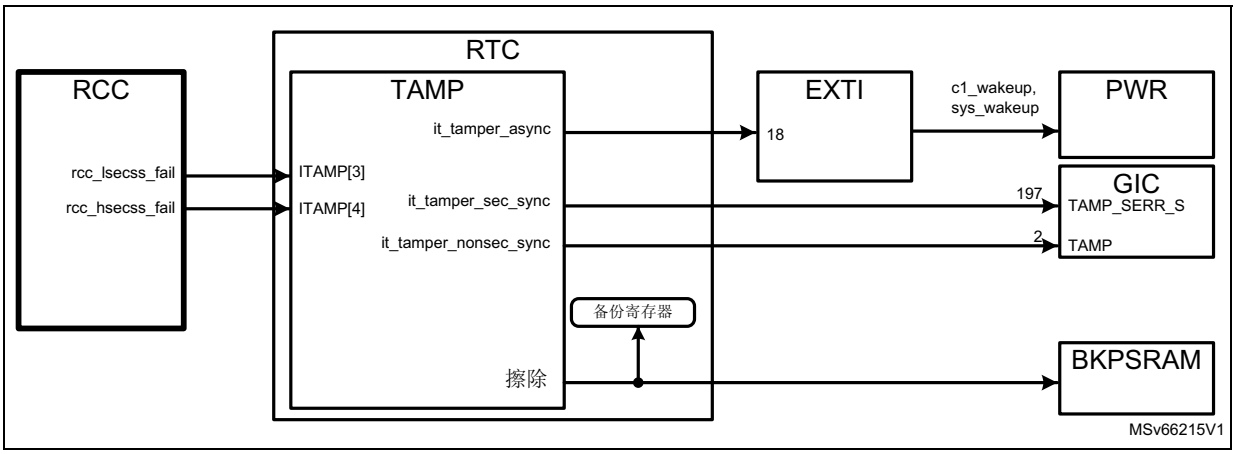
- RCC 产生应用复位（**app_rst** 和 **nreset**）。HCSSRSTF 标志置 1 以允许应用确定复位的根本原因。
- 将生成一个故障事件 (**rcc_hsecss_fail**)。该事件与 TAMP 模块连接，以保护备份寄存器和 BKPSRAM。

图 59 显示了与 HSE 和 LSE CSS 功能相关的模块。更多详细信息，请参见这些模块的说明。

注： *请注意，在发生 HSE 故障后，再次选择 HSE 时钟之前，RCC Boot ROM 复位状态清除寄存器 (RCC_BR_RSTSCLRR) 中的 HCSSRSTF 标志必须清零。发生 HSE 故障后，Boot ROM 通过复位* RCC_BR_RSTSCLRR *寄存器中的 HCSSRSTF 标志来执行此任务。然后，如果选择的启动源不需要 HSE 资源，则 Boot ROM 将重新启动电路。有关详细信息，请参见 Boot ROM 文档。*

此外，可以通过 LPTIM3 执行 HSE 时钟监视，防止系统受到超频攻击，LPTIM3 可将 HSE 时钟与 HSI 分频时钟进行比较，当 HSE 时钟比预期快时，生成一个 Tamper 事件。

图 59. LSE 和 HSE CSS 功能



LSE 上的 CSS

RCC 具有时钟安全系统，可检测是否存在 LSE 时钟故障：时钟是否因意外原因停止翻转，或者是否存在超频情况（遭受攻击等原因）。如果检测到故障，CSS 会生成故障事件，并保护 BKPSRAM 中的敏感数据。

激活 LSE 上的 CSS 时必须遵循以下顺序：

1. 将 LSEON 置“1”，并等待 LSERDY = “1”
2. 通过 RTCSRC 位字段选择 LSE 时钟
3. 将 LSECSSON 置“1”

LSECSSON 位只能通过备份域复位来禁止（请参见*第 10.3.4 节：备份域复位*）。

LSEON 和 RTCSRC 在 *RCC 备份域控制寄存器 (RCC_BDCR)* 中。

LSE 上的 CSS 在所有系统模式下均有效：运行模式、Stop 模式、待机模式和 VBAT 模式。

如果检测到 LSE 上发生故障：

- 将生成一个故障事件 (**rcc_lsecss_fail**)。该事件与 TAMP 模块连接，允许从待机模式唤醒，并可以保护备份寄存器和 BKPSRAM。
- 将激活一个中断标志 (LSECSSF) 以便在需要时产生 MPU 中断。

LSE 上的 CSS 提供两个标志，为了检测故障：RCC_BDCR 寄存器中的 LSECSSD 标志和 *RCC 时钟源中断标志寄存器 (RCC_MP_CIFR)* 中的 LSECSSF。

如果发生故障，软件必须执行以下操作：

- 通过 RCC 备份域控制寄存器 (RCC_BDCR) 中的 VSWRST 位复位备份域
- 如果为该事件使能了中断，则将 RCC 时钟源中断标志寄存器 (RCC_MP_CIFR) 中的 LSECSSF 清零

当然，软件可以采取任何其他必要操作来保证应用安全。

图 59 显示了与 HSE 和 LSE CSS 功能相关的模块。

10.4.4 时钟输出生成（MCO1 和 MCO2）

提供两个微控制器时钟输出 (MCO) 引脚：MCO1 和 MCO2。对于每个输出，都可以选择一个时钟源。可配置的预分频器实现了所选时钟的分频。有关信号选择的其他相关信息，请参见*图 3*。

MCO1 和 MCO2 输出通过 *RCC MCO1 配置寄存器 (RCC_MCO1CFGR)* 和 *RCC MCO2 配置寄存器 (RCC_MCO2CFGR)* 中的 MCO1DIV[3:0]、MCO1SEL[2:0]、MCO2DIV[2:0] 和 MCO2SEL[2:0] 控制。

对于不同的 MCO 引脚，必须在复用功能模式下对相应的 GPIO 端口进行编程。

分频器 MCODIV1 和 MCODIV2，为偶数分频值提供一个占空比为 50% 的时钟。占空比通常由以下公式指定：

$$DC(\%) = \frac{FLOOR\left(\frac{MCODIVx + 1}{2}\right)}{(MCODIVx + 1)} \times 100$$

MCODIVx = 0 时，占空比也为 50%。

请注意，MCO1 和 MCO2 输出仅在运行模式和任何 Stop 模式下可用。

小心： 为 MCO 输出提供的时钟不得超出最大引脚速度（有关支持的引脚速度的信息，请参见产品数据手册）。

10.4.5 PLL 描述

RCC 具有四个 PLL：

一个 VCO 频率高达 2 GHz (PLL_2000) 的 PLL (PLL1)，专用于为 Cortex-A7 MPU 子系统 (CA7-SS) 提供时钟。

一个 VCO 频率高达 1.6 Ghz 的 PLL (PLL2)，即 PLL_1600，用于 AXI-SS 的时钟生成。该 PLL 的一些输出也用于生成外设的内核时钟。

两个 VCO 频率高达 800 MHz 的 PLL (PLL_800)：

- 一个 PLL (PLL3) 用于为 MLAHB 子系统提供时钟。
- 一个 PLL (PLL4) 专用于生成外设的内核时钟。

PLL_2000 主要特性：

- VCO 频率范围在 992 到 2000 MHz 之间
- 输入频率范围：8 到 16 MHz
- 三种工作模式
  - 分数模式，使用 13 位 ΣΔ 调制器。Σ−Δ 调制器可实时更新，不会在 PLL 输出端产生频率过冲。
  - 整数模式
  - 展频模式，以减少电磁干扰
- 多达 3 个输出：
  - 3 个带有后分频器的输出

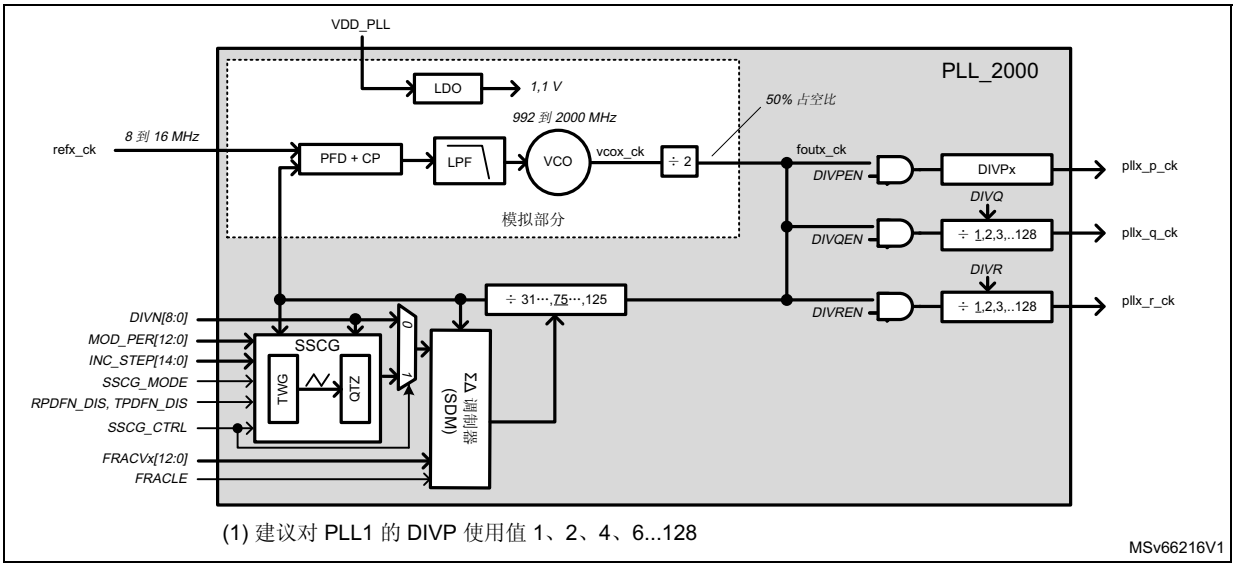
PLL_1600 主要特性：

- VCO 频率范围在 800 到 1600 MHz 之间
- 输入频率范围：8 到 16 MHz
- 三种工作模式
  - 分数模式，使用 13 位 ΣΔ 调制器。Σ−Δ 调制器可实时更新，不会在 PLL 输出端产生频率过冲。
  - 整数模式
  - 展频模式，以减少电磁干扰
- 多达 3 个输出：
  - 3 个带有后分频器的输出

PLL_800 主要特性：

- VCO 频率范围在 400 到 800 MHz 之间
- 输入频率范围：4 到 16 MHz
- 三种工作模式
  - 分数模式，使用 13 位 ΣΔ 调制器。Σ−Δ 调制器可实时更新，不会在 PLL 输出端产生频率过冲。
  - 整数模式
  - 展频模式，以减少电磁干扰
- 多达 3 个输出：
  - 3 个带有后分频器的输出

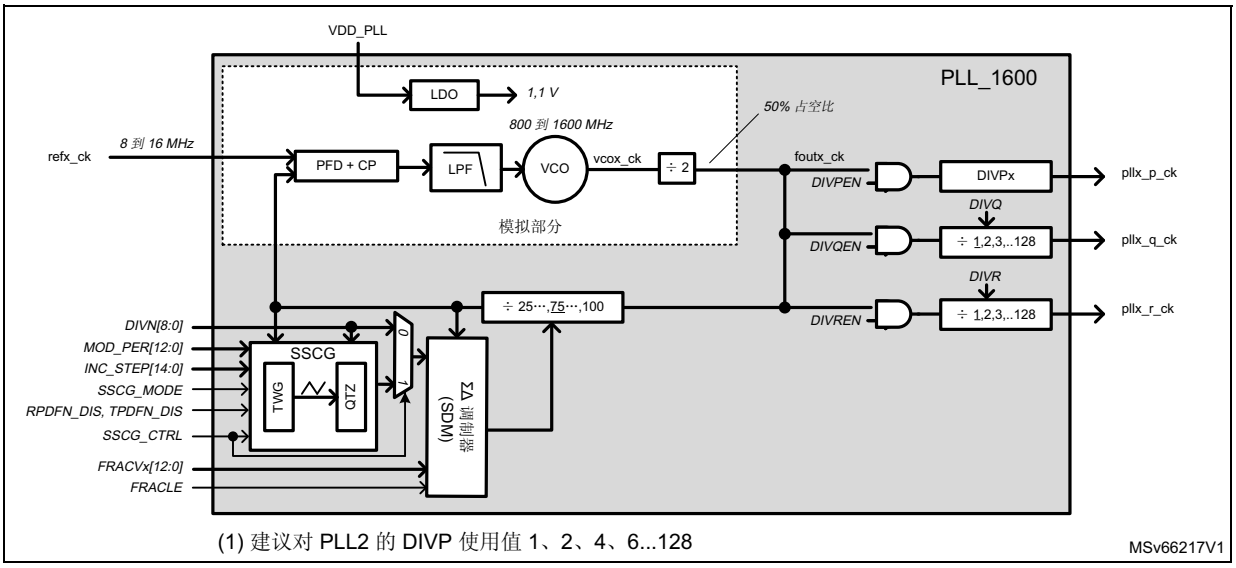
图 60. PLL1 框图



(1) 建议对 PLL1 的 DIVP 使用值 1、2、4、6...128

MSv66216V1

图 61. PLL2 框图



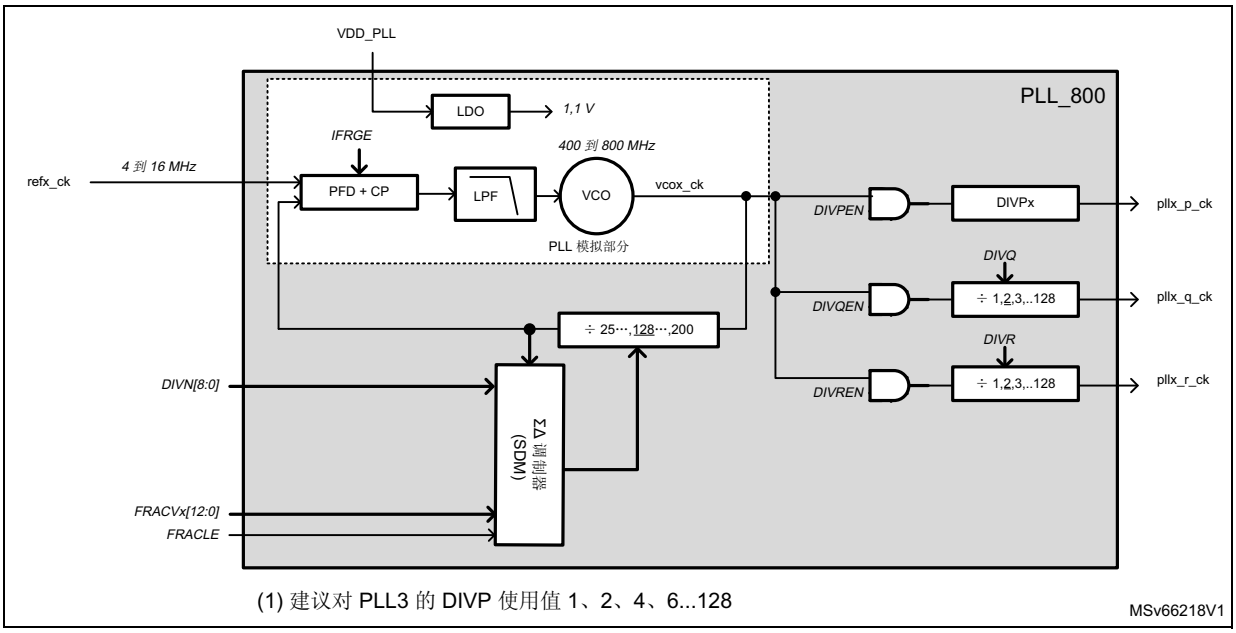
(1) 建议对 PLL2 的 DIVP 使用值 1、2、4、6...128

MSv66217V1

PLL1 和 PLL2 的模拟部分提供专用输出，以输出占空比为 50% 的干净的时钟信号 (**foutx_ck**)。**foutx_ck** 时钟的频率为 $F_{VCO}/2$，当后分频器被旁路时，可在 **pllx_p_ck**、**pllx_q_ck** 和 **pllx_r_ck** 输出上获得。

MPU 和 DDRC 对时钟抖动和占空比下降非常敏感，应用必须尽可能直接使用 **foutx_ck**。

图 62. PLL3 和 PLL4 框图



(1) 建议对 PLL3 的 DIVP 使用值 1、2、4、6...128

MSv66218V1

PLL 通过 RCC_PLLxCR、RCC_PLLxCFGR1、RCC_PLLxCFGR2 和 RCC_PLLxFRACR 寄存器进行控制。

用户必须为每个 PLL 提供一个输入频率在可接受范围内的干净的参考时钟 (**refx_ck**)，即：

- 8 到 16 MHz（PLL1 和 PLL2）
- 4 到 16 MHz（PLL3 和 PLL4）

应用必须正确编程 DIVM1、DIVM2、DIVM3 和 DIVM4 分频器以匹配这些条件。

此外，必须根据参考输入频率设置 IFRGE 位字段，以确保 PLL3 和 PLL4 达到最佳性能。

必须对 DIVN 环分频器进行编程，以使 VCO 输出达到预期频率。必须遵守 VCO 输出范围。

可通过将相应的 PLLON 位置“1”使能 PLL。用户可通过 PLLxRDY 位获知 PLL 已就绪（例如已锁定）。

警告： **对于 PLL1 和 PLL2 的 DIVP 输出，仅允许旁路（1 分频）和偶数分频。**
对于 PLL3 的 DIVP 输出，只允许偶数分频。
如果不遵守这些条件，电路的行为将无法得到保证。

通常，为了获得占空比接近 50% 的 PLL 输出频率，DIVP、DIVQ、DIVR 必须选择偶数分频。

PLL 编程建议

本节提供了一些建议，遵循这些建议可以有效使用 PLL。请注意，编程示例在*第 10.7.2 节：PLL 编程*中提供。

- 使能 PLL 之前，用户必须确保提供给 PLL 的参考频率 (**refx_ck**) 处于适当范围内。
- 使能 PLL 后，可以在不禁止相应 PLL 的情况下更改后分频器（DIVP、DIVQ 或 DIVR）的值。
  有关详细信息，请参见*重新编程后分频器*一节。
- 使能 PLL 后，如果更改 DIVMx、DIVNx、IFRGE 的值，会导致输出频率无效，可能会造成系统崩溃。没有硬件保护，需要由应用来避免这种情况。
- 如果有一个或多个 PLL 输出未被使用，应用必须将相应的使能位 DIVyEN 设为“0”。
- 为确保 PLL 正常工作，在禁止相应 PLL 之前，必须先将 DIVPEN、DIVQEN 和 DIVREN 位设为“0”。同样，仅当相应的位 PLLON 已经置“1”并且 PLL 就绪时，DIVPEN、DIVQEN 和 DIVREN 位才能置“1”。

注： *当相应的 PLLON 位置“1”时，**refx_ck** 时钟将提供给 PLLx。*

PLL 可以在 3 种不同的模式下工作：

- 整数模式
- 分数模式
- 展频模式（仅限 PLL1 和 PLL2）

以整数模式使用 PLL

当 Σ−Δ 调制器 (SDM) 装载了“0”，并且 SSCG_CTRL 位（如果有）=“0”时，PLL 以整数模式工作。

要将“0”装入 SDM，用户必须执行以下操作序列：

– 向 FRACV 写入“0”，向 FRACLE 写入“0”
– 向 FRACLE 写入“1”

FRACV[12:0] 和 FRACLE 位在 RCC PLLx 分数寄存器 (RCC_PLLxFRACR) 中，其中 x = 1、2、3 或 4。

对于 PLL1 和 PLL2，VCO 频率 (F_{VCO}) 由以下表达式计算得出：

$$F_{VCO} = F_{ref_ck} \times 2 \times (DIVN + 1)$$

后分频器输出的频率由下面的公式计算得出：

$$F_{pll_y_ck} = \frac{F_{VCO}}{2 \times (DIVy + 1)}$$

其中 y = P、Q 或 R

对于 PLL3 和 PLL4，VCO 频率 (F_{VCO}) 由以下表达式计算得出：

$$F_{VCO} = F_{ref_ck} \times (DIVN + 1)$$

后分频器输出的频率由下面的公式计算得出

其中 y = P、Q 或 R

$$F_{pll_y_ck} = \frac{F_{VCO}}{DIVy + 1}$$

以分数模式使用 PLL

当加载到 SDM 的值不为 0，并且 SSCG_CTRL 位（如果有）= “0”时，将激活分数模式。

应用可以随时通过执行以下操作序列来更新 SDM 值：

- 向 FRACV 写入新分数值，向 FRACLE 写入“0”
- 向 FRACLE 写入“1”。当 FRACLE 位从 0 变为“1”时，新的 FRACV 值便加载到 SDM 中。

Σ-Δ 调制器采用专门设计，可最大程度地降低抖动影响，同时实现非常小的频率阶跃。

当在分数模式下使用 PLL 时，必须先在使能 PLL 之前初始化 DIVN 分频器，之后可以实时对 FRACV 的值进行更改，而不会干扰 PLL 输出。
可使用该功能基于精度较高的晶振值生成特定频率，或实时执行频率微调。

对于 PLL1 和 PLL2，VCO 频率由以下表达式计算得出：

$$F_{VCO} = 2 \times F_{ref_ck} \times \left((DIVN + 1) + \frac{FRACV}{2^{13}} \right)$$

可使用*以整数模式使用 PLL*一节中的公式计算 $F_{pll_x_ck}$。

对于 PLL3 和 PLL4，VCO 频率由以下表达式计算得出：

$$F_{VCO} = F_{ref_ck} \times \left((DIVN + 1) + \frac{FRACV}{2^{13}} \right)$$

可使用*以整数模式使用 PLL*一节中的公式计算 $F_{pll_x_ck}$。

在展频模式下使用 PLL

当 SDM 载入 0，并且 SSCG_CTRL 位= “1” 时，展频模式即被激活。该功能在 PLL1 和 PLL2 上可用。

展频技术包括用低频信号（本文档中为三角波信号）调制 VCO 频率，以将时钟能量扩散到更广的频带中，从而减少了发射的 EMI 峰值的数量。

展频调制的参数通过以下位字段进行调整：

- MOD_PER[12:0]：用于调整调制频率

注： *请注意，调制频率通常在 20 到 50 kHz 的范围内。*

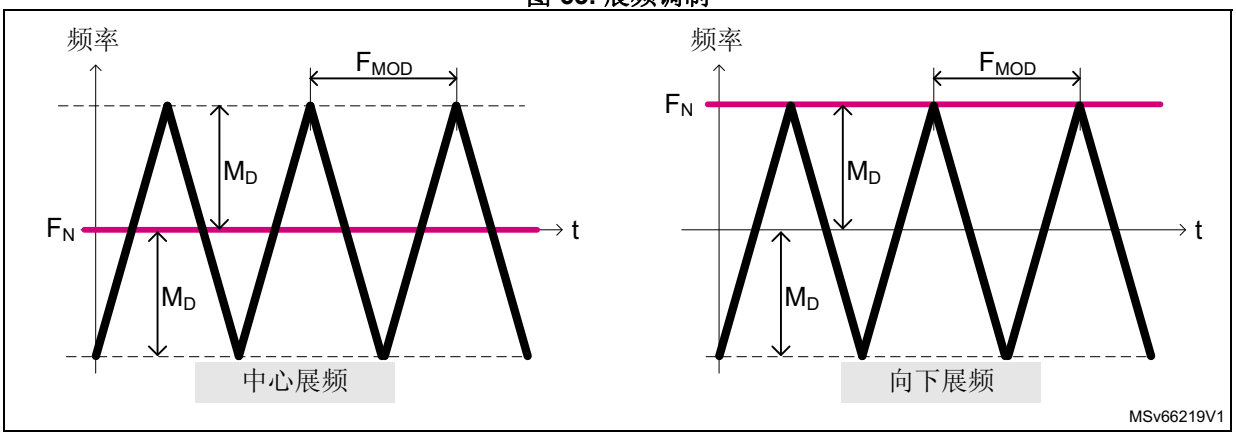
- INC_STEP[14:0]：用于调整调制深度（或调制指数）
- SSCG_MODE：用于定义调制是以 VCO 频率为中心（中心展频）还是相对于 VCO 频率降低（向下展频）。

MOD_PER[12:0]、INC_STEP[14:0] 和 SSCG_MODE 位在 RCC PLLx 展频时钟发生器寄存器 (RCC_PLLxCSGR) 中，其中 x = 1 或 2。

图 63 显示了在 SSCG_MODE = 0（中心展频）和 SSCG_MODE = “1” （向下展频）时调制标称频率 (F_N) 的信号。标称频率是 PLL 在整数模式下的输出频率（未应用时钟展频时）。

将 SSCG_MODE 位置 “1” （向下展频）可以保证当 SSCG 使能时，PLL 输出频率不会超出编程的频率值。

图 63. 展频调制



峰值调制深度（以百分比表示）由以下公式计算得出：

$$M_D\ (\%) = \frac{MOD_PER \times INC_STEP \times 100 \times 5}{(2^{15} - 1) \times (DIVN + 1)}$$

请注意，MOD_PER x INC_STEP 不得超出 (2^{15}-1)。

调制频率 (F_{mod}) 由以下公式计算得出：

$$F_{mod} = \frac{F_{ck_ref}}{4 \times MOD_PER}$$

为了使用展频功能，用户必须执行以下操作：

- 根据*以整数模式使用 PLL*一节编程 PLL，以根据应用目标调整标称频率 (F_N)。
- 根据所需的调制频率 (F_{mod}) 计算 MOD_PER 值：

$$MOD_PER = ROUND\left(\frac{F_{ck_ref}}{4 \times F_{mod}}\right)$$

- 根据所需的调制深度 (M_D) 计算 INC_STEP 值：

$$INC_STEP = ROUND\left(\frac{(2^{15} - 1) \times M_D \times (DIVN + 1)}{100 \times 5 \times MOD_PER}\right)$$

确认 MOD_PER x INC_STEP 未超出 (2^{15}-1)。

- 然后可以调整 DIVP、Q 或 R，将 SSCG_CTRL 位置“1”，PLL 使能。

用户可以按如下所述检查 F_{MIN}、F_{MAX} 和 F_C：

- 如果 SSCG_MODE =“0”（中心展频）

  F_C 由*以整数模式使用 PLL*一节中的公式计算得出

  $F_{MIN} = F_C \times (1 - M_D/100)$

  $F_{MAX} = F_C \times (1 + M_D/100)$

- 如果 SSCG_MODE =“1”（向下展频）

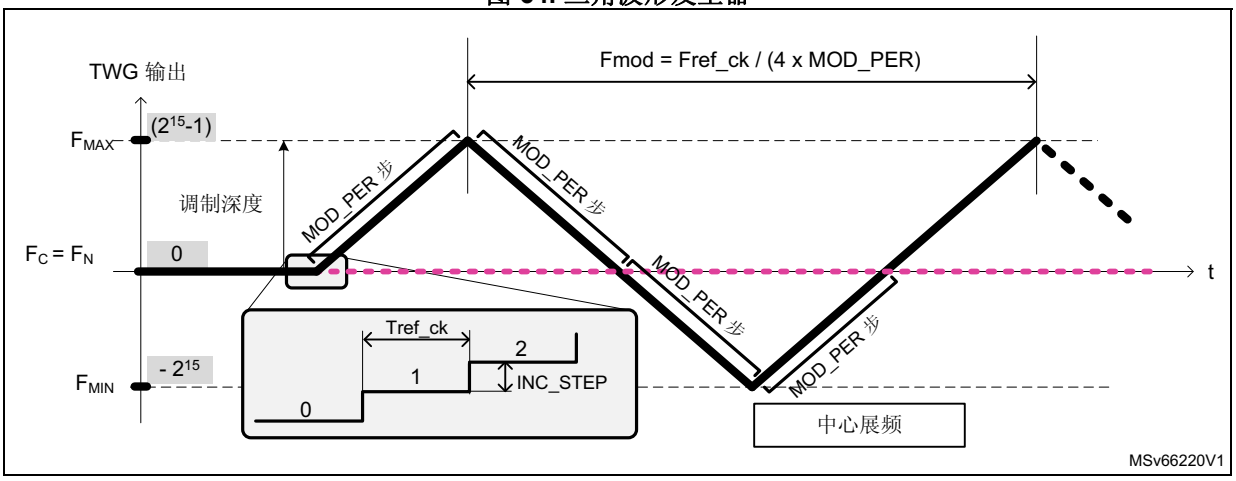
  F_{MAX} 由*以整数模式使用 PLL*一节中的公式计算得出

  $F_{MIN} = F_{MAX} \times (1 - 2 \times M_D/100)$

  $F_C = F_{MAX} \times (1 - M_D/100)$

图 64 显示由三角波形发生器（TWG 模块）生成的数字信号，以及 MOD_PER 和 INC_STEP 如何更改三角波形。

图 64. 三角波形发生器



有关 PLL 编程的其他相关信息，请参见*重新编程后分频器*一节和*PLL 初始化过程*一节。

10.4.6 子系统时钟

RCC 处理三个子系统时钟：**mpuss_ck**、**axiss_ck** 和 **mlahbss_ck**。

子系统时钟选择

系统复位后，将选择 HSI 作为整个系统的主时钟。因此使用 HSI 来提供时钟：

- **mpuss_ck**
- **axiss_ck**
- **mlahbss_ck**

系统复位后，所有 PLL 以及 CSI 和 HSE 均将关闭。

系统运行时，用户可以选择用于 MPU (**mpuss_ck**)、互连 (**axiss_ck**) 和 MLAHB (**mlahbss_ck**) 的时钟。

对于 **mpuss_ck**，应用可在以下来源中选择：

- HSI
- HSE
- **pll1_p_ck**
- **pll1_p_ck**/MPUDIV

对于 **axiss_ck**，应用可在以下来源中选择：

- HSI
- HSE
- **pll2_p_ck**

对于 **mlahbss_ck**，应用可在以下来源中选择：

- HSI
- HSE
- CSI
- **pll3_p_ck**

mpuss_ck 的时钟源由 *RCC MPU 时钟选择寄存器 (RCC_MPCKSELR)* 中的 MPUSRC 控制，**axiss_ck** 的时钟源由 *RCC AXI 子系统时钟选择寄存器 (RCC_ASSCKSELR)* 中的 AXISSRC 控制，而 **mlahbss_ck** 的时钟源则由 *RCC MLAHB 子系统时钟选择寄存器 (RCC_MSSCKSELR)* 中的 MLAHBSSRC 控制。

动态时钟切换

图 65 中的时钟选择器（包含字母 D 的方块）允许实时重新配置所选输入，而不会产生尖峰或时间违规，这样便实现了动态时钟切换。

因此，只有在原始输入端和新输入端均存在时钟时，才能从原始输入切换到新输入。否则不提供时钟。要从这种情况中恢复，用户必须为两个输入提供有效时钟。

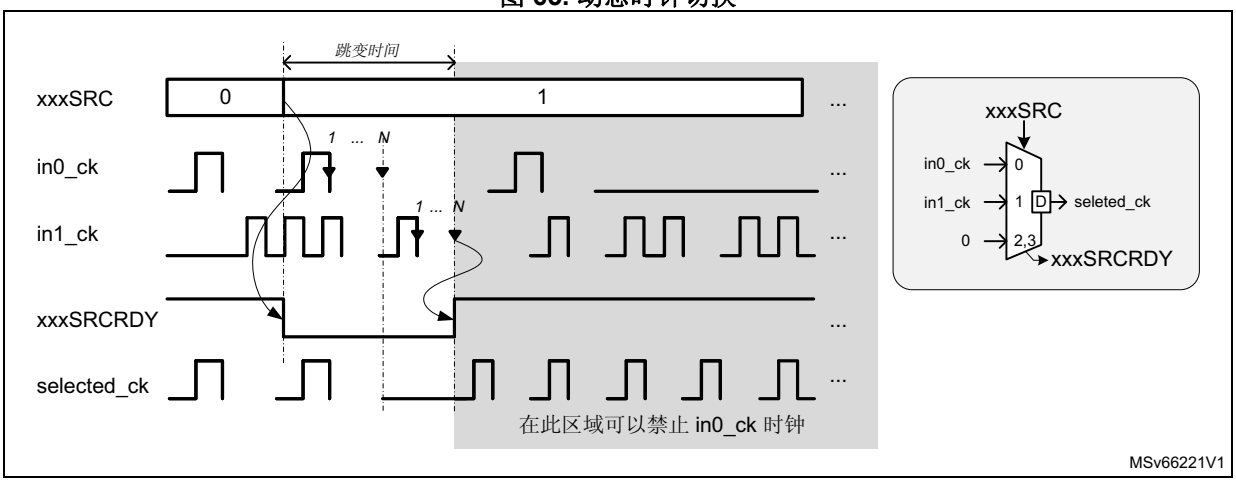
从一个输入跳变到另一个输入时，将对时钟输出进行门控，在最坏情况下，会在之前所选时钟的 3 个时钟周期内和新选择时钟的 3 个时钟周期内进行门控。如*图 65* 所示，在跳变期间，两个输入时钟均必须存在。

请注意，一些动态开关（例如由 PLL12SRC、PLL3SRC、PLL4SRC、MPUSRC、AXISSRC 和 MLAHBSSRC 控制的开关）提供一个就绪信号（*图 65* 中的 xxxSRCRDY）。当时钟源发生变化 (xxxSRC) 时，该就绪信号立即变为“0”，而当开关能够提供新选定的时钟时，变为“1”。该信号可用来确认切换跳变是否已正确完成。

当这些开关所在的位置没有选择输入时（*图 65* 的示例中的位置 2 和 3），不会向它们的输出端产生时钟，并且 xxxSRCRDY 标志设为“0”。

注： *但请注意，xxxSRCRDY 标志不能用来检测是否在开关输出位置产生了时钟。例如，如果开关已选择 **in0_ck** 并且 xxxSRCRDY 标志置“1”，则在禁止 **in0_ck** 时，xxxSRCRDY 标志不会变为“0”，而是保持为“1”，因为最后一次切换跳变已经正确执行。当然，如果应用更改了 xxxSRC 的值，则若禁止 **in0_ck**，开关将不能选择新的输入，并且 xxxSRCRDY 标志仍然保持为“0”。*

图 65. 动态时钟切换



切换跳变的硬件保护

只有在目标时钟源已就绪时（时钟在启动延迟或 PLL 锁定后稳定时），才可从一个时钟源切换到另一个。如果选择尚未就绪的时钟源，则在该时钟源就绪时才会进行切换。

RCC 振荡器和 MPU 与 AXI 时钟就绪寄存器 (RCC_OCRDYR) 中的状态位指示已就绪的时钟。

子系统时钟生成

图 66 显示了 CPU 和总线的时钟分配的简图。框图中显示的所有分频器均可实时更改，且不会违反时序。用户可以利用该功能以非常简单的方式调整总线频率以满足应用需求，从而优化功耗。

RCC MPU 时钟分频器寄存器 (RCC_MPCKDIVR) 中的 MPUDIV 分频器可用于动态调整 MPU 的时钟。

RCC AXI 时钟分频器寄存器 (RCC_AXIDIVR) 中的 AXIDIV 分频器可用于动态调整 AXI 矩阵的时钟。该分频器的值还影响 AHB5、AHB6、APB4 和 APB5 的时钟频率。

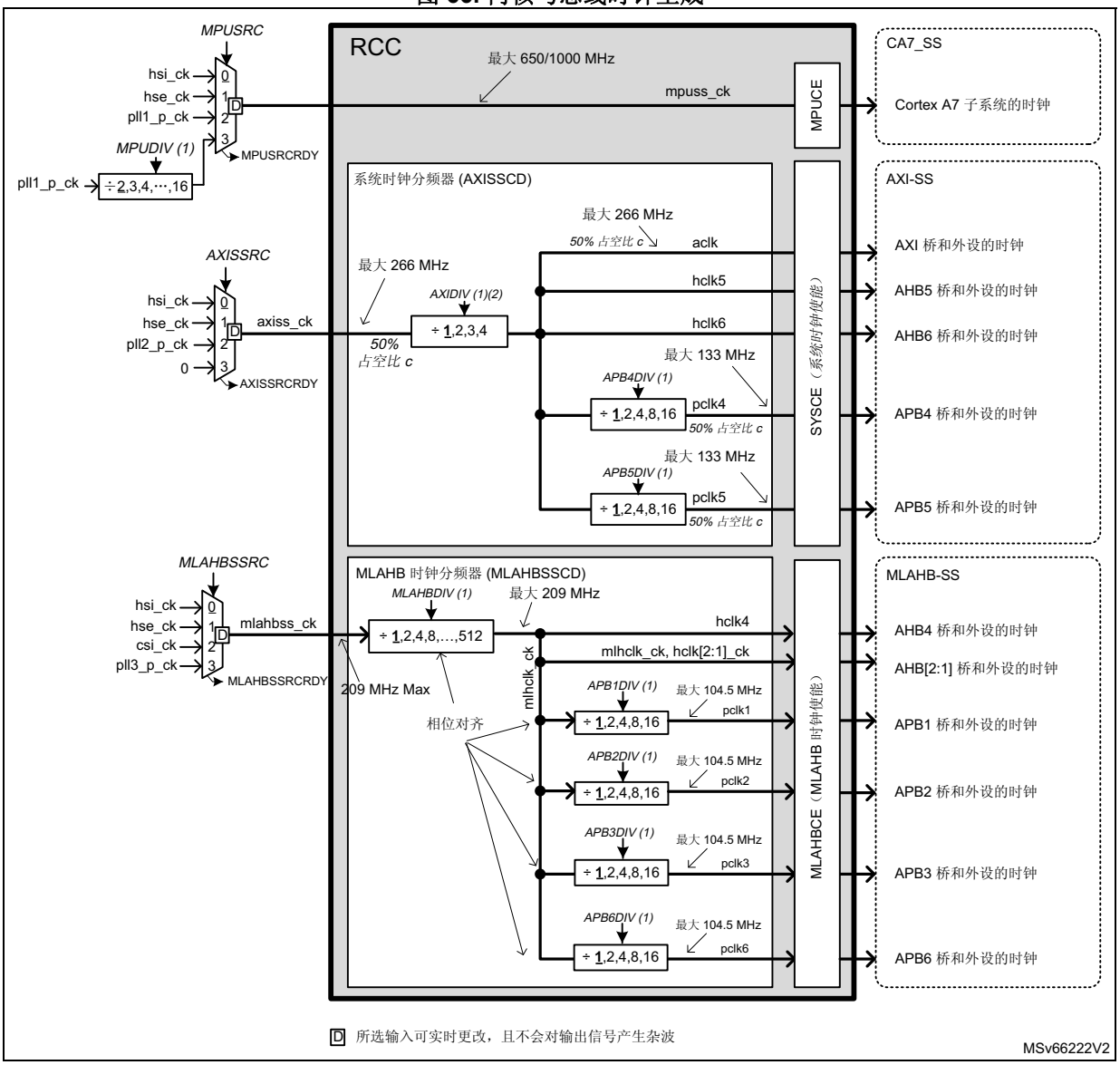
RCC APB4 时钟分频寄存器 (RCC_APB4DIVR) 中的 APB4DIV 分频器可用于动态调整 APB4 总线桥的时钟。同样，*RCC APB5 时钟分频器寄存器 (RCC_APB5DIVR)* 中的 APB5DIV 分频器可用于动态调整 APB5 总线桥的时钟。

在 MLAHB 时钟树中，*RCC MLAHB 时钟分频器寄存器 (RCC_MLAHBDIVR)* 中的 MLAHBDIV 分频器可用于动态调整 MLAHB 的时钟。该分频器的值还影响 AHB1、AHB2 和 AHB4 总线桥以及 APB1、APB2、APB3 和 APB6 的时钟频率。

RCC APB1 时钟分频器寄存器 (RCC_APB1DIVR) 中的 APB1DIV 分频器可用于动态调整 APB1 总线桥的时钟。同样，APB2、APB3 和 APB6 的时钟可分别通过 *RCC APB2 时钟分频器寄存器 (RCC_APB2DIVR)*、*RCC APB3 时钟分频器寄存器 (RCC_APB3DIVR)* 和 *RCC APB6 时钟分频器寄存器 (RCC_APB6DIVR)* 进行动态调整。。

请注意，可以实时更新的分频器提供一个就绪位，以便在考虑新的分频比生效时通知应用。

图 66. 内核与总线时钟生成



1. 可实时更改。
2. 即使是 3 分频，也保持 50% 的占空比。
3. **X** 表示系统复位后所选择的复用器输入。

10.4.7 Stop 模式和待机模式下的时钟生成

该产品支持四种系统 Stop 模式：

- 常规系统停止模式，称为 Stop
- 低功耗系统停止模式，称为 LP-Stop
- 低功耗、低电压系统停止模式，称为 LPLV-Stop 模式
- 低功耗、低电压系统停止 2 模式，称为 LPLV-Stop2

这些停止模式之间的主要区别在于 V_{DDCORE} 和 V_{DDCPU} 供电电压的处理：

- 在 Stop 模式下，V_{DDCORE} 供电电压与系统运行模式相同，具有相同的功率容量，
- 在 LP-Stop 模式下，V_{DDCORE} 供电电压与系统运行模式相同，但必须翻转 PWR_LP 信号，以允许外部稳压器在需要时减少或关闭电路板上的一些电源，
- 在 LPLV-Stop 模式下，V_{DDCORE} 和 V_{DDCPU} 供电电压降低（二者电压相同），
- 在 LPLV-Stop2 模式下，V_{DDCORE} 供电电压可降至与 LPLV-Stop 模式的电压相同，而 V_{DDCPU} 供电电压则被关断。

进入 Stop、LP-Stop、LPLV-Stop 或 LPLV-Stop2 模式

当 MPU 处于 CStop 状态时，系统可以进入 Stop、LP-Stop 或 LPLV-Stop 模式。当 MPU 处于 CStandby 状态时，系统可以进入 LPLV-Stop2 模式。请参见*第 9.5.1 节：工作模式*。

在进入 CStop 之前，处理器必须屏蔽除 RCC 唤醒中断以外的所有中断，以确保安全地进入和退出停止序列。RCC 提供安全和非安全唤醒中断：**rcc_sec_mpu_wkup_it** 和 **rcc_nonsec_mpu_wkup_it**，可通过 RCC_MP_CIER.WKUPIE 位使能。其他相关信息，请参见*第 10.6 节：RCC 中断*。

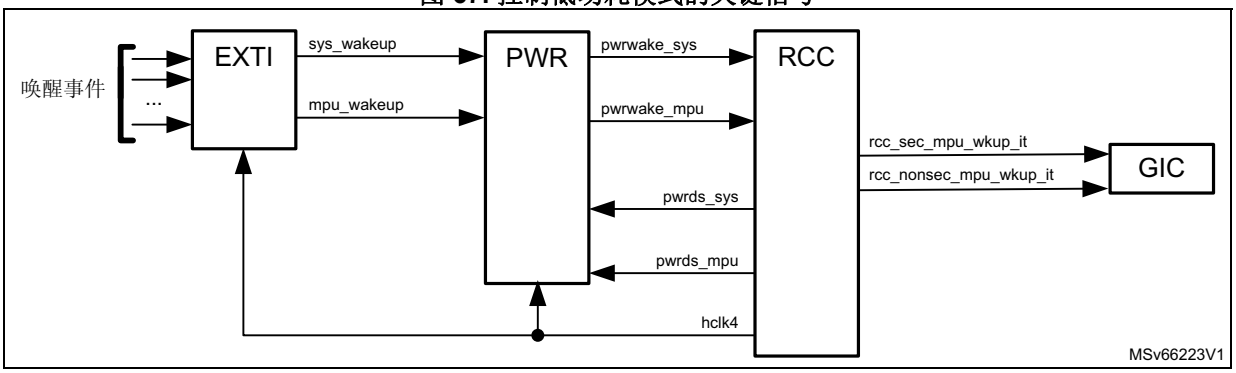
在门控所有时钟之前，RCC 等待 DDRC 进入自刷新模式。因此，可能会延迟进入 Stop 模式、LP-Stop 模式、LPLV-Stop 模式 或 LPLV-Stop 2 模式，直到 DDRC 允许才能进入。随后 RCC 会停止所有时钟（子系统和内核时钟），并将禁止以下时钟源：

- HSE（HSEKERON = “0”）
- HSI（HSIKERON = “0”）
- CSI（CSIKERON = “0”）
- PLL1、PLL2、PLL3 和 PLL4。

注： *在任何 Stop 模式中，HSE、HSI 和 CSI 均不能用作子系统时钟源。在任何 Stop 模式下，HSE、HSI 和 CSI 均可保持工作状态，以作为内核时钟的时钟源，前提是，RCC 振荡器时钟使能设置寄存器 (RCC_OCENSETR) 的 HSEKERON、HSIKERON 和 CSIKERON 均为 “1”，以避免在具有唤醒功能的外设请求内核时钟时需要振荡器稳定时间。*

注： *如果分别通过 RCC 备份域控制寄存器 (RCC_BDCR) 的 LSEON 控制位和 RCC 复位持续时间和 LSI 控制寄存器 (RCC_RDLSICR) 的 LSION 控制位使能，则 LSE 和 LSI 振荡器在任何 Stop 模式下均保持工作状态。*

图 67. 控制低功耗模式的关键信号



RCC 还可以通过信号 **pwrds_mpu** 和 **pwrds_sys**（参见*图 46*）通知 PWR 模块 MPU 子系统处于 Cstop 模式。

- **pwrds_mpu** 用于向 PWR 指示 MPU 处于 CStop 或 Cstandby 模式。
- **pwrds_sys** 用于向 PWR 指示系统处于 Stop 模式、LP-Stop 模式、LPLV-Stop 模式或 LPLV-Stop2 模式，并且不再提供时钟。

在 Stop、LP-Stop、LPLV-Stop 或 LPLV-Stop 2 模式期间

如上节所述，在系统 Stop、LP-Stop、LPLV-Stop 或 LPLV-Stop2 模式下，除了 LSE 和 LSI 保持其当前状态外，所有时钟、PLL 和振荡器均被禁止。但在任何 Stop 模式下，有两种特定情况可以使能 HSE、HSI 或 CSI：

- 专用外设请求内核时钟时：
  在这种情况下，外设将接收到 HSE、HSI 或 CSI，具体取决于为该外设选择的内核时钟源（通过 PERxSRC）。
- HSEKERON、HSIKERON 或 CSIKERON 位置 1 时：
  在这种情况下，HSE、HSI 或 CSI 在相关的 Stop 模式下保持运行，但不为外设提供时钟。这样，当系统退出该 Stop 模式或当外设请求内核时钟时，时钟立即可用。有关详细信息，请参见*表 53*。

HSEKERON、HSIKERON 和 CSIKERON 位在 *RCC 振荡器时钟使能设置寄存器 (RCC_OCENSETR)* 和 *RCC 振荡器时钟使能清零寄存器 (RCC_OCENCLRR)* 中。*表 53* 详细列出了有关这些位的行为。

另请注意，HSE、HSI 和 CSI 输出提供两条时钟路径（请参见*图 3*）：

- 一条路径用于处理器和互连时钟（**hse_ck**、**hsi_ck** 或 **csi_ck**），
- 一条路径用于外设内核时钟（**hse_ker_ck**、**hsi_ker_ck** 或 **csi_ker_ck**）。

在任何系统 Stop 模式下，当外设请求内核时钟时，只激活提供外设内核时钟的路径，处理器和互连时钟路径不激活。

表 53. HSI、CSI 和 HSE 状态与系统模式(1)

系统模式	HSION (CSION) (HSEON)	HSIKERON (CSIKERON) (HSEKERON)	内核时钟请求 (2)	-	HSI 状态（CSI 状态）（HSE 状态）	hsi_ck (csi_ck) (hse_ck)	hsi_ker_ck (csi_ker_ck) (hse_ker_ck)
运行	0	X	1	→	开启	门控	开启
			0	→	关闭		门控
	1		X	→	开启	开启	开启

表 53. HSI、CSI 和 HSE 状态与系统模式(1)（续）

系统模式	HSION (CSION) (HSEON)	HSIKERON (CSIKERON) (HSEKERON)	内核时钟请求 (2)	-	HSI 状态（CSI 状态）（HSE 状态）	hsi_ck (csi_ck) (hse_ck)	hsi_ker_ck (csi_ker_ck) (hse_ker_ck)
Stop、LP-Stop、LPLV-Stop 和 LPLV-Stop2	X	X	1	→	开启 (3)	门控	开启
		0	0	→	关闭		门控
		1	0	→	开启		开启 (4)
待机	X	X	X	→	关闭	门控	门控

1. “开启”状态以灰色突出显示。
2. 当外设请求内核时钟并且其相应的 PERxSRC 选择 HSE、HSI 或 CSI 振荡器，同时其 PERxEN 和 PERxLPEN 置“1”时，内核时钟请求等于“1”。有关详细信息，请参见*表 63：MPU 的外设时钟使能详细信息*。
3. 仅使能由请求内核时钟的外设通过 PERxSRC 选择的振荡器。
4. 内核时钟提供给 PCKE 模块，并且在外设不生成内核时钟请求时被门控。

退出 Stop、LP-Stop、LPLV-Stop 或 LPLV-Stop2 模式

当能够在所选 Stop 模式下运行的外设生成事件时，系统可以退出 Stop、LP-Stop、LPLV-Stop 或 LPLV-Stop2 模式。从任何系统 Stop 模式退出时，通常按以下方式进行（另请参见*图 67*）：

- 外设向 EXTI 模块发送唤醒中断。
- EXTI 模块对 PWR 模块生成一个事件 (**sys_wakeup**)。
- 对于任何低功耗 LPxx-Stopx 模式，PWR 模块通过 PWR_LP 信号通知外部 PMIC，应恢复主电源的信号。
- 在 LPLV-Stop2 模式下，PWR 模块通过 PWR_CPU_ON 信号通知外部电源稳压器，应将 V_{DDCPU} 恢复到标称电压。
- 在 LPLV-Stopx 模式下，PWR 模块等待一段时间 ($t_{VDDCORETEMPO}$) 以达到标称 V_{DDCORE} 电压。
- 当 V_{DDCORE} 和 V_{DDCPU} 达到标称值时，PWR 将 **pwrwake_sys** 置为有效并发送给 RCC。
- RCC 激活 HSI 振荡器（如果因处于 Stop 模式而尚未开启），提供 HSI 时钟和 **hclk4** 时钟给 EXTI 和 PWR 模块，并开始时钟恢复序列。
- 当 **hclk4** 可用时，EXTI 将 **mpu_wakeup** 信号置为有效，以指示 PWR 模块必须激活处理器。
- 随后 PWR 将 **pwrwake_mpu** 信号置为有效并发送到 RCC。
- RCC 将 **rcc_mpu_wkup_it** 置为有效，并在这些时钟可用时为 MPU 和 AXI 子系统提供时钟。
- 处理器便退出 Cstop，并重新使能在任何 Stop 模式期间可以被激活的外设的中断。这样，产生唤醒事件的外设就能够产生 MPU 中断。
- RCC 通过将 **pwrds_mpu** 信号置为无效来应答 **pwrwake_mpu** 请求。

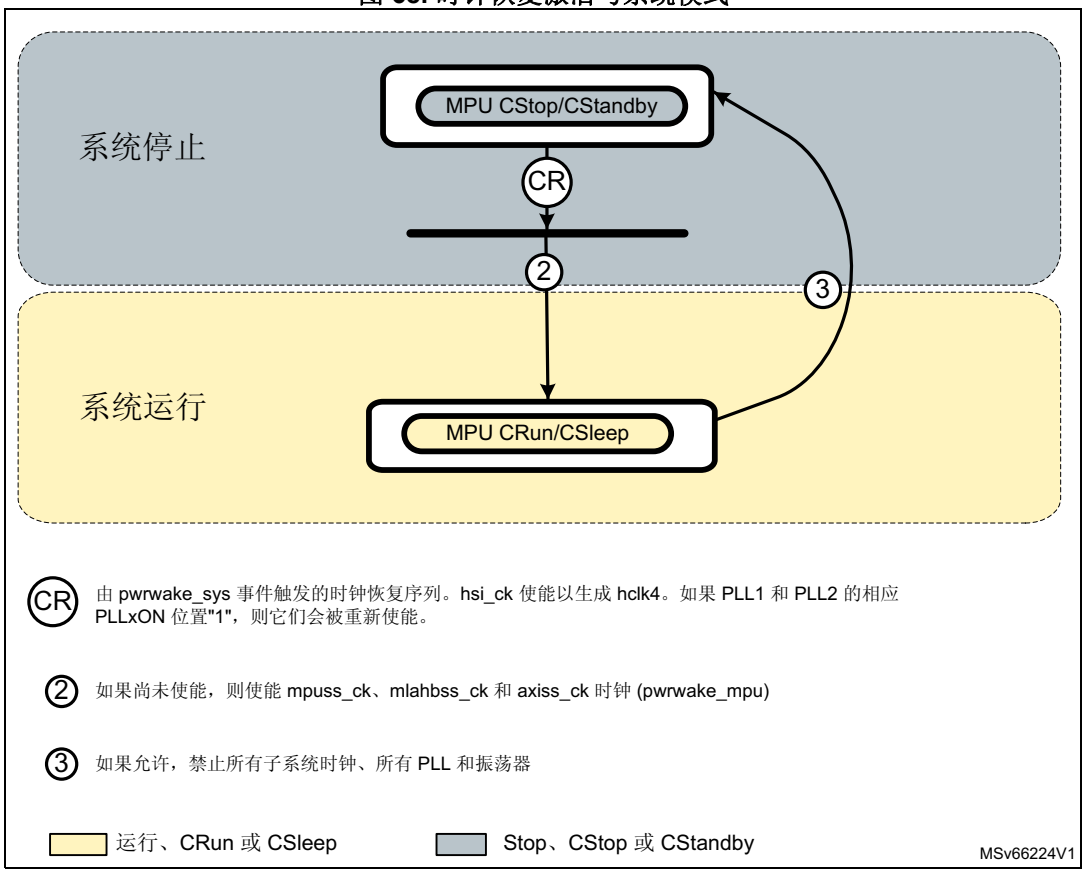
小心： RCC 集成了一个硬件状态机，能够恢复进入 Stop 模式之前使用的时钟配置。PLL3 和 PLL4 不按该机制处理。

当退出系统的一种 Stop 模式时，MLAHB 总是将 **hsi_ck** 用作子系统时钟，并将禁止 PLL3 和 PLL4。因此，在任何系统 Stop 后，以下位将设置为它们的复位值：

- HSION = “1”，MLAHBDIV = “0”，MLAHBSSRC = “0”
- PLL3ON = “0”，PLL3 的 DIVPEN、DIVQEN 和 DIVREN 设为 “0”
- PLL4ON = “0”，PLL4 的 DIVPEN、DIVQEN 和 DIVREN 设为 “0”

每次系统退出 Stop、LP-Stop、LPLV-Stop 或 LPLV-Stop2 时，RCC 均执行时钟恢复序列，如*图 67* 所示。当系统处于运行模式时，应用请求的所有振荡器和 PLL 都将被激活。

图 68. 时钟恢复激活与系统模式



处理 PWRLP_TEMPO

PWRLP_TEMPO 用于先自动等待外部 PMIC 从低功耗 (LP) 模式切换到高功耗 (HP) 模式，然后再使能耗能部分。应用可通过 *RCC 低功耗 Stop 模式延迟控制寄存器 (RCC_PWRLPDLYCR)* 控制延迟值。PWRLP_TEMPO 使用 **hsi_ck** 时钟作为参考。

为了实现快速重启，可以激活 **mlahbss_ck** 时钟来最小化 PWRLP_TEMPO 延迟（即 RCC_PWRLPDLYCR.PWRLP_DLY[21:0]=4，与复位后相同）。这仅在 MLAHB 子系统的功耗低至能够在 LP 模式下由外部电源稳压器支持时才可能实现。请注意，在这种情况下，外部电源稳压器仍需从 LP 模式跳变到 HP 模式。

警告： **RCC_PWRLPDLYCR.PWRLP_DLY[21:0]=4 是复位值，也是实现从任何 Stop 模式功能唤醒的最小 PWRLP_TEMPO 值。**

其他相关信息，请参见*第 10.7.5 节：时钟恢复序列示例*。

时钟恢复序列描述

每当系统退出一种 Stop 模式 (**pwrwake_sys**) 时，RCC 均将重新使能请求的振荡器和 PLL，以恢复 **mpuss_ck**、**axiss_ck** 和 **mlahbss_ck** 时钟。

当系统因 MPU 唤醒事件 (**pwrwake_mpu**) 而退出一种 Stop 模式时，**mpuss_ck**、**mlahbss_ck** 和 **axiss_ck** 时钟只有在 **mpuss_ck**、**mlahbss_ck** 和 **axiss_ck** 均就绪时才会同时重新使能。

更精确地说，时钟恢复序列分为以下主要步骤（另请参见*图 69* 和*图 70*）：

- 使能在进入一种系统 Stop 模式之前已使能的振荡器。
- 当 HSI 就绪时，RCC 可以触发 PWRLP_TEMPO（根据 PWRLP_DLY 值），以允许外部电源稳压器从 LP 切换到 HP 模式的延迟，然后再对 V_{DDCORE} 执行耗能操作，因为在退出 LP-Stop 模式时可能需要如此。
- 当 PLL1 和 PLL2 使用的振荡器就绪时，即可为 PLL 的模拟部分供电。由于 PLL 的模拟部分不是由 V_{DDCORE} 供电，因此可以使能 VCO，同时禁止所有输出（P、Q 和 R 分频器）。无需等待 PWRLP_TEMPO 这段延迟时间。
- 在使能 PLL 输出之前，RCC 必须等待 PWRLP_TEMPO 这段延迟时间并且 PLL 就绪。生成 **mlahbss_ck** 时钟的过程可能也需要等待 PWRLP_TEMPO 时间。
- 当系统因 MPU 唤醒事件而退出一种 Stop 模式时，如果 **mpuss_ck**、**mlahbss_ck** 和 **axiss_ck** 时钟就绪，则可以重新使能这些时钟。AXICKRDY 位也将置“1”。

其他相关信息，请参见*第 10.7.5 节：时钟恢复序列示例*。

图 69. 时钟恢复过程在任何系统 Stop 模式后激活

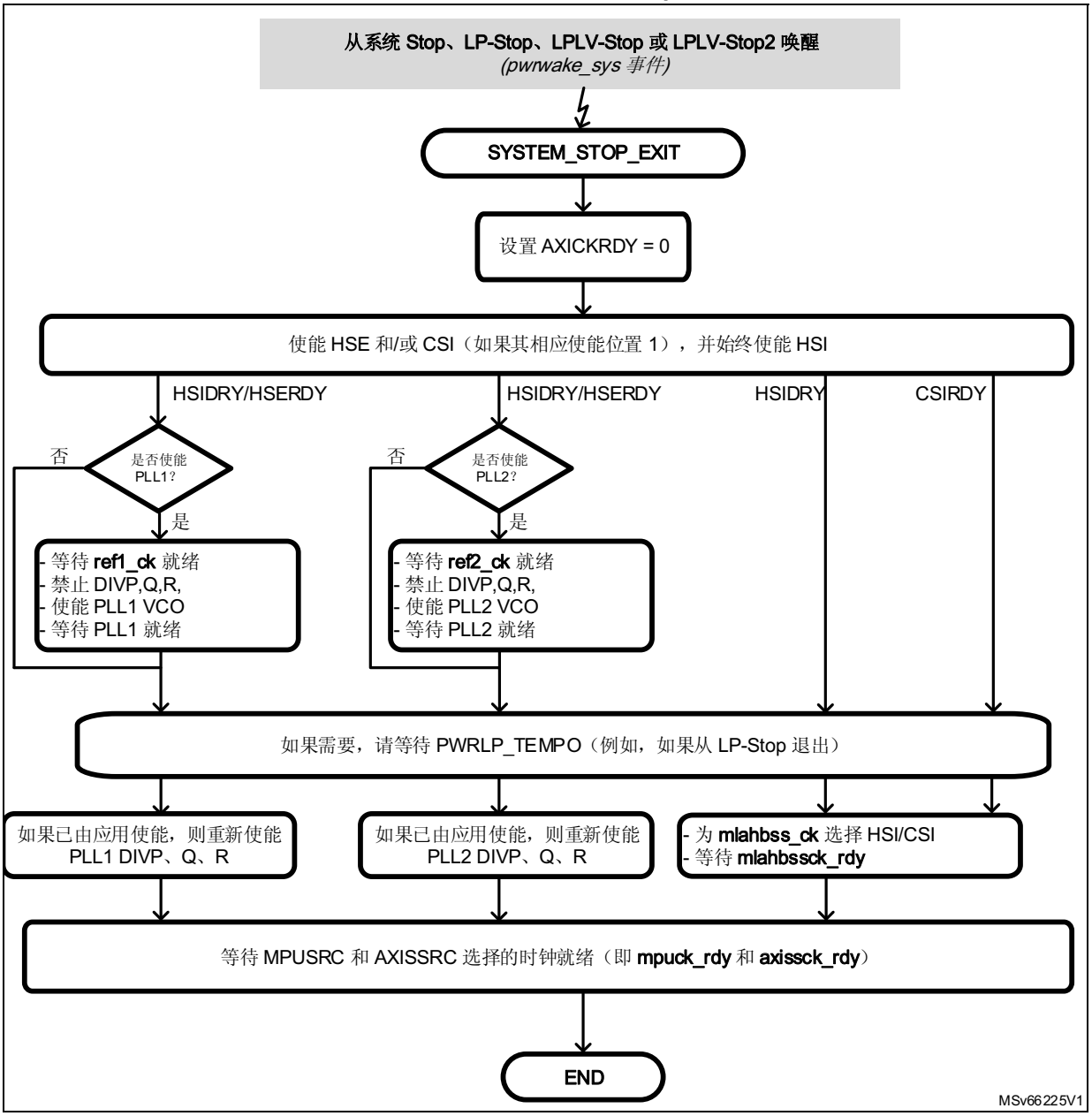


图 70. 从 CStop 过程唤醒

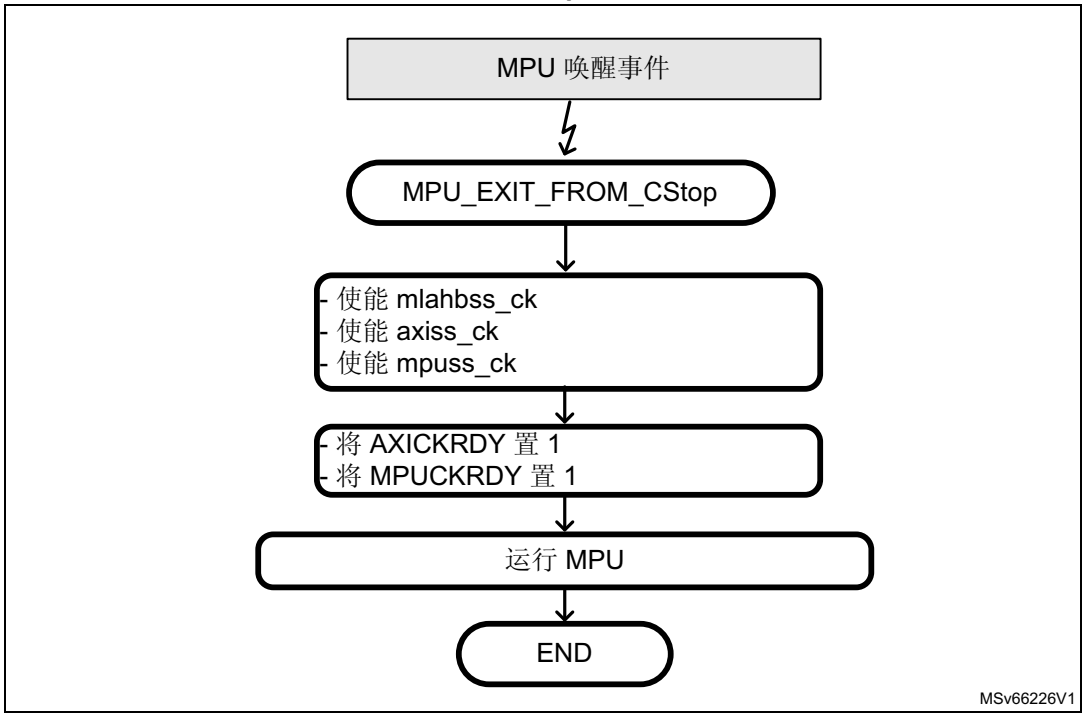


表 54. 在时钟恢复序列期间可能重新配置的资源

资源	-	时钟恢复期间对应的寄存器
子系统时钟切换	→	RCC_MPCKSELR、RCC_ASSCKSELR、RCC_MSSCKSELR
PLL1 和 PLL2 设置	→	RCC_PLL1CR、RCC_PLL1CFGR1、RCC_PLL1CFGR2、RCC_PLL1FRACR、RCC_PLL1CSGR RCC_PLL2CR、RCC_PLL2CFGR1、RCC_PLL2CFGR2、RCC_PLL2FRACR、RCC_PLL2CSGR
参考时钟选择	→	RCC_RCK12SELR
振荡器设置	→	RCC_OCENSETR、RCC_OCENCLRR、RCC_HSICFGR、RCC_CSICFGR
其他	→	RCC_PWRLPDLYCR

进入待机模式

当 MPU 处于 CStandby 模式时，系统可以进入待机模式。有关详细信息，请参见*第 9.6 节：PWR 低功耗工作模式*。

当整个系统进入待机模式时，RCC 停止所有时钟和 PLL（LSI 和 LSE 除外）。

RCC 还通过 **pwrds_mpu** 通知 PWR 模块 MPU 处于 Cstandby 模式。

然后 PWR 模块可以请求外部电源稳压器关闭 V_{DDCORE} 和 V_{DDCPU} 电压。只有使用 V_{DD} 和 V_{SW} 工作的部分在此模式下保持运行。

退出待机模式

当电路退出系统待机模式时，会选择 HSI 作为 **mpuss_ck**、**axiss_ck**、**mlahbss_ck** 的默认时钟，RCC 寄存器复位为初始值（RCC_BDCR、RCC_RDLSICR 和 RCC_BR_RSTSCLRR 寄存器除外。

请注意，当电路退出待机模式时将产生系统复位。

10.4.8 外设时钟分配

一些外设设计为使用两个彼此异步的不同时钟域：一个时钟域与寄存器和总线接口同步（**ckg_bus_perx** 时钟），另一个时钟域通常与外设同步（内核时钟）。

外设支持这两种时钟域的好处是：用户能更加自由地为 CPU、总线矩阵和外设的内核部分选择最佳时钟频率。

因此，用户无需对外设进行重新编程即可更改总线频率。例如，即使实时更改相应的 APB 时钟，也不会干扰正在通过 UART 进行的传输过程。

表 55 显示了 RCC 控制下的外设时钟（适用于大多数外设）。请注意，外设自身控制的时钟并未显示（例如来自外部接口和外设的时钟）。

表 55 醒目地显示了 RCC 提供的内核时钟（灰色阴影单元格），以及控制内核时钟选择器的位字段。列出了每个 MUX 和每个总线时钟输出的最大允许频率。

用户需遵循这些要求。

表 55. 外设时钟分配概述

外设	时钟类型		类型 (1)	最大频率 [MHz]	时钟复用器	复用器位置	时钟源
ADC1、ADC2	内核	←	A	133 (2)	ADC1SRC、ADC2SRC	0	pll4_r_ck
						1	per_ck
						2	pll3_q_ck
	总线	←	-	209	-	-	hclk2
BSEC	总线	←	-	67	-	-	pclk5
CRC1	总线	←	-	266	-	-	hclk6
CRYP1	总线	←	-	266	-	-	hclk5
DCMIPP	内核	←	A	266	DCMIPPSRC	0	aclk
						1	pll2_q_ck
						2	pll4_p_ck
						3	per_ck
	总线	←	A	133	-	-	pclk4
	总线	←	-	266	-	-	aclk
DDRC	内核	←	A	533	-	-	ker_ddrc_ck
	总线	←	-	133	-	-	pclk4
				266	-	-	aclk
DDRPERFM	内核	←	A	533	-	-	ker_ddrc_ckg
	总线	←	-	133	-	-	pclk4
DDRPHYC	内核	←	A	533	-	-	pll2_r_ck
	总线	←	-	133	-	-	pclk4
DFSDM	内核	←	A	133	SAI1SRC	0	pll4_q_ck
						1	pll3_q_ck
						2	I2S_CKIN
						3	per_ck
						4	pll3_r_ck
		←	S	209	-	-	mlhclk
	总线	←	-	104.5	-	-	pclk2
DLYBSD1、DLYBSD2	总线	←	-	266	-	-	hclk6
DLYBSD3	总线	←	-	209	-	-	hclk2
DLYBQS	总线	←	-	266	-	-	hclk6
DMA1、DMA2	总线	←	-	209	-	-	mlhclk
							hclk1
							hclk2
DMA3	总线	←	-	209	-	-	mlhclk
							hclk2
DMAMUX1、DMAMUX2	总线	←	-	209	-	-	hclk2
DTS	内核	←	A	10	-	-	lse_ck
	总线	←	-	104.5	-	-	pclk3

表 55. 外设时钟分配概述（续）

外设	时钟类型		类型 (1)	最大频率 [MHz]	时钟复用器	复用器位置	时钟源
ETH1、ETH2	内核	←	A	125	ETH1SRC、ETH2SRC	**0**	pll4_p_ck
						1	pll3_q_ck
	总线	←	-	266	-	-	hclk6
							aclk
ETZPC	总线	←	-	133	-	-	pclk5
EXTI	总线	←	-	209	-	-	hclk4
FDCAN	内核	←	A	100	FDCANSRC	**0**	hse_ker_ck
						1	pll3_q_ck
						2	pll4_q_ck
						3	pll4_r_ck
	总线	←	-	104.5	-	-	pclk2
FMC	内核	←	A	266	FMCSRC	**0**	aclk
						1	pll3_r_ck
						2	pll4_p_ck
						3	per_ck
	总线	←	-	266	-	-	hclk6
							aclk
GPIOA-I	总线	←	-	209	-	-	hclk4
HASH1	总线	←	-	266	-	-	hclk5
HDP	总线	←	-	104.5	-	-	pclk3
I2C1、I2C2	内核	←	A	104.5	I2C12SRC	**0**	pclk1
						1	pll4_r_ck
						2	hsi_ker_ck
						3	csi_ker_ck
	总线	←	-	104.5	-	-	pclk1
I2C3、I2C4 和 I2C5	内核	←	A	104.5	I2C3SRC、I2C4SRC 和 I2C5SRC	**0**	pclk6
						1	pll4_r_ck
						2	hsi_ker_ck
						3	csi_ker_ck
	总线	←	-	104.5	-	-	pclk6
IWDG1	内核	←	A	1	-	-	lsi_ck
	总线	←	-	133	-	-	pclk5
IWDG2	内核	←	A	1	-	-	lsi_ck
	总线	←	-	133	-	-	pclk4

表 55. 外设时钟分配概述（续）

外设	时钟类型		类型 (1)	最大频率 [MHz]	时钟复用器	复用器位置	时钟源
LPTIM1	内核	←	A	104.5	LPTIM1SRC	**0**	pclk1
						1	pll4_p_ck
						2	pll3_q_ck
						3	lse_ck
						4	lsi_ck
						5	per_ck
	总线	←	-	104.5	-	-	pclk1
LPTIM2、LPTIM3	内核	←	A	104.5	LPTIM2SRC、LPTIM3SRC	**0**	pclk3
						1	pll4_q_ck
						2	per_ck
						3	lse_ck
						4	lsi_ck
	总线	←	-	104.5	-	-	pclk3
LPTIM4、LPTIM5	内核	←	A	104.5	LPTIM45SRC	**0**	pclk3
						1	pll4_p_ck
						2	pll3_q_ck
						3	lse_ck
						4	lsi_ck
						5	per_ck
	总线	←	-	104.5	-	-	pclk3
LTDC	内核	←	A	90	-	-	pll4_q_ck
	总线	←	-	133	-	-	pclk4
				266	-	-	aclk
DDRMCE	总线	←	-	266	-	-	hclk6
				266	-	-	aclk
PKA	总线	←	-	266	-	-	hclk5
PWR	总线	←	-	209	-	-	hclk4
QUADSPI	内核	←	A	266	QSPISRC	**0**	aclk
						1	pll3_r_ck
						2	pll4_p_ck
						3	per_ck
	总线	←	-	266	-	-	hclk6
							aclk
RNG1	内核	←	A	360	RNG1SRC	**0**	csi_ker_ck
						1	pll4_r_ck
						2	保留
						3	lsi_ck
	总线	←	-	266	-	-	hclk5

表 55. 外设时钟分配概述（续）

外设	时钟类型		类型 (1)	最大频率 [MHz]	时钟复用器	复用器位置	时钟源
RTC/WUT(3)	内核	←	A	4	RTCSRC	1	lse_ck
						2	lsi_ck
						3	hse_ker_ck/(RTCDIV+1)
	总线	←	-	133	-	-	pclk5
SAES	内核	←	A	360	SAESSRC	**0**	axiss_ck
						1	per_ck
						2	pll4_r_ck
						3	lsi_ck
	总线	←	-	266	-	-	hclk5
SAI1	内核	←	A	133	SAI1SRC	**0**	pll4_q_ck
						1	pll3_q_ck
						2	I2S_CKIN
						3	per_ck
						4	pll3_r_ck
	总线	←	-	104.5	-	-	pclk2
SAI2	内核	←	A	133	SAI2SRC	**0**	pll4_q_ck
						1	pll3_q_ck
						2	I2S_CKIN
						3	per_ck
						4	spdif_ck_symb
						5	pll3_r_ck
	总线	←	-	104.5	-	-	pclk2
SDMMC1、SDMMC2	内核	←	A	266	SDMMC1SRC、SDMMC2SRC	0	hclk6
						1	pll3_r_ck
						2	pll4_p_ck
						3	hsi_ker_ck
	总线	←	-	266	-	-	hclk6
SPDIFRX	内核	←	A	200	SPDIFSRC	**0**	pll4_p_ck
						1	pll3_q_ck
						2	hsi_ker_ck
	总线	←	-	104.5	-	-	pclk1
SPI/I2S1	内核	←	A	200	SPI1SRC	**0**	pll4_p_ck
						1	pll3_q_ck
						2	I2S_CKIN
						3	per_ck
						4	pll3_r_ck
	总线	←	-	104.5	-	-	pclk2

表 55. 外设时钟分配概述（续）

外设	时钟类型		类型 (1)	最大频率 [MHz]	时钟复用器	复用器位置	时钟源
SPI/I2S4	内核	←	A	104.5	SPI4SRC	**0**	pclk6
						1	pll4_q_ck
						2	hsi_ker_ck
						3	csi_ker_ck
						4	hse_ker_ck
						5	I2S_CKIN
	总线	←	-	104.5	-	-	pclk6
SPI5	内核	←	A	104.5	SPI5SRC	**0**	pclk6
						1	pll4_q_ck
						2	hsi_ker_ck
						3	csi_ker_ck
						4	hse_ker_ck
	总线	←	-	104.5	-	-	pclk6
SPI/I2S3、SPI/I2S2	内核	←	A	200	SPI23SRC	**0**	pll4_p_ck
						1	pll3_q_ck
						2	I2S_CKIN
						3	per_ck
						4	pll3_r_ck
	总线	←	-	104.5	-	-	pclk1
STGEN	内核	←	A	64	STGENSRC	**0**	hsi_ker_ck
						1	hse_ker_ck
	总线	←	-	133	-	-	pclk4
							pclk5
SYSCFG	总线	←	-	104.5	-	-	pclk3
TIM1、TIM8	内核	←	S	209	-	-	timg2_ck
	总线	←	-	104.5	-	-	pclk2
TIM[7:2]	内核	←	S	209	-	-	timg1_ck
	总线	←	-	104.5	-	-	pclk1
TIM[17:12]	内核	←	S	209	-	-	timg3_ck
	总线	←	-	104.5	-	-	pclk6
USART1、USART2	内核	←	A	104.5	UART1SRC、UART2SRC	**0**	pclk6
						1	pll3_q_ck
						2	hsi_ker_ck
						3	csi_ker_ck
						4	pll4_q_ck
						5	hse_ker_ck
	总线	←	-	104.5	-	-	pclk6

表 55. 外设时钟分配概述（续）

外设	时钟类型		类型 (1)	最大频率 [MHz]	时钟复用器	复用器位置	时钟源
USART3、UART4、UART5 USART7、UART8	内核	←	A	104.5	UART35SRC、UART4SRC、UART78SRC	**0**	pclk1
						1	pll4_q_ck
						2	hsi_ker_ck
						3	csi_ker_ck
						4	hse_ker_ck
	总线	←	-	104.5	-	-	pclk1
USART6	内核	←	A	104.5	UART6SRC	**0**	pclk2
						1	pll4_q_ck
						2	hsi_ker_ck
						3	csi_ker_ck
						4	hse_ker_ck
	总线	←	-	104.5	-	-	pclk2
USBH	内核	←	A	60	-	-	rcc_ck_usbh_free
				60			rcc_ck_usbh_60m
				48			rcc_ck_usbh_48m
				12			rcc_ck_usbh_12m
	总线	←	-	266	-	-	hclk6
USBPHY	内核	←	A	38.4	USBPHYSRC	**0**	hse_ker_ck
						1	pll4_r_ck
						2	hse_ker_ck/2
	总线	←	-	133	-	-	pclk4
USBO	内核	←	A	50	USBOSRC	**0**	pll4_r_ck
						1	rcc_ck_usbo_48m
	内核	←	A	60	-	-	rcc_ck_usbo_60m
	总线	←	-	209	-	-	hclk2
VREFBUF	总线	←	-	104.5	-	-	pclk3

1. “**A**”表示内核时钟与总线接口时钟异步。
“**S**”表示内核时钟与总线接口时钟同步。

2. 对于 ADC1 和 ADC2，如果需要占空比为 50% 的时钟，则最大频率限制为 75 MHz。要获得 50% 的占空比，DIVQ 值必须是奇数，或者必须使用 ADC 内的预分频器。有关详细信息，请参见 *STGEN 和 ADC 的时钟分配*一节。如果同时使用两个 ADC，则对于这两个 ADC，内核时钟应相同，并且不能使用嵌入式 ADC 预分频器。
对于 SDMMCx，支持 DDR 时需要 50% 的占空比，如果 RCC 不能提供具有该占空比的时钟，则应用必须使用 SDMMCx 内部分频器。

3. RTC 开关位于 V_{SW} 电压域内。

针对每一个时钟源，下面的*表 56* 列出了内核时钟由（或者可由）该时钟源驱动的外设，具体取决于编程的选项（如果有）。

表 56. 内核时钟分配与外设

内核时钟源	连接的外设
hse_ker_ck	FDCAN、RTC/WUT、SPI4、SPI5、STGEN、USART1、USART2、USART3、UART4、UART5、USART6、USART7、UART8、USBPHY(1) 通过 per_ck：ADC1(2)、ADC2(2)、DCMIPP、DFSDM、FMC、LPTIM1、LPTIM2、LPTIM3、LPTIM4、LPTIM5、QUADSPI、SAES、SAI1、SAI2、SPI/I2S1、SPI/I2S2、SPI/I2S3
hsi_ker_ck	I2C1、I2C2、I2C3、I2C4、I2C5、I2C6、SDMMC1、SDMMC2、SPDIFRX、SPI/I2S4、SPI5、STGEN、USART1、USART2、USART3、UART4、UART5、USART6、USART7、UART8 通过 per_ck：ADC1(2)、ADC2(2)、DCMIPP、DFSDM、FMC、LPTIM1、LPTIM2、LPTIM3、LPTIM4、LPTIM5、QUADSPI、SAES、SAI1、SAI2、SPI/I2S1、SPI/I2S2、SPI/I2S3
csi_ker_ck	DTS、I2C1、I2C2、I2C3、I2C4、I2C5、I2C6、SPI/I2S4、SPI5、RNG1、RTC/WUT、USART1、USART2、USART3、UART4、UART5、USART6、UART7、UART8 通过 per_ck：ADC1(2)、ADC2(2)、DCMIPP、DFSDM、FMC、LPTIM1、LPTIM2、LPTIM3、LPTIM4、LPTIM5、QUADSPI、SAES、SAI1、SAI2、SPI/I2S1、SPI/I2S2、SPI/I2S3
lse_ck	DTS、LPTIM1、LPTIM2、LPTIM3、LPTIM4、LPTIM5、RTC/WUT
lsi_ck	IWDG1、IWDG2、LPTIM1、LPTIM2、LPTIM3、LPTIM4、LPTIM5、RNG1、RNG2、SAES
spdif_ck_symb	SAI2
per_ck	ADC1(2)、ADC2(2)、DCMIPP、DFSDM、FMC、LPTIM1、LPTIM2、LPTIM3、LPTIM4、LPTIM5、QUADSPI、SAES、SAI1、SAI2、SPI/I2S1、SPI/I2S2、SPI/I2S3
I2S_CKIN	DFSDM、SPI/I2S1、SPI/I2S2、SPI/I2S3、SPI/I2S4、SAI1、SAI2
pll2_q_ck	DCMIPP
pll2_r_ck	DDRPHYC
pll3_q_ck	ADC1(2)、ADC2(2)、DFSDM、ETH1、ETH2、FDCAN、LPTIM1、LPTIM4、LPTIM5、SAI1、SAI2、SPDIFRX、SPI/I2S1、SPI/I2S2、SPI/I2S3、I2C4、I2C6、USART1、USART2
pll3_r_ck	DFSM、FMC、QUADSPI、SAI1、SAI2、SDMMC1、SDMMC2、SPI/I2S1、SPI/I2S2、SPI/I2S3
pll4_p_ck	ET1、ETH2、FMC、LPTIM1、LPTIM4、LPTIM5、QUADSPI、SDMMC1、SDMMC2、SPDIFRX、SPI/I2S1、SPI/I2S2、SPI/I2S3
pll4_q_ck	DCMIPP、DFSDM、FDCAN、LPTIM2、LPTIM3、LTDC、SAI1、SAI2、SPI/I2S4、SPI5、USART1、USART2、USART3、UART4、UART5、USART6、USART7、UART8
pll4_r_ck	ADC1、ADC2、FDCAN、I2C1、I2C2、I2C3、I2C4、I2C5、RNG1、SAES、USBPHY、USBO
pclk1	I2C1、I2C2、LPTIM1、USART3、UART4、UART5、UART7、UART8
pclk2	USART6
pclk3	LPTIM2、LPTIM3、LPTIM4、LPTIM5
pclk4	
pclk5	
pclk6	I2C3、I2C4、I2C5、SPI/I2S4、SPI5、USART1、USART2
hclk1	
hclk2	
hclk4	
hclk5	
hclk6	SDMMC1、SDMMC2
mlhclk (APB1)	TIM2、TIM3、TIM4、TIM5、TIM6、TIM7
mlhclk (APB2)	DFSDM、TIM1、TIM8
mlhclk (APB6)	TIM12、TIM13、TIM14、TIM15、TIM16、TIM17
aclk	DCMIPP、FMC、QUADSPI

1. USBPHY 接收 **hse_ker_ck** 和 **hse_ker_ck**/2 时钟。
2. 如果同时使用两个 ADC，则对于这两个 ADC，内核时钟应相同，并且不能使用嵌入式 ADC 预分频器。

下面的章节提供了内核时钟和总线接口时钟分配的更详细视图。有关详细信息，请参见*第 10.4.11 节：外设时钟门控控制*。

为减少开关数，一些外设共用同一内核时钟源。不过，所有外设均具有各自的专用使能信号。

SAI、SPI/I2S、SPDIFRX 和 DFSDM 的时钟分配

音频外设通常需要特定的精确频率（SPDIFRX 除外）。如*图 72* 和*图 73* 所示，SAI 或 SPI(I2S) 的内核时钟可以由以下各项生成：

- PLL3 或 PLL4，可在频率生成方面实现最佳灵活性
- HSE、HSI 或 CSI，适用于对电流消耗极为敏感的用例
- I2S_CKIN（当需要使用外部时钟参考时）。DFSDM 可以与 SAI1 共用同一个内核时钟：这在 DFSDM 用于音频应用时很有用。

为正常工作，DFSDM 总是需要 **dfsdm_clk** 和 **fdsdm_pclk** 时钟。

时钟 **dfsdm_Aclk** 可用作 DFSDM_CKOUT 的时钟源。时钟 DFSDM_CKOUT 可由 **dfsdm_clk**（默认）或 **dfsdm_Aclk** 生成。

有关详细信息，请参见*第 29 章：∑Δ 调制器数字滤波器 (DFSDM)*。

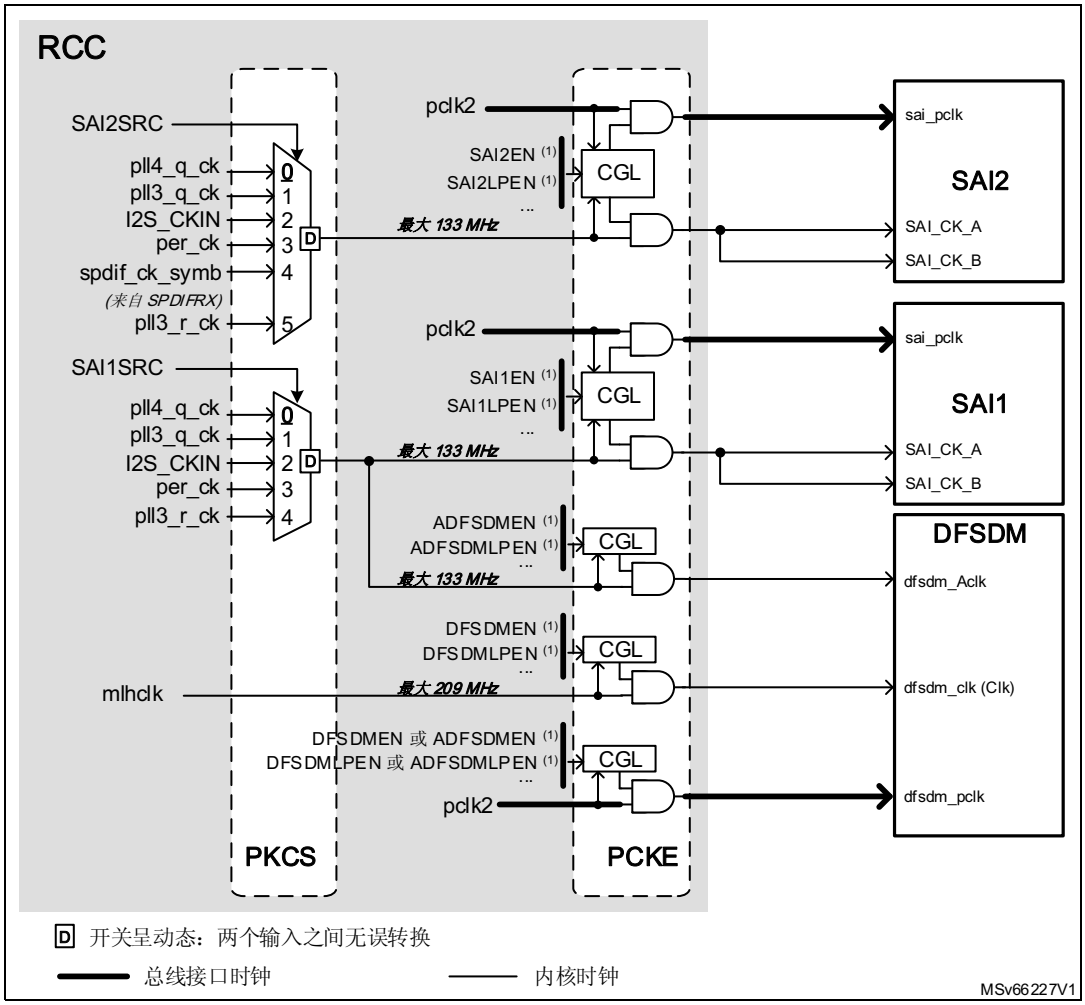
PLL3 的时钟设置示例

下表列出了使用 PLL3 生成内核时钟的编程示例。可以通过分数或整数模式的 PLL3 获得精度良好的音频频率（灰色单元格）。PLL3 还用于为 MLAHB 提供时钟（PLL3P 输出）。

表 57. 音频应用的内核时钟设置示例

配置	F_{HSE}	F_{ref3_ck}	F_{VCO}	DIVM+1	DIVN+1	FRACV	DIVP+1	DIVQ+1	DIVR+1	PLL3P 输出	PLL3Q 输出		PLL3R 输出	
											频率	48 kHz 的误差	频率	44.1 kHz 的误差
#1	24	12	417.71	2	34	6632	2	17	37	208.9	24.5715	-185	11.2896	-1
#2	24	12	417.79	2	34	6685	2	17	37	208.9	24.5760	1	11.2917	185
#3	24	12	417.75	2	34	6659	2	17	37	208.9	24.5738	-90	11.2907	94
#4	24	4.8	614.40	5	128	0	4	25	-	153.6	24.576	0	-	-
#5	24	4.8	609.6	5	127	0	4	-	27	152.4	-	-	22.5778	-63
	MHz	**MHz**	**MHz**							**MHz**	**MHz**	**ppm**	**MHz**	**ppm**

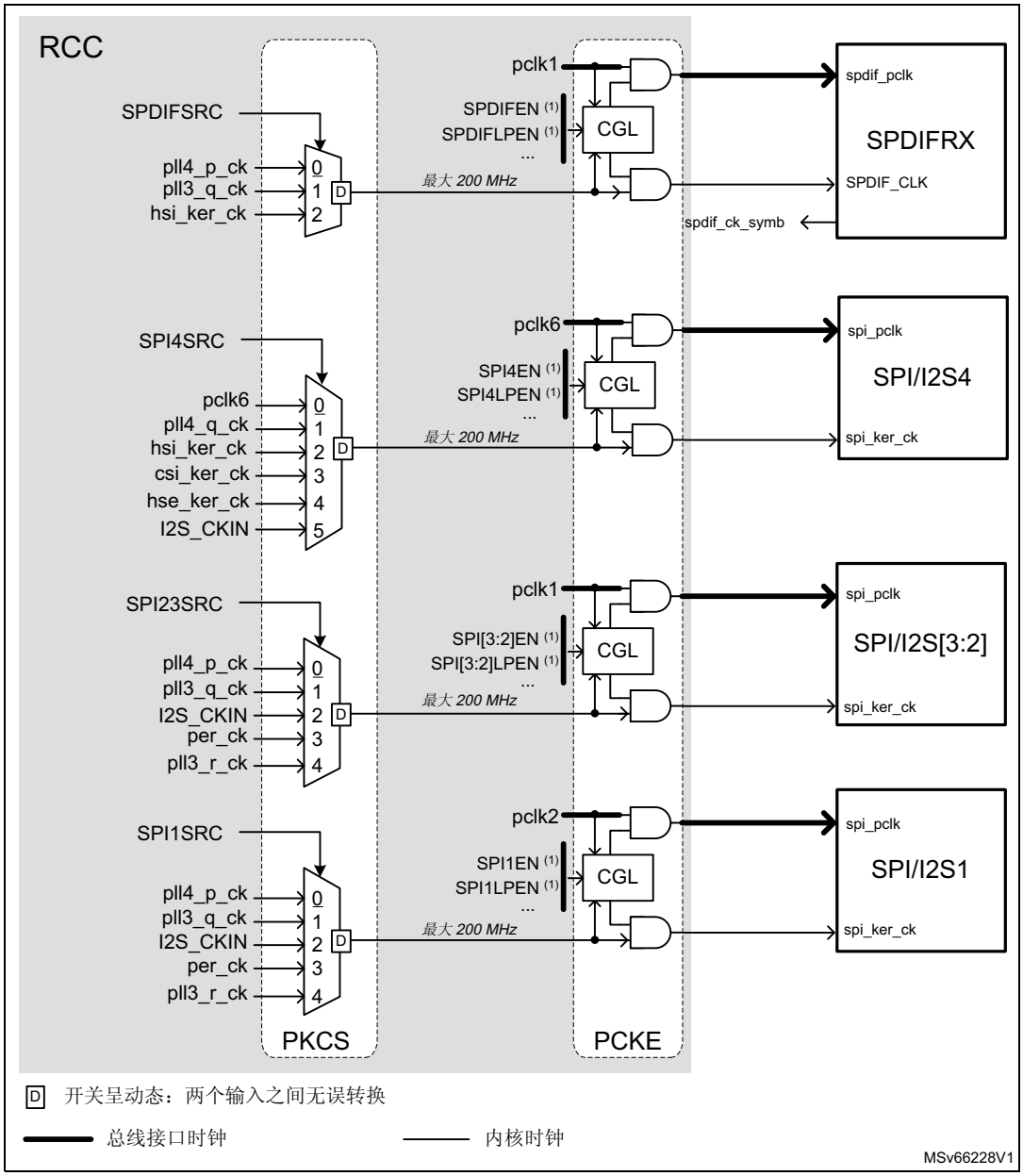
图 71. SAI 和 DFSDM 的外设时钟分配



1. MPU 有一个使能位和一个低功耗使能位。
   X 表示系统复位后所选择的复用器输入

注： *SPDIFRX 无需特定频率，只需一个足够高的内核时钟频率即可使外设正常工作。有关详细信息，请参见 SPDIFRX 的说明。*

图 72. SPI/I2S 和 SPDIFRX 的外设时钟分配



1. MPU 有一个使能位和一个低功耗使能位。
   X 表示系统复位后所选择的复用器输入。

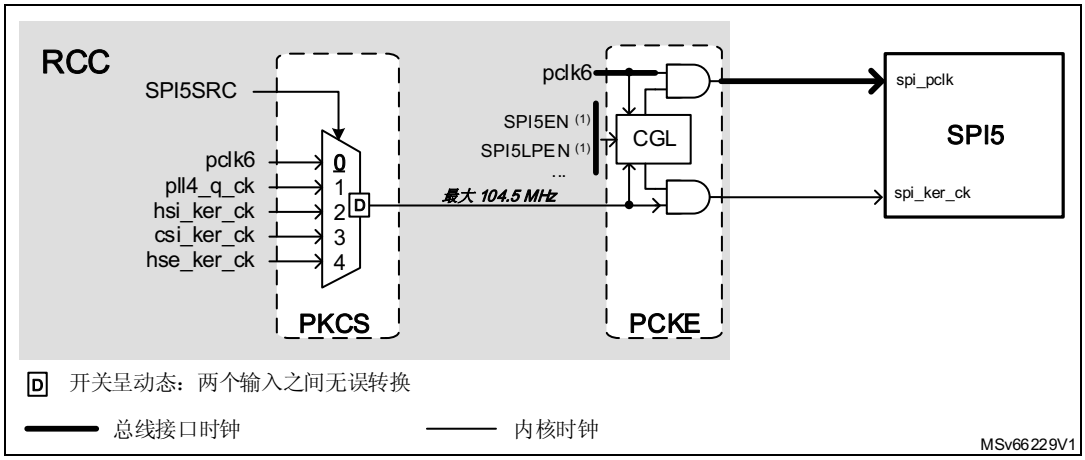
有关时钟门控逻辑 (CGL) 的行为的详细信息，请参见*表 63*。

SPI 的时钟分配

通常，SPI（或以 SPI 模式使用的 SPI/I2S）不需要特定的内核时钟频率，而是需要一个足够快的时钟来生成正确的数据速率。为此，通常可以从以下目标选择源：

- APB 总线，可用于避免激活其他 PLL，
- PLL4（如果需要提高灵活性），例如，它允许通过 PLL3 更改 MLAHB 总线的频率，而不影响某些串行接口的速度，
- HSI 和 CSI，适用于低功耗应用场景。

图 73. SPI 的外设时钟分配



1. MPU 有一个使能位和一个低功耗使能位。
   X 表示系统复位后所选择的复用器输入。

有关时钟门控逻辑 (CGL) 的行为的详细信息，请参见*表 63*。

I2C、UART 和 USART 的时钟分配

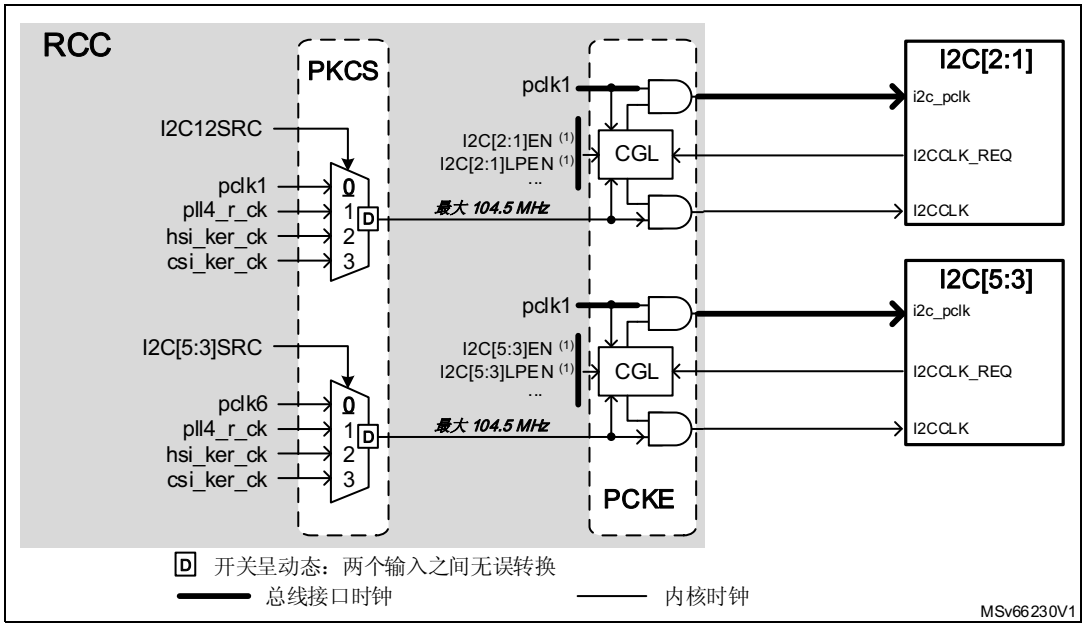
如前所述，SPI、I2C 和 UART 不需要特定内核时钟频率，但时钟频率应足以生成合适的波特率，或串行链路上所需的位时钟。为此，通常可以从以下目标选择源：

- PLL3，可用来避免激活其他 PLL，
- PLL4（如果需要提高灵活性），例如，它允许通过 PLL3 更改 MLAHB 总线的频率，而不影响某些串行接口的速度，
- HSI、CSI，适用于低功耗应用场景，也适用于外设必须快速从 Stop 模式唤醒的场景（例如 UART、I2C）。

请注意，UART 还接收 LSE 时钟，适用于不需要高波特率的应用场景。

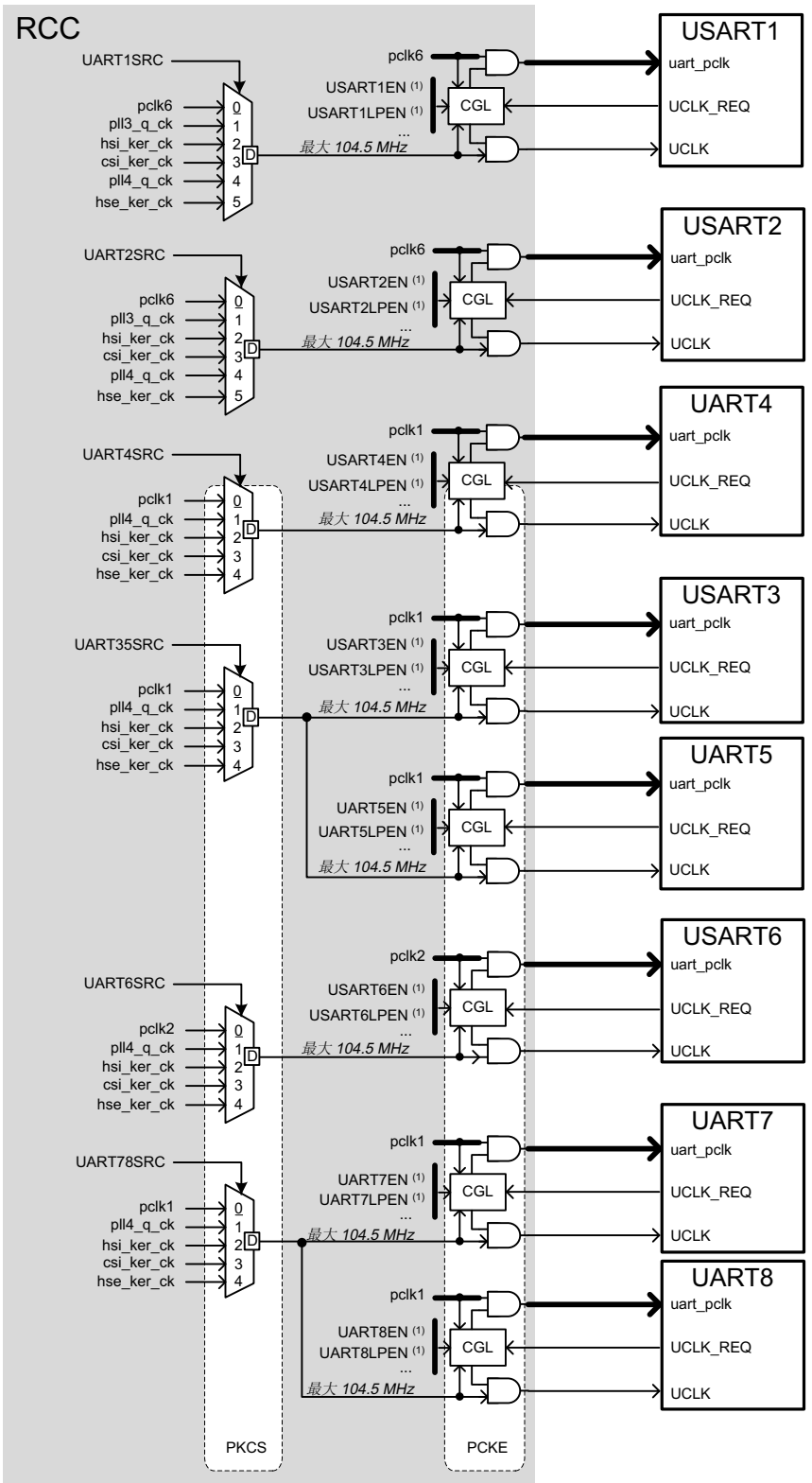
如*图 74* 和*图 75* 所示，I2C 和 UART 提供内核时钟请求信号（I2CCLK_REQ 和 UCLK_REQ）。当系统处于 Stop 模式时，可以使用这些信号向 RCC 请求选定的内核时钟。该功能仅在选择了 **hsi_ker_ck**、**csi_ker_ck** 或 **hse_ker_ck** 时有效。有关详细信息，请参见*第 10.4.11 节：外设时钟门控控制*。

图 74. I2C 的外设时钟分配



1. MPU 有一个使能位和一个低功耗使能位。
   X 表示系统复位后所选择的复用器输入。

图 75. UART 和 USART 的外设时钟分配



1. MPU 有一个使能位和一个低功耗使能位。
   X 表示系统复位后所选择的复用器输入。

注： *为了使 I2C 或 U(S)ART 能在 Stop 模式下工作，用户必须选择一个振荡器作为内核时钟：**hse_ker_ck**、**hsi_ker_ck** 或 **csi_ker_ck**。必须根据应用需求对 HSEKERON、HSIKERON 和 CSIKERON 位进行编程。*

有关时钟门控逻辑 (CGL) 的行为的详细信息，请参见*表 63*。

TIM 和 LPTIM 的时钟分配

TIM 定时器使用与总线接口时钟相位对齐的内核时钟 (**timg[3:1]_ck**)。

定时器时钟的频率取决于：

- 与定时器所连总线对应的 APB 预分频器
- 位 TIMG[3:1]PRE。

下表显示了如何选择定时器时钟频率。

表 58. 时钟定时器与 pclk 之比

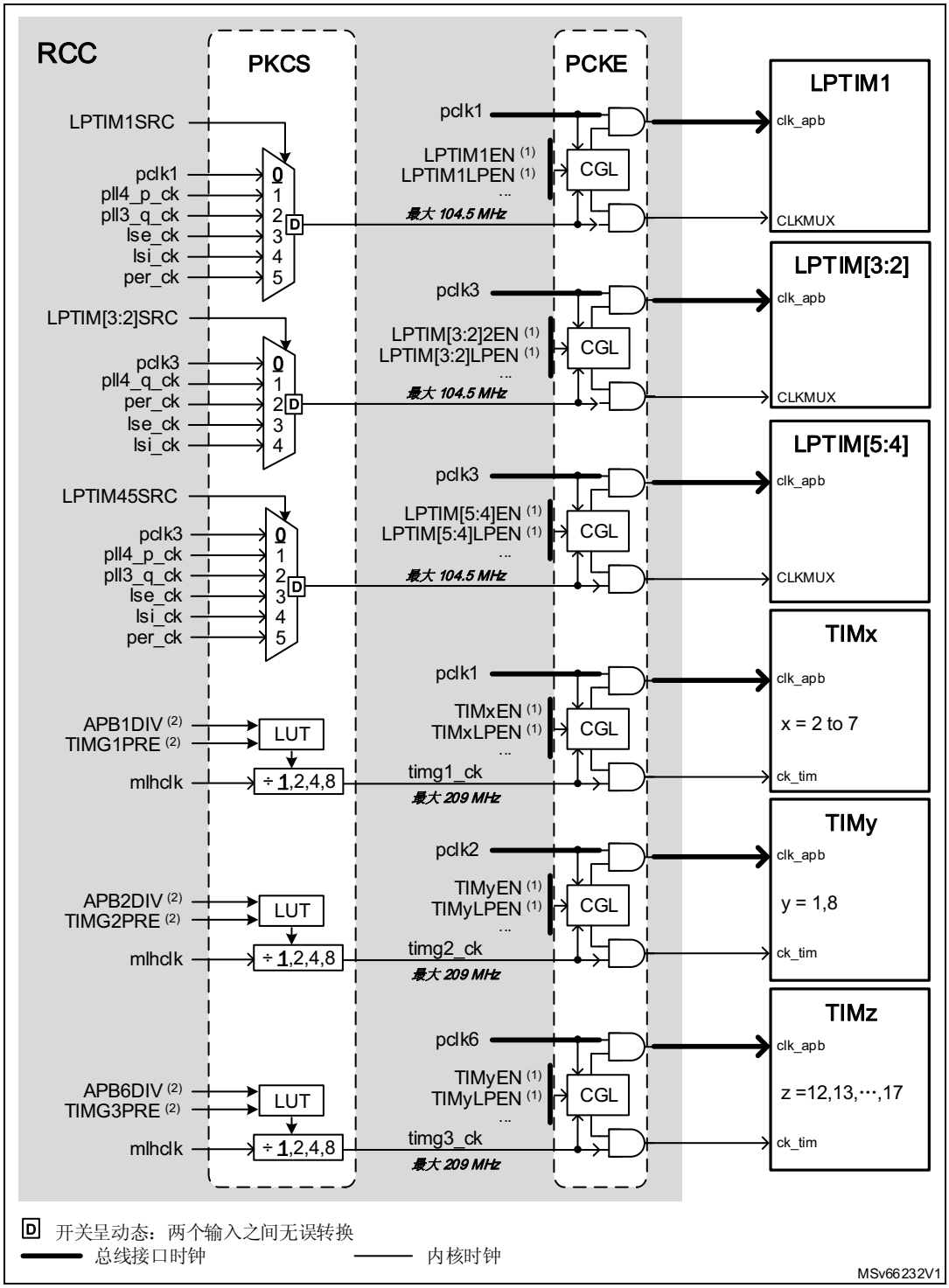
APB1DIV APB2DIV APB6DIV (1)	TIMG1PRE TIMG2PRE TIMG3PRE(2)	-	F_{timg1_ck} F_{timg2_ck} F_{timg3_ck}	F_{pclk1} F_{pclk2} F_{pclk6}	备注
“000”	x	→	F_{mlhclk}	F_{mlhclk}	定时器时钟等效于总线时钟。
“001”	x	→	F_{mlhclk}	$F_{mlhclk}/2$	定时器时钟频率是总线时钟频率的 2 倍。
“010”	“0”	→	$F_{mlhclk}/2$	$F_{mlhclk}/4$	
“011”	“0”	→	$F_{mlhclk}/4$	$F_{mlhclk}/8$	
“100”	“0”	→	$F_{mlhclk}/8$	$F_{mlhclk}/16$	
“010”	“1”	→	F_{mlhclk}	$F_{mlhclk}/4$	定时器时钟频率是总线时钟频率的 4 倍。
“011”	“1”	→	$F_{mlhclk}/2$	$F_{mlhclk}/8$	
“100”	“1”	→	$F_{mlhclk}/4$	$F_{mlhclk}/16$	

1. APB1DIV、APB2DIV 和 APB6DIV 在 *RCC APB1 时钟分频器寄存器 (RCC_APB1DIVR)*、*RCC APB2 时钟分频器寄存器 (RCC_APB2DIVR)* 和 *RCC APB6 时钟分频器寄存器 (RCC_APB6DIVR)* 中。
2. TIMG1PRE、TIMG2PRE 和 TIMG3PRE 在 *RCC APB1 定时器组 1 预分频器寄存器 (RCC_TIMG1PRER)*、*RCC APB2 定时器组 2 预分频器寄存器 (RCC_TIMG2PRER)* 和 *RCC APB6 定时器组 3 预分频器寄存器 (RCC_TIMG3PRER)* 中。

LPTIM 定时器具有一个与其总线接口时钟完全异步的内核时钟。

此外，LPTIMs 还可以使用 **lse_ck** 和 **lsi_ck** 作为参考时钟，使它们即使在系统处于 Stop 模式时也能工作。

图 76. TIM 和 LPTIM 的外设时钟分配



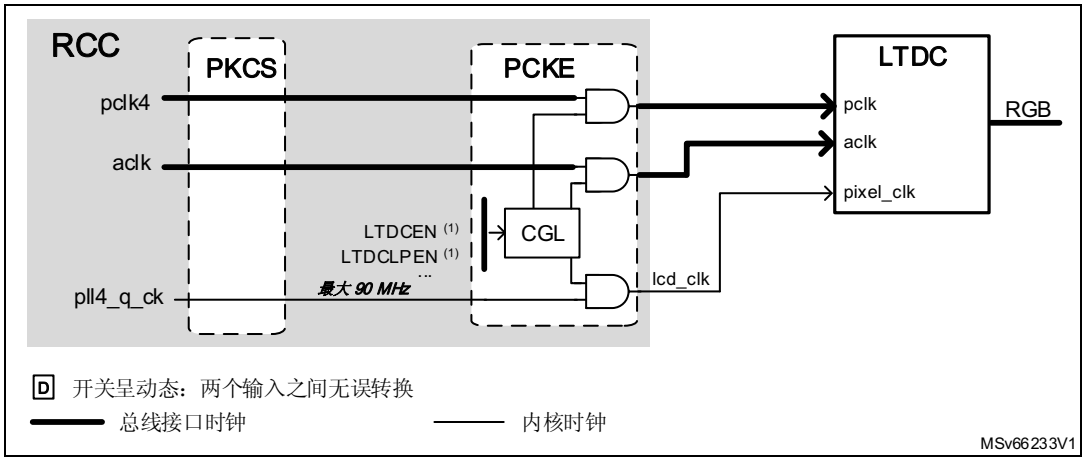
1. MPU 有一个使能位和一个低功耗使能位。
   X 表示系统复位后所选择的复用器输入。
2. 可实时更改。

有关时钟门控逻辑 (CGL) 的行为的详细信息，请参见*表 63*。

LTDC 的时钟分配

LTDC 的时钟由 RCC 提供，如*图 77* 所示。

图 77. LTDC 的外设时钟分配

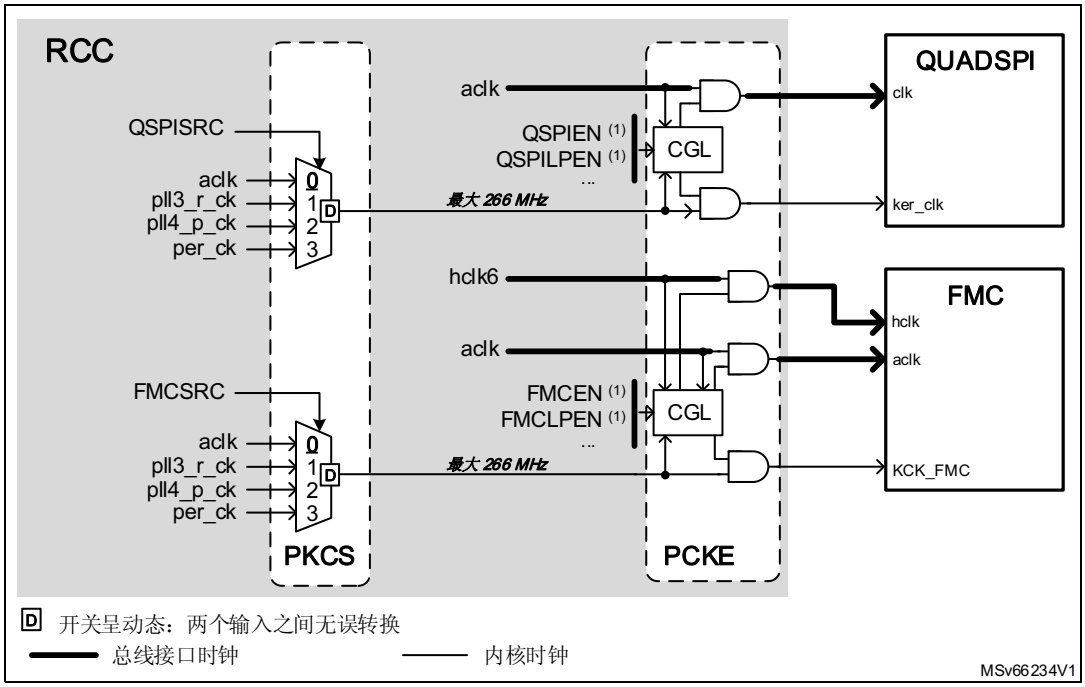


1. MPU 有一个使能位和一个低功耗使能位。
   X 表示系统复位后所选择的复用器输入。

SDMMC、FMC 和 QUADSPI 的时钟分配

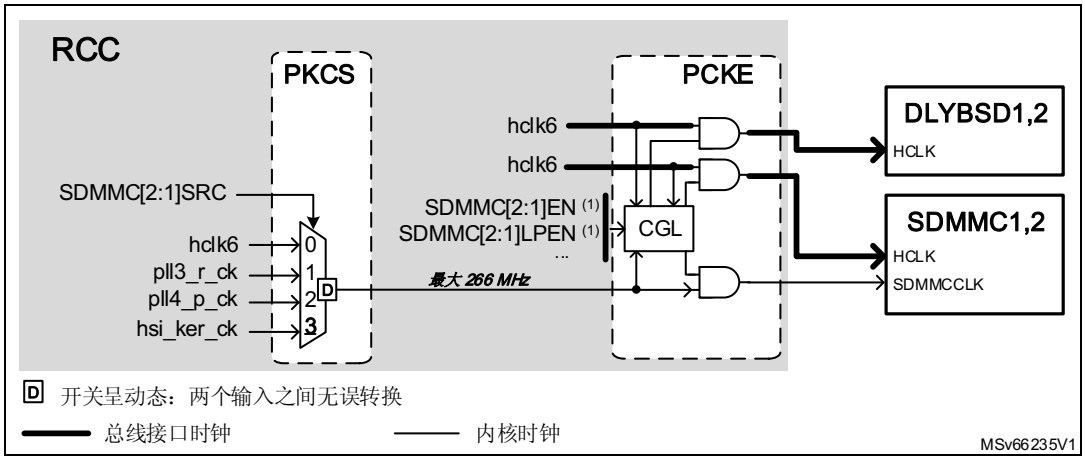
为提高灵活性，FMC 和 QUADSPI 还可以使用与总线接口时钟不同的时钟。

图 78. QUADSPI 和 FMC 的外设时钟分配



1. MPU 有一个使能位和一个低功耗使能位。
   X 表示系统复位后所选择的复用器输入。

图 79. SDMMC 的外设时钟分配



1. MPU 有一个使能位和一个低功耗使能位。
   X 表示系统复位后所选择的复用器输入。

USBH 和 USBO 的时钟分配

图 80 显示了 USB 模块的时钟分配。RCC 必须为 USB PHY 提供一个干净的参考时钟。USB PHY 具有自己的 PLL 并为 USBH (HOST) 和 USBO (OTG) 模块生成内核时钟。USBPHYSRC 所控制的复用器使 USB PHY 可接收两种不同的时钟。

要获得最佳的 USB 时钟质量，最好选择 **hse_ker_ck** 时钟或 **hse_ker_ck**/2。

请注意，应用必须为 USBPHY PLL 提供以下时钟参考：

- 介于 19.2 和 38.4 MHz 之间的时钟频率（如果 USBPHY PLL 工作在整数模式下）
- 介于 26 和 38.4 MHz 之间的时钟频率（如果 USBPHY PLL 工作在小数模式下）

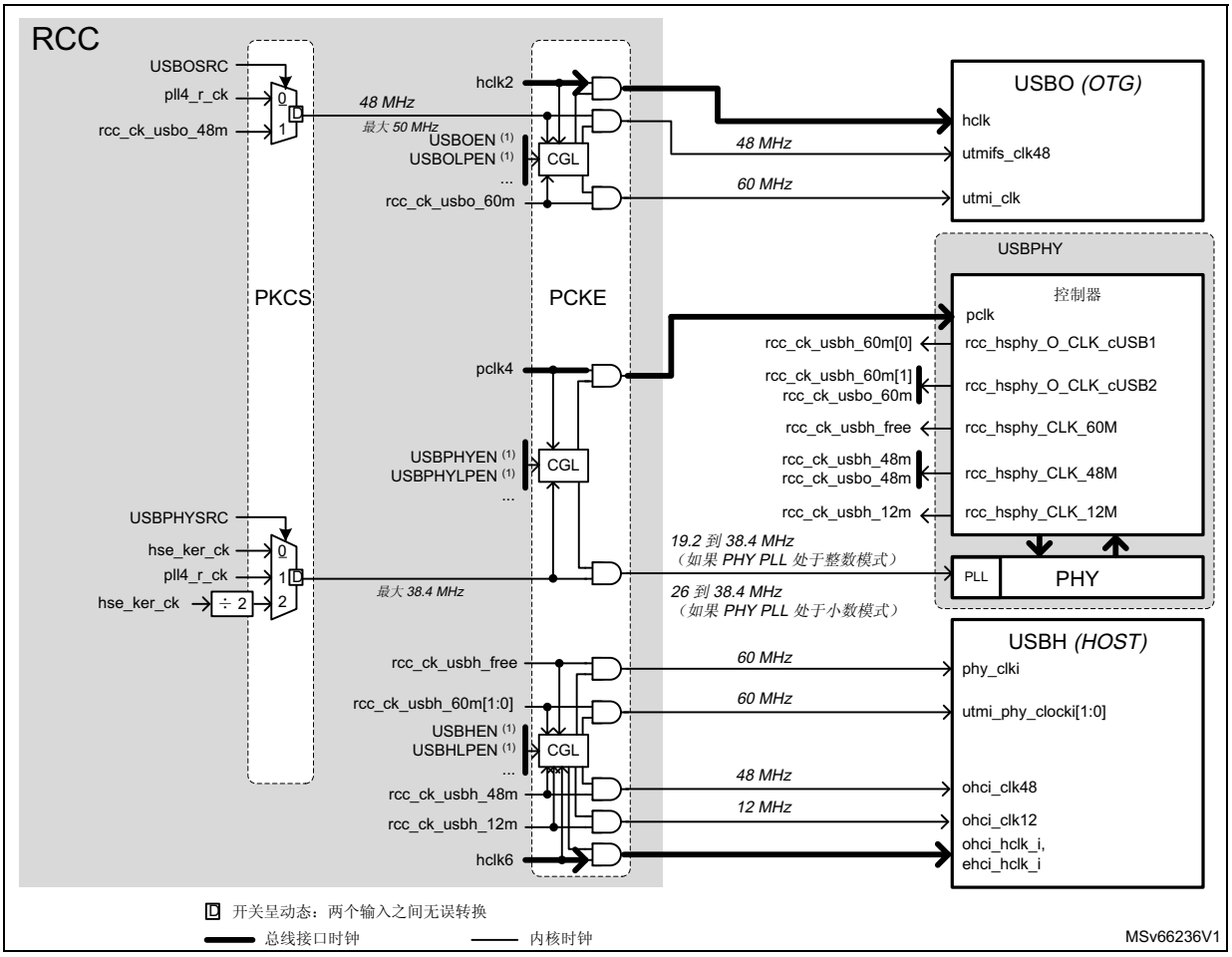
此外，如果 USBO 仅用在全速模式下，则应用可以选择提供给 USBO 48 MHz 的时钟源：

- 由 USBPHY 生成的时钟；在这种情况下，必须使用 PHY 的 PLL。
- **pll4_r_ck** 时钟；在这种情况下，无需使能 PHY 的 PLL。

时钟选择通过 USBOSRC 来执行。

这些配置位在 *RCC USB 内核时钟选择寄存器 (RCC_USBCKSELR)* 中。

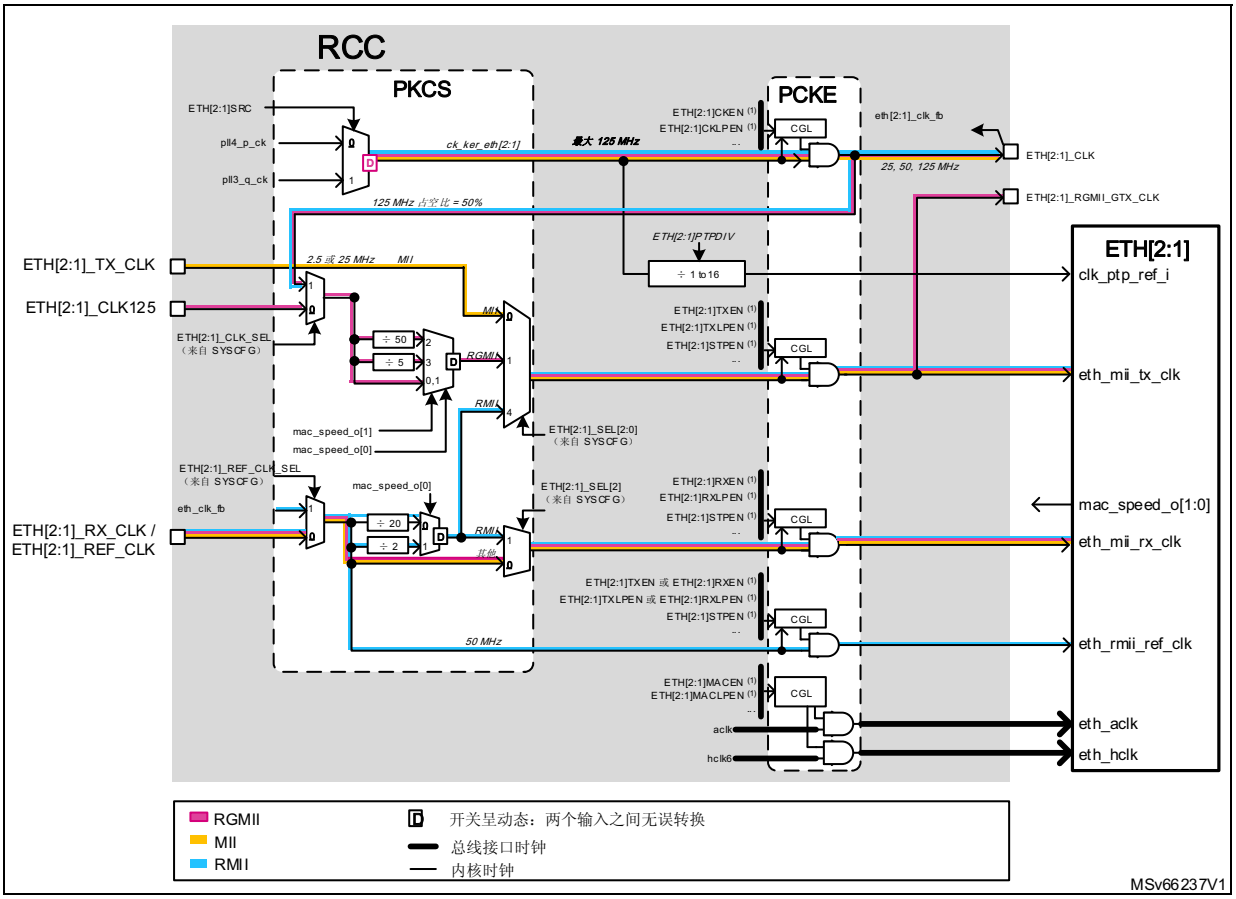
图 80. USB 的外设时钟分配



1. MPU 有一个使能位和一个低功耗使能位。
   X 表示系统复位后所选择的复用器输入。

以太网（ETH1 和 ETH2）的时钟分配

图 81. 以太网的外设时钟分配



1. MPU 有一个使能位和一个低功耗使能位。
   X 表示系统复位后所选择的复用器输入。

有两个以太网模块，ETH1 和 ETH2。每一个都具有相同且独立的配置、引脚、路由和时钟分配。

RCC 所提供的以太网时钟在 ETH[2:1]_CLK 引脚上可用。

RX 和 TX 路径的时钟选择通过 SYSCFG 模块进行控制。

信号 ETH[2:1]_SEL[2:0] 由 SYSCFG 块提供，并定义了所使用的接口类型：

- 在 MII 模式（橙色路径）下，
  - 发送时钟通过 ETH[2:1]_TX_CLK 引脚从外部 PHY 接收。时钟频率为 2.5 或 25 MHz。
  - 接收时钟通过 ETH[2:1]_RX_CLK 引脚从外部 PHY 接收。时钟频率为 2.5 或 25 MHz
  - 在该模式下，输出 ETH[2:1]_CLK 可用于向外部 PHY 提供参考时钟 (25 MHz)。

- 在 RGMII 模式（红色路径）下，
  - 发送时钟可以在 RCC 提供的内部时钟或在 ETH[2:1]_CLK125 引脚上接收到的时钟之间选择。该选择通过 SYSCFG 的 ETH[2:1]_CLK_SEL 控制。根据 ETH[2:1] 模块提供的信号 **mac_speed_o[1:0]**，时钟频率可以动态调整为 2.5、25 或 125 MHz
  - 接收时钟直接来自 ETH[2:1]_RX_CLK 引脚。时钟频率为 2.5、25 或 125 MHz。
  - 发送时钟通过 ETH[2:1]_RGMII_GTX_CLK 引脚提供给 PHY。
  - 还可以通过 ETH[2:1]_CLK 引脚为外部 PHY 提供 25 MHz 的参考时钟。
- 在 RMII 模式（蓝色路径）下
  - 可以在 RCC 提供的内部时钟或在 ETH[2:1]_REF_CLK 引脚上接收到的时钟之间选择连续的参考时钟 (50 MHz)。该选择通过 SYSCFG 的 ETH[2:1]_REF_CLK_SEL 控制。
  - 根据 ETH[2:1] 模块提供的信号 **mac_speed_o[0]**，发送和接收时钟频率可以动态调整为 2.5 或 25 MHz。
  - 还可以通过 ETH[2:1]_CLK 引脚为外部 PHY 提供连续的参考时钟 (50 MHz)。

ETH[2:1] 块提供关于 MAC 速度 (**mac_speed_o[1:0]**) 的信息，MAC 速度可能会动态变化。

mac_speed_o[1:0] 等于：

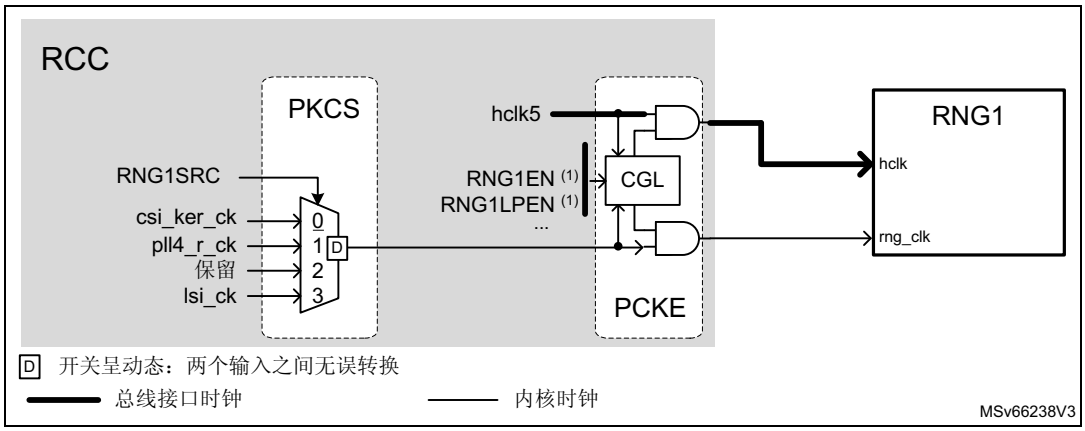
- 0x：对于 1000 Mbps
- 10：对于 10 Mbps
- 11：对于 100 Mbps

应用可以根据处理器模式（CRun、CSleep）控制时钟的门控。

此外，还可以通过 ETH[2:1]STPEN 位，在 CStop 和 Stop 模式下控制来自引脚 ETH[2:1]_RX_CLK 和 ETH[2:1]_TX_CLK 的时钟的门控。也可以通过 ETH[2:1] 块，在 EEE LPI 状态期间对提供给 PHY 的 ETH[2:1]_RGMII_GTX_CLK 时钟进行门控。

RNG1 的时钟分配

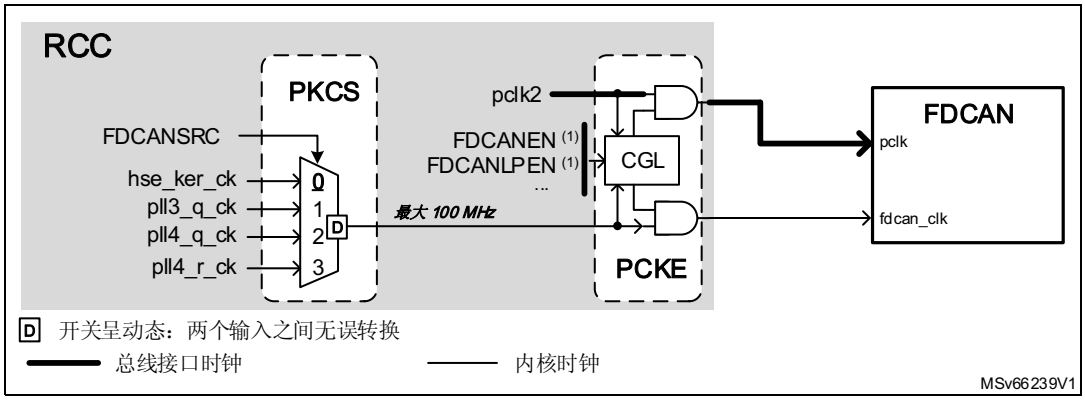
图 82. RNG1 的内核时钟分配



1. MPU 有一个使能位和一个低功耗使能位。
   X 表示系统复位后所选择的复用器输入。

FDCAN 的时钟分配

图 83. FDCAN 的内核时钟分配



1. MPU 有一个使能位和一个低功耗使能位。
   X 表示系统复位后所选择的复用器输入。

STGEN 和 ADC 的时钟分配

STGEN 模块的处理略有特殊：它需要一个单独的时钟用作总线接口和内核时钟。此外，在 Stop 模式下可能需要激活该时钟。

请注意，两个异步总线桥将 STGEN 连接到 APB4 和 APB5 总线，因此如果使能了 STGEN，RCC 必须提供这些 APB 时钟。

可通过两个不同的 APB 端口访问 STGEN：

- APB4 端口允许应用使用一些 STGEN 资源，但该接口为只读。为了通过 APB4 使用 STGEN，提供了使能位：STGENROEN、STGENROLPEN 和 STGENROSTPEN。
- APB5 端口允许应用控制 STGEN 控制器部分（也称为 STGENC）。为了通过 APB5 使用 STGEN，提供了使能位：STGENCEN、STGENCLPEN 和 STGENCSTPEN。

应用可以通过 *RCC STGEN 时钟选择寄存器 (RCC_STGENCKSELR)* 的 STGENSRC 位选择哪个时钟可以作为 STGEN 时钟（HSE 或 HSI）。

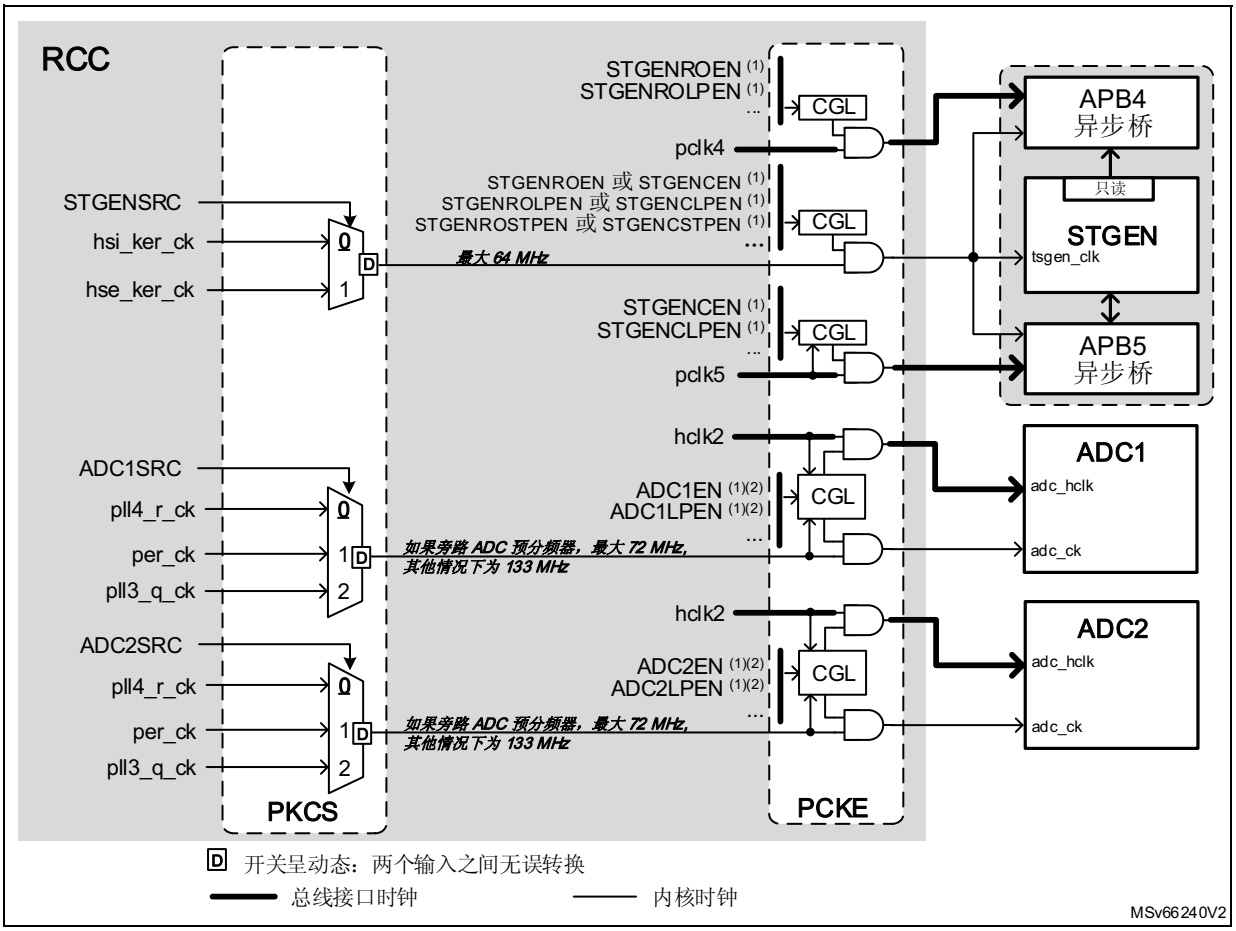
为了确保在系统处于 Stop 模式时向 STGEN 提供时钟，应用必须将 STGENCEN、STGENCLPEN 和 STGENCSTPEN 位置“1”，或将 STGENROEN、STGENROLPEN 和 STGENROSTPEN 位置“1”。此外，必须根据所选的时钟源将 HSEKERON 或 HSIKERON 位置“1”。

无论是 ADC1 还是 ADC2，ADC 都内嵌一个分频器，以便为 ADC 模拟单元提供占空比尽可能接近 50% 的时钟。

- 如果旁路了内嵌的 ADC 分频器，则应由 RCC 来提供一个占空比为 50% 的 **clk_adc[2:1]** 时钟。
  请注意，为了获得占空比为 50% 的 clk_adc[2:1]，后分频器（DIVQ 或 DIVR）必须编程为奇数值。如果使用 hsi_ker_ck，则 HSIDIV[1:0] 不得为 0。
- 如果使用了内嵌的 ADC[2:1] 分频器，则 RCC 可以提供一个频率高达 133 MHz 的时钟。

警告： **如果同时使用两个 ADC，则对于这两个 ADC，内核时钟应相同，并且不能使用嵌入式 ADC 预分频器。**

图 84. ADC 和 STGEN 的内核时钟分配

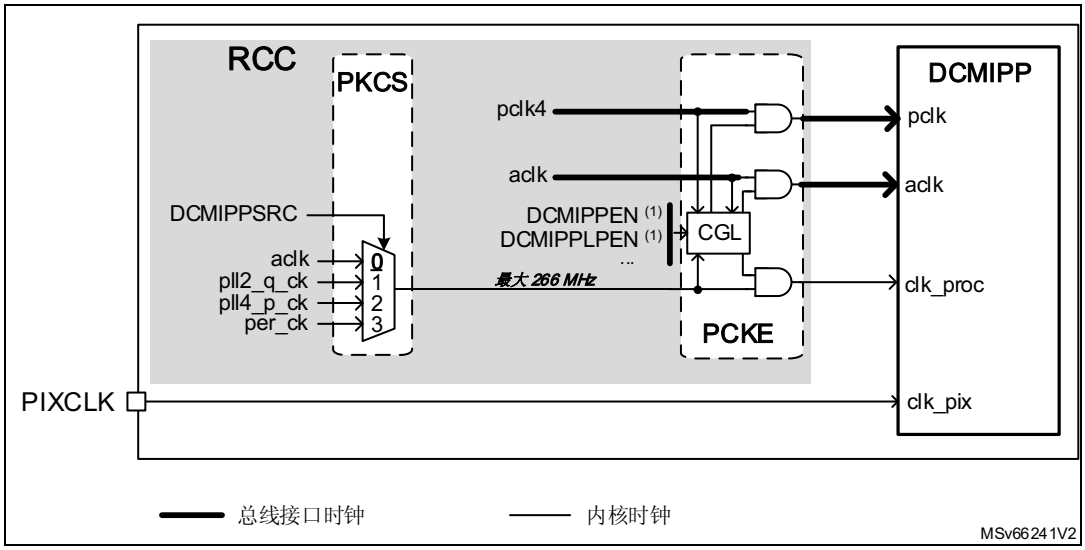


1. MPU 有一个使能位和一个低功耗使能位。
   X 表示系统复位后所选择的复用器输入。
2. 如果同时使用两个 ADC，则对于这两个 ADC，内核时钟应相同，并且不能使用嵌入式 ADC 预分频器。

DCMIPP 的时钟分配

PLL2 的 DIVQ 输出 (**pll2_q_ck**) 专用于 DCMIPP 内核时钟。此外，DCMIPP 内核时钟可以由 aclk、pll4_p_ck 或 per_ck 提供，如下图所示。

图 85. DCMIPP 的内核时钟分配

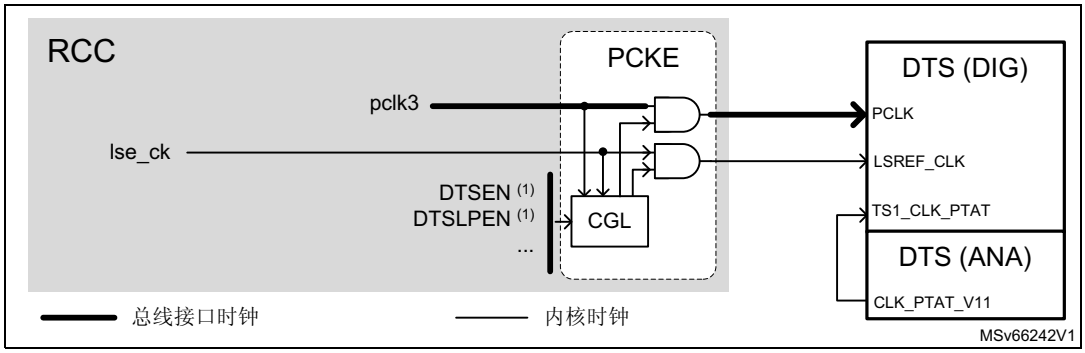


1. MPU 有一个使能位和一个低功耗使能位。

数字温度传感器 (DTS) 的时钟分配

DTS 模块需要一个 APB 时钟和 **lse_ck** 时钟。即使系统处于 Stop 模式，RCC 也能为 DTS 模块提供 lse_ck 时钟。其他相关信息，请参见*第 10.4.11 节：外设时钟门控控制*。

图 86. DTS 的内核时钟分配



1. MPU 有一个使能位和一个低功耗使能位。

RTC/WUT 的时钟分配

rtc_ck 时钟源可为：

- **hse_rtc_ck**（**hse_ck** 经过可编程预分频器分频获得）
- **lse_ck**，或
- **lsi_ck** 时钟。

时钟源通过 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 RTCSRC[1:0] 位和 *RCC RTC 时钟分频寄存器 (RCC_RTCDIVR)* 中的 RTCDIV[5:0] 位进行选择。

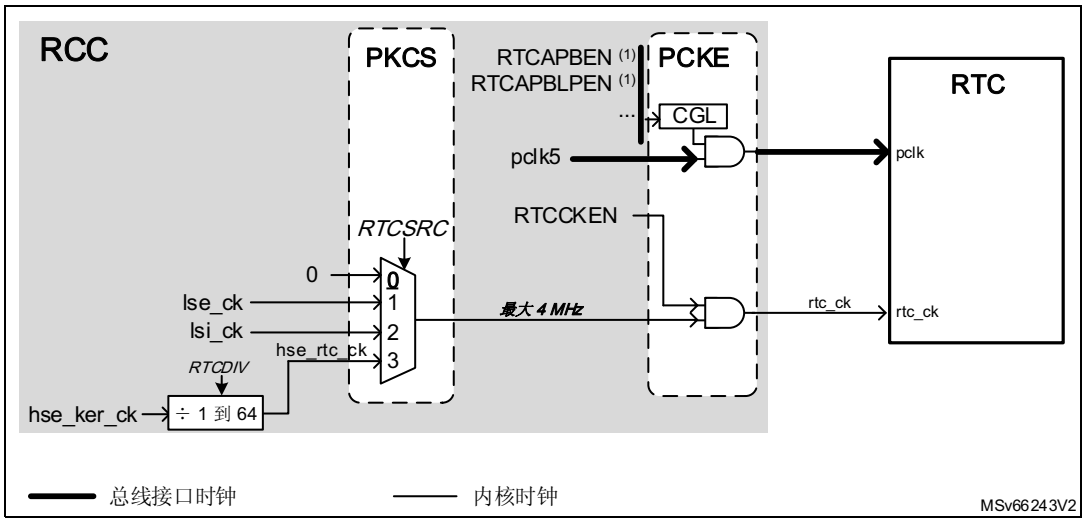
RTCSRC[1:0] 只能在备份域复位后或发生 LSE 故障后（LSECSSD = “1”）修改一次。

还请注意，要修改 RCC_BDCR 寄存器，PWR 控制寄存器 1 (PWR_CR1) 中的 DBP 位必须置“1”。

LSE 振荡器位于备份域中，而其他振荡器不在该域中。因此：

- 如果选择 LSE 作为 RTC 时钟：
  - 只要 VBAT 电源保持工作，即使 V_{DD} 电源关闭，RTC 仍可继续工作。
- 如果选择 LSI 作为 RTC 时钟：
  - 在 V_{DD} 电源掉电时，RTC 的状态将不能保证。
  - RTC 在 Stop、LP-Stop、LPLV-Stop 和待机模式下继续工作。
- 如果使用 HSE 时钟作为 RTC 时钟：
  - 如果 V_{DD} 电源关断或 V_{CORE} 电源关断，无法保证 RTC 状态。
  - 如果 HSEKERON = “1”，RTC 继续在 Stop 模式下工作。请注意，时钟会在 **nreset** 后丢失。

图 87. RTC 的时钟分配



1. MPU 有一个使能位和一个低功耗使能位。
   X 表示系统复位后所选择的复用器输入。

rtc_ck 时钟通过 RCC_BDCR 寄存器中的 RTCCKEN 位使能。
RTC 总线接口时钟（APB 时钟）通过 RCC_MP_APB5ENSETR、RCC_MC_APB5ENSETR、RCC_MP_APB5LPENSETR 和 RCC_MC_APB5LPENSETR 寄存器中的 RTCAPBEN 和 RTCAPBLPEN 位使能。

RTC 总线接口时钟（APB 时钟）通过 RCC_MP_APB5ENCLRR、RCC_MC_APB5ENCLRR、RCC_MP_APB5LPENCLRR 和 RCC_MC_APB5LPENCLRR 寄存器中的 RTCAPBEN 和 RTCAPBLPEN 位来禁止。

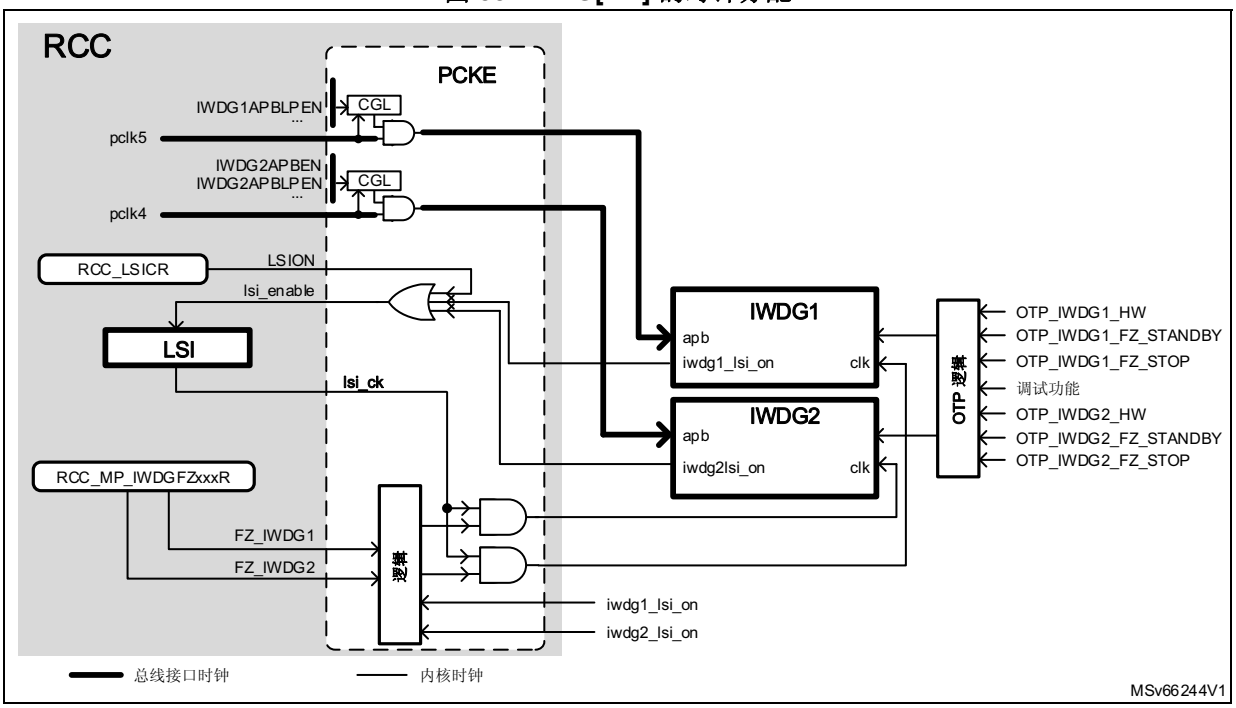
注： *在 APB 时钟频率低于 RTC 时钟频率的七倍 ($F_{APB} < 7 \times F_{RTCLCK}$) 时读取 RTC 日历寄存器，软件必须读取日历时间寄存器和日期寄存器两次。如果对 RTC_TR 的第二次读取访问得到的结果与第一次相同，则表明数据正确无误。否则必须执行第三次读取访问。*

独立看门狗 IWDG[2:1] 的时钟分配

RCC 为连接到 LSI 的两个独立看门狗（IWDG1 和 IWDG2）提供时钟。

如果独立看门狗已通过选项字节配置或软件访问的方式启动，则 LSI 将强制开启且不可禁止。在 LSI 振荡器设置延迟之后，时钟将提供给 IWDG。

图 88. IWDG[2:1] 的时钟分配



IWDG[2:1] 时钟控制

独立看门狗不需要 RCC 模块使能它们的时钟。有两种方式为 IWDG1/2 提供 LSI 时钟：

- 由软件使能 IWDG1/2，并使用*第 45.4.4 节：窗口选项*中描述的序列。IWDG1/2 自动向 RCC 请求激活 LSI 时钟。
- 通过 OTP_IWDG[2:1]_HW 选项字节，允许 IWDG1 和 IWDG2 在系统复位后自动使能。有关详细信息，请参见*第 45.4.4 节：窗口选项*看门狗配置。

另请注意，在访问 IWDGx 寄存器之前，必须使能 IWDG1 和 IWDG2 总线接口时钟（APB 时钟）。不需要访问 IWDGx 时，可以禁止 APB 时钟。即使 APB 时钟被禁止，IWDGx 仍进行倒计时。

另请注意，IWDG1 总线接口时钟（APB 时钟）通过 RCC_MP_APB5ENSETR、RCC_MP_APB5LPENSETR、RCC_MP_APB5ENCLRR 和 RCC_MP_APB5LPENCLRR 寄存器中的 IWDG1APBEN 和 IWDG1APBLPEN 位来控制。

同样，IWDG2 总线接口时钟（APB 时钟）通过 MPU 的 RCC_MP_APB4ENSETR、RCC_MP_APB4LPENSETR、RCC_MP_APB4ENCLRR 和 RCC_MP_APB4LPENCLRR 寄存器中的 IWDG2APBEN 和 IWDG2APBLPEN 位来控制。

位于 IWDG[2:1] 输入的 LSI 时钟在以下情况下可能被冻结：

- 当 FZ_IWDG[2:1] 位置 1 时（请参见*在 Boot ROM 代码执行期间处理 IWDG*）。这些位专用于 Boot ROM，在 Boot ROM 代码执行期间用来冻结 IWDG[2:1]。
- 当设置了选项字节 OTP_IWDG[2:1]_FZ_STANDBY 时。在这种情况下，系统进入待机模式时，相应的 IWDG[2:1] 会被冻结。
- 当设置了选项字节 OTP_IWDG[2:1]_FZ_STOP 时。在这种情况下，MPU 进入 CStop 时会冻结 IWDG[2:1]。

请注意，IWDG[2:1] 在调试模式下也可能冻结。

在 Boot ROM 代码执行期间处理 IWDG

在因系统复位或退出 LPLV-Stop2 所导致的 MPU 复位后，总是会激活 Boot ROM 代码。（请参见：*图 47：RCC 复位图*和*图 52：启动、冷/热复位以及从任何 Stop 模式唤醒的序列*。

只要 Boot ROM 代码被激活，Boot ROM 就会冻结 IWDG 时钟。
这是通过将 *RCC IWDG 时钟冻结设置寄存器 (RCC_MP_IWDGFZSETR)* 的 FZ_IWDG1 和 FZ_IWDG2 位置“1”来实现的。
在将控制权交给应用之前，Boot ROM 向 *RCC IWDG 时钟冻结清除寄存器 (RCC_MP_IWDGFZCLRR)* 的 FZ_IWDG1 和 FZ_IWDG2 位写入“1”，以解冻 IWDG。这样，如果 IWDG 是由应用使能的，则它们会被重新使能，否则保持禁止状态。

每次 MPU 复位时，硬件只允许写入 RCC_MP_IWDGFZSETR 一次。

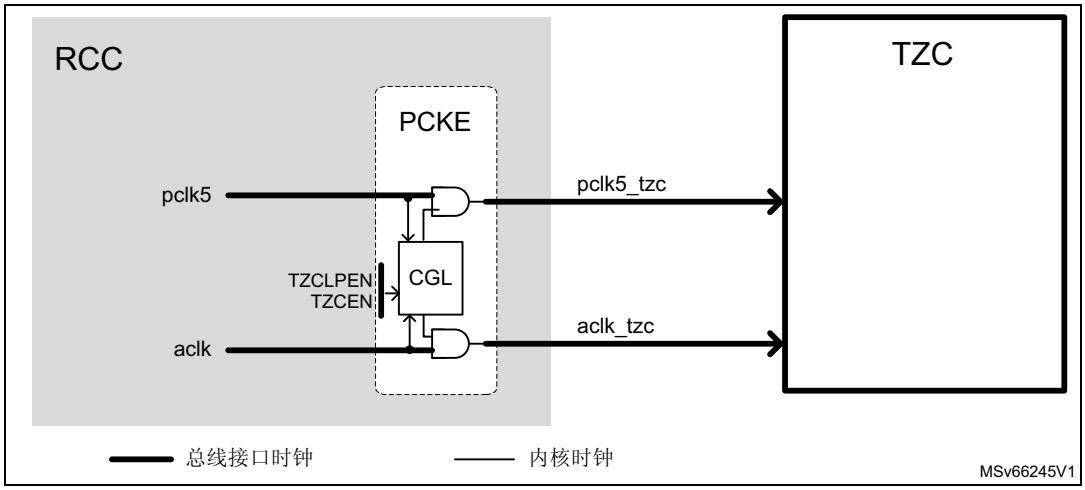
TZC 的时钟分配

下图为 TZC 的时钟分配的简图。TZC 接收以下时钟：

- 一个 APB 时钟 (**pclk5_tzc**)，
- 一个用于 AXI 端口 1 的 AXI 时钟 (**aclk_tzc1**)，
- 一个用于 AXI 端口 2 的 AXI 时钟 (**aclk_tzc2**)。

应用可以选择使能 APB 时钟和 AXI 端口的一个或两个时钟。

图 89. TZC 的时钟分配



1. MPU 有一个使能位和一个低功耗使能位。

为了使能 TZC，处理器必须将 TZCEN 置 1。如果应用想要使时钟在 CSleep 模式下保持工作状态，则相应的 TZCLPEN 位也必须置“1”。

每个使能位控制均可在 RCC_MP_APB5ENSETR/CLRR 和 RCC_MP_APB5LPENSETR/CLRR 寄存器上获得。

DDRC 和 DDRPHYC 的时钟分配

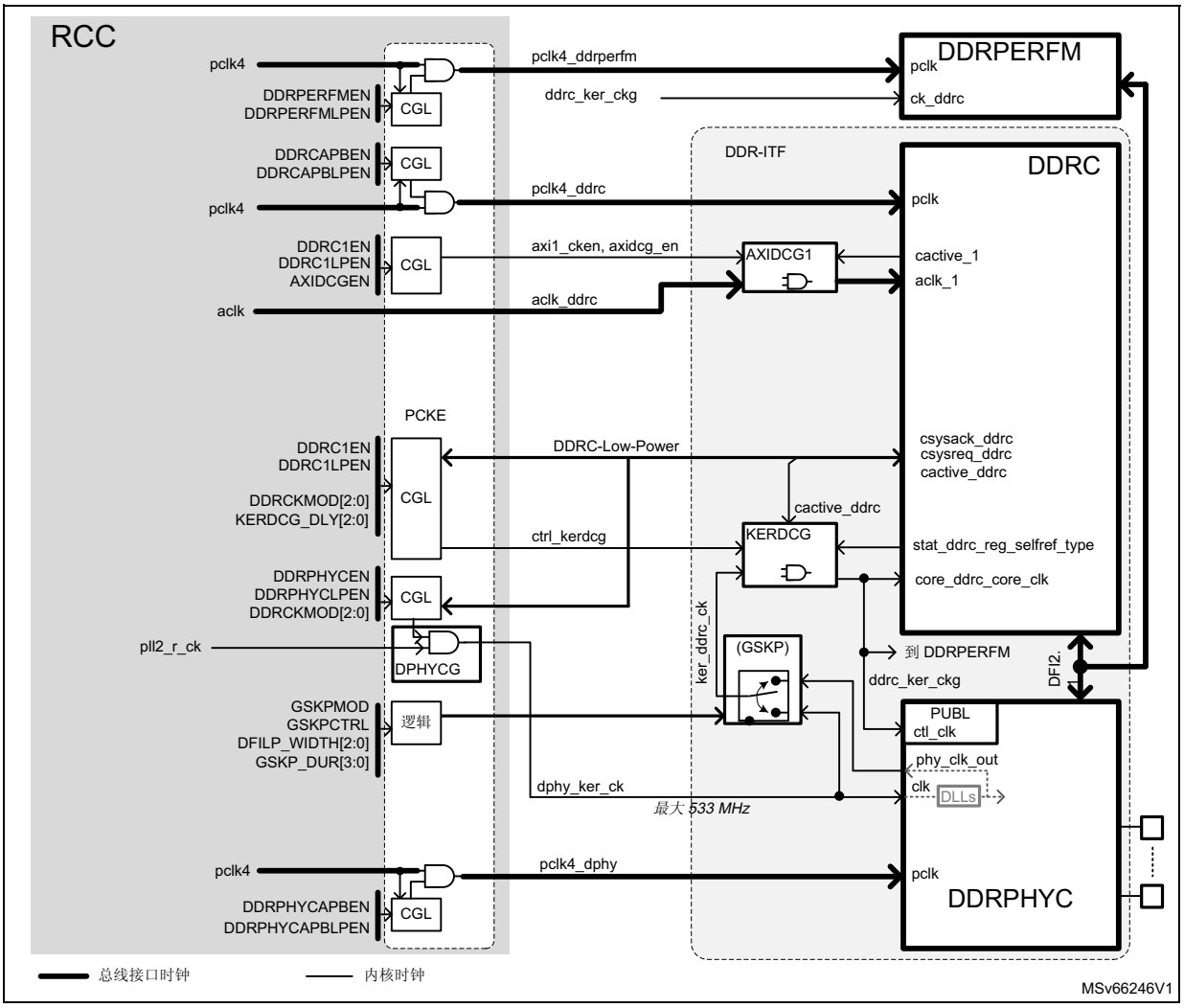
下图为 DDRC 和 DDRPHYC 模块的时钟分配简图。其他相关信息，请参见*第 5 章：DDR3/LPDDR2/LPDDR3 控制器 (DDRCTRL)*。

RCC 需要提供以下时钟：

- 寄存器接口的 APB 时钟（**pclk4_ddrperfm**、**pclk4_ddrc** 和 **pclk4_dphy**）。
- DDRC 单个 AXI 端口的 AXI 时钟 (**aclk**)。
- DDRPHYC 的内核时钟 (**dphy_ker_ck**)。请注意，DDRPHYC 提供时钟 **ddrc_ker_ck**，用于 DDRC、DDRPERFM 和 PUBL（DDRPHYC 的一部分）。

请注意，本节提及的所有 RCC 控制位均在 *RCC DDR 接口控制寄存器 (RCC_DDRITFCR)* 中。

图 90. DDRC、DDRPHYC 和 DDRPERFM 的时钟分配



APB 时钟的控制

APB 时钟门控的控制是通过专用位执行的：

- DDRCAPBEN 和 DDRCAPBLPEN 位控制 DDRC 接口的 APB 时钟门控 (**pclk4_ddrc**)
- DDRPHYCAPBEN 和 DDRPHYCAPBLPEN 位控制 DDRPHYC 接口的 APB 时钟门控 (**pclk4_dphy**)

有关详细信息，请参见*第 10.4.11 节：外设时钟门控控制*。

AXI 时钟的控制

AXIDCG1 模块处理提供给 DDRC AXI 端口的 AXI 时钟 (**aclk_1**) 的门控。AXIDCG1 位由 RCC 控制，可以按如下方式工作：

- 自动模式：**aclk_1** 时钟根据 DDRC 提供的 **cactive_1** 信号进行门控。
- 手动模式：RCC 可直接使能或禁止 AXI 时钟生成。

应用可以通过将相应的 DDRC1EN 设为“0”而禁止 AXI 端口的时钟（不使用时）。

表 59. aclk_1 时钟的门控控制

DDRC1EN	DDRC1LPEN	AXIDCGEN	MPU 模式	-	aclk_1 时钟	备注
0	X	X	任意	→	关闭	*由于 DDRC1EN = “0”，**aclk_1** 被门控。*
1	X	0	CRun	→	开启	*由于 DDRC1EN = “1”、AXIDCGEN = “0”并且 MPU 处于 CRun 状态，**aclk_1** 被强制开启，而与 **cactive_1** 无关。*
		1		→	**DCG** (1)	*当 DDRC1EN = AXIDCGEN = “1”时，**aclk_1** 根据 **cactive_1** 被自动门控。*
1	0	0	CSleep	→	关闭	*由于 DDRC1LPEN = “0”、AXIDCGEN = “0”并且 MPU 处于 CSleep 状态，**aclk_1** 被门控。*
	1			→	开启	*由于 DDRC1EN = DDRC1LPEN = “1”并且 AXIDCGEN = “0”，**aclk_1** 被强制开启，而与 **cactive_1** 无关。*
	X	1		→	**DCG**	*当 DDRC1EN = AXIDCGEN = “1”时，**aclk_1** 根据 **cactive_1** 被自动门控。*
X	X	X	CStop 或 CStandby	→	关闭	***aclk_1** 被门控。*

1. **DCG** 表示根据 **cactive_1** 信号进行动态时钟门控（灰色突出显示）。

DDRC 和 DDRPHYC 内核时钟的控制

KERDCG 和 DPHYCG 模块分别处理 **ddrc_ker_ckg** 和 **dphy_ker_ck** 的时钟门控。它们由 RCC 控制，并可根据 DDRC 和 RCC 的编程方式，在正常模式、自动模式或硬件低功耗自刷新模式下工作。

表 60 详细列出了根据 DDRCKMOD 的值处理时钟 **ddrc_ker_ckg** 和 **dphy_ker_ck** 的方式。

噪声跳过器 (GSKP)

RCC 还需要处理一个名为噪声跳过器 (GSKP) 的模块。GSKP 执行以下功能：

- GSKP 可阻止跳变时间不足 1 µs 的低功耗模式跳变。应用必须通过 DFILP_WIDTH[2:0] 位字段来编程超时延迟。
- GSKP 可防止在 DLL 重新激活期间时钟 **ddrc_ker_ck** 出现噪声。为此，当 GSKPMOD = “1” 时，GSKP 根据 GSKP_DUR[3:0] 位字段指定的时段向 DDRC 提供时钟 **dphy_ker_ck**。

  当 GSKPMOD = “0” 时，应用也可以通过 GSKPCTRL 位手动控制时钟切换。在这种情况下，当 GSKPCTRL 置 “1” 时，GSKP 向 DDRC 提供时钟 **dphy_ker_ck**，当 GSKPCTRL 设为 “0” 时，GSKP 在经过 GSKP_DUR[3:0] 指定的延迟后向 DDRC 提供时钟 **phy_clk_out**。

  DFILP_WIDTH、GSKPMOD、GSKP_DUR 和 GSKPCTRL 位字段在 *RCC DDR 接口控制寄存器 (RCC_DDRITFCR)* 中。

表 60. DDRCKMOD 控制描述

DDRCKMOD	-	KERDCG 模式	DPHYCG 模式	备注
“000”	→	正常	正常	**软件自刷新模式 (SSR)：**(1) **dphy_ker_ck** 时钟的门控取决于 DDRPHYCEN、DDRPHYCLPEN 和 MPU 模式。 **ddrc_ker_ckg** 时钟的门控取决于 DDRC1EN、DDRC1LPEN 和 MPU 模式。
“001”	→	自动	正常	**自动自刷新模式 (ASR1)：**(2) (3) 将根据 **cactive_ddrc** 信号自动门控时钟 **ddrc_ker_ckg**。 **dphy_ker_ck** 时钟的门控取决于 DDRPHYCEN、DDRPHYCLPEN 和 MPU 模式。 DDRC1EN 仍用于禁止/使能 DDR 接口。
“101”	→		自动	**全自动自刷新模式 (ASR2)：**(4) (3) 将根据 **cactive_ddrc** 信号自动门控时钟 **ddrc_ker_ckg** 和 **dphy_ker_ck**。 在该模式下，DDRPHYCEN 位必须置 “1”，DDRPHYCLPEN 必须设为 “0”。 DDRC1EN 仍用于禁止/使能 DDR 接口。
“010”	→	DDRC-LP	正常	**硬件自刷新模式 (HSR1)：**(2) (5) **ddrc_ker_ckg** 时钟的门控由连接到 DDRC 的 DDRC 低功耗接口控制。 **dphy_ker_ck** 时钟的门控取决于 DDRPHYCEN、DDRPHYCLPEN 和 MPU 模式。 DDRC1EN 仍用于禁止/使能 DDR 接口。
“110”	→		DDRC-LP	**全硬件自刷新模式 (HSR2)：**(4) (5) **ddrc_ker_ckg** 和 **dphy_ker_ck** 时钟的门控由连接到 DDRC 的 DDRC 低功耗接口控制。 在该模式下，DDRPHYCEN 位必须置 “1”，DDRPHYCLPEN 必须设为 “0”。 DDRC1EN 仍用于禁止/使能 DDR 接口。

1. 在 DDRC 和 DDRPHYC 初始化阶段，如果应用使用软件自刷新，则必须选择该模式。
2. 当应用想要保持 DLL 的时钟时，可以使用该模式。
3. 建议设置为 DDRC1EN = DDRC1LPEN = DDRPHYCEN = DDRPHYCLPEN = “1”。
4. 当 DLL 处于旁路状态时，可以使用该模式。
5. 建议设置为 DDRC1EN = DDRPHYCEN = “1”，DDRC1LPEN = DDRPHYCLPEN = “0”。

DDRC 低功耗接口

DDRC 模块提供了一个低功耗接口，该接口可用来对 DDRC 和 DDRPHYC 进入和退出低功耗模式加以控制。该接口还控制 **dphy_ker_ck** 和 **ddrc_ker_ckg** 时钟的门控。该低功耗接口符合 AMBA AXI 协议规范。为了让图例简洁，本文仅显示 DDRC 低功耗接口的最重要状态。*表 60* 详细列出了 DDRC 的低功耗状态。

DDRC 低功耗接口可以在 HSR1 和 HSR2 模式下使用，以便控制：

- DDRC 和 DDRPHYC 内核时钟的生成
- 外部 DDR 设备进入/退出低功耗（即自刷新）模式。

表 61. DDRC 低功耗 (DDRC-LP) 状态描述

DDRC 低功耗状态	-	备注
CKON	→	正常时钟操作：RCC 向 DDRC 和 DDRPHYC 提供内核时钟。**csysreq_ddrc**、**csysack_ddrc** 和 **cactive_ddrc** 均置“1”
CKOFF	→	低功耗运行：RCC 不向 DDRC 和 DDRPHYC 提供内核时钟。**csysreq_ddrc**、**csysack_ddrc** 和 **cactive_ddrc** 均设为“0”
CKON_Rreq	→	由 RCC 启动的低功耗退出：RCC 向 DDRC 和 DDRPHYC 提供内核时钟。RCC 将 **csysreq_ddrc** 置“1”，并等待 **csysack_ddrc** 和 **cactive_ddrc** 置“1”，之后再进入 CKON 状态。
CKON_Preq	→	由 DDRC 启动的低功耗退出：DDRC 通过将 **cactive_ddrc** 置“1”来请求内核时钟。RCC 提供内核时钟，将 **csysreq_ddrc** 置“1”，并等待 **csysack_ddrc** 置“1”，然后再进入 CKON 状态。
CKOFF_Rreq	→	由 RCC 启动的低功耗进入：RCC 通过将 **csysreq_ddrc** 设为“0”向 DDRC 请求进入低功耗状态。 DDRC 可以接受或拒绝该请求。如果请求遭到拒绝，接口将返回到 CKON 状态，RCC 重新请求进入低功耗状态。 当 DDRC 接受低功耗请求（**csysack_ddrc** 和 **cactive_ddrc** =“0”）时，RCC 会禁止内核时钟。然后切换到 CKOFF 状态。

自刷新模式控制

将外部 DDR 设备切换到自刷新对于节能至关重要。

当外部 DDR 设备处于自刷新模式时，应用可以关闭 DDRC 和 DDRPHYC 的时钟，而不会丢失外部 DDR 的内容。

当外部 DDR 设备处于自刷新模式时，应用将无法再访问外部 DDR 设备的内容。

DDR 设备退出自刷新模式可能需要几微秒的时间，应用必须考虑到这一延迟。该延迟是由外部 DDR 设备引起的，但与 DDRPHYC 中的 DLL 的处理方式也有关。

RCC 提供了 3 种控制 DDRC 和 DDRPHYC 时钟的方法：

- 软件自刷新模式 (SSR)
- 自动自刷新模式 (ASR)
- 硬件自刷新模式 (HSR)

RCC 的行为因应用所选择的自刷新模式而异。为此，应用必须根据 DDRC 和 DDRPHYC 的配置来编程 DDRCKMOD。

软件自刷新模式 (SSR)

在该模式下，应用必须直接编程 DDRC 以便立即将外部 DDR 设备切换到自刷新模式。通常情况下，当 MPU 想要进入 CSleep 或 CStop 状态时，必须确保不再需要 DDR 资源，并编程 DDRC 以将其切换到自刷新模式。
执行 DDRC 进入和退出自刷新的代码段必须位于 SYSRAM 中。这是因为当 DDR 设备处于自刷新模式时，需要运行一些软件程序。

自动自刷新模式（ASR1 和 ASR2）

在该模式下，软件不需要在每次必须将 DDRC 设置为自刷新模式时都对 DDRC 进行编程。当 DDRC 处于非工作状态的时间达到 DDRC 中定义的时间后，DDRC 会自动进入自刷新模式。

即使在 MPU 处于 CRun 模式时，如果其 AXI 端口上未发生活动，DDRC 也可能进入自刷新模式。自动自刷新模式允许进入自刷新模式，与处理器模式无关。

当 MPU 进入 CSleep 模式时，如果需要对 DDRC 的时钟和（可选的）DDRPHYC 的时钟进行门控，RCC 会延迟这些时钟的门控，直到 DDRC 切换到自刷新模式。

如果系统需要进入一种 Stop 模式，RCC 会延迟时钟的门控，直到 DDRC 切换到自刷新模式。

ASRx 模式只能在 DDRC 已编程为自动自刷新模式时使用，否则无法保证 DDR 接口的行为。

ASRx 模式仅影响 DDRC 和 DDRPHYC 内核时钟（即 **ddrc_ker_ckg** 和 **dphy_ker_ck**）的门控。

警告： **ASR2 模式只能在 DDRPHYC 的 DLL 被旁路时使用。因此，只有像 LPDDR2 这样工作频率约为 125 Mhz 的 DDR 设备才能使用该功能。**

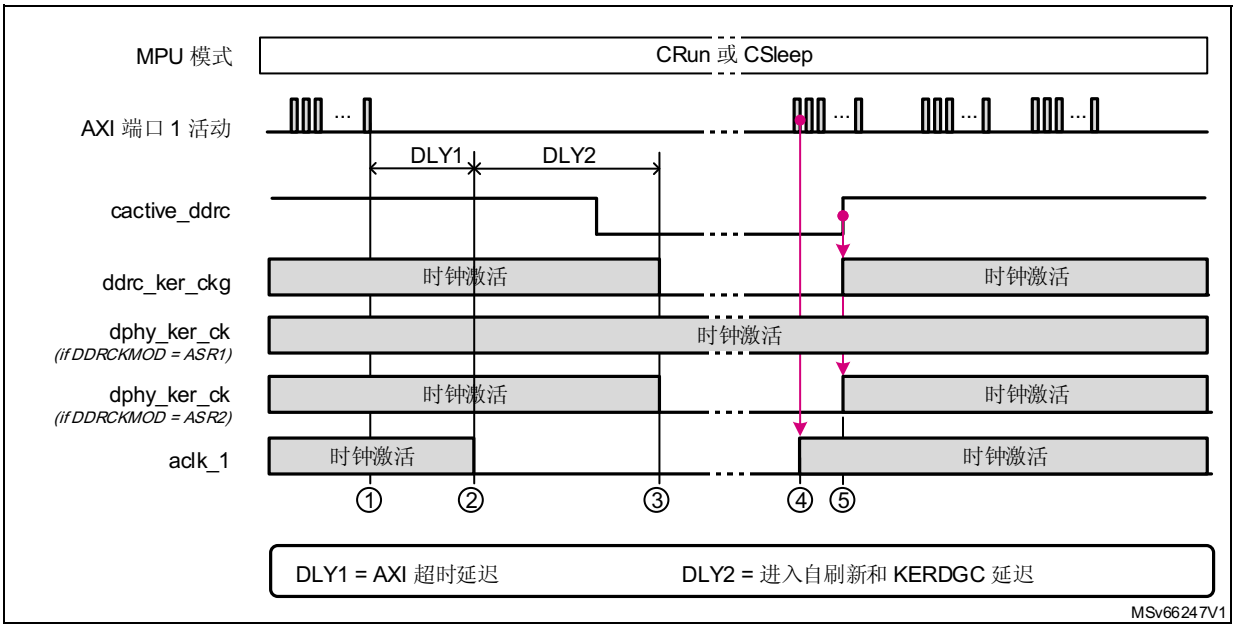
当使用 ASRx 模式时，至少必须将一个 DDRCxEN 位置“1”。

图 91 显示了进入和退出自刷新的简化时序。

在该示例中，DDRC1EN = DDRC1LPEN =“1”，DDRPHYCEN = DDRPHYCLPEN =“1”，AXIDCGEN =“1”。

1. AXI 端口 x 上的活动停止，其原因包括 MPU 进入 CSleep 模式等，但在两次活动突发之间也可能停止。
2. 当至少 UMCTL2_AXI_LOWPWR_NOPXCNT ***aclk_1*** 个周期 (DLY1) 未检测到 AXI 端口的活动时，如果 AXIDCGEN =“1”，则 AXI 时钟 (**aclk_1**) 将被门控。
3. 当 AXI 接口被门控后，如果在 DLY2 的延迟后没有切换回活动状态，则时钟 **ddrc_ker_ckg** 被门控。如果选择了 ASR1 模式，并且 DDRPHYCEN = DDRPHYCLPEN =“1”，则时钟 **dphy_ker_ck** 保持使能。如果选择了 ASR2 模式，则在 **ddrc_ker_ckg** 时钟被门控后不久，时钟 **dphy_ker_ck** 也会被门控。MPU 模式可以是 CRun 或 CSleep。
4. 如果在 AXI 端口执行了一个事务，AXI 时钟 (**aclk_1**) 会被重新使能。AXI 事务可以由使用 DDR 资源的任何主设备引发。
5. 经过几个 AXI 时钟周期后，DDRC 将 **cactive_ddrc** 置“1”，以请求内核时钟。DDRPHYC 时钟 **dphy_ker_ck** 和 DDRC 时钟 **ddrc_ker_ckg** 均重新使能。外部 DDR 随后就绪，可处理挂起的事务。

图 91. 在 ASR1 或 ASR2 模式下进入和退出自刷新



硬件自刷新模式（HSR1 和 HSR2）

硬件自刷新模式使用 DDRC 低功耗接口 (DDRC-LP) 来请求 DDR 进入或退出低功耗状态。

在以下情况下，DDRC-LP 允许 DDR 接口退出自刷新模式：

- 当通过 DDRC1EN 位使能接口时，
- 当 MPU 退出 CSleep 或 CStop 时，
- 当 DDRC 请求退出自刷新时。如果另有一个主设备正在访问 AXI 端口，就可能发生这种情况。

在以下情况下，DDRC-LP 允许 DDR 接口进入自刷新模式：

- 当通过 DDRC1EN 位禁止接口时
- 当 MPU 从 CRun 跳变到 CSleep 时
- 当 MPU 从 CRun 跳变到 CStop 时

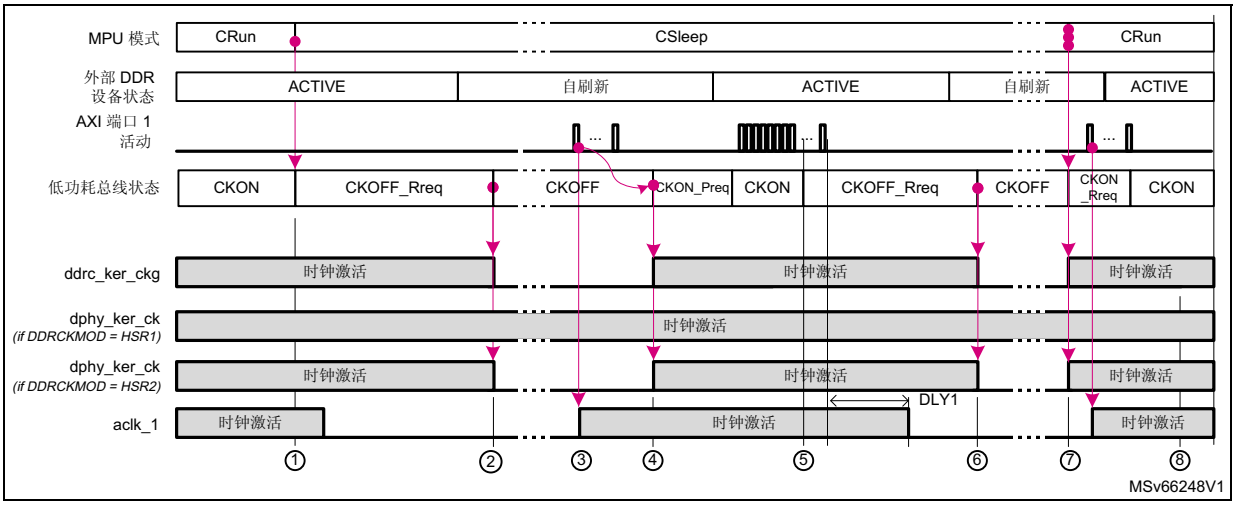
其节能效率虽然不如自动模式 (ASR)，但可以避免在 MPU 处于 CRun 模式时 DDRC 意外进入自刷新模式，从而提高了系统性能。

警告： **HSR2 模式只能在 DDRPHYC 的 DLL 被旁路时使用。因此，只有像 LPDDR2 这样工作频率约为 125 Mhz 的 DDR 设备才能使用该功能。**

图 92 显示了进入和退出自刷新模式的简化时序。当 MPU 在另有一个主设备仍处于使能状态的情况下进入 CSleep 时，通常会发生这种情况。

该示例基于*表 61* 中的配置 7 和配置 8，且 AXIDCGEN = “1”。

图 92. 在 HSR1 或 HSR2 模式下进入和退出自刷新

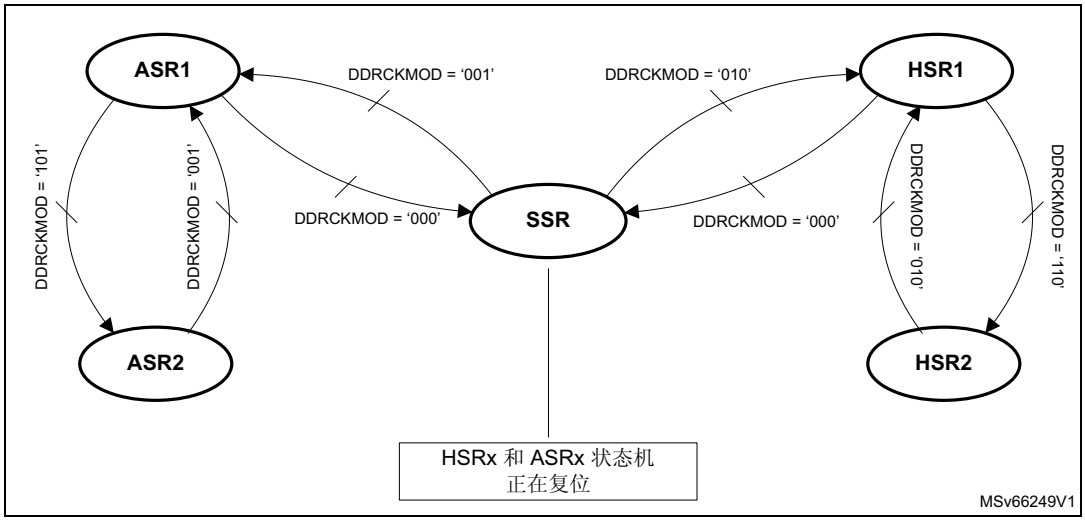


1. 当 MPU 进入 CSleep 时，RCC 使用 DDRC 低功耗总线 (CKOFF_Rreq) 向 DDRC 请求进入自刷新模式。另请注意，如果在编程设定的延迟时间 (DLY1) 内 AXI 端口上没有事务发生，则 AXI 时钟将被禁止。
2. 当满足 DDRC 中的所有条件（没有挂起的事务，DDR 处于自刷新模式）时，DDRC 将应答该请求。请注意，如果 DDRC 拒绝了 RCC 的请求，RCC 必须生成新的请求，直到 DDRC 接受，或者 MPU 退出 CSleep。
   当 DDRC 应答低功耗请求后，RCC 会禁止 DDRC 时钟 **ddrc_ker_ckg**。如果选择了 HSR1 模式，则若 DDRPHYCEN = DDRPHYCLPEN = “1”，**dphy_ker_ck** 时钟仍将保持使能状态。如果选择了 HSR2 模式，DDRPHYC 时钟 **dphy_ker_ck** 也将被禁止。DDRC 低功耗总线随后处于 CKOFF 状态。
3. 如果在两个 AXI 端口中的一个上执行了事务，则 AXI 时钟 (**aclk_1**) 将被重新使能。AXI 事务可以由使用 DDR 资源的任何主设备引发。
4. 经过几个 AXI 时钟周期后，DDRC 会使用 DDRC 低功耗总线 (CKON_Preq) 请求 RCC 重新使能内核时钟。只要 RCC 接收到请求，DDRPHYC 时钟 **dphy_ker_ck** 和 DDRC 时钟 **ddrc_ker_ckg** 就会被重新使能。此时外部 DDR 准备就绪，可处理挂起的事务。
5. 由于 MPU 仍处于 CSleep 模式，RCC 使用 DDRC 低功耗总线 (CKOFF_Rreq) 请求 DDRC 进入自刷新模式。如果不满足进入自刷新模式的所有条件，DDRC 可能会多次拒绝该请求。RCC 必须定期更新请求，直到被 DDRC 接受为止。如果其间 MPU 进入 CRun 模式，RCC 必须停止执行该请求。
6. 当满足 DDRC 中的所有条件（没有挂起的事务，DDR 处于自刷新模式）时，DDRC 将应答该请求。当 DDRC 应答低功耗请求后，RCC 会禁止 DDRC 时钟 **ddrc_ker_ckg**。如果选择了 HSR1 模式，则若 DDRPHYCEN = DDRPHYCLPEN = “1”，**dphy_ker_ck** 时钟仍将保持使能状态。如果选择了 HSR2 模式，DDRPHYC 时钟 **dphy_ker_ck** 也将被禁止。DDRC 低功耗总线随后处于 CKOFF 状态。
   如果 MPU 处于 CSleep 模式，则第 4 步到第 7 步的操作序列可能会重复多次。
7. 当 MPU 退出 CSleep 时，RCC 重新使能 DDRPHYC 时钟 **dphy_ker_ck** 和 DDRC 时钟 **ddrc_ker_ckg**，并使用 DDRC 低功耗总线 (CKON_Rreq) 请求 DDRC 让 DDR 退出自刷新模式。
8. DDRC 通过将 DDRC 低功耗总线设置为 CKON 状态来应答该请求。此时外部 DDR 准备就绪，可处理挂起的事务。

SSR、ASR 与 HSR 之间的跳变

应用可以在不禁止 DDR 接口的情况下更改 DDRCKMOD 的值。下图显示了允许在 SSR、ASR 与 HSR 模式之间进行的跳变。

图 93. 允许在 SSR、ASR 与 HSR 模式之间进行的跳变



不允许直接跳变：

- 从 ASRx 到 HSRx，应用需要先将 DDRCKMOD 设为“000”(SSR)，然后将 DDRCKMOD 设置为所需的 HSR 或 ASR 模式
- 从 SSR 到 ASR2，应用需要先将 DDRCKMOD 设为“001”(ASR1)，然后将 DDRCKMOD 设为“101”（ASR2 模式），
- 从 ASR2 到 SSR，应用需要先将 DDRCKMOD 设为“001”(ASR1)，然后再将 DDRCKMOD 设为“000”（SSR 模式），
- 从 SSR 到 HSR2，应用需要先将 DDRCKMOD 设为“010”(HSR1)，然后再将 DDRCKMOD 设为“110”（HSR2 模式），
- 从 HSR2 到 SSR，应用需要先将 DDRCKMOD 设为“010”(HSR1)，然后再将 DDRCKMOD 设“000”（SSR 模式）。

推荐配置

位 DDRC1EN 和 DDRPHYC 用于使能或禁止 DDR 接口。

表 62 显示了推荐的 RCC 配置。应用必须选择适合其 DDR 配置的配置。

表 62. 推荐配置(1)

配置编号	DDRCKMOD	DDRC1EN	DDRC1LPEN	DDRPHYCEN	DDRPHYCLPEN	-	MPU CRun 模式			MPU CSleep 模式			备注
							AXI 时钟(2)	DDRC 时钟	DDRPHYC 时钟	AXI 时钟(2)	DDRC 时钟	DDRPHYC 时钟	
1	任意	0	x	0	x	→	关闭	关闭	关闭	关闭	关闭	关闭	接口已禁止
2	SSR	1	0	1	0	→	开启	**开启**	**开启**	关闭	关闭	关闭	DDRC 和 DDRPHYC 时钟的门控取决于 MPU 模式。在门控时钟之前，由应用软件将 DDR 接口切换到低功耗模式。
3		1	0	1	1	→	开启	**开启**	**开启**	关闭	关闭	**开启**	
4		1	1	1	1	→	开启	**开启**	**开启**	开启	**开启**	**开启**	
5	ASR1	1	1	1	1	→	开启	DCG	**开启**	开启	DCG	**开启**	只有 DDRC 内核时钟是根据自动模式进行控制的。
6	ASR2	1	1	1	0	→	开启	DCG	DCG	开启	DCG	DCG	DDRC 和 DDRPHYC 的内核时钟均根据自动模式进行控制。 仅当 DDRPHYC 的 DLL 被旁路时，才能使用该模式。因此，只有像 LPDDR2 这样工作频率约为 125 Mhz 的 DDR 设备才能使用该功能。
7	HSR1	1	0	1	1	→	开启	**DLP**	**开启**	关闭	**DLP**	**开启**	DDRC 内核时钟根据 DDRC-LP 接口进行控制。 必须通过设置 DDRPHYCEN = DDRPHYCLPEN =“1”来使 DDRPHYC 内核时钟保持工作状态
8	HSR2	1	0	1	0	→	开启	**DLP**	**DLP**	关闭	**DLP**	**DLP**	DDRC 和 DDRPHYC 的内核时钟均根据 DDRC-LP 接口进行控制。 仅当 DDRPHYC 的 DLL 被旁路时，才能使用该模式。因此，只有像 LPDDR2 这样工作频率约为 125 Mhz 的 DDR 设备才能使用该功能。

1. DCG 表示时钟是使用动态时钟门控进行控制的（灰色突出显示）。
   如果是 DDRC 和 DDRPHYC，则表示时钟由 **cactive_ddrc** 信号门控。
   DLP 表示 DDRC 低功耗协议（以蓝色突出显示）。
2. 当位 AXIDCGEN =“1”时，AXI 时钟将根据 **cactive_1** (DCG) 自动门控（DDRC1EN =“0”时除外）。

在配置 2 到 4 中，DDRC 和 DDRPHYC 时钟的门控取决于 MPU 模式。在门控时钟之前，由应用软件将 DDR 接口切换到低功耗模式。

配置 2

- DDRC 和 DDRPHYC 的 APB 时钟和内核时钟在 MPU 处于 CRun 模式时使能，处于 CSleep 模式时禁止。

配置 3

- DDRC 的 APB 时钟在 MPU 处于 CRun 模式时使能，处于 CSleep 模式时禁止。
- DDRPHYC 的 APB 时钟在 MPU 处于 CRun 或 CSleep 时使能。
- DDRC 内核时钟 (**ddrc_ker_ckg**) 在 MPU 处于 CRun 时使能，处于 CSleep 时禁止。
- DDRPHYC 内核时钟 (**dphy_ker_ck**) 在 MPU 处于 CRun 或 CSleep 时使能。

配置 4

- DDRC 和 DDRPHYC 的 APB 时钟在 MPU 处于 CRun 或 CSleep 时使能。
- DDRC 和 DDRPHYC 的内核时钟（**ddrc_ker_ckg** 和 **dphy_ker_ck**）在 MPU 处于 CRun 或 CSleep 时使能。

在配置 5 和 6 中，DDRC 和 DDRPHYC 时钟的门控是自动进行的，并且取决于 DDRC 的活动。

配置 5

- DDRC 和 DDRPHYC 的 APB 时钟在 MPU 处于 CRun 或 CSleep 时使能。
- DDRC 内核时钟 (**ddrc_ker_ckg**) 根据 **cactive_ddrc** 进行门控，与 MPU 模式无关。
- DDRPHYC 内核时钟 (**dphy_ker_ck**) 的使能与 MPU 模式无关。当 DDRPHYC 的 DLL 需要保持锁定时，可以考虑使用该配置。

配置 6

- DDRC 和 DDRPHYC 的 APB 时钟在 MPU 处于 CRun 或 CSleep 时使能。
- DDRC 和 DDRPHYC 的内核时钟（**ddrc_ker_ckg** 和 **dphy_ker_ck**）根据 **cactive_ddrc** 进行门控，与 MPU 模式无关。
- 如果未使用（旁路）DDRPHYC 的 DLL，则该配置可节能。

在配置 7 到 8 中，DDRC 和 DDRPHYC 时钟的门控由 DDRC-LP 控制。DDRC-LP 也可以请求 DDR 接口进入低功耗（即自刷新）模式。

配置 7

- DDRC 的 APB 时钟在 MPU 处于 CRun 模式时使能，处于 CSleep 模式时禁止。
- DDRPHYC 的 APB 时钟在 MPU 处于 CRun 或 CSleep 模式时使能。
- DDRC 内核时钟 (**ddrc_ker_ckg**) 是根据 MPU 模式和 DDRC 请求，使用 DDRC 低功耗握手信号进行门控的。
- DDRPHYC 内核时钟 (**dphy_ker_ck**) 的使能与 MPU 模式无关。当 DDRPHYC 的 DLL 需要保持锁定时，可以考虑使用该配置。

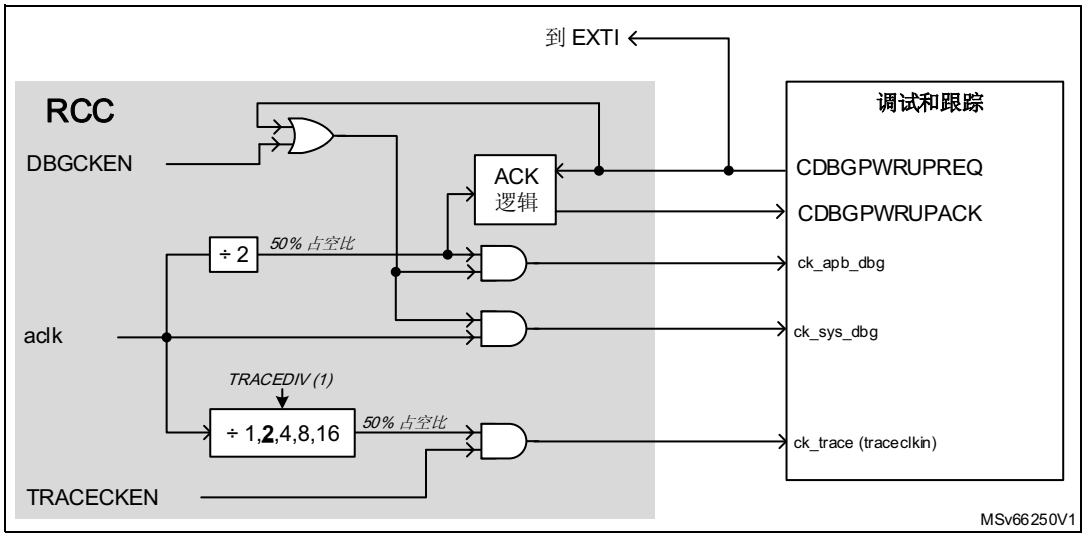
配置 8

- DDRC 的 APB 时钟在 MPU 处于 CRun 模式时使能，处于 CSleep 模式时禁止。
- DDRPHYC 的 APB 时钟在 MPU 处于 CRun 或 CSleep 模式时使能。
- DDRC 和 DDRPHYC 内核时钟（**ddrc_ker_ckg** 和 **dphy_ker_ck**）是根据 MPU 模式和 DDRC 请求，使用 DDRC 低功耗握手信号进行门控的。

DBG 和跟踪的时钟分配

跟踪和调试的时钟生成由 RCC 通过 *RCC 调试配置寄存器 (RCC_DBGCFGR)* 控制。

图 94. DBG 和跟踪的时钟说明



1. **X** 表示系统复位后的复位值。

IO 补偿的时钟分配

IO 补偿单元用于校准 I/O 的斜率。为此，RCC 向 IO 补偿单元块提供 **csi_ck** 时钟。

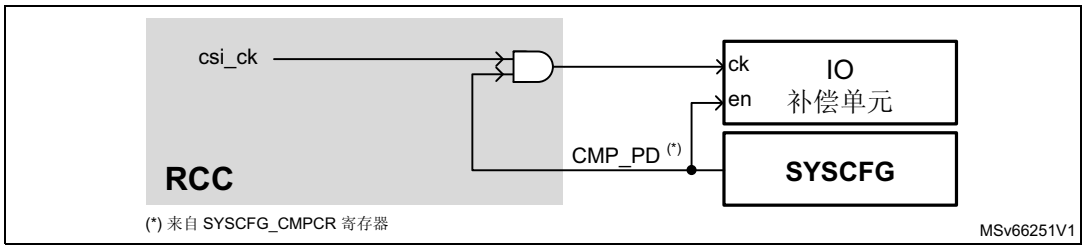
当需要激活补偿机制时，必须向 IO 补偿单元块提供时钟。这意味着应用必须将 CSION 位置“1”。

如果 CMP_PD 位设为“0”，则不会向 IO 补偿单元提供 **csi_ck** 时钟。更多信息，请参见*第 12 章：系统配置控制器 (SYSCFG)* 模块说明。

请注意，当系统进入 Stop 模式且 CSIKERON 位设为“0”时，**csi_ck** 时钟会自动被门控。如果将 CSIKERON 位置“1”，则即使处于系统 Stop 模式下，也可以使 **csi_ck** 时钟保持使能。

在系统待机模式下，**csi_ck** 时钟会被禁止。

图 95. IO 补偿单元的时钟说明



10.4.9 常规时钟概念概述

RCC 根据 MPU 的工作模式，处理系统 MPU、总线接口和外设时钟的分配。有关工作模式的详细信息，请参见*第 9.5.1 节：工作模式*。

有关时钟定义的详细信息，请参见*第 10.4.1 节：时钟命名约定*。

对于大多数外设，应用可以对它们的内核和总线接口时钟的使能和禁止加以控制。

在使用外设之前，MPU 必须根据需要将其使能（通过将 PERxEN 置“1”），并定义该外设在 CSleep 和 Cstop 模式下是否保持工作状态（通过将 PERxLPEN 置“1”）。该操作称作为处理器“分配”外设。

10.4.10 外设分配

MPU 可分配外设，从而可以控制外设的内核和总线接口时钟。当 MPU 将外设的 PERxEN 位置“1”时，即分配一个外设。

MPU 可以通过设置寄存器 RCC_MP_xxxxENSETR 和 RCC_MP_xxxxENCLRR 中的专用 PERxEN 位来分配或取消分配外设。

对于在 MPU 处于 CSleep 或 CStop 模式（PERxLPEN 位）时外设时钟的控制，实行类似的机制。

硬件使用外设分配位（PERxEN 位）向外设提供内核和总线接口时钟，但也可以使用这些位将外设与 MPU 链接。这样，硬件便能根据 MPU 状态安全地对外设时钟和总线矩阵时钟进行门控。PWR 模块还使用该信息来适当的控制域的状态。

时钟使能延迟

为避免产生尖峰，时钟门控逻辑会将使能命令（通常来自内核时钟请求或 PERxEN 位）与所选时钟同步。

- 在使能命令和时钟的第一个上升沿之间，可能会发生最长 2 个所使能时钟周期的延迟。使能命令可以是 RCC_xxxxENR 寄存器的 PERxEN 位的上升沿，或者是外设所发出的内核时钟请求。
- 在禁止命令和时钟的最后一个下降沿之间，可能会发生最长 1.5 个所禁止时钟周期的延迟。禁止命令可以是 RCC_xxxxENR 寄存器的 PERxEN 位的下降沿，或者是外设所释放的内核时钟请求。

注： *内核时钟和总线接口时钟均会受此再同步延迟的影响。*

此外，如果应用首次使能不在其子系统中的外设，则时钟使能延迟的时间会延长。这是因为外设所在的子系统可能还没有准备好其子系统时钟。必须先开启子系统时钟，之后应用才可以使用此外设。

为了正确分配外设，必须遵循以下操作顺序：

- 通过在 RCC_MP_xxxxENSETR 寄存器中的 PERxEN 位写入“1”来使能外设时钟（即，分配外设），
- 回读 RCC_MP_xxxxENSETR 寄存器。可将该操作视为一次空指令读取，
- 然后即可以使用外设。

注： *如果总线接口时钟未激活，则不支持对外设寄存器进行读访问或写访问。读访问会导致读取无效数据。写访问将被忽略，但不会产生任何总线错误。*

RCC 使用外设分配并根据 MPU 模式自动控制时钟门控。

当 MPU 进入 CStop 模式时，RCC 会自动禁止所有已分配外设的总线接口和内核时钟以及 MPU 时钟、AXI 和 MLAHB 互连时钟。

唤醒事件能够使系统退出待机或 Stop 模式。

注： *此外，有一些外设具有可编程启用控制位，用以在 MPU 处于 CStop 模式时使能（或不使能）各个外设，分别为：STGENRO、STGENC、ETH1 和 ETH2，这样它们便能生成唤醒事件并使系统退出 Stop 模式或待机模式，尽管它们的硬件中并没有请求内核时钟功能 (**req_ker_perx**)。*

内存设备处理

该产品中可用的嵌入式内存包括 SYSRAM、SRAM1、SRAM2、SRAM3 和 BKPSRAM。

当 MPU 处于 CRun 模式时：

- SYSRAM、SRAM1、SRAM2 和 SRAM3 会自动使能
- BKPSRAM 使能或禁止，具体取决于 RCC_MP_AHB5ENxxxR 寄存器的 BKPSRAMEN 位（xxx=SET/CLR）

当 MPU 处于 CSleep 模式时：

- SRAM1、SRAM2 和 SRAM3 会自动使能
- BKPSRAM 使能或禁止，具体取决于 RCC_MP_AHB5LPENxxxR 寄存器的 BKPSRAMLPEN 位（xxx=SET/CLR）
- SYSRAM 使能或禁止，具体取决于 RCC_MP_yy_AXIMLPENxxxR 寄存器的 SYSRAMLPEN 位（xxx=SET/CLR，yy=S/NS）。

有关时钟使能的详细信息，请参见*第 10.4.11 节：外设时钟门控控制*和*第 10.4.12 节：处理器和互连时钟门控*。

10.4.11 外设时钟门控控制

如前文所述，每个外设均需要一个名为 **ckg_bus_perx**（外设“x”）的总线接口时钟。该时钟可以是 APB、AHB 或 AXI 时钟，具体取决于与外设相连的总线。
用作 MPU 域中外设的总线接口的时钟可以是 **aclk**、**hclk5**、**hclk6**、**pclk4** 或 **pclk5**，具体取决于与每个外设相连的总线。为简单起见，将这些时钟称为 **mpu_bus_ck**。同样，名为 **mlahb_bus_ck** 的信号表示 **hclk[4:1]**、**pclk1**、**pclk2** 或 **pclk3**，具体取决于连接到 MLAHB 域各个外设的总线。

一些外设（如 SAI、UART）还需要一个专用的通信接口时钟，该时钟通常与总线接口时钟不同步，称为内核时钟 (**kckg_perx**)。

可根据下文详细介绍的各个条件对两个时钟进行门控。

如*图 96* 所示，每个外设的内核时钟和总线接口时钟的使能都取决于多个输入信号：

- MP_PERxEN、MP_PERxLPEN 位
- **mpu_mode**（参见*表 64：MPU 模式详细信息*）
- 外设自身的内核时钟请求 (**req_ker_perx**)，前提是功能可用。

MP_PERxEN 表示 MPU 的外设使能（分配）位。MPU 可以通过 RCC_MP_xxxxENSETR 寄存器将这些位写为“1”。

MPU 状态是处理器状态（WFI 或 WFE）和 STPREQ_P0 位的逻辑组合，详细信息请参见*表 69*。

更多详细信息，请参见*第 10.4.11 节：外设时钟门控控制*。

图 96. 外设内核时钟使能逻辑详细信息

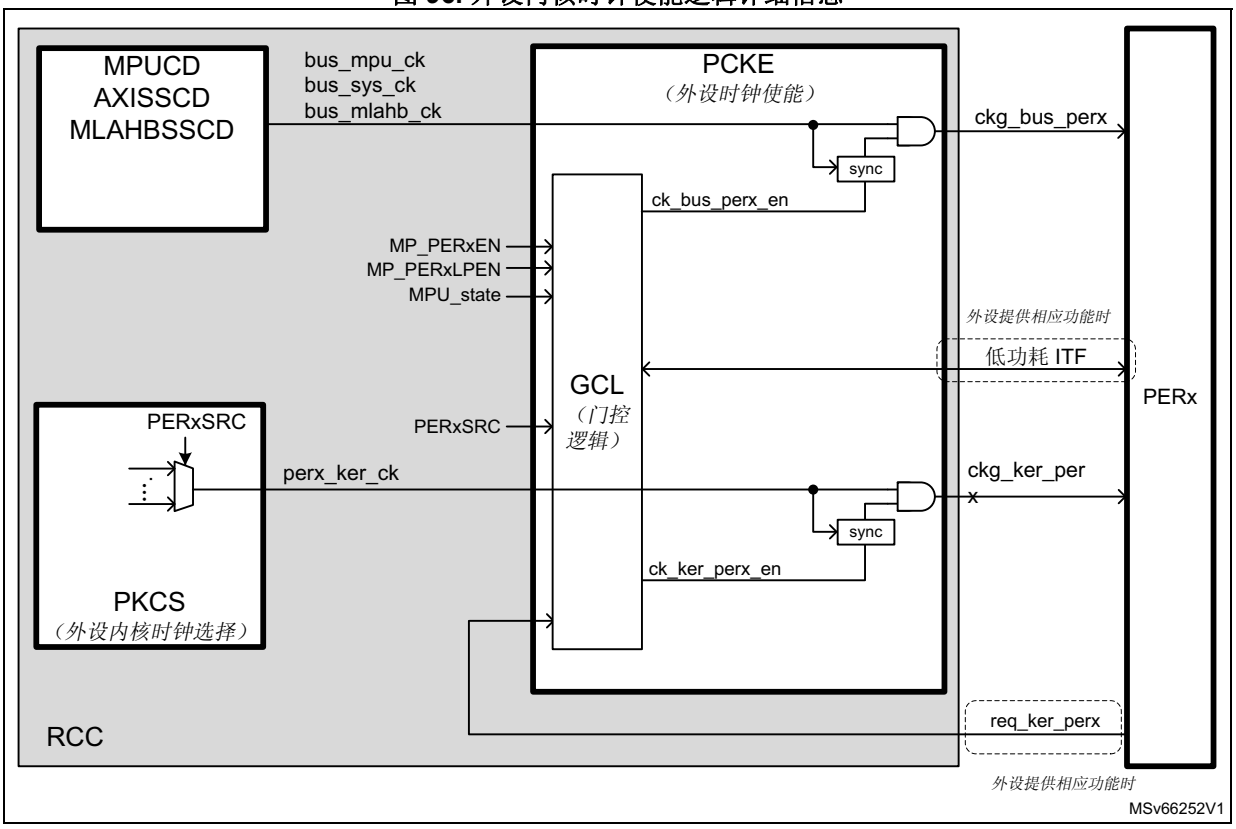


表 63 详细列出了由 MPU 分配的大多数外设时钟的使能逻辑。

表 63. MPU 的外设时钟使能详细信息

MP_PERxEN	MP_PERxLPEN	PERxSRC	req_ker_perx	MPU 状态	-	mp_perx_en_ker_ck	mp_perx_en_bus_ck	备注
0	X	X	X	X	→	0	0	没有向外设提供时钟，因为 PERxEN = 0
1	X	X	X	CRun	→	1	1	向外设提供内核和总线接口时钟，因为 CPU 处于 CRun 模式且 PERxEN = “1”
1	0	X	X	CSleep	→	0	0	没有向外设提供时钟，因为 MPU 处于 CSleep 模式且 PERxLPEN = 0
1	1	X	X		→	1	1	向外设提供内核和总线接口时钟，因为 MPU 处于 CSleep 模式且 PERxLPEN = “1”

表 63. MPU 的外设时钟使能详细信息（续）

MP_PERxEN	MP_PERxLPEN	PERxSRC	req_ker_perx	MPU 状态	-	mp_perx_en_ker_ck	mp_perx_en_bus_ck	备注
1	0	X	X	CStop	→	0	0	没有向外设提供时钟，因为 PERxLPEN 位被设为“0”。
1	1	非 lsi_ck 并且 非 lse_ck 并且非 hsi_ker_ck 非 csi_ker_ck 并且非 hse_ker_ck	X		→	0	0	没有向外设提供时钟，因为 MPU 处于 CStop 模式且未选择 **lse_ck**、**lsi_ck**、**hsi_ker_ck**、**csi_ker_ck** 或 **hse_ker_ck**。
1	1	lsi_ck 或 lse_ck	X		→	1 (1)	0	向外设提供内核时钟，因为 PERxEN = PERxLPEN =“1”且选择了 **lsi_ck** 或 **lse_ck**。 没有提供总线接口时钟，因为 MPU 处于 CStop 模式。 如果需要，由应用来使能 LSE 或 LSI。
1	1	hsi_ker_ck 或 csi_ker_ck 或 hse_ker_ck	1		→	1	0	向外设提供内核时钟，因为 **req_ker_perx** =“1”，PERxEN = PERxLPEN =“1”，并且选择了 **hsi_ker_ck**、**csi_ker_ck** 或 **hse_ker_ck**。 没有提供总线接口时钟，因为 MPU 处于 CStop 模式。 如果振荡器处于关闭状态，则当出现内核时钟请求时，RCC 会将振荡器使能。
1	1	hsi_ker_ck 或 csi_ker_ck 或 hse_ker_ck	0		→	0	0	没有向外设提供时钟，因为 MPU 处于 CStop 模式，且不存在挂起的内核时钟请求

1. 请注意，对于 RNG 模块，如果 MPU 为 CStop 状态，即使选择的时钟为 lsi_ck 或 lse_ck，也不会传送内核时钟。

注： *仅当 PERxSRC 选择了一个振荡器时，由外设生成的内核时钟请求才会使能相应的振荡器：HSI、CSI 或 HSE。*

总之，满足以下条件时会为外设提供内核时钟：

1. 当分配给外设的 CPU 处于 CRun 模式时。
2. 当分配给外设的 CPU 处于 CSleep 模式且 PERxLPEN =“1”时。
3. 当分配了外设的 CPU 处于 CStop 模式，并且 PERxLPEN =“1”，同时外设正在生成内核时钟请求，且所选时钟是 **hsi_ker_ck**、**csi_ker_ck** 或 **hse_ker_ck** 时。
4. 当分配了外设的 CPU 处于 CStop 模式，PERxLPEN =“1”，且外设的内核时钟源为 **lse_ck** 或 **lsi_ck** 时。

只有在满足条件 1 或 2 时才会向外设提供总线接口时钟。

注： *只有部分外设能够请求内核时钟：I2C 和 U(S)ART。借助该功能，外设能以最理想的功耗等待事件。*

注： *另一方面，STGEN 和 ETH[2:1] 等一些外设不能请求内核时钟功能。但 PERxSTPEN 位使外设能在 CStop 模式下部分运转，并提供唤醒事件（前提是 PERx_EN=PERxLPEN=PERxSTPEN=“1”）。（请参见：更多详细信息，请参见以太网（ETH1 和 ETH2）的时钟分配一节和 STGEN 和 ADC 的时钟分配一节*

10.4.12 处理器和互连时钟门控

CA7-SS (MPU) 时钟门控

处理器和 SCU 通过 RCC 提供的 mpuss_ck 时钟获得时钟。

当处理器进入等待中断 (WFI) 状态时，它会通过禁止其时钟而被设置为低功耗状态（用于将处理器从 WFI 唤醒的功能除外）。

当处理器进入等待事件 (WFE) 状态时，它会通过禁止其时钟而被设置为低功耗状态（用于将处理器从 WFE 唤醒的功能除外）。

当处理器处于 WFI 状态时，L2 内存系统完成挂起的事务，进入 WFI 状态，并且信号 **mpu_standbywfi_l2** 被置为有效。如果通过 STPREQ_P0 0 位请求了 CStop 模式，则 RCC 禁止 **mpuss_ck** 时钟和 AXI 接口时钟 (**aclk_mpu**)。

表 64 详细列出了基于处理器状态 (WFI) 和 RCC 寄存器值的 MPU 模式。

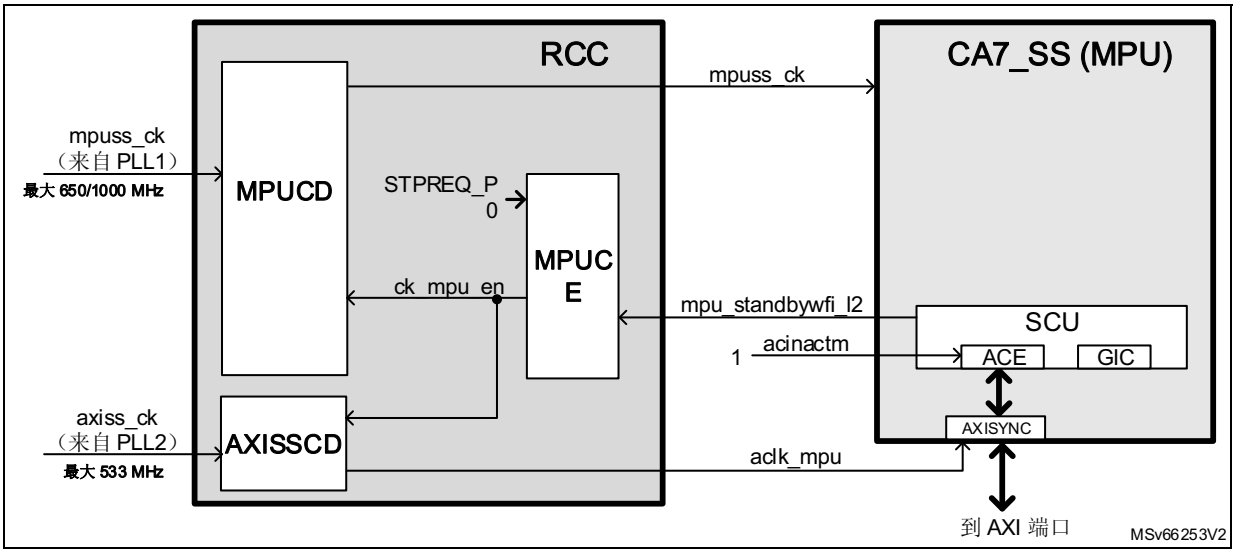
表 64. MPU 模式详细信息(1)

STPREQ_P0(2)	proc_0 处于 WFI 状态	-	MPU 模式
-	N	→	CRun
0	Y	→	CSleep
1	Y	→	CStop

1. “-”表示均可。
2. STPREQ_P0 位在 *RCC 停止请求设置寄存器 (RCC_MP_SREQSETR)* 和 *RCC 停止请求清除寄存器 (RCC_MP_SREQCLRR)* 中。

当 MPU 状态为 CStop 时，**mpuss_ck** 和 **aclk_mpu** 时钟由 RCC 进行门控。

图 97. MPU 低功耗接口



总线矩阵和总线桥时钟门控

总线桥和总线矩阵时钟是根据 MPU 模式和分配给处理器的外设进行门控的。

下表列出了 RCC 对总线矩阵和总线桥时钟进行门控的条件。

有关命名约定的信息，请参见*第 10.4.11 节：外设时钟门控控制*。

表 65. AXIM 的总线时钟使能详细信息

MPU 模式	mpu_peren_axi (1)	mpu_perlpen_axi (1)	-	AXIM 时钟	说明
CRun	X	X	→	工作	当以下条件之一为真时，将生成 AXIM 时钟： – MPU 处于 CRun 模式 – MPU 处于 CSleep 模式，并且至少有一个由 MPU 分配的 AXI-SS 外设激活了睡眠使能位（即 MP_ PERxEN = “1” 并且 MP_PERxLPEN = “1”）
CSleep	1	1	→		
	X	0	→	门控	在所有其他情况下，AXIM 时钟将被门控。
CStop/ CStandby	X	X	→		

1. 当至少有一个连接到 AXI-SS 的外设分配给 MPU 时，**mpu_peren_axi** 置“1”。
   当至少有一个连接到 AXI-SS 的外设分配给 MPU，并且其 PERxLPEN 位置“1”时，**mpu_perlpen_axi** 置“1”。

注： *当 mpu_peren_axi = “0” 时，mpu_perlpen_axi 也等于 “0”。*

当 mpu_peren_axi = mpu_perlpen_axi = “1” 时，表示至少对于一个连接到 AXI-SS 的外设，MP_PERxEN = MP_PERxLPEN = “1”。

表 66. AHB5 和 AHB6 的总线时钟使能详细信息

MPU 模式(1)	mpu_peren_ahb5 (mpu_peren_ahb6) (2)	mpu_perlpen_ahb5 (mpu_perlpen_ahb6) (2)	-	AHB5 (AHB6) 时钟	说明
CRun	1	X	→	工作	当以下条件之一为真时，将生成 AHB5 (AHB6) 时钟： – MPU 处于 CRun 模式，并且 AHB5 (AHB6) 上至少分配了一个外设（即，至少一个 MP_PERxEN = “1”） – MPU 处于 CSleep 模式，并且 AHB5 (AHB6) 上至少分配了一个外设，并且激活了睡眠使能位（即 MP_PERxEN = “1”并且 MP_PERxLPEN = “1”）
CSleep	1	1	→		
任意	0	X	→	门控	在所有其他情况下，AHB5 (AHB6) 时钟将被门控。
CSleep	X	0	→		
CStop/ CStandby	X	X	→		

1. “任何”指任何模式。MPU 可以是 CRun、CSleep、CStop 或 CStandby。
2. 当至少有一个连接到 AHB5 的外设被分配给 MPU 时，**mpu_peren_ahb5** 置“1”。
   当至少有一个连接到 AHB5 的外设分配给 MPU，并且其 PERxLPEN 位置“1”时，**mpu_perlpen_ahb5** 置“1”。
   当至少有一个连接到 AHB6 或 APB4 的外设分配给 MPU 时，**mpu_peren_ahb6** 置“1”。
   当至少有一个连接到 AHB6 或 APB4 的外设分配给 MPU，并且其 PERxLPEN 位置“1”时，**mpu_perlpen_ahb6** 置“1”。

注： *AXIMC 外设时钟和复位通过 RCC_MP_AHB5_xxx 寄存器控制，但该外设未连接到 AHB5。由于总线拓扑，APB4 上的外设可通过 AHB6 访问。*

表 67. MLAHB 和 AHB4 的总线时钟使能详细信息

MPU 模式	-	MLAHB (AHB4) 时钟	说明
CRun	→	工作	当以下条件之一为真时，将生成 MLAHB 和 AHB4 时钟： – MPU 处于 CRun 或 CSleep 模式 ==> 当系统处于运行模式时
CSleep	→		
CStop/ CStandby	→	门控	在所有其他情况下，MLAHB 和 AHB4 时钟将被门控。

表 68. AHB2 和 AHB1 的总线时钟使能详细信息

MPU 模式(1)	mpu_peren_ahb1 (mpu_peren_ahb2) (2)	mpu_perlpen_ahb1 (mpu_perlpen_ahb2) (2)	-	AHB1 (AHB2) 时钟	说明
CRun	1	X	→	工作	当以下条件之一为真时，将生成 AHB1 (AHB2) 时钟： – MPU 处于 CRun 模式，并且 AHB1 (AHB2) 上至少分配了一个外设。 – MPU 处于 CSleep 模式，并且 AHB1 (AHB2) 上至少分配了一个外设，并且激活了睡眠使能位。
CSleep	1	1	→		
任意	0	X	→	门控	在所有其他情况下，AHB1 (AHB2) 时钟将被门控
CSleep	X	0	→		
CStop/ CStandby	X	X	→		

1. “任何”指任何模式。MPU 可以是 CRun、CSleep、CStop 或 CStandby。
2. 当至少有一个连接到 AHB2、APB1 或 APB2 的外设被分配给 MPU 时，**mpu_peren_ahb2** 置“1”。
   当至少有一个连接到 AHB2、APB1 或 APB2 的外设分配给 MPU，并且其 PERxLPEN 位置“1”时，**mpu_perlpen_ahb2** 置“1”。
   当至少有一个连接到 AHB1 的外设被分配给 MPU 时，**mpu_peren_ahb1** 置“1”。
   当至少有一个连接到 AHB1 的外设分配给 MPU，并且其 PERxLPEN 位置“1”时，**mpu_perlpen_ahb1** 置“1”。

表 69. APB4 和 APB[2:1] 的总线时钟使能详细信息

MPU 模式 (1)	mpu_peren_apb4 (mpu_peren_apb[6;2:1]) (2)	mpu_perlpen_apb4 (mpu_perlpen_apb[6;2:1]) (2)	-	APB4 (APB[6;2:1]) 时钟	说明
CRun	1	X	→	工作	当以下条件之一为真时，将生成 APB4 (APB[6;2:1]) 时钟： – MPU 处于 CRun 模式，并且 APB4 (APB[6;2:1]) 上至少分配了一个外设（即至少一个 MC_PERxEN = “1”） – MPU 处于 CSleep 模式，并且 APB4 (APB[6;2:1]) 上至少分配了一个外设，并且激活了睡眠使能位（即 MP_PERxEN = “1”并且 MP_PERxLPEN = “1”）
CSleep	1	1	→		

表 69. APB4 和 APB[2:1] 的总线时钟使能详细信息（续）

MPU 模式 (1)	mpu_peren_apb4 (mpu_peren_apb[6;2:1]) (2)	mpu_perlpen_apb4 (mpu_perlpen_apb[6;2:1]) (2)	-	APB4 (APB[6;2:1]) 时钟	说明
任意	0	X	→	门控	在所有其他情况下，APB1、APB2、APB4 和 APB6 时钟被门控
CSleep	X	0	→		
CStop/ CStandby	X	X	→		

1. “任何”指任何模式。MPU 可以是 CRun、CSleep、CStop 或 CStandby。
2. 当至少有一个连接到 APB4 的外设被分配给 MPU 时，**mpu_peren_apb4** 置“1”。
   当至少有一个连接到 APB4 的外设分配给 MPU，并且其 PERxLPEN 位置“1”时，**mpu_perlpen_apb4** 置“1”。

备注

只要 MPU 不再处于 CStop 模式，就会为 MLAHB 和 AHB4 提供时钟。这是因为 RCC 和 PWR 模块被隐式分配给 MPU。

当 MPU 处于 CStop 模式时，AXIM 仍然保持时钟被门控。

10.4.13 低功耗仿真模式

RCC 能够处理针对 Stop 和待机模式的仿真模式，以简化电路调试。其他相关信息，请参见 DBG 一节中的*调试模式下的低功耗模式仿真*一节。

Stop 仿真模式

Stop 仿真模式由 DBGMCU_CR 寄存器的 DBGLP 位控制。

如果 MPU 进入 CStop，系统进入 Stop、LP-Stop 或 LPLV-Stop 中的一种模式，并且 DBGLP = “1”，则：

- 处理器时钟、所有已分配外设的时钟、调试部分和互连都将保持激活状态
- PWR 不通过 PWR_ON 和 PWR_LP 信号来指示低功耗模式。这可确保 V_{DDCORE} 域和 V_{DDCPU} 域在主模式下仍由稳压器供电
- 当发生唤醒事件时，PWR 和 EXTI 模块正常工作。
- 由于时钟仍在，RCC 旁路（或非常快速地执行）时钟恢复序列。

如果 MPU 进入 CStandby，系统进入 LPLV-Stop2，并且 DBGLP = “1”，则：

- 处理器时钟、所有已分配外设的时钟、调试部分和互连都将保持激活状态。
- 连接到 GIC 的中断线立即被门控，以防止 MPU 在复位前执行异常。
- PWR 不通过 PWR_ON 和 PWR_LP 信号来指示低功耗模式。这可确保 V_{DDCORE} 域和 V_{DDCPU} 域在主模式下仍由稳压器供电
- 当发生唤醒事件时，PWR 和 EXTI 模块正常工作，RCC 生成一个 V_{DDCPU} (**vcpu_rst**) 复位和一个 MPU 复位。
- 连接到 GIC 的中断线不再被门控。

待机仿真模式

待机仿真模式由 DBGMCU_CR 寄存器的 DBGSTBY 位控制。

如果 MPU 进入 CStandby，系统进入待机模式，并且 DBGSTBY = “1”，则：

- 处理器时钟、所有已分配外设的时钟、调试部分和互连都将保持激活状态。
- 连接到 GIC 的中断线立即被门控，以防止 MPU 在复位前执行异常。
- PWR 不通过 PWR_ON 和 PWR_LP 信号来指示低功耗模式。这可确保 V_{DDCORE} 域和 V_{DDCPU} 域在主模式下仍由稳压器供电
- 当发生唤醒事件时，PWR 和 EXTI 模块正常工作，RCC 生成一个完整 V_{DDCORE} (**vcore_rst**) 的复位，但不复位调试部分（**dbg_rstn** 未激活），并且 RCC 生成一个 V_{DDCPU} (**vcpu_rst**) 复位和一个 MPU 复位。
- 中断线不再被门控。

10.5 RCC 安全性

RCC 在其子功能级别上符合 TrustZone 硬件架构，将其所有资源进行分区，确保这些资源在任意指定时间要么存在于安全环境中，要么存在于非安全环境中。

安全软件代理可以独立地在位级别对始终安全的 RCC_SECCFGR 寄存器进行编程，以便将每个子功能/资源分配和划分为安全或非安全：

- 振荡器和源时钟：
  - HSI 时钟（通过 HSISEC）
  - CSI 时钟（通过 CSISEC）
  - HSE 时钟（通过 HSESEC）
  - LSI 时钟（通过 LSISEC）
  - LSE 时钟（通过 LSESEC）
- 中间时钟：
  - PLL1 和 PLL2 时钟（通过 PLL12SEC）
  - PLL3 时钟（通过 PLL3SEC）
  - PLL4 时钟（通过 PLL4SEC）
  - MPU 子系统时钟（通过 MPUSEC）
  - AXI 子系统时钟（通过 AXISEC）
  - MLAHB 子系统时钟（通过 MLAHBSEC）
- Sink 时钟预分频器：
  - APBx 分频时钟 (x=3,4,5,6)（通过 APBxDIVSEC）
  - 定时器组 3 预分频器（通过 TIMG3SEC）
- Sink/输出时钟：
  - 通用外设时钟（通过 CPERSEC）
  - MCO1 输出时钟（通过 MCO1SEC）
  - MCO2 输出时钟（通过 MCO2SEC）
- 其他功能，其安全状态为：
  - 复位（通过 RSTSEC）
  - Stop 模式（通过 STPSEC）
  - PWR 模块（通过 PWRSEC）（通过输出信号 rcc_pwr_sec 提供给 PWR 模块块）

当资源/子功能由安全代理配置为安全（例如 RCC_SECCFGR.xxxSEC =“1”）时，借助 RCC AHB TrustZone 从接口，在访问 RCC 寄存器字段（该位字段将影响此安全子功能的控制）时应用以下规则

- 写访问必须是安全写访问。
- 非安全写访问不影响寄存器字段，RCC 会产生总线错误，并作为硬故障通知给用户。
- 始终允许读访问。

如*表 70* 所示，除了编程的 RCC_SECFGR 外，RCC 还通过指定外设的安全状态来管理该外设的复位控制和时钟控制的安全性（运行模式和低功耗模式下的内核时钟源选择及外设时钟门控），如以下一种方式定义：

- ETZPC 寄存器的相应 DECPROT[1] 位字段，适用于任何集中式可保护外设（IWDG1、DDR 接口、I2C[5:3]、SPI5、SPI/I2S4、USART[2:1]、LPTIM[3:2]、ADC[2:1]、SDMMC[2:1]、ETH[2:1]、USBO、USBPHY、QSPI、FMC、RNG1、STGENC、DCMIPP、VREF、LTDC、TIM[17:12]、HASH、CRYP、SAES、PKA、DDRMCE、BKPSRAM、SRAM3、SRAM2、SRAM1）
- 由 TrustZone 感知外设自身指定的安全状态（对于 RTC）
- 始终安全的外设（SYSCFG、DMAMUX2、DMA3、BSEC、ETZPC、TZC）。

有一些额外的特定情况，特别是外设不完全是安全或非安全时的情况：

- GPIO[A-I] 端口、AXIM 和 MDMA 的复位控制被强制为安全写保护。
- SYSCFG、LTDC、GPIO[A:I] 端口、MDMA 和 SYSRAM 的时钟门控可以由安全和非安全软件代理独立控制，如下所述：
  - 外设时钟在任意（授权的）软件代理将其使能时被使能
  - 外设时钟在安全和非安全软件代理均将其禁止时（如系统复位后）被禁止。
- PWR 模块，其安全状态由 RCC_SECCFGR.PWRSEC 位定义。

表 70. RCC 安全性和外设/内存

安全策略	外设 内存
始终安全	BSEC
	DMA3
	DMAMUX2
	ETZPC
	SYSCFG
	TZC
TrustZone 感知	RCC
	RTC
RCC 驱动可保护	PWR

表 70. RCC 安全性和外设/内存（续）

安全策略	外设 内存
集中式 ETZPC 驱动可保护	ADC1
	ADC2
	BKPSRAM
	CRYP
	DCMIPP
	DDRC 和 DDRPHYC
	ETH1
	ETH2
	FMC
	HASH
	I2C3
	I2C4
	I2C5
	IWDG1
	LPTIM2
	LPTIM3
	LTDC
	DDRMCE
	PKA
	QSPI
	RNG1
	SAES
	SDMMC1
	SDMMC2
	SPI/I2S4
	SPI5
	SRAM1
	SRAM2
	SRAM3
	STGENC

表 70. RCC 安全性和外设/内存（续）

安全策略	外设 内存
集中式 ETZPC 驱动可保护	TIM12
	TIM13
	TIM14
	TIM15
	TIM16
	TIM17
	USART1
	USART2
	USBO
	USBPHY

10.6 RCC 中断

RCC 为 MPU 提供一个专用中断接口，包含三条中断线：

- 一条专用中断线 (**rcc_mpu_wkup_it**)，用于将 MPU 从 CStop 唤醒。
- 一条通用安全中断线 (**rcc_sec_mpu_it**)，用于收集任何安全中断事件（CStop 唤醒引发），如 RCC_SECCCFGR 寄存器中相应 RCC 子功能的安全状态所定义。
- 一条通用非安全中断线 (**rcc_nonsec_mpu_it**)，用于收集任何非安全中断事件（CStop 唤醒引发），如 RCC_SECCCFGR 寄存器中相应 RCC 子功能的安全状态所定义。

RCC 中断事件包括：

- 从 CStop 唤醒
- 检测到 LSE 时钟故障
- PLL（PLL1、PLL2、PLL3 或 PLL4）就绪
- 振荡器（LSI、LSE、HSI、HSE 或 CSI）就绪

任何中断线均为高电平有效，并且是电平触发。

任何中断事件都可以通过 *RCC 时钟源中断使能寄存器 (RCC_MP_CIER)* 中的独立中断使能控制位单独使能或禁止。每个中断使能位均受到安全写保护，具体取决于 RCC_SECCCFGR 寄存器定义的相应 RCC 子功能的安全状态。

任何中断标志均可通过 *RCC 时钟源中断标志寄存器 (RCC_MP_CIFR)* 检查并清除。任何中断标志均可以由安全或非安全软件代理读取。任何中断标志均可以单独、安全地在位字段级受到写保护，具体取决于 RCC_SECCCFGR 寄存器定义的相应 RCC 子功能的安全状态。

当安全中断事件标志置“1”，并且其相应的安全使能位也置“1”时，**rcc_sec_mpu_it** 线即被激活。要释放 **rcc_sec_mpu_it** 线，必须清除所有已使能的安全事件标志。

当非安全中断事件标志置“1”，并且其相应的非安全使能位也置“1”时，**rcc_nonsec_mpu_it** 线即被激活。要释放 **rcc_nonsec_mpu_it** 线，必须清除所有已使能的非安全事件标志。

如果 RCC 需要将 MPU 从 CStop 退出，RCC_MP_CIER.WKUPIE 位置“1”，同时 **rcc_mpu_wkup_it** 线被激活。要释放 **rcc_mpu_wkup_it** 线，MPU 必须向相应寄存器的 RCC_MP_CIFR.WKUPF 位写入“1”。

事件标志在相应就绪位（例如 xxxRDY）的上升沿变为“1”。

注： *即使相应的中断使能位未置 1，中断标志也相关。*

表 71 汇总了中断源及其控制方式。

表 71. 中断源和控制

中断线	中断事件	中断使能 (RCC_MP_CIER)	事件标志 (RCC_MP_CIFR)	事件清除方法 （RCC_MP_CIFR）
rcc_sec_mpu_it, rcc_nonsec_mpu_it	LSI 就绪	LSIRDYIE	LSIRDYF	将 LSIRDYF 置“1”
	LSE 就绪	LSERDYIE	LSERDYF	将 LSERDYF 置“1”
	HSI 就绪	HSIDRYIE	HSIDRYF	将 HSIRDYF 置“1”
	HSE 就绪	HSERDYIE	HSERDYF	将 HSERDYF 置“1”
	CSI 就绪	CSIRDYIE	CSIRDYF	将 CSIRDYF 置“1”
	PLL1 就绪	PLL1RDYIE	PLL1RDYF	将 PLL1RDYF 置“1”
	PLL2 就绪	PLL2RDYIE	PLL2RDYF	将 PLL2RDYF 置“1”
	PLL3 就绪	PLL3RDYIE	PLL3RDYF	将 PLL3RDYF 置“1”
	PLL4 就绪	PLL4RDYIE	PLL4RDYF	将 PLL4RDYF 置“1”
	LSE 时钟安全系统故障	LSECSSIE (1)	LSECSSF	将 LSECSSF 置“1”
rcc_mpu_wkup_it	从 CStop 事件唤醒	WKUPIE	WKUPF	将 WKUPF 置“1”

1. 为生成中断，还必须使能安全系统功能（LSECSSON = “1”）。

10.7 RCC 应用信息

10.7.1 处理动态时钟切换

实时更改处理器和总线桥的时钟源及时钟频率时，务必要小心：

- 应用必须确保在时钟切换过程中，所有必要资源均能获得时钟，以确保执行转换的处理器能正确获得时钟。例如，如果处理器正在进行需要分步执行的时钟更改，则必须使处理器、其内存、总线桥和 RCC 能获得时钟，以便软件正确执行时钟转换。如果其中一个元件不再能获得时钟，就无法完成转换，系统会被阻止。
- 应用还必须确保提供给处理器、总线桥和外围设备的时钟频率永远不会超出最大允许值。

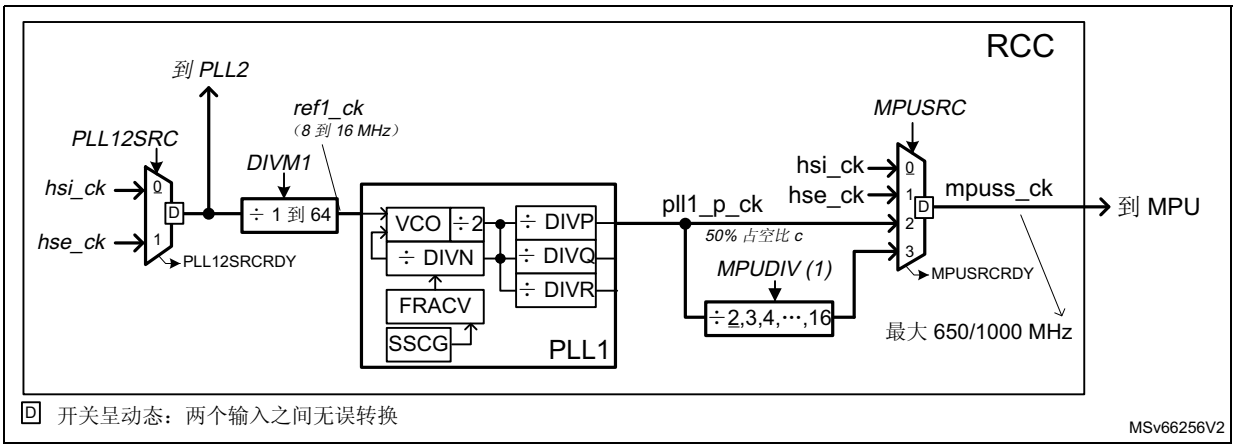
此外，更改互连上的时钟也可能会影响某些外设的性能，特别是当外设使用 APB、AHB 或 AXI 时钟作为内核时钟时。

更改 MPU 时钟频率

可通过以下不同方式更改 MPU 的时钟频率：

- 更改 MPUSRC 值
- 更改 MPUDIV 值
- 更改 PLL1 设置

图 98. MPU 时钟路径



1. 可实时更改 MPUDIV。
   X 表示系统复位后所选择的复用器输入。

通过 MPUSRC 更改 MPU 时钟频率

如果 MPU 的时钟是使用 MPUSRC 更改的，则应用必须：

– 确保在将 MPUSRC 切换到此输入之前，下一个时钟源已准备就绪
– 确保在禁止上一个时钟源之前已完成 MPUSRC 切换。

具体过程如下：

– 使能所需的新时钟
– 等待该时钟就绪
– 将 MPUSRC 切换到该新时钟
– 等待至位 MPUSRCRDY =“1”
– 之前的时钟源即可被禁止

通过 MPUDIV 更改 MPU 时钟频率

如果 MPU 时钟频率是使用 MPUDIV 更改的，则应用需注意两点：

- 如果 MPUSRC = “3”，则不要将 MPUDIV 设为 “000”，否则不再为处理器提供时钟
- 不要超过出 **mpuss_ck** 的最大允许频率。

通过 PLL1 更改 MPU 时钟频率

如果是通过 PLL1 更改 MPU 时钟频率，则只要向新时钟频率跳变的过程未完成，应用就必须为 MPU 提供一个备份时钟。

如果 MPU 已经在使用 PLL1，则必须遵循以下过程：

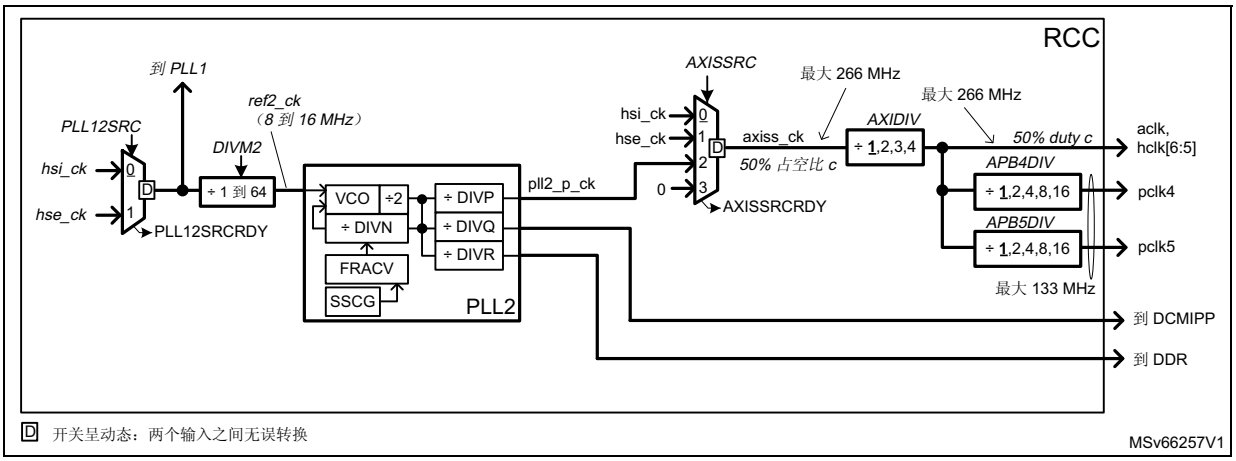
a) 使能备份时钟（如果尚未使能），例如 HSI 或 HSE
b) 等待该备份时钟就绪
c) 切换 MPUSRC 以选择该备份时钟
d) 等待切换跳变完成
e) 更改 PLL 设置：可以是 VCO 重新编程，由于 PLL 锁定，可能需要几十微秒的时间（参见 *PLL 初始化过程*一节），或者是更简单更快速的后分频器重新编程（参见*重新编程后分频器*一节）
f) 等待 PLL1 就绪
g) 将 MPUSRC 切换到 pll1_p_ck
h) 等待至 MPUSRCRDY = “1”
i) 即可禁止备份时钟。

更改 axiss_ck 时钟源和频率：

可通过以下不同方式更改 AXI 的时钟频率：

- 更改 AXISSRC 值
- 更改 AXIDIV 值
- 更改 PLL2 设置

图 99. PLL2 时钟路径



1. **X** 表示系统复位后所选择的复用器输入。

通过 AXISSRC 更改 axiss_ck 频率：

如果是使用 AXISSRC 更改 **axiss_ck** 时钟频率，应用必须执行以下操作：

- 确认当前 AXIDIV 设置与新时钟源未超出 AXIM、AHB[6:5] 和 APB[5:4] 的最大允许频率。如果未超出，则：
  - 更改 AXIDIV、APB4DIV 和 APB5DIV 以确保新的时钟频率不会超出 AXIM、APB4 和 APB5 总线的最大频率，
  - 等待 AXIDIV、APB4DIV 和 APB5DIV 就绪，以避免在新的分频比生效之前切换到新的时钟源。
- 生成新的时钟源，并等待该时钟源就绪，
- 切换 AXISSRC 以选择新的时钟源，
- 等待至 AXISSRCRDY =“1”，然后再关闭上一个时钟源。

注： *在关闭上一个时钟源之前，应确保该时钟未被其他子系统使用。*

通过 AXIDIV 更改 axiss_ck 频率：

只需更改 AXIDIV 值，即可实时更改 AXIDIV。

应用必须确保 AXIDIV 输出的频率永远不会超过 AXIM、AHB[6:5] 和 APB[5:4] 的最大允许频率。

新分频值可能需要一些时间才能生效，因此应用必须始终等待分频器的就绪位（AXIDIVRDY、APB[5:4]DIVRDY），以确保新的分频值生效。

通过 PLL2 更改 axiss_ck 频率：

如果是通过 PLL2 更改 **axiss_ck** 时钟频率，则只要向新时钟频率跳变的过程未完成，应用就必须通过备份时钟生成 **axiss_ck**。如果未提供备份时钟，MPU 将无法继续运行，因为 AXI 总线桥不可用。

通过以下操作可以只更改 PLL2 的 DIVP 输出：

- 使能备份时钟（如果尚未使能），例如 HSI 或 HSE
- 等待该备份时钟就绪
- 将 AXISSRC 切换到该备份时钟，
- 等待切换跳变完成（AXISSRCRDY =“1”），
- 确认当前 AXIDIV 设置与新时钟源未超出 AXI 的最大允许频率。如果未超出，则
  - 更改 AXIDIV 以确保新的时钟频率不会超出 AXI 总线的最大频率，
  - 等待 AXIDIV 就绪，以避免在新的分频比生效之前切换到新的时钟源。
- 将 PLL2 的 DIVPEN 位设为“0”
- 更改 DIVP 的值
- 将 PLL2 的 DIVPEN 位置“1”
- 将 AXISSRC 切换到该新时钟
- 等待至 AXISSRCRDY =“1”
- 即可禁止备份时钟。

更改 PLL2 的 VCO 频率的影响较大，因为 PLL2 还为 DDR 接口和 DCMIPP 提供时钟。

要更改 PLL2 的 VCO 频率，应用必须执行以下步骤（假设使用 DDR 来执行应用代码）：

- 停止或暂停 DCMIPP 活动以及使用 DDR 的所有其他外设，
- 将进行时钟跳变的代码加载到 SYSRAM 中，
- 将控制权交给 SYSRAM 中的代码。

在这种状态下，不得再使用 DDR，

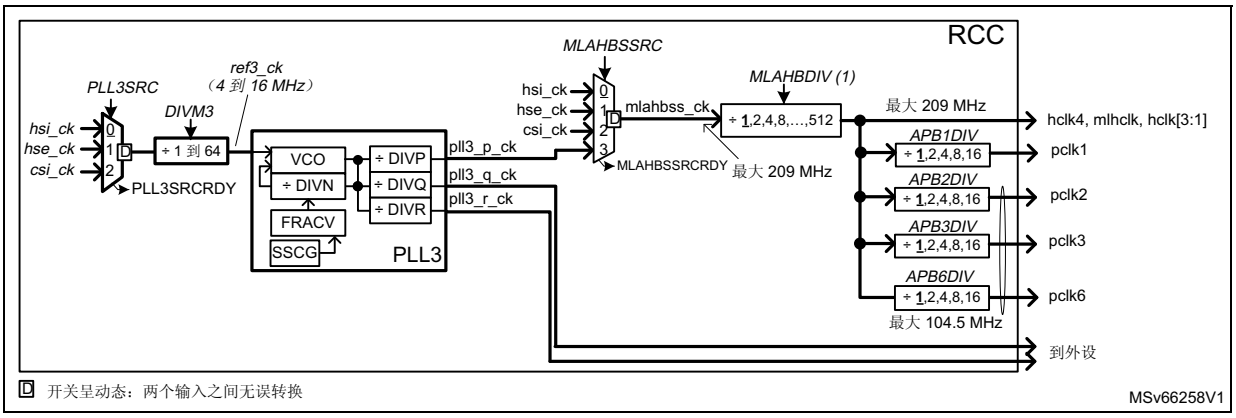
- 将 DDR 设置为自刷新状态，
- 使能备份时钟（如果尚未使能），例如 HSI 或 HSE
- 等待该备份时钟就绪
- 将 AXISSRC 切换到这个临时的备用时钟（AXISSRC = “0” 或 “1”），
- 等待切换跳变完成
- 确认当前 AXIDIV 设置与所需的新时钟频率未超出 AXI 的最大允许频率。如果未超出，则：
  - 更改 AXIDIV 以确保新的时钟频率不会超出 AXI 总线的最大频率，
  - 等待 AXIDIV 就绪，以避免在新的分频比生效之前切换到新的时钟源。
- 更改 PLL 设置：有关详细信息，请参见 *PLL 初始化过程*一节。由于 PLL 锁定时间，该步骤可能需要几微秒，
- 将 AXISSRC 切换到这个新时钟（AXISSRC = “2”），
- 等待至 AXISSRCRDY = “1”
- 然后即可禁止备份时钟，
- 必须重新初始化 DDR 控制器，因为 DDR 接口的时钟可能已经由于 PLL2 VCO 频率的变化而更改。
- 然后应用可以再次切换到 DDR，
- 可以重新使能使用 DDR 的 DCMIPP 和其他外设。

更改 MLAHB 时钟源和频率：

可以通过以下不同的方式更改 MLAHB 时钟频率 (**mlahbss_ck**)：

- 更改 MLAHBSSRC 值，
- 更改 MLAHBDIV 值，
- 更改 PLL3 设置。

图 100. MLAHB 时钟路径



1. **X** 表示系统复位后所选择的复用器输入。

通过 MLAHBSSRC 更改 MLAHB 时钟频率：

如果是使用 MLAHBSSRC 更改 **mlahbss_ck** 时钟，应用必须执行以下操作：

- 确认当前 MLAHBDIV 设置与新时钟源未超出 MLAHB 域的最大允许频率。如果未超出，则
  - 更改 MLAHBDIV、APB1DIV、APB2DIV、APB3DIV 和 APB6DIV，以确保新的时钟频率不会超出 MLAHB、AHB[4:1]、APB[3:1] 和 APB6 总线的最大频率，
  - 等待 MLAHBDIV、APB1DIV、APB2DIV、APB3DIV 和 APB6DIV 就绪，以避免在新的分频比生效前切换到新的时钟源。
- 使能新的时钟源，并等待该时钟源就绪，
- 将 MLAHBSSRC 切换到新的时钟源，
- 等待至 MLAHBSSRCRDY =“1”，然后再关闭上一个时钟源。

通过 MLAHBDIV 更改 MLAHB 时钟频率：

只需更改 MLAHBDIV 值，即可实时更改 MLAHBDIV。

应用必须确保 MLAHBDIV 输出的频率永远不会超过 MLAHB、AHB[4:1]、APB[3:1] 和 APB6 允许的最大频率。

新的分频值可能需要一些时间才能生效，因此应用必须始终等待分频器的就绪位（MLAHBDIVRDY、APB[6;3:1]DIVRDY），以确认新的分频值是否生效。

通过 PLL3 更改 MLAHB 时钟频率：

如果是通过 PLL3 更改 MLAHB 时钟频率，则只要向新时钟频率跳变的过程未完成，应用就必须为 MLAHB 提供一个备份时钟。

如果 MLAHB 已经在使用 PLL3，则必须遵循以下过程：

a) 使能备份时钟（如果尚未使能），例如 HSI、CSI 或 HSE
b) 等待该备份时钟就绪
c) 切换 MLAHBSSRC 以选择该备份时钟
d) 等待切换跳变完成（MLAHBSSRCRDY =“1”）
e) 更改 PLL 设置：可以是 VCO 重新编程，由于 PLL 锁定，可能需要几十微秒的时间（参见 *PLL 初始化过程*一节），或者是更简单更快速的后分频器重新编程（参见*重新编程后分频器*一节）
f) 等待 PLL3 就绪
g) 切换 MLAHBSSRC 以选择 pll3_p_ck
h) 等待切换跳变完成（MLAHBSSRCRDY =“1”），
i) 即可禁止备份时钟。

10.7.2 PLL 编程

PLL 初始化过程

以下为推荐的整数和小数模式 PLL 初始化序列：

- DLL 应为禁止状态：如果 DIVPEN、DIVQEN、DIVREN、PLLON 和 PLLxRDY 不为“0”，将其设为“0”，请参见 *PLL 禁止过程*。
- 使能所需的时钟源（HSE、HSI 或 CSI）并等待就绪标志。
- 选择时钟源：通过 RCC_RCK12SELR、RCC_RCK3SELR 和 RCC_RCK4SELR 寄存器。
- 等待时钟切换就绪。
- 初始化预分频器 (DIVMx)，以便为 PLL 提供有效的参考时钟。DIVMx 在 RCC_PLLxCFGR1 寄存器中。
- 配置 PLL：
  - 在整数模式或时钟展频模式下，应用必须确保 SDM 中加载了“0”。具体方法如下：
    - 将 FRACLE 设为“0”
    - 向 FRACV 写入“0”
    - 将 FRACLE 置“1”
  - 在小数模式下，应用必须通过执行与整数模式类似（但 FRACV = FracInitValue）的序列，将所需值 (FracInitValue) 加载到 SDM 中。
  - 编程位字段 IFRGE（用于 PLL3 和 4）、SSCG_CTRL、DIVN、DIVP、DIVQ 和 DIVR。
  - 如果使用时钟展频模式，还需编程位字段 MOD_PER、INC_STEP 和 SSCG_MODE。
- 使能 PLL：
  - 将相应的 PLLON 位置“1”，并等待 PLLxRDY 位置“1”，
  - 如果 PLLx 处于小数模式，则只要 PLLxRDY = “0”，FRACLE 位就不能设回“0”。
- 当 PLLxRDY 位等于“1”时，应用可以通过将 DIVPEN、DIVQEN、DIVREN 置“1”来使能所需输出。
- 如果应用要实时调整 PLL 频率（仅限小数模式），则：
  - 将 FRACLE 设为“0”，
    当 FRACLE = “0”时，∑Δ 调制器仍在以当前值运行，
  - 将新值写入 FRACV (FracValue(n))，
  - 将 FRACLE 置“1”，以便将 FRACV 的内容锁存到 SDM 中。

注： *当 PLLxRDY 变为“1”时，意味着 PLLx 输出频率在其目标值的 2% 内。*

重新编程后分频器

使能 PLL 后，可以通过执行以下序列，在不禁止相应 PLL 的情况下更改后分频器（DIVP、DIVQ 或 DIVR）的值。

- 通过将 DIVPEN、DIVQEN 或 DIVREN 位设为“0”来禁止所需输出，
- 更改相应的后分频器（DIVP、DIVQ 或 DIVR）的值，
- 通过将 DIVPEN、DIVQEN 或 DIVREN 位置“1”来重新使能该输出。

使用 SSCG 模块

如果应用需要使用 PLL2 产生具有以下特性的时钟信号：

- 输出频率 (**Fpll2_r_ck**) 应尽可能接近 533 MHz（但不能超过），
- 调制深度为 0.5% 的峰值 (M_D)，
- 调制频率为 25 kHz (F_{MOD})。

假设晶体振荡器是 24 MHz (F_{hse_ck})。

DIVM2 被编程为 2，以便为 PLL2 提供 12 MHz 的参考时钟 (F_{ref2_ck})。

可以使用 PLL2 提供的二分频从 12 MHz 参考时钟生成 528 MHz，以确保最佳占空比。

应用必须计算以下参数：

将 *以整数模式使用 PLL* 一节 中给出的公式组合起来计算 NDIV：

$$DIVN = \frac{F_{pll_r_ck}}{F_{ck_ref}} - 1 = \frac{528}{12} - 1 = 43$$

计算 MOD_PER：

$$MOD_PER = ROUND\left(\frac{F_{ref_ck}}{4 \times F_{MOD}}\right) = ROUND\left(\frac{12 \times 10^6}{4 \times 25 \times 10^3}\right) = 120$$

计算 INC_STEP：

$$INC_STEP = ROUND\left(\frac{(2^{15} - 1) \times M_D \times (DIVN + 1)}{100 \times 5 \times MOD_PER}\right) = ROUND\left(\frac{32767 \times 0.5 \times (43 + 1)}{100 \times 5 \times 120}\right) = 12$$

确认 MOD_PER x INC_STEP 小于 (2^{15}-1)

例如，MOD_PER x INC_STEP = 1440 ==> 可行

由于四舍五入运算的原因，调制周期和调制深度未完全达到目标。为选择中心展频或向下展频调制，必须确认最大频率是否超出 533 MHz 的要求。

实际调制深度是：

$$M_D\ (\%) = \frac{MOD_PER \times INC_STEP \times 100 \times 5}{(2^{15} - 1) \times (DIVN + 1)} = \frac{120 \times 12 \times 100 \times 5}{(2^{15} - 1) \times (43 + 1)} = 0.499\ \%$$

如果使用中心展频调制，F_{MAX} 为：

$F_{MAX} = F_C$ x (1 + M_D / 100) = 528 x (1 + 0.499 / 100) = 530.6 MHz。

所以可以使用中心展频调制。

PLL 禁止过程

以下为推荐的禁止序列：

- 禁止所有后分频器：将 PLLx 的 DIVPEN、DIVQEN 和 DIVREN 位设为“0”，
- 禁止 PLL：将 PLLx 的 PLLON 位设为“0”，
- 等待至 PLLxRDY =“0”。

MPU 频率规划示例

PLL1 专用于生成 MPU 时钟。应用必须为 MPU 提供具有以下特性的时钟：

- 占空比为 50%，
- 抖动尽可能小，尤其是当 MPU 频率接近其最大值时。

下表显示了针对使用最常见晶振的 MPU 所规划的可行频率。

表 72. MPU 的时钟设置示例

HSE (MHz)	PLL1							MPUSRC	mpuss_ck (MHz)	备注
	DIVM1+1	DIVN+1	FRACV	DIVP+1	F_{ref1_ck} (MHz)	F_{VCO} (MHz)	F_{fout1_ck} (MHz)			
16	1	62	4096	1	16	2000	1000	pll1_p_ck	**1000**	通过将 MPUSRC 改为“11”并将 MPUDIV 改为 /2、/3、/4,...,/16 可以获得其他频率
	1	56	2048		16	1800	900		**900**	
	2	100	0		8	1600	800		**800**	
	2	81			8	1296	648		**648**	
	1	40			16	1280	640		**640**	
	2	75			8	1200	600		**600**	
19.2	2	104	1365	1	9.6	2000	1000	pll1_p_ck	**1000**	
	2	93	6144	1	9.6	1800	900		**900**	
		83	0			1593.6	796.8		**796.8**	
		67				1286.4	643.2		**643.2**	
		63				1190.4	595.2		**595.2**	
24	2	83	2730	1	12	2000	1000	pll1_p_ck	**1000**	
	2	75	0	1	12	1800	900		**900**	
	3	100			8	1600	800		**800**	
	2	54			12	1296	648		**648**	
	2	50			12	1200	600		**600**	
26	2	76	7561	1	13	2000	1000	pll1_p_ck	**1000**	
	2	69	1890	1	13	1800	900		**900**	
		61	0			1600	1586		**793**	
		50				1300	650		**650**	
		46				1196	598		**598**	

若要尽可能接近 50% 的占空比，并获得最佳抖动，必须遵循以下规则：

改善抖动：

- F_{ref1_ck} 必须尽可能接近 PLL 所接受的最高输入频率范围。对于 PLL1 和 PLL2，该频率为 16 MHz。
- DIVN 值必须尽可能小。
- 最好在整数模式下使用 PLL。

避免占空比减小

- 更倾向于在旁路中使用 DIVP、Q、R（DIVP、Q、R = “0”），特别是对于 MPU 和 DDRC。
- 不要使用奇数值的 DIVP、Q、R 分频器，尤其是值为 3 时。

注： *当使用 HSI 提供参考时钟时，必须小心。应用必须考虑 HSI 的精度，以确保其不超出允许的最大频率。*

AXI、DDR 和 DCMIPP 频率规划示例

表 73. AXI、DDR 和 DCMIPP 的时钟设置示例

HSE (MHz)	PLL2								AXIDIV+1	AXI 时钟 (MHz)	DDR 时钟 (MHz)	DCMIPP 时钟 (MHz)
	DIVM2+1	DIVN+1	DIVP+1	DIVQ+1	DIVR+1	F_{ref2_ck} (MHz)	F_{VCO} (MHz)	F_{fout2_ck} (MHz)				
16	1	33	2	2	1	16	1056	528	1	**264**	**528**	**264**
	1	25	2	2	1	16	800	400	1	**200**	**400**	**200**
19.2	2	55	2	2	1	9.6	1056	528	1	**264**	**528**	**264**
24	2	44	2	2	1	12	1056	528	1	**264**	**528**	**264**
	3	50	2	2	1	8	800	400	1	**200**	**400**	**200**
	2	53	4	2	2	12	1280	640	1	**160**	**320**	**160**
	2	44	2	2	2	12	1056	528	1	**264**	**264**	**264**
	2	28	2	2	2	12	672	336	1	**168**	**168**	**168**
26	2	41	2	2	1	13	1066	533	1	**266**	**533**	**266**

AXI 和 DDR 频率规划示例

10.7.3 配置子系统时钟

该示例说明了寄存器编程序列，以正确配置子系统时钟（即 **mpuss_ck**、**axiss_ck** 和 **mlahbss_ck**）。

在本例中，假设：

- 连接到 HSE 的晶振频率为 24 MHz，
- MPU 和 MLAHB 均使用 HSE 作为时钟源，
- MPU 的工作频率为 600 MHz (PLL1)，MLAHB 为 200 MHz (PLL3)，AXI 为 264 MHz (PLL2)，
- 假设 **mpuss_ck**、**axiss_ck** 和 **mlahbss_ck** 正在使用 HSI（系统复位后的情况）。

以下是推荐的编程序列：

- 使能 HSE：
  - 确认 HSEON 和 HSERDY 是否设为“0”（上电复位后的情况），
  - 将 HSEON 置“1”，并等待 HSERDY =“1”。当 HSERDY =“1”时，时钟 **hse_ck** 就绪可供使用。
- **refx_ck** 时钟设置：
  - 将 PLL12SRC 置“1”，以选择 **hse_ck** 作为 MPU 和 AXI 的时钟，并等待 PLL12SRCRDY 标志等于“1”，
  - 将 DIVM1 和 DIVM2 置“1”，以将 **ref1_ck** 和 **ref2_ck** 时钟的频率设置为 12 MHz，
  - 将 PLL3SRC 置“1”，以选择 **hse_ck** 作为 MLAHB 子系统的时钟，并等待 PLL3SRCRDY 标志等于“1”，
  - 将 DIVM3 设置为 2，以提供 8 MHz 的 **ref3_ck** 时钟。
- PLL1 设置和使能：
  - 确保 PLL1 被禁止，并且所有 PLL 输出也被禁止，否则，必须执行 *PLL 禁止过程*一节中所述的序列，
  - 将 DIVN 设置为 49d，以获得 1200 MHz 的 VCO 频率，从而使 **Ffout1_ck** = 600 MHz，
  - 将 DIVP 设为“0”。
  - 将 PLL1 设置为整数模式：
    - 将 FRACLE 设为“0”
    - 将 FRACV 设为“0”
    - 然后将 FRACLE 置“1”
  - 禁止时钟展频：SSCG_CTRL =“0”
  - 将 PLLON 置“1”
  - 等待 PLL1RDY 标志置“1”
  - 将 DIVPEN 置“1”。
- PLL2 设置和使能：
  - 确保 PLL2 被禁止，并且所有 PLL 输出也被禁止，否则，必须执行 *PLL 禁止过程*一节中所述的序列，
  - 将 DIVN 设置为 43d，以获得 1056 MHz 的 VCO 频率，从而使 **Ffout2_ck** = 528 MHz，
  - 将 DIVP 置“1”（2 分频），

  - 将 PLL2 设置为整数模式：
    - 将 FRACLE 设为“0”
    - 将 FRACV 设为“0”
    - 然后将 FRACLE 置“1”
  - 设置 INC_STEP = xx、MOD_PER = yy 并且 SSCG_MODE =“1”（向下展频调制），
  - 将 SSCG_CTRL 置“1”，
  - 将 PLLON 置“1”，
  - 等待 PLL2RDY 标志置“1”，
  - 将 DIVPEN 置“1”。
- PLL3 设置和使能：
  - 确保 PLL3 被禁止，并且所有 PLL 输出也被禁止，否则，必须执行 *PLL 禁止过程*一节中所述的序列，
  - 将 DIVN 设置为 49d，以获得 400 MHz 的 VCO 频率，
  - 将 DIVP 置“1”，以实现 2 分频，
  - 将 IFRGE 设置为“x1”，
  - 将 PLL3 设置为整数模式：
    - 将 FRACLE 设为“0”
    - 将 FRACV 设为“0”
    - 然后将 FRACLE 置“1”
  - 将 PLLON 置“1”，
  - 等待 PLL3RDY 标志置“1”，
  - 将 DIVPEN 置“1”。
- 互连时钟设置：

  在增加 **mlahbss_ck**、**axiss_ck** 和 **mpuss_ck** 的频率之前，应用必须确保总线矩阵时钟分频器已正确编程，以避免无效频率。
  - 将 MLAHBDIV 设为“0”以向 MLAHB 提供 200 MHz 的时钟，
  - 将 AXIDIV 设为“0”以获得 264 MHz 的 AXI 频率，
  - 将 APB4DIV 设为非 0 值，以确保 **Fpclk4** 小于或等于 133 MHz，
  - 将 APB5DIV 设置为大于 1 的值，以确保 **Fpclk5** 小于或等于 133 MHz，
  - 将 APB1DIV、APB2DIV、APB3DIV 和 APB6DIV 设置为非 0 值，以确保 Fpclk[6; 3:1] 小于或等于 100 MHz。
- 子系统时钟切换：
  - 将 MPUSRC 设置为 2，以选择来自 PLL1 的直接 DIVP 输出，
  - 将 AXISSRC 设置为 2，以选择来自 PLL2 的 DIVP 输出，
  - 将 MLAHBSSRC 设置为 3，以选择来自 PLL3 的 DIVP 输出，
  - 在禁止 HSI 时钟之前，请等待 MLAHBSSRCRDY =“1”、MPUSRCRDY =“1”以及 AXISSRCRDY =“1”（如果系统中未使用），

10.7.4 使用 TIMx 进行时钟校准

可以通过 TIMx 的输入捕获测量大多数时钟源发生器的频率。

使用 LSE 校准 HSI 或 CSI

RCC 提供了两个专用信号 hsi_cal_ck 和 csi_cal_ck，以执行 HSI 和 CSI 振荡器的校准。

hsi_cal_ck 表示 HSI 振荡器输出，1024 分频。csi_cal_ck 表示 CSI 振荡器输出，128 分频。

信号 hsi_cal_ck 和 csi_cal_ck 连接到 TIM12 和 TIM15 输入捕获。对于此校准，应通过 PLL3 将 HSE 用作互连时钟源。通过 hsi_cal_ck（或 csi_cal_ck）信号的连续边沿之间的互连时钟周期数提供了 HSI（或 CSI）时钟周期的测量值。

校准精度取决于 TIM15/12 时钟与 hsi_cal_ck（或 csi_cal_ck）时钟之间的频率比。比率越大，测量结果越准确。

hsi_cal_ck 的频率通常为 62.5 kHz，csi_cal_ck 的频率为 31.25 kHz。

对于 50 MHz 的定时器时钟，HSI 的校准精度约为 0.13%，CSI 的校准精度约为 0.065%。

HSI 和 CSI 振荡器均设有针对此目的的专用校准位，且支持用户访问（请参见 *RCC PLL1 和 PLL2 参考时钟选择寄存器 (RCC_RCK12SELR)* 和 *RCC CSI 配置寄存器 (RCC_CSICFGR)*）。通过 PLLx 使用 HSI 或 CSI 时，还可通过 PLL 的小数分频器对互连时钟进行微调。

使用 LSE 校准 HSI 或 CSI

将 LSE 连接到 TIMx 输入捕获的主要目的是为了精确测量 HSI 或 CSI。这需要直接或通过 PLL3 将 HSI 或 CSI 用作互连时钟源。借助 LSE 信号连续边沿之间的互连时钟计数数量，即可对内部时钟周期进行测量。利用 LSE 晶振的高精度（通常为几十 ppm），用户能以同一分辨率测定内部时钟频率，并可通过对时钟源进行微调来补偿由生产工艺造成的和/或与温度和电压相关的频率偏差。

其基本原理是基于相对的测量（例如，HSI/LSE 比）：因此，精度与两个时钟源之比紧密相关。比率越大，测量结果越准确。

HSI 和 CSI 振荡器均设有针对此目的的专用校准位，且支持用户访问（请参见 *RCC PLL1 和 PLL2 参考时钟选择寄存器 (RCC_RCK12SELR)* 和 *RCC CSI 配置寄存器 (RCC_CSICFGR)*）。通过 PLLx 使用 HSI 或 CSI 时，还可通过 PLL 的小数分频器对互连时钟进行微调。

使用 HSI 校准 LSI

同时也可测量 LSI 频率：这对于没有晶振的应用场合非常实用。超低功耗 LSI 振荡器具有较大的生产工艺偏差：通过测量该振荡器与 HSI 时钟源的比率，可以借助 HSI 精度为其测定频率。可通过此测量值实现更加精确的 RTC 时基超时（当使用 LSI 作为 RTC 时钟源时）和/或实现精度水平可以接受的 IWDG 超时。

10.7.5 时钟恢复序列示例

以下一节提供了执行时钟恢复序列的一些简化时序示例。

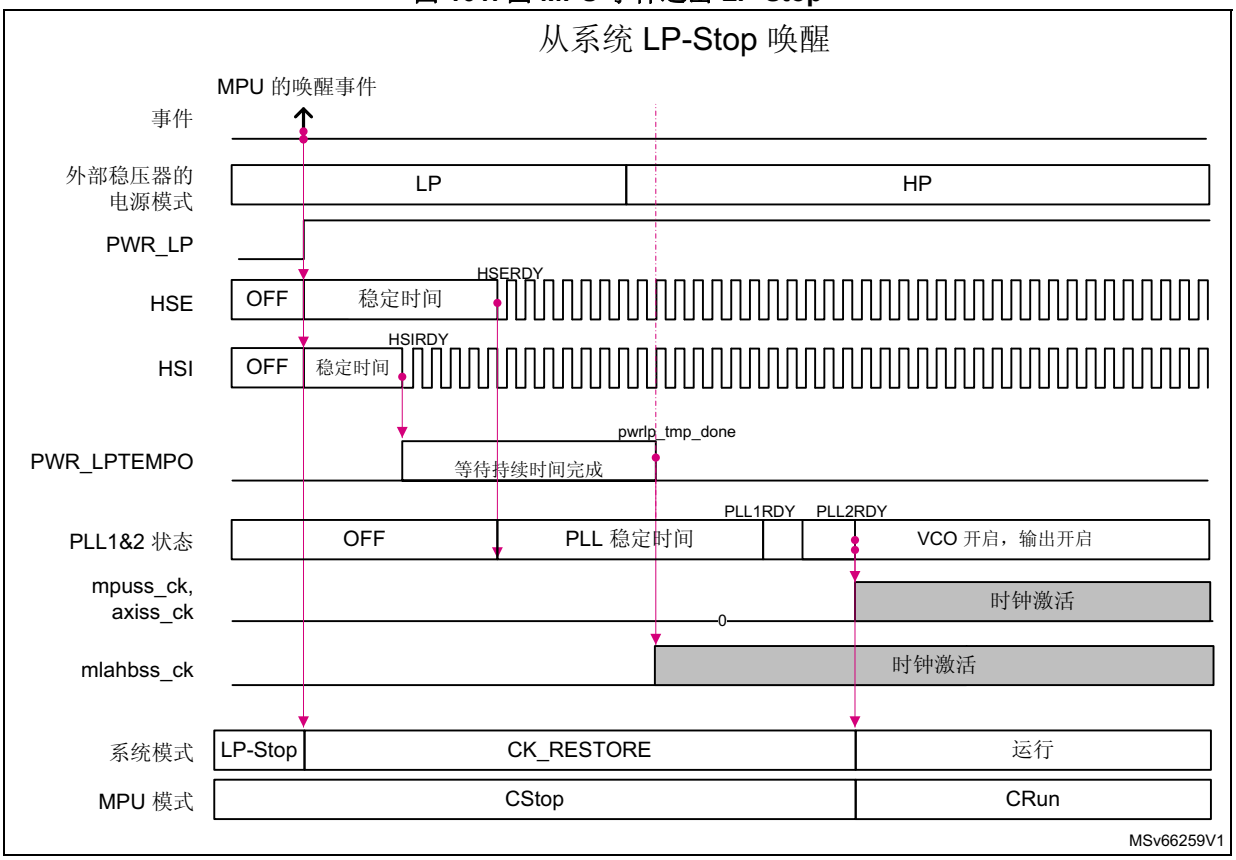
因 MPU 事件而退出 LP-Stop

在此示例中，系统因 MPU 事件而退出 LP-Stop 模式。

图 101 基于以下配置：

- **mpuss_ck** 由 PLL1 通过 HSE 提供，
- **axiss_ck** 由 PLL2 通过 HSE 提供，
- **mlahbss_ck** 由 HSI 提供

图 101. 因 MPU 事件退出 LP-Stop



其他相关信息，请参见*退出 Stop、LP-Stop、LPLV-Stop 或 LPLV-Stop2 模式*。

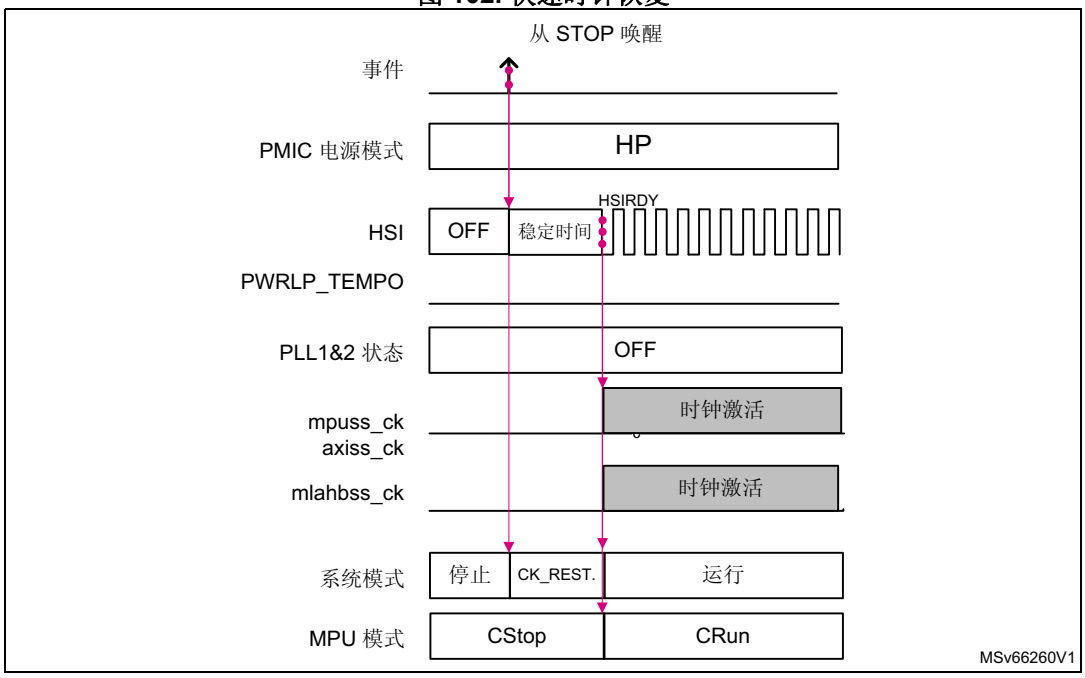
快速时钟恢复

在此示例中，系统因 MPU 事件而退出 Stop 模式。为了加速 CStop-CRun-CStop 序列的发生，系统切换回 HSI 用作 MPU、AXI 和 MLAHB 的时钟源。这样，时钟恢复序列缩短至 HSI 时钟激活。

图 102 基于以下配置：

- 在进入 CStop 之前，**mpuss_ck** 由 HSI 提供
- 在进入 CStop 之前，**axiss_ck** 由 HSI 提供
- **mlahbss_ck** 由 HSI 提供
- 位字段 PWRLP_DLY 设为“0”，以旁路低功耗开关延迟。

图 102. 快速时钟恢复



其他相关信息，请参见*退出 Stop、LP-Stop、LPLV-Stop 或 LPLV-Stop2 模式*。

10.8 RCC 寄存器

除非另有明确规定，否则可按字、半字和字节格式访问 RCC 寄存器。

除非另有明确规定，否则所有 RCC 寄存器都通过 **nreset** 信号复位。

请注意，不支持突发访问模式。

10.8.1 RCC 安全配置寄存器 (RCC_SECCFGR)

RCC secure configuration register

偏移地址：0x000

复位值：0x83FF 3F1F

该寄存器的写访问必须是安全访问。读访问可以是安全或非安全访问。

为了安全地将每一个相应的 RCC 资源/子功能分配和划分至安全环境或非安全环境，可以在位级别对该寄存器进行编程。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PWRSEC	Res.	Res.	Res.	Res.	Res.	RSTSEC	STPSEC	MCO2SEC	MCO1SEC	CPERSEC	TIMG3SEC	APB6DIVSEC	APB5DIVSEC	APB4DIVSEC	APB3DIVSEC
rw						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	MLAHBSEC	AXISEC	MPUSEC	PLL4SEC	PLL3SEC	PLL12SEC	Res.	Res.	Res.	LSESEC	LSISEC	HSESEC	CSISEC	HSISEC
		rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31 **PWRSEC**：PWR 模块的安全状态 (Secure state of the PWR block)
该控制位生成连接到 PWR 模块的输出信号 rcc_pwr_sec。
0：非安全
1：安全（复位后的默认值）

位 30:26 保留，必须保持复位值。

位 25 **RSTSEC**：复位的安全状态 (Secure state of the reset)
0：非安全
1：安全（复位后的默认值）

位 24 **STPSEC**：Stop 模式的安全状态 (Secure state of the Stop modes)
0：非安全
1：安全（复位后的默认值）

位 23 **MCO2SEC**：MCO2 时钟的安全状态 (Secure state of the MCO2 clock)
0：非安全
1：安全（复位后的默认值）

位 22 **MCO1SEC**：MCO1 时钟的安全状态 (Secure state of the MCO1 clock)
0：非安全
1：安全（复位后的默认值）

位 21 **CPERSEC**：通用外设时钟的安全状态 (Secure state of the common peripheral clock)
0：非安全
1：安全（复位后的默认值）

位 20 **TIMG3SEC**：TIMG3 预分频器的安全状态 (Secure state of the TIMG3 prescaler)
0：非安全
1：安全（复位后的默认值）

位 19 **APB6DIVSEC**：APB6DIV 分频器的安全状态 (Secure state of the APB6DIV divider)
0：非安全
1：安全（复位后的默认值）

位 18 **APB5DIVSEC**：APB5DIV 分频器的安全状态 (Secure state of the APB5DIV divider)
0：非安全
1：安全（复位后的默认值）

位 17 **APB4DIVSEC**：APB4DIV 分频器的安全状态 (Secure state of the APB4DIV divider)
0：非安全
1：安全（复位后的默认值）

位 16 **APB3DIVSEC**：APB3DIV 分频器的安全状态 (Secure state of the APB3DIV divider)
0：非安全
1：安全（复位后的默认值）

位 15:14 保留，必须保持复位值。

位 13 **MLAHBSEC**：MLAHB 子系统时钟的安全状态 (Secure state of the MLAHB sub-system clock)
0：非安全
1：安全（复位后的默认值）

位 12 **AXISEC**：AXI 子系统时钟的安全状态 (Secure state of the AXI sub-system clock)
0：非安全
1：安全（复位后的默认值）

位 11 **MPUSEC**：MPU 子系统时钟的安全状态 (Secure state of the MPU sub-system clock)
0：非安全
1：安全（复位后的默认值）

位 10 **PLL4SEC**：PLL4 时钟的安全状态 (Secure state of the PLL4 clock)
0：非安全
1：安全（复位后的默认值）

位 9 **PLL3SEC**：PLL3 时钟的安全状态 (Secure state of the PLL3 clock)
0：非安全
1：安全（复位后的默认值）

位 8 **PLL12SEC**：PLL1 和 PLL2 时钟的安全状态 (Secure state of the PLL1 and PLL2 clocks)
0：非安全
1：安全（复位后的默认值）

位 7:5 保留，必须保持复位值。

位 4 **LSESEC**：LSE 时钟的安全状态 (Secure state of the LSE clock)
0：非安全
1：安全（复位后的默认值）

位 3 **LSISEC**：LSI 时钟的安全状态 (Secure state of the LSI clock)
0：非安全
1：安全（复位后的默认值）

位 2 **HSESEC**：HSE 时钟的安全状态 (Secure state of the HSE clock)
0：非安全
1：安全（复位后的默认值）

位 1 **CSISEC**：CSI 时钟的安全状态 (Secure state of the CSI clock)
0：非安全
1：安全（复位后的默认值）

位 0 **HSISEC**：HSI 时钟的安全状态 (Secure state of the HSI clock)
0：非安全
1：安全（复位后的默认值）

10.8.2 RCC 停止请求设置寄存器 (RCC_MP_SREQSETR)

RCC stop request set register

偏移地址：0x100

复位值：0x0000 0000

如果 RCC_SECCFGR.STPSEC=1，则该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

写入“0”无影响，读取则会返回位的值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STPREQ_P0
															rs

位 31:1 保留，必须保持复位值。

位 0 **STPREQ_P0**：来自 MPU 处理器的停止请求 (Stop request from MPU processor)
由软件置 1
表示当 MPU 处于 WFI 状态时，MPU 处理器是否允许 MPU 域进入 CStop/CStandby 模式。
注： 选择进入 CStop 还是 CStandby 由 PWR_MPUCR.PDDS 和 PWR_CR1.STOP2 控制位控制。
0：写入“0”无影响，读取“0”表示 MPU 处理器不允许 MPU 域进入 CStop/CStandby 模式；因此，如果 MPU 处于 WFI 状态，它将保持在 CSleep 模式
1：写入“1”会将 STPREQ_P0 位置“1”，而读取“1”则表示 MPU 处理器允许 MPU 域进入 CStop

10.8.3 RCC 停止请求清除寄存器 (RCC_MP_SREQCLRR)

RCC stop request clear register

偏移地址：0x104

复位值：0x0000 0000

如果 RCC_SECCFGR.STPSEC=1，则该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

写入“0”无影响，读取则会返回位的值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STPREQ_P0
															rc_w1

位 31:1 保留，必须保持复位值。

位 0 **STPREQ_P0**：来自 MPU 处理器的停止请求 (Stop request from MPU processor)

由软件清零

表示当 MPU 处于 WFI 状态时，MPU 处理器是否允许 MPU 域进入 CStop/CStandby 模式。

注： 选择进入 CStop 还是 CStandby 由 PWR_MPUCR.PDDS 和 PWR_CR1.STOP2 控制位控制。

0：写入“0”无影响，读取“0”表示 MPU 处理器不允许 MPU 域进入 CStop/CStandby 模式；因此，如果 MPU 处于 WFI 状态，它将保持在 CSleep 模式

1：写入“1”会将 STPREQ_P0 位清零，读取“1”则表示 MPU 处理器允许 MPU 域进入 CStop

10.8.4 RCC 应用复位控制寄存器 (RCC_MP_APRSTCR)

RCC application reset control register

偏移地址：0x108

复位值：0x0000 7F00

如果 RCC_SECCFGR.RSTSEC=1，则该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制热复位的行为。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RSTTO[6:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDCTLEN
	rw	rw	rw	rw	rw	rw	rw								rw

位 31:15 保留，必须保持复位值。

位 14:8 **RSTTO[6:0]**：复位超时延迟调整 (Reset timeout delay adjust)

由软件写入，以定义 app_rst 延迟的最长时间。超时值使用 hsi_ck 时钟计数，可根据 HSIDIV 对该时钟分频。

仅当 RDCTLEN = “1” 时，该功能才可用。

如果 RDCTLEN = “1”，则不允许更改该位字段的值。有关超时值编程的其他相关信息，请参见*第 10.3.9 节：复位延迟控制 (RDCTL)*。

0x0：超时功能已禁止

0x1：超时设为 2 x 2^{HSIDIV} us

...

0x7F：超时设为 128 x 2^{HSIDIV} us（复位后的默认值）

位 7:1 保留，必须保持复位值。

位 0 **RDCTLEN**：复位延迟控制使能 (Reset delay control enable)

由软件置 1 和复位。

0：RDCTL 控制模块被旁路（复位后的默认值）

1：RDCTL 控制模块已使能。

10.8.5 RCC 应用复位状态寄存器 (RCC_MP_APRSTSR)

RCC application reset status register

偏移地址：0x10C

复位值：0x0000 0000

该寄存器提供 RDCTL 的状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RSTTOV[6:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	r	r	r	r	r	r	r								

位 31:15 保留，必须保持复位值。

位 14:8 **RSTTOV[6:0]**：复位超时延迟值 (Reset timeout delay value)
由硬件置 1，以提供超时计数器的值。仅当 RDCTLEN =“1”时，该功能才可用。
如果 RSTTOS =“0”，表示发生了超时。

位 7:0 保留，必须保持复位值。

10.8.6 RCC 低功耗 Stop 模式延迟控制寄存器 (RCC_PWRLPDLYCR)

RCC low-power Stop modes delay control register

偏移地址：0x110

复位值：0x0000 0004

如果 RCC_SECCFGR.STPSEC=1，则该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于编程从系统退出一种 Stop 模式到允许使能 PLL 并向总线桥和处理器提供时钟所间隔的延迟。有关详细信息，请参见*时钟恢复序列描述*一节。

注： *复位值为 4，是从任意一种 Stop 模式正常唤醒的最小值。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PWRLP_DLY[21:16]					
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRLP_DLY[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:0 **PWRLP_DLY[21:0]**：PWRLP_TEMPO 值

由软件写入。

该寄存器包含在一种系统 Stop 模式之后，在激活 PLL 和互连时钟之前必须遵守的延迟值。

延迟通过 HSI 时钟计算。用户必须考虑 HSIDIV 值。

延迟值由以下公式计算得出：PWRLP_DLY[21:0] / FHSI

0x0：保留

0x1：保留

0x2：保留

0x3：保留

0x4：取 5 个 HSI 周期 (64 Mhz) 的 PWRLP_TEMPO 延迟

0x5：取 6 个 HSI 周期 (64 MHz)的 PWRLP_TEMPO 延迟

...

0x3FFFFF：取约 65.5 毫秒的 PWRLP_TEMPO 延迟

10.8.7 RCC 全局复位控制设置寄存器 (RCC_MP_GRSTCSETR)

RCC global reset control set register

偏移地址：0x114

复位值：0x0000 0000

如果 RCC_SECCFGR.RSTSEC=1，则该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于生成系统复位或 MPU 处理器复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将激活复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPSYSRST
															rs

位 31:1 保留，必须保持复位值。

位 0 **MPSYSRST**：系统复位

由软件置 1，由硬件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将产生系统复位，参见*图 48*。

10.8.8 RCC Boot ROM 复位状态清除寄存器 (RCC_BR_RSTSCLRR)

RCC boot ROM reset status clear register

偏移地址：0x118

复位值：0x0000 0015

如果 RCC_SECCFGR.RSTSEC=1，则该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器由 Boot ROM 用来检查复位源。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位清零。要确定复位源，MPU 应用必须使用 *RCC 复位状态设置寄存器 (RCC_MP_RSTSSETR)*。有关详细信息，请参见*第 10.3.11 节：复位源标识*。

该寄存器在 V_{DD} 域中，通过 por_rst 复位信号来复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	IWDG2RSTF	IWDG1RSTF	Res.	MPSYSRSTF	VCPURSTF	VCORERSTF	HCSSRSTF	PADRSTF	BORRSTF	PORRSTF
						rc_w1	rc_w1		rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:10 保留，必须保持复位值。

位 9 **IWDG2RSTF**：IWDG2 复位标志 (IWDG2 reset flag)
由软件清零，发生 IWDG2 复位时由硬件置 1。
0：写入“0”无影响，读取“0”表示未发生 IWDG2 复位（por_rst 复位后的默认值）
1：写入“1”将清除 IWDG2RSTF 标志，读取“1”表示发生了 IWDG2 复位

位 8 **IWDG1RSTF**：IWDG1 复位标志 (IWDG1 reset flag)
由软件清零，发生 IWDG1 复位时由硬件置 1。
0：写入“0”无影响，读取“0”表示未发生 IWDG1 复位（por_rst 复位后的默认值）
1：写入“1”将清除 IWDG1RSTF 标志，读取“1”表示发生了 IWDG1 复位

位 7 保留，必须保持复位值。

位 6 **MPSYSRSTF**：MPU 系统复位标志 (MPU System reset flag)
由软件清零，发生 MPU 系统复位时由硬件置 1。
0：写入“0”无影响，读取“0”表示 MPU 未产生系统复位（por_rst 复位后的默认值）
1：写入“1”将清除 MPSYSRSTF 标志，读取“1”表示 MPU 产生了系统复位

位 5 **VCPURSTF**：V_{DDCPU} 复位标志 (V_{DDCPU} reset flag)
由软件清零，当因 V_{DDCPU} 低于预期值而发生复位时，由硬件置 1。
0：写入“0”无影响，读取“0”表示发生复位的起因并非 V_{DDCPU}
1：写入“1”将清除 VCPURSTF 标志，读取“1”表示 V_{DDCPU} 是发生复位的起因（por_rst 复位后的默认值）

位 4 **VCORERSTF**：V_{DDCORE} 复位标志 (V_{DDCORE} reset flag)
由软件清零，当因 V_{DDCORE} 低于预期值而发生复位时，由硬件置 1。
0：写入“0”无影响，读取“0”表示发生复位的起因并不是 V_{DDCORE}
1：写入“1”将清除 VCORERSTF 标志，读取“1”表示 V_{DDCORE} 是发生复位的起因（por_rst 复位后的默认值）

位 3 **HCSSRSTF**：HSE CSS 复位标志 (HSE CSS reset flag)

由软件清零，在 HSE 上检测到故障时由硬件置 1。

0：写入“0”无影响，读取“0”表示未发生 HSE CSS 复位（por_rst 复位后的默认值）

1：写入“1”将清除 HCSSRSTF 标志，读取“1”表示发生了 HSE CSS 复位

位 2 **PADRSTF**：NRST 复位标志 (NRST reset flag)

由软件清零，发生引脚复位时由硬件置 1。

0：写入“0”无影响，读取“0”表示未发生引脚复位（por_rst 复位后的默认值）

1：写入“1”将清除 PADRSTF 标志，读取“1”表示发生了引脚复位

位 1 **BORRSTF**：BOR 复位标志 (BOR reset flag)

由软件清零，发生 BOR 复位时由硬件置 1。

0：写入“0”无影响，读取“0”表示未发生 BOR 复位（por_rst 复位后的默认值）

1：写入“1”将清除 BORRSTF 标志，读取“1”表示发生了 BOR 复位

位 0 **PORRSTF**：POR/PDR 复位标志 (POR/PDR reset flag)

由软件清零，发生 POR/PDR 复位时由硬件置 1。

0：写入“0”无影响，读取“0”表示未发生 POR/PDR 复位

1：写入“1”将清除 PORRSTF 标志，读取“1”表示发生了 POR/PDR 复位（por_rst 复位后的默认值）

10.8.9 RCC 复位状态设置寄存器 (RCC_MP_RSTSSETR)

RCC reset status set register

偏移地址：0x11C

复位值：0x0000 0000

如果 RCC_SECCFGR.RSTSEC=1，则该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器专供 Boot ROM 代码用来更新复位源。该寄存器在上电复位 (por_rst)、退出待机模式、应用复位或退出 LPLV-Stop2 模式后将由 Boot ROM 代码更新。

写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

有关详细信息，请参见*第 10.3.11 节：复位源标识*。

该寄存器在 V_{DDCORE} 中

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPARE	Res.	Res.	CSTDBYRSTF	STDBYRSTF	STP2RSTF	IWDG2RSTF	IWDG1RSTF	Res.	MPSYSRSTF	VCPURSTF	VCORERSTF	HCSSRSTF	PADRSTF	BORRSTF	PORRSTF
rs			rs	rs	rs	rs	rs		rs	rs	rs	rs	rs	rs	rs

位 31:16 保留，必须保持复位值。

位 15 **SPARE**：备用位
由软件置 1 和复位。保留供将来使用。

位 14:13 保留，必须保持复位值。

位 12 **CSTDBYRSTF**：MPU CStandby 复位标志 (MPU CStandby reset flag)
当 MPU 退出 CStandby 模式时由 Boot ROM 代码置 1。
0：MPU 未处于 CStandby 模式
1：MPU 已处于 CStandby 模式

位 11 **STDBYRSTF**：系统待机复位标志 (System standby reset flag)
当退出待机模式时由 Boot ROM 代码置 1。
0：系统未处于待机模式
1：系统已处于待机模式

位 10 **STP2RSTF**：Stop2 复位标志 (Stop2 reset flag)
退出 LPLV-Stop2 时由 Boot ROM 代码置 1。
0：系统未处于 LPLV-Stop2 模式
1：系统已处于 LPLV-Stop2 模式

位 9 **IWDG2RSTF**：IWDG2 复位标志 (IWDG2 reset flag)
发生 IWDG2 复位时由 Boot ROM 代码置 1。
0：未发生 IWDG2 复位
1：发生了 IWDG2 复位

位 8 **IWDG1RSTF**：IWDG1 复位标志 (IWDG1 reset flag)
发生 IWDG1 复位时由 Boot ROM 代码置 1。
0：未发生 IWDG1 复位
1：发生了 IWDG1 复位

位 7 保留，必须保持复位值。

位 6 **MPSYSRSTF**：MPU 系统复位标志 (MPU System reset flag)
发生 MPU 系统复位时由 Boot ROM 代码置 1。
0：未发生由 MPU 产生的系统复位
1：发生了由 MPU 产生的系统复位

位 5 **VCPURSTF**：V_{CPU} 复位标志 (V_{CPU} reset flag)
当因 V_{CPU} 低于预期值而发生复位时由 Boot ROM 代码置 1。
0：V_{CPU} 不是复位的起因
1：V_{CPU} 是复位的起因

位 4 **VCORERSTF**：V_{DDCORE} 复位标志 (V_{DDCORE} reset flag)
当因 V_{DDCORE} 低于预期值而发生复位时由 Boot ROM 代码置 1。
0：V_{DDCORE} 不是复位的起因
1：V_{DDCORE} 是复位的起因

位 3 **HCSSRSTF**：HSE CSS 复位标志 (HSE CSS reset flag)
在 HSE 上检测到故障时由 Boot ROM 代码置 1。
0：未发生 HSE CSS 复位
1：发生了 HSE CSS 复位

位 2 **PADRSTF**：NRST 复位标志 (NRST reset flag)
当发生引脚复位时由 Boot ROM 代码置 1。
0：未发生引脚复位
1：发生了引脚复位

位 1 **BORRSTF**：BOR 复位标志 (BOR reset flag)
发生 BOR 复位时由 Boot ROM 代码置 1。
0：未发生 BOR 复位
1：发生了 BOR 复位

位 0 **PORRSTF**：POR/PDR 复位标志 (POR/PDR reset flag)
发生 POR/PDR 复位时由 Boot ROM 代码置 1。
0：未发生 POR/PDR 复位
1：发生了 POR/PDR 复位

10.8.10 RCC 复位状态清除寄存器 (RCC_MP_RSTSCLRR)

RCC reset status clear register

偏移地址：0x120

复位值：0x0000 0000

如果 RCC_SECCFGR.RSTSEC=1，则该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于检查复位源。该寄存器在上电复位 (por_rst)、退出待机模式、应用复位或退出 LPLV-Stop2 模式后将由 Boot ROM 代码更新。

写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位清零。

有关详细信息，请参见*第 10.3.11 节：复位源标识*。

该寄存器在 V_{DDCORE} 中

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPARE	Res.	Res.	CSTDBYRSTF	STDBYRSTF	STP2RSTF	IWDG2RSTF	IWDG1RSTF	Res.	MPSYSRSTF	VCPURSTF	VCORERSTF	HCSSRSTF	PADRSTF	BORRSTF	PORRSTF
rc_w1			rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:16 保留，必须保持复位值。

位 15 **SPARE**：备用位
由软件置 1 和复位。保留供将来使用。

位 14:13 保留，必须保持复位值。

位 12 **CSTDBYRSTF**：MPU CStandby 复位标志 (MPU CStandby reset flag)
由软件清零，当 MPU 退出 CStandby 模式时由 Boot ROM 代码置 1。
0：MPU 未处在 Cstandby 模式
1：MPU 已处在 CStandby 模式

位 11 **STDBYRSTF**：系统待机复位标志 (System standby reset flag)
由软件清零，退出待机模式时由 Boot ROM 代码置 1。
0：系统未处于待机模式
1：系统已处于待机模式

位 10 **STP2RSTF**：Stop2 复位标志 (Stop2 reset flag)
由软件清零，退出 LPLV-Stop2 模式时，由 Boot ROM 代码置 1。
0：系统未处于 LPLV-Stop2 模式
1：系统已处于 LPLV-Stop2 模式

位 9 **IWDG2RSTF**：IWDG2 复位标志 (IWDG2 reset flag)
由软件清零，发生 IWDG2 复位时，由 Boot ROM 代码置 1。
0：未发生 IWDG2 复位
1：发生了 IWDG2 复位

位 8 **IWDG1RSTF**：IWDG1 复位标志 (IWDG1 reset flag)
由软件清零，发生 IWDG1 复位时由 Boot ROM 代码置 1。
0：未发生 IWDG1 复位
1：发生了 IWDG1 复位

位 7 保留，必须保持复位值。

位 6 **MPSYSRSTF**：MPU 系统复位标志 (MPU System reset flag)
由软件清零，发生 MPU 系统复位时由 Boot ROM 代码置 1。
0：未发生由 MPU 产生的系统复位
1：发生了由 MPU 产生的系统复位

位 5 **VCPURSTF**：V_{DDCPU} 复位标志 (V_{DDCPU} reset flag)
由软件清零，当因 V_{DDCPU} 低于预期值而发生复位时由 Boot ROM 代码置 1。
0：V_{DDCPU} 不是复位的起因
1：V_{DDCPU} 是复位的起因

位 4 **VCORERSTF**：V_{DDCORE} 复位标志 (V_{DDCORE} reset flag)
由软件清零，当因 V_{DDCORE} 低于预期值而发生复位时由 Boot ROM 代码置 1。
0：V_{DDCORE} 不是复位的起因
1：V_{DDCORE} 是复位的起因

位 3 **HCSSRSTF**：HSE CSS 复位标志 (HSE CSS reset flag)
由软件清零，当在 HSE 上检测到故障时由 Boot ROM 代码置 1。
0：未发生 HSE CSS 复位
1：发生了 HSE CSS 复位

位 2 **PADRSTF**：NRST 复位标志 (NRST reset flag)
由软件清零，发生引脚复位时由 Boot ROM 代码置 1。
0：未发生引脚复位
1：发生了引脚复位

位 1 **BORRSTF**：BOR 复位标志 (BOR reset flag)
由软件清零，发生 BOR 复位时由 Boot ROM 代码置 1。
0：未发生 BOR 复位
1：发生了 BOR 复位

位 0 **PORRSTF**：POR/PDR 复位标志 (POR/PDR reset flag)
由软件清零，发生 POR/PDR 复位时由 Boot ROM 代码置 1。
0：未发生 POR/PDR 复位
1：发生了 POR/PDR 复位

10.8.11 RCC IWDG 时钟冻结设置寄存器 (RCC_MP_IWDGFZSETR)

RCC IWDG clock freeze set register

偏移地址：0x124

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器由 Boot ROM 用来冻结 IWDG 时钟。在 MPU 复位（因系统复位或退出 LPLV-Stop2 引起）之后，可对 MPU 执行一次写入。写入“0”无影响，而读取将返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FZ_IWDG2	FZ_IWDG1
														rs	rs

位 31:2 保留，必须保持复位值。

位 1 **FZ_IWDG2**：冻结 IWDG2 时钟 (Freeze the IWDG2 clock)

在 MPU 复位（因系统复位或退出 LPLV-Stop2 引起），由 Boot ROM 软件单次设置以冻结 IWDG2。

0：写入“0”无影响，读取“0”表示 IWDG2 时钟未冻结（系统复位后的默认值）

1：写入“1”将冻结 IWDG2 时钟，读取“1”表示 IWDG2 时钟已冻结

位 0 **FZ_IWDG1**：冻结 IWDG1 时钟 (Freeze the IWDG1 clock)

如果 IWDG1 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

在 MPU 复位（因系统复位或退出 LPLV-Stop2 引起），由 Boot ROM 软件单次设置以冻结 IWDG1。

0：写入“0”无影响，读取“0”表示 IWDG1 时钟未冻结（系统复位后的默认值）

1：写入“1”将冻结 IWDG1 时钟，读取“1”表示 IWDG1 时钟已冻结

10.8.12 RCC IWDG 时钟冻结清除寄存器 (RCC_MP_IWDGFZCLRR)

RCC IWDG clock freeze clear register

偏移地址：0x128

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器由 Boot ROM 用来解冻 IWDG 时钟。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FZ_IWDG2	FZ_IWDG1
														rc_w1	rc_w1

位 31:2 保留，必须保持复位值。

位 1 **FZ_IWDG2**：解冻 IWDG2 时钟 (Unfreeze the IWDG2 clock)

由 Boot ROM 软件清零以解冻 IWDG2。

0：写入“0”无影响，读取“0”表示 IWDG2 时钟未冻结（复位后的默认值）

1：写入“1”将解冻 IWDG2 时钟，读取“1”表示 IWDG2 时钟已冻结

位 0 **FZ_IWDG1**：解冻 IWDG1 时钟 (Unfreeze the IWDG1 clock)

如果 IWDG1 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由 Boot ROM 软件清零以解冻 IWDG1。

0：写入“0”无影响，读取“0”表示 IWDG1 时钟未冻结（复位后的默认值）

1：写入“1”将解冻 IWDG1 时钟，读取“1”表示 IWDG1 时钟已冻结

10.8.13 RCC 时钟源中断使能寄存器 (RCC_MP_CIER)

RCC clock source interrupt enable register

偏移地址：0x200

复位值：0x0000 0000

该寄存器可以单独、安全地在位字段级受到写保护，具体取决于 RCC_SECCFGR 寄存器定义的相应 RCC 子功能的安全状态。

必须使用该寄存器来控制中断源使能。更多详细信息，请参见*第 10.6 节：RCC 中断*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WKUPIE	Res.	Res.	Res.	LSECSSIE
											rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PLL4DYIE	PLL3DYIE	PLL2DYIE	PLL1DYIE	Res.	Res.	Res.	CSIRDYIE	HSERDYIE	HSIRDYIE	LSERDYIE	LSIRDYIE
				rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值。

位 20 **WKUPIE**：从 CStop 模式唤醒中断使能 (Wake up from CStop Interrupt Enable)

如果 RCC_SECCFGR.STPSEC=1，则该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和复位，以使能/禁止用于将 MPU 从 CStop 唤醒的中断的产生。

0：禁止唤醒中断（复位后的默认值）

1：使能唤醒中断

位 19:17 保留，必须保持复位值。

位 16 **LSECSSIE**：LSE 时钟安全系统中断使能 (LSE clock security system Interrupt Enable)

如果 RCC_SECCFGR.LSESEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和复位，用于使能/禁止由外部 32 kHz 振荡器上的时钟安全系统引起的中断。

0：禁止 LSE CSS 中断（复位后的默认值）

1：使能 LSE CSS 中断

位 15:12 保留，必须保持复位值。

位 11 **PLL4DYIE**：PLL4 就绪中断使能 (PLL4 ready Interrupt Enable)

如果 RCC_SECCFGR.PLL4SEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和复位，用于使能/禁止由 PLL4 锁定引起的中断。

0：禁止 PLL4 锁定中断（复位后的默认值）

1：使能 PLL4 锁定中断

位 10 **PLL3DYIE**：PLL3 就绪中断使能 (PLL3 ready Interrupt Enable)

如果 RCC_SECCFGR.PLL3SEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和复位，用于使能/禁止由 PLL3 锁定引起的中断。

0：禁止 PLL3 锁定中断（复位后的默认值）

1：使能 PLL3 锁定中断

位 9 **PLL2DYIE**：PLL2 就绪中断使能 (PLL2 ready Interrupt Enable)

如果 RCC_SECCFGR.PLL12SEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和复位，用于使能/禁止由 PLL2 锁定引起的中断。

0：禁止 PLL2 锁定中断（复位后的默认值）

1：使能 PLL2 锁定中断

位 8 **PLL1DYIE**：PLL1 就绪中断使能 (PLL1 ready Interrupt Enable)

如果 RCC_SECCFGR.PLL12SEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和复位，用于使能/禁止由 PLL1 锁定引起的中断。

0：禁止 PLL1 锁定中断（复位后的默认值）

1：使能 PLL1 锁定中断

位 7:5 保留，必须保持复位值。

位 4 **CSIRDYIE**：CSI 就绪中断使能 (CSI ready Interrupt Enable)

如果 RCC_SECCFGR.CSISEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和复位，用于使能/禁止由 CSI 振荡器稳定所引起的中断。

0：禁止 CSI 就绪中断（复位后的默认值）

1：使能 CSI 就绪中断

位 3 **HSERDYIE**：HSE 就绪中断使能 (HSE ready Interrupt Enable)

如果 RCC_SECCFGR.HSESEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和复位，用于使能/禁止由 HSE 振荡器稳定所引起的中断。

0：禁止 HSE 就绪中断（复位后的默认值）

1：使能 HSE 就绪中断

位 2 **HSIRDYIE**：HSI 就绪中断使能 (HSI ready Interrupt Enable)

如果 RCC_SECCFGR.HSISEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和复位，用于使能/禁止由 HSI 振荡器稳定所引起的中断。

0：禁止 HSI 就绪中断（复位后的默认值）

1：使能 HSI 就绪中断

位 1 **LSERDYIE**：LSE 就绪中断使能 (LSE ready Interrupt Enable)

如果 RCC_SECCFGR.LSESEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和复位，用于使能/禁止由 LSE 振荡器稳定所引起的中断。

0：禁止 LSE 就绪中断（复位后的默认值）

1：使能 LSE 就绪中断

位 0 **LSIRDYIE**：LSI 就绪中断使能 (LSI ready Interrupt Enable)

如果 RCC_SECCFGR.LSISEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和复位，用于使能/禁止由 LSI 振荡器稳定所引起的中断。

0：禁止 LSI 就绪中断（复位后的默认值）

1：使能 LSI 就绪中断

10.8.14 RCC 时钟源中断标志寄存器 (RCC_MP_CIFR)

RCC clock source interrupt flag register

偏移地址：0x204

复位值：0x0000 0000

该寄存器可以单独、安全地在位字段级受到写保护，具体取决于 RCC_SECCFGR 寄存器定义的相应 RCC 子功能的安全状态。

必须使用该寄存器来读取和清除中断标志。写入“0”无影响，写入“1”将清除相应的标志（前提是写访问配置为安全写访问）。更多详细信息，请参见*第 10.6 节：RCC 中断*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WKUPF	Res.	Res.	Res.	LSECSSF
											rc_w1				rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PLL4DYF	PLL3DYF	PLL2DYF	PLL1DYF	Res.	Res.	Res.	CSIRDYF	HSERDYF	HSIRDYF	LSERDYF	LSIRDYF
				rc_w1	rc_w1	rc_w1	rc_w1				rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:21 保留，必须保持复位值。

位 20 **WKUPF**：从 CStop 唤醒中断标志 (Wake up from CStop Interrupt Flag)

如果 RCC_SECCFGR.STPSEC=1，则该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件通过向该标志写入“1”来复位。

当 MPU 需要退出 CStop 时由硬件置 1。

0：无挂起的唤醒中断（复位后的默认值）

1：唤醒中断挂起，写入“1”将清除此标志

位 19:17 保留，必须保持复位值。

位 16 **LSECSSF**：LSE 时钟安全系统中断标志 (LSE clock security system Interrupt Flag)

如果 RCC_SECCFGR.LSESEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件通过向该标志写入“1”来复位。

当在外部 32 kHz 振荡器上检测到故障时由硬件置 1。

0：未在外部 32 kHz 振荡器上检测到故障（复位后的默认值）

1：在外部 32 kHz 振荡器上检测到故障，写入“1”将清除此标志

位 15:12 保留，必须保持复位值。

位 11 **PLL4DYF**：PLL4 就绪中断标志 (PLL4 ready Interrupt Flag)

如果 RCC_SECCFGR.PLLSEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件通过向该标志写入“1”来复位。

当 PLL4 锁定时由硬件置 1。

0：未因 PLL4 锁定而引起时钟就绪中断（复位后的默认值）

1：因 PLL4 锁定而引起时钟就绪中断，写入“1”将清除此标志

位 10 **PLL3DYF**：PLL3 就绪中断标志 (PLL3 ready Interrupt Flag)

如果 RCC_SECCFGR.PLLSEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件通过向该标志写入“1”来复位。

当 PLL3 锁定时由硬件置 1。

0：未因 PLL3 锁定而引起时钟就绪中断（复位后的默认值）

1：因 PLL3 锁定而引起时钟就绪中断，写入“1”将清除此标志

位 9 **PLL2DYF**：PLL2 就绪中断标志 (PLL2 ready Interrupt Flag)

如果 RCC_SECCFGR.PLLSEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件通过向该标志写入“1”来复位。

当 PLL2 锁定时由硬件置 1。

0：未因 PLL2 锁定而引起时钟就绪中断（复位后的默认值）

1：因 PLL2 锁定而引起时钟就绪中断，写入“1”将清除此标志

位 8 **PLL1DYF**：PLL1 就绪中断标志 (PLL1 ready Interrupt Flag)

如果 RCC_SECCFGR.PLLSEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件通过向该标志写入“1”来复位。

当 PLL1 锁定时由硬件置 1。

0：未因 PLL1 锁定而引起时钟就绪中断（复位后的默认值）

1：因 PLL1 锁定而引起时钟就绪中断，写入“1”将清除此标志

位 7:5 保留，必须保持复位值。

位 4 **CSIRDYF**：CSI 就绪中断标志 (CSI ready Interrupt Flag)

如果 RCC_SECCFGR.PLLSEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件通过向该标志写入“1”来复位。

当 CSI 时钟稳定时由硬件置 1。

0：未因 CSI 而引起时钟就绪中断（复位后的默认值）

1：因 CSI 而引起时钟就绪中断，写入“1”将清除此标志

位 3 **HSERDYF**：HSE 就绪中断标志 (HSE ready Interrupt Flag)

如果 RCC_SECCFGR.PLLSEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件通过向该标志写入“1”来复位。

当 HSE 时钟稳定时由硬件置 1。

0：未因 HSE 而引起时钟就绪中断（复位后的默认值）

1：因 HSE 而引起时钟就绪中断，写入“1”将清除此标志

位 2 **HSIRDYF**：HSI 就绪中断标志 (HSI ready Interrupt Flag)

如果 RCC_SECCFGR.PLLSEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件通过向该标志写入“1”来复位。

当 HSI 时钟稳定时由硬件置 1。

0：未因 HSI 而引起时钟就绪中断（复位后的默认值）

1：因 HSI 而引起时钟就绪中断，写入“1”将清除此标志

位 1 **LSERDYF**：LSE 就绪中断标志 (LSE ready Interrupt Flag)

如果 RCC_SECCFGR.PLLSEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件通过向该标志写入“1”来复位。

当 LSE 时钟稳定时由硬件置 1。

0：未因 LSE 而引起时钟就绪中断（复位后的默认值）

1：因 LSE 而引起时钟就绪中断，写入“1”将清除此标志

位 0 **LSIRDYF**：LSI 就绪中断标志 (LSI ready Interrupt Flag)

如果 RCC_SECCFGR.PLLSEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件通过向该标志写入“1”来复位。

当 LSI 时钟稳定时由硬件置 1。

0：未因 LSI 而引起时钟就绪中断（复位后的默认值）

1：因 LSI 而引起时钟就绪中断，写入“1”将清除此标志

10.8.15 RCC 备份域控制寄存器 (RCC_BDCR)

RCC backup domain control register

偏移地址：0x400

复位值：0x0000 0020

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该 V_{SW} 域寄存器通过备份域上电复位来复位（即 VDD 和 VBAT 已掉电）。当安全写入该寄存器并设置 RCC_BDCR.VSWRST=1 时，该寄存器也会复位（VSWRST 位字段除外）。

系统复位后，默认情况下，备份域（包括此 RCC_BDCR 寄存器）将受到写保护以防止任何写访问。在写入该寄存器之前，软件必须先禁止备份域写保护：写入并设置 PWR_CR1.DBP=1。

该寄存器用于控制备份域、RTC 和 LSE 功能。连续对该寄存器进行写访问时，将插入等待周期。等待周期的数量可多达 7 个 AHB4 时钟周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VSWRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTCCKEN	Res.	Res.	RTCSRC[1:0]	
rw											rw			rwo	rwo

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	LSECSSD	LSECSSON	Res.	Res.	LSEDRV[1:0]		DIGBYP	LSERDY	LSEBYP	LSEON
						r	rw			rw	rw	rw	r	rw	rw

位 31 **VSWRST**：V_{SW} 域软件复位 (V_{SW} domain software reset)

由软件置 1 和复位。

如果 RCC_SECCFGR.RSTSEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

0：写：释放软件复位；读：软件复位未激活（备份域复位后的默认值）

1：写：复位整个 V_{SW} 域；读：软件复位激活

位 30:21 保留，必须保持复位值。

位 20 **RTCCKEN**：RTC 时钟使能 (RTC clock enable)

由软件置 1 和复位。

如果 RTC 的至少一个功能通过 RTC_SECCFGR 寄存器配置为安全（即如果由 RTC 驱动的输入信号 rtc_sec 被置为有效），则对该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

0：禁止 rtc_ck 时钟（备份域复位后的默认值）

1：使能 rtc_ck 时钟

位 19:18 保留，必须保持复位值。

位 17:16 **RTCSRC[1:0]**：RTC 时钟源选择 (RTC clock source selection)

由软件置 1，用于选择 RTC 的时钟源。

如果 RTC 的至少一个功能通过 RTC_SECCFGR 寄存器配置为安全（即如果由 RTC 驱动的输入信号 rtc_sec 被置为有效），则对该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该位字段在备份域复位后只能写入一次。

该位字段必须在使能 LSECSSON 之前写入。

有关详细信息，请参见 *LSE 上的 CSS* 一节。

0x0：无时钟（备份域复位后的默认值）

0x1：LSE 时钟用作 RTC 时钟

0x2：LSI 时钟用作 RTC 时钟

0x3：将以 RTCDIV 值进行分频的 HSE 时钟用作 RTC 时钟

位 15:10 保留，必须保持复位值。

位 9 **LSECSSD**：LSE 时钟安全系统故障检测 (LSE clock security system failure detection)

由硬件置 1，用于指示外部 32 kHz 振荡器上的时钟安全系统检测到故障。

0：未在 32 kHz 振荡器上检测到故障（备份域复位后的默认值）

1：在 32 kHz 振荡器上检测到故障

位 8 **LSECSSON**：LSE 时钟安全系统使能 (LSE clock security system enable)

由软件置 1，用于使能 32 kHz 振荡器上的时钟安全系统。

如果 RCC_SECCFGR.LSESEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

有关激活和取消激活序列的详细信息，请参见 *LSE 上的 CSS* 一节。

0：32 kHz 振荡器上的时钟安全系统关闭（备份域复位后的默认值）

1：32 kHz 振荡器上的时钟安全系统开启

位 7:6 保留，必须保持复位值。

位 5:4 **LSEDRV[1:0]**：LSE 振荡器驱动能力 (LSE oscillator driving capability)

由软件写入，以选择 LSE 振荡器的驱动能力。

如果 RCC_SECCFGR.LSESEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

0x0：最低驱动

0x1：中低驱动

0x2：中高驱动（备份域复位后的默认值）

0x3：最高驱动

位 3 **DIGBYP**：LSE 数字旁路 (LSE digital bypass)

由软件置 1 和复位，以选择模拟或数字旁路模式。

如果 RCC_SECCFGR.LSESEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

0：LSE 处于模拟旁路模式（备份域复位后的默认值）

1：LSE 处于数字旁路模式

位 2 **LSERDY**：LSE 振荡器就绪 (LSE oscillator ready)

由硬件置 1 和复位，用于指示 LSE 已稳定。将 LSEON 设为“0”后，该位需要经过 6 个 lse_ck 时钟周期才能变为低电平。

0：LSE 振荡器未就绪（备份域复位后的默认值）

1：LSE 振荡器就绪 (LSE oscillator ready)

位 1 **LSEBYP**：LSE 振荡器旁路 (LSE oscillator bypass)

由软件置 1 和复位，用于在调试模式下旁路振荡器。

如果 RCC_SECCFGR.LSESEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

在 LSE 使能（LSEON = “1”）时，不得写入该位。

有关详细信息，请参见*另请注意，当系统进入 Stop 模式时，信号 csi_cal_ck 会被门控。*

0：不旁路 LSE 振荡器（备份域复位后的默认值）

1：旁路 LSE 振荡器

位 0 **LSEON**：LSE 振荡器使能 (LSE oscillator enabled)

由软件置 1 和复位。

如果 RCC_SECCFGR.LSESEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

0：LSE 振荡器关闭（备份域复位后的默认值）

1：LSE 振荡器开启

10.8.16 RCC 复位持续时间和 LSI 控制寄存器 (RCC_RDLSICR)

RCC reset duration and LSI control register

偏移地址：0x404

复位值：0x0000 0000

该寄存器可以单独、安全地在位字段级受到写保护，具体取决于 RCC_SECCFGR 寄存器定义的相应 RCC 子功能的安全状态。

该寄存器通过 por_rst 复位进行复位，位于 V_{DD} 域中。

该寄存器用于控制 NRST 激活的最小持续时间和 LSI 功能。

将为字、半字和字节访问插入 0 到 7 个等待周期。如果连续访问该寄存器，则插入等待周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPARE[4:0]					EADLY[2:0]			Res.	Res.	Res.	MRD[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSIRDY	LSION
														r	rw

位 31:27 **SPARE[4:0]**：备用位

由软件写入。

如果 RCC_SECCFGR.RSTSEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

保留供将来使用。

位 26:24 **EADLY[2:0]**: 外部访问延迟 (External access delays)

由软件写入。

如果 RCC_SECCFGR.RSTSEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

在 Boot ROM 执行任何外部访问（UART、USB、QUADSPI、FMC、SDMMC…）之前的等待时间

0x0：10 ms（复位后的默认值）

0x1：Boot ROM 不增加额外延迟

0x2：100 µs

0x3：200 µs

0x4：500 µs

0x5：1 ms

0x6：2 ms

0x7：5 ms

位 23:21 保留，必须保持复位值。

位 20:16 **MRD[4:0]**：最短复位持续时间 (Minimum reset duration)

由软件写入。

如果 RCC_SECCFGR.RSTSEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该位字段定义 NRST 低电平脉冲的最短持续时间。LSI 振荡器会在 RPCTL 需要时自动使能。

0x0：由引脚的脉冲延展器来保证 NRST 低电平脉冲持续时间。RPCTL 被旁路（复位后的默认值）

0x1：保证的 NRST 低电平脉冲持续时间约为 1 ms（1 x 32 个 lsi_ck 周期），

0x2：保证的 NRST 低电平脉冲持续时间约为 2 ms（2 x 32 个 lsi_ck 周期），

...

0x1F：保证的 NRST 低电平脉冲持续时间约为 31 ms（31 x 32 个 lsi_ck 周期）。

位 15:2 保留，必须保持复位值。

位 1 **LSIRDY**：LSI 振荡器就绪 (LSI oscillator ready)

由硬件置 1 和复位，用于指示 LSI 已稳定。将 LSION 设为“0”后，该位需要经过 3 个 lsi_ck 时钟周期才能变为低电平。

0：LSI 振荡器未就绪（复位后的默认值）

1：LSI 振荡器就绪 (LSI oscillator ready)

位 0 **LSION**：LSI 振荡器使能 (LSI oscillator enabled)

由软件置 1 和复位。

如果 RCC_SECCFGR.LSISEC=1，该位字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该位用于请求激活 LSI 振荡器。请注意，IWDG1 和 IWDG2 也可以请求激活 LSI。

0：如果没有其他请求方（例如 IWDG[2:1]）需要，LSI 振荡器将关闭（复位后的默认值）

1：LSI 振荡器开启

10.8.17 RCC 振荡器时钟使能设置寄存器 (RCC_OCENSETR)

RCC oscillator clock enable set register

偏移地址：0x420

复位值：0x0000 0001

如果 RCC 时钟的给定 xxx 资源配置为安全（即，如果给定的 xxx 资源配置为 RCC_SECCFGR.xxxSEC=1），则对属于该资源的寄存器字段进行的写访问必须是安全写访问。

读访问可以是安全或非安全访问。

该寄存器用于控制振荡器。向该寄存器写入“0”无影响，写入“1”会将相应位置 1。读取该寄存器可获得每个位的有效值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	HSECSSON	HSEBYP	HSEKERON	HSEON	DIGBYP	Res.	CSIKERON	CSION	Res.	Res.	HSIKERON	HSION
				rs	rs	rs	rs	rs		rs	rs			rs	rs

位 31:12 保留，必须保持复位值。

位 11 **HSECSSON**：将 HSECSSON 位置 1

设置 HSE 上的时钟安全系统。

如果 HSE 时钟配置为安全（即 RCC_SECCFGR.HSESEC=1），则必须由安全代理对该位进行写入。

该位“仅置 1”（由系统复位或在系统进入待机模式时禁止）。

当 HSECSSON 置 1 时，时钟检测器将在 HSE 就绪后由硬件使能。

0：读取“0”表示 HSE 上的时钟安全系统关闭（复位后的默认值）

1：写入“1”将使能 HSE 上的时钟安全系统，读取“1”表示 HSE 上的时钟安全系统已开启

位 10 **HSEBYP**：将 HSEBYP 位置 1

由软件置 1 以使能旁路模式。

如果 HSE 时钟配置为安全（即 RCC_SECCFGR.HSESEC=1），则必须由安全代理对该位进行写入。

0：无影响

1：将 HSEBYP 位置 1

位 9 **HSEKERON**：将 HSEKERON 位置 1

由软件置 1，即使在 (LP-)Stop 模式下也可强制 HSE 开启，从而可快速用作某些外设的内核时钟。

如果 HSE 时钟配置为安全（即 RCC_SECCFGR.HSESEC=1），则必须由安全代理对该位进行写入。

该位对 HSEON 的值没有任何影响。

0：无影响

1：将 HSEKERON 位置 1

位 8 **HSEON**：将 HSEON 位置 1

由软件置 1 以使能 HSE。

如果 HSE 时钟配置为安全（即 RCC_SECCFGR.HSESEC=1），则必须由安全代理对该位进行写入。

0：无影响

1：将 HSEON 位置 1

位 7 **DIGBYP**：将 DIGBYP 位置 1

当外部 HSE 时钟作为全摆幅数字信号连接到 OSC_IN 时由软件置 1。

如果 HSE 时钟配置为安全（即 RCC_SECCFGR.HSESEC=1），则必须由安全代理对该位进行写入。

0：无影响

1：将 DIGBYP 位置 1（数字旁路）

位 6 保留，必须保持复位值。

位 5 **CSIKERON**：将 CSIKERON 位置 1

由软件置 1，即使在 (LP-)Stop 模式下也可强制 CSI 开启，从而可快速用作某些外设的内核时钟。

如果 CSI 时钟配置为安全（即 RCC_SECCFGR.CSISEC=1），则必须由安全代理对该位进行写入。

该位对 CSION 的值没有任何影响。

0：无影响

1：将 CSIKERON 位置 1

位 4 **CSION**：将 CSION 位置 1

由软件置 1 以使能 CSI 时钟。

如果 CSI 时钟配置为安全（即 RCC_SECCFGR.CSISEC=1），则必须由安全代理对该位进行写入。

0：无影响

1：将 CSION 位置 1

位 3:2 保留，必须保持复位值。

位 1 **HSIKERON**：将 HSIKERON 位置 1

由软件置 1，即使在 (LP-)Stop 模式下也可强制 HSI 开启，从而可快速用作外设的内核时钟。

如果 HSI 时钟配置为安全（即 RCC_SECCFGR.HSISEC=1），则必须由安全代理对该位进行写入。

该位对 HSION 的值没有任何影响。

0：无影响

1：将 HSIKERON 位置 1

位 0 **HSION**：将 HSION 位置 1

由软件置 1 以使能 HSI 时钟。

如果 HSI 时钟配置为安全（即 RCC_SECCFGR.HSISEC=1），则必须由安全代理对该位进行写入。

当产品退出待机模式或一种 Stop 模式时，HSION 也会由硬件置 1 以强制 HSI 开启。

0：无影响

1：将 HSION 位置 1

10.8.18 RCC 振荡器时钟使能清零寄存器 (RCC_OCENCLRR)

RCC oscillator clock enable clear register

偏移地址：0x424

复位值：0x0000 0001

如果 RCC 时钟的给定 xxx 资源配置为安全（即，如果给定的 xxx 资源配置为 RCC_SECCFGR.xxxSEC=1），则对属于该资源的寄存器字段进行的写访问必须是安全写访问。

该寄存器用于控制振荡器。向该寄存器写入“0”无影响，写入“1”将清除相应的位。读取该寄存器会提供使能位的有效值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	HSEBYP	HSEKERON	HSEON	DIGBYP	Res.	CSIKERON	CSION	Res.	Res.	HSIKERON	HSION
					rc_w1	rc_w1	rc_w1	rc_w1		rc_w1	rc_w1			rc_w1	rc_w1

位 31:11 保留，必须保持复位值。

位 10 **HSEBYP**：将 HSEBYP 位清零
由软件清零以禁止旁路模式。
如果 HSE 时钟配置为安全（即 RCC_SECCFGR.HSESEC=1），则必须由安全代理对该位进行写入。
0：无影响
1：将 HSEBYP 位清零

位 9 **HSEKERON**：将 HSEKERON 位清零
由软件清零以允许 HSE 在一种 Stop 模式下关闭。
如果 HSE 时钟配置为安全（即 RCC_SECCFGR.HSESEC=1），则必须由安全代理对该位进行写入。
0：无影响
1：将 HSEKERON 位清零

位 8 **HSEON**：将 HSEON 位清零
由软件清零以禁止 HSE 时钟。
如果 HSE 时钟配置为安全（即 RCC_SECCFGR.HSESEC=1），则必须由安全代理对该位进行写入。
0：无影响
1：将 HSEON 位清零

位 7 **DIGBYP**：将 DIGBYP 位清零
当连接到 OSC_IN 的外部时钟为低摆幅信号时由软件清零。
如果 HSE 时钟配置为安全（即 RCC_SECCFGR.HSESEC=1），则必须由安全代理对该位进行写入。
0：无影响
1：将 DIGBYP 位清零（模拟旁路）

位 6 保留，必须保持复位值。

位 5 **CSIKERON**：将 CSIKERON 位清零

由软件清零以允许 CSI 在一种 Stop 模式下关闭。

如果 CSI 时钟配置为安全（即 RCC_SECCFGR.CSISEC=1），则必须由安全代理对该位进行写入。

该位对 CSION 的值没有任何影响。

0：无影响

1：将 CSIKERON 位清零

位 4 **CSION**：将 CSION 位清零

由软件清零以禁止 CSI 时钟。

如果 CSI 时钟配置为安全（即 RCC_SECCFGR.CSISEC=1），则必须由安全代理对该位进行写入。

0：CSI 关闭

1：将 CSION 位清零

位 3:2 保留，必须保持复位值。

位 1 **HSIKERON**：将 HSIKERON 位清零

由软件清零以允许 HSI 在一种 Stop 模式下关闭。

如果 HSI 时钟配置为安全（即 RCC_SECCFGR.HSISEC=1），则必须由安全代理对该位进行写入。

该位对 HSION 的值没有任何影响。

0：无影响

1：将 HSIKERON 位清零

位 0 **HSION**：将 HSION 位清零

由软件清零以禁止 HSI 时钟。

如果 HSI 时钟配置为安全（即 RCC_SECCFGR.HSISEC=1），则必须由安全代理对该位进行写入。

0：无影响

1：将 HSION 位清零

10.8.19 RCC 振荡器和 MPU 与 AXI 时钟就绪寄存器 (RCC_OCRDYR)

RCC oscillators and MPU and AXI clock ready register

偏移地址：0x428

复位值：0x0000 0000

该寄存器为只读访问寄存器，它包含振荡器的状态标志。写入无影响。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXICKRDY	MPUCKRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							r	r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSERDY	Res.	Res.	Res.	CSIRDY	Res.	HSIDIVRDY	Res.	HSIRDY
							r				r		r		r

位 31:25 保留，必须保持复位值。

位 24 **AXICKRDY**：AXI 子系统时钟就绪标志 (AXI sub-system clock ready flag)
由硬件置 1，用于指示 axiss_ck 时钟可用。
0：axiss_ck 时钟不可用（复位后的默认值）
1：axiss_ck 时钟可用

位 23 **MPUCKRDY**：MPU 时钟就绪标志 (MPU clock ready flag)
由硬件置 1，用于指示 mpuss_ck 时钟可用。
0：mpuss_ck 时钟不可用（复位后的默认值）
1：mpuss_ck 时钟可用

位 22:9 保留，必须保持复位值。

位 8 **HSERDY**：HSE 时钟就绪标志 (HSE clock ready flag)
由硬件置 1，用以指示 HSE 振荡器已稳定。
0：HSE 时钟未就绪（复位后的默认值）
1：HSE 时钟已就绪

位 7:5 保留，必须保持复位值。

位 4 **CSIRDY**：CSI 时钟就绪标志 (CSI clock ready flag)
由硬件置 1，用以指示 CSI 振荡器已稳定。
0：CSI 时钟未就绪（复位后的默认值）
1：CSI 时钟已就绪

位 3 保留，必须保持复位值。

位 2 **HSIDIVRDY**：HSI 分频器就绪标志 (HSI divider ready flag)

由硬件置 1 和复位。

由于对 HSIDIV 的写操作不会直接影响频率，因此该标志指示的是 HSI 分频器的当前状态。HSIDIVF 会在 HSIDIV 值发生变化时立即变为 0，并会在输出频率与编程到 HSIDIV 中的值匹配时重新置 1

0：新分频比尚未传播到 hsi_ck (hsi_ker_ck)（复位后的默认值）

1：hsi_ck (hsi_ker_ck) 时钟频率反映新 HSIDIV 值

位 1 保留，必须保持复位值。

位 0 **HSIRDY**：HSI 时钟就绪标志 (HSI clock ready flag)

由硬件置 1，用以指示 HSI 振荡器已稳定。

0：HSI 时钟未就绪（复位后的默认值）

1：HSI 时钟已就绪

10.8.20 RCC HSI 配置寄存器 (RCC_HSICFGR)

RCC HSI configuration register

偏移地址：0x440

复位值：0x0000 0000

如果 RCC_SECCFGR.HSISEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于配置 HSI。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	HSICAL[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	HSITRIM[6:0]							Res.	Res.	Res.	Res.	Res.	Res.	HSIDIV[1:0]	
	rw	rw	rw	rw	rw	rw	rw							rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **HSICAL[11:0]**：HSI 时钟校准 (HSI clock calibration)

由硬件通过系统复位期间的选项字节加载来置 1。
由软件通过微调位 HSITRIM 进行调节。
该位字段表示工程选项字节校准值与 HSITRIM 位值之和。

位 15 保留，必须保持复位值。

位 14:8 **HSITRIM[6:0]**：HSI 时钟微调 (HSI clock trimming)

由软件置 1，用于调整校准。HSITRIM 表示一个有符号值。
HSITRIM 位字段与复位阶段加载的工程选项字节 (bsec_hsi_cal[11:0]) 相加，以构成校准微调值。
HSICAL[11:0] = HSITRIM[6:0] + bsec_hsi_cal[11:0]。
0x40：bsec_hsi_cal[11:0] - 64
0x41：bsec_hsi_cal[11:0] - 63
...
0x0：bsec_hsi_cal[11:0]（复位后的默认值）
...
0x3E：bsec_hsi_cal[11:0] + 62
0x3F：bsec_hsi_cal[11:0] + 63

位 7:2 保留，必须保持复位值。

位 1:0 **HSIDIV[1:0]**：HSI 时钟分频器 (HSI clock divider)

由软件写入。
用户可使用这些位来选择一个分频比，以选择所需 HSI 时钟频率。*RCC 振荡器和 MPU 与 AXI 时钟就绪寄存器 (RCC_OCRDYR)* 中的 HSIDIVRDY 位指示需要考虑新的分频比。
0x0：1 分频，hsi_ck (hsi_ker_ck)= 64 MHz（复位后的默认值）
0x1：2 分频，hsi_ck (hsi_ker_ck)= 32 MHz
0x2：4 分频，hsi_ck (hsi_ker_ck) = 16 MHz
0x3：8 分频，hsi_ck (hsi_ker_ck) = 8 MHz

10.8.21 RCC CSI 配置寄存器 (RCC_CSICFGR)

RCC CSI configuration register

偏移地址：0x444

复位值：0x0000 1000

如果 RCC_SECCFGR.CSISEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于微调 CSI 频率。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSICAL[7:0]							
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CSITRIM[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
			rw	rw	rw	rw	rw								

位 31:24 保留，必须保持复位值。

位 23:16 **CSICAL[7:0]**：CSI 时钟校准 (CSI clock calibration)
由硬件通过系统复位期间的选项字节加载来置 1。
由软件通过微调位 CSITRIM 进行调节。
该位字段表示工程选项字节校准值与 CSITRIM 位值之和

位 15:13 保留，必须保持复位值。

位 12:8 **CSITRIM[4:0]**：CSI 时钟微调 (CSI clock trimming)
由软件置 1，用于调整校准。
CSITRIM 位字段与复位阶段加载的工程选项字节 (bsec_hsi_cal[7:0]) 相加，以构成校准微调值。
CSICAL[7:0] = CSITRIM[4:0] + bsec_csi_cal[7:0]。

位 7:0 保留，必须保持复位值。

10.8.22 RCC MCO1 配置寄存器 (RCC_MCO1CFGR)

RCC MCO1 configuration register

偏移地址：0x460

复位值：0x0000 0000

如果 RCC_SECCFGR.MCO1SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于选择在 MCO1 输出上生成的时钟。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	MCO1ON	Res.	Res.	Res.	Res.	MCO1DIV[3:0]				Res.	MCO1SEL[2:0]		
			rw					rw	rw	rw	rw		rw	rw	rw

位 31:13 保留，必须保持复位值。

位 12 **MCO1ON**：MCO1 输出的控制 (Control of the MCO1 output)
由软件置 1 和清零，用于使能 MCO1 输出。
0：禁止 MCO1 输出
1：使能 MCO1 输出

位 11:8 保留，必须保持复位值。

位 7:4 **MCO1DIV[3:0]**：MCO1 预分频器 (MCO1 prescaler)
由软件置 1 和清零，用于配置 MCO1 的预分频器。修改此预分频器可能会在 MCO1 上产生噪声。强烈建议仅在复位后且在使能外部振荡器和 PLL 之前更改此预分频器。
有关详细信息，请参见*第 10.4.4 节：时钟输出生成（MCO1 和 MCO2）*。
0x0：旁路（复位后的默认值）
0x1：2 分频
0x2：3 分频
...
0xF：16 分频

位 3 保留，必须保持复位值。

位 2:0 **MCO1SEL[2:0]**：MCO1 时钟输出选择 (MCO1 clock output selection)
由软件置 1 和清零。时钟源选择可能会在 MCO1 上产生噪声。
强烈建议仅在复位后且在使能外部振荡器和 PLL 之前来配置这些位。
0x0：选择 HSI 时钟 (hsi_ck)（复位后的默认值）
0x1：选择 HSE 时钟 (hse_ck)
0x2：选择 CSI 时钟 (csi_ck)
0x3：选择 LSI 时钟 (lsi_ck)
0x4：选择 LSE 振荡器时钟 (lse_ck)
其他值：保留

10.8.23 RCC MCO2 配置寄存器 (RCC_MCO2CFGR)

RCC MCO2 configuration register

偏移地址：0x464

复位值：0x0000 0000

如果 RCC_SECCFGR.MCO2SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于选择在 MCO2 输出上生成的时钟。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	MCO2ON	Res.	Res.	Res.	Res.	MCO2DIV[3:0]				Res.	MCO2SEL[2:0]		
			rw					rw	rw	rw	rw		rw	rw	rw

位 31:13 保留，必须保持复位值。

位 12 **MCO2ON**：MCO2 输出的控制 (Control of the MCO2 output)
由软件置 1 和清零，用于使能 MCO2 输出。
0：禁止 MCO2 输出
1：使能 MCO2 输出

位 11:8 保留，必须保持复位值。

位 7:4 **MCO2DIV[3:0]**：MCO2 预分频器 (MCO2 prescaler)
由软件置 1 和清零，用于配置 MCO2 的预分频器。修改此预分频器可能会在 MCO2 上产生噪声。强烈建议仅在复位后且在使能外部振荡器和 PLL 之前更改此预分频器。
有关详细信息，请参见第 1.4.4 节：时钟输出生成（MCO1 和 MCO2）。
0x0：旁路（复位后的默认值）
0x1：2 分频
0x2：3 分频
...
0xF：16 分频

位 3 保留，必须保持复位值。

位 2:0 **MCO2SEL[2:0]**：微控制器时钟输出 2 (Micro-controller clock output 2)
由软件置 1 和清零。时钟源选择可能会在 MCO2 上产生噪声。
强烈建议仅在复位后且在使能外部振荡器和 PLL 之前来配置这些位。
0x0：选择 MPU 时钟 (mpuss_ck)（复位后的默认值）
0x1：选择 AXI 时钟 (axiss_ck)
0x2：选择 MLAHB 时钟 (mlahbss_ck)
0x3：选择 PLL4 时钟 (pll4_p_ck)
0x4：选择 HSE 时钟 (hse_ck)
0x5：选择 HSI 时钟 (hsi_ck)
其他值：保留

10.8.24 RCC 调试配置寄存器 (RCC_DBGCFGR)

RCC debug configuration register

偏移地址：0x468

复位值：0x0000 0001

该寄存器包含调试和追踪功能的使能控制以及跟踪功能的时钟分频器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBGRST	Res.	Res.	TRACECKEN	DBGCKEN	Res.	Res.	Res.	Res.	Res.	TRACEDIV[2:0]		
			rw			rw	rw						rw	rw	rw

位 31:13 保留，必须保持复位值。

位 12 **DBGRST**：调试功能复位 (Reset of the debug function)
由软件置 1 和清零，用以控制跟踪和调试部分的复位生成 (dbg_rstn)。
0：跟踪和调试部分未复位。（复位后的默认值）
1：跟踪和调试部分正在复位。

位 11:10 保留，必须保持复位值。

位 9 **TRACECKEN**：跟踪功能的时钟使能 (Clock enable for trace function)
由软件置 1 和清零，用于控制 trace_ck 时钟的生成。
其他相关信息，请参见 *DBG 和跟踪的时钟分配*一节。
0：禁止跟踪功能的时钟（复位后的默认值）
1：使能跟踪功能的时钟

位 8 **DBGCKEN**：调试功能的时钟使能 (Clock enable for debug function)
由软件置 1 和清零，用于控制 ck_apb_dbg 时钟（调试 APB 时钟）、ck_sys_dbg 的生成。
其他相关信息，请参见 *DBG 和跟踪的时钟分配*一节。
0：调试功能的时钟的使能由来自 DAP 的 cdbgwrupreq 信号控制。（复位后的默认值）
1：使能调试功能的时钟

位 7:3 保留，必须保持复位值。

位 2:0 **TRACEDIV[2:0]**：跟踪时钟 (ck_trace) 的时钟分频比 (Clock divider for the trace clock)
由软件写入，用来控制 ck_trace 时钟的分频系数。
可以实时更改此分频比。
0x0：aclk
0x1：aclk/2（复位后的默认值）
0x2：aclk/4
0x3：aclk/8
其他值：aclk/16

10.8.25 RCC PLL1 和 PLL2 参考时钟选择寄存器 (RCC_RCK12SELR)

RCC PLL1 and PLL2 reference clock selection register

偏移地址：0x480

复位值：0x8000 0000

如果 RCC_SECCFGR.PLL12SEC=1，则该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于选择 PLL1 和 PLL2 的参考时钟源。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLL12SRCRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL12SRC[1:0]	
														rw	rw

位 31 **PLL12SRCRDY**：PLL12 源时钟切换状态 (PLL12 source clock switch status)

由硬件置 1 和复位，用来指示 PLL12 时钟切换跳变是否已正确执行。

0：PLL12 开关未就绪或处于“1x”位置：其输出上没有生成时钟

1：PLL12 开关已就绪：时钟切换正在选择由 PLL12SRC[1:0] 位字段指定的时钟。（复位后的默认值）

位 30:2 保留，必须保持复位值。

位 1:0 **PLL12SRC[1:0]**：PLL12 的源时钟选择 (Source clock selection for PLL12)

由软件写入，用来选择 PLL12 源时钟。

为实现节能，不使用 PLL12 时，必须将 PLL12SRC[1:0] 设为“1x”。

0x0：选择 HSI 作为 PLL12 时钟 (hsi_ck)（复位后的默认值）

0x1：选择 HSE 作为 PLL12 时钟 (hse_ck)

其他值：未向 DIVM1 分频器和 PLL12 发送任何时钟

10.8.26 RCC PLL3 参考时钟选择寄存器 (RCC_RCK3SELR)

RCC PLL3 reference clock selection register

偏移地址：0x484

复位值：0x8000 0000

如果 RCC_SECCFGR.PLL3SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于选择 PLL3 的参考时钟源。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLL3SRCRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL3SRC[1:0]	
														rw	rw

位 31 **PLL3SRCRDY**：PLL3 源时钟切换状态 (PLL3 source clock switch status)

由硬件置 1 和复位，用来指示 PLL3 时钟切换跳变是否已正确执行。

0：PLL3 切换未就绪或处于“11”位置：其输出上没有生成时钟

1：PLL3 切换已就绪：时钟切换正在选择由 PLL3SRC[1:0] 位字段指定的时钟。（复位后的默认值）

位 30:2 保留，必须保持复位值。

位 1:0 **PLL3SRC[1:0]**：PLL3 的源时钟选择 (Source clock selection for PLL3)

由软件写入，用来选择 PLL3 时钟源。

为实现节能，不使用 PLL3 时，必须将 PLL3SRC[1:0] 设为“11”。

0x0：选择 HSI 作为 PLL 时钟 (hsi_ck)（复位后的默认值）

0x1：选择 HSE 作为 PLL 时钟 (hse_ck)

0x2：选择 CSI 作为 PLL 时钟 (csi_ck)

0x3：未向 DIVM3 分频器和 PLL3 发送任何时钟

10.8.27 RCC PLL4 参考时钟选择寄存器 (RCC_RCK4SELR)

RCC PLL4 reference clock selection register

偏移地址：0x488

复位值：0x8000 0000

如果 RCC_SECCFGR.PLL4SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于选择 PLL4 的参考时钟源。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLL4SRCRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL4SRC[1:0]	
														rw	rw

位 31 **PLL4SRCRDY**：PLL4 源时钟切换状态 (PLL4 source clock switch status)

由硬件置 1 和复位，用来指示 PLL4 时钟切换跳变是否已正确执行。

0：PLL4 切换未就绪或处于“11”位置：其输出上没有生成时钟

1：PLL4 切换已就绪：时钟切换正在选择由 PLL4SRC[1:0] 位字段指定的时钟。（复位后的默认值）

位 30:2 保留，必须保持复位值。

位 1:0 **PLL4SRC[1:0]**：PLL4 的源时钟选择 (Source clock selection for PLL4)

由软件写入，用来选择 PLL4 时钟源。

0x0：选择 HSI 作为 PLL 时钟 (hsi_ck)（复位后的默认值）

0x1：选择 HSE 作为 PLL 时钟 (hse_ck)

0x2：选择 CSI 作为 PLL 时钟 (csi_ck)

0x3：信号 I2S_CKIN 用作源时钟

10.8.28 RCC PLL1 控制寄存器 (RCC_PLL1CR)

RCC PLL1 control register

偏移地址：0x4A0

复位值：0x0000 0000

如果 RCC_SECCFGR.PLL12SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制 PLL1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVREN	DIVQEN	DIVPEN	Res.	SSCG_CTRL	PLL1RDY	PLLON
									rw	rw	rw		rw	r	rw

位 31:7 保留，必须保持复位值。

位 6 **DIVREN**：PLL1 DIVR 分频器输出使能 (PLL1 DIVR divider output enable)
由软件置 1 和复位，用于使能 PLL1 的 pll1_r_ck 输出。
为实现节能，不需要 pll1_r_ck 时，必须将 DIVREN 和 DIVR 设为“0”。
0：禁止 pll1_r_ck 输出（复位后的默认值）
1：使能 pll1_r_ck 输出

位 5 **DIVQEN**：PLL1 DIVQ 分频器输出使能 (PLL1 DIVQ divider output enable)
由软件置 1 和复位，用于使能 PLL1 的 pll1_q_ck 输出。
为实现节能，不需要 pll1_q_ck 时，必须将 DIVQEN 和 DIVQ 设为“0”。
0：禁止 pll1_q_ck 输出（复位后的默认值）
1：使能 pll1_q_ck 输出

位 4 **DIVPEN**：PLL1 DIVP 分频器输出使能 (PLL1 DIVP divider output enable)
由软件置 1 和复位，用于使能 PLL1 的 pll1_p_ck 输出。
为实现节能，不需要 pll1_p_ck 时，必须将 DIVPEN 和 DIVP 设为“0”。
0：禁止 pll1_p_ck 输出（复位后的默认值）
1：使能 pll1_p_ck 输出

位 3 保留，必须保持复位值。

位 2 **SSCG_CTRL**：PLL1 的展频时钟发生器使能 (Spread Spectrum Clock Generator of PLL1 enable)

由软件置 1 和复位，用来使能 PLL1 的展频时钟发生器，以减少 EMI 峰值数量。

0：禁止时钟展频发生器（复位后的默认值）

1：使能时钟展频发生器

位 1 **PLL1RDY**：PLL1 时钟就绪标志 (PLL1 clock ready flag)

由硬件置 1，用以指示 PLL1 已锁定。

0：PLL1 失锁（复位后的默认值）

1：PLL1 已锁定

位 0 **PLLON**：PLL1 使能 (PLL1 enable)

由软件置 1 和清零，用于使能 PLL1。

请注意，在将 PLLON 设为“0”之前，必须将 PLL1 的 DIVPEN 设为“0”。有关详细信息，请参见 *PLL 禁止过程*一节。

0：PLL1 关闭（复位后的默认值）

1：PLL1 开启，并且 ref1_ck 提供给 PLL1

10.8.29 RCC PLL1 配置寄存器 1 (RCC_PLL1CFGR1)

RCC PLL1 configuration register 1

偏移地址：0x4A4

复位值：0x0001 0031

如果 RCC_SECCFGR.PLL12SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于配置 PLL1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVM1[5:0]					
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVN[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:16 **DIVM1[5:0]**：PLL1 的预分频器 (Prescaler for PLL1)

由软件置 1 和清零，用于配置 PLL1 的预分频器。

0x0：旁路

0x1：2 分频（复位后的默认值）

0x2：3 分频

...

0x3F：64 分频

位 15:9 保留，必须保持复位值。

位 8:0 **DIVN[8:0]**：PLL1 VCO 的倍频系数 (Multiplication factor for PLL1 VCO)

由软件写入，用于控制 VCO 的倍频系数。

警告：软件必须正确设置这些位，以确保 VCO 输出频率介于其有效频率范围内，即：992 至 2000 MHz。

当 FRACV 加载值为 0 时，VCO 输出频率 = Fref1_ck x 2 x (DIVN+1)，并且：

- (DIVN+1) 的有效分频比：31 到 125
- 输入频率 Fref1_ck：8 MHz 到 16 MHz

0x1E：分频比为 31

0x1F：分频比为 32

...

0x31：分频比为 50（复位后的默认值）

...

0x7C：分频比为 125

其他：错误配置

10.8.30 RCC PLL1 配置寄存器 2 (RCC_PLL1CFGR2)

RCC PLL1 configuration register 2

偏移地址：0x4A8

复位值：0x0001 0100

如果 RCC_SECCFGR.PLL12SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于配置 PLL1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVR[6:0]						
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DIVQ[6:0]							Res.	DIVP[6:0]						
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值。

位 22:16 **DIVR[6:0]**：PLL1 DIVR 分频系数 (PLL1 DIVR division factor)

由软件写入，用于控制 pll1_r_ck 时钟的频率。

0x0：pll1_r_ck = fout1_ck

0x1：pll1_r_ck = fout1_ck / 2（复位后的默认值）

0x2：pll1_r_ck = fout1_ck / 3

...

0x7F：pll1_r_ck = fout1_ck / 128

位 15 保留，必须保持复位值。

位 14:8 **DIVQ[6:0]**：PLL1 DIVQ 分频系数 (PLL1 DIVQ division factor)

由软件写入，用于控制 pll1_q_ck 时钟的频率。

0x0：pll1_q_ck = fout1_ck

0x1：pll1_q_ck = fout1_ck / 2（复位后的默认值）

0x2：pll1_q_ck = fout1_ck / 3

...

0x7F：pll1_q_ck = fout1_ck / 128

位 7 保留，必须保持复位值。

位 6:0 **DIVP[6:0]**：PLL1 DIVP 分频系数 (PLL1 DIVP division factor)

由软件写入，用于控制 pll1_p_ck 时钟的频率。

由于占空比降低，不推荐使用奇数分频系数。

0x0：pll1_p_ck = fout1_ck（复位后的默认值）

0x1：pll1_p_ck = fout1_ck / 2

0x2：pll1_p_ck = fout1_ck / 3

...

0x7F：pll1_p_ck = fout1_ck / 128

10.8.31 RCC PLL1 小数寄存器 (RCC_PLL1FRACR)

RCC PLL1 fractional register

偏移地址：0x4AC

复位值：0x0000 0000

如果 RCC_SECCFGR.PLL12SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于微调 PLL1 VCO 的频率。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRACLE
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRACV[12:0]													Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			

位 31:17 保留，必须保持复位值。

位 16 **FRACLE**：PLL1 小数锁存使能 (PLL1 fractional latch enable)

由软件置 1 和复位，用于将 FRACV 的内容锁存到 Σ−Δ 调制器中。

为将 FRACV 值锁存到 Σ−Δ 调制器中，必须将 FRACLE 设为“0”，然后将其置“1”：从 0 到 1 的跳变会将 FRACV 内容传送到调制器中。有关 PLL 编程的其他相关信息，请参见*重新编程后分频器*一节和 *PLL 初始化过程*一节。

位 15:3 **FRACV[12:0]**：PLL1 VCO 的倍频系数的小数部分 (Fractional part of the multiplication factor for PLL1 VCO)

由软件写入，用于控制 VCO 的倍频系数的小数部分。

这些位可随时写入，进而对 PLL1 VCO 进行动态微调。

警告：软件必须正确设置这些位，以确保 VCO 输出频率介于其有效频率范围内，即：992 到 2000 MHz

VCO 输出频率 = Fref1_ck x ((DIVN+1) + (FRACV / 8192))，并且：

- (DIVN+1) 必须在 31 到 125 之间
- FRACV 可以在 0 到 8191 之间
- 输入频率 Fref1_ck 必须在 4 MHz 到 16 MHz 之间

如果执意要在已使能 PLL 的情况下实时更改 FRACV 值，应用必须按如下过程进行操作：

- 将位 FRACLE 设为“0”，
- 将新的小数值写入 FRACV，
- 将位 FRACLE 置“1”。

位 2:0 保留，必须保持复位值。

10.8.32 RCC PLL1 时钟展频发生器寄存器 (RCC_PLL1CSGR)

RCC PLL1 clock spreading generator register

偏移地址：0x4B0

复位值：0x0000 0000

如果 RCC_SECCFGR.PLL12SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于配置 PLL1。不推荐在 PLL1 使能（PLLON = “1”）时更改此寄存器的内容。有关详细信息，请参见*在展频模式下使用 PLL*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	INC_STEP[14:0]														
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSCG_MODE	RPDFN_DIS	TPDFN_DIS	MOD_PER[12:0]												
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:16 **INC_STEP[14:0]**：PLL1 的调制深度调整 (Modulation Depth Adjustment for PLL1)
由软件写入，用于调整时钟展频发生器的调制深度。

位 15 **SSCG_MODE**：展频时钟发生器模式 (Spread spectrum clock generator mode)
由软件置 1 和复位，用于选择时钟展频模式。
0：选择中心展频调制（复位后的默认值）
1：选择向下展频调制

位 14 **RPDFN_DIS**：抖动 RPDF 噪声控制 (Dithering RPDF noise control)
由软件置 1 和复位。
该位用于使能或禁止向 SSCG 调制器注入抖动噪声。这种抖动噪声是使用矩形概率密度函数生成的。
0：抖动噪声注入使能（复位后的默认设置）
1：禁止抖动噪声注入

位 13 **TPDFN_DIS**：抖动 TPDF 噪声控制 (Dithering RPDF noise control)
由软件置 1 和复位。
该位用于使能或禁止向 SSCG 调制器注入抖动噪声。这种抖动噪声是使用三角概率密度函数生成的。
0：抖动噪声注入使能（复位后的默认设置）
1：禁止抖动噪声注入

位 12:0 **MOD_PER[12:0]**：PLL1 的调制周期调整 (Modulation Period Adjustment for PLL1)
由软件写入，用于调整时钟展频发生器的调制周期。

10.8.33 RCC PLL2 控制寄存器 (RCC_PLL2CR)

RCC PLL2 control register

偏移地址：0x4D0

复位值：0x0000 0000

如果 RCC_SECCFGR.PLL12SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制 PLL2。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVREN	DIVQEN	DIVPEN	Res.	SSCG_CTRL	PLL2RDY	PLLON
									rw	rw	rw		rw	r	rw

位 31:7 保留，必须保持复位值。

位 6 **DIVREN**：PLL2 DIVR 分频器输出使能 (PLL2 DIVP divider output enable)
由软件置 1 和复位，用于使能 PLL2 的 pll2_r_ck 输出。
为实现节能，不需要 pll2_r_ck 时，必须将 DIVREN 和 DIVR 设为“0”。
0：禁止 pll2_r_ck 输出（复位后的默认值）
1：使能 pll2_r_ck 输出

位 5 **DIVQEN**：PLL2 DIVQ 分频器输出使能 (PLL1 DIVP divider output enable)
由软件置 1 和复位，用于使能 PLL2 的 pll2_q_ck 输出。
为实现节能，不需要 pll2_q_ck 时，必须将 DIVQEN 和 DIVQ 设为“0”。
0：禁止 pll2_q_ck 输出（复位后的默认值）
1：使能 pll2_q_ck 输出

位 4 **DIVPEN**：PLL2 DIVP 分频器输出使能 (PLL2 DIVP divider output enable)
由软件置 1 和复位，用于使能 PLL2 的 pll2_p_ck 输出。
为实现节能，不需要 pll2_p_ck 时，必须将 DIVPEN 和 DIVP 设为“0”。
0：禁止 pll2_p_ck 输出（复位后的默认值）
1：使能 pll2_p_ck 输出

位 3 保留，必须保持复位值。

位 2 **SSCG_CTRL**：PLL2 的时钟展频发生器使能

由软件置 1 和复位，用来使能 PLL2 的时钟展频发生器，以减少 EMI 峰值数量。

0：禁止时钟展频发生器（复位后的默认值）

1：使能时钟展频发生器

位 1 **PLL2RDY**：PLL2 时钟就绪标志 (PLL2 clock ready flag)

由硬件置 1，用以指示 PLL2 已锁定。

0：PLL2 失锁（复位后的默认值）

1：PLL2 已锁定

位 0 **PLLON**：PLL2 使能 (PLL2 enable)

由软件置 1 和清零，用于使能 PLL2。

0：PLL2 关闭（复位后的默认值）

1：PLL2 开启，并向 PLL2 提供 ref2_ck

10.8.34 RCC PLL2 配置寄存器 1 (RCC_PLL2CFGR1)

RCC PLL2 configuration register 1

偏移地址：0x4D4

复位值：0x0001 0063

如果 RCC_SECCFGR.PLL12SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于配置 PLL2。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVM2[5:0]					
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVN[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:16 **DIVM2[5:0]**：PLL2 的预分频器 (Prescaler for PLL2)

由软件置 1 和清零，用于配置 PLL2 的预分频器。

0x0：旁路

0x1：2 分频（复位后的默认值）

0x2：3 分频

...

0x3F：64 分频

位 15:9 保留，必须保持复位值。

位 8:0 **DIVN[8:0]**：PLL2 VCO 的倍频系数 (Multiplication factor for PLL2 VCO)

由软件写入，用于控制 VCO 的倍频系数。

警告：软件必须正确设置这些位，以确保 VCO 输出频率介于其有效频率范围内，即：800 到 1600 MHz。

当 FRACV 加载值为 0 时，VCO 输出频率 = Fref2_ck x (DIVN+1)，并且：

- (DIVN+1) 的有效分频比：25 到 100
- 输入频率 Fref2_ck：8 MHz 到 16 MHz

其他：错误配置

0x18：分频比为 25

0x19：分频比为 26

...

0x31：分频比为 50

...

0x63：分频比为 100（复位后的默认值）

10.8.35 RCC PLL2 配置寄存器 2 (RCC_PLL2CFGR2)

RCC PLL2 configuration register 2

偏移地址：0x4D8

复位值：0x0001 0101

如果 RCC_SECCFGR.PLL12SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于配置 PLL2。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVR[6:0]						
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DIVQ[6:0]							Res.	DIVP[6:0]						
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值。

位 22:16 **DIVR[6:0]**：PLL2 DIVR 分频系数 (PLL2 DIVR division factor)

由软件写入，用于控制 pll2_r_ck 时钟的频率。

0x0：pll2_r_ck = fout2_ck

0x1：pll2_r_ck = fout2_ck / 2（复位后的默认值）

0x2：pll2_r_ck = fout2_ck / 3

...

0x7F：pll2_r_ck = fout2_ck / 128

位 15 保留，必须保持复位值。

位 14:8 **DIVQ[6:0]**：PLL2 DIVQ 分频系数 (PLL2 DIVQ division factor)

由软件写入，用于控制 pll2_q_ck 时钟的频率。

0x0：pll2_q_ck = fout2_ck

0x1：pll2_q_ck = fout2_ck / 2（复位后的默认值）

0x2：pll2_q_ck = fout2_ck / 3

...

0x7F：pll2_q_ck = fout2_ck / 128

位 7 保留，必须保持复位值。

位 6:0 **DIVP[6:0]**：PLL2 DIVP 分频系数 (PLL2 DIVP division factor)

由软件写入，用于控制 pll2_p_ck 时钟的频率。

0x0：pll2_p_ck = fout2_ck

0x1：pll2_p_ck = fout2_ck / 2（复位后的默认值）

0x2：pll2_p_ck = fout2_ck / 3

...

0x7F：pll2_p_ck = fout2_ck / 128

10.8.36 RCC PLL2 小数寄存器 (RCC_PLL2FRACR)

RCC PLL2 fractional register

偏移地址：0x4DC

复位值：0x0000 0000

如果 RCC_SECCFGR.PLL12SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于微调 PLL2 VCO 的频率。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRACLE
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRACV[12:0]													Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			

位 31:17 保留，必须保持复位值。

位 16 **FRACLE**：PLL2 小数锁存使能 (PLL2 fractional latch enable)

由软件置 1 和复位，用于将 FRACV 的内容锁存到 Σ−Δ 调制器中。

为将 FRACV 值锁存到 Σ−Δ 调制器中，必须将 FRACLE 设为“0”，然后将其置“1”：从 0 到 1 的跳变会将 FRACV 内容传送到调制器中。有关 PLL 编程的其他相关信息，请参见*重新编程后分频器*一节和 *PLL 初始化过程*一节。

位 15:3 **FRACV[12:0]**：PLL2 VCO 的倍频系数的小数部分 (Fractional part of the multiplication factor for PLL2 VCO)

由软件写入，用于控制 VCO 的倍频系数的小数部分。

这些位可随时写入，进而对 PLL2 VCO 进行动态微调。

警告：软件必须正确设置这些位，以确保 VCO 输出频率介于其有效频率范围内，即：800 到 1600 MHz

VCO 输出频率 = Fref2_ck x ((DIVN+1) + (FRACV / 8192))，并且：

- (DIVN+1) 必须在 25 到 100 之间
- FRACV 可以在 0 到 8191 之间
- 输入频率 Fref2_ck 必须在 4 MHz 到 16 MHz 之间

如果执意要在已使能 PLL 的情况下实时更改 FRACV 值，应用必须按如下过程进行操作：

- 将位 FRACLE 设为“0”，
- 将新的小数值写入 FRACV，
- 将位 FRACLE 置“1”。

位 2:0 保留，必须保持复位值。

10.8.37 RCC PLL2 时钟展频发生器寄存器 (RCC_PLL2CSGR)

RCC PLL2 clock spreading generator register

偏移地址：0x4E0

复位值：0x0000 0000

如果 RCC_SECCFGR.PLL12SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于配置 PLL2。不推荐在 PLL2 使能（PLLON = “1”）时更改此寄存器的内容。有关详细信息，请参见*在展频模式下使用 PLL*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	INC_STEP[14:0]														
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSCG_MODE	RPDFN_DIS	TPDFN_DIS	MOD_PER[12:0]												
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:16 **INC_STEP[14:0]**：PLL2 的调制深度调整 (Modulation Depth Adjustment for PLL2)
由软件写入，用于调整时钟展频发生器的调制深度。

位 15 **SSCG_MODE**：展频时钟发生器模式 (Spread spectrum clock generator mode)
由软件置 1 和复位，用于选择时钟展频模式。
0：选择中心展频调制（复位后的默认值）
1：选择向下展频调制

位 14 **RPDFN_DIS**：抖动 RPDF 噪声控制 (Dithering RPDF noise control)
由软件置 1 和复位。
该位用于使能或禁止向 SSCG 调制器注入抖动噪声。这种抖动噪声是使用矩形概率密度函数生成的。
0：抖动噪声注入使能（复位后的默认设置）
1：禁止抖动噪声注入

位 13 **TPDFN_DIS**：抖动 TPDF 噪声控制 (Dithering RPDF noise control)
由软件置 1 和复位。
该位用于使能或禁止向 SSCG 调制器注入抖动噪声。这种抖动噪声是使用三角概率密度函数生成的。
0：抖动噪声注入使能（复位后的默认设置）
1：禁止抖动噪声注入

位 12:0 **MOD_PER[12:0]**：PLL2 的调制周期调整 (Modulation Period Adjustment for PLL2)
由软件写入，用于调整时钟展频发生器的调制周期。

10.8.38 RCC PLL3 控制寄存器 (RCC_PLL3CR)

RCC PLL3 control register

偏移地址：0x500

复位值：0x0000 0000

如果 RCC_SECCFGR.PLL3SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制 PLL3。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVREN	DIVQEN	DIVPEN	Res.	SSCG_CTRL	PLL3RDY	PLLON
									rw	rw	rw		rw	r	rw

位 31:7 保留，必须保持复位值。

位 6 **DIVREN**：PLL3 DIVR 分频器输出使能 (PLL3 DIVR divider output enable)
由软件置 1 和复位，用于使能 PLL3 的 pll3_r_ck 输出。
当进入 Stop 模式或待机模式时由硬件清零。
为实现节能，不需要 pll3_r_ck 时，必须将 DIVREN 和 DIVR 设为“0”。
0：禁止 pll3_r_ck 输出（复位后的默认值）
1：使能 pll3_r_ck 输出

位 5 **DIVQEN**：PLL3 DIVQ 分频器输出使能 (PLL3 DIVQ divider output enable)
由软件置 1 和复位，用于使能 PLL3 的 pll3_q_ck 输出。
当进入 Stop 模式或待机模式时由硬件清零。
为实现节能，不需要 pll3_q_ck 时，必须将 DIVQEN 和 DIVQ 设为“0”。
0：禁止 pll3_q_ck 输出（复位后的默认值）
1：使能 pll3_q_ck 输出

位 4 **DIVPEN**：PLL3 DIVP 分频器输出使能 (PLL3 DIVR divider output enable)
由软件置 1 和复位，用于使能 PLL3 的 pll3_p_ck 输出。
当进入 Stop 模式或待机模式时由硬件清零。
为实现节能，不需要 pll3_p_ck 时，必须将 DIVPEN 和 DIVP 设为“0”。
0：禁止 pll3_p_ck 输出（复位后的默认值）
1：使能 pll3_p_ck 输出

位 3 保留，必须保持复位值。

位 2 **SSCG_CTRL**：PLL3 的时钟展频发生器使能 (Clock Spreading Generator of PLL3 enable)

由软件置 1 和复位，用来使能 PLL3 的时钟展频发生器，以减少 EMI 峰值数量。

0：禁止时钟展频发生器（复位后的默认值）

1：使能时钟展频发生器

位 1 **PLL3RDY**：PLL3 时钟就绪标志 (PLL3 clock ready flag)

由硬件置 1，用以指示 PLL3 已锁定。

0：PLL3 失锁（复位后的默认值）

1：PLL3 已锁定

位 0 **PLLON**：PLL3 使能 (PLL3 enable)

由软件置 1 和清零，用于使能 PLL3。

当进入 Stop 模式或待机模式时由硬件清零。

请注意，在将 PLLON 设为“0”之前，必须将 PLL3 的 DIVPEN、DIVQEN 和 DIVREN 设为“0”。有关详细信息，请参见 *PLL 禁止过程*一节。

0：PLL3 关闭（复位后的默认值）

1：PLL3 开启，并向 PLL3 提供 ref3_ck

10.8.39 RCC PLL3 配置寄存器 1 (RCC_PLL3CFGR1)

RCC PLL3 configuration register 1

偏移地址：0x504

复位值：0x0001 0031

如果 RCC_SECCFGR.PLL3SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于配置 PLL3。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	IFRGE[1:0]		Res.	Res.	DIVM3[5:0]					
						rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVN[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25:24 **IFRGE[1:0]**：PLL3 输入频率范围 (PLL3 input frequency range)

由软件写入，用于为 PLL3 选用合适的参考频率范围。

0x0：PLL3 输入 (ref3_ck) 时钟频率范围介于 4 MHz 到 8 MHz 之间（复位后的默认值）

0x1：PLL3 输入 (ref3_ck) 时钟频率范围介于 8 MHz 到 16 MHz 之间。

请注意，如果 ref3_ck 等于 8 MHz，推荐设置 IFRGE =“x1”

位 23:22 保留，必须保持复位值。

位 21:16 **DIVM3[5:0]**：PLL3 的预分频器 (Prescaler for PLL3)
由软件置 1 和清零，用于配置 PLL3 的预分频器。
0x0：旁路
0x1：2 分频（复位后的默认值）
0x2：3 分频
...
0x3F：64 分频

位 15:9 保留，必须保持复位值。

位 8:0 **DIVN[8:0]**：PLL3 VCO 的倍频系数 (Multiplication factor for PLL3 VCO)
由软件写入，用于控制 VCO 的倍频系数。
警告：软件必须正确设置这些位，以确保 VCO 输出频率
介于其有效频率范围内，即 400 到 800 MHz 之间。

当 FRACV 加载值为 0 时，VCO 输出频率 = Fref3_ck x (DIVN+1)，并且：
- (DIVN+1) 的有效分频比：介于 25 到 200 之间
- 输入频率 Fref3_ck：介于 4 MHz 到 16 Mhz 之间

0x18：分频比为 25
0x19：分频比为 26
0x1A：分频比为 27
...
0x31：分频比为 50（复位后的默认值）
...
0xC7：分频比为 200
其他：错误配置

10.8.40 RCC PLL3 配置寄存器 2 (RCC_PLL3CFGR2)

RCC PLL3 configuration register 2

偏移地址：0x508

复位值：0x0001 0101

如果 RCC_SECCFGR.PLL3SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于配置 PLL3。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVR[6:0]						
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DIVQ[6:0]							Res.	DIVP[6:0]						
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值。

位 22:16 **DIVR[6:0]**：PLL3 DIVR 分频系数 (PLL3 DIVR division factor)

由软件写入，用于控制 pll3_r_ck 时钟的频率。

0x0：pll3_r_ck = vco3_ck

0x1：pll3_r_ck = vco3_ck / 2（复位后的默认值）

0x2：pll3_r_ck = vco3_ck / 3

...

0x7F：pll3_r_ck = vco3_ck / 128

位 15 保留，必须保持复位值。

位 14:8 **DIVQ[6:0]**：PLL3 DIVQ 分频系数 (PLL3 DIVQ division factor)

由软件写入，用于控制 pll3_q_ck 时钟的频率。

0x0：pll3_q_ck = vco3_ck

0x1：pll3_q_ck = vco3_ck / 2（复位后的默认值）

0x2：pll3_q_ck = vco3_ck / 3

...

0x7F：pll3_q_ck = vco3_ck / 128

位 7 保留，必须保持复位值。

位 6:0 **DIVP[6:0]**：PLL3 DIVP 分频系数 (PLL3 DIVP division factor)

由软件写入，用于控制 pll3_p_ck 时钟的频率。

0x0：pll3_p_ck = vco3_ck

0x1：pll3_p_ck = vco3_ck / 2（复位后的默认值）

0x2：pll3_p_ck = vco3_ck / 3

...

0x7F：pll3_p_ck = vco3_ck / 128

10.8.41 RCC PLL3 小数寄存器 (RCC_PLL3FRACR)

RCC PLL3 Fractional register

偏移地址：0x50C

复位值：0x0000 0000

如果 RCC_SECCFGR.PLL3SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于微调 PLL3 VCO 的频率。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRACLE
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRACV[12:0]													Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			

位 31:17 保留，必须保持复位值。

位 16 **FRACLE**：PLL3 小数锁存使能 (PLL3 fractional latch enable)

由软件置 1 和复位，用于将 FRACV 的内容锁存到 Σ−Δ 调制器中。

为将 FRACV 值锁存到 Σ−Δ 调制器中，必须将 FRACLE 设为“0”，然后将其置“1”：从 0 到 1 的跳变会将 FRACV 内容传送到调制器中。有关 PLL 编程的其他相关信息，请参见 *PLL 初始化过程*一节和*重新编程后分频器*一节。

位 15:3 **FRACV[12:0]**：PLL3 VCO 的倍频系数的小数部分 (Fractional part of the multiplication factor for PLL3 VCO)

由软件写入，用于控制 VCO 的倍频系数的小数部分。

这些位可随时写入，进而对 PLL3 VCO 进行动态微调。

警告：软件必须正确设置这些位，以确保 VCO 输出频率介于其有效频率范围内，即 400 到 800 MHz 之间。

VCO 输出频率 = Fref3_ck x ((DIVN+1) + (FRACV / 8192))，并且：

- (DIVN+1) 必须在 25 到 200 之间
- FRACV 可以在 0 到 8191 之间
- 输入频率 Fref3_ck 必须介于 1 MHz 到 16 MHz 之间

如果执意要在已使能 PLL 的情况下实时更改 FRACV 值，应用必须按如下过程进行操作：

- 将位 FRACLE 设为“0”，
- 将新的小数值写入 FRACV，
- 将位 FRACLE 置“1”。

位 2:0 保留，必须保持复位值。

10.8.42 RCC PLL3 时钟展频发生器寄存器 (RCC_PLL3CSGR)

RCC PLL3 clock spreading generator register

偏移地址：0x510

复位值：0x0000 0000

如果 RCC_SECCFGR.PLL3SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于配置 PLL3。不推荐在 PLL3 使能（PLLON = “1”）时更改此寄存器的内容。有关详细信息，请参见*在展频模式下使用 PLL*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	INC_STEP[14:0]														
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSCG_MODE	RPDFN_DIS	TPDFN_DIS	MOD_PER[12:0]												
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:16 **INC_STEP[14:0]**：PLL3 的调制深度调整 (Modulation Depth Adjustment for PLL3)
由软件写入，用于调整时钟展频发生器的调制深度。

位 15 **SSCG_MODE**：展频时钟发生器模式 (Spread spectrum clock generator mode)
由软件置 1 和复位，用于选择时钟展频模式。
0：选择中心展频调制（复位后的默认值）
1：选择向下展频调制

位 14 **RPDFN_DIS**：抖动 RPDF 噪声控制 (Dithering RPDF noise control)
由软件置 1 和复位。
该位用于使能或禁止向 SSCG 调制器注入抖动噪声。这种抖动噪声是使用矩形概率密度函数生成的。
0：抖动噪声注入使能（复位后的默认设置）
1：禁止抖动噪声注入

位 13 **TPDFN_DIS**：抖动 TPDF 噪声控制 (Dithering RPDF noise control)
由软件置 1 和复位。
该位用于使能或禁止向 SSCG 调制器注入抖动噪声。这种抖动噪声是使用三角概率密度函数生成的。
0：抖动噪声注入使能（复位后的默认设置）
1：禁止抖动噪声注入

位 12:0 **MOD_PER[12:0]**：PLL3 的调制周期调整 (Modulation Period Adjustment for PLL3)
由软件写入，用于调整时钟展频发生器的调制周期。

10.8.43 RCC PLL4 控制寄存器 (RCC_PLL4CR)

RCC PLL4 control register

偏移地址：0x520

复位值：0x0000 0000

如果 RCC_SECCFGR.PLL4SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制 PLL4。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVREN	DIVQEN	DIVPEN	Res.	SSCG_CTRL	PLL4RDY	PLLON
									rw	rw	rw		rw	r	rw

位 31:7 保留，必须保持复位值。

位 6 **DIVREN**：PLL4 DIVR 分频器输出使能 (PLL4 DIVR divider output enable)
由软件置 1 和复位，用于使能 PLL4 的 pll4_r_ck 输出。
当进入 Stop 模式或待机模式时由硬件清零。
为实现节能，不需要 pll4_r_ck 时，必须将 DIVREN 设为“0”。
0：禁止 pll4_r_ck 输出（复位后的默认值）
1：使能 pll4_r_ck 输出

位 5 **DIVQEN**：PLL4 DIVQ 分频器输出使能 (PLL4 DIVQ divider output enable)
由软件置 1 和复位，用于使能 PLL4 的 pll4_q_ck 输出。
当进入 Stop 模式或待机模式时由硬件清零。
为实现节能，不需要 pll4_q_ck 时，必须将 DIVQEN 设为“0”。
0：禁止 pll4_q_ck 输出（复位后的默认值）
1：使能 pll4_q_ck 输出

位 4 **DIVPEN**：PLL4 DIVP 分频器输出使能 (PLL4 DIVP divider output enable)
由软件置 1 和复位，用于使能 PLL4 的 pll4_p_ck 输出。
当进入 Stop 模式或待机模式时由硬件清零。
为实现节能，不需要 pll4_p_ck 时，必须将 DIVPEN 设为“0”。
0：禁止 pll4_p_ck 输出（复位后的默认值）
1：使能 pll4_p_ck 输出

位 3 保留，必须保持复位值。

位 2 **SSCG_CTRL**：PLL4 的时钟展频发生器使能 (Clock Spreading Generator of PLL4 enable)

由软件置 1 和复位，用来使能 PLL4 的时钟展频发生器，以减少 EMI 峰值数量。

0：禁止时钟展频发生器（复位后的默认值）

1：使能时钟展频发生器

位 1 **PLL4RDY**：PLL4 时钟就绪标志 (PLL4 clock ready flag)

由硬件置 1，用以指示 PLL4 已锁定。

0：PLL4 失锁（复位后的默认值）

1：PLL4 锁定

位 0 **PLLON**：PLL4 使能 (PLL4 enable)

由软件置 1 和清零，用于使能 PLL4。

当进入 Stop 模式或待机模式时由硬件清零。

请注意，在将 PLLON 设为“0”之前，必须将 PLL4 的 DIVPEN、DIVQEN 和 DIVREN 设为“0”。有关详细信息，请参见 *PLL 禁止过程*一节。

0：PLL4 关闭（复位后的默认值）

1：PLL4 开启，并向 PLL4 提供 ref4_ck

10.8.44 RCC PLL4 配置寄存器 1 (RCC_PLL4CFGR1)

RCC PLL4 configuration register 1

偏移地址：0x524

复位值：0x0001 0031

如果 RCC_SECCFGR.PLL4SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于配置 PLL4。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	IFRGE[1:0]		Res.	Res.	DIVM4[5:0]					
						rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVN[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25:24 **IFRGE[1:0]**：PLL4 输入频率范围 (PLL4 input frequency range)

由软件写入，用于为 PLL4 选用合适的参考频率范围。

x0：PLL4 输入 (ref4_ck) 时钟频率范围介于 4 MHz 到 8 MHz 之间（复位后的默认值）

x1：PLL4 输入 (ref4_ck) 时钟频率范围介于 8 MHz 到 16 MHz 之间

请注意，如果 ref3_ck 等于 8 MHz，推荐设置 IFRGE =“x1”

位 23:22 保留，必须保持复位值。

位 21:16 **DIVM4[5:0]**：PLL4 的预分频器 (Prescaler for PLL4)
由软件置 1 和清零，用于配置 PLL4 的预分频器。
0x0：旁路
0x1：2 分频（复位后的默认值）
0x2：3 分频
...
0x3F：64 分频

位 15:9 保留，必须保持复位值。

位 8:0 **DIVN[8:0]**：PLL4 VCO 的倍频系数 (Multiplication factor for PLL4 VCO)
由软件写入，用于控制 VCO 的倍频系数。
警告：软件必须正确设置这些位，以确保 VCO 输出频率介于其有效频率范围内，即 400 到 800 MHz 之间。

当 FRACV 加载值为 0 时，VCO 输出频率 = Fref4_ck x (DIVN+1)，并且：
- (DIVN+1) 的有效分频比：介于 25 到 200 之间
- 输入频率 Fref4_ck：介于 4 MHz 到 16 Mhz 之间

0x18：分频比为 25
0x19：分频比为 26
0x1A：分频比为 27
...
0x31：分频比为 50（复位后的默认值）
...
0xC7：分频比为 200
其他：错误配置

10.8.45 RCC PLL4 配置寄存器 2 (RCC_PLL4CFGR2)

RCC PLL4 configuration register 2

偏移地址：0x528

复位值：0x0000 0000

如果 RCC_SECCFGR.PLL4SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于配置 PLL4。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVR[6:0]						
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DIVQ[6:0]							Res.	DIVP[6:0]						
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值。

位 22:16 **DIVR[6:0]**：PLL4 DIVR 分频系数 (PLL4 DIVR division factor)

由软件写入，用于控制 pll4_r_ck 时钟的频率。

0x0：pll4_r_ck = vco4_ck（复位后的默认值）

0x1：pll4_r_ck = vco4_ck / 2

0x2：pll4_r_ck = vco4_ck / 3

...

0x7F：pll4_r_ck = vco4_ck / 128

位 15 保留，必须保持复位值。

位 14:8 **DIVQ[6:0]**：PLL4 DIVQ 分频系数 (PLL4 DIVQ division factor)

由软件写入，用于控制 pll4_q_ck 时钟的频率。

0x0：pll4_q_ck = vco4_ck（复位后的默认值）

0x1：pll4_q_ck = vco4_ck / 2

0x2：pll4_q_ck = vco4_ck / 3

...

0x7F：pll4_q_ck = vco4_ck / 128

位 7 保留，必须保持复位值。

位 6:0 **DIVP[6:0]**：PLL4 DIVP 分频系数 (PLL4 DIVP division factor)

由软件写入，用于控制 pll4_p_ck 时钟的频率。

0x0：pll4_p_ck = vco4_ck（复位后的默认值）

0x1：pll4_p_ck = vco4_ck / 2

0x2：pll4_p_ck = vco4_ck / 3

...

0x7F：pll4_p_ck = vco4_ck / 128

10.8.46 RCC PLL4 小数寄存器 (RCC_PLL4FRACR)

RCC PLL4 fractional register

偏移地址：0x52C

复位值：0x0000 0000

如果 RCC_SECCFGR.PLL4SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于微调 PLL4 VCO 的频率。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRACLE
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRACV[12:0]													Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			

位 31:17 保留，必须保持复位值。

位 16 **FRACLE**：PLL4 小数锁存使能 (PLL4 fractional latch enable)

由软件置 1 和复位，用于将 FRACV 的内容锁存到 Σ−Δ 调制器中。

为将 FRACV 值锁存到 Σ−Δ 调制器中，必须将 FRACLE 设为“0”，然后将其置“1”：从 0 到 1 的跳变会将 FRACV 内容传送到调制器中。请参见：其他相关信息，请参见 *PLL 初始化过程*一节和*重新编程后分频器*一节。

位 15:3 **FRACV[12:0]**：PLL4 VCO 的倍频系数的小数部分 (Fractional part of the multiplication factor for PLL4 VCO)

由软件写入，用于控制 VCO 的倍频系数的小数部分。

这些位可随时写入，进而对 PLL4 VCO 进行动态微调。

警告：软件必须正确设置这些位，以确保 VCO 输出频率介于其有效频率范围内，即 400 到 800 MHz 之间。

VCO 输出频率 = Fref4_ck x ((DIVN+1) + (FRACV / 8192))，并且：

- (DIVN+1) 必须在 25 到 200 之间
- FRACV 可以在 0 到 8191 之间
- 输入频率 Fref4_ck 必须介于 1 MHz 到 16 MHz 之间

如果执意要在已使能 PLL 的情况下实时更改 FRACV 值，应用必须按如下过程进行操作：

- 将位 FRACLE 设为“0”，
- 将新的小数值写入 FRACV，
- 将位 FRACLE 置“1”。

位 2:0 保留，必须保持复位值。

10.8.47 RCC PLL4 时钟展频发生器寄存器 (RCC_PLL4CSGR)

RCC PLL4 clock spreading generator register

偏移地址：0x530

复位值：0x0000 0000

如果 RCC_SECCFGR.PLL4SEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于配置 PLL4。不推荐在 PLL4 使能（PLLON = “1”）时更改此寄存器的内容。有关详细信息，请参见*在展频模式下使用 PLL*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	INC_STEP[14:0]														
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSCG_MODE	RPDFN_DIS	TPDFN_DIS	MOD_PER[12:0]												
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:16 **INC_STEP[14:0]**：PLL4 的调制深度调整
由软件写入，用于调整时钟展频发生器的调制深度。

位 15 **SSCG_MODE**：展频时钟发生器模式 (Spread spectrum clock generator mode)
由软件置 1 和复位，用于选择时钟展频模式。
0：选择中心展频调制（复位后的默认值）
1：选择向下展频调制

位 14 **RPDFN_DIS**：抖动 RPDF 噪声控制 (Dithering RPDF noise control)
由软件置 1 和复位。
该位用于使能或禁止向 SSCG 调制器注入抖动噪声。这种抖动噪声是使用矩形概率密度函数生成的。
0：抖动噪声注入使能（复位后的默认设置）
1：禁止抖动噪声注入

位 13 **TPDFN_DIS**：抖动 TPDF 噪声控制 (Dithering RPDF noise control)
由软件置 1 和复位。
该位用于使能或禁止向 SSCG 调制器注入抖动噪声。这种抖动噪声是使用三角概率密度函数生成的。
0：抖动噪声注入使能（复位后的默认设置）
1：禁止抖动噪声注入

位 12:0 **MOD_PER[12:0]**：PLL4 的调制周期调整 (Modulation Period Adjustment for PLL4)
由软件写入，用于调整时钟展频发生器的调制周期。

10.8.48 RCC MPU 时钟选择寄存器 (RCC_MPCKSELR)

RCC MPU clock selection register

偏移地址：0x540

复位值：0x8000 0000

如果 RCC_SECCFGR.MPUSEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于选择 MPU 的时钟源。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MPUSRCRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPUSRC[1:0]	
														rw	rw

位 31 **MPUSRCRDY**：MPU 时钟切换状态 (MPU clock switch status)
由硬件置 1 和复位，以指示 MPU 时钟切换跳变是否已正确执行。
0：MPU 切换未就绪：其输出上未产生时钟
1：MPU 切换就绪：时钟切换选择由 MPUSRC 位字段指定的时钟。（复位后的默认值）

位 30:2 保留，必须保持复位值。

位 1:0 **MPUSRC[1:0]**：MPU 时钟切换 (MPU clock switch)
由软件写入，用来选择 MPU 子系统时钟源 (mpuss_ck)。
请注意，如果 MPUSRC[1:0] 不为“11”，将禁止 MPUDIV。
确保在将 MPUSRC 设置为“11”之前，MPUDIV 不为“000”。
0x0：选择 HSI 作为 MPU 子系统时钟 (hsi_ck)（复位后的默认值）
0x1：选择 HSE 作为 MPU 子系统时钟 (hse_ck)
0x2：选择 PLL1 作为 MPU 子系统时钟 (pll1_p_ck)
0x3：选择分频 PLL1 pll1_p_ck 时钟作为 MPU 子系统时钟，参见 RCC_MPCKDIVR.MPUDIV[3:0]

10.8.49 RCC AXI 子系统时钟选择寄存器 (RCC_ASSCKSELR)

RCC AXI sub-system clock selection register

偏移地址：0x544

复位值：0x8000 0000

如果 RCC_SECCFGR.AXISEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于选择 AXI 子系统的时钟源。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AXISSRCRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXISSRC[2:0]		
													rw	rw	rw

位 31 **AXISSRCRDY**：AXI 子系统时钟切换状态 (AXI sub-system clock switch status)
由硬件置 1 和复位，以指示 AXI 时钟切换跳变是否已正确执行。
0：AXI 子系统开关未就绪或处在高于“011”的位置：其输出上没有生成时钟
1：AXI 子系统开关已就绪：时钟切换正在选择 AXISSRC 位字段指定的时钟。（复位后的默认值）

位 30:3 保留，必须保持复位值。

位 2:0 **AXISSRC[2:0]**：AXI 子系统时钟切换 (AXI sub-system clock switch)
由软件写入，用来选择 AXI 子系统时钟源 (axiss_ck)。
0x0：选择 HSI 作为 AXI 子系统时钟 (hsi_ck)（复位后的默认值）
0x1：选择 HSE 作为 AXI 子系统时钟 (hse_ck)
0x2：选择 PLL2 作为 AXI 子系统时钟 (pll2_p_ck)
其他值：axiss_ck 被门控

10.8.50 RCC MLAHB 子系统时钟选择寄存器 (RCC_MSSCKSELR)

RCC MLAHB sub-system clock selection register

偏移地址：0x548

复位值：0x8000 0000

如果 RCC_SECCFGR.MLAHBSEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于选择 MLAHB 子系统的时钟源。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MLAHBSSRCRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MLAHBSSRC[1:0]	
														rw	rw

位 31 **MLAHBSSRCRDY**：MLAHB 子系统时钟切换状态 (MLAHB sub-system clock switch status)

由硬件置 1 和复位，以指示 MLAHB 时钟切换跳变是否已正确执行。

0：MLAHB 子系统开关未就绪或处在高于“011”的位置：其输出上没有生成时钟

1：MLAHB 子系统开关已就绪：时钟切换正在选择 MLAHBSSRC 位字段指定的时钟。（复位后的默认值）

位 30:2 保留，必须保持复位值。

位 1:0 **MLAHBSSRC[1:0]**：MLAHB 子系统时钟切换 (MLAHB sub-system clock switch)

由软件写入，用来选择 MLAHB 子系统时钟源 (mlahbss_ck)。

当系统退出一种 Stop 模式时，由硬件复位以选择 hsi_ck。

0x0：选择 HSI 作为 MLAHB 子系统时钟 (hsi_ck)（复位后的默认值）

0x1：选择 HSE 作为 MLAHB 子系统时钟 (hse_ck)

0x2：选择 CSI 作为 MLAHB 子系统时钟 (csi_ck)

0x3：选择 PLL3 作为 MLAHB 子系统时钟 (pll3_p_ck)

10.8.51 RCC 通用外设内核时钟选择寄存器 (RCC_CPERCKSELR)

RCC common peripheral kernel clock selection register

偏移地址：0x54C

复位值：0x0000 0000

如果 RCC_SECCFGR.CPERSEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于为 per_ck 时钟选择振荡器源。该时钟是一个通用内核时钟，可供多个外设选择：可安全的 (AHB2) ADC1/2、可安全的 (AHB6) FMC 和 QUADSPI、可安全的 (APB3) LPTIM2/3 和其他非安全 DFSM、LPTIM1/4/5、SAI1/2 和 SPI2S1/2。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKPERSRC[1:0]	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **CKPERSRC[1:0]**：内核时钟的振荡器选择 (Oscillator selection for kernel clock)
由软件置 1 和清零。
0x0：选择了 hsi_ker_ck 时钟（复位后的默认值）
0x1：选择了 csi_ker_ck 时钟
0x2：选择了 hse_ker_ck 时钟
0x3：禁止时钟

10.8.52 RCC RTC 时钟分频寄存器 (RCC_RTCDIVR)

RCC RTC clock division register

偏移地址：0x560

复位值：0x0000 0000

如果 RTC 的至少一个功能通过 RTC_SECCFGR 寄存器配置为安全（即如果由 RTC 驱动的输入信号 rtc_sec 置为有效），则对该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

注： *备份域复位后，RTC 默认处于非安全状态。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTCDIV[5:0]					
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持复位值。

位 5:0 **RTCDIV[5:0]**：适用于 RTC 时钟的 HSE 分频系数 (HSE division factor for RTC clock)

由软件置 1 和清零，用于对 HSE 进行分频，进而为 RTC 生成时钟。

小心：软件必须正确设置这些位，确保提供给 RTC 的时钟频率低于 4 MHz。在选择 RTC 时钟源之前必须配置这些位（如果需要）。

注意，当 RTCSRC 不为 3 时，会禁止该分频器（不提供时钟）。

0x0：HSE（复位后的默认值）

0x1：HSE/2

0x2：HSE/3

0x3：HSE/4

...

0x3E：HSE/63

0x3F：HSE/64

10.8.53 RCC MPU 时钟分频器寄存器 (RCC_MPCKDIVR)

RCC MPU clock divider register

偏移地址：0x564

复位值：0x8000 0001

如果 RCC_SECCFGR.MPUSEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制 MPU 时钟预分频器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MPUDIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPUDIV[3:0]			
												rw	rw	rw	rw

位 31 **MPUDIVRDY**：MPU 子系统时钟分频器状态 (MPU sub-system clock divider status)

由硬件置 1 和复位，用来指示是否考虑了新的分频系数。

该位仅在 MPUSRC[1:0] =“11”时有效。

0：尚未考虑新的分频系数。

1：已考虑新的分频系数。（复位后的默认值）

位 30:4 保留，必须保持复位值。

位 3:0 **MPUDIV[3:0]**：MPU 内核时钟分频器 (MPU Core clock divider)

由软件写入，用于控制当 MPUSRC 设置为位置 3 时的 MPU 时钟分频系数。

可以实时更改此分频比。它仅影响 MPU 时钟的频率。在 MPUDIV 更新后，时钟将以新的预分频系数（1 到 16 个 pll1_p_ck 时钟周期）进行分频。应用可通过回读 MPUDIVRDY 标志来检查是否将新分频系数考虑在内。请注意，如果 MPUSRC[1:0] 不为“11”，将禁止该分频器。

0x0：禁止 MPUDIV；即不产生时钟

0x1：mpuss_ck 为 pll1_p_ck 时钟 2 分频（复位后的默认值）

0x2：mpuss_ck 为 pll1_p_ck 时钟 3 分频

0x<n>：mpuss_ck 为 pll1_p_ck 时钟 0x<n>+1 分频

0xF：mpuss_ck 为 pll1_p_ck 时钟 16 分频

10.8.54 RCC AXI 时钟分频器寄存器 (RCC_AXIDIVR)

RCC AXI clock divider register

偏移地址：0x568

复位值：0x8000 0000

如果 RCC_SECCFGR.AXISEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制 AXI 矩阵时钟预分频器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AXIDIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXIDIV[2:0]		
													rw	rw	rw

位 31 **AXIDIVRDY**：AXI 子系统时钟分频器状态 (AXI sub-system clock divider status)

由硬件置 1 和复位，用来指示是否考虑了新的分频系数。

0：尚未考虑新的分频系数。

1：已考虑新的分频系数。（复位后的默认值）

位 30:3 保留，必须保持复位值。

位 2:0 **AXIDIV[2:0]**：AXI、AHB5 和 AHB6 时钟分频器 (AXI, AHB5 and AHB6 clock divider)

由软件写入，用来控制 AXI、AHB5 和 AHB6 的时钟分频系数。

可以实时更改此分频比。它影响 AXI、APB4、APB5、AHB5 及 AHB6 时钟的频率。

在 AXIDIV 更新后，时钟将以新的预分频系数（1 到 4 个 axiss_ck 时钟周期）进行分频。应用可通过回读 AXIDIVRDY 标志来检查是否将新分频系数考虑在内。

0x0：axiss_ck（复位后的默认值）

0x1：axiss_ck/2

0x2：axiss_ck/3

其他值：axiss_ck/4

10.8.55 RCC MLAHB 时钟分频器寄存器 (RCC_MLAHBDIVR)

RCC MLAHB clock divider register

偏移地址：0x56C

复位值：0x8000 0000

如果 RCC_SECCFGR.MLAHBSEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制 MLAHB 子系统时钟预分频器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MLAHBDIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MLAHBDIV[3:0]			
												rw	rw	rw	rw

位 31 **MLAHBDIVRDY**：MLAHB 时钟预分频器状态 (MLAHB clock prescaler status)

由硬件置 1 和复位，用来指示是否考虑了新的分频系数。

0：尚未考虑新的分频系数。

1：已考虑新的分频系数。（复位后的默认值）

位 30:4 保留，必须保持复位值。

位 3:0 **MLAHBDIV[3:0]**：MLAHB 时钟分频器 (MLAHB clock divider)

由软件写入，用来控制 MLAHB 时钟的分频系数。

更改此分频比会影响 MLAHB 时钟以及所有总线矩阵时钟的频率。在 MLAHBDIV 更新后，时钟将以新的预分频系数（1 到 512 个 mlahbss_ck 时钟周期）进行分频。应用可通过回读 MLAHBDIVRDY 标志来检查是否将新分频系数考虑在内。

0x0：mlahbss_ck 未分频（复位后的默认值）

0x1：mlahbss_ck 2 分频

0x2：mlahbss_ck 4 分频

0x3：mlahbss_ck 8 分频

0x4：mlahbss_ck 16 分频

0x5：mlahbss_ck 32 分频

0x6：mlahbss_ck 64 分频

0x7：mlahbss_ck 128 分频

0x8：mlahbss_ck 256 分频

其他值：mlahbss_ck 512 分频

10.8.56 RCC APB1 时钟分频器寄存器 (RCC_APB1DIVR)

RCC APB1 clock divider register

偏移地址：0x570

复位值：0x8000 0000

该寄存器用于控制 APB1 时钟预分频器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
APB1DIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APB1DIV[2:0]		
													rw	rw	rw

位 31 **APB1DIVRDY**：APB1 时钟预分频器状态 (APB1 clock prescaler status)

由硬件置 1 和复位，用来指示是否考虑了新的分频系数。

0：尚未考虑新的分频系数。

1：已考虑新的分频系数。（复位后的默认值）

位 30:3 保留，必须保持复位值。

位 2:0 **APB1DIV[2:0]**：APB1 时钟分频器 (APB1 clock divider)

由软件写入，用来控制 APB1 时钟的分频系数。

在 APB1DIV 更新后，APB1 外设时钟将以新的预分频系数（1 到 16 个 mlhclk 时钟周期）进行分频。应用可通过回读 APB1DIVRDY 标志来检查是否将新分频系数考虑在内。

0x0：mlhclk（复位后的默认值）

0x1：mlhclk/2

0x2：mlhclk/4

0x3：mlhclk/8

0x4：mlhclk/16

其他值：不允许

10.8.57 RCC APB2 时钟分频器寄存器 (RCC_APB2DIVR)

RCC APB2 clock divider register

偏移地址：0x574

复位值：0x8000 0000

该寄存器用于控制 APB2 时钟预分频器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
APB2DIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APB2DIV[2:0]		
													rw	rw	rw

位 31 **APB2DIVRDY**：APB2 时钟预分频器状态 (APB2 clock prescaler status)

由硬件置 1 和复位，用来指示是否考虑了新的分频系数。

0：尚未考虑新的分频系数。

1：已考虑新的分频系数。（复位后的默认值）

位 30:3 保留，必须保持复位值。

位 2:0 **APB2DIV[2:0]**：APB2 时钟分频器 (APB2 clock divider)

由软件写入，用来控制 APB2 时钟的分频系数。

在 APB2DIV 更新后，APB2 时钟将以新的预分频系数（1 到 16 个 mlhclk 时钟周期）进行分频。应用可通过回读 APB2DIVRDY 标志来检查是否将新分频系数考虑在内。

0x0：mlhclk（复位后的默认值）

0x1：mlhclk/2

0x2：mlhclk/4

0x3：mlhclk/8

0x4：mlhclk/16

其他值：不允许

10.8.58 RCC APB3 时钟分频器寄存器 (RCC_APB3DIVR)

RCC APB3 clock divider register

偏移地址：0x578

复位值：0x8000 0000

如果 RCC_SECCFGR.APB3DIVSEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制 APB3 时钟预分频器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
APB3DIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APB3DIV[2:0]		
													rw	rw	rw

位 31 **APB3DIVRDY**：APB3 时钟预分频器状态 (APB3 clock prescaler status)

由硬件置 1 和复位，用来指示是否考虑了新的分频系数。

0：尚未考虑新的分频系数。

1：已考虑新的分频系数。（复位后的默认值）

位 30:3 保留，必须保持复位值。

位 2:0 **APB3DIV[2:0]**：APB3 时钟分频器 (APB3 clock divider)

由软件写入，用来控制 APB3 时钟的分频系数。

在 APB3DIV 更新后，APB3 时钟将以新的预分频系数（1 到 16 个 hclk 时钟周期）进行分频。应用可通过回读 APB3DIVRDY 标志来检查是否将新分频系数考虑在内。

0x0：hclk（复位后的默认值）

0x1：hclk / 2

0x2：hclk / 4

0x3：hclk / 8

其他值：hclk/16

10.8.59 RCC APB4 时钟分频寄存器 (RCC_APB4DIVR)

RCC APB4 clock divider register

偏移地址：0x57C

复位值：0x8000 0000

如果 RCC_SECCFGR.APB4DIVSEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制 APB4 时钟分频器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
APB4DIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APB4DIV[2:0]		
													rw	rw	rw

位 31 **APB4DIVRDY**：APB4 时钟分频器状态 (APB4 clock divider status)

由硬件置 1 和复位，用来指示是否考虑了新的分频系数。

0：尚未考虑新的分频系数。

1：已考虑新的分频系数。（复位后的默认值）

位 30:3 保留，必须保持复位值。

位 2:0 **APB4DIV[2:0]**：APB4 时钟分频器 (APB4 clock divider)

由软件写入，用来控制 APB4 时钟的分频系数。

可以实时更改此分频比。它仅影响 APB4 时钟的频率。

在 APB4DIV 更新后，时钟将以新的预分频系数（1 到 16 个 aclk 时钟周期）进行分频。应用可通过回读 APB4DIVRDY 标志来检查是否将新分频系数考虑在内。

0x0：aclk（复位后的默认值）

0x1：aclk / 2

0x2：aclk/4

0x3：aclk/8

其他值：aclk/16

10.8.60 RCC APB5 时钟分频器寄存器 (RCC_APB5DIVR)

RCC APB5 clock divider register

偏移地址：0x580

复位值：0x8000 0000

如果 RCC_SECCFGR.APB5DIVSEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制 APB5 时钟分频器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
APB5DIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APB5DIV[2:0]		
													rw	rw	rw

位 31 **APB5DIVRDY**：APB5 时钟分频器状态 (APB5 clock divider status)

由硬件置 1 和复位，用来指示是否考虑了新的分频系数。

0：尚未考虑新的分频系数。

1：已考虑新的分频系数。（复位后的默认值）

位 30:3 保留，必须保持复位值。

位 2:0 **APB5DIV[2:0]**：APB5 时钟分频器 (APB5 clock divider)

由软件写入，用来控制 APB5 时钟的分频系数。

可以实时更改此分频比。它仅影响 APB5 时钟的频率。

在 APB5DIV 更新后，时钟将以新的预分频系数（1 到 16 个 aclk 时钟周期）进行分频。应用可通过读取 APB5DIVRDY 标志来检查是否将新分频系数考虑在内。

0x0：aclk（复位后的默认值）

0x1：aclk / 2

0x2：aclk/4

0x3：aclk/8

其他值：aclk/16

10.8.61 RCC APB6 时钟分频器寄存器 (RCC_APB6DIVR)

RCC APB6 clock divider register

偏移地址：0x584

复位值：0x8000 0000

如果 RCC_SECCFGR.APB6DIVSEC=1，该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制 APB6 时钟分频器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
APB6DIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APB6DIV[2:0]		
													rw	rw	rw

位 31 **APB6DIVRDY**：APB6 时钟分频器状态 (APB6 clock divider status)

由硬件置 1 和复位，用来指示是否考虑了新的分频系数。

0：尚未考虑新的分频系数。

1：已考虑新的分频系数。（复位后的默认值）

位 30:3 保留，必须保持复位值。

位 2:0 **APB6DIV[2:0]**：APB6 时钟分频器 (APB6 clock divider)

由软件写入，用来控制 APB6 时钟的分频系数。

可以实时更改此分频比。它仅影响 APB6 时钟的频率。

在 APB6DIV 更新后，时钟将以新的预分频系数（1 到 16 个 mlhclk 时钟周期）进行分频。应用可通过读取 APB6DIVRDY 标志来检查是否将新分频系数考虑在内。

0x0：mlhclk（复位后的默认值）

0x1：mlhclk/2

0x2：mlhclk/4

0x3：mlhclk/8

其他值：mlhclk/16

10.8.62 RCC APB1 定时器组 1 预分频器寄存器 (RCC_TIMG1PRER)

RCC APB1 timers group1 prescaler register

偏移地址：0x5A0

复位值：0x8000 0000

该寄存器用于控制位于 APB1 域内的定时器的预分频值。其涉及组 1 定时器：TIM2、TIM3、TIM4、TIM5、TIM6 和 TIM7。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMG1PRERDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMG1PRE
															rw

位 31 **TIMG1PRERDY**：定时器时钟预分频器状态 (Timers clocks prescalers status)
由硬件置 1 和复位，用来指示是否考虑了新的分频系数。
0：尚未考虑新的分频系数。
1：考虑新的分频系数（复位后的默认值）。

位 30:1 保留，必须保持复位值。

位 0 **TIMG1PRE**：定时器时钟预分频器选择 (Timers clocks prescaler selection)
该位由软件置 1 和复位，用于控制连接到 APB1 域的所有定时器的时钟频率。
请参见*表 58：时钟定时器与 pclk 之比*。
0：如果 APB1DIV 对应于 1 或 2 分频，则定时器内核时钟为 mlhclk；否则为 2 x Fpclk1（复位后的默认值）
1：如果 APB1DIV 对应于 1、2 或 4 分频，则定时器内核时钟为 mlhclk；否则为 4 x Fpclk1

10.8.63 RCC APB2 定时器组 2 预分频器寄存器 (RCC_TIMG2PRER)

RCC APB2 timers group2 prescaler register

偏移地址：0x5A4

复位值：0x8000 0000

该寄存器用于控制位于 APB2 域内的定时器的预分频值。其涉及组 2 定时器：TIM1 和 TIM8。其他相关信息，请参见*子系统时钟生成*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMG2PRERDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMG2PRE
															rw

位 31 **TIMG2PRERDY**：定时器时钟预分频器状态 (Timers clocks prescalers status)
由硬件置 1 和复位，用来指示是否考虑了新的分频系数。
0：尚未考虑新的分频系数。
1：考虑新的分频系数（复位后的默认值）。

位 30:1 保留，必须保持复位值。

位 0 **TIMG2PRE**：定时器时钟预分频器选择 (Timers clocks prescaler selection)
该位由软件置 1 和复位，用于控制连接到 APB2 域的所有定时器的时钟频率。
请参见*表 58：时钟定时器与 pclk 之比*。
0：如果 APB2DIV 对应于 1 或 2 分频，则定时器内核时钟为 mlhclk；否则为 2 x Fpclk2（复位后的默认值）
1：如果 APB2DIV 对应于 1、2 或 4 分频，则定时器内核时钟为 mlhclk；否则为 4 x Fpclk2

10.8.64 RCC APB6 定时器组 3 预分频器寄存器 (RCC_TIMG3PRER)

RCC APB6 timers group3 prescaler register

偏移地址：0x5A8

复位值：0x8000 0000

如果 RCC_SECCFGR.TIMG3SEC=1，则该寄存器的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制位于 APB6 域内可能安全的定时器的预分频值。其涉及组 3 定时器：TIM12、TIM13、TIM14、TIM15、TIM16 和 TIM17。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMG3PRERDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMG3PRE
															rw

位 31 **TIMG3PRERDY**：定时器时钟预分频器状态 (Timers clocks prescalers status)

由硬件置 1 和复位，用来指示是否考虑了新的分频系数。

0：尚未考虑新的分频系数。

1：考虑新的分频系数（复位后的默认值）。

位 30:1 保留，必须保持复位值。

位 0 **TIMG3PRE**：定时器时钟预分频器选择 (Timers clocks prescaler selection)

该位由软件置 1 和复位，用于控制连接到 APB6 域的所有定时器的时钟频率。

请参见*表 58：时钟定时器与 pclk 之比*。

0：如果 APB6DIV 对应于 1 或 2 分频，则定时器内核时钟为 mlhclk；否则为 2 x Fpclk6（复位后的默认值）

1：如果 APB6DIV 对应于 1、2 或 4 分频，则定时器内核时钟为 mlhclk；否则为 4 x Fpclk6

10.8.65 RCC DDR 接口控制寄存器 (RCC_DDRITFCR)

RCC DDR interface control register

偏移地址：0x5C0

复位值：0x000F D022

该寄存器根据 DDR 接口的安全状态（即根据 ETZPC 寄存器相应的 DECPROT[1] 位字段）而受到写保护。

该寄存器用于控制 DDR 接口，包括 DDRC 和 DDRPHYC。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GSKP_DUR[3:0]				DFILP_WIDTH[2:0]			GSKPCTRL	GSKPMOD	DDRCKMOD[2:0]			DPHYCTLRST	DPHYRST	DPHYAPBRST	DDRCORERST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDRCAXIRST	DDRCAPBRST	KERDCG_DLY[2:0]			DDRPHYCAPBLPEN	DDRPHYCAPBEN	AXIDCGEN	DDRCAPBLPEN	DDRCAPBEN	DDRPHYCLPEN	DDRPHYCEN	Res.	Res.	DDRC1LPEN	DDRC1EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw

位 31:28 **GSKP_DUR[3:0]**：自动模式下的噪声跳过器持续时间 (Glitch skipper duration in automatic mode)

如果 DDRC 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

该位字段允许应用定义噪声跳过器向 DDRC 模块提供时钟 dphy_ker_ck（而非时钟 ddrc_ker_ck）的时间量。该功能仅在噪声跳过器设置为自动模式（GSKPMOD =“1”）时有效。

该延迟等于 (GSKP_DUR + 1) x 32 x Tdphy_ker_ck

0x0：设置一个 32 x Tdphy_ker_ck 的延迟

0x1：设置一个 2 x 32 x Tdphy_ker_ck 的延迟

0x2：设置一个 3 x 32 x Tdphy_ker_ck 的延迟

...

0xF：设置一个 16 x 32 x Tdphy_ker_ck 的延迟

位 27:25 **DFILP_WIDTH[2:0]**：低功耗请求命令的最短持续时间 (Minimum duration of low-power request command)

如果 DDRC 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

低功耗请求持续时间必须始终大于或等于 1 微秒。如果不遵守该值，将无法保证 DDR 接口的行为。应用必须正确编程 DFILP_WIDTH 以确保延迟大于或等于 1 微秒。请注意，该延迟还取决于 dphy_ker_ck 时钟的频率。

0x0：旁路，禁止延迟

0x1：当 Fdphy_ker_ck 在 120 到 160 MHz 之间时，强制使用 160 x Tdphy_ker_ck 的延迟

0x2：当 Fdphy_ker_ck 在 160 到 220 MHz 之间时，强制使用 224 x Tdphy_ker_ck 的延迟

0x3：当 Fdphy_ker_ck 在 220 到 320 MHz 之间时，强制使用 320 x Tdphy_ker_ck 的延迟

0x4：当 Fdphy_ker_ck 在 320 到 410 MHz 之间时，强制使用 416 x Tdphy_ker_ck 的延迟

其他值：当 Fdphy_ker_ck 在 410 到 533 MHz 之间时，强制使用 544 x Tdphy_ker_ck 的延迟

位 24 **GSKPCTRL**：噪声跳过器 (GSKP) 控制 (Glitch skipper (GSKP) control)

如果 DDRC 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

仅当 GSKPMOD =“0”时有效。

0：GSKP 模块提供时钟 phy_out_ck（由 DDRPHYC 提供）

1：GSKP 模块提供时钟 dphy_ker_ck（由 RCC 提供）

位 23 **GSKPMOD**：噪声跳过器 (GSKP) 模式 (Glitch Skipper (GSKP) Mode)

如果 DDRC 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

0：GSKP 模块由 GSKPCTRL 位控制

1：GSKP 模块由 DFI 自动控制

位 22:20 **DDRCKMOD[2:0]**：用于 DDR 时钟控制的 RCC 模式 (RCC mode for DDR clock control)

如果 DDRC 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：正常模式：dphy_ker_ck 时钟的门控取决于 DDRPHYCEN、DDRPHYCLPEN 和 MPU 模式。ddrc_ker_ckg 时钟的门控取决于 DDRCxEN、DDRCxLPEN 和 MPU 模式。
在 DDRC 和 DDRPHYC 初始化阶段，如果应用使用软件自刷新 (SSR)，则必须选择该模式。

0x1：自动自刷新模式 (ASR1)：将根据 cactive_ddrc 信号自动门控 ddrc_ker_ckg 时钟。dphy_ker_ck 时钟的门控取决于 DDRPHYCEN、DDRPHYCLPEN 和 MPU 模式。
当应用想要保持 DLL 的时钟，可以使用该模式，以避免重新锁定序列。在该模式下，推荐设置 DDRCxEN = DDRCxLPEN = DDRPHYCEN = DDRPHYCLPEN =“1”。

0x5：全自动自刷新模式 (ASR2)：将根据 cactive_ddrc 信号自动门控 ddrc_ker_ckg 和 dphy_ker_ck 时钟。
当 DLL 处于旁路状态时，可以使用该模式。在该模式下，推荐设置 DDRCxEN = DDRCxLPEN = DDRPHYCEN = DDRPHYCLPEN =“1”。

0x2：硬件自刷新模式 (HSR1)：ddrc_ker_ckg 时钟的门控由连接到 DDRC 的 AXI 低功耗接口控制。dphy_ker_ck 时钟的门控取决于 DDRPHYCEN、DDRPHYCLPEN 和 MPU 模式。
当应用想要保持 DLL 的时钟时，可以使用该模式。在该模式下，推荐设置 DDRCxEN = DDRCxLPEN = DDRPHYCEN = DDRPHYCLPEN =“1”。

0x6：全硬件自刷新模式 (HSR2)：ddrc_ker_ckg 和 dphy_ker_ck 时钟的门控由连接到 DDRC 的 AXI 低功耗接口控制。
当 DLL 处于旁路状态时，可以使用该模式。在该模式下，推荐设置 DDRCxEN = DDRCxLPEN = DDRPHYCEN = DDRPHYCLPEN =“1”。

其他值：保留

位 19 **DPHYCTLRST**：DDRPHYC 控制复位 (DDRPHYC Control reset)

如果通过 ETZPC 寄存器相应的 DECPROT[1] 位字段将 DDRPHYC 模块配置为安全，则对该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

0：不复位 DDRPHYC 控制

1：复位 DDRPHYC 控制

位 18 **DPHYRST**：DDRPHYC 复位 (DDRPHYC reset)

如果通过 ETZPC 寄存器相应的 DECPROT[1] 位字段将 DDRPHYC 模块配置为安全，则对该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

0：不复位 DDRPHYC

1：复位 DDRPHYC

位 17 **DPHYAPBRST**：DDRPHYC APB 接口复位 (DDRPHYC APB interface reset)

如果通过 ETZPC 寄存器相应的 DECPROT[1] 位字段将 DDRPHYC 模块配置为安全，则对该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

0：不复位 DDRPHYC APB 接口

1：复位 DDRPHYC APB 接口

位 16 **DDRCORERST**：DDRC 内核复位 (DDRC core reset)

如果 DDRC 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

0：不复位 DDRC 内核

1：复位 DDRC 内核

位 15 **DDRCAXIRST**：DDRC AXI 接口复位 (DDRC AXI interface reset)

如果 DDRC 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

0：不复位 DDRC AXI 接口

1：复位 DDRC AXI 接口

位 14 **DDRCAPBRST**：DDRC APB 接口复位 (DDRC APB interface reset)

如果 DDRC 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

0：不复位 DDRC APB 接口

1：复位 DDRC APB 接口

位 13:11 **KERDCG_DLY[2:0]**：AXIDCG 延迟 (AXIDCG delay)

如果 DDRC 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

它表示从 cactive_ddrc 信号下降沿到 KERDCG 门控 ddrc_ker_ckg 时所间隔的延迟。

0x0：从 cactive_ddrc 下降沿到 ddrc_ker_ckg 门控时，间隔 1 个 ddrc_ker_ck 周期。

0x1：从 cactive_ddrc 下降沿到 ddrc_ker_ckg 门控时，间隔 3 个 ddrc_ker_ck 周期。

...

0x7：从 cactive_ddrc 下降沿到 ddrc_ker_ckg 门控时，间隔 15 个 ddrc_ker_ck 周期

位 10 **DDRPHYCAPBLPEN**：CSleep 模式期间的 DDRPHYC APB 时钟使能 (DDRPHYC APB clock enable during CSleep mode)

如果 DDRC 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

该位用于控制 CSleep 模式下 pclk4_dphy 时钟的门控。

0：表示在 CSleep 模式下禁止 APB 时钟

1：表示在 CSleep 模式下使能 APB 时钟

位 9 **DDRPHYCAPBEN**：DDRPHYC APB 时钟使能 (DDRPHYC APB clock enable)

如果 DDRC 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

该位用于控制 pclk4_dphy 时钟的门控。

0：表示禁止 APB 时钟

1：表示使能 APB 时钟

位 8 **AXIDCGEN**：MPU CRun 模式期间 AXIDCG 使能 (AXIDCG enable during MPU CRun mode)

如果 DDRC 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

当禁止动态时钟门控时，如果时钟是由 RCC 提供，则始终提供给 DDRC。

0：表示在 MPU CRun 期间禁止 AXIDCG[2:1] 的动态时钟门控，

1：表示在 MPU CRun 期间使能 AXIDCG[2:1] 的动态时钟门控

位 7 **DDRCAPBLPEN**：CSleep 模式期间的 DDRC APB 时钟使能 (DDRC APB clock enable during CSleep mode)

如果 DDRC 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

该位用于控制 CSleep 模式下 pclk4_ddrc 时钟的门控。

0：表示在 CSleep 模式下禁止 APB 时钟

1：表示在 CSleep 模式下使能 APB 时钟

位 6 **DDRCAPBEN**：DDRC APB 时钟使能 (DDRC APB clock enable)

如果 DDRC 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

该位用于控制 pclk4_ddrc 时钟的门控。

0：表示禁止 APB 时钟

1：表示使能 APB 时钟

位 5 **DDRPHYCLPEN**：CSleep 模式期间的 DDRPHYC 外设时钟使能 (DDRPHYC peripheral clocks enable during CSleep mode)

如果通过 ETZPC 寄存器相应的 DECPROT[1] 位字段将 DDRPHYC 模块配置为安全，则对该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

该位用于控制 CSleep 模式下 dphy_ker_ck 和 pclk4_dphy 的门控

0：表示在 CSleep 模式下禁止外设时钟

1：表示在 CSleep 模式下使能外设时钟

位 4 **DDRPHYCEN**：DDRPHYC 外设时钟使能 (DDRPHYC peripheral clocks enable)

如果通过 ETZPC 寄存器相应的 DECPROT[1] 位字段将 DDRPHYC 模块配置为安全，则对该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

该位用于控制 dphy_ker_ck 和 pclk4_dphy 的门控

0：表示禁止外设时钟

1：表示使能外设时钟

位 3:2 保留，必须保持复位值。

位 1 **DDRC1LPEN**：CSleep 模式期间的 DDRC 端口 1 外设时钟使能 (DDRC port 1 peripheral clocks enable during CSleep mode)

如果 DDRC 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

该位用于控制 CSleep 模式下 ddrc_ker_ckg、aclk1_ddrc 和 pclk4_ddrc 的门控。

0：表示在 CSleep 模式下禁止外设时钟

1：表示在 CSleep 模式下使能外设时钟

位 0 **DDRC1EN**：DDRC 端口 1 外设时钟使能 (DDRC port 1 peripheral clocks enable)

如果 DDRC 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1 和清零。

该位用于控制 ddrc_ker_ckg、aclk1_ddrc 和 pclk4_ddrc 时钟的门控。

0：表示禁止 DDRC 外设时钟

1：表示使能 DDRC 外设时钟

10.8.66 RCC I2C1 和 I2C2 内核时钟选择寄存器 (RCC_I2C12CKSELR)

RCC I2C1 and I2C2 kernel clock selection register

偏移地址：0x600

复位值：0x0000 0000

该寄存器用于控制 APB1 I2C1 和 I2C2 的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C12SRC[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **I2C12SRC[2:0]**：I2C1 和 I2C2 内核时钟源选择 (I2C1 and I2C2 kernel clock source selection)
由软件写入。
0x0：选择 pclk1 时钟作为内核外设时钟（复位后的默认值）
0x1：选择 pll4_r_ck 时钟作为内核外设时钟
0x2：选择 hsi_ker_ck 时钟作为内核外设时钟
0x3：选择 csi_ker_ck 时钟作为内核外设时钟
其他：保留，禁止内核时钟

10.8.67 RCC I2C3、I2C4 和 I2C5 内核时钟选择寄存器 (RCC_I2C345CKSELR)

RCC I2C3, I2C4 and I2C5 kernel clock selection register

偏移地址：0x604

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于控制 (APB6) I2C3、I2C4 和 I2C5 的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C5SRC[2:0]			I2C4SRC[2:0]			I2C3SRC[2:0]		
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:9 保留，必须保持复位值。

位 8:6 **I2C5SRC[2:0]**：I2C5 内核时钟源选择 (I2C5 kernel clock source selection)

如果 I2C5 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：选择 pclk6 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll4_r_ck 时钟作为内核外设时钟

0x2：选择 hsi_ker_ck 时钟作为内核外设时钟

0x3：选择 csi_ker_ck 时钟作为内核外设时钟

其他：保留，禁止内核时钟

位 5:3 **I2C4SRC[2:0]**：I2C4 内核时钟源选择 (I2C4 kernel clock source selection)

如果 I2C4 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：选择 pclk6 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll4_r_ck 时钟作为内核外设时钟

0x2：选择 hsi_ker_ck 时钟作为内核外设时钟

0x3：选择 csi_ker_ck 时钟作为内核外设时钟

其他：保留，禁止内核时钟

位 2:0 **I2C3SRC[2:0]**：I2C3 内核时钟源选择 (I2C3 kernel clock source selection)

如果 I2C3 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：选择 pclk6 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll4_r_ck 时钟作为内核外设时钟

0x2：选择 hsi_ker_ck 时钟作为内核外设时钟

0x3：选择 csi_ker_ck 时钟作为内核外设时钟

其他：保留，禁止内核时钟

10.8.68 RCC SPI/I2S1 内核时钟选择寄存器 (RCC_SPI2S1CKSELR)

RCC SPI/I2S1 kernel clock selection register

偏移地址：0x608

复位值：0x0000 0000

该寄存器用于控制 APB2 SPI/I2S1 的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPI1SRC[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **SPI1SRC[2:0]**：SPI/I2S1 内核时钟源选择 (SPI/I2S1 kernel clock source selection)
由软件写入。
0x0：选择 pll4_p_ck 时钟作为内核外设时钟（复位后的默认值）
0x1：选择 pll3_q_ck 时钟作为内核外设时钟
0x2：选择 I2S_CKIN 时钟作为内核外设时钟
0x3：选择 per_ck 时钟作为内核外设时钟
0x4：选择 pll3_r_ck 时钟作为内核外设时钟
其他：保留，禁止内核时钟

10.8.69 RCC SPI/I2S2 和 SPI/I2S3 内核时钟选择寄存器 (RCC_SPI2S23CKSELR)

RCC SPI/I2S2 and SPI/I2S3 kernel clock selection register

偏移地址：0x60C

复位值：0x0000 0000

该寄存器用于控制 APB1 SPI/I2S2 和 SPI/I2S3 的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPI23SRC[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **SPI23SRC[2:0]**：SPI/I2S2 和 SPI/I2S3 内核时钟源选择 (SPI/I2S2 and SPI/I2S3 kernel clock source selection)

由软件写入。

0x0：选择 pll4_p_ck 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll3_q_ck 时钟作为内核外设时钟

0x2：选择 I2S_CKIN 时钟作为内核外设时钟

0x3：选择 per_ck 时钟作为内核外设时钟

0x4：选择 pll3_r_ck 时钟作为内核外设时钟

其他：保留，禁止内核时钟

10.8.70 RCC SPI/I2S4 和 SPI5 内核时钟选择寄存器 (RCC_SPI45CKSELR)

RCC SPI/I2S4 and SPI5 kernel clock selection register

偏移地址：0x610

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于控制 (APB6) SPI/I2S4 和 SPI5 的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPI5SRC[2:0]			SPI4SRC[2:0]		
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持复位值。

位 5:3 **SPI5SRC[2:0]**：SPI5 内核时钟源选择 (SPI5 kernel clock source selection)

如果 SPI5 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：选择 pclk6 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll4_q_ck 时钟作为内核外设时钟

0x2：选择 hsi_ker_ck 时钟作为内核外设时钟

0x3：选择 csi_ker_ck 时钟作为内核外设时钟

0x4：选择 hse_ker_ck 时钟作为内核外设时钟

其他：保留，禁止内核时钟

位 2:0 **SPI4SRC[2:0]**：SPI/I2S4 内核时钟源选择 (SPI/I2S4 kernel clock source selection)

如果 SPI/I2S4 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：选择 pclk6 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll4_q_ck 时钟作为内核外设时钟

0x2：选择 hsi_ker_ck 时钟作为内核外设时钟

0x3：选择 csi_ker_ck 时钟作为内核外设时钟

0x4：选择 hse_ker_ck 时钟作为内核外设时钟

0x5：选择 I2S_CKIN 时钟作为内核外设时钟

其他：保留，禁止内核时钟

10.8.71 RCC USART1 和 USART2 内核时钟选择寄存器 (RCC_UART12CKSELR)

RCC USART1 and USART2 kernel clock selection register

偏移地址：0x614

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于控制 (APB6) USART1 和 USART2 的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UART2SRC[2:0]			UART1SRC[2:0]		
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持复位值。

位 5:3 **UART2SRC[2:0]**：USART2 内核时钟源选择 (USART2 kernel clock source selection)

如果 USART2 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：选择 pclk6 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll3_q_ck 时钟作为内核外设时钟

0x2：选择 hsi_ker_ck 时钟作为内核外设时钟

0x3：选择 csi_ker_ck 时钟作为内核外设时钟

0x4：选择 pll4_q_ck 时钟作为内核外设时钟

0x5：选择 hse_ker_ck 时钟作为内核外设时钟

其他：保留，禁止内核时钟

位 2:0 **UART1SRC[2:0]**：USART1 内核时钟源选择 (USART1 kernel clock source selection)

如果 USART1 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：选择 pclk6 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll3_q_ck 时钟作为内核外设时钟

0x2：选择 hsi_ker_ck 时钟作为内核外设时钟

0x3：选择 csi_ker_ck 时钟作为内核外设时钟

0x4：选择 pll4_q_ck 时钟作为内核外设时钟

0x5：选择 hse_ker_ck 时钟作为内核外设时钟

其他：保留，禁止内核时钟

10.8.72 RCC USART3 和 UART5 内核时钟选择寄存器 (RCC_UART35CKSELR)

RCC USART3 and UART5 kernel clock selection register

偏移地址：0x618

复位值：0x0000 0000

该寄存器用于控制 APB1 USART3 和 UART5 的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UART35SRC[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **UART35SRC[2:0]**：USART3 和 UART5 内核时钟源选择 (USART3 and UART5 kernel clock source selection)

由软件写入。

0x0：选择 pclk1 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll4_q_ck 时钟作为内核外设时钟

0x2：选择 hsi_ker_ck 时钟作为内核外设时钟

0x3：选择 csi_ker_ck 时钟作为内核外设时钟

0x4：选择 hse_ker_ck 时钟作为内核外设时钟

其他：保留，禁止内核时钟

10.8.73 RCC UART4 内核时钟选择寄存器 (RCC_UART4CKSELR)

RCC UART4 kernel clock selection register

偏移地址：0x61C

复位值：0x0000 0000

该寄存器用于控制 APB1 UART4 的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UART4SRC[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **UART4SRC[2:0]**：UART4 内核时钟源选择 (UART4 kernel clock source selection)

由软件写入。

0x0：选择 pclk1 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll4_q_ck 时钟作为内核外设时钟

0x2：选择 hsi_ker_ck 时钟作为内核外设时钟

0x3：选择 csi_ker_ck 时钟作为内核外设时钟

0x4：选择 hse_ker_ck 时钟作为内核外设时钟

其他：保留，禁止内核时钟

10.8.74 RCC USART6 内核时钟选择寄存器 (RCC_UART6CKSELR)

RCC USART6 kernel clock selection register

偏移地址：0x620

复位值：0x0000 0000

该寄存器用于控制 APB2 USART6 的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UART6SRC[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **UART6SRC[2:0]**：USART6 内核时钟源选择 (USART6 kernel clock source selection)

由软件写入。

0x0：选择 pclk2 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll4_q_ck 时钟作为内核外设时钟

0x2：选择 hsi_ker_ck 时钟作为内核外设时钟

0x3：选择 csi_ker_ck 时钟作为内核外设时钟

0x4：选择 hse_ker_ck 时钟作为内核外设时钟

其他：保留，禁止内核时钟

10.8.75 RCC UART7 和 UART8 内核时钟选择寄存器 (RCC_UART78CKSELR)

RCC UART7 and UART8 kernel clock selection register

偏移地址：0x624

复位值：0x0000 0000

该寄存器用于控制 APB1 UART7 和 UART8 的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UART78SRC[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **UART78SRC[2:0]**：UART7 和 UART8 内核时钟源选择 (UART7 and UART8 kernel clock source selection)

由软件写入。

0x0：选择 pclk1 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll4_q_ck 时钟作为内核外设时钟

0x2：选择 hsi_ker_ck 时钟作为内核外设时钟

0x3：选择 csi_ker_ck 时钟作为内核外设时钟

0x4：选择 hse_ker_ck 时钟作为内核外设时钟

其他：保留，禁止内核时钟

10.8.76 RCC LPTIM1 内核时钟选择寄存器 (RCC_LPTIM1CKSELR)

RCC LPTIM1 kernel clock selection register

偏移地址：0x628

复位值：0x0000 0000

该寄存器用于控制 APB1 LPTIM1 模块的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM1SRC[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **LPTIM1SRC[2:0]**：LPTIM1 内核时钟源选择 (LPTIM1 kernel clock source selection)
由软件写入。
0x0：选择 pclk1 时钟作为内核外设时钟（复位后的默认值）
0x1：选择 pll4_p_ck 时钟作为内核外设时钟
0x2：选择 pll3_q_ck 时钟作为内核外设时钟
0x3：选择 lse_ck 时钟作为内核外设时钟
0x4：选择 lsi_ck 时钟作为内核外设时钟
0x5：选择 per_ck 时钟作为内核外设时钟
其他：禁止内核时钟

10.8.77 RCC LPTIM2 和 LPTIM3 内核时钟选择寄存器 (RCC_LPTIM23CKSELR)

RCC LPTIM2 and LPTIM3 kernel clock selection register

偏移地址：0x62C

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于控制 APB3 LPTIM2 和 LPTIM3 模块的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM3SRC[2:0]			LPTIM2SRC[2:0]		
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持复位值。

位 5:3 **LPTIM3SRC[2:0]**：LPTIM3 内核时钟源选择 (LPTIM3 kernel clock source selection)

如果 LPTIM3 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：选择 pclk3 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll4_q_ck 时钟作为内核外设时钟

0x2：选择 per_ck 时钟作为内核外设时钟

0x3：选择 lse_ck 时钟作为内核外设时钟

0x4：选择 lsi_ck 时钟作为内核外设时钟

其他：禁止内核时钟

位 2:0 **LPTIM2SRC[2:0]**：LPTIM2 内核时钟源选择 (LPTIM2 kernel clock source selection)

如果 LPTIM2 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：选择 pclk3 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll4_q_ck 时钟作为内核外设时钟

0x2：选择 per_ck 时钟作为内核外设时钟

0x3：选择 lse_ck 时钟作为内核外设时钟

0x4：选择 lsi_ck 时钟作为内核外设时钟

其他：禁止内核时钟

10.8.78 RCC LPTIM4 和 LPTIM5 内核时钟选择寄存器 (RCC_LPTIM45CKSELR)

RCC LPTIM4 and LPTIM5 kernel clock selection register

偏移地址：0x630

复位值：0x0000 0000

该寄存器用于控制 LPTIM4 和 LPTIM5 模块的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM45SRC[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **LPTIM45SRC[2:0]**：LPTIM4 和 LPTIM5 内核时钟源选择 (LPTIM4 and LPTIM5 kernel clock source selection)

由软件写入。

0x0：选择 pclk3 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll4_p_ck 时钟作为内核外设时钟

0x2：选择 pll3_q_ck 时钟作为内核外设时钟

0x3：选择 lse_ck 时钟作为内核外设时钟

0x4：选择 lsi_ck 时钟作为内核外设时钟

0x5：选择 per_ck 时钟作为内核外设时钟

其他：禁止内核时钟

10.8.79 RCC SAI1 内核时钟选择寄存器 (RCC_SAI1CKSELR)

RCC SAI1 kernel clock selection register

偏移地址：0x634

复位值：0x0000 0000

该寄存器用于控制 SAI1 和 DFSDM 音频时钟的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI1SRC[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **SAI1SRC[2:0]**：SAI1 和 DFSDM 内核时钟源选择 (SAI1 and DFSDM kernel clock source selection)
由软件写入。
0x0：选择 pll4_q_ck 时钟作为内核外设时钟（复位后的默认值）
0x1：选择 pll3_q_ck 时钟作为内核外设时钟
0x2：选择 I2S_CKIN 时钟作为内核外设时钟
0x3：选择 per_ck 时钟作为内核外设时钟
0x4：选择 pll3_r_ck 时钟作为内核外设时钟
其他：保留，禁止内核时钟

10.8.80 RCC SAI2 内核时钟选择寄存器 (RCC_SAI2CKSELR)

RCC SAI2 kernel clock selection register

偏移地址：0x638

复位值：0x0000 0000

该寄存器用于控制 SAI2 的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI2SRC[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **SAI2SRC[2:0]**：SAI2 内核时钟源选择 (SAI2 kernel clock source selection)
由软件写入。
0x0：选择 pll4_q_ck 时钟作为内核外设时钟（复位后的默认值）
0x1：选择 pll3_q_ck 时钟作为内核外设时钟
0x2：选择 I2S_CKIN 时钟作为内核外设时钟
0x3：选择 per_ck 时钟作为内核外设时钟
0x4：选择来自 SPDIFRX 的 spdif_ck_symb 时钟作为内核外设时钟
0x5：选择 pll3_r_ck 时钟作为内核外设时钟
其他：保留，禁止内核时钟

10.8.81 RCC FDCAN 内核时钟选择寄存器 (RCC_FDCANCKSELR)

RCC FDCAN kernel clock selection register

偏移地址：0x63C

复位值：0x0000 0000

该寄存器用于控制 FDCAN 模块的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANSRC[1:0]	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **FDCANSRC[1:0]**：FDCAN 内核时钟源选择 (FDCAN kernel clock source selection)

由软件写入。

0x0：选择 hse_ker_ck 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll3_q_ck 时钟作为内核外设时钟

0x2：选择 pll4_q_ck 时钟作为内核外设时钟

0x3：选择 pll4_r_ck 时钟作为内核外设时钟

10.8.82 RCC SPDIFRX 内核时钟选择寄存器 (RCC_SPDIFCKSELR)

RCC SPDIFRX kernel clock selection register

偏移地址：0x640

复位值：0x0000 0000

该寄存器用于控制 SPDIFRX 的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPDIFSRC[1:0]	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **SPDIFSRC[1:0]**：SPDIFRX 内核时钟源选择 (SPDIFRX kernel clock source selection)
由软件写入。
0x0：选择 pll4_p_ck 时钟作为内核外设时钟（复位后的默认值）
0x1：选择 pll3_q_ck 时钟作为内核外设时钟
0x2：选择 hsi_ker_ck 时钟作为内核外设时钟
其他：保留，禁止内核时钟

10.8.83 RCC ADC1 和 ADC2 内核时钟选择寄存器 (RCC_ADC12CKSELR)

RCC ADC1 and ADC2 kernel clock selection register

偏移地址：0x644

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于控制 (AHB2) ADC1 和 ADC2 模块的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC2SRC[1:0]		ADC1SRC[1:0]	
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:2 **ADC2SRC[1:0]**：ADC2 内核时钟源选择 (ADC2 kernel clock source selection)

如果 ADC2 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：选择 pll4_r_ck 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 per_ck 时钟作为内核外设时钟

0x2：选择 pll3_q_ck 时钟作为内核外设时钟

其他：禁止内核时钟

位 1:0 **ADC1SRC[1:0]**：ADC1 内核时钟源选择 (ADC1 kernel clock source selection)

如果 ADC1 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：选择 pll4_r_ck 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 per_ck 时钟作为内核外设时钟

0x2：选择 pll3_q_ck 时钟作为内核外设时钟

其他：禁止内核时钟

10.8.84 RCC SDMMC1 和 SDMMC2 内核时钟选择寄存器 (RCC_SDMMC12CKSELR)

RCC SDMMC1 and SDMMC2 kernel clock selection register

偏移地址：0x648

复位值：0x0000 0003

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于控制 (AHB6) SDMMC1 和 SDMMC2 的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC2SRC[2:0]			SDMMC1SRC[2:0]		
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持复位值。

位 5:3 **SDMMC2SRC[2:0]**：SDMMC2 内核时钟源选择 (SDMMC2 kernel clock source selection)

如果 SDMMC2 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：选择 hclk6 时钟作为内核外设时钟

0x1：选择 pll3_r_ck 时钟作为内核外设时钟

0x2：选择 pll4_p_ck 时钟作为内核外设时钟

0x3：选择 hsi_ker_ck 时钟作为内核外设时钟（复位后的默认值）

其他：保留，禁止内核时钟

位 2:0 **SDMMC1SRC[2:0]**：SDMMC1 内核时钟源选择 (SDMMC1 kernel clock source selection)

如果 SDMMC1 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：选择 hclk6 时钟作为内核外设时钟

0x1：选择 pll3_r_ck 时钟作为内核外设时钟

0x2：选择 pll4_p_ck 时钟作为内核外设时钟

0x3：选择 hsi_ker_ck 时钟作为内核外设时钟（复位后的默认值）

其他：保留，禁止内核时钟

10.8.85 RCC ETH1 和 ETH2 内核时钟选择寄存器 (RCC_ETH12CKSELR)

RCC ETH1 and ETH2 kernel clock selection register

偏移地址：0x64C

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于控制 ETH1 和 ETH2 模块的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETH2PTPDIV[3:0]				Res.	Res.	ETH2SRC[1:0]		ETH1PTPDIV[3:0]				Res.	Res.	ETH1SRC[1:0]	
rw	rw	rw	rw			rw	rw	rw	rw	rw	rw			rw	rw

位 31:16 保留，必须保持复位值。

位 15:12 **ETH2PTPDIV[3:0]**：精确时间协议 (PTP) 的 ETH2 时钟分频器 (ETH2 clock divider for Precision Time Protocol (PTP))

如果 ETH2 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入，用来配置分频器值。

0x0：旁路（复位后的默认值）

0x1：2 分频

0x2：3 分频

...

0xF：16 分频

位 11:10 保留，必须保持复位值。

位 9:8 **ETH2SRC[1:0]**：ETH2 内核时钟源选择 (ETH2 kernel clock source selection)

如果 ETH2 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：选择 pll4_p_ck 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll3_q_ck 时钟作为内核外设时钟

其他：禁止内核时钟

位 7:4 **ETH1PTPDIV[3:0]**：精确时间协议 (PTP) 的 ETH1 时钟分频器 (ETH1 clock divider for Precision Time Protocol (PTP))

如果 ETH1 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入，用来配置分频器值。

0x0：旁路（复位后的默认值）

0x1：2 分频

0x2：3 分频

...

0xF：16 分频

位 3:2 保留，必须保持复位值。

位 1:0 **ETH1SRC[1:0]**：ETH1 内核时钟源选择 (ETH1 kernel clock source selection)

如果 ETH1 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：选择 pll4_p_ck 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll3_q_ck 时钟作为内核外设时钟

其他：禁止内核时钟

10.8.86 RCC USB 内核时钟选择寄存器 (RCC_USBCKSELR)

RCC USB kernel clock selection register

偏移地址：0x650

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于控制 USB 主机 (USBH) 和 USB On-The-Go (USBO) 的 USBPHY PLL 的内核时钟选择。它还控制 USBPHY 参考时钟的预分频器。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBOSRC	Res.	Res.	USBPHYSRC[1:0]	
											rw			rw	rw

位 31:5 保留，必须保持复位值。

位 4 **USBOSRC**：USB OTG 内核时钟源选择 (USB OTG kernel clock source selection)

如果 USBO 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0：选择 pll4_r_ck 时钟作为内核外设时钟（复位后的默认值）

1：选择 USB PHY 提供的时钟 (rcc_ck_usbo_48m) 作为内核外设时钟

位 3:2 保留，必须保持复位值。

位 1:0 **USBPHYSRC[1:0]**：USB PHY 内核时钟源选择 (USB PHY kernel clock source selection)

如果 USBPHY 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件写入。

0x0：选择 hse_ker_ck 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll4_r_ck 时钟作为内核外设时钟

0x2：选择 hse_ker_ck/2 时钟作为内核外设时钟

其他：禁止时钟

10.8.87 RCC QSPI 内核时钟选择寄存器 (RCC_QSPICKSELR)

RCC QSPI kernel clock selection register

偏移地址：0x654

复位值：0x0000 0000

如果 (AHB6) QUADSPI 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制 QUADSPI 的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	QSPISRC[1:0]	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **QSPISRC[1:0]**：QUADSPI 内核时钟源选择 (QUADSPI kernel clock source selection)
由软件写入。
0x0：选择 aclk 时钟作为内核外设时钟（复位后的默认值）
0x1：选择 pll3_r_ck 时钟作为内核外设时钟
0x2：选择 pll4_p_ck 时钟作为内核外设时钟
0x3：选择 per_ck 时钟作为内核外设时钟

10.8.88 RCC FMC 内核时钟选择寄存器 (RCC_FMCCKSELR)

RCC FMC kernel clock selection register

偏移地址：0x658

复位值：0x0000 0000

如果 (AHB6) FMC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制 FMC 模块的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FMCSRC[1:0]	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **FMCSRC[1:0]**：FMC 内核时钟源选择 (FMC kernel clock source selection)

由软件写入。

0x0：选择 aclk 时钟作为内核外设时钟（复位后的默认值）

0x1：选择 pll3_r_ck 时钟作为内核外设时钟

0x2：选择 pll4_p_ck 时钟作为内核外设时钟

0x3：选择 per_ck 时钟作为内核外设时钟

10.8.89 RCC RNG1 内核时钟选择寄存器 (RCC_RNG1CKSELR)

RCC RNG1 kernel clock selection register

偏移地址：0x65C

复位值：0x0000 0000

如果 (AHB5) RNG1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制 RNG1 的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNG1SRC[1:0]	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **RNG1SRC[1:0]**：RNG1 内核时钟源选择 (RNG1 kernel clock source selection)
由软件写入。
0x0：选择 csi_ker_ck 时钟作为内核外设时钟（复位后的默认值）
0x1：选择 pll4_r_ck 时钟作为内核外设时钟
0x2：保留
0x3：选择 lsi_ck 时钟作为内核外设时钟

10.8.90 RCC STGEN 时钟选择寄存器 (RCC_STGENCKSELR)

RCC STGEN clock selection register

偏移地址：0x660

复位值：0x0000 0000

如果 (APB5) STGENC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

注： *STGEN 是一个双 APB 接口 IP，具有 STGENC APB5 控制器部分，其可以是安全的，也可以是非安全的，具体取决于相应的 DECPROT[1]。它的另一个 STGENRO APB4 只读接口是非安全的。*

该寄存器用于选择 STGEN 模块的外设时钟。注意，该时钟用于为应用提供时间参考。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENSRC[1:0]	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **STGENSRC[1:0]**：内核时钟的振荡器选择 (Oscillator selection for kernel clock)
由软件置 1 和清零。
0x0：选择了 hsi_ker_ck 时钟（复位后的默认值）
0x1：选择了 hse_ker_ck 时钟
其他值：禁止时钟

10.8.91 RCC DCMIPP 内核时钟选择寄存器 (RCC_DCMIPPCKSELR)

RCC DCMIPP kernel clock selection register

偏移地址：0x664

复位值：0x0000 0000

如果 (APB4) DCMIPP 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制 DCMIPP 模块的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCMIPPSRC[1:0]	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **DCMIPPSRC[1:0]**：DCMIPP 内核时钟源选择 (DCMIPP kernel clock source selection)
由软件写入。
0x0：选择 aclk 时钟作为内核外设时钟（复位后的默认值）
0x1：选择 pll2_q_ck 时钟作为内核外设时钟
0x2：选择 pll4_p_ck 时钟作为内核外设时钟
0x3：选择 per_ck 时钟作为内核外设时钟

10.8.92 RCC SAES 内核时钟选择寄存器 (RCC_SAESCKSELR)

RCC SAES kernel clock selection register

偏移地址：0x668

复位值：0x0000 0000

如果 (AHB5) SAES 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于控制 SAES 模块的内核时钟选择。请注意，可以实时更换时钟源，并且不会产生任何时序违规，但用户必须确保在切换期间以及整个跳变时间内，之前的时钟源和新时钟源均存在。请参见*时钟使能延迟*一节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAESSRC[1:0]	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **SAESSRC[1:0]**：SAES 内核时钟源选择 (SAES kernel clock source selection)
由软件写入。
0x0：选择 axiss_ck 时钟作为内核外设时钟（复位后的默认值）
0x1：选择 per_ck 时钟作为内核外设时钟
0x2：选择 pll_4_r_ck 时钟作为内核外设时钟
0x3：选择 lsi_ck 时钟作为内核外设时钟

10.8.93 RCC APB1 外设复位设置寄存器 (RCC_APB1RSTSETR)

RCC APB1 peripheral reset set register

偏移地址：0x6A0

复位值：0x0000 0000

该寄存器用于激活相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”激活相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SPDIFRST	Res.	Res.	Res.	I2C2RST	I2C1RST	Res.	UART8RST	UART7RST	UART5RST	UART4RST
					rs				rs	rs		rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USART3RST	Res.	Res.	SPI3RST	SPI2RST	Res.	LPTIM1RST	Res.	Res.	Res.	TIM7RST	TIM6RST	TIM5RST	TIM4RST	TIM3RST	TIM2RST
rs			rs	rs		rs				rs	rs	rs	rs	rs	rs

位 31:27 保留，必须保持复位值。

位 26 **SPDIFRST**：SPDIFRX 模块复位 (SPDIFRX block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 25:23 保留，必须保持复位值。

位 22 **I2C2RST**：I2C2 模块复位 (I2C2 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 21 **I2C1RST**：I2C1 模块复位 (I2C1 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 20 保留，必须保持复位值。

位 19 **UART8RST**：UART8 模块复位 (UART8 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 18 **UART7RST**：UART7 模块复位 (UART7 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 17 **UART5RST**：UART5 模块复位 (UART5 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 16 **UART4RST**：UART4 模块复位 (UART4 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 15 **USART3RST**：USART3 模块复位 (USART3 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 14:13 保留，必须保持复位值。

位 12 **SPI3RST**：SPI3 模块复位 (SPI3 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 11 **SPI2RST**：SPI/I2S2 模块复位 (SPI/I2S2 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 10 保留，必须保持复位值。

位 9 **LPTIM1RST**：LPTIM1 模块复位 (LPTIM1 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 8:6 保留，必须保持复位值。

位 5 **TIM7RST**：TIM7 模块复位 (TIM7 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 4 **TIM6RST**：TIM6 模块复位 (TIM6 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 3 **TIM5RST**：TIM5 模块复位 (TIM5 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 2 **TIM4RST**：TIM4 模块复位 (TIM4 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 1 **TIM3RST**：TIM3 模块复位 (TIM3 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 0 **TIM2RST**：TIM2 模块复位 (TIM2 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

10.8.94 RCC APB1 外设复位清除寄存器 (RCC_APB1RSTCLRR)

RCC APB1 peripheral reset clear register

偏移地址：0x6A4

复位值：0x0000 0000

该寄存器用于释放相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”释放相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SPDIFRST	Res.	Res.	Res.	I2C2RST	I2C1RST	Res.	UART8RST	UART7RST	UART5RST	UART4RST
					rc_w1				rc_w1	rc_w1		rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USART3RST	Res.	Res.	SPI3RST	SPI2RST	Res.	LPTIM1RST	Res.	Res.	Res.	TIM7RST	TIM6RST	TIM5RST	TIM4RST	TIM3RST	TIM2RST
rc_w1			rc_w1	rc_w1		rc_w1				rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:27 保留，必须保持复位值。

位 26 **SPDIFRST**：SPDIFRX 模块复位 (SPDIFRX block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 25:23 保留，必须保持复位值。

位 22 **I2C2RST**：I2C2 模块复位 (I2C2 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 21 **I2C1RST**：I2C1 模块复位 (I2C1 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 20 保留，必须保持复位值。

位 19 **UART8RST**：UART8 模块复位 (UART8 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 18 **UART7RST**：UART7 模块复位 (UART7 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 17 **UART5RST**：UART5 模块复位 (UART5 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 16 **UART4RST**：UART4 模块复位 (UART4 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 15 **USART3RST**：USART3 模块复位 (USART3 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 14:13 保留，必须保持复位值。

位 12 **SPI3RST**：SPI3 模块复位 (SPI3 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 11 **SPI2RST**：SPI2 模块复位 (SPI2 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 10 保留，必须保持复位值。

位 9 **LPTIM1RST**：LPTIM1 模块复位 (LPTIM1 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 8:6 保留，必须保持复位值。

位 5 **TIM7RST**：TIM7 模块复位 (TIM7 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 4 **TIM6RST**：TIM6 模块复位 (TIM6 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 3 **TIM5RST**：TIM5 模块复位 (TIM5 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 2 **TIM4RST**：TIM4 模块复位 (TIM4 block reset)

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 1 **TIM3RST**：TIM3 模块复位 (TIM3 block reset)

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 0 **TIM2RST**：TIM2 模块复位 (TIM2 block reset)

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

10.8.95 RCC APB2 外设复位设置寄存器 (RCC_APB2RSTSETR)

RCC APB2 peripheral reset set register

偏移地址：0x6A8

复位值：0x0000 0000

该寄存器用于激活相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”激活相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANRST	Res.	Res.	Res.	DFSDMRST	Res.	Res.	SAI2RST	SAI1RST
							rs				rs			rs	rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	USART6RST	Res.	Res.	Res.	Res.	SPI1RST	Res.	Res.	Res.	Res.	Res.	Res.	TIM8RST	TIM1RST
		rs					rs							rs	rs

位 31:25 保留，必须保持复位值。

位 24 **FDCANRST**：FDCAN 模块复位 (FDCAN block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 23:21 保留，必须保持复位值。

位 20 **DFSDMRST**：DFSDM 模块复位 (DFSDM block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 19:18 保留，必须保持复位值。

位 17 **SAI2RST**：SAI2 模块复位 (SAI2 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 16 **SAI1RST**：SAI1 模块复位 (SAI1 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 15:14 保留，必须保持复位值。

位 13 **USART6RST**：USART6 模块复位 (USART6 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 12:9 保留，必须保持复位值。

位 8 **SPI1RST**：SPI/I2S1 模块复位 (SPI/I2S2 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 7:2 保留，必须保持复位值。

位 1 **TIM8RST**：TIM8 模块复位 (TIM8 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 0 **TIM1RST**：TIM1 模块复位 (TIM1 block reset)
由软件置 1。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

10.8.96 RCC APB2 外设复位清除寄存器 (RCC_APB2RSTCLRR)

RCC APB2 peripheral reset clear register

偏移地址：0x6AC

复位值：0x0000 0000

该寄存器用于释放相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”释放相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANRST	Res.	Res.	Res.	DFSDMRST	Res.	Res.	SAI2RST	SAI1RST
							rc_w1				rc_w1			rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	USART6RST	Res.	Res.	Res.	Res.	SPI1RST	Res.	Res.	Res.	Res.	Res.	Res.	TIM8RST	TIM1RST
		rc_w1					rc_w1							rc_w1	rc_w1

位 31:25 保留，必须保持复位值。

位 24 **FDCANRST**：FDCAN 模块复位 (FDCAN block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 23:21 保留，必须保持复位值。

位 20 **DFSDMRST**：DFSDM 模块复位 (DFSDM block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 19:18 保留，必须保持复位值。

位 17 **SAI2RST**：SAI2 模块复位 (SAI2 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 16 **SAI1RST**：SAI1 模块复位 (SAI1 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 15:14 保留，必须保持复位值。

位 13 **USART6RST**：USART6 模块复位 (USART6 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 12:9 保留，必须保持复位值。

位 8 **SPI1RST**：SPI/I2S1 模块复位 (SPI/I2S2 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 7:2 保留，必须保持复位值。

位 1 **TIM8RST**：TIM8 模块复位 (TIM8 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 0 **TIM1RST**：TIM1 模块复位 (TIM1 block reset)
由软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

10.8.97 RCC APB3 外设复位设置寄存器 (RCC_APB3RSTSETR)

RCC APB3 peripheral reset set register

偏移地址：0x6B0

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于激活相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”激活相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTSRST
															rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	VREFRST	Res.	SYSCFGRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM5RST	LPTIM4RST	LPTIM3RST	LPTIM2RST
		rs		rs								rs	rs	rs	rs

位 31:18 保留，必须保持复位值。

位 17 保留，必须保持复位值。

位 16 **DTSRST**：DTS 模块复位 (DTS block reset)

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 15:14 保留，必须保持复位值。

位 13 **VREFRST**：VREF 模块复位 (VREF block reset)

如果 VREF 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 12 保留，必须保持复位值。

位 11 **SYSCFGRST**：SYSCFG 模块复位 (SYSCFG block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 10:4 保留，必须保持复位值。

位 3 **LPTIM5RST**：LPTIM5 模块复位 (LPTIM5 block reset)

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 2 **LPTIM4RST**：LPTIM4 模块复位 (LPTIM4 block reset)

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 1 **LPTIM3RST**：LPTIM3 模块复位 (LPTIM3 block reset)

由软件置 1。

如果 LPTIM3 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 0 **LPTIM2RST**：LPTIM2 模块复位 (LPTIM2 block reset)

由软件置 1。

如果 LPTIM2 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

10.8.98 RCC APB3 外设复位清除寄存器 (RCC_APB3RSTCLRR)

RCC APB3 peripheral reset clear register

偏移地址：0x6B4

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于释放相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”释放相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTSRST
															rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	VREFRST	Res.	SYSCFGRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM5RST	LPTIM4RST	LPTIM3RST	LPTIM2RST
		rc_w1		rc_w1								rc_w1	rc_w1	rc_w1	rc_w1

位 31:18 保留，必须保持复位值。

位 17 保留，必须保持复位值。

位 16 **DTSRST**：DTS 模块复位 (DTS block reset)

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 15:14 保留，必须保持复位值。

位 13 **VREFRST**：VREF 模块复位 (VREF block reset)

如果 VREF 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 12 保留，必须保持复位值。

位 11 **SYSCFGRST**：SYSCFG 模块复位 (SYSCFG block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 10:4 保留，必须保持复位值。

位 3 **LPTIM5RST**：LPTIM5 模块复位 (LPTIM5 block reset)

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 2 **LPTIM4RST**：LPTIM4 模块复位 (LPTIM4 block reset)

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 1 **LPTIM3RST**：LPTIM3 模块复位 (LPTIM3 block reset)

如果 LPTIM3 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 0 **LPTIM2RST**：LPTIM2 模块复位 (LPTIM2 block reset)

如果 LPTIM2 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

10.8.99 RCC APB4 外设复位设置寄存器 (RCC_APB4RSTSETR)

RCC APB4 peripheral reset set register

偏移地址：0x6B8

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于激活相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”激活相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBPHYRST
															rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DDRPERFMRST	Res.	Res.	Res.	Res.	Res.	Res.	DCMIPPRST	LTDCRST
							rs							rs	rs

位 31:17 保留，必须保持复位值。

位 16 **USBPHYRST**：USBPHYC 模块复位 (USBPHYC block reset)

如果 USB PHY 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 15:9 保留，必须保持复位值。

位 8 **DDRPERFMRST**：DDRPERFM 模块复位 (DDRPERFM block reset)

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 7:2 保留，必须保持复位值。

位 1 **DCMIPPRST**：DCMIPP 模块复位 (DCMIPP block reset)

如果 DCMIPP 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 0 **LTDCRST**：LTDC 模块复位 (LTDC block reset)

如果 LTDC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

10.8.100 RCC APB4 外设复位清除寄存器 (RCC_APB4RSTCLRR)

RCC APB4 peripheral reset clear register

偏移地址：0x6BC

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于释放相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”释放相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBPHYRST
															rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DDRPERFMRST	Res.	Res.	Res.	Res.	Res.	Res.	DCMIPPRST	LTDCRST
							rc_w1							rc_w1	rc_w1

位 31:17 保留，必须保持复位值。

位 16 **USBPHYRST**：USBPHYC 模块复位 (USBPHYC block reset)

如果 USB PHY 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 15:9 保留，必须保持复位值。

位 8 **DDRPERFMRST**：DDRPERFM 模块复位 (DDRPERFM block reset)

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 7:2 保留，必须保持复位值。

位 1 **DCMIPPRST**：DCMIPP 模块复位 (DCMIPP block reset)

如果 DCMIPP 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 0 **LTDCRST**：LTDC 模块复位 (LTDC block reset)

如果 LTDC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

10.8.101 RCC APB5 外设复位设置寄存器 (RCC_APB5RSTSETR)

RCC APB5 peripheral reset set register

偏移地址：0x6C0

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于激活相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”激活相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENRST	Res.	Res.	Res.	Res.
											rs				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:21 保留，必须保持复位值。

位 20 **STGENRST**：STGEN 模块复位 (STGEN block reset)

如果 STGENC（STGEN 控制器子部分）通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 19:0 保留，必须保持复位值。

10.8.102 RCC APB5 外设复位清除寄存器 (RCC_APB5RSTCLRR)

RCC APB5 peripheral reset clear register

偏移地址：0x6C4

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于释放相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”释放相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENRST	Res.	Res.	Res.	Res.
											rc_w1				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:21 保留，必须保持复位值。

位 20 **STGENRST**：STGEN 模块复位 (STGEN block reset)

如果 STGENC（STGEN 控制器子部分）通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 19:0 保留，必须保持复位值。

10.8.103 RCC APB6 外设复位设置寄存器 (RCC_APB6RSTSETR)

RCC APB6 peripheral reset set register

偏移地址：0x6C8

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于激活相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”激活相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TIM17RST	TIM16RST	TIM15RST	TIM14RST	TIM13RST	TIM12RST	I2C5RST	I2C4RST	I2C3RST	SPI5RST	SPI4RST	USART2RST	USART1RST
			rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:13 保留，必须保持复位值。

位 12 **TIM17RST**：TIM17 模块复位 (TIM17 block reset)

如果 TIM17 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 11 **TIM16RST**：TIM16 模块复位 (TIM16 block reset)

如果 TIM16 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 10 **TIM15RST**：TIM15 模块复位 (TIM15 block reset)

如果 TIM15 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 9 **TIM14RST**：TIM14 模块复位 (TIM14 block reset)

如果 TIM14 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 8 **TIM13RST**：TIM13 模块复位 (TIM13 block reset)

如果 TIM13 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 7 **TIM12RST**：TIM12 模块复位 (TIM12 block reset)

如果 TIM12 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 6 **I2C5RST**：I2C5 模块复位 (I2C5 block reset)

如果 I2C5 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 5 **I2C4RST**：I2C4 模块复位 (I2C4 block reset)

如果 I2C4 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 4 **I2C3RST**：I2C3 模块复位 (I2C3 block reset)

如果 I2C3 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 3 **SPI5RST**：SPI5 模块复位 (SPI5 block reset)

如果 SPI5 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 2 **SPI4RST**：SPI4 模块复位 (SPI4 block reset)

如果 SPI4 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 1 **USART2RST**：USART2 模块复位 (USART2 block reset)

如果 USART2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 0 **USART1RST**：USART1 模块复位 (USART1 block reset)

如果 USART1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

10.8.104 RCC APB6 外设复位清除寄存器 (RCC_APB6RSTCLRR)

RCC APB6 peripheral reset clear register

偏移地址：0x6CC

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于释放相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”释放相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TIM17RST	TIM16RST	TIM15RST	TIM14RST	TIM13RST	TIM12RST	I2C5RST	I2C4RST	I2C3RST	SPI5RST	SPI4RST	USART2RST	USART1RST
			rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:13 保留，必须保持复位值。

位 12 **TIM17RST**：TIM17 模块复位 (TIM17 block reset)

如果 TIM17 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 11 **TIM16RST**：TIM16 模块复位 (TIM16 block reset)

如果 TIM16 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 10 **TIM15RST**：TIM15 模块复位 (TIM15 block reset)

如果 TIM15 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 9 **TIM14RST**：TIM14 模块复位 (TIM14 block reset)

如果 TIM14 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 8 **TIM13RST**：TIM13 模块复位 (TIM13 block reset)

如果 TIM13 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 7 **TIM12RST**：TIM12 模块复位 (TIM12 block reset)

如果 TIM12 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 6 **I2C5RST**：I2C5 模块复位 (I2C5 block reset)

如果 I2C5 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 5 **I2C4RST**：I2C4 模块复位 (I2C4 block reset)

如果 I2C4 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 4 **I2C3RST**：I2C3 模块复位 (I2C3 block reset)

如果 I2C3 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 3 **SPI5RST**：SPI5 模块复位 (SPI5 block reset)

如果 SPI5 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 2 **SPI4RST**：SPI4 模块复位 (SPI4 block reset)

如果 SPI4 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 1 **USART2RST**：USART2 模块复位 (USART2 block reset)

如果 USART2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 0 **USART1RST**：USART1 模块复位 (USART1 block reset)

如果 USART1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

10.8.105 RCC AHB2 外设复位设置寄存器 (RCC_AHB2RSTSETR)

RCC AHB2 peripheral reset set register

偏移地址：0x6D0

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于激活相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”激活相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBORST	Res.	ADC2RST	ADC1RST	DMAMUX2RST	DMA3RST	DMAMUX1RST	DMA2RST	DMA1RST
							rs		rs	rs	rs	rs	rs	rs	rs

位 31:9 保留，必须保持复位值。

位 8 **USBORST**：USBO 模块复位 (USBO block reset)

如果 USBO 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 7 保留，必须保持复位值。

位 6 **ADC2RST**：ADC2 模块复位 (ADC2 block reset)

如果 ADC2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 5 **ADC1RST**：ADC1 模块复位 (ADC1 block reset)

如果 ADC1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 4 **DMAMUX2RST**：DMAMUX2 模块复位 (DMAMUX2 block reset)

如果 DMAMUX2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 3 **DMA3RST**：DMA3 模块复位 (DMA3 block reset)

如果 DMA3 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 2 **DMAMUX1RST**：DMAMUX1 模块复位 (DMAMUX1 block reset)

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 1 **DMA2RST**：DMA2 模块复位 (DMA2 block reset)

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 0 **DMA1RST**：DMA1 模块复位 (DMA1 block reset)

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

10.8.106 RCC AHB2 外设复位清除寄存器 (RCC_AHB2RSTCLRR)

RCC AHB2 peripheral reset clear register

偏移地址：0x6D4

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于释放相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”释放相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBORST	Res.	ADC2RST	ADC1RST	DMAMUX2RST	DMA3RST	DMAMUX1RST	DMA2RST	DMA1RST
							rc_w1		rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:9 保留，必须保持复位值。

位 8 **USBORST**：USBO 模块复位 (USBO block reset)

如果 USBO 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 7 保留，必须保持复位值。

位 6 **ADC2RST**：ADC2 模块复位 (ADC2 block reset)

如果 ADC2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 5 **ADC1RST**：ADC1 模块复位 (ADC1 block reset)

如果 ADC1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 4 **DMAMUX2RST**：DMAMUX2 模块复位 (DMAMUX2 block reset)

如果 DMAMUX2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 3 **DMA3RST**：DMA3 模块复位 (DMA3 block reset)

如果 DMA3 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 2 **DMAMUX1RST**：DMAMUX1 模块复位 (DMAMUX1 block reset)

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 1 **DMA2RST**：DMA2 模块复位 (DMA2 block reset)

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 0 **DMA1RST**：DMA1 模块复位 (DMA1 block reset)

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

10.8.107 RCC AHB4 外设复位设置寄存器 (RCC_AHB4RSTSETR)

RCC AHB4 peripheral reset set register

偏移地址：0x6E0

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于激活相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”激活相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOIRST	GPIOHRST	GPIOGRST	GPIOFRST	GPIOERST	GPIODRST	GPIOCRST	GPIOBRST	GPIOARST
							rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:9 保留，必须保持复位值。

位 8 **GPIOIRST**：GPIOI 模块复位 (GPIOI block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 7 **GPIOHRST**：GPIOH 模块复位 (GPIOH block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 6 **GPIOGRST**：GPIOG 模块复位 (GPIOG block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 5 **GPIOFRST**：GPIOF 模块复位 (GPIOF block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 4 **GPIOERST**：GPIOE 模块复位 (GPIOE block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 3 **GPIODRST**：GPIOD 模块复位 (GPIOD block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 2 **GPIOCRST**：GPIOC 模块复位 (GPIOC block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 1 **GPIOBRST**：GPIOB 模块复位 (GPIOB block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 0 **GPIOARST**：GPIOA 模块复位 (GPIOA block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

10.8.108 RCC AHB4 外设复位清除寄存器 (RCC_AHB4RSTCLRR)

RCC AHB4 peripheral reset clear register

偏移地址：0x6E4

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于释放相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”释放相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOIRST	GPIOHRST	GPIOGRST	GPIOFRST	GPIOERST	GPIODRST	GPIOCRST	GPIOBRST	GPIOARST
							rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:9 保留，必须保持复位值。

位 8 **GPIOIRST**：GPIOI 模块复位 (GPIOI block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。
由（安全）软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 7 **GPIOHRST**：GPIOH 模块复位 (GPIOH block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。
由（安全）软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 6 **GPIOGRST**：GPIOG 模块复位 (GPIOG block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。
由（安全）软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 5 **GPIOFRST**：GPIOF 模块复位 (GPIOF block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。
由（安全）软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 4 **GPIOERST**：GPIOE 模块复位 (GPIOE block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。
由（安全）软件清零。
0：写入“0”无影响，读取“0”表示模块复位已释放
1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 3 **GPIODRST**：GPIOD 模块复位 (GPIOD block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 2 **GPIOCRST**：GPIOC 模块复位 (GPIOC block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 1 **GPIOBRST**：GPIOB 模块复位 (GPIOB block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 0 **GPIOARST**：GPIOA 模块复位 (GPIOA block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

10.8.109 RCC AHB5 外设复位设置寄存器 (RCC_AHB5RSTSETR)

RCC AHB5 peripheral reset set register

偏移地址：0x6E8

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于激活相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”激活相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXIMCRST
															rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNG1RST	HASH1RST	CRYP1RST	SAESRST	PKARST	Res.	Res.
									rs	rs	rs	rs	rs		

位 31:17 保留，必须保持复位值。

位 16 **AXIMCRST**：AXIMC 模块复位 (AXIMC block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 15:7 保留，必须保持复位值。

位 6 **RNG1RST**：RNG1 模块复位 (RNG1 block reset)

如果 RNG1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 5 **HASH1RST**：HASH 模块复位 (HASH block reset)

如果 HASH 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 4 **CRYP1RST**：CRYP (3DES/AES) 模块复位 (CRYP (3DES/AES) block reset)

如果 CRYP 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 3 **SAESRST**：SAES 模块复位 (SAES block reset)

如果 SAES 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 2 **PKARST**：PKA 模块复位 (PKA block reset)

如果 PKA 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 1:0 保留，必须保持复位值。

10.8.110 RCC AHB5 外设复位清除寄存器 (RCC_AHB5RSTCLRR)

RCC AHB5 peripheral reset clear register

偏移地址：0x6EC

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于释放相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”释放相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXIMCRST
															rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNG1RST	HASH1RST	CRYP1RST	SAESRST	PKARST	Res.	Res.
									rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		

位 31:17 保留，必须保持复位值。

位 16 **AXIMCRST**：AXIMC 模块复位 (AXIMC block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 15:7 保留，必须保持复位值。

位 6 **RNG1RST**：RNG1 模块复位 (RNG1 block reset)

如果 RNG1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 5 **HASH1RST**：HASH 模块复位 (HASH block reset)

如果 HASH 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 4 **CRYP1RST**：CRYP (3DES/AES) 模块复位 (CRYP (3DES/AES) block reset)

如果 CRYP 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 3 **SAESRST**：SAES 模块复位 (SAES block reset)

如果 SAES 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 2 **PKARST**：PKA 模块复位 (PKA block reset)

如果 PKA 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 1:0 保留，必须保持复位值。

10.8.111 RCC AHB6 外设复位设置寄存器 (RCC_AHB6RSTSETR)

RCC AHB6 peripheral reset set register

偏移地址：0x6F0

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于激活相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”激活相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	ETH2MACRST	Res.	Res.	Res.	Res.	Res.	USBHRST	Res.	Res.	Res.	CRC1RST	Res.	Res.	SDMMC2RST	SDMMC1RST
	rs						rs				rs			rs	rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	QSPIRST	Res.	FMCRST	Res.	ETH1MACRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCERST	MDMARST
	rs		rs		rs									rs	rs

位 31 保留，必须保持复位值。

位 30 **ETH2MACRST**：ETH2 MAC 模块复位 (ETH2 MAC block reset)

如果 ETH2 MAC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 29:25 保留，必须保持复位值。

位 24 **USBHRST**：USBH 模块复位 (USBH block reset)

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 23:21 保留，必须保持复位值。

位 20 **CRC1RST**：CRC 模块复位 (CRC block reset)

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 19:18 保留，必须保持复位值。

位 17 **SDMMC2RST**：SDMMC2 和 SDMMC2 延迟 (DLYBSD2) 模块复位 (SDMMC2 and the SDMMC2 delay (DLYBSD2) block reset)

如果 SDMMC2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 16 **SDMMC1RST**：SDMMC1 和 SDMMC1 延迟 (DLYBSD1) 模块复位 (SDMMC1 and the SDMMC1 delay (DLYBSD1) block reset)

如果 SDMMC1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 15 保留，必须保持复位值。

位 14 **QSPIRST**：QUADSPI 和 QUADSPI 延迟模块复位 (QUADSPI and the QUADSPI delay block reset)

如果 QSPI 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 13 保留，必须保持复位值。

位 12 **FMCRST**：FMC 模块复位 (FMC block reset)

如果 FMC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 11 保留，必须保持复位值。

位 10 **ETH1MACRST**：ETH1 MAC 模块复位 (ETH1 MAC block reset)

如果 ETH1 MAC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 9:2 保留，必须保持复位值。

位 1 **MCERST**：MCE 模块复位 (MCE block reset)

如果 MCE 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

位 0 **MDMARST**：MDMA 模块复位 (MDMA block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”将模块复位置为有效，读取“1”表示模块复位已置为有效

10.8.112 RCC AHB6 外设复位清除寄存器 (RCC_AHB6RSTCLRR)

RCC AHB6 peripheral reset clear register

偏移地址：0x6F4

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于释放相应外设的复位。写入“0”无影响，读取则会返回相应位的有效值。写入“1”释放相应外设的复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	ETH2MACRST	Res.	Res.	Res.	Res.	Res.	USBHRST	Res.	Res.	Res.	CRC1RST	Res.	Res.	SDMMC2RST	SDMMC1RST
	rc_w1						rc_w1				rc_w1			rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	QSPIRST	Res.	FMCRST	Res.	ETH1MACRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCERST	MDMARST
	rc_w1		rc_w1		rc_w1									rc_w1	rc_w1

位 31 保留，必须保持复位值。

位 30 **ETH2MACRST**：ETH2 MAC 模块复位 (ETH2 MAC block reset)

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 29:25 保留，必须保持复位值。

位 24 **USBHRST**：USBH 模块复位 (USBH block reset)

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 23:21 保留，必须保持复位值。

位 20 **CRC1RST**：CRC 模块复位 (CRC block reset)

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 19:18 保留，必须保持复位值。

位 17 **SDMMC2RST**：SDMMC2 和 SDMMC2 延迟 (DLYBSD2) 模块复位 (SDMMC2 and the SDMMC2 delay (DLYBSD2) block reset)

如果 SDMMC2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 16 **SDMMC1RST**：SDMMC1 和 SDMMC1 延迟 (DLYBSD1) 模块复位 (SDMMC1 and the SDMMC1 delay (DLYBSD1) block reset)

如果 SDMMC1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 15 保留，必须保持复位值。

位 14 **QSPIRST**：QUADSPI 和 QUADSPI 延迟模块复位 (QUADSPI and the QUADSPI delay block reset)

如果 QSPI 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 13 保留，必须保持复位值。

位 12 **FMCRST**：FMC 模块复位 (FMC block reset)

如果 FMC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 11 保留，必须保持复位值。

位 10 **ETH1MACRST**：ETH1 MAC 模块复位 (ETH1 MAC block reset)

如果 ETH1 MAC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 9:2 保留，必须保持复位值。

位 1 **MCERST**：MCE 模块复位 (MCE block reset)

如果 MCE 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

位 0 **MDMARST**：MDMA 模块复位 (MDMA block reset)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示模块复位已释放

1：写入“1”释放模块复位，读取“1”表示模块复位置为有效

10.8.113 RCC APB1 外设使能设置寄存器 (RCC_MP_APB1ENSETR)

RCC APB1 peripheral enable set register

偏移地址：0x700

复位值：0x0000 0000

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SPDIFEN	Res.	Res.	Res.	I2C2EN	I2C1EN	Res.	UART8EN	UART7EN	UART5EN	UART4EN
					rs				rs	rs		rs	rs	rs	rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USART3EN	Res.	Res.	SPI3EN	SPI2EN	Res.	LPTIM1EN	Res.	Res.	Res.	TIM7EN	TIM6EN	TIM5EN	TIM4EN	TIM3EN	TIM2EN
rs			rs	rs		rs				rs	rs	rs	rs	rs	rs

位 31:27 保留，必须保持复位值。

位 26 **SPDIFEN**：SPDIFRX 外设时钟使能 (SPDIFRX peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 25:23 保留，必须保持复位值。

位 22 **I2C2EN**：I2C2 外设时钟使能 (I2C2 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 21 **I2C1EN**：I2C1 外设时钟使能 (I2C1 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 20 保留，必须保持复位值。

位 19 **UART8EN**：UART8 外设时钟使能 (UART8 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 18 **UART7EN**：UART7 外设时钟使能 (UART7 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 17 **UART5EN**：UART5 外设时钟使能 (UART5 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 16 **UART4EN**：UART4 外设时钟使能 (UART4 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 15 **USART3EN**：USART3 外设时钟使能 (USART3 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 14:13 保留，必须保持复位值。

位 12 **SPI3EN**：SPI3 外设时钟使能 (SPI3 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 11 **SPI2EN**：SPI2 外设时钟使能 (SPI2 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 10 保留，必须保持复位值。

位 9 **LPTIM1EN**：LPTIM1 外设时钟使能 (LPTIM1 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 8:6 保留，必须保持复位值。

位 5 **TIM7EN**：TIM7 外设时钟使能 (TIM7 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 4 **TIM6EN**：TIM6 外设时钟使能 (TIM6 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 3 **TIM5EN**：TIM5 外设时钟使能 (TIM5 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 2 **TIM4EN**：TIM4 外设时钟使能 (TIM4 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 1 **TIM3EN**：TIM3 外设时钟使能 (TIM3 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 0 **TIM2EN**：TIM2 外设时钟使能 (TIM2 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

10.8.114 RCC APB1 外设使能清零寄存器 (RCC_MP_APB1ENCLRR)

RCC APB1 peripheral enable clear register

偏移地址：0x704

复位值：0x0000 0000

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SPDIFEN	Res.	Res.	Res.	I2C2EN	I2C1EN	Res.	UART8EN	UART7EN	UART5EN	UART4EN
					rc_w1				rc_w1	rc_w1		rc_w1	rc_w1	rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USART3EN	Res.	Res.	SPI3EN	SPI2EN	Res.	LPTIM1EN	Res.	Res.	Res.	TIM7EN	TIM6EN	TIM5EN	TIM4EN	TIM3EN	TIM2EN
rc_w1			rc_w1	rc_w1		rc_w1				rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:27 保留，必须保持复位值。

位 26 **SPDIFEN**：SPDIFRX 外设时钟使能 (SPDIFRX peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 25:23 保留，必须保持复位值。

位 22 **I2C2EN**：I2C2 外设时钟使能 (I2C2 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 21 **I2C1EN**：I2C1 外设时钟使能 (I2C1 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 20 保留，必须保持复位值。

位 19 **UART8EN**：UART8 外设时钟使能 (UART8 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 18 **UART7EN**：UART7 外设时钟使能 (UART7 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 17 **UART5EN**：UART5 外设时钟使能 (UART5 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 16 **UART4EN**：UART4 外设时钟使能 (UART4 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 15 **USART3EN**：USART3 外设时钟使能 (USART3 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 14:13 保留，必须保持复位值。

位 12 **SPI3EN**：SPI3 外设时钟使能 (SPI3 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 11 **SPI2EN**：SPI2 外设时钟使能 (SPI2 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 10 保留，必须保持复位值。

位 9 **LPTIM1EN**：LPTIM1 外设时钟使能 (LPTIM1 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 8:6 保留，必须保持复位值。

位 5 **TIM7EN**：TIM7 外设时钟使能 (TIM7 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 4 **TIM6EN**：TIM6 外设时钟使能 (TIM6 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 3 **TIM5EN**：TIM5 外设时钟使能 (TIM5 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 2 **TIM4EN**：TIM4 外设时钟使能 (TIM4 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 1 **TIM3EN**：TIM3 外设时钟使能 (TIM3 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 0 **TIM2EN**：TIM2 外设时钟使能 (TIM2 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

10.8.115 RCC APB2 外设使能设置寄存器 (RCC_MP_APB2ENSETR)

RCC APB2 peripheral enable set register

偏移地址：0x708

复位值：0x0000 0000

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANEN	Res.	Res.	ADFSDMEN	DFSDMEN	Res.	Res.	SAI2EN	SAI1EN
							rs			rs	rs			rs	rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	USART6EN	Res.	Res.	Res.	Res.	SPI1EN	Res.	Res.	Res.	Res.	Res.	Res.	TIM8EN	TIM1EN
		rs					rs							rs	rs

位 31:25 保留，必须保持复位值。

位 24 **FDCANEN**：FDCAN 和 CANRAM 外设时钟使能 (FDCAN and CANRAM peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 23:22 保留，必须保持复位值。

位 21 **ADFSDMEN**：音频 DFSDM 外设时钟使能 (Audio DFSDM peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 20 **DFSDMEN**：DFSDM 外设时钟使能 (DFSDM peripheral clock enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 19:18 保留，必须保持复位值。

位 17 **SAI2EN**：SAI2 外设时钟使能 (SAI2 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 16 **SAI1EN**：SAI1 外设时钟使能 (SAI1 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 15:14 保留，必须保持复位值。

位 13 **USART6EN**：USART6 外设时钟使能 (USART6 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 12:9 保留，必须保持复位值。

位 8 **SPI1EN**：SPI/I2S1 外设时钟使能 (SPI/I2S1 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 7:2 保留，必须保持复位值。

位 1 **TIM8EN**：TIM8 外设时钟使能 (TIM8 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 0 **TIM1EN**：TIM1 外设时钟使能 (TIM1 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

10.8.116 RCC APB2 外设使能清零寄存器 (RCC_MP_APB2ENCLRR)

RCC APB2 peripheral enable clear register

偏移地址：0x70C

复位值：0x0000 0000

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANEN	Res.	Res.	ADFSDMEN	DFSDMEN	Res.	Res.	SAI2EN	SAI1EN
							rc_w1			rc_w1	rc_w1			rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	USART6EN	Res.	Res.	Res.	Res.	SPI1EN	Res.	Res.	Res.	Res.	Res.	Res.	TIM8EN	TIM1EN
		rc_w1					rc_w1							rc_w1	rc_w1

位 31:25 保留，必须保持复位值。

位 24 **FDCANEN**：FDCAN 和 CANRAM 外设时钟使能 (FDCAN and CANRAM peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 23:22 保留，必须保持复位值。

位 21 **ADFSDMEN**：音频 DFSDM 外设时钟使能 (Audio DFSDM peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 20 **DFSDMEN**：DFSDM 外设时钟使能 (DFSDM peripheral clock enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 19:18 保留，必须保持复位值。

位 17 **SAI2EN**：SAI2 外设时钟使能 (SAI2 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 16 **SAI1EN**：SAI1 外设时钟使能 (SAI1 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 15:14 保留，必须保持复位值。

位 13 **USART6EN**：USART6 外设时钟使能 (USART6 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 12:9 保留，必须保持复位值。

位 8 **SPI1EN**：SPI/I2S1 外设时钟使能 (SPI/I2S1 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 7:2 保留，必须保持复位值。

位 1 **TIM8EN**：TIM8 外设时钟使能 (TIM8 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 0 **TIM1EN**：TIM1 外设时钟使能 (TIM1 peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

10.8.117 RCC APB3 外设使能设置寄存器 (RCC_MP_APB3ENSETR)

RCC APB3 peripheral enable set register

偏移地址：0x710

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDPEN	Res.	Res.	Res.	DTSEN
											rs				rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	VREFEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM5EN	LPTIM4EN	LPTIM3EN	LPTIM2EN
		rs										rs	rs	rs	rs

位 31:21 保留，必须保持复位值。

位 20 **HDPEN**：HDP 外设时钟使能 (HDP peripheral clock enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 19:18 保留，必须保持复位值。

位 17 保留，必须保持复位值。

位 16 **DTSEN**：DTS 外设时钟使能 (DTS peripheral clock enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 15:14 保留，必须保持复位值。

位 13 **VREFEN**：VREF 外设时钟使能 (VREF peripheral clock enable)
如果 VREF 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 12:4 保留，必须保持复位值。

位 3 **LPTIM5EN**：LPTIM5 外设时钟使能 (LPTIM5 peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 2 **LPTIM4EN**：LPTIM4 外设时钟使能 (LPTIM4 peripheral clocks enable)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 1 **LPTIM3EN**：LPTIM3 外设时钟使能 (LPTIM3 peripheral clocks enable)

如果 LPTIM3 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 0 **LPTIM2EN**：LPTIM2 外设时钟使能 (LPTIM2 peripheral clocks enable)

如果 LPTIM2 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

10.8.118 RCC APB3 外设使能清零寄存器 (RCC_MP_APB3ENCLRR)

RCC APB3 peripheral enable clear register

偏移地址：0x714

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDPEN	Res.	Res.	Res.	DTSEN
											rc_w1				rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	VREFEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM5EN	LPTIM4EN	LPTIM3EN	LPTIM2EN
		rc_w1										rc_w1	rc_w1	rc_w1	rc_w1

位 31:21 保留，必须保持复位值。

位 20 **HDPEN**：HDP 外设时钟使能 (HDP peripheral clock enable)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 19:18 保留，必须保持复位值。

位 17 保留，必须保持复位值。

位 16 **DTSEN**：DTS 外设时钟使能 (DTS peripheral clock enable)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 15:14 保留，必须保持复位值。

位 13 **VREFEN**：VREF 外设时钟使能 (VREF peripheral clock enable)

如果 VREF 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 12:4 保留，必须保持复位值。

位 3 **LPTIM5EN**：LPTIM5 外设时钟使能 (LPTIM5 peripheral clocks enable)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 2 **LPTIM4EN**：LPTIM4 外设时钟使能 (LPTIM4 peripheral clocks enable)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 1 **LPTIM3EN**：LPTIM3 外设时钟使能 (LPTIM3 peripheral clocks enable)

如果 LPTIM3 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 0 **LPTIM2EN**：LPTIM2 外设时钟使能 (LPTIM2 peripheral clocks enable)

如果 LPTIM2 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

10.8.119 RCC 安全 APB3 外设使能设置寄存器 (RCC_MP_S_APB3ENSETR)

RCC secure APB3 peripheral enable set register

偏移地址：0x718

复位值：0x0000 0000

该安全寄存器用于使能 SYSCFG 的时钟。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将相应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGEN
															rs

位 31:1 保留，必须保持复位值。

位 0 **SYSCFGEN**：SYSCFG 外设时钟使能 (SYSCFG peripheral clocks enable)

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

10.8.120 RCC 安全 APB3 外设使能清零寄存器 (RCC_MP_S_APB3ENCLRR)

RCC secure APB3 peripheral enable clear register

偏移地址：0x71C

复位值：0x0000 0000

该安全寄存器用于禁止 SYSCFG 的时钟。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGEN
															rc_w1

位 31:1 保留，必须保持复位值。

位 0 **SYSCFGEN**：SYSCFG 外设时钟使能 (SYSCFG peripheral clocks enable)
由（安全）软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

10.8.121 RCC 非安全 APB3 外设使能设置寄存器 (RCC_MP_NS_APB3ENSETR)

RCC non-secure APB3 peripheral enable set register

偏移地址：0x720

复位值：0x0000 0000

该非安全寄存器用于使能 SYSCFG 的时钟。

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将相应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGEN
															rs

位 31:1 保留，必须保持复位值。

位 0 **SYSCFGEN**：SYSCFG 外设时钟使能 (SYSCFG peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

10.8.122 RCC 非安全 APB3 外设使能清零寄存器 (RCC_MP_NS_APB3ENCLRR)

RCC non-secure APB3 peripheral enable clear register

偏移地址：0x724

复位值：0x0000 0000

该非安全寄存器用于禁止 SYSCFG 的时钟。

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGEN
															rs

位 31:1 保留，必须保持复位值。

位 0 **SYSCFGEN**：SYSCFG 外设时钟使能 (SYSCFG peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

10.8.123 RCC APB4 外设使能设置寄存器 (RCC_MP_APB4ENSETR)

RCC APB4 peripheral enable set register

偏移地址：0x728

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENROEN	Res.	Res.	Res.	USBPHYEN
											rs				rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDG2APBEN	Res.	Res.	Res.	Res.	Res.	Res.	DDRPERFMEN	Res.	Res.	Res.	Res.	Res.	Res.	DCMIPPEN	Res.
rs							rs							rs	

位 31:21 保留，必须保持复位值。

位 20 **STGENROEN**：STGEN 只读接口外设时钟使能 (STGEN read-only interface peripheral clocks enable)

由软件置 1。

STGEN 只读接口的外设时钟是 pclk4，以及由 RCC_STGENCKSELR 寄存器选定的内核时钟。

0：写入“0”无影响，读取“0”表示已禁止 pclk4，且内核时钟门控由 RCC_MP_APB5ENxxxR.STGENCEN (xxx=SET/CLR) 定义

1：写入“1”使能 pclk4 和内核时钟；读取“1”表示 pclk4 和内核时钟已使能

位 19:17 保留，必须保持复位值。

位 16 **USBPHYEN**：USBPHYC 外设时钟使能 (USBPHYC peripheral clock enable)

如果 USB PHY 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 15 **IWDG2APBEN**：IWDG2 APB 时钟使能 (IWDG2 APB clock enable)

由软件置 1。

0：写入“0”无影响，读取“0”表示 APB 时钟已禁止

1：写入“1”使能 APB 时钟，读取“1”表示 APB 时钟已使能

位 14:9 保留，必须保持复位值。

位 8 **DDRPERFMEN**：DDRPERFM APB 时钟使能 (DDRPERFM APB clock enable)

由软件置 1。

0：写入“0”无影响，读取“0”表示 APB 时钟已禁止

1：写入“1”使能 APB 时钟，读取“1”表示 APB 时钟已使能

位 7:2 保留，必须保持复位值。

位 1 **DCMIPPEN**：DCMIPP 外设时钟使能 (DCMIPP peripheral clock enable)

如果 DCMIPP 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 0 保留，必须保持复位值。

10.8.124 RCC APB4 外设使能清零寄存器 (RCC_MP_APB4ENCLRR)

RCC APB4 peripheral enable clear register

偏移地址：0x72C

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENROEN	Res.	Res.	Res.	USBPHYEN
											rc_w1				rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDG2APBEN	Res.	Res.	Res.	Res.	Res.	Res.	DDRPERFMEN	Res.	Res.	Res.	Res.	Res.	Res.	DCMIPPEN	Res.
rc_w1							rc_w1							rc_w1	

位 31:21 保留，必须保持复位值。

位 20 **STGENROEN**：STGEN 只读接口外设时钟使能 (STGEN read-only interface peripheral clocks enable)

由软件清零。

STGEN 只读接口的外设时钟是 pclk4，以及由 RCC_STGENCKSELR 寄存器选定的内核时钟。

0：写入“0”无影响，读取“0”表示已禁止 pclk4，且内核时钟门控由 RCC_MP_APB5ENxxxR.STGENCEN (xxx=SET/CLR) 定义

1：写入“1”禁止 pclk4，并会禁止内核时钟（当 RCC_MP_APB5ENSETR.STGENCEN=0 时，否则无影响）；读取“1”表示 pclk4 和内核时钟已使能

位 19:17 保留，必须保持复位值。

位 16 **USBPHYEN**：USBPHYC 外设时钟使能 (USBPHYC peripheral clock enable)

如果 USB PHY 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 15 **IWDG2APBEN**：IWDG2 APB 时钟使能 (IWDG2 APB clock enable)

由软件清零。

0：写入“0”无影响，读取“0”表示 APB 时钟已禁止

1：写入“1”禁止 APB 时钟，读取“1”表示 APB 时钟已使能

位 14:9 保留，必须保持复位值。

位 8 **DDRPERFMEN**：DDRPERFM APB 时钟使能 (DDRPERFM APB clock enable)

由软件清零。

0：写入“0”无影响，读取“0”表示 APB 时钟已禁止

1：写入“1”禁止 APB 时钟，读取“1”表示 APB 时钟已使能

位 7:2 保留，必须保持复位值。

位 1 **DCMIPPEN**：DCMIPP 外设时钟使能 (DCMIPP peripheral clock enable)

如果 DCMIPP 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 0 保留，必须保持复位值。

10.8.125 RCC 安全 APB4 外设使能设置寄存器 (RCC_MP_S_APB4ENSETR)

RCC secure APB4 peripheral enable set register

偏移地址：0x730

复位值：0x0000 0000

这是一个安全寄存器，用于在 LTDC 与安全合成层一起使用时使能其时钟。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCEN
															rs

位 31:1 保留，必须保持复位值。

位 0 **LTDCEN**：LTDC 外设时钟使能 (LTDC peripheral clock enable)

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

10.8.126 RCC 安全 APB4 外设使能清零寄存器 (RCC_MP_S_APB4ENCLRR)

RCC secure APB4 peripheral enable clear register

偏移地址：0x734

复位值：0x0000 0000

这是一个安全寄存器，用于在 LTDC 与安全合成层一起使用时禁止其时钟。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCEN
															rc_w1

位 31:1 保留，必须保持复位值。

位 0 **LTDCEN**：LTDC 外设时钟使能 (LTDC peripheral clock enable)

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

10.8.127 RCC 非安全 APB4 外设使能设置寄存器 (RCC_MP_NS_APB4ENSETR)

RCC non-secure APB4 peripheral enable set register

偏移地址：0x738

复位值：0x0000 0000

这是一个非安全寄存器，用于在 LTDC 与一个或两个非安全合成层一起使用时使能其时钟。

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCEN
															rs

位 31:1 保留，必须保持复位值。

位 0 **LTDCEN**：LTDC 外设时钟使能 (LTDC peripheral clock enable)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

10.8.128 RCC 非安全 APB4 外设使能清零寄存器 (RCC_MP_NS_APB4ENCLRR)

RCC non-secure APB4 peripheral enable clear register

偏移地址：0x73C

复位值：0x0000 0000

这是一个非安全寄存器，用于在 LTDC 与一个或两个非安全合成层一起使用时禁止其时钟。

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCEN
															rc_w1

位 31:1 保留，必须保持复位值。

位 0 **LTDCEN**：LTDC 外设时钟使能 (LTDC peripheral clock enable)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

10.8.129 RCC APB5 外设使能设置寄存器 (RCC_MP_APB5ENSETR)

RCC APB5 peripheral enable set register

偏移地址：0x740

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENCEN	Res.	Res.	Res.	BSECEN
											rs				rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDG1APBEN	Res.	ETZPCEN	Res.	TZCEN	Res.	Res.	RTCAPBEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs		rs		rs			rs								

位 31:21 保留，必须保持复位值。

位 20 **STGENCEN**：STGEN 控制器部分，外设时钟使能 (STGEN controller part, peripheral clocks enable)

如果 STGENC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

STGEN 控制器部分的外设时钟是 pclk5，以及由 RCC_STGENCKSELR 寄存器选定的内核时钟。

0：写入“0”无影响，读取“0”表示已禁止 pclk5，且内核时钟门控由 RCC_MP_APB4ENxxxR.STGENROEN (xxx=SET/CLR) 定义

1：写入“1”使能 pclk5 和内核时钟；读取“1”表示 pclk5 和内核时钟已使能

位 19:17 保留，必须保持复位值。

位 16 **BSECEN**：BSEC 外设时钟使能 (BSEC peripheral clock enable)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 15 **IWDG1APBEN**：IWDG1 APB 时钟使能 (IWDG1 APB clock enable)

如果 IWDG1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示 APB 时钟已禁止

1：写入“1”使能 APB 时钟，读取“1”表示 APB 时钟已使能

位 14 保留，必须保持复位值。

位 13 **ETZPCEN**：ETZPC 外设时钟使能 (ETZPC peripheral clock enable)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 12 保留，必须保持复位值。

位 11 **TZCEN**：TZC 时钟使能 (TZC clocks enable)

由（安全）软件置 1。

有关详细信息，请参见 *TZC 的时钟分配*一节。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能 pclk5 和 aclk_tzc1 时钟，读取“1”表示这些时钟已使能

位 10:9 保留，必须保持复位值。

位 8 **RTCAPBEN**：RTC APB 时钟使能 (RTC APB clock enable)

如果 RTC 的至少一个功能通过 RTC_SECCFGR 寄存器配置为安全（即如果由 RTC 驱动的输入信号 rtc_sec 被置为有效），则对该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 7:0 保留，必须保持复位值。

10.8.130 RCC APB5 外设使能清零寄存器 (RCC_MP_APB5ENCLRR)

RCC APB5 peripheral enable clear register

偏移地址：0x744

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENCEN	Res.	Res.	Res.	BSECEN
											rc_w1				rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDG1APBEN	Res.	ETZPCEN	Res.	TZCEN	Res.	Res.	RTCAPBEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rc_w1		rc_w1		rc_w1			rc_w1								

位 31:21 保留，必须保持复位值。

位 20 **STGENCEN**：STGEN 控制器部分，外设时钟使能 (STGEN controller part, peripheral clocks enable)

如果 STGENC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

STGEN 控制器部分的外设时钟是 pclk5，以及由 RCC_STGENCKSELR 寄存器选定的内核时钟。

0：写入“0”无影响，读取“0”表示已禁止 pclk5，且内核时钟门控由 RCC_MP_APB4ENxxxR.STGENROEN (xxx=SET/CLR) 定义

1：写入“1”禁止 pclk5，并会禁止内核时钟（当 RCC_MP_APB4ENSETR.STGENROEN=0 时，否则无影响）；读取“1”表示 pclk5 和内核时钟已使能

位 19:17 保留，必须保持复位值。

位 16 **BSECEN**：BSEC 外设时钟使能 (BSEC peripheral clock enable)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 15 **IWDG1APBEN**：IWDG1 APB 时钟使能 (IWDG1 APB clock enable)

如果 IWDG1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示 APB 时钟已禁止

1：写入“1”禁止 APB 时钟，读取“1”表示 APB 时钟已使能

位 14 保留，必须保持复位值。

位 13 **ETZPCEN**：ETZPC 外设时钟使能 (ETZPC peripheral clock enable)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 12 保留，必须保持复位值。

位 11 **TZCEN**：TZC 时钟使能 (TZC clocks enable)

由（安全）软件清零。

有关详细信息，请参见 *TZC 的时钟分配*一节。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止 pclk5 和 aclk_tzc1 时钟，读取“1”表示这些时钟已使能

位 10:9 保留，必须保持复位值。

位 8 **RTCAPBEN**：RTC APB 时钟使能 (RTC APB clock enable)

如果 RTC 的至少一个功能通过 RTC_SECCFGR 寄存器配置为安全（即如果由 RTC 驱动的输入信号 rtc_sec 被置为有效），则对该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 7:0 保留，必须保持复位值。

10.8.131 RCC APB6 外设使能设置寄存器 (RCC_MP_APB6ENSETR)

RCC APB6 peripheral enable set register

偏移地址：0x748

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TIM17EN	TIM16EN	TIM15EN	TIM14EN	TIM13EN	TIM12EN	I2C5EN	I2C4EN	I2C3EN	SPI5EN	SPI4EN	USART2EN	USART1EN
			rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:13 保留，必须保持复位值。

位 12 **TIM17EN**：TIM17 外设时钟 (TIM17 peripherals clocks)

如果 TIM17 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 11 **TIM16EN**：TIM16 外设时钟 (TIM16 peripherals clocks)

如果 TIM16 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 10 **TIM15EN**：TIM15 外设时钟 (TIM15 peripherals clocks)

如果 TIM15 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 9 **TIM14EN**：TIM14 外设时钟 (TIM14 peripherals clocks)

如果 TIM14 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 8 **TIM13EN**：TIM13 外设时钟 (TIM13 peripherals clocks)

如果 TIM13 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 7 **TIM12EN**：TIM12 外设时钟 (TIM12 peripherals clocks)

如果 TIM12 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 6 **I2C5EN**：I2C5 外设时钟 (I2C5 peripherals clocks)

如果 I2C5 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 5 **I2C4EN**：I2C4 外设时钟 (I2C4 peripherals clocks)

如果 I2C4 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 4 **I2C3EN**：I2C3 外设时钟 (I2C3 peripherals clocks)

如果 I2C3 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 3 **SPI5EN**：SPI5 外设时钟 (SPI5 peripherals clocks)

如果 SPI5 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 2 **SPI4EN**：SPI4 外设时钟 (SPI4 peripherals clocks)

如果 SPI4 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 1 **USART2EN**：USART2 外设时钟 (USART2 peripherals clocks)

如果 USART2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 0 **USART1EN**：USART1 外设时钟 (USART1 peripherals clocks)

如果 USART1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

10.8.132 RCC APB6 外设使能清零寄存器 (RCC_MP_APB6ENCLRR)

RCC APB6 peripheral enable clear register

偏移地址：0x74C

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TIM17EN	TIM16EN	TIM15EN	TIM14EN	TIM13EN	TIM12EN	I2C5EN	I2C4EN	I2C3EN	SPI5EN	SPI4EN	USART2EN	USART1EN
			rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:13 保留，必须保持复位值。

位 12 **TIM17EN**：TIM17 外设时钟 (TIM17 peripherals clocks)

如果 TIM17 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 11 **TIM16EN**：TIM16 外设时钟 (TIM16 peripherals clocks)

如果 TIM16 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 10 **TIM15EN**：TIM15 外设时钟 (TIM15 peripherals clocks)

如果 TIM15 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 9 **TIM14EN**：TIM14 外设时钟 (TIM14 peripherals clocks)

如果 TIM14 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 8 **TIM13EN**：TIM13 外设时钟 (TIM13 peripherals clocks)

如果 TIM13 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 7 **TIM12EN**：TIM12 外设时钟 (TIM12 peripherals clocks)

如果 TIM12 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 6 **I2C5EN**：I2C5 外设时钟 (I2C5 peripherals clocks)

如果 I2C5 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 5 **I2C4EN**：I2C4 外设时钟 (I2C4 peripherals clocks)

如果 I2C4 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 4 **I2C3EN**：I2C3 外设时钟 (I2C3 peripherals clocks)

如果 I2C3 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 3 **SPI5EN**：SPI5 外设时钟 (SPI5 peripherals clocks)

如果 SPI5 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 2 **SPI4EN**：SPI4 外设时钟 (SPI4 peripherals clocks)

如果 SPI4 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 1 **USART2EN**：USART2 外设时钟 (USART2 peripherals clocks)

如果 USART2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 0 **USART1EN**：USART1 外设时钟 (USART1 peripherals clocks)

如果 USART1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

10.8.133 RCC AHB2 外设使能设置寄存器 (RCC_MP_AHB2ENSETR)

RCC AHB2 peripheral enable set register

偏移地址：0x750

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBOEN	Res.	ADC2EN	ADC1EN	DMAMUX2EN	DMA3EN	DMAMUX1EN	DMA2EN	DMA1EN
							rs		rs	rs	rs	rs	rs	rs	rs

位 31:9 保留，必须保持复位值。

位 8 **USBOEN**：USBO 外设时钟使能 (USBO peripheral clocks enable)

如果 USBO 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 7 保留，必须保持复位值。

位 6 **ADC2EN**：ADC2 外设时钟使能 (ADC2 peripheral clocks enable)

如果 ADC2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 5 **ADC1EN**：ADC1 外设时钟使能 (ADC1 peripheral clocks enable)

如果 ADC1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 4 **DMAMUX2EN**：DMAMUX2 外设时钟使能 (DMAMUX2 peripheral clocks enable)

如果 DMAMUX2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 3 **DMA3EN**：DMA3 外设时钟使能 (DMA3 peripheral clocks enable)

如果 DMA3 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 2 **DMAMUX1EN**：DMAMUX1 外设时钟使能 (DMAMUX1 peripheral clocks enable)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 1 **DMA2EN**：DMA2 外设时钟使能 (DMA2 peripheral clocks enable)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 0 **DMA1EN**：DMA1 外设时钟使能 (DMA1 peripheral clocks enable)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

10.8.134 RCC AHB2 外设使能清零寄存器 (RCC_MP_AHB2ENCLRR)

RCC AHB2 peripheral enable clear register

偏移地址：0x754

复位值：0x0000 0000

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBOEN	Res.	ADC2EN	ADC1EN	DMAMUX2EN	DMA3EN	DMAMUX1EN	DMA2EN	DMA1EN
							rc_w1		rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:9 保留，必须保持复位值。

位 8 **USBOEN**：USBO 外设时钟使能 (USBO peripheral clocks enable)

如果 USBO 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 7 保留，必须保持复位值。

位 6 **ADC2EN**：ADC2 外设时钟使能 (ADC2 peripheral clocks enable)

如果 ADC2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 5 **ADC1EN**：ADC1 外设时钟使能 (ADC1 peripheral clocks enable)

如果 ADC1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 4 **DMAMUX2EN**：DMAMUX2 外设时钟使能 (DMAMUX2 peripheral clocks enable)

如果 DMAMUX2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 3 **DMA3EN**：DMA3 外设时钟使能 (DMA3 peripheral clocks enable)

如果 DMA3 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 2 **DMAMUX1EN**：DMAMUX1 外设时钟使能 (DMAMUX1 peripheral clocks enable)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 1 **DMA2EN**：DMA2 外设时钟使能 (DMA2 peripheral clocks enable)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 0 **DMA1EN**：DMA1 外设时钟使能 (DMA1 peripheral clocks enable)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

10.8.135 RCC 安全 AHB4 外设使能设置寄存器 (RCC_MP_S_AHB4ENSETR)

RCC secure AHB4 peripheral enable set register

偏移地址：0x768

复位值：0x0000 0000

该安全寄存器用于使能安全 (AHB4) GPIO 端口的时钟，如 GPIO 寄存器文件所定义，特别是 GPIOxSECCFGR.SECy 控制位（x=A..I，y=0..15，或者 x=I，y=2）。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
							rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:9 保留，必须保持复位值。

位 8 **GPIOIEN**：GPIOI 外设时钟使能 (GPIOI peripheral clocks enable)
由（安全）软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 7 **GPIOHEN**：GPIOH 外设时钟使能 (GPIOH peripheral clocks enable)
由（安全）软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 6 **GPIOGEN**：GPIOG 外设时钟使能 (GPIOG peripheral clocks enable)
由（安全）软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 5 **GPIOFEN**：GPIOF 外设时钟使能 (GPIOF peripheral clocks enable)
由（安全）软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 4 **GPIOEEN**：GPIOE 外设时钟使能 (GPIOE peripheral clocks enable)
由（安全）软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 3 **GPIODEN**：GPIOD 外设时钟使能 (GPIOD peripheral clocks enable)
由（安全）软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 2 **GPIOCEN**：GPIOC 外设时钟使能 (GPIOC peripheral clocks enable)
由（安全）软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 1 **GPIOBEN**：GPIOB 外设时钟使能 (GPIOB peripheral clocks enable)
由（安全）软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 0 **GPIOAEN**：GPIOA 外设时钟使能 (GPIOA peripheral clocks enable)
由（安全）软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

10.8.136 RCC 安全 AHB4 外设使能清零寄存器 (RCC_MP_S_AHB4ENCLRR)

RCC secure AHB4 peripheral enable clear register

偏移地址：0x76C

复位值：0x0000 0000

该安全寄存器用于禁止安全 (AHB4) GPIO 端口，如 GPIO 寄存器文件所定义，特别是 GPIOxSECCFGR.SECy 控制位（x=A..I，y=0..15，或者 x=I，y=2）。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
							rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:9 保留，必须保持复位值。

位 8 **GPIOIEN**：GPIOI 外设时钟使能 (GPIOI peripheral clocks enable)
由（安全）软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 7 **GPIOHEN**：GPIOH 外设时钟使能 (GPIOH peripheral clocks enable)
由（安全）软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 6 **GPIOGEN**：GPIOG 外设时钟使能 (GPIOG peripheral clocks enable)
由（安全）软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 5 **GPIOFEN**：GPIOF 外设时钟使能 (GPIOF peripheral clocks enable)
由（安全）软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 4 **GPIOEEN**：GPIOE 外设时钟使能 (GPIOE peripheral clocks enable)
由（安全）软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 3 **GPIODEN**：GPIOD 外设时钟使能 (GPIOD peripheral clocks enable)
由（安全）软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 2 **GPIOCEN**：GPIOC 外设时钟使能 (GPIOC peripheral clocks enable)
由（安全）软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 1 **GPIOBEN**：GPIOB 外设时钟使能 (GPIOB peripheral clocks enable)
由（安全）软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 0 **GPIOAEN**：GPIOA 外设时钟使能 (GPIOA peripheral clocks enable)
由（安全）软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

10.8.137 RCC 非安全 AHB4 外设使能设置寄存器 (RCC_MP_NS_AHB4ENSETR)

RCC non-secure AHB4 peripheral enable set register

偏移地址：0x770

复位值：0x0000 0000

该非安全寄存器用于使能非安全 GPIO 端口，如 GPIO 寄存器文件所定义，特别是 GPIOxSECCFGR.SECy 控制位（x=A..I，y=0..15，或者 x=I，y=2）。

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
							rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:9 保留，必须保持复位值。

位 8 **GPIOIEN**：GPIOI 外设时钟使能 (GPIOI peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 7 **GPIOHEN**：GPIOH 外设时钟使能 (GPIOH peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 6 **GPIOGEN**：GPIOG 外设时钟使能 (GPIOG peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 5 **GPIOFEN**：GPIOF 外设时钟使能 (GPIOF peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 4 **GPIOEEN**：GPIOE 外设时钟使能 (GPIOE peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 3 **GPIODEN**：GPIOD 外设时钟使能 (GPIOD peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 2 **GPIOCEN**：GPIOC 外设时钟使能 (GPIOC peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 1 **GPIOBEN**：GPIOB 外设时钟使能 (GPIOB peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 0 **GPIOAEN**：GPIOA 外设时钟使能 (GPIOA peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

10.8.138 RCC 非安全 AHB4 外设使能清零寄存器 (RCC_MP_NS_AHB4ENCLRR)

RCC non-secure AHB4 peripheral enable clear register

偏移地址：0x774

复位值：0x0000 0000

该非安全寄存器用于禁止非安全 GPIO 端口，如 GPIO 寄存器文件所定义，特别是 GPIOxSECCFGR.SECy 控制位（x=A..I，y=0..15，或者 x=I，y=2）。

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
							rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:9 保留，必须保持复位值。

位 8 **GPIOIEN**：GPIOI 外设时钟使能 (GPIOI peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 7 **GPIOHEN**：GPIOH 外设时钟使能 (GPIOH peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 6 **GPIOGEN**：GPIOG 外设时钟使能 (GPIOG peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 5 **GPIOFEN**：GPIOF 外设时钟使能 (GPIOF peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 4 **GPIOEEN**：GPIOE 外设时钟使能 (GPIOE peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 3 **GPIODEN**：GPIOD 外设时钟使能 (GPIOD peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 2 **GPIOCEN**：GPIOC 外设时钟使能 (GPIOC peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 1 **GPIOBEN**：GPIOB 外设时钟使能 (GPIOB peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 0 **GPIOAEN**：GPIOA 外设时钟使能 (GPIOA peripheral clocks enable)
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

10.8.139 RCC AHB5 外设使能设置寄存器 (RCC_MP_AHB5ENSETR)

RCC AHB5 peripheral enable set register

偏移地址：0x778

复位值：0x0001 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器为 MPU 分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXIMCEN
															rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKPSRAMEN	Res.	RNG1EN	HASH1EN	CRYP1EN	SAESEN	PKAEN	Res.	Res.
							rs		rs	rs	rs	rs	rs		

位 31:17 保留，必须保持复位值。

位 16 **AXIMCEN**：AXIMC 时钟使能 (AXIMC clocks enable)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 15:9 保留，必须保持复位值。

位 8 **BKPSRAMEN**：BKPSRAM 时钟使能 (BKPSRAM clocks enable)

如果 BKPSRAM 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 7 保留，必须保持复位值。

位 6 **RNG1EN**：RNG1 外设时钟使能 (RNG1 peripheral clocks enable)

如果 RNG1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 5 **HASH1EN**：HASH 外设时钟使能 (HASH peripheral clocks enable)

如果 HASH 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 4 **CRYP1EN**：CRYP (3DES/AES) 外设时钟使能 (CRYP (3DES/AES) peripheral clocks enable)

如果 CRYP 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 3 **SAESEN**：SAES 外设时钟使能 (SAES peripheral clocks enable)

如果 SAES 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 2 **PKAEN**：PKA 外设时钟使能 (PKA peripheral clocks enable)

如果 PKA 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 1:0 保留，必须保持复位值。

10.8.140 RCC AHB5 外设使能清零寄存器 (RCC_MP_AHB5ENCLRR)

RCC AHB5 peripheral enable clear register

偏移地址：0x77C

复位值：0x0001 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXIMCEN
															rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKPSRAMEN	Res.	RNG1EN	HASH1EN	CRYP1EN	SAESEN	PKAEN	Res.	Res.
							rc_w1		rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		

位 31:17 保留，必须保持复位值。

位 16 **AXIMCEN**：AXIMC 时钟使能 (AXIMC clocks enable)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。
由（安全）软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 15:9 保留，必须保持复位值。

位 8 **BKPSRAMEN**：BKPSRAM 时钟使能 (BKPSRAM clocks enable)

如果 BKPSRAM 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 7 保留，必须保持复位值。

位 6 **RNG1EN**：RNG1 外设时钟使能 (RNG1 peripheral clocks enable)

如果 RNG1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 5 **HASH1EN**：HASH 外设时钟使能 (HASH peripheral clocks enable)

如果 HASH 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。
由软件清零。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 4 **CRYP1EN**：CRYP (3DES/AES) 外设时钟使能 (CRYP (3DES/AES) peripheral clocks enable)

如果 CRYP 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 3 **SAESEN**：SAES 外设时钟使能 (SAES peripheral clocks enable)

如果 SAES 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 2 **PKAEN**：PKA 外设时钟使能 (PKA peripheral clocks enable)

如果 PKA 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 1:0 保留，必须保持复位值。

10.8.141 RCC AHB6 外设使能设置寄存器 (RCC_MP_AHB6ENSETR)

RCC AHB6 peripheral enable set register

偏移地址：0x780

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	ETH2MACEN	ETH2RXEN	ETH2TXEN	ETH2CKEN	Res.	Res.	USBHEN	Res.	Res.	Res.	CRC1EN	Res.	Res.	SDMMC2EN	SDMMC1EN
	rs	rs	rs	rs			rs				rs			rs	rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	QSPIEN	Res.	FMCEN	Res.	ETH1MACEN	ETH1RXEN	ETH1TXEN	ETH1CKEN	Res.	Res.	Res.	Res.	Res.	MCEEN	Res.
	rs		rs		rs	rs	rs	rs						rs	

位 31 保留，必须保持复位值。

位 30 **ETH2MACEN**：ETH2 MAC 总线接口时钟使能 (ETH2 MAC bus interface clock enable)（hclk6 和 aclk）

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示总线接口时钟已禁止

1：写入“1”使能总线接口时钟，读取“1”表示总线接口时钟已使能

位 29 **ETH2RXEN**：ETH2 接收时钟使能 (ETH2 reception clock enable)

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示接收时钟已禁止

1：写入“1”使能接收时钟，读取“1”表示接收时钟已使能

位 28 **ETH2TXEN**：ETH2 发送时钟使能 (ETH2 Transmission clock enable)

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示发送时钟已禁止

1：写入“1”使能发送时钟，读取“1”表示发送时钟已使能

位 27 **ETH2CKEN**：使能 RCC 生成的 ETH2 时钟 (Enable of the ETH2 clock generated by the RCC) (eth2_ker_ck)

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示 eth2_ker_ck 时钟已禁止

1：写入“1”使能 eth2_ker_ck 时钟，读取“1”表示 eth2_ker_ck 时钟已使能

位 26:25 保留，必须保持复位值。

位 24 **USBHEN**：USBH 外设时钟使能 (USBH peripheral clocks enable)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 23:21 保留，必须保持复位值。

位 20 **CRC1EN**：CRC 外设时钟使能 (CRC peripheral clocks enable)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 19:18 保留，必须保持复位值。

位 17 **SDMMC2EN**：SDMMC2 和 SDMMC2 延迟 (DLYBSD2) 模块外设时钟使能 (SDMMC2 and SDMMC2 delay (DLYBSD2) block peripheral clocks enable)

如果 SDMMC2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 16 **SDMMC1EN**：SDMMC1 和 SDMMC1 延迟 (DLYBSD1) 模块外设时钟使能 (SDMMC1 and SDMMC1 delay (DLYBSD1) block peripheral clocks enable)

如果 SDMMC1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 15 保留，必须保持复位值。

位 14 **QSPIEN**：QUADSPI 外设时钟使能 (QUADSPI peripheral clock enable)

如果 QSPI 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 13 保留，必须保持复位值。

位 12 **FMCEN**：FMC 外设时钟使能 (FMC peripheral clocks enable)

如果 FMC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 11 保留，必须保持复位值。

位 10 **ETH1MACEN**：ETH1 MAC 总线接口时钟使能 (ETH1 MAC bus interface clock enable)（hclk6 和 aclk）

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示总线接口时钟已禁止

1：写入“1”使能总线接口时钟，读取“1”表示总线接口时钟已使能

位 9 **ETH1RXEN**：ETH1 接收时钟使能 (ETH1 reception clock enable)

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示接收时钟已禁止

1：写入“1”使能接收时钟，读取“1”表示接收时钟已使能

位 8 **ETH1TXEN**：ETH1 发送时钟使能 (ETH1 transmission clock enable)

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示发送时钟已禁止

1：写入“1”使能发送时钟，读取“1”表示发送时钟已使能

位 7 **ETH1CKEN**：使能 RCC 生成的 ETH1 时钟 (Enable of the ETH1 clock generated by the RCC) (eth1_ker_ck)

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示 eth1_ker_ck 时钟已禁止

1：写入“1”使能 eth1_ker_ck 时钟，读取“1”表示 eth1_ker_ck 时钟已使能

位 6:2 保留，必须保持复位值。

位 1 **MCEEN**：MCE 外设时钟使能 (MCE peripheral clocks enable)

如果 MCE 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

位 0 保留，必须保持复位值。

10.8.142 RCC AHB6 外设使能清零寄存器 (RCC_MP_AHB6ENCLRR)

RCC AHB6 peripheral enable clear register

偏移地址：0x784

复位值：0x0000 0000

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	ETH2MACEN	ETH2RXEN	ETH2TXEN	ETH2CKEN	Res.	Res.	USBHEN	Res.	Res.	Res.	CRC1EN	Res.	Res.	SDMMC2EN	SDMMC1EN
	rc_w1	rc_w1	rc_w1	rc_w1			rc_w1				rc_w1			rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	QSPIEN	Res.	FMCEN	Res.	ETH1MACEN	ETH1RXEN	ETH1TXEN	ETH1CKEN	Res.	Res.	Res.	Res.	Res.	MCEEN	Res.
	rc_w1		rc_w1		rc_w1	rc_w1	rc_w1	rc_w1						rc_w1	

位 31 保留，必须保持复位值。

位 30 **ETH2MACEN**：ETH2 MAC 总线接口时钟使能 (ETH2 MAC bus interface clock enable)（hclk6 和 aclk）

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示总线接口时钟已禁止

1：写入“1”禁止总线接口时钟，读取“1”表示总线接口时钟已使能

位 29 **ETH2RXEN**：ETH2 接收时钟使能 (ETH2 reception clock enable)

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示接收时钟已禁止

1：写入“1”禁止接收时钟，读取“1”表示接收时钟已使能

位 28 **ETH2TXEN**：ETH2 发送时钟使能 (ETH2 Transmission clock enable)

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示发送时钟已禁止

1：写入“1”禁止发送时钟，读取“1”表示发送时钟已使能

位 27 **ETH2CKEN**：使能 RCC 生成的 ETH2 时钟 (Enable of the ETH2 clock generated by the RCC) (eth2_ker_ck)

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示 eth2_ker_ck 时钟已禁止

1：写入“1”禁止 eth2_ker_ck 时钟，读取“1”表示 eth2_ker_ck 时钟已使能

位 26:25 保留，必须保持复位值。

位 24 **USBHEN**：USBH 外设时钟使能 (USBH peripheral clocks enable)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 23:21 保留，必须保持复位值。

位 20 **CRC1EN**：CRC 外设时钟使能 (CRC peripheral clocks enable)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 19:18 保留，必须保持复位值。

位 17 **SDMMC2EN**：SDMMC2 和 SDMMC2 延迟 (DLYBSD2) 模块外设时钟使能 (SDMMC2 and SDMMC2 delay (DLYBSD2) block peripheral clocks enable)

如果 SDMMC2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 16 **SDMMC1EN**：SDMMC1 和 SDMMC1 延迟 (DLYBSD1) 模块外设时钟使能 (SDMMC1 and SDMMC1 delay (DLYBSD1) block peripheral clocks enable)

如果 SDMMC1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 15 保留，必须保持复位值。

位 14 **QSPIEN**：QUADSPI 外设时钟使能 (QUADSPI peripheral clock enable)

如果 QSPI 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 13 保留，必须保持复位值。

位 12 **FMCEN**：FMC 外设时钟使能 (FMC peripheral clocks enable)

如果 FMC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 11 保留，必须保持复位值。

位 10 **ETH1MACEN**：ETH1 MAC 总线接口时钟使能 (ETH1 MAC bus interface clock enable)（hclk6 和 aclk）

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示总线接口时钟已禁止

1：写入“1”禁止总线接口时钟，读取“1”表示总线接口时钟已使能

位 9 **ETH1RXEN**：ETH1 接收时钟使能 (ETH1 reception clock enable)

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示接收时钟已禁止

1：写入“1”禁止接收时钟，读取“1”表示接收时钟已使能

位 8 **ETH1TXEN**：ETH1 发送时钟使能 (ETH1 transmission clock enable)

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示发送时钟已禁止

1：写入“1”禁止发送时钟，读取“1”表示发送时钟已使能

位 7 **ETH1CKEN**：使能 RCC 生成的 ETH1 时钟 (Enable of the ETH1 clock generated by the RCC) (eth1_ker_ck)

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示 eth1_ker_ck 时钟已禁止

1：写入“1”禁止 eth1_ker_ck 时钟，读取“1”表示 eth1_ker_ck 时钟已使能

位 6:2 保留，必须保持复位值。

位 1 **MCEEN**：MCE 外设时钟使能 (MCE peripheral clocks enable)

如果 MCE 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

位 0 保留，必须保持复位值。

10.8.143 RCC 安全 AHB6 外设使能设置寄存器 (RCC_MP_S_AHB6ENSETR)

RCC secure AHB6 peripheral enable set register

偏移地址：0x788

复位值：0x0000 0000

这是一个安全寄存器，用于在 MDMA 作为安全使用时使能其时钟。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MDMAEN
															rs

位 31:1 保留，必须保持复位值。

位 0 **MDMAEN**：MDMA 外设时钟使能 (MDMA peripheral clocks enable)

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

10.8.144 RCC 安全 AHB6 外设使能清零寄存器 (RCC_MP_S_AHB6ENCLRR)

RCC secure AHB6 peripheral enable clear register

偏移地址：0x78C

复位值：0x0000 0000

这是一个安全寄存器，用于在 MDMA 作为安全使用时禁止其时钟。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MDMAEN
															rc_w1

位 31:1 保留，必须保持复位值。

位 0 **MDMAEN**：MDMA 外设时钟使能 (MDMA peripheral clocks enable)

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟已禁止

1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

10.8.145 RCC 非安全 AHB6 外设使能设置寄存器 (RCC_MP_NS_AHB6ENSETR)

RCC non-secure AHB6 peripheral enable set register

偏移地址：0x790

复位值：0x0000 0000

这是一个非安全寄存器，用于在 MDMA 作为非安全使用时使能其时钟。

该寄存器用于将相应外设的外设时钟使能位置 1。必须使用该寄存器分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MDMAEN
															rs

位 31:1 保留，必须保持复位值。

位 0 **MDMAEN**：MDMA 外设时钟使能 (MDMA peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”使能外设时钟，读取“1”表示外设时钟已使能

10.8.146 RCC 非安全 AHB6 外设使能清零寄存器 (RCC_MP_NS_AHB6ENCLRR)

RCC non-secure AHB6 peripheral enable clear register

偏移地址：0x794

复位值：0x0000 0000

这是一个非安全寄存器，用于在 MDMA 作为非安全使用时禁止其时钟。

该寄存器用于将相应外设的外设时钟使能位清零。必须使用该寄存器取消分配外设。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MDMAEN
															rc_w1

位 31:1 保留，必须保持复位值。

位 0 **MDMAEN**：MDMA 外设时钟使能 (MDMA peripheral clocks enable)
由软件置 1。
0：写入“0”无影响，读取“0”表示外设时钟已禁止
1：写入“1”禁止外设时钟，读取“1”表示外设时钟已使能

10.8.147 RCC APB1 睡眠时钟使能设置寄存器 (RCC_MP_APB1LPENSETR)

RCC APB1 Sleep clock enable set register

偏移地址：0x800

复位值：0x046F 9A3F

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SPDIFLPEN	Res.	Res.	Res.	I2C2LPEN	I2C1LPEN	Res.	UART8LPEN	UART7LPEN	UART5LPEN	UART4LPEN
					rs				rs	rs		rs	rs	rs	rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USART3LPEN	Res.	Res.	SPI3LPEN	SPI2LPEN	Res.	LPTIM1LPEN	Res.	Res.	Res.	TIM7LPEN	TIM6LPEN	TIM5LPEN	TIM4LPEN	TIM3LPEN	TIM2LPEN
rs			rs	rs		rs				rs	rs	rs	rs	rs	rs

位 31:27 保留，必须保持复位值。

位 26 **SPDIFLPEN**：CSleep 模式期间的 SPDIFRX 外设时钟使能 (SPDIFRX peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 25:23 保留，必须保持复位值。

位 22 **I2C2LPEN**：CSleep 模式期间的 I2C2 外设时钟使能 (I2C2 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 21 **I2C1LPEN**：CSleep 模式期间的 I2C1 外设时钟使能 (I2C1 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 20 保留，必须保持复位值。

位 19 **UART8LPEN**：CSleep 模式期间的 UART8 外设时钟使能 (UART8 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 18 **UART7LPEN**：CSleep 模式期间的 UART7 外设时钟使能 (UART7 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 17 **UART5LPEN**：CSleep 模式期间的 UART5 外设时钟使能 (UART5 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 16 **UART4LPEN**：CSleep 模式期间的 UART4 外设时钟使能 (UART4 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 15 **USART3LPEN**：CSleep 模式期间的 USART3 外设时钟使能 (USART3 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 14:13 保留，必须保持复位值。

位 12 **SPI3LPEN**：CSleep 模式期间的 SPI3 外设时钟使能 (SPI3 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 11 **SPI2LPEN**：CSleep 模式期间的 SPI2 外设时钟使能 (SPI2 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 10 保留，必须保持复位值。

位 9 **LPTIM1LPEN**：CSleep 模式期间的 LPTIM1 外设时钟使能 (LPTIM1 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 8:6 保留，必须保持复位值。

位 5 **TIM7LPEN**：CSleep 模式期间的 TIM7 外设时钟使能 (TIM7 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 4 **TIM6LPEN**：CSleep 模式期间的 TIM6 外设时钟使能 (TIM6 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 3 **TIM5LPEN**：CSleep 模式期间的 TIM5 外设时钟使能 (TIM5 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 2 **TIM4LPEN**：CSleep 模式期间的 TIM4 外设时钟使能 (TIM4 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 1 **TIM3LPEN**：CSleep 模式期间的 TIM3 外设时钟使能 (TIM3 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 **TIM2LPEN**：CSleep 模式期间的 TIM2 外设时钟使能 (TIM2 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.148 RCC APB1 睡眠时钟使能清零寄存器 (RCC_MP_APB1LPENCLRR)

RCC APB1 Sleep clock enable clear register

偏移地址：0x804

复位值：0x046F 9A3F

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SPDIFLPEN	Res.	Res.	Res.	I2C2LPEN	I2C1LPEN	Res.	UART8LPEN	UART7LPEN	UART5LPEN	UART4LPEN
					rc_w1				rc_w1	rc_w1		rc_w1	rc_w1	rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USART3LPEN	Res.	Res.	SPI3LPEN	SPI2LPEN	Res.	LPTIM1LPEN	Res.	Res.	Res.	TIM7LPEN	TIM6LPEN	TIM5LPEN	TIM4LPEN	TIM3LPEN	TIM2LPEN
rc_w1			rc_w1	rc_w1		rc_w1				rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:27 保留，必须保持复位值。

位 26 **SPDIFLPEN**：CSleep 模式期间的 SPDIFRX 外设时钟使能 (SPDIFRX peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 25:23 保留，必须保持复位值。

位 22 **I2C2LPEN**：CSleep 模式期间的 I2C2 外设时钟使能 (I2C2 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 21 **I2C1LPEN**：CSleep 模式期间的 I2C1 外设时钟使能 (I2C1 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 20 保留，必须保持复位值。

位 19 **UART8LPEN**：CSleep 模式期间的 UART8 外设时钟使能 (UART8 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 18 **UART7LPEN**：CSleep 模式期间的 UART7 外设时钟使能 (UART7 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 17 **UART5LPEN**：CSleep 模式期间的 UART5 外设时钟使能 (UART5 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 16 **UART4LPEN**：CSleep 模式期间的 UART4 外设时钟使能 (UART4 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 15 **USART3LPEN**：CSleep 模式期间的 USART3 外设时钟使能 (USART3 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 14:13 保留，必须保持复位值。

位 12 **SPI3LPEN**：CSleep 模式期间的 SPI3 外设时钟使能 (SPI3 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 11 **SPI2LPEN**：CSleep 模式期间的 SPI2 外设时钟使能 (SPI2 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 10 保留，必须保持复位值。

位 9 **LPTIM1LPEN**：CSleep 模式期间的 LPTIM1 外设时钟使能 (LPTIM1 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 8:6 保留，必须保持复位值。

位 5 **TIM7LPEN**：CSleep 模式期间的 TIM7 外设时钟使能 (TIM7 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 4 **TIM6LPEN**：CSleep 模式期间的 TIM6 外设时钟使能 (TIM6 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 3 **TIM5LPEN**：CSleep 模式期间的 TIM5 外设时钟使能 (TIM5 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 2 **TIM4LPEN**：CSleep 模式期间的 TIM4 外设时钟使能 (TIM4 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 1 **TIM3LPEN**：CSleep 模式期间的 TIM3 外设时钟使能 (TIM3 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 **TIM2LPEN**：CSleep 模式期间的 TIM2 外设时钟使能 (TIM2 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.149 RCC APB2 睡眠时钟使能设置寄存器 (RCC_MP_APB2LPENSETR)

RCC APB2 Sleep clock enable set register

偏移地址：0x808

复位值：0x0133 2103

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANLPEN	Res.	Res.	ADFSDMLPEN	DFSDMLPEN	Res.	Res.	SAI2LPEN	SAI1LPEN
							rs			rs	rs			rs	rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	USART6LPEN	Res.	Res.	Res.	Res.	SPI1LPEN	Res.	Res.	Res.	Res.	Res.	Res.	TIM8LPEN	TIM1LPEN
		rs					rs							rs	rs

位 31:25 保留，必须保持复位值。

位 24 **FDCANLPEN**：CSleep 模式期间的 FDCAN 和 CANRAM 外设时钟使能 (FDCAN and CANRAM peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 23:22 保留，必须保持复位值。

位 21 **ADFSDMLPEN**：CSleep 模式期间的音频 DFSDM 外设时钟使能 (Audio DFSDM peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 20 **DFSDMLPEN**：CSleep 模式期间的 DFSDM 外设时钟使能 (DFSDM peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 19:18 保留，必须保持复位值。

位 17 **SAI2LPEN**：CSleep 模式期间的 SAI2 外设时钟使能 (SAI2 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 16 **SAI1LPEN**：CSleep 模式期间的 SAI1 外设时钟使能 (SAI1 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 15:14 保留，必须保持复位值。

位 13 **USART6LPEN**：CSleep 模式期间的 USART6 外设时钟使能 (USART6 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 12:9 保留，必须保持复位值。

位 8 **SPI1LPEN**：CSleep 模式期间的 SPI/I2S1 外设时钟使能 (SPI/I2S1 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 7:2 保留，必须保持复位值。

位 1 **TIM8LPEN**：CSleep 模式期间的 TIM8 外设时钟使能 (TIM8 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 **TIM1LPEN**：CSleep 模式期间的 TIM1 外设时钟使能 (TIM1 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.150 RCC APB2 睡眠时钟使能清零寄存器 (RCC_MP_APB2LPENCLRR)

RCC APB2 Sleep clock enable clear register

偏移地址：0x80C

复位值：0x0133 2103

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANLPEN	Res.	Res.	ADFSDMLPEN	DFSDMLPEN	Res.	Res.	SAI2LPEN	SAI1LPEN
							rc_w1			rc_w1	rc_w1			rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	USART6LPEN	Res.	Res.	Res.	Res.	SPI1LPEN	Res.	Res.	Res.	Res.	Res.	Res.	TIM8LPEN	TIM1LPEN
		rc_w1					rc_w1							rc_w1	rc_w1

位 31:25 保留，必须保持复位值。

位 24 **FDCANLPEN**：CSleep 模式期间的 FDCAN 和 CANRAM 外设时钟使能 (FDCAN and CANRAM peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 23:22 保留，必须保持复位值。

位 21 **ADFSDMLPEN**：CSleep 模式期间的音频 DFSDM 外设时钟使能 (Audio DFSDM peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 20 **DFSDMLPEN**：CSleep 模式期间的 DFSDM 外设时钟使能 (DFSDM peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 19:18 保留，必须保持复位值。

位 17 **SAI2LPEN**：CSleep 模式期间的 SAI2 外设时钟使能 (SAI2 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 16 **SAI1LPEN**：CSleep 模式期间的 SAI1 外设时钟使能 (SAI1 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 15:14 保留，必须保持复位值。

位 13 **USART6LPEN**：CSleep 模式期间的 USART6 外设时钟使能 (USART6 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 12:9 保留，必须保持复位值。

位 8 **SPI1LPEN**：CSleep 模式期间的 SPI/I2S1 外设时钟使能 (SPI/I2S1 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 7:2 保留，必须保持复位值。

位 1 **TIM8LPEN**：CSleep 模式期间的 TIM8 外设时钟使能 (TIM8 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 **TIM1LPEN**：CSleep 模式期间的 TIM1 外设时钟使能 (TIM1 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.151 RCC APB3 睡眠时钟使能设置寄存器 (RCC_MP_APB3LPENSETR)

RCC APB3 Sleep clock enable set register

偏移地址：0x810

复位值：0x0003 200F

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTSLPEN
															rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	VREFLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM5LPEN	LPTIM4LPEN	LPTIM3LPEN	LPTIM2LPEN
		rs										rs	rs	rs	rs

位 31:18 保留，必须保持复位值。

位 17 保留，必须保持复位值。

位 16 **DTSLPEN**：CSleep 模式期间的 DTS 外设时钟使能 (DTS peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 15:14 保留，必须保持复位值。

位 13 **VREFLPEN**：CSleep 模式期间的 VREF 外设时钟使能 (VREF peripheral clocks enable during CSleep mode)

如果 VREF 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 12:4 保留，必须保持复位值。

位 3 **LPTIM5LPEN**：CSleep 模式期间的 LPTIM5 外设时钟使能 (LPTIM5 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 2 **LPTIM4LPEN**：CSleep 模式期间的 LPTIM4 外设时钟使能 (LPTIM4 peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 1 **LPTIM3LPEN**：CSleep 模式期间的 LPTIM3 外设时钟使能 (LPTIM3 peripheral clocks enable during CSleep mode)

如果 LPTIM3 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 **LPTIM2LPEN**：CSleep 模式期间的 LPTIM2 外设时钟使能 (LPTIM2 peripheral clocks enable during CSleep mode)

如果 LPTIM2 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.152 RCC APB3 睡眠时钟使能清零寄存器 (RCC_MP_APB3LPENCLRR)

RCC APB3 Sleep clock enable clear register

偏移地址：0x814

复位值：0x0003 200F

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTSLPEN
															rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	VREFLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM5LPEN	LPTIM4LPEN	LPTIM3LPEN	LPTIM2LPEN
		rc_w1										rc_w1	rc_w1	rc_w1	rc_w1

位 31:18 保留，必须保持复位值。

位 17 保留，必须保持复位值。

位 16 **DTSLPEN**：CSleep 模式期间的 DTS 外设时钟使能 (DTS peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 15:14 保留，必须保持复位值。

位 13 **VREFLPEN**：CSleep 模式期间的 VREF 外设时钟使能 (VREF peripheral clocks enable during CSleep mode)

如果 VREF 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 12:4 保留，必须保持复位值。

位 3 **LPTIM5LPEN**：CSleep 模式期间的 LPTIM5 外设时钟使能 (LPTIM5 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 2 **LPTIM4LPEN**：CSleep 模式期间的 LPTIM4 外设时钟使能 (LPTIM4 peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 1 **LPTIM3LPEN**：CSleep 模式期间的 LPTIM3 外设时钟使能 (LPTIM3 peripheral clocks enable during CSleep mode)

如果 LPTIM3 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 **LPTIM2LPEN**：CSleep 模式期间的 LPTIM2 外设时钟使能 (LPTIM2 peripheral clocks enable during CSleep mode)

如果 LPTIM2 模块通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.153 RCC 安全 APB3 睡眠时钟使能设置寄存器 (RCC_MP_S_APB3LPENSETR)

RCC secure APB3 Sleep clock enable set register

偏移地址：0x818

复位值：0x0000 0001

该安全寄存器用于使能 SYSCFG 的时钟。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGLPEN
															rs

位 31:1 保留，必须保持复位值。

位 0 **SYSCFGLPEN**：CSleep 模式期间的 SYSCFG 外设时钟使能 (SYSCFG peripheral clocks enable during CSleep mode)

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.154 RCC 安全 APB3 睡眠时钟使能清零寄存器 (RCC_MP_S_APB3LPENCLRR)

RCC secure APB3 Sleep clock enable clear register

偏移地址：0x81C

复位值：0x0000 0001

该安全寄存器用于禁止 SYSCFG 的时钟。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGLPEN
															rc_w1

位 31:1 保留，必须保持复位值。

位 0 **SYSCFGLPEN**：CSleep 模式期间的 SYSCFG 外设时钟使能 (SYSCFG peripheral clocks enable during CSleep mode)

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.155 RCC 非安全 APB3 睡眠时钟使能设置寄存器 (RCC_MP_NS_APB3LPENSETR)

RCC non-secure APB3 Sleep clock enable set register

偏移地址：0x820

复位值：0x0000 0001

该非安全寄存器用于使能 SYSCFG 的时钟。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGLPEN
															rs

位 31:1 保留，必须保持复位值。

位 0 **SYSCFGLPEN**：CSleep 模式期间的 SYSCFG 外设时钟使能 (SYSCFG peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.156 RCC 非安全 APB3 睡眠时钟使能清零寄存器 (RCC_MP_NS_APB3LPENCLRR)

RCC non-secure APB3 Sleep clock enable clear register

偏移地址：0x824

复位值：0x0000 0001

该非安全寄存器用于禁止 SYSCFG 的时钟。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGLPEN
															rs

位 31:1 保留，必须保持复位值。

位 0 **SYSCFGLPEN**：CSleep 模式期间的 SYSCFG 外设时钟使能 (SYSCFG peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.157 RCC APB4 睡眠时钟使能设置寄存器 (RCC_MP_APB4LPENSETR)

RCC APB4 Sleep clock enable set register

偏移地址：0x828

复位值：0x0011 8102

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENROSTPEN	STGENROLPEN	Res.	Res.	Res.	USBPHYLPEN
										rs	rs				rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDG2APBLPEN	Res.	Res.	Res.	Res.	Res.	Res.	DDRPERFMLPEN	Res.	Res.	Res.	Res.	Res.	Res.	DCMIPPLPEN	Res.
rs							rs							rs	

位 31:22 保留，必须保持复位值。

位 21 **STGENROSTPEN**：CStop 模式期间 STGEN 只读接口外设时钟使能 (STGEN read-only interface peripheral clocks enable during CStop mode)

由软件置 1。
STGEN 只读接口的外设时钟是 pclk4，以及由 RCC_STGENCKSELR 寄存器选定的内核时钟。
0：写入“0”无影响，读取“0”表示在 CStop 模式下已禁止 pclk4，且内核时钟门控由 RCC_MP_APB5ENxxxR.STGENCEN (xxx=SET/CLR) 定义
1：写入“1”在 CStop 模式下使能 pclk4 和内核时钟；读取“1”表示 pclk4 和内核时钟在 CStop 模式下已使能

位 20 **STGENROLPEN**：CSleep 模式期间 STGEN 只读接口外设时钟使能 (STGEN read-only interface peripheral clocks enable during CSleep mode)

由软件置 1。
STGEN 只读接口的外设时钟是 pclk4，以及由 RCC_STGENCKSELR 寄存器选定的内核时钟。
0：写入“0”无影响，读取“0”表示在 CSleep 模式下已禁止 pclk4，且内核时钟门控由 RCC_MP_APB5ENxxxR.STGENCEN (xxx=SET/CLR) 定义
1：写入“1”在 CSleep 模式下使能 pclk4 和内核时钟；读取“1”表示 pclk4 和内核时钟在 CSleep 模式下已使能

位 19:17 保留，必须保持复位值。

位 16 **USBPHYLPEN**：CSleep 模式期间 USBPHYC 外设时钟使能 (USBPHYC peripheral clocks enable during CSleep mode)

如果 USB PHY 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 15 **IWDG2APBLPEN**：CSleep 模式期间 IWDG2 APB 时钟使能 (IWDG2 APB clock enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示 APB 时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能 APB 时钟，读取“1”表示 APB 时钟在 CSleep 模式下已使能

位 14:9 保留，必须保持复位值。

位 8 **DDRPERFMLPEN**：CSleep 模式期间 DDRPERFM APB 时钟使能 (DDRPERFM APB clock enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示 APB 时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能 APB 时钟，读取“1”表示 APB 时钟在 CSleep 模式下已使能

位 7:2 保留，必须保持复位值。

位 1 **DCMIPPLPEN**：CSleep 模式期间 DCMIPP 外设时钟使能 (DCMIPP peripheral clocks enable during CSleep mode)

如果 DCMIPP 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 保留，必须保持复位值。

10.8.158 RCC APB4 睡眠时钟使能清零寄存器(RCC_MP_APB4LPENCLRR)

RCC APB4 Sleep clock enable clear register

偏移地址：0x82C

复位值：0x0011 8102

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENROSTPEN	STGENROLPEN	Res.	Res.	Res.	USBPHYLPEN
										rc_w1	rc_w1				rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDG2APBLPEN	Res.	Res.	Res.	Res.	Res.	Res.	DDRPERFMLPEN	Res.	Res.	Res.	Res.	Res.	Res.	DCMIPPLPEN	Res.
rc_w1							rc_w1							rc_w1	

位 31:22 保留，必须保持复位值。

位 21 STGENROSTPEN: CStop 模式期间 STGEN 只读接口外设时钟使能 (STGEN read-only interface peripheral clocks enable during CStop mode)

由软件清零。

STGEN 只读接口的外设时钟是 pclk4，以及由 RCC_STGENCKSELR 寄存器选定的内核时钟。

0：写入“0”无影响，读取“0”表示在 CStop 模式下已禁止 pclk4，且内核时钟门控由 RCC_MP_APB5LPENxxxR.STGENCLPEN (xxx=SET/CLR) 定义

1：写入“1”在 CStop 模式下禁止 pclk4 和内核时钟；读取“1”表示 pclk4 和内核时钟在 CStop 模式下已使能

位 20 **STGENROLPEN**：CSleep 模式期间 STGEN 只读接口外设时钟使能 (STGEN read-only interface peripheral clocks enable during CSleep mode)

由软件清零。

STGEN 只读接口的外设时钟是 pclk4，以及由 RCC_STGENCKSELR 寄存器选定的内核时钟。

0：写入“0”无影响，读取“0”表示在 CSleep 模式下已禁止 pclk4，且内核时钟门控由 RCC_MP_APB5LPENxxxR.STGENCLPEN (xxx=SET/CLR) 定义

1：写入“1”在 CSleep 模式下禁止 pclk4 和内核时钟；读取“1”表示 pclk4 和内核时钟在 CSleep 模式下已使能

位 19:17 保留，必须保持复位值。

位 16 **USBPHYLPEN**：CSleep 模式期间 USBPHYC 外设时钟使能 (USBPHYC peripheral clocks enable during CSleep mode)

如果 USB PHY 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 15 **IWDG2APBLPEN**：CSleep 模式期间 IWDG2 APB 时钟使能 (IWDG2 APB clock enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示 APB 时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止 APB 时钟，读取“1”表示 APB 时钟在 CSleep 模式下已使能

位 14:9 保留，必须保持复位值。

位 8 **DDRPERFMLPEN**：CSleep 模式期间 DDRPERFM APB 时钟使能 (DDRPERFM APB clock enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示 APB 时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止 APB 时钟，读取“1”表示 APB 时钟在 CSleep 模式下已使能

位 7:2 保留，必须保持复位值。

位 1 **DCMIPPLPEN**：CSleep 模式期间 DCMIPP 外设时钟使能 (DCMIPP peripheral clocks enable during CSleep mode)

如果 DCMIPP 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 保留，必须保持复位值。

10.8.159 RCC 安全 APB4 睡眠外设使能设置寄存器 (RCC_MP_S_APB4LPENSETR)

RCC secure APB4 sleep peripheral enable set register

偏移地址：0x830

复位值：0x0000 0001

这是一个安全寄存器，用于在 LTDC 与安全合成层一起使用时使能其时钟。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCLPEN
															rs

位 31:1 保留，必须保持复位值。

位 0 **LTDCLPEN**：CSleep 模式期间 LTDC 外设时钟使能 (LTDC peripheral clocks enable during CSleep mode)

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟已使能

10.8.160 RCC 安全 APB4 睡眠外设使能清零寄存器 (RCC_MP_S_APB4LPENCLRR)

RCC secure APB4 sleep peripheral enable clear register

偏移地址：0x834

复位值：0x0000 0001

这是一个安全寄存器，用于在 LTDC 与安全合成层一起使用时禁止其时钟。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCLPEN
															rc_w1

位 31:1 保留，必须保持复位值。

位 0 **LTDCLPEN**：CSleep 模式期间 LTDC 外设时钟使能 (LTDC peripheral clocks enable during CSleep mode)

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟已使能

10.8.161 RCC 非安全 APB4 睡眠外设使能设置寄存器 (RCC_MP_NS_APB4LPENSETR)

RCC non-secure APB4 sleep peripheral enable set register

偏移地址：0x838

复位值：0x0000 0001

这是一个非安全寄存器，用于在 LTDC 与一个或两个非安全合成层一起使用时使能其时钟。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCLPEN
															rs

位 31:1 保留，必须保持复位值。

位 0 **LTDCLPEN**：CSleep 模式期间 LTDC 外设时钟使能 (LTDC peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟已使能

10.8.162 RCC 非安全 APB4 睡眠外设使能清零寄存器 (RCC_MP_NS_APB4LPENCLRR)

RCC non-secure APB4 sleep peripheral enable clear register

偏移地址：0x83C

复位值：0x0000 0001

这是一个非安全寄存器，用于在 LTDC 与一个或两个非安全合成层一起使用时禁止其时钟。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCLPEN
															rc_w1

位 31:1 保留，必须保持复位值。

位 0 **LTDCLPEN**：CSleep 模式期间 LTDC 外设时钟使能 (LTDC peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟已使能

10.8.163 RCC APB5 睡眠时钟使能设置寄存器 (RCC_MP_APB5LPENSETR)

RCC APB5 Sleep clock enable set register

偏移地址：0x840

复位值：0x0011 A900

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENCSTPEN	STGENCLPEN	Res.	Res.	Res.	BSECLPEN
										rs	rs				rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDG1APBLPEN	Res.	ETZPCLPEN	Res.	TZCLPEN	Res.	Res.	RTCAPBLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs		rs		rs			rs								

位 31:22 保留，必须保持复位值。

位 21 **STGENCSTPEN**：CStop 模式期间 STGEN 控制器部分，外设时钟使能 (STGEN controller part, peripheral clocks enable during CStop mode)

如果 STGENC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

STGEN 控制器部分的外设时钟是 pclk5，以及由 RCC_STGENCKSELR 寄存器选定的内核时钟。

0：写入“0”无影响，读取“0”表示在 CStop 模式下已禁止 pclk5，且内核时钟门控由 RCC_MP_APB4ENxxxR.STGENROSTPEN (xxx=SET/CLR) 定义

1：写入“1”在 CStop 模式下使能 pclk5 和内核时钟；读取“1”表示 pclk5 和内核时钟在 CStop 模式下已使能

位 20 **STGENCLPEN**：CSleep 模式期间 STGEN 控制器部分，外设时钟使能 (STGEN controller part, peripheral clocks enable during CSleep mode)

如果 STGENC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

STGEN 控制器部分的外设时钟是 pclk5，以及由 RCC_STGENCKSELR 寄存器选定的内核时钟。

0：写入“0”无影响，读取“0”表示在 CSleep 模式下已禁止 pclk5，且内核时钟门控由 RCC_MP_APB4ENxxxR.STGENROLPEN (xxx=SET/CLR) 定义

1：写入“1”在 CSleep 模式下使能 pclk5 和内核时钟；读取“1”表示 pclk5 和内核时钟在 CSleep 模式下已使能

位 19:17 保留，必须保持复位值。

位 16 **BSECLPEN**：CSleep 模式期间 BSEC 外设时钟使能 (BSEC peripheral clocks enable during CSleep mode)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 15 **IWDG1APBLPEN**：CSleep 模式期间 IWDG1 APB 时钟使能 (IWDG1 APB clock enable during CSleep mode)

如果 IWDG1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示 APB 时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能 APB 时钟，读取“1”表示 APB 时钟在 CSleep 模式下已使能

位 14 保留，必须保持复位值。

位 13 **ETZPCLPEN**：CSleep 模式期间 ETZPC 外设时钟使能 (ETZPC peripheral clocks enable during CSleep mode)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 12 保留，必须保持复位值。

位 11 **TZCLPEN**：CSleep 模式期间 TZC 时钟使能 (TZC clocks enable during CSleep mode)

由（安全）软件置 1。

有关详细信息，请参见 *TZC 的时钟分配*一节。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能 pclk5 和 aclk_tzc1 时钟，读取“1”表示这些时钟在 CSleep 模式下已使能

位 10:9 保留，必须保持复位值。

位 8 **RTCAPBLPEN**：CSleep 模式期间 RTC APB 时钟使能 (RTC APB clock enable during CSleep mode)

如果 RTC 的至少一个功能通过 RTC_SECCFGR 寄存器配置为安全（即如果由 RTC 驱动的输入信号 rtc_sec 被置为有效），则对该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 7:0 保留，必须保持复位值。

10.8.164 RCC APB5 睡眠时钟使能清零寄存器 (RCC_MP_APB5LPENCLRR)

RCC APB5 Sleep clock enable clear register

偏移地址：0x844

复位值：0x0011 A900

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENCSTPEN	STGENCLPEN	Res.	Res.	Res.	BSECLPEN
										rc_w1	rc_w1				rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDG1APBLPEN	Res.	ETZPCLPEN	Res.	TZCLPEN	Res.	Res.	RTCAPBLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rc_w1		rc_w1		rc_w1			rc_w1								

位 31:22 保留，必须保持复位值。

位 21 **STGENCSTPEN**：CStop 模式期间 STGEN 控制器部分，外设时钟使能 (STGEN controller part, peripheral clocks enable during CStop mode)

如果 STGENC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

STGEN 控制器部分的外设时钟是 pclk5，以及由 RCC_STGENCKSELR 寄存器选定的内核时钟。

0：写入“0”无影响，读取“0”表示在 CStop 模式下已禁止 pclk5，且内核时钟门控由 RCC_MP_APB4LPENxxxR.STGENROSTPEN (xxx=SET/CLR) 定义

1：写入“1”在 CStop 模式下禁止 pclk5 和内核时钟；读取“1”表示 pclk5 和内核时钟在 CStop 模式下已使能

位 20 **STGENCLPEN**：CSleep 模式期间 STGEN 控制器部分，外设时钟使能 (STGEN controller part, peripheral clocks enable during CSleep mode)

如果 STGENC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

STGEN 控制器部分的外设时钟是 pclk5，以及由 RCC_STGENCKSELR 寄存器选定的内核时钟。

0：写入“0”无影响，读取“0”表示在 CSleep 模式下已禁止 pclk5，且内核时钟门控由 RCC_MP_APB4LPENxxxR.STGENROLPEN (xxx=SET/CLR) 定义

1：写入“1”在 CSleep 模式下禁止 pclk5 和内核时钟；读取“1”表示 pclk5 和内核时钟在 CSleep 模式下已使能

位 19:17 保留，必须保持复位值。

位 16 **BSECLPEN**：CSleep 模式期间 BSEC 外设时钟使能 (BSEC peripheral clocks enable during CSleep mode)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 15 **IWDG1APBLPEN**：CSleep 模式期间 IWDG1 APB 时钟使能 (IWDG1 APB clock enable during CSleep mode)

如果 IWDG1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示 APB 时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止 APB 时钟，读取“1”表示 APB 时钟在 CSleep 模式下已使能

位 14 保留，必须保持复位值。

位 13 **ETZPCLPEN**：CSleep 模式期间 ETZPC 外设时钟使能 (ETZPC peripheral clocks enable during CSleep mode)

该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 12 保留，必须保持复位值。

位 11 **TZCLPEN**：CSleep 模式期间 TZC 时钟使能 (TZC clocks enable during CSleep mode)

由（安全）软件清零。

有关详细信息，请参见 *TZC 的时钟分配*一节。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止 pclk5 和 aclk_tzc1 时钟，读取“1”表示这些时钟在 CSleep 模式下已使能

位 10:9 保留，必须保持复位值。

位 8 **RTCAPBLPEN**：CSleep 模式期间 RTC APB 时钟使能 (RTC APB clock enable during CSleep mode)

如果 RTC 的至少一个功能通过 RTC_SECCFGR 寄存器配置为安全（即如果由 RTC 驱动的输入信号 rtc_sec 被置为有效），则对该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 7:0 保留，必须保持复位值。

10.8.165 RCC APB6 睡眠时钟使能设置寄存器 (RCC_MP_APB6LPENSETR)

RCC APB6 sleep clock enable set register

偏移地址：0x848

复位值：0x0000 1FFF

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TIM17LPEN	TIM16LPEN	TIM15LPEN	TIM14LPEN	TIM13LPEN	TIM12LPEN	I2C5LPEN	I2C4LPEN	I2C3LPEN	SPI5LPEN	SPI4LPEN	USART2LPEN	USART1LPEN
			rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:13 保留，必须保持复位值。

位 12 **TIM17LPEN**：CSleep 模式期间 TIM17 外设时钟使能 (TIM17 peripheral clocks enable during CSleep mode)

如果 TIM17 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 11 **TIM16LPEN**：CSleep 模式期间 TIM16 外设时钟使能 (TIM16 peripheral clocks enable during CSleep mode)

如果 TIM16 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 10 **TIM15LPEN**：CSleep 模式期间 TIM15 外设时钟使能 (TIM15 peripheral clocks enable during CSleep mode)

如果 TIM15 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 9 **TIM14LPEN**：CSleep 模式期间 TIM14 外设时钟使能 (TIM14 peripheral clocks enable during CSleep mode)

如果 TIM14 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 8 **TIM13LPEN**：CSleep 模式期间 TIM13 外设时钟使能 (TIM13 peripheral clocks enable during CSleep mode)

如果 TIM13 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 7 **TIM12LPEN**：CSleep 模式期间 TIM12 外设时钟使能 (TIM12 peripheral clocks enable during CSleep mode)

如果 TIM12 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 6 **I2C5LPEN**：CSleep 模式期间 I2C5 外设时钟使能 (I2C5 peripheral clocks enable during CSleep mode)

如果 I2C5 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 5 **I2C4LPEN**：CSleep 模式期间 I2C4 外设时钟使能 (I2C4 peripheral clocks enable during CSleep mode)

如果 I2C4 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 4 **I2C3LPEN**：CSleep 模式期间 I2C3 外设时钟使能 (I2C3 peripheral clocks enable during CSleep mode)

如果 I2C3 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 3 **SPI5LPEN**：CSleep 模式期间 SPI5 外设时钟使能 (SPI5 peripheral clocks enable during CSleep mode)

如果 SPI5 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 2 **SPI4LPEN**：CSleep 模式期间 SPI4 外设时钟使能 (SPI4 peripheral clocks enable during CSleep mode)

如果 SPI4 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 1 **USART2LPEN**：CSleep 模式期间 USART2 外设时钟使能 (USART2 peripheral clocks enable during CSleep mode)

如果 USART2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 **USART1LPEN**：CSleep 模式期间 USART1 外设时钟使能 (USART1 peripheral clocks enable during CSleep mode)

如果 USART1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.166 RCC APB6 睡眠时钟使能清零寄存器 (RCC_MP_APB6LPENCLRR)

RCC APB6 sleep clock enable clear register

偏移地址：0x84C

复位值：0x0000 1FFF

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TIM17LPEN	TIM16LPEN	TIM15LPEN	TIM14LPEN	TIM13LPEN	TIM12LPEN	I2C5LPEN	I2C4LPEN	I2C3LPEN	SPI5LPEN	SPI4LPEN	USART2LPEN	USART1LPEN
			rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:13 保留，必须保持复位值。

位 12 **TIM17LPEN**：CSleep 模式期间 TIM17 外设时钟使能 (TIM17 peripheral clocks enable during CSleep mode)

如果 TIM17 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 11 **TIM16LPEN**：CSleep 模式期间 TIM16 外设时钟使能 (TIM16 peripheral clocks enable during CSleep mode)

如果 TIM16 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 10 **TIM15LPEN**：CSleep 模式期间 TIM15 外设时钟使能 (TIM15 peripheral clocks enable during CSleep mode)

如果 TIM15 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 9 **TIM14LPEN**：CSleep 模式期间 TIM14 外设时钟使能 (TIM14 peripheral clocks enable during CSleep mode)

如果 TIM14 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 8 **TIM13LPEN**：CSleep 模式期间 TIM13 外设时钟使能 (TIM13 peripheral clocks enable during CSleep mode)

如果 TIM13 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 7 **TIM12LPEN**：CSleep 模式期间 TIM12 外设时钟使能 (TIM12 peripheral clocks enable during CSleep mode)

如果 TIM12 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 6 **I2C5LPEN**：CSleep 模式期间 I2C5 外设时钟使能 (I2C5 peripheral clocks enable during CSleep mode)

如果 I2C5 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 5 **I2C4LPEN**：CSleep 模式期间 I2C4 外设时钟使能 (I2C4 peripheral clocks enable during CSleep mode)

如果 I2C4 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 4 **I2C3LPEN**：CSleep 模式期间 I2C3 外设时钟使能 (I2C3 peripheral clocks enable during CSleep mode)

如果 I2C3 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 3 **SPI5LPEN**：CSleep 模式期间 SPI5 外设时钟使能 (SPI5 peripheral clocks enable during CSleep mode)

如果 SPI5 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 2 **SPI4LPEN**：CSleep 模式期间 SPI4 外设时钟使能 (SPI4 peripheral clocks enable during CSleep mode)

如果 SPI4 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 1 **USART2LPEN**：CSleep 模式期间 USART2 外设时钟使能 (USART2 peripheral clocks enable during CSleep mode)

如果 USART2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 **USART1LPEN**：CSleep 模式期间 USART1 外设时钟使能 (USART1 peripheral clocks enable during CSleep mode)

如果 USART1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.167 RCC AHB2 睡眠时钟使能设置寄存器 (RCC_MP_AHB2LPENSETR)

RCC AHB2 Sleep clock enable set register

偏移地址：0x850

复位值：0x0000 017F

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBOLPEN	Res.	ADC2LPEN	ADC1LPEN	DMAMUX2LPEN	DMA3LPEN	DMAMUX1LPEN	DMA2LPEN	DMA1LPEN
							rs		rs	rs	rs	rs	rs	rs	rs

位 31:9 保留，必须保持复位值。

位 8 **USBOLPEN**：睡眠模式期间 USBO 外设时钟使能 (USBO peripheral clocks enable during Sleep mode)

如果 USBO 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 7 保留，必须保持复位值。

位 6 **ADC2LPEN**：睡眠模式期间 ADC2 外设时钟使能 (ADC2 peripheral clocks enable during Sleep mode)

如果 ADC2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 5 **ADC1LPEN**：睡眠模式期间 ADC1 外设时钟使能 (ADC1 peripheral clocks enable during Sleep mode)

如果 ADC1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 4 **DMAMUX2LPEN**：睡眠模式期间 DMAMUX2 外设时钟使能 (DMAMUX2 peripheral clocks enable during Sleep mode)

如果 DMAMUX2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 3 **DMA3LPEN**：睡眠模式期间 DMA3 外设时钟使能 (DMA3 peripheral clocks enable during Sleep mode)

如果 DMA3 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 2 **DMAMUX1LPEN**：睡眠模式期间 DMAMUX1 外设时钟使能 (DMAMUX1 peripheral clocks enable during Sleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 1 **DMA2LPEN**：睡眠模式期间 DMA2 外设时钟使能 (DMA2 peripheral clocks enable during Sleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 **DMA1LPEN**：睡眠模式期间 DMA1 外设时钟使能 (DMA1 peripheral clocks enable during Sleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.168 RCC AHB2 睡眠时钟使能清零寄存器 (RCC_MP_AHB2LPENCLRR)

RCC AHB2 Sleep clock enable clear register

偏移地址：0x854

复位值：0x0000 017F

该寄存器可在位字段级单独、安全地受到写保护，具体取决于相应外设的安全状态，即取决于 ETZPC 寄存器的相应 DECPROT[1] 位字段。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBOLPEN	Res.	ADC2LPEN	ADC1LPEN	DMAMUX2LPEN	DMA3LPEN	DMAMUX1LPEN	DMA2LPEN	DMA1LPEN
							rc_w1		rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:9 保留，必须保持复位值。

位 8 **USBOLPEN**：睡眠模式期间 USBO 外设时钟使能 (USBO peripheral clocks enable during Sleep mode)

如果 USBO 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 7 保留，必须保持复位值。

位 6 **ADC2LPEN**：睡眠模式期间 ADC2 外设时钟使能 (ADC2 peripheral clocks enable during Sleep mode)

如果 ADC2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 5 **ADC1LPEN**：睡眠模式期间 ADC1 外设时钟使能 (ADC1 peripheral clocks enable during Sleep mode)

如果 ADC1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 4 **DMAMUX2LPEN**：睡眠模式期间 DMAMUX2 外设时钟使能 (DMAMUX2 peripheral clocks enable during Sleep mode)

如果 DMAMUX2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 3 **DMA3LPEN**：睡眠模式期间 DMA3 外设时钟使能 (DMA3 peripheral clocks enable during Sleep mode)

如果 DMA3 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 2 **DMAMUX1LPEN**：睡眠模式期间 DMAMUX1 外设时钟使能 (DMAMUX1 peripheral clocks enable during Sleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 1 **DMA2LPEN**：睡眠模式期间 DMA2 外设时钟使能 (DMA2 peripheral clocks enable during Sleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 **DMA1LPEN**：睡眠模式期间 DMA1 外设时钟使能 (DMA1 peripheral clocks enable during Sleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.169 RCC 安全 AHB4 睡眠时钟使能设置寄存器 (RCC_MP_S_AHB4LPENSETR)

RCC secure AHB4 Sleep clock enable set register

偏移地址：0x868

复位值：0x0000 01FF

该安全寄存器用于使能安全 (AHB4) GPIO 端口的时钟，如 GPIO 寄存器文件所定义，特别是 GPIOxSECCFGR.SECy 控制位（x=A..I，y=0..15，或者 x=I，y=2）。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOILPEN	GPIOHLPEN	GPIOGLPEN	GPIOFLPEN	GPIOELPEN	GPIODLPEN	GPIOCLPEN	GPIOBLPEN	GPIOALPEN
							rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:9 保留，必须保持复位值。

位 8 **GPIOILPEN**：CSleep 模式期间 GPIOI 外设时钟使能 (GPIOI peripheral clocks enable during CSleep mode)

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 7 **GPIOHLPEN**：CSleep 模式期间 GPIOH 外设时钟使能 (GPIOH peripheral clocks enable during CSleep mode)

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 6 **GPIOGLPEN**：CSleep 模式期间 GPIOG 外设时钟使能 (GPIOG peripheral clocks enable during CSleep mode)

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 5 **GPIOFLPEN**：CSleep 模式期间 GPIOF 外设时钟使能 (GPIOF peripheral clocks enable during CSleep mode)

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 4 **GPIOELPEN**：CSleep 模式期间 GPIOE 外设时钟使能 (GPIOE peripheral clocks enable during CSleep mode)

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 3 **GPIODLPEN**：CSleep 模式期间 GPIOD 外设时钟使能 (GPIOD peripheral clocks enable during CSleep mode)

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 2 **GPIOCLPEN**：CSleep 模式期间 GPIOC 外设时钟使能 (GPIOC peripheral clocks enable during CSleep mode)

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 1 **GPIOBLPEN**：CSleep 模式期间 GPIOB 外设时钟使能 (GPIOB peripheral clocks enable during CSleep mode)

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 **GPIOALPEN**：CSleep 模式期间 GPIOA 外设时钟使能 (GPIOA peripheral clocks enable during CSleep mode)

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.170 RCC 安全 AHB4 睡眠时钟使能清零寄存器 (RCC_MP_S_AHB4LPENCLRR)

RCC secure AHB4 Sleep clock enable clear register

偏移地址：0x86C

复位值：0x0000 01FF

该安全寄存器用于禁止安全 (AHB4) GPIO 端口的时钟，如 GPIO 寄存器文件所定义，特别是 GPIOxSECCFGR.SECy 控制位（x=A..I，y=0..15，或者 x=I，y=2）。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOILPEN	GPIOHLPEN	GPIOGLPEN	GPIOFLPEN	GPIOELPEN	GPIODLPEN	GPIOCLPEN	GPIOBLPEN	GPIOALPEN
							rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:9 保留，必须保持复位值。

位 8 **GPIOILPEN**：CSleep 模式期间 GPIOI 外设时钟使能 (GPIOI peripheral clocks enable during CSleep mode)

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 7 **GPIOHLPEN**：CSleep 模式期间 GPIOH 外设时钟使能 (GPIOH peripheral clocks enable during CSleep mode)

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 6 **GPIOGLPEN**：CSleep 模式期间 GPIOG 外设时钟使能 (GPIOG peripheral clocks enable during CSleep mode)

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 5 **GPIOFLPEN**：CSleep 模式期间 GPIOF 外设时钟使能 (GPIOF peripheral clocks enable during CSleep mode)

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 4 **GPIOELPEN**：CSleep 模式期间 GPIOE 外设时钟使能 (GPIOE peripheral clocks enable during CSleep mode)

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 3 **GPIODLPEN**：CSleep 模式期间 GPIOD 外设时钟使能 (GPIOD peripheral clocks enable during CSleep mode)

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 2 **GPIOCLPEN**：CSleep 模式期间 GPIOC 外设时钟使能 (GPIOC peripheral clocks enable during CSleep mode)

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 1 **GPIOBLPEN**：CSleep 模式期间 GPIOB 外设时钟使能 (GPIOB peripheral clocks enable during CSleep mode)

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 **GPIOALPEN**：CSleep 模式期间 GPIOA 外设时钟使能 (GPIOA peripheral clocks enable during CSleep mode)

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.171 RCC 非安全 AHB4 睡眠时钟使能设置寄存器 (RCC_MP_NS_AHB4LPENSETR)

RCC non-secure AHB4 Sleep clock enable set register

偏移地址：0x870

复位值：0x0000 01FF

该非安全寄存器用于使能非安全 (AHB4) GPIO 端口的时钟，如 GPIO 寄存器文件所定义，特别是 GPIOxSECCFGR.SECy 控制位（x=A..I，y=0..15，或者 x=I，y=2）。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOILPEN	GPIOHLPEN	GPIOGLPEN	GPIOFLPEN	GPIOELPEN	GPIODLPEN	GPIOCLPEN	GPIOBLPEN	GPIOALPEN
							rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:9 保留，必须保持复位值。

位 8 **GPIOILPEN**：CSleep 模式期间 GPIOI 外设时钟使能 (GPIOI peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 7 **GPIOHLPEN**：CSleep 模式期间 GPIOH 外设时钟使能 (GPIOH peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 6 **GPIOGLPEN**：CSleep 模式期间 GPIOG 外设时钟使能 (GPIOG peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 5 **GPIOFLPEN**：CSleep 模式期间 GPIOF 外设时钟使能 (GPIOF peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 4 **GPIOELPEN**：CSleep 模式期间 GPIOE 外设时钟使能 (GPIOE peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 3 **GPIODLPEN**：CSleep 模式期间 GPIOD 外设时钟使能 (GPIOD peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 2 **GPIOCLPEN**：CSleep 模式期间 GPIOC 外设时钟使能 (GPIOC peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 1 **GPIOBLPEN**：CSleep 模式期间 GPIOB 外设时钟使能 (GPIOB peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 **GPIOALPEN**：CSleep 模式期间 GPIOA 外设时钟使能 (GPIOA peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.172 RCC 非安全 AHB4 睡眠时钟使能清零寄存器 (RCC_MP_NS_AHB4LPENCLRR)

RCC non-secure AHB4 Sleep clock enable clear register

偏移地址：0x874

复位值：0x0000 01FF

该非安全寄存器用于禁止非安全 (AHB4) GPIO 端口的时钟，如 GPIO 寄存器文件所定义，特别是 GPIOxSECCFGR.SECy 控制位（x=A..I，y=0..15，或者 x=I，y=2）。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将相应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOILPEN	GPIOHLPEN	GPIOGLPEN	GPIOFLPEN	GPIOELPEN	GPIODLPEN	GPIOCLPEN	GPIOBLPEN	GPIOALPEN
							rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:9 保留，必须保持复位值。

位 8 **GPIOILPEN**：CSleep 模式期间 GPIOI 外设时钟使能 (GPIOI peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 7 **GPIOHLPEN**：CSleep 模式期间 GPIOH 外设时钟使能 (GPIOH peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 6 **GPIOGLPEN**：CSleep 模式期间 GPIOG 外设时钟使能 (GPIOG peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 5 **GPIOFLPEN**：CSleep 模式期间 GPIOF 外设时钟使能 (GPIOF peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 4 **GPIOELPEN**：CSleep 模式期间 GPIOE 外设时钟使能 (GPIOE peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 3 **GPIODLPEN**：CSleep 模式期间 GPIOD 外设时钟使能 (GPIOD peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 2 **GPIOCLPEN**：CSleep 模式期间 GPIOC 外设时钟使能 (GPIOC peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 1 **GPIOBLPEN**：CSleep 模式期间 GPIOB 外设时钟使能 (GPIOB peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 **GPIOALPEN**：CSleep 模式期间 GPIOA 外设时钟使能 (GPIOA peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.173 RCC AHB5 睡眠时钟使能设置寄存器 (RCC_MP_AHB5LPENSETR)

RCC AHB5 Sleep clock enable set register

偏移地址：0x878

复位值：0x0000 017C

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKPSRAMLPEN	Res.	RNG1LPEN	HASH1LPEN	CRYP1LPEN	SAESLPEN	PKALPEN	Res.	Res.
							rs		rs	rs	rs	rs	rs		

位 31:9 保留，必须保持复位值。

位 8 **BKPSRAMLPEN**：CSleep 模式期间 BKPSRAM 时钟使能 (BKPSRAM clocks enable during CSleep mode)

如果 BKPSRAM 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 7 保留，必须保持复位值。

位 6 **RNG1LPEN**：CSleep 模式期间 RNG1 外设时钟使能 (RNG1 peripheral clocks enable during CSleep mode)

如果 RNG1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 5 **HASH1LPEN**：CSleep 模式期间 HASH 外设时钟使能 (HASH peripheral clocks enable during CSleep mode)

如果 HASH 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 4 **CRYP1LPEN**：CSleep 模式期间 CRYP (3DES/AES) 外设时钟使能 (CRYP (3DES/AES) peripheral clocks enable during CSleep mode)

如果 CRYP 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 3 **SAESLPEN**：CSleep 模式期间 SAES 外设时钟使能 (SAES peripheral clocks enable during CSleep mode)

如果 SAES 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 2 **PKALPEN**：CSleep 模式期间 PKA 外设时钟使能 (PKA peripheral clocks enable during CSleep mode)

如果 PKA 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 1:0 保留，必须保持复位值。

10.8.174 RCC AHB5 睡眠时钟使能清零寄存器 (RCC_MP_AHB5LPENCLRR)

RCC AHB5 Sleep clock enable clear register

偏移地址：0x87C

复位值：0x0000 017C

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKPSRAMLPEN	Res.	RNG1LPEN	HASH1LPEN	CRYP1LPEN	SAESLPEN	PKALPEN	Res.	Res.
							rc_w1		rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		

位 31:9 保留，必须保持复位值。

位 8 **BKPSRAMLPEN**：CSleep 模式期间 BKPSRAM 时钟使能 (BKPSRAM clocks enable during CSleep mode)

如果 BKPSRAM 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 7 保留，必须保持复位值。

位 6 **RNG1LPEN**：CSleep 模式期间 RNG1 外设时钟使能 (RNG1 peripheral clocks enable during CSleep mode)

如果 RNG1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 5 **HASH1LPEN**：CSleep 模式期间 HASH 外设时钟使能 (HASH peripheral clocks enable during CSleep mode)

如果 HASH 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 4 **CRYP1LPEN**：CSleep 模式期间 CRYP (3DES/AES) 外设时钟使能 (CRYP (3DES/AES) peripheral clocks enable during CSleep mode)

如果 CRYP 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 3 **SAESLPEN**：CSleep 模式期间 SAES 外设时钟使能 (SAES peripheral clocks enable during CSleep mode)

如果 SAES 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 2 **PKALPEN**：CSleep 模式期间 PKA 外设时钟使能 (PKA peripheral clocks enable during CSleep mode)

如果 PKA 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 1:0 保留，必须保持复位值。

10.8.175 RCC AHB6 睡眠时钟使能设置寄存器 (RCC_MP_AHB6LPENSETR)

RCC AHB6 Sleep clock enable set register

偏移地址：0x880

复位值：0x7913 5782

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETH2STPEN	ETH2MACLPEN	ETH2RXLPEN	ETH2TXLPEN	ETH2CKLPEN	Res.	Res.	USBHLPEN	Res.	Res.	Res.	CRC1LPEN	Res.	Res.	SDMMC2LPEN	SDMMC1LPEN
rs	rs	rs	rs	rs			rs				rs			rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	QSPILPEN	Res.	FMCLPEN	ETH1STPEN	ETH1MACLPEN	ETH1RXLPEN	ETH1TXLPEN	ETH1CKLPEN	Res.	Res.	Res.	Res.	Res.	MCELPEN	Res.
	rs		rs	rs	rs	rs	rs	rs						rs	

位 31 **ETH2STPEN**：CStop 模式期间 ETH2 外设时钟使能 (ETH2 peripheral clock enable during CStop mode)

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1，以允许 ETH2 模块在 CStop 期间从引脚接收 ETH2_TX_CLK 和 ETH2_RX_CLK 时钟。

0：写入“0”无影响，读取“0”表示 ETH2_TX_CLK 和 ETH2_RX_CLK 时钟在 CStop 模式下已禁止

1：写入“1”在 CStop 模式下使能 ETH2_TX_CLK 和 ETH2_RX_CLK 时钟，读取“1”表示 ETH2_TX_CLK 和 ETH2_RX_CLK 时钟在 CStop 模式下已使能

位 30 **ETH2MACLPEN**：CSleep 模式期间 ETH2 MAC 总线接口时钟使能 (ETH2 MAC bus interface clock enable during CSleep mode)（hclk6 和 aclk）

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示总线接口时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能总线接口时钟，读取“1”表示总线接口时钟在 CSleep 模式下已使能

位 29 **ETH2RXLPEN**：CSleep 模式期间 ETH2 接收时钟使能 (ETH2 reception clock enable during CSleep mode)

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示接收时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能接收时钟，读取“1”表示接收时钟在 CSleep 模式下已使能

位 28 **ETH2TXLPEN**：CSleep 模式期间 ETH2 发送时钟使能 (ETH2 transmission clock enable during CSleep mode)

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示发送时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能发送时钟，读取“1”表示发送时钟在 CSleep 模式下已使能

位 27 **ETH2CKLPEN**：使能 RCC 生成的 ETH2 时钟 (Enable of the ETH2 clock generated by the RCC) (eth2_ker_ck)

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示 eth2_ker_ck 时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能 eth2_ker_ck 时钟，读取“1”表示 eth2_ker_ck 时钟在 CSleep 模式下已使能

位 26:25 保留，必须保持复位值。

位 24 **USBHLPEN**：CSleep 模式期间 USBH 外设时钟使能 (USBH peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 23:21 保留，必须保持复位值。

位 20 **CRC1LPEN**：CSleep 模式期间 CRC 外设时钟使能 (CRC peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 19:18 保留，必须保持复位值。

位 17 **SDMMC2LPEN**：CSleep 模式期间 SDMMC2 和 SDMMC2 延迟 (DLYBSD2) 模块外设时钟使能 (SDMMC2 and SDMMC2 delay (DLYBSD2) block peripheral clocks enable during CSleep mode)

如果 SDMMC2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 16 **SDMMC1LPEN**：CSleep 模式期间 SDMMC1 和 SDMMC1 延迟 (DLYBSD1) 模块外设时钟使能 (SDMMC1 and SDMMC1 delay (DLYBSD1) block peripheral clocks enable during CSleep mode)

如果 SDMMC1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 15 保留，必须保持复位值。

位 14 **QSPILPEN**：CSleep 模式期间 QUADSPI 外设时钟使能 (QUADSPI peripheral clocks enable during CSleep mode)

如果 QSPI 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 13 保留，必须保持复位值。

位 12 **FMCLPEN**：CSleep 模式期间 FMC 外设时钟使能 (FMC peripheral clocks enable during CSleep mode)

如果 FMC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 11 **ETH1STPEN**：CStop 模式期间 ETH1 外设时钟使能 (ETH1 peripheral clock enable during CStop mode)

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1，以允许 ETH1 模块在 CStop 期间从引脚接收 ETH1_TX_CLK 和 ETH1_RX_CLK 时钟。

0：写入“0”无影响，读取“0”表示 ETH1_TX_CLK 和 ETH1_RX_CLK 时钟在 CStop 模式下已禁止

1：写入“1”在 CStop 模式下使能 ETH1_TX_CLK 和 ETH1_RX_CLK 时钟，读取“1”表示 ETH1_TX_CLK 和 ETH1_RX_CLK 时钟在 CStop 模式下已使能

位 10 **ETH1MACLPEN**：CSleep 模式期间 ETH1 MAC 总线接口时钟使能 (ETH1 MAC bus interface clock enable during CSleep mode)（hclk6 和 aclk）

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示总线接口时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能总线接口时钟，读取“1”表示总线接口时钟在 CSleep 模式下已使能

位 9 **ETH1RXLPEN**：CSleep 模式期间 ETH1 接收时钟使能 (ETH1 reception clock enable during CSleep mode)

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示接收时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能接收时钟，读取“1”表示接收时钟在 CSleep 模式下已使能

位 8 **ETH1TXLPEN**：CSleep 模式期间 ETH1 发送时钟使能 (ETH1 transmission clock enable during CSleep mode)

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示发送时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能发送时钟，读取“1”表示发送时钟在 CSleep 模式下已使能

位 7 **ETH1CKLPEN**：使能 RCC 生成的 ETH1 时钟 (Enable of the ETH1 clock generated by the RCC) (eth1_ker_ck)

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示 eth1_ker_ck 时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能 eth1_ker_ck 时钟，读取“1”表示 eth1_ker_ck 时钟在 CSleep 模式下已使能

位 6:2 保留，必须保持复位值。

位 1 **MCELPEN**：CSleep 模式期间 MCE 外设时钟使能 (MCE peripheral clocks enable during CSleep mode)

如果 MCE 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 保留，必须保持复位值。

10.8.176 RCC AHB6 睡眠时钟使能清零寄存器 (RCC_MP_AHB6LPENCLRR)

RCC AHB6 Sleep clock enable clear register

偏移地址：0x884

复位值：0x7913 5782

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETH2STPEN	ETH2MACLPEN	ETH2RXLPEN	ETH2TXLPEN	ETH2CKLPEN	Res.	Res.	USBHLPEN	Res.	Res.	Res.	CRC1LPEN	Res.	Res.	SDMMC2LPEN	SDMMC1LPEN
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1			rc_w1				rc_w1			rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	QSPILPEN	Res.	FMCLPEN	ETH1STPEN	ETH1MACLPEN	ETH1RXLPEN	ETH1TXLPEN	ETH1CKLPEN	Res.	Res.	Res.	Res.	Res.	MCELPEN	Res.
	rc_w1		rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1						rc_w1	

位 31 **ETH2STPEN**：CStop 模式期间 ETH2 外设时钟使能 (ETH2 peripheral clock enable during CStop mode)

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零，以允许 ETH2 模块在 CStop 期间从引脚接收 ETH2_TX_CLK 和 ETH2_RX_CLK 时钟。

0：写入“0”无影响，读取“0”表示 ETH2_TX_CLK 和 ETH2_RX_CLK 时钟在 CStop 模式下已禁止

1：写入“1”在 CStop 模式下禁止 ETH2_TX_CLK 和 ETH2_RX_CLK 时钟，读取“1”表示 ETH2_TX_CLK 和 ETH2_RX_CLK 时钟在 CStop 模式下已使能

位 30 **ETH2MACLPEN**：CSleep 模式期间 ETH2 MAC 总线接口时钟使能 (ETH2 MAC bus interface clock enable during CSleep mode)（hclk6 和 aclk）

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示总线接口时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止总线接口时钟，读取“1”表示总线接口时钟在 CSleep 模式下已使能

位 29 **ETH2RXLPEN**：CSleep 模式期间 ETH2 接收时钟使能 (ETH2 reception clock enable during CSleep mode)

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示接收时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止接收时钟，读取“1”表示接收时钟在 CSleep 模式下已使能

位 28 **ETH2TXLPEN**：CSleep 模式期间 ETH2 发送时钟使能 (ETH2 transmission clock enable during CSleep mode)

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示发送时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止发送时钟，读取“1”表示发送时钟在 CSleep 模式下已使能

位 27 **ETH2CKLPEN**：使能 RCC 生成的 ETH2 时钟 (Enable of the ETH2 clock generated by the RCC) (eth2_ker_ck)

如果 ETH2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示 eth2_ker_ck 时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止 eth2_ker_ck 时钟，读取“1”表示 eth2_ker_ck 时钟在 CSleep 模式下已使能

位 26:25 保留，必须保持复位值。

位 24 **USBHLPEN**：CSleep 模式期间 USBH 外设时钟使能 (USBH peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 23:21 保留，必须保持复位值。

位 20 **CRC1LPEN**：CSleep 模式期间 CRC 外设时钟使能 (CRC peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 19:18 保留，必须保持复位值。

位 17 **SDMMC2LPEN**：CSleep 模式期间 SDMMC2 和 SDMMC2 延迟 (DLYBSD2) 模块外设时钟使能 (SDMMC2 and SDMMC2 delay (DLYBSD2) block peripheral clocks enable during CSleep mode)

如果 SDMMC2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 16 **SDMMC1LPEN**：CSleep 模式期间 SDMMC1 和 SDMMC1 延迟 (DLYBSD1) 模块外设时钟使能 (SDMMC1 and SDMMC1 delay (DLYBSD1) block peripheral clocks enable during CSleep mode)

如果 SDMMC1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 15 保留，必须保持复位值。

位 14 **QSPILPEN**：CSleep 模式期间 QUADSPI 外设时钟使能 (QUADSPI peripheral clocks enable during CSleep mode)

如果 QSPI 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 13 保留，必须保持复位值。

位 12 **FMCLPEN**：CSleep 模式期间 FMC 外设时钟使能 (FMC peripheral clocks enable during CSleep mode)

如果 FMC 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 11 **ETH1STPEN**：CStop 模式期间 ETH1 外设时钟使能 (ETH1 peripheral clock enable during CStop mode)

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零，以允许 ETH1 模块在 CStop 期间从引脚接收 ETH1_TX_CLK 和 ETH1_RX_CLK 时钟。

0：写入“0”无影响，读取“0”表示 ETH1_TX_CLK 和 ETH1_RX_CLK 时钟在 CStop 模式下已禁止

1：写入“1”在 CStop 模式下禁止 ETH1_TX_CLK 和 ETH1_RX_CLK 时钟，读取“1”表示 ETH1_TX_CLK 和 ETH1_RX_CLK 时钟在 CStop 模式下已使能

位 10 **ETH1MACLPEN**：CSleep 模式期间 ETH1 MAC 总线接口时钟使能 (ETH1 MAC bus interface clock enable during CSleep mode)（hclk6 和 aclk）

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示总线接口时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止总线接口时钟，读取“1”表示总线接口时钟在 CSleep 模式下已使能

位 9 **ETH1RXLPEN**：CSleep 模式期间 ETH1 接收时钟使能 (ETH1 reception clock enable during CSleep mode)

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示接收时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止接收时钟，读取“1”表示接收时钟在 CSleep 模式下已使能

位 8 **ETH1TXLPEN**：CSleep 模式期间 ETH1 发送时钟使能 (ETH1 transmission clock enable during CSleep mode)

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示发送时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止发送时钟，读取“1”表示发送时钟在 CSleep 模式下已使能

位 7 **ETH1CKLPEN**：使能 RCC 生成的 ETH1 时钟 (Enable of the ETH1 clock generated by the RCC) (eth1_ker_ck)

如果 ETH1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示 eth1_ker_ck 时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止 eth1_ker_ck 时钟，读取“1”表示 eth1_ker_ck 时钟在 CSleep 模式下已使能

位 6:2 保留，必须保持复位值。

位 1 **MCELPEN**：CSleep 模式期间 MCE 外设时钟使能 (MCE peripheral clocks enable during CSleep mode)

如果 MCE 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

位 0 保留，必须保持复位值。

10.8.177 RCC 安全 AHB6 睡眠时钟使能设置寄存器 (RCC_MP_S_AHB6LPENSETR)

RCC secure AHB6 Sleep clock enable set register

偏移地址：0x888

复位值：0x0000 0001

这是一个安全寄存器，用于在 MDMA 以安全模式使用时使能其在睡眠模式下的时钟。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MDMALPEN
															rs

位 31:1 保留，必须保持复位值。

位 0 **MDMALPEN**：CSleep 模式期间 MDMA 外设时钟使能 (MDMA peripheral clocks enable during CSleep mode)

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.178 RCC 安全 AHB6 睡眠时钟使能清零寄存器 (RCC_MP_S_AHB6LPENCLRR)

RCC secure AHB6 Sleep clock enable clear register

偏移地址：0x88C

复位值：0x0000 0001

这是一个安全寄存器，用于在 MDMA 以安全模式使用时禁止其在睡眠模式下的时钟。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MDMALPEN
															rc_w1

位 31:1 保留，必须保持复位值。

位 0 **MDMALPEN**：CSleep 模式期间 MDMA 外设时钟使能 (MDMA peripheral clocks enable during CSleep mode)

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.179 RCC 非安全 AHB6 睡眠时钟使能设置寄存器 (RCC_MP_NS_AHB6LPENSETR)

RCC non-secure AHB6 Sleep clock enable set register

偏移地址：0x890

复位值：0x0000 0001

这是一个非安全寄存器，用于在 MDMA 以非安全模式使用时使能其在睡眠模式下的时钟。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MDMALPEN
															rs

位 31:1 保留，必须保持复位值。

位 0 **MDMALPEN**：CSleep 模式期间 MDMA 外设时钟使能 (MDMA peripheral clocks enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.180 RCC 非安全 AHB6 睡眠时钟使能清零寄存器 (RCC_MP_NS_AHB6LPENCLRR)

RCC non-secure AHB6 Sleep clock enable clear register

偏移地址：0x894

复位值：0x0000 0001

这是一个非安全寄存器，用于在 MDMA 以非安全模式使用时禁止其在睡眠模式下的时钟。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”会将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MDMALPEN
															rc_w1

位 31:1 保留，必须保持复位值。

位 0 **MDMALPEN**：CSleep 模式期间 MDMA 外设时钟使能 (MDMA peripheral clocks enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示外设时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止外设时钟，读取“1”表示外设时钟在 CSleep 模式下已使能

10.8.181 RCC 安全 AXI 睡眠时钟使能设置寄存器 (RCC_MP_S_AXIMLPENSETR)

RCC secure AXI Sleep clock enable set register

偏移地址：0x898

复位值：0x0000 0001

该安全寄存器用于在 CSleep 模式期间使能 SYSRAM 的时钟。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSRAMLPEN
															rs

位 31:1 保留，必须保持复位值。

位 0 **SYSRAMLPEN**：CSleep 模式期间 SYSRAM 接口时钟使能 (SYSRAM interface clock enable during CSleep mode)

由（安全）软件置 1。

0：写入“0”无影响，读取“0”表示内存接口时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能内存接口时钟，读取“1”表示内存接口时钟在 CSleep 模式下已使能

10.8.182 RCC 安全 AXI 睡眠时钟使能清零寄存器 (RCC_MP_S_AXIMLPENCLRR)

RCC secure AXI Sleep clock enable clear register

偏移地址：0x89C

复位值：0x0000 0001

该安全寄存器用于在 CSleep 模式期间禁止 SYSRAM 的时钟。该寄存器字段的写访问必须是安全写访问。读访问可以是安全或非安全访问。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSRAMLPEN
															rc_w1

位 31:1 保留，必须保持复位值。

位 0 **SYSRAMLPEN**：CSleep 模式期间 SYSRAM 接口时钟使能 (SYSRAM interface clock enable during CSleep mode)

由（安全）软件清零。

0：写入“0”无影响，读取“0”表示内存接口时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止内存接口时钟，读取“1”表示内存接口时钟在 CSleep 模式下已使能

10.8.183 RCC 非安全 AXI 睡眠时钟使能设置寄存器 (RCC_MP_NS_AXIMLPENSETR)

RCC non-secure AXI Sleep clock enable set register

偏移地址：0x8A0

复位值：0x0000 0001

该非安全寄存器用于在 CSleep 模式期间使能 SYSRAM 的时钟。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSRAMLPEN
															rs

位 31:1 保留，必须保持复位值。

位 0 **SYSRAMLPEN**：CSleep 模式期间 SYSRAM 接口时钟使能 (SYSRAM interface clock enable during CSleep mode)

由软件置 1。

0：写入“0”无影响，读取“0”表示内存接口时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能内存接口时钟，读取“1”表示内存接口时钟在 CSleep 模式下已使能

10.8.184 RCC 非安全 AXI 睡眠时钟使能清零寄存器 (RCC_MP_NS_AXIMLPENCLRR)

RCC non-secure AXI Sleep clock enable clear register

偏移地址：0x8A4

复位值：0x0000 0001

该非安全寄存器用于在 CSleep 模式期间禁止 SYSRAM 的时钟。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSRAMLPEN
															rc_w1

位 31:1 保留，必须保持复位值。

位 0 **SYSRAMLPEN**：CSleep 模式期间 SYSRAM 接口时钟使能 (SYSRAM interface clock enable during CSleep mode)

由软件清零。

0：写入“0”无影响，读取“0”表示内存接口时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止内存接口时钟，读取“1”表示内存接口时钟在 CSleep 模式下已使能

10.8.185 RCC MLAHB 睡眠时钟使能设置寄存器 (RCC_MP_MLAHBLPENSETR)

RCC MLAHB Sleep clock enable set register

偏移地址：0x8A8

复位值：0x0000 0007

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的 PERxLPEN 位置 1。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将对应位置“1”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAM3LPEN	SRAM2LPEN	SRAM1LPEN
													rs	rs	rs

位 31:3 保留，必须保持复位值。

位 2 **SRAM3LPEN**：CSleep 模式期间 SRAM3 接口时钟使能 (SRAM3 interface clock enable during CSleep mode)

如果 SRAM3 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示内存接口时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能内存接口时钟，读取“1”表示内存接口时钟在 CSleep 模式下已使能

位 1 **SRAM2LPEN**：CSleep 模式期间 SRAM2 接口时钟使能 (SRAM2 interface clock enable during CSleep mode)

如果 SRAM2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示内存接口时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能内存接口时钟，读取“1”表示内存接口时钟在 CSleep 模式下已使能

位 0 **SRAM1LPEN**：CSleep 模式期间 SRAM1 接口时钟使能 (SRAM1 interface clock enable during CSleep mode)

如果 SRAM1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件置 1。

0：写入“0”无影响，读取“0”表示内存接口时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下使能内存接口时钟，读取“1”表示内存接口时钟在 CSleep 模式下已使能

10.8.186 RCC MLAHB 睡眠时钟使能清零寄存器 (RCC_MP_MLAHBLPENCLRR)

RCC MLAHB Sleep clock enable clear register

偏移地址：0x8AC

复位值：0x0000 0007

该寄存器可在位字段级别单独、安全地实现写保护，如以下寄存器字段描述中的详细说明。

该寄存器用于将相应外设的 PERxLPEN 位清零。写入“0”无影响，读取则会返回相应位的有效值。写入“1”将相应位设为“0”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAM3LPEN	SRAM2LPEN	SRAM1LPEN
													rc_w1	rc_w1	rc_w1

位 31:3 保留，必须保持复位值。

位 2 **SRAM3LPEN**：CSleep 模式期间 SRAM3 接口时钟使能 (SRAM3 interface clock enable during CSleep mode)

如果 SRAM3 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示内存接口时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止内存接口时钟，读取“1”表示内存接口时钟在 CSleep 模式下已使能

位 1 **SRAM2LPEN**：CSleep 模式期间 SRAM2 接口时钟使能 (SRAM2 interface clock enable during CSleep mode)

如果 SRAM2 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示内存接口时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止内存接口时钟，读取“1”表示内存接口时钟在 CSleep 模式下已使能

位 0 **SRAM1LPEN**：CSleep 模式期间 SRAM1 接口时钟使能 (SRAM1 interface clock enable during CSleep mode)

如果 SRAM1 通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置为安全，则对该寄存器字段进行的写访问必须是安全写访问。读访问可以是安全或非安全访问。

由软件清零。

0：写入“0”无影响，读取“0”表示内存接口时钟在 CSleep 模式下已禁止

1：写入“1”在 CSleep 模式下禁止内存接口时钟，读取“1”表示内存接口时钟在 CSleep 模式下已使能

10.8.187 RCC APB3 外设安全状态寄存器 (RCC_APB3SECSR)

RCC APB3 peripheral secure state register

偏移地址：0x8C0

复位值：0x0000 4003

该读取寄存器在位级别反映了各个 APB3 可保护外设的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	VREFSECF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM3SECF	LPTIM2SECF
		r												r	r

位 31:14 保留，必须保持复位值。

位 13 **VREFSECF**：VREF 模块安全状态标志 (VREF block secure status flag)
该位反映了 VREF 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。
0：外设为非安全
1：外设为安全

位 12:2 保留，必须保持复位值。

位 1 **LPTIM3SECF**：LPTIM3 模块安全状态标志 (LPTIM3 block secure status flag)
该位反映了 LPTIM3 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。
0：外设为非安全
1：外设为安全

位 0 **LPTIM2SECF**：LPTIM2 模块安全状态标志 (LPTIM2 block secure status flag)
该位反映了 LPTIM2 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。
0：外设为非安全
1：外设为安全

10.8.188 RCC APB4 外设安全状态寄存器 (RCC_APB4SECSR)

RCC APB4 peripheral secure state register

偏移地址：0x8C4

复位值：0x0001 0002

该读取寄存器在位级别反映了各个 APB4 可保护外设的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBPHYSECF
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCMIPPSECF	Res.
														r	

位 31:17 保留，必须保持复位值。

位 16 **USBPHYSECF**：USBPHY 模块安全状态标志 (USBPHY block secure status flag)

该位反映了 USBPHY 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段配置）。

0：外设为非安全

1：外设为安全

位 15:2 保留，必须保持复位值。

位 1 **DCMIPPSECF**：DCMIPP 模块安全状态标志 (DCMIPP block secure status flag)

该位反映了 DCMIPP 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 0 保留，必须保持复位值。

10.8.189 RCC APB5 外设安全状态寄存器 (RCC_APB5SECSR)

RCC APB5 peripheral secure state register

偏移地址：0x8C8

复位值：0x0031 A800

该读取寄存器在位级别反映了各个 APB5 安全或可保护外设的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段或通过 TrustZone 感知外设本身进行的配置）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENCSECF[1:0]		Res.	Res.	Res.	BSECSECF
										r	r				r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDG1SECF	Res.	ETZPCSECF	Res.	TZCSECF	Res.	Res.	RTCSECF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r		r		r			r								

位 31:22 保留，必须保持复位值。

位 21:20 **STGENCSECF[1:0]**：STGENC 模块安全状态标志 (STGENC block secure status flag)

该位字段反映了 STGENC 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

00：外设为非安全

11：外设为安全

位 19:17 保留，必须保持复位值。

位 16 **BSECSECF**：BSEC 模块安全状态标志 (BSEC block secure status flag)

该位反映了 BSEC 模块的安全状态。

0：外设为非安全

1：外设为安全

位 15 **IWDG1SECF**：IWDG1 模块安全状态标志 (IWDG1 block secure status flag)

该位反映了 IWDG1 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 14 保留，必须保持复位值。

位 13 **ETZPCSECF**：ETZPC 模块安全状态标志 (ETZPC block secure status flag)

该位反映了 ETZPC 模块的安全状态。

0：外设为非安全

1：外设为安全

位 12 保留，必须保持复位值。

位 11 **TZCSECF**：TZC 模块安全状态标志 (TZC block secure status flag)

该位反映了 TZC 模块的安全状态。

0：外设为非安全

1：外设为安全

位 10:9 保留，必须保持复位值。

位 8 **RTCSECF**：RTC 模块安全状态标志 (RTC block secure status flag)

该位反映了 RTC 模块的安全状态。如果 RTC 的至少一个功能通过 RTC_SECCFGR 寄存器配置为安全（即如果由 RTC 驱动的输入信号 rtc_sec 置为有效），则该位置 1。

0：外设为非安全（默认）

1：外设为安全（至少一个 RTC 子功能）

位 7:0 保留，必须保持复位值。

10.8.190 RCC APB6 外设安全状态寄存器 (RCC_APB6SECSR)

RCC APB6 peripheral secure state register

偏移地址：0x8CC

复位值：0x0000 1FFF

该读取寄存器在位级别反映了各个 APB6 可保护外设的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TIM17SECF	TIM16SECF	TIM15SECF	TIM14SECF	TIM13SECF	TIM12SECF	I2C5SECF	I2C4SECF	I2C3SECF	SPI5SECF	SPI4SECF	USART2SECF	USART1SECF
			r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:13 保留，必须保持复位值。

位 12 **TIM17SECF**：TIM17 模块安全状态标志 (TIM17 block secure status flag)

该位反映了 TIM17 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 11 **TIM16SECF**：TIM16 模块安全状态标志 (TIM16 block secure status flag)

该位反映了 TIM16 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 10 **TIM15SECF**：TIM15 模块安全状态标志 (TIM15 block secure status flag)

该位反映了 TIM15 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 9 **TIM14SECF**：TIM14 模块安全状态标志 (TIM14 block secure status flag)

该位反映了 TIM14 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 8 **TIM13SECF**：TIM13 模块安全状态标志 (TIM13 block secure status flag)

该位反映了 TIM13 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 7 **TIM12SECF**：TIM12 模块安全状态标志 (TIM12 block secure status flag)

该位反映了 TIM12 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 6 **I2C5SECF**：I2C5 模块安全状态标志 (I2C5 block secure status flag)

该位反映了 I2C5 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 5 **I2C4SECF**：I2C4 模块安全状态标志 (I2C4 block secure status flag)

该位反映了 I2C4 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 4 **I2C3SECF**：I2C3 模块安全状态标志 (I2C3 block secure status flag)

该位反映了 I2C3 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 3 **SPI5SECF**：SPI5 模块安全状态标志 (SPI5 block secure status flag)

该位反映了 SPI5 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 2 **SPI4SECF**：SPI4 模块安全状态标志 (SPI4 block secure status flag)

该位反映了 SPI4 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 1 **USART2SECF**：USART2 模块安全状态标志 (USART2 block secure status flag)

该位反映了 USART2 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 0 **USART1SECF**：USART1 模块安全状态标志 (USART1 block secure status flag)

该位反映了 USART1 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

10.8.191 RCC AHB2 外设安全状态寄存器 (RCC_AHB2SECSR)

RCC AHB2 peripheral secure state register

偏移地址：0x8D0

复位值：0x0000 0178

该读取寄存器在位级别反映了各个 AHB2 可保护外设的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBOSECF	Res.	ADC2SECF	ADC1SECF	DMAMUX2SECF	DMA3SECF	Res.	Res.	Res.
							r		r	r	r	r			

位 31:9 保留，必须保持复位值。

位 8 **USBOSECF**：USBO 模块安全状态标志 (USBO block secure status flag)

该位反映了 USBO 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 7 保留，必须保持复位值。

位 6 **ADC2SECF**：ADC2 模块安全状态标志 (ADC2 block secure status flag)

该位反映了 ADC2 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 5 **ADC1SECF**：ADC1 模块安全状态标志 (ADC1 block secure status flag)
该位反映了 ADC1 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。
0：外设为非安全
1：外设为安全

位 4 **DMAMUX2SECF**：DMAMUX2 模块安全状态标志 (DMAMUX2 block secure status flag)
该位反映了 DMAMUX2 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。
0：外设为非安全
1：外设为安全

位 3 **DMA3SECF**：DMA3 模块安全状态标志 (DMA3 block secure status flag)
该位反映了 DMA3 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。
0：外设为非安全
1：外设为安全

位 2:0 保留，必须保持复位值。

10.8.192 RCC AHB5 外设安全状态寄存器 (RCC_AHB5SECSR)

RCC AHB5 peripheral secure state register

偏移地址：0x8D8

复位值：0x0000 017C

该读取寄存器在位级别反映了各个 AHB5 可保护外设的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKPSRAMSECF	Res.	RNG1SECF	HASH1SECF	CRYP1SECF	SAESSECF	PKASECF	Res.	Res.
							r		r	r	r	r	r		

位 31:9 保留，必须保持复位值。

位 8 **BKPSRAMSECF**：BKPSRAM 模块安全状态标志 (BKPSRAM block secure status flag)
该位反映了 BKPSRAM 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。
0：外设为非安全
1：外设为安全

位 7 保留，必须保持复位值。

位 6 **RNG1SECF**：RNG1 模块安全状态标志 (RNG1 block secure status flag)
该位反映了 RNG1 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。
0：外设为非安全
1：外设为安全

位 5 **HASH1SECF**：HASH1 模块安全状态标志 (HASH1 block secure status flag)
该位反映了 HASH1 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。
0：外设为非安全
1：外设为安全

位 4 **CRYP1SECF**：CRYP1 模块安全状态标志 (CRYP1 block secure status flag)
该位反映了 CRYP1 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。
0：外设为非安全
1：外设为安全

位 3 **SAESSECF**：SAES 模块安全状态标志 (SAES block secure status flag)
该位反映了 SAES 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。
0：外设为非安全
1：外设为安全

位 2 **PKASECF**：PKA 模块安全状态标志 (PKA block secure status flag)
该位反映了 PKA 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。
0：外设为非安全
1：外设为安全

位 1:0 保留，必须保持复位值。

10.8.193 RCC AHB6 外设安全状态寄存器 (RCC_AHB6SECSR)

RCC AHB6 peripheral secure state register

偏移地址：0x8DC

复位值：0xF803 5F82

该读取寄存器在位级别反映了各个 AHB6 可保护外设的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		ETH2SECF[4:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC2SECF	SDMMC1SECF
r	r	r	r	r										r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	QSPISECF	Res.	FMCSECF			ETH1SECF[4:0]			Res.	Res.	Res.	Res.	Res.	MCESECF	Res.
	r		r	r	r	r	r	r						r	

位 31:27 **ETH2SECF[4:0]**：ETH2 模块安全状态标志 (ETH2 block secure status flag)

该位字段反映了 ETH2 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

00000：外设为非安全

11111：外设为安全

位 26:18 保留，必须保持复位值。

位 17 **SDMMC2SECF**: SDMMC2 模块安全状态标志 (SDMMC2 block secure status flag)

该位反映了 SDMMC2 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 16 **SDMMC1SECF**：SDMMC1 模块安全状态标志 (SDMMC1 block secure status flag)

该位反映了 SDMMC1 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 15 保留，必须保持复位值。

位 14 **QSPISECF**：QSPI 模块安全状态标志 (QSPI block secure status flag)

该位反映了 QSPI 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 13 保留，必须保持复位值。

位 12 **FMCSECF**：FMC 模块安全状态标志 (FMC block secure status flag)

该位反映了 FMC 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 11:7 **ETH1SECF[4:0]**：ETH1 模块安全状态标志 (ETH1 block secure status flag)

该位字段反映了 ETH1 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

00000：外设为非安全

11111：外设为安全

位 6:2 保留，必须保持复位值。

位 1 **MCESECF**：MCE 模块安全状态标志 (MCE block secure status flag)

该位反映了 MCE 模块的安全状态（通过 ETZPC 寄存器的相应 DECPROT[1] 位字段进行的配置）。

0：外设为非安全

1：外设为安全

位 0 保留，必须保持复位值。

10.8.194 RCC 版本寄存器 (RCC_VERR)

RCC Version register

偏移地址：0xFF4

复位值：0x0000 0020

该寄存器指定 IP 版本

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：IP 的主版本 (Major Revision of the IP)

位 3:0 **MINREV[3:0]**：IP 的次版本 (Minor Revision of the IP)

10.8.195 RCC ID 寄存器 (RCC_IDR)

RCC ID register

偏移地址：0xFF8

复位值：0x0000 0001

该寄存器提供 RCC 的唯一标识符

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：RCC 的标识符

10.8.196 RCC 大小 ID 寄存器 (RCC_SIDR)

RCC Size ID register

偏移地址：0xFFC

复位值：0xA3C5 DD04

该寄存器提供解码空间，对于 RCC 其大小为 4 kB。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：解码空间为 4 kB

10.9 RCC 寄存器映射

表 74. RCC 寄存器映射和复位值

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**RCC_SECCFGR**	PWRSEC	Res.	Res.	Res.	Res.	Res.	RSTSEC	STPSEC	MCO2SEC	MCO1SEC	CPERSEC	TIMG3SEC	APB6DIVSEC	APB5DIVSEC	APB4DIVSEC	APB3DIVSEC	Res.	Res.	MLAHBSEC	AXISEC	MPUSEC	PLL4SEC	PLL3SEC	PLL12SEC	Res.	Res.	Res.	LSESEC	LSISEC	HSESEC	CSISEC	HSISEC
	Reset value	1						1	1	1	1	1	1	1	1	1	1			1	1	1	1	1	1				1	1	1	1	1
0x004 - 0x0FC	Reserved	Res.																															
0x100	**RCC_MP_ SREQSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STPREQ_P0
	Reset value																																0
0x104	**RCC_MP_ SREQCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STPREQ_P0
	Reset value																																0
0x108	**RCC_MP_ APRSTCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RSTTO[6:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDCTLEN
	Reset value																		1	1	1	1	1	1	1								0
0x10C	**RCC_MP_ APRSTSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RSTTOV[6:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																		0	0	0	0	0	0	0								
0x110	**RCC_ PWRLPDLYCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCTMPSKP	Res.	Res.	PWRLP_DLY[21:0]																					
	Reset value								0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0x114	**RCC_MP_ GRSTCSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPSYSRST
	Reset value																																0
0x118	**RCC_BR_ RSTSCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IWDG2RSTF	IWDG1RSTF	Res.	MPSYSRSTF	VCPURSTF	VCORERSTF	HCSSRSTF	PADRSTF	BORRSTF	PORRSTF
	Reset value																							0	0		0	0	1	0	1	0	1

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x11C	**RCC_MP_ RSTSSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPARE	Res.	Res.	CSTDBYRSTF	STDBYRSTF	STP2RSTF	IWDG2RSTF	IWDG1RSTF	Res.	MPSYSRSTF	VCPURSTF	VCORERSTF	HCSSRSTF	PADRSTF	BORRSTF	PORRSTF
	Reset value																	0			0	0	0	0	0		0	0	0	0	0	0	0
0x120	**RCC_MP_ RSTSCLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPARE	Res.	Res.	CSTDBYRSTF	STDBYRSTF	STP2RSTF	IWDG2RSTF	IWDG1RSTF	Res.	MPSYSRSTF	VCPURSTF	VCORERSTF	HCSSRSTF	PADRSTF	BORRSTF	PORRSTF
	Reset value																	0			0	0	0	0	0		0	0	0	0	0	0	0
0x124	**RCC_MP_ IWDGFZSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FZ_IWDG2	FZ_IWDG1
	Reset value																															0	0
0x128	**RCC_MP_ IWDGFZCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FZ_IWDG2	FZ_IWDG1
	Reset value																															0	0
0x12C - 0x1FC	Reserved	Res.																															
0x200	**RCC_MP_CIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WKUPIE	Res.	Res.	Res.	LSECSSIE	Res.	Res.	Res.	Res.	PLL4DYIE	PLL3DYIE	PLL2DYIE	PLL1DYIE	Res.	Res.	Res.	CSIRDYIE	HSERDYIE	HSIRDYIE	LSERDYIE	LSIRDYIE
	Reset value												0				0					0	0	0	0				0	0	0	0	0
0x204	**RCC_MP_CIFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WKUPF	Res.	Res.	Res.	LSECSSF	Res.	Res.	Res.	Res.	PLL4DYF	PLL3DYF	PLL2DYF	PLL1DYF	Res.	Res.	Res.	CSIRDYF	HSERDYF	HSIRDYF	LSERDYF	LSIRDYF
	Reset value												0				0					0	0	0	0				0	0	0	0	0
0x208 - 0x3FC	Reserved	Res.																															
0x400	**RCC_BDCR**	VSWRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTCCKEN	Res.	Res.	RTCSRC[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	LSECSSD	LSECSSON	Res.	Res.	LSEDRV[1:0]		DIGBYP	LSERDY	LSEBYP	LSEON
	Reset value	0											0			0	0							0	0			1	0	0	0	0	0

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x404	**RCC_RDLSICR**	SPARE[4:0]					EADLY[2:0]			Res.	Res.	Res.	MRD[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSIRDY	LSION
	Reset value	0	0	0	0	0	0	0	0				0	0	0	0	0														0	0
0x408 - 0x41C	Reserved	Res.																															
0x420	**RCC_OCENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSECSSON	HSEBYP	HSEKERON	HSEON	DIGBYP	Res.	CSIKERON	CSION	Res.	Res.	HSIKERON	HSION
	Reset value																					0	0	0	0	0		0	0			0	1
0x424	**RCC_OCENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSEBYP	HSEKERON	HSEON	DIGBYP	Res.	CSIKERON	CSION	Res.	Res.	HSIKERON	HSION
	Reset value																						0	0	0	0		0	0			0	1
0x428	**RCC_OCRDYR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXICKRDY	MPUCKRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSERDY	Res.	Res.	Res.	CSIRDY	Res.	HSIDIVRDY	Res.	HSIRDY
	Reset value								0	0															0				0		0		0
0x42C- 0x43C	Reserved	Res.																															
0x440	**RCC_HSICFGR**	Res.	Res.	Res.	Res.	HSICAL[11:0]												Res.	HSITRIM[6:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSIDIV[1:0]	
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0							0	0
0x444	**RCC_CSICFGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSICAL[7:0]								Res.	Res.	Res.	CSITRIM[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value									0	0	0	0	0	0	0	0				1	0	0	0	0								
0x448 - 0x45C	Reserved	Res.																															
0x460	**RCC_MCO1CFGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCO1ON	Res.	Res.	Res.	Res.	MCO1DIV[3:0]				Res.	MCO1SEL[2:0]		
	Reset value																				0					0	0	0	0		0	0	0

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x464	**RCC_MCO2CFGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCO2ON	Res.	Res.	Res.	Res.	MCO2DIV[3:0]				Res.	MCO2SEL[2:0]		
	Reset value																				0					0	0	0	0		0	0	0
0x468	**RCC_DBGCFGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGRST	Res.	Res.	TRACECKEN	DBGCKEN	Res.	Res.	Res.	Res.	Res.	TRACEDIV[2:0]		
	Reset value																				0			0	0						0	0	1
0x46C - 0x47C	Reserved	Res.																															
0x480	**RCC_RCK12SELR**	PLL2SRCRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL1SRC[1:0]	
	Reset value	1																													0	0
0x484	**RCC_RCK3SELR**	PLL3SRCRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL3SRC[1:0]	
	Reset value	1																													0	0
0x488	**RCC_RCK4SELR**	PLL4SRCRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL4SRC[1:0]	
	Reset value	1																													0	0
0x48C - 0x49C	Reserved	Res.																															
0x4A0	**RCC_PLL1CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVREN	DIVQEN	DIVPEN	Res.	SSCG_CTRL	PLL1RDY	PLLON
	Reset value																										0	0	0		0	0	0
0x4A4	**RCC_PLL1CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVM1[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVN[8:0]								
	Reset value											0	0	0	0	0	1								0	0	0	1	1	0	0	0	1

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x4A8	**RCC_PLL1CFGR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVR[6:0]							Res.	DIVQ[6:0]							Res.	DIVP[6:0]						
	Reset value										0	0	0	0	0	0	1		0	0	0	0	0	0	1		0	0	0	0	0	0	0
0x4AC	**RCC_PLL1FRACR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRACLE	FRACV[12:0]													Res.	Res.	Res.
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x4B0	**RCC_PLL1CSGR**	Res.	INC_STEP[14:0]															SSCG_MODE	RPDFN_DIS	TPDFN_DIS	MOD_PER[12:0]												
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4B4 - 0x4CC	Reserved	Res.																															
0x4D0	**RCC_PLL2CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVREN	DIVQEN	DIVPEN	Res.	SSCG_CTRL	PLL2RDY	PLLON
	Reset value																										0	0	0		0	0	0
0x4D4	**RCC_PLL2CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVM2[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVN[8:0]								
	Reset value											0	0	0	0	0	1								0	0	1	1	0	0	0	1	1
0x4D8	**RCC_PLL2CFGR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVR[6:0]							Res.	DIVQ[6:0]							Res.	DIVP[6:0]						
	Reset value										0	0	0	0	0	0	1		0	0	0	0	0	0	1		0	0	0	0	0	0	1
0x4DC	**RCC_PLL2FRACR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRACLE	FRACV[12:0]													Res.	Res.	Res.
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x4E0	**RCC_PLL2CSGR**	Res.	INC_STEP[14:0]															SSCG_MODE	RPDFN_DIS	TPDFN_DIS	MOD_PER[12:0]												
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4E4 - 0x4FC	Reserved	Res.																															
0x500	**RCC_PLL3CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVREN	DIVQEN	DIVPEN	Res.	SSCG_CTRL	PLL3RDY	PLLON
	Reset value																										0	0	0		0	0	0

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x504	**RCC_PLL3CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	IFRGE[1:0]		Res.	Res.	DIVM3[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVN[8:0]								
	Reset value							0	0			0	0	0	0	0	1								0	0	0	1	1	0	0	0	1
0x508	**RCC_PLL3CFGR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVR[6:0]							Res.	DIVQ[6:0]							Res.	DIVP[6:0]						
	Reset value										0	0	0	0	0	0	1		0	0	0	0	0	0	1		0	0	0	0	0	0	1
0x50C	**RCC_PLL3FRACR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRACLE	FRACV[12:0]													Res.	Res.	Res.
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x510	**RCC_PLL3CSGR**	Res.	INC_STEP[14:0]															SSCG_MODE	RPDFN_DIS	TPDFN_DIS	MOD_PER[12:0]												
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x514-0x51C	Reserved	Res.																															
0x520	**RCC_PLL4CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVREN	DIVQEN	DIVPEN	Res.	SSCG_CTRL	PLL4RDY	PLLON
	Reset value																										0	0	0		0	0	0
0x524	**RCC_PLL4CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	IFRGE[1:0]		Res.	Res.	DIVM4[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVN[8:0]								
	Reset value							0	0			0	0	0	0	0	1								0	0	0	1	1	0	0	0	1
0x528	**RCC_PLL4CFGR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVR[6:0]							Res.	DIVQ[6:0]							Res.	DIVP[6:0]						
	Reset value										0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x52C	**RCC_PLL4FRACR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRACLE	FRACV[12:0]													Res.	Res.	Res.
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x530	**RCC_PLL4CSGR**	Res.	INC_STEP[14:0]															SSCG_MODE	RPDFN_DIS	TPDFN_DIS	MOD_PER[12:0]												
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x534 - 0x53F	Reserved	Res.																															

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																														
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x540	**RCC_MPCKSELR**	MPUSRCRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPUSRC[1:0]	
	Reset value	1																														0	0
0x544	**RCC_ ASSCKSELR**	AXISSRCRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXISSRC[2:0]		
	Reset value	1																													0	0	0
0x548	**RCC_ MSSCKSELR**	MLAHBSSRCRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MLAHBSSRC[2:0]		
	Reset value	1																														0	0
0x54C	**RCC_ CPERCKSELR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKPERSRC[1:0]	
	Reset value																															0	0
0x550 - 0x55C	Reserved	Res.																															
0x560	**RCC_RTCDIVR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTCDIV[5:0]					
	Reset value																											0	0	0	0	0	0
0x564	**RCC_MPCKDIVR**	MPUDIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPUDIV[3:0]			
	Reset value	1																												0	0	0	1
0x568	**RCC_AXIDIVR**	AXIDIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXIDIV[2:0]		
	Reset value	1																													0	0	0

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																														
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x56C	**RCC_ MLAHBDIVR**	MLAHBDIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MLAHBDIV[3:0]			
	Reset value	1																												0	0	0	0
0x570	**RCC_APB1DIVR**	APB1DIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APB1DIV[2:0]		
	Reset value	1																													0	0	0
0x574	**RCC_APB2DIVR**	APB2DIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APB2DIV[2:0]		
	Reset value	1																													0	0	0
0x578	**RCC_APB3DIVR**	APB3DIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APB3DIV[2:0]		
	Reset value	1																													0	0	0
0x57C	**RCC_APB4DIVR**	APB4DIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APB4DIV[2:0]		
	Reset value	1																													0	0	0
0x580	**RCC_APB5DIVR**	APB5DIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APB5DIV[2:0]		
	Reset value	1																													0	0	0
0x584	**RCC_APB6DIVR**	APB6DIVRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APB6DIV[2:0]		
	Reset value	1																													0	0	0
0x588 - 0x59C	Reserved	Res.																															

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x5A0	**RCC_TIMG1PRER**	TIMG1PRERDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMG1PRE
	Reset value	1																															0
0x5A4	**RCC_TIMG2PRER**	TIMG2PRERDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMG2PRE
	Reset value	1																															0
0x5A8	**RCC_TIMG3PRER**	TIMG3PRERDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMG3PRE
	Reset value	1																															0
0x5AC - 0x5BC	Reserved	Res.																															
0x5C0	**RCC_DDRITFCR**	GSKP_DUR[3:0]				DFILP_WIDTH[2:0]			GSKPCTRL	GSKPMOD	DDRCKMOD[2:0]			DPHYCTLRST	DPHYRST	DPHYAPBRST	DDRCORERST	DDRCAXIRST	DDRCAPBRST	KERDCG_DLY[2:0]			DDRPHYCAPBLPEN	DDRPHYCAPBEN	AXIDCGEN	DDRCAPBLPEN	DDRCAPBEN	DDRPHYCLPEN	DDRPHYCEN	Res.	Res.	DDRC1LPEN	DDRC1EN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	0	0	0	0	0	0	1	0			1	0
0x5C4 - 0x5FC	Reserved	Res.																															
0x600	**RCC_ I2C12CKSELR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C12SRC[2:0]		
	Reset value																														0	0	0
0x604	**RCC_ I2C345CKSELR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C5SRC[2:0]			I2C4SRC[2:0]			I2C3SRC[2:0]		
	Reset value																								0	0	0	0	0	0	0	0	0

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																														
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x608	**RCC_ SPI2S1CKSELR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPI1SRC[2:0]		
	Reset value																														0	0	0
0x60C	**RCC_ SPI2S23CKSELR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPI23SRC[2:0]		
	Reset value																														0	0	0
0x610	**RCC_ SPI45CKSELR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPI5SRC[2:0]			SPI4SRC[2:0]		
	Reset value																											0	0	0	0	0	0
0x614	**RCC_ UART12CKSELR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UART2SRC[2:0]			UART1SRC[2:0]		
	Reset value																											0	0	0	0	0	0
0x618	**RCC_ UART35CKSELR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UART35SRC[2:0]		
	Reset value																														0	0	0
0x61C	**RCC_ UART4CKSELR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UART4SRC[2:0]		
	Reset value																														0	0	0
0x620	**RCC_ UART6CKSELR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UART6SRC[2:0]		
	Reset value																														0	0	0

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																														
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x624	RCC_ UART78CKSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UART78SRC[2:0]		
	Reset value																														0	0	0
0x628	RCC_ LPTIM1CKSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM1SRC[2:0]		
	Reset value																														0	0	0
0x62C	RCC_ LPTIM23CKSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM3SRC[2:0]			LPTIM2SRC[2:0]		
	Reset value																											0	0	0	0	0	0
0x630	RCC_ LPTIM45CKSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM45SRC[2:0]		
	Reset value																														0	0	0
0x634	RCC_ SAI1CKSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI1SRC[2:0]		
	Reset value																														0	0	0
0x638	RCC_ SAI2CKSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI2SRC[2:0]		
	Reset value																														0	0	0
0x63C	RCC_ FDCANCKSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANSRC[1:0]	
	Reset value																															0	0

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																														
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x640	RCC_ SPDIFCKSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPDIFSRC[1:0]	
	Reset value																															0	0
0x644	RCC_ ADC12CKSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC2SRC[1:0]		ADC1SRC[1:0]	
	Reset value																													0	0	0	0
0x648	RCC_ SDMMC12CKSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC2SRC[2:0]				SDMMC1SRC[2:0]	
	Reset value																											0	1	1	0	1	1
0x64C	RCC_ ETH12CKSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETH2PTPDIV[3:0]				Res.	Res.	ETH2SRC[1:0]		ETH1PTPDIV[3:0]				Res.	Res.	ETH1SRC[1:0]	
	Reset value																	0	0	0	0			0	0	0	0	0	0			0	0
0x650	RCC_ USBCKSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBOSRC	Res.	Res.	USBPHYSRC[1:0]	
	Reset value																												0			0	0
0x654	RCC_ QSPICKSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	QSPISRC[1:0]	
	Reset value																															0	0
0x658	RCC_ FMCCKSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FMCSRC[1:0]	
	Reset value																															0	0

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x65C	**RCC_ RNG1CKSELR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNG1SRC[1:0]	
	Reset value																															0	0
0x660	**RCC_ STGENCKSELR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENSRC[1:0]	
	Reset value																															0	0
0x664	**RCC_ DCMIPPCKSELR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCMIPPSRC[1:0]	
	Reset value																															0	0
0x668	**RCC_ SAESCKSELR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAESSRC[1:0]	
	Reset value																															0	0
0x66C-0x69C	Reserved	Res.																															
0x6A0	**RCC_ APB1RSTSETR**	Res.	Res.	Res.	Res.	Res.	SPDIFRST	Res.	Res.	Res.	I2C2RST	I2C1RST	Res.	UART8RST	UART7RST	UART5RST	UART4RST	USART3RST	Res.	Res.	SPI3RST	SPI2RST	Res.	LPTIM1RST	Res.	Res.	Res.	TIM7RST	TIM6RST	TIM5RST	TIM4RST	TIM3RST	TIM2RST
	Reset value						0				0	0		0	0	0	0	0			0	0		0				0	0	0	0	0	0
0x6A4	**RCC_ APB1RSTCLRR**	Res.	Res.	Res.	Res.	Res.	SPDIFRST	Res.	Res.	Res.	I2C2RST	I2C1RST	Res.	UART8RST	UART7RST	UART5RST	UART4RST	USART3RST	Res.	Res.	SPI3RST	SPI2RST	Res.	LPTIM1RST	Res.	Res.	Res.	TIM7RST	TIM6RST	TIM5RST	TIM4RST	TIM3RST	TIM2RST
	Reset value						0				0	0		0	0	0	0	0			0	0		0				0	0	0	0	0	0
0x6A8	**RCC_ APB2RSTSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANRST	Res.	Res.	Res.	DFSDMRST	Res.	Res.	SAI2RST	SAI1RST	Res.	Res.	USART6RST	Res.	Res.	Res.	Res.	SPI1RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM8RST	TIM1RST
	Reset value								0				0			0	0			0					0							0	0

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x6AC	**RCC_ APB2RSTCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANRST	Res.	Res.	Res.	DFSDMRST	Res.	Res.	SAI2RST	SAI1RST	Res.	Res.	USART6RST	Res.	Res.	Res.	Res.	SPI1RST	Res.	Res.	Res.	Res.	Res.	Res.	TIM8RST	TIM1RST
	Reset value								0				0			0	0			0					0							0	0
0x6B0	**RCC_ APB3RSTSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTSRST	Res.	Res.	VREFRST	Res.	SYSCFGRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM5RST	LPTIM4RST	LPTIM3RST	LPTIM2RST
	Reset value																0			0		0								0	0	0	0
0x6B4	**RCC_ APB3RSTCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTSRST	Res.	Res.	VREFRST	Res.	SYSCFGRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM5RST	LPTIM4RST	LPTIM3RST	LPTIM2RST
	Reset value																0			0		0								0	0	0	0
0x6B8	**RCC_ APB4RSTSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBPHYRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DDRPERFMRST	Res.	Res.	Res.	Res.	Res.	Res.	DCMIPPRST	LTDCRST
	Reset value																0								0							0	0
0x6BC	**RCC_ APB4RSTCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBPHYRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DDRPERFMRST	Res.	Res.	Res.	Res.	Res.	Res.	DCMIPPRST	LTDCRST
	Reset value																0								0							0	0
0x6C0	**RCC_ APB5RSTSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value												0																				
0x6C4	**RCC_ APB5RSTCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value												0																				
0x6C8	**RCC_ APB6RSTSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM17RST	TIM16RST	TIM15RST	TIM14RST	TIM13RST	TIM12RST	I2C5RST	I2C4RST	I2C3RST	SPI5RST	SPI4RST	USART2RST	USART1RST
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																												
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x6CC	**RCC_ APB6RSTCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM17RST	TIM16RST	TIM15RST	TIM14RST	TIM13RST	TIM12RST	I2C5RST	I2C4RST	I2C3RST	SPI5RST	SPI4RST	USART2RST	USART1RST
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x6D0	**RCC_ AHB2RSTSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBORST	Res.	ADC2RST	ADC1RST	DMAMUX2RST	DMA3RST	DMAMUX1RST	DMA2RST	DMA1RST
	Reset value																								0		0	0	0	0	0	0	0
0x6D4	**RCC_ AHB2RSTCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBORST	Res.	ADC2RST	ADC1RST	DMAMUX2RST	DMA3RST	DMAMUX1RST	DMA2RST	DMA1RST
	Reset value																								0		0	0	0	0	0	0	0
0x6D8-0x6DC	Reserved	Res.																															
0x6E0	**RCC_ AHB4RSTSETR**	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOIRST	GPIOHRST	GPIOGRST	GPIOFRST	GPIOERST	GPIODRST	GPIOCRST	GPIOBRST	GPIOARST
	Reset value																								0	0	0	0	0	0	0	0	0
0x6E4	**RCC_ AHB4RSTCLRR**	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOIRST	GPIOHRST	GPIOGRST	GPIOFRST	GPIOERST	GPIODRST	GPIOCRST	GPIOBRST	GPIOARST
	Reset value																								0	0	0	0	0	0	0	0	0
0x6E8	**RCC_ AHB5RSTSETR**	Res.	Res.	Res	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXIMCRST	Res.	Res.	Res.	Res.	Res.	Res	Res.	Res.	Res.	RNG1RST	HASH1RST	CRYP1RST	SAESRST	PKARST	Res.	Res.
	Reset value																0										0	0	0	0	0		
0x6EC	**RCC_ AHB5RSTCLRR**	Res.	Res.	Res	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXIMCRST	Res.	Res.	Res.	Res.	Res.	Res	Res.	Res.	Res.	RNG1RST	HASH1RST	CRYP1RST	SAESRST	PKARST	Res.	Res.
	Reset value																0										0	0	0	0	0		

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x6F0	**RCC_ AHB6RSTSETR**	Res.	ETH2MACRST	Res.	Res.	Res.	Res.	Res.	USBHRST	Res.	Res.	Res.	CRC1RST	Res.	Res.	SDMMC2RST	SDMMC1RST	Res.	QSPIRST	Res.	FMCRST	Res.	ETHMACRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCERST	MDMARST
	Reset value		0						0				0			0	0		0		0		0								0	0
0x6F4	**RCC_ AHB6RSTCLRR**	Res.	ETH2MACRST	Res.	Res.	Res.	Res.	Res.	USBHRST	Res.	Res.	Res.	CRC1RST	Res.	Res.	SDMMC2RST	SDMMC1RST	Res.	QSPIRST	Res.	FMCRST	Res.	ETHMACRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCERST	MDMARST
	Reset value		0						0				0			0	0		0		0		0								0	0
0x6F8-0x6FC	Reserved	Res.																															
0x700	**RCC_MP_ APB1ENSETR**	Res.	Res.	Res.	Res.	Res.	SPDIFEN	Res.	Res.	Res.	I2C2EN	I2C1EN	Res.	UART8EN	UART7EN	UART5EN	UART4EN	USART3EN	Res.	Res.	SPI3EN	SPI2EN	Res.	LPTIM1EN	Res.	Res.	Res.	TIM7EN	TIM6EN	TIM5EN	TIM4EN	TIM3EN	TIM2EN
	Reset value						0				0	0		0	0	0	0	0			0	0		0				0	0	0	0	0	0
0x704	**RCC_MP_ APB1ENCLRR**	Res.	Res.	Res.	Res.	Res.	SPDIFEN	Res.	Res.	Res.	I2C2EN	I2C1EN	Res.	UART8EN	UART7EN	UART5EN	UART4EN	USART3EN	Res.	Res.	SPI3EN	SPI2EN	Res.	LPTIM1EN	Res.	Res.	Res.	TIM7EN	TIM6EN	TIM5EN	TIM4EN	TIM3EN	TIM2EN
	Reset value						0				0	0		0	0	0	0	0			0	0		0				0	0	0	0	0	0
0x708	**RCC_MP_ APB2ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANEN	Res.	Res.	ADFSDMEN	DFSDMEN	Res.	Res.	SAI2EN	SAI1EN	Res.	Res.	USART6EN	Res.	Res.	Res.	Res.	SPI1EN	Res.	Res.	Res.	Res.	Res.	Res.	TIM8EN	TIM1EN
	Reset value								0			0	0			0	0			0					0							0	0
0x70C	**RCC_MP_ APB2ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANEN	Res.	Res.	ADFSDMEN	DFSDMEN	Res.	Res.	SAI2EN	SAI1EN	Res.	Res.	USART6EN	Res.	Res.	Res.	Res.	SPI1EN	Res.	Res.	Res.	Res.	Res.	Res.	TIM8EN	TIM1EN
	Reset value								0			0	0			0	0			0					0							0	0
0x710	**RCC_MP_ APB3ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDPEN	Res.	Res.	Res.	DTSEN	Res.	Res.	VREFEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM5EN	LPTIM4EN	LPTIM3EN	LPTIM2EN
	Reset value												0				0			0											0	0	0	0
0x714	**RCC_MP_ APB3ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDPEN	Res.	Res.	Res.	DTSEN	Res.	Res.	VREFEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM5EN	LPTIM4EN	LPTIM3EN	LPTIM2EN
	Reset value												0				0			0											0	0	0	0

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x718	**RCC_MP_S_ APB3ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGEN
	Reset value																																0
0x71C	**RCC_MP_S_ APB3ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGEN
	Reset value																																0
0x720	**RCC_MP_NS_ APB3ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGEN
	Reset value																																0
0x724	**RCC_MP_NS_ APB3ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGEN
	Reset value																																0
0x728	**RCC_MP_ APB4ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENROEN	Res.	Res.	Res.	USBPHYEN	IWDG2APBEN	Res.	Res.	Res.	Res.	Res.	Res.	DDRPERFMEN	Res.	Res.	Res.	Res.	Res.	Res.	DCMIPPEN	Res.
	Reset value												0				0	0							0							0	
0x72C	**RCC_MP_ APB4ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENROEN	Res.	Res.	Res.	USBPHYEN	IWDG2APBEN	Res.	Res.	Res.	Res.	Res.	Res.	DDRPERFMEN	Res.	Res.	Res.	Res.	Res.	Res.	DCMIPPEN	Res.
	Reset value												0				0	0							0							0	
0x730	**RCC_MP_S_ APB4ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCEN
	Reset value																																0
0x734	**RCC_MP_S_ APB4ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCEN
	Reset value																																0
0x738	**RCC_MP_NS_ APB4ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCEN
	Reset value																																0

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x73C	**RCC_MP_NS_ APB4ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCEN
	Reset value																																0
0x740	**RCC_MP_ APB5ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENCEN	Res.	Res.	Res.	BSECEN	IWDG1APBEN	Res.	ETZPCEN	Res.	TZCEN	Res.	Res.	RTCAPBEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value												0				0	0		0		0			0								
0x744	**RCC_MP_ APB5ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENCEN	Res.	Res.	Res.	BSECEN	IWDG1APBEN	Res.	ETZPCEN	Res.	TZCEN	Res.	Res.	RTCAPBEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value												0				0	0		0		0			0								
0x748	**RCC_MP_ APB6ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM17EN	TIM16EN	TIM15EN	TIM14EN	TIM13EN	TIM12EN	I2C5EN	I2C4EN	I2C3EN	SPI5EN	SPI4EN	USART2EN	USART1EN
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x74C	**RCC_MP_ APB6ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM17EN	TIM16EN	TIM15EN	TIM14EN	TIM13EN	TIM12EN	I2C5EN	I2C4EN	I2C3EN	SPI5EN	SPI4EN	USART2EN	USART1EN
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x750	**RCC_MP_ AHB2ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBOEN	Res.	ADC2EN	ADC1EN	DMAMUX2EN	DMA3EN	DMAMUX1EN	DMA2EN	DMA1EN
	Reset value																								0		0	0	0	0	0	0	0
0x754	**RCC_MP_ AHB2ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBOEN	Res.	ADC2EN	ADC1EN	DMAMUX2EN	DMA3EN	DMAMUX1EN	DMA2EN	DMA1EN
	Reset value																								0		0	0	0	0	0	0	0
0x758-0x764	Reserved	Res.																															
0x768	**RCC_MP_S_ AHB4ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
	Reset value																								0	0	0	0	0	0	0	0	0

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x76C	**RCC_MP_S_ AHB4ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
	Reset value																								0	0	0	0	0	0	0	0	0
0x770	**RCC_MP_NS_ AHB4ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
	Reset value																								0	0	0	0	0	0	0	0	0
0x774	**RCC_MP_NS_ AHB4ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
	Reset value																								0	0	0	0	0	0	0	0	0
0x778	**RCC_MP_ AHB5ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXIMCEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKPSRAMEN	Res.	RNG1EN	HASH1EN	CRYP1EN	SAESEN	PKAEN	Res.	Res.
	Reset value																0								0		0	0	0	0	0		
0x77C	**RCC_MP_ AHB5ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXIMCEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKPSRAMEN	Res.	RNG1EN	HASH1EN	CRYP1EN	SAESEN	PKAEN	Res.	Res.
	Reset value																0								0		0	0	0	0	0		
0x780	**RCC_MP_ AHB6ENSETR**	Reserved	ETH2MACEN	ETH2RXEN	ETH2TXEN	ETH2CKEN	Res.	Res.	USBHEN	Res.	Res.	Res.	CRC1EN	Res.	Res.	SDMMC2EN	SDMMC1EN	Res.	QSPIEN	Res.	FMCEN	Res.	ETH1MACEN	ETH1RXEN	ETH1TXEN	ETH1CKEN	Res.	Res.	Res.	Res.	Res.	Res.	MCEEN	Res.
	Reset value		0	0	0	0			0				0			0	0		0		0		0	0	0	0						0	
0x784	**RCC_MP_ AHB6ENCLRR**	Reserved	ETH2MACEN	ETH2RXEN	ETH2TXEN	ETH2CKEN	Res.	Res.	USBHEN	Res.	Res.	Res.	CRC1EN	Res.	Res.	SDMMC2EN	SDMMC1EN	Res.	QSPIEN	Res.	FMCEN	Res.	ETH1MACEN	ETH1RXEN	ETH1TXEN	ETH1CKEN	Res.	Res.	Res.	Res.	Res.	Res.	MCEEN	Res.
	Reset value		0	0	0	0			0				0			0	0		0		0		0	0	0	0						0	
0x788	**RCC_MP_S_ AHB6ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MDMAEN
	Reset value																																0
0x78C	**RCC_MP_S_ AHB6ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MDMAEN
	Reset value																																0

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x790	**RCC_MP_NS_AHB6ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MDMAEN
	Reset value																																0
0x794	**RCC_MP_NS_AHB6ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MDMAEN
	Reset value																																0
0x798 - 0x7FC	Reserved	Res.																															
0x800	**RCC_MP_APB1LPENSETR**	Res.	Res.	Res.	Res.	Res.	SPDIFLPEN	Res.	Res.	Res.	I2C2LPEN	I2C1LPEN	Res.	UART8LPEN	UART7LPEN	UART5LPEN	UART4LPEN	USART3LPEN	Res.	Res.	SPI3LPEN	SPI2LPEN	Res.	LPTIM1LPEN	Res.	Res.	Res.	TIM7LPEN	TIM6LPEN	TIM5LPEN	TIM4LPEN	TIM3LPEN	TIM2LPEN
	Reset value						1				1	1		1	1	1	1	1			1	1		1				1	1	1	1	1	1
0x804	**RCC_MP_APB1LPENCLRR**	Res.	Res.	Res.	Res.	Res.	SPDIFLPEN	Res.	Res.	Res.	I2C2LPEN	I2C1LPEN	Res.	UART8LPEN	UART7LPEN	UART5LPEN	UART4LPEN	USART3LPEN	Res.	Res.	SPI3LPEN	SPI2LPEN	Res.	LPTIM1LPEN	Res.	Res.	Res.	TIM7LPEN	TIM6LPEN	TIM5LPEN	TIM4LPEN	TIM3LPEN	TIM2LPEN
	Reset value						1				1	1		1	1	1	1	1			1	1		1				1	1	1	1	1	1
0x808	**RCC_MP_APB2LPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANLPEN	Res.	Res.	ADFSDMLPEN	DFSDMLPEN	Res.	Res.	SAI2LPEN	SAI1LPEN	Res.	Res.	USART6LPEN	Res.	Res.	Res.	Res.	SPI1LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM8LPEN	TIM1LPEN
	Reset value								1			1	1			1	1			1				1								1	1
0x80C	**RCC_MP_APB2LPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANLPEN	Res.	Res.	ADFSDMLPEN	DFSDMLPEN	Res.	Res.	SAI2LPEN	SAI1LPEN	Res.	Res.	USART6LPEN	Res.	Res.	Res.	Res.	SPI1LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM8LPEN	TIM1LPEN
	Reset value								1			1	1			1	1			1				1								1	1
0x810	**RCC_MP_APB3LPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTSLPEN	Res.	Res.	VREFLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM5LPEN	LPTIM4LPEN	LPTIM3LPEN	LPTIM2LPEN
	Reset value															1	1			1										1	1	1	1
0x814	**RCC_MP_APB3LPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTSLPEN	Res.	Res.	VREFLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM5LPEN	LPTIM4LPEN	LPTIM3LPEN	LPTIM2LPEN
	Reset value															1	1			1										1	1	1	1

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x818	**RCC_MP_S_ APB3LPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGLPEN
	Reset value																																1
0x81C	**RCC_MP_S_ APB3LPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGLPEN
	Reset value																																1
0x820	**RCC_MP_NS_ APB3LPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGLPEN
	Reset value																																1
0x824	**RCC_MP_NS_ APB3LPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGLPEN
	Reset value																																1
0x828	**RCC_MP_ APB4LPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENROSTPEN	STGENROLPEN	Res.	Res.	Res.	USBPHYLPEN	IWDG2APBLPEN	Res.	Res.	Res.	Res.	Res.	Res.	DDRPERFMLPEN	Res.	Res.	Res.	Res.	Res.	Res.	DCMIPPLPEN	Res.
	Reset value											0	1				1	1							1							1	
0x82C	**RCC_MP_ APB4LPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENROSTPEN	STGENROLPEN	Res.	Res.	Res.	USBPHYLPEN	IWDG2APBLPEN	Res.	Res.	Res.	Res.	Res.	Res.	DDRPERFMLPEN	Res.	Res.	Res.	Res.	Res.	Res.	DCMIPPLPEN	Res.
	Reset value											0	1				1	1							1							1	
0x830	**RCC_MP_S_ APB4LPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCLPEN
	Reset value																																1

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x834	**RCC_MP_S_ APB4LPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCLPEN
	Reset value																																1
0x838	**RCC_MP_NS_ APB4LPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCLPEN
	Reset value																																1
0x83C	**RCC_MP_NS_ APB4LPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCLPEN
	Reset value																																1
0x840	**RCC_MP_ APB5LPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENCSTPEN	STGENCLPEN	Res.	Res.	Res.	BSECLPEN	IWDG1APBLPEN	Res.	ETZPCLPEN	Res.	TZCLPEN	Res.	Res.	RTCAPBLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value											0	1				1	1		1		1			1								
0x844	**RCC_MP_ APB5LPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENCSTPEN	STGLPENCLPEN	Res.	Res.	Res.	BSECLPEN	IWDG1APBLPEN	Res.	ETZPCLPEN	Res.	TZCLPEN	Res.	Res.	RTCAPBLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value											0	1				1	1		1		1			1								
0x848	**RCC_MP_ APB6LPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM17LPEN	TIM16LPEN	TIM15LPEN	TIM14LPEN	TIM13LPEN	TIM12LPEN	I2C5LPEN	I2C4LPEN	I2C3LPEN	SPI5LPEN	SPI4LPEN	USART2LPEN	USART1LPEN
	Reset value																				1	1	1	1	1	1	1	1	1	1	1	1	1
0x84C	**RCC_MP_ APB6LPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM17LPEN	TIM16LPEN	TIM15LPEN	TIM14LPEN	TIM13LPEN	TIM12LPEN	I2C5LPEN	I2C4LPEN	I2C3LPEN	SPI5LPEN	SPI4LPEN	USART2LPEN	USART1LPEN
	Reset value																				1	1	1	1	1	1	1	1	1	1	1	1	1

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																												
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x850	**RCC_MP_ AHB2LPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBOLPEN	Res.	ADC2LPEN	ADC1LPEN	DMAMUX2LPEN	DMA3LPEN	DMAMUX1LPEN	DMA2LPEN	DMA1LPEN
	Reset value																								1		1	1	1	1	1	1	1
0x854	**RCC_MP_ AHB2LPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBOLPEN	Res.	ADC2LPEN	ADC1LPEN	DMAMUX2LPEN	DMA3LPEN	DMAMUX1LPEN	DMA2LPEN	DMA1LPEN
	Reset value																								1		1	1	1	1	1	1	1
0x858-0x864	Reserved	Res.																															
0x868	**RCC_MP_S_ AHB4LPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOILPEN	GPIOHLPEN	GPIOGLPEN	GPIOFLPEN	GPIOELPEN	GPIODLPEN	GPIOCLPEN	GPIOBLPEN	GPIOALPEN
	Reset value																								1	1	1	1	1	1	1	1	1
0x86C	**RCC_MP_S_ AHB4LPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOILPEN	GPIOHLPEN	GPIOGLPEN	GPIOFLPEN	GPIOELPEN	GPIODLPEN	GPIOCLPEN	GPIOBLPEN	GPIOALPEN
	Reset value																								1	1	1	1	1	1	1	1	1
0x870	**RCC_MP_NS_ AHB4LPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOILPEN	GPIOHLPEN	GPIOGLPEN	GPIOFLPEN	GPIOELPEN	GPIODLPEN	GPIOCLPEN	GPIOBLPEN	GPIOALPEN
	Reset value																								1	1	1	1	1	1	1	1	1
0x874	**RCC_MP_NS_ AHB4LPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOILPEN	GPIOHLPEN	GPIOGLPEN	GPIOFLPEN	GPIOELPEN	GPIODLPEN	GPIOCLPEN	GPIOBLPEN	GPIOALPEN
	Reset value																								1	1	1	1	1	1	1	1	1
0x878	**RCC_MP_ AHB5LPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKPSRAMLPEN	Res.	RNG1LPEN	HASH1LPEN	CRYP1LPEN	SAESLPEN	PKALPEN	Res.	Res.
	Reset value																								1		1	1	1	1	1		

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x87C	**RCC_MP_ AHB5LPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKPSRAMLPEN	Res.	RNG1LPEN	HASH1LPEN	CRYP1LPEN	SAESLPEN	PKALPEN	Res.	Res.
	Reset value																								1		1	1	1	1	1		
0x880	**RCC_MP_ AHB6LPENSETR**	ETH2STPEN	ETH2MACLPEN	ETH2RXLPEN	ETH2TXLPEN	ETH2CKLPEN	Res.	Res.	USBHLPEN	Res.	Res.	Res.	CRC1LPEN	Res.	Res.	SDMMC2LPEN	SDMMC1LPEN	Res.	QSPILPEN	Res.	FMCLPEN	Res.	ETH1MACLPEN	ETH1RXLPEN	ETH1TXLPEN	ETH1CKLPEN	Res.	Res.	Res.	Res.	Res.	MCELPEN	Res.
	Reset value	0	1	1	1	1			1				1			1	1		1		1		1	1	1	1						0	
0x884	**RCC_MP_ AHB6LPENCLRR**	ETH2STPEN	ETH2MACLPEN	ETH2RXLPEN	ETH2TXLPEN	ETH2CKLPEN	Res.	Res.	USBHLPEN	Res.	Res.	Res.	CRC1LPEN	Res.	Res.	SDMMC2LPEN	SDMMC1LPEN	Res.	QSPILPEN	Res.	FMCLPEN	Res.	ETH1MACLPEN	ETH1RXLPEN	ETH1TXLPEN	ETH1CKLPEN	Res.	Res.	Res.	Res.	Res.	MCELPEN	Res.
	Reset value	0	1	1	1	1			1				1			1	1		1		1		1	1	1	1						0	
0x888	**RCC_MP_S_ AHB6LPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MDMALPEN
	Reset value																																1
0x88C	**RCC_MP_S_ AHB6LPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MDMALPEN
	Reset value																																1
0x890	**RCC_MP_NS_ AHB6LPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MDMALPEN
	Reset value																																1
0x894	**RCC_MP_NS_ AHB6LPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MDMALPEN
	Reset value																																1

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x898	**RCC_MP_S_ AXIMLPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSRAMLPEN
	Reset value																																1
0x89C	**RCC_MP_S_ AXIMLPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSRAMLPEN
	Reset value																																1
0x8A0	**RCC_MP_NS_ AXIMLPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSRAMLPEN
	Reset value																																1
0x8A4	**RCC_MP_NS_ AXIMLPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSRAMLPEN
	Reset value																																1
0x8A8	**RCC_MP_ MLAHBLPENSET R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAM3LPEN	SRAM2LPEN	SRAM1LPEN
	Reset value																														1	1	1
0x8AC	**RCC_MP_ MLAHBLPENCLR R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAM3LPEN	SRAM2LPEN	SRAM1LPEN
	Reset value																														1	1	1
0x8B0 - 0x8BC	Reserved	Res.																															
0x8C0	**RCC_ APB3SECSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VREFSECF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM3SECF	LPTIM2SECF
	Reset value																			1												1	1

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x8C4	**RCC_ APB4SECSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBPHYSECF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCMIPPSECF	Res.
	Reset value																1															1	
0x8C8	**RCC_ APB5SECSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STGENCSECF[1:0]		Res.	Res.	Res.	BSECSECF	IWDG1SECF	Res.	ETZPCSECF	Res.	TZCSECF	Res.	Res.	RTCSECF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value											1	1				1	1		1		1			0								
0x8CC	**RCC_ APB6SECSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM17SECF	TIM16SECF	TIM15SECF	TIM14SECF	TIM13SECF	TIM12SECF	I2C5SECF	I2C4SECF	I2C3SECF	SPI5SECF	SPI4SECF	USART2SECF	USART1SECF
	Reset value																				1	1	1	1	1	1	1	1	1	1	1	1	1
0x8D0	**RCC_ AHB2SECSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBOSECF	Res.	ADC2SECF	ADC1SECF	DMAMUX2SECF	DMA3SECF	Res.	Res.	Res.
	Reset value																								1		1	1	1	1			
0x8D4	Reserved	Res.																															
0x8D8	**RCC_ AHB5SECSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKPSRAMSECF	Res.	RNG1SECF	HASH1SECF	CRYP1SECF	SAESSECF	PKASECF	Res.	Res.
	Reset value																								1		1	1	1	1	1		
0x8DC	**RCC_ AHB6SECSR**	ETH2SECF[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC2SECF	SDMMC1SECF	Res.	QSPISECF	Res.	FMCSECF	ETH1SECF[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	MCESECF	Res.
	Reset value	1	1	1	1	1										1	1		1		1	1	1	1	1	1						1	
0x8E0 - 0xFF0	Reserved	Res.																															
0xFF4	**RCC_VERR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	0	1	0	0	0	0	0

表 74. RCC 寄存器映射和复位值（续）

偏移	寄存器名称	寄存器位置																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFF8	**RCC_IDR**	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0xFFC	**RCC_SIDR**	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	1	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

11 通用 I/O (GPIO)

11.1 简介

每个通用 I/O 端口包括 4 个 32 位配置寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR 和 GPIOx_PUPDR）、2 个 32 位数据寄存器（GPIOx_IDR 和 GPIOx_ODR）和 1 个 32 位置位/复位寄存器 (GPIOx_BSRR)。此外，所有 GPIO 都包括 1 个 32 位锁定寄存器 (GPIOx_LCKR) 和 2 个 32 位复用功能选择寄存器（GPIOx_AFRH 和 GPIOx_AFRL）。

11.2 GPIO 主要特性

- 输出状态：推挽或开漏 + 上拉/下拉
- 从输出数据寄存器 (GPIOx_ODR) 或外设（复用功能输出）输出数据
- 可为每个 I/O 选择不同的速度
- 输入状态：浮空、上拉/下拉、模拟
- 将数据输入到输入数据寄存器 (GPIOx_IDR) 或外设（复用功能输入）
- 置位和复位寄存器 (GPIOx_BSRR)，对 GPIOx_ODR 具有按位写入访问
- 锁定机制 (GPIOx_LCKR)，可冻结 I/O 端口配置
- 模拟功能
- 复用功能选择寄存器
- 快速翻转，每次翻转最快只需要两个时钟周期
- 引脚复用非常灵活，允许将 I/O 引脚用作 GPIO 或多种外设功能中的一种

11.3 GPIO 功能描述

根据数据手册中列出的每个 I/O 端口的特性，可通过软件将通用 I/O (GPIO) 端口的各个端口位分别配置为多种模式：

- 输入浮空
- 输入上拉
- 输入下拉
- 模拟
- 具有上拉或下拉功能的开漏输出
- 具有上拉或下拉功能的推挽输出
- 具有上拉或下拉功能的复用功能推挽
- 具有上拉或下拉功能的复用功能开漏

每个 I/O 端口位均可自由编程，但 I/O 端口寄存器必须按 32 位字、半字或字节进行访问。GPIOx_BSRR 寄存器旨在实现对 GPIOx_ODR 寄存器进行原子读取/修改访问。这样便可确保在读取和修改访问之间发生中断请求也不会有问题。

图 103 和*图 104* 分别显示了标准和 5 V 容限 I/O 端口位的基本结构。*表 75* 给出了可能的端口位配置方案。

图 103. I/O 端口位的基本结构

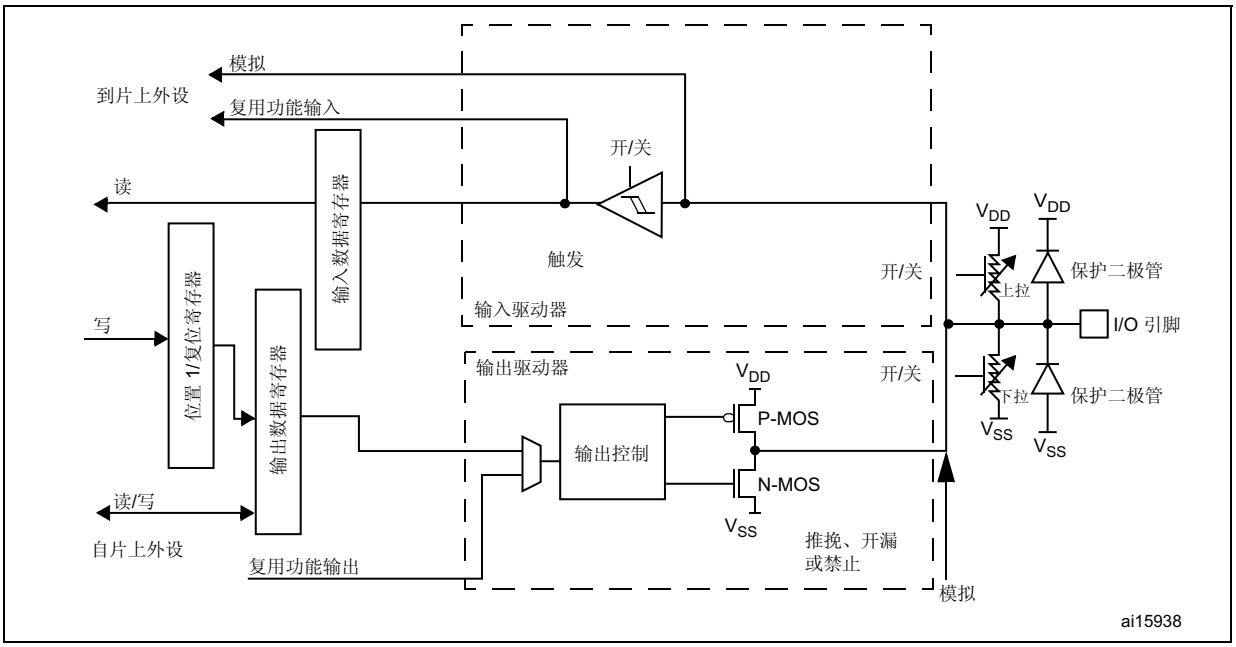
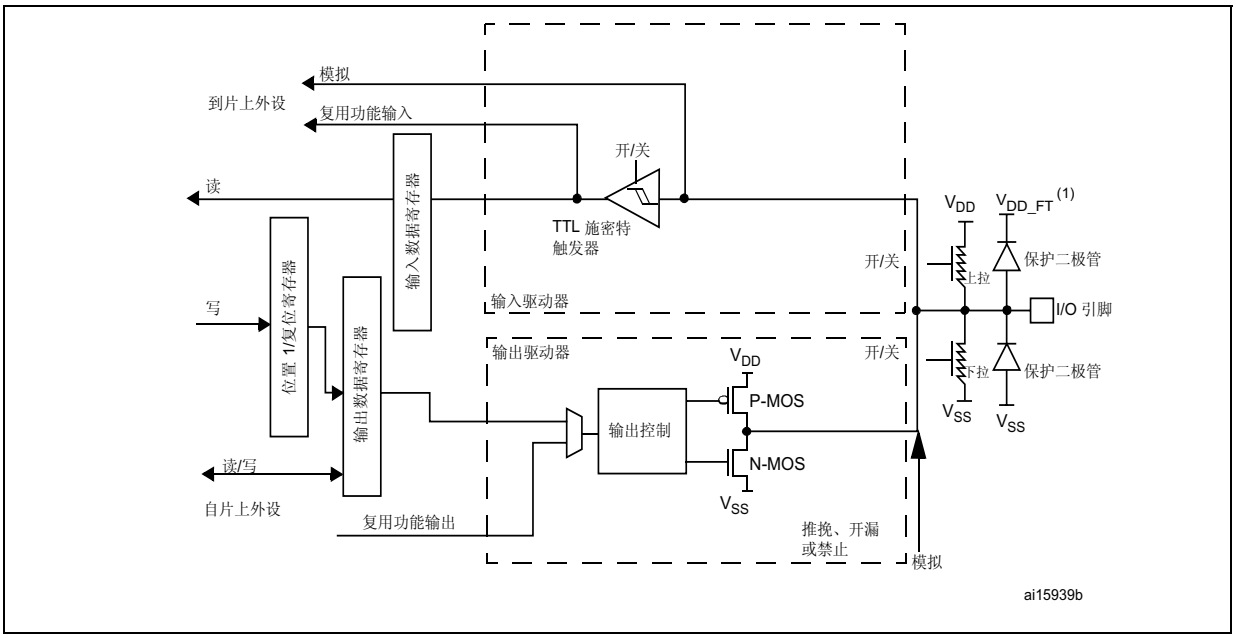


图 104. 5 V 容限 I/O 端口位的基本结构



1. V_{DD_FT} 是和 5 V 容限 I/O 相关的电位，与 V_{DD} 不同。

表 75. 端口位配置表(1)

MODE(i) [1:0]	OTYPER(i)	OSPEED(i) [1:0]		PUPD(i) [1:0]		I/O 配置	
01	0	SPEED [1:0]		0	0	GP 输出	PP
	0			0	1	GP 输出	PP + PU
	0			1	0	GP 输出	PP + PD
	0			1	1	保留	
	1			0	0	GP 输出	OD
	1			0	1	GP 输出	OD + PU
	1			1	0	GP 输出	OD + PD
	1			1	1	保留（GP 输出 OD）	
10	0	SPEED [1:0]		0	0	AF	PP
	0			0	1	AF	PP + PU
	0			1	0	AF	PP + PD
	0			1	1	保留	
	1			0	0	AF	OD
	1			0	1	AF	OD + PU
	1			1	0	AF	OD + PD
	1			1	1	保留	
00	x	x	x	0	0	输入	浮空
	x	x	x	0	1	输入	PU
	x	x	x	1	0	输入	PD
	x	x	x	1	1	保留（输入浮空）	
11	x	x	x	0	0	输入/输出	模拟
	x	x	x	0	1	保留	
	x	x	x	1	0		
	x	x	x	1	1		

1. GP = 通用、PP = 推挽、PU = 上拉、PD = 下拉、OD = 开漏、AF = 复用功能。

11.3.1 通用 I/O (GPIO)

在复位期间及复位刚刚完成后，复用功能尚未激活，大多数 I/O 端口被配置为输入浮空模式。

当引脚配置为输出后，写入到输出数据寄存器 (GPIOx_ODR) 的值将在 I/O 引脚上输出。可以在推挽模式下或开漏模式（仅驱动低电平，高电平为高阻态）下使用输出驱动器。

输入数据寄存器 (GPIOx_IDR) 每 1 个 AHB 时钟周期捕获一次 I/O 引脚的数据。

所有 GPIO 引脚都具有内部弱上拉及下拉电阻，可根据 GPIOx_PUPDR 寄存器中的值来打开/关闭。

11.3.2 I/O 引脚复用功能复用器和映射

设备 I/O 引脚通过一个复用器连接到板载外设/模块，该复用器一次仅允许一个外设的复用功能 (AF) 连接到 I/O 引脚。这可以确保共用同一个 I/O 引脚的外设之间不会发生冲突。

每个 I/O 引脚都有一个复用器，该复用器采用多达 16 路复用功能输入（AF0 到 AF15），可通过 GPIOx_AFRL（针对引脚 0 到 7）和 GPIOx_AFRH（针对引脚 8 到 15）寄存器对这些输入进行配置：

- 复位后，复用器选择为复用功能 0 (AF0)。在复用模式下通过 GPIOx_MODER 寄存器配置 I/O。
- 设备数据手册中详细说明了每个引脚的特定复用功能分配。

除了这种灵活的 I/O 复用架构之外，各外设还可以将复用功能映射到不同 I/O 引脚，这可以优化小型封装中可用外设的数量。

要在指定配置下使用 I/O，用户必须按照以下步骤操作：

- **系统功能**：必须将 MCOx 引脚配置为复用功能模式。
- **GPIO**：在 GPIOx_MODER 寄存器中将所需 I/O 配置为输出、输入或模拟通道。
- **外设复用功能**：
  - 在 GPIOx_AFRL 或 GPIOx_AFRH 寄存器中，将 I/O 连接到所需的 AFx。
  - 通过 GPIOx_OTYPER、GPIOx_PUPDR 和 GPIOx_OSPEEDER 寄存器，分别选择类型、上拉/下拉以及输出速度。
  - 在 GPIOx_MODER 寄存器中将所需 I/O 配置为复用功能。
- **其他功能**：
  - 对于 ADC，在 GPIOx_MODER 寄存器中将所需 I/O 配置为模拟模式，并在 ADC 寄存器中配置所需功能。
  - 对于 RTC_OUT、RTC_LSCO、TAMP_xxx、WKUPx 和振荡器的其他功能，在相关的 RTC、PWR 和 RCC 寄存器中配置所需功能。这些功能优先于标准 GPIO 寄存器中的配置。有关 RTC 的 I/O 控制的详细信息，请参见*第 46.3 节：RTC 功能描述*。

有关复用功能 I/O 引脚映射的详细信息，请参见设备数据手册中的“复用功能映射”表。

11.3.3 I/O 端口控制寄存器

每个 GPIO 端口有 4 个 32 位内存映射的控制寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR），可配置多达 16 个 I/O。GPIOx_MODER 寄存器用于选择 I/O 模式（输入、输出、AF、模拟）。GPIOx_OTYPER 和 GPIOx_OSPEEDR 寄存器用于选择输出类型（推挽或开漏）和速度。无论采用哪种 I/O 方向，GPIOx_PUPDR 寄存器都用于选择上拉/下拉。

11.3.4 I/O 端口数据寄存器

每个 GPIO 都具有 2 个 16 位内存映射的数据寄存器：输入和输出数据寄存器（GPIOx_IDR 和 GPIOx_ODR）。GPIOx_ODR 用于存储待输出数据，可对其进行读/写访问。通过 I/O 输入的数据存储到输入数据寄存器 (GPIOx_IDR) 中，它是一个只读寄存器。

有关寄存器说明的详细信息，请参见*第 11.4.5 节：GPIO 端口输入数据寄存器 (GPIOx_IDR)（x = A 到 I）*和*第 11.4.6 节：GPIO 端口输出数据寄存器 (GPIOx_ODR)（x = A 到 I）*。

11.3.5 I/O 数据位操作

置 1 复位寄存器 (GPIOx_BSRR) 是一个 32 位寄存器，它允许应用程序在输出数据寄存器 (GPIOx_ODR) 中对各个单独的数据位执行置 1 和复位操作。置位复位寄存器的大小是 GPIOx_ODR 的二倍。

GPIOx_ODR 中的每个数据位对应于 GPIOx_BSRR 中的两个控制位：BS(i) 和 BR(i)。当写入 1 时，BS(i) 位会将对应的 ODR(i) 位**置 1**。当写入 1 时，BR(i) 位会**复位** ODR(i) 对应的位。

在 GPIOx_BSRR 中向任何位写入 0 都不会对 GPIOx_ODR 中的对应位产生任何影响。如果在 GPIOx_BSRR 中同时尝试对某个位执行置 1 和复位操作，则置 1 操作优先。

使用 GPIOx_BSRR 寄存器更改 GPIOx_ODR 中各个位的值是一个“单次”操作，不会锁定 GPIOx_ODR 位。随时都可以直接访问 GPIOx_ODR 位。GPIOx_BSRR 寄存器提供了一种执行原子按位处理的方法。

在对 GPIOx_ODR 进行位操作时，软件无需禁止中断：在一次原子 AHB 写入访问中，可以修改一个或多个位。

11.3.6 GPIO 锁定机制

通过将特定的写入序列应用到 GPIOx_LCKR 寄存器，可以冻结 GPIO 控制寄存器。冻结的寄存器包括 GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL 和 GPIOx_AFRH。

要对 GPIOx_LCKR 寄存器执行写入操作，必须应用特定的写/读序列。当正确的 LOCK 序列应用到此寄存器的第 16 位后，会使用 LCKR[15:0] 的值来锁定 I/O 的配置（在写入序列期间，LCKR[15:0] 的值必须相同）。将 LOCK 序列应用到某个端口位后，在执行下一次 MCU 复位或外设复位之前，将无法对该端口位的值进行修改。每个 GPIOx_LCKR 位都会冻结控制寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL 和 GPIOx_AFRH）中的对应位。

LOCK 序列（参见*第 11.4.8 节：GPIO 端口配置锁定寄存器 (GPIOx_LCKR)（x = A 到 I）*）只能通过对 GPIOx_LCKR 寄存器进行字（32 位长）访问的方式来执行，因为 GPIOx_LCKR 的第 16 位必须与 [15:0] 位同时置位。

有关详细信息，请参见*第 11.4.8 节：GPIO 端口配置锁定寄存器 (GPIOx_LCKR)（x = A 到 I）*中的 LCKR 寄存器说明。

11.3.7 I/O 复用功能输入/输出

有两个寄存器可用来从每个 I/O 可用的复用功能输入/输出中进行选择。借助这些寄存器，用户可根据应用程序的要求将某个复用功能连接到其他某个引脚。

这意味着可使用 GPIOx_AFRL 和 GPIOx_AFRH 复用功能寄存器在每个 GPIO 上复用多个可用的外设功能。这样一来，应用程序可为每个 I/O 选择任何一个可用功能。由于 AF 选择信号由复用功能输入和复用功能输出共用，所以只需为给定 I/O 的复用功能输入/输出选择一个通道即可。

要了解在每个 GPIO 引脚上复用了哪些功能，请参见设备数据手册。

11.3.8 外部中断线/唤醒线

所有端口都具有外部中断功能。要使用外部中断线，切勿将端口配置为模拟模式。

请参见*第 20 章：扩展中断和事件控制器 (EXTI)*。

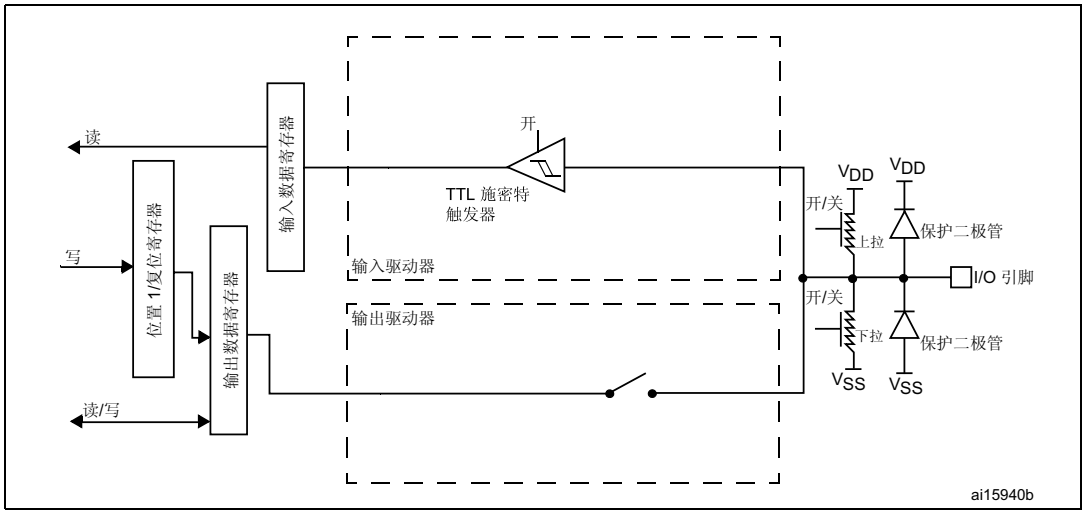
11.3.9 输入配置

对 I/O 端口进行编程作为输入时：

- 输出缓冲器被禁止
- 施密特触发器输入被打开
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开上拉和下拉电阻
- 输入数据寄存器每隔 1 个 AHB 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读取访问可获取 I/O 状态

图 105 说明了 I/O 端口位的输入配置。

图 105. 输入浮空/上拉/下拉配置



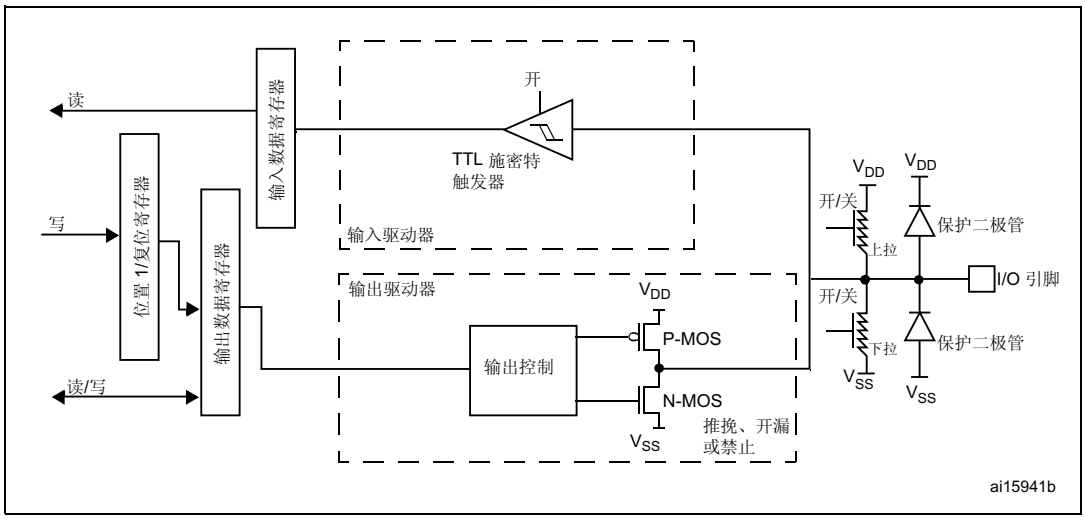
11.3.10 输出配置

对 I/O 端口进行编程作为输出时：

- 使能输出缓冲器：
  - 开漏模式：输出寄存器中的“0”可激活 N-MOS，而输出寄存器中的“1”会使端口保持高组态 (Hi-Z)（P-MOS 始终不激活）。
  - 推挽模式：输出寄存器中的“0”可激活 N-MOS，而输出寄存器中的“1”可激活 P-MOS
- 施密特触发器输入被打开
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开上拉和下拉电阻
- 输入数据寄存器每隔 1 个 AHB 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读取访问可获取 I/O 状态
- 对输出数据寄存器的读取访问可获取最后的写入值

图 106 说明了 I/O 端口位的输出配置。

图 106. 输出配置



11.3.11 I/O 补偿单元

补偿单元用于控制 I/O 换向斜率 (t_{fall}/t_{rise})，从而降低 I/O 噪声对众多 I/O 引脚电源的影响。

STM32MP13xx 设备中有 3 个补偿单元。

- 1 个用于专用电源 SDMMC1 部分（GPIO：PC[12:8]、PD2）
- 1 个用于专用电源 SDMMC2 部分（GPIO：PB[15,14,4,3]、PE3、PG6）
- 1 个用于其他 I/O 引脚。

该单元分为两个块：

- 第一个块提供当前 PVT 的最佳代码。当 SYSCFG_CMPCR 的 READY 标志置 1 时，可读取该块中存储的代码。
- 第二个块控制 I/O 斜率。用户选择要自动应用的代码并通过软件对其进行编程。

I/O 补偿单元具有 2 个电压范围：1.62 V 到 2.0 V 和 2.7 V 到 3.6 V。

11.3.12 高速低电压模式

用户可将部分 I/O 设置为高速低电压模式，从而在通过 HSLV 选项使用电压不到 2.5 V 典型值（最大值 2.7 V）的电源供电时，能够实现最快速度。

请参见*第 12.6.17 节：SYSCFG 高速低压使能寄存器 x (SYSCFG_HSLVENxR)* 和*表 79：高速低电压接口*以了解有关适用接口的详细信息，并参见数据手册以了解“使能 HSLV”的电气特性。

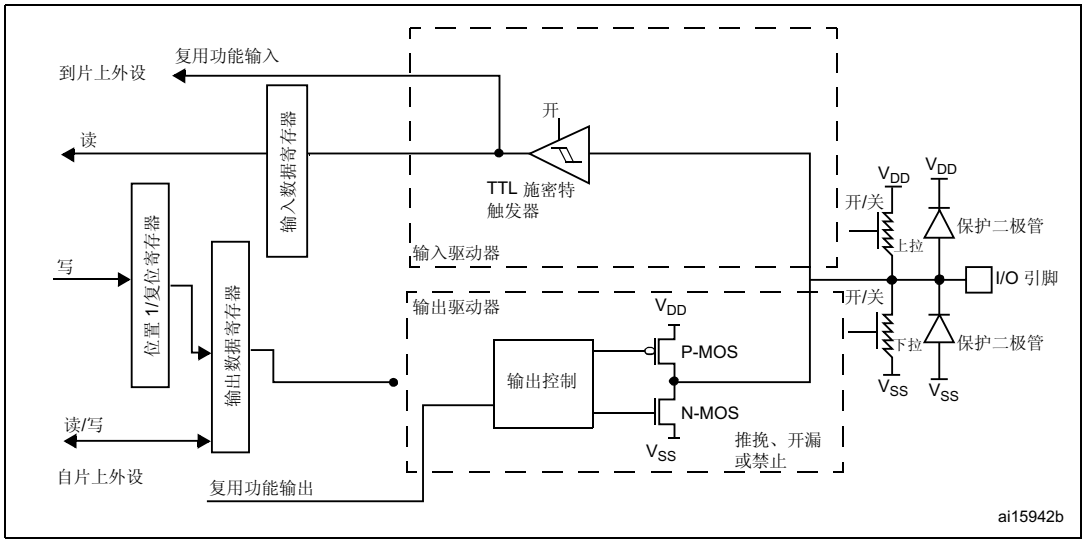
11.3.13 复用功能配置

对 I/O 端口进行编程作为复用功能时：

- 可将输出缓冲器配置为开漏或推挽模式
- 输出缓冲器由来自外设的信号驱动（发送器使能和数据）
- 施密特触发器输入被打开
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开弱上拉电阻和下拉电阻
- 输入数据寄存器每隔 1 个 AHB 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读取访问可获取 I/O 状态

图 107 说明了 I/O 端口位的复用功能配置。

图 107. 复用功能配置



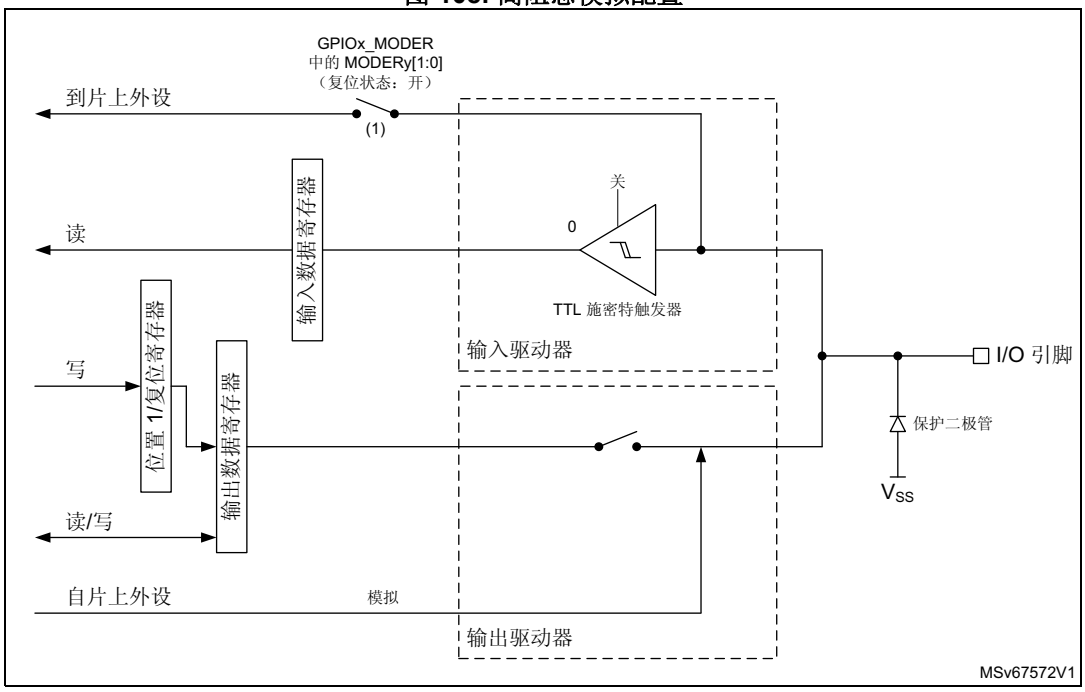
11.3.14 模拟配置

对 I/O 端口进行编程作为模拟配置时：

- 输出缓冲器被禁止
- 施密特触发器输入停用，I/O 引脚的每个模拟输入的功耗变为零。施密特触发器的输出被强制处理为恒定值 (0)。
- 弱上拉和下拉电阻被硬件关闭
- 对输入数据寄存器的读取访问值为“0”

图 108 说明了 I/O 端口位的高阻态模拟输入配置。

图 108. 高阻态模拟配置



1. 当 GPIOx_MODER 寄存器设置为“11”且已在 ADC_SQRx 寄存器或 ADC_JSQx 寄存器中选择了相应通道时，ADC 模拟开关将关闭。

11.3.15 将 HSE 或 LSE 振荡器引脚用作 GPIO

当 HSE 或 LSE 振荡器关闭（复位后的默认状态）时，可将相关的振荡器引脚用作常规 GPIO。

当 HSE 或 LSE 振荡器打开（通过设置 RCC_CSR 寄存器中的 HSEON 或 LSEON 位）时，振荡器会控制与其相关联的引脚，这些引脚的 GPIO 配置不起作用。这一点同样适用于外部时钟模式。

11.3.16 在备份电源域中使用 GPIO 引脚

当内核电源域掉电时（设备进入待机模式时），PC13/PC14/PC15/PI[3:0] GPIO 功能会丢失。在这种情况下，如果不通过 RTC 配置旁通其 GPIO 配置，这些引脚将被设置为模拟输入模式。

当内核电源域掉电时（设备进入待机模式时），PC13/PC14/PC15/PI[3:0] GPIO 功能会丢失。在这种情况下，如果未通过任何 RTC、TAMP 或 PWR(WKUP) 配置旁通其 GPIO 配置（包括 GPIO 上拉/下拉和开漏设置），这些引脚将强制进入模拟模式。

有关 RTC 的 I/O 控制的详细信息，请参见*第 46.3 节：RTC 功能描述*。

11.3.17 BOOT/GPIO 引脚共享

系统复位后（包括退出待机模式），BOOT 引脚 (PI[6:4]) 的复用功能 AF0 将默认处于选中状态，以便用户能够在 SYSCFG_BOOTR 寄存器中对 BOOT 引脚进行采样。随后，用户可以对相应的 GPIOx_MODER 和 GPIOx_PUPDR 寄存器进行编程，从而将这些引脚用作 GPIO。

小心： 在复位期间，必须使用正确的逻辑电平来驱动启动引脚，以选择所需的启动源（请参见产品数据手册）。

11.3.18 TrustZone 安全性

TrustZone 安全性始终处于激活状态。每个 I/O 引脚都可以通过 GPIOx_SECCFGR 寄存器单独配置为安全。

当所选 I/O 引脚配置为安全时，可保护其相应的配置位（用于复用功能 AFI、AFO、模式选择和 I/O 数据）免遭非安全访问：

- 输入数据不会被重定向到另一外设
- 输出数据不会被替换为另一外设
- 无法将安全 I/O 数据重定向到非安全 I/O，无论是将 I/O 配置为复用功能，还是通过外设将其配置为模拟、USB、RTC 或是 PWR 控制器中的唤醒引脚都无法实现这一点
- 无法将非安全 I/O 数据重定向到安全外设

复位后，所有 GPIOx I/O 引脚均安全。

表 76 对 GPIO_SECCFGR 寄存器中安全配置位之后的 I/O 端口安全位进行了汇总。

- 安全位：只允许通过安全访问进行读写操作。对安全位的非安全读取或写入访问为 RAZ/WI。不生成任何非法访问事件。
- 非安全位：无限制。可通过安全访问和非安全访问进行读写操作。

表 76. GPIO 安全位

安全配置位	安全位	寄存器名称	对安全位的非安全访问
GPIOx_SECCFGR 中的 SECy = 1	MODEy[1:0]	GPIOx_MODER	RAZ/WI
	OTy	GPIOx_OTYPER	
	OSPEEDy[1:0]	GPIOx_OSPEEDR	
	PUPDy[1:0]	GPIOx_PUPDR	
	IDy	GPIOx_IDR	
	ODy	GPIOx_ODR	
	BSy	GPIOx_BSRR	
	LCKy	GPIOx_LCKR	
	AFSELy[3:0]	GPIOx_AFRL	
	BRy	GPIOx_AFRH	
		GPIOx_BRR	

11.4 GPIO寄存器

本节对 GPIO 寄存器进行了详细介绍。

有关寄存器位、寄存器偏移地址和复位值的汇总，请参见*表 77*。

可按字、半字或字节模式写入外设寄存器。

11.4.1 GPIO 端口模式寄存器 (GPIOx_MODER)（x = A 到 I）

GPIO port mode register

偏移地址：0x00

复位值：0xFFFF FFFF

复位值：块 C：0xFFFF FFFC

复位值：块 F：0xAFFF FFFF

复位值：块 H：0x3FFF FAFF

复位值：块 I：0x0000 EAFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **MODER[15:0][1:0]**：端口 x 配置 I/O 引脚 y (Port x configuration I/O pin y)（y = 15 到 0）
这些位通过软件写入，用于配置 I/O 模式。
00：输入模式
01：通用输出模式
10：复用功能模式
11：模拟模式

11.4.2 GPIO 端口输出类型寄存器 (GPIOx_OTYPER)（x = A 到 I）

GPIO port output type register

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **OT[15:0]**：端口 x 配置 I/O 引脚 y (Port x configuration I/O pin y)（y = 15 到 0）
这些位通过软件写入，用于配置 I/O 输出类型。
0：输出推挽（复位状态）
1：输出开漏

11.4.3 GPIO 端口输出速度寄存器 (GPIOx_OSPEEDR)（x = A 到 I）

GPIO port output speed register

偏移地址：0x08

复位值：0x0000 0000

复位值：块 F：0xC000 0000

复位值：块 H：0x0000 0C00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEEDR15 [1:0]		OSPEEDR14 [1:0]		OSPEEDR13 [1:0]		OSPEEDR12 [1:0]		OSPEEDR11 [1:0]		OSPEEDR10 [1:0]		OSPEEDR9 [1:0]		OSPEEDR8 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEEDR7 [1:0]		OSPEEDR6 [1:0]		OSPEEDR5 [1:0]		OSPEEDR4 [1:0]		OSPEEDR3 [1:0]		OSPEEDR2 [1:0]		OSPEEDR1 [1:0]		OSPEEDR0 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **OSPEEDR[15:0][1:0]**：端口 x 配置 I/O 引脚 y (Port x configuration I/O pin y)（y = 15 到 0）
这些位通过软件写入，用于配置 I/O 输出速度。
00：低速
01：中速
10：高速
11：超高速
注意：有关 OSPEEDRy 位以及 V_{DD} 范围和外部负载的值，请参见产品数据手册。

11.4.4 GPIO 端口上拉/下拉寄存器 (GPIOx_PUPDR)（x = A 到 I）

GPIO port pull-up/pull-down register

偏移地址：0x0C

复位值：0x0000 0000

复位值：块 F：0x6000 0000

复位值：块 H：0x0000 0500

复位值：块 I：0x0000 2A00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **PUPDR[15:0][1:0]**：端口 x 配置 I/O 引脚 y (Port x configuration I/O pin y)（y = 15 到 0）
这些位通过软件写入，用于配置 I/O 上拉或下拉。
00：无上拉或下拉
01：上拉
10：下拉
11：保留

11.4.5 GPIO 端口输入数据寄存器 (GPIOx_IDR)（x = A 到 I）

GPIO port input data register

偏移地址：0x10

复位值：0x0000 XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **IDR[15:0]**：端口 x 输入数据 I/O 引脚 y (Port x input data I/O pin y)（y = 15 到 0）
这些位为只读。它们包含相应 I/O 端口的输入值。

11.4.6 GPIO 端口输出数据寄存器 (GPIOx_ODR)（x = A 到 I）

GPIO port output data register

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **ODR[15:0]**：端口输出数据 I/O 引脚 y (Port output data I/O pin y)（y = 15 到 0）
这些位可通过软件读取和写入。
注意：对于原子位置 1/复位，通过写入 GPIOx_BSRR 寄存器，可分别对 ODR 位进行置 1 和/或复位 (x = A..F)。

11.4.7 GPIO 端口位置 1/复位寄存器 (GPIOx_BSRR)（x = A 到 I）

GPIO port bit set/reset register

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:16 **BR[15:0]**：端口 x 复位 I/O 引脚 y (Port x reset I/O pin y)（y = 15 到 0）
这些位为只写。读取这些位会返回值 0x0000。
0：不会对相应的 ODRx 位执行任何操作
1：对相应的 ODRx 位进行复位
注意：如果同时对 BSx 和 BRx 置 1，则 BSx 的优先级更高。

位 15:0 **BS[15:0]**：端口 x 置 1 I/O 引脚 y (Port x set I/O pin y)（y = 15 到 0）
这些位为只写。读取这些位会返回值 0x0000。
0：不会对相应的 ODRx 位执行任何操作
1：对相应的 ODRx 位进行置位

11.4.8 GPIO 端口配置锁定寄存器 (GPIOx_LCKR)（x = A 到 I）

GPIO port configuration lock register

当正确的写入序列应用到位 16 (LCKK) 时，此寄存器将用于锁定端口位的配置。位 [15:0] 的值用于锁定 GPIO 的配置。在写入序列期间，不能更改 LCKR[15:0] 的值。将 LOCK 序列应用到某个端口位后，在执行下一次 MCU 复位或外设复位之前，将无法对该端口位的值进行修改。

注： *使用特定的写入序列对 GPIOx_LCKR 寄存器执行写入操作。在此锁定序列期间只允许使用字访问（32 位长）。*

每个锁定位冻结一个特定的配置寄存器（控制寄存器和复用功能寄存器）。

偏移地址：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **LCKK**：锁定键 (Lock key)

可随时读取该位。只可使用锁定键写入序列对其进行修改。

0：端口配置锁定键未激活。

1：端口配置锁定键已激活。GPIOx_LCKR 寄存器被锁定，直到下一次 MCU 复位或外设复位。

锁定键写入序列：

WR LCKR[16] =“1”+ LCKR[15:0]

WR LCKR[16] =“0”+ LCKR[15:0]

WR LCKR[16] =“1”+ LCKR[15:0]

RD LCKR

RD LCKR[16] =“1”（此读取操作为可选操作，但它可确认锁定已激活）

注意：在锁定键写入序列期间，不能更改 LCK[15:0] 的值。

锁定序列中的任何错误都将中止锁定操作。

在任一端口位上的第一个锁定序列之后，对 LCKK 位的任何读取访问都将返回“1”，直到下一次 MCU 复位或外设复位为止。

位 15:0 **LCK[15:0]**：端口 x 锁定 I/O 引脚 y (Port x lock I/O pin y)（y = 15 到 0）

这些位都是读/写位，但只能在 LCKK 位等于“0”时执行写入操作。

0：端口配置未锁定

1：端口配置已锁定

11.4.9 GPIO 复用功能低位寄存器 (GPIOx_AFRL)（x = A 到 I）

GPIO alternate function low register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR7[3:0]				AFR6[3:0]				AFR5[3:0]				AFR4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR3[3:0]				AFR2[3:0]				AFR1[3:0]				AFR0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **AFR[7:0][3:0]**：端口 x I/O 引脚 y 的复用功能选择 (Alternate function selection for port x I/O pin y)（y = 7 到 0）

这些位通过软件写入，用于配置复用功能 I/O。

0000：AF0
0001：AF1
0010：AF2
0011：AF3
0100：AF4
0101：AF5
0110：AF6
0111：AF7
1000：AF8
1001：AF9
1010：AF10
1011：AF11
1100：AF12
1101：AF13
1110：AF14
1111：AF15

11.4.10 GPIO 复用功能高位寄存器 (GPIOx_AFRH)（x = A 到 I）

GPIO alternate function high register

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR15[3:0]				AFR14[3:0]				AFR13[3:0]				AFR12[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR11[3:0]				AFR10[3:0]				AFR9[3:0]				AFR8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **AFR[15:8][3:0]**：端口 x I/O 引脚 y 的复用功能选择 (Alternate function selection for port x I/O pin y)（y = 15 到 8）

这些位通过软件写入，用于配置复用功能 I/O。

0000：AF0
0001：AF1
0010：AF2
0011：AF3
0100：AF4
0101：AF5
0110：AF6
0111：AF7
1000：AF8
1001：AF9
1010：AF10
1011：AF11
1100：AF12
1101：AF13
1110：AF14
1111：AF15

11.4.11 GPIO 端口位复位寄存器 (GPIOx_BRR)（x = A 到 I）

GPIO port bit reset register

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:16 保留，必须保持复位值。

位 15:0 **BR[15:0]**：端口 x 复位 IO 引脚 y (Port x reset IO pin y)（y = 15 到 0）
这些位为只写。读取这些位会返回值 0x0000。
0：不会对相应的 ODx 位执行任何操作
1：复位相应的 ODx 位

11.4.12 GPIO 安全配置寄存器 (GPIOx_SECCFGR)（x = A 到 I）

GPIO secure configuration register

此寄存器提供了访问安全控制，仅可通过安全访问进行写入操作。它用于将所选 I/O 配置为安全。将丢弃对该寄存器的非安全写入访问。读取将返回当前 GPIOx 引脚的安全设置。

偏移地址：0x30

复位值：0x0000 FFFF

复位值：块 H：0x0000 7FFF

复位值：块 I：0x0000 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC15	SEC14	SEC13	SEC12	SEC11	SEC10	SEC9	SEC8	SEC7	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	WR	rw	rw	rw

位 31:16 保留

位 15:0 **SEC[15:0]**：端口 x 的 I/O 引脚安全位使能 y (I/O pin of Port x secure bit enable y) (y= 0..15)
这些位将控制安全 I/O 端口引脚。
0：I/O 引脚非安全
1：I/O 引脚安全。有关所有相应的安全位，请参见*表 76：GPIO 安全位*。

11.4.13 GPIO 硬件配置寄存器 10 (GPIOx_HWCFGR10)（x = A 到 I）

GPIO hardware configuration register 10

偏移地址：0x3C8

复位值：0x0001 1240

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OR_CFG[3:0]				SEC_CFG[3:0]			
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOCK_CFG[3:0]				SPEED_CFG[3:0]				AF_SIZE[3:0]				AHB_IOP[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 保留，必须保持复位值。

位 23:20 **OR_CFG[3:0]**：选项寄存器配置 (Option register configuration)

0：未激活

1：激活

位 19:16 **SEC_CFG[3:0]**：安全机制激活 (Security mechanism activation)

0：未激活

1：激活

位 15:12 **LOCK_CFG[3:0]**：锁定机制激活 (Lock mechanism activation)

0：未激活

1：激活

位 11:8 **SPEED_CFG[3:0]**：每个 I/O 的速度线路数 (Number of speed lines for each I/O)

1：单线

2: 双线

位 7:4 **AF_SIZE[3:0]**：每个 I/O 的可用 AF 数 (Number of AF available for each I/O)

接受值：1 到 4

1：一个 AF/IO

2: 四个 AF/IO

3: 八个 AF/IO

4: 16 个 AF/IO

位 3:0 **AHB_IOP[3:0]**：总线接口配置 (Bus interface configuration)

0：AHB

1：IOP

11.4.14 GPIO 硬件配置寄存器 9 (GPIOx_HWCFGR9)（x = A 到 I）

GPIO hardware configuration register 9

偏移地址：0x3CC

复位值：0x0000 FFFF

复位值：块 H：0x0000 7FFF

复位值：块 I：0x0000 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN_IO[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **EN_IO[15:0]**：存在粒度，每个位均用来指示 IO 的存在情况
0：IO 不存在
1：IO 存在

11.4.15 GPIO 硬件配置寄存器 8 (GPIOx_HWCFGR8)（x = A 到 I）

GPIO hardware configuration register 8

偏移地址：0x3D0

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AF_PRIO15[3:0]				AF_PRIO14[3:0]				AF_PRIO13[3:0]				AF_PRIO12[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AF_PRIO11[3:0]				AF_PRIO10[3:0]				AF_PRIO9[3:0]				AF_PRIO8[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **AF_PRIO15[3:0]**：指示 I/O15 的优先级 AF（0 到 F）

位 27:24 **AF_PRIO14[3:0]**：指示 I/O14 的优先级 AF（0 到 F）

位 23:20 **AF_PRIO13[3:0]**：指示 I/O13 的优先级 AF（0 到 F）

位 19:16 **AF_PRIO12[3:0]**：指示 I/O12 的优先级 AF（0 到 F）

位 15:12 **AF_PRIO11[3:0]**：指示 I/O11 的优先级 AF（0 到 F）

位 11:8 **AF_PRIO10[3:0]**：指示 I/O10 的优先级 AF（0 到 F）

位 7:4 **AF_PRIO9[3:0]**：指示 I/O9 的优先级 AF（0 到 F）

位 3:0 **AF_PRIO8[3:0]**：指示 I/O8 的优先级 AF（0 到 F）

11.4.16 GPIO 硬件配置寄存器 7 (GPIOx_HWCFGR7)（x = A 到 I）

GPIO hardware configuration register 7

偏移地址：0x3D4

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AF_PRIO7[3:0]				AF_PRIO6[3:0]				AF_PRIO5[3:0]				AF_PRIO4[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AF_PRIO3[3:0]				AF_PRIO2[3:0]				AF_PRIO1[3:0]				AF_PRIO0[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **AF_PRIO7[3:0]**：指示 I/O7 的优先级 AF（0 到 F）

位 27:24 **AF_PRIO6[3:0]**：指示 I/O6 的优先级 AF（0 到 F）

位 23:20 **AF_PRIO5[3:0]**：指示 I/O5 的优先级 AF（0 到 F）

位 19:16 **AF_PRIO4[3:0]**：指示 I/O4 的优先级 AF（0 到 F）

位 15:12 **AF_PRIO3[3:0]**：指示 I/O3 的优先级 AF（0 到 F）

位 11:8 **AF_PRIO2[3:0]**：指示 I/O2 的优先级 AF（0 到 F）

位 7:4 **AF_PRIO1[3:0]**：指示 I/O1 的优先级 AF（0 到 F）

位 3:0 **AF_PRIO0[3:0]**：指示 I/O0 的优先级 AF（0 到 F）

11.4.17 GPIO 硬件配置寄存器 6 (GPIOx_HWCFGR6)（x = A 到 I）

GPIO hardware configuration register 6

偏移地址：0x3D8

复位值：0xFFFF FFFF

复位值：块 F：0xAFFF FFFF

复位值：块 H：0xFFFF FAFF

复位值：块 I：0xFFFF EAFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER_RES[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER_RES[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **MODER_RES[31:0]**：MODER 寄存器复位值 (MODER register reset value)

11.4.18 GPIO 硬件配置寄存器 5 (GPIOx_HWCFGR5)（x = A 到 I）

GPIO hardware configuration register 5

偏移地址：0x3DC

复位值：0x0000 0000

复位值：块 F：0x6000 0000

复位值：块 H：0x0000 0500

复位值：块 I：0x0000 2A00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR_RES[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR_RES[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **PUPDR_RES[31:0]**：上拉/下拉寄存器复位值 (Pull-up / pull-down register reset value)

11.4.19 GPIO 硬件配置寄存器 4 (GPIOx_HWCFGR4)（x = A 到 I）

GPIO hardware configuration register 4

偏移地址：0x3E0

复位值：0x0000 0000

复位值：块 F：0xC000 0000

复位值：块 H：0x0000 0C00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEED_RES[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEED_RES[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **OSPEED_RES[31:0]**：OSPEED 寄存器复位值 (OSPEED register reset value)

11.4.20 GPIO 硬件配置寄存器 3 (GPIOx_HWCFGR3)（x = A 到 I）

GPIO hardware configuration register 3

偏移地址：0x3E4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OTYPER_RES[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR_RES[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 **OTYPER_RES[31:0]**：输出类型寄存器复位值 (Output type register reset value)

位 15:0 **ODR_RES[31:0]**：输出数据寄存器复位值 (Output data register reset value)

11.4.21 GPIO 硬件配置寄存器 2 (GPIOx_HWCFGR2)（x = A 到 I）

GPIO hardware configuration register 2

偏移地址：0x3E8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFRL_RES[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFRL_RES[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **AFRL_RES[31:0]**：AF 寄存器低位复位值 (AF register low reset value)

11.4.22 GPIO 硬件配置寄存器 1 (GPIOx_HWCFGR1)（x = A 到 I）

GPIO hardware configuration register 1

偏移地址：0x3EC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFRH_RES[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFRH_RES[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **AFRH_RES[31:0]**：AF 寄存器高位复位值 (AF register high reset value)

11.4.23 GPIO 硬件配置寄存器 0 (GPIOx_HWCFGR0)（x = A 到 I）

GPIO hardware configuration register 0

偏移地址：0x3F0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OR_RES[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **OR_RES[15:0]**：选项寄存器复位值 (Option register reset value)

11.4.24 GPIO 版本寄存器 (GPIOx_VERR)（x = A 到 I）

GPIO version register

偏移地址：0x3F4

复位值：0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本 (Major revision)
这些位返回 GPIO 主版本。
主版本为 4。

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)
这些位返回 GPIO 次版本。
次版本为 0。

11.4.25 GPIO 标识寄存器 (GPIOx_IPIDR)（x = A 到 I）

GPIO identification register

偏移地址：0x3F8

复位值：0x000F 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPIDR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPIDR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **IPIDR[31:0]**：GPIO 标识符 (GPIO identifier)
这些位返回 GPIO 标识符值。

11.4.26 GPIO 大小标识寄存器 (GPIOx_SIDR)（x = A 到 I）

GPIO size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIDR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIDR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SIDR[31:0]**：大小标识符寄存器 (Size identifier register)
这些位返回分配给 GPIO 寄存器的存储区域的大小。

11.4.27 GPIO 寄存器映射

下表提供了 GPIO 寄存器映射和复位值。

表 77. GPIO 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**GPIOx_MODER** (where x = **A to I**)	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER[1:0]		MODER0[1:0]	
	Reset value for GPIOA, B, D, E, G	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Reset value for GPIOC	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	Reset value for GPIOF	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Reset value for GPIOH	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1
	Reset value for GPIOI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	1	1	1	1	1	1	1	1
0x04	**GPIOx_OTYPER** (where x = A to I)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	**GPIOx_OSPEEDR** (where x = A to I)	OSPEEDR15[1:0]		OSPEEDR14[1:0]		OSPEEDR13[1:0]		OSPEEDR12[1:0]		OSPEEDR11[1:0]		OSPEEDR10[1:0]		OSPEEDR9[1:0]		OSPEEDR8[1:0]		OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1[1:0]		OSPEEDR0[1:0]	
	Reset value for GPIOA..E, G, I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Reset value for GPIOF	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Reset value for GPIOH	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
0x0C	**GPIOx_PUPDR** (where x = A to I)	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
	Reset value for GPIOA..E, G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Reset value for GPIOF	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Reset value for GPIOH	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
	Reset value for GPIOI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0
0x10	**GPIOx_IDR** (where x = A to I)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
	Reset value																	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x14	**GPIOx_ODR** (where x = A to I)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	**GPIOx_BSRR** (where x = A to I)	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0	BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 77. GPIO 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1C	**GPIOx_LCKR** (where x = A to I)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK	LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**GPIOx_AFRL** (where x = A to I)	AFR7[3:0]				AFR6[3:0]				AFR5[3:0]				AFR4[3:0]				AFR3[3:0]				AFR2[3:0]				AFR1[3:0]				AFR0[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	**GPIOx_AFRH** (where x = A to I)	AFR15[3:0]				AFR14[3:0]				AFR13[3:0]				AFR12[3:0]				AFR11[3:0]				AFR10[3:0]				AFR9[3:0]				AFR8[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**GPIOx_BRR** (where x = A to I)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	**GPIOx_SECCFGR** (where x = A to I)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEC15	SEC14	SEC13	SEC12	SEC11	SEC10	SEC9	SEC8	SEC7	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
	Reset value for GPIOA..G																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Reset value for GPIOH																	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Reset value for GPIOI																	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0x03C8	**GPIOx_HWCFGR10 (where x = A to I)**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OR_CFG [3:0]				SEC_CFG [3:0]				LOCK_CFG [3:0]				SPEED_CFG [3:0]				AF_SIZE [3:0]				AHB_IOP [3:0]			
	Reset value									0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0
0x03CC	**GPIOx_HWCFGR9 (where x = A to I)**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN_IO[15:0]															
	Reset value for GPIOA..G																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Reset value for GPIOH																	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Reset value for GPIOI																	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0x03D0	**GPIOx_HWCFGR8 (where x = A to I)**	AF_PRIO15[3:0]				AF_PRIO14[3:0]				AF_PRIO13[3:0]				AF_PRIO12[3:0]				AF_PRIO11[3:0]				AF_PRIO10[3:0]				AF_PRIO9[3:0]				AF_PRIO8[3:0]			
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x03D4	**GPIOx_HWCFGR7 (where x = A to I)**	AF_PRIO7[3:0]				AF_PRIO6[3:0]				AF_PRIO5[3:0]				AF_PRIO4[3:0]				AF_PRIO3[3:0]				AF_PRIO2[3:0]				AF_PRIO1[3:0]				AF_PRIO0[3:0]			
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x03D8	**GPIOx_HWCFGR6 (where x = A to I)**	MODER_RES[31:16]																MODER_RES[15:0]															
	Reset value for GPIOA..E, G	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Reset value for GPIOF	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Reset value for GPIOH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1
	Reset value for GPIOI	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	1	1	1	1	1	1	1
0x03DC	**GPIOx_HWCFGR5 (where x = A to I)**	PUPDR_RES[31:16]																PUPDR_RES[15:0]															
	Reset value for GPIOA..E, G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Reset value for GPIOF	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Reset value for GPIOH	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
	Reset value for GPIOI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0

表 77. GPIO 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x03E0	**GPIOx_HWCFGR4 (where x = A to I)**	OSPEED_RES[31:16]																OSPEED_RES[15:0]															
	Reset value for GPIOA..E, G, I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Reset value for GPIOF	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Reset value for GPIOH	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
0x03E4	**GPIOx_HWCFGR3 (where x = A to I)**	OTYPER_RES[31:16]																ODR_RES[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03E8	**GPIOx_HWCFGR2 (where x = A to I)**	AFRL_RES[31:16]																AFRL_RES[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03EC	**GPIOx_HWCFGR1 (where x = A to I)**	AFRH_RES[31:16]																AFRH_RES[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03F0	**GPIOx_HWCFGR0 (where x = A to I)**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OR_RES[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03F4	**GPIO_VERR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	1	0	0	0	0	0	0
0x03F8	**GPIO_IPIDR**	ID[31:16]																ID[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0x03FC	**GPIO_SIDR**	SID[31:16]																SID[15:0]															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

12 系统配置控制器 (SYSCFG)

系统配置控制器主要用于管理补偿单元以及其他 I/O 和系统相关设置。除非特别说明，否则本节适用于整个 STM32MP13xx 系列。

12.1 TrustZone 安全性

系统配置控制器支持“TrustZone 感知”。某些寄存器位字段只能通过安全写入访问进行修改。这一点适用于负责控制安全外设，或是控制可能影响运行安全功能的特性的位字段。标记为安全的位字段仍然可以通过非安全读取访问进行读取。

12.2 I/O 补偿单元

I/O 集成了输出缓冲器的斜率控制功能，用以减少由快速切换导致的电源噪声。系统可以自动调整 I/O 的上升和下降时间，以补偿工艺、温度和老化所导致的变化；用户也可使用补偿单元并通过软件进行手动调整。

一共有三个 I/O 补偿单元，一个适用于 V_{DDSD1} 电源部分的 I/O，一个适用于 V_{DDSD2} 电源部分的 I/O，还有一个适用于所有其他的 I/O。

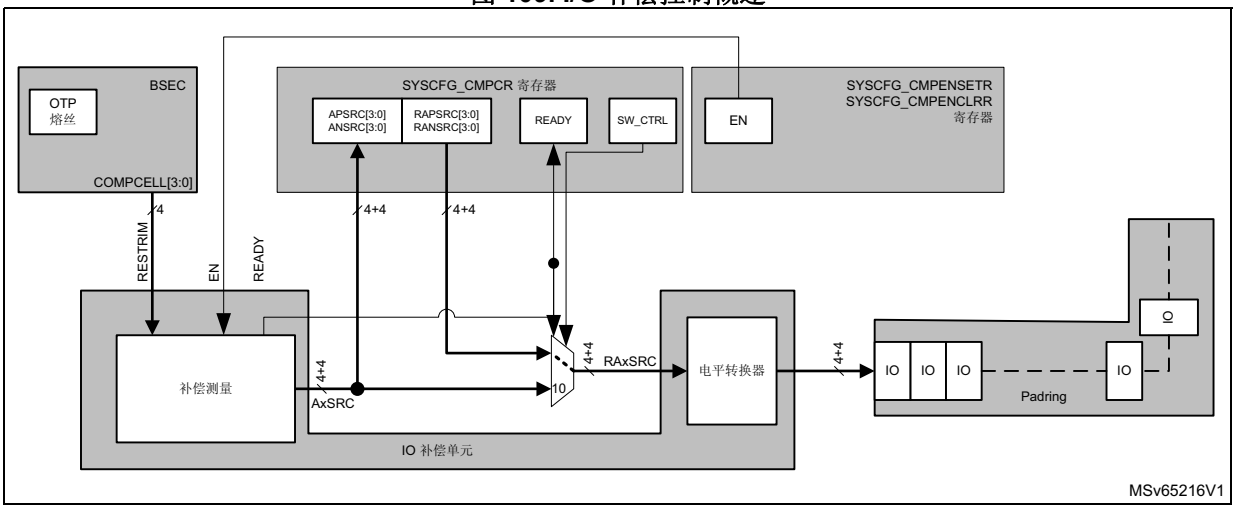
默认情况下，系统将禁止自动补偿。

在为不低于 50 MHz 的频率配置 I/O 输出缓冲区速度（GPIOx_OSPEEDRy = 0b10 或 0b11）时，建议使能自动补偿。

注： *使能后，补偿单元只会对 GPIOx_OSPEEDRy = 0b10 或 0b11 的 I/O 产生影响。*

补偿单元使能后，会设置一个“就绪”标志，指示补偿单元已就绪，可供使用。随后补偿值将会持续更新。

图 109. I/O 补偿控制概述



应使用以下序列来使能主 I/O 环的 I/O 补偿：

1. 确保 CSI 振荡器已（在 RCC 中）使能并就绪
2. 将 SYSCFG_CMPENSETR.EN 置 1（使能自动补偿）
3. 等待 SYSCFG_CMPCR.READY = 1
4. 将 SYSCFG_CMPCR.SW_CTRL 置 0（应用自动补偿）

使用以下序列来禁止 I/O 补偿，并保持当前补偿值（例如在进入停止模式之前）不变：

1. 将 SYSCFG_CMPCR.APSRC[3:0]/ANSRC[3:0] 的当前值复制到 SYSCFG_CMPCR.RAPSRC[3:0]/RANSRC[3:0]
2. 将 SYSCFG_CMPCR.SW_CTRL 置 1（应用软件补偿）
3. 将 SYSCFG_CMPENCLRR.EN 置 1（禁止自动补偿）

应为具有专用控制寄存器的 V_{DDSD1} 和 V_{DDSD2} 补偿单元使用相同的序列。

注： *如果在停止模式下未收到其他使用请求，系统将自动禁止 CSI 振荡器，且在待机模式下 CSI 将始终处于关闭状态。*

12.3 以太网 PHY 类型和时钟源选择

以太网 MAC 支持连接外部 PHY 的 RGMII、RMII 和 MII 接口。接口类型可在 SYSCFG_PMCSETR/CLRR 寄存器的 ETHx_SEL 位字段中进行选择。必须在使能以太网 MAC 之前设置该类型。

同一寄存器中的 ETHx_REF_CLK_SEL 和 ETHx_CLK_SEL 位字段将选择 MAC 和 PHY 所使用的参考时钟的外部或内部源。有关详细信息，请参阅“复位和时钟控制 (RCC)”一章。

12.4 模拟开关电源

ADC1 和 ADC2 所使用的某些 I/O 集成了模拟开关。这些开关的电阻会随着供电电压的降低而增大，进而导致模拟信号失真。SYSCFG_PMCSETR 和 SYSCFG_PMCCLRR 寄存器中存在以下两个位，可支持软件为这些开关选择电源：

ANASWVDD 可在 V_{DDA} 和 V_{DD} 电源之间进行选择。

EN_BOOST 可使能一个电荷泵，以提升 V_{DDA} 供电电压。

为了获得最佳的模拟性能，应将模拟开关的供电电压保持在 2.7V 以上：

如果 V_{DDA} > 2.7 V，则应将 ANASWVDD 编程为选择 V_{DDA}。

如果 V_{DD} > 2.7 V 且 V_{DDA} < 2.7 V，则应将 ANASWVDD 编程为选择 V_{DD}。

在这两种情况下，无需使用升压器并应将其关闭 (EN_BOOST = 0)，以节省电能。

如果 V_{DD} 和 V_{DDA} 都 < 2.7 V，则应选择 V_{DDA} 并通过将 EN_BOOST 置 1 来使能升压器

有关推荐设置，请参见*表 78*。

表 78. 模拟开关供电电压设置

条件	ANASWVDD	EN_BOOST	模拟开关电源
V_{DDA} > 2.7 V	0	0	V_{DDA}
V_{DD} > 2.7 V，V_{DDA} < 2.7 V	1	X	V_{DD}
V_{DD} < 2.7 V，V_{DDA} < 2.7 V	0	1	V_{DD} 提升至 3.3 V

由于所有模拟开关共用同一电源，更改电源配置会影响使用这些开关的所有外设的行为。因此，ANASWVDD 和 EN_BOOST 位始终处于安全状态，且只能通过安全写入进行修改。

只要 V_{DDCORE} 和 V_{DDA} 同时存在，ANASWVDD 设置就会始终处于应用状态。但请注意，如果 V_{DDA} 不存在或是 V_{DDCORE} 关闭（例如在待机模式下），模拟开关电源会自动切换到 V_{DD}，而不会影响 ANASWVDD 位的状态。

当 EN_BOOST = 1 且模拟外设（ADC1 或 ADC2）处于激活状态（即已使能其时钟）时，升压器将打开。一旦全部三个外设时钟均已禁止，升压器将自动关闭。

12.5 高速低压 (HSLV) 输出缓冲器

某些需要以高频率 (> 50 MHz) 进行传输的外部接口配备了支持 HSLV 的输出缓冲器。在正常模式下，当 V_{DD} = 1.8 V 时，输出频率要比 V_{DD} = 3.3 V 时更低。当使能 HSLV 模式时，缓冲器的驱动能力将有所提升，且 1.8 V 电压下的最大输出频率将与 3.3 V 电压的正常模式下的频率相同。

支持 HSLV 模式的接口已在*表 79* 中列出。

表 79. 高速低电压接口

索引 *x*	接口
0	跟踪端口 (TPIU)
1	QuadSPI (QSPI)
2	以太网 1 (ETH1)
3	以太网 2 (ETH2)
4	SDMMC1
5	SDMMC2
6	SPI1
7	SPI2
8	SPI3
9	SPI4
10	SPI5
11	LTDC

对于每个接口 *x*，必须对 SYSCFG_HSLVEN*x*R 寄存器进行编程以使能 HSLV 模式。可通过向寄存器写入一个特殊的密钥来完成此操作。写入任何其他值都将禁止 HSLV 模式。读取 SYSCFG_HSLVEN*x*R 寄存器将返回接口的 HSLV 使能状态。

警告： **如果 I/O 电源电压高于 2.7 V，则不得使能 HSLV 模式。这可能会导致设备损坏。为了降低意外使能 HSLV 模式的风险，我们设置了一个 OTP 位 (product_below_2v5)；用户必须先将其编程为高电平，然后才可以使能 HSLV 功能。如果 OTP 位为低电平，系统将忽略 HSLVENxR 寄存器设置。这一点适用于除 SDMMC1 和 SDMMC2 以外的所有接口。**

在使用 SDMMC 接口的内置电平转换器功能时，必须格外小心。在这种情况下，SDMMC1 接口的 I/O 电源电压将为 V_{DDSD1}，SDMMC2 接口的 I/O 电源电压将为 V_{DDSD2}。

当 SDMMC 接口在没有外部电平转换器的情况下以超高速 I 相 (UHS-I) 模式运行时，V_{DDSD1} 或 V_{DDSD2} 供电电压会在软件控制下进行动态切换；从 3.3 V 切换到 1.8 V 或是从 1.8 V 切换到 3.3 V。软件必须在将 $V_{DDSD1/2}$ 供电电压切换到 1.8 V（*第 24.8 节：超高速 I 相 (UHS-I) 电压切换*中的步骤 5）**之后**，为 SDMMC1/2 使能 HSLV 模式。同样，软件必须在将 $V_{DDSD1/2}$ 供电电压切换回 3.3 V **之前**，为 SDMMC1/2 禁止 HSLV 模式。

注： *当使用软件通过 RCC AHB6 外设复位设置寄存器 (RCC_AHB6RSTSETR) 来复位接口时，系统将禁止 SDMMC1 或 SDMMC2 接口的 HSLV 模式。*

12.6 SYSCFG 寄存器

12.6.1 SYSCFG 启动引脚控制寄存器 (SYSCFG_BOOTR)

SYSCFG boot pins control register

偏移地址：0x000

复位值：0x0000 000X

该寄存器用于读取上一次（从上电或待机模式）进行启动时 BOOT 引脚的状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOOT2	BOOT1	BOOT0
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **BOOT2**：BOOT2 引脚值 (BOOT2 pin value)
0：（低电平）BOOT2 引脚连接到 V_{SS}
1：（高电平）BOOT2 引脚连接到 V_{DD}

位 1 **BOOT1**：BOOT1 引脚值 (BOOT1 pin value)
0：（低电平）BOOT1 引脚连接到 V_{SS}
1：（高电平）BOOT1 引脚连接到 V_{DD}

位 0 **BOOT0**：BOOT0 引脚值 (BOOT0 pin value)
0：（低电平）BOOT0 引脚连接到 V_{SS}
1：（高电平）BOOT0 引脚连接到 V_{DD}

12.6.2 SYSCFG 外设模式配置设置寄存器 (SYSCFG_PMCSETR)

SYSCFG peripheral mode configuration set register

偏移地址：0x004

复位值：0x0000 0000

此寄存器支持 Trustzone 感知。与安全外设相对应的位只能通过安全写入访问进行修改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETH2_SEL[2:0]			Res.	Res.	Res.	ETH2_REF_CLK_SEL	ETH2_CLK_SEL	ETH1_SEL[2:0]			Res.	Res.	Res.	ETH1_REF_CLK_SEL	ETH1_CLK_SEL
rs	rs	rs				rs	rs	rs	rs	rs				rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	ANASWVDD	EN_BOOSTER	Res.	Res.	Res.	I2C5_FMP	I2C4_FMP	I2C3_FMP	I2C2_FMP	I2C1_FMP
						rs	rs				rs	rs	rs	rs	rs

位 31:29 **ETH2_SEL[2:0]**：以太网 PHY 接口选择。

由软件置 1。写入“0”无影响。通过写入“1”可将各个位置 1。读取意义如下：

000：MII

001：RGMII

010：保留

011：保留

100：RMII

101：保留

110：保留

111：保留

注意：必须在 ETH 处于复位状态且在使能 ETH 时钟之前完成此配置

如果在 ETZPC 中将 ETH2 设置为安全，则此位字段将处于安全状态。

位 28:26 保留，必须保持复位值。

位 25 **ETH2_REF_CLK_SEL**：以太网参考时钟选择。

由软件置 1。

0：写入“0”不会产生任何影响，而读取“0”则表示接收时钟（MII/RGMII 模式）或参考时钟（RMII 模式）来自 ETH2_RX_CLK/ETH2_REF_CLK 引脚。

1：写入“1”会将此位置 1，而读取“1”则表示参考时钟（RMII 模式）来自 RCC

如果在 ETZPC 中将 ETH2 设置为安全，则此位将处于安全状态。

位 24 **ETH2_CLK_SEL**：千兆以太网 125 MHz 时钟选择。仅适用于 RGMII PHY。

由软件置 1。

0：写入“0”不会产生任何影响，而读取“0”则表示发送时钟来自 ETHx_CLK125 引脚。

1：写入“1”会将此位置 1，而读取“1”则表示发送时钟来自 RCC。

如果在 ETZPC 中将 ETH2 设置为安全，则此位将处于安全状态。

位 23:21 **ETH1_SEL[2:0]**：以太网 PHY 接口选择。
由软件置 1。写入“0”无影响。通过写入“1”可将各个位置 1。读取意义如下：
000：MII
001：RGMII
010：保留
011：保留
100：RMII
101：保留
110：保留
111：保留
注意：必须在 ETH 处于复位状态且在使能 ETH 时钟之前完成此配置
如果在 ETZPC 中将 ETH1 设置为安全，则此位字段将处于安全状态。

位 20:18 保留，必须保持复位值。

位 17 **ETH1_REF_CLK_SEL**：以太网参考时钟选择。
由软件置 1。
0：写入“0”不会产生任何影响，而读取“0”则表示接收时钟（MII/RGMII 模式）或参考时钟（RMII 模式）来自 ETH1_RX_CLK/ETH1_REF_CLK 引脚。
1：写入“1”会将此位置 1，而读取“1”则表示参考时钟（RMII 模式）来自 RCC
如果在 ETZPC 中将 ETH1 设置为安全，则此位将处于安全状态。

位 16 **ETH1_CLK_SEL**：千兆以太网 125 MHz 时钟选择。仅适用于 RGMII PHY。
由软件置 1。
0：写入“0”不会产生任何影响，而读取“0”则表示发送时钟来自 ETHx_CLK125 引脚。
1：写入“1”会将此位置 1，而读取“1”则表示发送时钟来自 RCC。
如果在 ETZPC 中将 ETH1 设置为安全，则此位将处于安全状态。

位 15:10 保留，必须保持复位值。

位 9 **ANASWVDD**：GPIO 模拟开关供电电压选择 (GPIO analog switch supply voltage selection)
由软件清零。
0：写入“0”不会产生任何影响，而读取“0”则表示模拟开关由 V_{DDA}（如果存在且 **EN_BOOSTER**=0）或是升压器输出 (**EN_BOOSTER**=1) 进行供电
1：写入“1”会将此位置 1，而读取“1”则表示模拟开关由 V_{DD} 进行供电。
此位始终处于安全状态。

位 8 **EN_BOOSTER**：GPIO 模拟开关供电电压升压器使能。
由软件置 1。
用于当 V_{DDA} 和 V_{DD} 电源低于 2.7 V 时，降低 I/O 模拟开关的总谐波失真。激活升压器可以保证 I/O 模拟开关的交流性能。激活后，模拟开关将在全电压范围内实现相同的性能。
0：写入“0”不会产生任何影响，而读取“0”则表示将禁止升压器
1：写入“1”会将此位置 1，而读取“1”则表示将使能升压器。
此位始终处于安全状态。

位 7:5 保留，必须保持复位值。

位 4 **I2C5_FMP**：超快速模式 (FM+) 使能 (Fast mode plus (FM+) enable)
0：写入“0”不会产生任何影响，而读取“0”则表示 I2C5 的使用频率可能高达 400 kHz
1：写入“1”会将此位置 1，而读取“1”则表示当选择相应的复用功能时，I2C5 SCL 和 SDA I/O 将处于超快速模式 (FM+)。I/O 低驱动将得到增强，以支持高达 1 MHz 的频率。
如果在 ETZPC 中将 I2C5 设置为安全，则此位将处于安全状态。

位 3 **I2C4_FMP**：超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) Enable)

0：写入“0”不会产生任何影响，而读取“0”则表示 I2C4 的使用频率可能高达 400 kHz

1：写入“1”会将此位置 1，而读取“1”则表示当选择相应的复用功能时，I2C4 SCL 和 SDA I/O 将处于超快速模式 (FM+)。I/O 低驱动将得到增强，以支持高达 1 MHz 的频率。

如果在 ETZPC 中将 I2C4 设置为安全，则此位将处于安全状态。

位 2 **I2C3_FMP**：超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) Enable)

0：写入“0”不会产生任何影响，而读取“0”则表示 I2C3 的使用频率可能高达 400 kHz

1：写入“1”会将此位置 1，而读取“1”则表示当选择相应的复用功能时，I2C3 SCL 和 SDA I/O 将处于超快速模式 (FM+)。I/O 低驱动将得到增强，以支持高达 1 MHz 的频率。

如果在 ETZPC 中将 I2C3 设置为安全，则此位将处于安全状态。

位 1 **I2C2_FMP**：超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) Enable)

0：写入“0”不会产生任何影响，而读取“0”则表示 I2C2 的使用频率可能高达 400 kHz

1：写入“1”会将此位置 1，而读取“1”则表示当选择相应的复用功能时，I2C2 SCL 和 SDA I/O 将处于超快速模式 (FM+)。I/O 低驱动将得到增强，以支持高达 1 MHz 的频率。

位 0 **I2C1_FMP**：超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) Enable)

0：写入“0”不会产生任何影响，而读取“0”则表示 I2C1 的使用频率可能高达 400 kHz

1：写入“1”会将此位置 1，而读取“1”则表示当选择相应的复用功能时，I2C1 SCL 和 SDA I/O 将处于超快速模式 (FM+)。I/O 低驱动将得到增强，以支持高达 1 MHz 的频率。

12.6.3 SYSCFG 外设模式配置清除寄存器 (SYSCFG_PMCCLRR)

SYSCFG peripheral mode configuration clear register

偏移地址：0x008

复位值：0x0000 0000

此寄存器支持 Trustzone 感知。与安全外设相对应的位只能通过安全写入访问进行修改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETH2_SEL[2:0]			Res.	Res.	Res.	ETH2_REF_CLK_SEL	ETH2_CLK_SEL	ETH1_SEL[2:0]			Res.	Res.	Res.	ETH1_REF_CLK_SEL	ETH1_CLK_SEL
rc_w1	rc_w1	rc_w1				rc_w1	rc_w1	rc_w1	rc_w1	rc_w1				rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	ANASWVDD	EN_BOOSTER	Res.	Res.	Res.	I2C5_FMP	I2C4_FMP	I2C3_FMP	I2C2_FMP	I2C1_FMP
						rc_w1	rc_w1				rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:29 **ETH2_SEL[2:0]**：以太网 PHY 接口选择。
由软件置 1。写入“0”无影响。通过写入“1”可将各个位清零。读取意义如下：
000：MII
001：RGMII
010：保留
011：保留
100：RMII
101：保留
110：保留
111：保留
注意：必须在 ETH 处于复位状态且在使能 ETH 时钟之前完成此配置
如果在 ETZPC 中将 ETH2 设置为安全，则此位字段将处于安全状态。

位 28:26 保留，必须保持复位值。

位 25 **ETH2_REF_CLK_SEL**：以太网参考时钟选择。
由软件置 1。
0：写入“0”不会产生任何影响，而读取“0”则表示接收时钟（MII/RGMII 模式）或参考时钟（RMII 模式）来自 ETH2_RX_CLK/ETH2_REF_CLK 引脚。
1：写入“1”会将此位清零，而读取“1”则表示参考时钟（RMII 模式）来自 RCC
如果在 ETZPC 中将 ETH2 设置为安全，则此位将处于安全状态。

位 24 **ETH2_CLK_SEL**：千兆以太网 125 MHz 时钟选择。仅适用于 RGMII PHY。
由软件置 1。
0：写入“0”不会产生任何影响，而读取“0”则表示发送时钟来自 ETHx_CLK125 引脚。
1：写入“1”会将该位清零，而读取“1”则表示发送时钟来自 RCC。
如果在 ETZPC 中将 ETH2 设置为安全，则此位将处于安全状态。

位 23:21 **ETH1_SEL[2:0]**：以太网 PHY 接口选择。
由软件置 1。写入“0”无影响。通过写入“1”可将各个位清零。读取意义如下：
000：MII
001：RGMII
010：保留
011：保留
100：RMII
101：保留
110：保留
111：保留
注意：必须在 ETH 处于复位状态且在使能 ETH 时钟之前完成此配置
如果在 ETZPC 中将 ETH1 设置为安全，则此位字段将处于安全状态。

位 20:18 保留，必须保持复位值。

位 17 **ETH1_REF_CLK_SEL**：以太网参考时钟选择。
由软件置 1。
0：写入“0”不会产生任何影响，而读取“0”则表示接收时钟（MII/RGMII 模式）或参考时钟（RMII 模式）来自 ETH1_RX_CLK/ETH1_REF_CLK 引脚。
1：写入“1”会将此位清零，而读取“1”则表示参考时钟（RMII 模式）来自 RCC
如果在 ETZPC 中将 ETH1 设置为安全，则此位将处于安全状态

位 16 **ETH1_CLK_SEL**：千兆以太网 125 MHz 时钟选择。仅适用于 RGMII PHY。
由软件置 1。
0：写入“0”不会产生任何影响，而读取“0”则表示发送时钟来自 ETHx_CLK125 引脚。
1：写入“1”会将该位清零，而读取“1”则表示发送时钟来自 RCC。
如果在 ETZPC 中将 ETH1 设置为安全，则此位将处于安全状态

位 15:10 保留，必须保持复位值。

位 9 **ANASWVDD**：GPIO 模拟开关供电电压选择 (GPIO analog switch supply voltage selection)
由软件清零。
0：写入“0”不会产生任何影响，而读取“0”则表示模拟开关由 V_{DDA}（如果存在且 **EN_BOOSTER**=0）或是升压器输出 (**EN_BOOSTER**=1) 进行供电
1：写入“1”会将此位清零，而读取“1”则表示模拟开关由 V_{DD} 进行供电。
此位始终处于安全状态。

位 8 **EN_BOOSTER**：GPIO 模拟开关供电电压升压器使能。
由软件清零。
用于当 V_{DDA} 和 V_{DD} 电源低于 2.7 V 时，降低 I/O 模拟开关的总谐波失真。激活升压器可以保证 I/O 模拟开关的交流性能。激活后，模拟开关将在全电压范围内实现相同的性能。
0：写入“0”不会产生任何影响，而读取“0”则表示将禁止升压器
1：写入“1”会将此位清零，而读取“1”则表示将使能升压器。
此位始终处于安全状态。

位 7:5 保留，必须保持复位值。

位 4 **I2C5_FMP**：超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) Enable)
0：写入“0”不会产生任何影响，而读取“0”则表示 I2C5 的使用频率可能高达 400 kHz
1：写入“1”会将此位清零，而读取“1”则表示当选择相应的复用功能时，I2C5 SCL 和 SDA I/O 将处于超快速模式 (FM+)。I/O 低驱动将得到增强，以支持高达 1 MHz 的频率。
如果在 ETZPC 中将 I2C5 设置为安全，则此位将处于安全状态。

位 3 **I2C4_FMP**：超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) Enable)
0：写入“0”不会产生任何影响，而读取“0”则表示 I2C4 的使用频率可能高达 400 kHz
1：写入“1”会将此位清零，而读取“1”则表示当选择相应的复用功能时，I2C4 SCL 和 SDA I/O 将处于超快速模式 (FM+)。I/O 低驱动将得到增强，以支持高达 1 MHz 的频率。
如果在 ETZPC 中将 I2C4 设置为安全，则此位将处于安全状态。

位 2 **I2C3_FMP**：超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) Enable)
0：写入“0”不会产生任何影响，而读取“0”则表示 I2C3 的使用频率可能高达 400 kHz
1：写入“1”会将此位清零，而读取“1”则表示当选择相应的复用功能时，I2C3 SCL 和 SDA I/O 将处于超快速模式 (FM+)。I/O 低驱动将得到增强，以支持高达 1 MHz 的频率。
如果在 ETZPC 中将 I2C3 设置为安全，则此位将处于安全状态。

位 1 **I2C2_FMP**：超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) Enable)
0：写入“0”不会产生任何影响，而读取“0”则表示 I2C2 的使用频率可能高达 400 kHz
1：写入“1”会将此位清零，而读取“1”则表示当选择相应的复用功能时，I2C2 SCL 和 SDA I/O 将处于超快速模式 (FM+)。I/O 低驱动将得到增强，以支持高达 1 MHz 的频率。

位 0 **I2C1_FMP**：超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) Enable)
0：写入“0”不会产生任何影响，而读取“0”则表示 I2C1 的使用频率可能高达 400 kHz
1：写入“1”会将此位清零，而读取“1”则表示当选择相应的复用功能时，I2C1 SCL 和 SDA I/O 将处于超快速模式 (FM+)。I/O 低驱动将得到增强，以支持高达 1 MHz 的频率。

12.6.4 SYSCFG 启动控制寄存器 (SYSCFG_BOOTCR)

SYSCFG boot control register

偏移地址：0x00C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BMEN
															rw

位 31:1 保留，必须保持复位值。

位 0 **BMEN**：启动模式使能 (Boot mode enable)

此位只能通过安全特权代码进行写入。用户应仅在安全启动期间使用此位，以确保及时执行负责 Tamper 管理的固件。

0：当 TAMP 外设引发潜在 Tamper 事件时，设备中会发生多个阻塞和擦除操作。在这些操作中，SRAM3 将读为零，且系统将忽略对 SRAM3 的写入。同时，系统还会在 PKA、SAES、HASH 和 CRYP 外设上触发复位脉冲。详细信息请参见 TAMP 部分。

1：当 TAMP 外设引发潜在 Tamper 事件时，系统将允许对 SRAM3 进行读写访问，且不会复位 PKA、SAES、HASH 和 CRYP 外设。TAMP 外设中定义的其他阻塞和擦除操作不会受到影响。

当潜在 Tamper 事件处于激活状态时，此位将从 1 清除为 0，且系统将自动阻止对 SRAM3 进行访问（如上文所定义），并将对 PKA、SAES、HASH 和 CRYP 外设进行复位。

当 TAMP 外设引发确认的（非潜在）Tamper 事件时，不论 BMEN 值如何，系统都将擦除 SRAM3（如 TAMP 中所定义）并对 PKA、SAES、HASH 和 CRYP 外设进行复位。

12.6.5 SYSCFG SRAM3 擦除寄存器 (SYSCFG_SRAM3ERASER)

SYSCFG SRAM3 erase register

偏移地址：0x010

复位值：0x0000 0000

该寄存器允许软件触发对 SRAM3 内存的擦除，或是验证是否有擦除正在进行。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAM3 EO	SRAM3 ER
														r	rw

位 31:2 保留，必须保持复位值。

位 1 **SRAM3EO**：SRAM3 当前正在执行擦除 (SRAM3 erase ongoing)

此位指示当前是否正在执行 SRAM3 的擦除，无论使用何种触发器（Tamper 或软件）都是如此。

0：当前未执行擦除

1：当前正在执行擦除

位 0 **SRAM3ER**：SRAM3 擦除请求 (SRAM3 erase request)

0：（写入）无操作；（读取）当前未执行擦除

1：（写入）开始擦除；（读取）当前正在执行软件触发的擦除

此位由软件置 1，并在擦除操作结束后由硬件自动复位。仅当先前已将正确的解锁序列写入 SYSCFG_SRAM3KR.KEY[7:0] 寄存器位字段时，方可将此位置 1。如果在 ETZPC 中将 SRAM3 设置为安全，则此位将处于安全状态。

12.6.6 SYSCFG SRAM3 擦除密钥寄存器 (SYSCFG_SRAM3KR)

SYSCFG SRAM3 erase key register

偏移地址：0x014

复位值：0x0000 0000

该寄存器将解锁面向 SRAM3 擦除寄存器 SYSCFG_SRAM3ERASER 的写入访问保护。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY[7:0]							
								w	w	w	w	w	w	w	w

位 31:8 保留，必须保持复位值。

位 7:0 **KEY[7:0]**：SRAM3 软件擦除密钥 (SRAM3 software erase key)

将以下序列写入 KEY[7:0] 会解锁 SRAM3 软件擦除功能：

0xCA

0x53

写入其他任意值将解锁该功能。

12.6.7 SYSCFG 主补偿单元控制寄存器 (SYSCFG_CMPCR)

SYSCFG main compensation cell control register

偏移地址：0x020

复位值：0xXX87 0000

此寄存器只能通过安全写入访问进行修改

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
APSRC[3:0]				ANSRC[3:0]				RAPSRC[3:0]				RANSRC[3:0]			
r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	READY	Res.	Res.	Res.	Res.	Res.	Res.	SW_CTRL	Res.
							r							rw	

位 31:28 **APSRC[3:0]**：由补偿单元提供的 PMOS I/O 补偿值 (PMOS I/O compensation value provided by compensation cell)

当 SW_CTRL = 0 且 READY = 1 时，发送至 I/O 补偿的值。

位 27:24 **ANSRC[3:0]**：由补偿单元提供的 NMOS I/O 补偿值 (NMOS I/O compensation value provided by compensation cell)

当 SW_CTRL = 0 且 READY = 1 时，发送至 I/O 的值。

位 23:20 **RAPSRC[3:0]**：SW_CTRL = 1 时发送至 I/O 的 PMOS I/O 补偿值 (PMOS I/O Compensation value sent to I/Os when SW_CTRL = 1)

0000：保留
0001：快速条件下的最大补偿
....
1000：典型条件下的补偿
....
1110：慢速条件下的最大补偿
1111：保留

注意：如果需要补偿，则建议使用自动补偿

位 19:16 **RANSRC[3:0]**：SW_CTRL = 1 时发送至 I/O 的 NMOS I/O 补偿值 (NMOS I/O compensation value sent to I/Os when SW_CTRL = 1)

0000：保留
0001：慢速条件下的最大补偿
....
0111：典型条件下的补偿
....
1110：快速条件下的最大补偿
1111：保留

注意：如果需要补偿，则建议使用自动补偿

位 15:9 保留，必须保持复位值。

位 8 **READY**：补偿单元就绪标志 (Compensation cell ready flag)

0：I/O 补偿单元未就绪
1：I/O 补偿单元就绪，APSRC[3:0] 和 ANSRC[3:0] 的值有效

位 7:2 保留，必须保持复位值。

位 1 **SW_CTRL**：补偿软件控制 (Compensation software control)

0：I/O 补偿值来自 ANSRC[3:0] 和 APSRC[3:0] 中的补偿值

1：I/O 补偿值来自 RANSRC[3:0] 和 RAPSRC[3:0] 寄存器值

注意：在 READY = 1 之前，不考虑 SW_CTRL = 0。
这意味着无论 SW_CTRL 值如何，在补偿单元处于掉电模式时（即复位后 EN = 0 的情况），RANSRC[3:0] 和 RAPSRC[3:0] 都将用于 I/O 补偿。

位 0 保留，必须保持复位值。

12.6.8 SYSCFG 主补偿单元使能设置寄存器 (SYSCFG_CMPENSETR)

SYSCFG main compensation cell enable set register

偏移地址：0x024

复位值：0x0000 0000

此寄存器只能通过安全写入访问进行修改

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
															rs

位 31:1 保留，必须保持复位值。

位 0 **EN**：补偿单元使能 (Compensation cell enable)

由软件置 1。

0：(power_down) 写入“0”不会产生任何影响，而读取“0”则表示 I/O 补偿单元处于掉电模式

1：（使能）写入“1”将使能 I/O 补偿单元，而读取“1”则表示已使能 I/O 补偿单元

注意：必须使能 CSI 振荡器并使之就绪（在 RCC 中进行控制），然后方可将 EN 置 1。同理，只有当 EN 置 0 时，方可禁止 CSI 振荡器。

12.6.9 SYSCFG 主补偿单元使能清除寄存器 (SYSCFG_CMPENCLRR)

SYSCFG main compensation cell enable clear register

偏移地址：0x028

复位值：0x0000 0000

此寄存器只能通过安全写入访问进行修改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
															rc_w1

位 31:1 保留，必须保持复位值。

位 0 **EN**：补偿单元使能 (Compensation cell enable)

由软件清零。

0：(power_down) 写入“0”不会产生任何影响，而读取“0”则表示 I/O 补偿单元处于掉电模式

1：（使能）写入“1”会将 EN 位清零，而读取“1”则表示已使能 I/O 补偿单元

注意：必须使能 CSI 振荡器并使之就绪（在 RCC 中进行控制），然后方可将 EN 置 1。同理，只有当 EN 置 0 时，方可禁止 CSI 振荡器。

12.6.10 SYSCFG 控制定时器刹车寄存器 (SYSCFG_CBR)

SYSCFG control timer break register

偏移地址：0x02C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PVDL	Res.	Res.
													rs		

位 31:3 保留，必须保持复位值。

位 2 **PVDL**：PVD 锁定使能位 (PVD lock enable bit)

该位由软件置 1，只能通过系统复位清零。它可用于使能并锁定 TIM1/8/15/16/17 刹车输入的 PVD 连接以及 PWR_CR1 寄存器中的 PVDE 和 PLS[2:0]。

0：PVD 中断与 TIM1/8/15/16/17 刹车输入断开连接。PVDE 和 PLS[2:0] 位可由应用程序设定。

1：PVD 中断连接至 TIM1/8/15/16/17 刹车输入。PVDE 和 PLS[2:0] 位是只读的。

位 1:0 保留，必须保持复位值。

12.6.11 SYSCFG V_{DDSD1} 补偿单元控制寄存器 (SYSCFG_CMPSD1CR)

SYSCFG V_{DDSD1} compensation cell control register

偏移地址：0x030

复位值：0xXX87 0000

此寄存器支持 Trustzone 感知。某些位只能通过安全写入访问进行修改

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
APSRC[3:0]				ANSRC[3:0]				RAPSRC[3:0]				RANSRC[3:0]			
r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	READY	Res.	Res.	Res.	Res.	Res.	Res.	SW_CTRL	Res.
							r							rw	

位 31:28 **APSRC[3:0]**：由补偿单元提供的 PMOS I/O 补偿值 (PMOS I/O Compensation value provided by compensation cell)

当 SW_CTRL = 0 且 READY = 1 时，发送至 I/O 补偿的值。

位 27:24 **ANSRC[3:0]**：由补偿单元提供的 NMOS I/O 补偿值 (NMOS I/O Compensation value provided by compensation cell)

当 SW_CTRL = 0 且 READY = 1 时，发送至 I/O 的值。

位 23:20 **RAPSRC[3:0]**：SW_CTRL = 1 时发送至 I/O 的 PMOS I/O 补偿值 (PMOS I/O compensation value sent to I/Os when SW_CTRL = 1)

0000：保留
0001：快速条件下的最大补偿
....
1000：典型条件下的补偿
....
1110：慢速条件下的最大补偿
1111：保留

注意：如果需要补偿，则建议使用自动补偿

如果在 ETZPC 中将 SDMMC1 设置为安全，则此位字段将处于安全状态。

位 19:16 **RANSRC[3:0]**：SW_CTRL = 1 时发送至 I/O 的 NMOS I/O 补偿值 (NMOS I/O compensation value sent to I/Os when SW_CTRL = 1)

0000：保留
0001：慢速条件下的最大补偿
....
0111：典型条件下的补偿
....
1110：快速条件下的最大补偿
1111：保留

注意：如果需要补偿，则建议使用自动补偿

如果在 ETZPC 中将 SDMMC1 设置为安全，则此位字段将处于安全状态。

位 15:9 保留，必须保持复位值。

位 8 **READY**：补偿单元就绪标志 (Compensation cell ready flag)

0：I/O 补偿单元未就绪。
1：I/O 补偿单元就绪，APSRC[3:0] 和 ANSRC[3:0] 的值有效

位 7:2 保留，必须保持复位值。

位 1 **SW_CTRL**：补偿软件控制 (Compensation software control)

0：I/O 补偿值来自 ANSRC[3:0] 和 APSRC[3:0] 中的补偿值

1：I/O 补偿值来自 RANSRC[3:0] 和 RAPSRC[3:0] 寄存器值

注意：在 READY = 1 之前，不考虑 SW_CTRL = 0。

这意味着无论 SW_CTRL 值如何，在补偿单元处于掉电模式时（即复位后 EN = 0 的情况），RANSRC[3:0] 和 RAPSRC[3:0] 都将用于 I/O 补偿。

如果在 ETZPC 中将 SDMMC1 设置为安全，则此位将处于安全状态。

位 0 保留，必须保持复位值。

12.6.12 SYSCFG V_{DDSD1} 补偿单元使能设置寄存器 (SYSCFG_CMPSD1ENSETR)

SYSCFG V_{DDSD1} compensation cell enable set register

偏移地址：0x034

复位值：0x0000 0000

此寄存器支持 Trustzone 感知。某些位只能通过安全写入访问进行修改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
															rs

位 31:1 保留，必须保持复位值。

位 0 **EN**：补偿单元使能 (Compensation cell enable)

由软件置 1。

0：(power_down) 写入“0”不会产生任何影响，而读取“0”则表示 I/O 补偿单元处于掉电模式

1：（使能）写入“1”将使能 I/O 补偿单元，而读取“1”则表示已使能 I/O 补偿单元

注意：必须使能 CSI 振荡器并使之就绪（在 RCC 中进行控制），然后方可将 EN 置 1。同理，只有当 MPU_EN 置 0 时，方可禁止 CSI 振荡器。

如果在 ETZPC 中将 SDMMC1 设置为安全，则此位将处于安全状态。

12.6.13 SYSCFG V_{DDSD1} 补偿单元使能清除寄存器 (SYSCFG_CMPSD1ENCLRR)

SYSCFG V_{DDSD1} compensation cell enable clear register

偏移地址：0x038

复位值：0x0000 0000

此寄存器支持 Trustzone 感知。某些位只能通过安全写入访问进行修改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
															rc_w1

位 31:1 保留，必须保持复位值。

位 0 **EN**：补偿单元使能 (Compensation cell enable)

由软件清零。

0：(power_down) 写入“0”不会产生任何影响，而读取“0”则表示 I/O 补偿单元处于掉电模式

1：（使能）写入“1”会将 EN 位清零，而读取“1”则表示已使能 I/O 补偿单元

注意：必须使能 CSI 振荡器并使之就绪（在 RCC 中进行控制），然后方可将 EN 置 1。同理，只有当 EN 置 0 时，方可禁止 CSI 振荡器。

如果在 ETZPC 中将 SDMMC1 设置为安全，则此位将处于安全状态。

12.6.14 SYSCFG V_{DDSD2} 补偿单元控制寄存器 (SYSCFG_CMPSD2CR)

SYSCFG V_{DDSD2} compensation cell control register

偏移地址：0x040

复位值：0xXX87 0000

此寄存器支持 Trustzone 感知。某些位只能通过安全写入访问进行修改

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
APSRC[3:0]				ANSRC[3:0]				RAPSRC[3:0]				RANSRC[3:0]			
r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	READY	Res.	Res.	Res.	Res.	Res.	Res.	SW_CTRL	Res.
							r							rw	

位 31:28 **APSRC[3:0]**：由补偿单元提供的 PMOS I/O 补偿值 (PMOS I/O Compensation value provided by compensation cell)

当 SW_CTRL = 0 且 READY = 1 时，发送至 I/O 补偿的值

位 27:24 **ANSRC[3:0]**：由补偿单元提供的 NMOS I/O 补偿值 (NMOS I/O Compensation value provided by compensation cell)

当 SW_CTRL = 0 且 READY = 1 时，发送至 I/O 的值

位 23:20 **RAPSRC[3:0]**：SW_CTRL = 1 时发送至 I/O 的 PMOS I/O 补偿值 (PMOS I/O compensation value sent to I/Os when SW_CTRL = 1)

0000：保留
0001：快速条件下的最大补偿
....
1000：典型条件下的补偿
....
1110：慢速条件下的最大补偿
1111：保留

注意：如果需要补偿，则建议使用自动补偿

如果在 ETZPC 中将 SDMMC2 设置为安全，则此位字段将处于安全状态。

位 19:16 **RANSRC[3:0]**：SW_CTRL = 1 时发送至 I/O 的 NMOS I/O 补偿值 (NMOS I/O compensation value sent to I/Os when SW_CTRL = 1)

0000：保留
0001：慢速条件下的最大补偿
....
0111：典型条件下的补偿
....
1110：快速条件下的最大补偿
1111：保留

注意：如果需要补偿，则建议使用自动补偿

如果在 ETZPC 中将 SDMMC2 设置为安全，则此位字段将处于安全状态。

位 15:9 保留，必须保持复位值。

位 8 **READY**：补偿单元就绪标志 (Compensation cell ready flag)

0：I/O 补偿单元未就绪。
1：I/O 补偿单元就绪，APSRC[3:0] 和 ANSRC[3:0] 的值有效

位 7:2 保留，必须保持复位值。

位 1 **SW_CTRL**：补偿软件控制 (Compensation Software Control)

0：I/O 补偿值来自 ANSRC[3:0] 和 APSRC[3:0] 中的补偿值
1：I/O 补偿值来自 RANSRC[3:0] 和 RAPSRC[3:0] 寄存器值

注意：在 READY = 1 之前，不考虑 SW_CTRL = 0。
这意味着无论 SW_CTRL 值如何，在补偿单元处于掉电模式时（即复位后 EN = 0 的情况），RANSRC[3:0] 和 RAPSRC[3:0] 都将用于 I/O 补偿。

如果在 ETZPC 中将 SDMMC2 设置为安全，则此位将处于安全状态

位 0 保留，必须保持复位值。

12.6.15 SYSCFG V_{DDSD2} 补偿单元使能设置寄存器 (SYSCFG_CMPSD2ENSETR)

SYSCFG V_{DDSD2} compensation cell enable set register

偏移地址：0x044

复位值：0x0000 0000

此寄存器支持 Trustzone 感知。某些位只能通过安全写入访问进行修改

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
															rs

位 31:1 保留，必须保持复位值。

位 0 **EN**：补偿单元使能 (Compensation cell enable)

由软件置 1。

0：(power_down) 写入“0”不会产生任何影响，而读取“0”则表示 I/O 补偿单元处于掉电模式

1：（使能）写入“1”将使能 I/O 补偿单元，而读取“1”则表示已使能 I/O 补偿单元

注意：必须使能 CSI 振荡器并使之就绪（在 RCC 中进行控制），然后方可将 EN 置 1。同理，只有当 EN 置 0 时，方可禁止 CSI 振荡器。

如果在 ETZPC 中将 SDMMC2 设置为安全，则此位将处于安全状态。

12.6.16 SYSCFG V_{DDSD2} 补偿单元使能清除寄存器 (SYSCFG_CMPSD2ENCLRR)

SYSCFG V_{DDSD2} compensation cell enable clear register

偏移地址：0x048

复位值：0x0000 0000

此寄存器支持 Trustzone 感知。某些位只能通过安全写入访问进行修改

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
															rc_w1

位 31:1 保留，必须保持复位值。

位 0 **EN**：补偿单元使能 (Compensation cell enable)

由软件清零。

0：(power_down) 写入“0”不会产生任何影响，而读取“0”则表示 I/O 补偿单元处于掉电模式

1：（使能）写入“1”会将 MPU_EN 位清零，而读取“1”则表示已使能 I/O 补偿单元

注意：必须使能 CSI 振荡器并使之就绪（在 RCC 中进行控制），然后方可将 EN 置 1。同理，只有当 EN 置 0 时，方可禁止 CSI 振荡器。

如果在 ETZPC 中将 SDMMC2 设置为安全，则此位将处于安全状态。

12.6.17 SYSCFG 高速低压使能寄存器 x (SYSCFG_HSLVENxR)

SYSCFG high-speed low-voltage enable register x

偏移地址：0x050 + 0x4 * x（x = 0 到 11）

复位值：0x0000 0000

这些寄存器支持 Trustzone 感知。与安全外设相对应的寄存器只能通过安全写入访问进行修改。

有关与寄存器索引 x 相关的接口，请参见*表 79*。

警告： **当 V_{DD} > 2.7 V 时，使能高速模式可能会损坏 IC。有关详细信息，请参见 I/O 特性。**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSLVEN[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	rw

位 31:16 保留，必须保持复位值。

位 15:0 **HSLVEN[15:0]**：高速低电压 I/O 模式使能。
控制在 AFMUX 中选择某接口时，与该接口相关的 _h 和 _vh 类型 I/O 的速度
写入时：
0x1018：使能高速模式
其他值：禁止高速模式
读取时：
0x0001：已使能高速模式
0x0000：已禁止高速模式

12.6.18 SYSCFG 标识代码寄存器 (SYSCFG_IDC)

SYSCFG identity code register

偏移地址：0x380

复位值：0xXXXX 6501

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REV_ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DEV_ID[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 **REV_ID[15:0]**：设备版本 (device revision)

位 15:12 保留，必须保持复位值。

位 11:0 **DEV_ID[11:0]**：设备 ID (device ID)
0x501：STM32MP13xx

12.6.19 SYSCFG 版本寄存器 (SYSCFG_VERR)

SYSCFG version register

偏移地址：0x3F4

复位值：0x0000 0030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本 (Major revision)
这些位返回 SYSCFG 主版本。
主版本为 3。

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)
这些位返回 SYSCFG 次版本。
次版本为 0。

12.6.20 SYSCFG 标识寄存器 (SYSCFG_IPIDR)

SYSCFG identification register

偏移地址：0x3F8

复位值：0x0003 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：SYSCFG 标识符 (SYSCFG identifier)
这些位返回 SYSCFG 标识符值。

12.6.21 SYSCFG 大小标识寄存器 (SYSCFG_SIDR)

SYSCFG size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识 (Size identification)

这些位返回分配给 SPDIFRX 寄存器的存储区域的大小。

12.6.22 SYSCFG 接口寄存器映射

表 80 给出了 SYSCFG 接口寄存器映射和复位值。

表 80. SYSCFG 接口寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**SYSCFG_BOOTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOOT2	BOOT1	BOOT0
	Reset value																														x	x	x
0x004	**SYSCFG_ PMCSETR**	ETH2_SEL[2:0]			Res.	Res.	Res.	ETH2_REF_CLK_SEL	ETH2_CLK_SEL	ETH1_SEL[2:0]			Res.	Res.	Res.	ETH1_REF_CLK_SEL	ETH1_CLK_SEL	Res.	Res.	Res.	Res.	Res.	Res.	ANASWVDD	EN_BOOSTER	Res.	Res.	Res.	I2C5_FMP	I2C4_FMP	I2C3_FMP	I2C2_FMP	I2C1_FMP
	Reset value	0	0	0				0	0	0	0	0				0	0							0	0			0	0	0	0	0	0
0x008	**SYSCFG_ PMCCLRR**	ETH2_SEL[2:0]			Res.	Res.	Res.	ETH2_REF_CLK_SEL	ETH2_CLK_SEL	ETH1_SEL[2:0]			Res.	Res.	Res.	ETH1_REF_CLK_SEL	ETH1_CLK_SEL	Res.	Res.	Res.	Res.	Res.	Res.	ANASWDD	EN_BOOSTER	Res.	Res.	I2C6_FMP	I2C5_FMP	I2C4_FMP	I2C3_FMP	I2C2_FMP	I2C1_FMP
	Reset value	0	0	0				0	0	0	0	0				0	0							0	0			0	0	0	0	0	0
0x00C	**SYSCFG_ BOOTCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BMEN
	Reset value																																0
0x010	**SYSCFG_ SRAM3ERASER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAM3EO	SRAM3ER
	Reset value																															0	0
0x014	**SYSCFG_ SRAM3KR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x018-0x01C	Reserved	Res.																															
0x020	**SYSCFG_CMPCR**	APSRC[3:0]				ANSRC[3:0]				RAPSRC[3:0]				RANSRC[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	READY	Res.	Res.	Res.	Res.	Res.	Res.	SW_CTRL	Res.
	Reset value	x	x	x	x	x	x	x	x	1	0	0	0	0	1	1	1								0							0	

表 80. SYSCFG 接口寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x024	**SYSCFG_ CMPENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
	Reset value																																0
0x028	**SYSCFG_ CMPENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
	Reset value																																0
0x02C	**SYSCFG_CBR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PVDL	Res.	Res.
	Reset value																														0		
0x030	**SYSCFG_ CMPSD1CR**	APSRC [3:0]				ANSRC [3:0]				RAPSRC [3:0]				RANSRC [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	READY	Res.	Res.	Res.	Res.	Res.	Res.	SW_CTRL	Res.
	Reset value	-	-	-	-	-	-	-	-	1	0	0	0	0	1	1	1								0							0	
0x034	**SYSCFG_ CMPSD1ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
	Reset value																																0
0x038	**SYSCFG_ CMPSD1ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
	Reset value																																0
0x03C	Reserved	Res.																															
0x040	**SYSCFG_ CMPSD2CR**	APSRC [3:0]				ANSRC [3:0]				RAPSRC [3:0]				RANSRC [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	READY	Res.	Res.	Res.	Res.	Res.	Res.	SW_CTRL	Res.
	Reset value	-	-	-	-	-	-	-	-	1	0	0	0	0	1	1	1								0							0	
0x044	**SYSCFG_ CMPSD2ENSETR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
	Reset value																																0
0x048	**SYSCFG_ CMPSD2ENCLRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
	Reset value																																0
0x04C	Reserved	Res.																															
0x050	**SYSCFG_ HSLVEN0R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSLVEN[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x054	**SYSCFG_ HSLVEN1R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSLVEN[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 80. SYSCFG 接口寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x058	**SYSCFG_ HSLVEN2R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSLVEN[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x05C	**SYSCFG_ HSLVEN3R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSLVEN[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x060	**SYSCFG_ HSLVEN4R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSLVEN[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x064	**SYSCFG_ HSLVEN5R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSLVEN[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x068	**SYSCFG_ HSLVEN6R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSLVEN[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x06C	**SYSCFG_ HSLVEN7R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSLVEN[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x070	**SYSCFG_ HSLVEN8R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSLVEN[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x074	**SYSCFG_ HSLVEN9R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSLVEN[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x078	**SYSCFG_ HSLVEN10R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSLVEN[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07C	**SYSCFG_ HSLVEN11R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSLVEN[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x080 - 0x37C	Reserved	Res.																															
0x380	**SYSCFG_IDC**	REV_ID[15:0]																Res.	Res.	Res.	Res.	DEV_ID[11:0]											
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x					0	1	0	1	0	0	0	0	0	0	0	1
0x384 - 0x3F0	Reserved	Res.																															

表 80. SYSCFG 接口寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x3F4	**SYSCFG_VERR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJ REV[3:0]				MIN REV[3:0]			
	Reset value																									0	0	1	1	0	0	0	0
0x3F8	**SYSCFG_IPIDR**	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x3FC	**SYSCFG_SIDR**	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

13 扩展 TrustZone® 保护控制器 (ETZPC)

13.1 ETZPC 简介

ETZPC 可在 SoC 中配置 TrustZone 安全性；该 SoC 应配有具备可编程安全属性（受控安全资源）的总线主设备和从设备，例如：

- 具有可编程安全区域大小的片上 RAM/ROM
- 需设为安全的 AHB 和 APB 外设

13.2 ETZPC 特性

- 32 位 APB4 接口
- ETZPC 始终支持安全写入
- 用于控制以下 SoC 安全设置的寄存器组：
  - 受控安全片上 RAM 和 ROM 安全区域大小
  - 受控安全 AHB 和 APB 外设的访问权限
- 锁定每个存储区域和每个外设的安全配置
- 支持以 4 KB 为步长配置存储区域大小。

13.3 扩展 TrustZone 架构和 ETZPC

- Arm TrustZone 模型支持实现以下两者间的隔离：
  - 安全环境，可用于运行安全敏感型应用程序，并存放关键资源；AXI 总线上的安全事务将通过 AxPROT[1] = 0 的信号进行表示
  - 非安全或公共环境（Linux 用户空间）；AXI 总线上的非安全事务将通过 AxPROT[1] = 1 的信号进行表示。
  - STM32MP13xx TrustZone 架构在 AXI 和 Cortex-A 的基础上进行了扩展，可通过 AXI/AHB 边带信号将安全/非安全事务属性传播到 AHB 和 APB 代理。
- AHB 和 APB 外设分类如下：
  - **安全**：
    此类外设由 AHB/APB 防火墙存根提供保护；该存根用于拒绝一切非安全访问。
    非安全写入操作将被忽略，并导致总线错误。
    系统不会将非安全读取操作传达到外设，且总线将返回零并导致总线错误。
  - **受控安全**：
    此类外设由 ETZPC 控制的 AHB/APB 防火墙提供保护，并由 ETZPC 来定义其安全模式。
    当外设被 ETZPC 配置为安全时：非安全写入操作将被忽略，并导致总线错误；系统不会将非安全读取操作传达到外设，且总线将返回零并导致总线错误。只有安全读写操作可以传达到外设。
    当外设被 ETZPC 配置为非安全时，安全和非安全的读写操作均可传达到外设。
  - **非安全**：
    此类外设直接连接到 AHB/APB，且安全和非安全的读写操作均可传达到外设。
  - **TrustZone 感知**：
    此类外设直接连接到 AHB 或 APB 总线，并会实现特定的 TrustZone 行为，例如根据外设状态或其编程接口的情况，将某个寄存器子集设为安全状态。

注： *总线错误取决于互连时钟情况（例如，非法访问无时钟外设不会生成总线错误）。*

- AXI 总线主设备将根据其安全状态来驱动 AxPROT[1] 信号；AXI 总线主设备可为：
  a) 非安全设备（这种情况下 AxPROT[1]=1）
  b) 支持 TrustZone 的设备（Cortex-A7、MDMA）；这种情况下当总线主设备处于安全状态时，AxPROT[1] 将驱动为低电平。
- 除了 DMA3（始终安全）之外，AHB 主设备都将是非安全设备。
  预计系统会在安全启动期间对 ETZPC 进行编程，且稍后不会进行更改：
  – TZMA 安全区域大小只能由安全软件进行更改。

注： *请先将 RAM 清零，然后再通过对 TZMA 进行编程来缩减安全区域大小，从而将 RAM 区域释放至非安全环境。*

13.4 ETZPC 功能描述

ETZPC 包含了一组寄存器，可通过以下位来控制受控安全 IP：

- **r0size[]** 位，用于定义每个 TZMA 安全区域的大小。
  decprot[1:0] 位，用于控制每个 AHB/APB 受控安全外设的访问权限，具体如下：
  – decprot[1:0] = 00：该外设为安全外设；只允许在安全环境进行读/写访问。
  – decprot[1:0] = 01：该外设支持安全写入；安全环境和非安全环境均支持进行读取访问，但只允许在安全环境进行写入访问
  – decprot[1:0] = 10：保留 - 不使用
  – decprot[1:0] = 11：外设为非安全外设；安全环境和非安全环境均支持进行读写访问。

默认情况下，复位后，存储区域将处于安全状态，且所有外设均支持安全读/写。

r0size[] 和 **decprot[]** 支持单独锁定，以防止在下次复位之前安全配置发生更改。

13.5 STM32MP13xx 安全架构

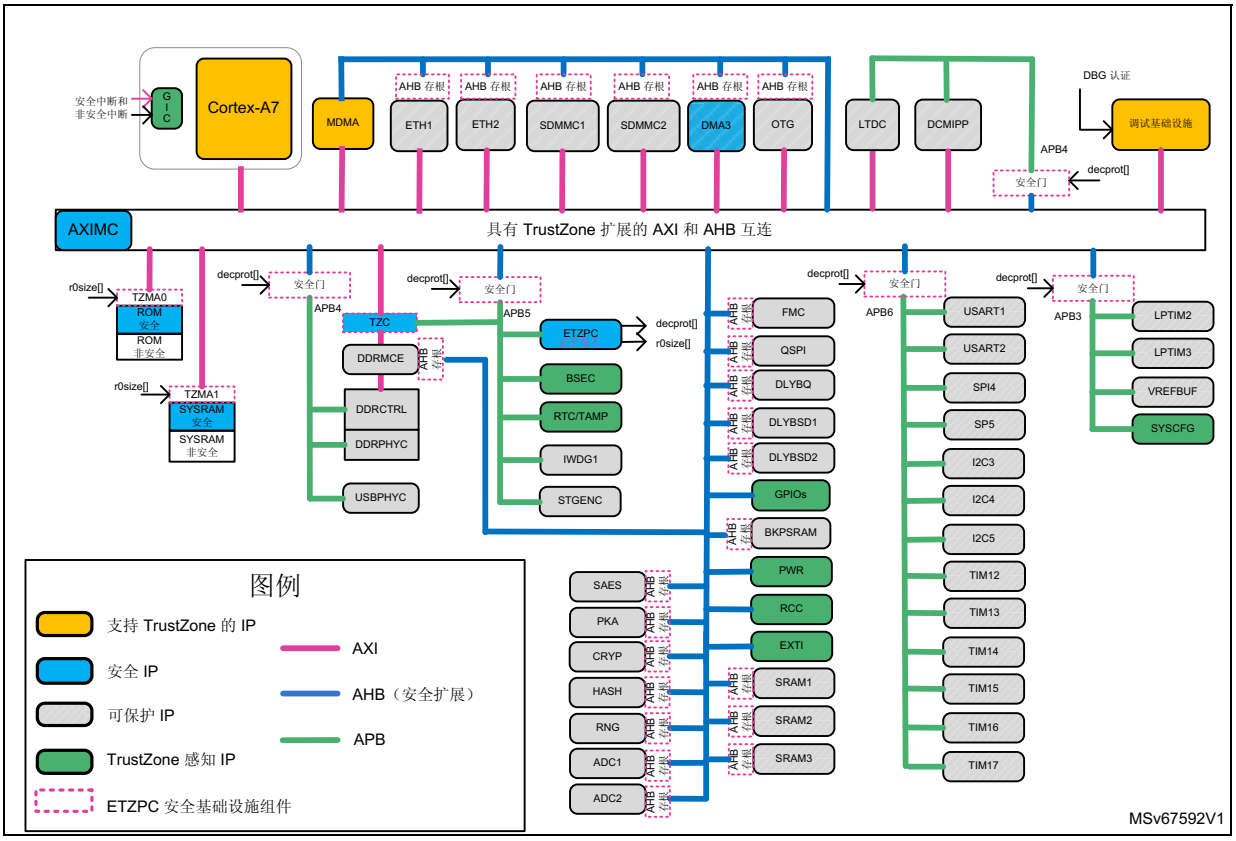
可对 IP 安全属性进行如下分类：

- 支持 TrustZone 的总线主设备 IP，如*表 84* 中所列。
- TrustZone 感知外设，其行为取决于事务安全性，如*表 82* 中所列。
- TrustZone 受控安全外设；此类外设依赖于某些 AHB 存根或 AHB2APB 总线桥，可根据 decprot[] 位和事务安全性来控制 IP 访问。此类外设已在*表 81* 中列出，其类型为 P。
- TrustZone 安全（始终安全）外设，如*表 81* 中所列，其类型为 S 或 WS（安全写入）。

注： *调试访问端口 (DAP) 的安全属性可通过 DBG 认证接口进行设置。*

STM32MP13xx 安全架构如*图 110* 中所示。

图 110. STM32MP13xx 安全架构



STM32MP13xx 安全架构基于 Arm TrustZone 的扩展版本，其中包括：

- 面向 AHB 的 TrustZone 扩展
- 用于保护受控安全 AHB 外设的 AHB 存根
- AHB2APB 总线桥，可使用 PPROT[1] 将信号传播到*表 82* 中列出的 TrustZone 感知 IP
- AHB2APB 总线桥，用于保护 中列出的受控安全 APB 外设 *表 81*
- 面向 ROM 和 SYSRAM 的 TZMA 防火墙，如*表 83* 中所示

表 81. STM32MP13xx 安全和受控安全外设

IP	类型(1)	总线	备注
VREFBUF	P	APB	电压参考缓冲器
LPTIM2/3	P	APB	低功耗定时器
LTDC	P	APB	LCD 显示控制端口
DCMIPP	P	APB	相机接口控制端口
USBPHYC	P	APB	USB PHY 控制端口
DDRCTRL	P	APB	DDR 控制器控制端口
DDRPHYC	P	APB	DDR PHY 控制端口
DDRMCE	P	APB	内存加密引擎
STGENC	P	APB	系统定时器生成器 (STGEN) 控制端口

表 81. STM32MP13xx 安全和受控安全外设（续）

IP	类型(1)	总线	备注
IWDG1	P	APB	独立看门狗
USART1/2	P	APB	通用同步/异步收发器
SPI4/5	P	APB	串行外设接口
I2C3/4/5	P	APB	内部 IC 接口
TIM12/13/14/15/16/17	P	APB	通用定时器
ADC1/2	P	AHB	模数转换器
OTG	P	AHB	USB OTG 控制器控制端口
RNG	P	AHB	随机数发生器
HASH	P	AHB	哈希处理器
CRYP	P	AHB	加密处理器
SAES	P	AHB	安全 AES 加速器
PKA	P	AHB	公钥加密加速器
BKPSRAM	P	AHB	Backup RAM (8 KB)
ETH1/2	P	AHB	以太网 MAC 控制端口
SDMMC1/2	P	AHB	SD/MMC 接口控制端口（包括延迟模块）
FMC	P	AHB	可变存储控制器控制端口
QSPI	P	AHB	四线 SPI 控制端口（包括延迟模块）
SRAM1/2/3	P	AHB	SRAM
TZC	S	APB	TrustZone 地址空间控制器
ETZPC	WS	APB	TrustZone 保护控制器（此 IP）
DMA3/DMAMUX2	S	AHB	安全 DMA 控制器和复用器

1. P：可编程安全，S：始终安全，WS：安全写入，非安全读取

表 82. STM32MP13xx TrustZone 感知外设

IP	总线	备注
BSEC	APB5	BSEC 包括了一个安全寄存器的子集
MDMA	AHB	MDMA 可为每个通道提供安全支持，且其从端口支持 TrustZone 感知，可禁止针对安全通道的非安全访问
GPIO	AHB	所有 GPIO 端口均支持 TrustZone 感知，并可提供引脚电平安全支持
RCC	AHB	RCC 包含支持安全写入的感应寄存器
PWR	AHB	PWR 包含支持安全写入的感应寄存器
EXTI	AHB	EXTI 包含支持安全写入的感应寄存器
RTC 和 Tamper	APB	RTC 包括了一个安全寄存器的子集
SYSCFG	APB	SYSCFG 包含某些支持安全写入的寄存器。多个寄存器与受控安全外设相关联，并且如果将外设 DECPROT 位编程为安全状态，则这些寄存器将支持安全写入。

有关 IP 安全功能的更多信息，请参见相应的 IP 章节。

表 83. STM32MP13xx 内存防火墙

片上内存	IP	备注
ROM	TZMA0	可利用 4 KB 边界将 ROM 划分为多个安全/非安全区域，其中低位区域为安全区域，其大小为 0 K、4 K、8 K 等等——最大可达 128 KB。 系统复位后，所有 ROM 均会处于安全状态。
SYSRAM	TZMA1	可利用 4 KB 边界将 SYSRAM 划分为多个安全/非安全区域，其中低位区域为安全区域，其大小为 0 K、4 K、8 K 等等——最大可达 128 KB。 系统复位后，所有 SYSRAM 均会处于安全状态。

表 84. STM32MP13xx 支持 TrustZone 的总线主设备

IP	I/F	备注
Cortex-A7	AXI	Cortex-A7 支持 TrustZone。
MDMA	AXI	可将 MDMA 通道设置为安全或非安全状态。
DAP AXI-AP	AXI	根据由 BSEC 控制的调试认证接口（spiden、spniden），已将 AXI-AP 设置为支持 TrustZone 感知。
ETH1/2	AXI	如果将以太网 MAC 的 DECPROT 位设置为“安全”，则系统会将其用作安全主设备。
LTDC 层 2	AXI	如果将 LTDC 层 2 的 DECPROT 位设置为“安全”，则系统会将其用作安全主设备。
DMA3	AHB	DMA3 是安全主设备。（请求方必须安全。）

表 85. DECPROT 分配

索引	decprot 位	IP	总线	默认	总线主设备	类型
0	DECPROT0[1:0]	VREFBUF	APB3	0b00	否	受控安全
1	DECPROT0[3:2]	LPTIM2	APB3	0b00	否	受控安全
2	DECPROT0[5:4]	LPTIM3	APB3	0b00	否	受控安全
3	DECPROT0[7:6]	LTDC 层 2	APB4	0b00	是	受控安全
4	DECPROT0[9:8]	DCMIPP	APB4	0b00	是	受控安全
5	DECPROT0[11:10]	USBPHYC	APB4	0b00	否	受控安全
6	DECPROT0[13:12]	DDRCTRL 和 DDRPHYC	APB4	0b00	否	受控安全
7	DECPROT0[15:14]	-	-	0b00	-	保留
8	DECPROT0[17:16]	-	-	0b00	-	保留
9	DECPROT0[19:18]	-	-	0b00	-	保留
10	DECPROT0[21:20]	-	-	0b00	-	保留
11	DECPROT0[23:22]	-	-	0b00	-	保留
12	DECPROT0[25:24]	IWDG1	APB5	0b00	否	受控安全
13	DECPROT0[27:26]	STGENC	APB5	0b00	否	受控安全
14	DECPROT0[29:28]	-	-	0b00	-	保留

表 85. DECPROT 分配（续）

索引	decprot 位	IP	总线	默认	总线主设备	类型
15	DECPROT0[31:30]	-	-	0b00	-	保留
16	DECPROT1[1:0]	USART1	APB6	0b00	否	受控安全
17	DECPROT1[3:2]	USART2	APB6	0b00	否	受控安全
18	DECPROT1[5:4]	SPI4	APB6	0b00	否	受控安全
19	DECPROT1[7:6]	SPI5	APB6	0b00	否	受控安全
20	DECPROT1[9:8]	I2C3	APB6	0b00	否	受控安全
21	DECPROT1[11:10]	I2C4	APB6	0b00	否	受控安全
22	DECPROT1[13:12]	I2C5	APB6	0b00	否	受控安全
23	DECPROT1[15:14]	TIM12	APB6	0b00	否	受控安全
24	DECPROT1[17:16]	TIM13	APB6	0b00	否	受控安全
25	DECPROT1[19:18]	TIM14	APB6	0b00	否	受控安全
26	DECPROT1[21:20]	TIM15	APB6	0b00	否	受控安全
27	DECPROT1[23:22]	TIM16	APB6	0b00	否	受控安全
28	DECPROT1[25:24]	TIM17	APB6	0b00	否	受控安全
29	DECPROT1[27:26]	-	-	0b00	-	保留
30	DECPROT1[29:28]	-	-	0b00	-	保留
31	DECPROT1[31:30]	-	-	0b00	-	保留
32	DECPROT2[1:0]	ADC1	AHB2	0b00	否	受控安全
33	DECPROT2[3:2]	ADC2	AHB2	0b00	否	受控安全
34	DECPROT2[5:4]	OTG	AHB2	0b00	是	受控安全
35	DECPROT2[7:6]	-	-	0b00	-	保留
36	DECPROT2[9:8]	-	-	0b00	-	保留
37	DECPROT2[11:10]	-	-	0b00	-	保留
38	DECPROT2[13:12]	-	-	0b00	-	保留
39	DECPROT2[15:14]	-	-	0b00	-	保留
40	DECPROT2[17:16]	RNG	AHB5	0b00	否	受控安全
41	DECPROT2[19:18]	HASH	AHB5	0b00	否	受控安全
42	DECPROT2[21:20]	CRYP	AHB5	0b00	否	受控安全
43	DECPROT2[23:22]	SAES	AHB5	0b00	否	受控安全
44	DECPROT2[25:24]	PKA	AHB5	0b00	否	受控安全
45	DECPROT2[27:26]	BKPSRAM	AHB5	0b00	否	受控安全
46	DECPROT2[29:28]	-	-	0b00	-	保留
47	DECPROT2[31:30]	-	-	0b00	-	保留
48	DECPROT3[1:0]	ETH1	AHB6	0b00	是	受控安全
49	DECPROT3[3:2]	ETH2	AHB6	0b00	是	受控安全

表 85. DECPROT 分配（续）

索引	decprot 位	IP	总线	默认	总线主设备	类型
50	DECPROT3[5:4]	SDMMC1 和 DLBSD1	AHB6	0b00	是	受控安全
51	DECPROT3[7:6]	SDMMC2 和 DLBSD2	AHB6	0b00	是	受控安全
52	DECPROT3[9:8]	-	-	0b00	-	保留
53	DECPROT3[11:10]	DDRMCE	AHB6	0b00	否	受控安全
54	DECPROT3[13:12]	FMC	AHB6	0b00	否	受控安全
55	DECPROT3[15:14]	QSPI 和 DLBQ	AHB6	0b00	否	受控安全
56	DECPROT3[17:16]	-	-	0b00	-	保留
57	DECPROT3[19:18]	-	-	0b00	-	保留
58	DECPROT3[21:20]	-	-	0b00	-	保留
59	DECPROT3[23:22]	-	-	0b00	-	保留
60	DECPROT3[25:24]	SRAM1	MLAHB	0b00	否	受控安全
61	DECPROT3[27:26]	SRAM2	MLAHB	0b00	否	受控安全
62	DECPROT3[29:28]	SRAM3	MLAHB	0b00	否	受控安全
63	DECPROT3[31:30]	-	-	0b00	-	保留

13.6 ETZPC 寄存器

ETZPC 寄存器只能按字（32 位）进行访问。

ETZPC 寄存器支持安全写入（即安全域和非安全域均支持进行读取访问，但只允许在安全域进行写入访问）。

ETZPC_DECPROTx 寄存器用于根据索引（从 0 到 63）控制*表 85* 中列出的所有外设的安全性和资源隔离。

小心： 用户需在早期启动期间对 ETZPC 进行编程，之后则不应进行更改；可通过锁定位来强制实现这一点；ETZPC decprot[] 位的传播会造成一定延迟，因此必须先观察两个 APB 周期，而后才能应用安全规则。

13.6.1 ETZPC ROM 安全大小定义 (ETZPC_TZMA0_SIZE)

ETZPC ROM secure size definition

偏移地址：0x000

复位值：0x0000 03FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	R0SIZE[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **LOCK**：用于锁定 TZMA 的 r0size 设置

0：r0size[] 可写入

1：下次复位之前 r0size[] 无法修改，因此针对 r0size 的写入操作将被静默忽略（例如不会引起 APB SLVERR）。

位 30:10 保留，必须保持复位值。

位 9:0 **R0SIZE[9:0]**

定义安全 ROM 的较低值大小和非安全 ROM 的较高值大小：

0x000：所有 ROM 均为非安全

0x001：4 KB ROM 安全

0x002：8 KB ROM 安全

....

0x020 到 0x3FF：128 KB ROM 安全（例如所有 ROM 均为安全）

13.6.2 ETZPC RAM 安全大小定义 (ETZPC_TZMA1_SIZE)

ETZPC RAM secure size definition

偏移地址：0x004

复位值：0x0000 03FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	R0SIZE[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **LOCK**：用于锁定 TZMA 的 r0size 设置

0：r0size[] 可写入

1：下次复位之前无法修改 r0size[]，因此针对 r0size 的写入操作将被静默忽略（例如 APB SLVERR）

位 30:10 保留，必须保持复位值。

位 9:0 **R0SIZE[9:0]**：定义安全 RAM 的较低值大小和非安全 RAM 的较高值大小：

0x000：所有 RAM 均为非安全

0x001：4 KB RAM 安全

0x002：8 KB RAM 安全

....

0x020 到 0x3FF：128 KB RAM 安全（例如所有 RAM 均为安全）

13.6.3 ETZPC 受控安全外设控制寄存器 x (ETZPC_DECPROTx)

ETZPC securable peripheral control register x

偏移地址：0x010 + 0x004 * (x- 0)（x = 0 到 3）

复位值：0x0000 0000、0x0000 0000、0x0000 0000、0x0000 0000

请根据*表 85* 列*默认*中的值来设置寄存器复位值，另请参见寄存器映射。

根据 DECPROTy[1:0] 并参照（x = 0 到 3）且（y= 0 到 15）的 ETZPC_DECPROTx 寄存器，为*表 85* 中具有索引 (16 * x + y) 的外设定义安全性和隔离：

00：安全读写

01：非安全读取，安全写入

10：保留

11：非安全。

DECPROTy[1:0] 允许使用的值需根据*表 85* 列*属性*进行设置。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DECPROT15 [1:0]		DECPROT14 [1:0]		DECPROT13 [1:0]		DECPROT12 [1:0]		DECPROT11 [1:0]		DECPROT10 [1:0]		DECPROT9 [1:0]		DECPROT8 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DECPROT7 [1:0]		DECPROT6 [1:0]		DECPROT5 [1:0]		DECPROT4 [1:0]		DECPROT3 [1:0]		DECPROT2 [1:0]		DECPROT1 [1:0]		DECPROT0 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 **DECPROT15[1:0]**：中具有索引 (16 * x + 15) 的外设的设置 *表 85*

位 29:28 **DECPROT14[1:0]**：中具有索引 (16 * x + 14) 的外设的设置 *表 85*

位 27:26 **DECPROT13[1:0]**：*表 85* 中具有索引 (16 * x + 13) 的外设的设置

位 25:24 **DECPROT12[1:0]**：*表 85* 中具有索引 (16 * x + 12) 的外设的设置

位 23:22 **DECPROT11[1:0]**：*表 85* 中具有索引 (16 * x + 11) 的外设的设置

位 21:20 **DECPROT10[1:0]**：*表 85* 中具有索引 (16 * x + 10) 的外设的设置

位 19:18 **DECPROT9[1:0]**：*表 85* 中具有索引 (16 * x + 9) 的外设的设置

位 17:16 **DECPROT8[1:0]**：*表 85* 中具有索引 (16 * x + 8) 的外设的设置

位 15:14 **DECPROT7[1:0]**：*表 85* 中具有索引 (16 * x + 7) 的外设的设置

位 13:12 **DECPROT6[1:0]**：*表 85* 中具有索引 (16 * x + 6) 的外设的设置

位 11:10 **DECPROT5[1:0]**：*表 85* 中具有索引 (16 * x + 5) 的外设的设置

位 9:8 **DECPROT4[1:0]**：*表 85* 中具有索引 (16 * x + 4) 的外设的设置

位 7:6 **DECPROT3[1:0]**：*表 85* 中具有索引 (16 * x + 3) 的外设的设置

位 5:4 **DECPROT2[1:0]**：*表 85* 中具有索引 (16 * x + 2) 的外设的设置

位 3:2 **DECPROT1[1:0]**：*表 85* 中具有索引 (16 * x + 1) 的外设的设置

位 1:0 **DECPROT0[1:0]**：*表 85* 中具有索引 (16 * x + 0) 的外设的设置

13.6.4 ETZPC decprot 锁定 x 寄存器 (ETZPC_DECPROT_LOCKx)

ETZPC decprot lock x register

偏移地址：0x030 + 0x004 * (x-0)（x = 0 到 1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK31	LOCK30	LOCK29	LOCK28	LOCK27	LOCK26	LOCK25	LOCK24	LOCK23	LOCK22	LOCK21	LOCK20	LOCK19	LOCK18	LOCK17	LOCK16
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOCK15	LOCK14	LOCK13	LOCK12	LOCK11	LOCK10	LOCK9	LOCK8	LOCK7	LOCK6	LOCK5	LOCK4	LOCK3	LOCK2	LOCK1	LOCK0
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31 **LOCK31**：*表 85* 中所定义的具有索引 (32 * x + 31) 的外设的安全锁定。

0：未锁定

1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 30 **LOCK30**：*表 85* 中所定义的具有索引 (32 * x + 30) 的外设的安全锁定。

0：未锁定

1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 29 **LOCK29**：*表 85* 中所定义的具有索引 (32 * x + 29) 的外设的安全锁定。

0：未锁定

1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 28 **LOCK28**：*表 85* 中所定义的具有索引 (32 * x + 28) 的外设的安全锁定。

0：未锁定

1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 27 **LOCK27**：*表 85* 中所定义的具有索引 (32 * x + 27) 的外设的安全锁定。

0：未锁定

1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 26 **LOCK26**：*表 85* 中所定义的具有索引 (32 * x + 26) 的外设的安全锁定。

0：未锁定

1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 25 **LOCK25**：*表 85* 中所定义的具有索引 (32 * x + 25) 的外设的安全锁定。

0：未锁定

1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 24 **LOCK24**：*表 85* 中所定义的具有索引 (32 * x + 24) 的外设的安全锁定。

0：未锁定

1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 23 **LOCK23**：*表 85* 中所定义的具有索引 (32 * x + 23) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 22 **LOCK22**：*表 85* 中所定义的具有索引 (32 * x + 22) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 21 **LOCK21**：*表 85* 中所定义的具有索引 (32 * x + 21) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 20 **LOCK20**：*表 85* 中所定义的具有索引 (32 * x + 20) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 19 **LOCK19**：*表 85* 中所定义的具有索引 (32 * x + 19) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 18 **LOCK18**：*表 85* 中所定义的具有索引 (32 * x + 18) 的外设的安全锁定。
0：未锁定
1：decprot[] 位在下次复位之前将处于锁定状态

位 17 **LOCK17**：*表 85* 中所定义的具有索引 (32 * x + 17) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 16 **LOCK16**：*表 85* 中所定义的具有索引 (32 * x + 16) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 15 **LOCK15**：*表 85* 中所定义的具有索引 (32 * x + 15) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 14 **LOCK14**：*表 85* 中所定义的具有索引 (32 * x + 14) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 13 **LOCK13**：*表 85* 中所定义的具有索引 (32 * x + 13) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 12 **LOCK12**：*表 85* 中所定义的具有索引 (32 * x + 12) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 11 **LOCK11**：*表 85* 中所定义的具有索引 (32 * x + 11) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 10 **LOCK10**：*表 85* 中所定义的具有索引 (32 * x + 10) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 9 **LOCK9**：*表 85* 中所定义的具有索引 (32 * x + 9) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 8 **LOCK8**：*表 85* 中所定义的具有索引 (32 * x + 8) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 7 **LOCK7**：*表 85* 中所定义的具有索引 (32 * x + 7) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 6 **LOCK6**：*表 85* 中所定义的具有索引 (32 * x + 6) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 5 **LOCK5**：*表 85* 中所定义的具有索引 (32 * x + 5) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 4 **LOCK4**：*表 85* 中所定义的具有索引 (32 * x + 4) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 3 **LOCK3**：*表 85* 中所定义的具有索引 (32 * x + 3) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 2 **LOCK2**：*表 85* 中所定义的具有索引 (32 * x + 2) 的外设的安全锁定。
0：未锁定
1：decprot[] 位在下次复位之前将处于锁定状态

位 1 **LOCK1**：*表 85* 中所定义的具有索引 (32 * x + 1) 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

位 0 **LOCK0**：*表 85* 中所定义的具有索引 32 * x 的外设的安全锁定。
0：未锁定
1：外设 decprot[] 位在下次复位之前将处于锁定状态，因此针对 decprot[] 的写入操作将被默默忽略（例如不会引起 APB SLVERR）

13.6.5 ETZPC IP HW 配置寄存器 (ETZPC_HWCFGR)

ETZPC IP HW configuration register

偏移地址：0x3F0

复位值：0x0000 4002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHUNKS1N4[7:0]								NUM_AHB_SEC[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUM_PER_SEC[7:0]								NUM_TZMA[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 **CHUNKS1N4[7:0]**：TZMA 大小步长 (TZMA size steps)（0：4 KB，1：1 KB）

位 23:16 **NUM_AHB_SEC[7:0]**：受控安全 AHB 主设备数 (Number of securable AHB masters)

位 15:8 **NUM_PER_SEC[7:0]**：受控安全外设数 (Number of securable peripherals)

位 7:0 **NUM_TZMA[7:0]**：TZMA 实例数 (Number of TZMA instances)

13.6.6 ETZPC IP 版本寄存器 (ETZPC_VERR)

ETZPC IP version register

偏移地址：0x3F4

复位值：0x0000 0021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本 (Major revision)
此位字段将返回 ETZPC 的主版本

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)
此位字段将返回 ETZPC 的次版本

13.6.7 ETZPC IP 版本寄存器 (ETZPC_IDR)

ETZPC IP version register

偏移地址：0x3F8

复位值：0x0010 0061

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：标识码。

此位字段将返回 ETZPC 的标识码

13.6.8 ETZPC IP 版本寄存器 (ETZPC_SIDR)

ETZPC IP version register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小和 ID。

此位字段将返回 ETZPC 的大小和 ID

13.6.9 ETZPC 寄存器映射

下表提供了 ETZPC 寄存器映射和复位值。

表 86. ETZPC 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	ETZPC_TZMA0_SIZE	LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res.	Res	Res	Res	R0SIZE									
	Reset value	0																						1	1	1	1	1	1	1	1	1	1
0x004	ETZPC_TZMA1_SIZE	LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res.	Res	Res	Res	R0SIZE									
	Reset value	0																						1	1	1	1	1	1	1	1	1	1
0x010	ETZPC_DECPROT0	DECPROT15.		DECPROT14		DECPROT13		DECPROT12		DECPROT11		DECPROT10		DECPROT9		DECPROT8		DECPROT7		DECPROT6		DECPROT5		DECPROT4		DECPROT3		DECPROT2		DECPROT1		DECPROT0	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014	ETZPC_DECPROT1	DECPROT15		DECPROT14		DECPROT13		DECPROT12		DECPROT11		DECPROT10		DECPROT9		DECPROT8		DECPROT7		DECPROT6		DECPROT5		DECPROT4		DECPROT3		DECPROT2		DECPROT1		DECPROT0	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	ETZPC_DECPROT2	DECPROT15.		DECPROT14		DECPROT13		DECPROT12		DECPROT11		DECPROT10		DECPROT9		DECPROT8		DECPROT7		DECPROT6		DECPROT5		DECPROT4		DECPROT3		DECPROT2		DECPROT1		DECPROT0	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x01C	ETZPC_DECPROT3	DECPROT15.		DECPROT14		DECPROT13		DECPROT12		DECPROT11		DECPROT10		DECPROT9		DECPROT8		DECPROT7		DECPROT6		DECPROT5		DECPROT4		DECPROT3		DECPROT2		DECPROT1		DECPROT0	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x024	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x030	ETZPC_DECPROT_LOCK0	LOCK31	LOCK30	LOCK29	LOCK28	LOCK27	LOCK26	LOCK25	LOCK24	LOCK23	LOCK22	LOCK21	LOCK20	LOCK19	LOCK18	LOCK17	LOCK16	LOCK15	LOCK14	LOCK13	LOCK12	LOCK11	LOCK10	LOCK9	LOCK8	LOCK7	LOCK6	LOCK5	LOCK4	LOCK3	LOCK2	LOCK1	LOCK0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x034	ETZPC_DECPROT_LOCK1	LOCK31	LOCK30	LOCK29	LOCK28	LOCK27	LOCK26	LOCK25	LOCK24	LOCK23	LOCK22	LOCK21	LOCK20	LOCK19	LOCK18	LOCK17	LOCK16	LOCK15	LOCK14	LOCK13	LOCK12	LOCK11	LOCK10	LOCK9	LOCK8	LOCK7	LOCK6	LOCK5	LOCK4	LOCK3	LOCK2	LOCK1	LOCK0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x038 to 0x3EC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x3F0	ETZPC_HWCFGR	CHUNKS1N4								NUM_AHB_SEC[7:0]								NUM_PER_SEC[7:0]								NUM_TZMA[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0
0x3F4	ETZPC_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	0	1	0	0	0	0	0

表 86. ETZPC 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x3F8	ETZPC_IDR	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1
0x3FC	ETZPC_SIDR	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

14 外设互联矩阵

14.1 外设互连

14.1.1 简介

多个外设间有直接互连。这可以在外设间实现自动通信和同步，节省了处理器资源并降低了电源功耗。

此外，这些硬件连接消除了软件延迟，允许设计可预测系统。

此举也节省了引脚和 GPIO。

这些互连可以在运行、CSleep 或停止模式下工作，具体取决于外设。

14.1.2 连接概述

连接可分为多种类型，具体如下所示。

异步连接 (A)

源输出信号通过目标时钟采样，从而可能在源输出事件到目标事件检测的延时中引入抖动。

同步连接 (S)

源和目标同步（它们采用相同的时钟），从源到目标的延时是确定的。不引入抖动。

TIM 输出的刹车/故障连接 (B)

源输出信号通过纯组合逻辑路径（无延时）禁止定时器输出。

表 87. 外设互连矩阵

源		目标																								
总线		AHB2		APB1					APB2						APB3					AXIM		APB5	APB6			
	外设	ADC1	ADC2	LPTIM1	TIM2	TIM3	TIM4	TIM5	FDCAN1	DFSDM	TIM1	TIM8	SAI1	SAI2	LPTIM2	LPTIM3	LPTIM4	LPTIM5	DTS	ETH1	ETH2	TAMP	TIM12	TIM15	TIM16	TIM17
-	AFMUX			A	A	A	A	A			A	A			A								A	A	A	A
APB1	LPTIM1	A	A							A									A							
	SPDIFRX																									A
	TIM2	A/S	A/S			S	S		A		S	S								S	S					
	TIM3	A/S	A/S		S		S	S	A	S	S									S	S			S		
	TIM4	A/S	A/S		S	S		S		S	S	S											S			
	TIM5											S											S			
	TIM6	S	S							S																
	TIM7									S																
APB2	FDCAN1							A/S												A	A					
	DFSDM										B	B												B	B	B
	TIM1	A/S	A/S		S	S	S	S		S		S												S		
	TIM8	A/S	A/S		S		S	S		S																
	SAI1				A									S		A										
	SAI2							A					S				A									
APB3	LPTIM2	A	A							A					S	S	S	S								
	LPTIM3	A	A							A							S	S	S			A (1)				
	LPTIM4														S			S								
	LPTIM5														S	S										
AHB2	ADC1									A	A	A														
	ADC2									A	A	A										A				
AHB4	RCC				A										A								A	A	A	A
	EXTI	A	A							A																
MLAHB	OTG							S																		
AXIM	ETH1				A	A			A																	
	ETH2				A	A			A																	
	USBH							A																		
APB5	RTC			A											A										A	
	TAMP			A											A											
APB6	TIM13																						S			
	TIM14																						S			
	TIM15	S	S			S					S															
	TIM16									A														S		
	TIM17																							S		

1. LPTIM3 输出专用于 TAMP 或其他外设。

表 88. 外设互连矩阵详细信息

源			目标			类型
总线	外设	信号	信号	外设	总线	
APB6	TIM15	TRGO	ITR0	TIM1	APB2	S
APB1	TIM2	TRGO	ITR1			S
	TIM3	TRGO	ITR2			S
	TIM4	TRGO	ITR3			S
-	AFMUX	TIM1_ETR	ETR0			A
AHB2	ADC1	AWD1	ETR3			A
		AWD2	ETR4			A
		AWD3	ETR5			A
	ADC2	AWD1	ETR6			A
		AWD2	ETR7			A
		AWD3	ETR8			A
-	AFMUX	TIM1_CH1	TI1_0			A
		TIM1_CH2	TI2_0			A
		TIM1_CH3	TI3_0			A
		TIM1_CH4	TI4_0			A
		TIM1_BKIN	BRK_0			A
APB2	DFSDM	BRK0	BRK_8			B
-	AFMUX	TIM1_BKIN2	BRK2_0	TIM1	APB2	A
APB2	DFSDM	BRK1	BRK2_8			B
	TIM1	TRGO	ITR0	TIM2	APB1	S
	TIM8	TRGO	ITR1			S
APB1	TIM3	TRGO	ITR2			S
	TIM4	TRGO	ITR3			S
-	AFMUX	TIM2_ETR	ETR0			A
AHB4	RCC	LSE	ETR3			A
APB2	SAI1	FS_A	ETR4			A
		FS_B	ETR5			A
AXIM	ETH1	PPS	ETR6			A
	ETH2	PPS	ETR7			A
-	AFMUX	TIM2_CH1	TI1_0			A
		TIM2_CH2	TI2_0			A
		TIM2_CH3	TI3_0			A
		TIM2_CH4	TI4_0			A

表 88. 外设互连矩阵详细信息（续）

源			目标			类型
总线	外设	信号	信号	外设	总线	
APB2	TIM1	TRGO	ITR0	TIM3	APB1	S
APB1	TIM2	TRGO	ITR1			S
APB6	TIM15	TRGO	ITR2			S
APB1	TIM4	TRGO	ITR3			S
-	AFMUX	TIM3_ETR	ETR0	TIM3	APB1	A
AXIM	ETH1	PPS	ETR6			A
	ETH2	PPS	ETR7			A
-	AFMUX	TIM3_CH1	TI1_0			A
		TIM3_CH2	TI2_0			A
		TIM3_CH3	TI3_0			A
		TIM3_CH4	TI4_0			A
APB2	TIM1	TRGO	ITR0	TIM4	APB1	S
APB1	TIM2	TRGO	ITR1			S
	TIM3	TRGO	ITR2			S
APB2	TIM8	TRGO	ITR3			S
-	AFMUX	TIM4_ETR	ETR0			A
		TIM4_CH1	TI1_0			A
		TIM4_CH2	TI2_0			A
		TIM4_CH3	TI3_0			A
		TIM4_CH4	TI4_0			A
APB2	TIM1	TRGO	ITR0	TIM5	APB1	S
	TIM8	TRGO	ITR1			S
APB1	TIM3	TRGO	ITR2			S
	TIM4	TRGO	ITR3			S
APB2	FDCAN1	SOC	ITR6			S
MLAHB	OTG	SOF	ITR8			S
-	AFMUX	TIM5_ETR	ETR0			A
APB2	SAI2	FS_A	ETR1			A
		FS_B	ETR2			A
AXIM	USBH	SOF	ETR3			A
-	AFMUX	TIM5_CH1	TI1_0			A
APB2	FDCAN1	TMP	TI1_1	TIM5	APB1	A
		RTP	TI1_2			A

表 88. 外设互连矩阵详细信息（续）

源			目标			类型
总线	外设	信号	信号	外设	总线	
-	AFMUX	TIM5_CH2	TI2_0	TIM5	APB1	A
		TIM5_CH3	TI3_0			A
		TIM5_CH4	TI4_0			A
APB2	TIM1	TRGO	ITR0	TIM8	APB2	S
APB1	TIM2	TRGO	ITR1			S
	TIM4	TRGO	ITR2			S
	TIM5	TRGO	ITR3			S
-	AFMUX	TIM8_ETR	ETR0			A
AHB2	ADC1	AWD1	ETR3			A
		AWD2	ETR4			A
		AWD3	ETR5			A
	ADC2	AWD1	ETR6			A
		AWD2	ETR7			A
		AWD3	ETR8			A
-	AFMUX	TIM8_CH1	TI1_0			A
		TIM8_CH2	TI2_0			A
		TIM8_CH3	TI3_0			A
		TIM8_CH4	TI4_0			A
		TIM8_BKIN	BRK_0			A
APB2	DFSDM	BRK2	BRK_8			B
-	AFMUX	TIM8_BKIN2	BRK2_0			A
APB2	DFSDM	BRK3	BRK2_8			B
APB1	TIM4	TRGO	ITR0	TIM12	APB6	S
	TIM5	TRGO	ITR1			S
APB6	TIM13	OC1	ITR2			S
	TIM14	OC1	ITR3			S
-	AFMUX	TIM12_CH1	TI1_0			A
AHB4	RCC	hsi_cal_ck	TI1_1			A
		csi_cal_ck	TI1_2			A
-	AFMUX	TIM12_CH2	TI2_0			A
		TIM13_CH1	TI1_0	TIM13	APB6	A
		TIM14_CH1	TI1_0	TIM14	APB6	A

表 88. 外设互连矩阵详细信息（续）

源			目标			类型
总线	外设	信号	信号	外设	总线	
APB2	TIM1	TRGO	ITR0	TIM15	APB6	S
APB1	TIM3	TRGO	ITR1			S
APB6	TIM16	OC1	ITR2			S
	TIM17	OC1	ITR3			S
-	AFMUX	TIM15_CH1	TI1_0			A
		TIM2_CH1	TI1_1			A
		TIM3_CH1	TI1_2			A
		TIM4_CH1	TI1_3			A
AHB4	RCC	LSE	TI1_4			A
		CSI	TI1_5			A
		MCO2	TI1_6			A
		hsi_cal_ck	TI1_7			A
		csi_cal_ck	TI1_8			A
-	AFMUX	TIM15_CH2	TI2_0			A
		TIM2_CH2	TI2_1			A
		TIM3_CH2	TI2_2			A
		TIM4_CH2	TI2_3			A
		TIM15_BKIN	BRK_0			A
APB2	DFSDM	BRK0	BRK_8			B
-	AFMUX	TIM16_CH1	TI1_0	TIM16	APB6	A
AHB4	RCC	LSI	TI1_1			A
		LSE	TI1_2			A
APB5	RTC	WKUP	TI1_3			A
-	AFMUX	TIM16_BKIN	BRK_0			A
APB2	DFSDM	BRK1	BRK_8			B
-	AFMUX	TIM17_CH1	TI1_0	TIM17	APB6	A
APB1	SPDIFRX	FS	TI1_1			A
AHB4	RCC	HSE_RTC	TI1_2			A
		MCO1	TI1_3			A
-	AFMUX	TIM17_BKIN	BRK_0	TIM17	APB6	A
APB2	DFSDM	BRK2	BRK_8			B

表 88. 外设互连矩阵详细信息（续）

源			目标			类型
总线	外设	信号	信号	外设	总线	
-	AFMUX	LPTIM1_ETR	ETR0	LPTIM1	APB1	A
APB5	RTC	ALA	ETR1			A
		ALB	ETR2			A
APB5	TAMP	TAMP1	ETR3			A
		TAMP2	ETR4			A
		TAMP3	ETR5			A
-	AFMUX	LPTIM1_IN1	IN1_0			A
		LPTIM1_IN2	IN2_0			A
		LPTIM2_ETR	ETR0	LPTIM2	APB3	A
APB5	RTC	ALA	ETR1			A
		ALB	ETR2			A
APB5	TAMP	TAMP1	ETR3			A
		TAMP2	ETR4			A
		TAMP3	ETR5			A
-	AFMUX	LPTIM2_IN1	IN1_0			A
		LPTIM2_IN2	IN2_0			A
APB3	LPTIM2	OUT	ETR0	LPTIM3	APB3	S
AHB4	RCC	hsi_cal_ck	ETR1			A
APB3	LPTIM4	OUT	ETR2			S
	LPTIM5	OUT	ETR3			S
APB2	SAI1	FS_A	ETR4			A
		FS_B	ETR5			A
AHB4	RCC	hse_ck	IN2_0			A
APB3	LPTIM2	OUT	ETR0	LPTIM4	APB3	S
	LPTIM3	OUT	ETR1			S
	LPTIM5	OUT	ETR3			S
APB2	SAI2	FS_A	ETR4			A
		FS_B	ETR5			A
APB3	LPTIM2	OUT	ETR0	LPTIM5	APB3	S
	LPTIM3	OUT	ETR1			S
	LPTIM4	OUT	ETR2			S

表 88. 外设互连矩阵详细信息（续）

源			目标			类型
总线	外设	信号	信号	外设	总线	
APB2	TIM1	TRGO	TRG0	DFSDM	APB2	S
		TRGO2	TRG1			S
	TIM8	TRGO	TRG2			S
		TRGO2	TRG3			S
APB1	TIM3	TRGO	TRG4			S
	TIM4	TRGO	TRG5			S
APB6	TIM16	OC1	TRG6			A
APB1	TIM6	TRGO	TRG7			S
	TIM7	TRGO	TRG8			S
AHB4	EXTI	EXTI11	TRG24			A
		EXTI15	TRG25			A
APB1	LPTIM1	OUT	TRG26			A
APB3	LPTIM2	OUT	TRG27			A
	LPTIM3	OUT	TRG28			A
AHB2	ADC1	DATA[15:0]	DAT_ADC0[15:0]			A
	ADC2	DATA[15:0]	DAT_ADC1[15:0]			A
APB2	TIM1	CC1	EXT0	ADC1	AHB2	A
		CC2	EXT1			A
		CC3	EXT2			A
APB1	TIM2	CC2	EXT3			A
	TIM3	TRGO	EXT4			S
	TIM4	CC4	EXT5			A
AHB4	EXTI	EXTI11	EXT6			A
APB2	TIM8	TRGO	EXT7			S
		TRGO2	EXT8			S
	TIM1	TRGO	EXT9			S
		TRGO2	EXT10			S
APB1	TIM2	TRGO	EXT11			S
	TIM4	TRGO	EXT12			S
	TIM6	TRGO	EXT13			S
APB6	TIM15	TRGO	EXT14			S
APB1	TIM3	CC4	EXT15			A
	LPTIM1	OUT	EXT18			A

表 88. 外设互连矩阵详细信息（续）

源			目标			类型
总线	外设	信号	信号	外设	总线	
APB3	LPTIM2	OUT	EXT19	ADC1	AHB2	A
	LPTIM3	OUT	EXT20			A
APB2	TIM1	TRGO	JEXT0			S
		CC4	JEXT1			A
APB1	TIM2	TRGO	JEXT2			S
		CC1	JEXT3			A
	TIM3	CC4	JEXT4			A
	TIM4	TRGO	JEXT5			S
AHB4	EXTI	EXTI15	JEXT6			A
APB2	TIM8	CC4	JEXT7			A
	TIM1	TRGO2	JEXT8			S
	TIM8	TRGO	JEXT9			S
		TRGO2	JEXT10			S
APB1	TIM3	CC3	JEXT11			A
		TRGO	JEXT12			S
		CC1	JEXT13			A
	TIM6	TRGO	JEXT14			S
APB6	TIM15	TRGO	JEXT15			S
APB1	LPTIM1	OUT	JEXT18			A
APB3	LPTIM2	OUT	JEXT19			A
	LPTIM3	OUT	JEXT20			A

表 88. 外设互连矩阵详细信息（续）

源			目标			类型
总线	外设	信号	信号	外设	总线	
APB2	TIM1	CC1	EXT0	ADC2	AHB2	A
		CC2	EXT1			A
		CC3	EXT2			A
APB1	TIM2	CC2	EXT3			A
	TIM3	TRGO	EXT4			S
	TIM4	CC4	EXT5			A
AHB4	EXTI	EXTI11	EXT6			A
APB2	TIM8	TRGO	EXT7			S
		TRGO2	EXT8			S
	TIM1	TRGO	EXT9			S
		TRGO2	EXT10			S
APB1	TIM2	TRGO	EXT11			S
	TIM4	TRGO	EXT12			S
	TIM6	TRGO	EXT13			S
APB6	TIM15	TRGO	EXT14			S
APB1	TIM3	CC4	EXT15			A
	LPTIM1	OUT	EXT18			A
APB3	LPTIM2	OUT	EXT19			A
	LPTIM3	OUT	EXT20			A
APB2	TIM1	TRGO	JEXT0			S
		CC4	JEXT1			A
APB1	TIM2	TRGO	JEXT2			S
		CC1	JEXT3			A
	TIM3	CC4	JEXT4			A
	TIM4	TRGO	JEXT5			S

表 88. 外设互连矩阵详细信息（续）

源			目标			类型
总线	外设	信号	信号	外设	总线	
AHB4	EXTI	EXTI15	JEXT6	ADC2	AHB2	A
APB2	TIM8	CC4	JEXT7			A
	TIM1	TRGO2	JEXT8			S
	TIM8	TRGO	JEXT9			S
		TRGO2	JEXT10			S
APB1	TIM3	CC3	JEXT11			A
		TRGO	JEXT12			S
		CC1	JEXT13			A
	TIM6	TRGO	JEXT14			S
APB6	TIM15	TRGO	JEXT15			S
APB1	LPTIM1	OUT	JEXT18			A
APB3	LPTIM2	OUT	JEXT19			A
	LPTIM3	OUT	JEXT20			A
APB1	LPTIM1	OUT	TRG1	DTS	APB3	A
APB3	LPTIM2	OUT	TRG2			S
	LPTIM3	OUT	TRG3			S
APB1	TIM2	TRGO	SWT0	FDCAN1	APB2	A
	TIM3	TRGO	SWT1			A
AXIM	ETH1	PPS	SWT2			A
	ETH2	PPS	SWT3			A
APB1	TIM2	TRGO	EVT0			A
	TIM3	TRGO	EVT1			A
AXIM	ETH1	PPS	EVT2			A
	ETH2	PPS	EVT3			A
APB1	TIM3	CNT[15:0]	EXT_TS[15:0]			A
	TIM2	TRGO	PTP0	ETH1	AXIM	S
	TIM3	TRGO	PTP1			S
APB2	FDCAN1	TMP	PTP3			A
APB1	TIM2	TRGO	PTP0	ETH2	AXIM	S
	TIM3	TRGO	PTP1			S
APB2	FDCAN1	TMP	PTP3			A
	SAI2	SYNCOUT_FS	SYNCIN1_FS	SAI1	APB2	S
		SYNCOUT_SCK	SYNCIN1_SCK			S

表 88. 外设互连矩阵详细信息（续）

源			目标			类型
总线	外设	信号	信号	外设	总线	
APB2	SAI1	SYNCOUT_FS	SYNCIN0_FS	SAI2	APB2	S
		SYNCOUT_SCK	SYNCIN0_SCK			S
APB3	LPTIM3	OUT	ITAMP4	TAMP	APB5	A(1)
AHB2	ADC2	AWD1	ITAMP7			A
		AWD2	ITAMP12			A
		AWD3	ITAMP13			A

1. LPTIM3 输出专用于 TAMP 或其他外设。

14.2 MDMA

表 89. MDMA 连接

源			目标			备注
总线	外设	信号	信号	外设	总线	
AHB1/ MLAHB	DMA1	tcif0	str0	MDMA	AXIM	DMA1 数据流 0 传输完成
		tcif1	str1			DMA1 数据流 1 传输完成
		tcif2	str2			DMA1 数据流 2 传输完成
		tcif3	str3			DMA1 数据流 3 传输完成
		tcif4	str4			DMA1 数据流 4 传输完成
		tcif5	str5			DMA1 数据流 5 传输完成
		tcif6	str6			DMA1 数据流 6 传输完成
		tcif7	str7			DMA1 数据流 7 传输完成
	DMA2	tcif0	str8			DMA2 数据流 0 传输完成
		tcif1	str9			DMA2 数据流 1 传输完成
		tcif2	str10			DMA2 数据流 2 传输完成
		tcif3	str11			DMA2 数据流 3 传输完成
		tcif4	str12			DMA2 数据流 4 传输完成
		tcif5	str13			DMA2 数据流 5 传输完成
		tcif6	str14			DMA2 数据流 6 传输完成
		tcif7	str15			DMA2 数据流 7 传输完成
MLAHB	DMA3	tcif0	str16			DMA3 数据流 0 传输完成
		tcif1	str17			DMA3 数据流 1 传输完成
		tcif2	str18			DMA3 数据流 2 传输完成
		tcif3	str19			DMA3 数据流 3 传输完成
		tcif4	str20			DMA3 数据流 4 传输完成

表 89. MDMA 连接（续）

源			目标			备注
总线	外设	信号	信号	外设	总线	
MLAHB	DMA3	tcif5	str21	MDMA	AXIM	DMA3 数据流 5 传输完成
		tcif6	str22			DMA3 数据流 6 传输完成
		tcif7	str23			DMA3 数据流 7 传输完成
AXIM	FMC	data_tx_rx	str24			NAND 数据传输（Tx 或 Rx）通道
		error_position	str25			NAND ECC/BCH 错误通道
	QUADSPI	ft_trg	str26			FIFO 阈值
		tc_trg	str27			传输完成
AHB5	CRYP	crypin_dmabreq	str28			来自输入的 4 字请求
		crypout_dmabreq	str29			来自输出的 4 字就绪
	HASH	dmabreq	str30			16 字请求
	SAES	dma_aes_wr	str31			输入数据写入请求
		dma_aes_rd	str32			输出数据读取请求

15 MDMA 控制器 (MDMA)

15.1 MDMA 简介

主机直接内存访问 (MDMA) 用于在内存与内存之间或外设与内存之间提供高速数据传输。可以在无需任何 CPU 操作的情况下通过 MDMA 快速移动数据。这样节省的 CPU 资源便可供其他操作使用。

MDMA 控制器为主内存和外设寄存器访问（系统访问端口）提供了一个主设备 AXI 接口。

MDMA 与标准 DMA 控制器（DMA1 或 DMA2）结合使用。它最多可提供 32 个通道，每个通道都专用于管理来自一个 DMA 数据流内存缓冲区或其他外设（带集成 FIFO）的内存访问请求。

15.2 MDMA 主要特性

- 采用 AXI 主设备总线架构。
- 32通道
- 多达 40 个硬件触发源
- 每个通道请求均可在任何请求源之间选择。可由软件配置，允许多个外设启动DMA请求。在一个块传输结束时，会自动更改触发选择。
- 所有通道均相同，可连接至一个标准 DMA 或一个外设请求（通过数据读/写进行确认）系统，或是连接至一个外设请求/确认系统
- 各个通道还支持软件触发
- 一个 256 级内存缓冲区，被分为两个 128 级先进先出 (FIFO)，将用于针对一个或两个连续缓冲区临时存储要传输的数据（突发或单次传输模式）。一个 FIFO 将存储要在当前通道块传输期间传输的数据（最大为块传输大小）。另一个 FIFO 可用于要传输的下一个缓冲区，既适用于同一通道也适用于下一通道传输。
- DMA 通道之间的优先级可用软件编程（四个级别：非常高、高、中、低），在软件优先级相同的情况下可以通过硬件决定优先级（例如，通道 0 的优先级高于通道 1）。
- 独立的源和目标传输宽度可分别设置（字节、半字、字、双字）：源和目标的数据宽度不相等时，MDMA 会适当地封装/解封数据来优化带宽。
- 可分别选择源和目标的数据长度及地址增量。

注： *借助这一分离特性，可在软件层面进行某些更高级的封装/解封操作。例如，2 个 16 位数据块可使用两个 MDMA 通道在目标内存中彼此交错，该过程的实现仅需编程 2 个通道（以 4 字节步长为增量，且数据大小为 16 位 + 2 个通道间的起始地址移 2 位）。*

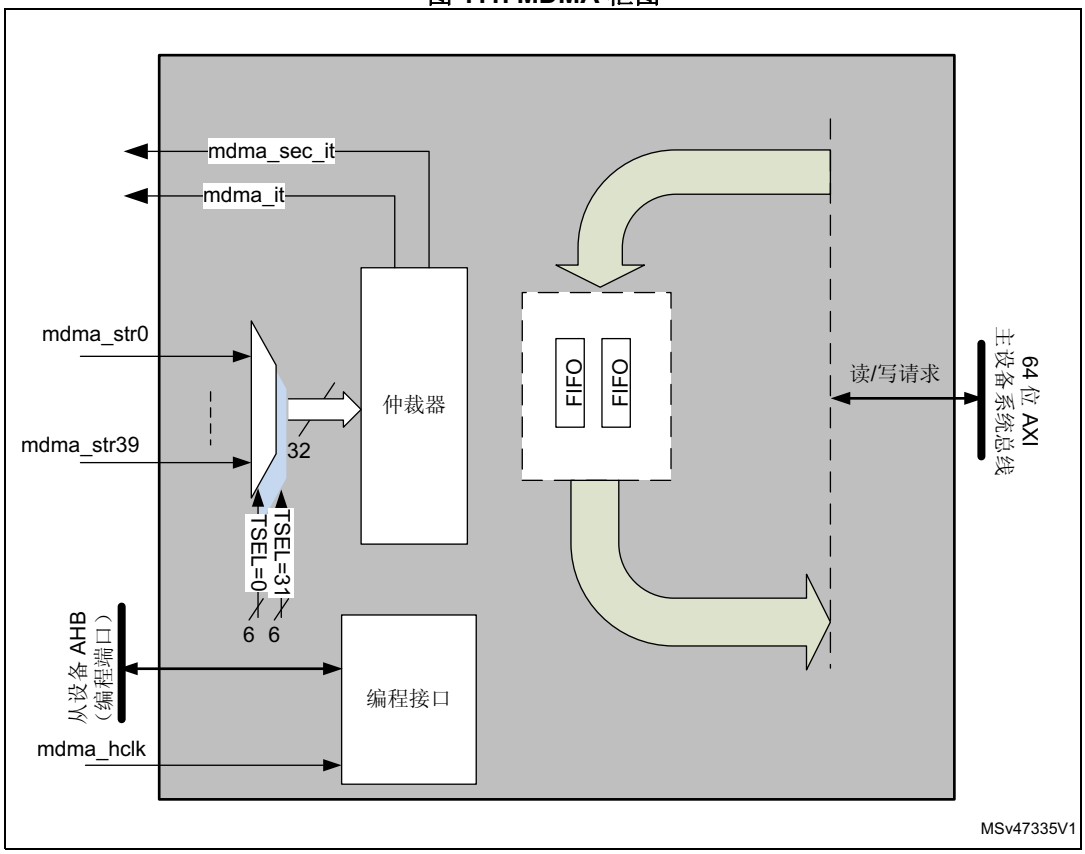
- 源和目标的递增、递减或非递增/固定寻址
- 始终根据小端模式约定进行数据封装 / 解封：数据实体（双字、字或半字）中的低地址始终包含最低有效字节。这与源和目标的地址递增/递减模式无关。
- 支持递增突发传输。突发大小可由软件进行配置，最大可为 128 字节。对于较大的数据单元，突发长度受限，需遵循最大 128 字节数据突发大小（例如 16 x 64 位或 32 x 32 位）。
- 对于 TCM 内存访问，只有在增量和数据大小等于或小于等于 32 位时，才允许突发访问。
- 提供五个事件标志（MDMA 通道传输完成、MDMA 块传输完成、MDMA 块重复传输完成、MDMA 缓冲区传输完成、MDMA 传输出错），且这些标志可生成中断。

15.3 MDMA 功能描述

15.3.1 MDMA 框图

下图显示了 MDMA 的框图。

图 111. MDMA 框图



15.3.2 MDMA 内部信号

下表显示了内部 MDMA 信号。

表 90. MDMA 内部输入/输出信号

信号名称	信号类型	说明
mdma_hclk	数字输入	MDMA AHB 时钟
mdma_it	数字输出	MDMA 中断
mdma_sec_it	数字输出	MDMA 安全中断
mdma_str[0:39]	数字输入	MDMA 数据流请求

15.3.3 MDMA 概述

MDMA 执行直接内存传输：做为一个 AXI 主设备，MDMA 可以控制 AXI 总线矩阵来发起 AXI 事务。

MDMA 可以执行下列事务：

- 内存到内存的传输（软件触发）
- 外设到内存的传输
- 内存到外设的传输

对于后两种传输类型，内存还可替换为内存映射的外设，其无法控制 MDMA 流。使用这些类型的传输且请求来自标准 DMA（DMA1 或 DMA2）时，外设寄存器访问会替换为该 DMA 所使用的内存缓冲区的内存访问。

注： *固定地址模式不能够用于内存访问。*

源和目标仅由地址指定（内存映射的外设也如此）。

AHB 从端口用于对 MDMA 进行编程（它支持 8、16 和 32 位访问）。

单次请求传输的数据数组长度为以下长度之一：

- 缓冲区传输大小
- 数据块大小
- 重复块
- 完整通道数据（直至通道的链表指针为空）

通过 TRGM[1:0]（触发模式）选择字段进行大小选择。

用户必须根据可用的数据数组大小（通常在 DMA1/2 内存缓冲区中）和其他 MDMA 通道的“实时”要求，选择其中的一种（因为 MDMA 缓冲区传输是不会在 MDMA 通道请求之间进行新仲裁的最小数据集合）。

对于每个通道，有三种数据数组长度非常重要：

- 突发大小：此为可在突发模式下执行的数据传输长度。此突发长度定义了不能被总线仲裁中断并且可阻止其他主设备访问该总线的数据传输最大长度。
- 缓冲区传输大小：此为在检查其他通道的 MDMA 请求之前，要在某个通道中传输的数据数组长度。在该长度范围内的数据传输时，不能够被其他 MDMA 通道请求所中断
- 块大小：该值有两个可同时使用的含义：
  - 主要功能：在 MDMA 链表的块结构中说明数据块的长度（对应于链表中的一个条目）
  - 可选功能：如果 TRGM[1:0] 等于 01，则该长度为在单次 MDMA 请求激活（针对相应通道）时传输的数据数组长度

15.3.4 MDMA通道

每个 MDMA 通道都能够提供源和目标之间的单向传输链路。

每个通道均可执行以下类型的传输：

- 单块传输：传输一个块。在块传输结束时，会禁止 MDMA 通道，并会生成通道传输结束中断。
- 重复块传输：在禁止相应通道前，传输大量块。
- 链表传输：当前数据块（或重复模式下的最后一个块）的传输完成后，会从内存加载新块控制结构，并会开始新块传输。

可编程要针对每个请求传输的最小数据量（缓冲区大小，最高 128 字节）。一个块中的数据总量是可编程的，最大为 64 KB。每次传输后，该值都会递减。此计数器达到 0 时，即表示达到块末尾，并会根据重复传输计数器（针对重复块传输）和/或链表结构值决定下一步操作。

注： *如果块长度不是缓冲区长度的倍数，则块中的最后一次缓冲区传输将变短，其中包含要在当前块中传输的剩余字节。*

如果链路结构地址指向有效内存地址，则 MDMA 会在该地址处从内存重载完整通道描述符结构寄存器内容。然后，将基于该信息执行新块传输（在出现下一个 MDMA 通道请求时）。

如果链路结构地址为 0x0，则在当前/重复块传输结束时，将禁止 MDMA 通道，并且会生成通道传输结束中断。

15.3.5 源、目标和传输模式

源传输和目标传输可在整个 4 GB 区域（地址在 0x0000 0000 和 0xFFFF FFFF 之间）寻址外设和内存。

源/目标地址可固定（例如 FIFO/单个数据寄存器外设）或递增/递减。可在单次访问或突发模式下进行传输（可编程）。

15.3.6 指针更新

根据 MDMA_CxCR 寄存器的 SINC[1:0] 和 DINC[1:0] 位字段，源和目标内存指针在每次传输后可以自动递增/递减或保持常量。

通过单个寄存器/FIFO 模式访问外设源或目标数据时，禁止递增模式十分有用。

如果使能了递增/递减模式，则根据在 MDMA_CxCR SINCOS[1:0] 或 DINCOS[1:0] 位中编程的增量大小，下一次数据传输的地址将是前一次传输的地址递增/递减 1、2、4 或 8。

为优化封装操作，可独立编程增量偏移大小和数据大小。

15.3.7 MDMA 缓冲区传输

此为发生 MDMA 请求事件时，在一个通道上传输的数据的最小逻辑量（最多 128 字节）。

MDMA 缓冲区传输包括一系列给定数据量传输（以单次或突发数据传输的形式进行）。要传输的数据项的数目及其宽度（8 位、16 位、32 位或 64 位）可用软件进行编程。用于数据传输的突发长度也可独立编程。

在产生数据数组传输请求事件后，DMA/外设会向 MDMA 发送请求信号，以便后者根据通道优先级处理该请求。

如果已设置掩码地址寄存器，则系统将通过向掩码寄存器指定的地址写入掩码数据值或是通过硬件 ACK 信号来确认这个请求。

如果掩码地址寄存器未设置（值 0x00），则向该外设读取/写入数据即可复位请求。此时，如果由目标外设完成请求，则必须将写入操作设为不可缓冲，以避免出现错误的新 MDMA 请求。

在 MDMA 请求之后当前通道需要传输的数据总量由 TRGM[1:0] 字段决定。

如果 TRGM[1:0] 等于 00，MDMA 会在一个缓冲区的数据传输完成后在同一个通道上等待另一个请求。

注： *在这种情况下，不会再考虑当前激活通道的硬件请求（FIFO 中的数据），直至该通道的写入阶段结束。即使通道在读取阶段结束后仍处于激活状态，另一通道（甚至是优先级较低的通道）也能够开始读取阶段。较低优先级的通道与当前通道的数据传输交错进行。*

如果 TRGM[1:0] 不为 00（需要传输多个缓冲区），则当前通道的 mdma_strx 保持激活（在内部存储），直至完成 TRGM 定义的完整传输（块、重复块或整个通道/链表数据）。传输完单一缓冲区后，MDMA 将进入新仲裁阶段（在新外部请求和内部存储请求之间）。如果不存在更高的其他优先级，则通道请求将激活，并且会对同一通道开始新缓冲区传输。

注： *如果 TRGM[1:0] 不为 00，则一次请求后会传输大量数据。但是，由于在每个缓冲区传输后都会进行通道仲裁，因此在较低优先级通道的数据传输将较高级别 MDMA 请求阻塞的时间不会超过一个缓冲区的传输周期。*

15.3.8 请求仲裁

仲裁器根据 MDMA 通道请求优先级对其进行管理。MDMA 空闲以及在各个缓冲区传输结束后，会对所有已使能通道检查所有 MDMA 请求（硬件或软件）。

优先级管理分为两个阶段：

- 软件：每个数据流优先级都可以在 MDMA_CxCR 寄存器中配置。分为四个级别：
  - 非常高优先级
  - 高优先级
  - 中优先级
  - 低优先级
- 硬件：就硬件而言，通道优先级固定。如果两个请求具有相同的软件优先级，则编号低的通道优先于编号高的通道。例如，如果通道 2 和通道 4 的软件优先级相同，则通道 2 的优先级较高。

15.3.9 FIFO

FIFO 结构用于在将源数据写入到目标之前，临时存储这些数据。所有通道共用一个中央 FIFO 结构。

为最大化数据带宽和总线使用率，使用了以下机制，从而可并行执行多个读/写操作。

- 在缓冲区传输期间，如果 FIFO 包含的数据足以用于目标突发传输，则会立即开始写入操作。
- 系统将用于一个缓冲区传输的所有数据读入 FIFO 时，仲裁将开始。之后，可将要传输的下一缓冲区数据读入 FIFO。

在缓冲区数据传输期间，如果正在工作的通道因为出错而被禁止，那么内部 FIFO 中的剩余数据将被丢弃。

15.3.10 块传输

块为“连续的”数据数组，最多支持 64 KB，通过连续缓冲区传输实现传输。

每个数据块均由起始地址和块长度进行定义。完成块传输后，可执行以下三种操作之一：

- 块为重复块传输的一部分：重载块长度，以及计算新块起始地址（基于 MDMA_CxBRUR 中的信息）。
- 块为单块或重复块传输中的最后一个块：从内存加载下一块的信息（使用 MDMA_CxLAR 中的链表地址信息）。
- 块为需要针对当前 MDMA 通道传输的最后一个块 (MDMA_CxLAR = 0)：通道被禁止，且不会为该通道接受更多 MDMA 请求。

15.3.11 块重复模式

块重复模式用于以源和目标的不同起始地址重复块传输。

重复块模式激活（重复传输计数器 ≠ 0）时，当前块传输结束后，将更新块参数（根据 BRSUM/BRDUM 配置重载 BNDT 值和 SAR/DAR 值），且重复传输计数器将减一。

重复块计数器达到 0 时，系统会将此最后一个块视为单块传输。

15.3.12 链表模式

链表模式支持从 MDMA_CxLAR 中给出的地址加载新的 MDMA 配置（MDMA_CxTCR、MDMA_CxBNDTR、MDMA_CxSAR、MDMA_CxDAR、MDMA_CxBRUR、MDMA_CxLAR、MDMA_CxTBR、MDMA_CxMAR 和 MDMA_CxMDR 寄存器）。该地址必须对 AXI 系统总线上映射的内存进行寻址。

在进行完该操作后，通道即准备好接受新请求（如上文的块/重复块模式中所述），或者会继续传输（如果 TRGM[1:0] = 11）。

加载 MDMA_CxTBR 值时，可自动更改触发源。

TRGM[1:0] = 11 时，TRGM 和 SWRM 值不能更改。

15.3.13 MDMA 传输完成

以下各种事件均可结束传输过程，并将状态寄存器 (MDMA_CxISR) 中的 CTCIF 位置 1：

- MDMA_CxBNDTR 计数器达到零，块重复传输计数器为 0 以及链表指针地址为 0
- 传输结束前禁止了通道（通过将 MDMA_CxCR 中的 EN 位清零），并且所有的剩余数据均已从 FIFO 传输到目标

15.3.14 MDMA 传输暂停

可以随时暂停 MDMA 传输以供稍后重新开始；也可以在 MDMA 传输结束前明确禁止传输。

分为两种情况：

- 通道被禁止，以后不从停止点重新开始。在这种情况下，只需将 MDMA_CxCR 寄存器中的 EN 位清零来禁止通道，除此之外不需要任何其他操作。禁止数据流可能要花费一些时间（需要首先完成正在进行的缓冲区传输）。传输完成中断标志被置 1 表明传输已结束。此时 MDMA_CxCR 中的 EN 位值为 0，这确认了通道的数据传输已中断。MDMA_CxNDTR 寄存器包含通道停止时剩余数据项的数目。这样软件便可以确定通道中断前已传输了多少数据项。
- 在 MDMA_CxBNDTR 寄存器中要传输的剩余字节数达到 0 之前，通道暂停传输。目的是以后通过重新使能通道重新开始传输。通道传输完成中断标志 CTCIF 被置 1 表明传输已结束。如果 MDMA_CxBNDTR、MDMA_CxSAR 和 MDMA_CxDAR 寄存器未由软件修改，则重新使能通道时会继续进行传输。在重新启动通道前，还必须将 CTCIF 复位。

注： *如果完成的缓冲区是最后一个块，则还会在禁止通道前更新配置寄存器，以便正确准备软重启。*

在重新编程通道前，软件必须等待 MDMA_CTCIF 寄存器置 1，以确保正在进行的操作均已完成。

15.3.15 错误管理

MDMA 能够检测到以下错误，且届时会将传输错误中断标志 (TEIF) 置 1：

- MDMA 读/写访问期间发生总线错误
- 地址对齐的位置与数据的大小不匹配
- 块大小不是（源和/或目标）数据大小的倍数：最后一次传输时会激活该错误，错误地址指向最后一次传输（无法完成）

15.4 MDMA 中断

对于每个 MDMA 通道，可在发生以下事件时生成中断：

- 通道传输完成
- 块传输完成
- 块传输重复完成
- 缓冲区传输完成
- 传输错误

可以使用单独的中断使能控制位以实现灵活性，如下表所示。

表 91. MDMA 中断请求

中断事件	事件标志	使能控制位
通道传输完成	CTCIF	CTCIE
块传输重复完成	BTRIF	BTRIE
块传输完成	BTIF	BTIE
缓冲区传输完成	TCIF	TCIE
传输错误	TEIF	TEIE

注： *在将使能控制位置 1 前，必须将相应的事件标志清零，否则如果该位已置 1，则会立即生成中断。*

如果至少有一个中断标志以及相应的使能控制位置 1，则会在 MDMA_SGISR 中将通道中断位置 1。中断输出或安全中断输出也会激活。如果在 NVIC 中使能了相应的中断通道，则会生成中断。

15.5 MDMA 寄存器

可按字、半字或字节格式访问 MDMA 寄存器。

15.5.1 MDMA 全局中断状态寄存器 (MDMA_GISR0)

MDMA global interrupt status register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GIF31	GIF30	GIF29	GIF28	GIF27	GIF26	GIF25	GIF24	GIF23	GIF22	GIF21	GIF20	GIF19	GIF18	GIF17	GIF16
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GIF15	GIF14	GIF13	GIF12	GIF11	GIF10	GIF9	GIF8	GIF7	GIF6	GIF5	GIF4	GIF3	GIF2	GIF1	GIF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **GIF[31:0]**：通道 x 全局中断标志（x = 31 到 0）

该位字段由硬件置 1 和复位。它是在 MDMA_CTCIEx、MDMA_BTIEx、MDMA_BRTIEx 和 MDMA_TEIEx 中使能的所有通道 x 中断标志（CTCIF、BTIF、BRTIF、TEIF）的逻辑或运算结果。

0：通道 x 未生成中断

1：通道 x 生成了中断

15.5.2 MDMA 安全全局中断状态寄存器 (MDMA_SGISR0)

MDMA secure global interrupt status register

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GIF31	GIF30	GIF29	GIF28	GIF27	GIF26	GIF25	GIF24	GIF23	GIF22	GIF21	GIF20	GIF19	GIF18	GIF17	GIF16
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GIF15	GIF14	GIF13	GIF12	GIF11	GIF10	GIF9	GIF8	GIF7	GIF6	GIF5	GIF4	GIF3	GIF2	GIF1	GIF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **GIF[31:0]**：通道 x 全局中断标志（x = 31 到 0）

该位字段由硬件置 1 和复位。它是在 MDMA_CTCIEx、MDMA_BTIEx、MDMA_BRTIEx 和 MDMA_TEIEx 中使能的所有通道 x 中断标志（CTCIF、BTIF、BRTIF、TEIF）的逻辑或运算结果。

0：通道 x 未生成中断

1：通道 x 生成了中断

15.5.3 MDMA 通道 x 中断状态寄存器 (MDMA_CxISR)

MDMA channel x interrupt status register

偏移地址：0x40 + 0x40 * x，（x = 0 到 31）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRQA
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCIF	BTIF	BRTIF	CTCIF	TEIF
											r	r	r	r	r

位 31:17 保留，必须保持复位值。

位 16 **CRQA**：通道 x 请求激活标志 (Channel x request active flag)

由软件通过向 MDMA_CxCR 中的 SWRQx 写入 1 来将该位置 1，以请求 MDMA 传输，且使能通道 x。

通道请求激活且使能通道时，该位还可由硬件置 1。硬件请求在处理前，始终被存储。

通道 x 请求完成时（因当前请求而进行最后一个缓冲区传输的源写入阶段后），该位由硬件清零。

0：对于通道 x，MDMA 传输 mdma_strx 未激活。

1：对于通道 x，MDMA 传输 mdma_strx 激活

注意：该位还会在通道被禁止时（发生传输错误，或通道数据传输结束——重复块 = 0，链表指针为空——或在结束前由软件将通道使能位编程为 0 时）由硬件复位。

位 15:5 保留，必须保持复位值。

位 4 **TCIF**：通道 x 缓冲区传输完成中断标志 (Channel x buffer transfer complete interrupt flag)

该位由硬件置 1，由软件清零，软件只需将 1 写入 MDMA_IFCRy 的相应位。

0：通道 x 上无缓冲区传输完成事件

1：通道 x 上发生缓冲区传输完成事件

传输单缓冲区时 TC 置 1。它将在发出每个通道传输请求时激活。

这可用作调试功能（无中断），指示自上次标志复位起（至少）已生成一个 MDMA 缓冲区传输。

位 3 **BTIF**：通道 x 块传输完成中断标志 (Channel x block transfer complete interrupt flag)

该位由硬件置 1，由软件清零，软件只需将 1 写入 MDMA_IFCRy 的相应位。

0：通道 x 上无块传输完成事件

1：通道 x 上发生块传输完成事件。

位 2 **BRTIF**：通道 x 块重复传输完成中断标志 (Channel x block repeat transfer complete interrupt flag)

该位由硬件置 1，由软件清零，软件只需将 1 写入 MDMA_IFCRy 的相应位。

0：通道 x 上无块重复传输完成事件

1：通道 x 上发生块重复传输完成事件

位 1 **CTCIF**：通道 x 通道传输完成中断标志 (Channel x channel transfer complete interrupt flag)

该位由硬件置 1，由软件清零，软件只需将 1 写入 MDMA_IFCRy 的相应位。

0：通道 x 上无通道传输完成事件

1：通道 x 上发生通道传输完成事件

最后一个块已传输且通道已自动禁止时 CTC 置 1。

当通道因将 EN 位写 0 而暂停传输时，CTC 也置 1。

位 0 **TEIF**：通道 x 传输错误中断标志 (Channel x transfer error interrupt flag)
该位由硬件置 1，由软件清零，软件只需将 1 写入 MDMA_IFCRy 的相应位。
0：数据流 x 上无传输错误
1：数据流 x 上发生传输错误

15.5.4 MDMA 通道 x 中断标志清零寄存器 (MDMA_CxIFCR)

MDMA channel x interrupt flag clear register

偏移地址：0x44 + 0x40 * x，（x = 0 到 31）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLTCIF	CBTIF	CBRTIF	CCTCIF	CTEIF
											w	w	w	w	w

位 31:5 保留，必须保持复位值。

位 4 **CLTCIF**：通道 x 的清零缓冲区传输完成中断标志 (Clear buffer transfer complete Interrupt flag for channel x)
向该位写入 1 可将 MDMA_ISRy 中的 TCIF 清零。

位 3 **CBTIF**：通道 x 清零块传输完成中断标志 (Channel x clear block transfer complete interrupt flag)
向该位写入 1 可将 MDMA_ISRy 中的 BTIF 清零。

位 2 **CBRTIF**：通道 x 清零块重复传输完成中断标志 (Channel x clear block repeat transfer complete interrupt flag)
向该位写入 1 可将 MDMA_ISRy 中的 BRTIF 清零。

位 1 **CCTCIF**：通道 x 的清零通道传输完成中断标志 (Clear channel transfer complete interrupt flag for channel x)
向该位写入 1 可将 MDMA_ISRy 中的 CTCIF 清零。

位 0 **CTEIF**：通道 x 清零传输错误中断标志 (Channel x clear transfer error interrupt flag)
向该位写入 1 可将 MDMA_ISRy 中的 TEIF 清零。

15.5.5 MDMA 通道 x 错误状态寄存器 (MDMA_CxESR)

MDMA channel x error status register

偏移地址：0x48 + 0x40 * x，（x = 0 到 31）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	BSE	ASE	TEMD	TELD	TED	TEA[6:0]						
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12 保留，必须保持复位值。

位 11 **BSE**：块大小错误 (Block size error)

块大小不是源或目标的数据大小的整数倍时，该位由硬件置 1。TED 将指示问题是出在源上还是出在目标上。通过软件向 MDMA_IFCRy 中的 CTEIF 写入 1 可将此位清零。

0：无块大小错误

1：编程的块大小不是数据大小的整数倍。

位 10 **ASE**：地址大小错误 (Address size error)

设置的地址与数据大小不匹配时，该位由硬件置 1。TED 将指示问题是出在源上还是出在目标上。通过软件向 MDMA_IFCRy 中的 CTEIF 写入 1 可将此位清零。

0：无地址大小错误

1：设置的地址与数据大小不匹配。

位 9 **TEMD**：传输错误掩码数据 (Transfer error mask data)

写入掩码数据发生传输错误时，该位由硬件置 1。通过软件向 MDMA_IFCRy 中的 CTEIF 写入 1 可将此位清零。

0：无掩码写入访问错误

1：通道上最后一次传输错误与掩码数据的写入有关。

位 8 **TELD**：传输错误链路数据 (Transfer error link data)

如果在读取块链路数据结构时发生传输错误，则该位由硬件置 1。通过软件向 MDMA_IFCRy 中的 CTEIF 写入 1 可将此位清零。

0：无链路数据读取访问错误

1：通道上最后一次传输错误与链路数据结构的读取有关。

位 7 **TED**：传输错误方向 (Transfer error direction)

在出现 MDMA 数据传输错误时，该位由硬件置 1 和清零。

0：通道上最后一次传输错误与读取访问有关。

1：通道上最后一次传输错误与写入访问有关。

位 6:0 **TEA[6:0]**：传输错误地址 (Transfer error address)

在出现 MDMA 数据传输错误时，这些位由硬件置 1 和清零。它们可与 TED 搭配使用。该字段指示生成传输/访问错误的地址的 7 个 LSB 位。

软件可以用这个字段来检索失效地址，方法是将该值（截取至缓冲区传输长度大小）与当前 SAR/DAR 值相加。

注意：因为存在 FIFO 管理系统，所以当前 SAR/DAR 值反映的并不是数据传输的最后一个地址。SAR/DAR 仅在结束一次缓冲区传输（长度为 TLEN+1 字节）后进行更新。

出现链路数据错误时该字段不会置 1。

15.5.6 MDMA 通道 x 控制寄存器 (MDMA_CxCR)

MDMA channel x control register

此寄存器用于控制相关通道。

偏移地址：0x4C + 0x40 * x，（x = 0 到 31）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWRQ
															w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	WEX	HEX	BEX	Res.	Res.	Res.	SM	PL[1:0]		TCIE	BTIE	BRTIE	CTCIE	TEIE	EN
	rw	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **SWRQ**：软件请求 (Software request)

向该位写入 1 会将 MDMA_ISRy 的 CRQA 置 1，这将激活通道 x 的请求。

注意：整个 MDMA_CxCR 寄存器或偏移地址为 0x4E + 0x40 × 通道编号的 8/16 位寄存器可用于 SWRQ 激活。

对于软件请求，不会产生一个回应（既没有硬件信号，也没有 MDMA_CxMAR 写入访问）。

位 15 保留，必须保持复位值。

位 14 **WEX**：字的字节序交换 (Word endianness exchange)

该位由软件置 1 和清零，

0：为字保留小端字节序

1：在双字中交换字顺序

该位置 1 时，会保留目标双字中的字顺序：较高地址字包含从源的较低地址中读取的数据。

如果目标不是双字，则该位的值无关紧要。

注意：此位受到保护，只有 EN 为 0 时才可以写入。

位 13 **HEX**：半字的字节序交换 (Half-word endianness exchange)

该位由软件置 1 和清零，

0：为半字保留小端字节序

1：在各个字中交换半字顺序

该位置 1 时，会保留各个目标字中的半字顺序：较高地址半字包含从源的较低地址中读取的数据。

如果目标长度短于字的长度，则该位的值无关紧要。

注意：此位受到保护，只有 EN 为 0 时才可以写入。

位 12 **BEX**：字节的字节序交换 (Byte endianness exchange)

该位由软件置 1 和清零，

0：为字节保留小端字节序

1：在各个半字中交换字节顺序

该位置 1 时，会保留各个目标半字中的字节顺序：较高地址字包含从源的较低地址中读取的数据。

如果目标为字节，则该位的值无关紧要。

注意：此位受到保护，只有 EN 为 0 时才可以写入。

位 11:9 保留，必须保持复位值。

位 8 **SM**：安全模式使能 (Secure mode enable)

该位由软件置 1 和清零，仅当 AHB 从端口处于安全模式时，才能写入该位。

0：禁止安全模式

1：安全模式使能 (Secure mode enable)

当此位读为 0 时，软件可对与当前通道相关的配置位进行编程，而无需考虑 AHB 从接口是否处于安全模式。

当此位置 1 时，如果 AHB 从端口的安全模式未激活，则当前通道的所有寄存器都将受到写保护。所有 MDMA 主设备事务都在安全模式下完成。

注意：此位受到保护，只有 EN 为 0 时才可以写入。

位 7:6 **PL[1:0]**：优先级 (Priority level)

这些位由软件置 1 和清零，

00：低

01：中

10：高

11：非常高

注意：这些位受到保护，只有 EN 为 0 时才可以写入。

位 5 **TCIE**：缓冲区传输完成中断使能 (Buffer transfer complete interrupt enable)

该位由软件置 1 和清零，

0：禁止 TC 中断

1：使能 TC 中断

位 4 **BTIE**：块传输中断使能 (Block Transfer interrupt enable)

该位由软件置 1 和清零，

0：禁止 BT 完成中断

1：使能 BT 完成中断

位 3 **BRTIE**：块重复传输中断使能 (Block repeat transfer interrupt enable)

该位由软件置 1 和清零，

0：禁止 BT 中断

1：使能 BT 中断

位 2 **CTCIE**：通道传输完成中断使能 (Channel transfer complete interrupt enable)

该位由软件置 1 和清零，

0：禁止 TC 中断

1：使能 TC 中断

位 1 **TEIE**：传输错误中断使能 (Transfer error interrupt enable)

该位由软件置 1 和清零，

0：禁止 TE 中断

1：使能 TE 中断

位 0 **EN**：通道使能/读作低电平时通道就绪标志 (Channel enable/flag channel ready when read low)

该位由软件置 1 和清零，

0：禁止通道

1：使能通道

在以下情况下，此位可由硬件清零：

- MDMA 传输结束时（准备好配置数据流）
- AXI 主总线出现传输错误（总线错误/HardFault）时
- 出现其他错误条件（数据对齐、块/数据大小不兼容）时

该位由软件复位时，将完成正在进行的缓冲区传输（如有）。所有状态/配置寄存器都将保持其当前值。如果在未写入这些寄存器的情况下重新使能通道，则该通道将从中断点继续。

将该位读作 0 时，允许软件对配置寄存器进行编程。将 EN 位读作 1 时，禁止向这些寄存器执行写入操作（写入操作将被忽略）。

注意：该位由软件复位时，建议等待 CTCIF = 1，以确保在重新编程通道之前，正在进行的缓冲区传输均已完成。

15.5.7 MDMA 通道 x 传输配置寄存器 (MDMA_CxTCR)

MDMA channel x transfer configuration register

偏移地址：0x50 + 0x40 * x，（x = 0 到 31）

复位值：0x0000 0000

此寄存器用于配置相关通道。在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从内存进行加载（从当前 LAR[31:0] + 0x00 提供的地址进行加载）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BWM	SWRM	TRGM[1:0]		PAM[1:0]		PKE	TLEN[6:0]							DBURST[2:1]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBURST[0]	SBURST[2:0]			DINCOS[1:0]		SINCOS[1:0]		DSIZE[1:0]		SSIZE[1:0]		DINC[1:0]		SINC[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **BWM**：可缓冲写入模式 (Bufferable write mode)

该位由软件置 1 和清零，

0：目标写入操作不可缓冲。

1：目标写入操作可缓冲。

注意：此位受到保护，只有 EN 为 0 时才可以写入。

所有 MDMA 目标访问均不可缓存。

位 30 **SWRM**：软件请求模式 (Software request mode)

该位由软件置 1 和清零，如果硬件或软件请求处于激活状态，则位更改会被延迟，直至完成当前传输为止。

0：采用硬件请求考虑在内：传输的启动由 TRGM 值定义以及由 MDMA ACKx 信号确认。如果 MDMA_CxMAR 包含有效地址，则还会在 MDMA_CxMAR 地址处写入 MDMA_CxMDR 值。

1：硬件请求被忽略。由软件通过向 SWRQ 位写入 1 来触发传输。

注意：此位受到保护，只有 EN 为 0 时才可以写入。

位 29:28 **TRGM[1:0]**：触发模式

这些位由软件置 1 和清零，

00：每个 MDMA 请求（软件或硬件）都会触发缓冲区传输。

01：每个 MDMA 请求（软件或硬件）都会触发块传输。

10：每个 MDMA 请求（软件或硬件）都会触发重复块传输（如果块重复值为 0，则会传输单块。

11：每个 MDMA 请求（软件或硬件）都会触发针对相应通道的完整数据（例如链表）传输，直至通道结束传输并被禁止。

注意：如果当前块的 TRGM 为 11，则在当前块末尾载入的所有值均必须通过链表机制保持相同值 (TRGM=11) 以及相同的 SWRM 值，否则结果是不确定的。

这些位受到保护，只有 EN 为 0 时才可以写入。

位 27:26 **PAM[1:0]**：填充/对齐模式 (Padding/alignment mode)

这些位由软件置 1 和清零，

00：右对齐，用 0 进行填充（默认值）。如果源数据大于目标大小，则系统仅会将源的 LSB 部分写入到目标地址。其余部分将被丢弃。

01：右对齐，符号扩展

10：左对齐（用 0 进行填充）。如果源数据大于目标大小，则系统仅会将源的 MSB 部分写入到目标地址。其余部分将被丢弃。

11：保留

注意：PKE = 1 或 DSIZE=SSIZE 时，将忽略这些位。

这些位受到保护，只有 EN 为 0 时才可以写入。

位 25 **PKE**：封装使能 (Pack enable)

该位由软件置 1 和清零，

0：按原样将源数据写入到目标中。如果源大小小于目标大小，则会根据 PAM 值进行填充。如果源数据大小大于目标数据大小，则会将其截断。将根据 PAM[1:0] 值进行对齐。

1：将源数据封装/解封为目标数据大小。所有数据均右对齐（采用小端模式）。

注意：此位受到保护，只有 EN 为 0 时才可以写入。

位 24:18 **TLEN[6:0]**：缓冲区传输长度 (Buffer transfer length)（字节数 - 1）

这些位由软件置 1 和清零，TLEN + 1 值表示要在单次传输过程中传输的字节数。

传输长度必须为（源和目标）数据大小的倍数

源/目标大小不同且使用填充/截断时，TLEN + 1 指的是源数据数组大小。

注意：这些位受到保护，只有 EN 为 0 时才可以写入。

必须对 DBURST 值进行编程，以确保突发大小小于传输大小。

位 17:15 **DBURST[2:0]**：目标突发传输配置 (Destination burst transfer configuration)

这些位由软件置 1 和清零，

000：单次传输

N：2^N 个节拍的突发

必须对 DBURST 值进行编程，以确保突发大小小于传输长度。如果无法保证这一点，将无法预测结果。

注意：这些位受到保护，只有 EN 为 0 时才可以写入。

如果 DINC = 00，则 DBURST 必须为最大值 100（16 个节拍的突发），否则将无法预测结果。

位 14:12 **SBURST[2:0]**：源突发传输配置 (Source burst transfer configuration)

这些位由软件置 1 和清零，

000：单次传输

N：2^N 个节拍的突发

必须对 SBURST 值进行编程，以确保突发大小小于传输长度。如果无法保证这一点，将无法预测结果。

注意：这些位受到保护，只有 EN 为 0 时才可以写入。

如果 SINC = 00，则 SBURST 必须为最大值 100（16 个节拍的突发），否则将无法预测结果。

位 11:10 **DINCOS[1:0]**：目标增量偏移量 (Destination increment offset size)

这些位由软件置 1 和清零，

00：字节（8 位）

01：半字（16 位）

10：字（32 位）

11：双字（64 位）-

如果位 DINC[1:0] = 00，则这些位没有意义。

注意：如果 DINCOS < DSIZE 且 DINC ≠ 00，将无法预测结果。

这些位受到保护，只有 EN = 0 时才可以写入。

位 9:8 **SINCOS[1:0]**：源增量偏移量 (Source increment offset size)

这些位由软件置 1 和清零，

00：字节（8 位）

01：半字（16 位）

10：字（32 位）

11：双字（64 位）-

如果位 SINC[1:0] = 00，则这些位没有意义。

注意：如果 SINCOS < SSIZE 且 SINC ≠ 00，将无法预测结果。

这些位受到保护，只有 EN = 0 时才可以写入。

位 7:6 **DSIZE[1:0]**：目标数据大小 (Destination data size)

这些位由软件置 1 和清零，

00：字节（8 位）

01：半字（16 位）

10：字（32 位）

11：双字（64 位）

注意：如果 DINCOS < DSIZE 且 DINC ≠ 00，将无法预测结果。

这些位受到保护，只有 EN = 0 时才可以写入。

位 5:4 **SSIZE[1:0]**：源数据大小 (Source data size)

这些位由软件置 1 和清零，

00：字节（8 位）

01：半字（16 位）

10：字（32 位）

11：双字（64 位）

注意：如果 SINCOS < SSIZE 且 SINC ≠ 00，将无法预测结果。

这些位受到保护，只有 EN 为 0 时才可以写入。

位 3:2 **DINC[1:0]**：目标增量模式 (Destination increment mode)

这些位由软件置 1 和清零，

00：目标地址指针固定。

10：每次数据传输后，目标地址指针递增（增量为 DINCOS 值）。

11：每次数据传输后，目标地址指针递减（减量为 DINCOS 值）。

这些位受到保护，只有 EN = 0 时才可以写入。

位 1:0 **SINC[1:0]**：源增量模式 (Source increment mode)

这些位由软件置 1 和清零，

00：源地址指针固定。

10：每次数据传输后，源地址指针递增（增量为 SINCOS 值）。

11：每次数据传输后，源地址指针递减（减量为 SINCOS 值）。

这些位受到保护，只有 EN = 0 时才可以写入。

15.5.8 MDMA 通道 x 块数据数寄存器 (MDMA_CxBNDTR)

MDMA channel x block number of data register

偏移地址：0x54 + 0x40 * x，（x = 0 到 31）

复位值：0x0000 0000

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从内存进行加载（从当前 LAR[31:0] + 0x04 提供的地址进行加载）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRC[11:0]												BRDUM	BRSUM	Res.	BNDT[16]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BNDT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:20 **BRC[11:0]**：块重复计数 (Block repeat count)

该字段包含当前块的重复数（0 到 4095）。使能通道后，此寄存器为只读型，用于指示除当前块之外的剩余块数。每次完成块传输后，此寄存器将递减。

最后一个块传输完成后，该寄存器既可保持为零，也可自动从内存重载（在链表模式下，意味着链路地址有效）。

注意：这些位受到保护，只有 EN = 0 时才可以写入。

位 19 **BRDUM**：块重复目标地址更新模式 (Block repeat destination address update mode)

0：块传输结束后，通过将 DUV 与当前 DAR 值（当前目标地址）相加来更新 MDMA_DAR 寄存器。

1：块传输结束后，通过从当前 DAR 值（当前目标地址）中减去 DUV 来更新 MDMA_DAR 寄存器。

注意：这些位受到保护，只有 EN = 0 时才可以写入。

位 18 **BRSUM**：块重复源地址更新模式 (Block repeat source address update mode)

0：块传输结束后，通过将 SUV 与当前 SAR 值（当前源地址）相加来更新 MDMA_SAR 寄存器。

1：块传输结束后，通过从当前 SAR 值（当前源地址）中减去 SUV 来更新 MDMA_SAR 寄存器。

注意：这些位受到保护，只有 EN = 0 时才可以写入。

位 17 保留，必须保持复位值。

位 16:0 **BNDT[16:0]**：块中待传输的数据字节数 (Block number of data bytes to transfer)

当前块中要传输的字节数（0 到 65536）。使能通道后，此寄存器为只读型，用于指示要传输的剩余数据项数。通道活动期间，该寄存器递减，指示当前块中剩余的数据项数。

块传输完成后，该寄存器可保持为零，或者，如果已将通道配置为块重复模式，则自动重载先前编程的值。

如果该寄存器的值为零，则即使使能数据流，也无法完成任何事务。

注意：如果 BNDT 值不是 TLEN + 1 值的整数倍，则最后一次传输将变短，仅包含块中的剩余数据。

块大小必须是源和目标数据大小的倍数。否则，会出现错误，且不会写入任何数据

这些位受到保护，只有 EN = 0 时才可以写入。

15.5.9 MDMA 通道 x 源地址寄存器 (MDMA_CxSAR)

MDMA channel x source address register

偏移地址：0x58 + 0x40 * x，（x = 0 到 31）

复位值：0x0000 0000

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从内存进行加载（从当前 LAR[31:0] + 0x08 提供的地址进行加载）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **SAR[31:0]**：源地址

这些位表示从其读取数据或将数据读取至的外设数据寄存器的基址。其必须与 SSIZE 对齐（SSIZE=10 时 SAR[1:0] = 00），但可不与 SINCOS 对齐。

在通道活动期间该寄存器将更新，反映了下次要从中读取数据的当前地址。

如果块重复模式激活，则在完成块传输后，通过将 SAU 值与当前值相加/相减来更新源地址（已在块中的最后一次传输后进行更新）。

链表模式激活时，在块（重复块或非重复块）传输结束后，将从内存（从地址 LSA + m）加载 SAR 值。

注意：这些位受到写保护，只有 EN 位为 0 时才可以写入。

15.5.10 MDMA 通道 x 目标地址寄存器 (MDMA_CxDAR)

MDMA channel x destination address register

偏移地址：0x5C + 0x40 * x，（x = 0 到 31）

复位值：0x0000 0000

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从内存进行加载（从当前 LAR[31:0] + 0x0C 提供的地址进行加载）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DAR[31:0]**：目标地址

要向其中写入数据的目标地址的基址。必须与 DSIZE 对齐（例如 DSIZE=01 时 DAR[0] = 0），但可不与 DINCOS 对齐。

在通道活动期间该寄存器将更新，反映了下次要向其中写入数据的当前地址。

如果块重复模式激活，则在完成块传输后，通过将 DAU 值与当前值相加/相减来更新目标地址（已在块中的最后一次传输后进行更新）。

链表模式激活时，在块（重复块或非重复块）传输结束后，将从内存（从地址 LSA + m）加载 DAR 值。

注意：这些位受到写保护，只有 EN 位为 0 时才可以写入。

15.5.11 MDMA 通道 x 块重复地址更新寄存器 (MDMA_CxBRUR)

MDMA channel x block repeat address update register

偏移地址：0x60 + 0x40 * x，（x = 0 到 31）

复位值：0x0000 0000

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从内存进行加载（从当前 LAR[31:0] + 0x10 提供的地址进行加载）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DUV[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUV[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **DUV[15:0]**：目标地址更新值 (Destination address update value)

该值用于在块传输结束后更新（通过加法或减法）当前目标地址。该值必须为 DSIZE 的整数倍，以便使 DAR 与 DSIZE 对齐（DSIZE=10 时 DAR[1:0] = 00）。

如果该值为 0，下次重复的块传输的目标地址将从下一地址继续。

块重复模式未激活 (BRC=0) 时，将忽略该字段。

注意：这些位受到写保护，只有 EN 位为 0 时才可以写入。

DINC[1:0] = 00 时必须将该字段编程为 0。

位 15:0 **SUV[15:0]**：源地址更新值 (Source address update value)

该值用于在块传输结束后更新（通过加法或减法）当前源地址。该值必须为 SSIZE 的整数倍，以便使 SAR 与 SSIZE 对齐（SSIZE = 10 时 SAR[1:0] = 00）。

如果该值为 0，下次重复的块传输的源地址将从下一地址继续。

块重复模式未激活 (BRC=0) 时，将忽略该字段。

注意：这些位受到写保护，只有 EN 位为 0 时才可以写入。

SINC[1:0] = 00 时必须将该字段编程为 0。

15.5.12 MDMA 通道 x 链路地址寄存器 (MDMA_CxLAR)

MDMA channel x link address register

偏移地址：0x64 + 0x40 * x，（x = 0 到 31）

复位值：0x0000 0000

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从内存进行加载（从当前 LAR[31:0] + 0x14 提供的地址进行加载）。

注： *只有在所有寄存器均更新后，才会为下次块结束采用新值。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LAR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LAR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	r

位 31:0 **LAR[31:0]**：链路地址寄存器 (Link address register)

（重复）块传输结束后，当前通道配置寄存器（MDMA_CxTCR、MDMA_CxBNDTR、MDMA_CxSAR、MDMA_CxDAR、MDMA_CxBRUR、MDMA_CxMAR、MDMA_CxMDR 和 MDMA_CxLAR 本身）使用该地址处的数据结构进行加载。

如果该寄存器的值为 0，则不会进行寄存器更新；系统将禁止该通道并将 CTCIF 置 1，表示此通道的传输已结束。

通道配置（LAR 地址）必须处于 AXI 地址空间内。

LAR 值必须在双字地址处匹配 (LAR[2:0] = 0x0)。

注意：这些位受到写保护，只有 EN 位为 0 时才可以写入。

15.5.13 MDMA 通道 x 触发和总线选择寄存器 (MDMA_CxTBR)

MDMA channel x trigger and bus selection register

偏移地址：0x68 + 0x40 * x，（x = 0 到 31）

复位值：0x0000 0000

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从内存进行加载（从当前 LAR[31:0] + 0x18 提供的地址进行加载）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSEL[5:0]					
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持复位值。

位 5:0 **TSEL[5:0]**：触发选择 (Trigger selection)

该字段用于为通道 x 选择硬件触发 (RQ) 输入。在具有相同索引值的 ACK 输出上发送 ACK。

如果 SWRM 位置 1（已选择软件请求），则会忽略该字段。

注意：这些位受到写保护，只有 EN 位为 0 时才可以写入。

如果同一事件触发多个通道（具有相同 TSEL 值），则会并行触发所有这些通道。只有具有最低索引的通道才会确认请求（通过软件或硬件确认）。

15.5.14 MDMA 通道 x 掩码地址寄存器 (MDMA_CxMAR)

MDMA channel x mask address register

偏移地址：0x70 + 0x40 * x，（x = 0 到 31）

复位值：0x0000 0000

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从内存进行加载（从当前 LAR[31:0] + 0x20 提供的地址进行加载）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **MAR[31:0]**：掩码地址 (Mask address)

当生成 ACK 信号时，系统也会将 MDR 值写入此地址。可将其用于通过写入中断清零寄存器来清除 DMA2 生成的 RQ 信号。

如果该寄存器的值为 0，则会禁止该功能。

注意：这些位受到写保护，只有 EN 位为 0 时才可以写入。

15.5.15 MDMA 通道 x 掩码数据寄存器 (MDMA_CxMDR)

MDMA channel x mask data register

偏移地址：0x74 + 0x40 * x，（x = 0 到 31）

复位值：0x0000 0000

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从内存进行加载（从当前 LAR[31:0] + 0x24 提供的地址进行加载）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MDR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MDR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **MDR[31:0]**：掩码数据 (Mask data)

当生成 ACK 信号时，系统也会将 MDR 值写入 MDMA_MAR 寄存器定义的地址。可将其用于通过写入中断清零寄存器来清除 DMA2 生成的 RQ 信号。

注意：这些位受到写保护，只有 EN 位为 0 时才可以写入。

15.5.16 MDMA 寄存器映射

表 92. MDMA 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	MDMA_GISR0	GIF31	GIF30	GIF29	GIF28	GIF27	GIF26	GIF25	GIF24	GIF23	GIF22	GIF21	GIF20	GIF19	GIF18	GIF17	GIF16	GIF15	GIF14	GIF13	GIF12	GIF11	GIF10	GIF9	GIF8	GIF7	GIF6	GIF5	GIF4	GIF3	GIF2	GIF1	GIF0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	Reserved	Reserved																															
0x08	MDMA_SGISR0	GIF31	GIF30	GIF29	GIF28	GIF27	GIF26	GIF25	GIF24	GIF23	GIF22	GIF21	GIF20	GIF19	GIF18	GIF17	GIF16	GIF15	GIF14	GIF13	GIF12	GIF11	GIF10	GIF9	GIF8	GIF7	GIF6	GIF5	GIF4	GIF3	GIF2	GIF1	GIF0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C - 0x3C	Reserved	Reserved																															
0x40 + 0x40 * x (x=0 to 31)	MDMA_CxISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCIF	BTIF	BRTIF	CTCIF	TEIF
	Reset value																												0	0	0	0	0
0x44 + 0x40 * x (x=0 to 31)	MDMA_CxIFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLTCIF	CBTIF	CBRTIF	CCTCIF	CTEIF
	Reset value																												0	0	0	0	0
0x48 + 0x40 * x (x=0 to 31)	MDMA_CxESR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BSE	ASE	TEMD	TELD	TED	TEA[6:0]						
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x4C + 0x40 * x (x=0 to 31)	MDMA_CxCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWRQ	Res.	WEX	HEX	BEX	Res.	Res.	Res.	SM	PL[1:0]		TCIE	BTIE	BRTIE	CTCIE	TEIE	EN
	Reset value																0		0	0	0				0	0	0	0	0	0	0	0	0
0x50 + 0x40 * x (x=0 to 31)	MDMA_CxTCR	BWM	SWRM	TRGM [1:0]		PAM[1:0]		PKE	TLEN[6:0].								DBURST [2:0]			SBURST [2:0]		DINCOS [1:0]		SINCOS [1:0]		DSIZE [1:0]		SSIZE [1:0]		DINC [1:0]		SINC [1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 92. MDMA 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x54 + 0x40 * x (x=0 to 31)	MDMA_CxBNDTR	BRC[11:0]												BRDUM	BRSUM	Res.	BNDT[16:0]																
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x58 + 0x40 * x (x=0 to 31)	MDMA_CxSAR	SAR[31:0].																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5C + 0x40 * x (x=0 to 31)	MDMA_CxDAR	DAR[31:0].																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60 + 0x40 * x (x=0 to 31)	MDMA_CxBRUR	DUV[15:0].															SUV[15:0].																
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x64 + 0x40 * x (x=0 to 31)	MDMA_CxLAR	LAR[31:0].																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x68 + 0x40 * x (x=0 to 31)	MDMA_CxTBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSEL[5:0]					
	Reset value																											0	0	0	0	0	0
0x6C+ 0x40 * x (x=0 to 31)	Reserved	Reserved																															
0x70 + 0x40 * x (x=0 to 31)	MDMA_CxMAR	MAR[31:0].																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x74 + 0x40 * x (x=0 to 31)	MDMA_CxMDR	MDR[31:0].																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 2.5 节*。

16 直接内存访问控制器 (DMA)

16.1 DMA 简介

直接内存访问 (DMA) 用于在外设与内存之间以及内存与内存之间提供高速数据传输。可以在无需任何 CPU 操作的情况下通过 DMA 快速移动数据。这样节省的 CPU 资源可供其他操作使用。

DMA 控制器基于复杂的总线矩阵架构，将功能强大的双 AHB 主总线架构与独立的 FIFO 结合在一起，优化了系统带宽。

这三个 DMA 控制器（DMA1、DMA2、DMA3）各有 8 个数据流，专门用于管理来自一个或多个外设的内存访问请求。

任何 DMA1 或 DMA2 数据流都由一个 DMAMUX1 输出通道（请求）进行驱动。任何 DMA3 数据流都由一个 DMAMUX2 输出通道（请求）进行驱动。

安全外设 DMA 请求将通过安全 DMAMUX2 路由到安全 DMA3。当受控安全外设配置为安全时，外设 DMA 请求将处于安全状态，如 ETZPC 寄存器的相应 DECPROT 位字段所定义。

非安全外设 DMA 请求将通过非安全 DMAMUX1 路由到非安全 DMA1 和 DMA2。

用户可对任何 DMAMUX 输出请求进行单独编程，以便从任何可用的请求输入信号中选择 DMA 请求源信号。

有关 DMA 请求和数据流映射的更多信息，请参见*第 17.3 节：DMAMUX 实现*。

每个 DMA 控制器都有一个仲裁器，用于处理 DMA 请求间的优先级。

16.2 DMA 主要特性

DMA 主要特性是：

- 双 AHB 主总线架构，一个用于内存访问，另一个用于外设访问
- 仅支持 32 位访问的 AHB 从编程接口
- 每个 DMA 控制器有 8 个数据流，DMAMUX1 上的每个数据流有多达 116 个通道（请求），而 DMAMUX2 上有多达 40 个通道（请求）
- 每个数据流有四个字深的 32 位先进先出内存缓冲区 (FIFO)，可用于 FIFO 模式或直接模式：
  - FIFO 模式：可通过软件将阈值级别选取为 FIFO 大小的 1/4、1/2 或 3/4
  - 直接模式：每个 DMA 请求会立即启动对内存的传输。当在直接模式（禁止 FIFO）下将 DMA 请求配置为以内存到外设模式传输数据时，DMA 仅会将一个数据从内存预加载到内部 FIFO，从而确保一旦外设触发 DMA 请求时则立即传输数据。
- 可以将每个数据流配置为：
  - 支持外设到内存、内存到外设和内存到内存传输的常规通道
  - 在内存端支持双缓冲的双缓冲区通道
- DMA 数据流请求之间的优先级可用软件编程（四个级别：非常高、高、中、低），在软件优先级相同的情况下可以通过硬件决定优先级（例如，请求 0 的优先级高于请求 1）
- 每个数据流还支持通过软件触发的内存间的传输
- 每个数据流请求都可以在所有可能的通道请求中进行选择。此选择可由 DMAMUX 通过软件配置，并支持在 DMAMUX1 上由 108 个外设，或是在 DMAMUX2 上由 32 个外设发起 DMA 请求

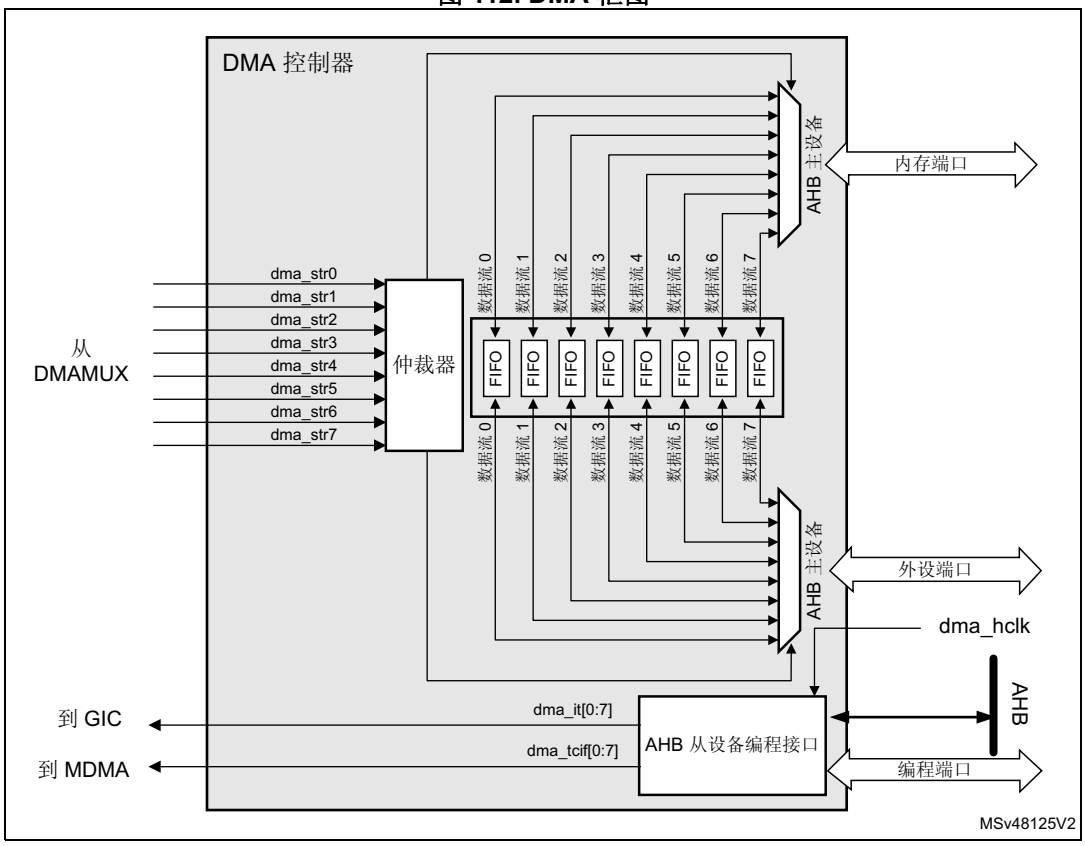
- 要传输的数据项的数目可以由 DMA 控制器或外设管理：
  - DMA 流控制器：要传输的数据项的数目可用软件编程，从 1 至 65535
  - 外设流控制器：要传输的数据项的数目未知并由源或目标外设控制，这些外设通过硬件发出传输结束的信号
- 独立的源和目标传输宽度（字节、半字、字）：源和目标的数据宽度不相等时，DMA 自动封装/解封必要的传输数据来优化带宽。这个特性仅在 FIFO 模式下可用
- 对源和目标的增量或非增量寻址
- 支持 4 个、8 个和 16 个节拍的增量突发传输。突发增量的大小可由软件配置，通常等于外设 FIFO 大小的一半
- 每个数据流都支持循环缓冲区管理
- 5 个事件标志（DMA 半传输、DMA 传输完成、DMA 传输错误、DMA FIFO 错误、直接模式错误），进行逻辑或运算，从而产生每个数据流的单个中断请求

16.3 DMA 功能描述

16.3.1 DMA 框图

下图显示了 DMA 的框图。

图 112. DMA 框图



16.3.2 DMA 概述

DMA 控制器执行直接内存传输：作为一个 AHB 主设备，DMA 控制器可以控制 AHB 总线矩阵来发起 AHB 传输。

DMA 控制器可以执行下列事务：

- 外设到内存的传输
- 内存到外设的传输
- 内存到内存的传输

DMA 控制器提供两个 AHB 主端口：AHB 内存端口（用于连接内存）和 AHB 外设端口（用于连接外设）。但是，要执行内存到内存的传输，AHB 外设端口必须也能访问内存。

AHB 从端口用于对 DMA 控制器进行编程（它仅支持 32 位访问）。

16.3.3 DMA 事务

DMA 事务由给定数目的数据传输序列组成。要传输的数据项的数目及其宽度（8 位、16 位或 32 位）可用软件编程。

每个 DMA 传输包含三项操作：

- 通过 DMA_SxPAR 或 DMA_SxM0AR 寄存器寻址，从外设数据寄存器或存储单元中加载数据
- 通过 DMA_SxPAR 或 DMA_SxM0AR 寄存器寻址，将加载的数据存储到外设数据寄存器或存储单元
- DMA_SxNDTR 计数器在数据存储结束后递减，该计数器中包含仍需执行的事务数

在产生事件后，外设会向 DMA 控制器发送请求信号。DMA 控制器根据通道优先级处理该请求。只要 DMA 控制器访问外设，DMA 控制器就会向外设发送确认信号。外设获得 DMA 控制器的确认信号后，便会立即释放其请求。一旦外设使请求失效，DMA 控制器就会释放确认信号。如有更多请求，外设可以启动下一个事务。

16.3.4 DMA 请求映射

*第 17.3 节：DMAMUX 实现*对从外设映射到 DMA 数据流的 DMA 请求进行了介绍。

16.3.5 仲裁器

仲裁器为两个 AHB 主端口（内存和外设端口）提供基于请求优先级的 8 个 DMA 数据流请求管理，并启动外设/内存访问序列。

优先级管理分为两个阶段：

- 软件：每个数据流优先级都可以在 DMA_SxCR 寄存器中配置。分为四个级别：
  - 非常高优先级
  - 高优先级
  - 中优先级
  - 低优先级
- 硬件：如果两个请求具有相同的软件优先级，则编号低的数据流优先于编号高的数据流。例如，数据流 2 的优先级高于数据流 4。

16.3.6 DMA 数据流

8 个 DMA 控制器数据流都能够提供源和目标之间的单向传输链路。

每个数据流配置后都可以执行：

- 常规类型事务：内存到外设、外设到内存或内存到内存的传输
- 双缓冲区类型事务：使用两个内存指针的双缓冲区传输（当 DMA 正在进行自/至缓冲区的读/写操作时，应用程序可以进行至/自其他缓冲区的写/读操作）。

要传输的数据量（多达 65535）可以编程，并与连接到外设 AHB 端口的外设（请求 DMA 传输）的源宽度相关。每个事务完成后，包含要传输的数据项总量的寄存器都会递减。

16.3.7 源、目标和传输模式

源传输和目标传输可在整个 4 GB 区域（地址在 0x0000 0000 和 0xFFFF FFFF 之间）寻址外设和内存。

传输方向使用 DMA_SxCR 寄存器中的 DIR[1:0] 位进行配置，有三种可能的传输方向：内存到外设、外设到内存或内存到内存。

下表介绍了相应的源和目标地址。

表 93. 源和目标地址

DMA_SxCR 寄存器的位 DIR[1:0]	方向	源地址	目标地址
00	外设到内存	DMA_SxPAR	DMA_SxM0AR
01	内存到外设	DMA_SxM0AR	DMA_SxPAR
10	内存到内存	DMA_SxPAR	DMA_SxM0AR
11	保留	-	-

当数据宽度（在 DMA_SxCR 寄存器的 PSIZE 或 MSIZE 位中编程）分别是半字或字时，写入 DMA_SxPAR 或 DMA_SxM0AR/M1AR 寄存器的外设或内存地址必须分别在字或半字地址的边界对齐。

外设到内存模式

图 113 介绍了这种模式。

使能这种模式（将 DMA_SxCR 寄存器中的位 EN 置 1）时，每次产生外设请求，数据流都会启动数据源到 FIFO 的传输。

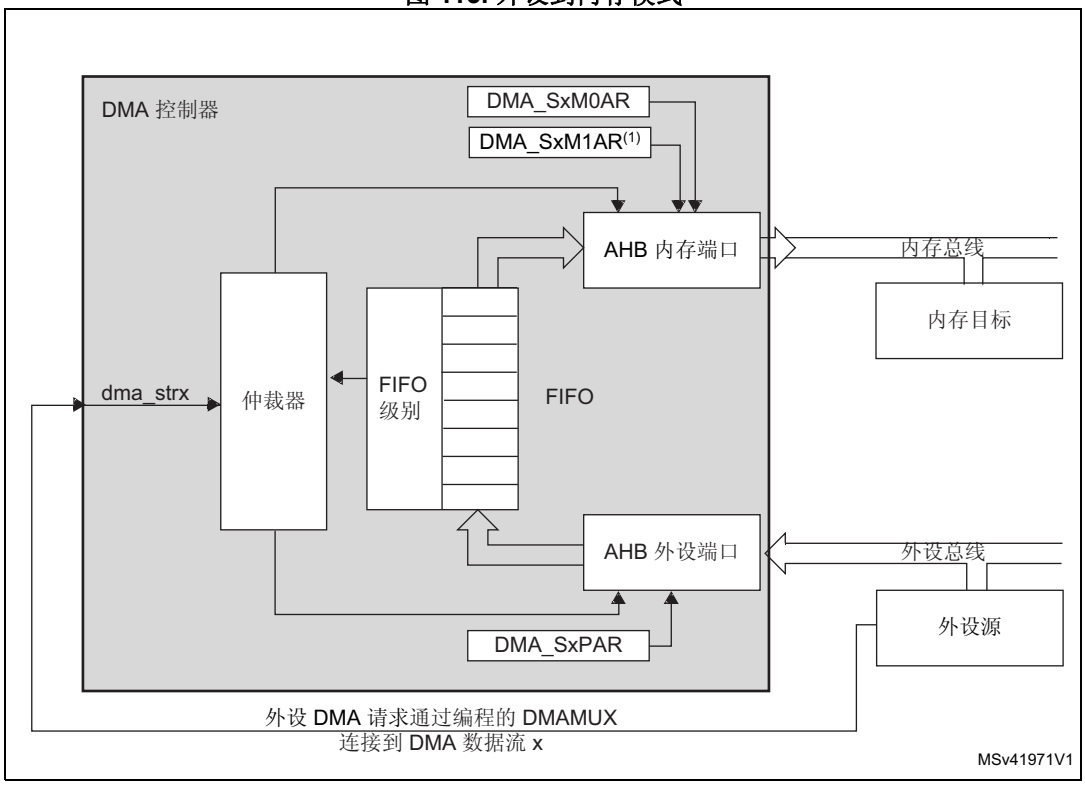
达到 FIFO 的阈值级别时，FIFO 的内容移出并存储到目标中。

如果 DMA_SxNDTR 寄存器达到零、外设请求传输终止（在使用外设流控制器的情况下）或 DMA_SxCR 寄存器中的 EN 位由软件清零，传输即会停止。

在直接模式下（当 DMA_SxFCR 寄存器中的 DMDIS 值为“0”时），不使用 FIFO 的阈值级别控制：每完成一次从外设到 FIFO 的数据传输后，相应的数据立即就会移出并存储到目标中。

只有赢得了数据流的仲裁后，相应数据流才有权访问 AHB 源或目标端口。系统使用在 DMA_SxCR 寄存器 PL[1:0] 位中为每个数据流定义的优先级执行仲裁。

图 113. 外设到内存模式



1. 用于双缓冲区模式。

内存到外设模式

图 114 介绍了这种模式。

使能这种模式（将 DMA_SxCR 寄存器中的 EN 位置 1）时，数据流会立即启动传输，从源完全填充 FIFO。

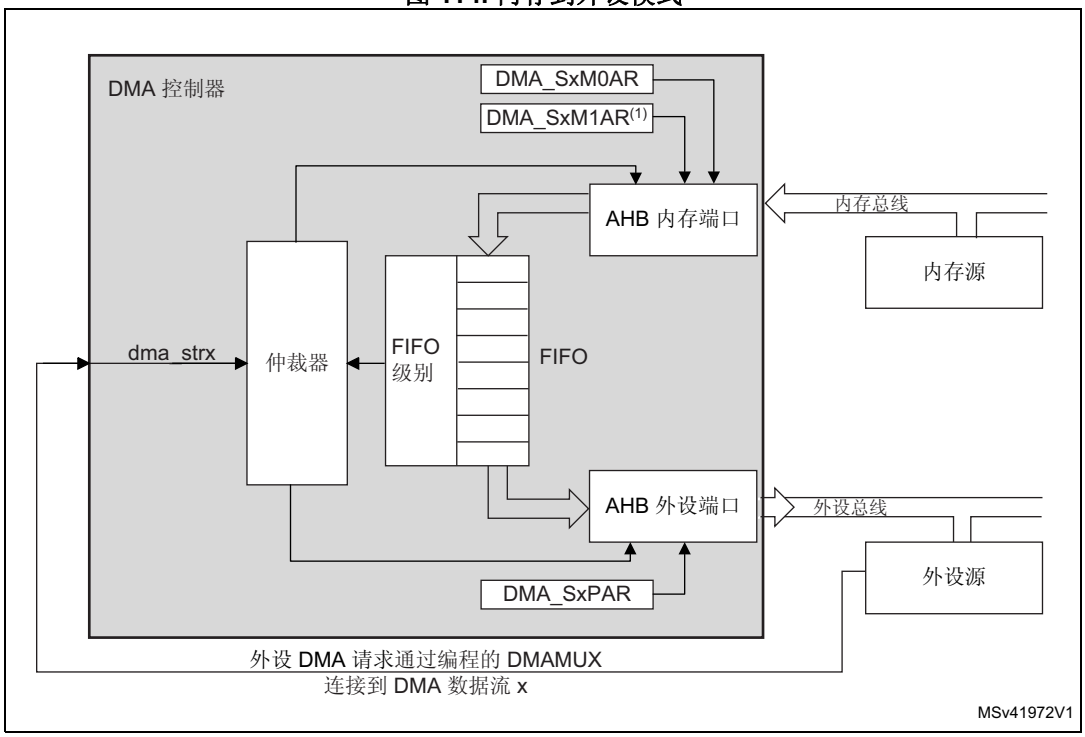
每次发生外设请求，FIFO 的内容都会移出并存储到目标中。当 FIFO 的级别小于或等于预定义的阈值级别时，将使用内存中的数据完全重载 FIFO。

如果 DMA_SxNDTR 寄存器达到零、外设请求传输终止（在使用外设流控制器的情况下）或 DMA_SxCR 寄存器中的 EN 位由软件清零，传输即会停止。

在直接模式下（当 DMA_SxFCR 寄存器中的 DMDIS 值为“0”时），不使用 FIFO 的阈值级别。一旦使能了数据流，DMA 便会预装载第一个数据，将其传输到内部 FIFO。一旦外设请求数据传输，DMA 便会将预装载的值传输到配置的目标。然后，它会使用要传输的下一个数据再次重载内部空 FIFO。预装载的数据大小为 DMA_SxCR 寄存器中 PSIZE 位字段的值。

只有赢得了数据流的仲裁后，相应数据流才有权访问 AHB 源或目标端口。系统使用在 DMA_SxCR 寄存器 PL[1:0] 位中为每个数据流定义的优先级执行仲裁。

图 114. 内存到外设模式



1. 用于双缓冲区模式。

内存到内存模式

DMA 通道在没有外设请求触发的情况下同样可以工作。此为*图 115* 中介绍的内存到内存模式。

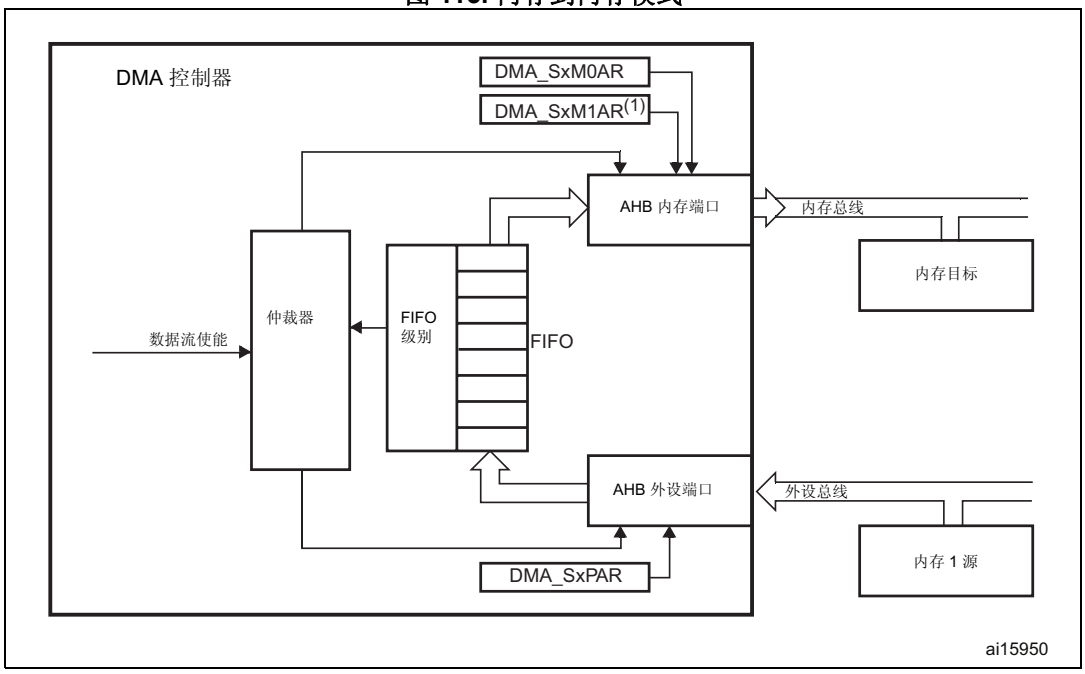
通过将 DMA_SxCR 寄存器中的使能位 (EN) 置 1 来使能数据流时，数据流会立即开始填充 FIFO，直至达到阈值级别。达到阈值级别后，FIFO 的内容便会移出，并存储到目标中。

如果 DMA_SxNDTR 寄存器达到零或 DMA_SxCR 寄存器中的 EN 位由软件清零，传输即会停止。

只有赢得了数据流的仲裁后，相应数据流才有权访问 AHB 源或目标端口。系统使用在 DMA_SxCR 寄存器 PL[1:0] 位中为每个数据流定义的优先级执行仲裁。

注： *使用内存到内存模式时，不允许循环模式和直接模式。*

图 115. 内存到内存模式



1. 用于双缓冲区模式。

16.3.8 指针递增

根据 DMA_SxCR 寄存器中 PINC 和 MINC 位的状态，外设和内存指针在每次传输后可以自动向后递增或保持常量。

通过单个寄存器访问外设源或目标数据时，禁止递增模式十分有用。

如果使能了递增模式，则根据在 DMA_SxCR 寄存器 PSIZE 或 MSIZE 位中编程的数据宽度，下一次传输的地址将是前一次传输的地址递增 1（对于字节）、2（对于半字）或 4（对于字）。

为了优化封装操作，可以不管 AHB 外设端口上传输的数据的大小，将外设地址的增量偏移大小固定下来。DMA_SxCR 寄存器中的 PINCOS 位用于将增量偏移大小与外设 AHB 端口或 32 位地址（此时地址递增 4）上的数据大小对齐。PINCOS 位仅对 AHB 外设端口有影响。

如果将 PINCOS 位置 1，则不论 PSIZE 值是多少，下一次传输的地址总是前一次传输的地址递增 4（自动与 32 位地址对齐）。但是，AHB 内存端口不受此操作影响。

16.3.9 循环模式 (Circular mode)

循环模式可用于处理循环缓冲区和连续数据流（例如 ADC 扫描模式）。可以使用 DMA_SxCR 寄存器中的 CIRC 位使能此特性。

当激活循环模式时，要传输的数据项的数目在数据流配置阶段自动用设置的初始值进行加载，并继续响应 DMA 请求。

注： *在循环模式下，如果为内存配置了突发模式，必须遵循下列规则：*

DMA_SxNDTR = ((Mburst 节拍) × (Msize)/(Psize)) 的倍数，其中：

- *(Mburst 节拍) = 4、8 或 16（取决于 DMA_SxCR 寄存器中的 MBURST 位）*
- *((Msize)/(Psize)) = 1、2、4、1/2 或 1/4（Msize 和 Psize 表示 DMA_SxCR 寄存器中的 MSIZE 和 PSIZE 位。它们与字节相关）*
- *DMA_SxNDTR = AHB 外设端口上要传输的数据项的数目*

例如：Mburst 节拍 = 8 (INCR8)，MSIZE = “00”（字节）和 PSIZE = “01”（半字），在此例中：DMA_SxNDTR 必须是 (8 × 1/2 = 4) 的倍数。

如果不遵循此公式，则 DMA 行为和数据完整性得不到保证。

NDTR 还必须是外设突发大小与外设数据大小乘积的倍数，否则会导致错误的 DMA 行为。

16.3.10 双缓冲区模式

此模式可用于所有 DMA1 和 DMA2 数据流。

通过将 DMA_SxCR 寄存器中的 DBM 位置1，即可使能双缓冲区模式。

除了有两个内存指针之外，双缓冲区数据流的工作方式与常规（单缓冲区）数据流的一样。使能双缓冲区模式时，将自动使能循环模式（DMA_SxCR 中的 CIRC 位的状态是“无关”），并在每次事务结束时交换内存指针。

在此模式下，每次事务结束时，DMA 控制器都从一个内存目标交换为另一个内存目标。这样，软件在处理一个存储区域的同时，DMA 传输还可以填充/使用第二个存储区域。如*表 94：双缓冲区模式下的源和目标地址寄存器 (DBM=1)* 所述，双缓冲区数据流可以双向工作（内存既可以是源也可以是目标）。

注： *在双缓冲区模式下使能数据流时，可遵循下列条件，实时更新 AHB 内存的基址*（DMA_SxM0AR 或 DMA_SxM1AR）*：*

- *当 DMA_SxCR 寄存器中的 CT 位为 0 时，可以写入 DMA_SxM1AR 寄存器。当 CT = 1 时，试图写入此寄存器会将错误标志位 (TEIF) 置 1，并自动禁止数据流。*
- *当 DMA_SxCR 寄存器中的 CT 位为 1 时，可以写入 DMA_SxM0AR 寄存器。当 CT = 0 时，试图写入此寄存器会将错误标志位 (TEIF) 置 1，并自动禁止数据流。*

为避免出现任何错误状态，建议在 TCIF 标志位置位时立即更改基址。因为此时根据上述两个条件之一，目标内存依据 DMA_SxCR 寄存器中 CT 值的情况，一定已从内存 0 更改为内存 1（或从内存 1 更改为内存 0）。

对于所有其他模式（双缓冲区模式除外），一旦使能数据流，内存地址寄存器即被写保护。

表 94. 双缓冲区模式下的源和目标地址寄存器 (DBM=1)

DMA_SxCR 寄存器的位 DIR[1:0]	方向	源地址	目标地址
00	外设到内存	DMA_SxPAR	DMA_SxM0AR/DMA_SxM1AR
01	内存到外设	DMA_SxM0AR/DMA_SxM1AR	DMA_SxPAR
10	不允许(1)		
11	保留	-	-

1. 使能双缓冲区模式时，自动使能循环模式。由于内存到内存模式与循环模式不兼容，所以当使能双缓冲区模式时，不允许配置内存到内存模式。

16.3.11 可编程数据宽度、封装/解封、字节序

要传输的数据项数目必须在使能数据流之前编程到 DMA_SxNDTR（要传输数据项数目位，NDT）中，当流控制器是外设且 DMA_SxCR 中的 PFCTRL 位置为 1 时除外。

当使用内部 FIFO 时，源和目标数据的数据宽度可以通过 DMA_SxCR 寄存器的 PSIZE 和 MSIZE 位（可以是 8、16 或32 位）编程。

当 PSIZE 和 MSIZE 不相等时：

- 在 DMA_SxNDTR 寄存器中配置的要传输的数据项数目的数据宽度等于外设总线的宽度（由 DMA_SxCR 寄存器中的 PSIZE 位配置）。例如，在外设到内存、内存到外设或内存到内存传输的情况下，如果将 PSIZE[1:0] 位配置为半字，则要传输的字节数等于 2 × NDT。
- DMA 控制器仅按小字节序寻址源和目标。相关内容在*表 95：封装/解封和字节序行为（位 PINC = MINC = 1）*中介绍。

在封装/解封数据的过程中，如果在数据完全封装/解封前中断操作，则有数据损坏的危险。因此，为了确保数据一致性，可将数据流配置成生成突发传输：在这种情况下，属于一个突发的每组传输不可分割（请参见*第 16.3.12 节：单次传输和突发传输*）。

在直接模式下（DMA_SxFCR 寄存器中的 DMDIS = 0），不能进行数据封装/解封。这种情况下，不允许源与目标的传输数据宽度不同，二者必须相等，并由 DMA_SxCR 寄存器中的 PSIZE 位定义。MSIZE 位的状态是“无关”。

表 95. 封装/解封和字节序行为（位 PINC = MINC = 1）

AHB 内存端口宽度	AHB 外设端口宽度	要传输的数据项的数目 (NDT)	内存传输数目	内存端口地址/字节通道	外设传输数目	外设端口地址/字节通道	
						PINCOS = 1	PINCOS = 0
8	8	4	1 2 3 4	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]	1 2 3 4	0x0/B0[7:0] 0x4/B1[7:0] 0x8/B2[7:0] 0xC/B3[7:0]	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]
8	16	2	1 2 3 4	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]	1 2	0x0/B1|B0[15:0] 0x4/B3|B2[15:0]	0x0/B1|B0[15:0] 0x2/B3|B2[15:0]
8	32	1	1 2 3 4	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]	1	0x0 / B3|B2|B1|B0[31:0]	0x0 / B3|B2|B1|B0[31:0]
16	8	4	1 2	0x0/B1|B0[15:0] 0x2/B3|B2[15:0]	1 2 3 4	0x0/B0[7:0] 0x4/B1[7:0] 0x8/B2[7:0] 0xC/B3[7:0]	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]
16	16	2	1 2	0x0/B1|B0[15:0] 0x2/B1|B0[15:0]	1 2	0x0/B1|B0[15:0] 0x4/B3|B2[15:0]	0x0/B1|B0[15:0] 0x2/B3|B2[15:0]
16	32	1	1 2	0x0/B1|B0[15:0] 0x2/B3|B2[15:0]	1	0x0/B3|B2|B1|B0[31:0]	0x0/B3|B2|B1|B0[31:0]
32	8	4	1	0x0/B3|B2|B1|B0[31:0]	1 2 3 4	0x0/B0[7:0] 0x4/B1[7:0] 0x8/B2[7:0] 0xC/B3[7:0]	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]
32	16	2	1	0x0/B3|B2|B1|B0[31:0]	1 2	0x0/B1|B0[15:0] 0x4/B3|B2[15:0]	0x0/B1|B0[15:0] 0x2/B3|B2[15:0]
32	32	1	1	0x0/B3|B2|B1|B0[31:0]	1	0x0 / B3|B2|B1|B0 [31:0]	0x0 / B3|B2|B1|B0[31:0]

注： *外设端口可以是源或目标（在内存到内存传输的情况下，也可能是内存源）。*

必须配置 PSIZE、MSIZE 和 NDT[15:0]，以确保最后一次传输的完整性。当外设端口的数据宽度（PSIZE 位）小于内存端口的数据宽度（MSIZE 位）时，可能会发生数据传输不完整的情况。下表对此限制进行了总结。

表 96. PSIZE 与 MSIZE 确定时对 NDT 的限制条件

DMA_SxCR 的 PSIZE[1:0]	DMA_SxCR 的 MSIZE[1:0]	DMA_SxNDTR 的 NDT[15:0]
00（8 位）	01（16 位）	必须是 2 的倍数。
00（8 位）	10（32 位）	必须是 4 的倍数。
01（16 位）	10（32 位）	必须是 2 的倍数。

16.3.12 单次传输和突发传输

DMA 控制器可以产生单次传输或 4 个、8 个和 16 个节拍的增量突发传输。

突发大小通过软件针对两个 AHB 端口独立配置，配置时使用 DMA_SxCR 寄存器中的 MBURST[1:0] 和 PBURST[1:0] 位。

突发大小指示突发中的节拍数，而不是传输的字节数。

为确保数据一致性，形成突发的每一组传输都不可分割：在突发传输序列期间，AHB 传输会锁定，并且 AHB 总线矩阵的仲裁器不解除对 DMA 主总线的授权。

根据单次或突发配置的情况，每个 DMA 请求在 AHB 外设端口上相应地启动不同数量的传输。

- 当 AHB 外设端口被配置为单次传输时，根据 DMA_SxCR 寄存器 PSIZE[1:0] 位的值，每个 DMA 请求产生一次字节、半字或字的数据传输。
- 当 AHB 外设端口被配置为突发传输时，根据 DMA_SxCR 寄存器 PBURST[1:0] 和 PSIZE[1:0] 位的值，每个 DMA 请求相应地生成 4 个、8 个或 16 个节拍的字节、半字或字的传输。

对于需要配置 MBURST 和 MSIZE 位的 AHB 内存端口，必须考虑与上述相同的内容。

在直接模式下，数据流只能生成单次传输，而 MBURST[1:0] 和 PBURST[1:0] 位由硬件强制配置。

必须选择地址指针（DMA_SxPAR 或 DMA_SxM0AR 寄存器），以确保一个突发块内的所有传输在等于传输大小的地址边界对齐。

选择突发配置必须要遵守 AHB 协议，即突发传输**不得**越过 1 KB 地址边界，因为可以分配给单个从设备的最小地址空间是 1 KB。这意味着突发块传输**不得**越过 1 KB 地址边界，否则就会产生一个 AHB 错误，并且 DMA 寄存器不会报告这个错误。

16.3.13 FIFO

FIFO 结构

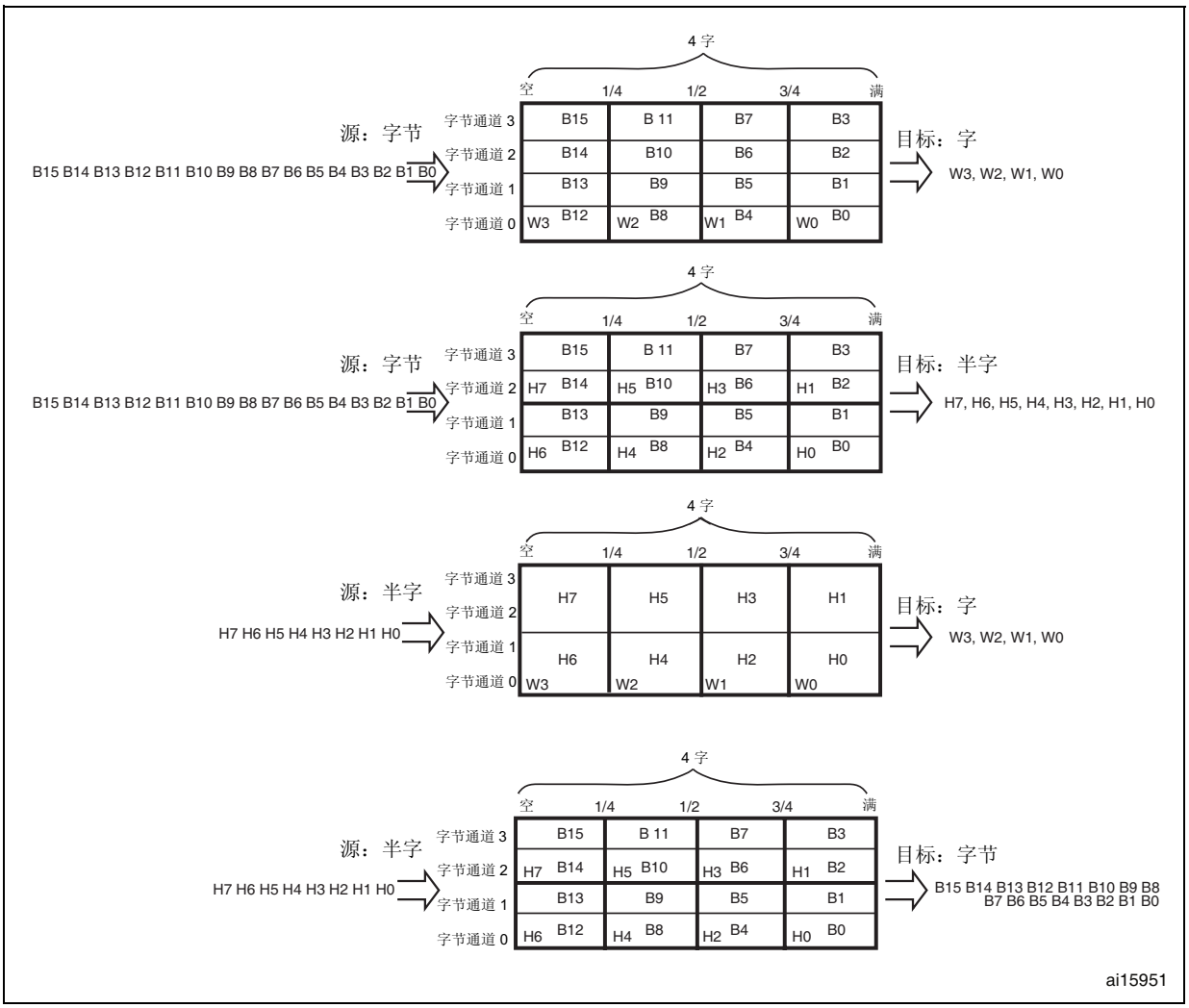
FIFO 用于在源数据传输到目标之前临时存储这些数据。

每个数据流都有一个独立的 4 字 FIFO，阈值级别可由软件配置为 1/4、1/2、3/4 或满。

为了使能 FIFO 阈值级别，必须通过将 DMA_SxFCR 寄存器中的 DMDIS 位置 1 来禁止直接模式。

FIFO 的结构随源与目标数据宽度而不同，如下图所示。

图 116. FIFO 结构


4 字
空
1/4
1/2
3/4
满
源：字节
B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
字节通道 3
字节通道 2
字节通道 1
字节通道 0
目标：字
W3, W2, W1, W0
源：字节
目标：半字
H7, H6, H5, H4, H3, H2, H1, H0
源：半字
H7 H6 H5 H4 H3 H2 H1 H0
目标：字
目标：字节
B15 B14 B13 B12 B11 B10 B9 B8
B7 B6 B5 B4 B3 B2 B1 B0
ai15951

FIFO 阈值与突发配置

选择 FIFO 阈值（DMA_SxFCR 寄存器的位 FTH[1:0]）和内存突发大小（DMA_SxCR 寄存器的 MBURST[1:0] 位）时需要小心：FIFO 阈值指向的内容必须与整数个内存突发传输完全匹配。如果不是这样，当使能数据流时将生成一个 FIFO 错误（DMA_HISR 或 DMA_LISR 寄存器的标志 FEIFx），然后将自动禁止数据流。下表介绍了允许的和禁止的配置。禁止的配置在表中以灰色突出显示。

表 97. FIFO 阈值配置

MSIZE	FIFO 级别	MBURST = INCR4	MBURST = INCR8	MBURST = INCR16
字节	1/4	4 个节拍的 1 次突发	禁止	禁止
	1/2	4 个节拍的 2 次突发	8 个节拍的 1 次突发	
	3/4	4 个节拍的 3 次突发	禁止	
	满	4 个节拍的 4 次突发	8 个节拍的 2 次突发	16 个节拍的 1 次突发
半字	1/4	禁止	禁止	禁止
	1/2	4 个节拍的 1 次突发		
	3/4	禁止		
	满	4 个节拍的 2 次突发	8 个节拍的 1 次突发	
字	1/4	禁止	禁止	
	1/2			
	3/4			
	满	4 个节拍的 1 次突发		

所有这些情况下，突发大小与数据大小的乘积不得超过 FIFO 大小（数据大小可以为：1（字节）、2（半字）或 4（字））。

如果发生下列一种情况，会导致 DMA 传输结束时出现不完整的突发传输：

- 对于 AHB 外设端口配置：数据项总数（在 DMA_SxNDTR 寄存器中设置）不是突发大小与数据大小乘积的倍数。
- 对于 AHB 内存端口配置：要传输到内存的 FIFO 中的剩余数据项的数目不是突发大小与数据大小乘积的倍数。

在这些情况下，即使在 DMA 数据流配置期间请求突发事务，要传输的剩余数据也将由 DMA 在单独模式下管理。

注： *当在外设 AHB 端口上请求突发传输并且使用 FIFO（DMA_SxCR 寄存器中 DMDIS = 1）时，必须根据 DMA 数据流方向遵守下列规则来避免出现上溢或下溢情况：*

如果 (PBURST × PSIZE) = FIFO_SIZE（4 字），则当 PSIZE = 1、2 或 4，PBURST = 4、8 或 16 时，禁止 FIFO_Threshold = 3/4。

此规则将确保一次释放足够的 FIFO 空间来处理外设的请求。

FIFO 刷新

当复位 DMA_SxCR 寄存器中的 EN 位来禁止数据流，以及配置数据流来管理外设到内存或内存到内存的传输时，可以刷新 FIFO。因为如果禁止数据流时仍有某些数据存留在 FIFO 中，DMA 控制器会将剩余的数据继续传输到目标（即使已经有效禁止了数据流）。刷新完成时，会将 DMA_LISR 或 DMA_HISR 寄存器中的传输完成状态位 (TCIFx) 置 1。

在这种情况下，剩余数据计数器 DMA_SxNDTR 保持的值指示在目标内存现有多少可用数据项。

请注意在 FIFO 刷新操作期间，如果 FIFO 中要传输到内存的剩余数据项的数目（以字节为单位）小于内存数据宽度（例如在 MSIZE 配置为字时 FIFO 中为 2 个字节），则会使用在 DMA_SxCR 寄存器中的 MSIZE 位设置的数据宽度发送数据。这意味着将使用非预期值写入内存。软件可以读取 DMA_SxNDTR 寄存器来确定包含良好数据的存储区域（起始地址和最后地址）。

如果 FIFO 中剩余数据项的数目小于突发大小，并且数据流通过将 DMA_SxCR 寄存器 MBURST 位置 1 进行配置来管理在 AHB 内存端口上的突发传输，则使用单次传输来完成 FIFO 的刷新。

直接模式

默认情况下，FIFO 以直接模式操作（将 DMA_SxFCR 中的 DMDIS 位置 1），不使用 FIFO 阈值级别。如果在每次 DMA 请求之后，系统需要至/自内存的立即和单独传输，这种模式非常有用。

当在直接模式（禁止 FIFO）下将 DMA 配置为以内存到外设模式传输数据时，DMA 会将一个数据从内存预加载到内部 FIFO，从而确保一旦外设触发 DMA 请求时则立即传输数据。

为了避免 FIFO 饱和，建议使用高优先级配置相应的数据流。

该模式仅限以下方式的传输：

- 源和目标传输宽度相等，并均由 DMA_SxCR 中的 PSIZE[1:0] 位定义（MSIZE[1:0] 位的状态是“无关”）
- 不可能进行突发传输（DMA_SxCR 中的 PBURST[1:0] 和 MBURST[1:0] 位的状态是“无关”）

当实现内存到内存传输时不得使用直接模式。

16.3.14 DMA 传输完成

以下各种事件均可以结束传输过程，并将 DMA_LISR 或 DMA_HISR 状态寄存器中的 TCIFx 位置 1：

- 在 DMA 流控制器模式下：
  - 在内存到外设模式下，DMA_SxNDTR 计数器已达到零。
  - 传输结束前禁止了数据流（通过将 DMA_SxCR 寄存器中的 EN 位清零），并且（当传输是外设到内存或内存到内存）所有的剩余数据均已从 FIFO 刷新到内存。
- 在外设流控制器模式下：
  - 已从外设生成最后的外部突发请求或单独请求，并当 DMA 在外设到内存模式下工作时，剩余数据已从 FIFO 传输到内存
  - 数据流由软件禁止，并当 DMA 在外设到内存模式下工作时，剩余数据已从 FIFO 传输到内存

注： *仅在外设到内存模式下，传输的完成取决于 FIFO 中要传输到内存的剩余数据。这种情况不适用于内存到外设模式。*

如果是在非循环模式下配置数据流，传输结束后（即要传输的数据数目达到零），除非软件重新对数据流编程并重新使能数据流（通过将 DMA_SxCR 寄存器中的 EN 位置 1），否则 DMA 即会停止传输（通过硬件将 DMA_SxCR 寄存器中的 EN 位清零）并且不再响应任何 DMA 请求。

16.3.15 DMA 传输暂停

可以随时暂停 DMA 传输以供稍后重新开始；也可以在 DMA 传输结束前明确禁止传输。

分为两种情况：

- 数据流禁止传输，以后不从停止点重新开始暂停。这种情况下，只需将 DMA_SxCR 寄存器中的 EN 位清零来禁止数据流，除此之外不需要任何其他操作。禁止数据流可能要花费一些时间（需要首先完成正在进行的传输）。需要将传输完成中断标志（DMA_LISR 或 DMA_HISR 寄存器中的 TCIF）置 1 来指示传输结束。当前 DMA_SxCR 中的 EN 位的值是 0，借此确认数据流已经终止传输。DMA_SxNDTR 寄存器包含数据流停止时剩余数据项的数目，这样软件便可以确定数据流中断前已传输了多少数据项。
- 数据流在 DMA_SxNDTR 寄存器中要传输的剩余数据项数目达到 0 之前暂停传输。目的是以后通过重新使能数据流重新开始传输。为了在传输停止点重新开始传输，软件必须在通过写入 DMA_SxCR 寄存器中的 EN 位（然后检查确认该位为 0）禁止数据流之后，首先读取 DMA_SxNDTR 寄存器来了解已经收集的数据项的数目。然后：
  - 必须更新外设和/或内存地址以调整地址指针
  - 必须使用要传输的剩余数据项的数目（禁止数据流时读取的值）更新 SxNDTR 寄存器
  - 然后可以重新使能数据流，从停止点重新开始传输

注： *传输完成中断标志（DMA_LISR 或 DMA_HISR 中的 TCIF）置 1 将指示因数据流中断而结束传输。*

16.3.16 流控制器

控制要传输的数据数目的实体称为流控制器。此流控制器使用 DMA_SxCR 寄存器中的 PFCTRL 位针对每个数据流独立配置。

流控制器可以是：

- DMA 控制器：在这种情况下，要传输的数据项的数目在使能 DMA 数据流之前由软件编程到 DMA_SxNDTR 寄存器。
- 外设源或目标：当要传输的数据项的数目未知时属于这种情况。当所传输的是最后的数据时，外设通过硬件向 DMA 控制器发出指示。仅限能够发出传输结束信号的外设支持此功能。

当外设流控制器用于给定数据流时，写入 DMA_SxNDTR 的值对 DMA 传输没有作用。实际上，不论写入什么值，一旦使能数据流，硬件即会将该值强制置为 0xFFFF 来执行下列方案：

- 预期的数据流中断：DMA_SxCR 寄存器中的 EN 位由软件复位为 0，以在外设发送最后的数据硬件信号（单独或突发）之前停止数据流。这样，在外设到内存 DMA 传输的情况下，数据流即会关闭并触发 FIFO 刷新。状态寄存器中相应数据流的 TCIFx 标志置 1 以指示 DMA 完成传输。要了解 DMA 传输期间传输的数据项的数目，请读取 DMA_SxNDTR 寄存器并应用下列公式：
  - 传输的数据数目 = 0xFFFF – DMA_SxNDTR
- 因接收到最后的数据硬件信号而引起的正常数据流中断：当外设请求最后的传输（单独或突发）并当此传输完成时，自动中断数据流。相应数据流的 TCIFx 标志在状态寄存器中置 1 以指示 DMA 传输完成。要了解传输的数据项的数目，请读取 DMA_SxNDTR 寄存器并应用与上面相同的公式。
- DMA_SxNDTR 寄存器达到 0：状态寄存器中相应数据流的 TCIFx 标志置 1 以指示强制的 DMA 传输完成。即使尚未置位最后的数据硬件信号（单独或突发），也会自动关闭数据流。已传输的数据不会丢失。这意味着即使在外设流控制模式下，DMA 在单独的事务中最多处理 65535 个数据项。

注： *当在内存到内存模式下配置时，DMA 始终是流控制器，而 PFCTRL 位由硬件强制置为 0。*

在外设流控制器模式下禁止循环模式。

16.3.17 可能的 DMA 配置汇总

下表汇总了各种可能的 DMA 配置。禁止的配置在表中以灰色突出显示。

表 98. 可能的 DMA 配置

DMA 传输模式	源	目标	流控制器	循环模式	传输类型	直接模式	双缓冲区模式
外设到内存	AHB 外设端口	AHB 内存端口	DMA	可能的	单侧	可能的	可能的
					突发	禁止	
			外设	禁止	单侧	可能的	禁止
					突发	禁止	
内存到外设	AHB 内存端口	AHB 外设端口	DMA	可能的	单侧	可能的	可能的
					突发	禁止	
			外设	禁止	单侧	可能的	禁止
					突发	禁止	
内存到内存	AHB 外设端口	AHB 内存端口	仅 DMA	禁止	单侧	禁止	禁止
					突发		

16.3.18 流配置过程

配置 DMA 数据流 x（其中 x 是数据流编号）时必须遵守下面的顺序：

1. 如果使能了数据流，通过复位 DMA_SxCR 寄存器中的 EN 位将其禁止，然后读取此位以确认没有正在进行的数据流操作。将此位写为 0 不会立即生效，因为实际上只有所有当前传输都已完成时才会将其写为 0。当所读取 EN 位的值为 0 时，才表示可以配置数据流。因此在开始任何数据流配置之前，需要等待 EN 位置 0。必须将先前的数据块 DMA 传输中在状态寄存器（DMA_LISR 和 DMA_HISR）中置 1 的所有数据流专用的位置 0，然后才可重新使能数据流。
2. 在 DMA_SxPAR 寄存器中设置外设端口寄存器地址。外设事件发生后，数据会从此地址移动到外设端口或从外设端口移动到此地址。
3. 在 DMA_SxMA0R 寄存器（在双缓冲区模式的情况下还有 DMA_SxMA1R 寄存器）中设置内存地址。外设事件发生后，将从此内存读取数据或将数据写入此内存。
4. 在 DMA_SxNDTR 寄存器中配置要传输的数据项的总数。每出现一次外设事件或每出现一个节拍的突发传输，该值都会递减。
5. 使用 DMAMUX 将 DMA 请求线接到 DMA 通道。
6. 如果外设用作流控制器而且支持此功能，请将 DMA_SxCR 寄存器中的 PFCTRL 位置 1。
7. 使用 DMA_SxCR 寄存器中的 PL[1:0] 位配置数据流优先级。
8. 配置 FIFO 的使用情况（使能或禁止，发送和接收阈值）
9. 配置数据传输方向、外设和内存增量 / 固定模式、单独或突发事务、外设和内存数据宽度、循环模式、双缓冲区模式和传输完成一半和 / 或全部完成，和 / 或 DMA_SxCR 寄存器中错误的中断。
10. 通过将 DMA_SxCR 寄存器中的 EN 位置 1 激活数据流。

一旦使能了流，即可响应连接到数据流的外设发出的任何 DMA 请求。

一旦在 AHB 目标端口上传输了一半数据，传输一半标志 (HTIF) 便会置 1，如果传输一半中断使能位 (HTIE) 置 1，还会生成中断。传输结束时，传输完成标志 (TCIF) 便会置 1，如果传输完成中断使能位 (TCIE) 置 1，还会生成中断。

警告： **要关闭连接到 DMA 数据流请求的外设，必须首先关闭外设连接的 DMA 数据流，然后等待 EN 位 = 0。只有这样才能安全地禁止外设。**

16.3.19 错误管理

DMA 控制器可以检测到以下错误：

- **传输错误**：当发生下列情况时，传输错误中断标志 (TEIFx) 将置 1：
  - DMA 读取或写入访问期间发生总线错误
  - 软件请求在双缓冲区模式下写入访问内存地址寄存器，但是，已使能数据流，并且当前目标内存是受写入内存地址寄存器操作影响的内存（请参见*第 16.3.10 节：双缓冲区模式*）
- **FIFO 错误**：如果发生下列情况，FIFO 错误中断标志 (FEIFx) 将置 1：
  - 检测到 FIFO 下溢情况
  - 检测到 FIFO 上溢情况（在内存到内存模式下由 DMA 内部管理请求和传输，所以在此模式下不检测溢出情况）
  - 当 FIFO 阈值级别与内存突发大小不兼容时使能流（请参见*表 97：FIFO 阈值配置*）
- **直接模式错误**：只有当在直接模式下工作并且已将 DMA_SxCR 寄存器中的 MINC 位清零时，才能在外设到内存模式下将直接模式错误中断标志 (DMEIFx) 置 1。当在先前数据未完全传输到内存（因为内存总线未得到授权）的情况下发生 DMA 请求时，该标志将置 1。在这种情况下，该标志指示有两个数据项相继传输到相同的目标地址，如果目标不能管理这种情况，会发生问题。

在直接模式下，如果出现下列条件，FIFO 错误标志也会置 1：

- 在外设到内存模式下，如果未对内存总线授权支持多个外设请求，FIFO 可能饱和（上溢）。
- 在内存到外设模式下，如果在外设请求发生前内存总线未得到授权，则可能发生下溢的情况。

如果由于突发大小与 FIFO 阈值级别之间的不兼容而引起 TEIFx 或 FEIFx 标志置 1，则硬件自动将相应数据流配置寄存器 (DMA_SxCR) 中的 EN 位清零，从而禁止错误的数据流。

如果由于上溢或下溢情况引起 DMEIFx 或 FEIFx 标志置 1，则不会自动禁止错误的数据流，而是由软件决定是否通过复位 DMA_SxCR 寄存器中的 EN 位禁止数据流。这是因为当发生这种错误时没有数据丢失。

当 DMA_LISR 或 DMA_HISR 寄存器中数据流的错误中断标志（TEIF、FEIF、DMEIF）置 1 时，如果 DMA_SxCR 或 DMA_SxFCR 寄存器中相应的中断使能位（TEIE、FEIE、DMIE）也置 1，则会生成一个中断。

注： *当 FIFO 上溢或下溢的情况发生时，因为直到上溢或下溢的情况被清除，数据流才会确认外设请求，所以数据不会丢失。如果此确认过程花费过多时间，则外设本身会检测到其内部缓冲器上溢或下溢的情况，数据可能丢失。*

16.4 DMA 中断

对于每个 DMA 数据流，可在发生以下事件时产生中断：

- 达到半传输
- 传输完成
- 传输错误
- FIFO 错误（上溢、下溢或 FIFO 级别错误）
- 直接模式错误

可以使用单独的中断使能控制位以实现灵活性，如下表所示。

表 99. DMA 中断请求

中断事件	事件标志	使能控制位
半传输	HTIF	HTIE
传输完成	TCIF	TCIE
传输错误	TEIF	TEIE
FIFO 上溢/下溢	FEIF	FEIE
直接模式错误	DMEIF	DMEIE

注： *在将使能控制位 EN 置 1 前，必须将相应的事件标志清零，否则会立即产生中断。*

16.5 DMA 寄存器

DMA 寄存器必须按字（32 位）进行访问。

16.5.1 DMA 低中断状态寄存器 (DMA_LISR)

DMA low interrupt status register

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TCIF3	HTIF3	TEIF3	DMEIF3	Res.	FEIF3	TCIF2	HTIF2	TEIF2	DMEIF2	Res.	FEIF2
				r	r	r	r		r	r	r	r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TCIF1	HTIF1	TEIF1	DMEIF1	Res.	FEIF1	TCIF0	HTIF0	TEIF0	DMEIF0	Res.	FEIF0
				r	r	r	r		r	r	r	r	r		r

位 31:28、15:12 保留，必须保持复位值。

位 27、21、11、5 **TCIF[3:0]**：数据流 x 传输完成中断标志 (Stream x transfer complete interrupt flag)（x = 3 到 0）
该位由硬件置 1，由软件清零，软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。
0：数据流 x 上无传输完成事件
1：数据流 x 上发生传输完成事件。

位 26、20、10、4 **HTIF[3:0]**：数据流 x 半传输中断标志 (Stream x half transfer interrupt flag)（x = 3 到 0）
该位由硬件置 1，由软件清零，软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。
0：数据流 x 上无半传输事件
1：数据流 x 上发生半传输事件。

位 25、19、9、3 **TEIF[3:0]**：数据流 x 传输错误中断标志 (Stream x transfer error interrupt flag)（x = 3 到 0）
该位由硬件置 1，由软件清零，软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。
0：数据流 x 上无传输错误
1：数据流 x 上发生传输错误。

位 24、18、8、2 **DMEIF[3:0]**：数据流 x 直接模式错误中断标志 (Stream x direct mode error interrupt flag)（x = 3 到 0）
该位由硬件置 1，由软件清零，软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。
0：数据流 x 上无直接模式错误
1：数据流 x 上发生直接模式错误。

位 23、17、7、1 保留，必须保持复位值。

位 22、16、6、0 **FEIF[3:0]**：数据流 x FIFO 错误中断标志 (Stream x FIFO error interrupt flag)（x = 3 到 0）
该位由硬件置 1，由软件清零，软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。
0：数据流 x 上无 FIFO 错误事件
1：数据流 x 上发生 FIFO 错误事件。

16.5.2 DMA 高中断状态寄存器 (DMA_HISR)

DMA high interrupt status register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TCIF7	HTIF7	TEIF7	DMEIF 7	Res.	FEIF7	TCIF6	HTIF6	TEIF6	DMEIF 6	Res.	FEIF6
				r	r	r	r		r	r	r	r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TCIF5	HTIF5	TEIF5	DMEIF 5	Res.	FEIF5	TCIF4	HTIF4	TEIF4	DMEIF 4	Res.	FEIF4
				r	r	r	r		r	r	r	r	r		r

位 31:28、15:12 保留，必须保持复位值。

位 27、21、11、5 **TCIF[7:4]**：数据流 x 传输完成中断标志 (Stream x transfer complete interrupt flag)（x = 7 到 4）
该位由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。
0：数据流 x 上无传输完成事件
1：数据流 x 上发生传输完成事件。

位 26、20、10、4 **HTIF[7:4]**：数据流 x 半传输中断标志 (Stream x half transfer interrupt flag)（x = 7 到 4）
该位由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。
0：数据流 x 上无半传输事件
1：数据流 x 上发生半传输事件。

位 25、19、9、3 **TEIF[7:4]**：数据流 x 传输错误中断标志 (Stream x transfer error interrupt flag)（x = 7 到 4）
该位由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。
0：数据流 x 上无传输错误
1：数据流 x 上发生传输错误。

位 24、18、8、2 **DMEIF[7:4]**：数据流 x 直接模式错误中断标志 (Stream x direct mode error interrupt flag)（x = 7 到 4）
该位由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。
0：数据流 x 上无直接模式错误
1：数据流 x 上发生直接模式错误。

位 23、17、7、1 保留，必须保持复位值。

位 22、16、6、0 **FEIF[7:4]**：数据流 x FIFO 错误中断标志 (Stream x FIFO error interrupt flag)（x = 7 到 4）
该位由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。
0：数据流 x 上无 FIFO 错误事件
1：数据流 x 上发生 FIFO 错误事件。

16.5.3 DMA 低中断标志清零寄存器 (DMA_LIFCR)

DMA low interrupt flag clear register

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CTCIF3	CHTIF3	CTEIF3	CDMEIF3	Res.	CFEIF3	CTCIF2	CHTIF2	CTEIF2	CDMEIF2	Res.	CFEIF2
				w	w	w	w		w	w	w	w	w		w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CTCIF1	CHTIF1	CTEIF1	CDMEIF1	Res.	CFEIF1	CTCIF0	CHTIF0	CTEIF0	CDMEIF0	Res.	CFEIF0
				w	w	w	w		w	w	w	w	w		w

位 31:28、15:12 保留，必须保持复位值。

位 27、21、11、5 **CTCIF[3:0]**：数据流 x 传输完成中断标志清零 (Stream x clear transfer complete interrupt flag)（x = 3 到 0）

将 1 写入此位时，DMA_LISR 寄存器中相应的 TCIFx 标志将清零。

位 26、20、10、4 **CHTIF[3:0]**：数据流 x 半传输中断标志清零 (Stream x clear half transfer interrupt flag)（x = 3 到 0）

将 1 写入此位时，DMA_LISR 寄存器中相应的 HTIFx 标志将清零

位 25、19、9、3 **CTEIF[3:0]**：数据流 x 传输错误中断标志清零 (Stream x clear transfer error interrupt flag)（x = 3 到 0）

将 1 写入此位时，DMA_LISR 寄存器中相应的 TEIFx 标志将清零。

位 24、18、8、2 **CDMEIF[3:0]**：数据流 x 直接模式错误中断标志清零 (Stream x clear direct mode error interrupt flag)（x = 3 到 0）

将 1 写入此位时，DMA_LISR 寄存器中相应的 DMEIFx 标志将清零。

位 23、17、7、1 保留，必须保持复位值。

位 22、16、6、0 **CFEIF[3:0]**：数据流 x FIFO 错误中断标志清零 (Stream x clear FIFO error interrupt flag)（x = 3 到 0）

将 1 写入此位时，DMA_LISR 寄存器中相应的 CFEIFx 标志将清零。

16.5.4 DMA 高中断标志清零寄存器 (DMA_HIFCR)

DMA high interrupt flag clear register

偏移地址：0x00C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CTCIF7	CHTIF7	CTEIF7	CDMEIF7	Res.	CFEIF7	CTCIF6	CHTIF6	CTEIF6	CDMEIF6	Res.	CFEIF6
				w	w	w	w		w	w	w	w	w		w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CTCIF5	CHTIF5	CTEIF5	CDMEIF5	Res.	CFEIF5	CTCIF4	CHTIF4	CTEIF4	CDMEIF4	Res.	CFEIF4
				w	w	w	w		w	w	w	w	w		w

位 31:28、15:12 保留，必须保持复位值。

位 27、21、11、5 **CTCIF[7:4]**：数据流 x 传输完成中断标志清零 (Stream x clear transfer complete interrupt flag)（x = 7 到 4）
将 1 写入此位时，DMA_HISR 寄存器中相应的 TCIFx 标志将清零。

位 26、20、10、4 **CHTIF[7:4]**：数据流 x 半传输中断标志清零 (Stream x clear half transfer interrupt flag)（x = 7 到 4）
将 1 写入此位时，DMA_HISR 寄存器中相应的 HTIFx 标志将清零。

位 25、19、9、3 **CTEIF[7:4]**：数据流 x 传输错误中断标志清零 (Stream x clear transfer error interrupt flag)（x = 7 到 4）
将 1 写入此位时，DMA_HISR 寄存器中相应的 TEIFx 标志将清零。

位 24、18、8、2 **CDMEIF[7:4]**：数据流 x 直接模式错误中断标志清零 (Stream x clear direct mode error interrupt flag)（x = 7 到 4）
将 1 写入此位时，DMA_HISR 寄存器中相应的 DMEIFx 标志将清零。

位 23、17、7、1 保留，必须保持复位值。

位 22、16、6、0 **CFEIF[7:4]**：数据流 x FIFO 错误中断标志清零 (Stream x clear FIFO error interrupt flag)（x = 7 到 4）
将 1 写入此位时，DMA_HISR 寄存器中相应的 CFEIFx 标志将清零。

16.5.5 DMA 数据流 x 配置寄存器 (DMA_SxCR)

DMA stream x configuration register

此寄存器用于配置相关数据流。

偏移地址：0x010 + 0x18 * x，（x = 0 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	MBURST[1:0]		PBURST[1:0]		TRBUFF	CT	DBM	PL[1:0]	
							rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24:23 **MBURST[1:0]**：内存突发传输配置 (Memory burst transfer configuration)
这些位由软件置 1 和清零，
00：单次传输
01：INCR4（4 个节拍的增量突发传输）
10：INCR8（8 个节拍的增量突发传输）
11：INCR16（16 个节拍的增量突发传输）
这些位受到保护，只有 EN = 0 时才可以写入。
在直接模式中，当位 EN = 1 时，这些位由硬件强制置为 0x0。

位 22:21 **PBURST[1:0]**：外设突发传输配置 (Peripheral burst transfer configuration)
这些位由软件置 1 和清零，
00：单次传输
01：INCR4（4 个节拍的增量突发传输）
10：INCR8（8 个节拍的增量突发传输）
11：INCR16（16 个节拍的增量突发传输）
这些位受到保护，只有 EN = 0 时才可以写入。
在直接模式下，这些位由硬件强制置为 0x0。

位 20 **TRBUFF**：使能 DMA 处理可缓冲传输 (Enable the DMA to handle bufferable transfers)
0：未使能可缓冲传输
1：使能可缓冲传输
注意：如果 DMA 数据流管理 UART/USART 传输，则必须将此位置 1。

位 19 **CT**：当前目标（仅在双缓冲区模式下）(Current target (only in double buffer mode))
该位由硬件置 1 和清零。也可由软件写入。
0：当前目标内存为内存 0（使用 DMA_SxM0AR 指针寻址）。
1：当前目标内存为内存 1（使用 DMA_SxM1AR 指针寻址）。
只有 EN 为 0 时，此位才可以写入，以指示第一次传输的目标存储区。在使能数据流后，此位相当于一个状态标志，用于指示作为当前目标的存储区。

位 18 **DBM**：双缓冲区模式
该位由软件置 1 和清零，
0：传输结束时不切换缓冲区
1：DMA 传输结束时切换目标存储区
此位受到保护，只有 EN 为 0 时才可以写入。

位 17:16 **PL[1:0]**：优先级 (priority level)
这些位由软件置 1 和清零，
00：低
01：中
10：高
11：非常高
这些位受到保护，只有 EN = 0 时才可以写入。

位 15 **PINCOS**：外设增量偏移量 (Peripheral increment offset size)
该位由软件置 1 和清零
0：用于计算外设地址的偏移量与 PSIZE 相关。
1：用于计算外设地址的偏移量固定为 4（32 位对齐）。
如果位 PINC = 0，则此位没有意义。
此位受到保护，只有 EN 为 0 时才可以写入。
如果选择直接模式或者 PBURST 不等于 00，则当使能数据流 (EN = 1) 时，此位由硬件强制置为低电平。

位 14:13 **MSIZE[1:0]**：内存数据大小
这些位由软件置 1 和清零，
00：字节（8 位）
01：半字（16 位）
10：字（32 位）
11：保留
这些位受到保护，只有 EN = 0 时才可以写入。
在直接模式下，当 EN = 1 时，MSIZE 位由硬件强制置为与 PSIZE 相同的值。

位 12:11 **PSIZE[1:0]**：外设数据大小
这些位由软件置 1 和清零，
00：字节（8 位）
01：半字（16 位）
10：字（32 位）
11：保留
这些位受到保护，只有 EN = 0 时才可以写入。

位 10 **MINC**：内存递增模式 (Memory increment mode)
该位由软件置 1 和清零，
0：内存地址指针固定
1：每次数据传输后，内存地址指针递增（增量为 MSIZE 值）
此位受到保护，只有 EN 为 0 时才可以写入。

位 9 **PINC**：外设递增模式 (Peripheral increment mode)
该位由软件置 1 和清零，
0：外设地址指针固定
1：每次数据传输后，外设地址指针递增（增量为 PSIZE 值）
此位受到保护，只有 EN 为 0 时才可以写入。

位 8 **CIRC**：循环模式 (Circular mode)
此位由软件置 1 和清零，并可由硬件清零。
0：禁止循环模式
1：使能循环模式
如果外设为流控制器（位 PFCTRL=1）且使能数据流 (EN = 1)，此位由硬件自动强制清零。
如果 DBM 位置 1，当使能数据流 (EN = 1) 时，此位由硬件自动强制置 1。

位 7:6 **DIR[1:0]**：数据传输方向 (Data transfer direction)
这些位由软件置 1 和清零，
00：外设到内存
01：内存到外设
10：内存到内存
11：保留
这些位受到保护，只有 EN = 0 时才可以写入。

位 5 **PFCTRL**：外设流控制器 (Peripheral flow controller)
该位由软件置 1 和清零，
0：DMA 是流控制器。
1：外设是流控制器。
此位受到保护，只有 EN 为 0 时才可以写入。
选择内存到内存模式（位 DIR[1:0]=10）后，此位由硬件自动强制清零。

位 4 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)
该位由软件置 1 和清零，
0：禁止 TC 中断
1：使能 TC 中断

位 3 **HTIE**：半传输中断使能 (Half transfer interrupt enable)
该位由软件置 1 和清零，
0：禁止 HT 中断
1：使能 HT 中断

位 2 **TEIE**：传输错误中断使能 (Transfer error interrupt enable)
该位由软件置 1 和清零，
0：禁止 TE 中断
1：使能 TE 中断

位 1 **DMEIE**：直接模式错误中断使能 (Direct mode error interrupt enable)

该位由软件置 1 和清零，

0：禁止 DME 中断

1：使能 DME 中断

位 0 **EN**：数据流使能/读作低电平时数据流就绪标志 (Stream enable / flag stream ready when read low)

该位由软件置 1 和清零，

0：禁止数据流

1：使能数据流

以下情况下此位由硬件清零：

– DMA 传输结束时（准备好配置数据流）

– AHB 主总线出现传输错误时

– 内存 AHB 端口上的 FIFO 阈值与突发大小不兼容时

此位读作 0 时，软件可以对配置和 FIFO 位寄存器编程。EN 位读作 1 时，禁止向这些寄存器执行写入操作。

注意：将 EN 位置 1 以启动新传输之前，DMA_LISR 或 DMA_HISR 寄存器中与数据流相对应的事件标志必须清零。

16.5.6 DMA 数据流 x 数据项数寄存器 (DMA_SxNDTR)

DMA stream x number of data register

偏移地址：0x014 + 0x18 * x，（x = 0 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NDT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **NDT[15:0]**：要传输的数据项数目（0 到 65535）

只有在禁止数据流时，才能向此位字段执行写入操作。使能数据流后，此位字段为只读，用于指示要传输的剩余数据项数。每次 DMA 传输后，此位字段将递减。

传输完成后，此位字段保持为零（数据流处于正常模式时），或者在以下情况下自动以先前编程的值重载：

– 以循环模式配置数据流时

– 通过将 EN 位置 1 来重新使能数据流时

如果该位字段的值为零，则即使使能数据流，也无法完成任何事务。

16.5.7 DMA 数据流 x 外设地址寄存器 (DMA_SxPAR)

DMA stream x peripheral address register

偏移地址：0x018 + 0x18 * x，（x = 0 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PAR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **PAR[31:0]**：外设地址

读/写数据的外设数据寄存器的基址。

这些位受到写保护，只有 DMA_SxCR 中的 EN 位为 0 时才可以写入。

16.5.8 DMA 数据流 x 内存 0 地址寄存器 (DMA_SxM0AR)

DMA stream x memory 0 address register

偏移地址：0x01C + 0x18 * x，（x = 0 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M0A[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M0A[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **M0A[31:0]**：内存 0 地址 (Memory 0 address)

读/写数据的存储区 0 的基址。

这些位受到写保护，只有在以下情况下才可以写入：

- 禁止数据流（DMA_SxCR 中的 EN = 0）或
- 使能数据流（DMA_SxCR 中的 EN = 1）并且 DMA_SxCR 中的 CT = 1（在双缓冲区模式下）。

16.5.9 DMA 数据流 x 内存 1 地址寄存器 (DMA_SxM1AR)

DMA stream x memory 1 address register

偏移地址：0x020 + 0x18 * x，（x = 0 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M1A[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M1A[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **M1A[31:0]**：内存 1 地址（用于双缓冲区模式）(Memory 1 address (used in case of double-buffer mode))

读/写数据的存储区 1 的基址。

此位字段仅用于双缓冲区模式。

这些位受到写保护，只有在以下情况下才可以写入：

– 禁止数据流（DMA_SxCR 中的 EN = 0）或

– 使能数据流（DMA_SxCR 中的 EN = 1）并且 DMA_SxCR 中的位 CT = 0。

16.5.10 DMA 数据流 x FIFO 控制寄存器 (DMA_SxFCR)

DMA stream x FIFO control register

偏移地址：0x024 + 0x18 * x，（x = 0 到 7）

复位值：0x0000 0021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE	Res.	FS[2:0]			DMDIS	FTH[1:0]	
								rw		r	r	r	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7 **FEIE**：FIFO 错误中断使能 (FIFO error interrupt enable)

该位由软件置 1 和清零，

0：禁止 FE 中断

1：使能 FE 中断

位 6 保留，必须保持复位值。

位 5:3 **FS[2:0]**：FIFO 状态 (FIFO status)

这些位为只读。

000：0 < fifo_level < 1/4

001：1/4 ≤ fifo_level < 1/2

010：1/2 ≤ fifo_level < 3/4

011：3/4 ≤ fifo_level < 满

100：FIFO 为空。

101：FIFO 已满。

其他值：保留（无意义）

在直接模式 (DMDIS = 0) 下，这些位的状态是“无关”。

位 2 **DMDIS**：直接模式禁止 (Direct mode disable)

该位由软件置 1 和清零，它可由硬件置 1。

0：使能直接模式

1：禁止直接模式

此位受到保护，只有 EN 为 0 时才可以写入。

如果选择内存到内存模式（DMA_SxCR 中的 DIR = 10），并且 DMA_SxCR 中的 EN = 1，则此位由硬件置 1，因为在内存到内存配置不能使用直接模式。

位 1:0 **FTH[1:0]**：FIFO 阈值选择 (FIFO threshold selection)

这些位由软件置 1 和清零，

00：FIFO 容量的 1/4

01：FIFO 容量的 1/2

10：FIFO 容量的 3/4

11：FIFO 完整容量

在直接模式 (DMIS = 0) 下，不使用这些位。

这些位受到保护，只有 EN = 0 时才可以写入。

16.5.11 DMA 硬件配置 2 寄存器 (DMA_HWCFGR2)

DMA hardware configuration 2 register

偏移地址：0x3EC

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CHSEL_WIDTH[2:0]			Res.	Res.	Res.	WRITE_BUFFERABLE	Res.	Res.	FIFO_SIZE[1:0]	
					r	r	r				r			r	r

位 31:11 保留，必须保持复位值。

位 10:8 **CHSEL_WIDTH[2:0]**：任何 DMA_SxCR 寄存器中 CHSEL 位字段的宽度，对所有数据流通用
在 [0..6] 范围内：
0：无可编程选择
1：2 个通道可编程选择
2：最多 8 个通道可编程选择
3：最多 16 个通道可编程选择
4：最多 32 个通道可编程选择
5：最多 64 个通道可编程选择
6：最多 128 个通道可编程选择

位 7:5 保留，必须保持复位值。

位 4 **WRITE_BUFFERABLE**：
0：DMA 确认信号将在同时满足以下两个条件之时起一个周期后变为无效：1) 到达其 AHB 外设主端口总线访问数据阶段的最后一个周期；2) 外设请求变为无效
1：DMA 确认信号将在到达其 AHB 外设主端口总线访问数据阶段的最后一个周期时，于一个周期后变为无效
在任何情况下，DMA 确认信号都将在到达其 AHB 外设主端口总线访问地址阶段起一个周期后生效。

位 3:2 保留，必须保持复位值。

位 1:0 **FIFO_SIZE[1:0]**：FIFO 大小，对所有数据流通用
在 [0..3] 范围内：
0：2 字 FIFO
1：4 字 FIFO
2：8 字 FIFO
3：16 字 FIFO

16.5.12 DMA 硬件配置 1 寄存器 (DMA_HWCFGR1)

DMA hardware configuration 1 register

偏移地址：0x3F0

复位值：0x2222 2222

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	DMA_DEF7 [1:0]		Res.	Res.	DMA_DEF6 [1:0]		Res.	Res.	DMA_DEF5 [1:0]		Res.	Res.	DMA_DEF4 [1:0]	
		r	r			r	r			r	r			r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DMA_DEF3 [1:0]		Res.	Res.	DMA_DEF2 [1:0]		Res.	Res.	DMA_DEF1 [1:0]		Res.	Res.	DMA_DEF0 [1:0]	
		r	r			r	r			r	r			r	r

位 31:30 保留，必须保持复位值。

位 29:28 **DMA_DEF7[1:0]**：数据流 7 类型 (Stream 7 type)

0：无
1：常规
2：双缓冲区
3：保留

位 27:26 保留，必须保持复位值。

位 25:24 **DMA_DEF6[1:0]**：数据流 6 类型 (Stream 6 type)

0：无
1：常规
2：双缓冲区
3：保留

位 23:22 保留，必须保持复位值。

位 21:20 **DMA_DEF5[1:0]**：数据流 5 类型 (Stream 5 type)

0：无
1：常规
2：双缓冲区
3：保留

位 19:18 保留，必须保持复位值。

位 17:16 **DMA_DEF4[1:0]**：数据流 4 类型 (Stream 4 type)

0：无
1：常规
2：双缓冲区
3：保留

位 15:14 保留，必须保持复位值。

位 13:12 **DMA_DEF3[1:0]**：数据流 3 类型 (Stream 3 type)

0：无
1：常规
2：双缓冲区
3：保留

位 11:10 保留，必须保持复位值。

位 9:8 **DMA_DEF2[1:0]**：数据流 2 类型 (Stream 2 type)
0：无
1：常规
2：双缓冲区
3：保留

位 7:6 保留，必须保持复位值。

位 5:4 **DMA_DEF1[1:0]**：数据流 1 类型 (Stream 1 type)
0：无
1：常规
2：双缓冲区
3：保留

位 3:2 保留，必须保持复位值。

位 1:0 **DMA_DEF0[1:0]**：数据流 0 类型 (Stream 0 type)
0：无
1：常规
2：双缓冲区
3：保留

16.5.13 DMA 版本寄存器 (DMA_VERR)

DMA version register

偏移地址：0x3F4

复位值：0x0000 0014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：DMA 主版本 (DMA major revision)

位 3:0 **MINREV[3:0]**：DMA 次版本 (DMA minor revision)

16.5.14 DMA 标识寄存器 (DMA_IPDR)

DMA identification register

偏移地址：0x3F8

复位值：0x0010 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：DMA 大小标识 (DMA size identification)

16.5.15 DMA 大小标识寄存器 (DMA_SIDR)

DMA size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识 (size identification)

该寄存器会将 DMA 标识为具有 1 KB 地址空间的外设。

16.5.16 DMA 寄存器映射

表 100. DMA 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	DMA_LISR	Res.	Res.	Res.	Res.	TCIF3	HTIF3	TEIF3	DMEIF3	Res.	FEIF3	TCIF2	HTIF2	TEIF2	DMEIF2	Res.	FEIF2	Res.	Res.	Res.	Res.	TCIF1	HTIF1	TEIF1	DMEIF1	Res.	FEIF1	TCIF0	HTIF0	TEIF0	DMEIF0	Res.	FEIF0
	Reset value					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0		0
0x004	DMA_HISR	Res.	Res.	Res.	Res.	TCIF7	HTIF7	TEIF7	DMEIF7	Res.	FEIF7	TCIF6	HTIF6	TEIF6	DMEIF6	Res.	FEIF6	Res.	Res.	Res.	Res.	TCIF5	HTIF5	TEIF5	DMEIF5	Res.	FEIF5	TCIF4	HTIF4	TEIF4	DMEIF4	Res.	FEIF4
	Reset value					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0		0
0x008	DMA_LIFCR	Res.	Res.	Res.	Res.	CTCIF3	CHTIF3	TEIF3	CDMEIF3	Res.	CFEIF3	CTCIF2	CHTIF2	CTEIF2	CDMEIF2	Res.	CFEIF2	Res.	Res.	Res.	Res.	CTCIF1	CHTIF1	CTEIF1	CDMEIF1	Res.	CFEIF1	CTCIF0	CHTIF0	CTEIF0	CDMEIF0	Res.	CFEIF0
	Reset value					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0		0

表 100. DMA 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00C	DMA_HIFCR	Res	Res	Res	Res	CTCIF7	CHTIF7	CTEIF7	CDMEIF7	Res	CFEIF7	CTCIF6	CHTIF6	CTEIF6	CDMEIF6	Res	CFEIF6	Res	Res	Res	Res	CTCIF5	CHTIF5	CTEIF5	CDMEIF5	Res	CFEIF5	CTCIF4	CHTIF4	CTEIF4	CDMEIF4	Res	CFEIF4
	Reset value					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0		0
0x010	DMA_S0CR	Res	Res	Res	Res	Res	Res	Res	MBURST [1:0]		PBURST [1:0]		TRBUFF	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014	DMA_S0NDTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	DMA_S0PAR	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x01C	DMA_S0M0AR	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	DMA_S0M1AR	M1A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	DMA_S0FCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FEIE	Res	FS[2:0]			DMDIS	FTH[1:0]	
	Reset value																									0		1	0	0	0	0	1
0x028	DMA_S1CR	Res	Res	Res	Res	Res	Res	Res	MBURST[1:0]		PBURST[1:0]		TRBUFF	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x02C	DMA_S1NDTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x030	DMA_S1PAR	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x034	DMA_S1M0AR	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x038	DMA_S1M1AR	M1A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03C	DMA_S1FCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FEIE	Res	FS[2:0]			DMDIS	FTH[1:0]	
	Reset value																									0		1	0	0	0	0	1
0x040	DMA_S2CR	Res	Res	Res	Res	Res	Res	Res	MBURST[1:0]		PBURST[1:0]		TRBUFF	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 100. DMA 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x044	DMA_S2NDTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x048	DMA_S2PAR	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04C	DMA_S2M0AR	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x050	DMA_S2M1AR	M1A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x054	DMA_S2FCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FEIE	Res	FS[2:0]			DMDIS	FTH[1:0]	
	Reset value																									0		1	0	0	0	0	1
0x058	DMA_S3CR	Res	Res	Res	Res	Res	Res	Res	MBURST [1:0]		PBURST [1:0]		TRBUFF	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x05C	DMA_S3NDTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x060	DMA_S3PAR	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x064	DMA_S3M0AR	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x068	DMA_S3M1AR	M1A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x06C	DMA_S3FCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FEIE	Res	FS[2:0]			DMDIS	FTH[1:0]	
	Reset value																									0		1	0	0	0	0	1
0x070	DMA_S4CR	Res	Res.	Res.	Res.	Res	Res	Res.	MBURST [1:0]		PBURST [1:0]		TRBUFF	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x074	DMA_S4NDTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x078	DMA_S4PAR	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07C	DMA_S4M0AR	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 100. DMA 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x080	DMA_S4M1AR	M1A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	DMA_S4FCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FEIE	Res	FS[2:0]			DMDIS	FTH[1:0]	
	Reset value																									0		1	0	0	0	0	1
0x088	DMA_S5CR	Res	Res	Res	Res	Res	Res	Res	MBURST [1:0]		PBURST [1:0]		TRBUFF	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08C	DMA_S5NDTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x090	DMA_S5PAR	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x094	DMA_S5M0AR	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x098	DMA_S5M1AR	M1A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x09C	DMA_S5FCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FEIE	Res	FS[2:0]			DMDIS	FTH[1:0]	
	Reset value																									0		1	0	0	0	0	1
0x0A0	DMA_S6CR	Res	Res	Res	Res	Res	Res	Res	MBURST [1:0]		PBURST [1:0]		TRBUFF	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0A4	DMA_S6NDTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0A8	DMA_S6PAR	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0AC	DMA_S6M0AR	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B0	DMA_S6M1AR	M1A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B4	DMA_S6FCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FEIE	Res	FS[2:0]			DMDIS	FTH[1:0]	
	Reset value																									0		1	0	0	0	0	1

表 100. DMA 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0B8	DMA_S7CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MBURST [1:0]		PBURST [1:0]		TRBUFF	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0BC	DMA_S7NDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C0	DMA_S7PAR	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C4	DMA_S7M0AR	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C8	DMA_S7M1AR	M1A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0CC	DMA_S7FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE	Res.	FS[2:0]			DMDIS	FTH[1:0]	
	Reset value																									0		1	0	0	0	0	1
0x0D0-0x3D8	Reserved	Reserved																															
0x3EC	DMA_HWCFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHSEL_WIDTH[2:0]			Res.	Res.	Res.	WRITE_BUFFERABLE	Res.	Res.	FIFO_SIZE[1:0]	
	Reset value																						0	0	0				0			0	1
0x3F0	DMA_HWCFGR1	Res.	Res.	DMA_DEF7		Res.	Res.	DMA_DEF6		Res.	Res.	DMA_DEF5		Res.	Res.	DMA_DEF4		Res.	Res.	DMA_DEF3		Res.	Res.	DMA_DEF2		Res.	Res.	DMA_DEF1		Res.	Res.	DMA_DEF0	
	Reset value			1	0			1	0			1	0			1	0			1	0			1	0			1	0			1	0
0x3F4	DMA_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	0	0	1	0	1	0	0
0x3F8	DMA_IPIDR	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0x3FC	DMA_SIDR	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 2.5 节*。

17 DMA 请求复用器 (DMAMUX)

17.1 简介

外设通过设置其 DMA 请求信号来指示存在 DMA 传输请求。DMA 请求信号在 DMA 控制器处理 DMA 请求并生成 DMA 确认信号之前一直处于挂起状态，之后相应的 DMA 请求信号就将变为无效。

在本文档中，DMA 请求/确认协议所需的控制信号组并未明确列出和说明，而是称作 DMA 请求线。

DMAMUX 请求复用器可在产品的外设和 DMA 控制器之间重新配置（路由）DMA 请求线。该路由功能通过可编程的多通道 DMA 请求线复用器来确保实现。每条通道可不受限制地选择一个 DMA 请求线，或者按照与来自其 DMAMUX 同步输入的事件相同步的方式选择一个 DMA 请求线。此外，DMAMUX 还可用作其输入触发信号的可编程事件的 DMA 请求发生器。

*第 17.3.1 节*规定了 DMAMUX 实例的数量及其主要特性。

以下的信号分配取决于产品实现：DMAMUX 请求复用器输入到外设 DMA 请求线和 DMAMUX 请求发生器输出的分配，DMAMUX 请求复用器输出到 DMA 控制器通道的分配以及 DMAMUX 同步和触发输入到内部和外部信号的分配，详细描述见*第 17.3.2 节*。

17.2 DMAMUX 主要特性

- 多达 16 通道可编程 DMA 请求线复用器输出
- 8 通道 DMA 请求发生器
- 8 个 DMA 请求发生器的触发输入
- 8 个同步输入
- 每个 DMA 请求发生器通道均具有：
  - DMA 请求触发输入选择器
  - DMA 请求计数器
  - 所选 DMA 请求触发输入的事件溢出标志
- 每个 DMA 请求线复用器通道输出均具有：
  - 多达 108 个来自外设的输入 DMA 请求线
  - 一个 DMA 请求线输出
  - 同步输入选择器
  - DMA 请求计数器
  - 所选同步输入的事件溢出标志
  - 一个事件输出，用于 DMA 请求链接
- 带有相关 IP 版本和硬件配置的标识寄存器。

17.3 DMAMUX 实现

17.3.1 DMAMUX 实例化

DMAMUX1 和 DMAMUX2通过下表中列出的硬件配置参数进行实例化。

表 101. DMAMUX1 和 DMAMUX2 实例化

功能	DMAMUX1	DMAMUX2
DMAMUX 输出请求通道数	16	8
DMAMUX 请求发生器通道数	8	8
DMAMUX 请求触发输入数	8	8
DMAMUX 同步输入数	8	8
DMAMUX 外设请求输入数	108	32

17.3.2 DMAMUX1 映射

资源到 DMAMUX1 的映射通过硬接线实现。

DMAMUX1 与 DMA1 和 DMA2 配合使用

- DMAMUX1 通道 0 到 7 与 DMA1 通道 0 到 7 相连
- DMAMUX1 通道 8 到 15 与 DMA2 通道 0 到 7 相连

表 102. DMAMUX1：复用器输入到资源的分配

DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源
1	dmamux1_req_gen0	44	USART2_TX	87	SAI1_A
2	dmamux1_req_gen1	45	USART3_RX	88	SAI1_B
3	dmamux1_req_gen2	46	USART3_TX	89	SAI2_A
4	dmamux1_req_gen3	47	TIM8_CH1	90	SAI2_B
5	dmamux1_req_gen4	48	TIM8_CH2	91	-
6	dmamux1_req_gen5	49	TIM8_CH3	92	-
7	dmamux1_req_gen6	50	TIM8_CH4	93	SPDIFRX_DT
8	dmamux1_req_gen7	51	TIM8_UP	94	SPDIFRX_CS
9	ADC1	52	TIM8_TRIG	95	-
10	ADC2	53	TIM8_COM	96	-
11	TIM1_CH1	54	-	97	-
12	TIM1_CH2	55	TIM5_CH1	98	-
13	TIM1_CH3	56	TIM5_CH2	99	-
14	TIM1_CH4	57	TIM5_CH3	100	-
15	TIM1_UP	58	TIM5_CH4	101	DFSDM1_FLT0
16	TIM1_TRIG	59	TIM5_UP	102	DFSDM1_FLT1
17	TIM1_COM	60	TIM5_TRIG	103	-

表 102. DMAMUX1：复用器输入到资源的分配（续）

DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源
18	TIM2_CH1	61	SPI3_RX	104	-
19	TIM2_CH2	62	SPI3_TX	105	TIM15_CH1
20	TIM2_CH3	63	UART4_RX	106	TIM15_UP
21	TIM2_CH4	64	UART4_TX	107	TIM15_TRIG
22	TIM2_UP	65	UART5_RX	108	TIM15_COM
23	TIM3_CH1	66	UART5_TX	109	TIM16_CH1
24	TIM3_CH2	67	-	110	TIM16_UP
25	TIM3_CH3	68	-	111	TIM17_CH1
26	TIM3_CH4	69	TIM6_UP	112	TIM17_UP
27	TIM3_UP	70	TIM7_UP	113	-
28	TIM3_TRIG	71	USART6_RX	114	-
29	TIM4_CH1	72	USART6_TX	115	I2C5_RX
30	TIM4_CH2	73	I2C3_RX	116	I2C5_TX
31	TIM4_CH3	74	I2C3_TX	117	保留
32	TIM4_UP	75	I2C4_RX	118	保留
33	I2C1_RX	76	I2C4_TX	119	保留
34	I2C1_TX	77	-	120	保留
35	I2C2_RX	78	-	121	保留
36	I2C2_TX	79	UART7_RX	122	保留
37	SPI1_RX	80	UART7_TX	123	保留
38	SPI1_TX	81	UART8_RX	124	保留
39	SPI2_RX	82	UART8_TX	125	保留
40	SPI2_TX	83	SPI4_RX	126	保留
41	USART1_RX	84	SPI4_TX	127	保留
42	USART1_TX	85	SPI5_RX	-	-
43	USART2_RX	86	SPI5_TX	-	-

表 103. DMAMUX1：触发输入到资源的分配

触发输入	资源	触发输入	资源
0	dmamux1_evt0	5	LPTIMER2_OUT
1	dmamux1_evt1	6	LPTIMER3_OUT
2	dmamux1_evt2	7	EXTI0
3	LPTIMER1_OUT	8	TIM12_TRGO

表 104. DMAMUX1：同步输入到资源的分配

同步输入	资源	同步输入	资源
0	dmamux1_evt0	5	LPTIMER2_OUT
1	dmamux1_evt1	6	LPTIMER3_OUT
2	dmamux1_evt2	7	EXTI0
3	LPTIMER1_OUT	8	TIM12_TRGO

17.3.3 DMAMUX2 映射

资源到 DMAMUX2 的映射通过硬接线实现。

DMAMUX2 与 DMA3 搭配使用：DMAMUX2 通道 0 到 7 与 DMA3 通道 0 到 7 相连。

表 105. DMAMUX2：复用器输入到资源的分配

DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源
1	dmamux2_req_gen0	33	保留
2	dmamux2_req_gen1	34	保留
3	dmamux2_req_gen2	35	保留
4	dmamux2_req_gen3	36	保留
5	dmamux2_req_gen4	37	保留
6	dmamux2_req_gen5	38	保留
7	dmamux2_req_gen6	39	保留
8	dmamux2_req_gen7	40	保留
9	ADC1	41	保留
10	ADC2	42	保留
11	I2C3_RX	43	保留
12	I2C3_TX	44	保留
13	SPI4_RX	45	保留
14	SPI4_TX	46	保留
15	USART1_RX	47	保留
16	USART1_TX	48	保留
17	USART2_RX	49	保留
18	USART2_TX	50	保留
19	I2C4_RX	51	保留
20	I2C4_TX	52	保留
21	I2C5_RX	53	保留
22	I2C5_TX	54	保留
23	SPI5_RX	55	保留
24	SPI5_TX	56	保留
25	TIM15_CH1	57	保留

表 105. DMAMUX2：复用器输入到资源的分配（续）

DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源
26	TIM15_UP	58	保留
27	TIM15_TRIG	59	保留
28	TIM15_COM	60	保留
29	TIM16_CH1	61	保留
30	TIM16_UP	62	保留
31	TIM17_CH1	63	保留
32	TIM17_UP	64	保留

表 106. DMAMUX2：触发输入到资源的分配

触发输入	资源	触发输入	资源
0	dmamux2_evt0	4	保留
1	dmamux2_evt1	5	保留
2	dmamux2_evt2	6	保留
3	保留	7	TIM12_TRGO

表 107. DMAMUX2：同步输入到资源的分配

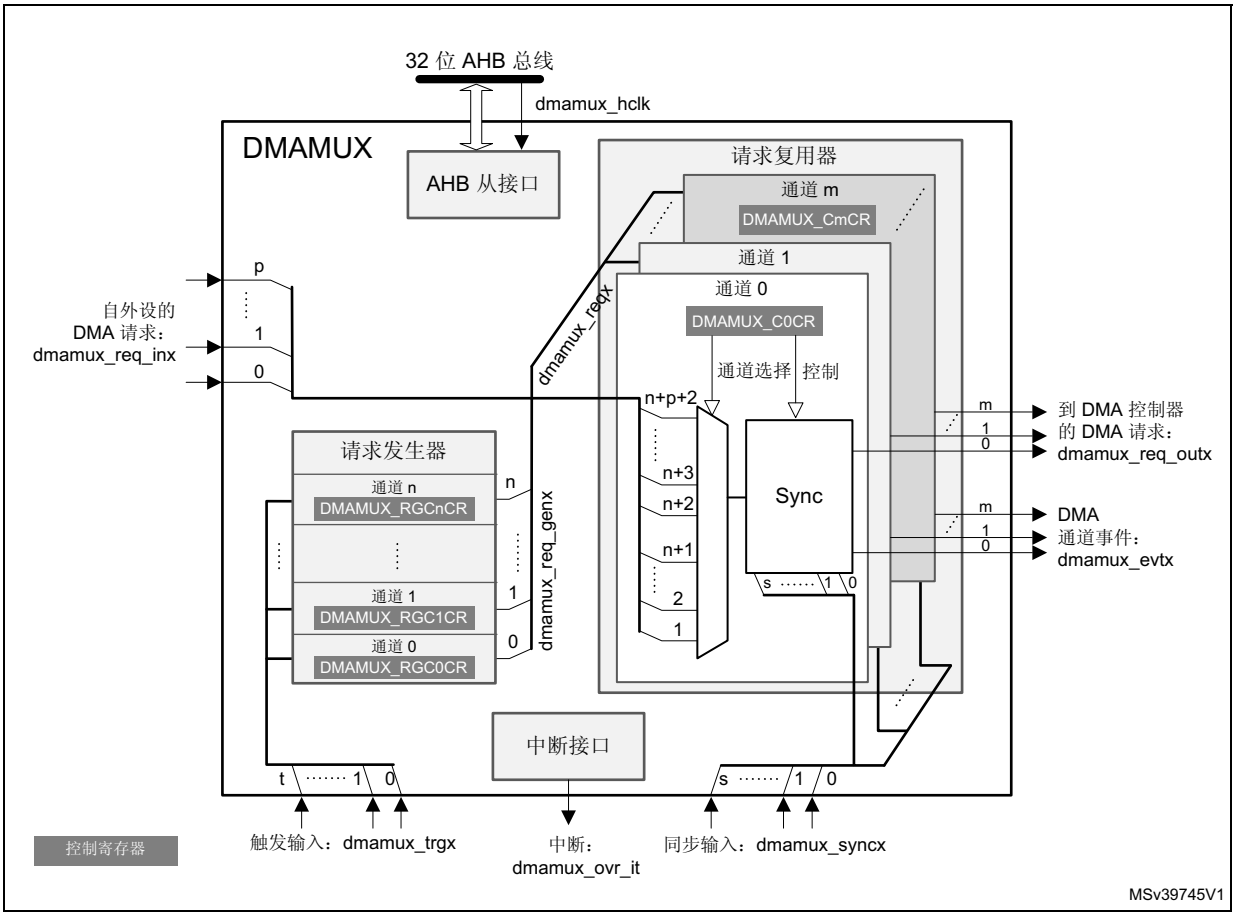
同步输入	资源	同步输入	资源
0	dmamux2_evt0	4	保留
1	dmamux2_evt1	5	保留
2	dmamux2_evt2	6	保留
3	保留	7	TIM12_TRGO

17.4 DMAMUX 功能描述

17.4.1 DMAMUX 框图

图 117 显示了 DMAMUX 框图。

图 117. DMAMUX 框图



DMAMUX 具有两个主要子模块：请求线复用器和请求线发生器。

该实现可：

- 分配 DMAMUX 请求复用器子模块输入 (dmamux_reqx)，这些输入来自外设 (dmamux_req_inx) 和 DMAMUX 请求发生器子模块的通道 (dmamux_req_genx)
- 将 DMAMUX 请求输出分配到 DMA 控制器的通道 (dmamux_req_outx)
- 将内部或外部信号分配到 DMA 请求触发输入 (dmamux_trgx)
- 将内部或外部信号分配到同步输入 (dmamux_syncx)

17.4.2 DMAMUX 信号

表 108 列出了 DMAMUX 信号。

表 108. DMAMUX 信号

信号名称	说明
dmamux_hclk	DMAMUX AHB 时钟
dmamux_req_inx	DMAMUX DMA 请求线输入（来自外设）
dmamux_trgx	DMAMUX DMA 请求触发输入（到请求发生器子模块）
dmamux_req_genx	DMAMUX 请求发生器子模块通道输出
dmamux_reqx	DMAMUX 请求复用器子模块输入（来自外设请求和请求发生器通道）
dmamux_syncx	DMAMUX 同步输入（到请求复用器子模块）
dmamux_req_outx	DMAMUX 请求输出（到 DMA 控制器）
dmamux_evtx	DMAMUX 事件输出
dmamux_ovr_it	DMAMUX 溢出中断

17.4.3 DMAMUX 通道

DMAMUX 通道是一个可能包括额外的 DMAMUX 请求发生器通道的请求复用器通道，具体取决于所选的请求复用器输入。

一个 DMAMUX 请求复用器通道连接到 DMA 控制器的一个通道且专用于这个通道。

通道配置流程

请遵循以下顺序来配置 DMAMUX x 通道和相关的 DMA 通道 y：

1. 对 DMA 通道 y 进行完整的设置和配置，但不要使能通道 y。
2. 对相关 DMAMUX y 通道进行完整的设置和配置。
3. 最后，将 DMA y 通道寄存器中的 EN 位置 1 以激活 DMA 通道。

17.4.4 DMAMUX 请求线复用器

DMAMUX 请求复用器及其多条通道可确保 DMA 请求/确认控制信号（称为 DMA 请求线）的实际路由。

每个 DMA 请求线并行连接到 DMAMUX 请求线复用器的所有通道。

DMA 请求来自外设或 DMAMUX 请求发生器。

DMAMUX 请求线复用器通道 x 按照 DMAMUX_CxCR 寄存器中的 DMAREQ_ID 字段的配置来选择 DMA 请求线编号。

注： *字段 DMAREQ_ID 为空值表示未选择任何 DMA 请求线。*

小心： 除非应用程序可保证所连接的两个 DMA 通道不会同时激活，否则不得为不同的 x 和 y DMAMUX 请求复用器通道编程相同的非空 DMAREQ_ID（通过 DMAMUX_CxCR 和 DMAMUX CyCR）。

除了 DMA 请求选择外，还可根据需要配置和使能同步模式和/或事件生成。

同步模式和通道事件生成

每个 DMAMUX 请求线复用器通道 x 可单独同步，方法是将 DMAMUX_CxCR 寄存器中的同步使能 (SE) 位置 1。

DMAMUX 具有多个同步输入。同步输入并行连接到请求复用器的所有通道。

同步输入通过给定通道 x 的 DMAMUX_CxCR 寄存器中的 SYNC_ID 字段选择。

当某个通道处于此同步模式时，如果通过 DMAMUX_CxCR 寄存器的 SPOL[1:0] 字段在所选输入同步信号上检测到可配置的上升沿/下降沿，则所选输入 DMA 请求线信号将被传送到复用器通道输出。

此外，DMAMUX 请求复用器内部有一个可编程 DMA 请求计数器，可用于实现通道请求输出生成功能以及事件生成功能。通道 x 输出上的事件生成功能通过 DMAMUX_CxCR 寄存器的 EGE 位（事件生成使能）使能。

如*图 119* 所示，检测到同步输入的边沿后，挂起的所选输入 DMA 请求线将连接到 DMAMUX 复用器通道 x 输出。

注： *如果在不存在挂起的所选输入 DMA 请求线的情况下发生同步事件，则会将其丢弃。再次发生同步事件之前，后续有效的输入请求线不会连接至 DMAMUX 复用器通道输出。*

此后，DMA 控制器每次处理连接的 DMAMUX 请求时（已处理请求被禁止），DMAMUX 请求计数器都会递减。下溢时，DMA 请求计数器将自动加载 DMAMUX_CxCR 寄存器的 NBREQ 字段中的值，输入 DMA 请求线将与复用器通道 x 输出断开。

因此，检测到同步事件后传送到复用器通道 x 输出的 DMA 请求数等于 NBREQ 字段中的值加 1。

注： *只能在相应复用器通道 x 的同步使能位 (SE) 和事件生成使能位 (EGE) 均禁止时通过软件写入 NBREQ 字段的值。*

图 118. DMAMUX 请求线复用器通道的同步模式

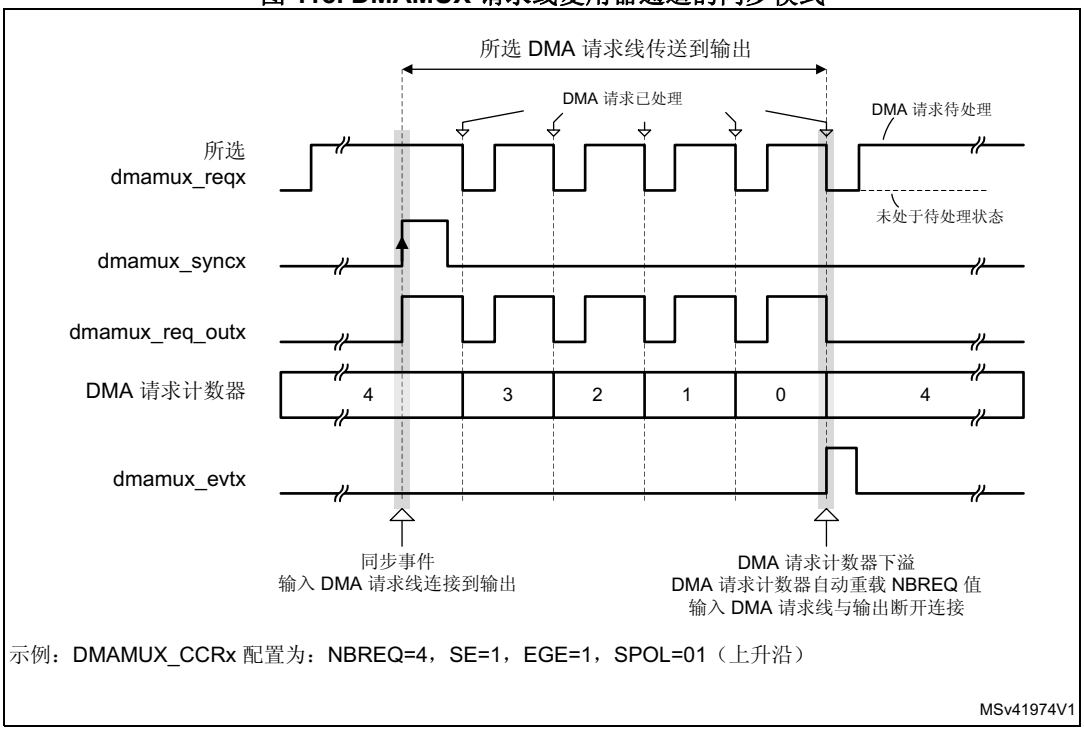
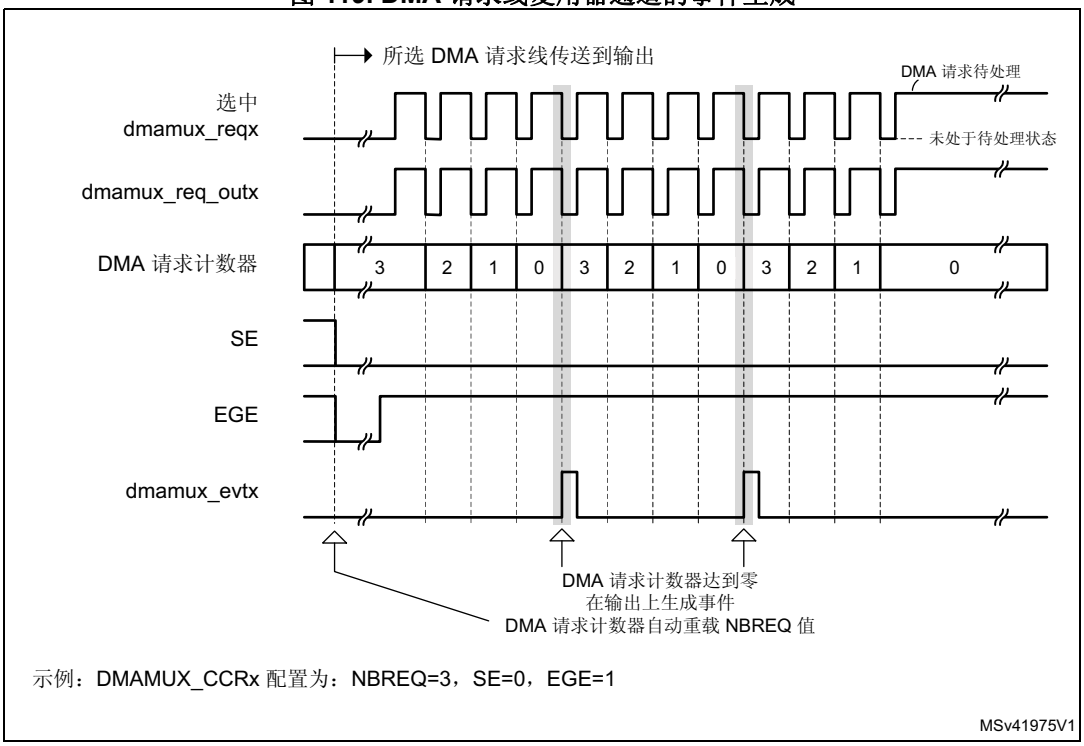


图 119. DMA 请求线复用器通道的事件生成



如果 EGE 使能，当复用器通道的 DMA 请求计数器自动重新装入已编程 NBREQ 字段的值时，复用器通道会生成一个通道事件（即，一个 AHB 时钟周期的脉冲），如*图 118* 和*图 119* 所示。

注： *如果 EGE 使能且 NBREQ = 0，则在每个处理的 DMA 请求后都会生成一个事件。*

注： *如果边沿后的状态保持稳定的时间持续 2 个 AHB 时钟周期以上，则会检测到同步事件（边沿）。*

写入 DMAMUX_CxCR 寄存器后，同步事件将被屏蔽 3 个 AHB 时钟周期。

同步溢出和中断

如果在请求计数器下溢（通过 DMAMUX_CxCR 寄存器的 NBREQ 字段编程的内部请求计数器）之前发生新的同步事件，则 DMAMUX_CSR 寄存器中的同步溢出标志位 SOFx 会置 1。

注： *DMA 控制器的相关通道使用结束后，必须禁止请求复用器通道 x 同步 (DMAMUX_CxCR.SE = 0)。否则，在新检测到同步事件后，由于未从 DMA 控制器收到 DMA 确认（即无已处理请求），将会发生同步溢出。*

溢出标志 SOFx 的复位方式是将 DMAMUX_CFR 寄存器中的相关清除同步溢出标志位 CSOFx 置 1。

如果 DMAMUX_CxCR 寄存器中的同步溢出中断使能位 SOIE 置 1，则将同步溢出标志置 1 会产生中断。

17.4.5 DMAMUX 请求发生器

DMAMUX 请求发生器会在其 DMA 请求触发输入上出现触发事件后产生 DMA 请求。

DMAMUX 请求发生器有多个通道。DMA 请求触发输入并行连接到所有通道。

DMAMUX 请求发生器通道的输出是 DMAMUX 请求线复用器的输入。

每个 DMAMUX 请求发生器通道 x 在相应的 DMAMUX_RGxCR 寄存器中都有一个使能位 GE（发生器使能）。

DMAMUX 请求发生器通道 x 的 DMA 请求触发输入通过相应 DMAMUX_RGxCR 寄存器中的 SIG_ID（触发信号 ID）字段选择。

DMA 请求触发输入上的触发事件可以是上升沿、下降沿或任一边沿。有效边沿通过相应 DMAMUX_RGxCR 寄存器中的 GPOL（发生器极性）字段选择。

触发事件后，相应发生器通道开始在其输出上生成 DMA 请求。连接的 DMA 控制器每次处理 DMAMUX 生成的请求时（已处理请求被禁止），内置（DMAMUX 请求发生器内）DMA 请求计数器都会递减。出现下溢时，请求生成器通道会停止生成 DMA 请求，DMA 请求计数器将在收到下一个触发事件时自动重新加载其设定值。

因此，触发事件后生成的 DMA 请求的数量为 GNBREQ + 1。

注： *只能在相应发生器通道 x 的使能位 GE 被禁止时通过软件写入 GNBREQ 字段的值。*

没有硬件写保护。

如果边沿后的状态保持稳定的时间持续 2 个 AHB 时钟周期以上，则会检测到触发事件（边沿）。

写入 DMAMUX_RGxCR 寄存器后，触发事件将被屏蔽 3 个 AHB 时钟周期。

触发溢出和中断

如果在 DMAMUX 请求发生器计数器下溢（通过 DMAMUX_RGxCR 寄存器的 GNBREQ 字段编程的内部计数器）之前发生新的 DMA 请求触发事件，并且请求发生器通道 x 通过 GE 使能，则 DMAMUX_RGSR 寄存器中的请求触发事件溢出标志位 OFx 将通过硬件置为有效。

注： *DMA 控制器的相关通道使用结束后，必须禁止请求发生器通道 x (DMAMUX_RGxCR.GE = 0)。否则，在新检测到触发事件后，由于未从 DMA 收到确认（即无已处理请求），将会发生触发溢出。*

溢出标志 OFx 的复位方式是将 DMAMUX_RGCFR 寄存器中的相关清除溢出标志位 COFx 置 1。

如果 DMAMUX_RGxCR 寄存器中的 DMA 请求触发事件溢出中断使能位 OIE 置 1，则将 DMAMUX 请求触发溢出标志置 1 会产生中断。

17.5 DMAMUX 中断

发生以下事件时会产生中断：

- 每个 DMA 请求线复用器通道发生同步事件溢出
- 每个 DMA 请求发生器通道发生触发事件溢出

对于任一事件，每个通道的中断使能、状态和清除标志寄存器位均可单独使用。

表 109. DMAMUX 中断

中断信号	中断事件	事件标志	清除位	使能位
dmamuxovr_it	DMAMUX 请求线复用器的通道 x 上发生同步事件溢出	SOFx	CSOFx	SOIE
	DMAMUX 请求发生器的通道 x 上发生触发事件溢出	OFx	COFx	OIE

17.6 DMAMUX 寄存器

有关 DMAMUX 基址的信息，请参见包括寄存器边界地址的表格。

DMAMUX 寄存器可按字节（8 位）、半字（16 位）或字（32 位）进行访问。地址必须与数据大小对齐。

17.6.1 DMAMUX 请求线复用器通道 x 配置寄存器 (DMAMUX_CxCR)

DMAMUX request line multiplexer channel x configuration register

偏移地址：0x000 + 0x04 * x（DMAMUX1 为 x = 0 到 15）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SYNC_ID[2:0]			NBREQ[4:0]					SPOL[1:0]		SE
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMA REQ_ID[6]	DMAREQ_ID[5:0]					
						rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:27 保留，必须保持复位值。

位 26:24 **SYNC_ID[2:0]**：同步标识 (Synchronization identification)

选择同步输入（请参见*表 104：DMAMUX1：同步输入到资源的分配*和*表 107：DMAMUX2：同步输入到资源的分配*）。

位 23:19 **NBREQ[4:0]**：要转发的 DMA 请求数减 1 (Number of DMA requests minus 1 to forward)

定义在同步事件后要转发到 DMA 控制器的 DMA 请求的数量，和/或生成输出事件前的 DMA 请求的数量。

只有当 SE 和 EGE 位均为低电平时，才能写入该字段。

位 18:17 **SPOL[1:0]**：同步极性 (Synchronization polarity)

定义所选同步输入的边沿极性：

00：无事件（无同步也无检测）。

01：上升沿

10：下降沿

11：上升沿和下降沿

位 16 **SE**：同步使能 (Synchronization enable)

0：禁止同步

1：使能同步

位 15:10 保留，必须保持复位值。

位 9 **EGE**：事件生成使能 (Event generation enable)

0：禁止事件生成

1：使能事件生成

位 8 **SOIE**：同步上溢中断使能 (Synchronization overrun interrupt enable)

0：禁止中断

1：使能中断

位 7 保留，必须保持复位值。

位 6 **DMAREQ_ID[6]**：DMA 请求标识 (DMA request identification)
选择输入 DMA 请求。有关复用器输入到资源的分配的信息，请参见 DMAMUX 表。
对于 DMAMUX2，该位保留且必须保持复位值。

位 5:0 **DMAREQ_ID[5:0]**：DMA 请求标识 (DMA request identification)
选择输入 DMA 请求。有关复用器输入到资源的分配的信息，请参见 DMAMUX 表。

17.6.2 DMAMUX 请求线复用器通道 x 配置寄存器 [备用] (DMAMUX_CxCR)

DMAMUX request line multiplexer channel x configuration register

偏移地址：0x000 + 0x04 * x（DMAMUX2 为 x = 0 到 7）

复位值：0x0000 0000

根据目标 DMAMUX 请求线复用器通道 x 的安全模式，基于所连接的 DMA 控制器通道 y 的安全模式位，该寄存器必须通过非安全写入或安全写入操作进行写入。前提条件是 DMAMUX x 通道输出连接到 DMA 的 y 通道（请参见 DMAMUX 映射实现部分）。

根据目标 DMAMUX 请求线复用器通道 x 的特权模式，基于所连接的 DMA 控制器通道 y 的特权控制位，该寄存器必须通过特权或非特权读/写操作进行访问。前提条件是 DMAMUX x 通道输出连接到 DMA 的 y 通道（请参见 DMAMUX 映射实现部分）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SYNC_ID[2:0]			NBREQ[4:0]					SPOL[1:0]		SE
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMA REQ_ID[6]	DMAREQ_ID[5:0]					
						rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:27 保留，必须保持复位值。

位 26:24 **SYNC_ID[2:0]**：同步标识 (Synchronization identification)
选择同步输入（请参见*表 104：DMAMUX1：同步输入到资源的分配*和*表 107：DMAMUX2：同步输入到资源的分配*）。

位 23:19 **NBREQ[4:0]**：要转发的 DMA 请求数减 1 (Number of DMA requests minus 1 to forward)
定义在同步事件后要转发到 DMA 控制器的 DMA 请求的数量，和/或生成输出事件前的 DMA 请求的数量。
只有当 SE 和 EGE 位均为低电平时，才能写入该字段。

位 18:17 **SPOL[1:0]**：同步极性 (Synchronization polarity)
定义所选同步输入的边沿极性：
00：无事件（无同步也无检测）。
01：上升沿
10：下降沿
11：上升沿和下降沿

位 16 **SE**：同步使能 (Synchronization enable)
0：禁止同步
1：使能同步

位 15:10 保留，必须保持复位值。

位 9 **EGE**：事件生成使能 (Event generation enable)
0：禁止事件生成
1：使能事件生成

位 8 **SOIE**：同步上溢中断使能 (Synchronization overrun interrupt enable)
0：禁止中断
1：使能中断

位 7 保留，必须保持复位值。

位 6 **DMAREQ_ID[6]**：DMA 请求标识 (DMA request identification)
选择输入 DMA 请求。有关复用器输入到资源的分配的信息，请参见 DMAMUX 表。
对于 DMAMUX2，该位保留且必须保持复位值。

位 5:0 **DMAREQ_ID[5:0]**：DMA 请求标识 (DMA request identification)
选择输入 DMA 请求。有关复用器输入到资源的分配的信息，请参见 DMAMUX 表。

17.6.3 DMAMUX 请求线复用器中断通道状态寄存器 (DMAMUX_CSR)

DMAMUX request line multiplexer interrupt channel status register

偏移地址：0x080

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOF15	SOF14	SOF13	SOF12	SOF11	SOF10	SOF9	SOF8	SOF7	SOF 6	SOF 5	SOF4	SOF3	SOF2	SOF1	SOF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **SOF[15:0]**：同步溢出事件标志 (Synchronization overrun event flag)
当 DMA 请求线复用器通道 x 上发生同步事件且 DMA 请求计数器的值小于 NBREQ 时，该标志将置 1。
该标志的清零方式是向 DMAMUX_CFR 寄存器中的相应 CSOFx 位写入 1。对于 DMAMUX2，位 15:8 将保留且必须保持复位值。

17.6.4 DMAMUX 请求线复用器中断清除标志寄存器 (DMAMUX_CFR)

DMAMUX request line multiplexer interrupt clear flag register

偏移地址：0x084

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSOF 15	CSOF 14	CSOF 13	CSOF 12	CSOF 11	CSOF 10	CSOF 9	CSOF 8	CSOF 7	CSOF 6	CSOF 5	CSOF 4	CSOF 3	CSOF 2	CSOF 1	CSOF 0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:16 保留，必须保持复位值。

位 15:0 **CSOF[15:0]**：清零同步溢出事件标志 (Clear synchronization overrun event flag)
将 1 写入每个位时，DMAMUX_CSR 寄存器中相应的溢出标志 SOFx 将清零。对于 DMAMUX2，位 15:8 将保留且必须保持复位值。

17.6.5 DMAMUX 请求发生器通道 x 配置寄存器 (DMAMUX_RGxCR)

DMAMUX request generator channel x configuration register

偏移地址：0x100 + 0x04 * x（x = 0 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL[1:0]		GE
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	Res.	Res.	SIG_ID[2:0]		
							rw						rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:19 **GNBREQ[4:0]**：要生成的 DMA 请求数（减 1）(Number of DMA requests to be generated (minus 1))
定义触发事件后生成的 DMA 请求的数量。生成的 DMA 请求的实际数量为 GNBREQ+1。
注意：只有在禁止 GE 后才能写入该字段。

位 18:17 **GPOL[1:0]**：DMA 请求发生器触发极性 (DMA request generator trigger polarity)
定义所选触发输入的边沿极性
00：无事件，即无触发检测和生成。
01：上升沿
10：下降沿
11：上升沿和下降沿

位 16 **GE**：DMA 请求发生器通道 x 使能 (DMA request generator channel x enable)
0：禁止 DMA 请求发生器通道 x
1：使能 DMA 请求发生器通道 x

位 15:9 保留，必须保持复位值。

位 8 **OIE**：触发上溢中断使能 (Trigger overrun interrupt enable)
0：禁止在出现触发溢出事件时产生中断。
1：使能在出现触发溢出事件时产生中断。

位 7:3 保留，必须保持复位值。

位 2:0 **SIG_ID[2:0]**：信号标识 (Signal identification)
选择用于 DMA 请求发生器的通道 x 的 DMA 请求触发输入

17.6.6 DMAMUX 请求发生器中断状态寄存器 (DMAMUX_RGSR)

DMAMUX request generator interrupt status register

偏移地址：0x140

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OF7	OF6	OF5	OF4	OF3	OF2	OF1	OF0
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **OF[7:0]**：触发溢出事件标志 (Trigger overrun event flag)
在请求计数器下溢（通过 DMAMUX_RGxCR 寄存器的 GNBREQ 字段编程的内部请求计数器）之前，在 DMA 请求发生器通道 x 上发生新触发事件时，该标志将置 1。
该标志的清零方式是向 DMAMUX_RGCFR 寄存器中的相应 COFx 位写入 1。

17.6.7 DMAMUX 请求发生器中断清除标志寄存器 (DMAMUX_RGCFR)

DMAMUX request generator interrupt clear flag register

偏移地址：0x144

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COF7	COF6	COF5	COF4	COF3	COF2	COF1	COF0
								w	w	w	w	w	w	w	w

位 31:8 保留，必须保持复位值。

位 7:0 **COF[7:0]**：清零触发溢出事件标志 (Clear trigger overrun event flag)
将 1 写入每个位时，DMAMUX_RGSR 寄存器中相应的溢出标志 OFx 将清零。

17.6.8 DMAMUX 大小标识寄存器 (DMAMUX_SIDR)

DMAMUX size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识 (Size identification)
此寄存器将标识 1 KB 地址空间的 IP 大小。

17.6.9 DMAMUX IP 标识寄存器 (DMAMUX_IPIDR)

DMAMUX IP identification register

偏移地址：0x3F8

复位值：0x0010 0011

此寄存器将标识 IP。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：IP 标识 (IP identification)

17.6.10 DMAMUX 版本寄存器 (DMAMUX_VERR)

DMAMUX version register

偏移地址：0x3F4

复位值：0x0000 0011

此寄存器将标识 IP 版本。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主 IP 版本 (Major IP revision)

位 3:0 **MINREV[3:0]**：次 IP 版本 (Minor IP revision)

17.6.11 DMAMUX 硬件配置 1 寄存器 (DMAMUX_HWCFGR1)

DMAMUX hardware configuration 1 register

偏移地址：0x3F0

复位值：0x0808 6C10（适用于 DMAMUX1，DMAMUX2 为 0x08082008）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUM_DMA_REQGEN[7:0]								NUM_DMA_TRIG[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUM_DMA_PERIPH_REQ[7:0]								NUM_DMA_STREAMS[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 **NUM_DMA_REQGEN[7:0]**：DMA 请求发生器通道数 (number of DMA request generator channels)

位 23:16 **NUM_DMA_TRIG[7:0]**：同步输入数 (number of synchronization inputs)

位 15:8 **NUM_DMA_PERIPH_REQ[7:0]**：来自外设的 DMA 请求线数 (number of DMA request lines from peripherals)

位 7:0 **NUM_DMA_STREAMS[7:0]**：DMA 请求线复用器（输出）通道数 (number of DMA request line multiplexer (output) channels)

17.6.12 DMAMUX 硬件配置 2 寄存器 (DMAMUX_HWCFGR2)

DMAMUX hardware configuration 2 register

偏移地址：0x3EC

复位值：0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NUM_DMA_EXT_REQ[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **NUM_DMA_EXT_REQ[7:0]:** DMA 请求触发输入数 (Number of DMA request trigger inputs)

17.6.13 DMAMUX 寄存器映射

下表汇总了 DMAMUX 寄存器和复位值。有关 DMAMUX 寄存器的基址，请参见寄存器边界地址表。

表 110. DMAMUX 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	DMAMUX_C0CR	Res.	Res.	Res.	Res.	Res.	SYNC_ID [2:0]			NBREQ[4:0]					SPOL [1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0](1)						
	Reset value						0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x004	DMAMUX_C1CR	Res.	Res.	Res.	Res.	Res.	SYNC_ID [2:0]			NBREQ[4:0]					SPOL [1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0](1)						
	Reset value						0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x008	DMAMUX_C2CR	Res.	Res.	Res.	Res.	Res.	SYNC_ID [2:0]			NBREQ[4:0]					SPOL [1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0](1)						
	Reset value						0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x00C	DMAMUX_C3CR	Res.	Res.	Res.	Res.	Res.	SYNC_ID [2:0]			NBREQ[4:0]					SPOL [1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0](1)						
	Reset value						0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x010	DMAMUX_C4CR	Res.	Res.	Res.	Res.	Res.	SYNC_ID [2:0]			NBREQ[4:0]					SPOL [1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0](1)						
	Reset value						0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x014	DMAMUX_C5CR	Res.	Res.	Res.	Res.	Res.	SYNC_ID [2:0]			NBREQ[4:0]					SPOL [1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0](1)						
	Reset value						0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x018	DMAMUX_C6CR	Res.	Res.	Res.	Res.	Res.	SYNC_ID [2:0]			NBREQ[4:0]					SPOL [1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0](1)						
	Reset value						0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x01C	DMAMUX_C7CR	Res.	Res.	Res.	Res.	Res.	SYNC_ID [2:0]			NBREQ[4:0]					SPOL [1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0](1)						
	Reset value						0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x020	DMAMUX_C8CR(2)	Res.	Res.	Res.	Res.	Res.	SYNC_ID [2:0]			NBREQ[4:0]					SPOL [1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value						0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0

表 110. DMAMUX 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x024	DMAMUX_C9CR(2)	Res.	Res.	Res.	Res.	Res.	SYNC_ID [2:0]			NBREQ[4:0]					SPOL [1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value						0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x028	DMAMUX_C10CR(2)	Res.	Res.	Res.	Res.	Res.	SYNC_ID [2:0]			NBREQ[4:0]					SPOL [1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value						0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x02C	DMAMUX_C11CR(2)	Res.	Res.	Res.	Res.	Res.	SYNC_ID [2:0]			NBREQ[4:0]					SPOL [1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x030	DMAMUX_C12CR(2)	Res.	Res.	Res.	Res.	Res.	SYNC_ID [2:0]			NBREQ[4:0]					SPOL [1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value						0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x034	DMAMUX_C13CR(2)	Res.	Res.	Res.	Res.	Res.	SYNC_ID [2:0]			NBREQ[4:0]					SPOL [1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value						0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x038	DMAMUX_C14CR(2)	Res.	Res.	Res.	Res.	Res.	SYNC_ID [2:0]			NBREQ[4:0]					SPOL [1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value						0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x03C	DMAMUX_C15CR(2)	Res.	Res.	Res.	Res.	Res.	SYNC_ID [2:0]			NBREQ[4:0]					SPOL [1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x040 - 0x07C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x080	DMAMUX_CSR(3)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOF15	SOF14	SOF13	SOF12	SOF11	SOF10	SOF9	SOF8	SOF7	SOF6	SOF5	SOF4	SOF3	SOF2	SOF1	SOF0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	DMAMUX_CFR(4)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSOF15	CSOF14	CSOF13	CSOF12	CSOF11	CSOF10	CSOF9	CSOF8	CSOF7	CSOF6	CSOF5	CSOF4	CSOF3	CSOF2	CSOF1	CSOF0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x088 - 0x0FC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x100	DMAMUX_RG0CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL [1:0]		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	Res.	Res.	SIG_ID [3:0]		
	Reset value									0	0	0	0	0	0	0	0								0						0	0	0
0x104	DMAMUX_RG1CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL [1:0]		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	Res.	Res.	SIG_ID [3:0]		
	Reset value									0	0	0	0	0	0	0	0								0						0	0	0
0x108	DMAMUX_RG2CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL [1:0]		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	Res.	Res.	SIG_ID [3:0]		
	Reset value									0	0	0	0	0	0	0	0								0						0	0	0
0x10C	DMAMUX_RG3CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL [1:0]		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	Res.	Res.	SIG_ID [3:0]		
	Reset value									0	0	0	0	0	0	0	0								0						0	0	0
0x110	DMAMUX_RG4CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL [1:0]		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	Res.	Res.	SIG_ID [3:0]		
	Reset value									0	0	0	0	0	0	0	0								0						0	0	0
0x114	DMAMUX_RG5CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL [1:0]		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	Res.	Res.	SIG_ID [3:0]		
	Reset value									0	0	0	0	0	0	0	0								0						0	0	0
0x118	DMAMUX_RG6CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL [1:0]		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	Res.	Res.	SIG_ID [3:0]		
	Reset value									0	0	0	0	0	0	0	0								0						0	0	0
0x11C	DMAMUX_RG7CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL [1:0]		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	Res.	Res.	SIG_ID [3:0]		
	Reset value									0	0	0	0	0	0	0	0								0						0	0	0

表 110. DMAMUX 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x120 - 0x13C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x140	DMAMUX_RGSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OF7	OF6	OF5	OF4	OF3	OF2	OF1	OF0
	Reset value																									0	0	0	0	0	0	0	0
0x144	DMAMUX_RGCFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COF7	COF6	COF5	COF4	COF3	COF2	COF1	COF0
	Reset value																									0	0	0	0	0	0	0	0
0x148 - 0x3E8	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x3EC	DMAMUX _HWCFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NUM_DMA_EXT_REQ							
	Reset value																									0	0	0	0	1	0	0	0
0x3F0	DMAMUX _HW_CFGR1	NUM_DMA_REQGEN								NUM_DMA_TRIG								NUM_DMA_PERIF_REQ								NUM_DMA_STREAMS							
	Reset value[5]	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	1	0	0	0	0	0	1	0	0	0	0
0x3F4	DMAMUX_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	0	0	1	0	0	0	1
0x3F8	DMAMUX_IPIDR	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
0x3FC	DMAMUX_SIDR	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

1. 对于 DMAMUX2，位 DMAREQ[6] 将保留且必须保持复位值。
2. 仅适用于 DMAMUX1。
3. 对于 DMAMUX2，位 SOF[15:8] 将保留且必须保持复位值。
4. 对于 DMAMUX2，位 CSOF[15:8] 将保留且必须保持复位值。
5. 有关 DMAMUX2 的信息，请参见*第 17.6.11 节*。

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

18 中断列表

18.1 GIC 中断

表 111. 基于 STM32MP13xx 上 Cortex®-A7 的 GIC 中断映射

标号	ID	缩略语	说明	EXTI 事件(1)
SGI（上升沿触发）				
0	0	SGI0	软件触发中断 0（建议在非安全模式下）	-
1	1	SGI1	软件触发中断 1（建议在非安全模式下）	-
2	2	SGI2	软件触发中断 2（建议在非安全模式下）	-
3	3	SGI3	软件触发中断 3（建议在非安全模式下）	-
4	4	SGI4	软件触发中断 4（建议在非安全模式下）	-
5	5	SGI5	软件触发中断 5（建议在非安全模式下）	-
6	6	SGI6	软件触发中断 6（建议在非安全模式下）	-
7	7	SGI7	软件触发中断 7（建议在非安全模式下）	-
8	8	SGI8	软件触发中断 8（建议在安全模式下）	-
9	9	SGI9	软件触发中断 9（建议在安全模式下）	-
10	10	SGI10	软件触发中断 10（建议在安全模式下）	-
11	11	SGI11	软件触发中断 11（建议在安全模式下）	-
12	12	SGI12	软件触发中断 12（建议在安全模式下）	-
13	13	SGI13	软件触发中断 13（建议在安全模式下）	-
14	14	SGI14	软件触发中断 14（建议在安全模式下）	-
15	15	SGI15	软件触发中断 15（建议在安全模式下）	-
PPI（有效低电平敏感）				
-	16 到 14	-	保留	-
-	25	PPI6	虚拟维护中断。	-
-	26	PPI5	管理程序定时器事件。	-
-	27	PPI4	虚拟定时器事件。	-
-	28	PPI0	传统 nFIQ 信号。未使用。	-
-	29	PPI1	安全物理定时器事件。	-
-	30	PPI2	非安全物理定时器事件。	-
-	31	PPI3	传统 nIRQ 信号。未使用。	-
SPI（有效高电平敏感或上升沿触发）				
0	32	-	保留	-
1	33	PVD_AVD	PVD & AVD 通过 EXTI 检测	16

表 111. 基于 STM32MP13xx 上 Cortex®-A7 的 GIC 中断映射（续）

标号	ID	缩略语	说明	EXTI 事件(1)
2	34	TAMP	防篡改中断（包括 LSECSS 中断）	(18)
3	35	RTC_WKUP_ALARM	RTC 唤醒定时器和闹钟（A 和 B）中断	(19)
4	36	TZC_IT	TrustZone DDR 地址空间控制器	-
5	37	RCC	RCC 全局中断	-
6	38	EXTI0	EXTI 线 0 中断	0
7	39	EXTI1	EXTI 线 1 中断	1
8	40	EXTI2	EXTI 线 2 中断	2
9	41	EXTI3	EXTI 线 3 中断	3
10	42	EXTI4	EXTI 线 4 中断	4
11	43	DMA1_STR0	DMA1 数据流 0 全局中断	-
12	44	DMA1_STR1	DMA1 数据流 1 全局中断	-
13	45	DMA1_STR2	DMA1 数据流 2 全局中断	-
14	46	DMA1_STR3	DMA1 数据流 3 全局中断	-
15	47	DMA1_STR4	DMA1 数据流 4 全局中断	-
16	48	DMA1_STR5	DMA1 数据流 5 全局中断	-
17	49	DMA1_STR6	DMA1 数据流 6 全局中断	-
18	50	ADC1	ADC1 全局中断	-
19	51	ADC2	ADC2 全局中断	-
20	52	FDCAN1_IT0	FDCAN1 中断 0	-
21	53	FDCAN2_IT0	FDCAN2 中断 0	-
22	54	FDCAN1_IT1	FDCAN3 中断 1	-
23	55	FDCAN2_IT1	FDCAN2 中断 1	-
24	56	EXTI5	EXTI 线 5 中断	5
25	57	TIM1_BRK	TIM1 暂停中断	-
26	58	TIM1_UP	TIM1 更新中断	-
27	59	TIM1_TRG_COM	TIM1 触发和换向中断	-
28	60	TIM1_CC	TIM1 捕获/比较中断	-
29	61	TIM2	TIM2 全局中断	-
30	62	TIM3	TIM3 全局中断	-
31	63	TIM4	TIM4 全局中断	-
32	64	I2C1_EVT	I2C1 事件中断	(21)
33	65	I2C1_ERR	I2C1 全局错误中断	-
34	66	I2C2_EVT	I2C2 事件中断	(22)

表 111. 基于 STM32MP13xx 上 Cortex®-A7 的 GIC 中断映射（续）

标号	ID	缩略语	说明	EXTI 事件(1)
35	67	I2C2_ERR	I2C2 全局错误中断	-
36	68	SPI1	SPI1 全局中断	(36)
37	69	SPI2	SPI2 全局中断	(37)
38	70	USART1	USART1 全局中断	(26)
39	71	USART2	USART2 全局中断	(27)
40	72	USART3	USART3 全局中断	(28)
41	73	EXTI10	EXTI 线 10 中断	10
42	74	RTC_TS	RTC 时间戳中断	(17)
43	75	EXTI11	EXTI 线 11 中断	11
44	76	TIM8_BRK	TIM8 暂停中断	-
45	77	TIM8_UP	TIM8 更新中断	-
46	78	TIM8_TRG_COM	TIM8 触发和换向中断	-
47	79	TIM8_CC	TIM8 捕获/比较中断	-
48	80	DMA1_STR7	DMA1 数据流 7 全局中断	-
49	81	FMC	FMC 全局中断	-
50	82	SDMMC1	SDMMC 全局中断	-
51	83	TIM5	TIM5 全局中断	-
52	84	SPI3	SPI3 全局中断	(38)
53	85	UART4	UART4 全局中断	(30)
54	86	UART5	UART5 全局中断	(31)
55	87	TIM6	TIM6 全局中断	-
56	88	TIM7	TIM7 全局中断	-
57	89	DMA2_STR0	DMA2 数据流 0 全局中断	-
58	90	DMA2_STR1	DMA2 数据流 1 全局中断	-
59	91	DMA2_STR2	DMA2 数据流 2 全局中断	-
60	92	DMA2_STR3	DMA2 数据流 3 全局中断	-
61	93	DMA2_STR4	DMA2 数据流 4 全局中断	-
62	94	ETH1	以太网 1 全局中断	-
63	95	ETH1_WKUP	以太网 1 唤醒中断 (PMT)	(68)
64	96	FDCAN_CAL	FDCAN CCU 中断	-
65	97	EXTI6	EXTI 线 6 中断	6
66	98	EXTI7	EXTI 线 7 中断	7
67	99	EXTI8	EXTI 线 8 中断	8

表 111. 基于 STM32MP13xx 上 Cortex®-A7 的 GIC 中断映射（续）

标号	ID	缩略语	说明	EXTI 事件(1)
68	100	EXTI9	EXTI 线 9 中断	9
69	101	DMA2_STR5	DMA2 数据流 5 全局中断	-
70	102	DMA2_STR6	DMA2 数据流 6 全局中断	-
71	103	DMA2_STR7	DMA2 数据流 7 全局中断	-
72	104	USART6	USART6 全局中断	(29)
73	105	I2C3_EVT	I2C3 事件中断	(23)
74	106	I2C3_ERR	I2C3 错误中断	-
75	107	USBH_OHCI	USB 主机 OHCI 中断	（42、43）
76	108	USBH_EHCI	USB 主机 EHCI 中断	
77	109	EXTI12	EXTI 线 12 中断	12
78	110	EXTI13	EXTI 线 13 中断	13
79	111	DCMIPP	DCMIPP 全局中断	-
80	112	CRYP(2)	CRYP 全局中断	-
81	113	HASH	HASH 中断	-
82	114	SAES	安全 AES 全局中断	-
83	115	UART7	UART7 全局中断	(32)
84	116	UART8	UART8 全局中断	(33)
85	117	SPI4	SPI4 全局中断	(39)
86	118	SPI5	SPI5 全局中断	(40)
87	119	SAI1	SAI1 全局中断	-
88	120	LTDC	LTCD 全局中断	-
89	121	LTDC_ER	LTCD 全局错误中断	-
90	122	SAI2	SAI2 全局中断	-
91	123	QUADSPI	QUADSPI 全局中断	-
92	124	LPTIM1	LPTIMER1 全局中断	(47)
93	125	I2C4_EVT	I2C4 事件中断	(24)
94	126	I2C4_ERR	I2C4 错误中断	-
95	127	SPDIFRX	SPDIFRX 全局中断	-
96	128	OTG	USB On-The-Go 全局中断	(44)
97	129	ETH2	以太网 2 全局中断	-
98	130	ETH2_WKUP	以太网 2 唤醒中断 (PMT)	(70)
99	131	DMAMUX1_OVR_REQ	DMAMUX1 溢出中断	-
100	132	DMAMUX2_OVR_REQ	DMAMUX2 溢出中断	-

表 111. 基于 STM32MP13xx 上 Cortex®-A7 的 GIC 中断映射（续）

标号	ID	缩略语	说明	EXTI 事件(1)
101	133	TIM15	TIM15 全局中断	-
102	134	TIM16	TIM16 全局中断	-
103	135	TIM17	TIM17 全局中断	-
104	136	TIM12	TIM12 全局中断	-
105	137	PKA	PKA 中断	-
106	138	EXTI14	EXTI 线 14 中断	14
107	139	MDMA	MDMA 全局中断	-
108	140	SDMMC2	SDMMC2 全局中断	-
109	141	EXTI15	EXTI 线 15 中断	15
110	142	MDMA_SEC_IT	MDMA 全局安全中断	-
111	143	TIM13	TIM13 全局中断	-
112	144	TIM14	TIM14 全局中断	-
113	145	RNG	RNG 中断	-
114	146	I2C5_EVT	I2C5 事件中断	(25)
115	147	I2C5_ERR	I2C5 错误中断	-
116	148	LPTIM2	LPTIMER2 全局中断	(48)
117	149	LPTIM3	LPTIMER3 全局中断	(50)
118	150	LPTIM4	LPTIMER4 全局中断	(52)
119	151	LPTIM5	LPTIMER5 全局中断	(53)
120	152	ETH1_LPI	ETH1 LPI 中断	(69)
121	153	ETH2_LPI	ETH2 LPI 中断	(71)
122	154	RCC_WAKEUP	RCC MPU 唤醒中断	-
123	155	RCC_SEC	RCC 安全全局中断	-
124	156	DTS	数字温度传感器中断	(72)
125	157	MPU_WAKEUP_PIN	由 MPU 使能的全部 6 个唤醒的中断	（55、56、57、58、59、60）
126	158	IWDG1_IT	IWDG1 提前唤醒	(45)
127	159	IWDG2_IT	IWDG2 提前唤醒	(46)
128	160	TAMP_S	TAMP 防篡改安全中断	(18)
129	161	RTC_WKUP_ALARM_S	RTC 唤醒定时器和闹钟（A 和 B）安全中断	(19)
130	162	RTC_TS_S	RTC 时间戳安全中断	(17)
131	163	LTDC_SEC	LTDC 安全全局中断	-

表 111. 基于 STM32MP13xx 上 Cortex®-A7 的 GIC 中断映射（续）

标号	ID	缩略语	说明	EXTI 事件(1)
132	164	LTDC_SEC_ER	LTDC 安全全局错误中断	-
133	165	PMUIRQ0	Cortex®-A7 内核性能监控中断	-
134	166	COMMRX0	Cortex®-A7 内核调试通信接收通道中断	-
135	167	COMMTX0	Cortex®-A7 内核调试通信发送通道中断	-
136	168	AXIERRIRQ	异步 AXI 中止中断	-
137	169	DDRPERFM	DDR 性能监控中断	-
138	170	nCTIIRQ0	Cortex®-A7 内核 CTI 中断	-
139	171	DDRMCE	DDRMCE 中断	-
140	172	DFSDM_FLT1	DFSDM 滤波器 1 中断	-
141	173	DFSDM_FLT2	DFSDM 滤波器 2 中断	-
142	174	DMA3_STR0	DMA3 数据流 0 全局中断	-
143	175	DMA3_STR1	DMA3 数据流 1 全局中断	-
144	176	DMA3_STR2	DMA3 数据流 2 全局中断	-
145	177	DMA3_STR3	DMA3 数据流 3 全局中断	-
146	178	DMA3_STR4	DMA3 数据流 4 全局中断	-
147	179	DMA3_STR5	DMA3 数据流 5 全局中断	-
148	180	DMA3_STR6	DMA3 数据流 6 全局中断	-
149	181	DMA3_STR7	DMA3 数据流 7 全局中断	-
150 到 159	182 到 191	-	保留	-

1. - EXTI 事件编号周围没有括号表示 EXTI 输出已连接到 GIC。EXTI 配置可能会影响中断线状态（即 EXTI 可能会屏蔽中断）。
   - EXTI 事件编号周围有括号 () 表示 EXTI 输出未连接到 GIC。EXTI 配置无法影响中断线状态。
   有关详细信息，请参见*第 20 章：扩展中断和事件控制器 (EXTI)*。
2. 请检查设备数据表以确认可用性。

18.2 EXTI 事件

表 112. STM32MP13xx EXTI 事件

事件输入	信号源	事件类型	连接至 rxev	TrustZone
0	EXTI[0]	可配置	是	是
1	EXTI[1]	可配置	是	是
2	EXTI[2]	可配置	是	是
3	EXTI[3]	可配置	是	是

表 112. STM32MP13xx EXTI 事件（续）

事件输入	信号源	事件类型	连接至 rxev	TrustZone
4	EXTI[4]	可配置	是	是
5	EXTI[5]	可配置	是	是
6	EXTI[6]	可配置	是	是
7	EXTI[7]	可配置	是	是
8	EXTI[8]	可配置	是	是
9	EXTI[9]	可配置	是	是
10	EXTI[10]	可配置	是	是
11	EXTI[11]	可配置	是	是
12	EXTI[12]	可配置	是	是
13	EXTI[13]	可配置	是	是
14	EXTI[14]	可配置	是	是
15	EXTI[15]	可配置	是	是
16	PVD 和 AVD	可配置	否	否
17	RTC 时间戳唤醒（OR 运算非安全和安全）	直接	是	是
18	TAMP 防篡改唤醒（OR 运算非安全和安全）	直接	是	是
19	RTC 唤醒定时器和闹钟（A 和 B）唤醒（OR 运算非安全和安全）	直接	是	是
20	保留	-	-	-
21	I2C1 唤醒	直接	否	否
22	I2C2 唤醒	直接	否	否
23	I2C3 唤醒	直接	否	是
24	I2C4 唤醒	直接	否	是
25	I2C5 唤醒	直接	否	是
26	USART1 唤醒	直接	否	是
27	USART2 唤醒	直接	否	是
28	USART3 唤醒	直接	否	否
29	USART6 唤醒	直接	否	否
30	UART4 唤醒	直接	否	否
31	UART5 唤醒	直接	否	否
32	UART7 唤醒	直接	否	否
33	UART8 唤醒	直接	否	否
34	保留	-	-	-
35	保留	-	-	-
36	SPI1 唤醒	直接	否	否

表 112. STM32MP13xx EXTI 事件（续）

事件输入	信号源	事件类型	连接至 rxev	TrustZone
37	SPI2 唤醒	直接	否	否
38	SPI3 唤醒	直接	否	否
39	SPI4 唤醒	直接	否	是
40	SPI5 唤醒	直接	否	是
41	保留	-	-	-
42	USBH 端口 1 唤醒	直接	否	是
43	USBH 端口 2 唤醒	直接	否	是
44	OTG 唤醒	直接	否	是
45	IWDG1 提前唤醒	直接	否	是
46	IWDG2 提前唤醒	直接	否	否
47	LPTIM1 唤醒	直接	否	否
48	LPTIM2 唤醒	直接	否	是
49	保留	-	-	-
50	LPTIM3 唤醒	直接	否	是
51	保留	-	-	-
52	LPTIM4 唤醒	直接	否	否
53	LPTIM5 唤醒	直接	否	否
54	保留	-	-	-
55	WKUP1 唤醒	直接	否	是
56	WKUP2 唤醒	直接	否	是
57	WKUP3 唤醒	直接	否	是
58	WKUP4 唤醒	直接	否	是
59	WKUP5 唤醒	直接	否	是
60	WKUP6 唤醒	直接	否	是
61	保留	-	-	-
62	保留	-	-	-
63	保留	-	-	-
64	保留	-	-	-
65	保留	-	-	-
66	保留	-	-	-
67	保留	-	-	-
68	ETH1 pmt_intr_o 唤醒	直接	否	是
69	ETH1 lpi_intr_o 唤醒	直接	否	是

表 112. STM32MP13xx EXTI 事件（续）

事件输入	信号源	事件类型	连接至 rxev	TrustZone
70	ETH2 pmt_intr_o 唤醒	直接	否	是
71	ETH2 lpi_intr_o 唤醒	直接	否	是
72	DTS 唤醒	直接	否	否
73	保留	-	-	-
74	保留	-	-	-
75	CDBGPWRUPREQ 事件	直接	否	否

19 全局中断控制器 (GIC)

19.1 简介

以下 Arm® 规范提供了关于 GIC 的其他详细信息：

- Arm® 通用中断控制器 - 架构规范 v2.0
- Cortex®-A7 MPCore 技术参考手册版本：r0p5

19.2 GIC 主要特性

集成 GIC 可从大量中断源进行整理和仲裁操作。

它可提供以下功能：

- 中断屏蔽
- 中断优先级设置
- 将中断分配至目标处理器
- 跟踪中断的状态
- 通过软件生成中断
- 支持安全扩展
- 支持虚拟化扩展

19.3 GIC 功能描述

GIC 是位于 Cortex-A7 MPCore 处理器中的单一功能单元。GIC 包含了一个共享的分配器模块和多个 CPU 接口。多处理器设备中的每个处理器都有：

- 一个 CPU 接口
- 一个虚拟接口控制
- 一个虚拟 CPU 接口

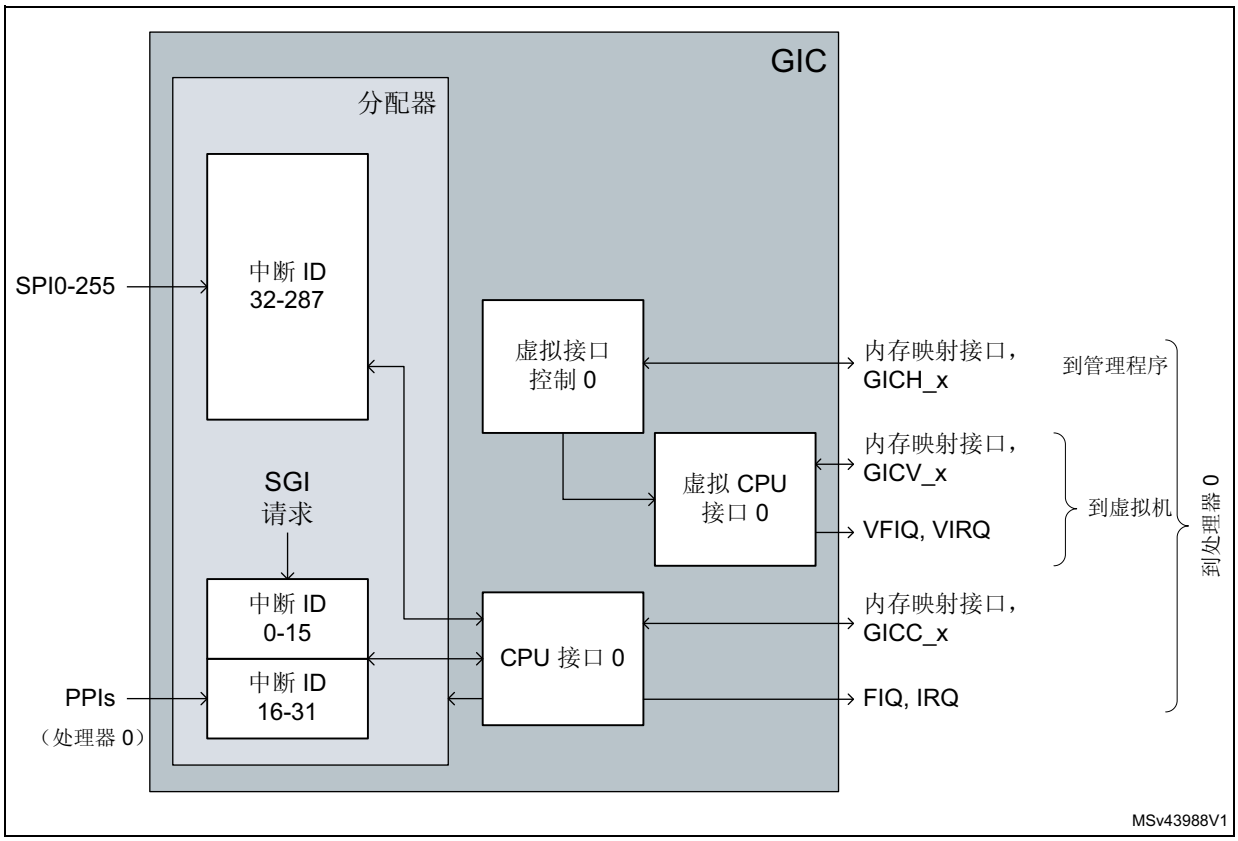
GIC 寄存器为内存映射，且该基址在集群中每个处理器的配置基址 (CBAR) 中均可见。

用于这些寄存器的内存区域必须在转换表中标记为设备内存或强顺序内存

标记为普通内存的内存区域无法访问任何 GIC 寄存器，而是将根据需要访问缓存或外部内存。

图 120 显示了 GIC 框图。

图 120. GIC 框图



19.4 中断源

所有中断源都由唯一 ID 进行标识。

Cortex-A7 MPCore 处理器具有以下中断源：

19.4.1 软件生成中断 (SGI)

SGI 需通过向软件生成中断寄存器 (GICD_SGIR) 进行写入操作来生成。针对每个处理器接口，最多可以生成 ID0-ID15 共 16 个 SGI。SGI 具有边沿触发属性。中断的软件触发相当于外设输入上中断信号的边沿转换。

Arm 建议将 SGI ID0-ID7 用于非安全中断，并将 SGI ID8-ID15 用于安全中断。

19.4.2 私有外设中断 (PPI)

PPI 是由外设产生的中断信号，专门与单个处理器相关联。每个 CPU 接口都具有以下七个 PPI：

传统 nFIQ 信号 (PPI0) - 未使用

当 GIC 中断旁路生效时（例如复位后），外部 nFIQ 信号将绕过中断分配器逻辑，并直接将中断请求发送到相应的处理器。

当处理器使用 GIC 而非外部 nFIQ 信号时，一旦使能其自己的 CPU 接口，则系统会将 nFIQ 信号视作其他中断线，并将使用 ID28。该中断为有效低电平敏感。

安全物理定时器事件 (PPI1)

此事件是由安全物理定时器生成的事件，将使用 ID29。该中断为电平敏感。

非安全物理定时器事件 (PPI2)

此事件是由非安全物理定时器生成的事件，将使用 ID30。该中断为电平敏感。

传统 nIRQ 信号 (PPI3) - 未使用

当 GIC 中断旁路生效时（例如复位后），外部 nIRQ 信号将绕过中断分配器逻辑，并直接将中断请求发送到相应的处理器。

当处理器使用 GIC 而非外部 nIRQ 信号时，一旦使能其自己的 CPU 接口，则系统会将 nIRQ 信号视作其他中断线，并将使用 ID31。该中断为有效低电平敏感。

虚拟定时器事件 (PPI4)

此事件是由虚拟定时器生成的事件，将使用 ID27。该中断为电平敏感。

管理程序定时器事件 (PPI5)

此事件是由物理定时器在管理程序模式下生成的事件，将使用 ID26。该中断为电平敏感。

虚拟维护中断 (PPI6)

Arm 通用中断控制器架构规范中的虚拟化扩展支持，允许在多种情况下生成维护中断。虚拟维护中断将使用 ID25。该中断为电平敏感。

19.4.3 共享外设中断 (SPI)

SPI 由相关中断输入线上生成的事件触发。该 GIC 支持多达 160 个对应于外部 IRQS 信号的 SPI。

SPI 将从 ID32 开始。用户可将 SPI 配置为边沿触发或有效高电平敏感。

19.4.4 中断优先级格式

Cortex-A7 MPCore 处理器可实现一个 5 位版本的中断优先级字段，用于 32 个中断优先级。

19.5 GIC 分配器 (GICD)

GIC 分配器 (GICD) 可集中所有中断源，确定每个中断的优先级，并针对每个 CPU 接口将优先级最高的中断转发至接口，以便进行优先级屏蔽和抢占处理。

GICD 提供了可实现以下功能的编程接口：

- 全局使能将中断转发至 CPU 接口
- 使能或禁止每个中断
- 设置每个中断的优先级
- 设置每个中断的目标处理器列表
- 将每个外设中断设置为电平敏感或边沿触发
- 将每个中断设置为组 0 或组 1
- 向一个或多个目标处理器发送 SGI
- 显示每个中断的状态
- 提供用于设置或清除外设中断挂起状态的软件机制。

GICD_IPRIORITYRx、GICD_ITARGETSRx、GICD_CPENDSGIRx 和 GICD_SPENDSGIRx 寄存器支持字节访问和半字访问。

所有其他寄存器均支持字访问。

19.5.1 GICD 控制寄存器 (GICD_CTLR)

GICD control register

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENABLEGRP1	ENABLEGRP0
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **ENABLEGRP1**：使能组 1 中断 (enable group 1 interrupts)
全局使能将挂起的组 1 中断从 GICD 转发至 CPU 接口

位 0 **ENABLEGRP0**：使能组 0 中断 (enable group 0 interrupts)
全局使能将挂起的组 0 中断从 GICD 转发至 CPU 接口

19.5.2 GICD 控制非安全访问寄存器 (GICD_CTLRNS)

GICD control non-secure access register

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENABLE
															rw

位 31:1 保留，必须保持复位值。

位 0 **ENABLE**：
全局使能将挂起的组 1 中断从 GICD 转发至 CPU 接口

19.5.3 GICD 中断控制器类型寄存器 (GICD_TYPER)

GICD interrupt controller type register

偏移地址：0x004

复位值：0x0000 FC05

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LSPI[4:0]					SECURITYEXTN	Res.	Res.	CPUNUMBER[2:0]			ITLINESNUMBER[4:0]				
r	r	r	r	r	r			r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:11 **LSPI[4:0]**：可锁定共享外设中断 (lockable shared peripheral interrupt)
返回中断控制器包含的 LSPI 数（0b1111 = 31 个 LSPI，此为 ID 32-62 的中断）。

位 10 **SECURITYEXTN**：安全扩展 (security extension)
指示 GIC 是否实现了安全扩展。此位始终返回值 1，表明已实现安全扩展。

位 9:8 保留，必须保持复位值。

位 7:5 **CPUNUMBER[2:0]**：处理器接口数 (number of processors interfaces)
指示 GIC 中已实现的处理器接口数（0b000 = 1 个处理器）

位 4:0 **ITLINESNUMBER[4:0]**：中断线数 (number of interrupt lines)
指示 GIC 支持的中断数（0b00101 = 最多 192 个中断，160 个外部中断）

19.5.4 GICD 实现器标识寄存器 (GICD_IIDR)

GICD implementer identification register

偏移地址：0x008

复位值：0x0100 143B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRODUCTID[7:0]								Res.	Res.	Res.	Res.	VARIANT[3:0]			
r	r	r	r	r	r	r	r					r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION[3:0]				IMPLEMENTER[11:0]											
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 **PRODUCTID[7:0]**：GIC 的产品 ID (product ID of the GIC)

位 23:20 保留，必须保持复位值。

位 19:16 **VARIANT[3:0]**：表示 GIC 的主版本号

位 15:12 **REVISION[3:0]**：表示 GIC 的次版本号

位 11:0 **IMPLEMENTER[11:0]**：GIC 实现器（0x43B Arm 实现）

19.5.5 GICD 中断组寄存器 x (GICD_IGROUPRx)

GICD interrupt group register x

偏移地址：0x080 + 0x4 * x，（x = 0 到 5）

复位值：0x0000 0000

用于中断 ID = x * 32 到 ID = x * 32 + 31

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IGROUPRx[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IGROUPRx[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IGROUPRx[31:0]**：寄存器 x 中断组 (Register x group of interrupts)

19.5.6 GICD 中断设置使能寄存器 (GICD_ISENABLER0)

GICD interrupt set-enable register

偏移地址：0x100

复位值：0b0000 000X XXXX XXXX 1111 1111 1111 1111

用于中断 ID = 0 到 ID = 31

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ISENABLER0[31:16]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISENABLER0[15:0]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:0 **ISENABLER0[31:0]**：中断设置使能 0 (interrupt set-enable 0)

19.5.7 GICD 中断设置使能寄存器 x (GICD_ISENABLERx)

GICD interrupt set-enable register x

偏移地址：0x104 + 0x4 * (x - 1)（x = 1 到 5）

复位值：0x0000 0000

用于中断 ID = x * 32 到 ID = x * 32 + 31

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ISENABLERx[31:16]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISENABLERx[15:0]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:0 **ISENABLERx[31:0]**：中断设置使能 x (interrupt set-enable x)

19.5.8 GICD 中断清除使能寄存器 (GICD_ICENABLER0)

GICD interrupt clear-enable register

偏移地址：0x180

复位值：0b0000 000X XXXX XXXX 1111 1111 1111 1111

用于中断 ID = 0 到 ID = 31

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ICENABLER0[31:16]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICENABLER0[15:0]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:0 **ICENABLER0[31:0]**：中断清除使能 0 (interrupt clear-enable 0)

19.5.9 GICD 中断清除使能寄存器 x (GICD_ICENABLERx)

GICD interrupt clear-enable register x

偏移地址：0x184 + 0x4 * (x - 1)（x = 1 到 5）

复位值：0x0000 0000

用于中断 ID = x * 32 到 ID = x * 32 + 31

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ICENABLERx[31:16]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICENABLERx[15:0]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:0 **ICENABLERx[31:0]**：中断清除使能 x (interrupt clear-enable x)

19.5.10 GICD 中断设置挂起寄存器 x (GICD_ISPENDRx)

GICD interrupt set-pending register x

偏移地址：0x200 + 0x4 * x，（x = 0 到 5）

复位值：0x0000 0000

用于中断 ID = x * 32 到 ID = x * 32 + 31

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ISPENDRx[31:16]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISPENDRx[15:0]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:0 **ISPENDRx[31:0]**：中断设置挂起 x (interrupt set-pending x)

19.5.11 GICD 中断清除挂起寄存器 x (GICD_ICPENDRx)

GICD interrupt clear-pending register x

偏移地址：0x280 + 0x4 * x，（x = 0 到 5）

复位值：0x0000 0000

用于中断 ID = x * 32 到 ID = x * 32 + 31

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ICPENDRx[31:16]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICPENDRx[15:0]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:0 **ICPENDRx[31:0]**：中断清除挂起 x (interrupt clear-pending x)

19.5.12 GICD 中断设置有效寄存器 x (GICD_ISACTIVERx)

GICD interrupt set-active register x

偏移地址：0x300 + 0x4 * x，（x = 0 到 5）

复位值：0x0000 0000

用于中断 ID = x * 32 到 ID = x * 32 + 31

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ISACTIVERx[31:16]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISACTIVERx[15:0]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:0 **ISACTIVERx[31:0]**：中断设置有效 x (interrupt set-active x)

19.5.13 GICD 中断清除有效寄存器 x (GICD_ICACTIVERx)

GICD interrupt clear-active register x

偏移地址：0x380 + 0x4 * x，（x = 0 到 5）

复位值：0x0000 0000

用于中断 ID = x * 32 到 ID = x * 32 + 31

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ICACTIVERx[31:16]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICACTIVERx[15:0]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:0 **ICACTIVERx[31:0]**：中断清除有效 x (interrupt clear-active x)

19.5.14 GICD 中断优先级寄存器 x (GICD_IPRIORITYRx)

GICD interrupt priority register x

偏移地址：0x400 + 0x4 * x，（x = 0 到 47）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRIORITY3[4:0]					Res.	Res.	Res.	PRIORITY2[4:0]					Res.	Res.	Res.
rw	rw	rw	rw	rw				rw	rw	rw	rw	rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIORITY1[4:0]					Res.	Res.	Res.	PRIORITY0[4:0]					Res.	Res.	Res.
rw	rw	rw	rw	rw				rw	rw	rw	rw	rw			

位 31:27 **PRIORITY3[4:0]**：中断 ID 的优先级 = x * 4 + 3

位 26:24 保留，必须保持复位值。

位 23:19 **PRIORITY2[4:0]**：中断 ID 的优先级 = x * 4 + 2

位 18:16 保留，必须保持复位值。

位 15:11 **PRIORITY1[4:0]**：中断 ID 的优先级 = x * 4 + 1

位 10:8 保留，必须保持复位值。

位 7:3 **PRIORITY0[4:0]**：中断 ID 的优先级 = x * 4

位 2:0 保留，必须保持复位值。

19.5.15 GICD 中断处理器目标寄存器 x (GICD_ITARGETSRx)

GICD interrupt processor target register x

偏移地址：0x800 + 0x4 * x，（x = 0 到 7）

复位值：0xXXXX XXXX

对于现有的 SGI 和 PPI，读取 CPU 目标字段后，系统会返回执行读取操作的处理器的编号。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	CPU_TARGETS3[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	CPU_TARGETS2[1:0]	
						r	r							r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CPU_TARGETS1[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	CPU_TARGETS0[1:0]	
						r	r							r	r

位 31:26 保留，必须保持复位值。

位 25:24 **CPU_TARGETS3[1:0]**：中断 ID 的 CPU 目标 = x * 4 + 3

位 23:18 保留，必须保持复位值。

位 17:16 **CPU_TARGETS2[1:0]**：中断 ID 的 CPU 目标 = x * 4 + 2

位 15:10 保留，必须保持复位值。

位 9:8 **CPU_TARGETS1[1:0]**：中断 ID 的 CPU 目标 = x * 4 + 1

位 7:2 保留，必须保持复位值。

位 1:0 **CPU_TARGETS0[1:0]**：中断 ID 的 CPU 目标 = x * 4

19.5.16 GICD 中断处理器目标寄存器 x (GICD_ITARGETSRx)

GICD interrupt processor target register x

偏移地址：0x820 + 0x4 * (x-8)，（x = 8 到 47）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	CPU_TARGETS3[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	CPU_TARGETS2[1:0]	
						rw	rw							rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CPU_TARGETS1[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	CPU_TARGETS0[1:0]	
						rw	rw							rw	rw

位 31:26 保留，必须保持复位值。

位 25:24 **CPU_TARGETS3[1:0]**：中断 ID 的 CPU 目标 = x * 4 + 3

位 23:18 保留，必须保持复位值。

位 17:16 **CPU_TARGETS2[1:0]**：中断 ID 的 CPU 目标 = x * 4 + 2

位 15:10 保留，必须保持复位值。

位 9:8 **CPU_TARGETS1[1:0]**：中断 ID 的 CPU 目标 = x * 4 + 1

位 7:2 保留，必须保持复位值。

位 1:0 **CPU_TARGETS0[1:0]**：中断 ID 的 CPU 目标 = x * 4

19.5.17 GICD 中断配置寄存器 (GICD_ICFGR0)

GICD interrupt configuration register

偏移地址：0xC00

复位值：0xAAAA AAAA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT_CONFIG15[1:0]		INT_CONFIG14[1:0]		INT_CONFIG13[1:0]		INT_CONFIG12[1:0]		INT_CONFIG11[1:0]		INT_CONFIG10[1:0]		INT_CONFIG9[1:0]		INT_CONFIG8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_CONFIG7[1:0]		INT_CONFIG6[1:0]		INT_CONFIG5[1:0]		INT_CONFIG4[1:0]		INT_CONFIG3[1:0]		INT_CONFIG2[1:0]		INT_CONFIG1[1:0]		INT_CONFIG0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 **INT_CONFIG15[1:0]**：中断 ID 的中断配置 = 15

位 29:28 **INT_CONFIG14[1:0]**：中断 ID 的中断配置 = 14

位 27:26 **INT_CONFIG13[1:0]**：中断 ID 的中断配置 = 13

位 25:24 **INT_CONFIG12[1:0]**：中断 ID 的中断配置 = 12

位 23:22 **INT_CONFIG11[1:0]**：中断 ID 的中断配置 = 11

位 21:20 **INT_CONFIG10[1:0]**：中断 ID 的中断配置 = 10

位 19:18 **INT_CONFIG9[1:0]**：中断 ID 的中断配置 = 9

位 17:16 **INT_CONFIG8[1:0]**：中断 ID 的中断配置 = 8

位 15:14 **INT_CONFIG7[1:0]**：中断 ID 的中断配置 = 7

位 13:12 **INT_CONFIG6[1:0]**：中断 ID 的中断配置 = 6

位 11:10 **INT_CONFIG5[1:0]**：中断 ID 的中断配置 = 5

位 9:8 **INT_CONFIG4[1:0]**：中断 ID 的中断配置 = 4

位 7:6 **INT_CONFIG3[1:0]**：中断 ID 的中断配置 = 3

位 5:4 **INT_CONFIG2[1:0]**：中断 ID 的中断配置 = 2

位 3:2 **INT_CONFIG1[1:0]**：中断 ID 的中断配置 = 1

位 1:0 **INT_CONFIG0[1:0]**：中断 ID 的中断配置 = 0

19.5.18 GICD 中断配置寄存器 (GICD_ICFGR1)

GICD interrupt configuration register

偏移地址：0xC04

复位值：0b0101 0101 0101 01XX XXXX XXXX XXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT_CONFIG15[1:0]		INT_CONFIG14[1:0]		INT_CONFIG13[1:0]		INT_CONFIG12[1:0]		INT_CONFIG11[1:0]		INT_CONFIG10[1:0]		INT_CONFIG9[1:0]		INT_CONFIG8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_CONFIG7[1:0]		INT_CONFIG6[1:0]		INT_CONFIG5[1:0]		INT_CONFIG4[1:0]		INT_CONFIG3[1:0]		INT_CONFIG2[1:0]		INT_CONFIG1[1:0]		INT_CONFIG0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 **INT_CONFIG15[1:0]**：中断 ID 的中断配置 = 31

位 29:28 **INT_CONFIG14[1:0]**：中断 ID 的中断配置 = 30

位 27:26 **INT_CONFIG13[1:0]**：中断 ID 的中断配置 = 29

位 25:24 **INT_CONFIG12[1:0]**：中断 ID 的中断配置 = 28

位 23:22 **INT_CONFIG11[1:0]**：中断 ID 的中断配置 = 27

位 21:20 **INT_CONFIG10[1:0]**：中断 ID 的中断配置 = 26

位 19:18 **INT_CONFIG9[1:0]**：中断 ID 的中断配置 = 25

位 17:16 **INT_CONFIG8[1:0]**：中断 ID 的中断配置 = 24

位 15:14 **INT_CONFIG7[1:0]**：中断 ID 的中断配置 = 23

位 13:12 **INT_CONFIG6[1:0]**：中断 ID 的中断配置 = 22

位 11:10 **INT_CONFIG5[1:0]**：中断 ID 的中断配置 = 21

位 9:8 **INT_CONFIG4[1:0]**：中断 ID 的中断配置 = 20

位 7:6 **INT_CONFIG3[1:0]**：中断 ID 的中断配置 = 19

位 5:4 **INT_CONFIG2[1:0]**：中断 ID 的中断配置 = 18

位 3:2 **INT_CONFIG1[1:0]**：中断 ID 的中断配置 = 17

位 1:0 **INT_CONFIG0[1:0]**：中断 ID 的中断配置 = 16

19.5.19 GICD 中断配置寄存器 x (GICD_ICFGRx)

GICD interrupt configuration register x

偏移地址：0xC08 + 0x4 * (x-2)，（x = 2 到 11）

复位值：0x5555 5555

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT_CONFIG15[1:0]		INT_CONFIG14[1:0]		INT_CONFIG13[1:0]		INT_CONFIG12[1:0]		INT_CONFIG11[1:0]		INT_CONFIG10[1:0]		INT_CONFIG9[1:0]		INT_CONFIG8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_CONFIG7[1:0]		INT_CONFIG6[1:0]		INT_CONFIG5[1:0]		INT_CONFIG4[1:0]		INT_CONFIG3[1:0]		INT_CONFIG2[1:0]		INT_CONFIG1[1:0]		INT_CONFIG0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 **INT_CONFIG15[1:0]**：中断 ID 的中断配置 = x * 16 + 15

位 29:28 **INT_CONFIG14[1:0]**：中断 ID 的中断配置 = x * 16 + 14

位 27:26 **INT_CONFIG13[1:0]**：中断 ID 的中断配置 = x * 16 + 13

位 25:24 **INT_CONFIG12[1:0]**：中断 ID 的中断配置 = x * 16 + 12

位 23:22 **INT_CONFIG11[1:0]**：中断 ID 的中断配置 = x * 16 + 11

位 21:20 **INT_CONFIG10[1:0]**：中断 ID 的中断配置 = x * 16 + 10

位 19:18 **INT_CONFIG9[1:0]**：中断 ID 的中断配置 = x * 16 + 9

位 17:16 **INT_CONFIG8[1:0]**：中断 ID 的中断配置 = x * 16 + 8

位 15:14 **INT_CONFIG7[1:0]**：中断 ID 的中断配置 = x * 16 + 7

位 13:12 **INT_CONFIG6[1:0]**：中断 ID 的中断配置 = x * 16 + 6

位 11:10 **INT_CONFIG5[1:0]**：中断 ID 的中断配置 = x * 16 + 5

位 9:8 **INT_CONFIG4[1:0]**：中断 ID 的中断配置 = x * 16 + 4

位 7:6 **INT_CONFIG3[1:0]**：中断 ID 的中断配置 = x * 16 + 3

位 5:4 **INT_CONFIG2[1:0]**：中断 ID 的中断配置 = x * 16 + 2

位 3:2 **INT_CONFIG1[1:0]**：中断 ID 的中断配置 = x * 16 + 1

位 1:0 **INT_CONFIG0[1:0]**：中断 ID 的中断配置 = x * 16

19.5.20 GICD 私有外设中断状态寄存器 (GICD_PPISR)

GICD private peripheral interrupt status register

偏移地址：0xD00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPI3	PPI2	PPI1	PPI0	PPI4	PPI5	PPI6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r	r	r									

位 31:16 保留，必须保持复位值。

位 15 **PPI3**：nIRQ（未使用）

位 14 **PPI2**：安全物理定时器事件

位 13 **PPI1**：安全物理定时器事件

位 12 **PPI0**：nFIQ（未使用）

位 11 **PPI4**：虚拟定时器事件

位 10 **PPI5**：管理程序定时器事件

位 9 **PPI6**：虚拟维护中断

位 8:0 保留，必须保持复位值。

19.5.21 GICD 共享外设中断寄存器 x (GICD_SPISRx)

GICD shared peripheral interrupt register x

偏移地址：0xD04 + 0x4 * x，（x = 0 到 4）

复位值：0x0000 0000

用于中断 ID = SPI 数 + 32，从 SPI [x * 32 + 31] 到 SPI [x * 32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPISRx[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPISRx[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SPISRx[31:0]**：寄存器 x 共享外设中断 (Register x shared peripheral interrupt)

19.5.22 GICD 软件生成中断寄存器 (GICD_SGIR)

GICD software generated interrupt register

偏移地址：0xF00

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	TARGETLISTFILTER[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	CPUTARGETLIST[1:0]	
						w	w							w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSATT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SGIINTID[3:0]			
w												w	w	w	w

位 31:26 保留，必须保持复位值。

位 25:24 **TARGETLISTFILTER[1:0]**：目标列表过滤器 (target list filter)

确定分配器必须如何处理所请求的 SGI

0：将中断转发到 CPUTARGETLIST 字段中指定的 CPU 接口

1：将中断转发到除了请求中断的处理器以外的所有 CPU 接口

2：只将中断转发到请求中断的处理器的 CPU 接口

3：保留，必须保持复位值。

位 23:18 保留，必须保持复位值。

位 17:16 **CPUTARGETLIST[1:0]**：CPU 目标列表 (CPU target list)

当 TARGETLISTFILTER = 0b00 时，用于定义 GICD 必须将中断转发到哪个 CPU 接口。CPUTARGETLIST[1:0] 的每个位都指向相应的 CPU 接口。如果在 TARGETLISTFILTER 为 0b00 时此字段为 0，GICD 不会将中断转发到任何 CPU 接口。

位 15 **NSATT**：非安全属性 (non-secure attribute)

此位将指定 SGI 所需的安全值：

0 - 仅当指定的 CPU 接口上的 SGI 配置为组 0 时，才会将 SGIINTID 字段中指定的 SGI 转发到该接口。

1 - 仅当指定的 CPU 接口上的 SGI 配置为组 1 时，才会将 SGIINTID 字段中指定的 SGI 转发到该接口。

此字段仅能通过安全访问进行写入。只有当指定的 SGI 编程为组 1 时，对 GICD_SGIR 进行任意非安全写入才会生成一个 SGI。

位 14:4 保留，必须保持复位值。

位 3:0 **SGIINTID[3:0]**：SGI 中断 ID (SGI interrupt ID)

此字段的值为要转发到特定 CPU 接口的 SGI 的中断 ID，其范围是 0-15（例如：值 0b0011 将指定中断 ID 3）。

19.5.23 GICD SGI 清除挂起寄存器 x (GICD_CPENDSGIRx)

GICD SGI clear-pending register x

偏移地址：0xF10 + 0x4 * x，（x = 0 到 3）

复位值：0x0000 0000

用于 SGI x * 4 到 SGI x * 4 + 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	SGI_CLEAR_PENDING3[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	SGI_CLEAR_PENDING2[1:0]	
						rw	rw							rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	SGI_CLEAR_PENDING1[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	SGI_CLEAR_PENDING0[1:0]	
						rw	rw							rw	rw

位 31:26 保留，必须保持复位值。

位 25:24 **SGI_CLEAR_PENDING3[1:0]**：SGI [x * 4 + 3] 的清除挂起状态

写入 1 将清除相应源处理器的 SGI [x * 4 + 3] 挂起状态，并且不再将执行写入的处理器设为目标处理器。

写入 0 不会起任何作用。

读取一个位将识别读取处理器上的 SGI [x * 4 + 3] 是否已在相应的源处理器中处于挂起状态。

位 23:18 保留，必须保持复位值。

位 17:16 **SGI_CLEAR_PENDING2[1:0]**：SGI [x * 4 + 2] 的清除挂起状态

写入 1 将清除相应源处理器的 SGI [x * 4 + 2] 挂起状态，并且不再将执行写入的处理器设为目标处理器。

写入 0 不会起任何作用。

读取一个位将识别读取处理器上的 SGI [x * 4 + 2] 是否已在相应的源处理器中处于挂起状态。

位 15:10 保留，必须保持复位值。

位 9:8 **SGI_CLEAR_PENDING1[1:0]**：SGI [x * 4 + 1] 的清除挂起状态

写入 1 将清除相应源处理器的 SGI [x * 4 + 1] 挂起状态，并且不再将执行写入的处理器设为目标处理器。

写入 0 不会起任何作用。

读取一个位将识别读取处理器上的 SGI [x * 4 + 1] 是否已在相应的源处理器中处于挂起状态。

位 7:2 保留，必须保持复位值。

位 1:0 **SGI_CLEAR_PENDING0[1:0]**：SGI [x * 4] 的清除挂起状态

写入 1 将清除相应源处理器的 SGI [x * 4] 挂起状态，并且不再将执行写入的处理器设为目标处理器。

写入 0 不会起任何作用。

读取一个位将识别读取处理器上的 SGI [x * 4] 是否已在相应的源处理器中处于挂起状态。

19.5.24 GICD SGI 设置挂起寄存器 x (GICD_SPENDSGIRx)

GICD SGI set-pending register x

偏移地址：0xF20 + 0x4 * x，（x = 0 到 3）

复位值：0x0000 0000

用于 SGI x * 4 到 SGI x * 4 + 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	SGI_SET_PENDING3[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	SGI_SET_PENDING2[1:0]	
						rw	rw							rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	SGI_SET_PENDING1[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	SGI_SET_PENDING0[1:0]	
						rw	rw							rw	rw

位 31:26 保留，必须保持复位值。

位 25:24 **SGI_SET_PENDING3[1:0]**：SGI [x * 4 + 3] 的设置挂起状态

写入 1 将为相应的源处理器设置 SGI [x * 4 + 3] 的挂起状态。

写入 0 不会起任何作用。

读取一个位将识别读取处理器上的 SGI [x * 4 + 3] 是否已在相应的源处理器中处于挂起状态。

位 23:18 保留，必须保持复位值。

位 17:16 **SGI_SET_PENDING2[1:0]**：SGI [x * 4 + 2] 的设置挂起状态

写入 1 将为相应的源处理器设置 SGI [x * 4 + 2] 的挂起状态。

写入 0 不会起任何作用。

读取一个位将识别读取处理器上的 SGI [x * 4 + 2] 是否已在相应的源处理器中处于挂起状态。

位 15:10 保留，必须保持复位值。

位 9:8 **SGI_SET_PENDING1[1:0]**：SGI [x * 4 + 1] 的设置挂起状态
写入 1 将为相应的源处理器设置 SGI [x * 4 + 1] 的挂起状态。
写入 0 不会起任何作用。
读取一个位将识别读取处理器上的 SGI [x * 4 + 1] 是否已在相应的源处理器中处于挂起状态。

位 7:2 保留，必须保持复位值。

位 1:0 **SGI_SET_PENDING0[1:0]**：SGI [x * 4] 的设置挂起状态
写入 1 将为相应的源处理器设置 SGI [x * 4] 的挂起状态。
写入 0 不会起任何作用。
读取一个位将识别读取处理器上的 SGI [x * 4] 是否已在相应的源处理器中处于挂起状态。

19.5.25 GICD 外设 ID4 寄存器 (GICD_PIDR4)

GICD peripheral ID4 register

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIDR4[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIDR4[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **PIDR4[31:0]**：外设 ID4 (peripheral ID4)

19.5.26 GICD 外设 ID5 到 ID7 寄存器 x (GICD_PIDRx)

GICD peripheral ID5 to ID7 register x

偏移地址：0xFD4 + 0x4* (x-5)，（x = 5 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIDRx[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIDRx[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **PIDRx[31:0]**：寄存器 x 外设 ID5 到 ID7

19.5.27 GICD 外设 ID0 寄存器 (GICD_PIDR0)

GICD peripheral ID0 register

偏移地址：0xFE0

复位值：0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIDR0[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIDR0[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **PIDR0[31:0]**：外设 ID0 (peripheral ID0)

19.5.28 GICD 外设 ID1 寄存器 (GICD_PIDR1)

GICD peripheral ID1 register

偏移地址：0xFE4

复位值：0x0000 00B4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIDR1[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIDR1[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **PIDR1[31:0]**：外设 ID1 (peripheral ID1)

19.5.29 GICD 外设 ID2 寄存器 (GICD_PIDR2)

GICD peripheral ID2 register

偏移地址：0xFE8

复位值：0x0000 002B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIDR2[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIDR2[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **PIDR2[31:0]**：外设 ID2 (peripheral ID2)

19.5.30 GICD 外设 ID3 寄存器 (GICD_PIDR3)

GICD peripheral ID3 register

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIDR3[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIDR3[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **PIDR3[31:0]**：外设 ID3 (peripheral ID3)

19.5.31 GICD 组件 ID0 寄存器 (GICD_CIDR0)

GICD component ID0 register

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CIDR0[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CIDR0[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CIDR0[31:0]**：组件 ID0 (component ID0)

19.5.32 GICD 组件 ID1 寄存器 (GICD_CIDR1)

GICD component ID1 register

偏移地址：0xFF4

复位值：0x0000 00F0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CIDR1[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CIDR1[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CIDR1[31:0]**：组件 ID1 (component ID1)

19.5.33 GICD 组件 ID2 寄存器 (GICD_CIDR2)

GICD component ID2 register

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CIDR2[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CIDR2[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CIDR2[31:0]**：组件 ID2 (component ID2)

19.5.34 GICD 组件 ID3 寄存器 (GICD_CIDR3)

GICD component ID3 register

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CIDR3[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CIDR3[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CIDR3[31:0]**：组件 ID3 (component ID3)

19.5.35 GICD 寄存器映射

表 113. GICD 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	GICD_CTLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENABLEGRP1	ENABLEGRP0
	Reset value																															0	0
0x000	GICD_CTLRNS	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ENABLE
	Reset value																																0

表 113. GICD 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x004	GICD_TYPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSPI[4:0]					SECURITYEXTN	Res.	Res.	CPUNUMBER[2:0]			ITLINESNUMBER[4:0]				
	Reset value																	1	1	1	1	1	1			0	0	0	0	0	1	0	1
0x008	GICD_IIDR	PRODUCTID[7:0]								Res.	Res.	Res.	Res.	VARIANT[3:0]				REVISION[3:0]				IMPLEMENTER[11:0]											
	Reset value	0	0	0	0	0	0	0	1					0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	1	1
0x00C - 0x07C	Reserved	Reserved																															
0x080 to 0x094	GICD_IGROUPR0 to GICD_IGROUPR5	IGROUPRx[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x098 - 0x0FC	Reserved	Reserved																															
0x100	GICD_ISENABLER0	ISENABLER0[31:0]																															
	Reset value	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x104 to 0x114	GICD_ISENABLER1 to GICD_ISENABLER5	ISENABLERx[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x118 - 0x17C	Reserved	Reserved																															
0x180	GICD_ICENABLER0	ICENABLER0[31:0]																															
	Reset value	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x184 to 0x194	GICD_ICENABLER1 to GICD_ICENABLER5	ICENABLERx[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x198 - 0x1FC	Reserved	Reserved																															
0x200 to 0x214	GICD_ISPENDR0 to GICD_ISPENDR5	ISPENDRx[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x218 - 0x27C	Reserved	Reserved																															
0x280 to 0x294	GICD_ICPENDR0 to GICD_ICPENDR5	ICPENDRx[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x298 - 0x2FC	Reserved	Reserved																															
0x300 to 0x314	GICD_ISACTIVER0 to GICD_ISACTIVER5	ISACTIVERx[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x318 - 0x37C	Reserved	Reserved																															
0x380 to 0x394	GICD_ICACTIVER0 to GICD_ICACTIVER5	ICACTIVERx[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x398 - 0x3FC	Reserved	Reserved																															

表 113. GICD 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x400 to 0x4BC	GICD_IPRIORITYR0 to GICD_IPRIORITYR47	PRIORITY3[4:0]					Res.	Res.	Res.	PRIORITY2[4:0]					Res.	Res.	Res.	PRIORITY1[4:0]					Res.	Res.	Res.	PRIORITY0[4:0]					Res.	Res.	Res.
	Reset value	0	0	0	0	0				0	0	0	0	0				0	0	0	0	0				0	0	0	0	0			
0x4C0 - 0x7FC	Reserved	Reserved																															
0x800 to 0x81C	GICD_ITARGETSR0 to GICD_ITARGETSR7	Res.	Res.	Res.	Res.	Res.	Res.	CPU_TARGETS3[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	CPU_TARGETS2[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	CPU_TARGETS1[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	CPU_TARGETS0[1:0]	
	Reset value							X	X							X	X							X	X							X	X
0x820 to 0x8BC	GICD_ITARGETSR8 to GICD_ITARGETSR47	Res.	Res.	Res.	Res.	Res.	Res.	CPU_TARGETS3[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	CPU_TARGETS2[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	CPU_TARGETS1[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	CPU_TARGETS0[1:0]	
	Reset value							0	0							0	0							0	0							0	0
0x8C0 - 0xBFC	Reserved	Reserved																															
0xC00	GICD_ICFGR0	INT_CONFIG15[1:0]		INT_CONFIG14[1:0]		INT_CONFIG13[1:0]		INT_CONFIG12[1:0]		INT_CONFIG11[1:0]		INT_CONFIG10[1:0]		INT_CONFIG9[1:0]		INT_CONFIG8[1:0]		INT_CONFIG7[1:0]		INT_CONFIG6[1:0]		INT_CONFIG5[1:0]		INT_CONFIG4[1:0]		INT_CONFIG3[1:0]		INT_CONFIG2[1:0]		INT_CONFIG1[1:0]		INT_CONFIG0[1:0]	
	Reset value	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0xC04	GICD_ICFGR1	INT_CONFIG15[1:0]		INT_CONFIG14[1:0]		INT_CONFIG13[1:0]		INT_CONFIG12[1:0]		INT_CONFIG11[1:0]		INT_CONFIG10[1:0]		INT_CONFIG9[1:0]		INT_CONFIG8[1:0]		INT_CONFIG7[1:0]		INT_CONFIG6[1:0]		INT_CONFIG5[1:0]		INT_CONFIG4[1:0]		INT_CONFIG3[1:0]		INT_CONFIG2[1:0]		INT_CONFIG1[1:0]		INT_CONFIG0[1:0]	
	Reset value	0	1	0	1	0	1	0	1	0	1	0	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0xC08 to 0xC2C	GICD_ICFGR2 to GICD_ICFGR11	INT_CONFIG15[1:0]		INT_CONFIG14[1:0]		INT_CONFIG13[1:0]		INT_CONFIG12[1:0]		INT_CONFIG11[1:0]		INT_CONFIG10[1:0]		INT_CONFIG9[1:0]		INT_CONFIG8[1:0]		INT_CONFIG7[1:0]		INT_CONFIG6[1:0]		INT_CONFIG5[1:0]		INT_CONFIG4[1:0]		INT_CONFIG3[1:0]		INT_CONFIG2[1:0]		INT_CONFIG1[1:0]		INT_CONFIG0[1:0]	
	Reset value	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0xC30 - 0xCFC	Reserved	Reserved																															
0xD00	GICD_PPISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PPI3	PPI2	PPI1	PPI0	PPI4	PPI5	PPI6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																	0	0	0	0	0	0	0									
0xD04 to 0xD14	GICD_SPISR0 to GICD_SPISR4	SPISRx[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 113. GICD 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xD18 - 0xEFC	Reserved	Reserved																															
0xF00	GICD_SGIR	Res.	Res.	Res.	Res.	Res.	Res.	TARGETLISTFILTER[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	CPUTARGETLIST[1:0]		NSATT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SGIINTID[3:0]			
	Reset value							X	X							X	X	X												X	X	X	X
0xF04 - 0xF0C	Reserved	Reserved																															
0xF10 to 0xF1C	GICD_CPENDSGIR0 to GICD_CPENDSGIR3	Res.	Res.	Res.	Res.	Res.	Res.	SGI_CLEAR_PENDING3[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	SGI_CLEAR_PENDING2[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	SGI_CLEAR_PENDING1[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	SGI_CLEAR_PENDING0[1:0]	
	Reset value							0	0							0	0							0	0							0	0
0xF20 to 0xF2C	GICD_SPENDSGIR0 to GICD_SPENDSGIR3	Res.	Res.	Res.	Res.	Res.	Res.	SGI_SET_PENDING3[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	SGI_SET_PENDING2[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	SGI_SET_PENDING1[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	SGI_SET_PENDING0[1:0]	
	Reset value							0	0							0	0							0	0							0	0
0xF30 - 0xFCC	Reserved	Reserved																															
0xFD0	GICD_PIDR4	PIDR4[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0xFD4 to 0xFDC	GICD_PIDR5 to GICD_PIDR7	PIDRx[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xFE0	GICD_PIDR0	PIDR0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
0xFE4	GICD_PIDR1	PIDR1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0
0xFE8	GICD_PIDR2	PIDR2[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1
0xFEC	GICD_PIDR3	PIDR3[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xFF0	GICD_CIDR0	CIDR0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
0xFF4	GICD_CIDR1	CIDR1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
0xFF8	GICD_CIDR2	CIDR2[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
0xFFC	GICD_CIDR3	CIDR3[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1

有关寄存器边界地址的信息，请参见第 147 页的第 2.5 节。

19.6 GIC CPU 接口 (GICC)

每个 CPU 接口均提供了可实现以下功能的编程接口：

- 使能 CPU 接口发送中断请求信号
- 确认中断
- 指示中断处理完成
- 为处理器设置中断优先级屏蔽
- 定义处理器的抢占策略
- 确定处理器的最高优先级挂起中断。

所有寄存器均支持字访问。

19.6.1 GICC 控制寄存器 (GICC_CTLR)

GICC control register

偏移地址：0x000

复位值：0bXXXX XXXX XXXX XXXX XXXX X000 0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	EOIMODENS	EOIMODES	IRQBYPDISGRP1	FIQBYPDISGRP1	IRQBYPDISGRP0	FIQBYPDISGRP0	CBPR	FIQEN	ACKCTL	ENABLEGRP1	ENABLEGRP0
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10 **EOIMODENS**：此寄存器的非安全副本中 EOIMODENS 的别名。

位 9 **EOIMODES**：安全访问的 EOI 模式
控制对 GICC_EOIR 和 GICC_DIR 寄存器的访问行为。此控制仅适用于安全访问。

位 8 **IRQBYPDISGRP1**：此寄存器的非安全副本中 IRQBYPDISGRP1 的别名。

位 7 **FIQBYPDISGRP1**：此寄存器的非安全副本中 FIQBYPDISGRP1 的别名。

位 6 **IRQBYPDISGRP0**：组 0 中断的 IRQ 中断绕过 (IRQ bypass disable for group 0 interrupts)
当禁止 CPU 接口发送 IRQ 信号时，此位将部分控制是否向处理器发出旁路 IRQ 信号。

位 5 **FIQBYPDISGRP0**：组 0 中断的 FIQ 中断绕过 (FIQ bypass disable for group 0 interrupts)
当禁止 CPU 接口发送 FIQ 信号时，此位将部分控制是否向处理器发出旁路 FIQ 信号。

位 4 **CBPR**：BPR 控制 (BPR control)
控制 GICC_BPR 是否为组 0 和组 1 中断提供通用控制。

位 3 **FIQEN**：组 0 中断的 FIQ 使能 (FIQ enable for group 0 interrupts)
控制 CPU 接口是否使用 FIQ 或 IRQ 信号将组 0 中断信号发送至目标处理器。
GIC 始终使用 IRQ 信号来发送组 1 中断信号。

位 2 **ACKCTL**：应答控制 (Acknowledge control)
当最高优先级待处理中断是组 1 中断时，这个位决定了以下两点：
- 读取 GICC_IAR 时，是否确认该中断，或者返回一个虚假的中断 ID。
- 读取 GICC_HPPIR 时，是否返回最高优先级的待处理中断的 ID，或者返回一个虚假的中断 ID。
Arm 不推荐使用此位，并强烈建议使用将此位置 0 的软件模型。

位 1 **ENABLEGRP1**：使能组 1 中断 (Enable group 1 interrupts)
此位支持 CPU 接口向所连接的处理器发送组 1 中断信号。

位 0 **ENABLEGRP0**：使能组 0 中断 (Enable group 0 interrupts)
此位支持 CPU 接口向所连接的处理器发送组 0 中断信号。

19.6.2 GICC 控制非安全访问寄存器 (GICC_CTLRNS)

GICC control non-secure access register

偏移地址：0x000

复位值：0bXXXX XXXX XXXX XXXX XXXX XX0X X00X XXX0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	EOIMODENS	Res.	Res.	IRQBYPDISGRP1	FIQBYPDISGRP1	Res.	Res.	Res.	Res.	ENABLEGRP1
						rw			rw	rw					rw

位 31:10 保留，必须保持复位值。

位 9 **EOIMODENS**：非安全访问的 EOI 模式 (EOI mode for non-secure accesses)
此位将控制对 GICC_EOIR 和 GICC_DIR 寄存器的访问行为。

位 8:7 保留，必须保持复位值。

位 6 **IRQBYPDISGRP1**：组 1 中断的 IRQ 中断的跳过 (IRQ bypass for group 1 interrupts)
当禁止 CPU 接口发送 IRQ 信号时，此位部分控制是否将跳过的 IRQ 信号传递给处理器。

位 5 **FIQBYPDISGRP1**：禁用组 1 中断的 FIQ 跳过 (FIQ bypass disable for group 1 interrupts)
当禁止 CPU 接口发送 FIQ 信号时，此位部分控制是否将跳过的 FIQ 信号传递给处理器。

位 4:1 保留，必须保持复位值。

位 0 **ENABLEGRP1**：使能组 1 中断 (Enable group1 interrupts)
支持 CPU 接口向所连接的处理器发送组 1 中断信号。

19.6.3 GICC 输入优先级屏蔽寄存器 (GICC_PMR)

GICC input priority mask register

偏移地址：0x004

复位值：0bXXXX XXXX XXXX XXXX XXXX XXXX 0000 0XXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIORITY[4:0]					Res.	Res.	Res.
								rw	rw	rw	rw	rw			

位 31:8 保留，必须保持复位值。

位 7:3 **PRIORITY[4:0]**：CPU 接口的优先级屏蔽级别

如果某中断的优先级高于此字段所指示的值，则接口会将中断信号发送给处理器。

位 2:0 保留，必须保持复位值。

19.6.4 GICC 二进制点寄存器 (GICC_BPR)

GICC binary point register

偏移地址：0x008

复位值：0bXXXX XXXX XXXX XXXX XXXX XXXX XXXX X010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BINARY_POINT[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **BINARY_POINT[2:0]**：

该字段的值将控制如何将 8 位中断优先级字段划分为组优先级字段。它用于确定中断抢占优先级和子优先级。最小值为 2。

19.6.5 GICC 二进制点非安全访问寄存器 (GICC_BPRNS)

GICC binary point non-secure access register

偏移地址：0x008

复位值：0bXXXX XXXX XXXX XXXX XXXX XXXX XXXX X011

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BINARY_POINT[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **BINARY_POINT[2:0]**：

该字段的值将控制如何将 8 位中断优先级字段划分为组优先级字段。它用于确定中断抢占优先级和子优先级。最小值为 3。

19.6.6 GICC 中断确认寄存器 (GICC_IAR)

GICC interrupt acknowledge register

偏移地址：0x00C

复位值：0bXXXX XXXX XXXX XXXX XXXX X011 1111 1111

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CPUID	INTERRUPT_ID[9:0]									
					r	r	r	r	r	r	r	r	r	r	r

位 31:11 保留，必须保持复位值。

位 10 **CPUID**：

对于多处理器实现中的 SGI，该字段将标识请求中断的处理器。它将返回发出请求的 CPU 接口的编号。

例如，值 1 表示该请求是通过对 CPU 接口 1 上的 GICD_SGIR 进行写入而生成的。

位 9:0 **INTERRUPT_ID[9:0]**：中断 ID (interrupt ID)

19.6.7 GICC 中断结束寄存器 (GICC_EOIR)

GICC end of interrupt register

偏移地址：0x010

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CPUID	EOIINTID[9:0]									
					w	w	w	w	w	w	w	w	w	w	w

位 31:11 保留，必须保持复位值。

位 10 **CPUID**：
在多处理器实现中，如果写入操作涉及 SGI，则此字段将包含相应 GICC_IAR 访问中的 CPUID 值。

位 9:0 **EOIINTID[9:0]**：
它将包含来自相应 GICC_IAR 访问的中断 ID 值。

19.6.8 GICC 运行优先级寄存器 (GICC_RPR)

GICC running priority register

偏移地址：0x014

复位值：0bXXXX XXXX XXXX XXXX XXXX XXXX 1111 1111

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIORITY[4:0]					Res.	Res.	Res.
								r	r	r	r	r			

位 31:8 保留，必须保持复位值。

位 7:3 **PRIORITY[4:0]**：CPU 接口上的当前运行优先级

位 2:0 保留，必须保持复位值。

19.6.9 GICC 最高优先级挂起中断寄存器 (GICC_HPPIR)

GICC highest priority pending interrupt register

偏移地址：0x018

复位值：0bXXXX XXXX XXXX XXXX XXX0 0011 1111 1111

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CPUID	PENDINTID[9:0]									
					r	r	r	r	r	r	r	r	r	r	r

位 31:11 保留，必须保持复位值。

位 10 **CPUID**：
在多处理器实现中，如果 PENDINTID 字段返回了 SGI 的 ID，则该字段将包含该中断的 CPUID 值。这将标识生成中断的处理器。

位 9:0 **PENDINTID[9:0]**：最高优先级挂起中断的中断 ID (interrupt ID of the highest-priority pending interrupt)

19.6.10 GICC 别名二进制点寄存器 (GICC_ABPR)

GICC aliased binary point register

偏移地址：0x01C

复位值：0bXXXX XXXX XXXX XXXX XXXX XXXX XXXX X011

GICC_ABPR 是非安全 GICC_BPR 的别名。当 GICC_CTLR.CBPR 置 0 时，对此寄存器的安全访问将等同于对 GICC_BPR 的非安全访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BINARY_POINT[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **BINARY_POINT[2:0]**：

该字段的值将控制如何将 8 位中断优先级字段划分为组优先级字段。它用于确定中断抢占优先级和子优先级。最小值为 3。

19.6.11 GICC 别名中断确认寄存器 (GICC_AIAR)

GICC aliased interrupt acknowledge register

偏移地址：0x020

复位值：0bXXXX XXXX XXXX XXXX XXXX X011 1111 1111

GICC_AIAR 是 GICC_IAR 非安全视图的别名。对此寄存器的安全访问将与对 GICC_IAR 的非安全访问相同。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CPUID	INTERRUPT_ID[9:0]									
					r	r	r	r	r	r	r	r	r	r	r

位 31:11 保留，必须保持复位值。

位 10 **CPUID**：

对于多处理器实现中的 SGI，该字段将标识请求中断的处理器。它将返回发出请求的 CPU 接口的编号。

例如，值 1 表示该请求是通过对 CPU 接口 1 上的 GICD_SGIR 进行写入而生成的。

位 9:0 **INTERRUPT_ID[9:0]**：中断 ID (interrupt ID)

19.6.12 GICC 别名中断结束寄存器 (GICC_AEOIR)

GICC aliased end of interrupt register

偏移地址：0x024

复位值：0xXXXX XXXX

GICC_AEOIR 是非安全 GICC_EOIR 的别名。对此寄存器的安全访问与对 GICC_EOIR 的非安全访问相似，不同之处在于前者使用了 GICC_CTLR.EOImodeS 位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CPUID	EOIINTID[9:0]									
					w	w	w	w	w	w	w	w	w	w	w

位 31:11 保留，必须保持复位值。

位 10 **CPUID**：

在多处理器实现中处理 SGI 时，此字段必须包含相应 GICC_AIAR 访问或非安全 GICC_IAR 访问中的 CPUID 值。

位 9:0 **EOIINTID[9:0]**：

它将包含来自相应 GICC_AIAR 访问或非安全 GICC_IAR 访问的中断 ID 值。

19.6.13 GICC 别名最高优先级挂起中断寄存器 (GICC_AHPPIR)

GICC aliased highest priority pending interrupt register

偏移地址：0x028

复位值：0bXXXX XXXX XXXX XXXX XXX0 0011 1111 1111

ICC_AHPPIR 是非安全 GICC_HPPIR 的别名。对此寄存器的安全访问将等同于对 GICC_HPPIR 的非安全访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CPUID	PENDINTID[9:0]									
					r	r	r	r	r	r	r	r	r	r	r

位 31:11 保留，必须保持复位值。

位 10 **CPUID**：

在多处理器实现中，如果 PENDINTID 字段返回了 SGI 的 ID，则该字段将包含该中断的 CPUID 值。这将标识生成中断的处理器。

位 9:0 **PENDINTID[9:0]**：最高优先级挂起中断的中断 ID (interrupt ID of the highest-priority pending interrupt)

如果最高优先级挂起中断是组 1 中断，则此位将包含该中断的中断 ID。否则，其值将为虚假中断 ID，1023。

19.6.14 GICC 有效优先级寄存器 (GICC_APR0)

GICC active priority register

偏移地址：0x0D0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
APR0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APR0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **APR0[31:0]**：有效优先级 (active priority)

19.6.15 GICC 非安全有效优先级寄存器 (GICC_NSAPR0)

GICC non-secure active priority register

偏移地址：0x0E0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NSAPR0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSAPR0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **NSAPR0[31:0]**：非安全有效优先级 (non-secure active priority)

19.6.16 GICC 接口标识寄存器 (GICC_IIDR)

GICC interface identification register

偏移地址：0x0FC

复位值：0x0102 143B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRODUCTID[11:0]												ARCH[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION[3:0]				IMPLEMENTER[11:0]											
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:20 **PRODUCTID[11:0]**：产品 ID (product ID)

位 19:16 **ARCH[3:0]**：GIC 的架构版本 (architecture version of the GIC)

位 15:12 **REVISION[3:0]**：CPU 接口的版本号 (revision number for the CPU interface)

位 11:0 **IMPLEMENTER[11:0]**：GIC 实现器（0x43B Arm 实现）

19.6.17 GICC 禁用中断寄存器 (GICC_DIR)

GICC deactivate interrupt register

偏移地址：0x1000

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CPUID	INTERRUPT_ID[9:0]									
					w	w	w	w	w	w	w	w	w	w	w

位 31:11 保留，必须保持复位值。

位 10 **CPUID**：CPU ID

对于多处理器实现中的 SGI，该字段将标识请求中断的处理器

位 9:0 **INTERRUPT_ID[9:0]**：中断 ID (interrupt ID)

19.6.18 GICC 寄存器映射

表 114. GICC 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	GICC_CTLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EOIMODENS	EOIMODES	IRQBYPDISGRP1	FIQBYPDISGRP1	IRQBYPDISGRP0	FIQBYPDISGRP0	CBPR	FIQEN	ACKCTL	ENABLEGRP1	ENABLEGRP0
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x000	GICC_CTLRNS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EOIMODENS	Res.	Res.	IRQBYPDISGRP1	FIQBYPDISGRP1	Res.	Res.	Res.	Res.	ENABLEGRP1
	Reset value																							0			0	0					0

表 114. GICC 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x004	GICC_PMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIORITY[4:0]					Res.	Res.	Res.
	Reset value																									0	0	0	0	0			
0x008	GICC_BPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BINARY_POINT[2:0]		
	Reset value																														0	1	0
0x008	GICC_BPRNS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BINARY_POINT[2:0]		
	Reset value																														0	1	1
0x00C	GICC_IAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPUID	INTERRUPT_ID[9:0]									
	Reset value																						0	1	1	1	1	1	1	1	1	1	1
0x010	GICC_EOIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPUID	EOIINTID[9:0]									
	Reset value																						X	X	X	X	X	X	X	X	X	X	X
0x014	GICC_RPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIORITY[4:0]					Res.	Res.	Res.
	Reset value																									1	1	1	1	1			
0x018	GICC_HPPIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPUID	PENDINTID[9:0]									
	Reset value																						0	1	1	1	1	1	1	1	1	1	1
0x01C	GICC_ABPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BINARY_POINT[2:0]		
	Reset value																														0	1	1
0x020	GICC_AIAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPUID	INTERRUPT_ID[9:0]									
	Reset value																						0	1	1	1	1	1	1	1	1	1	1
0x024	GICC_AEOIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPUID	EOIINTID[9:0]									
	Reset value																						X	X	X	X	X	X	X	X	X	X	X
0x028	GICC_AHPPIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPUID	PENDINTID[9:0]									
	Reset value																						0	1	1	1	1	1	1	1	1	1	1
0x02C - 0x0CC	Reserved	Reserved																															

表 114. GICC 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0D0	GICC_APR0	APR0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0D4 - 0x0DC	Reserved	Reserved																															
0x0E0	GICC_NSAPR0	NSAPR0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0E4 - 0x0F8	Reserved	Reserved																															
0x0FC	GICC_IIDR	PRODUCTID[11:0]												ARCH[3:0]				REVISION[3:0]				IMPLEMENTER[11:0]											
	Reset value	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	1	1
0x100 - 0xFFC	Reserved	Reserved																															
0x1000	GICC_DIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPUID	INTERRUPT_ID[9:0]									
	Reset value																						X	X	X	X	X	X	X	X	X	X	X

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

19.7 GIC 虚拟接口控制，通用 (GICH)

虚拟接口控制寄存器为管理寄存器。Cortex-A7 MPCore 处理器上的配置软件必须确保这些寄存器仅可通过管理程序或类似软件进行访问。

所有寄存器均支持字访问。

19.7.1 GICH 管理程序控制寄存器 (GICH_HCR)

GICH hypervisor control register

偏移地址：0x000

复位值：0b0000 0XXX XXXX XXXX XXXX XXXX 0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EOICOUNT[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VGRP1DIE	VGRP1EIE	VGRP0DIE	VGRP0EIE	NPIE	LRENPIE	UIE	EN
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:27 **EOICOUNT[4:0]**：中断结束计数器 (end-of-interrupt counter)

计数接收到的没有对应条目的 EOIs（结束中断信号）。虚拟 CPU 接口在接收到匹配的 EOI 时会自动递增此字段。当 EOIs 没有清零活动优先级寄存器中的位时，GICH_APR 不会导致递增。

如果在此字段的值为 31 时发生 EOI（尽管在正确操作的情况下不可能发生），此字段将返回至 0。

只要此字段为非零且 LRENPIE 位置 1，维护中断就会生效。

位 26:8 保留，必须保持复位值。

位 7 **VGRP1DIE**：虚拟机禁止组 1 中断使能 (virtual machine disable group 1 interrupt enable)

当系统禁止将组 1 中断信号从虚拟 CPU 接口发送到所连接的虚拟机时，使能发送维护中断信号。

位 6 **VGRP1EIE**：虚拟机使能组 1 中断使能 (virtual machine enable group 1 interrupt enable)

当系统使能将组 1 中断信号从虚拟 CPU 接口发送到所连接的虚拟机时，使能发送维护中断信号。

位 5 **VGRP0DIE**：虚拟机禁止组 0 中断使能 (virtual machine disable group 0 interrupt enable)

当系统禁止将组 0 中断信号从虚拟 CPU 接口发送到所连接的虚拟机时，使能发送维护中断信号。

位 4 **VGRP0EIE**：虚拟机使能组 0 中断使能 (virtual machine enable group 0 interrupt enable)

当系统使能将组 0 中断信号从虚拟 CPU 接口发送到所连接的虚拟机时，使能发送维护中断信号。

位 3 **NPIE**：无挂起中断使能 (no pending interrupt enable)

当列表寄存器中没有挂起中断时，使能发送维护中断信号。

位 2 **LRENPIE**：列表寄存器条目不存在中断使能 (list register entry not present interrupt enable)

当虚拟 CPU 接口没有与 EOI 请求相对应的有效列表寄存器条目时，使能发送维护中断信号。

位 1 **UIE**：下溢中断使能。

当列表寄存器为空或只包含一个有效条目时，使能发送维护中断信号。

位 0 **EN**：虚拟 CPU 接口的全局使能位 (global enable bit for the virtual CPU interface)

19.7.2 GICH VGIC 类型寄存器 (GICH_VTR)

GICH VGIC type register

偏移地址：0x004

复位值：0x9000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRIBITS[2:0]			PREBITS[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r	r										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LISTREGS[4:0]				
											r	r	r	r	r

位 31:29 **PRIBITS[2:0]**：优先级位 (priority bits)
表示已实现的优先级位数减一。
例如，0x4 表示 5 个优先级位和 32 个优先级。

位 28:26 **PREBITS[2:0]**：抢占位 (preemption bits)
表示已实现的抢占位数减一。
例如，0x4 表示 5 个抢占位和 32 个抢占级别

位 25:5 保留，必须保持复位值。

位 4:0 **LISTREGS[4:0]**：列表寄存器 (list registers)
表示已实现的列表寄存器数减一。
例如，0x3 表示 4 个列表寄存器。

19.7.3 GICH 虚拟机控制寄存器 (GICH_VMCR)

GICH virtual machine control register

偏移地址：0x008

复位值：0b0000 0XXX 0100 11X1 XXXX XX0X XXX0 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VMPRIMASK[4:0]					Res.	Res.	Res.	VMBP[2:0]			VMABP[2:0]			Res.	Res.
rw	rw	rw	rw	rw				rw	rw	rw	rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	VEM	Res.	Res.	Res.	Res.	VMCBPR	VMFIQEN	VMACKCTL	VMGRP1EN	VMGRP0EN
						rw					rw	rw	rw	rw	rw

位 31:27 **VMPRIMASK[4:0]**：GICV_PMR.PRIORITY 的别名

位 26:24 保留，必须保持复位值。

位 23:21 **VMBP[2:0]**：GICV_BPR.BINARY_POINT 的别名

位 20:18 **VMABP[2:0]**：GICV_ABPR.BINARY_POINT 的别名。

位 17:10 保留，必须保持复位值。

位 9 **VEM**：GICV_CTLR.EOIMODE 的别名

位 8:5 保留，必须保持复位值。

位 4 **VMCBPR**：GICV_CTLR.CBPR 的别名

位 3 **VMFIQEN**：GICV_CTLR.FIQEN 的别名

位 2 **VMACKCTL**：GICV_CTLR.ACKCTL 的别名

位 1 **VMGRP1EN**：GICV_CTLR.ENABLEGRP1 的别名

位 0 **VMGRP0EN**：GICV_CTLR.ENABLEGRP0 的别名

19.7.4 GICH 维护中断状态寄存器 (GICH_MISR)

GICH maintenance interrupt status register

偏移地址：0x010

复位值：0bXXXX XXXX XXXX XXXX XXXX XXXX 0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VGRP1D	VGRP1E	VGRP0D	VGRP0E	NP	LRENP	U	EOI
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7 **VGRP1D**：已禁止组 1 维护中断 (disabled group 1 maintenance interrupt)
当 GICH_HCR.VGRP1DIE 置 1 且 GICH_VMCR.VMGRP1EN=0 时生效。

位 6 **VGRP1E**：已使能组 1 维护中断 (enabled group 1 maintenance interrupt)
当 GICH_HCR.VGRP1EIE 置 1 且 GICH_VMCR.VMGRP1EN=1 时生效。

位 5 **VGRP0D**：已禁止组 0 维护中断 (disabled group 0 maintenance interrupt)
当 GICH_HCR.VGRP0DIE 置 1 且 GICH_VMCR.VMGRP0EN=0 时生效。

位 4 **VGRP0E**：已使能组 0 维护中断 (enabled group 0 maintenance interrupt)
当 GICH_HCR.VGRP0EIE 置 1 且 GICH_VMCR.VMGRP0EN=1 时生效。

位 3 **NP**：无挂起维护中断 (no pending maintenance interrupt)
当 GICH_HCR.NPIE=1 且无任何列表寄存器处于挂起状态时生效。

位 2 **LRENP**：列表寄存器条目不存在维护中断 (list register entry not present maintenance interrupt)
当 GICH_HCR.LRENPIE=1 且 GICH_HCR.EOICOUNT 为非零时生效。

位 1 **U**：下溢维护中断 (underflow maintenance interrupt)
当 GICH_HCR.UIE 置 1 且列表寄存器条目均未被标记或仅有一个被标记为有效中断（即相应的 GICH_LRx.STATE 位不等于 0x0）时生效。

位 0 **EOI**：中断结束维护中断 (End of interrupt maintenance interrupt)
当至少有一个列表寄存器触发 EOI 中断（这意味着 GICH_EISRx 中至少有一个位 =1）时生效。

19.7.5 GICH 中断结束状态寄存器 (GICH_EISR0)

GICH end of interrupt status register

偏移地址：0x020

复位值：0bXXXX XXXX XXXX XXXX XXXX XXXX XXXX 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EISR0[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EISR0[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **EISR0[31:0]**：中断结束状态 (end of interrupt status)

19.7.6 GICH 空列表状态寄存器 (GICH_ELSR0)

GICH empty list status register

偏移地址：0x030

复位值：0bXXXX XXXX XXXX XXXX XXXX XXXX XXXX 1111

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ELSR0[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ELSR0[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ELSR0[31:0]**：空列表状态 (empty list status)

19.7.7 GICH 有效优先级寄存器 (GICH_APR0)

GICH active priority register

偏移地址：0x0F0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
APR0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APR0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **APR0[31:0]**：有效优先级 (active priority)

19.7.8 GICH 列表寄存器 x (GICH_LRx)

GICH list register x

偏移地址：0x100 + 0x4 * x，（x = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HW	GRP1	STATE[1:0]		PRIORITY[4:0]					Res.	Res.	Res.	PHYSICALID[9:6]			
rw	rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHYSICALID[5:0]						VIRTUALID[9:0]									
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **HW**：硬件中断 (hardware interrupt)

指示此虚拟中断是否为硬件中断，即该中断是否与物理中断相对应。禁用虚拟中断的同时也会禁用具有 PhysicalID 字段所示 ID 的物理中断。

位 30 **GRP1**：指示此虚拟中断是否为组 1 虚拟中断

位 29:28 **STATE[1:0]**：中断状态 (state of the interrupt)

位 27:23 **PRIORITY[4:0]**：中断优先级 (priority of the interrupt)

位 22:20 保留，必须保持复位值。

位 19:10 **PHYSICALID[9:0]**：物理 ID (physical ID)

此位的功能取决于 GICH_LR.HW 位的值，如下所示。

1) 当 GICH_LR.HW 置 0 时，位 [9:0] 将具有以下含义：

- [19] EOI（当此中断触发 EOI 维护中断时）

- [12:10] CPUID，前提是此中断拥有 SGI 的 VirtualID，即 0-15。此字段将显示所请求的 CPU ID。此情况会出现在虚拟机中断确认寄存器 GICV_IAR 或 GICV_AIAR 的相关字段中。否则，此字段必须置 0。

2) 当 GICH_LR.HW 置 1 时，PHYSICALID[9:0] 将表示管理程序转发到 GICD 的物理中断 ID。

如果 PHYSICALID 的值为 0-15 或 1020-1023，则其行为将变得**不可预测**。

如果 PHYSICALID 的值为 16-31，则此字段将适用于物理 CPUID 与请求禁用的虚拟 CPU 接口相同的相关 PPI。

位 9:0 **VIRTUALID[9:0]**：虚拟 ID (virtual ID)

当通过 VM 中断确认寄存器 GICV_IAR 确认中断时，系统会将此 ID 返回给用户操作系统。每个存储在列表寄存器中的有效中断都必须拥有一个面向该虚拟 CPU 接口的唯一虚拟 ID。

如果 VIRTUALID 的值为 1020-1023，则其行为将变得**不可预测**。

19.7.9 GICH 寄存器映射

表 115. GICH 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	GICH_HCR	EOICOUNT[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VGRP1DIE	VGRP1EIE	VGRP0DIE	VGRP0EIE	NPIE	LRENPIE	UIE	EN
	Reset value	0	0	0	0	0																				0	0	0	0	0	0	0	0

表 115. GICH 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x004	GICH_VTR	PRIBITS[2:0]			PREBITS[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LISTREGS[4:0]				
	Reset value	1	0	0	1	0	0																						0	0	0	1	1
0x008	GICH_VMCR	VMPRIMASK[4:0]					Res.	Res.	Res.	VMBP[2:0]			VMABP[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VMCBPR	VMFIQEN	VMACKCTL	VMGRP1EN	VMGRP0EN
	Reset value	0	0	0	0	0				0	1	0	0	1	1														0	0	0	0	0
0x00C	Reserved	Reserved																															
0x010	GICH_MISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VGRP1D	VGRP1E	VGRP0D	VGRP0E	NP	LRENP	U	EOI
	Reset value																									0	0	0	0	0	0	0	0
0x014 - 0x01C	Reserved	Reserved																															
0x020	GICH_EISR0	EISR0[31:0]																															
	Reset value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0
0x024 - 0x02C	Reserved	Reserved																															
0x030	GICH_ELSR0	ELSR0[31:0]																															
	Reset value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1	1	1
0x034 - 0x0EC	Reserved	Reserved																															
0x0F0	GICH_APR0	APR0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0F4 - 0x0FC	Reserved	Reserved																															
0x100 to 0x10C	GICH_LR0 to GICH_LR3	HW	GRP1	STATE[1:0]			PRIORITY[4:0]						Res.	Res.	Res.	PHYSICALID[9:0]										VIRTUALID[9:0]									
	Reset value	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

19.8 GIC 虚拟 CPU 接口 (GICV)

此虚拟 CPU 接口会遵循正常的 GIC 处理和优先级规则，将虚拟中断转发到所连接的 Cortex-A7 MPCore 处理器。虚拟接口控制寄存器将控制虚拟 CPU 接口的操作，尤其是虚拟 CPU 接口将使用列表寄存器中的内容来确定发送虚拟中断信号的时机。

当处理器访问虚拟 CPU 接口时，系统将更新列表寄存器。

所有寄存器均支持字访问。

19.8.1 GICV 虚拟机控制寄存器 (GICV_CTLR)

GICV virtual machine control register

偏移地址：0x000

复位值：0bXXXX XXXX XXXX XXXX XXXX X000 0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	EOIMODE	Res.	Res.	Res.	Res.	CBPR	FIQEN	ACKCTL	ENABLEGRP1	ENABLEGRP0
						rw					rw	rw	rw	rw	rw

位 31:10 保留，必须保持复位值。

位 9 **EOIMODE**：中断结束模式 (end of interrupt mode)
控制与 GICV_EOIR、GICV_AEOIR 和 GICV_DIR 寄存器相关的行为。

位 8:5 保留，必须保持复位值。

位 4 **CBPR**：BPR 控制 (BPR control)
控制 GICV_BPR 是否可控制组 0 和组 1 的虚拟中断。

位 3 **FIQEN**：组 0 中断的 FIQ 使能 (FIQ enable for group 0 interrupts)
控制是否以虚拟 FIQ 的形式呈现标记为组 0 的中断。

位 2 **ACKCTL**：确认控制 (acknowledge control)
控制当最高优先级挂起中断为组 1 中断时，读取 GICV_IAR 是否会导致 CPU 接口确认该中断。
Arm 不推荐使用此位。Arm 强烈建议在编写软件时，始终将此位置 0。

位 1 **ENABLEGRP1**：
使能虚拟 CPU 接口向虚拟机发送组 1 虚拟中断信号。

位 0 **ENABLEGRP0**：
使能虚拟 CPU 接口向虚拟机发送组 0 虚拟中断信号。

19.8.2 GICV VM 优先级屏蔽寄存器 (GICV_PMR)

GICV VM priority mask register

偏移地址：0x004

复位值：0bXXXX XXXX XXXX XXXX XXXX XXXX 0000 0XXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIORITY[4:0]					Res.	Res.	Res.
								rw	rw	rw	rw	rw			

位 31:8 保留，必须保持复位值。

位 7:3 **PRIORITY[4:0]**：虚拟 CPU 接口的优先级屏蔽级别
系统仅会将优先级高于此寄存器中值的虚拟中断信号发送给处理器

位 2:0 保留，必须保持复位值。

19.8.3 GICV VM 二进制点寄存器 (GICV_BPR)

GICV VM binary point register

偏移地址：0x008

复位值：0bXXXX XXXX XXXX XXXX XXXX XXXX XXXX X010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BINARY_POINT[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **BINARY_POINT[2:0]**：
此字段的值将控制如何将 8 位虚拟中断优先级字段划分为组优先级字段，以确定虚拟中断抢占优先级和子优先级。最小值为 2。

19.8.4 GICV VM 中断确认寄存器 (GICV_IAR)

GICV VM interrupt acknowledge register

偏移地址：0x00C

复位值：0bXXXX XXXX XXXX XXXX XXX0 0011 1111 1111

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CPUID	INTERRUPT_ID[9:0]									
					r	r	r	r	r	r	r	r	r	r	r

位 31:11 保留，必须保持复位值。

位 10 **CPUID**：
如果 GICH_LRx.HW 位置 0（表示软件中已触发该中断），则 GICV_IAR.CPUID 字段中将返回用于指示 CPU ID 的 GICH_LRx CPUID。否则 GICV_IAR.CPUID 字段将读为零。

位 9:0 **INTERRUPT_ID[9:0]**：中断 ID (interrupt ID)

19.8.5 GICV VM 中断结束寄存器 (GICV_EOIR)

GICV VM end of interrupt register

偏移地址：0x010

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CPUID	EOIINTID[9:0]									
					w	w	w	w	w	w	w	w	w	w	w

位 31:11 保留，必须保持复位值。

位 10 **CPUID**：

在多处理器实现中，如果写入操作涉及 SGI，则此字段将包含相应 GICV_IAR 访问中的 CPUID 值。

位 9:0 **EOIINTID[9:0]**：

如果匹配的列表寄存器中的 GICH_LRx.HW 位置 1（表示硬件中断），则系统会向物理分配器发送一个禁用请求，并标识列表寄存器的相应字段中的物理 ID。

19.8.6 GICV VM 运行优先级寄存器 (GICV_RPR)

GICV VM running priority register

偏移地址：0x014

复位值：0bXXXX XXXX XXXX XXXX XXXX XXXX 1111 1111

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIORITY[4:0]					Res.	Res.	Res.
								r	r	r	r	r			

位 31:8 保留，必须保持复位值。

位 7:3 **优先级 [4:0]**：虚拟 CPU 接口上的当前运行优先级

位 2:0 保留，必须保持复位值。

19.8.7 GICV VM 最高优先级挂起中断寄存器 (GICV_HPPIR)

GICV VM highest priority pending interrupt register

偏移地址：0x018

复位值：0bXXXX XXXX XXXX XXXX XXX0 0011 1111 1111

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CPUID	PENDINTID[9:0]									
					r	r	r	r	r	r	r	r	r	r	r

位 31:11 保留，必须保持复位值。

位 10 **CPUID**：在多处理器实现中，如果 GICH_LRx.HW 位=0，则此字段将包含该虚拟中断的 CPUID 值。这将标识生成虚拟中断的虚拟处理器。

位 9:0 **PENDINTID[9:0]**：最高优先级挂起虚拟中断的虚拟中断 ID

19.8.8 GICV VM 别名二进制点寄存器 (GICV_ABPR)

GICV VM aliased binary point register

偏移地址：0x01C

复位值：0bXXXX XXXX XXXX XXXX XXXX XXXX XXXX X011

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BINARY_POINT[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **BINARY_POINT[2:0]**：

此字段的值将控制如何将 8 位虚拟中断优先级字段划分为组优先级字段，以确定虚拟中断抢占优先级和子优先级。最小值为 3。

19.8.9 GICV VM 别名中断寄存器 (GICV_AIAR)

GICV VM aliased interrupt register

偏移地址：0x020

复位值：0bXXXX XXXX XXXX XXXX XXX0 0011 1111 1111

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CPUID	INTERRUPT_ID[9:0]									
					r	r	r	r	r	r	r	r	r	r	r	r

位 31:11 保留，必须保持复位值。

位 10 **CPUID**：如果 GICH_LRx.HW 位置 0（表示软件中已触发该中断），则 GICV_AIAR.CPUID 字段中将返回用于指示 CPU ID 的 GICH_LRx CPUID。否则 GICV_AIAR.CPUID 字段将读为零。

位 9:0 **INTERRUPT_ID[9:0]**：中断 ID。如果 GICH_LRx.Grp1 位为 0，则中断将为组 0。系统将返回虚假中断 ID 1023，并且不会确认此中断。

19.8.10 GICV VM 别名中断结束寄存器 (GICV_AEOIR)

GICV VM aliased end of interrupt register

偏移地址：0x024

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CPUID	EOIINTID[9:0]									
					w	w	w	w	w	w	w	w	w	w	w	w

位 31:11 保留，必须保持复位值。

位 10 **CPUID**：

在多处理器实现中，如果写入操作涉及 SGI，则此字段将包含相应 GICV_IAR 访问中的 CPUID 值。

位 9:0 **EOIINTID[9:0]**：

如果匹配的列表寄存器中的 GICH_LRx.HW 位置 1（表示硬件中断），则系统会向物理分配器发送一个禁用请求，并标识列表寄存器的相应字段中的物理 ID。

19.8.11 GICV VM 别名最高优先级挂起中断寄存器 (GICV_AHPPIR)

GICV VM aliased highest priority pending interrupt register

偏移地址：0x028

复位值：0bXXXX XXXX XXXX XXXX XXX0 0011 1111 1111

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CPUID	PENDINTID[9:0]									
					r	r	r	r	r	r	r	r	r	r	r

位 31:11 保留，必须保持复位值。

位 10 **CPUID**：

如果匹配的列表寄存器中的 GICH_LRx.HW 位置 1（表示硬件中断），则系统会向物理分配器发送一个禁用请求，并标识列表寄存器的相应字段中的物理 ID。

位 9:0 **PENDINTID[9:0]**：

最高优先级挂起虚拟中断的虚拟中断 ID，前提是该虚拟中断是组 1 虚拟中断。否则此 ID 将为虚假虚拟中断 ID，1023。

19.8.12 GICV VM 有效优先级寄存器 (GICV_APR0)

GICV VM active priority register

偏移地址：0x0D0

复位值：0x0000 0000

GICV_APR0 是 GICH_APR 的别名。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
APR0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APR0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **APR0[31:0]**：有效优先级 (active priority)

19.8.13 GICV VM CPU 接口标识寄存器 (GICV_IIDR)

GICV VM CPU interface identification register

偏移地址：0x0FC

复位值：0x0102 143B

GICV_IIDR 是 GICC_IIDR 的别名。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IIDR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IIDR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **IIDR[31:0]**：请参见 GICC_IIDR 寄存器的说明

19.8.14 GICV VM 禁用中断寄存器 (GICV_DIR)

GICV VM deactivate interrupt register

偏移地址：0x1000

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CPUID	INTERRUPT_ID[9:0]									
					w	w	w	w	w	w	w	w	w	w	w

位 31:11 保留，必须保持复位值。

位 10 **CPUID**：此字段将标识请求中断的处理器。

位 9:0 **INTERRUPT_ID[9:0]**：中断 ID (interrupt ID)

19.8.15 GICV 寄存器映射

表 116. GICV 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	GICV_CTLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EOIMODE	Res.	Res.	Res.	Res.	CBPR	FIQEN	ACKCTL	ENABLEGRP1	ENABLEGRP0
	Reset value																							0					0	0	0	0	0

表 116. GICV 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x004	GICV_PMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIORITY[4:0]					Res.	Res.	Res.
	Reset value																									0	0	0	0	0			
0x008	GICV_BPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BINARY_POINT[2:0]		
	Reset value																														0	1	0
0x00C	GICV_IAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPUID	INTERRUPT_ID[9:0]									
	Reset value																						0	1	1	1	1	1	1	1	1	1	1
0x010	GICV_EOIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPUID	EOIINTID[9:0]									
	Reset value																						X	X	X	X	X	X	X	X	X	X	X
0x014	GICV_RPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIORITY[4:0]					Res.	Res.	Res.
	Reset value																									1	1	1	1	1			
0x018	GICV_HPPIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPUID	PENDINTID[9:0]									
	Reset value																						0	1	1	1	1	1	1	1	1	1	1
0x01C	GICV_ABPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BINARY_POINT[2:0]		
	Reset value																														0	1	1
0x020	GICV_AIAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPUID	INTERRUPT_ID[9:0]									
	Reset value																						0	1	1	1	1	1	1	1	1	1	1
0x024	GICV_AEOIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPUID	EOIINTID[9:0]									
	Reset value																						X	X	X	X	X	X	X	X	X	X	X
0x028	GICV_AHPPIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPUID	PENDINTID[9:0]									
	Reset value																						0	1	1	1	1	1	1	1	1	1	1
0x02C - 0x0CC	Reserved	Reserved																															
0x0D0	GICV_APR0	APR0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0D4 - 0x0F8	Reserved	Reserved																															
0x0FC	GICV_IIDR	IIDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	1	1
0x100 - 0xFFC	Reserved	Reserved																															

表 116. GICV 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1000	GICV_DIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPUID	INTERRUPT_ID[9:0]									
	Reset value																						X	X	X	X	X	X	X	X	X	X	X

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

20 扩展中断和事件控制器 (EXTI)

扩展中断和事件控制器 (EXTI) 通过可配置的事件输入和直接事件输入来管理单独的 CPU 和系统唤醒。它可以为电源控制提供唤醒请求，给 CPU 中断控制器产生中断请求以及给 CPU 的事件输入产生事件。对于 CPU，需通过额外的事件生成模块 (EVG) 来生成 CPU 事件信号。

EXTI 唤醒请求可以唤醒处于停止模式的系统。

此外，还可以在运行模式下生成中断请求和事件请求。

EXTI 还包含 EXTI 多路复用输入/输出端口选择功能。

CPU = MPU

20.1 EXTI 主要特性

EXTI 的主要特性如下：

- 支持 76 个输入事件
- 所有事件输入都允许唤醒系统
- 没有关联唤醒标志的外设事件，在 EXTI 中会有一个标志，并通过 EXTI 生成中断请求到 CPU

异步事件输入分为两组：

- 可配置事件（来自能够生成脉冲的 I/O 或外设的信号）
  - 可配置事件具有以下特性：
    - 可选择的有效触发边沿
    - 独立于上升沿和下降沿的中断挂起状态寄存器位
    - 单独的中断和事件生成屏蔽，用于调节 CPU 唤醒、中断和事件生成
    - 支持软件触发
- 直接事件（来自具有相关标志的外设的中断和唤醒源，需要在外设中清除）
  - 直接事件具有以下特性：
    - 固定上升沿有效触发
    - EXTI 中无中断挂起状态寄存器位（中断挂起状态标志由生成事件的外设提供）
    - 单独的中断和事件生成屏蔽，用于调节 CPU 唤醒和事件生成
    - 不支持软件触发
- TrustZone 安全事件
  - 对安全输入事件的控制和配置位的访问可设为安全访问。
- EXTI IO 端口选择

20.2 EXTI 框图

EXTI 包含一个通过 AHB 接口访问的寄存器模块、一个事件输入触发模块、一个屏蔽模块和一个 EXTI 复用器，如*图 121* 所示。

寄存器模块包含所有 EXTI 寄存器。

事件输入触发模块提供事件输入边沿触发逻辑。

屏蔽模块为不同的唤醒、中断和事件输出及其屏蔽功能提供事件输入分配。

EXTI 复用器为 EXTI 事件信号选择 IO 端口。

图 121. EXTI 框图

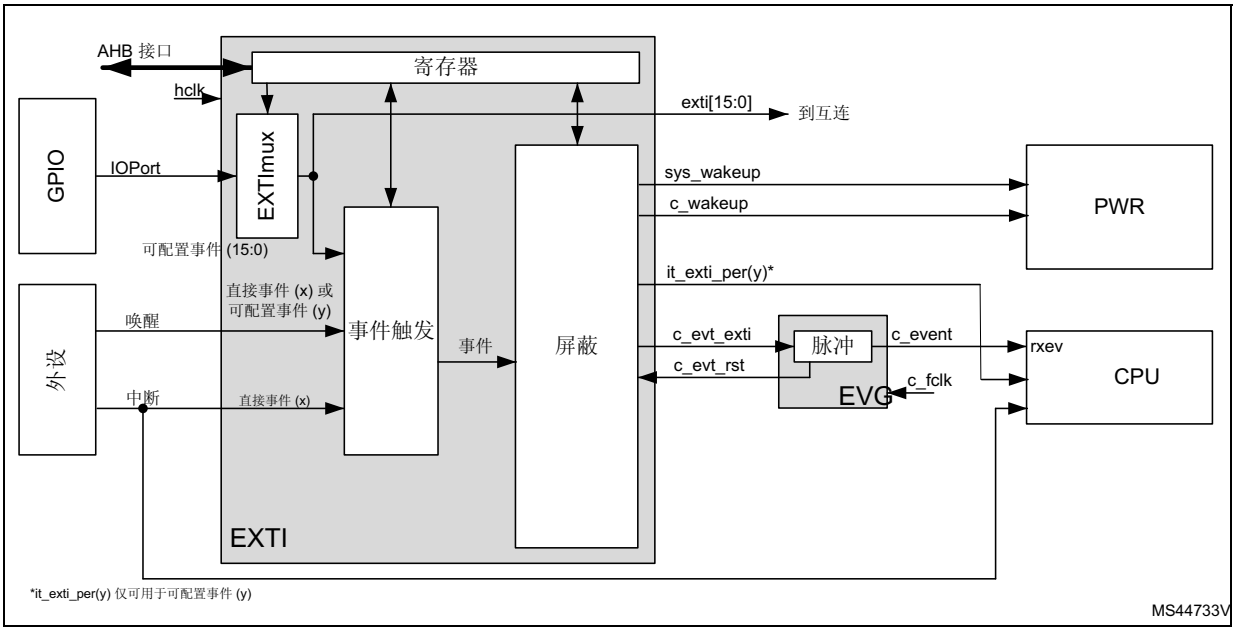


表 117. EXTI 引脚概述

引脚名称	I/O	说明
AHB 接口	I/O	EXTI 寄存器总线接口。当将一个事件配置为允许安全性时，AHB 接口支持安全访问。
hclk	I	AHB 总线时钟和 EXTI 系统时钟。
可配置事件 (y)	I	来自没有相关中断和标志的外设的异步唤醒事件。
直接事件 (x)	I	来自具有相关中断和标志的外设的同步与异步唤醒事件。
IOPort(n)	I	GPIO 模块 IO 端口[15:0]。
exti[15:0]	O	用于触发其他 IP 的 EXTI 输出端口。
it_exti_per (y)	O	与可配置事件 (y) 关联的 CPU 的中断。
c_evt_exti	O	面向 CPU 的高电平触发事件输出，与 hclk 同步。
c_evt_rst	I	用于清除 c_evt_exti 的异步复位输入。
sys_wakeup	O	面向 ck_sys 和 hclk 的 PWR 的异步系统唤醒请求。
c_wakeup	O	CPU 向 PWR 发出的唤醒请求，与 hclk 同步。

表 118. EVG 引脚概述

引脚名称	I/O	说明
c_fclk	I	CPU 自由运行时钟。
c_evt_in	I	来自 EXTI 的高电平触发事件输入，与 CPU 时钟异步。
c_event	O	事件脉冲，与 CPU 时钟同步。
c_evt_rst	O	事件复位信号，与 CPU 时钟同步。

20.2.1 外设和 CPU 之间的 EXTI 连接

对于能够在系统处于停止模式时生成唤醒或中断事件的外设，将连接至 EXTI。

- 对于生成脉冲或在外设中没有中断状态位的外设唤醒信号，将连接至 EXTI 可配置事件输入。对于这类事件，EXTI 提供一个必须清零的状态挂起位。与状态位相关的 EXTI 中断将中断 CPU。
- 对于在外设中有状态位（必须在外设中清零）的外设中断和唤醒信号，将连接至 EXTI 直接事件输入。EXTI 中没有状态挂起位。中断或唤醒由外设中的 CPU 清除。外设中断会直接中断 CPU。
- 所有 GPIO 端口输入到 EXTI 复用器，可以选择端口引脚以便通过可配置的事件来唤醒系统。

EXTI 可配置事件中断连接到 CPU 的中断控制器。

专用 EXTI/EVG CPU 事件连接至 CPU rxev 输入。

EXTI CPU 唤醒信号连接至 PWR 模块，用于唤醒系统和 CPU 子系统总线时钟。

20.3 EXTI 功能描述

根据 EXTI 事件输入类型和唤醒目标，将使用不同的逻辑实现。适用的功能通过寄存器位进行控制：

- 有效触发边沿使能，具体包括上升沿选择
  EXTI 上升沿触发选择寄存器 (EXTI_RTSR1),
  EXTI 上升沿触发选择寄存器 (EXTI_RTSR2),
  EXTI 上升沿触发选择寄存器 (EXTI_RTSR3),
  和下降沿选择
  EXTI 下降沿触发选择寄存器 (EXTI_FTSR1),
  EXTI 下降沿触发选择寄存器 (EXTI_FTSR2),
  EXTI 下降沿触发选择寄存器 (EXTI_FTSR3).
- 软件触发，具体包括
  EXTI 软件中断事件寄存器 (EXTI_SWIER1),
  EXTI 软件中断事件寄存器 (EXTI_SWIER2),
  EXTI 软件中断事件寄存器 (EXTI_SWIER3).
- 中断挂起标志，具体包括
  EXTI 上升沿挂起寄存器 (EXTI_RPR1),
  EXTI 上升沿挂起寄存器 (EXTI_RPR2)
  EXTI 上升沿挂起寄存器 (EXTI_RPR3)
  EXTI 下降沿挂起寄存器 (EXTI_FPR1)
  EXTI 下降沿挂起寄存器 (EXTI_FPR2)
  EXTI 下降沿挂起寄存器 (EXTI_FPR3).
- CPU 唤醒和中断使能，具体包括
  EXTI CPU 唤醒中断屏蔽寄存器 (EXTI_IMR1),
  EXTI CPU 唤醒中断屏蔽寄存器 (EXTI_IMR2),
  EXTI CPU 唤醒中断屏蔽寄存器 (EXTI_IMR3).
- CPU 唤醒和事件使能，具体包括
  EXTI CPU 唤醒事件屏蔽寄存器 (EXTI_EMR1)
  EXTI CPU 唤醒事件屏蔽寄存器 (EXTI_EMR2),
  EXTI CPU 唤醒事件屏蔽寄存器 (EXTI_EMR3).

表 119. EXTI 事件输入配置和寄存器控制

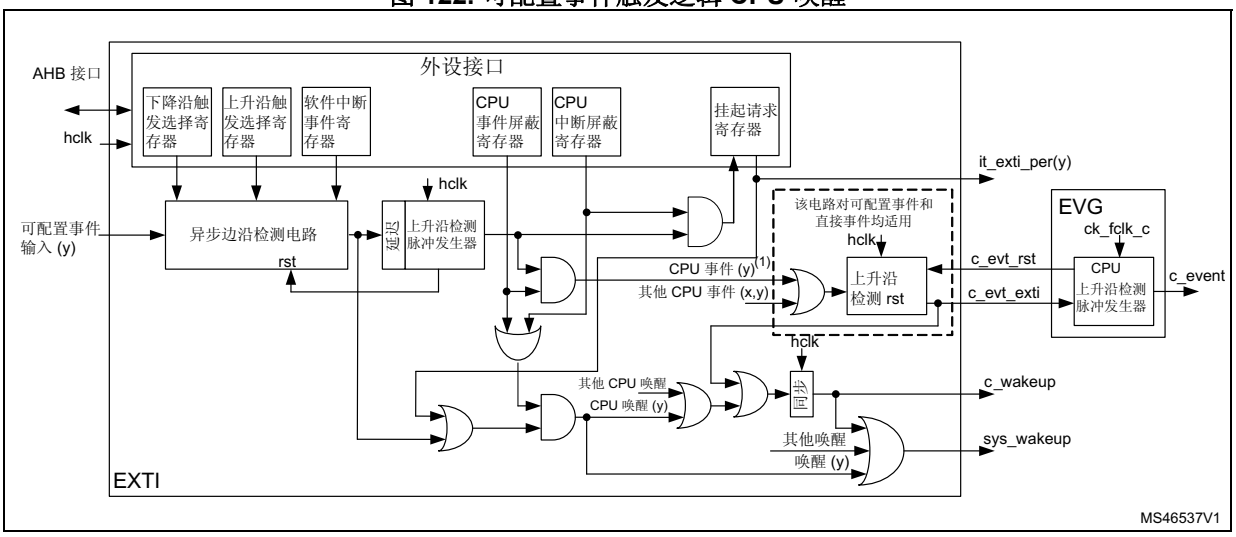
事件输入类型	逻辑实现	EXTI_RTSR	EXTI_FTSR	EXTI_SWIER	EXTI_R/FPR	EXTI_IMR	EXTI_EMR(1)
可配置	可配置事件输入唤醒逻辑	x	x	x	x	x	x
直接	直接事件输入唤醒逻辑	-	-	-	-	x	x

1. 仅适用于使能了“rxev 生成”配置的输入事件。

20.3.1 EXTI 可配置事件输入唤醒

图 122 所示为具有如下用途的可配置事件输入的相关逻辑详图：唤醒 CPU 子系统总线时钟并生成 CPU 的 EXTI 挂起标志与中断以及 CPU 唤醒事件。

图 122. 可配置事件触发逻辑 CPU 唤醒



1. 仅适用于支持 CPU rxev 生成 c_event 的输入事件。

软件中断事件寄存器允许通过软件写入相应的寄存器位来触发可配置事件，而边沿选择设置无关紧要。

上升沿和下降沿选择寄存器允许使能和选择可配置事件有效触发边沿或两种边沿。

CPU 具有专用的中断屏蔽寄存器和专用的事件屏蔽寄存器。使能的事件允许在 CPU 上生成一个事件。CPU 的所有事件会一起进行逻辑或运算，生成单个 CPU 事件信号。事件挂起寄存器（EXTI_RPR 和 EXTI_FPR）不会针对未屏蔽的 CPU 事件而置位。

可配置事件具有唯一的中断挂起请求寄存器（由 CPU 使用）。挂起寄存器只会针对未屏蔽的中断而置位。每个可配置事件均提供对 CPU 的中断。可配置事件中断需要由软件通过 EXTI_RPR 和/或 EXTI_FPR 寄存器确认。

使能 CPU 中断或 CPU 事件时，异步边沿检测电路由时钟延迟和上升沿检测脉冲发生器复位。这可以确保 EXTI hclk 时钟在异步边沿检测电路复位之前唤醒。

注： *检测到的可配置事件中断挂起请求，可由任何 CPU 清除。只要中断挂起请求处于活动状态，系统就无法进入低功耗模式。*

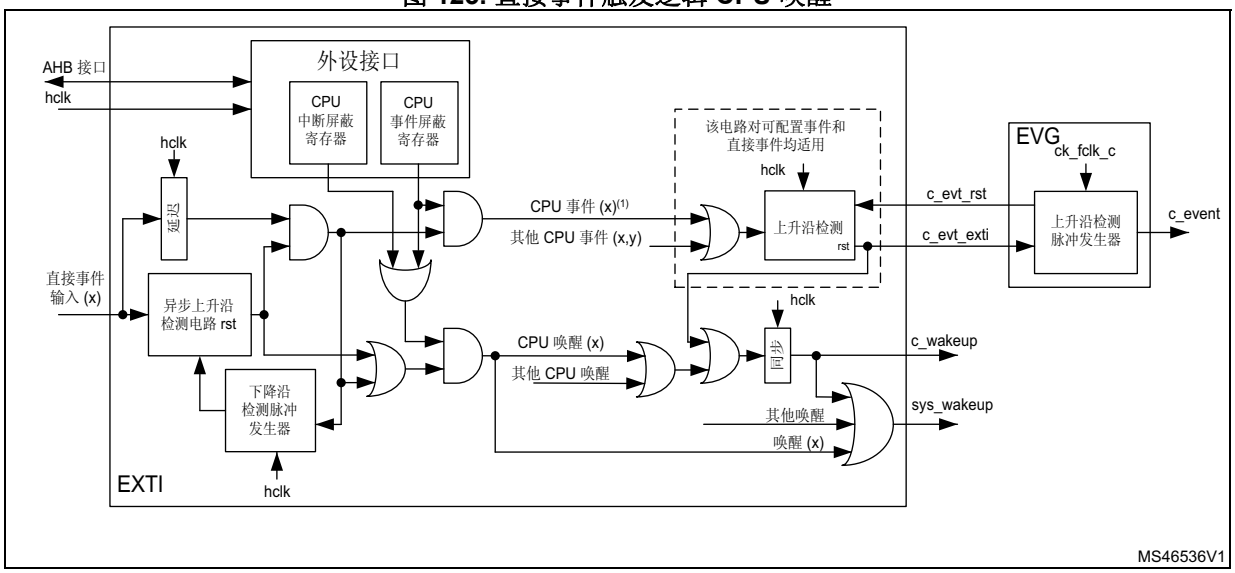
20.3.2 EXTI 直接事件输入唤醒

图 123 所示为唤醒系统的直接事件输入的相关逻辑详图。

直接事件没有关联的 EXTI 中断。EXTI 仅唤醒系统和 CPU 子系统时钟，并可能生成 CPU 唤醒事件。与直接唤醒事件相关的外设同步中断会唤醒 CPU。

EXTI 直接事件能够生成 CPU 事件。此 CPU 事件会唤醒 CPU。CPU 事件可能在相关外设中断标志置 1 之前发生。

图 123. 直接事件触发逻辑 CPU 唤醒

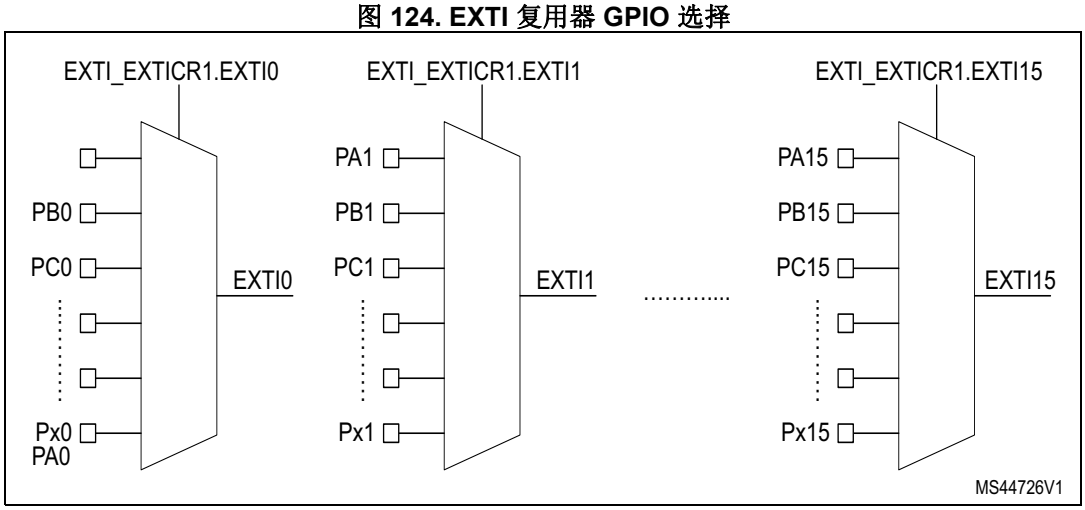


1. 仅适用于支持 CPU rxev 生成 c_event 的输入事件。

20.3.3 EXTI 复用器选择

EXTI 复用器可选择作为中断和唤醒的 GPIO。各 GPIO 通过 16 根 EXTI 复用器线连接到前 16 个 EXTI 事件（均为可配置事件）。选择哪个 GPIO 端口作为 EXTI 复用器输出，由以下寄存器控制：*EXTI 外部中断选择寄存器 1 (EXTI_EXTICR1)* 到 *EXTI 外部中断选择寄存器 4 (EXTI_EXTICR4)*。

图 124. EXTI 复用器 GPIO 选择



EXTI 复用器输出可作为 EXTI 的输出信号，用于触发其他 IP。EXTI 复用器输出的使用不受 EXTI_IMR1 和 EXTI_EMR1 中任何屏蔽的影响。

20.4 EXTI 功能行为

在生成唤醒事件的相应外设中使能直接事件输入。通过使能至少一个触发沿来使能可配置事件。

使能事件输入后，是否生成 CPU 唤醒取决于 CPU 中断屏蔽和 CPU 事件屏蔽。

表 120. 屏蔽功能

CPU 中断使能 EXTI_IMR.IMn	CPU 事件使能 EXTI_EMR.EMn	可配置事件输入 EXTI_RPR.RPIFn EXTI_FPR.FPIFn	exti(n) 中断(1)	CPU 事件	CPU 唤醒
0	0	否	已屏蔽	已屏蔽	已屏蔽
	1		已屏蔽	是	是
1	0	状态已锁存	是	已屏蔽	是(2)
	1		是	是	是

1. 单个 exti(n) 中断将面向 CPU。如果 CPU 不需要中断，则必须在 CPU 中断控制器中屏蔽 exti(n) 中断。
2. 仅限 EXTI_IMR.IMn 中已使能 CPU 中断的情况。

对于可配置事件输入，当事件输入上出现使能的边沿时，会生成一个事件请求。相关 CPU 中断未被屏蔽时，相应的挂起位 EXTI_RPR.RPIFn 和/或 EXTI_FPR.FPIFn 会置 1，CPU 子系统被唤醒，且 CPU 中断信号被激活。EXTI_RPR.RPIFn 和/或 EXTI_FPR.FPIFn 挂起位必须通过软件写入“1”来清零。此操作会清除 CPU 中断。

对于直接事件输入，当在相关外设中使能时，只会在上升沿生成事件请求。EXTI 中没有相应的 CPU 挂起位。当相关 CPU 中断未被屏蔽时，会唤醒相应的 CPU 子系统。CPU 由外设同步中断唤醒（中断）。

要生成事件，CPU 事件必须处于未屏蔽状态。当事件输入上出现使能的边沿时，会生成 CPU 事件脉冲。不存在事件挂起位。

对于可配置事件输入，可通过软件在软件中断/事件寄存器 EXTI_SWIER 中写入“1”来生成事件请求，同时基于事件生成上升沿。上升沿事件挂起位在 EXTI_RPR 中置 1，与 EXTI_RTSR 中的设置无关。

20.5 EXTI TrustZone 安全性

EXTI 可以保护寄存器位免受非安全访问的修改。通过 EXTI_TZENRn 中的寄存器位，可以为每个输入事件单独激活 TrustZone 安全性。在 EXTI 级别，安全性在于防止通过非安全写入访问来：

- 更改安全可配置事件的设置。
- 更改安全输入事件的屏蔽。
- 清除安全输入事件的挂起状态。

请注意，EXTI 不会区分来自各个 CPU 的安全访问。任何 CPU 安全访问都可以修改安全寄存器位。

表 121. 寄存器安全性概述

寄存器名称	访问类型	安全性(1)
EXTI_RTSRn	RW	非安全，安全性可在 EXTI_TZENRn 中逐位使能
EXTI_FTSRn	RW	
EXTI_SWIERn	RW	
EXTI_RPRn	RW	
EXTI_FPRn	RW	
EXTI_TZENRn	RW	始终安全
EXTI_EXTICRx	RW	非安全，安全性可在 EXTI_TZENRn 中逐字段使能
EXTI_IMRn	RW	非安全，安全性可在 EXTI_TZENRn 中逐位使能
EXTI_EMRn	RW	
EXTI_HWCFGRn	R	非安全
EXTI_VER	R	
EXTI_ID	R	
EXTI_SID	R	

1. 安全性通过单独的输入事件使能。EXTI_TZENRn 寄存器。

为输入事件使能 TZ 安全性后，相关输入事件配置和控制位只能通过安全访问进行修改和读取；非安全写入访问会被丢弃，且非安全读取访问会返回 0。

输入事件为非安全事件时，TZ 安全性将被禁止。相关输入事件配置和控制位可通过安全访问和非安全访问进行修改和读取。

20.6 EXTI 寄存器

表 122 中对 EXTI 寄存器映射进行了详细划分。

表 122. EXTI 寄存器映射的各个部分

地址	说明
0x000 - 0x01C	通用可配置事件 [31:0] 配置
0x020 - 0x03C	通用可配置事件 [63:32] 配置
0x040 - 0x05C	通用可配置事件 [95:64] 配置
0x060 - 0x06C	EXTI IO 端口复用器选择
0x080 - 0x0BC	CPU 输入事件配置

所有寄存器均可支持按字（32 位）、半字（16 位）或字节（8 位）访问。

20.6.1 EXTI 上升沿触发选择寄存器 (EXTI_RTSR1)

EXTI rising trigger selection register

偏移地址：0x000

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RT16
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:0 **RT[16:0]**：可配置事件输入 x 的上升沿触发事件配置位 (Rising trigger event configuration bit of configurable event input x)（x = 16 到 0）(1)。

TZENx 处于禁止状态时，RTx 可通过非安全和安全访问进行访问。

TZENx 处于使能状态时，RTx 只能通过安全访问进行访问。对 RTx 执行的非安全写入会被丢弃，而非安全读取会返回 0。

0：禁止输入线上升沿触发（事件和中断）

1：使能输入线上升沿触发（事件和中断）

1. 可配置事件输入为边沿触发，在这些输入上不能出现毛刺信号。如果在对寄存器执行写操作时可配置事件输入上出现上升沿，则相关挂起位不会被置 1。在同一可配置事件输入上，可同时设置上升沿和下降沿触发。在这种情况下，两种边沿均会生成触发信号。

20.6.2 EXTI 下降沿触发选择寄存器 (EXTI_FTSR1)

EXTI falling trigger selection register

偏移地址：0x004

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FT16
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FT15	FT14	FT13	FT12	FT11	FT10	FT9	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:0 **FT[16:0]**：可配置事件输入 x 的下降沿触发事件配置位 (Falling trigger event configuration bit of configurable event input x)（x = 16 到 0）(1)。

TZENx 处于禁止状态时，FTx 可通过非安全和安全访问进行访问。

TZENx 处于使能状态时，FTx 只能通过安全访问进行访问。对 FTx 执行的非安全写入会被丢弃，而非安全读取会返回 0。

0：禁止输入线下降沿触发（事件和中断）
1：使能输入线下降沿触发（事件和中断）

1. 可配置事件输入为边沿触发，在这些输入上不能出现毛刺信号。如果在对寄存器执行写操作时可配置事件输入上出现下降沿，则相关挂起位不会被置 1。在同一可配置事件输入上，可同时设置上升沿和下降沿触发。在这种情况下，两种边沿均会生成触发信号。

20.6.3 EXTI 软件中断事件寄存器 (EXTI_SWIER1)

EXTI software interrupt event register

偏移地址：0x008

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWI16
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWI15	SWI14	SWI13	SWI12	SWI11	SWI10	SWI9	SWI8	SWI7	SWI6	SWI5	SWI4	SWI3	SWI2	SWI1	SWI0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:0 **SWI[16:0]**：事件 x 的软件中断 (Software interrupt on event x)（x = 16 到 0）

TZENx 处于禁止状态时，SWIx 可通过非安全和安全访问进行访问。

TZENx 处于使能状态时，SWIx 只能通过安全访问进行访问。对 SWx 执行的非安全写入会被丢弃。

软件中断的生成与 EXTI_RTSR 和 EXTI_FTSR 中的设置无关。被读取时将始终返回 0。

0：写入 0 无影响。

1：向此位写入 1 会触发事件 x 上的上升沿事件。

此位由硬件自动清零。

20.6.4 EXTI 上升沿挂起寄存器 (EXTI_RPR1)

EXTI rising edge pending register

偏移地址：0x00C

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RPIF16
															rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPIF15	RPIF14	RPIF13	RPIF12	RPIF11	RPIF10	RPIF9	RPIF8	RPIF7	RPIF6	RPIF5	RPIF4	RPIF3	RPIF2	RPIF1	RPIF0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:17 保留，必须保持复位值。

位 16:0 **RPIF[16:0]**：可配置事件输入 x（x = 16 到 0）上升沿挂起位。

TZENx 处于禁止状态时，RPIFx 可通过非安全和安全访问进行访问。

TZENx 处于使能状态时，RPIFx 只能通过安全访问进行访问。对 RPIFx 执行的非安全写入会被丢弃，而非安全读取会返回 0。

0：未发生上升沿触发请求

1：发生了上升沿触发请求

当在可配置事件线上发生了上升沿事件或 EXTI_SWIER 软件触发时，此位将置 1。向此位写入 1 可将其清零。

20.6.5 EXTI 下降沿挂起寄存器 (EXTI_FPR1)

EXTI falling edge pending register

偏移地址：0x010

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FPIF16
															rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FPIF15	FPIF14	FPIF13	FPIF12	FPIF11	FPIF10	FPIF9	FPIF8	FPIF7	FPIF6	FPIF5	FPIF4	FPIF3	FPIF2	FPIF1	FPIF0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:17 保留，必须保持复位值。

位 16:0 **FPIF[16:0]**：可配置事件输入 x（x = 16 到 0）下降沿挂起位。
TZENx 处于禁止状态时，FPIFx 可通过非安全和安全访问进行访问。
TZENx 处于使能状态时，FPIFx 只能通过安全访问进行访问。对 FPIFx 执行的非安全写入会被丢弃，而非安全读取会返回 0。
0：未发生下降沿触发请求
1：发生了上升沿触发请求
当可配置事件线上发生了下降沿事件时，此位被置 1。向此位写入 1 可将其清零。

20.6.6 EXTI TrustZone 使能寄存器 (EXTI_TZENR1)

EXTI TrustZone enable register

偏移地址：0x014

复位值：0x0000 0000

该寄存器用于确保 TrustZone 写入访问的安全性；非安全写入访问将生成总线错误。非安全读取会返回寄存器数据。

仅包含支持 TrustZone 的输入事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TZEN 27	TZEN 26	TZEN 25	TZEN 24	TZEN 23	Res.	Res.	Res.	TZEN 19	TZEN 18	TZEN 17	Res.
				rw	rw	rw	rw	rw				rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TZEN 15	TZEN 14	TZEN 13	TZEN 12	TZEN 11	TZEN 10	TZEN9	TZEN8	TZEN7	TZEN6	TZEN5	TZEN4	TZEN3	TZEN2	TZEN1	TZEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:23 **TZEN[27:23]**：事件输入 x 上的 TrustZone 安全性使能 (TrustZone security enable on event input x)（x = 27 到 23）
0：事件 TrustZone 被禁止（非安全）
1：事件 TrustZone 被使能（安全）

位 22:20 保留，必须保持复位值。

位 19:17 **TZEN[19:17]**：事件输入 x 上的 TrustZone 安全性使能 (TrustZone security enable on event input x)（x = 19 到 17）
0：事件 TrustZone 被禁止（非安全）
1：事件 TrustZone 被使能（安全）

位 16 保留，必须保持复位值。

位 15:0 **TZEN[15:0]**：事件输入 x 上的 TrustZone 安全性使能 (TrustZone security enable on event input x)（x = 15 到 0）
0：事件 TrustZone 被禁止（非安全）
1：事件 TrustZone 被使能（安全）

20.6.7 EXTI 上升沿触发选择寄存器 (EXTI_RTSR2)

EXTI rising trigger selection register

偏移地址：0x020

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

20.6.8 EXTI 下降沿触发选择寄存器 (EXTI_FTSR2)

EXTI falling trigger selection register

偏移地址：0x024

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

20.6.9 EXTI 软件中断事件寄存器 (EXTI_SWIER2)

EXTI software interrupt event register

偏移地址：0x028

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

20.6.10 EXTI 上升沿挂起寄存器 (EXTI_RPR2)

EXTI rising edge pending register

偏移地址：0x02C

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

20.6.11 EXTI 下降沿挂起寄存器 (EXTI_FPR2)

EXTI falling edge pending register

偏移地址：0x030

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

20.6.12 EXTI TrustZone 使能寄存器 (EXTI_TZENR2)

EXTI TrustZone enable register

偏移地址：0x034

复位值：0x0000 0000

该寄存器用于确保 TrustZone 写入访问的安全性；非安全写入访问将生成总线错误。非安全读取会返回寄存器数据。

仅包含支持 TrustZone 的输入事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	TZEN 60	TZEN 59	TZEN 58	TZEN 57	TZEN 56	TZEN 55	Res.	Res.	Res.	Res.	TZEN 50	Res.	TZEN 48
			rw	rw	rw	rw	rw	rw					rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	TZEN 45	TZEN 44	TZEN 43	TZEN 42	Res.	TZEN 40	TZEN 39	Res.	Res.	Res.	Res.	Res.	Res.	Res.
		rw	rw	rw	rw		rw	rw							

位 31:29 保留，必须保持复位值。

位 28:23 **TZEN[60:55]**：事件输入 x 上的 TrustZone 安全性使能 (TrustZone security enable on Event input x)（x = 60 到 55）

0：事件 TrustZone 被禁止（非安全）
1：事件 TrustZone 被使能（安全）

位 22:19 保留，必须保持复位值。

位 18 **TZEN50**：事件输入 50 上的 TrustZone 安全性使能 (TrustZone security enable on Event input 50)

0：事件 TrustZone 被禁止（非安全）
1：事件 TrustZone 被使能（安全）

位 17 保留，必须保持复位值。

位 16 **TZEN48**：事件输入 48 上的 TrustZone 安全性使能 (TrustZone security enable on Event input 48)

0：事件 TrustZone 被禁止（非安全）
1：事件 TrustZone 被使能（安全）

位 15:14 保留，必须保持复位值。

位 13:10 **TZEN[45:42]**：事件输入 x 上的 TrustZone 安全性使能 (TrustZone security enable on Event input x)（x = 45 到 42）

0：事件 TrustZone 被禁止（非安全）
1：事件 TrustZone 被使能（安全）

位 9 保留，必须保持复位值。

位 8:7 **TZEN[40:39]**：事件输入 x 上的 TrustZone 安全性使能 (TrustZone security enable on Event input x)（x = 40 到 39）

0：事件 TrustZone 被禁止（非安全）
1：事件 TrustZone 被使能（安全）

位 6:0 保留，必须保持复位值。

20.6.13 EXTI 上升沿触发选择寄存器 (EXTI_RTSR3)

EXTI rising trigger selection register

偏移地址：0x040

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

20.6.14 EXTI 下降沿触发选择寄存器 (EXTI_FTSR3)

EXTI falling trigger selection register

偏移地址：0x044

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

20.6.15 EXTI 软件中断事件寄存器 (EXTI_SWIER3)

EXTI software interrupt event register

偏移地址：0x048

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

20.6.16 EXTI 上升沿挂起寄存器 (EXTI_RPR3)

EXTI rising edge pending register

偏移地址：0x04C

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

20.6.17 EXTI 下降沿挂起寄存器 (EXTI_FPR3)

EXTI falling edge pending register

偏移地址：0x050

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

20.6.18 EXTI TrustZone 使能寄存器 (EXTI_TZENR3)

EXTI TrustZone enable register

偏移地址：0x054

复位值：0x0000 0000

该寄存器用于确保 TrustZone 写入访问的安全性；非安全写入访问将生成总线错误。非安全读取会返回寄存器数据。

仅包含支持 TrustZone 的输入事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TZEN 71	TZEN 70	TZEN 69	TZEN 68	Res.	Res.	Res.	Res.
								rw	rw	rw	rw				

位 31:8 保留，必须保持复位值。

位 7:4 **TZEN[71:68]**：事件输入 x 上的 TrustZone 安全性使能 (TrustZone security enable on Event input x)（x = 71 到 68）

0：事件 TrustZone 被禁止（非安全）

1：事件 TrustZone 被使能（安全）

位 3:0 保留，必须保持复位值。

20.6.19 EXTI 外部中断选择寄存器 1 (EXTI_EXTICR1)

EXTI external interrupt selection register 1

偏移地址：0x060

复位值：0x0000 0000

EXTIm 字段包含的位数与 NBIOPORT[7:0] 配置的位数完全一致。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI3[7:0]								EXTI2[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI1[7:0]								EXTI0[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **EXTI3[7:0]**：EXTI3 GPIO 端口选择 (EXTI3 GPIO port selection)

这些位通过软件写入，用于选择 EXTI3 外部中断的源输入。

TZEN3 处于禁止状态时，EXTI3 可通过非安全和安全访问进行访问。

TZEN3 处于使能状态时，EXTI3 只能通过安全访问进行访问。非安全写入会被丢弃，而非安全读取会返回 0。

0x00：PA[3] 引脚
0x01：PB[3] 引脚
0x02：PC[3] 引脚
0x03：PD[3] 引脚
0x04：PE[3] 引脚
0x05：PF[3] 引脚
0x06：PG[3] 引脚
0x07：PH[3] 引脚
0x08：PI[3] 引脚
其他值：保留

位 23:16 **EXTI2[7:0]**：EXTI2 GPIO 端口选择 (EXTI2 GPIO port selection)

这些位通过软件写入，用于选择 EXTI2 外部中断的源输入。

TZEN2 处于禁止状态时，EXTI2 可通过非安全和安全访问进行访问。

TZEN2 处于使能状态时，EXTI2 只能通过安全访问进行访问。非安全写入会被丢弃，而非安全读取会返回 0。

0x00：PA[2] 引脚
0x01：PB[2] 引脚
0x02：PC[2] 引脚
0x03：PD[2] 引脚
0x04：PE[2] 引脚
0x05：PF[2] 引脚
0x06：PG[2] 引脚
0x07：PH[2] 引脚
0x08：PI[2] 引脚
其他值：保留

位 15:8 **EXTI1[7:0]**：EXTI1 GPIO 端口选择 (EXTI1 GPIO port selection)

这些位通过软件写入，用于选择 EXTI1 外部中断的源输入。

TZEN1 处于禁止状态时，EXTI1 可通过非安全和安全访问进行访问。

TZEN1 处于使能状态时，EXTI1 只能通过安全访问进行访问。非安全写入会被丢弃，而非安全读取会返回 0。

0x00：PA[1] 引脚
0x01：PB[1] 引脚
0x02：PC[1] 引脚
0x03：PD[1] 引脚
0x04：PE[1] 引脚
0x05：PF[1] 引脚
0x06：PG[1] 引脚
0x07：PH[1] 引脚
0x08：PI[1] 引脚
其他值：保留

位 7:0 **EXTI0[7:0]**：EXTI0 GPIO 端口选择 (EXTI0 GPIO port selection)

这些位通过软件写入，用于选择 EXTI0 外部中断的源输入。

TZEN0 处于禁止状态时，EXTI0 可通过非安全和安全访问进行访问。

TZEN0 处于使能状态时，EXTI0 只能通过安全访问进行访问。非安全写入会被丢弃，而非安全读取会返回 0。

0x00：PA[0] 引脚
0x01：PB[0] 引脚
0x02：PC[0] 引脚
0x03：PD[0] 引脚
0x04：PE[0] 引脚
0x05：PF[0] 引脚
0x06：PG[0] 引脚
0x07：PH[0] 引脚
0x08：PI[0] 引脚
其他值：保留

20.6.20 EXTI 外部中断选择寄存器 2 (EXTI_EXTICR2)

EXTI external interrupt selection register 2

偏移地址：0x064

复位值：0x0000 0000

EXTIm 字段包含的位数与 NBIOPORT[7:0] 配置的位数完全一致。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI7[7:0]								EXTI6[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI5[7:0]								EXTI4[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **EXTI7[7:0]**：EXTI7 GPIO 端口选择 (EXTI7 GPIO port selection)

这些位通过软件写入，用于选择 EXTI7 外部中断的源输入。

TZEN7 处于禁止状态时，EXTI7 可通过非安全和安全访问进行访问。

TZEN7 处于使能状态时，EXTI7 只能通过安全访问进行访问。非安全写入会被丢弃，而非安全读取会返回 0。

0x00：PA[7] 引脚
0x01：PB[7] 引脚
0x02：PC[7] 引脚
0x03：PD[7] 引脚
0x04：PE[7] 引脚
0x05：PF[7] 引脚
0x06：PG[7] 引脚
0x07：PH[7] 引脚
0x08：PI[7] 引脚
其他值：保留

位 23:16 **EXTI6[7:0]**：EXTI6 GPIO 端口选择 (EXTI6 GPIO port selection)

这些位通过软件写入，用于选择 EXTI6 外部中断的源输入。

TZEN6 处于禁止状态时，EXTI6 可通过非安全和安全访问进行访问。

TZEN6 处于使能状态时，EXTI6 只能通过安全访问进行访问。非安全写入会被丢弃，而非安全读取会返回 0。

0x00：PA[6] 引脚
0x01：PB[6] 引脚
0x02：PC[6] 引脚
0x03：PD[6] 引脚
0x04：PE[6] 引脚
0x05：PF[6] 引脚
0x06：PG[6] 引脚
0x07：PH[6] 引脚
0x08：PI[6] 引脚
其他值：保留

位 15:8 **EXTI5[7:0]**：EXTI5 GPIO 端口选择 (EXTI5 GPIO port selection)

这些位通过软件写入，用于选择 EXTI5 外部中断的源输入。

TZEN5 处于禁止状态时，EXTI5 可通过非安全和安全访问进行访问。

TZEN5 处于使能状态时，EXTI5 只能通过安全访问进行访问。非安全写入会被丢弃，而非安全读取会返回 0。

0x00：PA[5] 引脚
0x01：PB[5] 引脚
0x02：PC[5] 引脚
0x03：PD[5] 引脚
0x04：PE[5] 引脚
0x05：PF[5] 引脚
0x06：PG[5] 引脚
0x07：PH[5] 引脚
0x08：PI[5] 引脚
其他值：保留

位 7:0 **EXTI4[7:0]**：EXTI4 GPIO 端口选择 (EXTI4 GPIO port selection)

这些位通过软件写入，用于选择 EXTI4 外部中断的源输入。

TZEN4 处于禁止状态时，EXTI4 可通过非安全和安全访问进行访问。

TZEN4 处于使能状态时，EXTI4 只能通过安全访问进行访问。非安全写入会被丢弃，而非安全读取会返回 0。

0x00：PA[4] 引脚
0x01：PB[4] 引脚
0x02：PC[4] 引脚
0x03：PD[4] 引脚
0x04：PE[4] 引脚
0x05：PF[4] 引脚
0x06：PG[4] 引脚
0x07：PH[4] 引脚
0x08：PI[4] 引脚
其他值：保留

20.6.21 EXTI 外部中断选择寄存器 3 (EXTI_EXTICR3)

EXTI external interrupt selection register 3

偏移地址：0x068

复位值：0x0000 0000

EXTIm 字段包含的位数与 NBIOPORT[7:0] 配置的位数完全一致。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI11[7:0]								EXTI10[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI9[7:0]								EXTI8[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **EXTI11[7:0]**：EXTI11 GPIO 端口选择 (EXTI11 GPIO port selection)

这些位通过软件写入，用于选择 EXTI11 外部中断的源输入。

TZEN11 处于禁止状态时，EXTI11 可通过非安全和安全访问进行访问。

TZEN11 处于使能状态时，EXTI11 只能通过安全访问进行访问。非安全写入会被丢弃，而非安全读取会返回 0。

0x00：PA[11] 引脚
0x01：PB[11] 引脚
0x02：PC[11] 引脚
0x03：PD[11] 引脚
0x04：PE[11] 引脚
0x05：PF[11] 引脚
0x06：PG[11] 引脚
0x07：PH[11] 引脚
其他值：保留

位 23:16 **EXTI10[7:0]**：EXTI10 GPIO 端口选择 (EXTI10 GPIO port selection)

这些位通过软件写入，用于选择 EXTI10 外部中断的源输入。

TZEN10 处于禁止状态时，EXTI10 可通过非安全和安全访问进行访问。

TZEN10 处于使能状态时，EXTI10 只能通过安全访问进行访问。非安全写入会被丢弃，而非安全读取会返回 0。

0x00：PA[10] 引脚
0x01：PB[10] 引脚
0x02：PC[10] 引脚
0x03：PD[10] 引脚
0x04：PE[10] 引脚
0x05：PF[10] 引脚
0x06：PG[10] 引脚
0x07：PH[10] 引脚
其他值：保留

位 15:8 **EXTI9[7:0]**：EXTI9 GPIO 端口选择 (EXTI9 GPIO port selection)

这些位通过软件写入，用于选择 EXTI9 外部中断的源输入。

TZEN9 处于禁止状态时，EXTI9 可通过非安全和安全访问进行访问。

TZEN9 处于使能状态时，EXTI9 只能通过安全访问进行访问。非安全写入会被丢弃，而非安全读取会返回 0。

0x00：PA[9] 引脚
0x01：PB[9] 引脚
0x02：PC[9] 引脚
0x03：PD[9] 引脚
0x04：PE[9] 引脚
0x05：PF[9] 引脚
0x06：PG[9] 引脚
0x07：PH[9] 引脚
其他值：保留

位 7:0 **EXTI8[7:0]**：EXTI8 GPIO 端口选择 (EXTI8 GPIO port selection)

这些位通过软件写入，用于选择 EXTI8 外部中断的源输入。

TZEN8 处于禁止状态时，EXTI8 可通过非安全和安全访问进行访问。

TZEN8 处于使能状态时，EXTI8 只能通过安全访问进行访问。非安全写入会被丢弃，而非安全读取会返回 0。

0x00：PA[8] 引脚
0x01：PB[8] 引脚
0x02：PC[8] 引脚
0x03：PD[8] 引脚
0x04：PE[8] 引脚
0x05：PF[8] 引脚
0x06：PG[8] 引脚
0x07：PH[8] 引脚
其他值：保留

20.6.22 EXTI 外部中断选择寄存器 4 (EXTI_EXTICR4)

EXTI external interrupt selection register 4

偏移地址：0x06C

复位值：0x0000 0000

EXTIm 字段包含的位数与 NBIOPORT[7:0] 配置的位数完全一致。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI15[7:0]								EXTI14[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI13[7:0]								EXTI12[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **EXTI15[7:0]**：EXTI15 GPIO 端口选择 (EXTI15 GPIO port selection)

这些位通过软件写入，用于选择 EXTI15 外部中断的源输入。

TZEN15 处于禁止状态时，EXTI15 可通过非安全和安全访问进行访问。

TZEN15 处于使能状态时，EXTI15 只能通过安全访问进行访问。非安全写入会被丢弃，而非安全读取会返回 0。

0x00：PA[15] 引脚
0x01：PB[15] 引脚
0x02：PC[15] 引脚
0x03：PD[15] 引脚
0x04：PE[15] 引脚
0x05：PF[15] 引脚
0x06：PG[15] 引脚
其他值：保留

位 23:16 **EXTI14[7:0]**：EXTI14 GPIO 端口选择 (EXTI14 GPIO port selection)

这些位通过软件写入，用于选择 EXTI14 外部中断的源输入。

TZEN14 处于禁止状态时，EXTI14 可通过非安全和安全访问进行访问。

TZEN14 处于使能状态时，EXTI14 只能通过安全访问进行访问。非安全写入会被丢弃，而非安全读取会返回 0。

0x00：PA[14] 引脚
0x01：PB[14] 引脚
0x02：PC[14] 引脚
0x03：PD[14] 引脚
0x04：PE[14] 引脚
0x05：PF[14] 引脚
0x06：PG[14] 引脚
0x07：PH[14] 引脚
其他值：保留

位 15:8 **EXTI13[7:0]**：EXTI13 GPIO 端口选择 (EXTI13 GPIO port selection)

这些位通过软件写入，用于选择 EXTI13 外部中断的源输入。

TZEN13 处于禁止状态时，EXTI13 可通过非安全和安全访问进行访问。

TZEN13 处于使能状态时，EXTI13 只能通过安全访问进行访问。非安全写入会被丢弃，而非安全读取会返回 0。

0x00：PA[13] 引脚
0x01：PB[13] 引脚
0x02：PC[13] 引脚
0x03：PD[13] 引脚
0x04：PE[13] 引脚
0x05：PF[13] 引脚
0x06：PG[13] 引脚
0x07：PH[13] 引脚
其他值：保留

位 7:0 **EXTI12[7:0]**：EXTI12 GPIO 端口选择 (EXTI12 GPIO port selection)

这些位通过软件写入，用于选择 EXTI12 外部中断的源输入。

TZEN12 处于禁止状态时，EXTI12 可通过非安全和安全访问进行访问。

TZEN12 处于使能状态时，EXTI12 只能通过安全访问进行访问。非安全写入会被丢弃，而非安全读取会返回 0。

0x00：PA[12] 引脚
0x01：PB[12] 引脚
0x02：PC[12] 引脚
0x03：PD[12] 引脚
0x04：PE[12] 引脚
0x05：PF[12] 引脚
0x06：PG[12] 引脚
0x07：PH[12] 引脚
其他值：保留

20.6.23 EXTI CPU 唤醒中断屏蔽寄存器 (EXTI_IMR1)

EXTI CPU wakeup with interrupt mask register

偏移地址：0x080

复位值：0xFFFE 0000

包含可配置事件和直接事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IM31	IM30	IM29	IM28	IM27	IM26	IM25	IM24	IM23	IM22	IM21	IM20	IM19	IM18	IM17	IM16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IM[31:0]**：事件输入 x 上的 CPU 唤醒中断屏蔽 (x = 31:0)(1)(2)。

TZENx 处于禁止状态时，IMx 可通过非安全和安全访问进行访问。

TZENx 处于使能状态时，IMx 只能通过安全访问进行访问。对 IMx 执行的非安全写入会被丢弃，而非安全读取会返回 0。

0：屏蔽来自线 x 的唤醒中断请求
1：取消屏蔽来自线 x 的唤醒中断请求

1. 可配置事件输入的复位值置“0”，以在默认情况下禁止中断。
2. 直接事件输入的复位值置“1”，以在默认情况下使能中断。

复位值：0xFFFE 0000

20.6.24 EXTI CPU 唤醒事件屏蔽寄存器 (EXTI_EMR1)

EXTI CPU wakeup with event mask register

偏移地址：0x084

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EM19	EM18	EM17	Res.
												rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EM15	EM14	EM13	EM12	EM11	EM10	EM9	EM8	EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:20 保留，必须保持复位值。

位 19:17 **EM[19:17]**：事件输入 x 上的 CPU 唤醒事件生成屏蔽 (CPU wakeup with event generation mask on event input x)（x = 19 到 17）

TZENx 处于禁止状态时，EMx 可通过非安全和安全访问进行访问。

TZENx 处于使能状态时，EMx 只能通过安全访问进行访问。对 EMx 执行的非安全写入会被丢弃，而非安全读取会返回 0。

0：屏蔽线 x 上的唤醒事件生成
1：取消屏蔽线 x 上的唤醒事件生成

位 16 保留，必须保持复位值。

位 15:0 **EM[15:0]**：事件输入 x 上的 CPU 唤醒事件生成屏蔽 (CPU wakeup with event generation mask on event input x)（x = 15 到 0）

TZENx 处于禁止状态时，EMx 可通过非安全和安全访问进行访问。

TZENx 处于使能状态时，EMx 只能通过安全访问进行访问。对 EMx 执行的非安全写入会被丢弃，而非安全读取会返回 0。

0：屏蔽线 x 上的唤醒事件生成
1：取消屏蔽线 x 上的唤醒事件生成

20.6.25 EXTI CPU 唤醒中断屏蔽寄存器 (EXTI_IMR2)

EXTI CPU wakeup with interrupt mask register

偏移地址：0x090

复位值：0xFFFF FFFF

包含可配置事件和直接事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IM63	IM62	IM61	IM60	IM59	IM58	IM57	IM56	IM55	IM54	IM53	IM52	IM51	IM50	IM49	IM48
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IM47	IM46	IM45	IM44	IM43	IM42	IM41	IM40	IM39	IM38	IM37	IM36	IM35	IM34	IM33	IM32
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IM[63:32]**：事件输入 x 上的 CPU 唤醒中断屏蔽 (CPU wakeup with interrupt mask on event input x)（x = 63 到 32）(1)(2)

TZENx 处于禁止状态时，IMx 可通过非安全和安全访问进行访问。

TZENx 处于使能状态时，IMx 只能通过安全访问进行访问。对 IMx 执行的非安全写入会被丢弃，而非安全读取会返回 0。

0：屏蔽来自线 x 的唤醒中断请求

1：取消屏蔽来自线 x 的唤醒中断请求

1. 可配置事件输入的复位值置“0”，以在默认情况下禁止中断。
2. 直接事件输入的复位值置“1”，以在默认情况下使能中断。

复位值：0xFFFF FFFF

20.6.26 EXTI CPU 唤醒事件屏蔽寄存器 (EXTI_EMR2)

EXTI CPU wakeup with event mask register

偏移地址：0x094

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

20.6.27 EXTI CPU 唤醒中断屏蔽寄存器 (EXTI_IMR3)

EXTI CPU wakeup with interrupt mask register

偏移地址：0x0A0

复位值：0x0000 0FFF

包含可配置事件和直接事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	IM75	IM74	IM73	IM72	IM71	IM70	IM69	IM68	IM67	IM66	IM65	IM64
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:0 **IM[75:64]**：事件输入 x 上的 CPU 唤醒中断屏蔽 (CPU wakeup with interrupt mask on event input x)（x = 75 到 64）(1)(2)

TZENx 处于禁止状态时，IMx 可通过非安全和安全访问进行访问。

TZENx 处于使能状态时，IMx 只能通过安全访问进行访问。对 IMx 执行的非安全写入会被丢弃，而非安全读取会返回 0。

0：屏蔽来自线 x 的唤醒中断请求

1：取消屏蔽来自线 x 的唤醒中断请求

1. 可配置事件输入的复位值置“0”，以在默认情况下禁止中断。
2. 直接事件输入的复位值置“1”，以在默认情况下使能中断。

20.6.28 EXTI CPU 唤醒事件屏蔽寄存器 (EXTI_EMR3)

EXTI CPU wakeup with event mask register

偏移地址：0x0A4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

20.6.29 EXTI 硬件配置寄存器 x (EXTI_HWCFGRx)

EXTI hardware configuration register x

偏移地址：0x3C8 - 0x4 * (x-11)，（x = 11 到 13）

复位值：0x0F8E FFFF (x = 11)

复位值：0x1F85 3D80 (x = 12)

复位值：0x0000 00F0 (x = 13)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TZ[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TZ[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TZ[31:0]**：事件 TrustZone 安全能力 HW 配置。

0：非安全事件。

1：支持 TrustZone 安全的事件。

20.6.30 EXTI 硬件配置寄存器 x (EXTI_HWCFGRx)

EXTI hardware configuration register x

偏移地址：0x3D4 - 0x4 * (x - 8)，（x = 8 到 10）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

20.6.31 EXTI 硬件配置寄存器 x (EXTI_HWCFGRx)

EXTI hardware configuration register x

偏移地址：0x3E0 - 0x4 * (x - 5)，（x = 5 到 7）

复位值：0x000E FFFF (x = 5)

复位值：0x0000 0000 (x = 6)

复位值：0x0000 0000 (x = 7)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CPUEVENT[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPUEVENT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CPUEVENT[31:0]**：CPU 事件生成 HW 配置 (HW configuration CPU event generation)

0：未生成 CPU 事件

1：已生成 CPU 事件

20.6.32 EXTI 硬件配置寄存器 x (EXTI_HWCFGRx)

EXTI hardware configuration register x

偏移地址：0x3EC - 0x4 * (x - 2)，（x = 2 到 4）

复位值：0x0001 FFFF (x = 2)

复位值：0x0000 0000 (x = 3)

复位值：0x0000 0000 (x = 4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EVENT_TRG[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT_TRG[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **EVENT_TRG[31:0]**：事件触发类型 HW 配置 (HW configuration event trigger type)

0：直接事件

1：可配置事件

20.6.33 EXTI 硬件配置寄存器 1 (EXTI_HWCFGR1)

EXTI hardware configuration register 1

偏移地址：0x3F0

复位值：0x000B 204B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBIOPORT[7:0]							
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPUEVTEN[3:0]				NBCPUS[3:0]				NBEVENTS[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 保留，必须保持复位值。

位 23:16 **NBIOPORT[7:0]**：EXTI 上 IO 端口数 (n+1) 的 HW 配置。

位 15:12 **CPUEVTEN[3:0]**：CPU 事件输出使能的 HW 配置。

位 11:8 **NBCPUS[3:0]**：CPU 数 (n+1) HW 配置

位 7:0 **NBEVENTS[7:0]**：事件数 (n+1) HW 配置

20.6.34 EXTI IP 版本寄存器 (EXTI_VERR)

EXTI IP version register

偏移地址：0x3F4

复位值：0x0000 0030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本号 (Major Revision number)

位 3:0 **MINREV[3:0]**：次版本号 (Minor Revision number)

20.6.35 EXTI 标识寄存器 (EXTI_IPIDR)

EXTI identification register

偏移地址：0x3F8

复位值：0x000E 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **IPID[31:0]**：IP 标识 (IP Identification)

20.6.36 EXTI 大小 ID 寄存器 (EXTI_SIDR)

EXTI size ID register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识 (Size identification)

20.6.37 EXTI 寄存器映射

下表列出了 EXTI 寄存器映射和复位值。

表 123. EXTI 寄存器映射和复位值

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**EXTI_RTSR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RT[16:0]																
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	**EXTI_FTSR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FT[16:0]																
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x008	**EXTI_SWIER1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWI[16:0]																
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	**EXTI_RPR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RPIF[16:0]																
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	**EXTI_FPR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FPIF[16:0]																
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014	**EXTI_TZENR1**	Res.	Res.	Res.	Res.	TZEN[27:23]					Res.	Res.	Res.	TZEN [19:17]			Res.	TZEN[15:0]															
	Reset value					0	0	0	0	0				0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018-0x01C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x020	**EXTI_RTSR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x024	**EXTI_FTSR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x028	**EXTI_SWIER2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x02C	**EXTI_RPR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x030	**EXTI_FPR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x034	**EXTI_TZENR2**	Res.	Res.	Res.	TZEN[60:55]						Res.	Res.	Res.	Res.	TZEN50	Res.	TZEN48	Res.	Res.	TZEN[45:42]				Res.	TZEN [40:39]		Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value				0	0	0	0	0	0					0		0			0	0	0	0		0	0							
0x038-0x03C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 123. EXTI 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x040	**EXTI_RTSR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x044	**EXTI_FTSR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x048	**EXTI_SWIER3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x04C	**EXTI_RPR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x050	**EXTI_FPR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x054	**EXTI_TZENR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TZEN[71:68]				Res.	Res.	Res.	Res.
	Reset value																									0	0	0	0				
0x058-0x05C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x060	**EXTI_EXTICR1**	EXTI3[7:0]								EXTI2[7:0]								EXTI1[7:0]								EXTI0[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x064	**EXTI_EXTICR2**	EXTI7[7:0]								EXTI6[7:0]								EXTI5[7:0]								EXTI4[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x068	**EXTI_EXTICR3**	EXTI11[7:0]								EXTI10[7:0]								EXTI9[7:0]								EXTI8[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x06C	**EXTI_EXTICR4**	EXTI15[7:0]								EXTI14[7:0]								EXTI13[7:0]								EXTI2[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x070-0x07C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x080	**EXTI_IMR1**	IM[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	**EXTI_EMR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EM[19:17]			Res.	EM[15:0]															
	Reset value													0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x088-0x08C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x090	**EXTI_IMR2**	IM[63:32]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

表 123. EXTI 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x094	**EXTI_EMR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x098-0x09C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0A0	**EXTI_IMR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IM[75:64]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x0A4	**EXTI_EMR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x0A8 - 0x3BC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x3C0	**EXTI_ HWCFGR13**	TZ[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
0x3C4	**EXTI_ HWCFGR12**	TZ[31:0]																															
	Reset value	0	0	0	1	1	1	1	1	1	0	0	0	0	1	0	1	0	0	1	1	1	1	0	1	1	0	0	0	0	0	0	0
0x3C8	**EXTI_ HWCFGR11**	TZ[31:0]																															
	Reset value	0	0	0	0	1	1	1	1	1	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x3D4 - 0x4 * (x - 8), (x = 8 to 10)	**EXTI_ HWCFGRx**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x3D8	**EXTI_ HWCFGR7**	CPUEVENT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3DC	**EXTI_ HWCFGR6**	CPUEVENT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3E0	**EXTI_ HWCFGR5**	CPUEVENT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x3E4	**EXTI_ HWCFGR4**	EVENT_TRG[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3E8	**EXTI_ HWCFGR3**	EVENT_TRG[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3EC	**EXTI_ HWCFGR2**	EVENT_TRG[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x3F0	**EXTI_HWCFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBIOPORT[7:0]								CPUEVTEN [3:0]				NBCPUS[3:0]				NBEVENTS[7:0]							
	Reset value									0	0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	1	1

表 123. EXTI 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x3F4	**EXTI_VERR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	0	1	1	0	0	0	0
0x3F8	**EXTI_IPIDR**	IPID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x3FC	**EXTI_SIDR**	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	0	1	1	1	0	1	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

21 循环冗余校验计算单元 (CRC)

21.1 简介

CRC（循环冗余校验）计算单元使用多项式发生器从一个 8 位、16 位或 32 位的数据字中产生 CRC 码。

在众多的应用中，基于 CRC 的技术还常用来验证数据传输或存储的完整性。根据功能安全标准的规定，这些技术提供了验证 Flash 完整性的方法。CRC 计算单元有助于在运行期间计算软件的签名，并将该签名与链接时生成并存储在指定存储单元的参考签名加以比较。

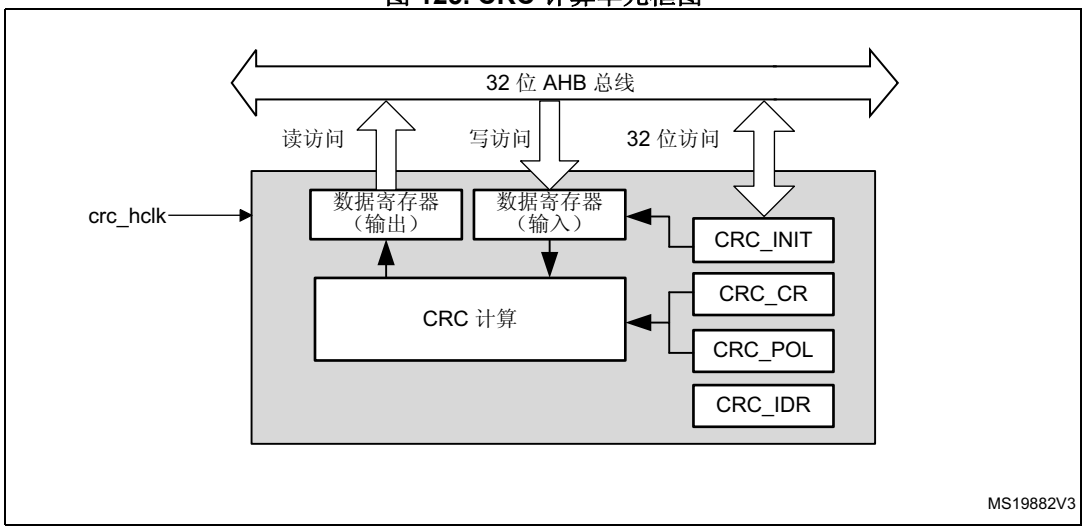
21.2 CRC 主要特性

- 使用 CRC-32（以太网）多项式：0x4C11DB7
  $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- 或者，使用位数可编程的（7 位、8 位、16 位和 32 位）的完全可编程多项式
- 处理 8 位、16 位、32 位数据大小
- 可编程 CRC 初始值
- 单输入/输出 32 位数据寄存器
- 输入缓冲器可避免计算期间发生总线阻塞
- 对于 32 位数据大小，CRC 计算在 4 个 AHB 时钟周期 (HCLK) 内完成
- 8 位通用寄存器（可用于临时存储）
- 输入输出数据位序可反转
- 仅支持按 32 位字的方式通过 AHB 从设备访问（例外情况：CRC_DR 寄存器可按字、右对齐半字和右对齐字节进行访问）

21.3 CRC 功能描述

21.3.1 CRC 框图

图 125. CRC 计算单元框图



21.3.2 CRC 内部信号

表 124. CRC 内部输入/输出信号

信号名称	信号类型	说明
crc_hclk	数字输入	AHB 时钟

21.3.3 CRC 运算

CRC 计算单元具有单个 32 位读/写数据寄存器 (CRC_DR)。它用于输入新数据（写入访问）和保存之前 CRC 计算的结果（读取访问）。

对数据寄存器的每个写入操作都会对之前的 CRC 值（存储在 CRC_DR 中）和新值再做一次 CRC 计算。CRC 计算针对整个 32 位数据字或逐个字节完成，具体取决于数据的写入格式。

CRC_DR 寄存器可按字、右对齐半字和右对齐字节进行访问。对于其他寄存器，只允许按 32 位进行访问。

计算时间取决于数据宽度：

- 32 位数据需要 4 个 AHB 时钟周期
- 16 位数据需要 2 个 AHB 时钟周期
- 8 位数据需要 1 个 AHB 时钟周期

输入缓冲区允许立即写入第二个数据，而无需等待由于先前的 CRC 计算而导致的任何等待状态。

可动态调整数据大小，从而能最大程度地减少给定字节数的写入访问次数。例如，对 5 个字节进行 CRC 计算时，可先写入字，然后写入字节。

为管理各种数据存放方式（大端/小端），输入数据的顺序可以反转。可对 8 位、16 位和 32 位数据执行反转操作，具体取决于 CRC_CR 寄存器中的 REV_IN[1:0] 位。

例如，0x1A2B3C4D 输入数据在 CRC 计算中用作：

- 0x58D43CB2，按字节执行位序反转
- 0xD458B23C，按半字执行位序反转
- 0xB23CD458，按全字执行位序反转

通过将 CRC_CR 寄存器中 REV_OUT 位置 1 也可以将输出数据反转。

该操作按位进行。例如，输出数据 0x11223344 将转换为 0x22CC4488。

使用 CRC_CR 寄存器中的 RESET 控制位可将 CRC 计算器初始化为可编程值（默认值为 0xFFFFFFFF）。

可使用 CRC_INIT 寄存器对 CRC 初始值进行编程。对 CRC_INIT 寄存器进行写入访问时会自动初始化 CRC_DR 寄存器。

CRC_IDR 寄存器可用于保存与 CRC 计算相关的临时值。它不受 CRC_CR 寄存器中的 RESET 位影响。

多项式可编程

多项式系数可完全通过 CRC_POL 寄存器进行编程，通过编程 CRC_CR 寄存器中的 POLYSIZE[1:0] 位可将多项式大小配置为 7 位、8 位、16 位或 32 位。不支持偶数多项式。

注： *偶数多项式的形式为 $X+X^2+..+X^n$，而奇数多项式的形式为 $1+X+X^2+..+X^n$。*

如果 CRC 数据小于 32 位，可从 CRC_DR 寄存器的最低有效位读取 CRC 的值。

为实现可靠的 CRC 计算，CRC 计算期间不能实时更改多项式的值或大小。因此，如果正在执行 CRC 计算，则在更改多项式前，应用程序必须先复位，或者先执行 CRC_DR 读取操作。

默认的多项式值为 CRC-32（以太网）多项式：0x4C11DB7。

21.4 CRC 寄存器

CRC_DR 寄存器可按字、右对齐半字和右对齐字节进行访问。对于其他寄存器，只允许按 32 位进行访问。

21.4.1 CRC 数据寄存器 (CRC_DR)

CRC data register

偏移地址：0x00

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DR[31:0]**：数据寄存器位

该寄存器用于向 CRC 计算器写入新数据。

读取寄存器时可读出之前的 CRC 计算结果。

如果数据大小小于 32 位，则最低有效位可用于写入/读取正确值。

21.4.2 CRC 独立数据寄存器 (CRC_IDR)

CRC independent data register

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IDR[31:0]**：通用的 32 位数据寄存器位

这些位可用作四个字节的临时存储单元。

该寄存器不受 CRC_CR 寄存器中 RESET 位产生的 CRC 复位影响

21.4.3 CRC 控制寄存器 (CRC_CR)

CRC control register

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REV_OUT	REV_IN[1:0]		POLYSIZE[1:0]		Res.	Res.	RESET
								rw	rw	rw	rw	rw			rs

位 31:8 保留，必须保持复位值。

位 7 **REV_OUT**：反转输出数据 (Reverse output data)

该位字段用于控制输出数据的位序反转。

0：位序不受影响

1：按位反转输出

位 6:5 **REV_IN[1:0]**：反转输入数据 (Reverse input data)

该位字段用于控制输入数据的位序反转。

00：位序不受影响

01：按字节执行位反转

10：按半字执行位反转

11：按字执行位反转

位 4:3 **POLYSIZE[1:0]**：多项式大小 (Polynomial size)

这些位用于控制多项式的大小。

00：32 位多项式

01：16 位多项式

10：8 位多项式

11：7 位多项式

位 2:1 保留，必须保持复位值。

位 0 **RESET**：RESET 位

该位由软件置 1，用于复位 CRC 计算单元并将数据寄存器设置为存储在 CRC_INIT 寄存器中的值。该位只能置 1，将由硬件自动进行清零

21.4.4 CRC 初始值 (CRC_INIT)

CRC initial value

偏移地址：0x10

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CRC_INIT[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC_INIT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CRC_INIT[31:0]**：可编程 CRC 初始值 (Programmable initial CRC value)
该寄存器用于写入 CRC 初始值。

21.4.5 CRC 多项式 (CRC_POL)

CRC polynomial

偏移地址：0x14

复位值：0x04C1 1DB7

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POL[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POL[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **POL[31:0]**：可编程多项式 (Programmable polynomial)
该寄存器用于写入要用于 CRC 计算的多项式系数。
如果多项式大小小于 32 位，则必须使用最低有效位编程正确值。

21.4.6 CRC 硬件配置寄存器 (CRC_HWCFGR)

CRC hardware configuration register

偏移地址：0x3F0

复位值：0x0000 00XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FULL_POLY[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **FULL_POLY[7:0]**：多项式类型 (Polynomial type)
00000000：固定多项式 0x4C11DB7
00000001：可编程多项式系数和大小（7、8、16、32）
其他值：保留，不得使用。

21.4.7 CRC 版本寄存器 (CRC_VERR)

CRC version register

偏移地址：0x3F4

复位值：0x0000 0023

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本 (Major revision)
这些位返回 CRC 主版本。

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)
这些位返回 CRC 次版本。

21.4.8 CRC 标识寄存器 (CRC_IPIDR)

CRC identification register

偏移地址：0x3F8

复位值：0x0017 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：外设标识符 (Peripheral identifier)

这些位返回 CRC 标识符：
ID[31:0] = 0x0017 0001

21.4.9 CRC 大小标识寄存器 (CRC_SIDR)

CRC size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识 (Size identification)

这些位返回分配给 CRC 寄存器的存储区域的大小。该存储区域的大小为 1 KB：
SID[31:0] = 0xA3C5 DD01

21.4.10 CRC 寄存器映射

表 125. CRC 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CRC_DR	DR[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x04	CRC_IDR	IDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	CRC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REV_OUT	REV_IN[1:0]		POLYSIZE[1:0]		Res.	Res.	RESET
	Reset value																									0	0	0	0	0			0
0x10	CRC_INIT	CRC_INIT[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x14	CRC_POL	POL[31:0]																															
	Reset value	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	1	1	1	0	1	1	0	1	1	0	1	1	1
0x3F0	**CRC_HWCFGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FULL_POLY[7:0]							
	Reset value																									X	X	X	X	X	X	X	X
0x3F4	**CRC_VERR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV [3:0]				MINREV [3:0]			
	Reset value																									0	0	1	0	0	0	1	1
0x3F8	**CRC_IPIDR**	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x3FC	**CRC_SIDR**	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

22 可变存储控制器 (FMC)

可变存储控制器 (FMC) 包括以下两个存储控制器：

- NOR/PSRAM 存储控制器
- NAND 存储控制器

22.1 FMC 主要特性

FMC 功能块可连接：同步/异步静态内存和 NAND Flash。其主要用途有：

- 将 AXI 数据通信事务转换为适当的外部设备协议
- 满足外部内存设备的访问时间要求

所有外部内存与控制器共享地址、数据和控制信号，但有各自的片选信号。FMC 一次只能访问一个外部设备。

FMC 的主要特性如下：

- 连接静态内存映射的设备，包括：
  - 静态随机访问内存 (SRAM)
  - NOR Flash/OneNAND Flash
  - PSRAM（4 个存储子区域）
  - 铁电 RAM (FRAM)
  - 带有硬件 ECC 的 NAND Flash，可检查多达 8 KB 的数据
- 支持 Burst 模式，能够更快速地访问同步设备（如 NOR Flash 和 PSRAM）
- 可编程连续时钟输出以支持异步和同步访问
- 具有 8 位、16 位宽的数据总线
- 每个存储区域有独立的片选控制
- 每个存储区域可独立配置
- 写使能和字节通道选择输出，可配合 PSRAM 和 SRAM 设备使用
- 外部异步等待控制

启动时，必须通过用户应用程序对 FMC 引脚进行配置。应用程序未使用的 FMC 输入/输出引脚可用于其他用途。

复位后，FMC 将禁止。当所有使用的存储控制器完成配置后，必须通过将 FMCEN 位置 1 来使能 FMC。

定义外部设备类型和其特性的 FMC 寄存器在启动时进行设置，并且在下次上电或复位前保持不变。如果必须重新配置某些寄存器，必须禁止 FMC（将 FMCEN 位设置为 0x0）。但是，只有几个位可实时更改：

- FMCEN 位
- FMC_PCR 寄存器中的 ECCEN 和 PBEN 位
- FMC_CSQICR 和 FMC_CSQIER 寄存器位
- FMC 命令定序器寄存器和 FMC BCH 寄存器
- FMC_PCR 寄存器中的 WEN 位

如果在使能 FMC 的情况下必须修改某些参数，则必须先禁止 FMC，以确保在寄存器修改期间不允许进一步访问任何存储控制器。当所有需要的配置完成更新后，请按以下步骤使能 FMC：

1. 通过复位 FMCEN 位来禁止 FMC。
2. 等待 FMC_SR 寄存器中的 ISOST[1:0] 位设置为 0b11。
3. 等待 FMC_SR 寄存器中的 PEF 标志置 1。
4. 使用新值编程寄存器。
5. 通过将 FMCEN 位置 1 使能 FMC。

22.2 FMC 实现

表 126. FMC 特性

模式/特性	FMC
NAND Flash 片选输出	2
NAND Flash 就绪/繁忙	1
使用公共 WAIT 引脚的 NAND/NOR Flash	支持 (NWAIT)

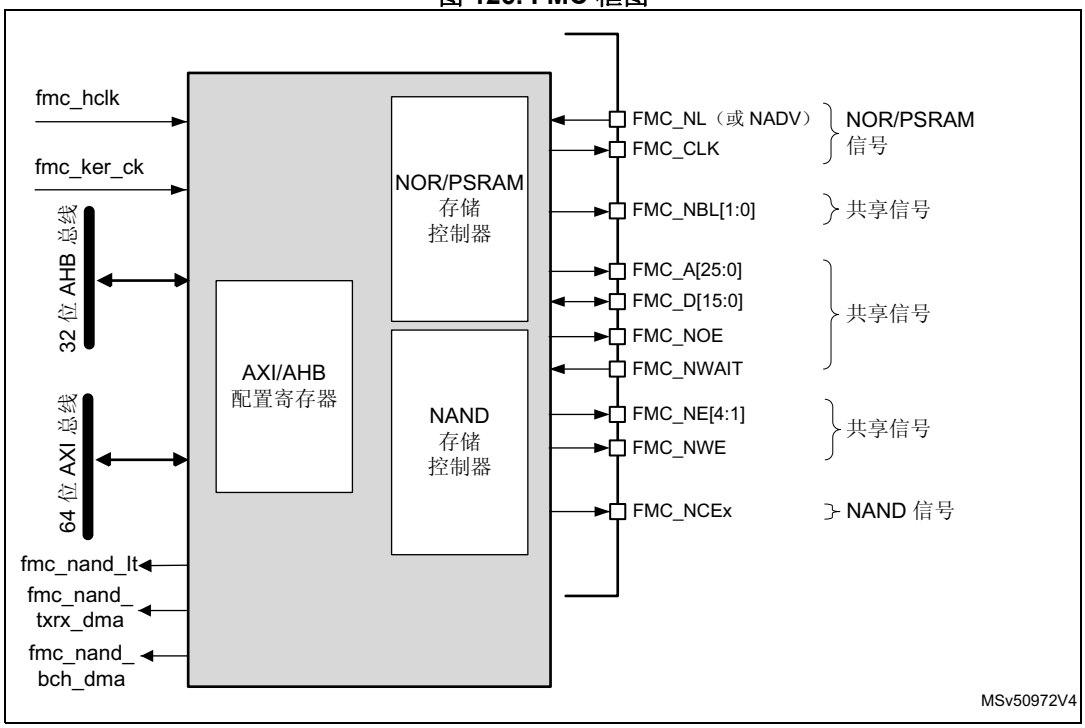
22.3 FMC 框图

FMC 包含以下主要模块：

- NOR Flash/PSRAM/SRAM 控制器
- NAND 控制器
- AXI 接口
- AHB 接口（包括 FMC 配置寄存器）

框图如下图所示。

图 126. FMC 框图



1. FMC 具有两个 NCEx 片选输出。

22.4 FMC 内部信号

表 127 列出了 FMC 内部信号。FMC 引脚（即外部信号）在*第 22.8.1 节：外部内存接口信号*中进行了说明。

表 127. FMC 内部信号

名称	信号类型	说明
fmc_nand_it	数字输出	FMC NAND Flash 中断
fmc_ker_ck	数字输入	FMC 内核时钟
fmc_hclk	数字输入	FMC 接口时钟
fmc_nand_txrx_dma	数字输出	用于对 NAND Flash 读取/写入数据的 DMA 请求
fmc_nand_bch_dma	数字输出	用于获取 BCH 结果的 DMA 请求

22.5 AHB 接口

AHB 从接口使得内部 CPU 能够配置 FMC 寄存器。

AHB 时钟 (HCLK) 是 FMC 寄存器访问的参考时钟。

因此只能以 32 位的方式访问 FMC 寄存器。对于非 32 位 AHB 访问，FMC 不会返回 AHB 总线错误，寄存器值可能会损坏。

22.6 AXI 接口

内部 CPU 和其他 AXI 总线主设备可通过该 AXI 从设备接口访问外部内存。

AXI 事务会转换为外部设备协议。由于 AXI 数据总线为 64 位宽，因此 AXI 事务可根据数据大小访问拆分成几个连续的 32 位、16 位或 8 位访问。

当满足下列其中一个条件时，FMC 将生成 AXI 从错误：

- 读取或写入未使能的 FMC 存储区域（存储区域 1 到 4）。
- 在 FMC 禁止（FMCEN 位复位）时读取或写入任何 FMC 存储区域
- 命令定序器读取或写入禁止的 NAND 存储区域
- 在 FMC_BCRx 寄存器中的 MBKEN 位复位时读取或写入 NOR Flash 存储区域

FMC 的内核时钟是异步 fmc_ker_ck 时钟（有关 fmc_ker_ck 时钟源选择的详细信息，请参见复位和时钟控制 (RCC) 一章）。

22.6.1 支持的内存和事务

通用事务规则

所请求的 AXI 事务数据宽度可以是 8、16、32 或 64 位，但访问的外部设备具有固定的数据宽度。这可能会导致传输不一致。

因此，必须遵循一些简单的事务规则：

- AXI 事务大小与内存数据大小必须相等，以防止问题的发生。
- AXI 事务数据宽度大于内存宽度：
  在此情况下，FMC 会将 AXI 事务分为多个较小的连续内存访问，以符合外部数据宽度。
- AXI 事务数据宽度小于内存宽度：
  传送可能一致，也可能不一致，具体取决于外部设备的类型：
  - 对具有字节选择功能的设备（SRAM、ROM、PSRAM）进行访问
    FMC 允许读/写事务并通过其字节选择通道 NBL[1:0] 访问恰当的数据。
    通过 NBL[1:0] 寻址要写入的字节。
    读取所有内存字节（NBL[1:0] 在读取事务期间保持为低电平），并丢弃无用的字节。
  - 访问不具有字节选择功能的设备（NOR 和 NAND Flash）
    当请求对 16 位宽的 Flash 进行字节访问时会发生此情形。由于无法在字节模式下访问设备（只能向 Flash 读取或写入 16 位字），因此允许读事务而不允许写事务（控制器会读取全部 16 位内存字，但只使用所需字节）。

NOR Flash/PSRAM 的回卷支持

由于并非所有主设备都能发出回卷事务，因此同步内存必须按未定义长度的线性 Burst 模式进行配置。

如果主设备生成回卷事务：

- 读操作将分为两个线性突发事务。
- 写操作将分为两个线性突发事务。

配置寄存器

可通过一组寄存器配置 FMC。有关 NOR Flash/PSRAM 控制寄存器的详细说明，请参见*第 22.8.6 节*。有关 NAND Flash 寄存器的详细说明，请参见*第 22.9.9 节*。

22.7 外部设备地址映射

从 FMC 的角度，外部内存被划分为固定大小的存储区域，每个存储区域的大小为 256 MB（请参见*表 128*）：

- 存储区域 1 可连接多达 4 个 NOR Flash 或 PSRAM 设备。此存储区域被划分为如下 4 个 NOR/PSRAM 子区域，带 4 个专用片选信号：
  - 子区域 1 - NOR/PSRAM 1
  - 子区域 2 - NOR/PSRAM 2
  - 子区域 3 - NOR/PSRAM 3
  - 子区域 4 - NOR/PSRAM 4

注： *必须将存储区域 1 的类型属性声明为“设备”。*

- 存储区域 3 用于寻址 NAND Flash 设备。此空间的 MPU 内存属性必须通过软件重新配置为设备。

对于每个存储区域，所要使用的内存类型可由用户应用程序通过配置寄存器配置。

表 128. FMC 存储区域（默认映射）

地址	存储区域	支持的内存类型
0x6000 0000 - 0x6FFF FFFF	存储区域 1	NOR Flash/PSRAM/SRAM
0x8000 0000 - 0x8FFF FFFF	存储区域 3	NAND Flash

22.7.1 NOR/PSRAM 地址映射

ADDR[27:26] 位用于从*表 129* 中所示的四个存储子区域之中选择其中一个存储区域。

表 129. NOR/PSRAM 子区域选择

ADDR[27:26](1)	所选子区域
00	子区域 1 - NOR/PSRAM 1
01	子区域 2 - NOR/PSRAM 2
10	子区域 3 - NOR/PSRAM 3
11	子区域 4 - NOR/PSRAM 4

1. ADDR 是内部地址线，但也会参与对外部内存的寻址。

ADDR[25:0] 位包含外部内存地址。由于 ADDR 为字节地址，而内存按字寻址，所以根据内存数据宽度不同，实际向内存发送的地址也将有所不同，如下表所示。

表 130. NOR/PSRAM 外部内存地址

内存宽度(1)	向内存发出的数据地址	最大内存容量（位）
8 位	ADDR[25:0]	64 MB * 8 = 512 Mb
16 位	ADDR[25:1] >> 1	64 MB/2 * 16 = 512 Mb

1. 如果外部内存的宽度为 16 位，FMC 将使用内部的 ADDR[25:1] 地址来作为对外部内存的寻址地址 FMC_A[24:0]。
   无论外部内存的宽度是多少，FMC_A[0] 都必须连接到外部内存地址 A[0]。

22.7.2 NAND Flash 地址映射

NAND 存储区域分为几个存储区，如*表 131* 所示。

表 131. NAND 存储映射和时序寄存器

起始地址	结束地址	FMC 存储区域	存储空间	时序寄存器
0x8800 0000	0x8FFF FFFF	存储区域 3 - NAND Flash	特性区	FMC_PATT (0x8C)
0x8000 0000	0x87FF FFFF		通用区	FMC_PMEM (0x88)

对于 NAND Flash，通用区和特性区存储空间分为三个部分，均位于低位 256 KB 中（见下面的*表 132*）：

- 数据区域（通用/特性存储空间中的第一个 64 KB）
- 命令区域（通用/特性存储空间中的第二个 64 KB）
- 地址区域（通用/特性存储空间中的下一个 128 KB）

表 132. NAND Flash 区域选择

区域名称	ADDR[17:16]	地址范围
地址区域	10	0x20000-0x3FFFF
命令区域	01	0x10000-0x1FFFF
数据区域	00	0x00000-0x0FFFF

ADDR[25:24] 位可用于选择对 NAND 内存发出的 NAND Flash 芯片使能信号，如*表 133* 所示。使用这两位可以从一个芯片使能信号切换到另一个芯片使能信号，以便寻址多芯片封装内的一个给定 NAND Flash。

表 133. NAND 芯片使能信号选择

ADDR[25:24]	片选
00	NCE1
01	NCE2
其他配置	保留

应用程序软件使用这 3 个区域来访问 NAND Flash：

- **向 NAND Flash 存储器发送命令**，软件必须向命令区域中的任意存储单元写入命令值。
- **指定读取或写入的 NAND Flash 地址**，软件必须向地址区域中的任意存储单元写入地址值。由于地址的长度可以是 4 或 5 个字节（具体取决于实际内存大小），要指定完整的地址，需要对地址区域执行多个连续写入操作。
- **读取或写入数据**，软件将从数据区域中的任意存储单元读取数据，或者向其中写入数据。

由于 NAND Flash 会自动递增地址，所以在访问连续存储单元时，无需递增数据区域的地址。

注： *必须将 NAND 区域作为设备而不是内存来进行访问。对命令区域和地址区域进行写访问时，必须采用字节模式并与区域基址（偏移量 = 0）对齐。*

对 NAND 数据区域进行读写访问时，对于 8 位 NAND Flash，必须采用字节模式；对于 16 位 NAND Flash，必须采用字节模式或半字模式。

读写突发操作必须从区域基址（偏移量 = 0）开始，并且不得跨越 16 字节区域边界。

22.8 NOR Flash/PSRAM 控制器

FMC 会生成适当的信号时序，以驱动以下类型的内存：

- 异步 SRAM、FRAM 和 ROM
  - 8 位
  - 16 位
- PSRAM (CellularRAM™)
  - 异步模式
  - 用于同步访问的 Burst 模式，具有在跨越 CRAM 1.5 的边界页时用于拆分突发访问的可配置选项。
  - 复用或非复用
- NOR Flash
  - 异步模式
  - 同步访问的 Burst 模式
  - 复用或非复用

FMC 会为每个存储区域输出唯一的片选信号 NE[4:1]。所有其他信号（地址、数据和控制）均为共享信号。

FMC 通过可编程时序支持多种设备，其中：

- 等待周期可编程（最多 15 个时钟周期）
- 总线周转周期可编程（最多 15 个时钟周期）
- 输出使能和写入使能延迟可编程（最多 15 个时钟周期）
- 独立的读和写时序和协议，以支持各种内存和时序
- 可编程连续时钟 (FMC_CLK) 输出。

FMC 输出时钟 (FMC_CLK) 是 fmc_ker_ck 时钟的约数。该时钟可传送到选定的外部设备，根据 FMC_BCR1 寄存器中 CCKEN 位的配置，可决定只在同步访问期间传送还是同步异步访问期间都传送：

- 如果 CCLKEN 位复位，则 FMC 仅在同步访问（读/写事务）期间生成时钟 (FMC_CLK)。
- 如果 CCLKEN 位置 1，则 FMC 将在异步和同步访问期间生成连续时钟。要生成 FMC_CLK 连续时钟，存储区域 1 还必须配置为支持同步模式（请参见*第 22.8.6 节：NOR/PSRAM 控制寄存器*）。由于所有同步内存均使用同一个时钟，因此在生成连续输出时钟和执行同步访问时，AXI 数据大小必须与内存数据宽度 (MWID[1:0]) 相同，否则 FMC_CLK 频率将根据 AXI 数据事务发生变化（有关 FMC_CLK 分频比公式的信息，请参见*第 22.8.5 节：同步事务*）。

每个存储区域的大小固定，均为 64 MB。每个存储区域都通过专用的寄存器配置（参见*第 22.8.6 节：NOR/PSRAM 控制寄存器*）。

内存的可编程参数包括访问时间（参见*表 134*）和对等待管理的支持（用于在 Burst 模式下访问 PSRAM 和 NOR Flash）。

表 134. NOR/PSRAM 的可编程访问参数

参数	功能	访问模式	单位	最小值	最大值
地址建立	地址建立阶段的持续时间	异步	FMC 时钟周期 (fmc_ker_ck)	0	15
地址保持	地址保持阶段的持续时间	异步，复用 I/O	FMC 时钟周期 (fmc_ker_ck)	1	15
NBL 建立	字节通道建立阶段的持续时间	异步	FMC 时钟周期 (fmc_ker_ck)	0	3
数据建立	数据建立阶段的持续时间	异步	FMC 时钟周期 (fmc_ker_ck)	1	256
数据保持	数据保持阶段的持续时间	异步	FMC 时钟周期 (fmc_ker_ck)	0	3
总线周转	总线周转阶段的持续时间	异步和同步读取	FMC 时钟周期 (fmc_ker_ck)	0	15
时钟分频比	构建一个内存时钟周期 (CLK) 所需的 FMC 时钟周期 (fmc_ker_ck) 数量	同步	FMC 时钟周期 (fmc_ker_ck)	2	16
数据延迟	在发出突发的第一个数据前向内存发出的时钟周期数量	同步	内存时钟周期 (fmc_ker_ck)	2	17

22.8.1 外部内存接口信号

表 135、*表 136* 和*表 137* 列出了通常用于连接 NOR Flash、SRAM 和 PSRAM 的信号。

注： *前缀“N”标识低电平有效的信号。*

NOR Flash，非复用 I/O

表 135. 非复用 I/O NOR Flash

FMC 信号名称	I/O	功能
CLK	O	时钟（用于同步访问）
A[25:0]	O	地址总线
D[15:0]	I/O	双向数据总线
NE[x]	O	片选，x = 1 到 4
NOE	O	输出使能
NWE	O	写入使能
NL(= NADV)	O	锁存使能（对于部分 NOR Flash 设备，此信号也称为地址有效 (NADV)）
NWAIT	I	FMC 的 NOR Flash 等待输入信号

最大容量为 512 Mb（26 个地址线）。

NOR Flash，16 位复用 I/O

表 136. 16 位复用 I/O NOR Flash

FMC 信号名称	I/O	功能
CLK	O	时钟（用于同步访问）
A[25:16]	O	地址总线
AD[15:0]	I/O	16 位复用，双向地址/数据总线 （16 位地址 A[15:0] 和数据 D[15:0] 在数据总线上复用）
NE[x]	O	片选，x = 1 到 4
NOE	O	输出使能
NWE	O	写入使能
NL(= NADV)	O	锁存使能 （对于部分 NOR Flash 设备，此信号也称为地址有效 (NADV)）
NWAIT	I	FMC 的 NOR Flash 等待输入信号

最大容量为 512 Mb。

PSRAM/SRAM/FRAM，非复用 I/O

表 137. 非复用 I/O PSRAM/SRAM

FMC 信号名称	I/O	功能
CLK	O	时钟（仅用于 PSRAM 同步访问）
A[25:0]	O	地址总线
D[15:0]	I/O	数据双向总线
NE[x]	O	片选，x = 1 到 4（在 PSRAM 应用中被称作 NCE （CellularRAM™，即 CRAM））
NOE	O	输出使能
NWE	O	写入使能
NL(= NADV)	O	仅用于 PSRAM 输入的地址有效信号（内存信号名称：NADV）
NWAIT	I	PSRAM 发送给 FMC 的等待输入信号
NBL[1:0]	O	字节通道输出（高字节和低字节使能）

最大容量为 512 Mb。

PSRAM，16 位复用 I/O

表 138. 16 位复用 I/O PSRAM

FMC 信号名称	I/O	功能
CLK	O	时钟（用于同步访问）
A[25:16]	O	地址总线

表 138. 16 位复用 I/O PSRAM（续）

FMC 信号名称	I/O	功能
AD[15:0]	I/O	16 位复用，双向地址/数据总线 （16 位地址 A[15:0] 和数据 D[15:0] 在数据总线上复用）
NE[x]	O	片选，x = 1 到 4（在 PSRAM 应用中被称作 NCE （CellularRAM™，即 CRAM））
NOE	O	输出使能
NWE	O	写入使能
NL(= NADV)	O	仅用于 PSRAM 输入的地址有效信号（内存信号名称：NADV）
NWAIT	I	PSRAM 发送给 FMC 的等待输入信号
NBL[1:0]	O	字节通道输出（高字节和低字节使能）

最大容量为 512 Mb（26 个地址线）。

22.8.2 支持的内存和事务

下面的*表 139* 显示了当 NOR Flash、PSRAM 和 SRAM 的内存数据总线为 16 位宽时所支持的设备、访问模式和事务的示例。在本例中，FMC 不允许（或不支持）的事务显示为灰色。

表 139. NOR Flash/PSRAM：支持的内存和事务示例

设备	模式	R/W	AXI 数据大小	内存数据大小	是否允许	备注
NOR Flash（复用 I/O 和非复用 I/O）	异步	R	8	16	Y	-
	异步	W	8	16	N	-
	异步	R	16	16	Y	-
	异步	W	16	16	Y	-
	异步	R	32	16	Y	分为 2 次 FMC 访问
	异步	W	32	16	Y	分为 2 次 FMC 访问
	异步	R	64	16	Y	分为 4 次 FMC 访问
	异步	W	64	16	Y	分为 4 次 FMC 访问
	异步页	R	-	16	N	不支持该模式
	同步	R	8	16	N	-
	同步	R	16	16	Y	-
	同步	R	32	16	Y	-
	同步	R	64	16	Y	分为 4 次 FMC 访问

表 139. NOR Flash/PSRAM：支持的内存和事务示例（续）

设备	模式	R/W	AXI 数据大小	内存数据大小	是否允许	备注
PSRAM（复用 I/O 和非复用 I/O）	异步	R	8	16	Y	-
	异步	W	8	16	Y	使用字节通道 NBL[1:0]
	异步	R	16	16	Y	-
	异步	W	16	16	Y	-
	异步	R	32	16	Y	分为 2 次 FMC 访问
	异步	W	32	16	Y	分为 2 次 FMC 访问
	异步	R	64	16	Y	分为 4 次 FMC 访问
	异步	W	64	16	Y	分为 4 次 FMC 访问
	异步页	R	-	16	N	不支持该模式
	同步	R	8	16	N	-
	同步	R	16	16	Y	-
	同步	R	32/64	16	Y	-
	同步	W	8	16	Y	使用字节通道 NBL[1:0]
	同步	W	16/32/64	16	Y	-
SRAM 和 ROM	异步	R	8 / 16	16	Y	-
	异步	W	8 / 16	16	Y	使用字节通道 NBL[1:0]
	异步	R	32	16	Y	分为 2 次 FMC 访问
	异步	W	32	16	Y	分为 2 次 FMC 访问
	异步	R	64	16	Y	分为 4 次 FMC 访问
	异步	W	64	16	Y	分为 4 次 FMC 访问

22.8.3 通用时序规则

信号同步

- 所有的控制器输出信号均在 fmc_ker_ck 时钟的上升沿变化
- 在同步模式（读取或写入）下，所有输出信号均在 fmc_ker_ck 时钟的上升沿变化。
- NBL 信号

  这些信号适用于支持字节使能的内存。
  - NBL 信号在下列情况下驱动为低电平：

    开始进行读取内存传输时，并且在设备从 NBL 到 NCE 需要一定的建立时间时可选择提前（根据 NBLSET）

    针对数据传输期间要写入的字节，并且根据地址传输期间的第一个数据字节。
  - NBL 信号在读/写传输完成后的两个 FMC_CLK 时钟周期后驱动为高电平。

22.8.4 NOR Flash/PSRAM 控制器异步事务

异步静态内存（NOR Flash、PSRAM、SRAM、FRAM）

- 信号通过内部时钟 fmc_ker_ck 进行同步。不会将此时钟发送到内存
- FMC 总是会先对数据进行采样，而后再禁止片选信号 NE。这样可以确保符合内存数据保持时序的要求（最小芯片使能高电平到数据转换通常为 0 ns）
- 如果使能扩展模式（FMC_BCRx 寄存器中的 EXTMOD 位置 1），则最多可提供四种扩展模式（A、B、C 和 D）。可以混合使用 A、B、C 和 D 模式来进行读取和写入操作。例如，可以在模式 A 下执行读取操作，而在模式 B 下执行写入操作。扩展模式与复用模式不兼容（在复用模式下，EXTMOD 位必须保持清零状态）。
- 如果禁止扩展模式（FMC_BCRx 寄存器中的 EXTMOD 位复位），则 FMC 可以在模式 1 或模式 2 下运行，如下所述：
  - 当选择 SRAM/PSRAM 内存类型时，模式 1 为默认模式（FMC_BCRx 寄存器中 MTYP[1:0] = 0x0 或 0x01）
  - 当选择 NOR 内存类型时，模式 2 为默认模式（FMC_BCRx 寄存器中 MTYP[1:0] = 0x10）。

模式 1 - SRAM/FRAM/PSRAM (CRAM)

下图显示了所支持模式的读取和写入事务以及所需的 FMC_BCRx 和 FMC_BTRx/FMC_BWTRx 寄存器配置。

图 127. 模式 1 读取访问波形

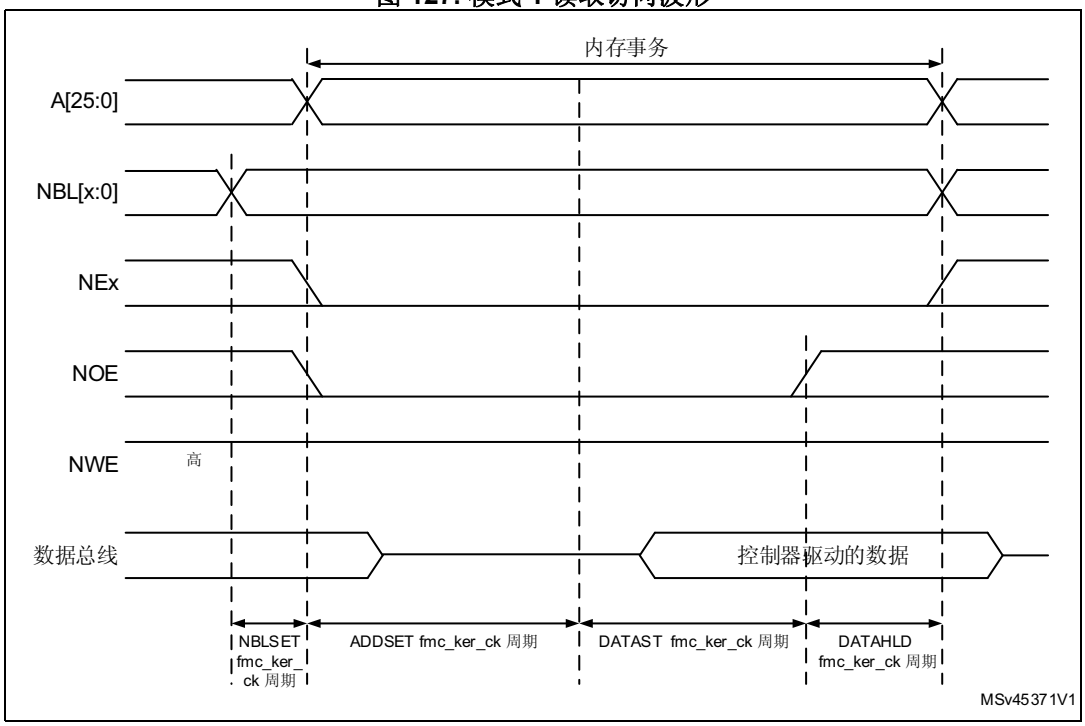
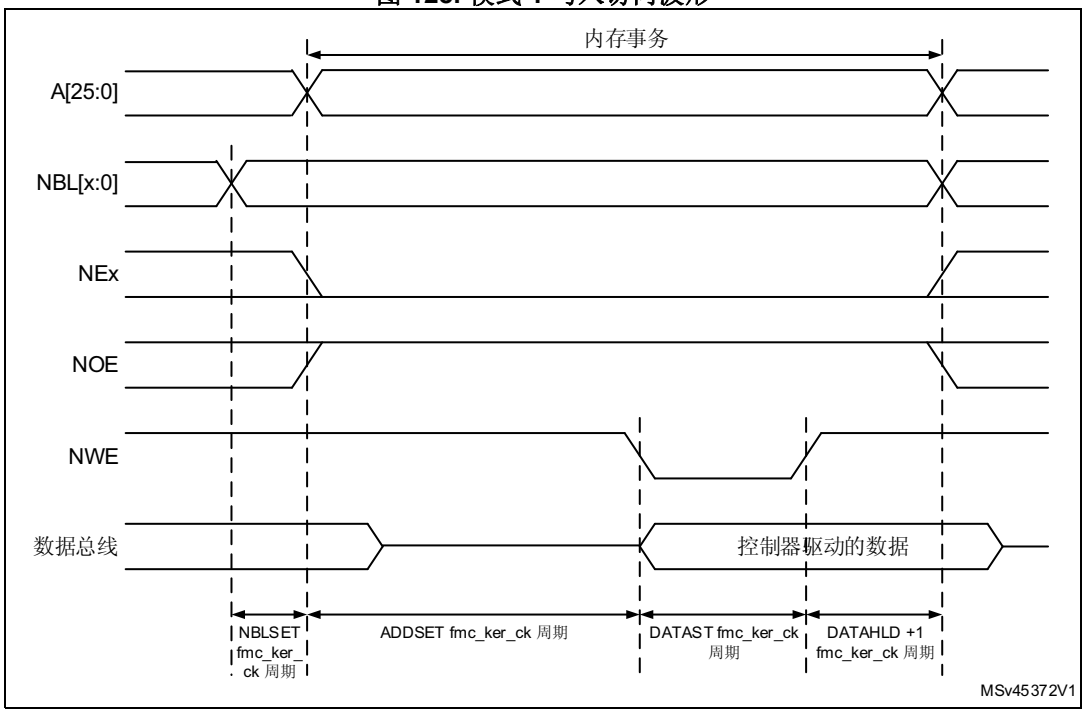


图 128. 模式 1 写入访问波形



位于读写事务末尾的 DATAHLD 时间有助于确保 NOE/NWE 上升沿之后的地址和数据保持时间。对于写入访问，DATAST 值必须大于零 (DATAST > 0)。

表 140. FMC_BCRx 位字段（模式 1）

位号	位名	要设置的值
31	FMCEN	0x1
30-24	保留	0x000
23:22	NBLSET[1:0]	根据需要进行设置
21	保留	0x0
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	在内存支持该特性的情况下置 1，否则保持清零状态。
14	EXTMOD	0x0
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	无关
5-4	MWID[1:0]	根据需要进行设置
3-2	MTYP	根据需要进行设置，0x2 (NOR Flash) 除外
1	MUXE	0x0
0	MBKEN	0x1

表 141. FMC_BTRx 位字段（模式 1）

位号	位名	要设置的值
31:30	DATAHLD	读取访问的数据保持阶段的持续时间（DATAHLD 个 fmc_ker_ck 周期）。
29-28	ACCMOD	无关
27-24	DATLAT	无关
23-20	CLKDIV	无关
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	第二个访问阶段的持续时间（DATAST 个 fmc_ker_ck 周期）。

表 141. FMC_BTRx 位字段（模式 1）（续）

位号	位名	要设置的值
7-4	ADDHLD	无关
3-0	ADDSET	第一个访问阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。 ADDSET 的最小值为 0。

模式 A - SRAM/FRAM/PSRAM (CRAM) OE 切换

图 129. 模式 A 读取访问波形

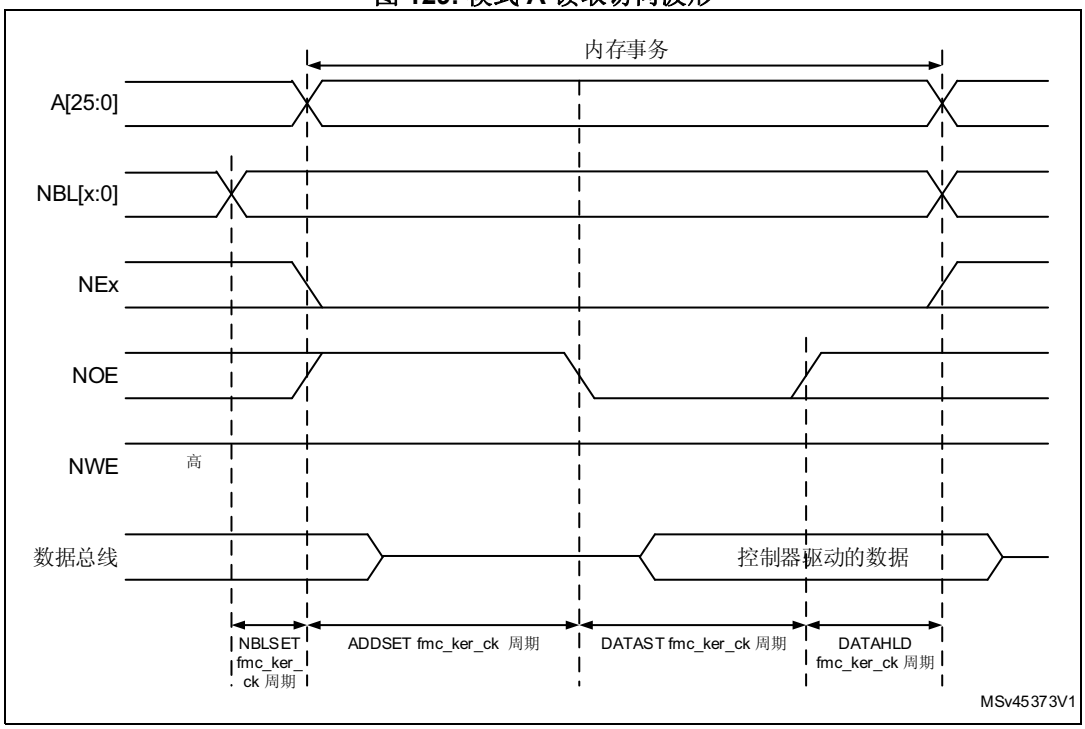
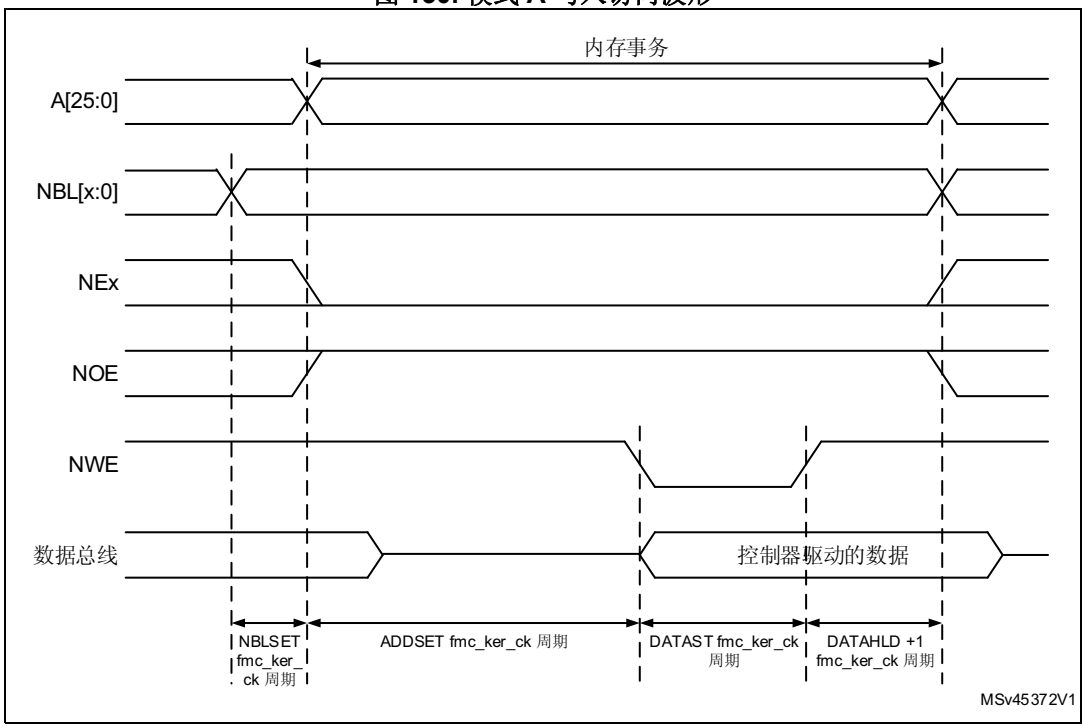


图 130. 模式 A 写入访问波形



与模式 1 的不同之处在于 NOE 的切换与独立的读取和写入时序。

表 142. FMC_BCRx 位字段（模式 A）

位号	位名	要设置的值
31	FMCEN	0x1
30-24	保留	0x000
23:22	NBLSET[1:0]	根据需要进行设置
21	保留	0x0
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	在内存支持该特性的情况下置 1，否则保持清零状态。
14	EXTMOD	0x1
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1

表 142. FMC_BCRx 位字段（模式 A）（续）

位号	位名	要设置的值
6	FACCEN	无关
5-4	MWID[1:0]	根据需要进行设置
3-2	MTYP	根据需要进行设置，0x2 (NOR Flash) 除外
1	MUXEN	0x0
0	MBKEN	0x1

表 143. FMC_BTRx 位字段（模式 A）

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（对于读取访问，为 DATAHLD 个 HCLK 周期）。
29-28	ACCMOD	0x0
27-24	DATLAT	无关
23-20	CLKDIV	无关
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	读取访问第二个阶段的持续时间（DATAST 个 fmc_ker_ck 周期）。
7-4	ADDHLD	无关
3-0	ADDSET	读取访问第一个阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。ADDSET 的最小值为 0。

表 144. FMC_BWTRx 位字段（模式 A）

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（对于写入访问，为 DATAHLD+1 个 fmc_ker_ck 周期）。
29-28	ACCMOD	0x0
27:20	保留	0x0
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	写入访问第二个阶段的持续时间（DATAST 个 fmc_ker_ck 周期）。
7-4	ADDHLD	无关
3-0	ADDSET	写入访问第一个阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。ADDSET 的最小值为 0。

模式 2/B - NOR Flash

图 131. 模式 2 和模式 B 读取访问波形

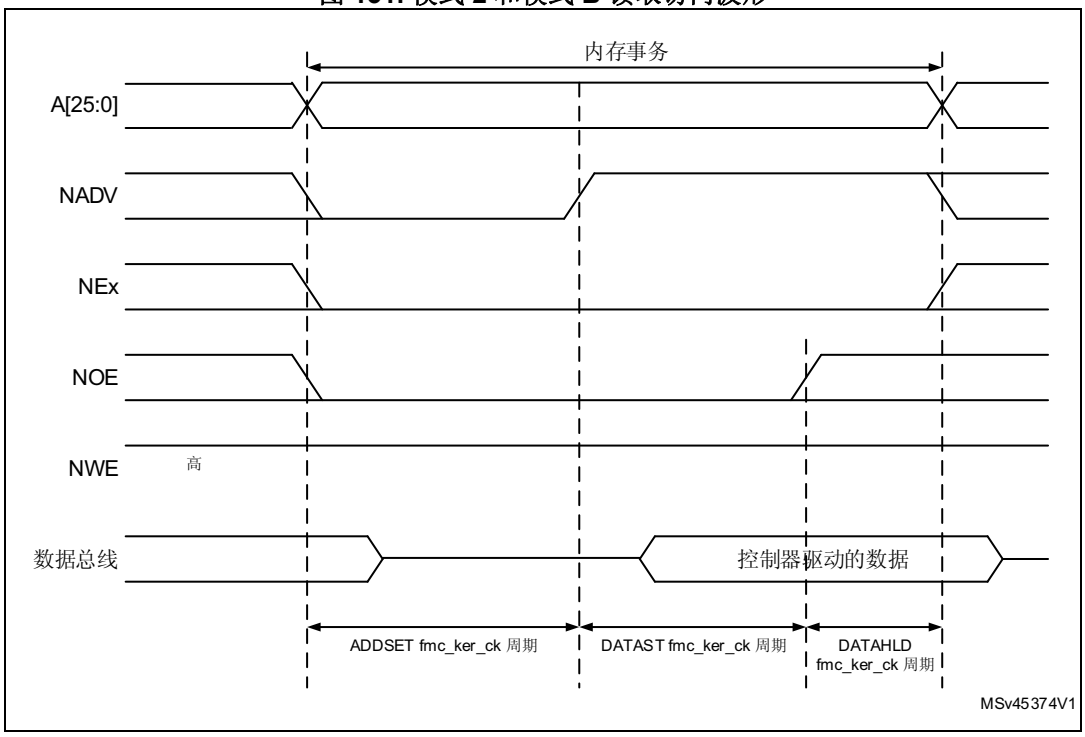


图 132. 模式 2 写入访问波形

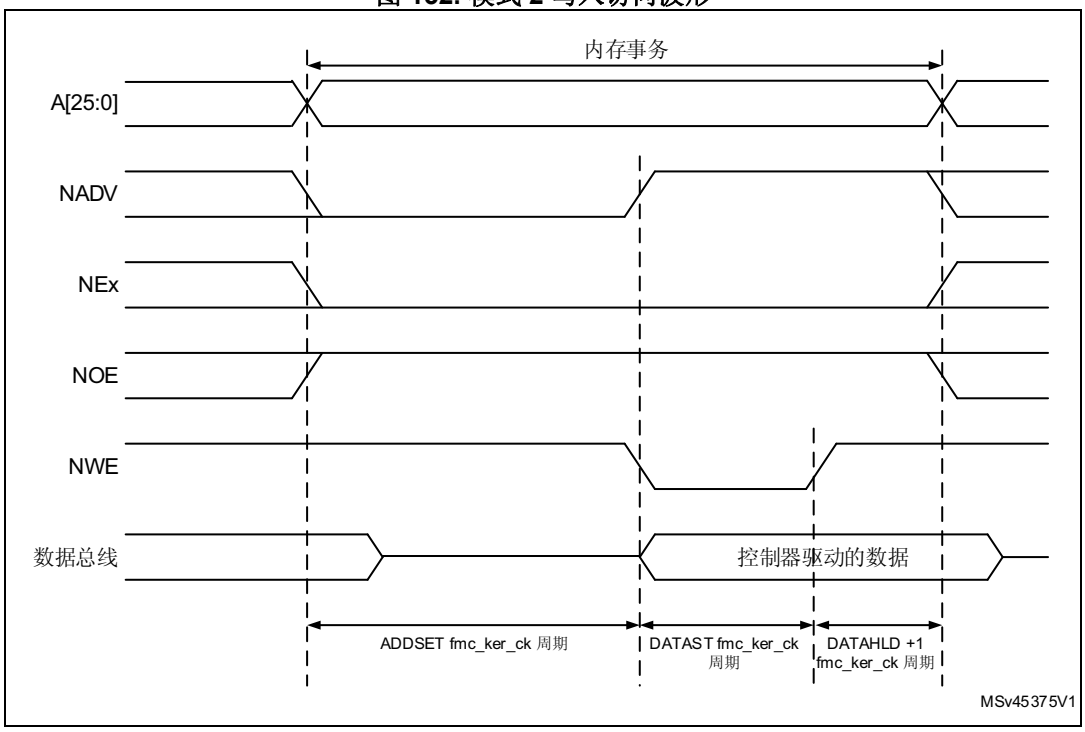
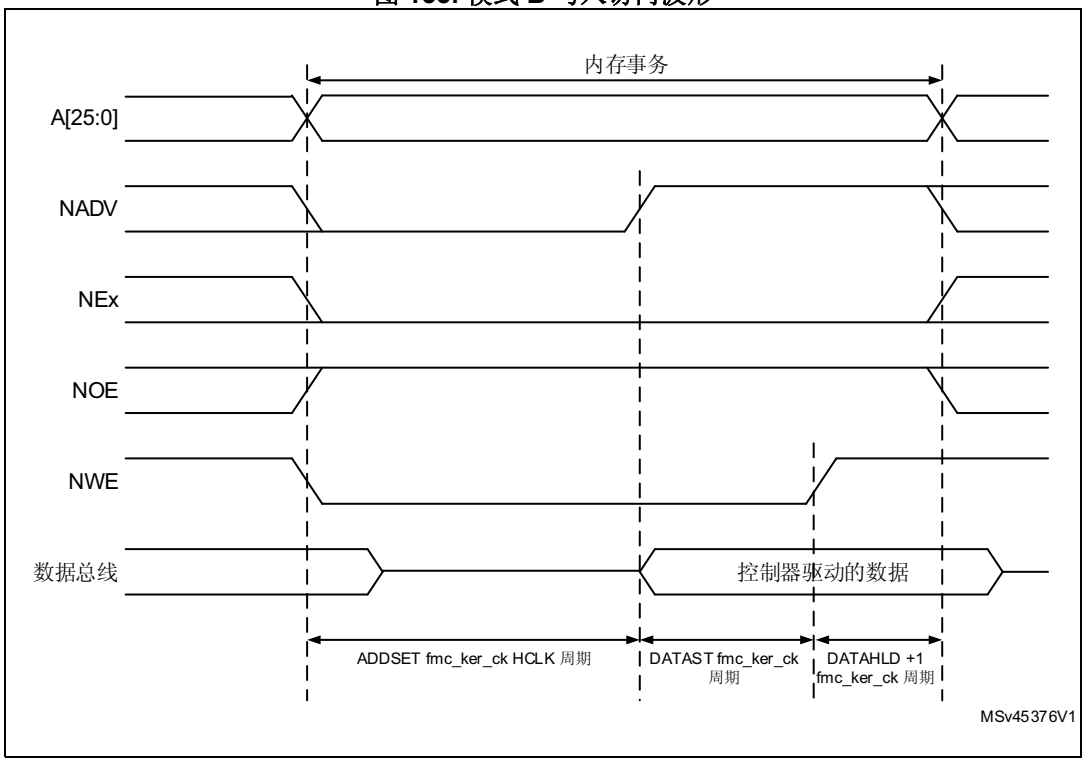


图 133. 模式 B 写入访问波形



与模式 1 的不同之处在于设置扩展模式（模式 B）时 NWE 的切换与独立的读取和写入时序。

表 145. FMC_BCRx 位字段（模式 2/B）

位号	位名	要设置的值
31	FMCEN	0x1
30-24	保留	0x000
23:22	NBLSET[1:0]	根据需要进行设置
21	保留	0x0
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	在内存支持该特性的情况下置 1，否则保持清零状态。
14	EXTMOD	模式 B 为 0x1，模式 2 为 0x0
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1

表 145. FMC_BCRx 位字段（模式 2/B）（续）

位号	位名	要设置的值
6	FACCEN	0x1
5-4	MWID[1:0]	根据需要进行设置
3-2	MTYP	0x2 (NOR Flash)
1	MUXEN	0x0
0	MBKEN	0x1

表 146. FMC_BTRx 位字段（模式 2/B）

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（对于读取访问，为 DATAHLD 个 HCLK 周期）。
29-28	ACCMOD	如果设置了扩展模式，则为 0x1
27-24	DATLAT	无关
23-20	CLKDIV	无关
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	读取访问第二个阶段的持续时间（DATAST 个 fmc_ker_ck 周期）。
7-4	ADDHLD	无关
3-0	ADDSET	读取访问第一个阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。ADDSET 的最小值为 0。

表 147. FMC_BWTRx 位字段（模式 2/B）

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（对于写入访问，为 DATAHLD+1 个 fmc_ker_ck 周期）。
29-28	ACCMOD	如果设置了扩展模式，则为 0x1
27:20	保留	0x0
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	写入访问第二个阶段的持续时间（DATAST 个 fmc_ker_ck 周期）。
7-4	ADDHLD	无关
3-0	ADDSET	写入访问第一个阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。ADDSET 的最小值为 0。

注：　仅当设置了扩展模式（模式 B）时，FMC_BWTRx 寄存器才有效，否则其内容均为“无关”。

模式 C - NOR Flash - OE 切换

图 134. 模式 C 读取访问波形

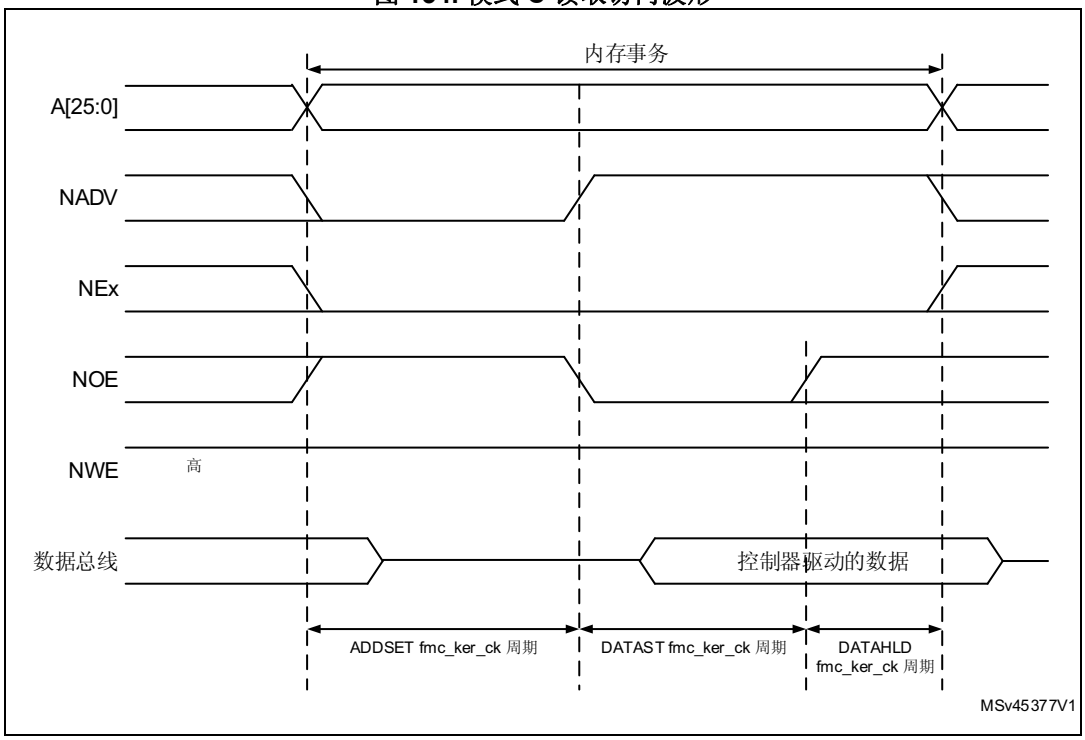
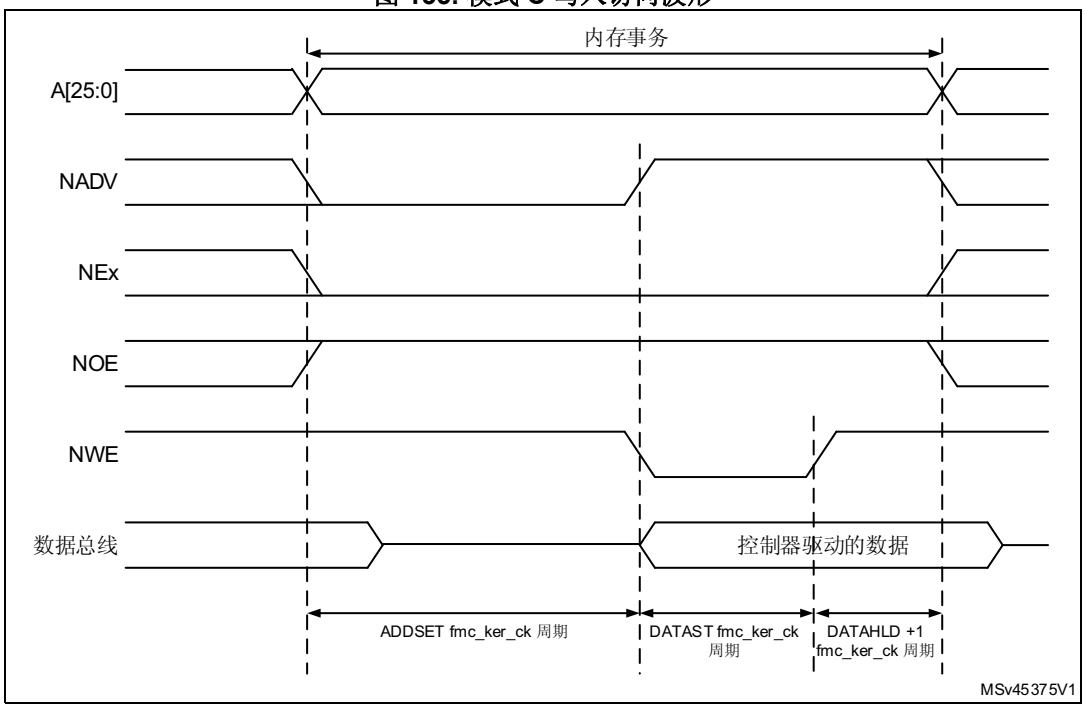


图 135. 模式 C 写入访问波形



与模式 1 的不同之处在于 NOE 的切换与独立的读取和写入时序。

表 148. FMC_BCRx 位字段（模式 C）

位号	位名	要设置的值
31	FMCEN	0x1
30-24	保留	0x000
23:22	NBLSET[1:0]	根据需要进行设置
21	保留	0x0
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	在内存支持该特性的情况下置 1，否则保持清零状态。
14	EXTMOD	0x1
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	0x1
5-4	MWID[1:0]	根据需要进行设置
3-2	MTYP[1:0]	0x02 (NOR Flash)
1	MUXEN	0x0
0	MBKEN	0x1

表 149. FMC_BTRx 位字段（模式 C）

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（对于读取访问，为 DATAHLD 个 HCLK 周期）。
29-28	ACCMOD	0x2
27-24	DATLAT	0x0
23-20	CLKDIV	0x0
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	读取访问第二个阶段的持续时间（DATAST 个 fmc_ker_ck 周期）。
7-4	ADDHLD	无关
3-0	ADDSET	读取访问第一个阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。ADDSET 的最小值为 0。

表 150. FMC_BWTRx 位字段（模式 C）

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（对于写入访问，为 DATAHLD+1 个 fmc_ker_ck 周期）。
29-28	ACCMOD	0x2
27:20	保留	0x0
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	写入访问第二个阶段的持续时间（DATAST 个 fmc_ker_ck 周期）。
7-4	ADDHLD	无关
3-0	ADDSET	写入访问第一个阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。ADDSET 的最小值为 0。

模式 D - 扩展地址异步访问

图 136. 模式 D 读取访问波形

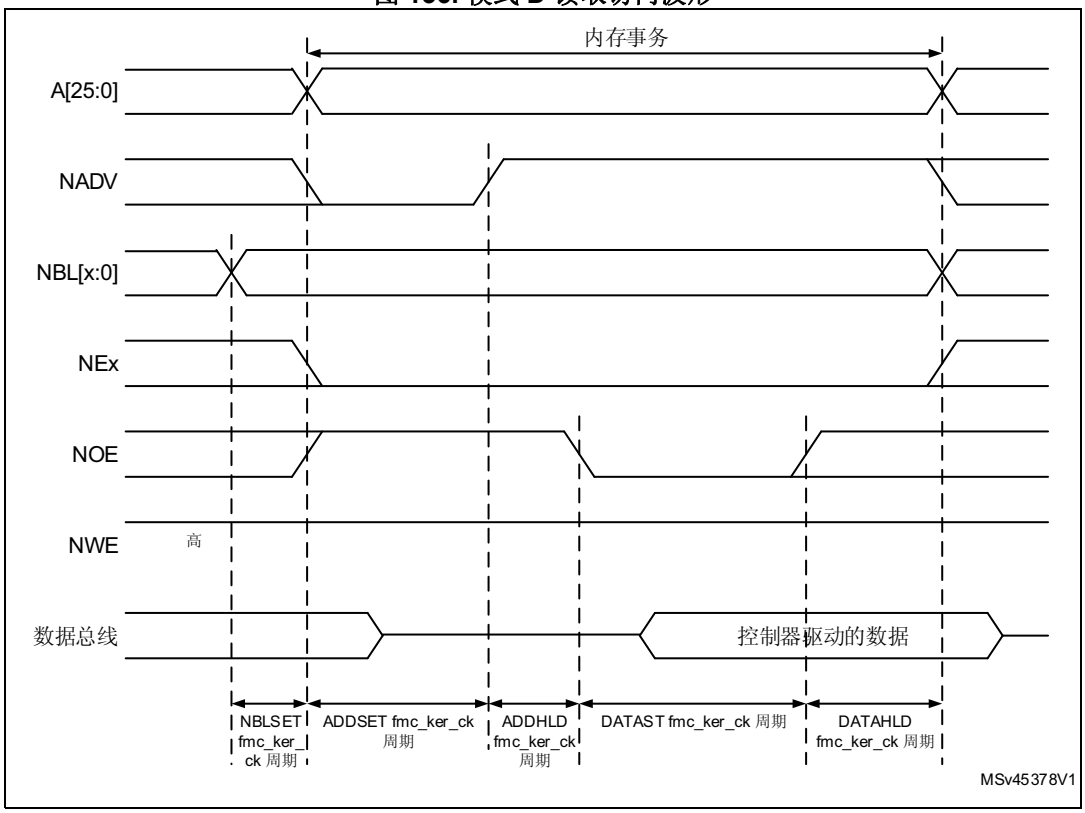
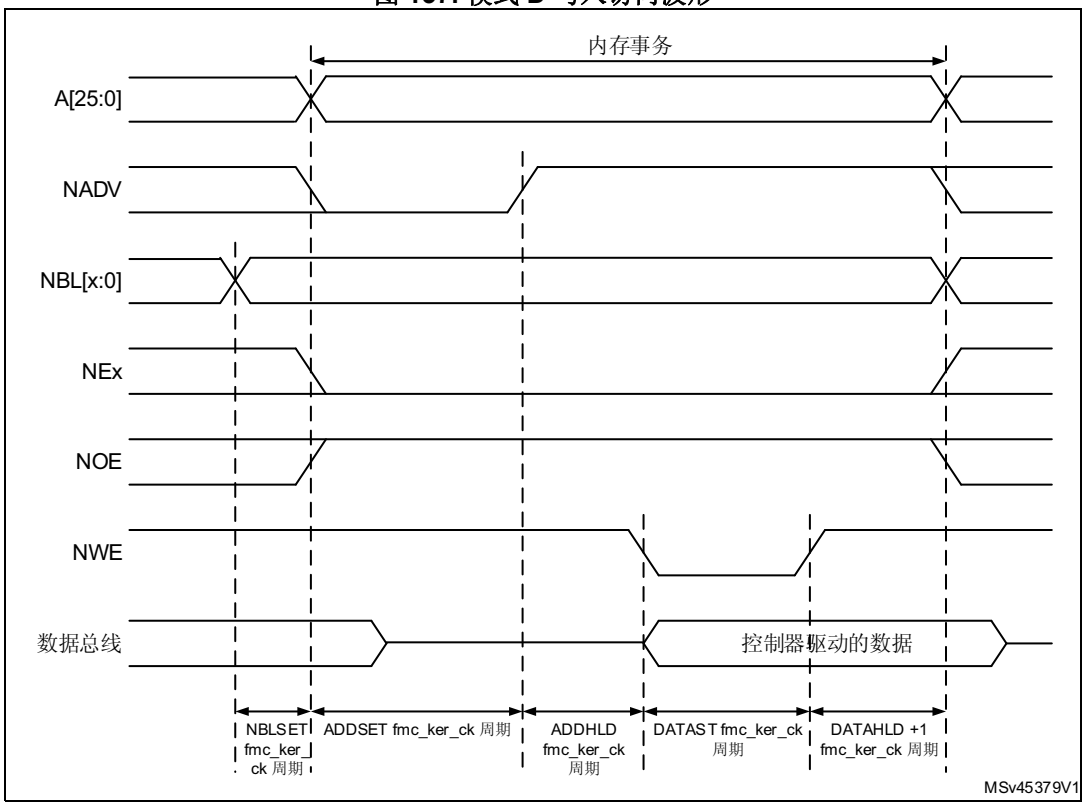


图 137. 模式 D 写入访问波形



与模式 1 的不同之处在于 NADV 变化后 NOE 的切换与独立的读取和写入时序。

表 151. FMC_BCRx 位字段（模式 D）

位号	位名	要设置的值
31	FMCEN	0x1
30-24	保留	0x000
23:22	NBLSET[1:0]	根据需要进行设置
21	保留	0x0
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	在内存支持该特性的情况下置 1，否则保持清零状态。 在模式 D 下，不支持 ASYNCWAIT 功能。
14	EXTMOD	0x1
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义

表 151. FMC_BCRx 位字段（模式 D）（续）

位号	位名	要设置的值
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	根据内存支持情况进行设置
5-4	MWID[1:0]	根据需要进行设置
3-2	MTYP[1:0]	根据需要进行设置
1	MUXEN	0x0
0	MBKEN	0x1

表 152. FMC_BTRx 位字段（模式 D）

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（对于读取访问，为 DATAHLD 个 HCLK 周期）。
29-28	ACCMOD	0x3
27-24	DATLAT	无关
23-20	CLKDIV	无关
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	读取访问第二个阶段的持续时间（DATAST 个 fmc_ker_ck 周期）。
7-4	ADDHLD	读取访问中间阶段的持续时间（ADDHLD 个 fmc_ker_ck 周期）。
3-0	ADDSET	读取访问第一个阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。ADDSET 的最小值为 1。

表 153. FMC_BWTRx 位字段（模式 D）

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（对于写入访问，为 DATAHLD+1 个 fmc_ker_ck 周期）。
29-28	ACCMOD	0x3
27:20	保留	0x0
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	写入访问第二个阶段的持续时间（DATAST + 1 个 fmc_ker_ck 周期）
7-4	ADDHLD	写入访问中间阶段的持续时间（ADDHLD 个 fmc_ker_ck 周期）
3-0	ADDSET	写入访问第一个阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。ADDSET 的最小值为 1。

复用模式 - 复用异步访问 NOR Flash

图 138. 复用读取访问波形

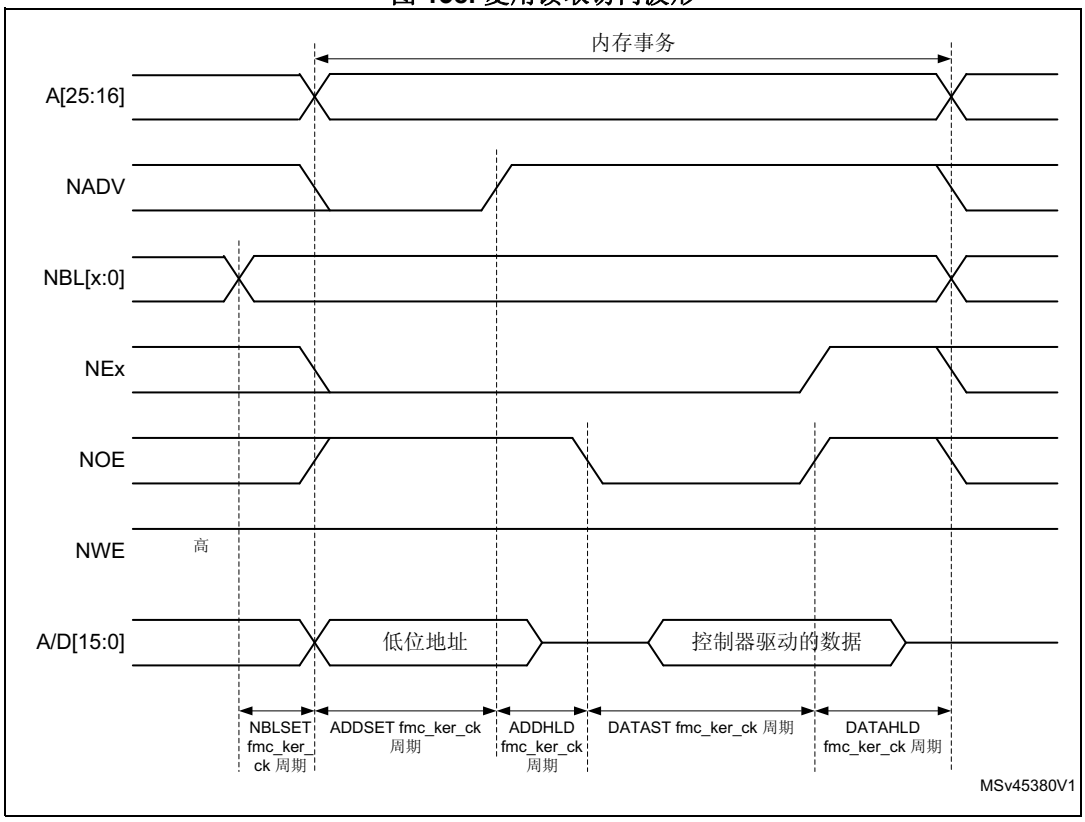
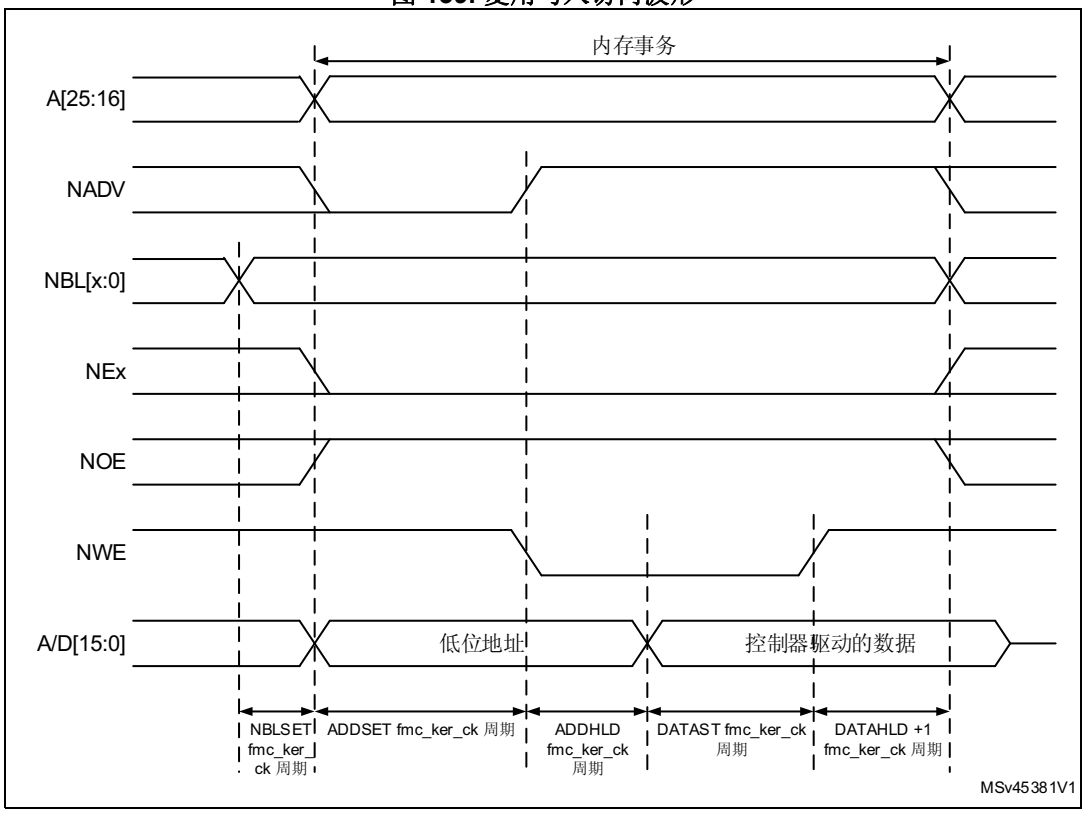


图 139. 复用写入访问波形



与模式 D 的不同之处在于在数据总线上驱动低地址字节。

表 154. FMC_BCRx 位字段（复用模式）

位号	位名	要设置的值
31	FMCEN	0x1
30-24	保留	0x000
23:22	NBLSET[1:0]	根据需要进行设置
21	保留	0x0
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	在内存支持该特性的情况下置 1，否则保持清零状态。 在复用模式下，不支持 ASYNCWAIT 功能。
14	EXTMOD	0x0
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义

表 154. FMC_BCRx 位字段（复用模式）（续）

位号	位名	要设置的值
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	0x1
5-4	MWID[1:0]	根据需要进行设置
3-2	MTYP[1:0]	0x2 (NOR Flash)
1	MUXEN	0x1
0	MBKEN	0x1

表 155. FMC_BTRx 位字段（复用模式）

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（读取访问为 DATAHLD 个 fmc_ker_ck 周期，写入访问为 DATAHLD+1 个 fmc_ker_ck 周期）。
29-28	ACCMOD	0x0
27-24	DATLAT	无关
23-20	CLKDIV	无关
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	第二个访问阶段的持续时间（读取访问为 DATAST 个 fmc_ker_ck 周期，写入访问为 DATAST+1 个 fmc_ker_ck 周期）。
7-4	ADDHLD	访问中间阶段的持续时间（ADDHLD 个 fmc_ker_ck 周期）。
3-0	ADDSET	第一个访问阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。ADDSET 的最小值为 1。

异步访问的 WAIT 管理

如果异步内存发出 WAIT 信号，指示尚未准备好接受或提供数据，则 FMC_BCRx 寄存器中的 ASYNCWAIT 位必须置 1。

如果 WAIT 信号处于有效状态（电平高低取决于 WAITPOL 位），则由 DATAST 位控制的第二个访问阶段（数据建立阶段）将延长，直到 WAIT 变为无效状态。与数据建立阶段不同，由 ADDSET 和 ADDHLD 位控制的第一个访问阶段（地址建立和地址保持阶段）不受 WAIT 影响，因此第一个访问阶段不会延长。

必须配置数据建立阶段，以便在内存事务结束前 4 个 fmc_ker_ck 周期检测到 WAIT。必须考虑以下情况：

1. 内存发出的 WAIT 信号和 NOE/NWE 信号对齐：

$$DATAST \geq (4 \times FMCCLK) + max_wait_assertion_time$$

2. 内存发出的 WAIT 信号和 NEx 对齐（或者 NOE/NWE 信号不翻转）：

   如果

$$max_wait_assertion_time > address_phase + hold_phase$$

   那么：

$$DATAST \geq (4 \times FMCCLK) + (max_wait_assertion_time - address_phase - hold_phase)$$

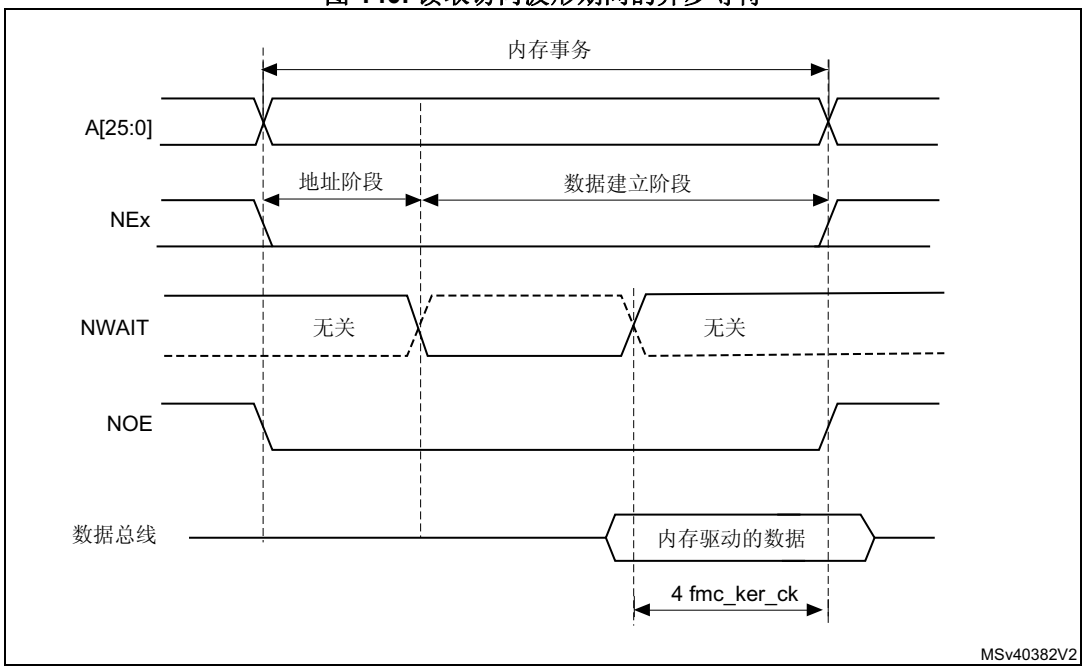
   否则

$$DATAST \geq 4 \times FMCCLK$$

   其中，max_wait_assertion_time 是在 NEx/NOE/NWE 变为低电平后内存使能 WAIT 信号所花费的最长时间。

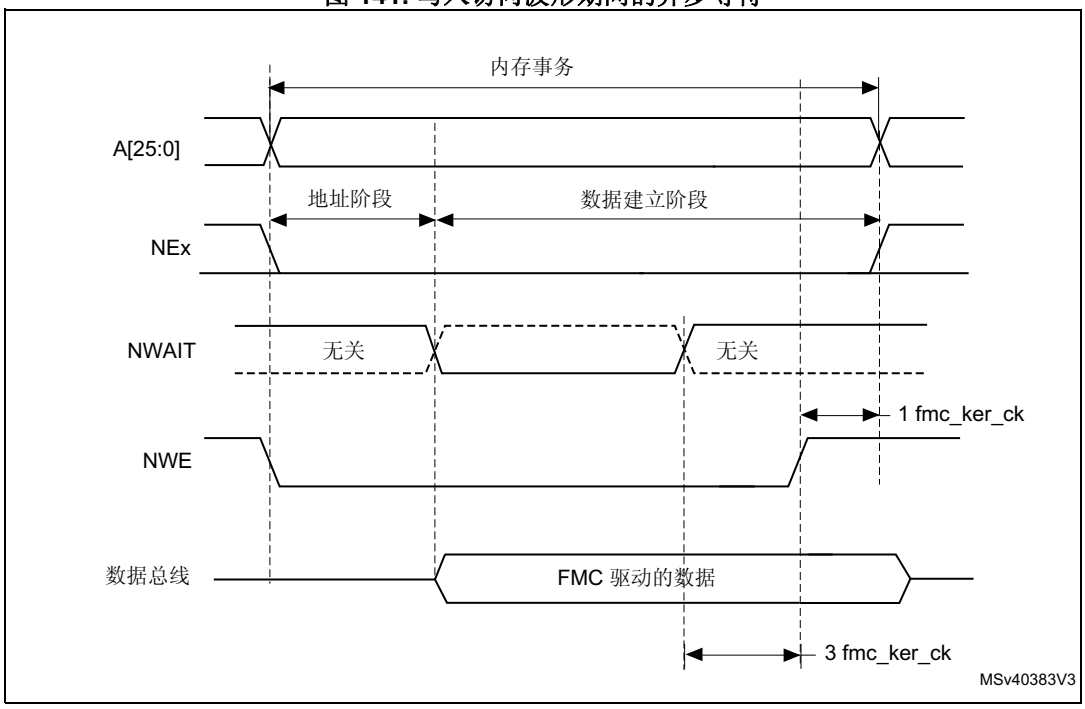
图 140 和*图 141* 显示了异步内存释放 WAIT 之后，在内存访问阶段增加的 fmc_ker_ck 时钟周期的个数（与上述情况无关）。

图 140. 读取访问波形期间的异步等待



1. NWAIT 极性取决于 FMC_BCRx 寄存器中的 WAITPOL 位设置。

图 141. 写入访问波形期间的异步等待



1. NWAIT 极性取决于 FMC_BCRx 寄存器中的 WAITPOL 位设置。

CellularRAM™ (PSRAM) 刷新管理

CellularRAM™ 不允许片选信号 (NE) 保持低电平的时间长于内存中指定的 t_{CEM} 时间。该时间可在 FMC_PCSCNTR 寄存器中设定。它定义了同步模式下 PSRAM 读取访问或写入访问的 NE 低电平脉冲的最大持续时间（以 fmc_ker_ck 周期数表示）。

22.8.5 同步事务

内存时钟 FMC_CLK 是 fmc_ker_ck 的约数。它取决于 CLKDIV 的值。

NOR Flash 指定了从 NADV 使能到 FMC_CLK 高电平的最短时间。为了符合这一限制，FMC 不会在同步访问的第一个内部时钟周期内（NADV 使能之前）将时钟发到内存。这样可以确保内存时钟的上升沿出现在 NADV 低脉冲的中间。

数据延迟与 NOR 内存延迟

数据延迟是对数据进行采样之前需要等待的周期数。DATLAT 的值必须与 NOR Flash 配置寄存器中指定的延迟值一致。当数据延迟计数中 NADV 为低电平时，FMC 不会计入时钟周期。

小心： 一些 NOR Flash 将 NADV 低电平周期计入数据延迟计数，这样 NOR Flash 延迟和 FMC DATLAT 参数之间的确切关系可以是以下任一种：

- NOR Flash 延迟 = (DATLAT + 2) 个 CLK 时钟周期
- 或 NOR Flash 延迟 = (DATLAT + 3) 个 CLK 时钟周期

近来有一些内存会在延迟阶段使能 NWAIT。在这种情况下，可以将 DATLAT 设置为最小值。然后，FMC 会对数据进行采样，并且等待足够长的时间来评估数据是否有效。这样，FMC 就能检测到内存存在延迟的时间，从而处理真实数据。

其他内存不会在延迟期间使能 NWAIT。在这种情况下，必须正确设置 FMC 和内存的延迟，否则可能会将无效数据误用为有效数据，或者在内存访问初始阶段丢失有效数据。

单次突发传输

当所选存储区域配置为同步 Burst 模式时，例如，如果向 16 位内存请求了一个单次突发事务，则 FMC 会执行长度为 1 的突发事务（如果 AXI 传输为 16 位）或者长度为 2 的突发事务（如果 AXI 传输为 32 位），然后在最后一个数据选通时禁止片选信号。

与异步读取操作相比，就周期而言这并不是最有效的传输方法。但是，随机异步读取需要先重新编程内存访问模式，这样总时间会更长。

越过 CellularRAM™ 1.5 的边界页

CellularRAM™ 1.5 不允许突发访问越过页边界。根据内存页大小配置 FMC_BCR1 寄存器中的 CPSIZE 位来达到内存页大小时，FMC 会自动分离突发访问。

等待管理

对于同步 NOR Flash，会在配置的延迟周期（相当于 (DATLAT+2) 个 FMC_CLK 时钟周期）之后对 NWAIT 进行评估。

如果 NWAIT 有效（WAITPOL = 0 时为低电平，WAITPOL = 1 时为高电平），会插入等待周期，直到 NWAIT 无效（WAITPOL = 0 时为高电平，WAITPOL = 1 时为低电平）。

当 NWAIT 无效时，数据将立即（位 WAITCFG = 1）或在下一个时钟边沿（位 WAITCFG = 0）被视为有效。

通过 NWAIT 信号插入等待周期期间，控制器会继续将时钟脉冲发送到内存，保持片选和输出使能信号有效。但不将数据视为有效。

Burst 模式下，NOR Flash NWAIT 信号有两种时序配置：

- Flash 在等待周期之前一个数据周期发出 NWAIT 信号（复位后的默认值）。
- Flash 在等待周期期间发出 NWAIT 信号

FMC 支持这两种 NOR Flash 等待周期配置，通过 FMC_BCRx 寄存器的 WAITCFG 位（x = 0 到 3）针对每个片选进行配置。

图 142. 等待配置波形

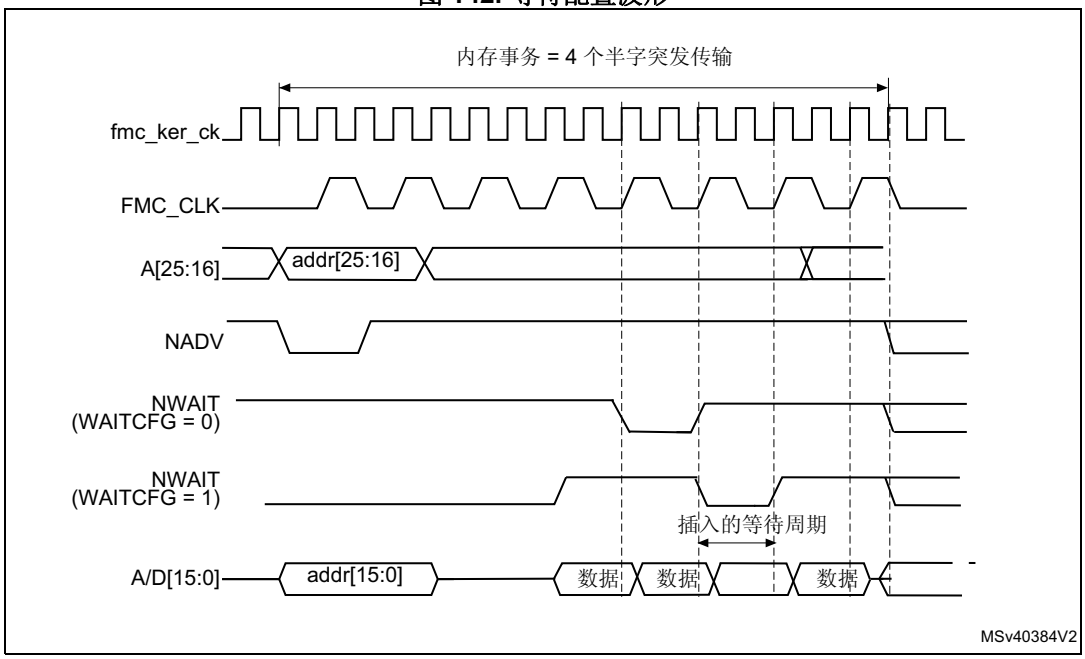


图 143. 同步复用读取模式波形 - NOR、PSRAM (CRAM)

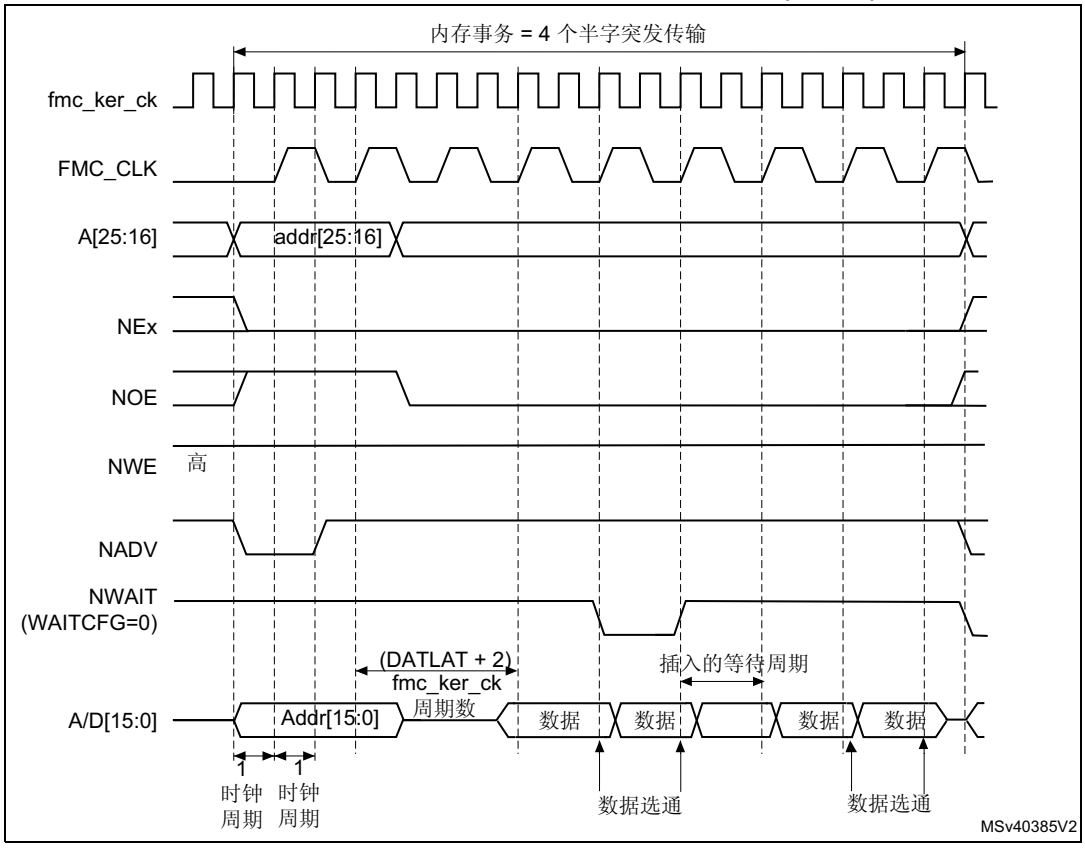


表 156. FMC_BCRx 位字段（同步复用读取模式）

位号	位名	要设置的值
31	FMCEN	0x1
30-24	保留	0x000
23:22	NBLSET[1:0]	根据需要进行设置
21	保留	0x0
20	CCLKEN	根据需要进行设置
19	CBURSTRW	根据需要进行设置
18:16	CPSIZE	根据需要进行设置（使用 CRAM 1.5 时为 0x1）
15	ASYNCWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	在内存支持该特性的情况下置 1，否则保持清零状态
12	WREN	对同步读取没有影响
11	WAITCFG	是否置 1 视内存情况而定
10	保留	0x0
9	WAITPOL	是否置 1 视内存情况而定
8	BURSTEN	0x1

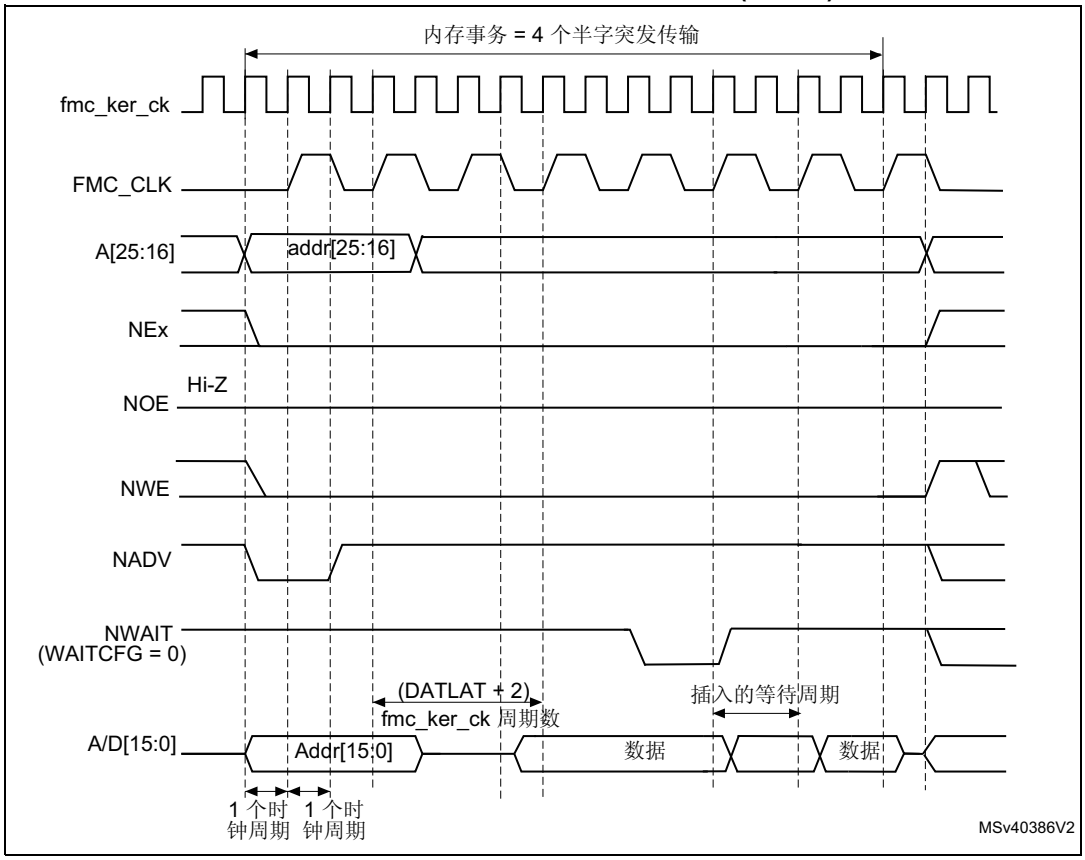
表 156. FMC_BCRx 位字段（同步复用读取模式）（续）

位号	位名	要设置的值
7	保留	0x1
6	FACCEN	在内存支持的情况下置 1 (NOR Flash)
5-4	MWID[1:0]	根据需要进行设置
3-2	MTYP[1:0]	0x1 或 0x2
1	MUXEN	根据需要进行设置
0	MBKEN	0x1

表 157. FMC_BTRx 位字段（同步复用读取模式）

位号	位名	要设置的值
31:30	DATAHLD	无关
29:28	ACCMOD	0x0
27-24	DATLAT	数据延迟
27-24	DATLAT	数据延迟
23-20	CLKDIV	FMC_CLK 分频比
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	无关
7-4	ADDHLD	无关
3-0	ADDSET	无关

图 144. 同步复用写入模式波形 - PSRAM (CRAM)



1. 内存必须提前一个周期发出 NWAIT 信号，相应地 WAITCFG 必须编程为 0。

表 158. FMC_BCRx 位字段（同步复用写入模式）

位号	位名	要设置的值
31	FMCEN	0x1
30-24	保留	0x000
23:22	NBLSET[1:0]	根据需要进行设置
21	保留	0x0
20	CCLKEN	根据需要进行设置
19	CBURSTRW	根据需要进行设置
18:16	CPSIZE	根据需要进行设置（使用 CRAM 1.5 时为 0x1）
15	ASYNCWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	在内存支持该特性的情况下置 1，否则保持清零状态。
12	WREN	0x1
11	WAITCFG	0x0
10	保留	0x0
9	WAITPOL	是否置 1 视内存情况而定

表 158. FMC_BCRx 位字段（同步复用写入模式）（续）

位号	位名	要设置的值
8	BURSTEN	0x1
7	保留	0x1
6	FACCEN	根据内存支持情况进行设置
5-4	MWID[1:0]	根据需要进行设置
3-2	MTYP[1:0]	0x1
1	MUXEN	根据需要进行设置
0	MBKEN	0x1

表 159. FMC_BTRx 位字段（同步复用写入模式）

位号	位名	要设置的值
31:30	DATAHLD	无关
29:28	ACCMOD	0x0
27-24	DATLAT	数据延迟
23-20	CLKDIV	FMC_CLK 分频比
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	无关
7-4	ADDHLD	无关
3-0	ADDSET	无关

22.8.6 NOR/PSRAM 控制寄存器

外设寄存器必须按字（32 位）进行访问

SRAM/NOR Flash 存储区域 x 的片选控制寄存器 (FMC_BCRx)

SRAM/NOR flash chip-select control register for memory region x

偏移地址：0x00 + 0x08 * (x - 1)（x = 1 到 4）

复位值：0x0000 30DB、0x0000 30D2、0x0000 30D2、0x0000 30D2

该寄存器包含每个存储区域的控制信息，用于 SRAM、PSRAM、FRAM 和 NOR Flash。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FMCEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBLSET[1:0]		Res.	CCLKEN	CBURSTRW	CPSIZE[2:0]		
rw								rw	rw		rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID[1:0]		MTYP[1:0]		MUXEN	MBKEN
rw	rw	rw	rw	rw		rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31 **FMCEN**：FMC 使能 (FMC enable)

该位用于使能/禁止 FMC。

0：禁止 FMC

1：使能 FMC

注意：FMC_BCR2 寄存器的 FMCEN 位为“无关”位。只能通过 FMC_BCR1 寄存器使能。

位 30:24 保留，必须保持复位值。

位 23:22 **NBLSET[1:0]**：字节通道 (NBL) 建立 (Byte lane (NBL) setup)

这些位配置了从 NBLx 低电平到片选 NEx 低电平的 NBL 设置时间。

00：NBL 建立时间为 0 个 fmc_ker_ck 时钟周期。

01：NBL 建立时间为 1 个 fmc_ker_ck 时钟周期。

10：NBL 建立时间为 2 个 fmc_ker_ck 时钟周期。

11：NBL 建立时间为 3 个 fmc_ker_ck 时钟周期。

位 21 保留，必须保持复位值。

位 20 **CCLKEN**：连续时钟使能 (Continuous clock enable)

该位可使能向外部内存设备输出 FMC_CLK 时钟。

0：仅在同步内存访问（读/写事务）期间生成 FMC_CLK。FMC_CLK 时钟分频比由 FMC_BCRx 寄存器中配置的 CLKDIV 值指定（复位后的默认值）。

1：异步和同步访问期间连续生成 FMC_CLK。CCLKEN 置 1 将激活 FMC_CLK 时钟。

注意：仅通过 FMC_BCR1 寄存器的 CCLKEN 位使能 FMC_CLK 连续时钟。

当 CCLKEN 位置 1 时，FMC_BCR2 到 4 寄存器的 CCLKEN 位为“无关”位。

使能 FMC_CLK 连续时钟时，必须将设备 1 配置为同步模式（即，必须将 BURSTEN 和/或 CBURSTRW 置 1），并且必须通过 FMC_BTR1 寄存器中的 CLKDIV 指定 FMC_CLK 时钟分频比。所有配置为同步模式的设备共用同一个时钟，该时钟的分频比取决于 FMC_BTR1 中的 CLKDIV。FMC_BTR2 到 4 和 FMC_BWTR1 到 4 寄存器中的 CLKDIV 不起作用。

位 19 **CBURSTRW**：写入突发使能 (Write burst enable)

PSRAM (CRAM) 在 Burst 模式下工作时，该位可使能同步写访问。同步读取访问的使能位为 FMC_BCRx 寄存器中的 BURSTEN 位。

0：始终在异步模式下进行写入操作

1：在同步模式下进行写入操作。

位 18:16 **CPSIZE[2:0]**：CRAM 页大小 (CRAM page size)

这些位用于 CellularRAM™ 1.5，CellularRAM™ 1.5 不允许突发访问越过页之间的地址边界。如果配置这些位，FMC 会在达到内存页大小后自动分离突发访问（请参见内存数据手册了解页大小）。

000：跨页边界时不进行突发分割（复位后的默认值）。

001：128 字节

010：256 字节

011：512 字节

100：1024 字节

其他值：保留。

位 15 **ASYNCWAIT**：异步传输期间的等待信号 (Wait signal during asynchronous transfers)

该位用于使能/禁止 FMC 在异步传输期间使用等待信号。

0：异步传输期间不考虑 NWAIT 信号（复位后的默认值）

1：异步传输期间考虑 NWAIT 信号

位 14 **EXTMOD**：扩展模式使能 (Extended mode enable)

该位用于使能 FMC_BWTR 寄存器中异步访问的写入时序编程，进而使读取和写入操作采用不同时序。

0：不考虑 FMC_BWTR 寄存器中的值（复位后的默认值）

1：考虑 FMC_BWTR 寄存器中的值

注意：如果禁止扩展模式，FMC 可以在模式 1 或模式 2 下运行，如下所述：

- *当选择 SRAM/PSRAM 内存类型时，模式 1 为默认模式（MTYP[1:0] = 0x0 或 0x01）*
- *当选择 NOR 内存类型时，模式 2 为默认模式 (MTYP[1:0] = 0x10)。*

位 13 **WAITEN**：等待使能位 (Wait enable bit)

该位可使能/禁止在同步模式下访问内存时通过 NWAIT 信号插入等待周期。

0：禁止 NWAIT 信号（不考虑其电平，不在配置过的 Flash 延迟周期后插入等待周期）

1：使能 NWAIT 信号（考虑其电平，如果使能，在配置过的延迟周期后插入等待周期）（复位后的默认值）

位 12 **WREN**：写入使能位 (Write enable bit)

该位指示是使能还是禁止 FMC 对存储区域的写操作：

0：禁止 FMC 对存储区域的写操作。报告 AXI 从错误。

1：使能 FMC 对存储区域的写操作（复位后的默认值）。

位 11 **WAITCFG**：等待时序配置 (Wait timing configuration)

NWAIT 信号指示内存中的数据是否有效，或者在同步模式下访问内存时是否必须插入等待周期。该配置位决定内存是在等待周期之前的一个时钟周期还是等待周期期间使能 NWAIT：

0：NWAIT 信号在等待周期之前的一个数据周期有效（复位后的默认值），

1：NWAIT 信号在等待周期期间有效（不适用于 PSRAM）。

位 10 保留，必须保持复位值。

位 9 **WAITPOL**：等待信号极性位 (Wait signal polarity bit)

该位定义同步或异步模式下使用的内存的等待信号极性：

0：NWAIT 低电平有效（复位后的默认值），

1：NWAIT 高电平有效。

位 8 **BURSTEN**：突发使能位 (Burst enable bit)

该位可使能/禁止同步读取访问。该位仅对 Burst 模式下工作的同步内存有效：

0：禁止 Burst 模式（复位后的默认值）。在异步模式下进行读取访问。

1：使能 Burst 模式。在同步模式下进行读取访问。

位 7 保留，必须保持复位值。

位 6 **FACCEN**：Flash 硬件复位 (Flash memory hardware reset)

该位用于使能通过专用输出引脚对 Flash 进行硬件复位。

0：相应的 NOR Flash 复位输出为低电平

1：相应的 NOR Flash 复位输出为高电平（复位后的默认值）

位 5:4 **MWID[1:0]**：内存数据总线宽度 (Memory data bus width)

该位定义外部内存设备宽度，对所有类型的内存均有效。

00：8 位

01：16 位（复位后的默认值）

10：保留。

11：保留。

位 3:2 **MTYP[1:0]**：内存类型 (Memory type)

该位定义与相应存储区域相连的外部内存类型：

00：SRAM/FRAM（对于存储区域 2...4，复位后的默认值）

01：PSRAM (CRAM)/FRAM

10：NOR Flash/OneNAND Flash（对于存储区域 1，复位后的默认值）

11：保留。

位 1 **MUXEN**：地址/数据复用使能位 (Address/data multiplexing enable bit)

该位置 1 时，地址和数据值在数据总线上复用，仅对 NOR 和 PSRAM 内存有效：

0：地址/数据非复用

1：地址/数据在数据总线上复用（复位后的默认值）

位 0 **MBKEN**：存储区域使能位 (Memory region enable bit)

该位用于使能存储区域。复位后使能存储区域 1，其他存储区域均禁止。访问禁止的存储区域会引起 AXI 总线上的错误。

0：禁止相应的存储区域

1：使能相应的存储区域

SRAM/NOR Flash 存储区域 x 的片选时序寄存器 (FMC_BTRx)

SRAM/NOR flash chip-select timing registers for memory region x

偏移地址：0x04 + 0x8 * (x - 1)（x = 1 到 4）

复位值：0x0FFF FFFF

该寄存器包含每个存储区域的控制信息，用于 SRAM、PSRAM 和 NOR Flash。如果 FMC_BCRx 寄存器中的 EXTMOD 位置 1，该寄存器将和另外一个寄存器配合来配置写入和读取访问，也就是说有 2 个寄存器可用：一个用于配置读取访问（该寄存器），另一个用于配置写入访问（FMC_BWTRx 寄存器）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATAHLD[1:0]		ACCMOD[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAST[7:0]								ADDHLD[3:0]				ADDSET[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 **DATAHLD[1:0]**：数据保持阶段的持续时间 (Data Hold phase duration)

通过软件写入这些位可定义以 fmc_ker_ck 周期表示的*数据保持*阶段持续时间（参见*图 140* 到*图 141*），适用于异步访问模式：

对于读取/写入访问：

00：DATAHLD 阶段持续时间 = fmc_ker_ck 时钟周期 * 0（读取访问的默认值）/1（写入访问的默认值）

01：DATAHLD 阶段持续时间 = fmc_ker_ck 时钟周期 * 1（读取访问）/2（写入访问）

10：DATAHLD 阶段持续时间 = fmc_ker_ck 时钟周期 * 2（读取访问）/3（写入访问）

11：DATAHLD 阶段持续时间 = fmc_ker_ck 时钟周期 * 3（读取访问）/4（写入访问）

位 29:28 **ACCMOD[1:0]**：访问模式 (Access mode)

这些位指定异步访问模式，如时序图所示。仅当 FMC_BCRx 寄存器中的 EXTMOD 位为 1 时才考虑这些位。

00：访问模式 A

01：访问模式 B

10：访问模式 C

11：访问模式 D

位 27:24 **DATLAT[3:0]**：同步内存的数据延迟 (Data latency for synchronous memory)（请参见位说明下的注释）

对于使能了读/写 Burst 模式（BURSTEN/CBURSTRW 位置 1）的同步访问，这些位定义读写首个数据前要发送给内存的内存时钟周期数 (+2)。

该时序参数以 FMC_CLK 周期而非 fmc_ker_ck 周期表示。

该值与异步访问模式无关。

0000：首次突发访问时，2 个 FMC_CLK 时钟周期的数据延迟

1111：首次突发访问时，17 个 FMC_CLK 时钟周期的数据延迟（复位后的默认值）

位 23:20 **CLKDIV[3:0]**：FMC_CLK 信号的时钟分频比 (Clock divide ratio (for FMC_CLK signal))

这些位定义 FMC_CLK 时钟输出信号周期，以 fmc_ker_ck 周期数表示：

0000：保留。

0001：FMC_CLK 周期 = 2 × fmc_ker_ck 周期

0010：FMC_CLK 周期 = 3 × fmc_ker_ck 周期

1111：FMC_CLK 周期 = 16 × fmc_ker_ck 周期（复位后的默认值）

在异步模式下访问 NOR Flash、SRAM 和 PSRAM 时，该值为无关值。

注意：有关 FMC_CLK *分频比公式的信息，请参见第 22.8.5 节：同步事务。*

位 19:16 **BUSTURN[3:0]**：总线周转阶段的持续时间 (Bus turnaround phase duration)

通过软件写入这些位可在当前读取/写入事务的结尾与同一存储区域的下一事务之间添加延迟。该延迟可以匹配连续事务之间的最短时间（t_{EHEL}，由 NEx 高电平变为 NEx 低电平的时间）以及内存在读取访问后释放数据总线所需的最长时间（t_{EHQZ}，芯片使能高电平到输出高阻态所经过的时间）。对于模式 D 和复用模式，建议使用此延迟。对于非复用内存，总线周转延迟可设置为满足以下条件的最小值：

(BUSTRUN + 1)fmc_ker_ck 周期 ≥ $t_{EHELmin}$ max ($t_{EHELmin}$, $t_{EHQZmax}$)

对于 FRAM 内存，总线周转延迟必须配置为匹配最小预充电时间 (t_{PC})。该延迟插入到同一存储区域的所有连续事务（读/读、写/写、读/写和写/读）之间以匹配 t_{PC} 内存时序，片选会在连续访问之间翻转。

(BUSTRUN + 1) fmc_ker_ck 周期 ≥ t_{PCmin}

0000：BUSTURN 阶段的持续时间 = 增加 1 个 fmc_ker_ck 时钟周期

...

1111：BUSTURN 阶段的持续时间 = 增加 16 个 fmc_ker_ck 时钟周期（复位后的默认值）

位 15:8 **DATAST[7:0]**：数据阶段的持续时间 (Data-phase duration)

通过软件写入这些位可定义数据阶段的持续时间（参见*图 140* 到*图 141*），适用于异步访问模式：

0000 0000：保留。

0000 0001：DATAST 阶段的持续时间 = 1 个 fmc_ker_ck 时钟周期

0000 0010：DATAST 阶段的持续时间 = 2 个 fmc_ker_ck 时钟周期

...

1111 1111：DATAST 阶段的持续时间 = 255 个 fmc_ker_ck 时钟周期（复位后的默认值）

有关每种内存类型和访问模式数据阶段持续时间的信息，请参见相应图片（*图 140* 到*图 141*）。

模式 1 示例，写入访问，DATAST=1：

数据阶段的持续时间 = DATAST+1 = 2 个 fmc_ker_ck 时钟周期。

注意：在同步访问中，该值为无关值。

位 7:4 **ADDHLD[3:0]**：地址保持阶段的持续时间 (Address-hold phase duration)

通过软件写入这些位可定义*地址保持*阶段的持续时间（参见*图 136* 到*图 139*），适用于模式 D 或复用访问：

0000：保留。

0001：ADDHLD 阶段的持续时间 = 1 个 fmc_ker_ck 时钟周期

0010：ADDHLD 阶段的持续时间 = 2 个 fmc_ker_ck 时钟周期

...

1111：ADDHLD 阶段的持续时间 = 15 个 fmc_ker_ck 时钟周期（复位后的默认值）

有关每种访问模式地址保持阶段持续时间的信息，请参见相应图片（*图 136* 到*图 139*）。

注意：在同步访问中不使用该值，地址保持阶段始终持续 1 个内存时钟周期。

位 3:0 **ADDSET[3:0]**：地址建立阶段的持续时间 (Address setup phase duration)

通过软件写入这些位可定义*地址建立*阶段的持续时间（请参见*图 140* 到*图 141*），适用于 SRAM、ROM 和异步 NOR Flash 访问模式：

0000：ADDSET 阶段的持续时间 = 0 个 fmc_ker_ck 时钟周期

...

1111：ADDSET 阶段的持续时间 = 15 个 fmc_ker_ck 时钟周期（复位后的默认值）

有关每种访问模式地址建立阶段持续时间的信息，请参见相应图片（请参见*图 140* 到*图 141*）。

注意：在同步访问中，该值为无关值。

复用模式或模式 D 下，ADDSET 的最小值为 1。

注： *PSRAM (CRAM) 由于内部刷新而导致数据延时时间长度不确定。因此，这些内存会在整个延迟阶段发送 NWAIT 信号，以便按照需要延长延迟。*

对于 PSRAM (CRAM)，位字段 DATLAT 必须设置为 0，这样 FMC 会立即退出延迟阶段，开始对内存中的 NWAIT 采样，然后在内存准备就绪后开始读取或写入操作。

此方法也适用于最新一代的同步 Flash，同早期 Flash 不同的是，此类 Flash 会发送 NWAIT 信号（查看相应 Flash 的数据手册）。

SRAM/NOR Flash 存储区域 x 的写入时序寄存器 (FMC_BWTRx)

SRAM/NOR-flash write timing registers for memory region x

偏移地址：0x104 + 8 * (x - 1)（x = 1 到 4）

复位值：0x000F FFFF

该寄存器包含每个存储区域的控制信息，用于 SRAM、FRAM、PSRAM 和 NOR Flash。当 FMC_BCRx 寄存器中的 EXTMOD 位置 1 时，该寄存器将处于有效状态，可以进行写入访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATAHLD[1:0]		ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN[3:0]			
rw	rw	rw	rw									rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAST[7:0]								ADDHLD[3:0]				ADDSET[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 **DATAHLD[1:0]**：数据保持阶段的持续时间 (Data Hold phase duration)

通过软件写入这些位可定义以 fmc_ker_ck 周期表示的*数据保持*阶段持续时间（参见*图 141*），适用于异步写入访问模式：

00：DATAHLD 阶段的持续时间 = 1 个 fmc_ker_ck 时钟周期（默认值）

01：DATAHLD 阶段的持续时间 = 2 个 fmc_ker_ck 时钟周期

10：DATAHLD 阶段的持续时间 = 3 个 fmc_ker_ck 时钟周期

11：DATAHLD 阶段的持续时间 = 4 个 fmc_ker_ck 时钟周期

位 29:28 **ACCMOD[1:0]**：访问模式 (Access mode)。

这些位指定异步访问模式，如下一个时序图所示。仅当 FMC_BCRx 寄存器中的 EXTMOD 位置 1 时，这些位才有效。

00：访问模式 A

01：访问模式 B

10：访问模式 C

11：访问模式 D

位 27:20 保留，必须保持复位值。

位 19:16 **BUSTURN[3:0]**：总线周转阶段的持续时间 (Bus turnaround phase duration)

通过软件写入这些位可在当前写入事务的结尾与同一存储区域的下一事务之间添加延迟。

对于 FRAM 内存，总线周转延迟必须配置为匹配最小预充电时间 (t_{PC})。总线周转延迟插入到同一存储区域的所有连续事务（读/读、写/写、读/写和写/读）之间，片选会在连续访问之间翻转。

(BUSTRUN + 1) fmc_ker_ck 周期 ≥ t_{PCmin}

0000：BUSTURN 阶段的持续时间 = 增加 1 个 fmc_ker_ck 时钟周期

...

1111：BUSTURN 阶段的持续时间 = 增加 16 个 fmc_ker_ck 时钟周期（复位后的默认值）

位 15:8 **DATAST[7:0]**：数据阶段的持续时间 (Data-phase duration)。

通过软件写入这些位可定义数据阶段的持续时间（参见*图 140* 到*图 141*），适用于异步 SRAM、PSRAM 和 NOR Flash 访问模式：

0000 0000：保留。

0000 0001：DATAST 阶段的持续时间 = 1 个 fmc_ker_ck 时钟周期

0000 0010：DATAST 阶段的持续时间 = 2 个 fmc_ker_ck 时钟周期

...

1111 1111：DATAST 阶段的持续时间 = 255 个 fmc_ker_ck 时钟周期（复位后的默认值）

位 7:4 **ADDHLD[3:0]**：地址保持阶段的持续时间 (Address-hold phase duration)。

通过软件写入这些位可定义地址保持阶段的持续时间（参见*图 138* 到*图 139*），适用于异步复用访问：

0000：保留。

0001：ADDHLD 阶段的持续时间 = 1 个 fmc_ker_ck 时钟周期

0010：ADDHLD 阶段的持续时间 = 2 个 fmc_ker_ck 时钟周期

...

1111：ADDHLD 阶段的持续时间 = 15 个 fmc_ker_ck 时钟周期（复位后的默认值）

注意：在同步 NOR Flash 访问模式下，该值不使用，因为地址保持阶段的持续时间始终是 1 个 Flash 时钟周期。

位 3:0 **ADDSET[3:0]**：地址建立阶段的持续时间 (Address setup phase duration)。

通过软件写入这些位可定义以 fmc_ker_ck 周期表示的地址建立阶段持续时间（请参见*图 140* 到*图 141*），适用于异步访问模式：

0000：ADDSET 阶段的持续时间 = 0 个 fmc_ker_ck 时钟周期

...

1111：ADDSET 阶段的持续时间 = 15 个 fmc_ker_ck 时钟周期（复位后的默认值）

注意：在同步访问模式下，该值不使用，因为地址建立阶段的持续时间始终是 1 个 Flash 时钟周期。复用模式下，ADDSET 的最小值为 1。

PSRAM 片选计数器寄存器 (FMC_PCSCNTR)

PSRAM chip select counter register

偏移地址：0x20

复位值：0x0000 0000

该寄存器包含同步模式的 PSRAM 片选计数器值。片选计数器对所有存储区域通用，并且可以在每个存储区域上单独使能。在 PSRAM 读/写访问期间，该值装载到定时器中，定时器在 NE 信号保持低电平时使用 fmc_ker_ck 进行递减。当定时器达到 0 时，PSRAM 控制器分离当前访问，翻转 NE 以允许 PSRAM 设备刷新，并重新开始新的访问。编程的计数器值可确保最大 NE 脉冲宽度 (t_{CEM})，如为 PSRAM 设备所指定的值。每次通过 NE 从高电平变为低电平来启动新访问时，该计数器都会重新装载并开始递减。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNTB4EN	CNTB3EN	CNTB2EN	CNTB1EN
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSCOUNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:20 保留，必须保持复位值。

位 19 **CNTB4EN**：计数器存储区域 4 使能 (Counter memory region 4 enable)
该位用于为 PSRAM/NOR 存储区域 4 使能 CS 计数器。
0：为存储区域 4 禁止计数器
1：为存储区域 4 使能计数器

位 18 **CNTB3EN**：计数器存储区域 3 使能 (Counter memory region 3 enable)
该位用于为 PSRAM/NOR 存储区域 3 使能 CS 计数器。
0：为存储区域 3 禁止计数器
1：为存储区域 3 使能计数器

位 17 **CNTB2EN**：计数器存储区域 2 使能 (Counter memory region 2 enable)
该位用于为 PSRAM/NOR 存储区域 2 使能 CS 计数器。
0：为存储区域 2 禁止计数器
1：为存储区域 2 使能计数器

位 16 **CNTB1EN**：计数器存储区域 1 使能 (Counter memory region 1 enable)
该位用于为 PSRAM/NOR 存储区域 1 使能 CS 计数器。
0：为存储区域 1 禁止计数器
1：为存储区域 1 使能计数器

位 15:0 **CSCOUNT[15:0]**：片选 (CS) 计数器 (Chip Select (CS) counter)。
通过软件写入这些位可定义同步访问的的最大片选低电平脉冲持续时间（以 fmc_ker_ck 周期数表示）。
如果编程值为 0，则禁止计数器。
fmc_ker_ck 时钟周期数与编程值的关系如下面的公式所示：
fmc_ker_ck 周期数 = (CLKDIV+1)*ROUNDUP(CSCOUNT[15:0]/(CLKDIV+1);0)

FMC 大小标识寄存器 (FMC_SIDR)

FMC size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识 (Size Identification)
这些位返回分配给 FMC 寄存器的存储区域的大小。

FMC 标识寄存器 (FMC_IPIDR)

FMC identification register

偏移地址：0x3F8

复位值：0x0014 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：FMC 标识符 (FMC Identifier)
这些位返回 FMC 标识符

FMC 版本触发器 (FMC_VERR)

FMC version register

偏移地址：0x3F4

复位值：0x0000 0011

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本 (Major revision)
这些位返回 FMC 主版本

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)
这些位返回 FMC 次版本

FMC 硬件配置寄存器 1 (FMC_HWCFGR1)

FMC hardware configuration register 1

偏移地址：0x3F0

复位值：0x2232 B011

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RA_LN2DPTH[3:0]				WR_LN2DPTH[3:0]				WD_LN2DPTH[3:0]				WA_LN2DPTH[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID_SIZE[3:0]				Res.	Res.	Res.	SDRAM_SEL	Res.	Res.	Res.	NAND_ECC	Res.	Res.	Res.	NAND_SEL
r	r	r	r				r				r				r

位 31:28 **RA_LN2DPTH[3:0]**：AXI 读取地址 FIFO 深度 (AXI read address FIFO depth)
这些位返回 AXI 读取地址 FIFO 深度。深度为 $2^{RA_LN2DPTH}$。

位 27:24 **WR_LN2DPTH[3:0]**：AXI 写入响应 FIFO 深度 (AXI write response FIFO depth)
这些位返回 AXI 写入响应 FIFO 深度。深度为 $2^{WR_LN2DPTH}$。

位 23:20 **WD_LN2DPTH[3:0]**：AXI 写入数据 FIFO 深度 (AXI write data FIFO depth)
这些位返回 AXI 写入数据 FIFO 深度。深度为 $2^{WD_LN2DPTH}$。

位 19:16 **WA_LN2DPTH[3:0]**：AXI 写入地址 FIFO 深度 (AXI write address FIFO depth)
这些位返回 AXI 写入地址 FIFO 深度。深度为 $2^{WA_LN2DPTH}$。

位 15:12 **ID_SIZE[3:0]**：AXI ID 宽度 (AXI ID width)
这些位返回 AXI ID 宽度。

位 11:9 保留，必须保持复位值。

位 8 **SDRAM_SEL**：SDRAM 控制器选择 (SDRAM controller selection)
0x0：不支持 SDRAM 控制器
0x1：支持 SDRAM 控制器

位 7:5 保留，必须保持复位值。

位 4 **NAND_ECC**：NAND ECC
0x0：汉明码（仅在不带定序器的传统模式下支持）
0x1：汉明码和 BCH 码

位 3:1 保留，必须保持复位值。

位 0 **NAND_SEL**：NAND 控制器选择 (NAND Controller Selection)
0x0：不支持 NAND 控制器
0x1：支持 NAND 控制器

FMC 硬件配置寄存器 2 (FMC_HWCFGR2)

FMC hardware configuration register 2

偏移地址：0x3EC

复位值：0x00DC 8762

基址仅在 FMC 包括相应的存储控制器时才可用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDRAM2_BASE[3:0]				SDRAM1_BASE[3:0]			
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAND_BASE[3:0]				SDRAM_RBASE[3:0]				NOR_BASE[3:0]				RD_LN2DPTH[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 保留，必须保持复位值。

位 23:20 **SDRAM2_BASE[3:0]**：SDRAM bank 2 基址 (SDRAM bank 2 base address)
当支持 SDRAM 控制器时，这些位返回 SDRAM bank 2 基址空间的位 [31:28]。当不支持 SDRAM 控制器时，这些位将被忽略。

位 19:16 **SDRAM1_BASE[3:0]**：SDRAM bank 1 基址 (SDRAM bank 1 base address)
当支持 SDRAM 控制器时，这些位返回 SDRAM bank 1 基址空间的位 [31:28]。当不支持 SDRAM 控制器时，这些位将被忽略。

位 15:12 **NAND_BASE[3:0]**：NAND 基址 (NAND base address)
这些位返回 NAND 基址空间的位 [31:28]

位 11:8 **SDRAM_RBASE[3:0]**：SDRAM 重映射基址 (SDRAM remap base address)
当支持 SDRAM 控制器时，这些位返回 SDRAM 重映射基址空间的位 [31:28]。当不支持 SDRAM 控制器时，这些位将被忽略。

位 7:4 **NOR_BASE[3:0]**：NOR 基址 (NOR base address)
这些位返回 NOR 基址空间的位 [31:28]

位 3:0 **RD_LN2DPTH[3:0]**：AXI 读取数据 FIFO 深度 (AXI read data FIFO depth)
这些位返回 AXI 读取数据 FIFO 深度。深度为 $2^{RD_LN2DPTH}$

22.9 NAND Flash 控制器

FMC 会生成适当的信号时序，以驱动 8 位和 16 位 NAND Flash

NAND 存储区域通过专用的寄存器配置（*第 22.9.9 节*）。可编程的内存参数包括访问时序（如*表 160* 所示）和 ECC 配置。

NAND 控制器支持以下特性：

- 可将页大小设定为 256、512、1024、2048、4096 或 8192 字节
- 可设定内存时序
- 可使用 BCH 码或汉明码设定纠错功能
- 命令定序器：一种用于页内或块内（大小可配置）的读取/写入访问的硬件加速器。该功能在执行 ECC 计算时很有用。
- 可使用命令定序器为 ECC 字节编程设定偏移地址

- 可在检测到 ECC 错误时以及在页或扇区传输完成时生成中断
- 支持多芯片封装。一次只能选择一颗芯片。

表 160. 可编程的 NAND Flash 访问参数

参数	功能	访问模式	单位	最小值	最大值
内存建立时间	命令使能前地址建立所需时钟周期 (fmc_ker_ck) 数	读/写	FMC 时钟周期 (fmc_ker_ck)	1	256
内存等待	命令使能的最小持续时间（按 fmc_ker_ck 时钟周期计）	读/写	FMC 时钟周期 (fmc_ker_ck)	1	256
内存保持	命令禁止后，必须保持地址（如果进行了写访问还需保持数据）的时钟周期 (fmc_ker_ck) 数	读/写	FMC 时钟周期 (fmc_ker_ck)	1	256
内存数据总线高阻态	开始进行写访问后，数据总线保持高阻状态期间的时钟周期 (fmc_ker_ck) 数	写	FMC 时钟周期 (fmc_ker_ck)	1	256

22.9.1 外部内存接口信号

下表列出了用于连接 NAND Flash 的典型信号。

注： *前缀“N”标识低电平有效的信号。*

8 位 NAND Flash

表 161. 8 位 NAND Flash

FMC 信号名称	I/O	功能
A[17]	O	NAND Flash 地址锁存使能 (ALE) 信号
A[16]	O	NAND Flash 命令锁存使能 (CLE) 信号
D[7:0]	I/O	8 位复用双向地址/数据总线
NCEx	O	片选（x = 1 和 2）
NOE(= NRE)	O	输出使能（内存信号名称：读取使能，NRE）
NWE	O	写入使能
NWAIT	I	输入 FMC 的 NAND Flash 就绪/繁忙 (RNB) 信号

注： *由于 FMC 能够管理足够多的地址周期，因此理论上不存在容量限制。*

16 位 NAND Flash

表 162. 16 位 NAND Flash

FMC 信号名称	I/O	功能
A[17]	O	NAND Flash 地址锁存使能 (ALE) 信号
A[16]	O	NAND Flash 命令锁存使能 (CLE) 信号
D[15:0]	I/O	16 位复用双向地址/数据总线
NCEx	O	片选（x = 1 和 2）
NOE(= NRE)	O	输出使能（内存信号名称：读取使能，NRE）
NWE	O	写入使能
NWAIT	I	输入 FMC 的 NAND Flash 就绪/繁忙 (RNB) 信号

注： *由于 FMC 能够管理足够多的地址周期，因此理论上不存在容量限制。*

22.9.2 NAND Flash 支持的内存和事务

表 163 介绍了所支持的设备、访问模式和事务。NAND Flash 控制器不允许（或不支持）的事务以灰色显示。

表 163. 支持的内存和事务

设备	模式	R/W	AXI 数据大小	内存数据大小	允许/不允许	备注
8 位 NAND	异步	R	8	8	Y	
	异步	W	8	8	Y	
	异步	R	16	8	Y	分为 2 次 FMC 访问
	异步	W	16	8	Y	分为 2 次 FMC 访问
	异步	R	32	8	Y	分为 4 次 FMC 访问
	异步	W	32	8	Y	分为 4 次 FMC 访问
	异步	R	64	8	Y	分为 8 次 FMC 访问
	异步	W	64	8	Y	分为 8 次 FMC 访问
16 位 NAND	异步	R	8	16	Y	
	异步	W	8	16	N	
	异步	R	16	16	Y	
	异步	W	16	16	Y	
	异步	R	32	16	Y	分为 2 次 FMC 访问
	异步	W	32	16	Y	分为 2 次 FMC 访问
	异步	R	64	16	Y	分为 4 次 FMC 访问
	异步	W	64	16	Y	分为 4 次 FMC 访问

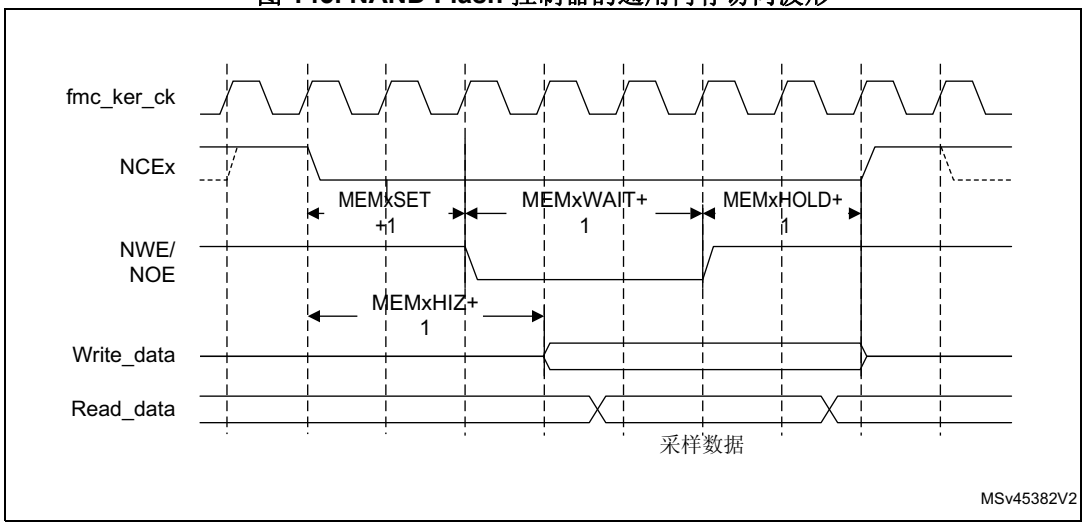
22.9.3 NAND Flash 的时序图

NAND Flash 区域通过以下一组寄存器进行管理：

- 控制寄存器：FMC_PCR
- 中断状态寄存器：FMC_SR
- ECC 寄存器：FMC_HECCR
- 通用内存空间的时序寄存器：FMC_PMEM
- 特性内存空间的时序寄存器：FMC_PATT

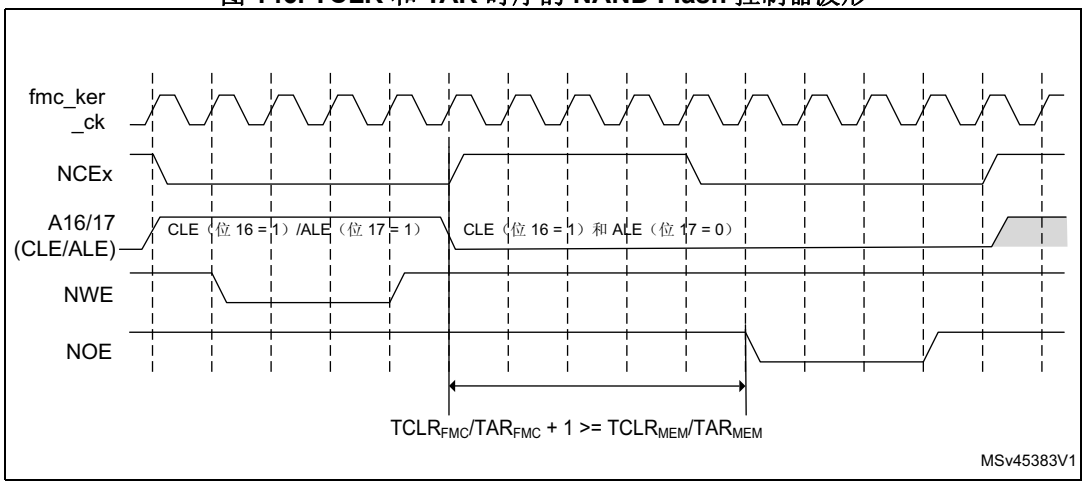
每个时序配置寄存器包含四个参数，其中三个参数用于定义 NAND Flash 的不同访问阶段的 fmc_ker_ck 周期数，另一个参数用于定义进行写访问时开始驱动数据总线的时序。*图 145* 介绍了通用内存访问的时序参数定义，特性内存空间的访问时序与此类似。

图 145. NAND Flash 控制器的通用内存访问波形



1. 进行写访问期间 NOE 保持高电平（无效）。进行读访问期间 NWE 保持高电平（无效）。
2. MEMxSET + MEMxHOLD 的编程值必须 > 1。

图 146. TCLR 和 TAR 时序的 NAND Flash 控制器波形



1. 进行写访问期间 NOE 保持高电平（无效）。进行读访问期间 NWE 保持高电平（无效）。

22.9.4 NAND Flash 操作

NAND Flash 设备的命令锁存使能 (CLE) 信号和地址锁存使能 (ALE) 信号由 FMC 的地址信号驱动。这意味着要向 NAND Flash 发送命令或地址，CPU 必须对其内存空间中的特定地址执行写操作。

从 NAND Flash 设备进行的典型页读取操作需要执行以下步骤：

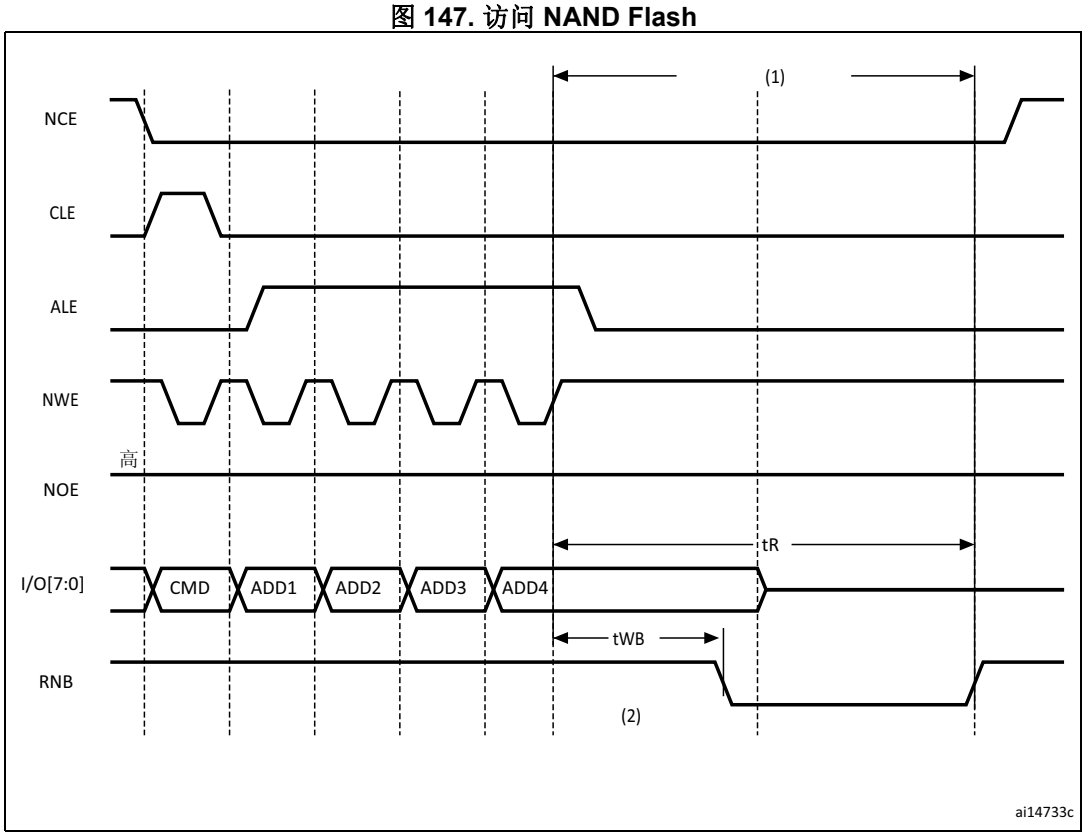
1. 根据 NAND Flash 的特性（PWID 位用于配置 NAND Flash 数据总线宽度，PWAITEN 根据需要置 1 或清零；有关时序配置，请参见*第 22.7.2 节：NAND Flash 地址映射*），配置 FMC_PCR 和 FMC_PMEM（对于某些设备，配置 FMC_PATT，请参见*第 22.9.5 节：NAND Flash 预等待功能*）寄存器，进而配置和使能 NAND 存储区域。
2. CPU 向通用内存空间执行字节写操作，此时数据字节值等于一个 Flash 命令字节（例如 Samsung NAND Flash 设备的字节为 0x00）。在写入选通期间（NWE 上为低电平脉冲），NAND Flash CLE 输入有效，因此将已写入的字节视为 NAND Flash 的一个命令。该命令由内存锁存后，随后进行页读取操作时，不需要再次写入该命令。
3. CPU 可通过在将五字节地址写入通用内存或特性空间后发送第二个读命令，来发送读取操作的起始地址。在写入选通期间（NWE 上为低电平脉冲），NAND FlashALE 输入有效，因此将已写入的字节视为读操取作的起始地址。借助特性内存空间，可使用另一种不同的 FMC 时序配置，实现某些 NAND Flash 所需的预等待功能（有关详细信息，请参见*第 22.9.5 节：NAND Flash 预等待功能*）。
4. 控制器在对同一个或另一个存储区域进行新访问之前，等待 NAND Flash 准备就绪（RNB 信号处于高电平）。
5. 然后 CPU 能够从通用内存空间执行字节读操作，进而按字节读取 NAND Flash 页（数据字段 + 备用字段）。
6. 无需 CPU 指令或地址写入操作即可读取下一个 NAND Flash 页。可采用以下三种不同的方式实现：
   - 执行步骤 5 中介绍的操作，
   - 通过重新开始步骤 3 中的操作随机访问一个新地址，
   - 通过重新开始步骤 2 向 NAND Flash 设备发送新命令。

注： *不支持非“CE 无关”NAND Flash。*

22.9.5 NAND Flash 预等待功能

NAND 控制器必须等待 RNB 置为低电平后才能发送下一个命令（请参见*图 147*）。该时序限制根据 t_{WB} 使用 FMC_PMEM 中的 MEMHOLD[7:0] 位字段或 FMC_PATT 中的 ATTHOLD[7:0] 进行设置。

图 147. 访问 NAND Flash



1. 对于 NAND Flash，可将 NCE 置为无效，因为 CE 无关，但本例中未考虑的非常早期的 NAND Flash 设备除外。
2. 通过使用 FMC_PATT 的 ATTHOLD[7:0] 位，可将 RNG 采样延迟到 t_{WB} 之后。例如，如果 fmc_ker_clk = 14 ns 且 ATTHOLD[7:0] = 7，则延迟为 8*14 = 112 ns。t_{WB} 最大值通常为 100 ns，设定的延迟必须大于 t_{WB} + 2*fmc_ker_clk。

任何对 NAND Flash 的 CPU 读取或写入访问，在 NWE 信号置为有效之后都有 (MEMHOLD[7:0] + 1) 个 fmc_ker_ck 周期的延迟。

必须确定 FMC_PATT 寄存器中的 ATTHOLD[7:0] 位字段以满足 t_{WB} 时序限制，并选择性地对 CPU 模式下的最后一个命令或地址字节应用更严格的时序限制。设置 ATTHOLD[7:0] 时，还必须考虑 NAND 定序器采样 RNB 所需的 2 * fmc_ker_ck 延迟。

22.9.6 NAND ECC 控制器

NAND Flash 在每个页的末尾包含一个备用存储区域（或 OOB）。该区域通常用于存储一些元数据（例如，使整个页失效的坏块标记信息）或用于错误管理功能。

FMC 支持在对 NAND Flash 读取/编程数据的过程中实时进行纠错码 (ECC) 计算。该机制可在软件处理 ECC 时减少主机 CPU 工作负载。

用户应用程序可以从以下两种算法中选择一种：

- 汉明码算法，每 256、512、1024、2048、4096 或 8192 个字节纠正 1 位错误并检测出 2 位错误。
- BCH（Bose、Chaudhuri 和 Hocquenghem）码，每 512 个字节纠正 4 位错误并检测出 8 位错误或者纠正 8 位错误并检测出 16 位错误。

两种 ECC 控制器的原理类似：每当 NAND Flash 存储区域处于活动状态时，ECC 模块均会监视 NAND Flash 数据总线和读/写信号（NCE 和 NWE）。ECC 控制器的操作如下：

- 当访问 NAND Flash 存储区域时，将锁存 D[7:0] 或 D[15:0] 总线上出现的数据并将其用于 ECC 计算。
- 当访问 NAND Flash 中的任何其他地址时，ECC 逻辑会进入空闲状态，不执行任何操作。因此，进行 ECC 计算时，用于定义 NAND Flash 命令或地址的写操作无效。

主机 CPU 或 DMA 对 NAND Flash 完成所需字节数的读取/写入操作后，将立即更新 ECC 结果寄存器。

汉明码

对 NAND Flash 执行读取或写入操作时，相应每 256、512、1024、2048、4096 或 8192 个字节，汉明码可纠正 1 位错误并检测出 2 位错误。包括计算行和列的奇偶校验。8 位和 16 位 NAND Flash 均支持该算法。

下面介绍了使用汉明码的页写入和读取序列。

使用汉明 ECC 计算来执行页编程：

1. 使能 FMC_PCR 寄存器中的 ECCEN 位。
2. 将数据写入 NAND Flash 页。在写入 NAND 页期间，ECC 控制器将计算 ECC 值。
3. 等待 ECC 码就绪（FMC_SR 寄存器中的 NWRF 置 1）。
4. 读取 FMC_HECCR 寄存器中所提供的 ECC 值，并将其存储到 NAND Flash 备用区域。
5. 清零 ECCEN 位并启动页编程。

使用汉明 ECC 计算来执行页读取：

1. 在读取 NAND 页之前，使能 FMC_PCR 寄存器中的 ECCEN 位。在读取 NAND 页期间，ECC 模块将计算 ECC 值。
2. 等待 ECC 码就绪（FMC_SR 寄存器中的 NWRF 置 1）。
3. 读取备用区域中所提供的 ECC 值，以及 FMC_HECCR 寄存器中所提供的 ECC 值（经过汉明控制器处理后）。
4. 如果两个 ECC 值相同，则无需纠正。如果两者不同，则会发生 ECC 错误，此时软件纠正程序会返回有关错误是否可纠正的信息，并尽可能进行纠正。

根据备用区域的大小，页可以划分成几个大小相同的扇区。这样可以增强每个页的纠错能力，因为每个扇区都可以纠正 1 个错误并检测出 2 个错误。

示例

对于具有 2 KB + 64 字节页的 NAND Flash，每个页可以划分为 8 个 256 字节的扇区。每个扇区需要 3 字节（22 位）的 ECC，因此整个页的备用区域需要存储 24 字节（8 * 3 字节）。对于整个 2 KB 页，最多可以纠正 8 个错误（相应地最多可以检测出 16 个错误），前提是每个扇区的错误少于 1 个错误（相应地少于 2 个错误）。

BCH（Bose、Chaudhuri 和 Hocquenghem）码

为了增强纠错能力，FMC 内置 BCH（Bose、Chaudhuri 和 Hocquenghem）编码器和解码器。

每个扇区可纠正 4 位错误并检测出 8 位错误或者纠正 8 位错误并检测出 16 位错误。BCH 编码器/解码器模块处理固定大小的扇区，每个扇区大小为 512 字节。因此，每个页必须划分为多个 512 字节的扇区。*表 164* 列出了每个扇区生成的奇偶校验字节数。对于 16 位 NAND Flash，增添了填充功能以将奇偶校验字节数对齐到偶数（4 位和 8 位 BCH 码分别为 8 字节和 14 字节）。

表 164. 每个扇区的 ECC 奇偶校验字节数

功能	BCH 码	
纠错	4 位	8 位
奇偶校验字节数	7 字节	13 字节

BCH 模块检测每个扇区的错误并返回对应的位位置。为了纠正错误，软件需要翻转存储在 FMC_BCHPBRx 寄存器中的位位置上提供的错误位。

BCH ECC 控制器支持 8 位和 16 位 NAND Flash。

在页编程操作期间，BCH 编码器为每个 512 字节的扇区生成 ECC 字节（7 或 13 字节），ECC 字节必须以一定的偏移量写入 NAND Flash 备用区域，该偏移量可编程设定，但必须考虑到备用区域开头存放有一些元数据。每个扇区都必须执行该操作。BCH 编码器不处理备用区域：元数据必须有自己的错误保护机制。

ECC 字节总数 (N) 如下面的公式所示：

$$N = \text{每扇区 ECC 字节数} \times \text{每页扇区数}$$

示例

对于具有 4096 字节页且每 512 字节可纠正 4 位错误的 8 位 NAND Flash，每页的扇区数量为 8 (4096 / 512)。因此，ECC 字节总数如下：

$$N = 7 \times 8 = 56 \text{ 字节}$$

用户应用程序必须确保备用区域足够大，以便存储元数据和奇偶校验字节。

下面介绍了使用 BCH 码的页编程和读取序列。

执行页编程：

1. 通过将 FMC_PCR 寄存器中的 WEN 位置 1 来使能写入访问。
2. 通过将 FMC_PCR 寄存器中的 ECCEN 位置 1 来使能 ECC。
3. 向 NAND Flash 页中的扇区写入 512 个数据字节。在写入扇区时，BCH 编码器将计算 ECC 值。
4. 通过监视 FMC_BCHISR 寄存器的 EPBRF 位，等待 BCH 编码器奇偶校验位可用。
5. 读取 FMC_BCHPBRx 寄存器中所提供的 ECC 值，并将其存储到 NAND Flash 备用区域。
6. 通过复位 FMC_PCR 寄存器中的 ECCEN 位来禁止 ECC。
7. 对所有要写入 NAND Flash 的扇区重复执行步骤 2 到 6。
8. 启动页编程。

执行页读取：

1. 通过将 FMC_PCR 寄存器中的 WEN 位复位来使能读取访问。
2. 通过将 FMC_PCR 寄存器中的 ECCEN 位置 1 来使能 ECC。
3. 读取 NAND 页扇区（512 字节）。
4. 从 NAND Flash 备用区域读取相应的 ECC 字节（7 或 13 字节）。
5. 在读取数据和奇偶校验位期间，计算校正子。然后启动错误位置搜索。
6. 等待解码错误就绪（FMC_BCHISR 寄存器中的 DERF 位置 1），然后从 FMC_BCHDSR0 和 FMC_BCHDSRx 寄存器中读取解码结果（检测到的错误（如果有）、是否可纠正、错误数量、位位置）。
7. 软件可基于解码结果纠正扇区中的错误（在可能的情况下）。
8. 通过复位 FMC_PCR 寄存器中的 ECCEN 位来禁止 ECC。
9. 对页的所有相关扇区重复执行步骤 2 到 7。

22.9.7 FMC 命令定序器

如前几节所述，CPU 可以执行 NAND Flash 页编程和读取。
对于命令和地址传输，还可以通过 FMC 命令定序器自动管理数据扇区编程/读取以及 ECC 字节编程/读取；对于数据传输，则通过 DMA 自动管理。

命令定序器旨在简化带 ECC 计算的 NAND Flash 页编程/读取，以及将 CPU 从定序任务中释放出来。通过使用命令定序器，软件可以对整个 NAND Flash 页（而不是扇区）进行操作，这样更加高效，对于实现高吞吐量至关重要。定序器处理页中的所有扇区。它不支持子页读取或编程操作。

写入操作需要一条 DMA 通道，读取操作需要两条 DMA 通道。

使用 BCH 算法时，必须遵循特定的 NAND Flash 页布局。有关 2KB 页 BCH 布局的示例，请参见*图 148*。

图 148. 带 BCH 的 NAND Flash 页 (2 KB) 布局

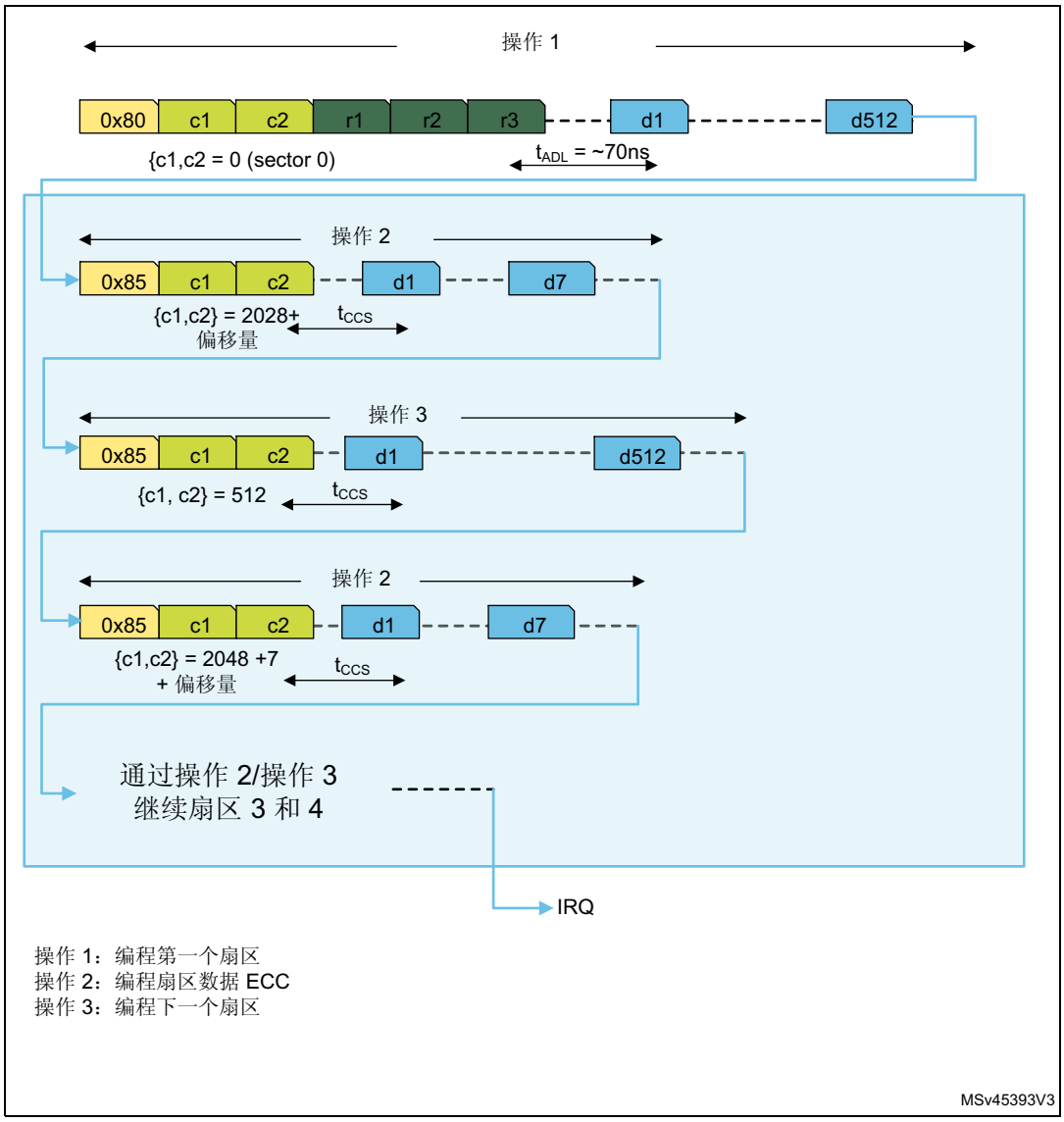
页编程序列

1. 必须使用要写入 NAND Flash 页的数据对位于 SRAM 中的数据缓冲区进行初始化。
2. CPU 配置 FMC 命令定序器寄存器：页程序命令 (0x80)、地址字节、更改写入列命令 (0x85)、ECC 偏移地址、页大小、扇区大小和扇区编号。另外还选择芯片使能信号。

   此外，还选择 ECC 算法（汉明或 BCH）并配置一条 DMA 通道以将缓冲区中的数据传输到 NAND Flash 页。
3. 然后，CPU 通过将 FMC_CSQCR 寄存器中的 CSQSTART 位置 1 来启动定序器。
4. 命令定序器遵循在 FMC_PMEM 或 FMC_PATT 寄存器中编程配置的时序，自动生成对 NAND Flash 的控制周期和地址周期。
5. 发出 DMA 请求以开始从数据缓冲区到 NAND Flash 的数据传输。
6. 当整个扇区完成编程时，如果使能了 ECC，则继续执行步骤 7。否则，跳转到步骤 9。
7. FMC 命令定序器自动发出更改写入列命令 (0x85)，后跟两个地址周期。要发出的命令和地址周期数在 FMC_CSQCFGR2 寄存器中进行编程。地址基于设定的 ECC 偏移地址自动生成。
8. 一旦奇偶校验字节在 ECC 编码器（汉明或 BCH）模块中可用，将自动传输到 NAND Flash 备用区域。如果之前已使能扇区传输完成中断，则FMC 命令定序器可以生成该中断。
9. 然后，定序器重复执行步骤 4（但执行的是更改写入列命令，而不执行页编程命令）到步骤 8，以编程后续扇区及其相应的 ECC 奇偶校验字节。对于每次扇区传输，ECC 偏移地址会自动增加。当禁止 ECC 时，定序器重复执行步骤 5。
10. FMC 命令定序器既可以按扇区生成传输完成中断或传输完成错误中断，也可以在写入所有扇区后生成该中断。
11. CPU 可以在 NAND Flash 区域中编程任何其他元数据，注意不要覆盖奇偶校验字节。
12. 然后，可以将编程命令 (0x10) 写入命令部分，以编程 NAND Flash。
13. 如果使能了中断，则会在就绪/繁忙 (RNB) 信号的上升沿生成中断，以指示页编程操作已完成。

有关使用 BCH 的页编程序列的概览，请参见*图 149*。

图 149. 带 4 位 BCH 的 2 KB 页编程序列示例



页读取序列

1. 在 SRAM 中分配一个数据缓冲区以存储 NAND Flash 页数据，并分配一个解码状态缓冲区以存储 ECC 解码的结果。
2. CPU 编程命令定序器寄存器：页读取第一周期命令 (0x00)、地址字节、页读取第二周期命令 (0x30)、更改读取列命令 (0x05)、备用区域偏移、页大小、扇区大小和扇区编号。另外还选择芯片使能信号。

   根据需要配置 ECC 算法（汉明或 BCH），并配置一条 DMA 通道处理从 NAND Flash 页到数据缓冲区的数据传输，以及第二条 DMA 通道将 ECC 解码结果从 ECC 状态寄存器传输到 RAM 解码状态缓冲区。
3. 然后，CPU 通过将 FMC_CSQCR 寄存器中的 CSQSTART 位置 1 来启动 FMC 定序器。
4. FMC 命令定序器向 NAND Flash 发出命令周期和地址周期。
5. 命令定序器等待就绪/繁忙 (RNB) 上升沿以开始数据传输。
6. 命令定序器发送 DMA 请求以启动从 NAND Flash 页到数据缓冲区的数据传输。
7. 当所有扇区数据读取完毕后，如果使能了 ECC，则继续执行步骤 8，否则执行步骤 9。如果之前已使能扇区传输完成中断，则FMC 命令定序器可以生成该中断。
8. 当 ECC 状态寄存器为空（由于前一个扇区的汉明错误码或 BCH 错误位置位已被读取）时，ECC 命令定序器自动发出更改读取列命令 (0xE0)，后跟两个地址周期，然后是 0xE0 命令，以从 NAND Flash 中读取 ECC 字节并将其存储到 ECC 寄存器中。这些命令以及地址周期数在 FMC_CSQCFGR2 寄存器中进行编程（请参见*图 150*）。

   备用区域中的地址是从设定的 ECC 偏移地址中提取。

   当使用汉明算法时，将向第二条 DMA 通道发送 DMA 请求，以将 ECC 码（FMC_HPR 和 FMC_HECCR 寄存器）传输到解码状态缓冲区。

   当使用 BCH 算法时，一旦所有奇偶校验字节均读取完毕，BCH 解码器首先处理校正子以检测错误，然后搜索错误位置（如果有错误）。

   由于 BCH 解码器为流水线式，即使前一个扇区的解码尚未完成，也可以开始读取下一个扇区。

   在执行以下步骤的过程中，如果 BCH 扇区解码完成，则将新的错误状态存储在 ECC 状态寄存器 (FMC_BCHDSR0) 中，然后向第二条 DMA 通道发送 DMA 请求，以将状态寄存器中的数据传输到位于 RAM 中的解码状态缓冲区。
9. 然后，FMC 命令定序器重复执行步骤 4（但发出的是更改读取列命令，而不发出页读取第一周期命令）到步骤 8，以读取其他扇区及其对应的 ECC 字节。在页大小、扇区数、备用区偏移地址和 BCH 码选择后，发出地址并递增。当读取序列完成时，命令定序器等待 t_{RHWL} 后发出新的命令序列（请参见*图 149*）。当禁止 ECC 时，定序器重复执行步骤 6。
10. 一旦命令定序器将页内的所有扇区和状态寄存器 (FMC_BCHDSRx) 传输完毕，ECC 控制器可以生成传输完成中断或传输完成错误中断（如果使能了该中断）。
11. 通过读取解码状态缓冲区，CPU 可以开始处理与每个扇区对应的错误。如果未检测到错误，则确认存储在 RAM 中的页有效。如果检测到错误，CPU 会先纠正错误，然后再将确认 RAM 中的页缓冲区有效。如果错误太多，CPU 可以检测无法纠正的页，并使数据缓冲区失效。

在所选的 BCH 码之后，如果检测到的错误数量超过可纠正的错误数量，则该位无意义，并且 DUE 位置 1。如果 BCH 解码器检测到的错误数量超过可检测的最大错误数量，则 DEN 位和 DUE 位无意义。

DMA 通道也可以在每次数据传输完成时发送中断。这样，CPU 便可即时处理扇区中的错误。

注： *使用定序器和 DMA 通道时，CPU 必须等到所有扇区的数据传输完成（即，FMC_CSQISR 寄存器中的 TCF 标志置 1 时）后才能读取 FMC_BCHPBRx 和 FMC_BCHDSRx 寄存器。*

图 150. 链式页读命令时序

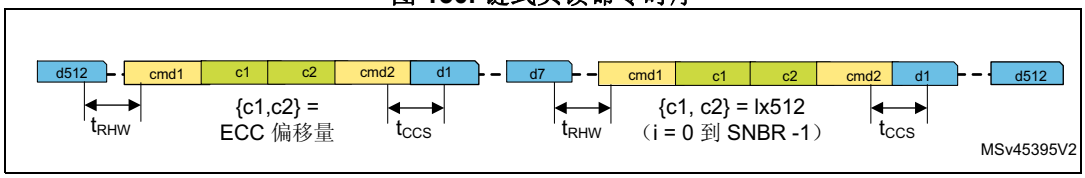
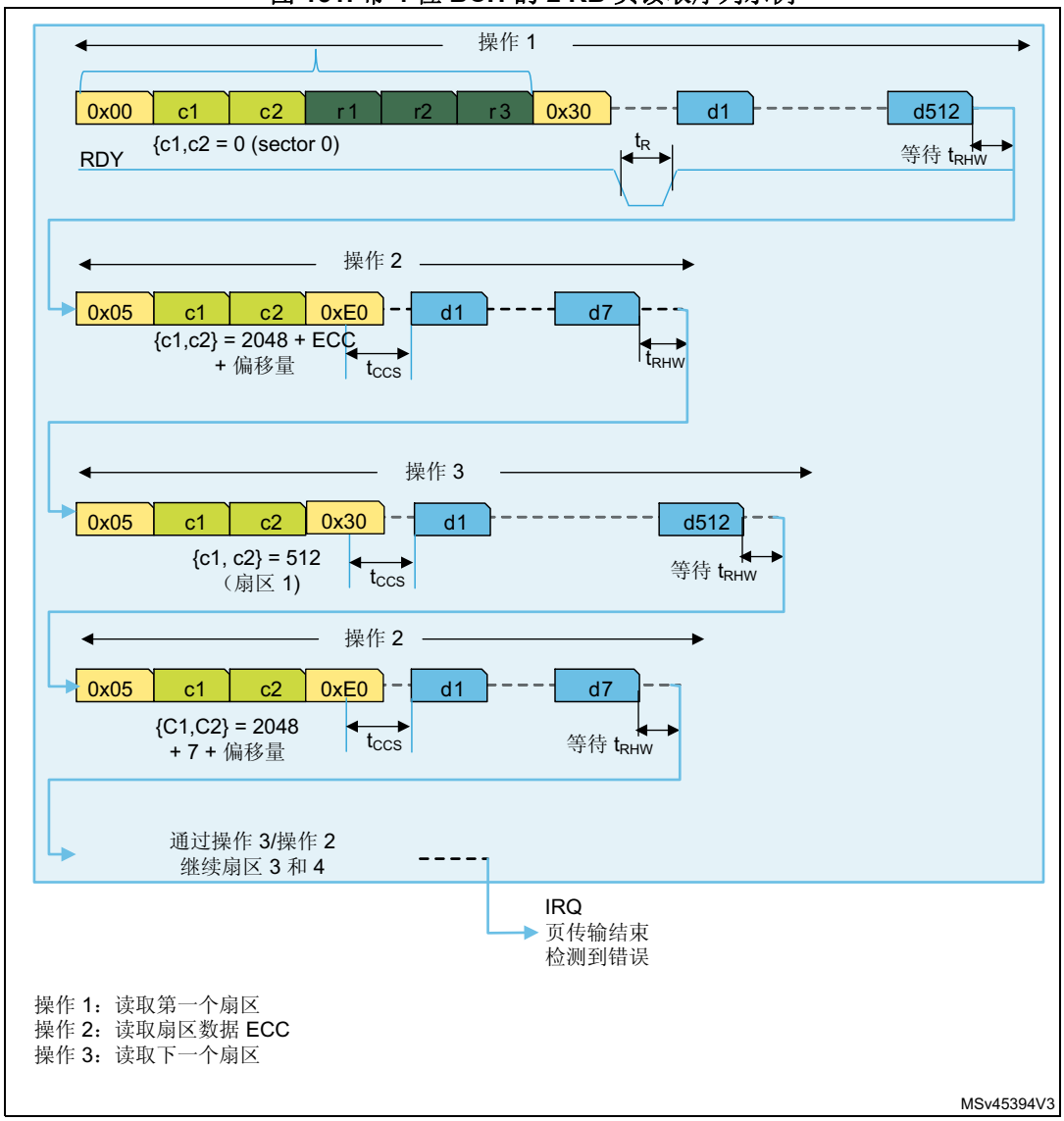


图 151 给出了一个使用 BCH 的 2 KB 页读取序列的示例。该示例中包含四个 512 字节的扇区，每 512 字节数据包含 7 个 ECC 字节。

图 151. 带 4 位 BCH 的 2 KB 页读取序列示例



每当发出页读取 (0x00)、页编程 (0x80)、随机读取 (0x05) 或随机编程命令 (0x85) 时，BCH 控制器都会复位 ECC 计算。

当使用 BCH 写入/读取页的一部分时，命令定序器根据要读取/写入的扇区数量发出随机读取/写入命令序列。这通过设定要处理的扇区数量来实现。

FMC 命令定序器配置

1. 配置 FMC_CSQCFGR1 寄存器：设定命令 1 和命令 2（在读取访问的情况下）、要发出的地址周期数，配置读取或写入序列，以及选择要使用的时序。
2. 在 FMC_CSQAR1 和 FMC_CSQAR2 寄存器中配置地址字节。
3. 在 FMC_CSQCFGR2 寄存器中，配置更改写入列命令或更改读取列命令以及用于 ECC 的时序。在 FMC_CSQAR2 寄存器中设定随机列的偏移地址。根据需要使能 DMA，以进行 ECC 状态寄存器传输。
4. 在 FMC_CSQCFGR3 寄存器中配置扇区数和地址周期时序。
5. 在 FMC_PCR 寄存器中配置 ECC 扇区大小并选择读取访问或写入访问。
6. 使能 FMC_CSQFCGR2 寄存器中的 SQSDTEN 位。FMC_PCR 寄存器中的 ECCEN 位保持为 0。
7. 可通过 FMC_CSQIER 寄存器使能传输完成、传输错误完成或扇区传输完成。

22.9.8 NAND Flash 控制器中断

发生如下事件时可生成中断：

- RNB 上升沿：当检测到 RNB NAND Flash 信号出现上升沿时，将产生中断。
- RNB 下降沿：当检测到 RNB NAND Flash 信号出现下降沿时，将产生中断。
- RNB 高电平：当检测到 RNB NAND Flash 信号出现高电平时，将产生中断。
- 传输完成：当 FMC 命令定序器完成所有扇区的所有数据传输，或当命令定序器被中止时，将产生中断。
- 扇区完成中断：当 FMC 命令定序器完成一个扇区的所有数据传输时，将产生中断。
- 扇区错误中断：当 FMC 命令定序器完成扇区传输并至少检测到一个错误时，将产生中断。
- 扇区不可纠正错误中断：当 FMC 命令定序器完成扇区传输并检测到不可纠正的错误时，将产生中断。
- 命令传输完成：命令定序器已完成设定的命令和地址的传输。
- BCH 编码器奇偶校验位就绪：当 BCH 编码器处理完奇偶校验位时，将产生中断。
- BCH 解码器校正子就绪：当 BCH 解码器处理完校正子并已知是否检测到错误时，将产生中断。
- 发现 BCH 解码器错误：当 BCH 解码器检测到错误时，将产生中断。
- BCH 解码器错误就绪：当 BCH 解码器处理完错误、确定错误位置并且随后可以进行纠正时，将产生中断。
- BCH 解码器不可纠正错误：当 BCH 解码器检测到错误无法纠正（错误数量超过四个或八个，具体取决于 BCH 纠错能力配置）时，将产生中断。

可以分别通过 FMC_IER、FMC_CSQIER 和 FMC_BCHIER 寄存器使能或禁止中断源。

表 165. FMC NAND 控制器中断请求

中断事件	事件标志	使能控制位
传输完成	FMC_CSQISR 中的 TCF	FMC_CSQIER 中的 TCIE
扇区完成	FMC_CSQISR 中的 SCF	FMC_CSQIER 中的 SCIE
扇区错误	FMC_CSQISR 中的 SEF	FMC_CSQIER 中的 SEIE
扇区不可纠正错误	FMC_CSQISR 中的 SUEF	FMC_CSQIER 中的 SUEIE
命令传输完成	FMC_CSQISR 中的 CMDTCF	FMC_CSQIER 中的 CMDTCIE

表 165. FMC NAND 控制器中断请求（续）

中断事件	事件标志	使能控制位
BCH 编码器奇偶校验位就绪	FMC_BCHISR 中的 EPBRF	FMC_BCHIER 中的 EPBRIE
BCH 解码器校正子就绪	FMC_BCHISR 中的 DSRF	FMC_BCHIER 中的 DSRIE
发现 BCH 解码器错误	FMC_BCHISR 中的 DEFF	FMC_BCHIER 中的 DEFIE
BCH 解码器错误就绪	FMC_BCHISR 中的 DERF	FMC_BCHIER 中的 DERIE
BCH 解码器不可纠正错误	FMC_BCHISR 中的 DUEF	FMC_BCHIER 中的 DUEIE

22.9.9 NAND Flash 控制器寄存器

NAND Flash 可编程控制寄存器 (FMC_PCR)

NAND flash controller registers

偏移地址：0x80

复位值：0x0007 FE08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	WEN	BCHECC	Res.	Res.	Res.	Res.	ECCSS[2:0]			TAR[3]
						rw	rw					rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAR[2:0]			TCLR[3:0]				ECCALG	Res.	ECCEN	PWID[1:0]		Res.	PBKEN	PWAITEN	Res.
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw		rw	rw	

位 31:26 保留，必须保持复位值。

位 25 **WEN**：写入使能

该位用于使能读取访问或写入访问。当使用 FMC 定序器或使能带 BCH 码的 ECC 时，必须配置该位。该位只能在 ECCEN 位复位时进行修改。

0：使能读取访问

1：使能写入访问

位 24 **BCHECC**：BCH 纠错能力 (BCH error correction capability)

0：4 位 BCH（每 512 字节纠正 4 位错误并检测出 8 位错误）

1：8 位 BCH（每 512 字节纠正 8 位错误并检测出 16 位错误）

位 23:20 保留，必须保持复位值。

位 19:17 **ECCSS[2:0]**：ECC 扇区大小 (ECC sector size)（用于访问备用区域）
这些位定义用于 ECC 的扇区大小：
000：256 字节
001：512 字节
010：1024 字节
011：2048 字节（默认）
100：4096 字节
101：保留。
11X：保留。
当使能带 BCH 码的 ECC 时，扇区大小必须设定为 512 字节。

位 16:13 **TAR[3:0]**：ALE 到 RE 的延迟 (ALE to RE delay)。
这些位以 fmc_ker_ck 时钟周期数定义从 ALE 低电平到 RE 低电平的时间：
$t_ar \leq (TAR +1) \times t_{fmc_ker_ck}$
其中 $t_{fmc_ker_ck}$ 是 FMC 时钟周期，*t*_ar 是 NAND Flash 的 ALE 到 RE 的时间。
0000：1 个 fmc_ker_ck 周期
1111：16 个 fmc_ker_ck 周期（默认）

位 12:9 **TCLR[3:0]**：CLE 到 RE 的延迟 (CLE to RE delay)。
这些位以 fmc_ker_ck 时钟周期数定义从 CLE 低电平到 RE 低电平的时间：
$t_clr \leq (TCLR +1) \times t_{fmc_ker_ck}$
其中 $t_{fmc_ker_ck}$ 是 FMC 时钟周期，*t*_clr 是 NAND Flash 的 CLE 到 RE 的时间。
0000：1 个 fmc_ker_ck 周期
1111：16 个 fmc_ker_ck 周期（默认）

位 8 **ECCALG**：ECC 算法 (ECC algorithm)
0：选择汉明码（默认）。
1：选择 BCH 码。

位 7 保留，必须保持复位值。

位 6 **ECCEN**：ECC 计算逻辑使能位 (ECC computation logic enable bit)
0：禁止和复位 ECC 逻辑（复位后为默认值）。
1：使能 ECC 逻辑。
使用 FMC 定序器时，该位必须保持复位状态

位 5:4 **PWID[1:0]**：数据总线宽度
这些位定义外部存储设备宽度。
00：8 位（复位后为默认值）。
01：16 位
1X：保留。

位 3 保留，必须保持复位值。

位 2 **PBKEN**：NAND Flash 存储区域使能位 (NAND flash memory region enable bit)
该位用于使能存储区域。访问禁止的存储区域会引起 AXI 总线错误。
0：禁止相应的存储区域（复位后为默认值）
1：使能相应的存储区域。
注意：仅当所有 NAND Flash 控制器命令完成后，才能禁止 NAND Flash 存储区域。

位 1 **PWAITEN**：等待特性使能位 (Wait feature enable bit)
该位用于使能 NAND Flash 存储区域的等待特性：
0：禁止（默认）
1：使能

位 0 保留，必须保持复位值。

FMC 状态寄存器 (FMC_SR)

FMC status register

偏移地址：0x84

复位值：0x0000 0053

该寄存器包含有关 AXI 接口隔离状态和 NAND 写入请求状态的信息。在修改某些寄存器之前，必须禁止 FMC。由于可能有请求挂起，因此必须等到 AXI 接口稳定且模块的内核与其 AXI 接口完全隔离之后再重新配置寄存器。

PEF 和 NWEF 位指示管道的状态。如果使用汉明算法，则在数据写入内存时计算 ECC。为了读取正确的 ECC，软件必须等到 NAND 控制器没有挂起的写入请求（通过轮询 PEF 和 NWRF 位确定）之后再进行读取。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NWRF	Res.	PEF	Res.	Res.	ISOST[1:0]	
									r		r			r	r

位 31:7 保留，必须保持复位值。

位 6 **NWRF**：NAND 写入请求标志 (NAND write request flag)

该位提供发给 NAND Flash 的写入请求的状态。当该位置 1 时，将处理对 NAND 控制器的所有写入请求。

0：存在挂起的 NAND Flash 写入请求

1：不存在挂起的 NAND Flash 写入请求

位 5 保留，必须保持复位值。

位 4 **PEF**：管道为空标志 (Pipe Empty Flag)

该位指示管道中是否包含请求。

位 3:2 保留，必须保持复位值。

位 1:0 **ISOST[1:0]**：FMC 与 AXI 接口之间的隔离状态 (FMC isolation state with respect to the AXI interface)

00：FMC 未隔离；AXI 事务由 FMC 及其控制器处理。

01：FMC 已使能（FMCEN = 1），并等待挂起的 AXI 事务结束（这将引发错误响应）后再进入“非隔离”状态。

10：FMC 已禁止（FMCEN = 0），并等待挂起的 AXI 事务结束后再进入隔离状态。

11：FMC 与其 AXI 接口隔离。所有 AXI 请求都会引发错误响应。

FMC 通用内存空间时序寄存器 (FMC_PMEM)

FMC common memory space timing register

偏移地址：0x88

复位值：0x0A0A 0A0A

FMC_PMEM 读/写寄存器包含 NAND Flash 区域时序信息。该信息用于访问 NAND Flash 的通用存储空间来实现命令、地址写访问或数据读/写访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MEMHIZ[7:0]								MEMHOLD[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEMWAIT[7:0]								MEMSET[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **MEMHIZ[7:0]**：通用内存数据总线高阻态时间 (Common memory data bus Hi-Z time)

这些位定义在对 NAND Flash 的通用存储空间开始执行写访问之后，数据总线保持高阻态所持续的 fmc_ker_ck 时钟周期数 (+1)。仅对写入事务有效：

0000 0000：1 个 fmc_ker_ck 周期

0000 0001：2 个 fmc_ker_ck 周期

.....

1111 1111：256 个 fmc_ker_ck 周期

位 23:16 **MEMHOLD[7:0]**：通用内存保持时间 (Common memory hold time)

这些位定义在对 NAND Flash 的通用存储空间执行读或写访问期间，禁止命令（NWE、NOE）之后保持地址（和写访问数据）的 fmc_ker_ck 时钟周期数 (+1)：

0000 0000：1 个 fmc_ker_ck 周期

0000 0001：2 个 fmc_ker_ck 周期

.....

1111 1111：256 个 fmc_ker_ck 周期

位 15:8 **MEMWAIT[7:0]**：通用内存等待时间 (Common memory wait time)

这些位定义在对 NAND Flash 的通用存储空间执行读或写访问期间，使能命令（NWE、NOE）所需的 fmc_ker_ck 时钟周期数最小值 (+1)。如果等待信号 (NWAIT) 在编程的 fmc_ker_ck 值末尾处有效（低电平），则命令使能的持续时间将延长：

0000 0000：1 个 fmc_ker_ck 周期（+ 禁止 NWAIT 时引入的等待周期）

0000 0001：2 个 fmc_ker_ck 周期（+ 禁止 NWAIT 时引入的等待周期）

.....

1111 1111：256 个 fmc_ker_ck 周期（+ 禁止 NWAIT 时引入的等待周期）

位 7:0 **MEMSET[7:0]**：通用内存建立时间 (Common memory setup time)

这些位定义在对 NAND Flash 的通用存储空间执行读或写访问期间，使能命令（NWE、NOE）前建立地址所需的 fmc_ker_ck 时钟周期数 (+1)：

0000 0000：1 个 fmc_ker_ck 周期

0000 0001：2 个 fmc_ker_ck 周期

.....

1111 1111：256 个 fmc_ker_ck 周期

FMC 特性内存空间时序寄存器 (FMC_PATT)

FMC attribute memory space timing registers

偏移地址：0x8C

复位值：0x0A0A 0A0A

FMC_PATT 读/写寄存器包含 NAND Flash 区域时序信息。在最后一次地址写访问期间对 NAND Flash 的特性存储空间进行 8 位访问时，如果时序与先前访问的时序不同，则使用该寄存器（有关就绪/繁忙管理的信息，请参见*第 22.9.5 节：NAND Flash 预等待功能*）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ATTHIZ[7:0]								ATTHOLD[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATTWAIT[7:0]								ATTSET[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **ATTHIZ[7:0]**：特性内存数据总线高阻态时间 (Attribute memory data bus Hi-Z time)

这些位定义在对 NAND Flash 的特性存储空间开始执行写访问之后，数据总线保持高阻态所持续的 fmc_ker_ck 时钟周期数 (+1)。这些位仅对写入事务有效：

0000 0000：1 个 fmc_ker_ck 周期

0000 0001：2 个 fmc_ker_ck 周期

.....

1111 1111：256 个 fmc_ker_ck 周期

位 23:16 **ATTHOLD[7:0]**：特性内存保持时间 (Attribute memory hold time)

这些位定义在对 NAND Flash 的特性存储空间执行读或写访问期间，禁止命令（NWE、NOE）之后保持地址（和写访问数据）的 fmc_ker_ck 时钟周期数 (+1)：

0000 0000：1 个 fmc_ker_ck 周期

0000 0001：2 个 fmc_ker_ck 周期

.....

1111 1111：256 个 fmc_ker_ck 周期

位 15:8 **ATTWAIT[7:0]**：特性内存等待时间 (Attribute memory wait time)

这些位定义在对 NAND Flash 的特性存储空间执行读或写访问期间，使能命令（NWE、NOE）所需的 fmc_ker_ck 时钟周期数最小值 (+1)。如果等待信号 (NWAIT) 在编程的 fmc_ker_ck 值末尾处有效（低电平），则命令使能的持续时间将延长：

0000 0000：1 个 fmc_ker_ck 周期（+ 禁止 NWAIT 时引入的等待周期）

0000 0001：2 个 fmc_ker_ck 周期（+ 禁止 NWAIT 时引入的等待周期）

.....

1111 1111：256 个 fmc_ker_ck 周期（+ 禁止 NWAIT 时引入的等待周期）

位 7:0 **ATTSET[7:0]**：特性内存建立时间 (Attribute memory setup time)

这些位定义在对 NAND Flash 的特性存储空间执行读或写访问期间，使能命令（NWE、NOE）前建立地址所需的 fmc_ker_ck 时钟周期数 (+1)：

0000 0000：1 个 fmc_ker_ck 周期

0000 0001：2 个 fmc_ker_ck 周期

.....

1111 1111：256 个 fmc_ker_ck 周期

FMC 汉明奇偶校验结果寄存器 (FMC_HPR)

FMC Hamming parity result registers

偏移地址：0x90

复位值：0x0000 0000

该寄存器与 FMC 定序器搭配用于读取访问。其中包含由 FMC NAND 控制器汉明模块计算得出的当前纠错码值。当 FMC 定序器从 NAND Flash 页中的正确地址处读取数据时，汉明计算模块会自动处理读取的数据。读取 X 个字节后（依据 FMC_PCR 寄存器中的扇区大小 ECCSS 位字段），CPU 必须从 FMC_HECCR 寄存器读取计算所得的 ECC 值。然后验证这些计算的奇偶校验数据是否与备用区记录的奇偶校验数据和 FMC_HPR 中存储的奇偶校验数据相同，从而确定该页是否有效，如不同则进行纠正。FMC_HPR 寄存器被读取后应通过将 ECCEN 位置 0 进行清零。如要计算新的数据块，则必须将 ECCEN 位置 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HPR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HPR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **HPR[31:0]**：汉明奇偶校验结果 (Hamming parity result)

该位字段包含由汉明 ECC 计算模块计算所得的奇偶校验值。*表 166* 介绍了这些位字段的内容。

表 166. HPR 相关位

ECCSS[2:0]	扇区大小（字节）	HPR 位
000	256	HPR[21:0]
001	512	HPR[23:0]
010	1024	HPR[25:0]
011	2048	HPR[27:0]
100	4096	HPR[29:0]
101	8192	HPR[31:0]

FMC 汉明码 ECC 结果寄存器 (FMC_HECCR)

FMC Hamming code ECC result register

偏移地址：0x94

复位值：0x0000 0000

该寄存器包含由 FMC NAND 控制器汉明模块计算所得的当前纠错码值。当 CPU 在 NAND Flash 页中的正确地址处读取/写入数据时（请参见*第 22.9.6 节：NAND ECC 控制器*），汉明计算模块会自动处理对 NAND Flash 读取/写入的数据。读取 X 个字节后（依据 FMC_PCR 寄存器中的扇区大小 ECCSS 位字段），CPU 必须从 FMC_HECCR 寄存器读取计算所得的 ECC 值。然后验证这些计算的奇偶校验数据是否与备用区记录的奇偶校验数据相同，从而确

定该页是否有效，如不同则进行纠正。FMC_HECCR 寄存器被读取后应通过将 ECCEN 位置 0 进行清零。如要计算新的数据块，则必须将 ECCEN 位置 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HECC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HECC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **HECC[31:0]**：ECC 结果 (ECC result)

该位字段包含由汉明 ECC 计算逻辑计算所得的值。*表 167* 介绍了这些位字段的内容。

表 167. ECC 结果相关位(1)

ECCSS[2:0]	扇区大小（字节）	HECC 位
000	256	HECC[21:0]
001	512	HECC[23:0]
010	1024	HECC[25:0]
011	2048	HECC[27:0]
100	4096	HECC[29:0]
101	8192	HECC[31：0]

1. 有效位数取决于扇区大小。

FMC NAND 中断使能寄存器 (FMC_IER)

FMC NAND interrupt enable register

偏移地址：0x180

复位值：0x0000 0000

该寄存器用于配置在 FMC 中断引脚上连接 RNB 信号时的中断生成功能，具体可以选择在检测到 RNB 信号上出现上升沿、下降沿或高电平时生成中断。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IFEE	IHLE	IREE
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2 **IFEE**：中断下降沿检测使能位 (Interrupt falling edge detection enable bit)

0：禁止中断下降沿检测请求

1：使能中断下降沿检测请求

位 1 **IHLE**：中断高电平检测使能位 (Interrupt high-level detection enable bit)
0：禁止中断高电平检测请求
1：使能中断高电平检测请求

位 0 **IREE**：中断上升沿检测使能位 (Interrupt rising edge detection enable bit)
0：禁止中断上升沿检测请求
1：使能中断上升沿检测请求

FMC 中断状态寄存器 (FMC_ISR)

FMC interrupt status register

偏移地址：0x184

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IFEF	IHLF	IREF
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **IFEF**：中断下降沿标志 (Interrupt falling edge flag)
该标志由硬件置 1，并通过将 CFEIF 位置 1 来清零。
0：未出现中断下降沿
1：出现中断下降沿

位 1 **IHLF**：中断高电平标志 (Interrupt high-level flag)
该标志由硬件置 1，并通过将 CHLIF 位置 1 来清零。
0：未出现中断高电平
1：出现中断高电平

位 0 **IREF**：中断上升沿标志 (Interrupt rising edge flag)
该标志由硬件置 1，并通过将 CREIF 位置 1 来清零。
0：未出现中断上升沿
1：出现中断上升沿

FMC NAND 控制器中断清零寄存器 (FMC_ICR)

FMC NAND controller interrupt clear register

偏移地址：0x188

复位值：0x0000 0000

向该寄存器中的一个位写入 1 即可清零中断状态寄存器 (FMC_ISR) 中对应的标志。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CIFEF	CIHLF	CIREF
													w	w	w

位 31:3 保留，必须保持复位值。

位 2 **CIFEF**：清零中断下降沿标志 (Clear Interrupt falling edge flag)
0：无影响
1：将 FMC_ISR 寄存器中的 IFEF 标志清零

位 1 **CIHLF**：清零中断高电平标志 (Clear Interrupt high-level flag)
0：无影响
1：将 FMC_ISR 寄存器中的 IHLF 标志清零

位 0 **CIREF**：清零中断上升沿标志 (Clear Interrupt rising edge flag)
0：无影响
将 FMC_ISR 寄存器中的 IREF 标志清零

FMC NAND 命令定序器控制寄存器 (FMC_CSQCR)

FMC NAND command sequencer control register

偏移地址：0x200

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSQSTART
															w

位 31:1 保留，必须保持复位值。

位 0 **CSQSTART**：命令定序器使能 (Command Sequencer Enable)

写入 1 会启动命令定序器。

写入 0 无影响。

注意：仅当通过将 FMC_CFGR 的 FMCEN 位置 1 来使能 FMC，并通过将 FMC_PCR 的 PBKEN 位置 1 来激活 NAND Flash 控制器时，向该位写入 1 才会启动定序器。

FMC NAND 命令定序器配置寄存器 1 (FMC_CSQCFGR1)

FMC NAND command sequencer configuration register 1

偏移地址：0x204

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	CMD2T	CMD1T	CMD2[7:0]							
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD1[7:0]								Res.	ACYNBR[2:0]			Res.	DMADEN	CMD2EN	Res.
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw		rw	rw	

位 31:26 保留，必须保持复位值。

位 25 **CMD2T**：命令 2 定序器时序 (Command 2 Sequencer timings)

该位指定是以 FMC_PMEM 还是 FMC_PATT 寄存器中设定的时序向 NAND Flash 发出命令 2 (CMD2)。

0：以 FMC_PMEM 中设定的时序发出 CMD2

1：以 FMC_PATT 中设定的时序发出 CMD2

位 24 **CMD1T**：命令 1 定序器时序 (Command 1 Sequencer timings)

该位指定是以 FMC_PMEM 还是 FMC_PATT 寄存器中设定的时序向 NAND Flash 发出命令 1 (CMD1)。

0：以 FMC_PMEM 中设定的时序发出 CMD1

1：以 FMC_PATT 中设定的时序发出 CMD1

位 23:16 **CMD2[7:0]**：命令 2 定序器 (Command 2 sequencer)

这些位指定在读取操作期间是否由命令定序器在命令第二周期发出命令 2。仅当 CMD2EN 位置 1 时才发出该命令。

示例：必须将 CMD2[7:0] 位编程为 0x30 才能发出读列取命令。

位 15:8 **CMD1[7:0]**：命令 1 定序器 (Command 1 sequencer)

这些位指定在读取或写入操作期间是否由命令定序器在第一周期发出命令 1。

位 7 保留，必须保持复位值。

位 6:4 **ACYNBR[2:0]**：地址周期数 (Address Cycle number)

这些位定义在发出寄存器中设定的命令时命令定序器要生成的地址周期数。

000：无地址周期

001：1 个地址周期

010：2 个地址周期

011：3 个地址周期

100：4 个地址周期

101：5 个地址周期

其他值：保留。

位 3 保留，必须保持复位值。

位 2 **DMADEN**：命令定序器 DMA 请求数据使能 (Command sequencer DMA request data enable)

当该位置 1 时，命令定序器会向第一个 DMA 通道发出 DMA 请求，以与 NAND Flash 之间传输数据。

0：未请求 DMA 传输

1：请求 DMA 传输

位 1 **CMD2EN**：命令周期 2 使能 (Command cycle 2 Enable)

该位指定命令定序器必须生成命令周期 2。

0：未发出命令周期 2。

1：命令定序器在地址周期后将命令周期 2（设定的 CMD2[7:0]）发送至 NAND Flash。

位 0 保留，必须保持复位值。

FMC NAND 命令定序器配置寄存器 2 (FMC_CSQCFGR2)

FMC NAND command sequencer configuration register 2

偏移地址：0x208

复位值：0x0000 0000

该寄存器用于配置命令定序器发出随机读/写命令以按扇区读取/写入数据，并自动以可编程的偏移地址从 NAND Flash 读取/写入数据。在执行扇区读/写操作后接着对 NAND Flash 备用区域进行 ECC 读/写操作时，该寄存器很有用。命令定序器会一直生成随机命令，直到所有扇区均完成读取/写入为止。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	RCMD2T	RCMD1T	RCMD2[7:0]							
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCMD1[7:0]								Res.	Res.	Res.	Res.	Res.	DMASEN	RCMD2EN	SQSDTEN
rw	rw	rw	rw	rw	rw	rw	rw						rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25 **RCMD2T**：命令 1 定序器时序 (Command 1 sequencer timings)

该位指定是以 FMC_PMEM 还是 FMC_PATT 寄存器中设定的时序向 NAND Flash 发出 CMD1。

0：以 FMC_PMEM 中设定的时序发出 CMD1

1：以 FMC_PATT 中设定的时序发出 CMD1

位 24 **RCMD1T**：命令 1 定序器时序 (Command 1 sequencer timings)

该位指定是以 FMC_PMEM 还是 FMC_PATT 寄存器中设定的时序向 NAND Flash 发出 CMD1。

0：以 FMC_PMEM 中设定的时序发出 CMD1

1：以 FMC_PATT 中设定的时序发出 CMD1

位 23:16 **RCMD2[7:0]**：随机命令 2 定序器 (Random Command 2 sequencer)

这些位指定在读取操作期间是否由命令定序器在第二周期发出命令 2。仅当 RCMD2EN 位置 1 时才发出该命令。

示例：需要使用 RCMD2 (0xE0) 来发出随机读取列命令。

位 15:8 **RCMD1[7:0]**：随机命令 1 定序器 (Random Command 1 sequencer)

这些位指定在读取或写入访问期间是否由命令定序器在第一周期发出命令 1。

位 7:3 保留，必须保持复位值。

位 2 **DMASEN**：命令定序器 DMA 请求解码状态使能 (Command sequencer DMA request decoding status enable)

当该位置 1 时，如果在 ECC 处理（解码）结束时 ECC 状态寄存器中有有效位，则命令定序器将生成对第二个 DMA 通道的 DMA 请求。

0：未使用 DMA 请求传输 ECC 状态寄存器中的内容

1：使用 DMA 请求传输 ECC 状态寄存器中的内容

注意：该位只能在页读取序列期间置 1，这样可以使用第二个 DMA 通道，无需保存扇区之间的 ECC 结果。

位 1 **RCMD2EN**：随机命令 2 定序器使能 (Random Command 2 sequencer enable)

该位用于允许命令定序器在读取访问期间发出命令 2。仅当 CMD2SQEN 位置 1 时才发出该命令。

0：未发出命令 2。

1：命令定序器在地址周期后将命令 2 (CMD2SQ[7:0]) 发送至 NAND Flash。

位 0 **SQSDTEN**：定序器备用数据传输使能 (Sequencer spare data transfer enable)

该位用于允许定序器在每次扇区传输后访问备用数据区域。使用定序器时，该位还用于使能 ECC。FMC_PCR 寄存器中的 ECCEN 位必须复位。

0：禁止 ECC，定序器未访问备用数据区域

1：使能 ECC，定序器在每次扇区传输后读取或编程备用数据区域

FMC NAND 定序器配置寄存器 3 (FMC_CSQCFGR3)

FMC NAND sequencer configuration register 3

偏移地址：0x20C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RAC2T	RAC1T	SDT	AC5T	AC4T	AC3T	AC2T	AC1T
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	SNBR[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
		rw	rw	rw	rw	rw	rw								

位 31:24 保留，必须保持复位值。

位 23 **RAC2T**：随机地址周期 2 定序器时序 (Random Address cycle 2 sequencer timings)

该位指定是以 FMC_PMEM 还是 FMC_PATT 寄存器中设定的时序向 NAND Flash 发出随机地址周期 2 (ADDC2)。

0：以 FMC_PMEM 中设定的时序发出随机 ADDC2

1：以 FMC_PATT 中设定的时序发出随机 ADDC2

位 22 **RAC1T**：随机地址周期 1 定序器时序 (Random Address cycle 1 sequencer timings)

该位指定是以 FMC_PMEM 还是 FMC_PATT 寄存器中设定的时序向 NAND Flash 发出随机地址周期 1 (ADDC1)。

0：以 FMC_PMEM 中设定的时序发出随机 ADDC1

1：以 FMC_PATT 中设定的时序发出随机 ADDC1

位 21 **SDT**：备用数据传输定序器时序 (Spare data transfer sequencer timings)

该位指定备用数据传输是以 FMC_PMEM 还是 FMC_PATT 寄存器中设定的时序对 NAND Flash 进行读取和写入操作。

0：以 FMC_PMEM 中设定的时序发起备用数据传输

1：以 FMC_PATT 中设定的时序发起备用数据传输

位 20 **AC5T**：地址周期 5 定序器时序 (Address cycle 5 sequencer timings)

该位定义是以 FMC_PMEM 还是 FMC_PATT 寄存器中设定的时序向 NAND Flash 发出地址周期 5 (ADDC5)。

0：以 FMC_PMEM 中设定的时序发出 ADDC5

1：以 FMC_PATT 中设定的时序发出 ADDC5

位 19 **AC4T**：地址周期 4 定序器时序 (Address cycle 4 sequencer timings)

该位定义是以 FMC_PMEM 还是 FMC_PATT 寄存器中设定的时序向 NAND Flash 发出地址周期 4 (ADDC4)。

0：以 FMC_PMEM 中设定的时序发出 ADDC4

1：以 FMC_PATT 中设定的时序发出 ADDC4

位 18 **AC3T**：地址周期 3 定序器时序 (Address cycle 3 sequencer timings)

该位定义是以 FMC_PMEM 还是 FMC_PATT 寄存器中设定的时序向 NAND Flash 发出地址周期 3 (ADDC3)。

0：以 FMC_PMEM 中设定的时序发出 ADDC3

1：以 FMC_PATT 中设定的时序发出 ADDC3

位 17 **AC2T**：地址周期 2 定序器时序 (Address cycle 2 sequencer timings)
该位定义是以 FMC_PMEM 还是 FMC_PATT 寄存器中设定的时序向 NAND Flash 发出地址周期 2 (ADDC2)。
0：以 FMC_PMEM 中设定的时序发出 ADDC2
1：以 FMC_PATT 中设定的时序发出 ADDC2

位 16 **AC1T**：地址周期 1 定序器时序 (Address cycle 1 sequencer timings)
该位定义是以 FMC_PMEM 还是 FMC_PATT 寄存器中设定的时序向 NAND Flash 发出地址周期 1 (ADDC1)。
0：以 FMC_PMEM 中设定的时序发出 ADDC1
1：以 FMC_PATT 中设定的时序发出 ADDC1

位 15:14 保留，必须保持复位值。

位 13:8 **SNBR[5:0]**：要读取/写入的扇区数 (Number of sectors to be read/written)
这些位定义由定序器处理的扇区数。
这些位用于读取/写入页的一部分或使用 ECC（特别是需要 512 字节扇区的 BCH 码）：
000000：1 个扇区
000001：2 个扇区
...
111111：16 个扇区
其他值：保留。

位 7:0 保留，必须保持复位值。

FMC NAND 命令定序器地址寄存器 1 (FMC_CSQAR1)

FMC NAND command sequencer address register 1

偏移地址：0x210

复位值：0x0000 0000

该寄存器用于定义由命令定序器发出的地址周期 1 到 4 的值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDC4[7:0]								ADDC3[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDC2[7:0]								ADDC1[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **ADDC4[7:0]**：地址周期 4 (Address Cycle 4)
这些位定义命令定序器在读取或写入访问期间发出的地址周期 4 的值。

位 23:16 **ADDC3[7:0]**：地址周期 3 (Address Cycle 3)
这些位定义命令定序器在读取或写入访问期间发出的地址周期 2 的值。

位 15:8 **ADDC2[7:0]**：地址周期 2 (Address Cycle 2)
这些位定义命令定序器在读取或写入访问期间发出的地址周期 2 的值。

位 7:0 **ADDC1[7:0]**：地址周期 1 (Address Cycle 1)
这些位定义命令定序器在读取或写入访问期间发出的地址周期 1 的值。

FMC NAND 命令定序器地址寄存器 2 (FMC_CSQAR2)

FMC NAND command sequencer address register 2

偏移地址：0x214

复位值：0x0000 0000

该寄存器用于编程第五个地址周期和备用区域的偏移地址。另外还选择芯片使能信号。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAO[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	NAND CEN1	NAND CEN0	Res.	Res.	ADDC5[7:0]							
				rw	rw			rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **SAO[15:0]**：备用区域偏移地址 (Spare Area Address Offset)
这些位定义编程序列中备用区域的第一个 ECC 字节的偏移。
注意：最小 SAQ 值为 2，因为备用区的前两个字节保留用于坏块标记和元数据。

位 15:12 保留，必须保持复位值。

位 11:10 **NANDCEN[1:0]**：NAND Flash 芯片使能信号编号 (NAND flash chip enable number)
这些位定义 NAND Flash 命令定序器要选择的芯片使能信号。
00：选择 NCE1 芯片使能信号
01：选择 NCE2 芯片使能信号
其他值：保留

位 9:8 保留，必须保持复位值。

位 7:0 **ADDC5[7:0]**：地址周期 5 (Address Cycle 5)
这些位定义命令定序器在读取或写入访问期间发出的地址周期 5 的值。

FMC NAND 命令定序器中断使能寄存器 (FMC_CSQIER)

FMC NAND command sequencer interrupt enable register

偏移地址：0x220

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDTCIE	SUEIE	SEIE	SCIE	TCIE
											rw	rw	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4 **CMDTCIE**：命令传输完成中断使能 (Command Transfer Complete interrupt enable)
0：禁止命令传输完成中断
1：使能命令传输完成中断

位 3 **SUEIE**：扇区不可纠正错误中断使能 (Sector Uncorrectable Error interrupt enable)
0：禁止命令传输完成中断
1：使能命令传输完成中断

位 2 **SEIE**：扇区错误中断使能 (Sector Error interrupt enable)
0：禁止扇区错误中断
1：使能扇区错误中断

位 1 **SCIE**：扇区完成中断使能 (Sector Complete interrupt enable)
0：禁止扇区完成中断
1：使能扇区完成中断

位 0 **TCIE**：传输完成中断使能 (Transfer Complete Interrupt enable)
0：禁止传输完成中断
1：传输完成中断使能 (Transfer Complete Interrupt enable)

FMC NAND 命令定序器中断状态寄存器 (FMC_CSQISR)

FMC NAND command sequencer interrupt status register

偏移地址：0x224

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDTCF	SUEF	SEF	SCF	TCF
											rw	rw	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4 **CMDTCF**：命令传输完成标志 (Command Transfer Complete flag)
当命令定序器完成设定的命令和地址传输时，该位置 1。
该位通过将 CCMDTCF 位置 1 来清零。如果 FMC_CSQICR 寄存器中的 CMDTCIE 位置 1，可产生中断。
注意：在定序器模式下，不需要扇区传输完成中断。

位 3 **SUEF**：扇区不可纠正错误标志 (Sector Uncorrectable Error flag)
该位由硬件置 1，并通过向 FMC_CSQICR 中的 CSUEF 位写入 1 来清零。
当该位置 1 时，表示在解码扇区时命令定序器检测到不可纠正的错误。如果 FMC_CSQICR 寄存器中的 SUEIE 位置 1，可产生中断。
注意：SUEF 位只与 BCH 码相关，用于指示不可纠正的错误。对于汉明码，必须忽略该位。

位 2 **SEF**：扇区错误标志 (Sector Error flag)

该位由硬件置 1，并通过向 FMC_CSQICR 中的 CSEF 位写入 1 来清零。

当命令定序器完成扇区数据传输并至少检测到一个错误时，该位置 1。如果 FMC_CSQICR 寄存器中的 SEIE 位置 1，可产生中断。

注意：SEF 位仅与 BCH 码相关，用于指示存在错误。对于汉明码，必须忽略该位。

位 1 **SCF**：扇区完成标志 (Sector Complete flag)

该位由硬件置 1，并通过向 FMC_CSQICR 中的 CSCF 位写入 1 来清零。

当命令定序器完成扇区数据传输时，该位置 1。如果 FMC_CSQICR 寄存器中的 SCIE 位置 1，可产生中断。

位 0 **TCF**：传输完成标志 (Transfer Complete flag)

该位由硬件置 1，并通过向 FMC_CSQICR 中的 CTCF 位写入 1 来清零。

当命令定序器完成所有扇区的所有数据传输，或命令定序器被中止时，该位置 1。如果 FMC_CSQICR 寄存器中的 TCIE 位置 1，可产生中断。

FMC NAND 命令定序器中断清零寄存器 (FMC_CSQICR)

FMC NAND command sequencer interrupt clear register

偏移地址：0x228

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCMDTCF	CSUEF	CSEF	CSCF	CTCF
											w	w	w	w	w

位 31:5 保留，必须保持复位值。

位 4 **CCMDTCF**：清零命令传输完成标志 (Clear Command Transfer Complete flag)

写入 1 可将 FMC_CSQISR 寄存器中的 CMTCF 标志清零

位 3 **CSUEF**：清零扇区不可纠正错误标志 (Clear Sector uncorrectable Error flag)

写入 1 可将 FMC_CSQISR 寄存器中的 SUEF 标志清零

位 2 **CSEF**：清零扇区错误标志 (Clear Sector Error flag)

写入 1 可将 FMC_CSQISR 寄存器中的 SEF 标志清零

位 1 **CSCF**：清零扇区完成标志 (Clear Sector Complete flag)

写入 1 可将 FMC_CSQISR 寄存器中的 SCF 标志清零

位 0 **CTCF**：清零传输完成标志 (Clear Transfer Complete flag)

写入 1 可将 FMC_CSQISR 寄存器中的 TCF 标志清零

FMC 命令定序器错误映射状态寄存器 (FMC_CSQEMSR)

FMC command sequencer error mapping status register

偏移地址：0x230

复位值：0x0000 0000

当整个传输完成时，该寄存器保存扇区错误映射状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEM[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **SEM[15:0]**：扇区错误映射 (Sector Error mapping)

当 SEM[i] 位 (0 ≤ i ≤ 15) 置 1 时，表示在扇区 i+1 至少检测到一个错误。

例如，当在扇区 3 和扇区 2 检测到错误时，SEM[2] 位和 SEM[1] 位置 1。

FMC BCH 中断使能寄存器 (FMC_BCHIER)

FMC BCH interrupt enable register

偏移地址：0x250

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EPBRIE	DSRIE	DEFIE	DERIE	DUEIE
											rw	rw	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4 **EPBRIE**：解码器奇偶校验位就绪中断使能 (Decoder Parity Bits Ready Interrupt enable)

0：禁止解码器奇偶校验位就绪中断

1：使能解码器奇偶校验位就绪中断

位 3 **DSRIE**：解码器校正子就绪中断使能 (Decoder Syndrome Ready Interrupt enable)

0：禁止解码器校正子就绪中断

1：使能解码器校正子就绪中断

位 2 **DEFIE**：发现解码器错误中断使能 (Decoder Error Found Interrupt enable)

0：禁止发现解码器错误中断

1：使能发现解码器错误中断

位 1 **DERIE**：解码器错误就绪中断使能 (Decoder Error Ready Interrupt enable)
0：禁止解码器错误就绪中断
1：使能解码器错误就绪中断

位 0 **DUEIE**：解码器不可纠正错误中断使能 (Decoder Uncorrectable Errors Interrupt enable)
0：禁止解码器不可纠正错误中断
1：使能解码器不可纠正错误中断

FMC BCH 中断和状态寄存器 (FMC_BCHISR)

FMC BCH interrupt and status register

偏移地址：0x254

复位值：0x0000 0000

该寄存器在每个扇区完成处理后保存 BCH 编码器/解码器的状态。当使用定序器时，该寄存器会自动清零，并且不得通过 FMC_BCHIER 寄存器使能相应的中断。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EPBRF	DSRF	DEFF	DERF	DUEF
											r	r	r	r	r

位 31:5 保留，必须保持复位值。

位 4 **EPBRF**：编码器奇偶校验位就绪标志 (Encoder Parity Bits Ready flag)
当 FMC_BCHPBR1 到 4 中的 BCH 编码器奇偶校验位可用时，该位置 1。

位 3 **DSRF**：解码器校正子就绪标志 (Decoder Syndrome Ready flag)
当校正子就绪且 DEFF 位有效时，该位置 1。

位 2 **DEFF**：发现解码器错误标志 (Decoder Error Found flag)
该位由硬件置 1，并通过向 CDEFF 写入 1 来清零。
该位指示在扇区中至少检测到一个错误。

位 1 **DERF**：解码器错误就绪标志 (Decoder Error Ready flag)
该位指示解码器已完成错误搜索并识别到错误位置。
错误的数量及其位置分别在 FMC_BCHDSR0 和 FMC_BCHDSR1 到 4 中提供。

位 0 **DUEF**：解码器不可纠正错误标志 (Decoder Uncorrectable Errors flag)
该位指示在 BCH 解码过程中检测到扇区中的错误过多，无法纠正。

FMC BCH 中断清零寄存器 (FMC_BCHICR)

FMC BCH interrupt clear register

偏移地址：0x258

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CEPBRF	CDSRF	CDEFF	CDERF	CDUEF
											w	w	w	w	w

位 31:5 保留，必须保持复位值。

位 4 **CEPBRF**：清零编码器奇偶校验位就绪标志 (Clear Encoder Parity Bits Ready flag)
写入 1 可将 FMC_BCHISR 中的 EPBRF 标志清零。

位 3 **CDSRF**：清零解码器校正子就绪标志 (Clear Decoder Syndrome Ready flag)
写入 1 可将 FMC_BCHISR 中的 DSRF 标志清零。

位 2 **CDEFF**：清零发现解码器错误标志 (Clear Decoder Error Found flag)
写入 1 可将 FMC_BCHISR 中的 DEFF 标志清零。

位 1 **CDERF**：清零解码器错误就绪标志 (Clear Decoder Error ready flag)
写入 1 可将 FMC_BCHISR 中的 DERF 标志清零。

位 0 **CDUEF**：清零解码器不可纠正错误标志 (Clear Decoder Uncorrectable Error flag)
写入 1 可将 FMC_BCHISR 中的 DUEF 标志清零。

FMC BCH 奇偶校验位寄存器 1 (FMC_BCHPBR1)

FMC BCH parity bits register 1

偏移地址：0x260

复位值：0x0000 0000

这些寄存器包含 BCH 奇偶校验位 (BCHPB)。对于 4 位 BCH，仅 BCHPB[51:0] 有意义；对于 8 位 BCH，BCHPB[103:0] 有意义。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCHPB[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCHPB[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **BCHPB[31:0]**：BCH 奇偶校验位 (BCH parity bits)

FMC BCH 奇偶校验位寄存器 2 (FMC_BCHPBR2)

FMC BCH parity bits register 2

偏移地址：0x264

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCHPB[63:48]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCHPB[47:32]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **BCHPB[63:32]**：BCH 奇偶校验位 (BCH parity bits)

FMC BCH 奇偶校验位寄存器 3 (FMC_BCHPBR3)

FMC BCH parity bits register 3

偏移地址：0x268

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCHPB[95:80]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCHPB[79:64]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **BCHPB[95:64]**：BCH 奇偶校验位 (BCH parity bits)

FMC BCH 奇偶校验位寄存器 4 (FMC_BCHPBR4)

FMC BCH parity bits register 4

偏移地址：0x26C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BCHPB[103:96]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **BCHPB[103:96]**：BCH 奇偶校验位 (BCH parity bits)

FMC BCH 解码器状态寄存器 0 (FMC_BCHDSR0)

FMC BCH decoder status register 0

偏移地址：0x27C

复位值：0x0000 0000

该寄存器包含一些在其他寄存器中已有的位字段，但在同时对多个扇区（例如整个 NAND Flash 页）进行纠错时需要保存这些位字段。这样，可以通过 DMA 通道将 FMC_BCHDSR0 到 4 的内容传输到解码状态缓冲区。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEN[3:0]				Res.	Res.	DEF	DUE
								r	r	r	r			r	r

位 31:8 保留，必须保持复位值。

位 7:4 **DEN[3:0]**：解码器错误数量 (Decoder error number)

当 DEF 位置 1 时，该位字段表示 BCH 解码器检测到的错误数量。

在所选的 BCH 码之后，如果检测到的错误数量超过可纠正的错误数量，则该位无意义，并且 DUE 位置 1。

如果 BCH 解码器检测到的错误数量超过可检测的最大错误数量，则 DEN 无意义。

位 3:2 保留，必须保持复位值。

位 1 **DEF**：发现解码器错误 (Decoder error found)

该位字段指示解码器已完成错误搜索并识别到错误位置。

错误的数量及其位置分别在 FMC_BCHDSR0 和 FMC_BCHDSR1 到 4 中提供。

位 0 **DUE**：解码器不可纠正错误 (Decoder uncorrectable error)

该位指示在 BCH 解码过程中检测到扇区中的错误过多，无法纠正。

如果 BCH 解码器检测到的错误数量超过可检测的最大错误数量，则 DUE 无意义。

FMC 存储区域 1 的 BCH 解码器状态寄存器 (FMC_BCHDSR1)

FMC BCH decoder status register for memory region 1

偏移地址：0x280

复位值：0x0000 0000

FMC 内置的 BCH 模块最多可纠正 8 个错误。该寄存器的 EBP1 位字段和 EPB2 位字段分别包含第 1 个错误位和第 2 个错误位的位置。如果检测到的错误少于 8 个，则未使用的 EPBx 字段（x = 1 到 8）为“无关”位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	EBP2[12:0]												
			r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	EBP1[12:0]												
			r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:29 保留，必须保持复位值。

位 28:16 **EBP2[12:0]**：错误编号 2 的错误位位置 (Error bit position for error number 2)

位 15:13 保留，必须保持复位值。

位 12:0 **EBP1[12:0]**：错误编号 1 的错误位位置 (Error bit position for error number 1)

FMC 存储区域 2 的 BCH 解码器状态寄存器 (FMC_BCHDSR2)

FMC BCH decoder status register for memory region 2

偏移地址：0x284

复位值：0x0000 0000

FMC 内置的 BCH 模块最多可纠正 8 个错误。该寄存器的 EBP3 位字段和 EPB4 位字段分别包含第 3 个错误位和第 4 个错误位的位置。如果检测到的错误少于 8 个，则未使用的 EPBx 字段（x = 1 到 8）为“无关”位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	EBP4[12:0]												
			r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	EBP3[12:0]												
			r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:29 保留，必须保持复位值。

位 28:16 **EBP4[12:0]**：错误编号 4 的错误位位置 (Error bit position for error number 4)

位 15:13 保留，必须保持复位值。

位 12:0 **EBP3[12:0]**：错误编号 3 的错误位位置 (Error bit position for error number 3)

FMC 存储区域 3 的 BCH 解码器状态寄存器 (FMC_BCHDSR3)

FMC BCH decoder status register for memory region 3

偏移地址：0x288

复位值：0x0000 0000

FMC 内置的 BCH 模块最多可纠正 8 个错误。该寄存器的 EBP5 位字段和 EPB6 位字段分别包含第 5 个错误位和第 6 个错误位的位置。如果检测到的错误少于 8 个，则未使用的 EPBx 字段（x = 1 到 8）为“无关”位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	EBP6[12:0]												
			r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	EBP5[12:0]												
			r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:29 保留，必须保持复位值。

位 28:16 **EBP6[12:0]**：错误编号 6 的错误位位置 (Error bit position for error number 6)

位 15:13 保留，必须保持复位值。

位 12:0 **EBP5[12:0]**：错误编号 5 的错误位位置 (Error bit position for error number 5)

FMC 存储区域 4 的 BCH 解码器状态寄存器 (FMC_BCHDSR4)

FMC BCH decoder status register for memory region 4

偏移地址：0x28C

复位值：0x0000 0000

FMC 内置的 BCH 模块最多可纠正 8 个错误。该寄存器的 EBP7 位字段和 EPB8 位字段分别包含第 7 个错误位和第 8 个错误位的位置。如果检测到的错误少于 8 个，则未使用的 EPBx 字段（x = 1 到 8）为“无关”位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	EBP8[12:0]												
			r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	EBP7[12:0]												
			r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:29 保留，必须保持复位值。

位 28:16 **EBP8[12:0]**：错误编号 8 的错误位位置 (Error bit position for error number 8)

位 15:13 保留，必须保持复位值。

位 12:0 **EBP7[12:0]**：错误编号 7 的错误位位置 (Error bit position for error number 7)

22.10 FMC 寄存器映射

表 168. FMC 寄存器映射

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	FMC_BCR1	FMCEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBLSET[1:0]		Res.	CCLKEN	CBURSTRW	CPSIZE[2:0]			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID[1:0]		MTYP[1:0]		MUXEN	MBKEN
	Reset value	0								0	0		0	0	0	0	0	0	0	0	0	0		1	1		1	0	1	1	0	1	1
0x08	FMC_BCR2	FMCEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBLSET[1:0]		Res.	CCLKEN	CBURSTRW	CPSIZE[2:0]			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID[1:0]		MTYP[1:0]		MUXEN	MBKEN
	Reset value	0								0	0		0	0	0	0	0	0	0	0	0	0		1	1		1	0	1	0	0	1	0
0x10	FMC_BCR3	FMCEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBLSET[1:0]		Res.	CCLKEN	CBURSTRW	CPSIZE[2:0]			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID[1:0]		MTYP[1:0]		MUXEN	MBKEN
	Reset value	0								0	0		0	0	0	0	0	0	0	0	0	0		1	1		1	0	1	0	0	1	0
0x18	FMC_BCR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBLSET[1:0]		Res.	CCLKEN	CBURSTRW	CPSIZE[2:0]			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID[1:0]		MTYP[1:0]		MUXEN	MBKEN
	Reset value	0								0	0		0	0	0	0	0	0	0	0	0	0		1	1		1	0	1	0	0	1	0
0x04	FMC_BTR1	DATAHLD		ACCMOD		DATLAT				CLKDIV[3:0]				BUSTURN				DATAST								ADDHLD				ADDSET			
	Reset value	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0C	FMC_BTR2	DATAHLD		ACCMOD		DATLAT				CLKDIV[3:0]				BUSTURN				DATAST								ADDHLD				ADDSET			
	Reset value	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x14	FMC_BTR3	DATAHLD		ACCMOD		DATLAT				CLKDIV[3:0]				BUSTURN				DATAST								ADDHLD				ADDSET			
	Reset value	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x1C	FMC_BTR4	DATAHLD		ACCMOD		DATLAT				CLKDIV[3:0]				BUSTURN				DATAST								ADDHLD				ADDSET			
	Reset value	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x20	FMC_PCSCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNTB4EN	CNTB3EN	CNTB2EN	CNTB1EN	CSCOUNT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x80	FMC_PCR	Res.	Res.	Res.	Res.	Res.	Res.	WEN	BCHECC	Res.	Res.	Res.	Res.	ECCSS[2:0]				TAR[3:0]				TCLR[3:0]			ECCALG	Res.	ECCEN	PWID		Res.	PBKEN	PWAITEN	Res.
	Reset value							0	0					0	1	1	1	1	1	1	1	1	1	1	0		0	0	0		0	0	
0x84	FMC_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NWRF	Res.	PEF	Res.	Res.	ISOST	
	Reset value																										1		1			1	1
0x88	FMC_PMEM	MEMHIZ								MEMHOLD								MEMWAIT								MEMSET							
	Reset value	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0
0x8C	FMC_PATT	ATTHIZ								ATTHOLD								ATTWAIT								ATTSET							
	Reset value	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0

表 168. FMC 寄存器映射（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x90	FMC_HPR	HPR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x94	FMC_HECCR	HECC[[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x98	Reserved	Reserved																															
0x104	FMC_BWTR1	DATAHLD		ACCMOD		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN				DATAST								ADDHLD				ADDSET			
	Reset value	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x10C	FMC_BWTR2	DATAHLD		ACCMOD		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN				DATAST								ADDHLD				ADDSET			
	Reset value	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x114	FMC_BWTR3	DATAHLD		ACCMOD		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN				DATAST								ADDHLD				ADDSET			
	Reset value	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x11C	FMC_BWTR4	DATAHLD		ACCMOD		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN				DATAST								ADDHLD				ADDSET			
	Reset value	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x180	FMC_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IFEE	IHLE	IREE
	Reset value																														0	0	0
0x184	FMC_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IFEF	IHLF	IREF
	Reset value																														0	0	0
0x188	FMC_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CIFEF	CIHLF	CIREF
	Reset value																														0	0	0
0x200	FMC_CSQCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSQSTART
	Reset value																																0
0x204	FMC_ CSQCFGR1	Res.	Res.	Res.	Res.	Res.	Res.	CMD2T	CMD1T	CMD2								CMD1								ACYNBR			Res.	Res.	DMADEN	CMD2EN	Res.
	Reset value							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	
0x208	FMC_ CSQCFGR2	Res.	Res.	Res.	Res.	Res.	Res.	RCMD2T	RCMD1T	RCMD2								RCMD1								Res.	Res.	Res.	Res.	Res.	DMADEN	RCMD2EN	SQSDTEN
	Reset value							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0
0x20C	FMC_ CSQCFGR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RAC2T	RAC1T	SDT	AC5T	AC4T	AC3T	AC2T	AC1T	Res.	Res.	SNBR						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value									0	0	0	0	0	0	0	0			0	0	0	0	0	0								
0x210	FMC_CSQAR1	ADDCA4								ADDC3								ADDC2								ADDC1							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x214	FMC_CSQAR2	SAO																Res.	Res.	Res.	Res.	NANDCEN1	NANDCEN0	Res.	Res.	ADDC5							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0					0	0			0	0	0	0	0	0	0	0

表 168. FMC 寄存器映射（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x220	FMC_CSQIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDTCIE	SUEIE	SEIE	SCIE	TCIE
	Reset value																												0	0	0	0	0
0x224	FMC_CSQISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDTCF	SUEF	SEF	SCF	TCF
	Reset value																												0	0	0	0	0
0x228	FMC_CSQICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCMDTCF	CSUEF	CSEF	CSCF	CTCF
	Reset value																												0	0	0	0	0
0x22C	Reserved	Reserved																															
0x230	FMC_CSQEMSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEM[15:0]														
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x240	Reserved	Reserved																															
0x250	FMC_BCHIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EPBRIE	DSRIE	DEFIE	DERIE	DUEIE
	Reset value																												0	0	0	0	0
0x254	FMC_BCHISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EPBRF	DSRF	DEFF	DERF	DUEF
	Reset value																												0	0	0	0	0
0x258	FMC_BCHICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CEPBRF	CDSRF	CDEFF	CDERF	CDUEF
	Reset value																												0	0	0	0	0
0x260	FMC_BCHPBR1	BCHPB[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x264	FMC_BCHPBR2	BCHPB[63:32]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x268	FMC_BCHPBR3	BCHPB[95:64]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x26C	FMC_BCHPBR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BCHPB[103:96]							
	Reset value																									0	0	0	0	0	0	0	0
0x27C	FMC_BCHDSR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEN					Res.	Res.	DEF	DUE
	Reset value																									0	0	0	0			0	0
0x280	FMC_BCHDSR1	Res.	Res.	Res.	EBP2[12:0]													Res.	Res.	Res.	EBP1[12:0]												
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0
0x284	FMC_BCHDSR2	Res.	Res.	Res.	EBP4[12:0]													Res.	Res.	Res.	EBP3[12:0]												
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0
0x288	FMC_BCHDSR3	Res.	Res.	Res.	EBP6[12:0]													Res.	Res.	Res.	EBP5[12:0]												
	Reset value																																
0x28C	FMC_BCHDSR4	Res.	Res.	Res.	EBP8[12:0]													Res.	Res.	Res.	EBP7[12:0]												
	Reset value																																

表 168. FMC 寄存器映射（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x3EC	FMC_HWCFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDRAM2_BASE [3:0]				SDRAM1_BASE 3:0]				NAND_BASE [3:0]				SDRAM_RBASE [3:0]				NOR_BASE [3:0]				RD_LN2PTH [3:0]			
	Reset value	0x00DC 8762																															
0x3F0	FMC_HWCFGR1	RA_LN2DPTH [3:0]				WR_LN2DPTH [3:0]				WD_LN2DPTH [3:0]				WA_LN2DPTH [3:0]				ID_SIZE[3:0]				Res.	Res.	Res.	SDRAM_SEL	Res.	Res.	Res.	NAND_ECC	Res.	Res.	Res.	NAND_SEL
	Reset value	0x2232 B011																															
0x3F4	FMC_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value	0x0000 0011																															
0x3F8	FMC_IPIDR	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0x3FC	FMC_SIDR	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 2.5 节*。

23 四线 SPI 接口 (QUADSPI)

23.1 简介

QUADSPI 是一种专用的通信接口，连接单线、双线或四线 SPI Flash 存储介质。该接口可以在以下三种模式下工作：

- 间接模式：使用 QUADSPI 寄存器执行全部操作。
- 自动状态轮询模式：周期性读取外部 Flash 状态寄存器，而且标志位置 1 时会产生中断。
- 内存映射模式：外部 Flash 映射到设备地址空间，从而系统将其视作内部内存。

采用双闪存模式时，将同时访问两个四线 SPI Flash，吞吐量和容量均可提高二倍。

23.2 QUADSPI 主要特性

- 三种功能模式：间接模式、自动状态轮询模式和内存映射模式
- 双闪存模式，通过并行访问两个 Flash，可同时发送/接收 8 位数据
- 支持 SDR 和 DDR 模式
- 针对间接模式和内存映射模式，完全可编程操作码
- 针对间接模式和内存映射模式，完全可编程帧格式
- 集成 FIFO，用于发送和接收
- 允许进行 8 位、16 位和 32 位数据访问
- 具有适用于间接模式操作的 DMA 通道
- 在达到 FIFO 阈值、超时、操作完成以及发生访问错误时产生中断

23.3 QUADSPI 功能描述

23.3.1 QUADSPI 框图

图 152. QUADSPI 框图

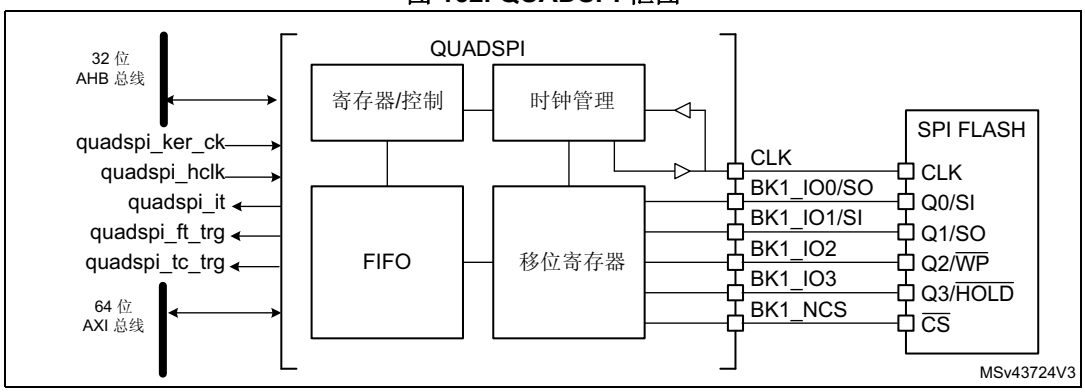
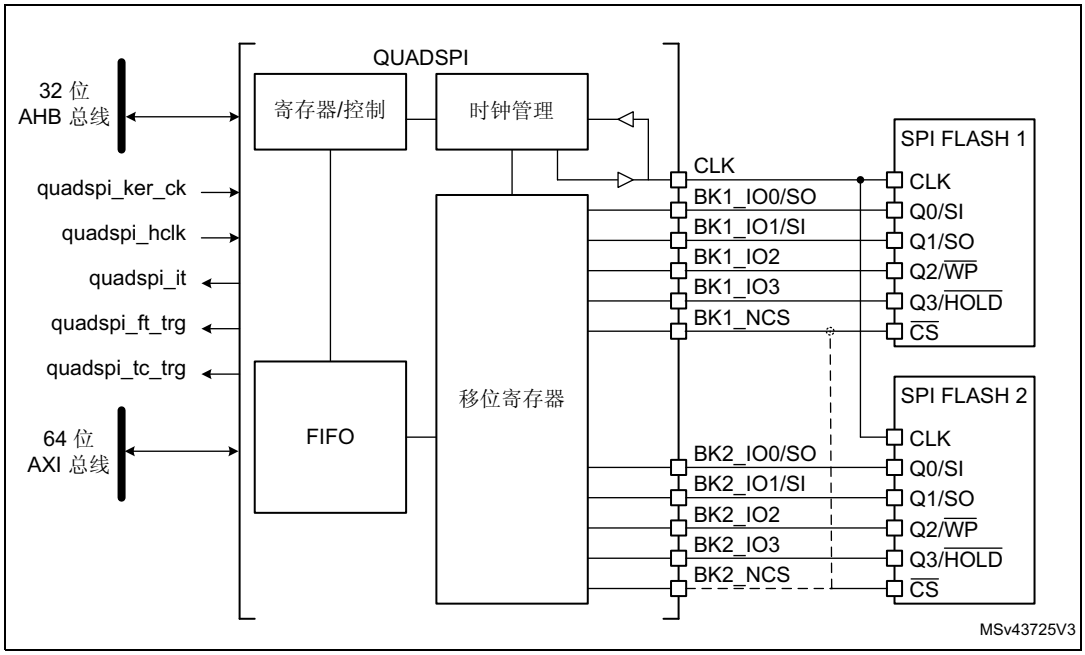


图 153. QUADSPI 功能框图（双闪存模式使能）



23.3.2 QUADSPI 引脚和内部信号

下表列出了 QUADSPI 内部信号。

表 169. QUADSPI 内部信号

信号名称	信号类型	说明
quadspi_ker_ck	数字输入	QUADSPI 内核时钟
quadspi_hclk	数字输入	QUADSPI 寄存器接口时钟
quadspi_it	数字输出	QUADSPI 全局中断
quadspi_ft_trg	数字输出	MDMA 的 QUADSPI FIFO 阈值触发信号
quadspi_tc_trg	数字输出	MDMA 的 QUADSPI 传输完成触发信号

下表列出了 QUADSPI 引脚，双闪存模式下使用 6 个信号连接单个 FLASH，使用 10 到 11 个信号连接两个 FLASH（FLASH 1 和 FLASH 2）。

表 170. QUADSPI 引脚

信号名称	信号类型	说明
CLK	数字输出	FLASH 1 和 FLASH 2 的时钟
BK1_IO0/SO	数字输入/输出	在双线/四线模式中为双向 IO，单线模式中为串行输出，适用于 FLASH 1
BK1_IO1/SI	数字输入/输出	在双线/四线模式中为双向 IO，单线模式中为串行输入，适用于 FLASH 1
BK1_IO2	数字输入/输出	在四线模式中为双向 IO，适用于 FLASH 1
BK1_IO3	数字输入/输出	在四线模式中为双向 IO，适用于 FLASH 1
BK2_IO0/SO	数字输入/输出	在双线/四线模式中为双向 IO，单线模式中为串行输出，适用于 FLASH 2

表 170. QUADSPI 引脚（续）

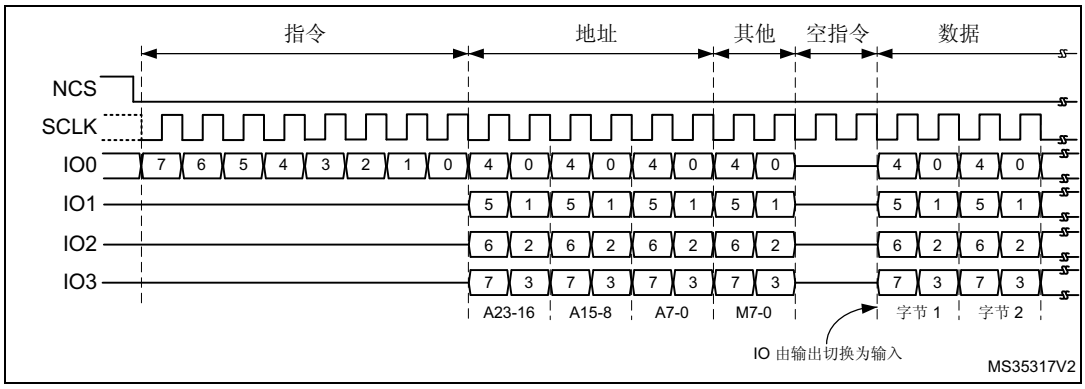
信号名称	信号类型	说明
BK2_IO1/SI	数字输入/输出	在双线/四线模式中为双向 IO，单线模式中为串行输入，适用于 FLASH 2
BK2_IO2	数字输入/输出	在四线模式中为双向 IO，适用于 FLASH 2
BK2_IO3	数字输入/输出	在四线模式中为双向 IO，适用于 FLASH 2
BK1_NCS	数字输出	片选（低电平有效），适用于 FLASH 1。如果 QUADSPI 始终在双闪存模式下工作，则其也可用于 FLASH 2。
BK2_NCS	数字输出	片选（低电平有效），适用于 FLASH 2。如果 QUADSPI 始终在双闪存模式下工作，则其也可用于 FLASH 1。

23.3.3 QUADSPI 命令序列

QUADSPI 通过命令与 Flash 通信。每条命令包括指令、地址、交替字节、空指令和数据这五个阶段。任一阶段均可跳过，但至少要包含指令、地址、交替字节或数据阶段之一。

NCS 在每条指令开始前下降，在每条指令完成后再次上升。

图 154. 四线 SPI 模式下的读命令示例



指令阶段

这一阶段，将在 QUADSPI_CCR[7:0] 寄存器的 INSTRUCTION 位字段中配置的一条 8 位指令发送到 Flash，指定待执行操作的类型。

大多数 FLASH 从 IO0/SO 信号（单线 SPI 模式）只能以一次 1 位的方式接收指令，但指令阶段可选择一次发送 2 位（在双线 SPI 模式中通过 IO0/IO1）或一次发送 4 位（在四线 SPI 模式中通过 IO0/IO1/IO2/IO3）。这可通过 QUADSPI_CCR[9:8] 寄存器中的 IMODE[1:0] 位字段进行配置。

若 IMODE = 00，则跳过指令阶段，命令序列从地址阶段（如果存在）开始。

地址阶段

在地址阶段，将 1-4 字节发送到 Flash，指示操作地址。待发送的地址字节数在 QUADSPI_CCR[13:12] 寄存器的 ADSIZE[1:0] 位字段中进行配置。在间接模式和自动状态轮询模式下，待发送的地址字节在 QUADSPI_AR 寄存器的 ADDRESS[31:0] 位字段中指定。在内存映射模式下，则通过 AHB（来自于 Cortex 或 DMA）直接给出地址。

地址阶段可一次发送 1 位（在单线 SPI 模式中通过 SO）、2 位（在双线 SPI 模式中通过 IO0/IO1）或 4 位（在四线 SPI 模式中通过 IO0/IO1/IO2/IO3）。这可通过 QUADSPI_CCR[11:10] 寄存器中的 ADMODE[1:0] 位字段进行配置。

若 ADMODE = 00，则跳过地址阶段，命令序列直接进入下一阶段（如果存在）。

交替字节阶段

在交替字节阶段，将 1-4 字节发送到 Flash，一般用于控制操作模式。待发送的交替字节数在 QUADSPI_CCR[17:16] 寄存器的 [1:0] 位字段中进行配置。待发送的字节在 QUADSPI_ABR 寄存器中指定。

交替字节阶段可一次发送 1 位（在单线 SPI 模式中通过 SO）、2 位（在双线 SPI 模式中通过 IO0/IO1）或 4 位（在四线 SPI 模式中通过 IO0/IO1/IO2/IO3）。这可通过 QUADSPI_CCR[15:14] 寄存器中的 ABMODE[1:0] 位字段进行配置。

若 ABMODE = 00，则跳过交替字节阶段，命令序列直接进入下一阶段（如果存在）。

交替字节阶段存在仅需发送单个半字节而不是一个全字节的情况，比如采用双线模式并且仅使用两个周期发送交替字节时。在这种情况下，固件可采用四线模式 (ABMODE = 11) 并发送一个字节，方法是 ALTERNATE 的位 7 和 3 置“1”（IO3 保持高电平）且位 6 和 2 置“0”（IO2 线保持低电平）。此时，半字节的高 2 位存放在 ALTERNATE 的位 4:3，低 2 位存放在位 1 和位 0 中。例如，如果半字节 2 (0010) 通过 IO0/IO1 发送，则 ALTERNATE 必须设置为 0x8A (1000_1010)。

空指令周期阶段

在空指令周期阶段，给定的 1 到 31 个周期内不发送或接收任何数据，目的是当采用更高的时钟频率时，给 FLASH 留出准备数据阶段的时间。这一阶段中给定的周期数在 QUADSPI_CCR[22:18] 寄存器的 DCYC[4:0] 位字段中指定。在 SDR 和 DDR 模式下，持续时间被指定为一定个数的全 CLK 周期。

若 DCYC 为零，则跳过空指令周期阶段，命令序列直接进入数据阶段（如果存在）。

空指令周期阶段的操作模式由 DMODE 确定。

为确保数据信号从输出模式转变为输入模式有足够的“转向”时间，使用双线和四线模式从 Flash 接收数据时，至少需要指定一个空指令周期。

数据阶段

在数据阶段，可从 Flash 接收或向其发送任意数量的字节。

在间接模式和自动轮询模式下，待发送/接收的字节数在 QUADSPI_DLR 寄存器中指定。

在间接写模式中，要发送到闪存的数据必须写入 QUADSPI_DR 寄存器。在间接读取模式下，从外部 Flash 接收的数据通过读取 QUADSPI_DR 获得。

在内存映射模式下，读取的数据通过 AHB 直接发送回 Cortex 或 DMA。

数据阶段可一次发送/接收 1 位（在单线 SPI 模式中通过 SO/SI）、2 位（在双线 SPI 模式中通过 IO0/IO1）或 4 位（在四线 SPI 模式中通过 IO0/IO1/IO2/IO3）。这可通过 QUADSPI_CCR[15:14] 寄存器中的 ABMODE[1:0] 位字段进行配置。

若 DMODE = 00，则跳过数据阶段，命令序列在拉高 NCS 时立即完成。此配置只能用于仅间接写入模式。

23.3.4 QUADSPI 信号接口协议模式

单线 SPI 模式

传统 SPI 模式允许串行发送/接收仅仅 1 位。在此模式下，数据通过 SO 信号（其 I/O 与 IO0 共享）发送到 Flash。从 Flash 接收到的数据通过 SI（其 I/O 与 IO1 共享）送达。

通过将（QUADSPI_CCR 中的）IMODE/ADMODE/ABMODE/DMODE 位字段设置为 01，可对不同的命令阶段分别进行配置，以使用此模式。

在每个已配置为双线 SPI 模式的阶段中：

- IO0 (SO) 处于输出模式。
- IO1 (SI) 处于输入模式（高阻抗）。
- IO2 处于输出模式并强制置 0。
- IO3 处于输出模式并强制置 1（以禁止“保持”功能）。

若 DMODE = 01，这对于空指令阶段也同样如此。

双线 SPI 模式

在双线 SPI 模式下，通过 IO0/IO1 信号同时发送/接收两位。

通过将 QUADSPI_CCR 寄存器的 IMODE/ADMODE/ABMODE/DMODE 位字段设置为 10，可对不同的命令阶段分别进行配置，以使用双线 SPI 模式。

在每个已配置为双线 SPI 模式的阶段中：

- IO0/IO1 在数据阶段进行读取操作时处于高阻态（输入），在其他情况下为输出。
- IO2 处于输出模式并强制置 0。
- IO3 处于输出模式并强制置 1。

在空指令阶段，若 DMODE = 01，则 IO0/IO1 始终保持高阻态。

四线 SPI 模式

在四线 SPI 模式下，通过 IO0/IO1/IO2/IO3 信号同时发送/接收四位。

通过将 QUADSPI_CCR 寄存器的 IMODE/ADMODE/ABMODE/DMODE 位字段设置为 11，可对不同的命令阶段分别进行配置，以使用四线 SPI 模式。

在每个已配置为此模式的阶段中，IO0/IO1/IO2/IO3 在数据阶段进行读取操作时均处于高阻态（输入），在其他情况下为输出。

在空指令阶段中，若 DMODE = 11，则 IO0/IO1/IO2/IO3 均为高阻态。

IO2 和 IO3 仅用于四线 SPI 模式。如果未配置任何阶段使用四线 SPI 模式，即使 QUADSPI 激活，对应 IO2 和 IO3 的引脚也可用于其他功能。

SDR 模式

默认情况下，DDRM 位 (QUADSPI_CCR[31]) 为 0，QUADSPI 在单倍数据速率 (SDR) 模式下工作。

在 SDR 模式下，当 QUADSPI 驱动 IO0/SO、IO1、IO2、IO3 信号时，这些信号仅在 CLK 的下降沿发生转变。

在 SDR 模式下接收数据时，QUADSPI 假定 Flash 也通过 CLK 的下降沿发送数据。默认情况下（SSHIFT = 0 时），将使用 CLK 后续的边沿（上升沿）对信号进行采样。

DDR 模式

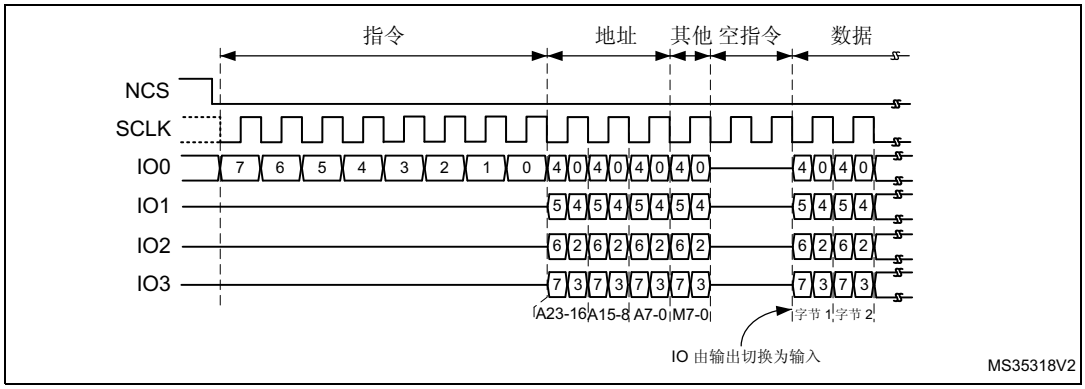
若 DDRM 位 (QUADSPI_CCR[31]) 置 1，则 QUADSPI 在双倍数据速率 (DDR) 模式下工作。

在 DDR 模式下，当 QUADSPI 在地址/交替字节/数据阶段驱动 IO0/SO、IO1、IO2、IO3 信号时，将在 CLK 的每个上升沿和下降沿发送 1 位。

指令阶段不受 DDRM 影响。指令始终使用 CLK 下降沿发送。

在 DDR 模式下接收数据时，QUADSPI 假定 Flash 也通过 CLK 的上升沿和下降沿均发送数据。若 DDRM = 1，固件必须清零 SSHIFT 位（QUADSPI_CR 的位 4）。因此，在半个 CLK 周期后（下一个反向边沿）对信号采样。

图 155. 四线模式下 DDR 命令示例



双闪存模式

若 DFM 位（QUADSPI_CR 的位 6）为 1，QUADSPI 处于双闪存模式。QUADSPI 使用两个外部四线 SPI FLASH（FLASH 1 和 FLASH 2），在每个周期中发送/接收 8 位（在 DDR 模式下为 16 位），能够有效地将吞吐量和容量扩大一倍。

每个 Flash 使用同一个 CLK 并可选择使用同一个 NCS 信号，但其 IO0、IO1、IO2 和 IO3 信号是各自独立的。

双闪存模式可与单线、双线以及四线 SPI 模式结合使用，也可与 SDR 或 DDR 模式相结合。

Flash 的大小在 FSIZE[4:0] (QUADSPI_DCR[20:16]) 中指定，指定的值应能够反映 Flash 的总容量，即单个组件容量的 2 倍。

如果地址 X 为偶数，QUADSPI 赋给地址 X 的字节是存放于 FLASH 1 的地址 X/2 中的字节，QUADSPI 赋给地址 X+1 的字节是存放于 FLASH 2 的地址 X/2 中的字节。也就是说，偶地址中的字节存储于 FLASH 1，奇地址中的字节存储于 FLASH 2。

在双闪存模式下读取 Flash 状态寄存器时，需要读取的字节数是单闪存模式下的 2 倍。这意味着在状态寄存器获取指令到达后，如果每个 Flash 给出 8 个有效位，则 QUADSPI 必须配置为 2 个字节（16 位）的数据长度，它将从每个 Flash 接收 1 个字节。如果每个 Flash 给出一个 16 位的状态，则 QUADSPI 必须配置为读取 4 字节，以在双闪存模式下可获取两个 Flash 的所有状态位。结果（在数据寄存器中）的最低有效字节是 FLASH 1 状态寄存器的最低有效字节，而下一个字节是 FLASH 2 状态寄存器的最低有效字节。数据寄存器的第三个字节是 FLASH 1 的第二个字节，第四个字节是 FLASH 2 的第二个字节（Flash 具有 16 位状态寄存器时）。

偶数个字节必须始终在双闪存模式下访问。因此，若 DRM = 1，则数据长度字段 (QUADSPI_DLR[0]) 的位 0 始终保持为 1。

在双闪存模式下，FLASH 1 接口信号的行为基本上与正常模式下相同。在指令、地址、交替字节以及空指令周期阶段，FLASH 2 接口信号具有与 FLASH 1 接口信号完全相同的波形。也就是说，每个 Flash 总是接收相同的指令与地址。然后，在数据阶段，BK1_IOx 和 BK2_IOx 总线并行传输数据，但发送到 FLASH 1（或从其接收）的数据与 FLASH 2 中的不同。

23.3.5 QUADSPI 间接模式

在间接模式下，通过写入 QUADSPI 寄存器来触发命令；并通过读写数据寄存器来传输数据，就如同对待其他通信外设那样。

若 FMODE = 00 (QUADSPI_CCR[27:26])，则 QUADSPI 处于间接写入模式，字节在数据阶段中发送到 Flash。通过写入数据寄存器 (QUADSPI_DR) 的方式提供数据。

若 FMODE = 01，则 QUADSPI 处于间接读取模式，在数据阶段中从 Flash 接收字节。通过读取 QUADSPI_DR 来获取数据。

读取/写入的字节数在数据长度寄存器 (QUADSPI_DLR) 中指定。如果 QUADSPI_DLR = 0xFFFF_FFFF（全为“1”），则数据长度视为未定义，QUADSPI 将继续传输数据，直到到达（由 FSIZE 定义的）Flash 的结尾。如果不传输任何字节，DMODE (QUADSPI_CCR[25:24]) 必须设置为 00。

如果 QUADSPI_DLR = 0xFFFF_FFFF 并且 FSIZE = 0x1F（最大值指示一个 4 GB 的 Flash），在此特殊情况下，传输将无限继续下去，仅在出现终止请求或 QUADSPI 被禁止后停止。在读取最后一个内存地址后（地址为 0xFFFF_FFFF），将从地址 = 0x0000_0000 开始继续读取。

当发送或接收的字节数达到编程设定值时，如果 TCIE = 1，则 TCF 置 1 并产生中断。在数据数量不确定的情况下，将根据 QUADSPI_CR 中定义的 Flash 大小，在达到外部 SPI 的限制时，TCF 置 1。

触发命令启动

从本质上讲，在固件给出命令所需的最后一点信息时，命令即会启动。根据 QUADSPI 的配置，在间接模式下有三种触发命令启动的方式。在出现以下情形时，命令立即启动：

- 对 INSTRUCTION[7:0] (QUADSPI_CCR) 执行写入操作，如果没有地址是必需的（当 ADMODE = 00）并且不需要固件提供数据（当 FMODE = 01 或 DMODE = 00）
- 对 ADDRESS[31:0] (QUADSPI_AR) 执行写入操作，如果地址是必需的（当 ADMODE ≠ 00）并且不需要固件提供数据（当 FMODE = 01 或 DMODE = 00）
- 对 DATA[31:0] (QUADSPI_DR) 执行写入操作后，如果需要固件提供数据（当 FMODE = 00 并且 DMODE != 00）

写入交替字节寄存器 (QUADSPI_ABR) 始终不会触发命令启动。如果需要交替字节，必须预先进行编程。

如果命令启动，BUSY（QUADSPI_SR 的位 5）将自动置 1。

FIFO 和数据管理

在间接模式中，数据通过 QUADSPI 内部的 32 字节 FIFO 传输。FLEVEL[5:0] (QUADSPI_SR[13:8]) 指示 FIFO 目前保存的字节数。

在间接写入模式下 (FMODE = 00)，当数据写入到 QUADSPI_DR 中时，固件会将其添加到 FIFO 中。字写入将在 FIFO 中增加 4 个字节，半字写入增加 2 个字节，而字节写入仅增加 1 个字节。如果固件在 FIFO 中加入的数据过多（超过 DL[31:0] 指示的值），将在写入操作结束（TCF 置 1）时从 FIFO 中清除超出的字节。

对 QUADSPI_DR 的字节/半字访问必须仅针对该 32 位寄存器的最低有效字节/半字。

FTHRES[4:0] 用于定义 FIFO 的阈值。如果达到阈值，FTF（FIFO 阈值标志）置 1。在间接读取模式下，从 FIFO 中读取的有效字节数超过阈值时，FTF 置 1。从 Flash 中读取最后一个字节后，如果 FIFO 中依然有数据，则无论 FTHRES 的设置为何，FTF 也都会置 1。在间接写入模式下，当 FIFO 中的空字节数超过阈值时，FTF 置 1。

如果 FTIE = 1，则 FTF 置 1 时产生中断。如果 DMAEN = 1，则 FTF 置 1 时启动 DMA 传送。如果阈值条件不再为“真”（CPU 或 DMA 传输了足够的数据后），则 FTF 由硬件清零。

在间接模式下，当 FIFO 已满，QUADSPI 将暂时停止从 Flash 读取字节以避免上溢。请注意，只有在 FIFO 中的 4 个字节为空 (FLEVEL ≤ 11) 时才会重新开始读取 FLASH。因此，若 FTHRES ≥ 29，应用程序必须读取足够的字节以确保 QUADSPI 再次从 FLASH 检索数据。否则，只要 28 < FLEVEL < FTHRES，FTF 标志就会保持为 0。

23.3.6 QUADSPI 自动状态轮询模式

在自动轮询模式下，QUADSPI 周期性启动命令以读取一定数量的状态字节（最多 4 个）。可屏蔽接收的字节以隔离一些状态位，从而在所选的位具有定义的值时可产生中断。

对 Flash 的访问最初与在间接读取模式下相同：如果不需要地址 (AMODE = 00)，则在写入 QUADSPI_CCR 时即开始访问。否则，如果需要地址，则在写入 QUADSPI_AR 时开始第一次访问。BUSY 在此时变为高电平，即使在周期性访问期间也保持不变。

在自动轮询模式下，MASK[31:0] (QUADSPI_PSMAR) 的内容用于屏蔽来自 Flash 的数据。如果 MASK[n] = 0，则屏蔽结果的位 n，从而不考虑此位。如果 MASK[n] = 1 并且位 [n] 的内容与 MATCH[n] (QUADSPI_PSMAR) 相同，说明存在位 n 匹配。

如果轮询匹配模式位（QUADSPI_CR 的位 23）为 0，将激活“AND”匹配模式。这意味着状态匹配标志 (SMF) 仅在全部未屏蔽位均存在匹配时置 1。

如果 PMM = 1，则激活“OR”匹配模式。这意味着 SMF 在任意未屏蔽位存在匹配时置 1。

如果 SMIE = 1，则在 SMF 置 1 时调用一个中断。

如果自动轮询模式停止 (APMS) 位置 1，则操作停止并且 BUSY 位在检测到匹配时清零。否则，BUSY 位保持为“1”，在发生中止或禁止 QUADSPI (EN = 0) 前继续进行周期性访问。

数据寄存器 (QUADSPI_DR) 包含最新接收的状态字节（FIFO 停用）。数据寄存器的内容不受匹配逻辑所用屏蔽方法的影响。FTF 状态位在新一次状态读取完成后置 1，并且 FTF 在数据读取后清零。

23.3.7 QUADSPI 内存映射模式

在配置为内存映射模式时，外部 SPI 设备被视为内部内存。

QUADSPI 外设若没有正确配置并使能，禁止访问四线 SPI Flash 的 bank 区域。

即使 Flash 容量更大，寻址空间也无法超过 256 MB。

如果访问的地址超出 FSIZE 定义的范围但仍在 256 MB 范围内，则生成总线错误。此错误的影响具体取决于尝试进行访问的总线主设备：

- 如果为 Cortex CPU，则会在使能总线故障时发生总线故障异常，在禁止总线故障时发生硬性故障异常
- 如果为 DMA，则生成 DMA 传输错误，并自动禁用相应的 DMA 通道。

支持字节、半字和字访问类型。

支持芯片内执行 (XIP) 操作，QUADSPI 预期下一次访问并提前加载后续地址中的字节。如果之后访问的是连续地址，由于值已经预取，访问将更快完成。

默认情况下，即便在很长时间内不访问 Flash，QUADSPI 也不会停止预取操作，之前的读取操作将保持激活状态并且 NCS 保持低电平。由于 NCS 保持低电平时，Flash 功耗增加，应用程序可能会激活超时计数器（TCEN = 1，QUADSPI_CR 的位 3），从而在 FIFO 中写满预取的数据后，若在 TIMEOUT[15:0] (QUADSPI_LPTR) 个周期的时长内没有访问，则释放 NCS。

BUSY 在第一次内存映射访问发生时变为高电平。由于进行预取操作，BUSY 在发生超时、中止或外设禁止前不会下降。

23.3.8 QUADSPI 自由运行时钟模式

当配置为自由运行时钟模式时，QUADSPI 不断输出时钟以进行测试和校准。

QUADSPI_CCR 寄存器中的 FRCM 位置 1 后，立即进入自由运行时钟模式。该模式的退出方式是将 QUADSPI_CR 的 ABORT 位置 1。

当 QUADSPI 以自由运行时钟模式运行时：

- 时钟连续运行
- NCS 保持高电平（取消选择外部设备）
- 释放数据线（高阻态）
- QUADSPI_SR 的 BUSY 标志置 1

23.3.9 QUADSPI Flash 配置

设备配置寄存器 (QUADSPI_DCR) 可用于指定外部 SPI Flash 的特性。

FSIZE[4:0] 位字段使用下面的公式定义外部内存的大小：

Flash 中的字节数 = $2^{[FSIZE+1]}$

FSIZE+1 是对 Flash 寻址所需的地址位数。在间接模式下，设备容量最高可达 4 GB（使用 32 位进行寻址），但在内存映射模式下，可寻址空间限制为 256 MB。

如果 DFM = 1，FSIZE 表示两个 Flash 容量的总和。

QUADSPI 连续执行两条命令时，它在两条命令之间将 NCS 置为高电平默认仅一个 CLK 周期时长。如果 Flash 需要命令之间的时间更长，可使用 CSHT 位字段指定 NCS 必须保持高电平的最少 CLK 周期数（最大为 8）。

时钟模式 (CKMODE) 位指示命令之间的 CLK 信号逻辑电平（NCS = 1 时）。

23.3.10 QUADSPI 延迟数据采样

默认情况下，QUADSPI 在 Flash 驱动信号后半个 CLK 周期才对 Flash 驱动的数据采样。

在外部信号延迟时，这有利于推迟数据采样。使用 SSHIFT 位（QUADSPI_CR 的位 4），可将数据采样移位半个 CLK 周期。

DDR 模式下不支持时钟移位：若 DDRM 位置 1，SSHIFT 位必须清零。

23.3.11 QUADSPI 配置

QUADSPI 配置分两个阶段：

1. QUADSPI 外设配置
2. QUADSPI Flash 配置

QUADSPI 在配置完毕并使能后，即可在间接模式、自动状态轮询模式和内存映射模式这三种操作模式之一下工作。

QUADSPI 配置

QUADSPI 使用 QUADSPI_CR 进行配置。用户必须配置传入数据的时钟预分频器的分频系数以及采样移位设置。

DDR 模式可通过 DDRM 位进行设置。在 DDR 模式下设置四线 SPI 接口时，内核时钟的内部分频器的分频比必须设置为 2 或更大的值。使能该模式后，在每个时钟的上升沿和下降沿都会发送地址和交替字节以及发送/接收数据。无论 DDRM 位如何设置，都将在 SDR 模式下发送指令。

DMA 请求通过 DMAEN 位置 1 使能。若是用于中断，则相关使能位也可在该阶段置 1。

生成 DMA 请求或生成中断的 FIFO 电平在 FTHRES 位中进行编程。

如果需要超时计数器，则可将 TCEN 位置 1 并在 QUADSPI_LPTR 寄存器中编程超时值。

双闪存模式可通过将 DFM 置 1 来激活。

QUADSPI Flash 配置

与外部目标 Flash 相关的参数通过 QUADSPI_DCR 寄存器进行配置。用户必须在 FSIZE 位中编程 Flash 的大小、在 CSHT 位中编程片选保持高电平的最短时间以及在 MODE 位中编程功能模式（模式 0 或模式 3）。

23.3.12 QUADSPI 使用

使用 FMODE[1:0] (QUADSPI_CCR[27:26]) 选择工作模式。

间接模式

FMODE 编程为 00 可选择间接写入模式，将数据发送到 Flash。FMODE 编程为 01 可选择间接读取模式，读取 Flash 中的数据。

QUADSPI 用于间接模式时，采用以下方式构建帧：

1. 在 QUADSPI_DLR 中指定待读取或写入的字节数。
2. 在 QUADSPI_CCR 中指定帧格式、模式和指令代码。
3. 在 QUADSPI_ABR 中指定要在地址阶段后立即发送的可选交替字节。
4. 在 QUADSPI_CR 中指定工作模式。若 FMODE = 00（间接写入模式）并且 DMAEN = 1，则应在 QUADSPI_CR 前指定 QUADSPI_AR。否则在 QUADSPI_AR 更新前（如果已经使能 DMA 控制器），DMA 便可以写入 QUADSPI_DR。
5. 在 QUADSPI_AR 中指定目标地址。
6. 通过 QUADSPI_DR 从 FIFO 读取数据/向 FIFO 写入数据。

在写入 QUADSPI_CR 时，用户可指定以下设置：

- 使能位 (EN) 设置为 1
- DMA 使能位 (DMAEN)，用于向/从 RAM 传输/接收数据
- 超时计数器使能位 (TCEN)
- 采样移位设置 (SSHIFT)
- FIFO 阈值 (FTRHES)，以指示 FTF 标志在何时置 1
- 中断使能
- 自动状态轮询模式参数：匹配模式和停止模式（在 FMODE = 11 时有效）
- 时钟预分频器

在写入 QUADSPI_CCR 时，用户可指定以下参数：

- 通过 INSTRUCTION 位指定指令字节
- 通过 IMODE 位指定指令发送方式（1/2/4 线）
- 通过 ADMODE 位指定地址发送方式（无/1/2/4 线）
- 通过 ADSIZE 位指定地址长度（8/16/24/32 位）
- 通过 ABMODE 位指定交替字节发送方式（无/1/2/4 线）
- 通过 ABSIZE 位指定交替字节数 (1/2/3/4)
- 通过 DBMODE 位指定是否存在空指令字节
- 通过 DCYC 位指定空指令字节数
- 通过 DMODE 位指定数据发送/接收方式（无/1/2/4 线）

如果无需为某个命令更新 QUADSPI_AR 与 QUADSPI_DR，则在写入 QUADSPI_CCR 时，该命令序列便立即启动。在 ADMODE 和 DMODE 均为 00 时，或在间接读取模式 (FMODE = 01) 下仅 ADMODE = 00 时，便属于此情况。

在需要地址（ADMODE 不为 00），但无需写入数据寄存器（FMODE = 01 或 DMODE = 00）时，通过写入 QUADSPI_AR 更新地址后，命令序列便立即启动。

在数据传输（FMODE = 00 并且 DMODE != 00）中，通过 QUADSPI_DR 写入 FIFO 触发通信启动。

自动状态轮询模式

将 FMODE 位字段 (QUADSPI_CCR[27:26]) 设置为 10，使能此模式。在此模式下，发送编程的帧并周期性检索数据。

每帧中读取的最大数据量为 4 字节。如果 QUADSPI_DLR 请求更多的数据，则忽略多余的部分并仅读取 4 个字节。

在 QUADSPI_PISR 寄存器中指定周期性。

在检索到状态数据后，可在内部进行处理：

- 将状态匹配标志位置 1，如果使能，还将产生中断
- 自动停止周期性检索状态字节

接收到的值可通过存储于 QUADSPI_PSMKR 中的值进行屏蔽，并与存储在 QUADSPI_PSMAR 中的值进行或运算或与运算。

若是存在匹配，则状态匹配标志置 1，并且在使能了中断的情况下还将产生中断；如果 AMPS 位置 1，则 QUADSPI 自动停止。

在任何情况下，最新的检索值都在 OCTOSPI_DR 中可用。

内存映射模式

在内存映射模式下，外部 Flash 被视为内部内存，只是存在访问延迟。在该模式下，仅允许对外部 Flash 执行读取操作。

将 QUADSPI_CCR 中的 FMODE 设置为 11 可进入内存映射模式。

当主设备访问内存映射空间时,将发送已编程的指令和帧。

FIFO 用作预取缓冲区以接受线性读取。在此模式中，对于 QUADSPI_DR 的任何访问均返回零。

QUADSPI_DLR 在内存映射模式下不起作用。

23.3.13 指令仅发送一次

一些 Flash（例如：Winbound）提供了一种模式，指令在该模式中仅通过第一个命令序列进行发送，后续的命令根据地址直接启动。用户通过使用 SIOO 位 (QUADSPI_CCR[28]) 可利用此功能的优势。

SIOO 对于所有功能模式（间接模式、自动状态轮询模式和内存映射模式）均有效。如果 SIOO 位置 1，则在对 QUADSPI_CCR 执行写入操作后，仅针对第一条命令发送指令。后续命令序列都将跳过指令阶段，直到 QUADSPI_CCR 被写入为止。

在 IMODE = 00（无指令）时，SIOO 不起作用。

23.3.14 QUADSPI 错误管理

在以下情况下可能产生错误：

- 在间接模式或自动状态轮询模式下，如果在 QUADSPI_AR 中编程了错误的地址（根据 QUADSPI_DCR 中 FSIZE[4:0] 定义的 Flash 大小）：TEF 将置 1，如果使能，还将产生中断。

  另外，在间接模式下，如果地址加数据的长度超过 Flash 的大小，TEF 将在访问被触发时置 1。
- 在内存映射模式下，当主设备执行的访问超出范围或 QUADSPI 被禁止时：将产生总线错误，以响应故障总线主设备请求。

  当主设备访问内存映射空间，但内存映射模式被禁止时：将产生总线错误，以响应故障总线主设备请求。

23.3.15 QUADSPI 的 BUSY 位和中止功能

在 QUADSPI 启动对 Flash 的操作时，QUADSPI_SR 中的 BUSY 位自动置 1。

在间接模式下，在 QUADSPI 完成了请求的命令序列并且 FIFO 为空时，BUSY 复位。

在自动状态轮询模式下，仅当最后一次周期性访问完成时（因 APMS = 1 时发生匹配，或因中止），BUSY 位才变为低电平。

在内存映射模式下进行第一次访问后，仅在发生超时事件或中止时 BUSY 位变为低电平。

任何操作都可以通过将 QUADSPI_CR 中的 ABORT 置 1 来中止。在完成中止后，BUSY 和 ABORT 自动复位，FIFO 清空。

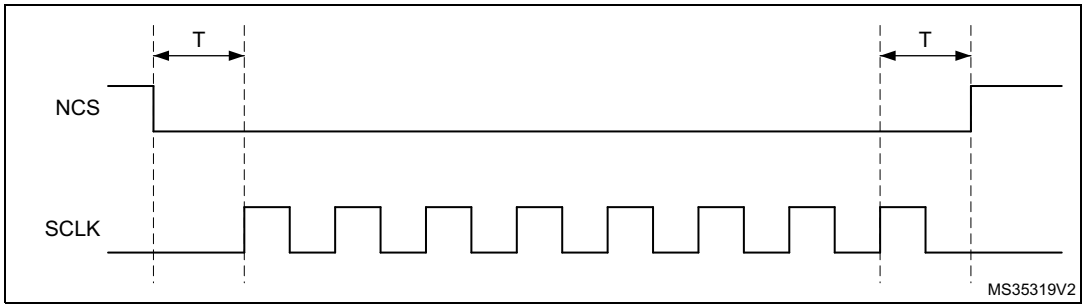
注： *如果中止对状态寄存器的写入操作，有些 Flash 可能发生错误行为。*

23.3.16 NCS 行为

默认情况下，NCS 为高电平，取消选择外部 Flash。NCS 在操作开始前降至低电平，在操作完成时立即升至高电平。

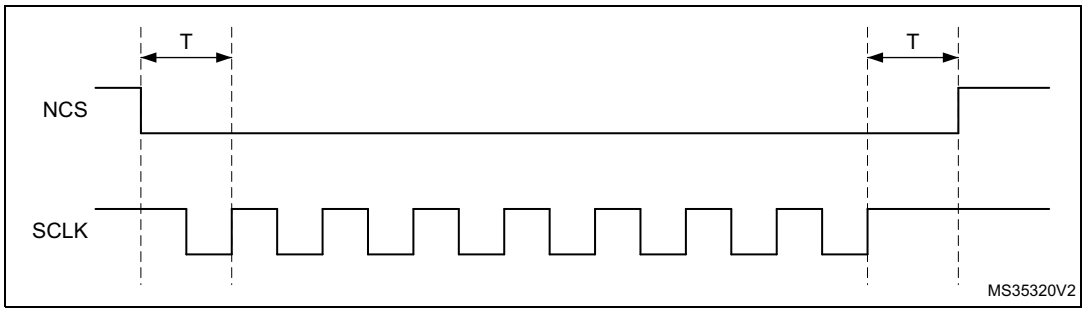
当 CKMODE = 0（“模式 0”，在未进行任何操作时 CLK 保持低电平）时，NCS 在操作首次升高 CLK 边沿时的一个 CLK 周期前降至低电平，在操作最后一次升高 CLK 边沿时的一个 CLK 周期后升至高电平，如下图所示。

图 156. CKMODE = 0 时的 NCS（T = CLK 周期）



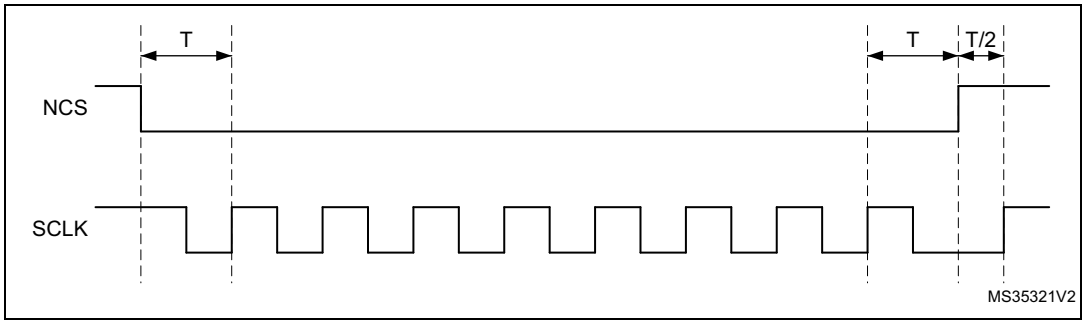
当 CKMODE = 1（“模式 3”，在未进行任何操作时 CLK 升至高电平）且 DDRM = 0（SDR 模式）时，NCS 仍在操作首次升高 CLK 边沿时的一个 CLK 周期前降至低电平，在操作最后一次升高 CLK 边沿时的一个 CLK 周期后升至高电平，如下图所示。

图 157. SDR 模式下 CKMODE = 1 时的 NCS（T = CLK 周期）



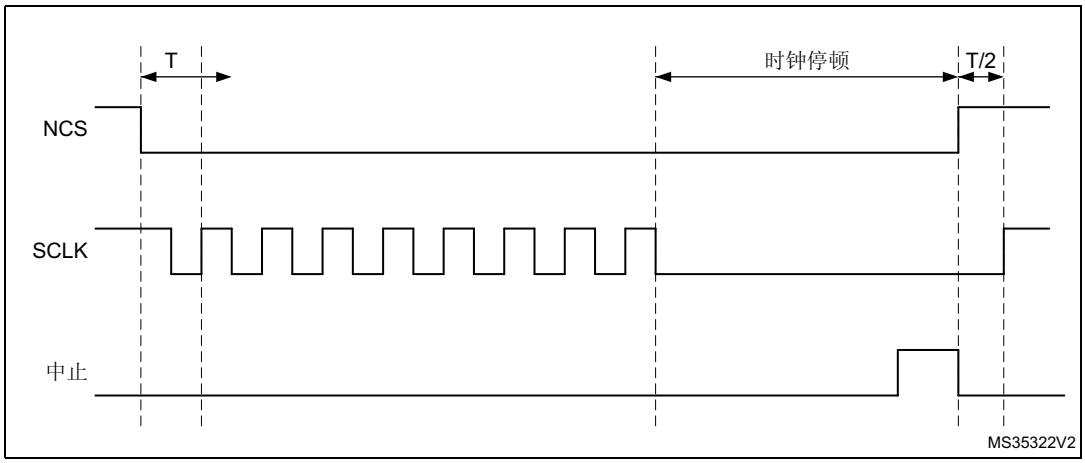
当 CKMODE = 1（模式 3）且 DDRM = 1（DDR 模式）时，NCS 在操作首次升高 CLK 边沿时的一个 CLK 周期前降至低电平，在操作最后一次升高 CLK 边沿时的一个 CLK 周期后升至高电平，如下图所示。由于 DDR 操作必须伴随下降沿完成，当 NCS 变为高电平时，CLK 为低电平并在经过半个周期后恢复高电平。

图 158. DDR 模式下 CKMODE = 1 时的 NCS（T = CLK 周期）



若 FIFO 在读取操作中保持写满状态或在写入操作中保持为空，则在固件干预 FIFO 前，操作停止并且 CLK 保持低电平。若操作停止时发生中止，则 NCS 在请求中止后即升至高电平，CLK 则在半个 CLK 周期后升至高电平，如*图 159* 中所示。

图 159. CKMODE = 1 且发生中止时的 NCS（T = CLK 周期）



不处于双闪存模式 (DFM = 0) 且 FSEL = 0（默认值）时，仅访问 FLASH 1，因此 BK2_nCS 保持高电平，若 FSEL = 1，则仅访问 FLASH 2 且 BK1_NCS 保持高电平。在双闪存模式下，BK2_NCS 的行为与 BK1_NCS 完全相同。因此，如果存在 FLASH 2 并且应用程序仅处于双闪存模式，则 BK1_NCS 信号也可用于 FLASH 2，而专用于 BK2_NCS 的引脚可用于其他功能。

23.4 QUADSPI 中断

发生如下事件时可生成中断：

- 超时
- 状态匹配
- FIFO 阈值
- 传输完成
- 传输错误

可以使用单独的中断使能位以提高灵活性。

表 171. QUADSPI 中断请求

中断事件	事件标志	使能控制位
超时	TOF	TOIE
状态匹配	SMF	SMIE
FIFO 阈值	FTF	FTIE
传输完成	TCF	TCIE
传输错误	TEF	TEIE

23.5 QUADSPI 寄存器

23.5.1 QUADSPI 控制寄存器 (QUADSPI_CR)

QUADSPI control register

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESCALER[7:0]								PMM	APMS	Res.	TOIE	SMIE	FTIE	TCIE	TEIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	FTHRES[4:0]					FSEL	DFM	Res.	SSHIFT	TCEN	DMAEN	ABORT	EN
			rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:24 **PRESCALER[7:0]**：时钟预分频器 (Clock prescaler)

该位字段定义基于 quadspi_ker_ck 时钟生成 CLK 所用的分频系数（值 + 1）。

0：$F_{CLK} = F_{quadspi_ker_ck}$，quadspi_ker_ck 时钟直接用作 QUADSPI CLK（预分频器被旁路）

1：$F_{CLK} = F_{quadspi_ker_ck}/2$

2：$F_{CLK} = F_{quadspi_ker_ck}/3$

...

255：$F_{CLK} = F_{quadspi_ker_ck}/256$

对于奇数时钟分频系数，CLK 的占空比并非 50%。时钟信号的低电平持续时间比高电平持续时间多一个周期。

在 DDR 模式下设置四线 SPI 接口时，预分频器的分频比必须设置为 2 或更大的值。

注意：仅可在 BUSY = 0 时修改该位字段。

位 23 **PMM**：轮询匹配模式 (Polling match mode)

此位指示在自动状态轮询模式期间用来确定是否“匹配”的方法。

0：AND 匹配模式。如果从 Flash 接收的所有未屏蔽位均与匹配寄存器中的对应位相匹配，则 SMF 置 1。

1：OR 匹配模式。如果从 Flash 接收的任意一个未屏蔽位与匹配寄存器中的对应位相匹配，则 SMF 置 1。

注意：仅可在 BUSY = 0 时修改此位。

位 22 **APMS**：自动状态轮询模式停止 (Automatic status-polling mode stop)

此位确定在匹配后自动状态轮询是否停止。

0：仅通过中止或禁止 QUADSPI 停止自动状态轮询模式。

1：发生匹配时，自动状态轮询模式停止。

注意：仅可在 BUSY = 0 时修改此位。

位 21 保留，必须保持复位值。

位 20 **TOIE**：超时中断使能 (Timeout interrupt enable)

该位使能超时中断。

0：禁止中断

1：使能中断

位 19 **SMIE**：状态匹配中断使能
该位使能状态匹配中断。
0：禁止中断
1：使能中断

位 18 **FTIE**：FIFO 阈值中断使能 (FIFO threshold interrupt enable)
该位使能 FIFO 阈值中断。
0：禁止中断
1：使能中断

位 17 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)
该位使能传输完成中断。
0：禁止中断
1：使能中断

位 16 **TEIE**：传输错误中断使能 (Transfer error interrupt enable)
该位使能传输错误中断。
0：禁止中断
1：使能中断

位 15:13 保留，必须保持复位值。

位 12:8 **FTHRES[4:0]**：FIFO 阈值级别 (FIFO threshold level)
此位字段定义在间接模式下 FIFO 中将导致 FIFO 阈值标志（QUADSPI_SR 中的 FTF 位）置 1 的字节数阈值。
0：在间接写入模式 (FMODE = 00) 下，如果 FIFO 中存在 1 个或更多空闲字节位置，则 FTF 置 1；或者在间接读取模式 (FMODE = 01) 下，如果 FIFO 中存在 1 个或更多有效字节可供读取，则 FTF 置 1。
1：在间接写入模式 (FMODE = 00) 下，如果 FIFO 中存在 2 个或更多空闲字节位置，则 FTF 置 1；或者在间接读取模式 (FMODE = 01) 下，如果 FIFO 中存在 2 个或更多有效字节可供读取，则 FTF 置 1。
...
31：在间接写入模式 (FMODE = 00) 下，如果 FIFO 中存在 32 个空闲字节位置，则 FTF 置 1；或者在间接读取模式 (FMODE = 01) 下，如果 FIFO 中存在 32 个有效字节可供读取，则 FTF 置 1。
如果 DMAEN = 1，则在更改 FTHRES 值前，必须禁止相应通道的 DMA 控制器。

位 7 **FSEL**：Flash 选择
该位选择单闪存模式 (DFM = 0) 下要寻址的 Flash。
0：选择 FLASH 1
1：选择 FLASH 2
注意：仅可在 BUSY = 0 时修改此位。在 DFM = 1 时忽略该位。

位 6 **DFM**：双闪存模式
该位激活双闪存模式，同时使用两个外部 Flash 以将吞吐量和容量扩大一倍。
0：禁止双闪存模式
1：使能双闪存模式
注意：仅可在 BUSY = 0 时修改此位。

位 5 保留，必须保持复位值。

位 4 **SSHIFT**：采样移位

默认情况下，QUADSPI 在 Flash 驱动数据后过半个 CLK 周期开始采集数据。使用此位，可考虑外部信号延迟，推迟数据采集。

0：不移位

1：移位半个周期

在 DDR 模式下 (DDRM = 1)，固件必须确保 SSHIFT = 0。

注意：仅可在 BUSY = 0 时修改该位字段。

位 3 **TCEN**：超时计数器使能 (Timeout counter enable)

此位仅在内存映射模式 (FMODE = 11) 下有效。激活此位后，如果在一段时间（通过 TIMEOUT[15:0] (QUADSPI_LPTR) 定义）内一直没有进行访问，将释放 NCS（从而降低功耗）。该位使能超时计数器。

默认情况下，即便在很长时间内不访问 Flash，QUADSPI 也不会停止预取操作，之前的读取操作将保持激活状态并且 NCS 保持低电平。由于 NCS 保持低电平时，FLASH 功耗增加，应用程序可能会激活超时计数器（TCEN = 1，QUADSPI_CR 的位 3）。从而在 FIFO 中写满预取的数据后，若在 TIMEOUT[15:0] (QUADSPI_LPTR) 个周期的时长内没有访问，则释放 NCS。

0：禁止超时计数器，在内存映射模式中进行访问后，NCS 保持激活

1：使能超时计数器，在内存映射模式下，Flash 持续不活动 TIMEOUT[15:0] 个周期后释放 NCS。

注意：仅可在 BUSY = 0 时修改此位。

位 2 **DMAEN**：DMA 使能

在间接模式下，通过 QUADSPI_DR 寄存器可使用 DMA 输入或输出数据。FIFO 阈值标志 FTF 置 1 时，将触发 DMA 传输。

0：间接模式下禁止 DMA

1：间接模式下使能 DMA

位 1 **ABORT**：中止请求 (Abort request)

该位中止执行中的命令序列。在中止完成时自动复位。该位可停止当前的传输。

在自动状态轮询模式或内存映射模式下，此位也用以复位 APM 位或 DM 位。

0：不请求中止

1：请求中止

位 0 **EN**：QUADSPI 使能

0：QUADSPI 禁止

1：使能 QUADSPI

23.5.2 QUADSPI 设备配置寄存器 (QUADSPI_DCR)

QUADSPI device configuration register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSIZE[4:0]				
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CSHT[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKMODE
					rw	rw	rw								rw

位 31:21 保留，必须保持复位值。

位 20:16 **FSIZE[4:0]**：Flash 大小

此位字段使用下面的公式定义了外部内存的大小：

Flash 中的字节数 = $2^{[FSIZE+1]}$

FSIZE+1 是对 Flash 寻址所需的地址位数。在间接模式下，设备容量最高可达 4 GB（使用 32 位进行寻址），但在内存映射模式下，可寻址空间限制为 256 MB。

如果 DFM = 1，FSIZE 表示两个 Flash 容量的总和。

注意：仅可在 BUSY = 0 时修改该位字段。

位 15:11 保留，必须保持复位值。

位 10:8 **CSHT[2:0]**：片选高电平时间 (Chip select high time)

CSHT+1 定义片选 (NCS) 在发送至 Flash 的命令之间必须保持高电平的最少 CLK 周期数。

0：NCS 在 Flash 命令之间保持高电平至少 1 个周期

1：NCS 在 Flash 命令之间保持高电平至少 2 个周期

...

7：NCS 在 Flash 命令之间保持高电平至少 8 个周期

注意：仅可在 BUSY = 0 时修改该位字段。

位 7:1 保留，必须保持复位值。

位 0 **CKMODE**：模式 0/模式 3

此位指示 CLK 在命令之间（NCS = 1 时）的电平。

0：NCS 为高电平（片选释放）时，CLK 必须保持低电平。这称为模式 0。

1：NCS 为高电平（片选释放）时，CLK 必须保持高电平。这称为模式 3。

注意：仅可在 BUSY = 0 时修改该位字段。

23.5.3 QUADSPI 状态寄存器 (QUADSPI_SR)

QUADSPI status register

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	FLEVEL[5:0]						Res.	Res.	BUSY	TOF	SMF	FTF	TCF	TEF
		r	r	r	r	r	r			r	r	r	r	r	r

位 31:14 保留，必须保持复位值。

位 13:8 **FLEVEL[5:0]**：FIFO 级别

该位字段指定 FIFO 中保留的有效字节数。如果 FIFO 为空，则 FLEVEL = 0；如果 FIFO 已满，则 FLEVEL = 32。在内存映射模式和自动状态轮询模式下，FLEVEL 为零

位 7:6 保留，必须保持复位值。

位 5 **BUSY**：忙

操作进行时，此位置 1。在对 Flash 的操作完成并且 FIFO 为空时，此位自动清零。

位 4 **TOF**：超时标志 (Timeout flag)

发生超时时，该位置 1。向 CTOF 写入 1 可将该位清零。

位 3 **SMF**：状态匹配标志 (Status match flag)

在自动状态轮询模式下，若未屏蔽的接收数据与匹配寄存器 (OCTOSPI_PSMAR) 中的对应位相匹配，则该位置 1。向 CSMF 写入 1 可将此位清零。

位 2 **FTF**：FIFO 阈值标志 (FIFO threshold flag)

在间接模式下，若达到 FIFO 阈值，或从 Flash 读取完成后，FIFO 中留有数据时，此位置 1。只要阈值条件不再为“真”，此位就自动清零。

在自动状态轮询模式下，每次读取状态寄存器时，该位即置 1，读取数据寄存器时，该位清零。

位 1 **TCF**：传输完成标志 (Transfer complete flag)

在间接模式下，当传输的数据数量达到编程设定值，或在任何模式下传输中止时，此位置 1。向 CTCF 写入 1 时，此位清零。

位 0 **TEF**：传输错误标志 (Transfer error flag)

在间接模式下访问无效地址时，该位置 1。向 CTEF 写入 1 可将此位清零。

23.5.4 QUADSPI 标志清零寄存器 (QUADSPI_FCR)

QUADSPI flag clear register

偏移地址：0x00C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTOF	CSMF	Res.	CTCF	CTEF
											w	w		w	w

位 31:5 保留，必须保持复位值。

位 4 **CTOF**：清零超时标志
写入 1 可将 QUADSPI_SR 寄存器中的 TOF 标志清零

位 3 **CSMF**：清零状态匹配标志 (Clear status match flag)
写入 1 可将 QUADSPI_SR 寄存器中的 SMF 标志清零

位 2 保留，必须保持复位值。

位 1 **CTCF**：清零传输完成标志 (Clear transfer complete flag)
写入 1 可将 QUADSPI_SR 寄存器中的 TCF 标志清零

位 0 **CTEF**：清零传输错误标志 (Clear transfer error flag)
写入 1 可将 QUADSPI_SR 寄存器中的 TEF 标志清零

23.5.5 QUADSPI 数据长度寄存器 (QUADSPI_DLR)

QUADSPI data length register

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DL[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DL[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DL[31:0]**：数据长度 (Data length)

在间接模式和自动状态轮询模式下待检索的数据量（值 + 1）。对自动状态轮询模式必须使用不大于 3 的值（表示 4 字节）。

在间接模式下，所有位置 1 表示未定义长度，QUADSPI 将继续传输数据直到到达由 FSIZE 定义的内存末尾。

0x0000_0000：传输 1 个字节

0x0000_0001：传输 2 个字节

0x0000_0002：传输 3 个字节

0x0000_0003：传输 4 个字节

...

0xFFFF_FFFD：传输 4,294,967,294 (4G-2) 个字节

0xFFFF_FFFE：传输 4,294,967,295 (4G-1) 个字节

0xFFFF_FFFF：未定义长度 -- 传输所有字节直到到达由 FSIZE 定义的 Flash 的结尾。如果 FSIZE = 0x1F，则读取无限继续下去。

在双闪存模式 (DFM = 1) 下，即使该位写入“0”，DL[0] 也始终保持为“1”，因此保证了每次访问均传输偶数个字节。

此位字段在内存映射模式 (FMODE = 10) 下不起作用。

注意：仅可在 BUSY = 0 时修改该位字段。

23.5.6 QUADSPI 通信配置寄存器 (QUADSPI_CCR)

QUADSPI communication configuration register

偏移地址：0x014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DDRM	DHHC	FRCM	SIOO	FMODE[1:0]		DMODE[1:0]		Res.	DCYC[4:0]					ABSIZE[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABMODE[1:0]		ADSIZE[1:0]		ADMODE[1:0]		IMODE[1:0]		INSTRUCTION[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **DDRM**：双倍数据速率模式 (Double data rate mode)

此位为地址、交替字节和数据阶段设置 DDR 模式：

0：禁止 DDR 模式

1：使能 DDR 模式

注意：仅可在 BUSY = 0 时写入该位。

位 30 **DHHC**：DDR 保持

DDR 模式下数据输出延迟 1/4 个 QUADSPI 输出时钟周期：

0：使用模拟延迟延迟数据输出

1：数据输出延迟 1/4 个 QUADSPI 输出时钟周期。

该特性仅在 DDR 模式下激活。

注意：仅可在 BUSY = 0 时写入该位。

当 DHHC = 1 时，PRESCALER > 0 是必须的。

位 29 **FRCM**：自由运行时钟模式
该位置 1 时，无论 FMODE 位如何，QUADSPI 外设均进入自由运行时钟模式。
0：正常模式
1：自由运行时钟模式
注意：仅可在 BUSY = 0 时写入该位。

位 28 **SIOO**：仅发送指令一次模式 (Send instruction only once mode)
当 IMODE = 00 时，此位无效。详见*第 23.3.13 节*。
0：在每个事务中发送指令
1：仅为第一条命令发送指令
注意：仅可在 BUSY = 0 时写入该位。

位 27:26 **FMODE[1:0]**：功能模式
该位字段定义 OCTOSPI 操作的功能模式。
00：间接写入模式
01：间接读取模式
10：自动状态轮询模式
11：内存映射模式
如果 DMAEN = 1，则在更改 FMODE 值前，必须禁止相应通道的 DMA 控制器。
注意：仅可在 BUSY = 0 时修改该位字段。

位 25:24 **DMODE[1:0]**：数据模式
该位字段定义数据阶段的操作模式：
00：无数据
01：单线传输数据
10：双线传输数据
11：四线传输数据
此位字段还定义空指令阶段的操作模式。
注意：仅可在 BUSY = 0 时修改该位字段。

位 23 保留，必须保持复位值。

位 22:18 **DCYC[4:0]**：空指令周期数 (Number of dummy cycles)
此位字段定义空指令阶段的持续时间。在 SDR 和 DDR 模式下，它指定 CLK 周期数 (0-31)。
注意：仅可在 BUSY = 0 时修改该位字段。

位 17:16 **ABSIZE[1:0]**：交替字节大小
此位定义替代字节的大小。
00：8 位交替字节
01：16 位交替字节
10：24 位交替字节
11：32 位交替字节
注意：仅可在 BUSY = 0 时修改该位字段。

位 15:14 **ABMODE[1:0]**：交替字节模式
该位字段定义交替字节阶段的操作模式。
00：无交替字节
01：单线传输交替字节
10：双线传输交替字节
11：四线传输交替字节
注意：仅可在 BUSY = 0 时修改该位字段。

位 13:12 **ADSIZE[1:0]**：地址长度
此位定义地址长度：
00：8 位地址
01：16 位地址
10：24 位地址
11：32 位地址
注意：仅可在 BUSY = 0 时修改该位字段。

位 11:10 **ADMODE[1:0]**：地址模式
该位字段定义地址阶段的操作模式。
00：无地址
01：单线传输地址
10：双线传输地址
11：四线传输地址
注意：仅可在 BUSY = 0 时修改该位字段。

位 9:8 **IMODE[1:0]**：指令模式
该位字段定义指令阶段的操作模式。
00：无指令
01：单线传输指令
10：双线传输指令
11：四线传输指令
注意：仅可在 BUSY = 0 时修改该位字段。

位 7:0 **INSTRUCTION[7:0]**：指令 (Instruction)
指定要发送到外部 SPI 设备的指令。
注意：仅可在 BUSY = 0 时修改该位字段。

23.5.7 QUADSPI 地址寄存器 (QUADSPI_AR)

QUADSPI address register

偏移地址：0x018

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRESS[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **ADDRESS[31:0]**：地址
此位字段包含要发送到外部 Flash 的地址。
BUSY = 1 或 FMODE = 11（内存映射模式）时，将忽略写入此位字段。
在双闪存模式下，由于地址始终为偶地址，ADDRESS[0] 自动保持为“0”

23.5.8 QUADSPI 交替字节寄存器 (QUADSPI_ABR)

QUADSPI alternate-byte register

偏移地址：0x01C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALTERNATE[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTERNATE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **ALTERNATE[31:0]**：交替字节 (Alternate bytes)

指定要在地址后立即发送到外部 SPI 设备的可选数据。

注意：仅可在 BUSY = 0 时修改该位字段。

23.5.9 QUADSPI 数据寄存器 (QUADSPI_DR)

QUADSPI data register

偏移地址：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DATA[31:0]**：数据 (Data)

指定要与外部 SPI 设备交换的数据。

在间接写入模式下，写入该寄存器的数据在数据阶段发送到 Flash，在此之前则存储于 FIFO。如果 FIFO 太满，将暂停写入，直到 FIFO 具有足够的空间接受要写入的数据才继续。

在间接模式下，读取该寄存器可获得（通过 FIFO）已从 Flash 接收的数据。如果 FIFO 所含字节数比读取操作要求的字节数少并且 BUSY=1，将暂停读取，直到足够的数据出现或传输完成（不分先后）才继续。

在自动状态轮询模式下，该寄存器包含最后从 Flash 读取的数据（未进行屏蔽）。

支持对该寄存器进行字、半字以及字节访问。在间接写入模式下，字节写入将在 FIFO 中增加 1 个字节，半字写入增加 2 个，而字写入则增加 4 个。类似地，在间接读取模式下，字节读取将擦除 FIFO 中的 1 个字节，半字读取擦除 2 个，而字读取则擦除 4 个。间接模式下的访问必须与此寄存器的最低位对齐：字节读取必须读取 DATA[7:0] 而半字读取必须读取 DATA[15:0]。

23.5.10 QUADSPI 轮询状态屏蔽寄存器 (QUADSPI _PSMKR)

QUADSPI polling status mask register

偏移地址：0x024

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MASK[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **MASK[31:0]**：状态屏蔽

对在自动状态轮询模式下接收的状态字节进行屏蔽。

对于位 n：

0：屏蔽在自动状态轮询模式下所接收数据的位 n，在匹配逻辑中不考虑其值。

1：不屏蔽在自动状态轮询模式下所接收数据的位 n，在匹配逻辑中考虑其值。

注意：仅可在 BUSY = 0 时修改该位字段。

23.5.11 QUADSPI 轮询状态匹配寄存器 (QUADSPI _PSMAR)

QUADSPI polling status match register

偏移地址：0x028

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MATCH[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MATCH[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **MATCH[31:0]**：状态匹配

该值将与屏蔽状态寄存器比较以进行匹配。

注意：仅可在 BUSY = 0 时修改该位字段。

23.5.12 QUADSPI 轮询间隔寄存器 (QUADSPI_PIR)

QUADSPI polling interval register

偏移地址：0x02C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERVAL[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **INTERVAL[15:0]**：轮询间隔

自动状态轮询阶段读取操作之间的 CLK 周期数。

注意：仅可在 BUSY = 0 时修改该位字段。

23.5.13 QUADSPI 低功耗超时寄存器 (QUADSPI_LPTR)

QUADSPI low-power timeout register

偏移地址：0x030

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEOUT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **TIMEOUT[15:0]**：超时时长 (Timeout period)

在内存映射模式下，每次访问结束后，QUADSPI 将预取后续字节并将其存放在 FIFO 中。该位字段指示在时钟变为无效后，QUADSPI 等待多少个 CLK 时钟周期才让 NCS 升至高电平，将外部设备置为低功耗状态。

注意：仅可在 BUSY = 0 时修改该位字段。

23.5.14 QUADSPI 硬件配置寄存器 (QUADSPI_HWCFGR)

QUADSPI hardware configuration register

偏移地址：0x3F0

复位值：0x0000 B058

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDLENGTH[3:0]				PRESCVAL[3:0]				FIFOPTR[3:0]				FIFOSIZE[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:12 **IDLENGTH[3:0]**：ID 长度
此位字段返回 AXI ID 的长度。

位 11:8 **PRESCVAL[3:0]**：预分频器复位值
此位字段返回预分频器的复位值。

位 7:4 **FIFOPTR[3:0]**：FIFO 指针大小
此位字段返回 FIFO 指针的位大小。

位 3:0 **FIFOSIZE[3:0]**：FIFO 大小
此位字段返回 FIFO 的字大小。

23.5.15 QUADSPI 版本寄存器 (QUADSPI_VERR)

QUADSPI version register

偏移地址：0x3F4

复位值：0x0000 0042

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：QUADSPI 主版本

位 3:0 **MINREV[3:0]**：QUADSPI 次版本

23.5.16 QUADSPI 标识寄存器 (QUADSPI_IPIDR)

QUADSPI identification register

偏移地址：0x3F8

复位值：0x0014 0031

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：QUADSPI 标识码

23.5.17 QUADSPI 大小标识寄存器 (QUADSPI_SIDR)

QUADSPI size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：QUADSPI 大小和标识码

23.5.18 QUADSPI 寄存器映射

表 172. QUADSPI 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	QUADSPI_CR	PRESCALER[7:0]								PMM	APMS	Res.	TOIE	SMIE	FTIE	TCIE	TEIE	Res.	Res.	Res.	FTHRES[4:0]					FSEL	DFM	Res.	SSHIFT	TCEN	DMAEN	ABORT	EN
	Reset value	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0				0	0	0	0	0	0	0		0	0	0	0	0
0x004	QUADSPI_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSIZE[4:0]					Res.	Res.	Res.	Res.	Res.	CSHT			Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKMODE
	Reset value												0	0	0	0	0						0	0	0								0
0x008	QUADSPI_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLEVEL[5:0]						Res.	Res.	BUSY	TOF	SMF	FTF	TCF	TEF
	Reset value																			0	0	0	0	0	0			0	0	0	0	0	0

表 172. QUADSPI 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00C	QUADSPI_FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTOF	CSMF	Res.	CTCF	CTEF
	Reset value																												0	0		0	0
0x010	QUADSPI_DLR	DL[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014	QUADSPI_CCR	DDRM	DHHC	FRCM	SIOO	FMODE [1:0]		DMODE [1:0]		Res.	DCYC[4:0]					ABSIZE [1:0]		ABMODE [1:0]		ADSIZE [1:0]		ADMODE [1:0]		IMODE [1:0]		INSTRUCTION[7:0]							
	Reset value	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	QUADSPI_AR	ADDRESS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x01C	QUADSPI_ABR	ALTERNATE[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	QUADSPI_DR	DATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	QUADSPI_ PSMKR	MASK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x028	QUADSPI_ PSMAR	MATCH[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x02C	QUADSPI_PIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTERVAL[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x030	QUADSPI_ LPTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMEOUT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x034-0x3EC	Reserved	Reserved																															
0x3F0	QUADSPI_ HWCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDLENGTH [3:0]				PRESCVAL [3:0]				FIFOPTR [3:0]				FIFOSIZE [3:0]			
	Reset value	0x0000 B058																															
0x3F4	QUADSPI _VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV [3:0]				MINREV [3:0]		
	Reset value	0x0000 0042																															
0x3F8	QUADSPI_IDR	ID[31:0]																															
	Reset value	0x0014 0031																															
0x3FC	QUADSPI_SIDR	SID[31:0]																															
	Reset value	0xA3C5 DD01																															

有关寄存器边界地址的信息，请参见*第 2.5 节*。

24 安全数字输入/输出多媒体卡接口 (SDMMC)

24.1 SDMMC 主要特性

SD/SDIO 嵌入式多媒体卡 (*e*•MMC) 主机接口 (SDMMC) 提供 AHB 总线与 SD 存储卡、SDIO 卡以及 *e*•MMC 设备之间的接口。

多媒体卡协会网站 www.jedec.org 中提供了由 MMCA 技术委员会发布的多媒体卡系统规范。

SD 卡协会网站 www.sdcard.org 中提供了 SD 存储卡和 SD I/O 卡系统规范。

SDMMC 具有以下特性：

- 兼容*嵌入式多媒体卡系统规范版本 5.1*。
  卡支持三种不同数据总线模式：1 位（默认）、4 位和 8 位。
  （HS200 SDMMC_CK 速度限制为允许的最大 I/O 速度）（不支持 HS400）。
- 完全兼容先前版本的多媒体卡（向后兼容性）
- 完全兼容 *SD 存储卡规范版本 6.0*。
  （SDR104 SDMMC_CK 速度限制为允许的最大 I/O 速度，不支持 SPI 模式和 UHS-II 模式）。
- 完全兼容 *SDIO 卡规范版本 4.0*。
  卡支持两种不同数据总线模式：1 位（默认）和 4 位。
  （SDR104 SDMMC_CK 速度限制为允许的最大 I/O 速度，不支持 SPI 模式和 UHS-II 模式）。
- 对于 8 位模式，数据传输高达 208 MB/s。
  （取决于允许的最大 I/O 速度）。
- 数据和命令输出使能信号，控制外部双向驱动程序。
- 支持 IDMA 链表

多媒体卡/SD 总线将卡连接到主机。

当前版本的 SDMMC 每次只支持一个 SD/SDIO/*e*•MMC 卡，但支持多个 *e*•MMC。

24.2 SDMMC 实现

表 173. SDMMC 特性

SDMMC 模式/特性(1)	SDMMC1	SDMMC2
可变延迟	X	X
SDMMC_CKIN	X	X
SDMMC_CDIR、SDMMC_D0DIR	X	X
SDMMC_D123DIR	X	X

1. X = 支持。

24.3 SDMMC 总线拓扑

总线的通信基于命令/响应和数据传输。

SD/SDIO/*e*•MMC 总线上的基本事务是命令/响应事务。这些类型的总线事务直接在命令或响应结构中传输其信息。此外，某些操作具有数据令牌。

数据传输通过以下方式完成：

- 块模式：块大小为 2^N 字节的数据块，N 为 0-14 范围内的值。
- SDIO 多字节模式：块大小为 1-512 字节的单一数据块
- *e*•MMC 流模式：连续数据流

与 *e*•MMC 卡的数据相互传输在数据块或流中执行。

图 160. SDMMC“无响应”和“无数据”操作

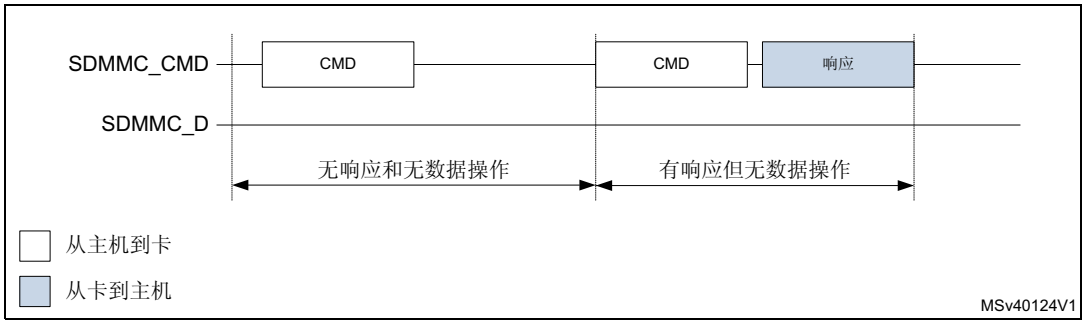
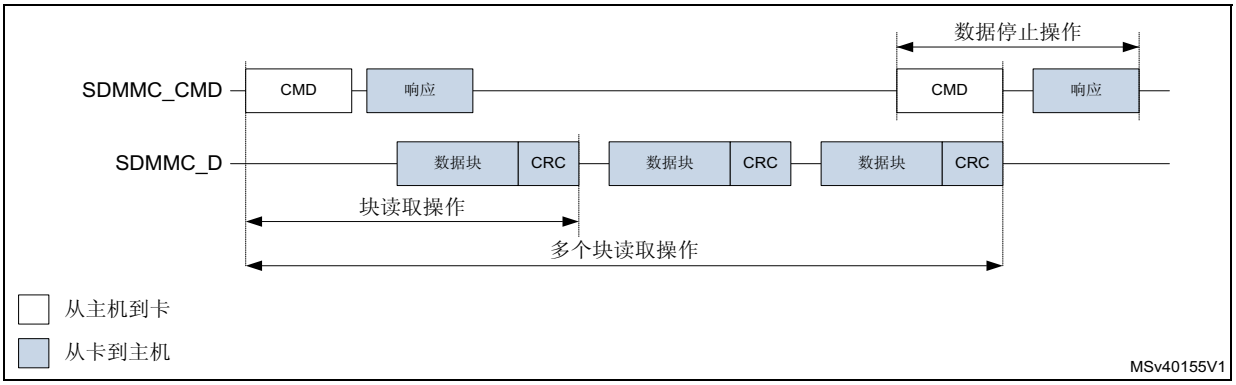
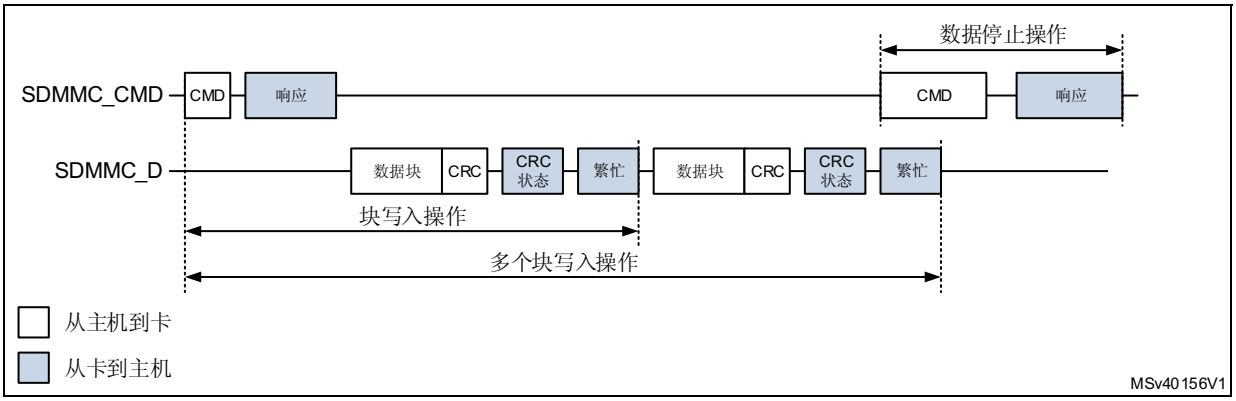


图 161. SDMMC（多个）块读取操作



注：　在具有预定义块数的 e•MMC 多块读取操作结束时，不需要停止传输命令。

图 162. SDMMC（多个）块写入操作



注：　在具有预定义块数的 e•MMC 多块写入操作结束时，不需要停止传输命令。

注：　只要发出“繁忙”信号（SDMMC_D0 被拉到低电平），SDMMC 便不会发送任何数据。

图 163. SDMMC（连续）流读取操作

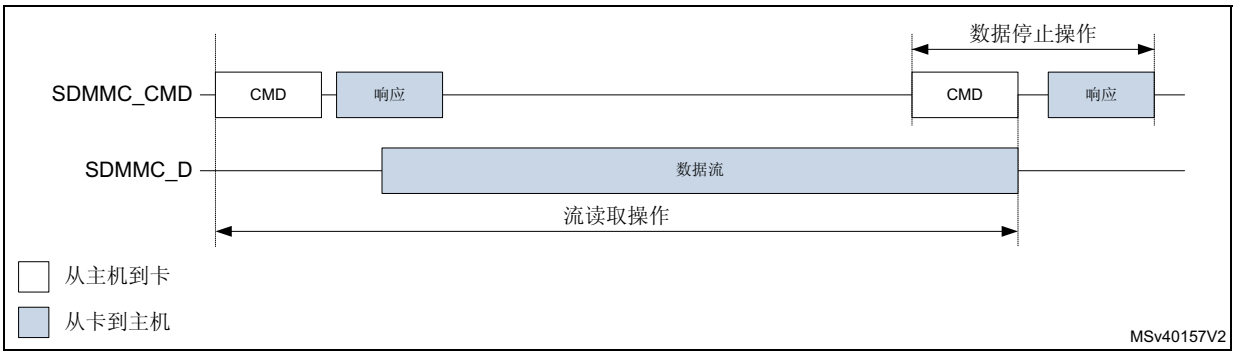
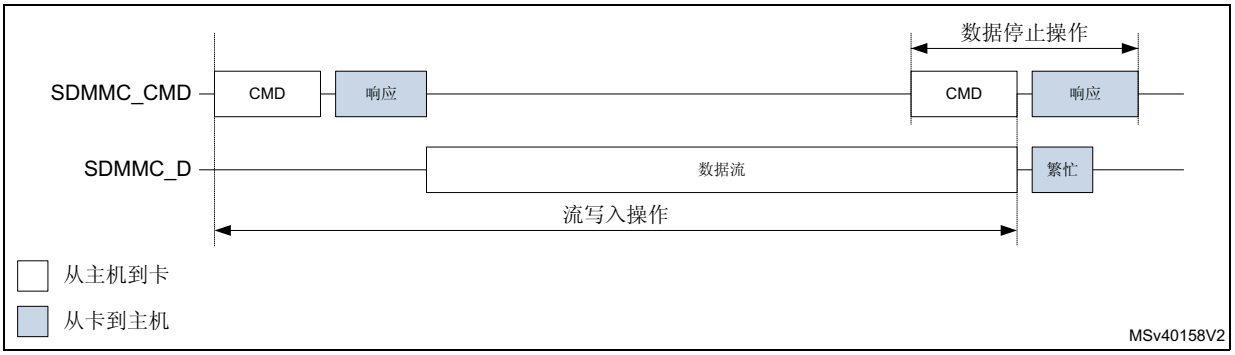


图 164. SDMMC（连续）流写入操作



数据流传输操作只适用于总线配置位宽为 1，且工作模式为单倍速率模式（DS、HS 和 SDR）下的 SDMMC_D0 上。

24.4 SDMMC 工作模式

表 174. SDMMC 工作模式 SD 和 SDIO

SDIO 总线速度模式(1)(2)	最大总线速度 (3) [MB/s]	最大时钟频率 [MHz](4)	信号电压 [V]
DS（默认速度）	12.5	25	3.3
HS（高速）	25	50	3.3
SDR12	12.5	25	1.8
SDR25	25	50	1.8
DDR50	50	50	1.8
SDR50	50	100	1.8
SDR104	104	208	1.8

1. SDR 单倍数据速率信号传输。
2. DDR 双倍数据速率信号传输。（在 SDMMC_CK 的两种时钟边沿上均采样数据）。
3. SDIO 总线速度，4 位总线宽度。
4. 最大频率取决于允许的最大 IO 速度。

SDR104 模式需要使用一个可变延迟来支持采样点的调节，而对于 SDR50 模式来说，这个可变延迟则是可选的。

表 175. SDMMC 工作模式 *e*•MMC

e•MMC 总线速度模式 (1)(2)	最大总线速度 (3) [MB/s]	最大时钟频率 [MHz](4)	信号电压 [V](5)
传统兼容型	26	26	3/1.8/1.2V
高速 SDR	52	52	3/1.8/1.2V
高速 DDR	104	52	3/1.8/1.2V
高速 HS200	200	200	1.8/1.2V

1. SDR 单倍数据速率信号传输。
2. DDR 双倍数据速率信号传输。（在 SDMMC_CK 的两种时钟边沿上均采样数据）。
3. *e*•MMC 总线速度，8 位总线宽度。
4. 最大频率取决于允许的最大 I/O 速度。
5. 支持的信号电压取决于 I/O 端口特性，请参见设备数据手册。

HS200 模式需要使用一个可变延迟来支持采样点的调节。

24.5 SDMMC 功能描述

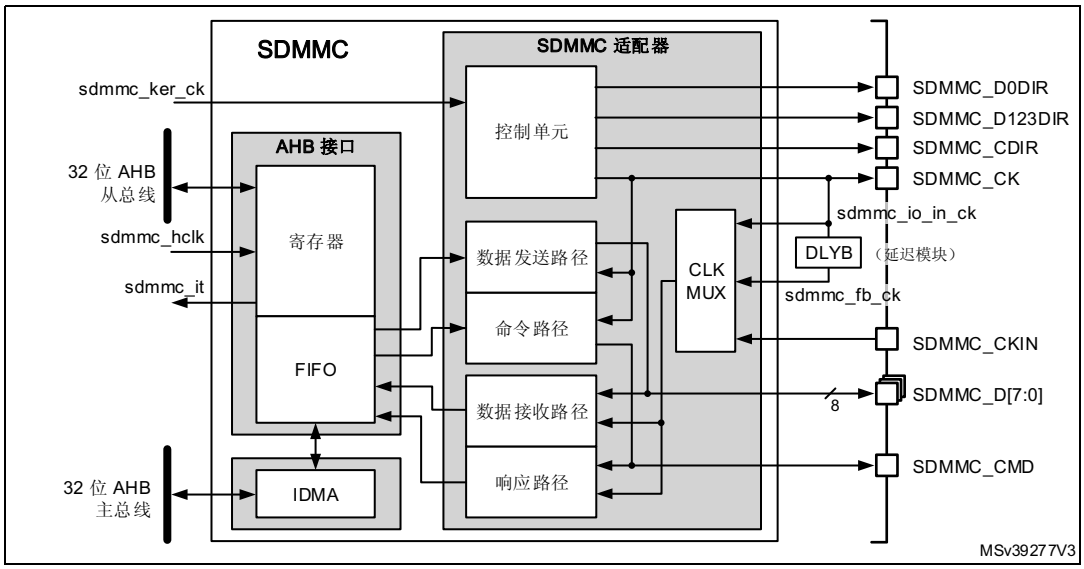
SDMMC 由四部分组成：

- AHB 从接口访问 SDMMC 适配器寄存器，并且生成中断信号和 IDMA 控制信号。
- SDMMC 适配器块提供特定于 *e*•MMC/SD/SD I/O 卡的所有功能，如时钟生成单元、命令和数据传输。
- 内部 DMA（IDMA）模块及其 AHB 主接口。
- 负责接收数据采样时钟对齐的延迟模块 (DLYB)。延迟模块不是 SDMMC 的一部分。支持 SDR104 或 HS200 时，强制使用延迟模块。

24.5.1 SDMMC 框图

图 165 显示了 SDMMC 框图。

图 165. SDMMC 框图



24.5.2 SDMMC 引脚和内部信号

表 176 列出了 SDMMC 内部输入/输出信号，*表 177* 列出了 SDMMC 引脚（复用功能）。

表 176. SDMMC 内部输入/输出信号

信号名称	信号类型	说明
sdmmc_ker_ck	数字输入	SDMMC 内核时钟
sdmmc_hclk	数字输入	AHB 时钟
sdmmc_it	数字输出	SDMMC 全局中断
sdmmc_io_in_ck	数字输入	SD/SDIO/*e*•MMC 卡反馈时钟。此信号在内部连接到 SDMMC_CK 引脚（针对 DS 和 HS 模式）。
sdmmc_fb_ck	数字输入	DLYB 延迟模块调节后的 SD/SDIO/*e*•MMC 卡反馈时钟（用于 SDR50、DDR50、SDR104 和 HS200）

表 177. SDMMC 引脚

信号名称	信号类型	说明
SDMMC_CK	数字输出	SD/SDIO/*e*•MMC 卡的时钟
SDMMC_CKIN	数字输入	来自 SD/SDIO/*e*•MMC 卡的外部驱动器的时钟反馈。（用于 SDR12、SDR25、SDR50 和 DDR50）
SDMMC_CMD	数字输入/输出	SD/SDIO/*e*•MMC 卡双向命令/响应信号。
SDMMC_CDIR	数字输出	SDMMC_CMD 信号的 SD/SDIO/*e*•MMC 卡 I/O 方向指示。
SDMMC_D[7:0]	数字输入/输出	SD/SDIO/*e*•MMC 卡双向数据线。
SDMMC_D0DIR	数字输出	SDMMC_D0 数据线的 SD/SDIO/*e*•MMC 卡 I/O 方向指示。
SDMMC_D123DIR	数字输出	SDMMC_D[3:1] 数据线的 SD/SDIO/*e*•MMC 卡 I/O 方向指示。

24.5.3 一般说明

SDMMC_D[7:0] 线具有不同的工作模式：

- 默认情况下，SDMMC_D0 线用于数据传输。初始化后，主机可以更改数据总线宽度。
- 对于 *e*•MMC，可以使用 1 位 (SDMMC_D0)、4 位 (SDMMC_D[3:0]) 或 8 位 (SDMMC_D[7:0]) 的数据总线宽度。
- 对于 SD 或 SDIO 卡，可以使用 1 位 (SDMMC_D0) 或 4 位 (SDMMC_D[3:0]) 的数据总线宽度。所有数据线均以推挽模式运行。

为了连接外部驱动器（电压切换收发器），使用 I/O 方向信号指示数据线上的数据流方向。**SDMMC_D0DIR** 信号指示 SDMMC_D0 数据线的 I/O 方向，**SDMMC_D123DIR** 则指示 SDMMC_D[3:1] 数据线的方向。

SDMMC_CMD 仅在推挽模式下运行：

为了连接外部驱动器（电压切换收发器），使用 I/O 方向信号 **SDMMC_CDIR** 指示 SDMMC_CMD 线上的数据流方向。

卡的 **SDMMC_CK** 时钟源自 **sdmmc_ker_ck**：

- 当 sdmmc_ker_ck 时钟的占空比为 50% 时，即使在旁路模式 (CLKDIV = 0) 下也可以使用。
- 当 sdmmc_ker_ck 占空比不是 50% 时，必须使用 CLKDIV 以 2 或更大值 (CLKDIV > 0) 对其进行分频。
- SDMMC_CMD/SDMMC_D[7:0] 输出和 SDMMC_CK 之间的相位关系可通过 NEGEDGE 位进行选择。相位关系取决于 CLKDIV、NEGEDGE 和 DDR 设置。请参见*图 166*。

图 166. SDMMC 命令和数据相位关系

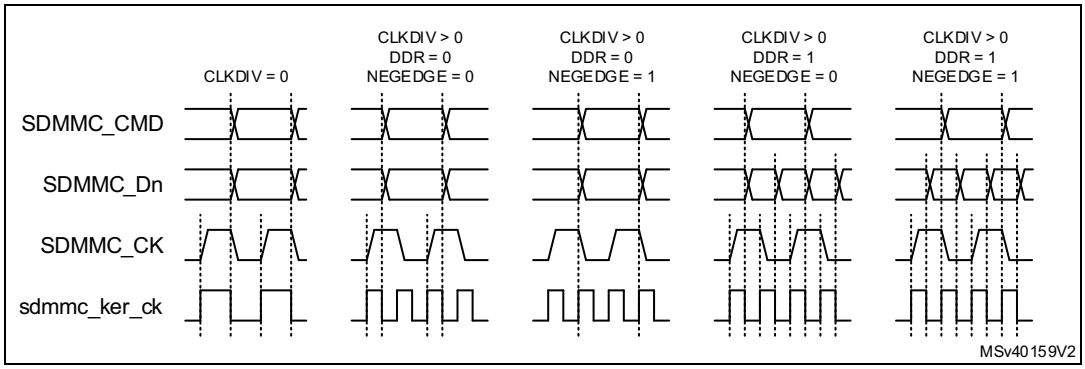


表 178. SDMMC 命令和数据相位选择

CLKDIV	DDR	NEGEDGE	SDMMC_CK	命令输出	数据输出
0	x	x	= sdmmc_ker_ck	在 sdmmc_ker_ck 下降沿生成	
>0	0	0	在 sdmmc_ker_ck 上升沿生成	在 SDMMC_CK 上升沿后的 sdmmc_ker_ck 下降沿生成。	
		1		在生成 SDMMC_CK 下降沿的同一 sdmmc_ker_ck 上升沿生成。	
	1	0		在 SDMMC_CK 上升沿后的 sdmmc_ker_ck 下降沿生成。	在 SDMMC_CK 边沿后的 sdmmc_ker_ck 下降沿生成。
		1		在生成 SDMMC_CK 下降沿的同一 sdmmc_ker_ck 上升沿生成。	

默认情况下，选择 **sdmmc_io_in_ck** 反馈时钟输入来采样 SDMMC 接收路径中的传入数据。它源自 SDMMC_CK 引脚。

为调整采样时钟的相位以适应接收数据时序，可将设备上的 DLYB 延迟模块连接在 **sdmmc_io_in_ck** 信号（DLYB 输入 dlyb_in_ck）和 SDMMC 的 **sdmmc_fb_ck** 时钟输入（DLYB 输出 dlyb_out_ck）之间。在接收路径中选择 **sdmmc_fb_ck** 时钟输入，然后可以使用相位校准的采样时钟来处理传入的数据。这是 SDMMC 支持 SDR104 和 HS200 工作模式所必需的，对于 SDR50 和 DDR50 模式则是可选的。

当使用外部驱动器（电压切换收发器）时，可以选择 SDMMC_CKIN 反馈时钟输入来对接收数据进行采样。

对于 SD/SDIO/*e*•MMC 卡，时钟频率可在 0 到 208 MHz 之间变化（受限于最大 I/O 速度）。

根据选择的总线模式（SDR 或 DDR），每个时钟周期在 SDMMC_D[7:0] 线上传输一位或两位。SDMMC_CMD 线每个时钟周期只传输一位。

24.5.4 SDMMC 适配器

SDMMC 适配器（请参见*图 165：SDMMC 框图*）是一个多媒体卡/安全数字存储卡总线主设备，提供与多媒体卡堆栈或安全数字存储卡的接口。它由以下子单元组成：

- 控制单元
- 数据发送路径
- 命令路径
- 数据接收路径
- 响应路径
- 接收数据路径时钟复用器
- 延迟模块 (DLYB)，在 SDMMC 外部
- 适配器寄存器模块
- 数据 FIFO
- 内部 DMA (IDMA)

注： *适配器寄存器和 FIFO 使用 AHB 时钟域 (sdmmc_hclk)。控制单元、命令路径和数据发送路径使用 SDMMC 适配器时钟域 (sdmmc_ker_ck)。响应路径和数据接收路径使用来自 sdmmc_io_in_ck、SDMMC_CKIN 或 sdmmc_fb_ck（由 DLYB 生成）的 SDMMC 适配器反馈时钟域。*

设备上的 DLYB 延迟模块可与 SDMMC 适配器搭配用来调整 SDMMC 接收模式下传入数据的采样时钟相位。要使 SDMMC 支持 SDR104 和 HS200 工作模式则必需使用 DLYB 模块，而对于 SDR50 和 DDR50 模式，DYLB 模块则是可选的。

适配器寄存器模块

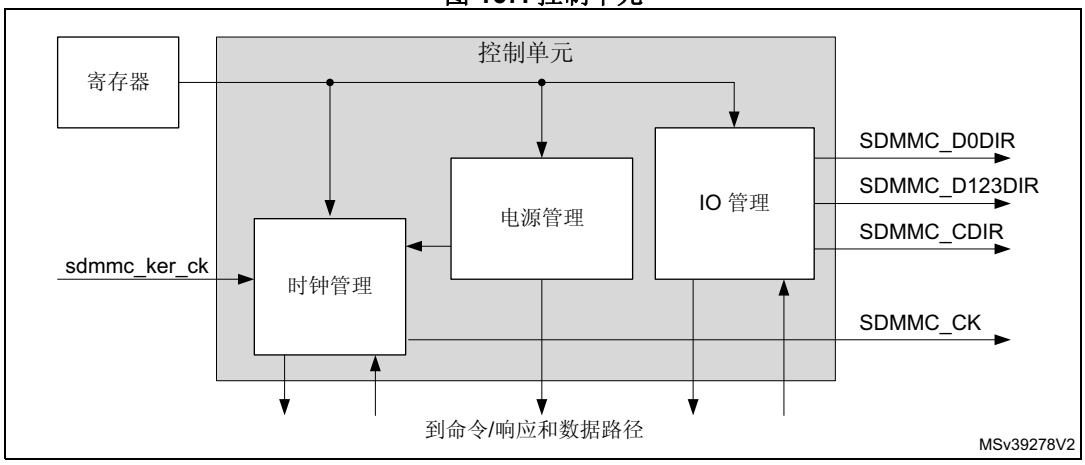
适配器寄存器模块包含所有系统控制寄存器、SDMMC 命令和响应寄存器以及数据 FIFO。

该模块还从 SDMMC 清零寄存器中的相应位单元生成信号，该寄存器用于清零 SDMMC 适配器中的静态标志。

控制单元

控制单元如*图 167* 所示，其包含电源管理功能、带有分频器的 SDMMC_CK 时钟管理功能和 I/O 方向管理功能。

图 167. 控制单元



电源管理子单元会在断电阶段和上电阶段中禁止卡总线输出信号。

有三个电源阶段：

- 掉电
- 上电
- 通电

时钟管理子单元使用 sdmmc_ker_ck 来生成 SDMMC_CK，并提供分频控制。它还负责停止 SDMMC_CK 以进行流控制等。

时钟输出在以下情况下无效：

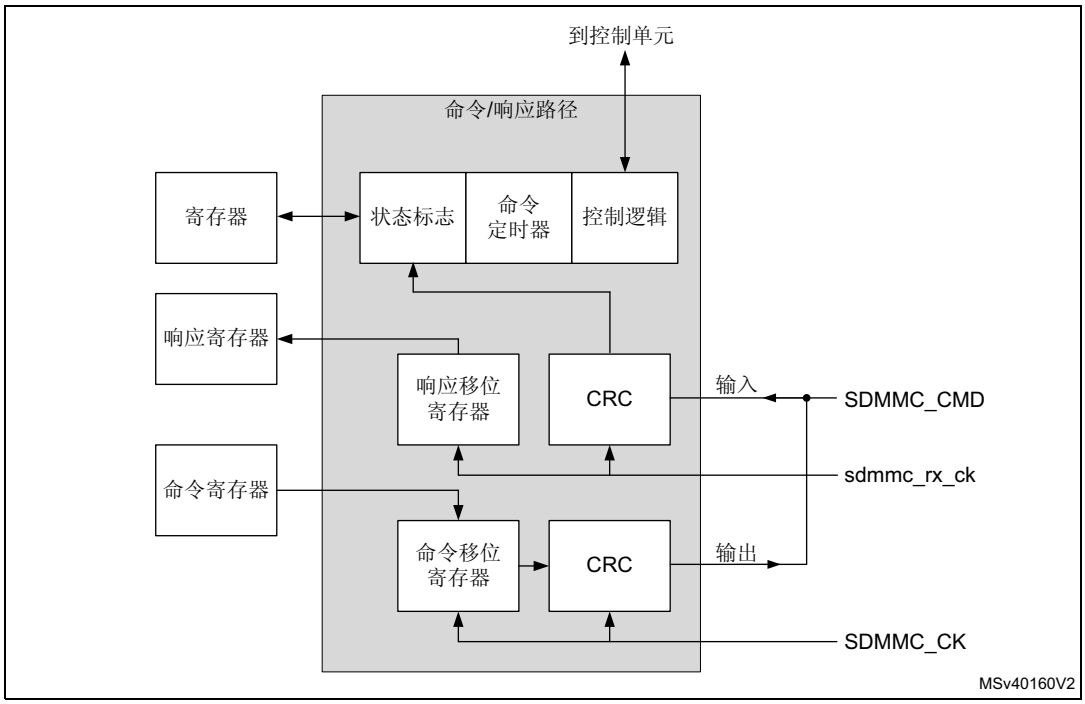
- 复位后
- 在掉电或上电阶段中
- 节能模式（寄存器位 PWRSAV）已使能并且卡总线处于空闲状态的时间持续 8 个时钟周期。在命令/响应 CPSM 和数据路径 DPSM 子单元进入空闲阶段后的八个周期后，时钟停止。当命令/响应 CPSM 或数据路径 DPSM 激活（使能）时，时钟重新启动。

I/O 管理子单元负责处理用于控制外部电压收发器的 SDMMC_Dn 和 SDMMC_CMD I/O 方向信号。

命令/响应路径

命令/响应路径子单元在 SDMMC_CMD 线上传送命令和响应。命令路径由 SDMMC_CK 提供时钟，并向卡发送命令。响应路径由 sdmmc_rx_ck 提供时钟，并从卡接收响应。

图 168. 命令/响应路径

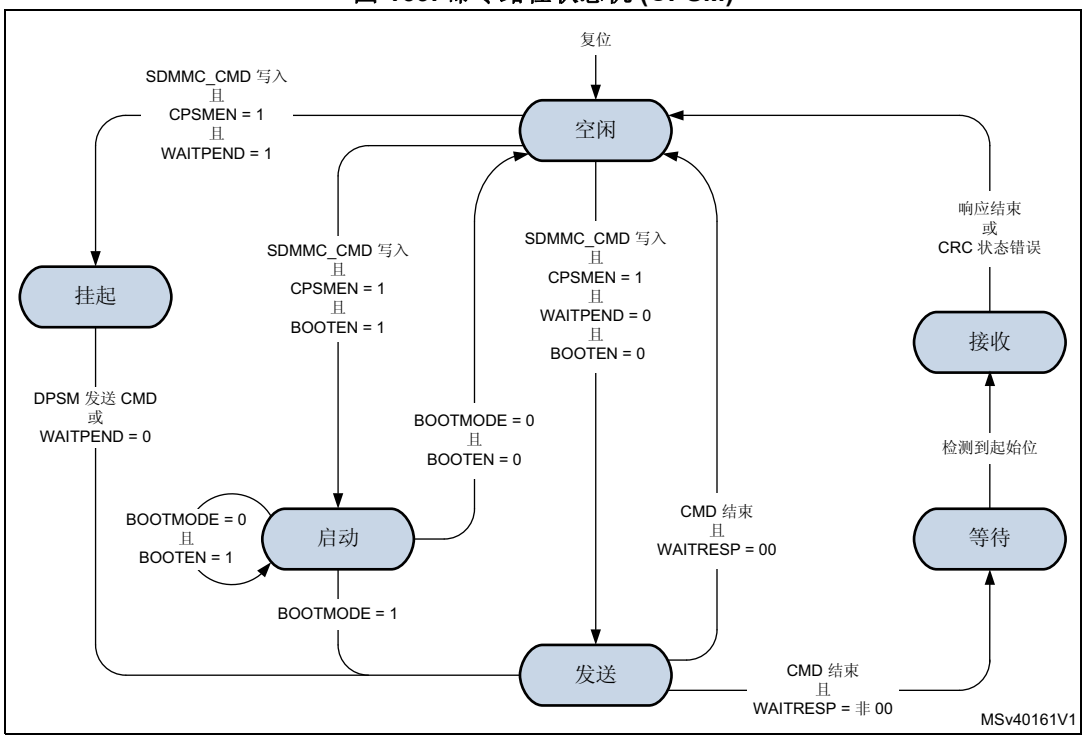


命令/响应路径状态机 (CPSM)

- 写入命令寄存器并且将使能位置 1 后，命令传输将开始。命令发送完成后，将附加 CRC，命令路径状态机 (CPSM) 将设置状态标志，并且：
  - 如果不需要响应，则进入空闲状态。
  - 如果需要响应，则等待响应。
- 接收到响应时，
  - 对于带有 CRC 的响应，将比较接收到的 CRC 代码和内部生成代码，并且会根据结果将相应的状态标志置 1。
  - 对于没有 CRC 的响应，不会进行 CRC 校验，并且不会将相应的状态标志置 1。

当 CPSM 有效（即不处于空闲状态）时，CPSMACT 位置 1。

图 169. 命令路径状态机 (CPSM)



- **空闲**：命令路径无效。当写入命令控制寄存器并且使能位 (CPSMEN) 置 1 后，CPSM 激活 SDMMC_CK 时钟（因节能 PWRSAV 位而停止）并且
  - 在 WAITPEND = 0 且 BOOTEN = 0 时变为发送状态。
  - 在 WAITPEND = 1 时变为挂起状态。
  - 在 BOOTEN = 1 时变为启动状态。
- **发送**：发送命令并附加 CRC。
  - 当 CMDTRANS 位置 1 或 BOOTEN 位置 1、BOOTMODE 为备用启动状态且 DTDIR = 接收时，CPSM 数据使能信号在命令结束时发送给 DPSM。
  - 当 CMDTRANS 位置 1 且 CMDSUSPEND 位为 0 时，中断周期在命令结束时终止。
  - 当 CMDSTOP 位置 1 时，CPSM 中止信号在命令结束时发送给 DPSM。

– 如果不需要响应 (WAITRESP = 00)，CPSM 将变为空闲状态并且 CMDSENT 标志已置 1。当 BOOTMODE = 1 且 BOOTEN = 0 时，CMDSENT 标志将在命令结束位后延迟 56 个周期生成，否则 CMDSENT 标志将在命令结束位后立即生成。
不会修改 RESPCMDR 和 RESPxR 寄存器。

– 如果需要命令响应（WAITRESP 不为 00），CPSM 将变为等待状态并启动响应超时。

- **等待**：命令路径等待响应。
  – 当 WAITINT 位为 0 时，命令定时器开始运行，CPSM 等待起始位。
  a) 如果在超时之前检测到起始位，CPSM 将变为接收状态。
  b) 如果在 CPSM 检测到响应起始位之前便已超时，则超时标志 (CTIMEOUT) 将置 1，CPSM 将变为空闲状态。
  不会修改 RESPCMDR 和 RESPxR 寄存器。
  – 当 WAITINT 位为 1 时，定时器将禁止并且 CPSM 将等待来自其中一个卡的中断请求（响应起始位）。
  a) 如果检测到起始位，CPSM 将变为接收状态。
  b) 向 WAITINT 写入 0（中断模式中止）时，主机自行发送响应，并且检测到起始位时，CPSM 变为接收状态。
- **接收**：已接收到命令响应。响应可短可长，可带 CRC 可不带 CRC，具体取决于命令控制寄存器中的响应模式位 WAITRESP。将根据内部生成的 CRC 代码对接收到的 CRC 代码（存在时）进行校验。
  – 当 CMDSUSPEND 位置 1 且 SDIO 响应位 BS = 0（响应位 [39]）时，中断周期在响应后启动。
  当 CMDSUSPEND 位清零或 CMDSUSPEND 位为 1 且 SDIO 响应位 BS = 1（响应位 [39]）时，不会有中断周期启动。
  – 当 CMDTRANS 位置 1、CMDSUSPEND 位置 1 且 SDIO 响应位 DF = 1（响应位 [32]）时，中断周期在响应后终止。
  – 当 CRC 状态通过校验或不存在 CRC 时，CMDREND 标志将置 1，CPSM 变为空闲状态。
  RESPCMDR 和 RESPxR 寄存器通过接收到的响应进行更新。
  - 当 BOOTMODE = 1 且 BOOTEN = 0 时，CMDREND 标志将在响应结束位后延迟 56 个周期生成，否则 CMDREND 标志将在响应结束位后立即生成。
  - 当 CMDTRANS 位置 1 且 DTDIR = 发送时，CPSM 数据使能信号将在命令响应结束时发送给 DPSM。
  – 当 CRC 状态未通过校验时，CCRCFAIL 标志将置 1，CPSM 变为空闲状态。
  RESPCMDR 和 RESPxR 寄存器通过接收到的响应进行更新。
- **挂起**：CPSM 根据命令寄存器中的挂起 WAITPEND 位进入挂起状态。
  – 当 DATALENGTH =< 5 个字节时，CPSM 变为发送状态，并生成数据使能信号以启动与 CMD12 停止传输命令对齐的数据传输。
  – 当 DATALENGTH > 5 个字节时，将向 DPSM 发出 CPSM 数据使能信号以启动数据传输。CPSM 会等待来自 DPSM 的发送 CMD 信号，然后再变为发送状态。这样一来，便能够发送与数据对齐的 CMD12 停止传输命令。
  – 向 WAITPEND 写入 0 时，CPSM 变为发送状态。

- **启动**：如果命令寄存器中的 BOOTEN 位置 1，则 CPSM 将进入启动状态，并且：
  - 当 BOOTMODE = 0 时，SDMMC_CMD 线驱动为低电平；当 CMDTRANS 位置 1 且 DTDIR = 接收时，CPSM 数据使能信号发送到 DPSM，从而进入正常启动操作。启动过程结束后，将寄存器位 BOOTEN 清零以退出启动状态，此时会将 SDMMC_CMD 线驱动为高电平并将 CPSM 中止信号发送到 DPSM，然后再进入空闲状态。CMDSENT 标志在 SDMMC_CMD 线变为高电平后的 56 个周期后生成。
  - 当 BOOTMODE = 1 时，变为发送状态。此时可以发送 CMD0（启动）。将 BOOTEN 清零不起作用。

注： *CPSM 保持空闲状态至少八个 SDMMC_CK 周期以满足 N_{CC} 和 N_{RC} 时序限制。N_{CC} 是两个主机命令之间的最小延迟，N_{RC} 是主机命令和卡响应之间的最小延迟。*

注： *响应超时为固定值：64 个 SDMMC_CK 时钟周期。*

命令是用于启动操作的令牌。命令从主机发送到单个卡（编址命令），或发送到所有已连接的卡（*e*•MMC V3.31 或更低版本可以使用广播命令）。命令在 SDMMC_CMD 线上以串行方式传输。所有命令都为固定长度 48 位。*表 179.* 中显示了 SD 存储卡、SDIO 卡和 *e*•MMC 卡的命令令牌的常规格式。

命令令牌数据取自 2 个寄存器，一个包含 32 位参数，另一个包含 6 位命令索引（发送到卡的 6 位）。

表 179. 命令令牌格式

位的位置	宽度	值	说明
47	1	0	起始位
46	1	1	传输位
[45:40]	6	x	命令索引
[39:8]	32	x	参数
[7:1]	7	x	CRC7
0	1	1	结束位

命令数据旁为命令类型 (WAITRESP) 位，用于控制命令路径状态机 (CPSM)。这些位还用于确定命令是否需要响应，响应是短（48 位）还是长（136 位）以及是否存在 CRC。

响应是一个令牌，它作为对先前接收命令的应答，从编址卡或从所有已连接的卡同步发送到主机。所有响应都是通过命令线 SDMMC_CMD 来发送。响应传输始终从对应于响应代码字的位字符串的左侧位开始。代码长度取决于响应类型。响应令牌 R1、R2、R3、R4、R5 和 R6 具有多种编码方案，具体取决于其内容。*表 180*、*表 181* 和*表 182* 分别给出了 SD 存储卡、SDIO 卡和 *e*•MMC 卡的响应令牌的常规格式。

响应始终从起始位（始终为 0）开始，后跟用于指示传输方向的位 (card = 0)。下表中标有 x 的值表示变量项。大多数响应（部分除外）都受到 CRC 的保护。每个命令代码字由结束位（始终为 1）来终止。

响应令牌数据存储在 5 个寄存器中，其中 4 个寄存器包含 32 位卡状态、OCR 寄存器、参数或包括内部 CRC 的 127 位 CID 或 CSD 寄存器，另一个寄存器包含 6 位命令索引。

表 180. 带 CRC 的短响应令牌格式

位的位置	宽度	值	说明
47	1	0	起始位
46	1	0	传输位
[45:40]	6	x	命令索引（或保留为 111111）
[39:8]	32	x	参数
[7:1]	7	x	CRC7
0	1	1	结束位

表 181. 不带 CRC 的短响应令牌格式

位的位置	宽度	值	说明
47	1	0	起始位
46	1	0	传输位
[45:40]	6	x	命令索引（或保留为 111111）
[39:8]	32	x	参数
[7:1]	7	1111111	（保留为 1111111）
0	1	1	结束位

表 182. 带 CRC 的长响应令牌格式

位的位置	宽度	值	说明
135	1	0	起始位
134	1	0	传输位
[133:128]	6	111111	保留
[127:1]	127:8	x	CID 或 CSD 片段
	7:1	x	CRC7（包含在 CID 或 CSD 中）
0	1	1	结束位

命令/响应路径以半双工模式运行，因此既可发送命令，也可接收响应。如果 CPSM 未处于发送状态，则 SDMMC_CMD 输出处于高阻抗状态。在 SDMMC_CMD 上发送的数据将根据 NEGEDGE 寄存器位与 SDMMC_CK 进行同步，请参见*图 166*。

对于命令和带 CRC 的短响应，CRC 生成器计算 CRC 代码前面全部 40 个位的 CRC 校验和。这包括起始位、传输位、命令索引和命令参数（或卡状态）。

对于长响应，仅计算 R2 CID 或 CSD 的 120 个位的 CRC 校验和。请注意，CRC 计算中不使用起始位、传输位和六个保留位。

CRC 校验和是一个 7 位值：

CRC[6:0] = 余数 [(M(x) * x^7) / G(x)]

$G(x) = x^7 + x^3 + 1$

M(x) = (第 1 个位) * x^n + (第 2 个位) * x^{n-1}+...+（CRC 前的最后一位）* x^0

其中，n = 39 或 119。

CPSM 可在 CPSMEN 置 1 时发送一些特定命令来处理各种工作模式，请参见*表 183*。

表 183. 特定命令概述

VSWITCH	BOOTEN	BOOTMOD	CMDTRAN	WAITPEND	CMDSTOP	WAITINT	说明
1	x	x	x	x	x	x	启动电压切换序列
0	1	x	x	x	x	x	开始正常启动
0	1	1	x	x	x	x	开始备用启动
0	0	1	x	x	x	x	停止备用启动
0	0	0	1	x	x	x	发送包含相关数据传输的命令。
0	0	0	0	1	1	x	*e*•MMC 流数据传输，命令 (STOP_TRANSMISSION) 挂起至数据传输结束。
0	0	0	0	1	0	x	*e*•MMC 流数据传输，(STOP_TRANSMISSION) 以外的命令挂起至数据传输结束。
0	0	0	0	0	1	x	发送命令 (STOP_TRANSMISSION)，停止任何正在进行的数据传输。
0	0	0	0	0	0	1	进入 *e*•MMC 等待中断 (Wait-IRQ) 模式。
0	0	0	0	0	0	0	任何其他非特定命令

命令/响应路径实现了*表 184* 所示的状态标志和相关清零位：

表 184. 命令路径状态标志

标志	说明
CMDSENT	在命令（无响应）结束时置 1。（CPSM 从发送状态变为空闲状态）
CMDREND	当 CRC 正常时，在命令响应结束时置 1。（CPSM 从接收状态变为空闲状态）
CCRCFAIL	当 CRC 失败时，在命令响应结束时置 1。（CPSM 从接收状态变为空闲状态）
CTIMEOUT	如果未在超时前接收到响应起始位，则在命令后置 1。（CPSM 从等待状态变为空闲状态）
CKSTOP	当 CRC 正常并且 SDMMC_CK 停止时，在电压切换 (VSWITCHEN = 1) 命令响应后置 1。（对 CPSM 无影响）
VSWEND	在 5 ms + 1 ms 的电压切换 (VSWITCH = 1) 超时后置 1。（对 CPSM 无影响）
CPSMACT	正在进行命令传输。（CPSM 不处于空闲状态）

命令路径错误处理如*表 185* 所示：

表 185. 命令路径错误处理

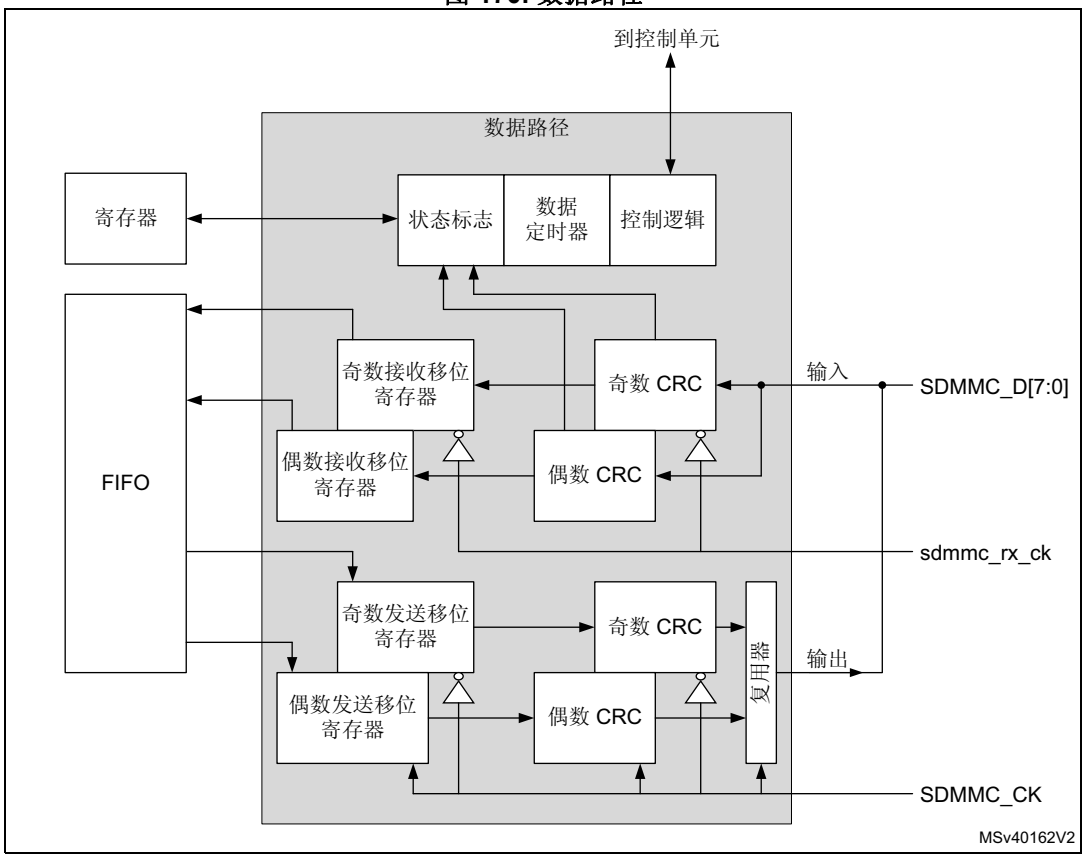
错误	CPSM 状态	原因	卡操作	主机操作	CPSM 操作
超时	等待	未及时出现起始位	未知	对卡进行复位或掉电再上电(1)	变为空闲状态
CRC 状态	接收	否定状态	忽略命令	重新发送命令(1)	变为空闲状态
		传输错误	接受命令	重新发送命令(1)	

1. 当 CMDTRANS 置 1 时，还必须发送 stop_transmission 命令才能使 DPSM 变为空闲状态。

数据路径

数据路径子单元负责通过 SDMMC_D[7:0] 线与卡相互传输数据。数据发送路径由 SDMMC_CK 提供时钟，并向卡发送数据。数据接收路径由 sdmmc_rx_ck 提供时钟，并从卡接收数据。*图 170* 显示了数据路径框图。

图 170. 数据路径



可以使用时钟控制寄存器位 WIDBUS 对卡数据总线宽度进行编程。支持的数据总线宽度模式如下：

- 如果未使能宽总线模式，则仅通过 SDMMC_D0 传输一位。
- 如果使能了 4 位宽度的总线模式，则通过 SDMMC_D[3:0] 传输 4 个数据位。
- 如果使能了 8 位宽度的总线模式，则通过 SDMMC_D[7:0] 传输 8 个数据位。

与数据总线宽度相似，数据采样模式可以在时钟控制寄存器位 DDR 中进行编程。支持的数据采样模式如下：

- 单倍数据速率信号传输 (SDR)，在时钟的上升沿驱动数据。
- 双倍数据速率信号传输 (DDR)，在时钟的两个边沿驱动数据。只有宽总线模式（4 位宽和 8 位宽）才支持 DDR 模式。

注： *数据采样模式仅适用于 SDMMC_D[7:0] 线。（不适用于 SDMMC_CMD 线。）*

在 DDR 模式下，根据以下规则在 SDMMC_CK 的两个边沿对数据进行采样，另请参见*图 171* 和*图 172*：

- 在时钟的上升沿，对奇数字节进行采样。
- 在时钟的下降沿，对偶数字节进行采样。
- 数据有效负载大小始终为 2 字节的倍数。
- 每个数据线计算两个 CRC16
  - 奇数位 CRC16，在时钟的下降沿驱动。
  - 偶数位 CRC16，在时钟的上升沿驱动。
- 起始位、结束位和空闲条件均为完整周期。
- CRC 状态/启动确认和繁忙信号均为完整周期，仅在时钟的上升沿进行采样。

在 DDR 模式下，SDMMC_CK 时钟分频比必须 >= 2。

图 171. DDR 模式数据包时钟驱动

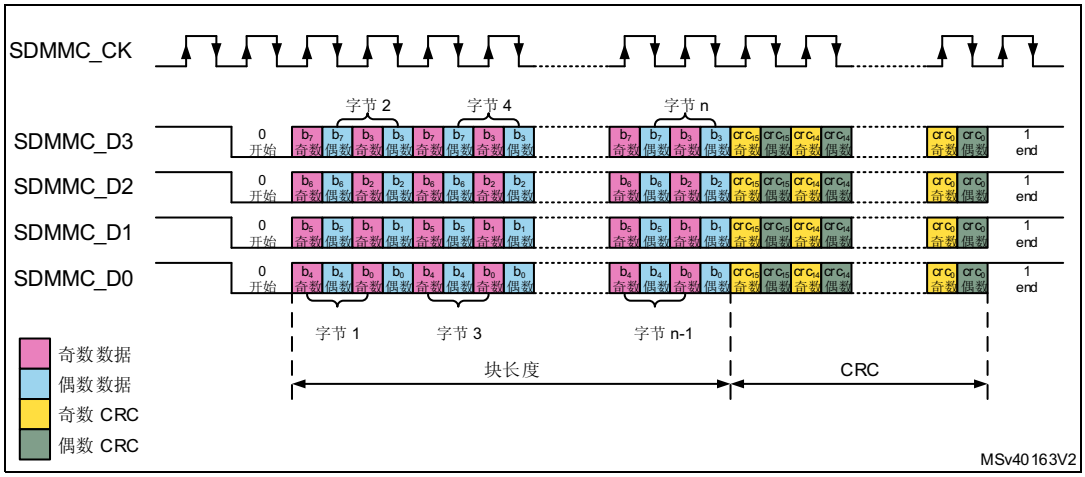
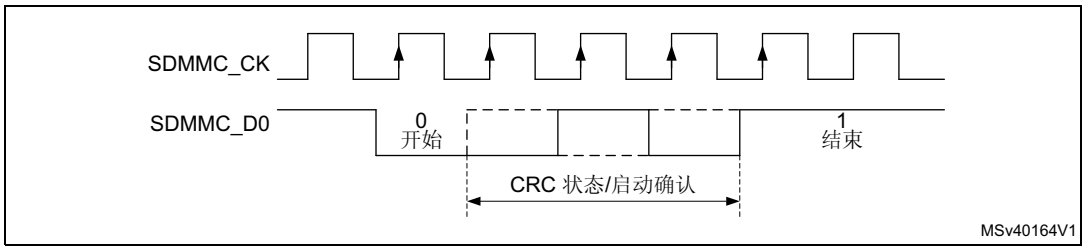


图 172. DDR 模式 CRC 状态/启动确认时钟驱动



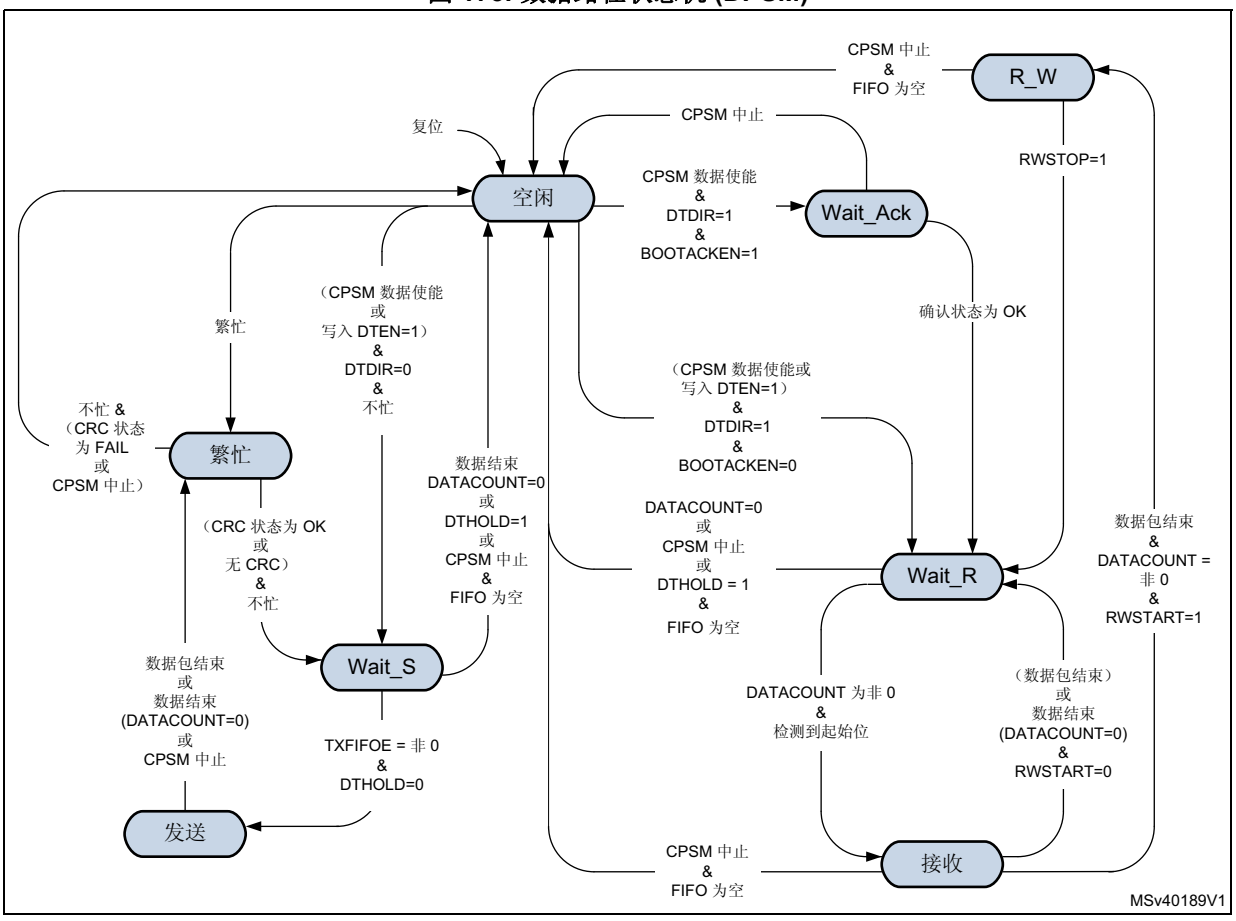
数据路径状态机 (DPSM)

根据传输方向（发送或接收），数据路径状态机 (DPSM) 将变为 Wait_S 或 Wait_R 状态（如果已使能）：

- 发送：DPSM 变为 Wait_S 状态。如果发送 FIFO 中有数据，则 DPSM 变为发送状态并且数据路径子单元开始向卡发送数据。
- 接收：DPSM 变为 Wait_R 状态并等待起始位。在 DPSM 收到起始位时，将变为接收状态，并且数据路径子单元开始从卡中接收数据。
  对于包含确认的启动操作，DPSM 变为 Wait_Ack 状态并等待启动确认，然后再变为 Wait_R 状态。

DPSM 以 SDMMC_CK 工作。DPSM 具有以下状态，如*图 173* 所示。当 DPSM 有效（即不处于空闲状态）时，DPSMACT 位置 1。

图 173. 数据路径状态机 (DPSM)



- **空闲**状态：数据路径无效，SDMMC_D[7:0] 输出取决于 PWRCTRL 设置。DPSM 通过以下方式激活：发送将 CMDTRANS 位置 1 的命令或将 DTEN 位置 1，或者检测 SDMMC_D0 上的繁忙状态（即，在包含 R1b 响应的命令之后）。

  当不忙时，DPSM 激活 SDMMC_CK 时钟（因节能 PWRSAV 位而停止时），并使用新的 (DATALENGTH) 值加载数据计数器，并且：

  – 当数据方向位 (DTDIR) 指示发送时，变为 Wait_S。
  – 当数据方向位 (DTDIR) 指示接收时，
    - 如果 BOOTACKEN 寄存器位清零，则变为 Wait_R。
    - 如果 BOOTACKEN 寄存器位置 1 且启动确认超时，则变为 Wait_Ack。

  当繁忙时，DPSM 使 SDMMC_CK 时钟保持有效并变为繁忙状态。

注： *DTEN 不得用于启动与 SD、SDIO 和 e MMC 卡之间的数据传输。*

- **Wait_Ack** 状态：数据路径等待启动确认令牌。
  - 如果 DPSM 在超时之前收到无错确认，则它将变为 Wait_R 状态。
  - 当接收到确认以外的模式时，将生成确认状态错误，并且确认失败状态标志 (ACKFAIL) 将置 1。DPSM 保持 Wait_Ack 状态。
  - 如果在检测到起始位前便已超时 (ACKTIME)，则它会将超时状态标志 (ACKTIMEOUT) 置 1。DPSM 保持 Wait_Ack 状态。
  - 当 CPSM 中止信号置 1 时，它将变为空闲状态并将 DABORT 标志置 1。
- **Wait_R** 状态：如果数据计数器不为零并且数据未保存，则数据路径将等待 SDMMC_D[n:0] 上的起始位。如果数据计数器为零或数据已保存，则数据路径将等待 FIFO 变为空。
  - 在块模式下，如果在超时之前接收到起始位，DPSM 将变为接收状态，并使用 DBLOCKSIZE 加载数据块计数器。
  - 在 SDIO 多字节模式下，如果在超时之前接收到起始位，DPSM 将变为接收状态，并使用 DATALENGTH 加载数据块计数器。
  - 在流模式下，如果在超时之前接收到起始位，DPSM 将变为接收状态，并使用 DATALENGTH 加载数据计数器。
  - 如果数据计数器 (DATACOUNT) 等于零（数据结束），当接收 FIFO 为空并且 DATAEND 标志置 1 时，DPSM 将变为空闲状态。
  - 如果在检测到起始位前便已超时 (DATATIME)，则它会将超时状态标志 (DTIMEOUT) 置 1 且 DPSM 会保持 Wait_R 状态。
  - 如果 CPSM 中止信号置 1：
    如果 DATACOUNT > 0，则当 FIFO 为空时，DPSM 变为空闲状态，当 IDMAEN = 0 时，使用 FIFORST 复位，并将 DABORT 标志置 1。
    如果 DATACOUNT 为零，则继续正常操作，由于传输已经正常完成，将不会有 DABORT 标志。
  - 如果 DTHOLD 位置 1：
    - 如果 DATACOUNT > 0，则当接收 FIFO 为空时，DPSM 变为空闲状态，当 IDMAEN = 0 时，使用 FIFORST 复位，并发出 DHOLD 标志。保持期间禁止超时。如果在保持期间接收到 CPSM 中止信号，则传输中止。
    - 当 DATACOUNT = 0 时，传输正常完成，不会有 DHOLD 标志。
  - 如果 DPSM 通过 DTEN 启动，在发生错误 (DTIMEOUT) 之后，当 FIFO 为空时，DPSM 将变为空闲状态，当 IDMAEN = 0 时，将使用 FIFORST 复位。
- **R_W** 状态：数据路径对总线进行读取等待。
  - 当读取等待停止位 (RWSTOP) 置 1 时，DPSM 将变为 Wait_R 状态，并且启动接收超时。
  - 如果 CPSM 中止信号置 1，则等待 FIFO 变为空，并且当 IDMAEN = 0 时，使用 FIFORST 复位，然后变为空闲状态并将 DABORT 标志置 1。

- **接收**状态：数据路径从卡接收串行数据。将数据以字节为单位打包，并写入到数据 FIFO。根据数据控制寄存器中选择的传输模式 (DTMODE)，数据传输模式可以是块或流：
  - 在块模式中，当接收到的数据字节数达到数据块大小 (DBLOCKSIZE) 时，DPSM 将等待直至其收到 CRC 代码。
  - 在 SDIO 多字节模式中，当接收到的数据字节数达到数据块大小 (DATALENGTH) 时，DPSM 将等待直至其收到 CRC 代码。

  a) 如果收到的 CRC 代码与内部生成的 CRC 代码相匹配，则
  - 当 RWSTART = 1 且 DATACOUNT > 0 时，DPSM 变为 R_W 状态，DBCKEND 标志置 1。
  - 否则，DPSM 变为 Wait_R 状态。

  b) 如果收到的 CRC 代码与内部生成的 CRC 代码不匹配，则将阻止任何后续的数据接收操作。
  - 当并未接收到所有数据 (DATACOUNT > 0) 时，CRC 失败状态标志 (DCRCFAIL) 将置 1，DPSM 将保持接收状态。
  - 当接收到所有数据 (DATACOUNT = 0) 时，将等待 FIFO 变为空，之后 CRC 失败状态标志 (DCRCFAIL) 将置 1，DPSM 将变为空闲状态。

  - 在流模式中，DPSM 在数据计数器 DATACOUNT > 0 时接收数据。当计数器为零时，会将移位寄存器中其余数据写入到数据 FIFO，并且 DPSM 将变为 Wait_R 状态。
  - 如果发生 FIFO 溢出错误，则 DPSM 将 FIFO 溢出错误标志 (RXOVERR) 置 1 并且会阻止任何后续的数据接收操作。DPSM 保持接收状态。
  - 收到 CPSM 中止信号时：
    - 如果 DATACOUNT = 0 时在数据的最后 2 位之前接收到 CPSM 中止信号，则传输中止。移位寄存器中的剩余数据将写入数据 FIFO，等待 FIFO 变为空，当 IDMAEN = 0 时，使用 FIFORST 复位，DPSM 随后变为空闲状态且 DABORT 标志置 1。
    - 如果 DATACOUNT = 0 时在传输的最后 2 位期间或之后接收到 CPSM 中止信号，则传输正常完成。DPSM 保持接收状态，不生成 DABORT 标志。
  - 如果 DPSM 通过 DTEN 启动，在发生错误（DATACOUNT > 0 时为 DCRCFAIL，否则为 RXOVERR）之后，当 FIFO 为空时，DPSM 将变为空闲状态，当 IDMAEN = 0 时，将使用 FIFORST 复位。

- **Wait_S** 状态：数据路径等待 FIFO 中有数据可用。
  - 如果数据计数器 DATACOUNT > 0，则等待直到数据 FIFO 为空标志 (TXFIFOE) 置为无效并且 DTHOLD 未置 1，然后变为发送状态。
  - 如果数据计数器 (DATACOUNT) = 0，DPSM 将变为空闲状态。
    - 当禁止 DTHOLD 时，DATAEND 标志置 1。
    - 当使能 DTHOLD 时，DHOLD 标志置 1。
  - 当 DTHOLD 置 1 并且 DATACOUNT > 0 时
    - 当使能 IDMA 时，DBCKEND 标志置 1，FIFO 随后清空，并且 DPSM 变为空闲状态，DHOLD 标志置 1。
    - 当禁止 IDMA 时，DBCKEND 标志置 1。等待 FIFO 由软件使用 FIFORST 复位，之后 DPSM 变为空闲状态并发出 DHOLD 标志。
  - 当 DTHOLD 置 1 且 DATACOUNT = 0 时，传输正常完成。

– 当收到 CPSM 中止信号时
- 如果 DATACOUNT = 0 时在数据的最后 2 位之前接收到 CPSM 中止信号，则传输将中止，等待 FIFO 变为空并且当 IDMAEN = 0 时，使用 FIFORST 复位，之后 DPSM 变为空闲状态并将 DABORT 标志置 1。
- 如果 DATACOUNT = 0 时在传输的最后 2 位期间或之后接收到 CPSM 中止信号，则继续正常操作，并且由于传输正常完成，将没有 DABORT 标志。

注： *DPSM 在 Wait_S 状态下需要至少保持两个时钟周期以满足 N_{WR} 计时要求，N_{WR} 是从收到卡响应到开始从主机传输数据这一期间的时钟周期数。*

- **发送**状态：DPSM 开始向卡发送数据。根据数据控制寄存器中的传输模式，数据传输模式可以是块、SDIO 多字节或流：
  – 在块模式中，当发送的数据字节数达到数据块大小 (DBLOCKSIZE) 时，DPSM 将发送一个内部生成的 CRC 代码和结束位，然后变为繁忙状态并启动发送超时。
  – 在 SDIO 多字节模式下，当发送的数据字节数达到数据块大小 (DATALENGTH) 时，DPSM 将发送一个内部生成的 CRC 代码和结束位，然后变为繁忙状态并启动发送超时。
  – 在流模式中，DPSM 在数据计数器 DATACOUNT > 0 时向卡发送数据。当数据计数器达到 0 时，DPSM 变为繁忙状态并启动发送超时。
  在根据 DATACOUNT 发送最后一个流字节之前，DPSM 会触发发送 CMD 信号。CPSM 使用该信号来发送任何挂起的命令。（即 CMD12 停止传输命令）
  – 如果发生 FIFO 下溢错误，DPSM 会将 FIFO 下溢错误标志 (TXUNDERR) 置 1。DPSM 保持发送状态。
  – 当收到 CPSM 中止信号时
  - 如果 DATACOUNT = 0 时在传输的最后 2 位之前接收到 CPSM 中止信号，则传输中止。DPSM 发送最后一个数据位，后跟一个结束位。FIFO 被禁止/清空，并且 DPSM 变为繁忙状态，等待至不忙状态后将 DABORT 标志置 1。
  - 如果 DATACOUNT = 0 时在传输的最后 2 位期间或之后接收到 CPSM 中止信号，则传输正常完成，将不会有 DABORT 标志。
- **繁忙**状态：DPSM 等待 CRC 状态令牌（需要时），并等待不忙信号：
  – 如果需要 CRC 状态令牌并且该令牌指示“无错误传输”或者不需要 CRC 时：
  - 如果 SDMMC_D0 未处于低电平（卡未处于繁忙状态），则它变为 Wait_S 状态。
  - 当卡处于繁忙状态时，SDMMC_D0 处于低电平，它保持繁忙状态。
  – 如果需要 CRC 状态令牌并且该令牌指示“错误传输”。
  - 当并未发送所有数据 (DATACOUNT > 0) 时。DPSM 等待至不忙状态后将 CRC 失败状态标志 (DCRCFAIL) 置 1。FIFO 被禁止/清空并且 DPSM 保持繁忙状态。
  - 当已发送所有数据 (DATACOUNT = 0) 时。DPSM 等待至不忙状态后将 CRC 失败状态标志 (DCRCFAIL) 置 1，并且 DPSM 将变为空闲状态。
  – 如果 DPSM 处于繁忙状态时发生 CRC 状态 (Ncrc) 超时，则它会将数据超时标志 (DTIMEOUT) 置 1 并保持繁忙状态。
  – 如果 DPSM 处于繁忙状态时发生繁忙超时，则它会将数据超时标志 (DTIMEOUT) 置 1 并保持繁忙状态。
  – 当在繁忙状态下接收到 CPSM 中止信号时：
  - 如果 DATACOUNT > 0 时在 CRC 响应的最后 2 位之前接收到 CPSM 中止信号，则数据传输中止。DPSM 等待至不忙状态且 FIFO 被禁止/清空后变为空闲状态，并将 DABORT 标志置 1。
  - 如果 DATACOUNT = 0 时在 CRC 响应的最后 2 位期间或之后接收到 CPSM 中止信号，或者当不需要 CRC、DATACOUNT = 0 且没有 DTIMEOUT 错误时，DPSM 将保持繁忙状态，并且由于传输可正常完成，将不会生成 DABORT 标志。

- 如果在已出现 DTIMEOUT 错误时接收到 CPSM 中止信号，则 DPSM 将等待至不忙状态且 FIFO 禁止/清空后变为空闲状态，并将 DABORT 标志置 1。

– 当由于在发送状态下中止而进入繁忙状态时，DPSM 将等待至不忙状态后变为空闲状态，并将 DABORT 标志置 1。

– 如果 DPSM 通过 DTEN 启动，在发生错误（DATACOUNT > 0 时为 DCRCFAIL，否则为 DTIMEOUT）之后，当 FIFO 复位时，DPSM 将变为空闲状态。

– 当 DPSM 因 SDMMC_D0 上繁忙而启动时，它将等待至不忙状态后将繁忙结束状态标志 (BUSYD0END) 置 1，并且 DPSM 将变为空闲状态。

当 DPSM 在数据块结束位、数据读取命令结束位或 R1b 响应后 2 个周期后进入 Wait_R 或繁忙状态时，将使能数据定时器 (DATATIME)，并会生成数据超时错误 (DTIMEOUT)：

- 发送数据时，发生超时
  – 当需要 CRC 状态并且在 8 个 SDMMC_CK 周期内未接收到起始位时，DTIMEOUT 标志将置 1。
  – 当繁忙状态的时长超过编程的超时周期时，DTIMEOUT 标志将置 1。
- 接收数据时，发生超时
  – 当仍然有数据要接收、DATACOUNT > 0 并且未在编程的超时周期前接收到起始位时，DTIMEOUT 标志将置 1。
- 在 R1b 响应后，发生超时
  – 当繁忙状态的时长超过编程的超时周期时，DTIMEOUT 标志将置 1。

DATATIME = 0 时，

- 接收期间，起始位必须在数据块结束位或数据读取命令结束位后的 2 个周期后出现。
- 发送期间，繁忙状态在 CRC 令牌结束位或流数据结束位后超时 2 个周期。
- 在 R1b 响应之后，繁忙状态在响应结束位后超时 2 个周期。

可以将数据从卡传输到主机（发送），或从主机传输到卡（接收）。数据通过 SDMMC_Dn 数据线传输，并且存储在 FIFO 中。

表 186. 数据令牌格式

说明	起始位	数据(1)	CRC16	结束位	DTMODE
块数据	0	（DBLOCKSIZE 和 DATALENGTH）	是	1	00
SDIO 多字节	0	(DATALENGTH)	是	1	01
e•MMC 数据流	0	(DATALENGTH)	否	1	10

1. 要传输的数据总量由 DATALENGTH 给出。对于块数据而言，每个块中的数据量由 DBLOCKSIZE 给出。

数据令牌格式通过相应的寄存器位 DTMODE 选择。

数据路径实现了*表 187* 所示的状态标志和相关清零位：

表 187. 数据路径状态标志和清零位

标志		说明
DATAEND	TX	如果 CRC 正常且繁忙状态已结束，同时 DTHOLD = 0 且 DATACOUNT = 0，则在完整数据传输结束时置 1。（DPSM 从 Wait_S 状态变为空闲状态）
	RX	如果 CRC 正常且所有数据读取完毕，则在完整数据传输结束时置 1（DATACOUNT = 0 且 FIFO 为空）。（DPSM 从 Wait_R 状态变为空闲状态）
	启动	
DCRCFAIL	TX	如果 CRC 失败且繁忙状态已结束，则在 CRC 结束时置 1。（当仍然有数据要发送时，DPSM 保持繁忙状态并等待 CPSM 中止）（当所有数据发送完毕或 DPSM 已通过 DTEN 启动时，DPSM 将从繁忙状态变为空闲状态）
	RX	如果 CRC 失败且 FIFO 为空，则在 CRC 结束时置 1。（当仍然有数据要接收时，DPSM 保持接收状态并等待 CPSM 中止）（当所有数据接收完毕或 DPSM 已通过 DTEN 启动时，DPSM 将从接收状态变为空闲状态）
	启动	
ACKFAIL	启动	如果失败，则在启动确认结束时置 1。（DPSM 保持 Wait_Ack 状态并等待 CPSM 中止）
DTIMEOUT	CMD R1b	如果未在超时前接收到繁忙结束信号，则在命令响应后置 1。（DPSM 保持繁忙状态并等待 CPSM 中止）
	TX	当未在 Ncrc 内接收到 CRC 令牌起始位或未在超时前接收到繁忙结束信号时置 1。（DPSM 保持繁忙状态并等待 CPSM 中止）（当 DPSM 已通过 DTEN 启动时，将变为空闲状态） 注意：在繁忙超时之前，如果 CRC 失败，则还会将 DCRCFAIL 标志置 1。
	RX	如果未在超时前接收到起始位，则置 1。（DPSM 保持 Wait_R 状态并等待 CPSM 中止）（当 DPSM 已通过 DTEN 启动时，将变为空闲状态）
	启动	
ACKTIMEOUT	启动	如果未在超时前接收到起始位，则置 1。（DPSM 保持 Wait_Ack 状态并等待 CPSM 中止）
DBCKEND	TX	当 DTHOLD = 1 且 IDMAEN = 0 时：在数据传输未完成的情况下 (DATACOUNT >0)，如果 CRC 正常且繁忙状态已结束，则在数据块传输结束时置 1。（DPSM 从繁忙状态变为 Wait_S 状态）
	RX	当 RWSTART = 1 时：在数据传输未完成的情况下 (DATACOUNT >0)，如果 CRC 正常，则在数据块传输结束时置 1。（DPSM 从接收状态变为 R_W 状态）
	启动	
DHOLD	TX	当 DTHOLD = 1 时：如果 CRC 正常且繁忙状态已结束，则在数据块传输结束时置 1。（DPSM 从 Wait_S 状态变为空闲状态）
	RX	当 DTHOLD = 1 时：在数据传输未完成的情况下 (DATACOUNT >0)，如果 CRC 正常且所有数据读取完毕（FIFO 为空），则在数据块传输结束时置 1。（DPSM 从 Wait_R 状态变为空闲状态）
DABORT	CMD R1b	当 CPSM 已发送 CPSM 中止事件且繁忙状态结束时。（DPSM 从繁忙状态变为空闲状态）
	TX	当 CPSM 在传输的最后 2 位之前已发送 CPSM 中止事件时。（DPSM 从任何状态变为空闲状态）
	RX	
	启动	
BUSYD0END	CMD R1b	如果繁忙状态在超时前结束，则在命令响应后置 1。（DPSM 从繁忙状态变为空闲状态）
DPSMACT		正在进行数据传输。（DPSM 不处于空闲状态）

数据路径错误处理如*表 188* 所示：

表 188. 数据路径错误处理

错误	DPSM 状态	原因	卡操作	主机操作	DPSM 操作
超时	Wait_Ack	未及时确认	未知	对卡进行掉电再上电	保持 Wait_Ack 状态（通过 RCC.SDMMCxRST 寄存器位复位 SDMMC）
	Wait_R	未及时出现起始位	未知	停止数据接收 发送停止传输命令	在 CPSM 中止时，DPSM 转变为空闲状态时
			未知	停止启动过程	
	忙	繁忙状态太久（由于数据传输）	未知	停止数据接收 发送停止传输命令	
		繁忙状态太久（由于 R1b）	未知	发送复位命令	
CRC	接收	传输错误	发送后续数据	停止数据接收 发送停止传输命令	在 CPSM 中止时，DPSM 转变为空闲状态时
CRC 状态	忙	否定状态	忽略后续数据	停止数据发送 发送停止传输命令	在 CPSM 中止时，DPSM 转变为空闲状态时
		传输错误	等待后续数据		
确认状态	Wait_Ack	传输错误	发送启动数据	停止启动过程	在 CPSM 中止时，DPSM 转变为空闲状态时
上溢	接收	FIFO 已满	发送后续数据	停止数据接收 发送停止传输命令	在 CPSM 中止时，DPSM 转变为空闲状态时
下溢	发送	FIFO 为空	接收后续数据	停止数据发送 发送停止传输命令	在 CPSM 中止时，DPSM 转变为空闲状态时

数据 FIFO

数据 FIFO（先进先出）子单元包含发送和接收数据缓冲器。单个 FIFO 用于由 DTDIR 位选择的发送或接收操作。FIFO 包含一个宽度为 32 位且深度为 16 字的数据缓冲器和控制逻辑。由于数据 FIFO 在 AHB 时钟域 (sdmmc_hclk) 中运行，因此来自 SDMMC 时钟域 (SDMMC_CK/sdmmc_rx_ck) 的子单元中的所有信号都将重新同步。

FIFO 可进入以下状态之一：

- 向卡发送数据时，发送 FIFO 引用发送逻辑和数据缓冲器。(DTDIR = 0)
- 从卡接收数据时，接收 FIFO 引用接收逻辑和数据缓冲器。(DTDIR = 1)

当来自 FIFO 的 SDMMC 数据传输正确完成时，将由数据路径子单元驱动 DATAEND 标志来指示。当来自 FIFO 的 SDMMC 数据传输出错（中止）时，将由数据路径子单元驱动其中一个错误标志（DCRCFAIL、DTIMEOUT 和 DABORT）或由 FIFO 控制驱动其中一个 FIFO 错误标志（TXUNDERR 和 RXOVERR）来指示。

可以通过以下方式访问数据 FIFO，请参见*表 189*。

表 189. 数据 FIFO 访问

数据 FIFO 访问	IDMAEN
在固件的控制下通过 AHB 从接口访问	0
在 IDMA 的控制下通过 AHB 主接口访问	1

发送 FIFO：

当 DPSM 已激活 (DPSMACT = 1) 时，数据可以写入发送 FIFO。

当 IDMAEN = 1 时，FIFO 由 IDMA 完全处理。

当 IDMAEN = 0 时，FIFO 由固件通过 AHB 从接口控制。可以通过连续地址来访问发送 FIFO。发送 FIFO 包含一个数据输出寄存器，其中存储了读取指针所指向的数据字。在数据路径子单元已加载了其移位寄存器之后，该子单元会将读取指针递增并将新数据推出。发送 FIFO 的处理方式如下：

1. 将数据长度写入 DATALENGTH，将块长度写入 DBLOCKSIZE。
   – 对于块数据传输 (DTMODE = 0)，DATALENGTH 必须为 DBLOCKSIZE 的倍数。
2. 将 SDMMC 设置为发送模式 (DTDIR = 0)。
   – 将 FIFO 配置为发送模式。
3. 使能数据传输
   – 通过将 CMDTRANS 位置 1 从 CPSM 发送一条命令
   – 通过将 DTEN 位置 1
4. 当 (DPSMACT = 1) 将数据写入 FIFO 时。
   – DPSM 保持 Wait_S 状态，直到 FIFO 已满 (TXFIFOF = 1) 或者达到 DATALENGTH 指示的数字。
   - 只要 FIFO 不为空，SDMMC 就会继续发送数据，数据传输过程中的硬件流控制用于防止 FIFO 下溢。
5. 向 FIFO 写入数据。
   – 当 FIFO 由软件处理时，等待至 FIFO 为半空（TXFIFOHE 标志），然后将数据写入 FIFO，直到 FIFO 已满 (TXFIFOF = 1)，或者写入最后一个数据。
   – 当 FIFO 由 IDMA 处理时，IDMA 传输 FIFO 数据。
6. 当写入最后一个数据后，等待数据结束（DATAEND 标志）
   – SDMMC 已发送完所有数据且 DPSM 处于禁止状态 (DPSMACT = 0)。

如果在 IDMAEN = 0 时发生数据传输错误或传输暂停，固件必须停止写入 FIFO，并使用 FIFORST 寄存器位清空和复位 FIFO。

表 190 中列出了发送 FIFO 状态标志。

表 190. 发送 FIFO 状态标志

标志	说明
TXFIFOF	当所有发送 FIFO 字都包含有效数据时，设置为高电平。
TXFIFOE	当发送 FIFO 不包含有效数据时，设置为高电平。
TXFIFOHE	当一半或更多个发送 FIFO 字为空时，设置为高电平。
TXUNDERR	发生下溢错误时，设置为高电平。通过写入到 SDMMC 清零寄存器可将该位清零。

接收 FIFO：

当 DPSM 已激活 (DPSMACT = 1) 时，可以从接收 FIFO 读取数据。

当 IDMAEN = 1 时，FIFO 由 IDMA 完全处理。

当 IDMAEN = 0 时，FIFO 由固件通过 AHB 从接口控制。当数据路径子单元接收到数据字时，它会驱动数据到写数据总线上。在写入操作完成后，写入指针递增。在读取端，读取指针的当前值指向的 FIFO 字内容被推入读取数据总线中。可以通过连续地址来访问接收 FIFO。

接收 FIFO 的处理方式如下：

1. 将数据长度写入 DATALENGTH，将块长度写入 DBLOCKSIZE。
   – 对于块数据传输 (DTMODE = 0)，DATALENGTH 必须为 DBLOCKSIZE 的倍数。
2. 将 SDMMC 设置为接收模式 (DTDIR = 1)。
   – 将 FIFO 配置为接收模式。
3. 使能 DPSM 传输
   – 通过将 CMDTRANS 位置 1 从 CPSM 发送一条命令
   – 或者通过将 DTEN 位置 1。
4. 当 (DPSMACT = 1) FIFO 准备好接收数据时。
   – DPSM 会将接收到的数据写入 FIFO。
   - 只要 FIFO 未满，SDMMC 就会继续接收数据，数据传输过程中的硬件流控制用于防止 FIFO 上溢。
5. 从 FIFO 读取数据。
   – 当 FIFO 由软件处理时，等待至 FIFO 为半满（RXFIFOHF 标志），然后从 FIFO 读取数据，直到 FIFO 为空 (RXFIFOE = 1)。
   - 当接收到最后一个数据时，从 FIFO 读取数据，直到 FIFO 为空 (DATAEND = 1)。
   – 当 FIFO 由 IDMA 处理时，IDMA 传输 FIFO 数据。
6. SDMMC 已接收完所有数据且 DPSM 处于禁止状态 (DPSMACT = 0)。

如果在 IDMAEN = 0 时暂停数据传输，固件必须读取剩余数据，直到 FIFO 为空，并使用 FIFORST 寄存器位复位 FIFO。这会导致 DPSM 进入空闲状态 (DPSMACT = 0)。

如果在 IDMAEN = 0 时出现数据传输错误，固件必须停止读取 FIFO，并使用 FIFORST 寄存器位清空和复位 FIFO。这会导致 DPSM 进入空闲状态 (DPSMACT = 0)。

表 191 中列出了接收 FIFO 状态标志。

表 191. 接收 FIFO 状态标志

标志	说明
RXFIFOF	当所有接收 FIFO 字都包含有效数据时，设置为高电平
RXFIFOE	当接收 FIFO 不包含有效数据时，设置为高电平。
RXFIFOHF	当一半或更多个接收 FIFO 字包含有效数据时，设置为高电平。
RXOVERR	发生溢出错误时，设置为高电平。通过写入到 SDMMC 清零寄存器可将该位清零。

CLKMUX 单元

CLKMUX 选择用于与接收到的数据和命令响应搭配使用的 sdmmc_rx_ck 时钟源。可以通过时钟控制寄存器位 SELCLKRX 从以下几项中选择接收数据时钟源：

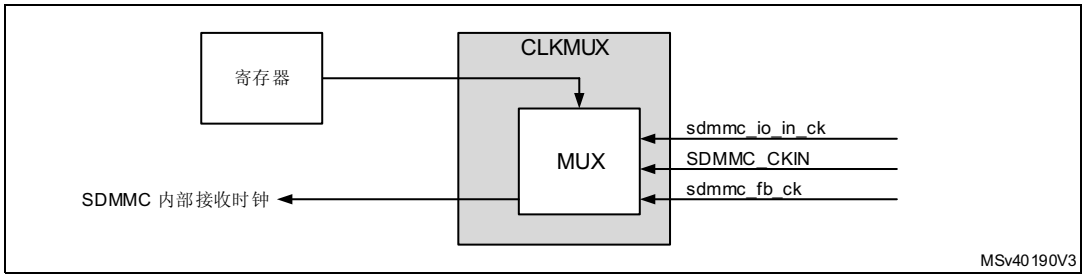
- sdmmc_io_in_ck 总线主设备主反馈时钟。
- SDMMC_CKIN 外部总线反馈时钟。
- sdmmc_fb_ck 总线调节反馈时钟。

当没有外部驱动器且使用 DS 和 HS 时，选择 sdmmc_io_in_ck。

当有外部驱动器且使用 SDR12、SDR25、SDR50 和 DDR50 时，选择 SDMMC_CKIN。

如果设备上的 DLYB 模块与 SDR104、HS200 模式搭配使用并且可选择与 SDR50 和 DDR50 模式搭配使用，则必须选择 sdmmc_fb_ck 时钟输入。

图 174. CLKMUX 单元



当 CPSM 和 DPSM 处于空闲状态时，必须更改 sdmmc_rx_ck 源。

24.5.5 SDMMC AHB 从接口

AHB 从接口生成中断请求，并且访问 SDMMC 适配器寄存器和数据 FIFO。该接口由数据路径、寄存器解码器和中断逻辑构成。

SDMMC FIFO

FIFO 访问仅限于字访问：

- 在发送 FIFO 模式下
  - 数据按字（32 位）写入 FIFO，直到 DATALENGTH 所指示的所有数据均传输完毕。当 DATALENGTH 不是 4 的倍数时，最后剩余的数据（1、2 或 3 字节）通过字传输的方式写入。
- 在接收 FIFO 模式下
  - 数据按字（32 位）从 FIFO 读取，直到 DATALENGTH 所指示的所有数据均传输完毕。当 DATALENGTH 不是 4 的倍数时，最后剩余的数据（1、2 或 3 字节）通过以 0 值字节填充的字传输读取。

当通过半字或字节访问的方式访问 FIFO 时，会生成 AHB 总线故障。

SDMMC 中断

中断逻辑生成一个中断请求信号，当至少其中一个非屏蔽状态标志有效时，便发出此信号。提供了一个屏蔽寄存器，用于选择生成中断的条件。如果对应的屏蔽标志置 1，则状态标志将生成中断请求。一些状态标志需要在清零寄存器中隐式清零。

24.5.6 SDMMC AHB 主接口

AHB 主接口用于在内存和 FIFO 之间使用 SDMMC IDMA 传输数据。

SDMMC IDMA

直接内存访问 (DMA) 用于在 SDMMC FIFO 与内存之间提供高速传输。AHB 主设备可优化系统总线的带宽。SDMMC 内部 DMA (IDMA) 提供一个用于发送或接收的通道。

IDMA 通过 IDMAEN 位使能，并支持 8 节拍的突发传输。

- 在发送突发传输模式下：
  - 只要 FIFO 中的突发传输次数为空，就会以突发形式从内存中获取数据，直到 DATALENGTH 所指示的所有数据均已传输完毕。如果 DATALENGTH 不是突发大小的倍数，则不足突发大小的剩余数据将使用单次传输模式进行传输。
    当 DATALENGTH 不是 4 的倍数时，最后剩余的数据（1、2 或 3 字节）通过字传输的方式获取。
- 在接收突发传输模式下：
  - 只要 FIFO 中包含突发传输次数，就会以突发形式将数据存储到内存中，直到 DATALENGTH 所指示的所有数据均已传输完毕。
    如果 DATALENGTH 不是突发传输的倍数，则不足突发大小的剩余数据将使用单次传输模式进行传输。
    当 DATALENGTH 不是 4 的倍数时，最后剩余的数据（1、2 或 3 字节）通过半字或字节传输的方式存储。

此外，IDMA 还提供以下通道配置（通过位 IDMABMODE 选择）：

- 单缓冲通道
- 链表通道

单缓冲通道

在单缓冲区配置中，内存端的数据从基址 IDMABASE 开始以线性方式进行访问。当 IDMA 完成所有数据的传输并且 DPSM 也完成传输后，DATAEND 标志将置 1。

链表通道

在链表配置中 (IDMAMODE = 1)，内存端的数据从位于基址 IDMABASE 处的链接缓冲区访问。内存缓冲区的大小由 IDMABSIZE 定义。缓冲区大小必须是突发大小的整数倍。位 ULA 用于指示是否必须从链表加载新的链表缓冲区配置。如果当前链表项的 ULA 位置 1，则加载新的链表配置。

第一个链表项配置由固件直接在 SDMMC 寄存器中编程。

当 IDMA 完成一个链表缓冲区中所有数据的传输（基于 IDMABSIZE）后，以及当链表项 ULA 位置 1 时，IDMA 从链表中加载新的链表项，然后继续传输下一个链表缓冲区中的数据。当 IDMA 完成所有数据的传输（基于 IDMABSIZE 和 ULA）并且 DPSM 也完成传输（基于 DATALENGHT）后，DATAEND 标志将置 1。

在以下情况下，链表提供的缓冲区空间大于要传输的数据，这意味着当前链表缓冲区数据尚未完全传输：

- ULA 位置 1，并且所有 SDMMC 数据（基于 DATALENGTH）均已传输完毕（DATAEND 标志）
- 出现传输错误（DATACOUNT > 0 时为 DCRCFAIL，否则为 RXOVERR 或 TXUNDERR）
- 传输暂停 (DTHOLD)

在上述所有情况下，IDMA 链表将停止，FIFO 将清空/复位。在开始或重新开始新的 SDMMC 传输之前，软件必须根据正确的 IDMABASE 和 IDMABSIZE 初始化一个新的链表。

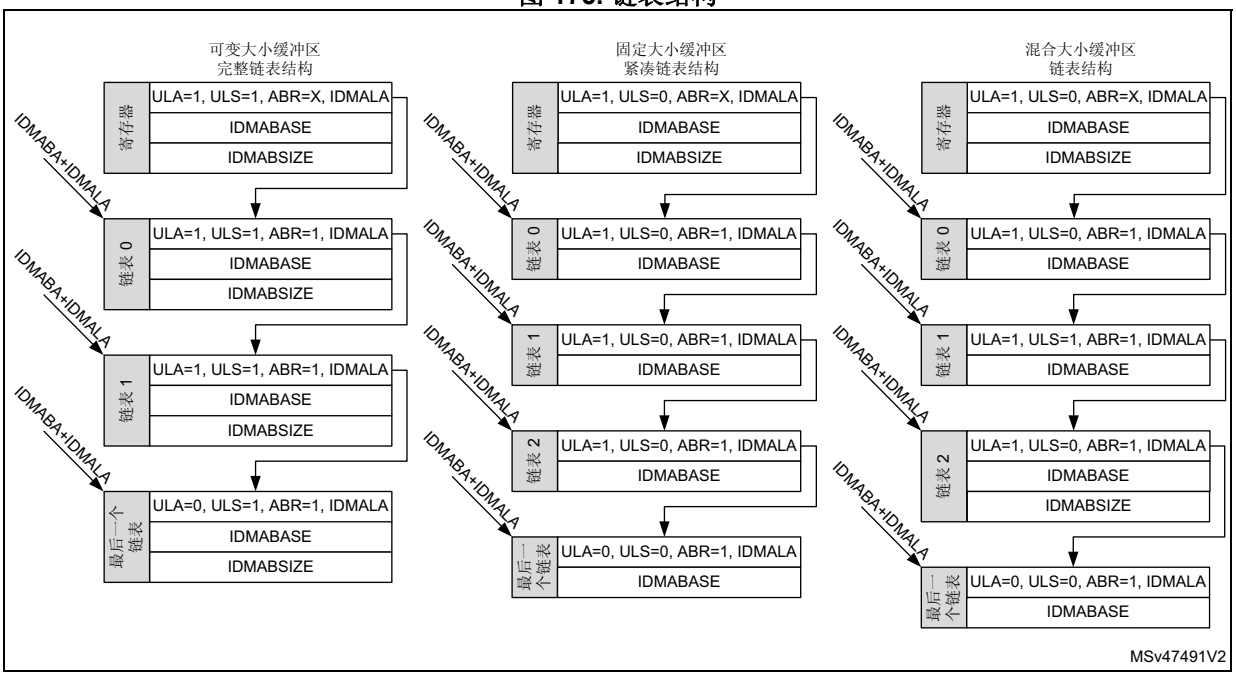
当出现 IDMA 传输错误（请参见 *IDMA 传输错误管理*一节）或链表未提供足够的缓冲区空间时：

- 链表以 ULA = 0 结束，所有最后的链表缓冲区数据均已传输完毕，但并非所有 SDMMC 数据（基于 DATALENGTH）均传输完毕。SDMMC 传输停止并生成 IDMA 传输错误（请参见 *IDMA 传输错误管理*一节）。

对于给定的链表项，基址由链表基址 IDMABA 寄存器值加上链表偏移 IDMALA 寄存器值给出。

每个链表项的内容可以由 ULS 位指定，因此可以选择性地加载 IDMABSIZE，从而产生一个 3 字的链表结构。当不加载 IDMABSIZE（即固定大小的缓冲区）时，可以使用压缩的 2 字链表结构，其中仅包含 IDMABASER 和 IDMALAR 值。

图 175. 链表结构



在单个列表中混合两个链表项结构时没有限制，因此仅在需要时更新 IDMABSIZE。

每当链表缓冲区传输完毕并且当前缓冲区 ULA = 1 时，就可能会生成链表缓冲区传输完成结束中断 (IDMABTC)（如果使能了中断）。

链表确认

在 SDMMC 传输期间软件动态更新链表的情况下，可通过确认缓冲区就绪 (ABR) 位确认新链表缓冲区的可用性。

当 ABR 确认新链表缓冲区已就绪时，IDMA 继续传输新链表缓冲区中的数据。

当 ABR 指示新链表缓冲区未就绪时，会生成 IDMA 传输错误（请参见 *IDMA 传输错误管理* 一节）。当发生 IDMA 传输错误时，这通常会导致生成 TXUNDERR 或 RXOVERR 错误。如果未及时确认链表缓冲区，则停止 SDMMC 传输。

当从软件编程的寄存器信息启动链表时，ABR 信息为“无关”。在开始 SDMMC 传输之前，第一个链表缓冲区必须准备好以供使用。

IDMA 传输错误管理

在以下情况下，可能发生 IDMA 传输错误：

- 当读取或写入保留的地址空间（对于数据或链表信息）时。
- 当没有更多的链表缓冲区空间来存储接收到的 SDMMC 数据时。
- 当所有链表缓冲区数据均传输完毕，并且仍需要发送更多 SDMMC 数据时。
- 当未确认链表缓冲区的可用性时。

在出现 IDMA 传输错误时，后续的 IDMA 传输将禁止，IDMATE 标志将置 1 并且硬件流控制将禁止。当发生 IDMA 传输错误时，这通常会导致生成 TXUNDERR 或 RXOVERR 错误。

IDMATE 标志的行为取决于 SDMMC 传输期间出现 IDMA 传输错误的时间：

- 在出现任何 SDMMC 传输错误（TXUNDERR、RXOVERR、DCRCFAIL 或 DTIMEOUT）之前检测到 IDMA 传输错误：
  - IDMATE 标志与 SDMMC 传输错误标志同时置 1。
  - 生成 TXUNDERR、RXOVERR、DCRCFAIL 或 DTIMEOUT 中断。
- 在 STOP_TRANSNMISSION 命令期间检测到 IDMA 传输错误：
  - IDMATE 标志与 DABORT 标志同时置 1。
  - 生成 DABORT 中断。
- 在 SDMMC 传输（DHOLD 或 DATAEND）结束时检测到 IDMA 传输错误。
  - IDMATE 标志在 SDMMC 传输结束时置 1。
  - 生成 SDMMC 传输结束中断且 DHOLD 或 DATAEND 标志置 1。

IDMATE 在出现其他 SDMMC 传输中断（TXUNDERR、RXOVERR、DCRCFAIL、DTIMEOUT、DABORT、DHOLD 或 DATAEND）时生成。

24.5.7 AHB 和 SDMMC_CK 时钟的关系

AHB 带宽必须至少为 SDMMC 总线带宽的 3 倍，即对于 SDR50 4 位模式 (50 MB/s)，最小 sdmmc_hclk 频率为 37.5 MHz (150 MB/s)。

表 192. AHB 和 SDMMC_CK 时钟频率的关系

SDMMC 总线模式	SDMMC 总线带宽	最大 SDMMC_CK [MHz]	最小 AHB 时钟 [MHz]
e•MMC DS	8	26	19.5
e•MMC HS	8	52	39
e•MMC DDR52	8	52	78
e•MMC HS200	8	200	150
SD DS / SDR12	4	25	9.4
SD HS / SDR25	4	50	18.8
SD DDR50	4	50	37.5
SD SDR50	4	100	37.5
SD SDR104	4	208	78

24.6 卡功能描述

24.6.1 SD I/O 模式

以下功能是专用于 SDMMC 的操作：

- SDIO 中断
- SDIO 挂起/恢复操作（写入和读取挂起）
- 通过停止时钟执行的 SDIO 读取等待操作
- 通过触发 SDMMC_D2 执行的 SDIO 读取等待操作

表 193. SDIO 特殊操作控制

操作模式	SDIOEN	RWMOD	RWSTOP	RWSTART	DTDIR
中断检测	1	X	X	X	X
挂起/恢复操作	X	X	X	X	X
读取等待 SDMMC_CK 时钟停止 (START)	X	1	0	1	1
读取等待 SDMMC_CK 时钟停止 (STOP)	X	1	1	1	1
读取等待 SDMMC_D2 信号 (START)	X	0	0	1	1
读取等待 SDMMC_D2 信号 (STOP)	X	0	1	1	1

SD I/O 中断

为使 SD I/O 卡能够中断主机，在 SD 接口的引脚 8（4 位模式下与 SDMMC_D1 共用）上提供了一个中断功能。每个卡或者卡中的功能都可以使用中断。SD I/O 中断为电平敏感型，这意味着，在主机识别出中断线并处理或者由于中断周期结束而使其失效之前，中断线必须保持有效状态（低电平）。在主机处理完中断后，将通过在 SD I/O 卡内部寄存器中的相应位中进行 I/O 写入来清除中断状态位。所有 SD I/O 卡的中断输出均为低电平有效，并且应用程序必须在所有数据线 (SDMMC_D[3:0]) 上提供外部上拉电阻。

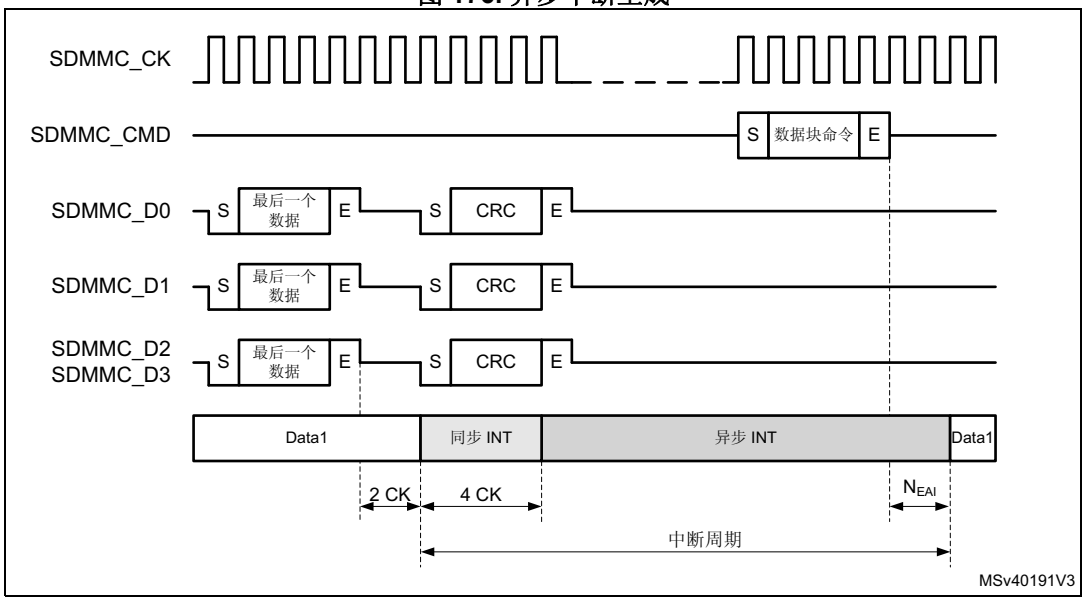
在 SD 1 位模式下，引脚 8 专用于中断功能 (IRQ)，并且中断没有时序限制。

在 SD 4 位模式下，只有在中断周期内，主机才会将引脚 8 (SDMMC_D1/IRQ) 的电平采样到中断检测器中。而在所有其他时候，主机中断将忽略此值。当卡上使能中断且 SDIOEN 位置 1 时，中断周期开始，请参见*表 193* 中的寄存器设置。

在 4 位模式下，卡可以生成同步或异步中断，分别通过卡 CCCR 寄存器中的 SAI 和 EAI 位指示。

- 同步中断，要求 SDMMC_CK 有效。
- 异步中断，可以在 SDMMC_CK 停止时生成，即在最后一个数据块后的卡中断周期开始后的 4 个周期后生成。

图 176. 异步中断生成



中断周期的时序取决于总线速度模式：

在 DS、HS、SDR12 和 SDR25 模式（通过寄存器位 BUSSPEED 选择）下，中断周期与 SD 时钟同步。

- 中断周期在数据块传输命令（在 CMDTRANS 位置 1 时发送命令）的结束位的下一个时钟结束或在 DTEN 位置 1 时结束。
- 中断周期在数据块完成传输后的 2 个 SDMMC_CK 后恢复。
- 在数据块间隙，中断周期被限制为 2 个 SDMMC_CK 周期。

注： *DTEN 不得用于启动与 SD 和 e MMC 卡的数据传输。*

图 177. 同步中断周期数据读取

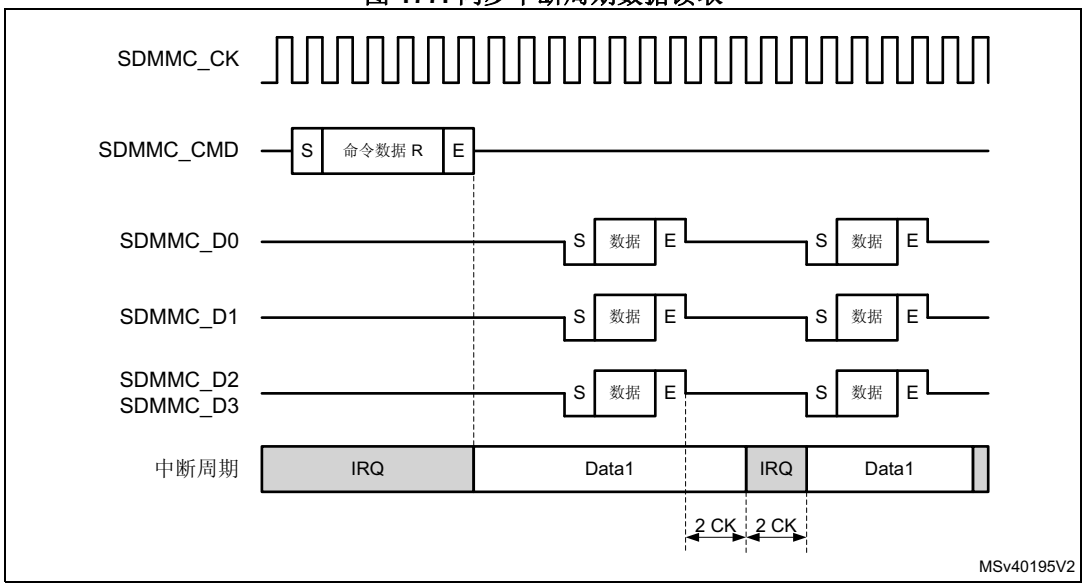
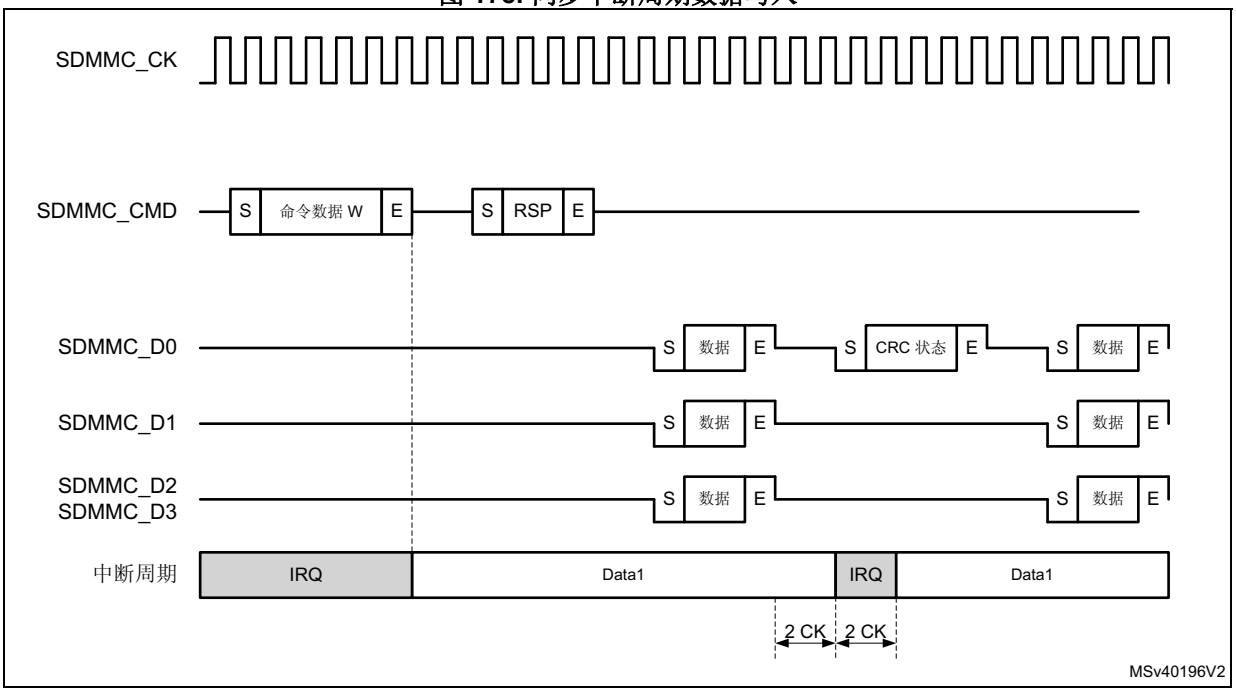


图 178. 同步中断周期数据写入



在 SDR50、SDR104 和 DDR50（通过寄存器位 BUSSPEED 选择）下，由于卡与主机之间存在传播延迟，因此中断周期是异步的。

- 卡中断周期在数据块传输命令（在 CMDTRANS 位置 1 时发送命令）的结束位后的 0 到 2 个 SDMMC_CK 周期后结束或在 DTEN 位置 1 时结束。在主机中，中断周期在数据块传输命令的结束位之后结束。如果在此中断周期内主机未检测到命令结束位，则将在 1 到 2 个周期后发出卡中断。
- 卡中断周期在最后一个数据块完成传输后的 2 到 4 个 SDMMC_CK 后恢复。主机始终在最后一个数据块之后的 2 个周期后恢复中断周期。
- 数据块间隙内没有中断周期。

注： *DTEN 不得用于启动与 SD 和 e MMC 卡的数据传输。*

图 179. 异步中断周期数据读取

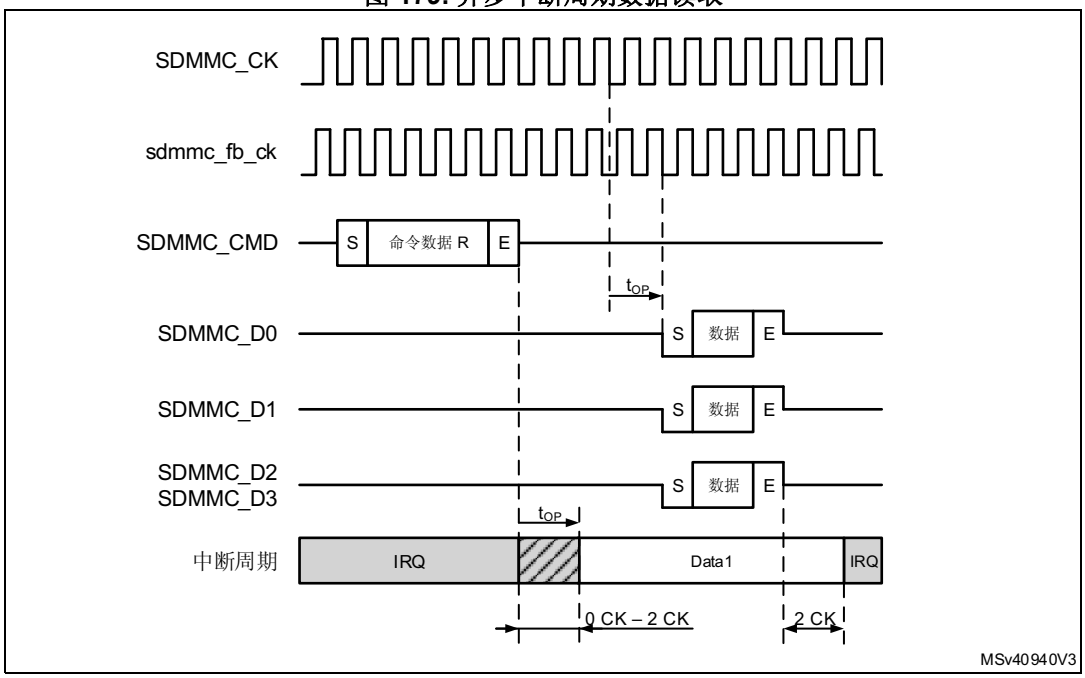
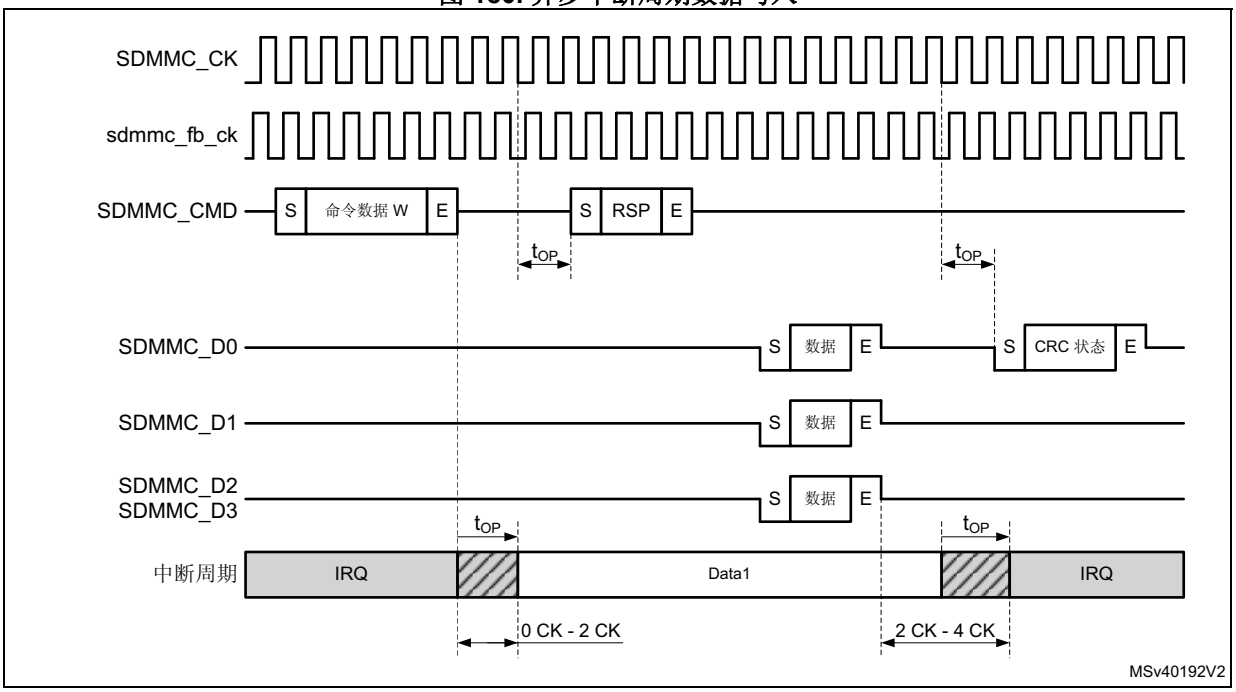


图 180. 异步中断周期数据写入



当传输开放式多块数据并使用 DTMODE“以 STOP_TRANSMISSION 命令结束的块数据传输”时，SDMMC 在最后一个数据块之后屏蔽中断周期，直到 CMD12 STOP_TRANSMISSION 命令结束。

中断周期适用于内存和 I/O 操作。

在 4 位模式下，可以根据*表 194* 来区分中断与其他信号。

表 194. 4 位模式启动、中断和 CRC 状态信号检测

SDMMC 数据线	启动	中断	CRC 状态
SDMMC_D0	0	1 或 CRC 状态	0
SDMMC_D1	0	0	X
SDMMC_D2	0	1 或读取等待	X
SDMMC_D3	0	1	X

SD I/O 挂起和恢复

SDIO V4.00 或更高版本不支持此功能。

在多功能 SD I/O 中或者在同时具备 I/O 和内存功能的卡中，有多个设备（I/O 和内存）共享对 *e*•MMC/SD 总线的访问权限。要在多个设备之间共享对主机的访问权限，SD I/O 和组合卡可以有选择性地实施“挂起/恢复”这一概念。如果卡支持挂起/恢复，则主机可以临时停止（挂起）对某一功能或内存的数据传输操作以释放总线，从而将总线用于其他功能或内存的更高优先级传输。此更高优先级传输完成后，原始传输将从中断位置重新开始（恢复）。

要在总线上执行挂起/恢复操作，主机将执行以下步骤：

1. 确定当前使用 SDMMC_D [3:0] 线的功能
2. 请求将优先级较低或速度较慢的事务挂起
3. 等待事务挂起完成
4. 开始更高优先级的事务
5. 等待更高优先级事务完成
6. 恢复挂起的事务

接收到挂起命令的卡使用其当前总线状态进行响应。只有当总线通过卡挂起后，总线状态才会指示挂起完成。

有多种挂起情况：

- 在数据传输开始之前接受挂起请求。
- 未接受挂起请求（由于同时在传输数据），主机持续检查请求，直到接受请求。（数据传输已停止）
- 写繁忙时的挂起请求。
- 多次写入时的挂起请求。
- 读取等待时的挂起请求。

为了让主机获知总线是否已释放，必须检查挂起请求的状态（挂起完成）。

当挂起请求响应的总线状态指示挂起完成时，表示卡已释放总线。此时，必须保存挂起操作的状态，之后可启动其他操作。

挂起命令必须在 CMDSUSPEND 位置 1 时发送。这样，当总线挂起时（响应位 BS = 0），可在挂起命令响应之后启动中断周期。

硬件不会保存挂起的操作恢复后还需传输的剩余数据量。固件负责确定已传输的数据，从而保证恢复后的剩余字节数是正确的。

从卡中接收数据时，SDMMC 可以在读取数据块结束（DPSM 处于 Wait_R 状态）之后挂起读取操作。在收到卡的挂起确认响应后，固件必须执行以下步骤：

1. 必须通过将 DTHOLD 位置 1 来停止正常接收过程。
   a) 必须读取 FIFO 中的剩余数据字节数，直到接收 FIFO 为空（RXFIFOE 标志置 1），当 IDMAEN = 0 时，FIFO 必须使用 FIFORST 复位。
2. 通过 DHOLD 标志指示对已从 FIFO 中读取所有数据以及挂起完成的确认。
   a) 恢复操作后仍需读取的剩余数据字节数（数据块的倍数）必须通过 DATACOUNT 指示的剩余字节数确定。

注： *挂起过程中出现的 DTIMEOUT 标志必须忽略。*

要继续从卡中接收数据，固件必须执行以下步骤：

1. 必须在 DATALENGTH 中编程剩余的数据字节数（数据块的倍数）。
2. 必须在 DTDIR 位中将 DPSM 配置为接收数据。
3. 必须从 CPSM 发送恢复命令，此时 CMDTRANS 位置 1 且 CMDSUSPEND 位置 1，该命令在数据传输恢复（响应位 DF = 1）且 DPSM 使能时结束中断周期，之后卡会继续发送数据。

在向卡发送数据时，SDMMC 可以在写入数据块 CRC 状态结束（DPSM 处于繁忙状态）之后挂起写入操作。在向卡发送挂起命令前，固件必须执行以下步骤：

1. 使能 DHOLD 标志（当 IDMAEN = 0 时使能 DBCKEND 标志）
2. 必须防止 DPSM 通过将 DTHOLD 置 1 开始发送新的数据块。
3. 当 IDMAEN = 0 时：当接收到 DBCKEND 标志时，将停止数据传输。固件可以停止填充 FIFO，之后必须使用 FIFORST 复位 FIFO。恢复操作后，仍保留在 FIFO 中的任何字节都需要重新写入。
4. 当接收到 DHOLD 标志时，将停止数据传输。恢复操作后仍需写入的剩余数据字节数必须通过 DATACOUNT 指示的剩余字节数确定。
5. 要挂起卡，必须在 CMDSUSPEND 位置 1 时通过 CPSM 发送挂起命令。这样，当总线挂起时（响应位 BS = 0），可在挂起命令响应之后启动中断周期。

要继续向卡发送数据，固件必须执行以下步骤：

1. 必须在 DATALENGTH 中编程剩余的数据字节数。
2. 必须通过设置 DTDIR 来配置 DPSM 的传输方向，通过 CPSM 发送恢复命令（CMDTRANS 位置 1 且 CMDSUSPEND 位置 1）的方式来使能 DPSM。这会结束中断周期并开始数据传输。DPSM 在 SDMMC_D0 未发出繁忙信号时进入 Wait_S 状态，而在 SDMMC_D0 发出繁忙信号时进入繁忙状态。
3. 当 IDMAEN = 1 时：需要针对要传输的剩余字节重新编程 IDMA。
4. 当 IDMAEN = 0 时：固件必须开始使用剩余的数据填充 FIFO。

SD I/O 读取等待

有两种方法可以在块间隙内暂停数据传输：

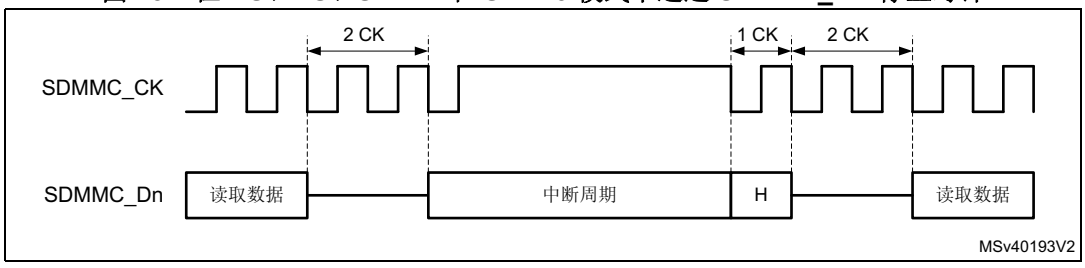
1. 停止 SDMMC_CK。
2. 在 SDMMC_D2 上使用读取等待信号。

SDMMC 可以根据*表 193* 通过寄存器设置执行读取等待。

根据 SDMMC 工作模式（DS、HS、SDR12 和 SDR25）或（SDR50、SDR104 和 DDR），每种方法都有不同的特性。

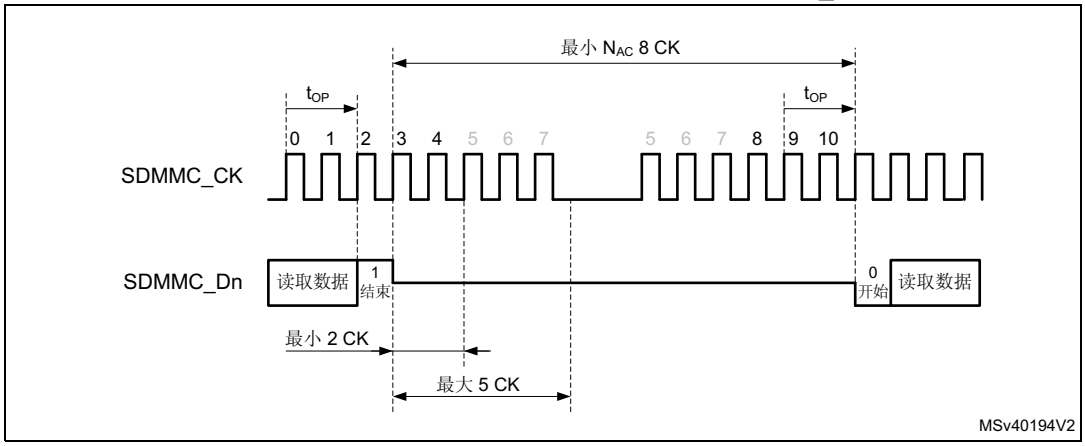
在 DS、HS、SDR12 和 SDR25 模式下通过停止 SDMMC_CK 实现暂停读取操作的时序中，SDMMC_CK 可在结束位之后的 2 个 SDMMC_CK 周期后停止。主机就绪后将通过重启时钟进行恢复，请参见*图 181*。

图 181. 在 DS、HS、SDR12 和 SDR25 模式下通过 SDMMC_CK 停止时钟



在 SDR50、SDR104 和 DDR50 模式下通过停止 SDMMC_CK 实现暂停读取操作的时序中，SDMMC_CK 可在结束位之后的 2-5 个 SDMMC_CK 周期后停止。主机就绪后将通过重启时钟进行恢复，请参见*图 182*。（在 DDR50 模式下，如果时钟线为低电平，则必须在下降沿后停止 SDMMC_CK）。

图 182. 在 DDR50、SDR50 和 SDR104 模式下通过 SDMMC_CK 停止时钟



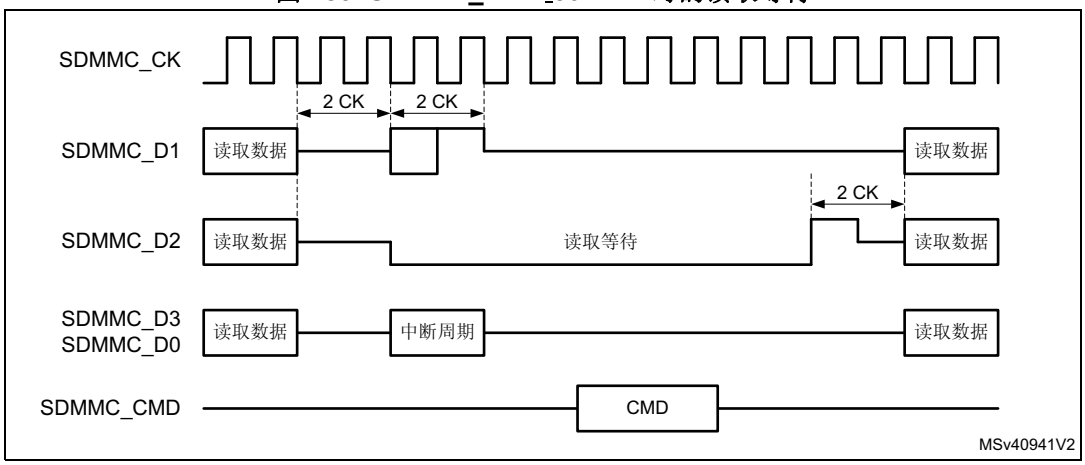
在读取等待 SDMMC_CK 时钟停止期间，如果 RWSTART 置 1，DSPM 会在当前接收到的数据块 CRC 的结束位之后停止时钟。时钟将在向 RWSTOP 位写入 1 后再次启动，之后 DPSM 将等待来自卡的起始位。

SDMMC_CK 停止时，无法向卡发出任何命令。在读取等待间隔期间，SDMMC 仍可以在 SDMMC_D1 上检测 SDIO 中断。

SDMMC_D2 (RW) 上可选的读取等待信号操作仅针对 SD 的 1 位和 4 位模式进行了定义。“读取等待”操作允许主机发出如下信号：卡正在读取多个寄存器 (IO_RW_EXTENDED, CMD53) 以临时停止数据传输，同时允许主机向 SD I/O 设备中的任何功能发送命令。要确定卡是否支持“读取等待”协议，主机必须测试内部卡寄存器中的功能位。

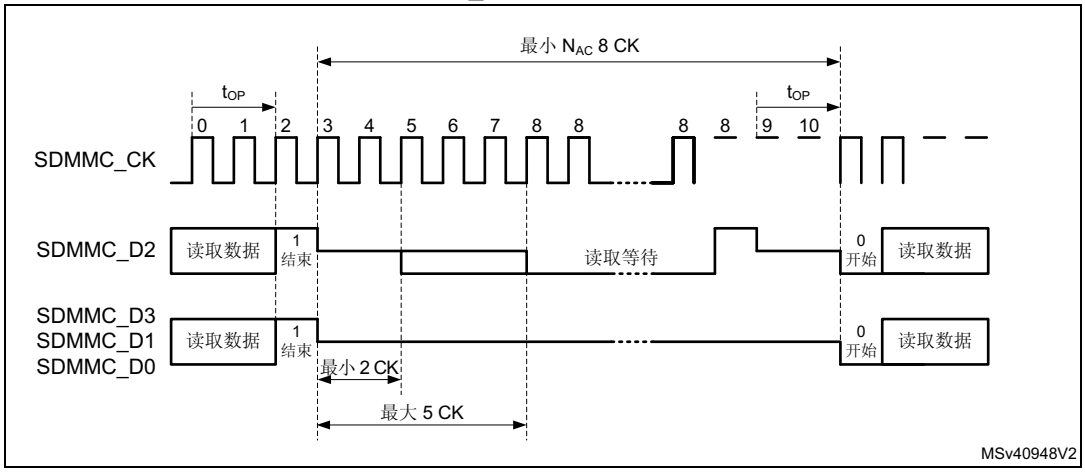
SDMMC_CK 小于 50 MHz（DS、HS、SDR12 和 SDR25）的读取等待的时序基于卡在 SDMMC_D1 上生成的中断周期。主机在中断周期内将 SDMMC_D2 置为低电平，请求卡进入读取等待状态。要退出读取等待状态，主机必须在将 SDMMC_D2 设置为高阻态之前的一个 SDMMC_CK 周期内将 SDMMC_D2 升为高电平，请参见*图 183*。

图 183. SDMMC_CK < 50 MHz 时的读取等待



对于SDMMC_CK 大于 50 MHz 的 SDR50 和 SDR104 以及 DDR50，卡会将 SDMMC_D2 上的读取等待请求视为异步事件。主机在 2-5 个 SDMMC_CK 周期后将 SDMMC_D2 置为低电平，请求卡进入读取等待状态。要退出读取等待状态，主机必须在将 SDMMC_D2 设置为高阻态之前的一个 SDMMC_CK 周期内将 SDMMC_D2 升为高电平。主机必须基于 SDMMC_CK 时钟将 SDMMC_D2 升为高电平（请参见*图 184*）。

图 184. SDMMC_CK ≥ 50 MHz 时的读取等待



在读取等待 SDMMC_D2 信号期间，如果 RWSTART 置 1，DPSM 将在当前接收的数据块 CRC 的结束位后驱动 SDMMC_D2。当将 1 写入 RWSTOP 位时，SDMMC_D2 上的读取等待信号将被删除。DPSM 保持 R_W 状态的时间将延长两个 SDMMC_CK 时钟周期，以将 SDMMC_D2 驱动为 1 并保持一个时钟周期（根据 SDIO 规范），之后 DPSM 将等待来自卡的起始位。

在 SDMMC_D2 上的读取等待信号传输期间，可以向卡发出命令。在读取等待间隔期间，SDMMC 可以在 SDMMC_D1 上检测 SDIO 中断。

24.6.2 CMD12 发送时序

CMD12 用于停止/中止数据传输，卡数据传输在停止传输命令的结束位后的两个时钟周期后终止。

表 195. CMD12 用例

数据操作	停止传输命令 CMD12 的说明
SDMMC 流写入	通过发送停止传输命令来停止/中止数据传输。
SDMMC 开放式多块写入	通过发送停止传输命令来停止/中止数据传输。 如果卡检测到错误，主机必须通过发送停止传输命令来中止操作。
预定义块数的 SDMMC 块写入	在该类型的多块写入操作结束时，不需要停止传输命令。（在卡接收到最后一个块后发送停止传输命令被视为非法命令。） 如果卡检测到错误，主机必须通过发送停止传输命令来中止操作。
SDMMC 流读取	通过发送停止传输命令来停止/中止数据传输。
SDMMC 开放式多块读取	通过发送停止传输命令来停止/中止数据传输。 如果卡检测到错误，主机必须通过发送停止传输命令来中止操作。
预定义块数的 SDMMC 块读取	在该类型的多块读取操作结束时，不需要停止传输命令。（在卡发送最后一个块后发送停止传输命令被视为非法命令。） 通过发送停止传输命令可中止事务。 如果卡检测到错误，主机必须通过发送停止传输命令来中止操作。

所有数据写入和读取命令都可以随时通过停止传输命令 CMD12 中止。如果正在进行数据传输，请按照以下数据中止步骤操作：

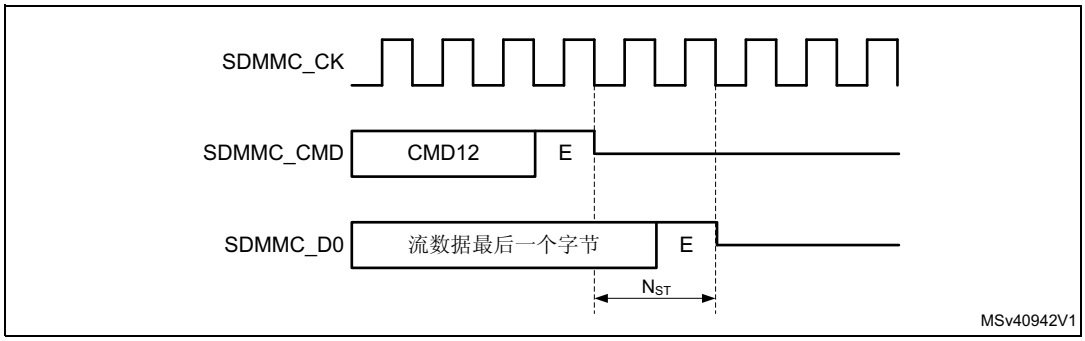
1. 在寄存器中加载 CMD12 停止传输命令并将 CMDSTOP 位置 1。
   a) 当命令发送到 DPSM 时，这将导致生成 CPSM 中止信号。
2. 将 CPSM 配置为立即发送命令（将 WAITPEND 位清零）。
   a) 发送数据时，卡在停止传输命令结束位后的 2 个周期后停止数据传输。
   未发送数据时，卡不会开始发送任何新数据。
   b) 发送数据时，主机会在停止传输命令结束位之后发送最后一个数据位，后跟一个结束位。
   未发送数据时，主机不会开始发送任何新数据。
3. 当 IDMAEN = 0 时，需要使用 FIFORST 复位 FIFO。
   a) 向卡写入数据时。出现 CMDREND 标志时，固件必须停止向 FIFO 写入数据。随后，必须使用 FIFORST 复位 FIFO，这会清空 FIFO。
   b) 从卡读取数据时。出现 CMDREND 标志时，固件必须从 FIFO 读取剩余数据。随后，必须使用 FIFORST 复位 FIFO。
4. 当 IDMAEN = 1 时，硬件处理 FIFO。
   a) 向卡写入数据时。出现 CPSM 中止信号时，硬件会停止 IDMA，随后 FIFO 会清空。
   b) 从卡读取数据时。出现 CPSM 中止信号时，硬件指示 IDMA 将剩余数据从 FIFO 传输到 RAM。
5. 当 FIFO 为空/复位时，将生成 DABORT 标志。

流操作和 CMD12

要在待传输的最后一个字节之后停止流传输，必须在数据流的最后一个字节结束时发送 CMD12 结束位时序。请按照以下流数据写入步骤操作：

1. 在 DPSM 中初始化流数据，DTMODE = MCC 流数据传输。
2. 从 CPSM 发送 WRITE_DATA_STREAM 命令 (CMDTRANS = 1)。
3. 在命令寄存器中预加载 CMD12，CMDSTOP 位置 1。
4. 将 CPSM 配置为仅在最后一个数据（根据 DATALENGTH）的等待挂起 (WAITPEND = 1) 结束后发送命令。
5. 使能 CPSM 发送 STOP_TRANSMISSION 命令，流数据结束位和命令结束位对齐。
   a) 当 DATALENGTH > 5 字节时，CPSM 中等待命令 CMD12 与数据传输结束位对齐。
   b) 当 DATALENGHT < 5 字节时，命令 CMD12 提前启动，DPSM 保持 Wait_S 状态，以将数据传输结束位与 CMD12 结束位对齐。
6. 可通过将 WAITPEND 位清零随时中止流数据写入过程。这会导致预加载的 CMD12 立即发送，并停止数据流写入过程。

图 185. CMD12 流时序



要在最后一个字节之后停止流传输读取过程，必须在数据流最后一个字节后发送 CMD12 结束位时序。请按照以下流数据读取步骤操作：

1. 等待 DPSM 接收到所有数据且从 FIFO 读取了所有数据（DATAEND 标志）。
   a) 即使卡发送更多数据，DPSM 接收的数据也不会超过 DATALENGTH 所指示的数据量。
2. 通过 CPSM 发送 CMD12。
   a) CMD12 使卡停止发送数据。

注： *即使卡继续发送数据，SDMMC 也不会在 DATACOUNT = 0 时从卡中继续接收任何数据。*

块操作和 CMD12

要在数据结束时停止块传输，必须在最后一个块结束位之后发送 CMD12 结束位。

向卡写入数据时，必须在写入数据块 CRC 令牌结束位之后发送 CMD12 结束位。这要求 CMD12 发送过程遵循数据块传输时序。要停止开放式多块写入操作，请按照以下步骤操作：

1. 在开始数据传输之前，将 DTMODE 设置为“以 STOP_TRANSMISSION 命令结束块数据传输”。
2. 等待 DPSM 发送完所有数据并等待接收到 CRC 令牌（DATAEND 标志）。
   a) DPSM 发送的数据不会超过 DATALENGTH 所指示的数据量。
3. 通过 CPSM 发送 CMD12。
   a) CMD12 将卡设置为空闲模式。

当从卡读取数据时，CMD12 结束位必须最早发送，即在卡读取数据块最后一个数据位时发送。这要求 CMD12 发送过程遵循数据块接收时序。请按照以下停止开放式多块读取数据块步骤操作：

1. 在开始数据传输之前，将 DTMODE 设置为“以 STOP_TRANSMISSION 命令结束块数据传输”。
2. 等待 DPSM 接收到所有数据且从 FIFO 读取了所有数据（DATAEND 标志）。
   a) 即使卡发送更多数据，DPSM 接收的数据也不会超过 DATALENGTH 所指示的数据量。
3. 通过 CPSM 发送 CMD12 并将 CMDSTOP 位置 1。
   a) CMD12 使卡停止发送更多数据，并将卡设置为空闲模式。卡中止正在进行的任何块传输操作。

注： *即使卡继续发送数据，SDMMC 也不会在 DATACOUNT = 0 时从卡中继续接收任何数据。*

24.6.3 睡眠 (CMD5)

e•MMC 卡可以通过 CMD5 在睡眠状态和待机状态之间转换。在睡眠状态下，卡的功耗降至最低，此时可关闭 Vcc 电源。

CMD5 (SLEEP) 用于启动从待机状态到睡眠状态的状态转换。在转换阶段，卡指示繁忙，同时会将 SDMMC_D0 拉为低电平。当卡停止将 SDMMC_DO 线拉为低电平时，表示已进入睡眠状态。

要将卡设置为睡眠状态，请按照以下步骤操作：

1. 使能 BUSYD0END 中断。
2. 发送 CMD5 (SLEEP)。
3. 出现 BUSYD0END 中断时，卡处于睡眠状态。
4. 可以关闭 Vcc 电源

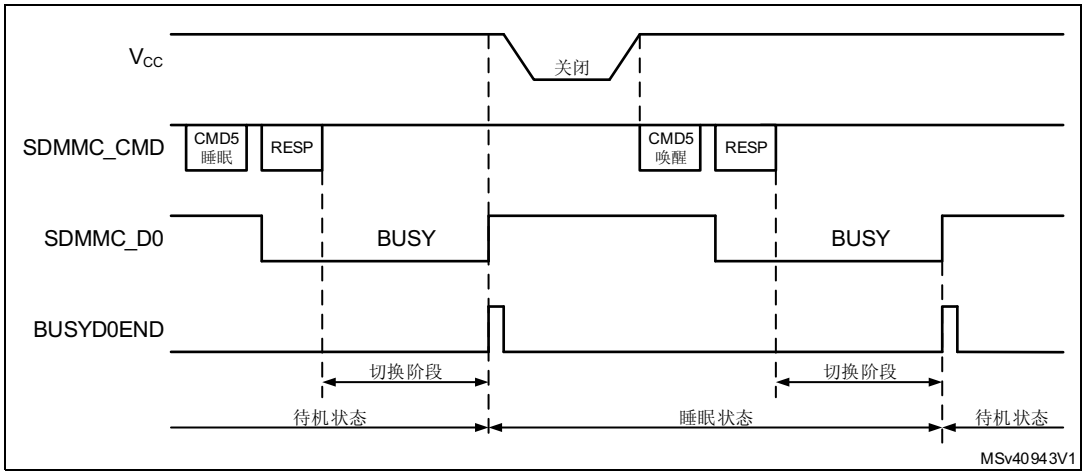
CMD5 (AWAKE) 用于启动从睡眠状态到待机状态的状态转换。在转换阶段，卡指示繁忙，同时会将 SDMMC_D0 拉为低电平。当卡停止将 SDMMC_DO 线拉为低电平时，表示已进入待机状态。

要将卡设置为睡眠状态，请按照以下步骤操作：

1. 接通 Vcc 电源，等待其达到最低工作电压。
2. 使能 BUSYD0END 中断。
3. 发送 CMD5 (AWAKE)。
4. 出现 BUSYD0END 中断时，卡处于待机状态。

只有在进入睡眠状态后，才能关闭 Vcc 电源。在发送 CMD5 (AWAKE) 之前，必须重新安装 Vcc 电源。

图 186. CMD5 睡眠唤醒过程



24.6.4 中断模式 (Wait-IRQ)

主机和卡同时进入和退出中断模式 (Wait-IRQ)。在中断模式下，不会进行任何数据传输。唯一允许的消息是来自卡或主机的中断服务请求响应。为了使中断模式正常工作，必须根据开漏模式下可实现的 SDMMC_CMD 数据速率设置 SDMMC_CK 频率，具体取决于电容负载和上拉电阻。CLKDIV 必须设置为 > 1，SETCLKRX 必须选择 sdmmc_io_in_ck 或 SDMMC_CLKin 源。

主机必须确保发出 CMD40 (GO_IRQ_STATE) 之前卡处于待机状态。在等待中断响应时，SDMMC_CK 时钟信号必须保持有效。

中断模式下的卡（IRQ 状态）：

- 等待内部卡中断事件。出现中断事件后，卡开始发送中断服务请求响应。该响应在开漏模式下发送。
- 等待内部卡中断事件时，卡还会监视 SDMMC_CMD 线上的起始位。检测到起始位后，卡中止中断模式并转换到待机状态。

中断模式下的主机（CPSM 等待状态，等待中断）：

- 等待卡中断服务请求响应（起始位）。
- 等待卡中断服务请求响应时，主机可中止中断模式（通过将 WAITINT 寄存器位清零），这会导致主机在开漏模式下发送 RCA = 0x0000 的中断服务请求响应 R5。

发送中断服务请求响应时，发送方逐位监视 SDMMC_CMD 位流。如果发送方的中断服务请求响应位与 SDMMC_CMD 线上的位不对应，则将停止发送。当存在多个发送方时，只有一个发送方成功发送其完整的中断服务请求响应。如果主机与其中一个发送方同时发送，则主机在传输位之后停止发送。

要处理中断模式，请按照以下步骤操作：

1. 根据开漏模式下可实现的 SDMMC_CMD 数据速率设置 SDMMC_CK 频率，CLKDIV 必须设置为 > 1，SETCLKRX 必须选择 sdmmc_io_in_ck。
2. 在命令寄存器中加载 CMD40 (GO_IRQ_STATE)。
3. 通过将 WAITINT 寄存器位置 1 使能等待中断。
4. 将 CPSM 配置为立即发送命令。
   a) 这会导致发送 CMD40，并且使 CPSM 在等待状态下停止，等待中断服务请求响应。
5. 要退出等待中断状态（CPSM 等待状态）：
   a) 检测到中断服务请求响应起始位后，CPSM 将变为接收状态以接收响应。是否成功接收到响应通过 CMDREND 或命令 CRC 错误标志来指示。
   b) 为中止中断模式，主机会将 WAITINT 寄存器位清零。这会导致主机自行发送中断服务请求响应，从而使 CPSM 变为接收状态。是否成功接收到响应通过 CMDREND 或命令 CRC 错误标志来指示。

注： *当同步发送中断服务请求响应起始位发生冲突时，主机在传输位之后丢失总线访问。*

24.6.5 启动操作

在启动操作模式下，主机可以通过 2 种启动操作功能从卡中读取启动数据：

1. 正常启动。（使 CMD 线保持低电平）
2. 备用启动（发送具有参数 0xFFFFFFFA 的 CMD0）

可以根据以下配置选项读取启动数据，具体取决于卡寄存器设置：

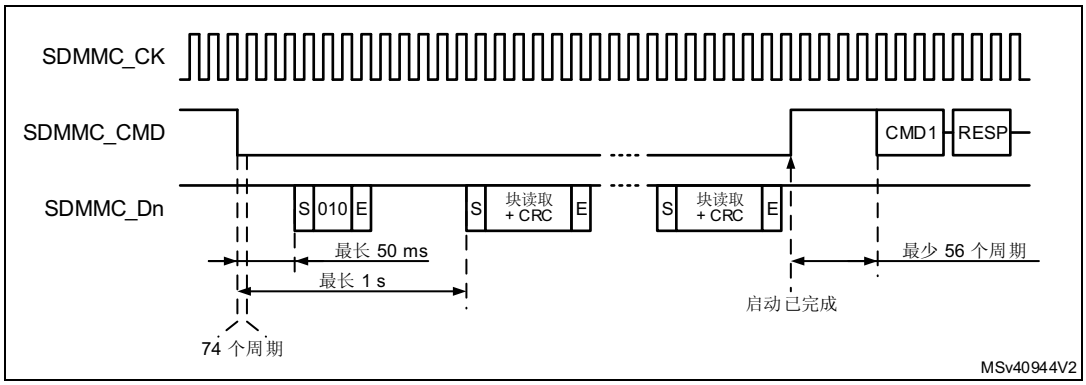
- 从中读取启动数据的分区（EXT_CSD 字节 [179]）
- 启动数据大小（EXT_CSD 字节 [226]）
- 启动期间的总线配置（EXT_CSD 字节 [177]）
- 从卡接收启动确认。（EXT_CSD 字节 [179]）

如果使能启动确认，则卡将在请求启动模式（通过将 CMD 线置为低电平）或发送具有参数 0xFFFFFFFA 的 CMD0 后的 50 ms 内在 SDMMC_D0 上发送模式 010，并将提供启动确认超时 (ACKTIMEOUT) 和确认状态 (ACKFAIL)。

正常启动操作

如果在卡上电或复位后 SDMMC_CMD 线的低电平状态持续至少 74 个时钟周期，则卡将在发出第一个命令之前识别出正在发起启动模式。在 CMD 线变为低电平后的 1 秒内，卡开始在 SDMMC_Dn 线上发送第一个启动代码数据。主机必须使 SDMMC_CMD 线保持低电平，直到读取完所有启动数据为止。主机可通过将 SDMMC_CMD 线拉高来终止启动模式。

图 187. 正常启动模式操作



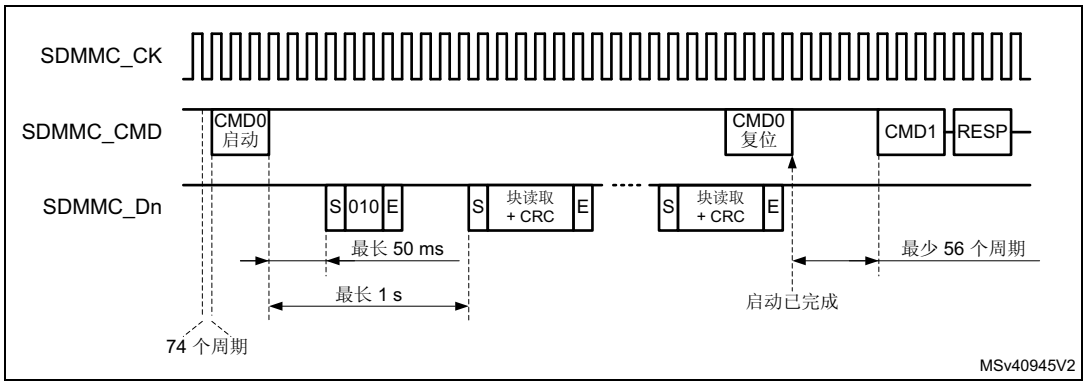
要执行正常的启动过程，需按照以下步骤操作：

1. 将卡复位。
2. 如果请求启动确认，需使能 BOOTACKEN、设置 ACKTIME 并使能 ACKFAIL 和 ACKTIMEOUT 中断。
3. 将 DPSM 设置为接收模式 (DTDIR) 并在 DATALENGTH 中设置要接收的数据字节数，以使能数据接收。
4. 使能 DTIMEOUT、DATAEND 和 CMDSENT 中断，用于完成启动命令确认。
5. 通过 BOOTMODE 选择正常启动操作模式，并通过 BOOTEN 使能启动。启动过程的启动方法是通过 CPSMEN 使能 CPSM。这会导致：
   - SDMMC_CMD 被驱动为低电平。（BOOTMODE = 正常启动）。
   - ACK 超时被启动。
   - DPSM 被使能。
6. 可通过 ACKFAIL 标志或 ACKTIMEOUT 标志（使能时）检测是否接收到不正确的启动确认。
   - 当接收到不正确的启动确认时，将出现 ACKFAIL 标志。
   - 当未及时接收到启动确认时，将出现 ACKTIMEOUT 标志。
7. 当接收到所有启动数据时，会出现 DATAEND 标志。
   - 当数据 CRC 失败时，还会生成 DCRCFAIL 标志。
   - 当数据发生超时时，还会生成 DTIMEOUT 标志。
8. 当接收到最后一个数据时，从 FIFO 读取数据，直到 FIFO 为空，之后将生成数据结束 DATAEND 标志。
   - SDMMC 已接收完所有数据且 DPSM 处于禁止状态。
9. 通过固件清零 BOOTEN 将终止启动过程，这会导致 SDMMC_CMD 线变为高电平。56 个周期后将生成 CMDSENT 标志，以指示可以发送新命令。
   a) 如果在接收到所有数据之前通过固件中止启动过程，CPSM 中止信号会停止数据接收并禁止 DPSM，此时会触发 DABORT 标志（使能时）。
10. CMDSENT 标志用于指示启动过程结束，卡已准备好接收新命令。

备用启动操作

卡上电或复位后，如果主机在 74 个时钟周期后发送具有参数 0xFFFFFFFA 的 CMD0，则卡将在发出 CMD0 之前识别出正在发起启动模式。在发送具有参数 0xFFFFFFFA 的 CMD0 后的 1 秒内，卡开始在 SDMMC_Dn 线上发送第一个启动代码数据。主机将通过发送 CMD0（复位）来终止启动操作。

图 188. 备用启动模式操作



要执行备用启动过程，需按照以下步骤操作：

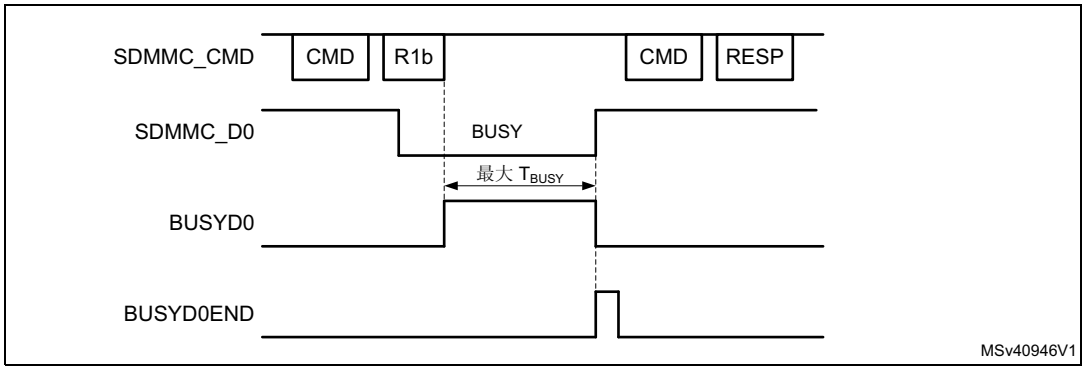
1. 使 SDMMC 变为掉电状态，并将卡复位
2. 使 SDMMC 变为上电状态。这可保证在任何命令之前提供 74 个 SCDMMC_CK 周期的时钟驱动。
3. 如果请求启动确认，需使能 BOOTACKEN、设置 ACKTIME 并使能 ACKTIMEOUT 标志。
4. 将 DPSM 设置为接收模式 (DTDIR) 并在 DATALENGTH 中设置要接收的数据量，以使能数据接收。使能 DTIMEOUT 和 DATAEND 标志。
5. 通过 BOOTMODE 选择备用启动操作模式，在命令寄存器中加载具有参数 0xFFFFFFFA 的 CMD0。使能 CMDSENT 标志以指示完成启动命令确认，并通过 BOOTEN 使能启动。启动过程的启动方法是通过 CPSMEN 使能 CPSM。这会导致：
   - 加载的命令和参数被发送出去。（BOOTMODE = 备用启动）。
   - ACK 超时被启动。
   - DPSM 被使能。
6. 发送命令后，将生成 CMDSENT 标志，此时 BOOTEN 位必须清零。
7. 可通过 ACKFAIL 标志（使能时）检测是否接收到启动确认。
   - 当未及时接收到启动确认时，将出现 ACKTIMEOUT 标志。
8. 当接收到所有启动数据时，会出现 DATAEND 标志。
   - 当数据 CRC 失败时，还会生成 DCRCFAIL 标志。
   - 当数据发生超时时，还会生成 DTIMEOUT 标志。
9. 当接收到最后一个数据时，从 FIFO 读取数据，直到 FIFO 为空，之后将生成数据结束 DATAEND 标志。
   - SDMMC 已接收完所有数据且 DPSM 处于禁止状态。

10. 在 BOOTMODE = 备用启动时，通过发送 CMD0（复位）以中止启动过程之前，必须将 BOOTEN 位清零。这会导致在命令后的 56 个周期后出现 CMDSENT 标志。
    – 如果在接收到所有数据之前通过固件中止启动过程，CPSM 中止信号会停止数据传输并禁止 DPSM，此时会触发 DABORT 标志（使能时）。
11. CMDSENT 标志用于指示启动过程结束，卡已准备好接收新命令。当 RESET 命令成功发送后，必须将 BOOTMODE 控制位清零才能终止启动操作。

24.6.6 响应 R1b 的处理

当发送具有 R1b 响应的命令时，繁忙信号将反映在 BUSYD0 寄存器位上，而繁忙释放信号将通过 BUSYD0END 标志反映。SDMMC_D0 线在 R1b 响应结束时进行采样，在 BUSYD0 寄存器位中进行指示。当 SDMMC_D0 线释放繁忙状态时，BUSYD0 寄存器位将复位为不忙，同时将生成 BUSYD0END 标志。

图 189. 命令响应 R1b 繁忙信号



发送命令之前，必须在 DATATIME 寄存器中设置预期的最长繁忙时间。当 R1b 响应繁忙状态保持有效的时间超过编程的时间时，DTIMEOUT 标志（使能时）置 1。

发送具有 R1b 响应的命令时，如果要检测 SDMMC_D0 繁忙信号，请按照以下步骤操作：

- 使能 CMDREND 标志。
- 通过 CPSM 发送命令。
- 出现 CMDREND 标志时，检查 BUSYD0 寄存器位。
  – 如果 BUSYD0 指示不忙，则向固件发出繁忙释放信号
  – 如果 BUSYD0 指示繁忙，则等待 BUSYD0END 标志
- 出现 BUSYD0END 标志时，向固件发出繁忙已释放信号。
- 出现 DTIMEOUT 标志时，表示繁忙状态保持有效的时间超过编程的时间。

24.6.7 复位和卡掉电再上电

复位

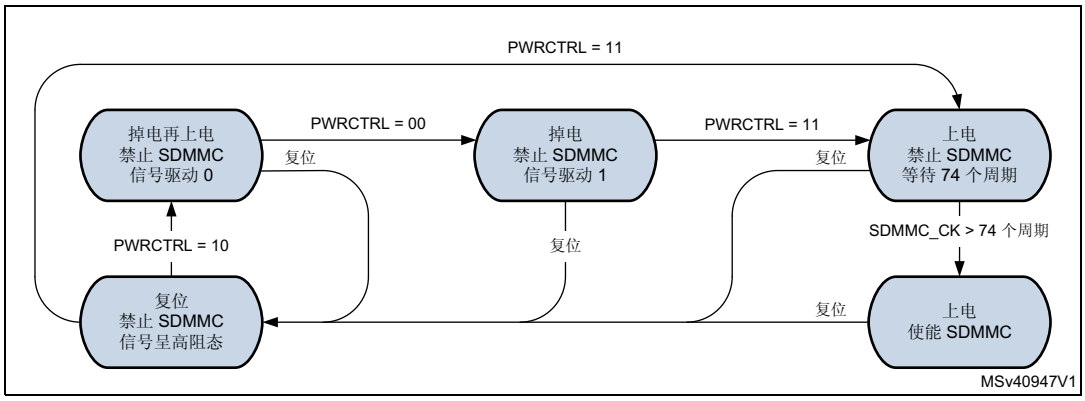
复位后，SDMMC 处于复位状态。在此状态下，SDMMC 将被禁止，不能传输任何命令和数据。SDMMC_D[7:0] 和 SDMMC_CMD 处于高阻态，SDMMC_CK 驱动为低电平。

SDMMC 必须在变为上电状态之前进行配置。

在上电状态下，SDMMC_CK 时钟处于运行状态。首先提供 74 个 SDMMC_CK 周期的时钟驱动，之后将使能 SDMMC，此时即可传输命令和数据。

SDMMC 状态由固件通过 PWRCTL 寄存器位进行控制，如*图 190.* 所示。

图 190. SDMMC 状态控制

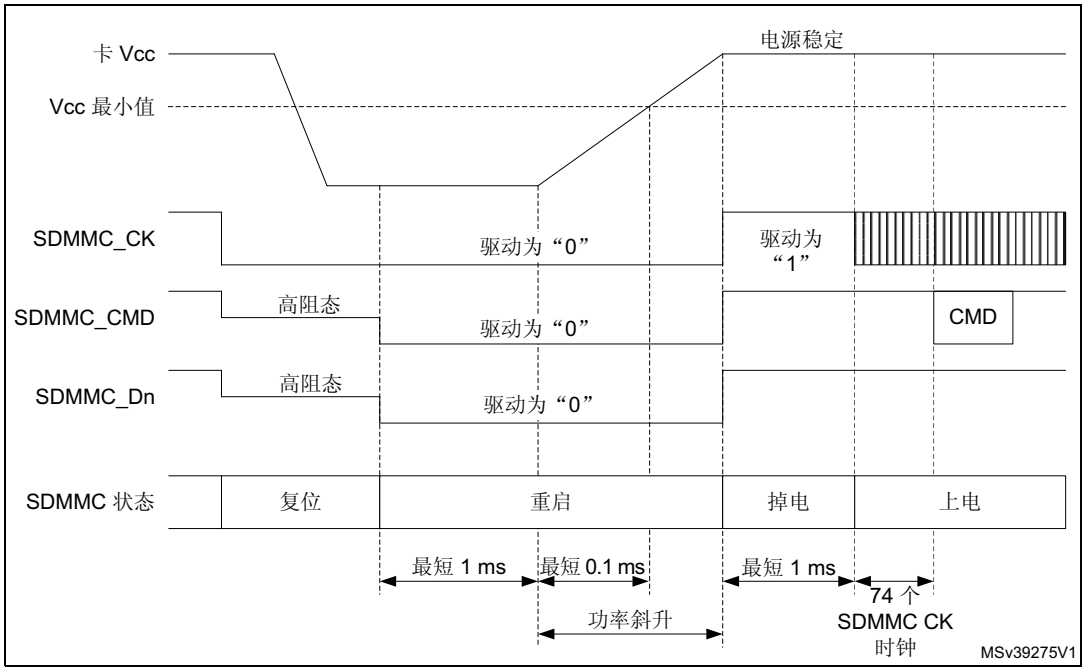


对卡进行掉电再上电

要对卡进行掉电再上电，请按照以下步骤进行操作：

1. 通过 RCC.SDMMCxRST 寄存器位复位 SDMMC。这会使 SDMMC 复位为复位状态，CPSM 和 DPSM 复位为空闲状态。
2. 禁止卡的 Vcc 电源。
3. 将 SDMMC 设置为掉电再上电状态。这会使 SDMMC_D[7:0]、SDMMC_CMD 和 SDMMC_CK 被驱动为低电平，以防止卡通过信号线供电。
4. 至少等待 1 ms 后，才能使能卡的 Vcc 电源。
5. 在电源斜升周期后，将 SDMMC 设置为掉电状态并至少持续 1 ms。SDMMC_D[7:0]、SDMMC_CMD 和 SDMMC_CK 将设置为“1”。
6. 经过 1 ms 的延迟后，将 SDMMC 设置为上电状态，在此期间使能 SDMMC_CK 时钟。
7. 经过 74 个 SDMMC_CK 周期后，可以向卡发送第一个命令。

图 191. 卡掉电再上电/上电图



24.7 硬件流控制

数据传输期间的硬件流控制功能用于避免 FIFO 下溢（发送模式）和上溢（接收模式）错误。

具体行为是在数据传输过程中停止 SDMMC_CK 并冻结 SDMMC 状态机。当 FIFO 无法发送或接收数据时，将停止数据传输。数据传输一直保持停止状态，直到发送 FIFO 半满或已存储 DATALENGHT 所指示的所有数据，或者直到接收 FIFO 半空。只有由 SDMMC_CK 提供时钟的状态机才会冻结，AHB 接口仍保持活动状态。因此，即使激活流控制，仍可填充或清空 FIFO。

发生 IDMA 链表传输错误时，会禁止硬件流控制。因此，根据 IDMA 链表传输错误发生的时间，还可能发生下溢或上溢错误（请参见 *IDMA 传输错误管理*一节）。

要在数据传输期间使能硬件流控制，HWFC_EN 寄存器位必须置 1。复位后，硬件流控制将禁止。

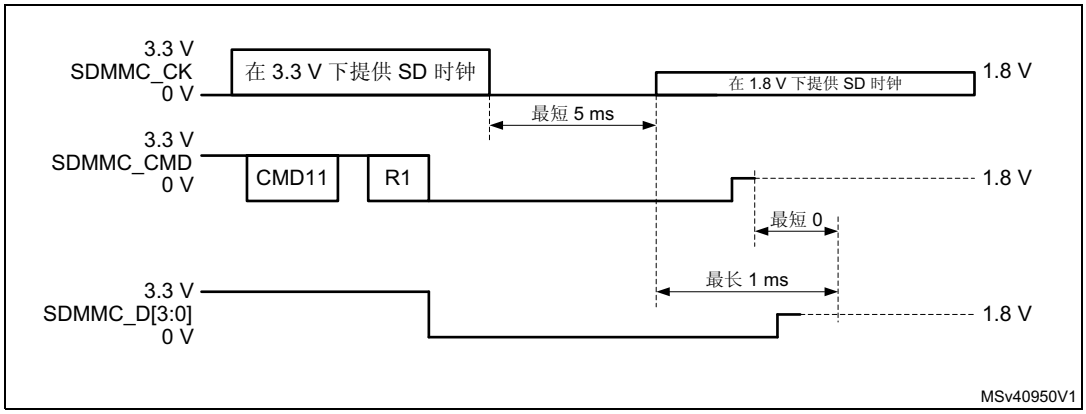
仅当 SDMMC_Dn 接收数据与 SDMMC_CK 循环对齐或发送数据时，才必须使用硬件流控制。当接收数据并使用 DLYB 延迟块的 sdmmc_fb_ck 时。在 SDR104 接收模式下，如果 t_{OP} 和 Dt_{OP} 延迟 > 1 个周期，则无法使用硬件流控制。

警告： **当使用可变延迟并使能硬件流控制时，在数据传输期间发送命令可能会干扰命令响应接收。（CMD12 不受此影响，除非在数据传输过程中使用 CMD12 来中止。）**

24.8 超高速 I 相 (UHS-I) 电压切换

UHS-I 模式（SDR12、SDR25、SDR50、SDR104 和 DDR50）需要 1.8V 信号支持。上电后，卡将以 3.3V 模式启动。CMD11 将调用电压切换序列以切换到 1.8V 模式。当电压序列成功完成时，卡将以默认 SDR12 进入 UHS-I 模式，卡输入和输出时序将发生变化。

图 192. CMD11 信号电压切换序列



要执行信号电压切换序列，需按以下步骤进行操作：

1. 在启动电压切换过程之前，必须将 SDMMC_CK 频率设置在 100 kHz - 400 kHz 范围内。
2. 主机通过在发送 CMD11 之前将 VSWITCHEN 位置 1 来启动电压切换过程。
3. 卡返回 R1 响应。
   - 如果响应 CRC 通过，电压切换过程将继续执行，主机将不再驱动 CMD 和 SDMMC_D[3:0] 信号，直到电压切换序列完成。在响应后再经过若干周期，SDMMC_CK 停止，CKSTOP 标志置 1。
   - 如果响应 CRC 失败（CCRCFAIL 标志）或未在超时（CTIMEOUT 标志）之前收到响应，则电压切换过程将停止。
4. 在 R1 响应之后的下一个时钟，卡将 CMD 和 SDMMC_D[3:0] 驱动为低电平。
5. 收到 R1 响应后，主机可以使用 BUSYD0 寄存器位监视 SDMMC_D0 线。SDMMC_D0 线在响应后的两个 SDMMC_CK 时钟周期后进行采样。固件可以读取 CKSTOP 标志后的 BUSYD0 寄存器位。
   - 当检测到 BUSYD0 为低电平时，主机固件会将稳压器切换到 1.8V，然后通过将寄存器位 VSWITCH 置 1 来指示 SDMMC 启动电压切换序列的时序关键部分。硬件继续停止 SDMMC_CK，方法是使 SDMMC_CK 保持低电平并持续至少 5 ms。
   - 当检测到 BUSYD0 为高电平时，主机会中止电压切换序列并对卡进行掉电再上电。
6. 检测到 SDMMC_CK 为低电平后，卡开始将信号电压切换到 1.8 V。
7. 主机 SDMMC 硬件在至少 5 ms 后重新启动 SDMMC_CK。
8. 在检测到 SDMMC_CK 切换后的 1 ms 内，卡会将 CMD 和 DAT[3:0] 驱动为高电平并至少持续 1 个 SDMMC_CK 周期，然后停止驱动 CMD 和 DAT[3:0]。

9. 在 SDMMC_CK 重启后的 1 ms 后，主机 SDMMC 硬件会将 SDMMC_D0 采样到 BUSYD0 中并将 VSWEND 标志置 1。
10. 出现 VSWEND 标志时，主机使用 BUSYD0 寄存器位检查 SDMMC_D0 线，以确认电压切换序列是否完成：
    - 当检测到 BUSYD0 为高电平时，表示电压切换成功完成。
    - 当检测到 BUSYD0 为低电平时，表示电压切换失败，主机将对卡执行掉电再上电。

停止 SDMMC_CK 所需的 5 ms 最短时间来自内部非门控 SDMMC_CK 时钟，其最大频率为 25 MHz（SD 模式），该频率由时钟分频比 CLKDIV 设置。超过 5 ms 的时间将以 2^{12} 个周期（400 kHz 时为 10.24 ms）计数。如果通过时钟分频比 CLKDIV 选择较低的 SDMMC_CK 频率，则停止 SDMMC_CK 时钟所需的时间会更长。

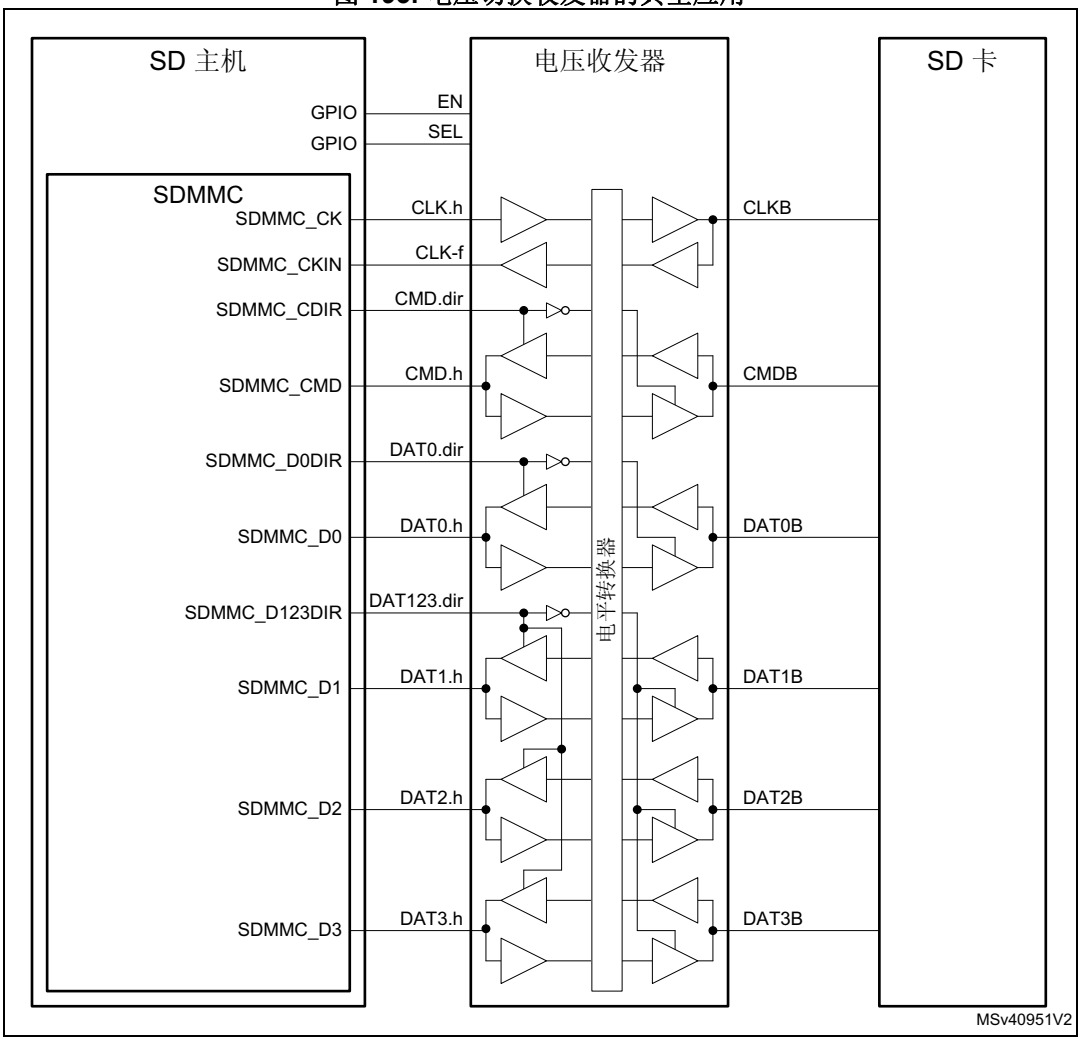
卡将 SDMMC_Dn 和 SDMMC_CMD 线驱动为高电平所需的 1 ms 最长时间来自内部非门控 SDMMC_CK，其最大频率为 25 MHz（SD 模式），该频率由时钟分频比 CLKDIV 设置。SDMMC 在 > 1 ms 时间后检查线路，该时间以 2^9 个周期（25 MHz 时为 1.28 ms）计数。如果通过时钟分频比 CLKDIV 选择较低的 SDMMC_CK 频率，则检查线路所需的时间会更长。

信号电压通过直接设备 SDMMC GPIO V_{DD} 电源电平切换或类似意法半导体 ST6G3244ME 的外部电平切换收发器来提供支持。

使用直接设备 SDMMC GPIO V_{DD} 电平切换时，在 1.8 V 信号下，设备 SDMMC GPIO 必须切换到高速模式，请参见*第 12 章：系统配置控制器 (SYSCFG)*。

警告： **当设备 SDMMC GPIO V_{DD} > 2.7 V 时，使能高速模式可能会损坏设备。**

图 193. 电压切换收发器的典型应用



为了连接标准信号旁的外部驱动器（电压切换收发器），SDMMC 使用以下信号：

SDMMC_CKIN 反馈输入时钟

SDMMC_CDIR CMD 的 I/O 方向控制信号。

SDMMC_D0DIR SDMMC_D0 的 I/O 方向控制信号。

SDMMC_D123DIR SDMMC_D1、SDMMC_D2 和 SDMMC_D3 的 I/O 方向控制信号。

电压收发器信号 **EN** 和 **SEL** 将通过通用 I/O 进行处理。

SDMMC_CDIR、SDMMC_D0DIR 和 SDMMC_D123DIR 信号的极性可通过 SDMMC_POWER.DIRPOL 控制位进行选择。

24.9 SDMMC 中断

表 196. SDMMC 中断

中断缩写	中断事件	事件标志	使能控制位	中断清除方法	退出睡眠模式
SDMMC	命令响应 CRC 失败	CCRCFAIL	CCRCFAILIE	CCRCFAILC	是
SDMMC	数据块 CRC 失败	DCRCFAIL	DCRCFAILIE	DCRCFAILC	是
SDMMC	命令响应超时	CTIMEOUT	CTIMEOUTIE	CTIMEOUTC	是
SDMMC	数据超时	DTIMEOUT	DTIMEOUTIE	DTIMEOUTC	是
SDMMC	发送 FIFO 下溢	TXUNDERR	TXUNDERRIE	TXUNDERRC	是
SDMMC	接收 FIFO 上溢	RXOVERR	RXOVERRIE	RXOVERRC	是
SDMMC	接收到命令响应	CMDREND	CMDRENDIE	CMDRENDC	是
SDMMC	已发送命令	CMDSENT	CMDSENTIE	CMDSENTC	是
SDMMC	数据传输结束	DATAEND	DATAENDIE	DATAENDC	是
SDMMC	数据传输保持	DHOLD	DHOLDIE	DHOLDC	是
SDMMC	已发送或接收到数据块	DBCKEND	DBCKENDIE	DBCKENDC	是
SDMMC	数据传输中止	DABORT	DABORTIE	DABORTC	是
SDMMC	发送 FIFO 半空	TXFIFOHE	TXFIFOHEIE	n.a.	是
SDMMC	接收 FIFO 半满	RXFIFOHF	RXFIFOHFIE	n.a.	是
SDMMC	发送 FIFO 已满	TXFIFOF	n.a.	n.a.	是
SDMMC	接收 FIFO 已满	RXFIFOF	RXFIFOFIE	n.a.	是
SDMMC	发送 FIFO 为空	TXFIFOE	TXFIFOEIE	n.a.	是
SDMMC	接收 FIFO 为空	RXFIFOE	n.a.	n.a.	是
SDMMC	命令响应繁忙结束	BUSYD0END	BUSYD0ENDIE	BUSYD0ENDC	是
SDMMC	SDIO 中断	SDIOIT	SDIOITIE	SDIOITC	是
SDMMC	启动确认失败	ACKFAIL	ACKFAILIE	ACKFAILC	是
SDMMC	启动确认超时	ACKTIMEOUT	ACKTIMEOUTIE	ACKTIMEOUTC	是
SDMMC	电压切换时序	VSWEND	VSWENDIE	VSWENDC	是
SDMMC	SDMM_CK 在电压切换时停止	CKSTOP	CKSTOPIE	CKSTOPC	是
SDMMC	IDMA 传输错误	IDMATE	IDMATEIE	IDMATEC	是
SDMMC	IDMA 缓冲区传输完成	IDMABTC	IDMABTCIE	IDMABTCC	是

24.10 SDMMC 寄存器

设备通过 32 位控制寄存器（可通过 AHB 从接口访问）与系统进行通信。

外设寄存器必须按字（32 位）进行访问。按字节（8 位）和半字（16 位）访问会触发 AHB 总线错误。

24.10.1 SDMMC 电源控制寄存器 (SDMMC_POWER)

SDMMC power control register

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIR POL	VSWI TCHEN	VSWI TCH	PWRCTRL[1:0]	
											rw	rw	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4 **DIRPOL**：数据和命令方向信号极性选择 (Data and command direction signals polarity selection)

只有在 SDMMC 处于掉电状态时 (PWRCTRL = 00) 才能写入该位。

0：当方向信号为低电平时，电压收发器 IO 驱动为输出。

1：当方向信号为高电平时，电压收发器 IO 驱动为输出。

位 3 **VSWITCHEN**：电压切换过程使能 (Voltage switch procedure enable)

只有在禁止 CPSM (CPSMEN = 0) 时才能通过固件写入该位。

该位用于在电压切换命令响应后停止 SDMMC_CK：

0：SDMMC_CK 时钟在成功接收到命令响应后保持不变。

1：SDMMC_CK 时钟在成功接收到命令响应后停止。

位 2 **VSWITCH**：电压切换序列启动 (Voltage switch sequence start)

该位用于启动电压切换序列的时序关键部分：

0：电压切换序列未启动且未激活。

1：电压切换序列已启动或已激活。

位 1:0 **PWRCTRL[1:0]**：SDMMC 状态控制位 (SDMMC state control bits)

只有在 SDMMC 未处于上电状态时 (PWRCTRL ≠ 11) 才能写入这些位。

这些位用于定义 SDMMC 信号的功能状态：

00：复位后，复位：SDMMC 禁止，卡的时钟停止，SDMMC_D[7:0] 和 SDMMC_CMD 处于高阻态，SDMMC_CK 驱动为低电平。
写入 00 时，掉电：SDMMC 禁止，卡的时钟停止，SDMMC_D[7:0]、SDMMC_CMD 和 SDMMC_CK 驱动为高电平。

01：保留。（写入 01 时，PWRCTRL 值不会改变）

10：掉电再上电，SDMMC 禁止，卡的时钟停止，SDMMC_D[7:0]、SDMMC_CMD 和 SDMMC_CK 驱动为低电平。

11：上电：卡由时钟驱动，前 74 个 SDMMC_CK 周期内，SDMMC 仍禁止。74 个周期后，使能 SDMMC，并根据 SDMMC 操作控制 SDMMC_D[7:0]、SDMMC_CMD 和 SDMMC_CK。

任何后续写操作都被忽略，PWRCTRL 值保持为 11。

24.10.2 SDMMC 时钟控制寄存器 (SDMMC_CLKCR)

SDMMC clock control register

偏移地址：0x004

复位值：0x0000 0000

SDMMC_CLKCR 寄存器控制 SDMMC_CK 输出时钟、sdmmc_rx_ck 接收时钟和总线宽度。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELCLKRX[1:0]		BUS SPEED	DDR	HWFC_EN	NEG EDGE
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WID BUS[1:0]		Res.	PWR SAV	Res.	Res.	CLKDIV[9:0]									
rw	rw		rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:20 **SELCLKRX[1:0]**：接收时钟选择 (Receive clock selection)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入这些位

00：选择 sdmmc_io_in_ck 作为接收时钟

01：选择 SDMMC_CKIN 反馈时钟作为接收时钟

10：选择 sdmmc_fb_ck 调节反馈时钟作为接收时钟

11：保留（选择 sdmmc_io_in_ck）

位 19 **BUSSPEED**：用于选择 SDMMC 操作模式的总线速度 (Bus speed for selection of SDMMC operating modes)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位

0：选择 DS、HS、SDR12、SDR25、传统兼容型、高速 SDR、高速 DDR 总线速度模式

1：选择 SDR50、DDR50、SDR104、HS200 总线速度模式。

位 18 **DDR**：数据速率信号选择 (Data rate signaling selection)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位

必须仅在 4 位或 8 位宽的总线模式下选择 DDR 速率。(WIDBUS > 00)。当 WIDBUS = 00（1 位宽总线）时，DDR = 1 不起作用。

必须仅在时钟分频比 > 1 的情况下选择 DDR 速率。(CLKDIV > 0)

0：SDR 单倍数据速率信号传输

1：DDR 双倍数据速率信号传输

位 17 **HWFC_EN**：硬件流控制使能 (Hardware flow control enable)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位

0：禁止硬件流控制

1：使能硬件流控制

如果使能硬件流控制，TXFIFOE 和 RXFIFOF 标志的含义将发生变化，请参见*第 24.10.11 节*中的 SDMMC 状态寄存器定义。

位 16 **NEGEDGE**：数据和命令的 SDMMC_CK 移相选择位 (SDMMC_CK dephasing selection bit for data and command)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位。
当时钟分频比 = 1 (CLKDIV = 0) 时，该位不起作用。数据和命令在 SDMMC_CK 下降沿发生变化。
0：当时钟分频比 > 1 (CLKDIV > 0) 且 DDR = 0 时：
– 命令和数据在 SDMMC_CK 上升沿后的 sdmmc_ker_ck 下降沿发生变化。
– SDMMC_CK 边沿出现在 sdmmc_ker_ck 上升沿。
当时钟分频比 > 1 (CLKDIV > 0) 且 DDR = 1 时：
– 命令在 SDMMC_CK 上升沿后的 sdmmc_ker_ck 下降沿发生变化。
– 数据在 SDMMC_CK 边沿后的 sdmmc_ker_ck 下降沿发生变化。
– SDMMC_CK 边沿出现在 sdmmc_ker_ck 上升沿。
1：当时钟分频比 > 1 (CLKDIV > 0) 且 DDR = 0 时：
– 命令和数据在生成 SDMMC_CK 下降沿的同一 sdmmc_ker_ck 上升沿发生变化。
当时钟分频比 > 1 (CLKDIV > 0) 且 DDR = 1 时：
– 命令在生成 SDMMC_CK 下降沿的同一 sdmmc_ker_ck 上升沿发生变化。
– 数据在 SDMMC_CK 边沿后的 SDMMC_CK 下降沿发生变化。
– SDMMC_CK 边沿出现在 sdmmc_ker_ck 上升沿。

位 15:14 **WIDBUS[1:0]**：宽总线模式使能位 (Wide bus mode enable bit)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位
00：默认 1 位宽总线模式：使用 SDMMC_D0（不支持 DDR）
01：4 位宽总线模式：使用 SDMMC_D[3:0]
10：8 位宽总线模式：使用 SDMMC_D[7:0]

位 13 保留，必须保持复位值。

位 12 **PWRSAV**：节能模式配置位 (Power saving configuration bit)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位
要实现节能模式，可在总线空闲时通过将 PWRSAV 置 1 来禁止 SDMMC_CK 时钟输出：
0：始终使能 SDMMC_CK 时钟
1：仅在总线激活时使能 SDMMC_CK

位 11:10 保留，必须保持复位值。

位 9:0 **CLKDIV[9:0]**：时钟分频系数 (Clock divide factor)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位。
该位字段定义输入时钟 (sdmmc_ker_ck) 与输出时钟 (SDMMC_CK) 之间的分频系数：
SDMMC_CK 频率 = sdmmc_ker_ck / [2 * CLKDIV].
0x000：SDMMC_CK 频率 = sdmmc_ker_ck / 1（不支持 DDR）
0x001：SDMMC_CK 频率 = sdmmc_ker_ck / 2
0x002：SDMMC_CK 频率 = sdmmc_ker_ck / 4
0x0XX：...
0x080：SDMMC_CK 频率 = sdmmc_ker_ck / 256
0xXXX：...
0x3FF：SDMMC_CK 频率 = sdmmc_ker_ck / 2046

注： *1 当 SD/SDIO 卡或 e•MMC 处于识别模式时，SDMMC_CK 频率必须小于 400 kHz。*

2 如果为所有卡分配相对卡地址，则可以将时钟频率更改为卡总线最大频率。

3 此寄存器的两次写访问至少需要相隔七个 sdmmc_hclk 时钟周期。在 SD I/O 卡的读取等待间隔期间也可以停止 SDMMC_CK；在此情况下，SDMMC_CLKCR 寄存器不对 SDMMC_CK 进行控制。

24.10.3 SDMMC 参数寄存器 (SDMMC_ARGR)

SDMMC argument register

偏移地址：0x008

复位值：0x0000 0000

SDMMC_ARGR 寄存器包含一个 32 位命令参数，该参数作为命令消息的一部分发送到卡。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMDARG[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMDARG[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CMDARG[31:0]**：命令参数 (Command argument)

只有在禁止 CPSM (CPSMEN = 0) 时才能通过固件写入这些位。

作为命令消息的一部分发送给卡的命令参数。如果命令包含参数，则在将命令写入到命令寄存器之前，必须将参数加载到此寄存器中。

24.10.4 SDMMC 命令寄存器 (SDMMC_CMDR)

SDMMC command register

偏移地址：0x00C

复位值：0x0000 0000

SDMMC_CMDR 寄存器包含命令索引和命令类型位。命令索引作为命令消息的一部分而发送给卡。命令类型位控制命令路径状态机 (CPSM)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMD SUS PEND
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT EN	BOOT MODE	DT HOLD	CPSM EN	WAITP END	WAIT INT	WAITRESP[1:0]		CMD STOP	CMD TRANS	CMDINDEX[5:0]					
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **CMDSUSPEND**：CPSM 将命令视为挂起或恢复命令并发出中断周期开始/结束信号 (The CPSM treats the command as a Suspend or Resume command and signals interrupt period start/end)

只有在禁止 CPSM (CPSMEN = 0) 时才能通过固件写入该位。

CMDSUSPEND = 1 且 CMDTRANS = 0 时为挂起命令，响应位 BS = 0 时开始中断周期。

CMDSUSPEND = 1 且 CMDTRANS = 1 时为含数据的恢复命令，响应位 DF = 1 时结束中断周期。

位 15 **BOOTEN**：使能启动模式过程 (Enable boot mode procedure)

0：禁止启动模式过程

1：使能启动模式过程

位 14 **BOOTMODE**：选择要使用的启动模式过程 (Select the boot mode procedure to be used)

只有在禁止 CPSM (CPSMEN = 0) 时才能写入该位

0：选择正常启动模式过程

1：选择备用启动模式过程。

位 13 **DTHOLD**：在 DPSM 中暂停新的数据块发送和接收 (Hold new data block transmission and reception in the DPSM)

如果该位置 1，DPSM 不会从 Wait_S 状态变为发送状态或从 Wait_R 状态变为接收状态。

位 12 **CPSMEN**：命令路径状态机 (CPSM) 使能位 (Command path state machine (CPSM) enable bit)

该位由固件写 1，并在 CPSM 进入空闲状态时由硬件清零。

如果该位置 1，则使能 CPSM。

当 DTEN = 1 时，不会传输命令，也不会开始启动过程。CPSMEN 将清零。

在读取等待期间，当 SDMMC_CK 停止时，不会发送任何命令，CPSMEN 保持为 0。

位 11 **WAITPEND**：CPSM 等待来自 DPSM 的数据传输结束（CmdPend 内部信号）(CPSM waits for end of data transfer (CmdPend internal signal) from DPSM)

如果该位置 1，则 CPSM 将等到数据传输结束触发信号后才开始发送命令。

仅当 DTMODE = *e*•MMC 流数据传输、WIDBUS = 1 位宽总线模式、DPSMACT = 1 且 DTDIR = 从主机到卡时，才考虑 WAITPEND。

位 10 **WAITINT**：CPSM 等待中断请求 (CPSM waits for interrupt request)

如果该位置 1，则 CPSM 禁止命令超时并等待卡中断请求（响应）。

如果该位在 CPSM 等待状态下清零，会导致中断模式中止。

位 9:8 **WAITRESP[1:0]**：等待响应位 (Wait for response bits)

只有在禁止 CPSM (CPSMEN = 0) 时才能通过固件写入该位。

这些位用于配置 CPSM 是否等待响应，如果等待，将等待哪种类型的响应。

00：无响应，但 CMDSENT 标志除外

01：短响应，但 CMDREND 或 CCRCFAIL 标志除外

10：短响应，将出现 CMDREND 标志（无 CRC）

11：长响应，但 CMDREND 或 CCRCFAIL 标志除外

位 7 **CMDSTOP**：CPSM 将命令视为停止传输命令并向 DPSM 发出中止信号 (The CPSM treats the command as a Stop Transmission command and signals abort to the DPSM)

只有在禁止 CPSM (CPSMEN = 0) 时才能通过固件写入该位。

如果该位置 1，当发送命令后，CPSM 将向 DPSM 发出中止信号。

位 6 **CMDTRANS**：CPSM 将命令视为数据传输命令、停止中断周期并向 DPSM 发出数据使能信号 (The CPSM treats the command as a data transfer command, stops the interrupt period, and signals DataEnable to the DPSM)

只有在禁止 CPSM (CPSMEN = 0) 时才能通过固件写入该位。

如果该位置 1，当发送命令时，CPSM 将发出中断周期结束信号并向 DPSM 发出数据使能信号。

位 5:0 **CMDINDEX[5:0]**：命令索引

只有在禁止 CPSM (CPSMEN = 0) 时才能通过固件写入该位。

命令索引作为命令消息的一部分发送给卡。

注： *1 此寄存器的两次写访问至少需要相隔七个 sdmmc_hclk 时钟周期。*

2 多媒体卡可以发送两种类型的响应：短响应（48 位）或长响应（136 位）。SD 卡和 SD I/O 卡则只能发送短响应，参数因响应类型而异：软件根据发送的命令区分响应类型。

24.10.5 SDMMC 命令响应寄存器 (SDMMC_RESPCMDR)

SDMMC command response register

偏移地址：0x010

复位值：0x0000 0000

SDMMC_RESPCMDR 寄存器包含接收到的最后一个命令响应的命令索引位字段。如果命令响应传输不包含命令索引位字段（长响应或 OCR 响应），则 RESPCMD 位字段为未知，但该位字段必须包含 111111b（响应中保留位字段的值）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RESPCMD[5:0]					
										r	r	r	r	r	r

位 31:6 保留，必须保持复位值。

位 5:0 **RESPCMD[5:0]**：响应命令索引 (Response command index)
只读位字段。包含接收到的最后一个命令响应的命令索引。

24.10.6 SDMMC 响应 x 寄存器 (SDMMC_RESPxR)

SDMMC response x register

偏移地址：0x010 + 0x004 * x（x = 1 到 4）

复位值：0x0000 0000

SDMMC_RESP1/2/3/4R 寄存器包含卡的状态，该状态来自接收的响应。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CARDSTATUS[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARDSTATUS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CARDSTATUS[31:0]**：下表所示的卡状态
请参见*表 197*。

卡状态为 32 位或 128 位，具体取决于响应类型。

表 197. 响应类型和 SDMMC_RESPxR 寄存器

寄存器(1)	短响应	长响应
SDMMC_RESP1R	卡状态 [31:0]	卡状态 [127:96]
SDMMC_RESP2R	全 0	卡状态 [95:64]
SDMMC_RESP3R	全 0	卡状态 [63:32]
SDMMC_RESP4R	全 0	卡状态 [31:0](2)

1. 首先接收卡状态的最高有效位。
2. SDMMC_RESP4R 寄存器 LSB 始终为 0。

24.10.7 SDMMC 数据定时器寄存器 (SDMMC_DTIMER)

SDMMC data timer register

偏移地址：0x024

复位值：0x0000 0000

SDMMC_DTIMER 寄存器包含以卡总线时钟周期表示的数据超时周期。

计数器加载 SDMMC_DTIMER 寄存器的值，并在数据路径状态机 (DPSM) 进入 Wait_R 或繁忙状态后开始递减。如果定时器达到 0 而 DPSM 处于上述一种状态，则超时状态标志置 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATATIME[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATATIME[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DATATIME[31:0]**：数据和 R1b 繁忙超时周期 (Data and R1b busy timeout period)
只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位。
以卡总线时钟周期表示的数据和 R1b 繁忙超时周期。

注： *数据传输必须首先写入到数据计时器寄存器和数据长度寄存器，然后才写入到数据控制寄存器。*

24.10.8 SDMMC 数据长度寄存器 (SDMMC_DLENR)

SDMMC data length register

偏移地址：0x028

复位值：0x0000 0000

SDMMC_DLENR 寄存器包含要传输的数据字节数。当数据传输开始时，值将加载到数据计数器中。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATALENGTH[24:16]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATALENGTH[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24:0 **DATALENGTH[24:0]**：数据长度值 (Data length value)

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此寄存器。

要传输的数据字节数量。

当 DDR = 1 时，DATALENGTH 被截断为 2 的倍数。（不传输最后一个奇数字节）

当 DATALENGTH = 0 时，不会传输任何数据，当通过 CPSMEN 和 CMDTRANS = 1 请求时，也不会传输任何命令。DTEN 和 CPSMEN 均清零。

注： *对于块数据传输，数据长度寄存器中的值必须是块大小的倍数（请参见 SDMMC_DCTRL）。数据传输必须首先写入到数据计时器寄存器和数据长度寄存器，然后才写入到数据控制寄存器。*

对于 SDMMC 多字节传输，数据长度寄存器中的值必须在 1 到 512 之间。

24.10.9 SDMMC 数据控制寄存器 (SDMMC_DCTRL)

SDMMC data control register

偏移地址：0x02C

复位值：0x0000 0000

SDMMC_DCTRL 寄存器控制数据路径状态机 (DPSM)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	FIFO RST	BOOT ACK EN	SDIO EN	RW MOD	RW STOP	RW START	DBLOCKSIZE[3:0]				DTMODE[1:0]		DTDIR	DTEN
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:14 保留，必须保持复位值。

位 13 **FIFORST**：FIFO 复位，清空任何剩余的数据 (FIFO reset, flushes any remaining data)

只有在 IDMAEN = 0 且未激活 DPSM (DPSMACT = 1) 时才能通过固件写入该位。只有在发生传输错误或传输保持时，该位才会生效。

0：FIFO 不受影响。

1：清空任何剩余的数据并复位 FIFO 指针。当 DPSM 变为非激活状态 (DPSMACT = 0) 时，该位由硬件自动清零。

位 12 **BOOTACKEN**：使能启动确认接收 (Enable the reception of the boot acknowledgment)

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入该位。

0：禁止启动确认，将不会接收

1：使能启动确认，将会接收

位 11 **SDIOEN**：SD I/O 中断使能功能 (SD I/O interrupt enable functions)
只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入该位。
如果将该位置 1，则 DPSM 使能特定于 SD I/O 卡的中断操作。

位 10 **RWMOD**：读取等待模式 (Read Wait mode)
只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入该位。
0：使用 SDMMC_D2 进行读取等待控制
1：通过停止 SDMMC_CK 进行读取等待控制

位 9 **RWSTOP**：读取等待停止 (Read Wait stop)
该位由固件写入，并在 DPSM 从 R_W 状态变为 Wait_R 或空闲状态时由硬件自动清零。
0：无读取等待停止。
1：当 DPSM 处于 R_W 状态时，使能读取等待停止。

位 8 **RWSTART**：读取等待开始 (Read Wait start)
如果将该位置 1，则读取等待操作开始。

位 7:4 **DBLOCKSIZE[3:0]**：数据块大小 (Data block size)
只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入该位。
定义在选择了块数据传输模式时数据块的长度：
0000：块长度 = 2^0 = 1 字节
0001：块长度 = 2^1 = 2 字节
0010：块长度 = 2^2 = 4 字节
0011：块长度 = 2^3 = 8 字节
0100：块长度 = 2^4 = 16 字节
0101：块长度 = 2^5 = 32 字节
0110：块长度 = 2^6 = 64 字节
0111：块长度 = 2^7 = 128 字节
1000：块长度 = 2^8 = 256 字节
1001：块长度 = 2^9 = 512 字节
1010：块长度 = 2^{10} = 1024 字节
1011：块长度 = 2^{11} = 2048 字节
1100：块长度 = 2^{12} = 4096 字节
1101：块长度 = 2^{13} = 8192 字节
1110：块长度 = 2^{14} = 16384 字节
1111：保留
当 DATALENGTH 不是 DBLOCKSIZE 的倍数时，传输的数据将截断为 DBLOCKSIZE 的倍数。（不会传输任何剩余数据。）
当 DDR = 1 时，不得使用 DBLOCKSIZE = 0000。（不会传输任何数据）

位 3:2 **DTMODE[1:0]**：数据传输模式选择 (Data transfer mode selection)
只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入该位。
00：按块数结束的块数据传输。
01：SDIO 多字节数据传输。
10：*e*•MMC 流数据传输。（WIDBUS 必须选择 1 位宽总线模式）
11：以 STOP_TRANSMISSION 命令结束的块数据传输（不与 DTEN 启动的数据传输一起使用）。

位 1 **DTDIR**：数据传输方向选择 (Data transfer direction selection)
只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入该位。
0：从主机到卡。
1：从卡到主机。

位 0 **DTEN**：数据传输使能位 (Data transfer enable bit)

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入该位。当数据传输完成时，该位由硬件清零。

该位必须仅在不使用相关数据传输命令时才用于传输数据，即不能与 SD 或 *e*•MMC 卡一起使用。

0：不使用 CPSM 数据传输命令时不启动数据传输。

1：不使用 CPSM 数据传输命令时启动数据传输。

24.10.10 SDMMC 数据计数器寄存器 (SDMMC_DCNTR)

SDMMC data counter register

偏移地址：0x030

复位值：0x0000 0000

当 DPSM 从空闲状态变为 Wait_R 或 Wait_S 状态时，SDMMC_DCNTR 寄存器将从数据长度寄存器中加载值（请参见 SDMMC_DLENR）。在传输数据时，计数器将值递减直至计数器达到 0。然后 DPSM 将变为空闲状态，并且在无错误时且没有发送数据传输保持的情况下，将数据状态结束标志 (DATAEND) 置 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATACOUNT[24:16]								
							r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATACOUNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:25 保留，必须保持复位值。

位 24:0 **DATACOUNT[24:0]**：数据计数值 (Data count value)

读取时，将返回要传输的剩余数据字节数。写入没有任何效果。

注： *必须仅当数据传输完成或暂停后才读取该寄存器。在错误事件后读取时，读取的数据计数值可能与传输的实际数据字节数不同。*

24.10.11 SDMMC 状态寄存器 (SDMMC_STAR)

SDMMC status register

偏移地址：0x034

复位值：0x0000 0000

SDMMC_STAR 寄存器是一个只读寄存器。它包含两种类型的标志：

- 静态标志（位 [28, 21, 11:0]）：在通过写入到 SDMMC 中断清零寄存器来清零这些位之前，会一直保持这些位（请参见 SDMMC_ICR）
- 动态标志（位 [20:12]）：这些位根据底层逻辑的状态来更改状态（例如，随着数据写入到 FIFO，发出和停止发出 FIFO 满和空标志）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	IDMABTC	IDMATE	CKSTOP	VSWEND	ACKTIMEOUT	ACKFAIL	SDIOIT	BUSYD0END	BUSYD0	RXFIFOE	TXFIFOE	RXFIFOF	TXFIFOF
			r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXFIFOHF	TXFIFOHE	CPSMACT	DPSMACT	DABORT	DBCKEND	DHOLD	DATAEND	CMDSENT	CMDREND	RXOVERR	TXUNDERR	DTIMEOUT	CTIMEOUT	DCRCFAIL	CCRCFAIL
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:29 保留，必须保持复位值。

位 28 **IDMABTC**：IDMA 缓冲区传输完成

通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 27 **IDMATE**：IDMA 传输错误

通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 26 **CKSTOP**：SDMMC_CK 在电压切换过程中停止 (SDMMC_CK stopped in Voltage switch procedure)

通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 25 **VSWEND**：电压切换时序关键部分完成 (Voltage switch critical timing section completion)

通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 24 **ACKTIMEOUT**：启动确认超时

通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 23 **ACKFAIL**：接收到启动确认（启动确认校验失败）(Boot acknowledgment received (boot acknowledgment check fail))

通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 22 **SDIOIT**：收到了 SDIO 中断 (SDIO interrupt received)

通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 21 **BUSYD0END**：检测到 CMD 响应后 SDMMC_D0 繁忙状态结束 (end of SDMMC_D0 Busy following a CMD response detected)

该位仅指示 CMD 响应后繁忙状态结束，该位不会指示因数据传输引起的繁忙状态。通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

0：卡 SDMMC_D0 信号未从繁忙状态变为不忙状态。

1：卡 SDMMC_D0 信号从繁忙状态变为不忙状态。

位 20 **BUSYD0**：SDMMC_D0 线的反转值（繁忙）(Inverted value of SDMMC_D0 line (Busy))，在 CMD 响应结束时进行采样，并在 CMD 响应后的 2 个 SDMMC_CK 周期后进行二次采样

当 SDMMCD0 线从繁忙状态变为不忙状态时，该位复位为不忙状态。该位不会指示因数据传输引起的繁忙状态。它只是一个硬件状态标志，不会生成中断。

0：卡在 SDMMC_D0 上发出不忙信号。

1：卡在 SDMMC_D0 上发出繁忙信号。

位 19 **RXFIFOE**：接收 FIFO 为空

它只是一个硬件状态标志，不会生成中断。该位在一个 FIFO 单元已满时清零。

位 18 **TXFIFOE**：发送 FIFO 为空

该位在一个 FIFO 单元已满时清零。

位 17 **RXFIFOF**：接收 FIFO 已满

该位在一个 FIFO 单元为空时清零。

位 16 **TXFIFOF**：发送 FIFO 已满 (Transmit FIFO full)
它只是一个硬件状态标志，不会生成中断。该位在一个 FIFO 单元为空时清零。

位 15 **RXFIFOHF**：接收 FIFO 半满
FIFO 中至少有一半字数。该位在 FIFO 为“半 + 1”空时清零。

位 14 **TXFIFOHE**：发送 FIFO 半空
FIFO 中至少可写入一半字数。该位在 FIFO 为“半 + 1”满时清零。

位 13 **CPSMACT**：命令路径状态机激活，即不处于空闲状态 (Command path state machine active, i.e. not in Idle state)
它只是一个硬件状态标志，不会生成中断。

位 12 **DPSMACT**：数据路径状态机激活，即不处于空闲状态 (Data path state machine active, i.e. not in Idle state)
它只是一个硬件状态标志，不会生成中断。

位 11 **DABORT**：数据传输由 CMD12 中止 (Data transfer aborted by CMD12)
通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 10 **DBCKEND**：已发送/接收数据块 (Data block sent/received)
在以下情况下，DBCKEND 置 1：
- CRC 校验已通过且 DPSM 变为 R_W 状态
或
- IDMAEN = 0、发送数据传输保持、DATACOUNT >0 且 DPSM 变为 Wait_S 状态。
通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 9 **DHOLD**：数据传输保持 (Data transfer Hold)
通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 8 **DATAEND**：数据传输正确结束 (Data transfer ended correctly)
如果数据计数器 DATACOUNT 为零，未出现错误且没有发送数据传输保持，则 DATAEND 置 1。
通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 7 **CMDSENT**：命令已发送（不需要响应）(Command sent (no response required))
通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 6 **CMDREND**：已接收命令响应（CRC 校验通过或无 CRC）(Command response received (CRC check passed, or no CRC))
通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 5 **RXOVERR**：收到了 FIFO 上溢错误 (Received FIFO overrun error)（IDMA 使能时由硬件屏蔽）
通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 4 **TXUNDERR**：发送 FIFO 下溢错误 (Transmit FIFO underrun error)（IDMA 使能时由硬件屏蔽）
通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 3 **DTIMEOUT**：数据超时
通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 2 **CTIMEOUT**：命令响应超时
通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。
命令超时周期为固定值 64 个 SDMMC_CK 时钟周期。

位 1 **DCRCFAIL**：已发送/接收数据块（CRC 校验失败）(Data block sent/received (CRC check failed))
通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

位 0 **CCRCFAIL**：已接收命令响应（CRC 校验失败）(Command response received (CRC check failed))

通过写入 SDMMC_ICR 中的相应中断清零位来清零中断标志。

注： *使用 IDMA 模式时，SDMMC_MASKR 中必须屏蔽 FIFO 中断标志。*

24.10.12 SDMMC 中断清零寄存器 (SDMMC_ICR)

SDMMC interrupt clear register

偏移地址：0x038

复位值：0x0000 0000

SDMMC_ICR 寄存器是一个只写寄存器。向某个位写入 1 会将 SDMMC_STAR 状态寄存器中的对应位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	IDMA BTCC	IDMA TEC	CK STOPC	VSW ENDC	ACK TIME OUTC	ACK FAILC	SDIO ITC	BUSY D0 ENDC	Res.	Res.	Res.	Res.	Res.
			rw	rw	rw	rw	rw	rw	rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	D ABORT C	DBCK ENDC	DHOLD C	DATA ENDC	CMD SENTC	CMDR ENDC	RX OVERR C	TX UNDER RC	D TIME OUTC	C TIME OUTC	DCRC FAILC	CCRC FAILC
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28 **IDMABTCC**：IDMA 缓冲区传输完成清零位 (IDMA buffer transfer complete clear bit)

由软件置 1，用于将 IDMABTC 标志清零。

0：未将 IDMABTC 清零

1：已将 IDMABTC 清零

位 27 **IDMATEC**：IDMA 传输错误清零位 (IDMA transfer error clear bit)

由软件置 1，用于将 IDMATE 标志清零。

0：未将 IDMATE 清零

1：已将 IDMATE 清零

位 26 **CKSTOPC**：CKSTOP 标志清零位 (CKSTOP flag clear bit)

由软件置 1，用于将 CKSTOP 标志清零。

0：未将 CKSTOP 清零

1：已将 CKSTOP 清零

位 25 **VSWENDC**：VSWEND 标志清零位 (VSWEND flag clear bit)

由软件置 1，用于将 VSWEND 标志清零。

0：未将 VSWEND 清零

1：已将 VSWEND 清零

位 24 **ACKTIMEOUTC**：ACKTIMEOUT 标志清零位 (ACKTIMEOUT flag clear bit)

由软件置 1，用于将 ACKTIMEOUT 标志清零。

0：未将 ACKTIMEOUT 清零

1：已将 ACKTIMEOUT 清零

位 23 **ACKFAILC**：ACKFAIL 标志清零位 (ACKFAIL flag clear bit)
由软件置 1，用于将 ACKFAIL 标志清零。
0：未将 ACKFAIL 清零
1：已将 ACKFAIL 清零

位 22 **SDIOITC**：SDIOIT 标志清零位 (SDIOIT flag clear bit)
由软件置 1，用于将 SDIOIT 标志清零。
0：未将 SDIOIT 清零
1：已将 SDIOIT 清零

位 21 **BUSYD0ENDC**：BUSYD0END 标志清零位 (BUSYD0END flag clear bit)
由软件置 1，用于将 BUSYD0END 标志清零。
0：未将 BUSYD0END 清零
1：已将 BUSYD0END 清零

位 20:12 保留，必须保持复位值。

位 11 **DABORTC**：DABORT 标志清零位 (DABORT flag clear bit)
由软件置 1，用于将 DABORT 标志清零。
0：未将 DABORT 清零
1：已将 DABORT 清零

位 10 **DBCKENDC**：DBCKEND 标志清零位 (DBCKEND flag clear bit)
由软件置 1，用于将 DBCKEND 标志清零。
0：未将 DBCKEND 清零
1：已将 DBCKEND 清零

位 9 **DHOLDC**：DHOLD 标志清零位 (DHOLD flag clear bit)
由软件置 1，用于将 DHOLD 标志清零。
0：未将 DHOLD 清零
1：已将 DHOLD 清零

位 8 **DATAENDC**：DATAEND 标志清零位 (DATAEND flag clear bit)
由软件置 1，用于将 DATAEND 标志清零。
0：未将 DATAEND 清零
1：已将 DATAEND 清零

位 7 **CMDSENTC**：CMDSENT 标志清零位 (CMDSENT flag clear bit)
由软件置 1，用于将 CMDSENT 标志清零。
0：未将 CMDSENT 清零
1：已将 CMDSENT 清零

位 6 **CMDRENDC**：CMDREND 标志清零位 (CMDREND flag clear bit)
由软件置 1，用于将 CMDREND 标志清零。
0：未将 CMDREND 清零
1：已将 CMDREND 清零

位 5 **RXOVERRC**：RXOVERR 标志清零位 (RXOVERR flag clear bit)
由软件置 1，用于将 RXOVERR 标志清零。
0：未将 RXOVERR 清零
1：已将 RXOVERR 清零

位 4 **TXUNDERRC**：TXUNDERR 标志清零位 (TXUNDERR flag clear bit)
由软件置 1，用于将 TXUNDERR 标志清零。
0：未将 TXUNDERR 清零
1：已将 TXUNDERR 清零

位 3 **DTIMEOUTC**：DTIMEOUT 标志清零位 (DTIMEOUT flag clear bit)
由软件置 1，用于将 DTIMEOUT 标志清零。
0：未将 DTIMEOUT 清零
1：已将 DTIMEOUT 清零

位 2 **CTIMEOUTC**：CTIMEOUT 标志清零位 (CTIMEOUT flag clear bit)
由软件置 1，用于将 CTIMEOUT 标志清零。
0：未将 CTIMEOUT 清零
1：已将 CTIMEOUT 清零

位 1 **DCRCFAILC**：DCRCFAIL 标志清零位 (DCRCFAIL flag clear bit)
由软件置 1，用于将 DCRCFAIL 标志清零。
0：未将 DCRCFAIL 清零
1：已将 DCRCFAIL 清零

位 0 **CCRCFAILC**：CCRCFAIL 标志清零位 (CCRCFAIL flag clear bit)
由软件置 1，用于将 CCRCFAIL 标志清零。
0：未将 CCRCFAIL 清零
1：已将 CCRCFAIL 清零

24.10.13 SDMMC 屏蔽寄存器 (SDMMC_MASKR)

SDMMC mask register

偏移地址：0x03C

复位值：0x0000 0000

中断屏蔽寄存器通过将对应的位置 1 来确定哪一个状态标志位可以产生中断请求。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	IDMA BTCIE	Res.	CK STOP IE	VSW ENDIE	ACK TIME OUTIE	ACK FAILIE	SDIO ITIE	BUSY D0 ENDIE	Res.	Res.	TX FIFO EIE	RX FIFO FIE	Res.
			rw		rw	rw	rw	rw	rw	rw			rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX FIFO HFIE	TX FIFO HEIE	Res.	Res.	DA BORT IE	DBCK ENDIE	DHOLD IE	DATA ENDIE	CMD SENT IE	CMDR ENDIE	RX OVER RIE	TX UNDER RIE	D TIME OUTIE	C TIME OUTIE	DCRC FAILIE	CCRC FAILIE
rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28 **IDMABTCIE**：IDMA 缓冲区传输完成中断使能 (IDMA buffer transfer complete interrupt enable)
由软件置 1 和清零，用于使能/禁止 IDMA 传输完内存缓冲区内的所有数据时生成的中断。
0：禁止 IDMA 缓冲区传输完成中断
1：使能 IDMA 缓冲区传输完成中断

位 27 保留，必须保持复位值。

位 26 **CKSTOPIE**：电压切换时钟停止中断使能 (Voltage Switch clock stopped interrupt enable)
由软件置 1 和清零，用于使能/禁止由电压切换时钟停止引起的中断。
0：禁止电压切换时钟停止中断
1：使能电压切换时钟停止中断

位 25 **VSWENDIE**：电压切换时序关键部分完成中断使能 (Voltage switch critical timing section completion interrupt enable)

由软件置 1 和清零，用于使能/禁止电压切换时序关键部分完成时生成的中断。

0：禁止电压切换时序关键部分完成中断

1：使能电压切换时序关键部分完成中断

位 24 **ACKTIMEOUTIE**：确认超时中断使能 (Acknowledgment timeout interrupt enable)

由软件置 1 和清零，用于使能/禁止由确认超时引起的中断。

0：禁止确认超时中断

1：使能确认超时中断

位 23 **ACKFAILIE**：确认失败中断使能 (Acknowledgment Fail interrupt enable)

由软件置 1 和清零，用于使能/禁止由确认失败引起的中断。

0：禁止确认失败中断

1：使能确认失败中断

位 22 **SDIOITIE**：接收到 SDIO 模式中断时中断使能 (SDIO mode interrupt received interrupt enable)

由软件置 1 和清零，用于使能/禁止收到 SDIO 模式中断时生成中断。

0：禁止接收到 SDIO 模式中断时中断

1：使能接收到 SDIO 模式中断时中断

位 21 **BUSYD0ENDIE**：BUSYD0END 中断使能 (BUSYD0END interrupt enable)

由软件置 1 和清零，用于使能/禁止 CMD 响应后 SDMMC_D0 信号从繁忙状态变为不忙状态时生成的中断。

0：禁止 BUSYD0END 中断

1：使能 BUSYD0END 中断

位 20:19 保留，必须保持复位值。

位 18 **TXFIFOEIE**：Tx FIFO 为空时中断使能 (Tx FIFO empty interrupt enable)

由软件置 1 和清零，用于使能/禁止由 Tx FIFO 为空引起的中断。

0：禁止 Tx FIFO 为空时中断

1：使能 Tx FIFO 为空时中断

位 17 **RXFIFOFIE**：Rx FIFO 变满时中断使能 (Rx FIFO full interrupt enable)

由软件置 1 和清零，用于使能/禁止由 Rx FIFO 变满引起的中断。

0：禁止 Rx FIFO 变满时中断

1：使能 Rx FIFO 变满时中断

位 16 保留，必须保持复位值。

位 15 **RXFIFOHFIE**：Rx FIFO 半满时中断使能 (Rx half full interrupt enable)

由软件置 1 和清零，用于使能/禁止由 Rx FIFO 半满引起的中断。

0：禁止 Rx FIFO 半满时中断

1：使能 Rx FIFO 半满时中断

位 14 **TXFIFOHEIE**：Tx FIFO 半空时中断使能 (Tx FIFO half empty interrupt enable)

由软件置 1 和清零，用于使能/禁止由 Tx FIFO 半空引起的中断。

0：禁止 Tx FIFO 半空时中断

1：使能 Tx FIFO 半空时中断

位 13:12 保留，必须保持复位值。

位 11 **DABORTIE**：数据传输中止中断使能 (Data transfer aborted interrupt enable)

由软件置 1 和清零，用于使能/禁止由数据传输中止引起的中断。

0：禁止数据传输中止中断

1：使能数据传输中止中断

位 10 **DBCKENDIE**：数据块结束中断使能 (Data block end interrupt enable)
由软件置 1 和清零，用于使能/禁止由数据块结束引起的中断。
0：禁止数据块结束中断
1：使能数据块结束中断

位 9 **DHOLDIE**：数据保持中断使能 (Data hold interrupt enable)
由软件置 1 和清零，用于使能/禁止发送过程中新数据保持 DPSM Wait_S 状态时生成的中断。
0：禁止数据保持中断
1：使能数据保持中断

位 8 **DATAENDIE**：数据结束中断使能 (Data end interrupt enable)
由软件置 1 和清零，用于使能/禁止由数据结束引起的中断。
0：禁止数据结束中断
1：使能数据结束中断

位 7 **CMDSENTIE**：命令发送中断使能 (Command sent interrupt enable)
由软件置 1 和清零，用于使能/禁止由发送命令引起的中断。
0：禁止命令发送中断
1：使能命令发送中断

位 6 **CMDRENDIE**：命令响应接收中断使能 (Command response received interrupt enable)
由软件置 1 和清零，用于使能/禁止由接收命令响应引起的中断。
0：禁止命令响应接收中断
1：使能命令响应接收中断

位 5 **RXOVERRIE**：Rx FIFO 上溢错误中断使能 (Rx FIFO overrun error interrupt enable)
由软件置 1 和清零，用于使能/禁止由 Rx FIFO 上溢错误引起的中断。
0：禁止 Rx FIFO 上溢错误中断
1：使能 Rx FIFO 上溢错误中断

位 4 **TXUNDERRIE**：Tx FIFO 下溢错误中断使能 (Tx FIFO underrun error interrupt enable)
由软件置 1 和清零，用于使能/禁止由 Tx FIFO 下溢错误引起的中断。
0：禁止 Tx FIFO 下溢错误中断
1：使能 Tx FIFO 下溢错误中断

位 3 **DTIMEOUTIE**：数据超时中断使能 (Data timeout interrupt enable)
由软件置 1 和清零，用于使能/禁止由数据超时引起的中断。
0：禁止数据超时中断
1：使能数据超时中断

位 2 **CTIMEOUTIE**：命令超时中断使能 (Command timeout interrupt enable)
由软件置 1 和清零，用于使能/禁止由命令超时引起的中断。
0：禁止命令超时中断
1：使能命令超时中断

位 1 **DCRCFAILIE**：数据 CRC 失败中断使能 (Data CRC fail interrupt enable)
由软件置 1 和清零，用于使能/禁止由数据 CRC 失败引起的中断。
0：禁止数据 CRC 失败中断
1：使能数据 CRC 失败中断

位 0 **CCRCFAILIE**：命令 CRC 失败中断使能 (Command CRC fail interrupt enable)
由软件置 1 和清零，用于使能/禁止由命令 CRC 失败引起的中断。
0：禁止命令 CRC 失败中断
1：使能命令 CRC 失败中断

24.10.14 SDMMC 确认定时器寄存器 (SDMMC_ACKTIMER)

SDMMC acknowledgment timer register

偏移地址：0x040

复位值：0x0000 0000

SDMMC_ACKTIMER 寄存器包含以 SDMMC_CK 总线时钟周期表示的确认超时周期。

计数器加载 SDMMC_ACKTIMER 寄存器的值，并在数据路径状态机 (DPSM) 进入 Wait_Ack 状态后开始递减。如果定时器达到 0 而 DPSM 处于上述状态，则确认超时状态标志置 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACKTIME[24:16]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACKTIME[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24:0 **ACKTIME[24:0]**：启动确认超时周期 (Boot acknowledgment timeout period)
只有在禁止 CPSM (CPSMEN = 0) 时才能通过固件写入该位。
以卡总线时钟周期表示的启动确认超时周期。

注： *数据传输必须首先写入到确认定时器寄存器，然后才写入到数据控制寄存器。*

24.10.15 SDMMC 数据 FIFO 寄存器 x (SDMMC_FIFORx)

SDMMC data FIFO registers x

偏移地址：0x080 + 0x004 * x（x = 0 到 15）

复位值：0x0000 0000

接收和发送 FIFO 只能被当作 1 字（32 位）宽的寄存器来读出或写入。FIFO 在连续地址上包含 16 个入口。这使 CPU 可以使用其加载 (load) 和存储 (store) 多操作符命令来读写 FIFO。FIFO 寄存器接口负责正确对齐 FIFO 内部的数据，CPU 使用的 FIFO 寄存器地址很重要。

当通过半字或字节访问的方式访问 SDMMC_FIFOR 时，会生成 AHB 总线故障。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIFODATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFODATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **FIFODATA[31:0]**：接收和发送 FIFO 数据 (Receive and transmit FIFO data)
只有在激活 DPSM (DPSMACT = 1) 时才能通过固件读取或写入此寄存器。
FIFO 数据占用 32 位字的 16 个入口。

24.10.16 SDMMC DMA 控制寄存器 (SDMMC_IDMACTRLR)

SDMMC DMA control register

偏移地址：0x050

复位值：0x0000 0000

接收和发送 FIFO 可以被当做 32 位宽的寄存器来读出或写入。FIFO 在 32 个连续地址上包含 32 个入口。这使 CPU 可以使用其加载 (load) 和存储 (store) 多操作符命令来读写 FIFO。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDMABMODE	IDMAEN
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **IDMABMODE**：缓冲区模式选择 (Buffer mode selection)
只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入该位。
0：单缓冲区模式。
1：链表模式。

位 0 **IDMAEN**：IDMA 使能 (IDMA enable)
只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入该位。
0：禁止 IDMA
1：使能 IDMA

24.10.17 SDMMC IDMA 缓冲区大小寄存器 (SDMMC_IDMABSIZER)

SDMMC IDMA buffer size register

偏移地址：0x054

复位值：0x0000 0000

SDMMC_IDMABSIZER 寄存器包含链表配置下的缓冲区大小。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDMABNDT[11]
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDMABNDT[10:0]											Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw					

位 31:17 保留，必须保持复位值。

位 16:5 **IDMABNDT[11:0]**：每个缓冲区的字节数 (Number of bytes per buffer)

该 12 位值必须乘以 8（可得到用 32 位字表示的缓冲区大小）和 32（可得到用字节表示的缓冲区大小）。

示例：IDMABNDT = 0x001：缓冲区大小 = 8 字 = 32 字节。

示例：IDMABNDT = 0x800：缓冲区大小 = 16384 字 = 64 KB。

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入这些位。

位 4:0 保留，必须保持复位值。

24.10.18 SDMMC IDMA 缓冲区基址寄存器 (SDMMC_IDMABASER)

SDMMC IDMA buffer base address register

偏移地址：0x058

复位值：0x0000 0000

SDMMC_IDMABASER 寄存器包含单缓冲区配置和链表配置下的内存缓冲区基址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDMABASE[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDMABASE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r

位 31:0 **IDMABASE[31:0]**：缓冲区内存基址位 [31:2]，必须按字对齐（位 [1:0] 始终为 0 且为只读位）

当 DPSM 未激活 (DPSMACT = 0) 时，可通过固件写入该寄存器，当 DPSM 激活 (DPSMACT = 1) 时，可通过固件动态写入该寄存器。

24.10.19 SDMMC IDMA 链表地址寄存器 (SDMMC_IDMALAR)

SDMMC IDMA linked list address register

偏移地址：0x064

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ULA	ULS	ABR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDMALA[13:0]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31 **ULA**：在链表模式下（SDMMC_IDMACTRLR.IDMABMODE 选择链表模式）从链表中更新 SDMMC_IDMALAR

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入该位。

0：SDMMC_IDMALAR 不会更新，保持上一个链表项。

1：SDMMC_IDMALAR 将从链表更新。

位 30 **ULS**：在链表模式（SDMMC_IDMACTRLR.IDMABMODE 选择链表模式且 ULA = 1）下从下一个链表中更新 SDMMC_IDMABSIZE

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入该位。

0：SDMMC_IDMABSIZER 将不会从下一个链表更新。

1：SDMMC_IDMABSIZER 将从下一个链表更新。

位 29 **ABR**：确认链表缓冲区就绪 (Acknowledge linked list buffer ready)

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入该位。

当第一个链表缓冲区以软件编程的寄存器信息开头时，不考虑该位。仅在随后加载的链表项中考虑 ABR。

0：加载的链表缓冲区未就绪（这会导致生成链表 IDMA 传输错误）。

1：加载的链表缓冲区就绪确认。链表缓冲区数据由 IDMA 传输。

位 28:16 保留，必须保持复位值。

位 15:2 **IDMALA[13:0]**：字对齐链表项偏移地址 (Word aligned linked list item address offset)

指向下一个链表项结构基址的链表项偏移指针。

链表项基址为 IDMABA + IDMALA。

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入这些位。

位 1:0 保留，必须保持复位值。

24.10.20 SDMMC IDMA 链表内存基址寄存器 (SDMMC_IDMABAR)

SDMMC IDMA linked list memory base register

偏移地址：0x068

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDMABA[29:14]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDMABA[13:0]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31:2 **IDMABA[29:0]**：字对齐链表内存基址 (Word aligned Linked list memory base address)

链表内存基址指针。

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入这些位。

位 1:0 保留，必须保持复位值。

24.10.21 SDMMC 版本寄存器 (SDMMC_VERR)

SDMMC version register

偏移地址：0x3F4

复位值：0x0000 0022

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：SDMMC 主版本号

位 3:0 **MINREV[3:0]**：SDMMC 次版本号

24.10.22 SDMMC 标识寄存器 (SDMMC_IPIDR)

SDMMC identification register

偏移地址：0x3F8

复位值：0x0014 0022

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IP_ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP_ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **IP_ID[31:0]**：SDMMC 标识

24.10.23 SDMMC 大小标识寄存器 (SDMMC_SIDR)

SDMMC size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：SDMMC 大小标识

24.10.24 SDMMC 寄存器映射

表 198. SDMMC 寄存器映射

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	SDMMC_ POWER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIRPOL	VSWITCHEN	VSWITCH	PWRCTRL[1:0]	
	Reset value																												0	0	0	0	0
0x004	SDMMC_ CLKCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELCLKRX[1:0]		BUSSPEED	DDR	HWFC_EN	NEGEDGE	WIDBUS[1:0]		Res.	PWRSAV	Res.	Res.	CLKDIV[9:0]									
	Reset value											0	0	0	0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0
0x008	SDMMC_ ARGR	CMDARG[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	SDMMC_ CMDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDSUSPEND	BOOTEN	BOOTMODE	DTHOLD	CPSMEN	WAITPEND	WAITINT	WAITRESP[1:0]		CMDSTOP	CMDTRANS	CMDINDEX[5:0]					
	Reset value																																
0x010	SDMMC_ RESPCMDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RESPCMD[5:0]					
	Reset value																											0	0	0	0	0	0
0x014	SDMMC_ RESP1R	CARDSTATUS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	SDMMC_ RESP2R	CARDSTATUS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x01C	SDMMC_ RESP3R	CARDSTATUS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	SDMMC_ RESP4R	CARDSTATUS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	SDMMC_ DTIMER	DATATIME[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x028	SDMMC_ DLENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATALENGTH[24:0]																								
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x02C	SDMMC_ DCTRLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FIFORST	BOOTACKEN	SDIOEN	RWMOD	RWSTOP	RWSTART	DBLOCK SIZE[3:0]			DTMODE[1:0]		DTDIR	DTEN	
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 198. SDMMC 寄存器映射（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x030	SDMMC_ DCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATACOUNT[24:0]																								
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x034	SDMMC_ STAR	Res.	Res.	Res.	IDMABTC	IDMATE	CKSTOP	VSWEND	ACKTIMEOUT	ACKFAIL	SDIOIT	BUSYD0END	BUSYD0	RXFIFOE	TXFIFOE	RXFIFOF	TXFIFOF	RXFIFOHF	TXFIFOHE	CPSMACT	DPSMACT	DABORT	DBCKEND	DHOLD	DATAEND	CMDSENT	CMDREND	RXOVERR	TXUNDERR	DTIMEOUT	CTIMEOUT	DCRCFAIL	CCRCFAIL
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x038	SDMMC_ ICR	Res.	Res.	Res.	IDMABTCC	IDMATEC	CKSTOPC	VSWENDC	ACKTIMEOUTC	ACKFAILC	SDIOITC	BUSYD0ENDC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DABORTC	DBCKENDC	DHOLDC	DATAENDC	CMDSENTC	CMDRENDC	RXOVERRC	TXUNDERRC	DTIMEOUTC	CTIMEOUTC	DCRCFAILC	CCRCFAILC
	Reset value				0	0	0	0	0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0
0x03C	SDMMC_ MASKR	Res.	Res.	Res.	IDMABTCIE	Res.	CKSTOPIE	VSWENDIE	ACKTIMEOUTIE	ACKFAILIE	SDIOITIE	BUSYD0ENDIE	Res.	Res.	TXFIFOEIE	RXFIFOFIE	Res.	RXFIFOHFIE	TXFIFOHEIE	Res.	Res.	DABORTIE	DBCKENDIE	DHOLDIE	DATAENDIE	CMDSENTIE	CMDRENDIE	RXOVERRIE	TXUNDERRIE	DTIMEOUTIE	CTIMEOUTIE	DCRCFAILIE	CCRCFAILIE
	Reset value				0		0	0	0	0	0	0			0	0		0	0			0	0	0	0	0	0	0	0	0	0	0	0
0x040	SDMMC_ ACKTIMER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACKTIME[24:0]																								
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x044 - 0x04C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x050	SDMMC_ IDMACTRLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDMABMODE	IDMAEN
	Reset value																															0	0
0x054	SDMMC_ IDMABSIZER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDMABNDT[11:0]												Res.	Res.	Res.	Res.	Res.
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0					
0x058	SDMMC_ IDMABASER	IDMABASE[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x05C - 0x060	Reserved	Res.																															
0x064	SDMMC_ IDMALAR	ULA	ULS	ABR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDMALA[13:0]														Res.	Res.
	Reset value	0	0	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x068	SDMMC_ IDMABAR	IDMABA[29:0]																														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x06C - 0x07C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x080 + 0x04 * x, (x=0..15)	SDMMC_ FIFOR	FIFODATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 198. SDMMC 寄存器映射（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0C0 - 0x3F0	Reserved	Res.																															
0x3F4	SDMMC_ VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	0	1	0	0	0	1	0
0x3F8	SDMMC_ IPIDR	IP_ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
0x3FC	SDMMC_ SIDR	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

25 延迟模块 (DLYB)

25.1 简介

延迟模块 (DLYB) 用于生成和输入时钟不同相位的输出时钟，输出时钟的相位必须通过用户应用程序进行编程。然后，输出时钟用来对由另一个外设（例如 SDMMC 或者四线 SPI 接口）接收到的数据进行时钟控制。

该延迟与电压和温度相关，可能需要应用程序根据接收到的数据重新配置并重新对齐输出时钟的相位。

25.2 DLYB 主要特性

延迟模块具有以下特性：

- 输入时钟频率从 25 MHz 到通信接口支持的最大频率（请参见数据手册）
- 多达 12 个过采样相位

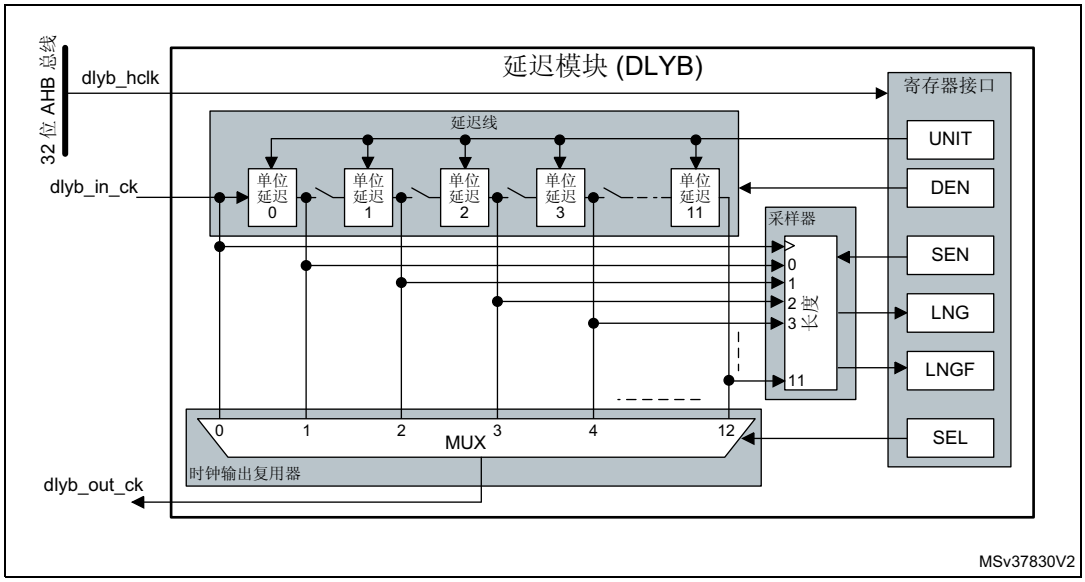
25.3 DLYB 功能描述

25.3.1 DLYB 框图

延迟模块包括以下子模块（如下图所示）：

- 寄存器接口模块，提供对 DLYB 寄存器的 AHB 访问
- 延迟线，用来支持单位延迟
- 延迟线长度采样
- 输出时钟选择多路复用器

图 194. DLYB 框图



25.3.2 DLYB 引脚和内部信号

表 199 列出了 DLYB 内部信号。

表 199. DLYB 内部输入/输出信号

信号名称	信号类型	说明
dlyb_hclk	数字输入	延迟模块寄存器接口时钟
dlyb_in_ck	数字输入	延迟模块输入时钟
dlyb_out_ck	数字输出	延迟模块输出时钟

表 200. DLYB 互连

信号名称	源/目标		
	SDMMC1	SDMMC2	QUADSPI
dlyb_in_ck	sdmmc_io_in_ck		qspi_clk_o 来自 IO 焊盘
dlyb_out_ck	sdmmc_fb_ck		qspi_clk_fb_i

25.3.3 一般说明

可通过将 DLYB 控制寄存器 (DLYB_CR) 中的 DEN 位置 1 来使能延迟模块。可通过 DLYB_CR 寄存器 SEN 位来使能长度采样器。

当使能延迟模块时，由单位延迟增加的延迟是由 DLYB 配置寄存器 (DLYB_CFGR) 中的 UNIT[6:0] 字段来定义的。

注： *只有在禁止输出时钟 (SEN = 1) 时，才能对 UNIT[6:0] 进行编程。*

当使能延迟模块时，可通过 DLYB_CFGR 寄存器中的 SEL[3:0] 位字段来选择输出时钟相位。

注： *只有在禁止输出时钟 (SEN = 1) 时，才能对 SEL 进行编程。*

通过将延迟线长度配置为涵盖一个周期，可以在一个输入时钟周期内对输出时钟进行相移。延迟线长度可以通过 SEN 位启用长度采样器来配置，这样可以访问 DLYB_CFGR 中的延迟线长度 (LNG[11:0]) 和长度有效标志 (LNGF)。

如果需要短于一个输入时钟周期的输出时钟延迟，可以减小延迟线长度。这样可以实现更小的单位延迟，从而提供更高的分辨率。

一旦完成延迟线长度配置，就可以通过输出时钟复用器选择经过移相的输出时钟。此操作通过 SEL[3:0] 完成。当 SEN 设置为 0 时，只有选定相位的输出时钟可供使用。

下表对延迟模块控制进行了总结。

表 201. 延迟模块控制

DEN	SEN	UNIT	SEL	LNG	LNGF	输出时钟
0	0	无关	无关	无关	无关	使能（= 输入时钟）
x	1	单位延迟	输出时钟相位	长度	长度标志	禁止
1	0	单位延迟(1)	输出时钟相位(2)	无关	无关	使能（= 选定相位的输出时钟）

1. 只有当 SEN = 1 时，才可以更改单位延迟。
2. 只有当 SEN = 1 时，才可以更改输出时钟相位。

25.3.4 延迟线长度配置程序

LNG[11:0] 用于确定延迟线长度，其与输入时钟周期相关。配置的延迟线长度必须覆盖一个完整的输入时钟周期。

请注意，尽管延迟线有 12 个单位延迟单元，但以下过程描述返回的长度在 0 到 10 之间，因为使用了延迟输出值上限来确保在一个完整的输入时钟周期内校准延迟。对于不同的时钟频率和 UNIT 值，还可以从时钟周期长度截断为单位延迟单元 10。

在整个调谐过程中，必须存在时钟输入（自由运行时钟）。

要将延迟线长度配置为一个输入时钟周期，请按照以下顺序进行：

1. 通过将 DEN 位置 1 来使能延迟模块。
2. 通过将 SEN 位置 1 来使能长度采样。
3. 通过将 SEL[3:0] 设置为 12 来使能所有延迟单元。
4. 对于 UNIT[6:0] = 0 到 127（必须重复此步骤，直到配置了延迟线长度）：
   a) 更新 UNIT[6:0] 值并等待长度标志 LNGF 置 1。
   b) 读取 LNG[11:0]。
   如果 (LNG[10:0] > 0) 且（LNG[11] 或 LNG[10] = 0），则将延迟线长度配置为一个输入时钟周期。
5. 确定需要多少个单位延迟 (N) 才能涵盖一个输入时钟周期：当 N = 0 到 10 时，如果 LNG[N] = 1，则涵盖输入时钟周期的单位延迟数量 = N。
6. 通过将 SEN 清零来禁止长度采样。

如果需要短于一个输入时钟周期的输出时钟延迟，可以将延迟线长度减小到不足一个输入时钟周期。这样可以实现更小的单位延迟，从而在更短的时间间隔内提供更高的分辨率。

25.3.5 输出时钟相位配置程序

当将延迟线长度配置为一个输入时钟周期时，可以在涵盖一个输入时钟周期的单位延迟之间选择输出时钟相位。

请按照以下步骤选择输出时钟相位：

1. 通过将 SEN 位置 1 来禁止输出时钟并使能访问相位选择 SEL[3:0] 位。
2. 使用期望的输出时钟相位值对 SEL[3:0] 进行编程。
3. 通过将 SEN 清零来使能所选相位的输出时钟。

SDMMC 用例：

延迟模块与 SDMMC 接口可变延迟搭配使用。为了正确调整采样点，延迟值必须覆盖整个 SDMMC_CK 时钟周期。调整延迟线长度后，将使用各个延迟在采样点调整中找到最佳采样点。

25.4 DLYB 寄存器

可以对所有寄存器进行字、半字和字节访问。

25.4.1 DLYB 控制寄存器 (DLYB_CR)

DLYB control register

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEN	DEN
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **SEN**：采样器长度使能位 (Sampler length enable bit)

0：禁止采样器长度和寄存器访问 UNIT[6:0] 和 SEL[3:0]，使能输出时钟。

1：使能采样器长度和寄存器访问 UNIT[6:0] 和 SEL[3:0]，禁止输出时钟。

位 0 **DEN**：延迟模块使能位 (Delay block enable bit)

0：禁止 DLYB。

1：使能 DLYB。

25.4.2 DLYB 配置寄存器 (DLYB_CFGR)

DLYB configuration register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LNGF	Res.	Res.	Res.	LNG[11:0]											
r				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	UNIT[6:0]							Res.	Res.	Res.	Res.	SEL[3:0]			
	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw

位 31 **LNGF**：长度有效标志 (Length valid flag)

该标志指示在 UNIT[6:0] 位更改之后，LNG[11:0] 中包含的延迟线长度值是否有效。

0：LNG 中的长度值无效。

1：LNG 中的长度值有效。

位 30:28 保留，必须保持复位值。

位 27:16 **LNG[11:0]**：延迟线长度值 (Delay line length value)
这些位反映了在输入时钟的上升沿采样的 12 个单位延迟值。
此值仅在 LNGF = 1 时有效。

位 15 保留，必须保持复位值。

位 14:8 **UNIT[6:0]**：单位延迟单元的延迟 (Delay of a unit delay cell)
这些位只能在 SEN = 1 时写入。
单位延迟 = 初始延迟 + UNIT[6:0] x 延迟步长

位 7:4 保留，必须保持复位值。

位 3:0 **SEL[3:0]**：输出时钟的相位 (Phase for the output clock)
这些位只能在 SEN = 1 时写入。
输出时钟相位 = 输入时钟 + SEL[3:0] x 单位延迟

25.4.3 DLYB 寄存器映射

表 202. DLYB 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	DLYB_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEN	DEN
	Reset value																															0	0
0x004	DLYB_CFGR	LNGF	Res.	Res.	Res.	LNG[11:0]												Res.	UNIT[6:0]							Res.	Res.	Res.	Res.	SEL[3:0]			
	Reset value	0				0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0					0	0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

26 模数转换器 (ADC1/2)

26.1 简介

ADC 由 12 位逐次逼近模数转换器组成。

ADC 的复用通道多达 19 条。各种不同通道的 A/D 转换可在单次、连续、扫描或不连续采样模式下进行。ADC 的结果存储在一个左对齐或右对齐的 16 位数据寄存器中。

ADC 映射到 AHB 总线，从而可实现快速数据处理。

ADC 具有模拟看门狗特性，允许应用检测输入电压是否超过了用户自定义的阈值上限或下限。

内置硬件过采样器，可提高模拟性能，同时还能减轻 CPU 进行相关计算的负担。

采用了高效的低功耗模式，在低频下可实现极低的功耗。

26.2 ADC 主要特性

- 高性能特性
  - 20 个外部通道和 6 个内部通道
  - 可配置 12 位、10 位、8 位或 6 位分辨率
  - ADC 转换时间与 AHB 总线时钟频率无关
  - 通过降低分辨率缩短转换时间
  - 可管理单端或差分输入
  - AHB 从总线接口，可实现快速数据处理
  - 自校准
  - 可独立设置各通道采样时间
  - 灵活控制采样时间
  - 多达 4 条注入通道（对常规通道或注入通道的模拟输入分配完全可配置）
  - 硬件辅助准备注入通道的上下文，从而实现快速上下文切换
  - 数据对齐以保持内置数据一致性
  - 数据可由 DMA 管理，以实现常规通道转换
  - 数据可连接到 DFSDM 进行后期处理
  - 四个专用数据寄存器供注入通道使用
- 低功耗特性
  - 速度自适应低功耗模式，可降低 ADC 在低频下工作时的功耗
  - 可在实现慢速总线频率应用的同时保持最佳 ADC 性能
  - 提供自动控制，可避免 ADC 在低 AHB 总线时钟频率应用中溢出（自动延迟模式）
- 过采样器
  - 16 位数据寄存器
  - 过采样率可在 2x 到 256x 之间进行调整
  - 可编程数据最多可移位 8 位

- 数据预处理
  - 偏移补偿
- 模拟输入通道
  - 外部模拟输入：
    GPIO 引脚有多达 6 条快速通道
    GPIO 引脚有多达 10 条慢速通道
  - 1 条用于内部温度传感器 (V_{SENSE}) 的通道
  - 2 条用于内部参考电压 (V_{REFINT}) 的通道
  - 1 条用于监控外部 VBAT 电源引脚的通道
  - 1 条用于监视内部 V_{DDCORE} 电源的通道
  - 1 通道供 V_{DDCPU} 使用
  - 1 通道供 V_{DDQ_DDR} 使用
- 可通过以下方式启动转换过程：
  - 通过软件启动常规转换和注入转换
  - 通过硬件触发，具有可配置极性的（内部定时器事件或 GPIO 输入事件），用于常规转换和注入转换。
- 转换模式
  - ADC 可转换单条通道，也可扫描一系列通道
  - 单次模式会在每次触发时对选定的输入执行一次转换
  - 连续模式可连续转换选定的输入
  - 不连续模式
- 在 ADC 准备就绪、采样结束、转换结束（常规转换或注入转换）、序列转换结束（常规转换或注入转换）、模拟看门狗 1/2/3 事件或溢出事件时生成中断
- 3 个模拟看门狗
  - 看门狗可以执行滤波以忽略超出范围的数据
- ADC 输入范围：$V_{SSA} \le V_{IN} \le V_{REF+}$

图 195 给出了一个 ADC 的框图。

26.3 ADC 实现

表 203. ADC 特性

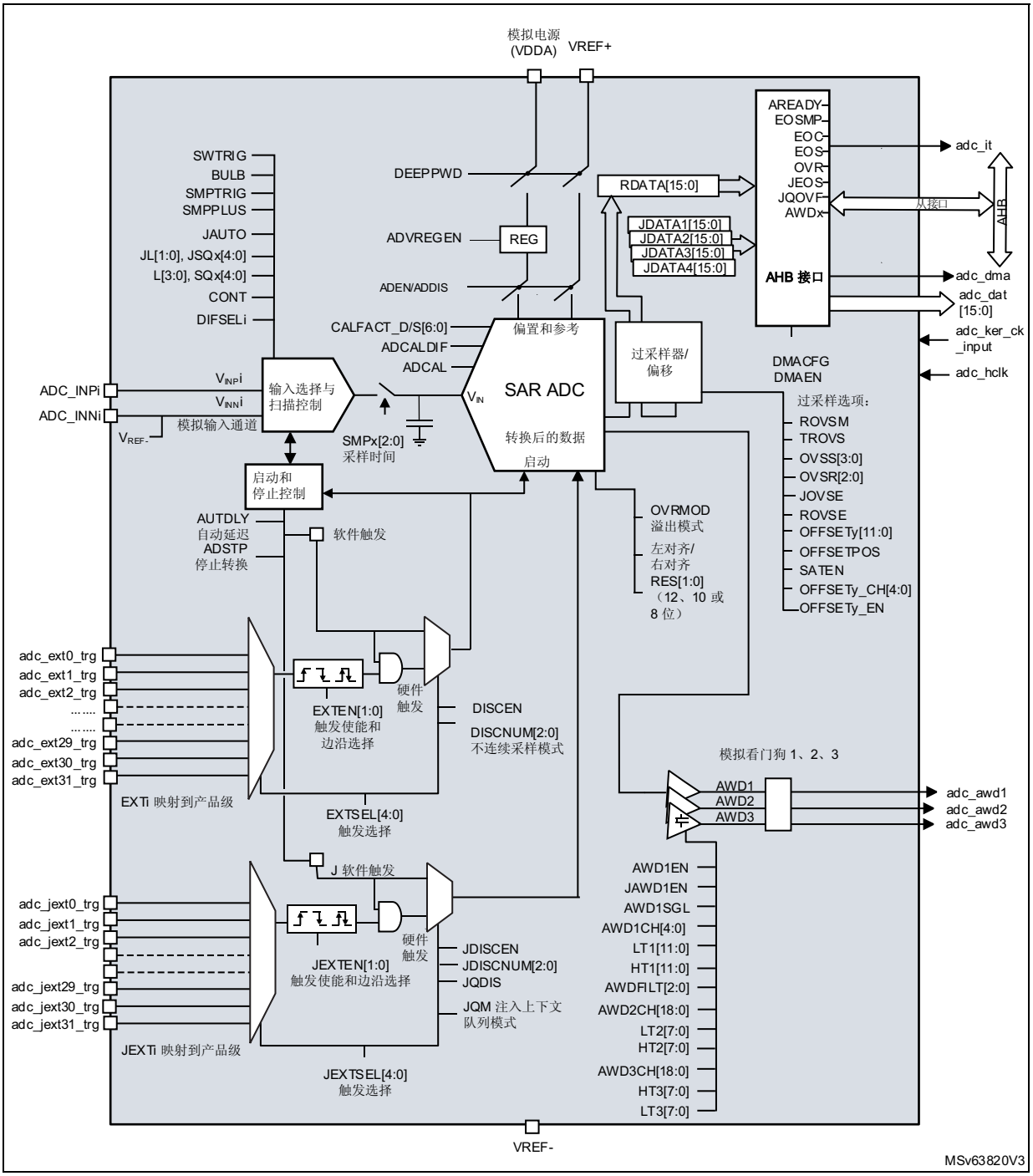
ADC 模式/特性	ADC1	ADC2
分辨率	12 位	
最大采样速度	5 Msps （12 位分辨率）	
双模式操作	-	
硬件偏移校准	X	
硬件线性校准	-	
单端输入	X	
差分输入	X	
注入通道转换	X	
过采样	高达 x256	
数据寄存器	16 位	
数据寄存器 FIFO 深度	3 级	
支持 DMA	X	
并行数据输出到 DFSDM	X	
偏移补偿	X	
增益补偿	-	
模拟看门狗数	3	
选项寄存器	-	X

26.4 ADC 功能描述

26.4.1 ADC 框图

图 195 给出了 ADC 框图，*表 204* 列出了 ADC 的引脚说明。

图 195. ADC 框图



26.4.2 ADC 引脚和内部信号

表 204. ADC 输入/输出引脚

引脚名称	信号类型	说明
VDDA	模拟电源输入	ADC 的模拟电源和正参考电压
VSSA	模拟电源地输入	模拟电源地电压，等于 V_{SS}
VREF+	正模拟参考电压输入	ADC 高/正参考电压。
VREF-	负模拟参考电压输入	ADC 低/负参考电压。V_{REF-} 内部连接到 V_{SSA}
ADC1/2_INNi/INPi	负/正外部模拟输入信号	19 负/正外部模拟输入通道（有关详细信息，请参见*第 26.4.4 节：ADC 连接*）

表 205. ADC 内部输入/输出信号

内部信号名称	信号类型	说明
$V_{INP}i$	正模拟输入通道	正向内部模拟输入通道连接到 ADC1/2_INPi 外部通道或内部通道。
$V_{INN}i$	负模拟输入通道	负向内部模拟输入通道连接到 ADC1/2_INNi 外部通道或内部通道。
adc_ext_trgi	输入	用于常规转换的 ADC 外部触发输入。
adc_jext_trgi	输入	用于注入转换的 ADC 外部触发输入。
adc_awdx	输出	内部模拟看门狗输出信号，连接至片上定时器。（x = 模拟看门狗编号 1、2、3）
adc_ker_ck_input	输出	ADC 内核时钟
adc_hclk	输入	ADC 外设时钟
adc_it	输出	ADC 中断
adc_dma	输出	ADC DMA 请求
adc_dat[15:0]	输出	ADC 数据输出

表 206. ADC 互连

信号名称	源/目标
ADC2 V_{INP}[12]	V_{SENSE}（内部温度传感器输出电压）
ADC1 V_{INP}[18]，ADC2 V_{INP}[13]	V_{REFINT}（内部参考电压的输出电压）
ADC2 V_{INP}[15]	V_{BAT}/4（VBAT 引脚输入电压除以 4）
ADC2 V_{INP}[14]	V_{DDCORE}
ADC2 V_{INP}[16]	V_{DDCPU}
ADC2 V_{INP}[17]	V_{DDQ_DDR}（DDR PHY 电源电压）
adc_dat[15:0]	dfsdm_adc1_dat[15:0]、dfsdm_adc2_dat[15:0]
adc_ext_trg0	tim1_oc1
adc_ext_trg1	tim1_oc2
adc_ext_trg2	tim1_oc3
adc_ext_trg3	tim2_oc2

表 206. ADC 互连 （续）

信号名称	源/目标
adc_ext_trg4	tim3_trgo
adc_ext_trg5	tim4_oc4
adc_ext_trg6	exti11
adc_ext_trg7	tim8_trgo
adc_ext_trg8	tim8_trgo2
adc_ext_trg9	tim1_trgo
adc_ext_trg10	tim1_trgo2
adc_ext_trg11	tim2_trgo
adc_ext_trg12	tim4_trgo
adc_ext_trg13	tim6_trgo
adc_ext_trg14	tim15_trgo
adc_ext_trg15	tim3_oc4
adc_ext_trg16	保留
adc_ext_trg17	保留
adc_ext_trg18	lptim1_out
adc_ext_trg19	lptim2_out
adc_ext_trg20	lptim3_out
adc_ext_trg21	保留
adc_ext_trg22	保留
adc_ext_trg23	保留
adc_ext_trg24	保留
adc_ext_trg25	保留
adc_ext_trg26	保留
adc_ext_trg27	保留
adc_ext_trg28	保留
adc_ext_trg29	保留
adc_ext_trg30	保留
adc_ext_trg31	保留
adc_jext_trg0	tim1_trgo
adc_jext_trg1	tim1_oc4
adc_jext_trg2	tim2_trgo
adc_jext_trg3	tim2_oc1
adc_jext_trg4	tim3_oc4
adc_jext_trg5	tim4_trgo
adc_jext_trg6	exti15

表 206. ADC 互连 （续）

信号名称	源/目标
adc_jext_trg7	tim8_oc4
adc_jext_trg8	tim1_trgo2
adc_jext_trg9	tim8_trgo
adc_jext_trg10	tim8_trgo2
adc_jext_trg11	tim3_oc3
adc_jext_trg12	tim3_trgo
adc_jext_trg13	tim3_oc1
adc_jext_trg14	tim6_trgo
adc_jext_trg15	tim15_trgo
adc_jext_trg16	保留
adc_jext_trg17	保留
adc_jext_trg18	lptim1_out
adc_jext_trg19	lptim2_out
adc_jext_trg20	lptim3_out
adc_jext_trg21	保留
adc_jext_trg22	保留
adc_jext_trg23	保留
adc_jext_trg24	保留
adc_jext_trg25	保留
adc_jext_trg26	保留
adc_jext_trg27	保留
adc_jext_trg28	保留
adc_jext_trg29	保留
adc_jext_trg30	保留
adc_jext_trg31	保留

26.4.3 ADC 时钟

双时钟域架构

双时钟域架构意味着 ADC 时钟独立于 AHB 总线时钟。

可以在两个不同的时钟源之间选择 ADC 输入时钟（参见*图 196：ADC 时钟方案*）：

1. ADC 时钟可以是特定时钟源 (adc_ker_ck_input)，该时钟源独立于 AHB 时钟，并与 AHB 时钟异步。

   有关如何生成 ADC 专用时钟的更多信息，请参见*复位和时钟控制 (RCC)* 部分。要选择此时钟方案，ADC_CCR 寄存器的 CKMODE[1:0] 位必须设置为 00。

2. ADC 时钟可由 AHB 时钟除以一个可编程的因数 1、2 或 4 来提供。要选择此时钟方案，ADC_CCR 寄存器的 CKMODE[1:0] 位不得为 00。可编程的分频系数可以通过 ADC_CCR 的 CKMODE[1:0] 位来配置。

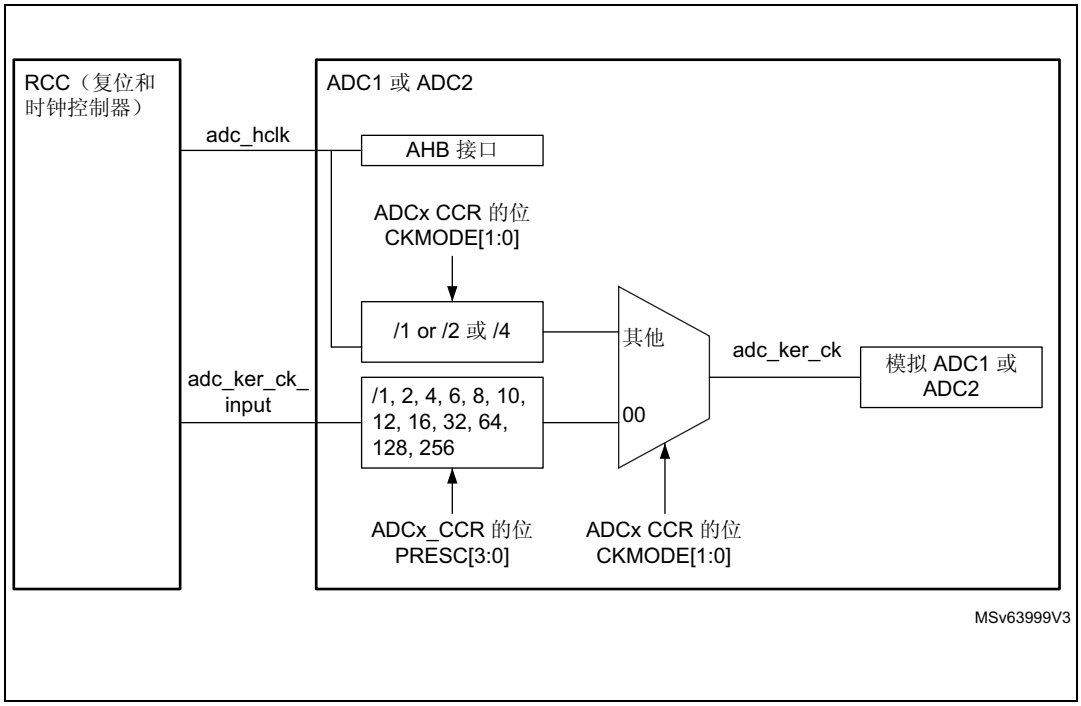
   仅当 AHB 预分频器置 1（RCC_CFGR 寄存器中的 HPRE[3:0] = 0xxx）时，才能使用预分频系数 1 (CKMODE[1:0] = 01)。

选项 **1** 的优点在于无论选择哪种 AHB 时钟方案，都可以达到最大 ADC 时钟频率。ADC 时钟最终可以通过以下比率进行分频：1、2、4、6、8、12、16、32、64、128、256，分频比通过在 ADC_CCR 寄存器中的 PRESC[3:0] 位配置。

选项 **2** 的优势在于绕过了时钟域重新同步。如果 ADC 由定时器触发，并且应用要求 ADC 精确触发（不存在任何不确定性），可使用该选项（否则，时钟域之间的重新同步会增加触发时刻的不确定性）。

通过 CKMODE[1:0] 位配置的时钟必须符合设备数据手册中指定的工作频率。

图 196. ADC 时钟方案



ADC 时钟与 AHB 时钟的时钟比例限制

通常来讲，ADC 时钟与 AHB 时钟之比没有限制，但一些注入通道已编程的情况除外，此时，ADC 时钟与 AHB 时钟之比必须遵循以下规定：

- 如果所有通道的分辨率均为 12 位或 10 位，则 $F_{adc_hclk} \geq F_{ADC}/4$
- 如果一些通道的分辨率为 8 位（且所有通道的分辨率均不低于 8 位），则 $F_{adc_hclk} \geq F_{ADC}/3$
- 如果一些通道的分辨率为 6 位，则 $F_{adc_hclk} \geq F_{ADC}/2$

ADC 时钟之间的限制

如果同时使用多个 ADC 接口，则必须对所有 ADC 接口使用来自 RCC 模块的相同时钟，并且不使用预分频比。

26.4.4 ADC 连接

ADC 输入连接到外部通道和内部源，如下所述。

图 197. ADC1 连接功能

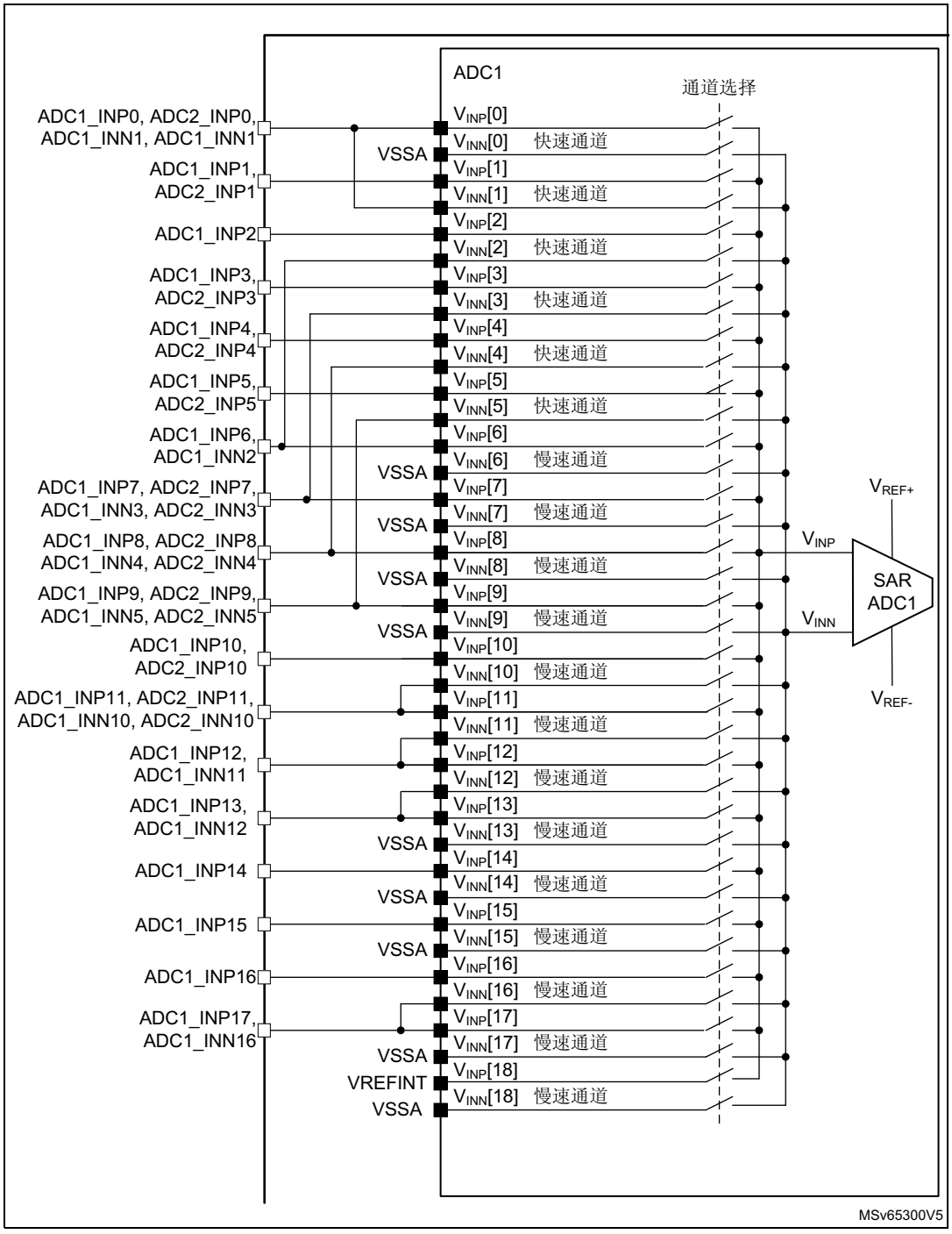
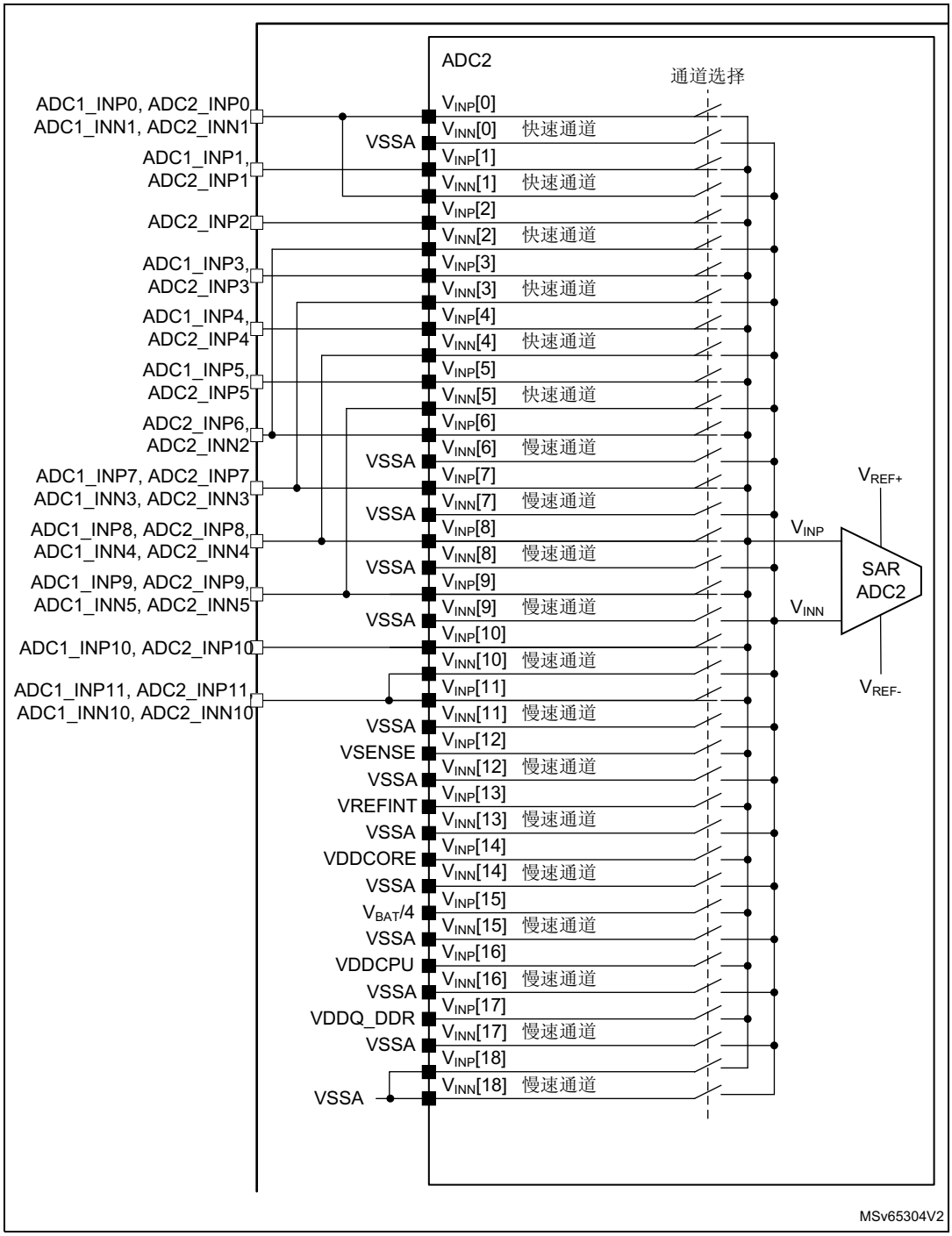


图 198. ADC2连接功能



26.4.5 从设备 AHB 接口

ADC 利用 AHB 从设备端口实现控制/状态寄存器访问和数据访问。AHB 接口的特性如下：

- 字（32 位）访问
- 单周期响应
- 以零等待状态响应对寄存器的所有读/写访问

AHB 从设备接口不支持分离/重试请求，且绝不会生成 AHB 错误。

26.4.6 ADC 深度掉电模式 (DEEPPWD) 和 ADC 稳压器 (ADVREGEN)

ADC 默认处于深度掉电模式，在该模式下，其电源会在内部切断以降低泄漏电流（ADC_CR 寄存器中的 DEEPPWD 位的复位状态为 1）。

要开始 ADC 操作，需要先将 DEEPPWD 位清零，使 ADC 退出深度掉电模式。

然后，必须将 ADC_CR 寄存器中的 ADVREGEN 位置 1，以使能 ADC 内部稳压器。软件在启动 ADC 校准或使能 ADC 之前，必须等待 ADC 稳压器的启动时间 (TADCVREG_STUP)。该延迟必须通过软件实现。

有关 ADC 稳压器的启动时间，请参见设备数据手册中的 $T_{ADCVREG_STUP}$ 参数。

ADC 操作完成后，可禁止 ADC (ADEN = 0)。此外，还可通过禁止 ADC 稳压器实现节能。这可通过向 ADVREGEN 位写入 0 来实现。

之后，为了通过降低泄漏电流的方式进一步实现节能，还可以将 ADC_CR 寄存器中的 DEEPPWD 位置 1，以使 ADC 再次进入深度掉电模式。进入停止模式之前，这一点特别值得关注。

注： *写入 DEEPPWD = 1 会自动禁止 ADC 稳压器，并且 ADVREGEN 位会自动清零。*
如果内部稳压器已禁止 (ADVREGEN = 0)，则会保持内部模拟校准。

在 ADC 深度掉电模式下 (DEEPPWD = 1)，内部模拟校准会丢失，需要重新启动校准或重新应用之前保存的校准系数（请参见*第 26.4.8 节：校准（ADCAL、ADCALDIF、ADC_CALFACT）*）。

26.4.7 单端输入通道和差分输入通道

可通过编程 ADC_DIFSEL 寄存器中的 DIFSEL[i] 位将通道配置为单端输入或差分输入。必须在禁止 ADC (ADEN = 0) 时写入此配置。注意，与各个单端通道对应的 DIFSEL[i] 位始终编程为 0。

在单端输入模式下，要通过通道“i”转换的模拟电压是外部电压 $V_{INP[i]}$（正输入）和 V_{REF-}（负输入）之差。

在差分输入模式下，要为通道“i”转换的模拟电压是外部电压 $V_{INP[i]}$（正输入）与 $V_{INN[i]}$（负输入）之差。

差分模式的输出数据是无符号数据。当 $V_{INP[i]}$ 等于 V_{REF-} 时，$V_{INN[i]}$ 等于 V_{REF+}，并且输出数据为 0x000（12 位分辨率模式）。当 $V_{INP[i]}$ 等于 V_{REF+} 时，$V_{INN[i]}$ 等于 V_{REF-}，并且输出数据为 0xFFF。

$$\text{转换后的值} = \frac{\text{ADC_Full_Scale}}{2} \times \left[1 + \frac{V_{INP} - V_{INN}}{V_{REF+}}\right]$$

当 ADC 配置为差分模式时，两路输入的偏置电压均应为 $(V_{REF+})/2$。

输入信号应为差分信号（共模电压应固定）。

内部通道（例如 V_{REFINT} 和 V_{SENSE}）仅用于单端模式。

有关输入通道连接方式的完整说明，请参见*第 26.4.4 节：ADC 连接*。

小心： 将通道“i”配置为差分输入模式时，其负输入电压 $V_{INN[i]}$ 连接至另一个通道。因此，在单端模式或差分模式下，该通道不再可用，并且不得配置为进行转换。

26.4.8 校准（ADCAL、ADCALDIF、ADC_CALFACT）

ADC 提供有自动校准流程，用于驱动包括 ADC 上电/掉电序列在内的所有校准序列。在校准过程中，ADC 会计算 7 位宽度的校准系数，并会在下一次 ADC 掉电之前在内部应用该系数。在校准过程中，应用不得使用 ADC，必须等待至校准完成。

执行任何 ADC 操作之前都必须进行校准。校准可消除偏移误差，由于制造工艺或带隙的不同，各芯片的偏移误差也有所不同。

单端输入转换与差分输入转换应用的校准系数有所不同：

- 启动要为单端输入转换应用的校准之前，写入 ADCALDIF = 0。
- 启动要为差分输入转换应用的校准之前，写入 ADCALDIF = 1。

随后，通过软件将 ADCAL 位置 1 发起校准。仅当 ADC 禁止 (ADEN = 0) 后，才能发起校准。ADCAL 位在所有校准序列过程中保持为 1。校准完成后，该位会立即由硬件清零。此时，相关校准系数会存储在模拟 ADC 内部，同时还会存储在 ADC_CALFACT 寄存器的 CALFACT_S[6:0] 或 CALFACT_D[6:0] 位中（具体取决于是单端输入校准还是差分输入校准）

如果禁止 ADC (ADEN = 0)，则会保留内部模拟校准。但如果长时间禁止 ADC，建议在重新使能 ADC 之前运行新的校准周期。

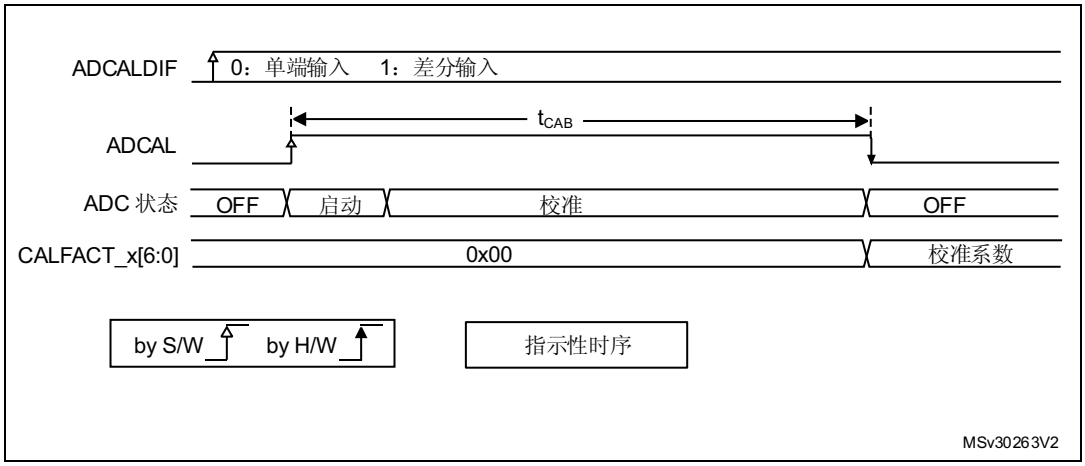
每次 ADC 掉电时，内部模拟校准都会丢失（例如，产品进入待机模式或 VBAT 模式时）。这种情况下，为了避免浪费时间重新校准 ADC，可将校准系数重新写入 ADC_CALFACT 寄存器，但前提是软件之前保存了上次校准时得出的校准系数。

如果 ADC 已使能但未进行转换（ADEN = 1、ADSTART = 0 且 JADSTART = 0），可写入校准系数。随后，下次转换启动时，校准系数会自动添加到模拟 ADC 中。这一载入过程是透明的，不会对转换的启动造成延迟。建议在 V_{REF+} 电压变化超过 10% 时重新进行校准。

ADC 校准的软件流程

1. 确保 DEEPPWD = 0、ADVREGEN = 1，并确保 ADC 稳压器启动时间已过。
2. 确保 ADEN = 0。
3. 设置 ADCALDIF = 0（单端输入）或 ADCALDIF = 1（差分输入），选择此校准的输入模式。
4. 将 ADCAL 置 1。
5. 等待至 ADCAL = 0。
6. 可从 ADC_CALFACT 寄存器读取校准系数。

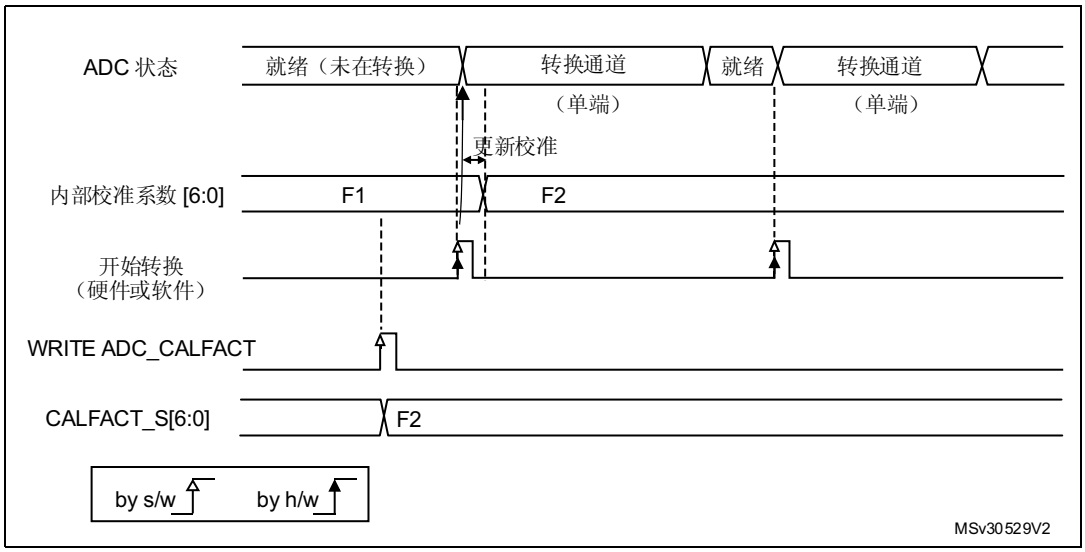
图 199. ADC 校准



将校准系数重新注入 ADC 的软件流程

1. 确保 ADEN = 1、ADSTART = 0 且 JADSTART = 0（ADC 已使能，且未进行任何转换）。
2. 将新的校准系数写入 CALFACT_S 和 CALFACT_D。
3. 启动转换时，仅当内部模拟校准系数与 CALFACT_S 位（单端输入通道）或 CALFACT_D 位（差分输入通道）中存储的值不同时，校准系数才会注入到模拟 ADC。

图 200. 更新 ADC 校准系数

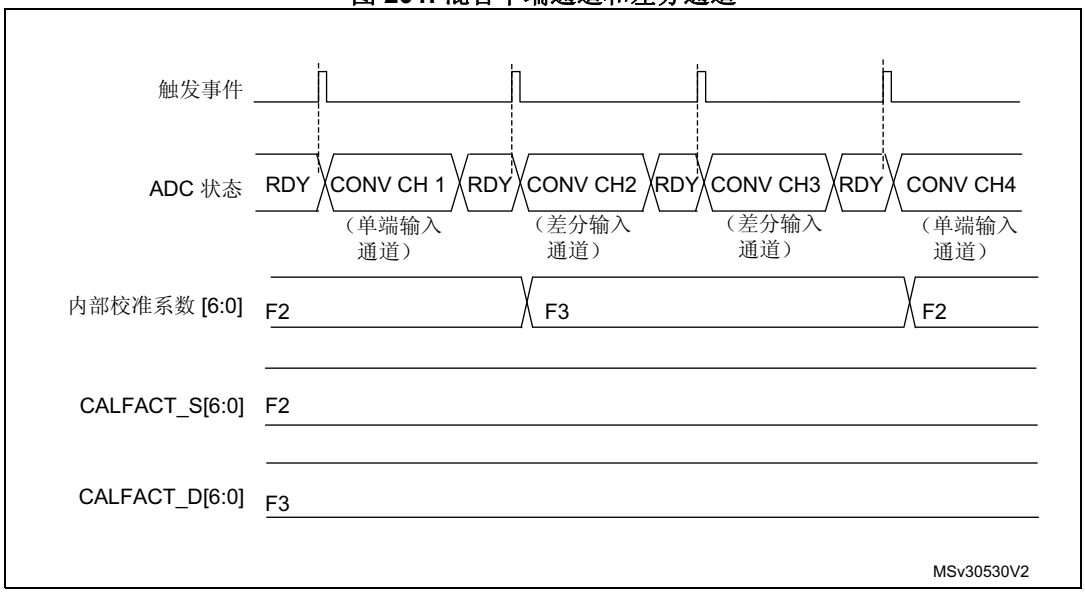


单个 ADC 转换单端模拟输入和差分模拟输入

如果 ADC 既要转换差分输入，也要转换单端输入，则必须执行两次校准，两次校准分别在 ADCALDIF = 0 和 ADCALDIF = 1 的情况下执行。过程如下：

1. 禁止 ADC。
2. 在单端输入模式下 (ADCALDIF = 0) 校准 ADC。此操作会更新寄存器 CALFACT_S[6:0]。
3. 在差分输入模式下 (ADCALDIF = 1) 校准 ADC。此操作会更新寄存器 CALFACT_D[6:0]。
4. 使能 ADC、配置通道并启动转换。每次从单端输入通道切换为差分输入通道（或进行相反切换）时，校准都将自动注入到模拟 ADC 中。

图 201. 混合单端通道和差分通道



26.4.9 ADC 开关控制（ADEN、ADDIS、ADRDY）

首先按照*第 26.4.6 节：ADC 深度掉电模式 (DEEPPWD) 和 ADC 稳压器 (ADVREGEN)* 中的流程执行操作。

DEEPPWD = 0 且 ADVREGEN = 1 后，可使能 ADC，并且 ADC 在开始精确转换之前需要 t_{STAB} 的稳定时间，如*图 202* 所示。以下两个控制位可使能或禁止 ADC：

- 将 ADEN 置 1 可使能 ADC。ADC 准备就绪可以运行后，ADRDY 标志会立即置 1。
- 将 ADDIS 置 1 可禁止 ADC。模拟 ADC 被有效禁止后，ADEN 和 ADDIS 会立即由硬件自动清零。

随后可通过将 ADSTART 置 1（请参见*第 26.4.18 节：外部触发转换和触发极性（EXTSEL、EXTEN、JEXTSEL、JEXTEN）*）开始进行常规转换，如果触发器已使能，也可在发生外部触发事件时开始进行常规转换。

可通过将 JADSTART 置 1 开始进行注入转换，如果注入触发器已使能，也可在发生外部注入触发事件时开始进行注入转换。

通过软件使能 ADC 的流程

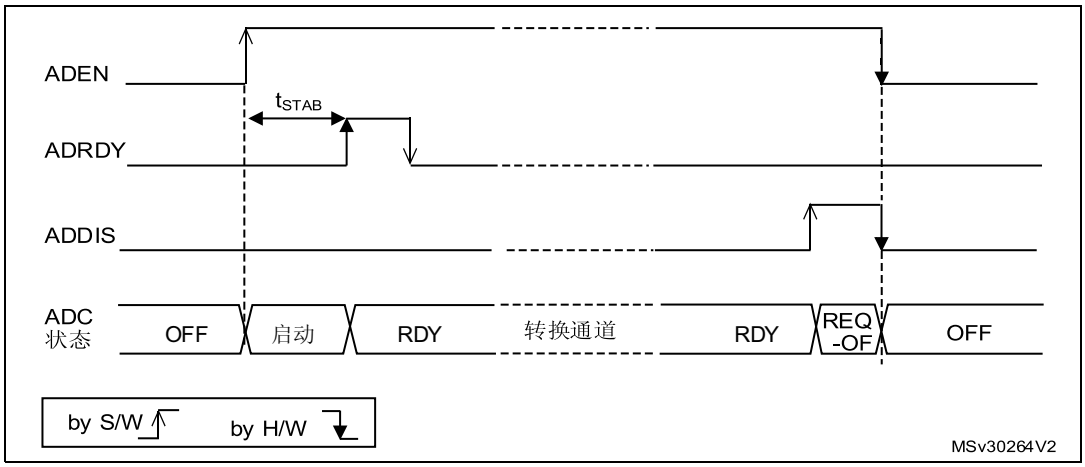
1. 向 ADC_ISR 寄存器中的 ADRDY 位写入“1”，将其清零。
2. 将 ADEN 置 1。
3. 等待至 ADRDY = 1（ADRDY 会在 ADC 启动时间后置 1）。这可以使用关联中断来实现（将 ADRDYIE 置 1）。
4. 向 ADC_ISR 寄存器中的 ADRDY 位写入“1”，将其清零（可选）。

小心： 如果 ADCAL 置 1 并且是在 ADCAL 位由硬件清零（校准结束）后的 4 个 ADC 时钟周期内，则 ADEN 位无法置 1。

通过软件禁止 ADC 的流程

1. 检查 ADSTART 和 JADSTART 是否均为 0，以确保当前未执行任何转换。如有需要，可将 ADSTP 置 1，并将 JADSTP 置 1，随后等待至 ADSTP = 0 且 JADSTP = 0，以停止任何正在进行的常规转换和注入转换。
2. 将 ADDIS 置 1。
3. 如果应用要求，可等待 ADEN = 0，直至模拟 ADC 已有效禁止（ADEN = 0 后，ADDIS 将自动复位）。

图 202. 使能/禁止 ADC



26.4.10 写入 ADC 控制位时的限制

仅当 ADC 已禁止（ADEN 必须等于 0）时，软件才可通过写入 RCC 控制位的方式配置和使能 ADC 时钟（请参见 RCC 部分）、ADC_DIFSEL 寄存器中的 DIFSEL[i] 控制位、ADC_CR 寄存器中的 ADCAL 和 ADEN 控制位。

只有在 ADC 已使能、并且没有待处理的禁止 ADC 的请求（ADEN 必须等于 1 且 ADDIS 必须等于 0）时，才允许软件向 ADC_CR 寄存器的控制位 ADSTART、JADSTART 和 ADDIS 执行写操作。

对于 ADC_CFGR、ADC_SMPRx、ADC_TRy、ADC_SQRy、ADC_JDRy、ADC_OFRy、ADC_OFCHRy 和 ADC_IER 寄存器的所有其他控制位：

- 对于与常规转换配置相关的控制位，仅当 ADC 已使能 (ADEN = 1) 且未进行常规转换（ADSTART 必须等于 0）时，才允许软件对这些位执行写操作。
- 对于与注入转换配置相关的控制位，仅当 ADC 已使能 (ADEN = 1) 且未进行注入转换（JADSTART 必须等于 0）时，才允许软件对这些位执行写操作。
- 可以在进行模数转换时修改 ADC_TRy 寄存器（有关详细信息，请参见*第 26.4.29 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*）。

仅当 ADC 已使能（可能会进行转换）且没有待处理的禁止 ADC 的请求（ADSTART 或 JADSTART 必须等于 1 且 ADDIS 必须等于 0）时，软件才可以对 ADC_CR 寄存器的 ADSTP 或 JADSTP 控制位执行写操作。

如果 ADC 已使能 (ADEN = 1)，软件可随时对 ADC_JSQR 寄存器执行写操作。更多详细信息，请参见*第 26.6.16 节：ADC 注入序列寄存器 (ADC_JSQR)*。

注： *没有硬件保护可以防止执行此类禁止的写操作，ADC 操作可能进入未知状态。要从此状态恢复，必须禁止 ADC（将 ADEN 清零，并且将 ADC_CR 寄存器的所有位都清零）。*

26.4.11 通道选择（SQRx、JSQRx）

ADC 具有多达 19 个复用通道，其中：

- 最多 20 个模拟输入来自 GPIO 引脚 (ADC_INP/INN[i])，视产品而定，并非所有这些输入都在 GPIO 引脚上提供。
- ADC1 连接 1 个内部模拟输入：
  - 内部参考电压 (V_{REFINT})
- ADC 连接 6 个内部模拟输入：
  - 内部温度传感器 (V_{SENSE})
  - 内部参考电压 (V_{REFINT})
  - V_{BAT} 监测通道 ($V_{BAT}/4$)
  - 内部数字内核电压 (V_{DDCORE})
  - V_{DDCPU} 电压
  - V_{DDQ_DDR} 电压

  要对其中一条内部模拟通道进行转换，必须先对 ADC_CCR 寄存器中的 VREFEN、VREFEN 或 TSEN 位进行编程，以使能相应的模拟源。

有关上述内部模拟输入与外部 ADC 引脚或内部信号的连接，请参见*第 26.4.2 节：ADC 引脚和内部信号*中的 *ADC 互连表*。

转换可以分为两组：常规转换和注入转换。每个组包含一个转换序列，该序列可按任意顺序在任意通道上完成。例如，可按以下顺序对序列进行转换：ADC1/2_INP/INN3，ADC1/2_INP/INN8，ADC1/2_INP/INN2，ADC1/2_INN/INP2，ADC1/2_INP/INN0，ADC1/2_INP/INN2，ADC1/2_INP/INN2，ADC1/2_INP/INN15。

- 一个**常规转换组**最多由 16 个转换构成。必须在 ADC_SQRy 寄存器中选择转换序列的常规通道及其顺序。常规转换组中的转换总数必须写入 ADC_SQR1 寄存器中的 L[3:0] 位。
- 一个**注入转换组**最多由 4 个转换构成。必须在 ADC_JSQR 寄存器中选择转换序列的注入通道及其顺序。注入转换组中的转换总数必须写入 ADC_JSQR 寄存器中的 L[1:0] 位。

不得在进行常规转换时对 ADC_SQRy 寄存器进行修改。因此，必须先写入 ADSTP = 1 停止 ADC 常规转换（请参见*第 26.4.17 节：停止正在进行的转换（ADSTP、JADSTP）*）。

只有当 JADSTART 设置为 1（注入转换正在进行）且上下文队列启用（ADC_CFGR 寄存器中的 JQDIS = 0）时，软件才允许动态修改 ADC_JSQR 寄存器。请参见*第 26.4.21 节：注入转换的上下文队列*

26.4.12 可独立设置各通道采样时间（SMPR1、SMPR2）

开始转换之前，ADC 必须在待测量电压源与 ADC 内置采样电容之间建立直接连接。这个采样时间必须足够长，以便输入电压源能够将内部电容充电至输入电压水平。

对各通道进行采样时可以使用不同的采样时间，采样时间可通过 ADC_SMPR1 和 ADC 寄存器中的 SMP[2:0] 位编程。可选采样时间值如下：

- SMP = 000：2.5 个 ADC 时钟周期
- SMP = 001：6.5 个 ADC 时钟周期
- SMP = 010：12.5 个 ADC 时钟周期
- SMP = 011：24.5 个 ADC 时钟周期
- SMP = 100：47.5 个 ADC 时钟周期
- SMP = 101：92.5 个 ADC 时钟周期
- SMP = 110：247.5 个 ADC 时钟周期
- SMP = 111：640.5 个 ADC 时钟周期

总转换时间的计算公式如下：

T_{CONV} = 采样时间 + 12.5 个 ADC 时钟周期

示例：

如果 $F_{adc_ker_ck}$ = 30 MHz，采样时间为 2.5 个 ADC 时钟周期：

T_{CONV} = (2.5 + 12.5) 个 ADC 时钟周期 = 15 个 ADC 时钟周期 = 500 ns

ADC 通过将状态位 EOSMP 置 1 来指示采样阶段结束（仅限常规转换）。

注： *在不同的 ADC 转换模式下，实时采样时间可能会不同于上述设定的 SMP 值，而总转换时间 (T_{CONV}) 不变：*

- *对于扫描或连续模式下的第一次转换以及不连续模式下的所有转换，实时采样时间比上述配置值少 0.5 个时钟周期。*
- *对于扫描或连续模式下的第二次转换和后续转换，配置的采样时间值要加上 0.5 个时钟周期。增加的这 0.5 个时钟周期与上一个转换周期重叠。*

采样时间限制

对于每条通道，必须对 SMP[2:0] 位进行编程，以符合数据手册 ADC 特性部分规定的最短采样时间要求。

Bulb 采样模式

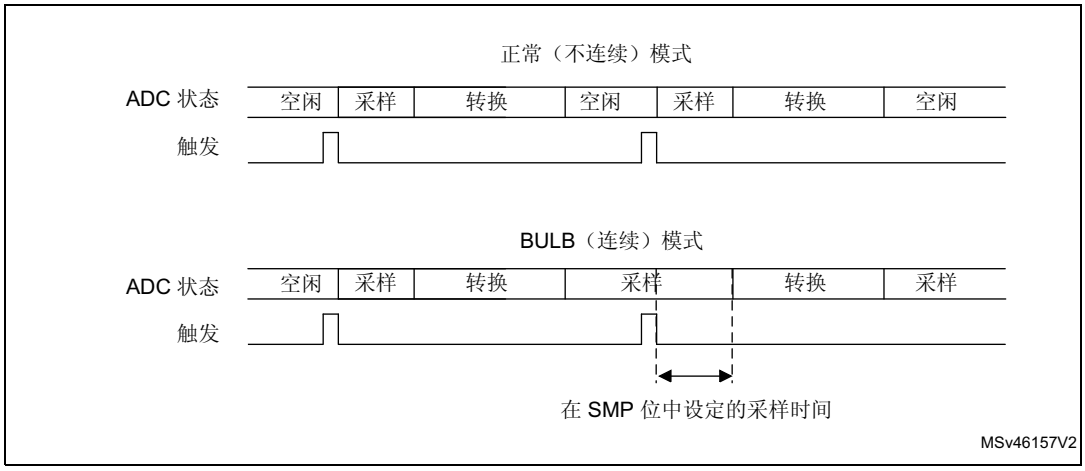
当 ADC 寄存器中的 BULB 位置 1 时，采样周期在上一次 ADC 转换后立即开始。在 ADC_SMPR1 寄存器中设定了采样时间后，会通过硬件触发或软件触发启动转换。使能 ADC 后的第一次 ADC 转换是根据 SMP 位中编程的采样时间进行的。Bulb 模式从第二次转换开始生效。

最大采样时间存在限制（请参见数据手册中的 ADC 特性部分）。

Bulb 模式既不兼容连续转换模式，也不兼容注入通道转换。

当 BULB 位置 1 时，不允许将 ADC_CFGR2 中的 SMPTRIG 位置 1。

图 203. Bulb 模式时序图



采样时间控制触发模式

当 SMPTRIG 位置 1 时，通过 SMPx 位设定的采样时间不适用。采样时间受触发信号边沿控制。

如果选择了硬件触发，则触发信号的每个上升沿均会启动采样周期。下降沿会结束采样周期，并开始进行转换。

如果选择了软件触发，则软件触发并不是 ADC_CR 中的 ADSTART 位，而是 SWTRIG 位。SWTRIG 位必须置 1 才能启动采样周期，SWTRIG 位必须清零才能结束采样周期并开始进行转换。

最大采样时间存在限制（请参见数据手册中的 ADC 特性部分）。

此模式既不兼容连续转换模式，也不兼容注入通道转换。

当 SMPTRIG 位置 1 时，不允许将 BULB 位置 1。

I/O 模拟开关升压器

当 V_{DDA} 电压过低时，I/O 模拟开关的电阻会增大。因此必须相应地调整采样时间（有关相应的电气特性，请参见设备数据手册）。通过 BOOSTEN 位使能内部升压器，或者在 V_{DD} > 2.7 V 时通过 SBS_PMCR 寄存器的 BOOSTVDDSEL 位选择 V_{DD} 增益器电压，均可以在 V_{DDA} 为低电压值时最小化该电阻。

SMPPLUS 控制位

SMPPLUS 位可用于将采样时间从 2.5 个 ADC 时钟周期改为 3.5 个 ADC 时钟周期。

26.4.13 单次转换模式 (CONT = 0)

在单次转换模式下，ADC 会将通道的所有转换执行一次。CONT 位为 0 时，可通过以下方式启动此模式：

- 将 ADC_CR 寄存器中的 ADSTART 位置 1（适用于常规通道）
- 将 ADC_CR 寄存器中的 JADSTART 位置 1（适用于注入通道）
- 外部硬件触发事件（适用于常规通道或注入通道）

在常规序列中，每次转换完成后：

- 转换数据存储在 16 位 ADC_DR 寄存器中
- EOC（常规转换结束）标志置 1
- EOCIE 位置 1 时将产生中断

在注入序列中，每次转换完成后：

- 转换数据存储在四个 16 位 ADC_JDRy 寄存器的其中一个寄存器中
- JEOC（注入转换结束）标志置 1
- JEOCIE 位置 1 时将产生中断

常规序列完成后：

- EOS（常规序列结束）标志置 1
- EOSIE 位置 1 时将产生中断

注入序列完成后：

- JEOS（注入序列结束）标志置 1
- JEOSIE 位置 1 时将产生中断

随后，ADC 会停止工作，直至发生新的外部常规或注入触发，或者 ADSTART 或 JADSTART 位再次置 1。

注： *要转换单个通道，可将序列长度编程为 1。*

26.4.14 连续转换模式 (CONT = 1)

该模式仅适用于常规通道。

在连续转换模式下，如果发生软件或硬件常规触发事件，ADC 会将通道的所有常规转换执行一次，随后会自动重启并持续执行序列的每个转换。CONT 位为 1 时，可通过外部触发或将 ADC_CR 寄存器中的 ADSTART 位置 1 来启动此模式。

在常规序列中，每次转换完成后：

- 转换数据存储在 16 位 ADC_DR 寄存器中
- EOC（转换结束）标志置 1
- EOCIE 位置 1 时将产生中断

转换序列完成后：

- EOS（序列结束）标志置 1
- EOSIE 位置 1 时将产生中断

随后，会立即重启新序列，ADC 会继续重复执行转换序列。

注： *要转换单个通道，可将序列长度编程为 1。*

不能同时使能不连续模式和连续模式：禁止同时将 DISCEN 和 CONT 位置 1。

注入通道不能连续转换，唯一例外的是，在连续转换模式下（使用 JAUTO 位）注入通道配置为在常规通道后的自动转换，请参见自动注入模式一节。

26.4.15 开始转换（ADSTART、JADSTART）

软件通过将 ADSTART 置 1 的方式开始进行 ADC 常规转换。

ADSTART 置 1 后，会开始进行转换：

- 立即开始转换：EXTEN = 0x0 时（软件触发）
- 在所选常规硬件触发的下一有效边沿开始转换：EXTEN 不等于 0x0 时

软件通过将 JADSTART 置 1 的方式开始进行 ADC 注入转换。

JADSTART 置 1 后，会开始进行转换：

- 立即开始转换：JEXTEN = 0x0 时（软件触发）
- 在所选注入硬件触发的下一有效边沿开始转换：JEXTEN 不等于 0x0 时

注： *在自动注入模式下 (JAUTO = 1)，使用 ADSTART 位开始常规转换，然后再进行自动注入转换（JADSTART 必须保持清零）。*

ADSTART 位和 JADSTART 位还提供当前是否正在进行 ADC 操作的信息。可以在 ADSTART = 0 且 JADSTART = 0（指示 ADC 处于空闲状态）时重新配置 ADC。

ADSTART 通过硬件清零：

- 在使用软件常规触发的单次模式下（CONT = 0，EXTSEL = 0x0）
  - 如果 DISCEN = 1，只要常规转换序列结束（EOS 置 1）或子组处理结束就清零
- 在所有情况下（CONT = x、EXTSEL = x）
  - 执行由软件调用的 ADSTP 程序之后清零。

注： *在连续模式下 (CONT = 1)，由于序列会自动重新启动，因此，当 EOS 置 1 时，ADSTART 位不会通过硬件清零。*

如果在单次模式下选择了硬件触发（CONT = 0 且 EXTSEL 不等于 0x00），当 EOS 置 1 时，ADSTART 位不会通过硬件清零，借此软件无需再次为下一个硬件触发事件复位 ADSTART。这样可确保不会错过任何后续的硬件触发。

JADSTART 通过硬件清零：

- 在使用软件注入触发的单次模式下 (JEXTSEL = 0x0)
  - 如果 JDISCEN = 1，只要注入转换序列结束（JEOS 置 1）或子组处理结束就清零
- 在所有情况下 (JEXTSEL = x)
  - 执行由软件调用的 JADSTP 程序之后清零。

注： *选择软件触发时，如果 EOC 标志仍为高电平，则不应将 ADSTART 位置 1。*

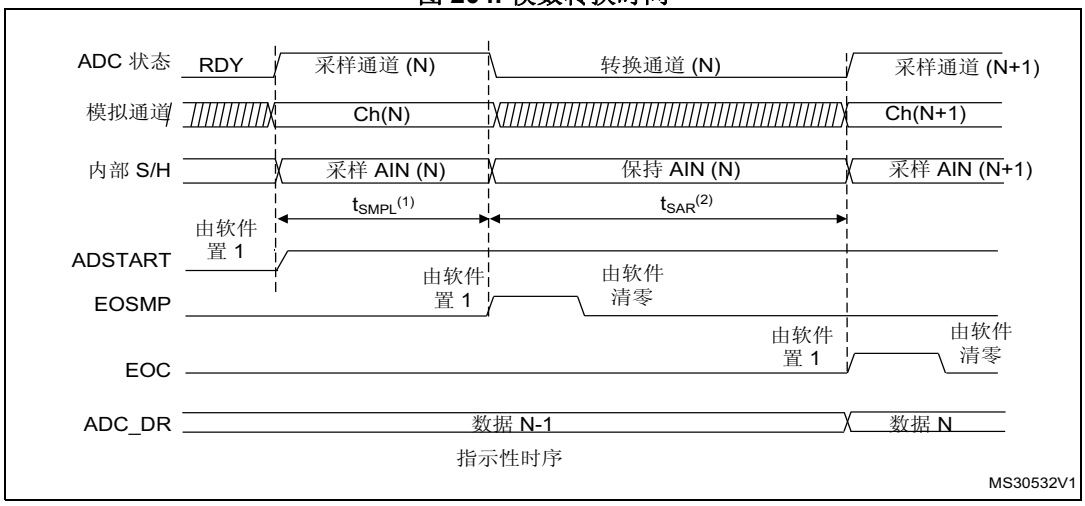
26.4.16 ADC 时序

从转换开始到转换结束所经过的时间是配置的采样时间与逐次逼近时间（具体取决于数据分辨率）的总和：

$$T_{CONV} = T_{SMPL} + T_{SAR} = [2.5\ _{|min} + 12.5\ _{|12bit}] \times T_{ADC_CLK}$$

$$T_{CONV} = T_{SMPL} + T_{SAR} = 83.33\ ns\ _{|min} + 416.67\ ns\ _{|12bit} = 500.0\ ns（F_{ADC_CLK} = 30\ MHz\ 时）$$

图 204. 模数转换时间



1. T_{SMPL} 取决于 SMP[2:0]。
2. T_{SAR} 取决于 RES[2:0]。

26.4.17 停止正在进行的转换（ADSTP、JADSTP）

软件决定是否停止转换，要停止正在进行的常规转换，应将 ADSTP 置 1；要停止正在进行的注入转换，应将 JADSTP 置 1。

停止转换将复位正在进行的 ADC 操作。随后可重新配置 ADC（例如：更改通道选择或触发），为新操作做好准备。

请注意，在常规转换仍在进行的情况下，可以停止注入转换，反之亦然。这样便可在常规转换仍在进行时重新配置注入转换序列和触发（反之亦然）。

如果 ADSTP 位由软件置 1，则会中止任何正在进行的常规转换，并会丢弃部分转换结果（ADC_DR 寄存器不会更新为当前转换结果）。

如果 JADSTP 位由软件置 1，则会中止任何正在进行的注入转换，并会丢弃部分转换结果（ADC_JDRy 寄存器不会更新为当前转换结果）。扫描序列也会中止并会复位（这意味着重启 ADC 将重新开始新的序列）。

该程序执行完毕后，ADSTP/ADSTART 位（常规转换时）或 ADSTP/JADSTART 位（注入转换时）会由硬件清零，软件必须轮询 ADSTART（或 JADSTART）直至其复位，然后才能判定 ADC 已完全停止运行。

注： *在自动注入模式下 (JAUTO = 1)，将 ADSTP 位置 1 会中止常规转换和注入转换（不得使用 JADSTP）。*

图 205. 停止正在进行的常规转换

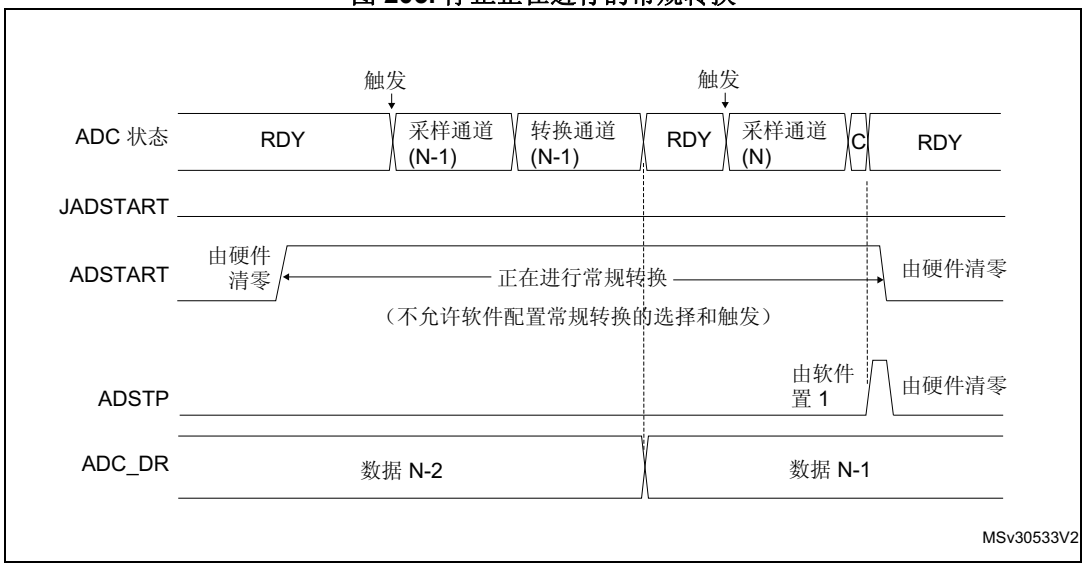
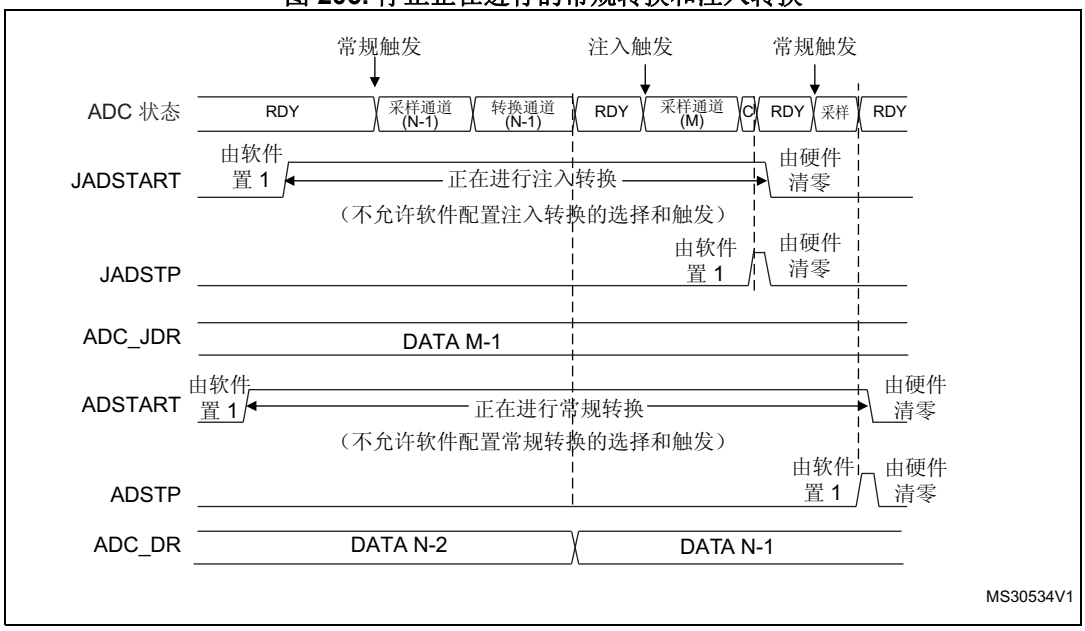


图 206. 停止正在进行的常规转换和注入转换



26.4.18 外部触发转换和触发极性（EXTSEL、EXTEN、JEXTSEL、JEXTEN）

可通过软件或外部事件（例如定时器捕获、输入引脚）触发转换或转换序列。如果 EXTEN[1:0] 控制位（对于常规转换）或 JEXTEN[1:0] 位（对于注入转换）不等于“0b00”，则外部事件能够以所选极性触发转换。

注入队列使能时（位 JQDIS = 0），不能实现注入软件触发。

软件将 ADSTART 位置 1 后，常规触发选择有效；软件将 JADSTART 位置 1 后，注入触发选择有效。

在转换进行时发生的硬件触发会被忽略。

- 如果位 ADSTART = 0，则会忽略发生的任何常规硬件触发。
- 如果位 JADSTART = 0，则会忽略发生的任何注入硬件触发。

表 207 提供了 EXTEN[1:0] 和 JEXTEN[1:0] 值与触发极性之间的对应关系。

表 207. 为常规外部触发配置触发极性

EXTEN[1:0]	触发源
00	禁止硬件触发检测，使能软件触发检测
01	在上升沿执行硬件触发检测
10	在下降沿执行硬件触发检测
11	在上升沿和下降沿均执行硬件触发检测

注： *不能实时更改常规触发的极性。*

表 208. 为注入外部触发配置触发极性

JEXTEN[1:0]	触发源
00	– 如果 JQDIS = 1（队列禁止）：禁止硬件触发检测，使能软件触发检测 – 如果 JQDIS = 0（队列使能），同时禁止硬件和软件触发检测
01	在上升沿执行硬件触发检测
10	在下降沿执行硬件触发检测
11	在上升沿和下降沿均执行硬件触发检测

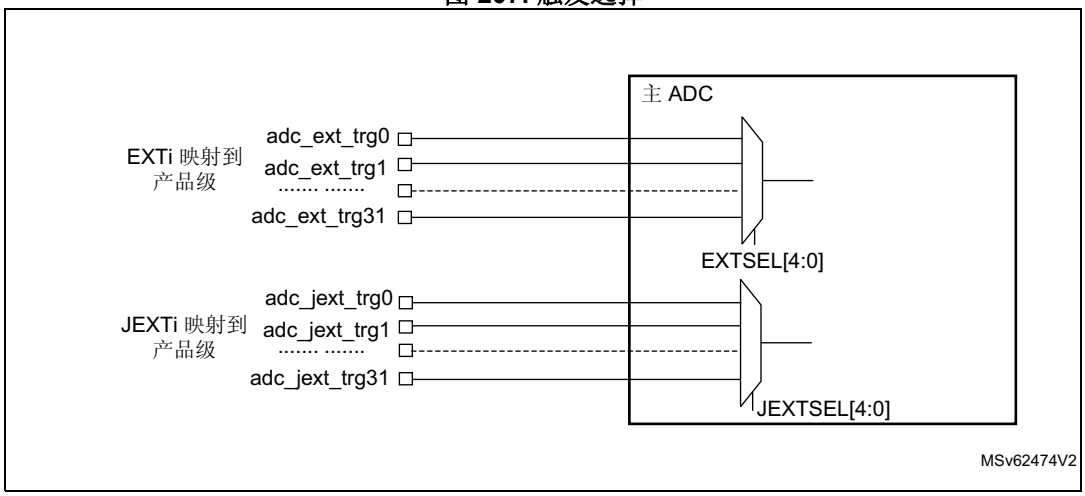
注： *如果队列已使能 (JQDIS = 0)，可预计并实时更改注入触发的极性。请参见第 26.4.21 节：注入转换的上下文队列。*

EXTSEL 和 JEXTSEL 控制位用于从 32 个可能事件中选择可触发常规组转换和注入组转换的事件。

可通过注入触发中断常规组转换。

注： *不能实时更改常规触发选择。可以预计并实时更改注入触发选择。请参见第 1405 页的第 26.4.21 节：注入转换的上下文队列。*

图 207. 触发选择



有关所有可用于进行常规转换的外部触发，请参见*第 26.4.2 节：ADC 引脚和内部信号*中的 ADC 互连表。

26.4.19 注入通道管理

触发注入模式

要使用触发注入，必须将 ADC_CFGR 寄存器中的 JAUTO 位清零。

1. 通过外部触发或将 ADC_CR 寄存器中的 ADSTART 位置 1 来启动常规通道组转换。
2. 如果在常规通道组转换期间出现外部注入触发或者 ADC_CR 寄存器中的 JADSTART 位置 1，则当前的转换会复位，并会启动注入通道序列切换（所有注入通道都会转换一次）。
3. 然后，常规通道组的常规转换会从上次中断的常规转换处恢复。
4. 如果在注入转换期间出现常规事件，注入转换不会中断，但在注入序列结束时会执行常规序列。*图 208* 显示了相应的时序图。

注： *使用触发注入时，必须确保触发事件之间的间隔长于注入序列。例如，如果序列长度为 30 个 ADC 时钟周期（即，采样时间为 2.5 个时钟周期的两次转换），则触发事件的最小间隔不能小于 31 个 ADC 时钟周期。*

自动注入模式

如果将 ADC_CFGR 寄存器中的 JAUTO 位置 1，则注入组中的通道会在常规组通道之后自动转换。这可用于转换最多由 20 个转换构成的序列，这些转换在 ADC_SQRy 和 ADC_JSQR 寄存器中编程。

在该模式下，必须将 ADC_CR 寄存器中的 ADSTART 位置 1 以开始常规转换，然后再进行注入转换（JADSTART 必须保持清零）。将 ADSTP 位置 1 会中止常规转换和注入转换（不得使用 JADSTP 位）。

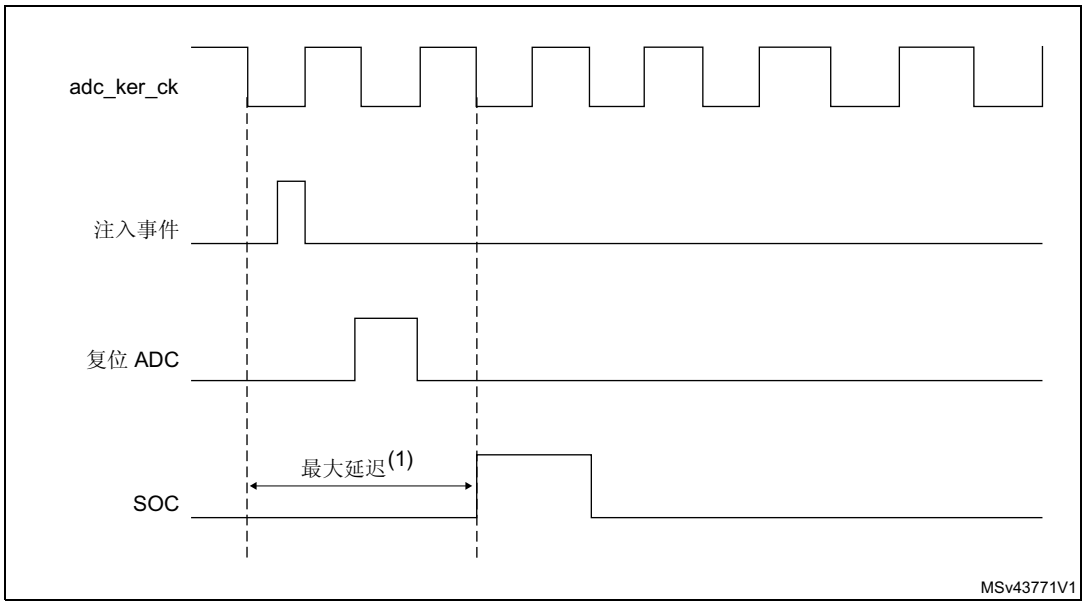
在此模式下，必须禁止注入通道上的外部触发。

如果 CONT 位和 JAUTO 位均已置 1，则在转换常规通道之后会继续转换注入通道。

注： *不能同时使用自动注入和不连续采样模式。*

如果在 JAUTO 模式下使用 DMA 导出常规定序器的数据，需要将其设定为循环模式（将 DMA_CCRx 寄存器中的 CIRC 位置 1）。如果 CIRC 位复位（单发模式），JAUTO 序列将在出现 DMA 传输完成事件时停止。

图 208. 注入转换延迟



1. 有关延迟的最大值，请参见设备数据手册中的电气特性部分。

26.4.20 不连续模式（DISCEN、DISCNUM、JDISCEN）

常规组模式

可将 ADC_CFGR 寄存器中的 DISCEN 位置 1 来使能此模式。

该模式用于转换含有 n (n ≤ 8) 个转换的短序列（子组），该短序列是在 ADC_SQRy 寄存器中选择的转换序列的一部分。可通过写入 ADC_CFGR 寄存器中的 DISCNUM[2:0] 位来指定 n 的值。

出现外部触发时，将启动在 ADC_SQRy 寄存器中选择的接下来 n 个转换，直到序列中的所有转换均完成为止。通过 ADC_SQR1 寄存器中的 L[3:0] 位定义总序列长度。

示例：

- DISCEN = 1，n = 3，要转换的通道 = 1、2、3、6、7、8、9、10、11
  - 第一次触发：转换的通道为 1、2、3（每次转换时都生成 EOC 事件）。
  - 第二次触发：转换的通道为 6、7、8（每次转换时都生成 EOC 事件）。
  - 第三次触发：转换的通道为 9、10、11（每次转换时都生成 EOC 事件），并会在通道 11 转换完成后生成 EOS 事件。
  - 第四次触发：转换的通道为 1、2、3（每次转换时都生成 EOC 事件）。
  - ...

- DISCEN = 0，要转换的通道 = 1、2、3、6、7、8、9、10、11
  - 第一次触发：转换整个序列：通道 1、然后是通道 2、3、6、7、8、9、10 和 11。每次转换都会生成 EOC 事件，最后一次转换还会生成 EOS 事件。
  - 所有后续触发事件都将重启整个序列。

注： *上例中提及的通道数并非在所有微控制器上均能提供。*

在不连续模式下转换常规组时，不会出现翻转（序列最后一个子组的转换次数少于 n 次）。

转换完所有子组后，下一个触发信号将启动第一个子组的转换。在上述示例中，第四次触发重新转换了第一个子组中的通道 1、2 和 3。

不能同时使能不连续模式和连续模式。如果同时使能两种模式（即 DISCEN = 1、CONT = 1），ADC 会认定连续模式已禁止并继续执行相关操作。

注入组模式

可将 ADC_CFGR 寄存器中的 JDISCEN 位置 1 来使能此模式。在出现外部注入触发事件之后，该模式会逐通道转换在 ADC_JSQR 寄存器中选择的序列，相当于不连续模式下常规通道“n”固定为 1 的情况。

出现外部触发时，将启动在 ADC_JSQR 寄存器中选择的下一个通道转换，直到序列中的所有转换均完成为止。通过 ADC_JSQR 寄存器中的 JL[1:0] 位定义总序列长度。

示例：

- JDISCEN = 1，要转换的通道 = 1、2、3
  - 第一次触发：转换通道 1（生成 JEOC 事件）
  - 第二次触发：转换通道 2（生成 JEOC 事件）
  - 第三次触发：转换通道 3 并生成 JEOC 事件和 JEOS 事件
  - ...

注： *上例中提及的通道数并非在所有微控制器上均能提供。*

转换完所有注入通道后，下一个触发信号将启动第一个注入通道的转换。在上述示例中，第四次触发重新转换了第一个注入通道 1。

不能同时使用自动注入模式和不连续模式：当 JAUTO 置 1 时，DISCEN 和 JDISCEN 位必须通过软件保持清零状态。

26.4.21 注入转换的上下文队列

实现上下文队列可为下一个注入转换序列准备多达 2 个上下文。必须将 ADC_CFGR 寄存器的 JQDIS 位复位才能使能此功能。上下文队列使能时，只能进行硬件触发转换。

该上下文包括：

- 注入触发的配置（ADC_JSQR 寄存器中的 JEXTEN[1:0] 位和 JEXTSEL 位）
- 注入序列的定义（ADC_JSQR 寄存器中的 JSQx[4:0] 位和 JL[1:0] 位）

上下文的所有参数都会在 ADC_JSQR 这一寄存器中定义，该寄存器会实现一个双缓冲区队列，可缓冲多达 2 组参数：

- JSQR 寄存器可随时写入，正在进行注入转换时也不例外。
- 每个写入到 JSQR 寄存器中的数据均会存储在上下文队列中。
- 队列开始时为空，对 JSQR 寄存器进行的第一次写访问时会立即更改上下文，随即 ADC 会准备好接收注入触发。
- 注入序列完成后，队列会被占用，上下文会根据队列中存储的后续 JSQR 参数进行更改。这一新的上下文会用于下一个注入转换序列。
- 如果在队列已满的情况下向 JSQR 寄存器执行写操作，会发生队列溢出。这种溢出情况会通过 JQOVF 标志置为有效来指示。发生溢出时，会忽略造成溢出的 JSQR 寄存器写访问，并且上下文队列保持不变。如果 JQOVFIE 位置 1，可产生中断。
- 队列变空时可能执行两种操作，具体取决于寄存器 ADC_CFGR 的控制位 JQM 的值：
  - 如果 JQM = 0，队列刚好在使能 ADC 后为空，但在随后的运行操作期间绝不会变空：队列始终保留上一个有效的上下文，并会根据上一个有效的上下文处理后续的有效注入序列启动。
  - 如果 JQM = 1，队列会在注入序列结束后或队列被清空时变空。这种情况下，队列中不存在任何上下文，硬件触发也会被禁止。因此会忽略后续的所有硬件注入触发，直至软件重新向 JSQR 寄存器写入新的注入上下文。
- 读取 JSQR 寄存器会返回当前有效的 JSQR 上下文。如果 JSQR 上下文为空，JSQR 的读出值为 0x0000。
- 如果通过将 JADSTP 置 1 的方式停止注入转换、或者通过将 ADDIS 置 1 的方式禁止 ADC，队列会被刷新：
  - 如果 JQM = 0，队列会保留上一个有效的上下文。
  - 如果 JQM = 1，队列会变空，并会忽略触发。

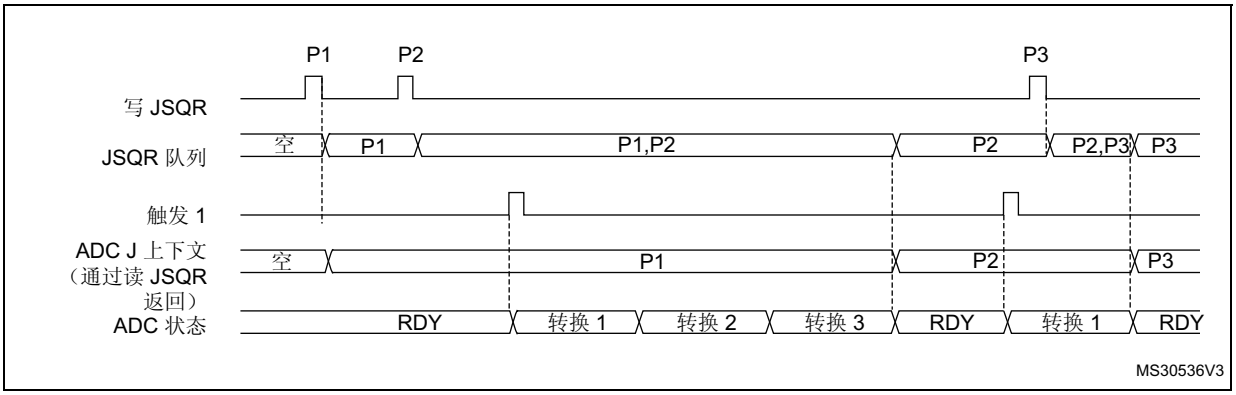
注： *如果配置为不连续模式（位 JDISCEN = 1），只有注入序列的最后一次触发会更改上下文并占用队列。第一次触发仅会占用队列，但其他触发仍为有效触发，如下文中的不连续模式示例所示（两个上下文的长度均为 3）：*

- *第一次触发，不连续。序列 1：上下文 1 已占用，已执行第一次转换*
- *第二次触发，不连续。序列 1：第二次转换。*
- *第三次触发，不连续。序列 1：第三次转换。*
- *第四次触发，不连续。序列 2：上下文 2 已占用，已执行第一次转换。*
- *第五次触发，不连续。序列 2：第二次转换。*
- *第六次触发，不连续。序列 2：第三次转换。*

更改触发或序列上下文时的操作

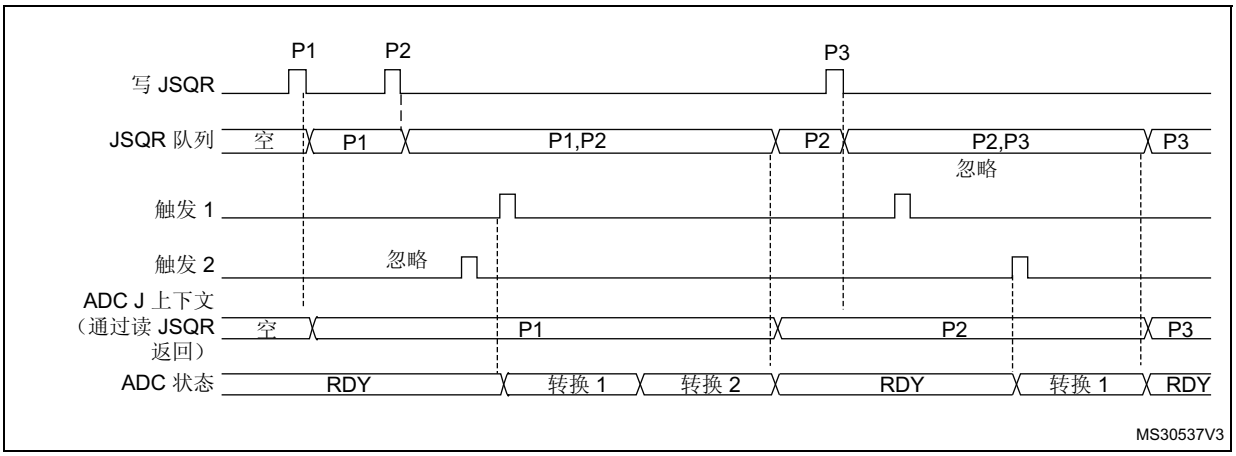
图 209 和*图 210* 显示了更改序列或触发时上下文队列的操作。

图 209. JSQR 上下文队列示例（队列更改）



1. 参数：
   P1：3 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：4 个转换组成的序列，硬件触发 1

图 210. JSQR 上下文队列示例（触发更改）

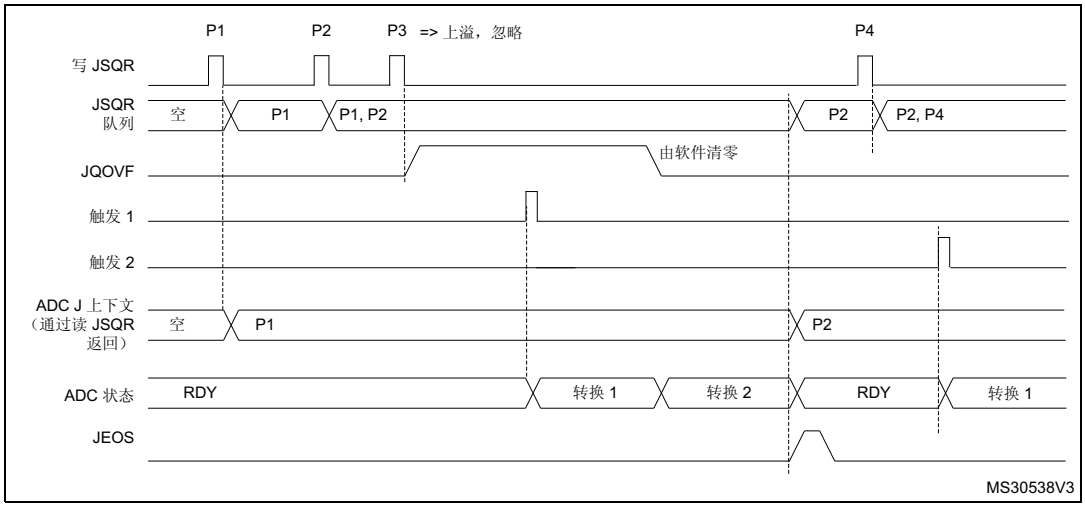


1. 参数：
   P1：2 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 2
   P3：4 个转换组成的序列，硬件触发 1

上下文队列：发生队列溢出时的操作

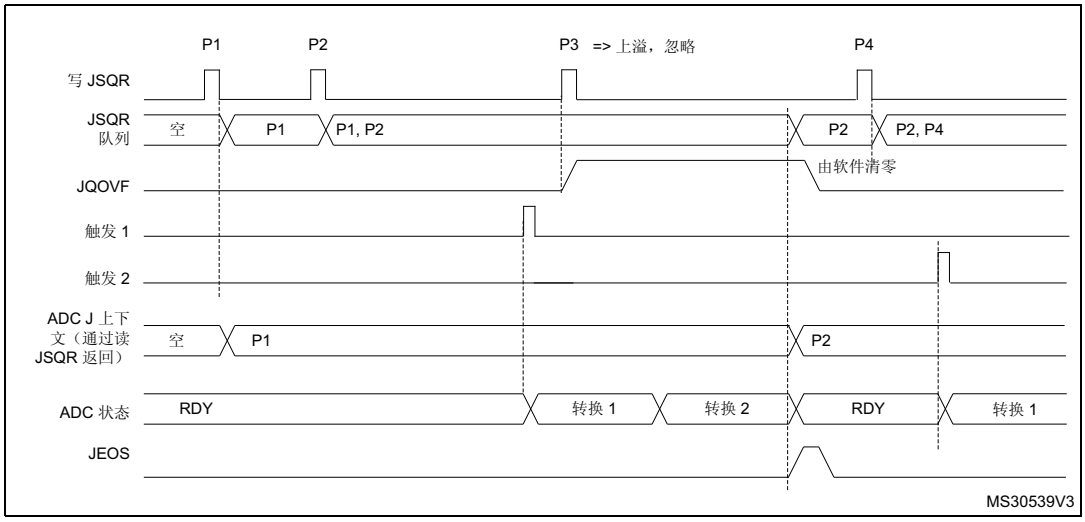
图 211 和*图 212* 显示了转换之前或之后发生溢出时上下文队列的操作。

图 211. 转换前发生溢出的 JSQR 上下文队列示例



1. 参数：
   P1：2 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 2
   P3：3 个转换组成的序列，硬件触发 1
   P4：4 个转换组成的序列，硬件触发 1

图 212. 转换期间发生溢出的 JSQR 上下文队列示例



1. 参数：
   P1：2 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 2
   P3：3 个转换组成的序列，硬件触发 1
   P4：4 个转换组成的序列，硬件触发 1

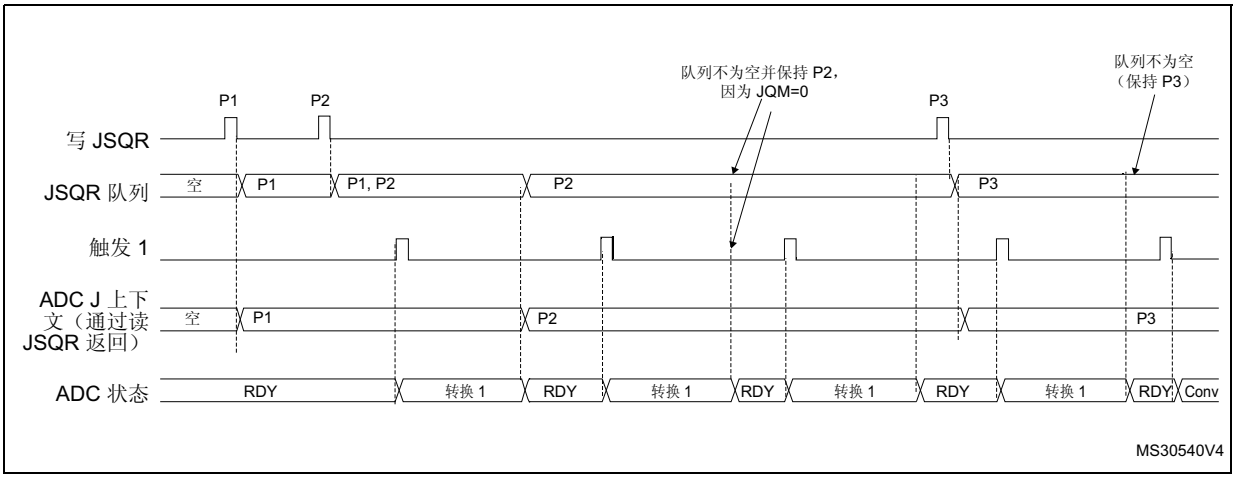
建议按照下述方法管理队列溢出：

- 将每个 P 上下文写入 JSQR 寄存器后，通过标志 JQOVF 指示写操作是否已被忽略（可生成中断）。
- 为避免队列溢出，仅在上一个上下文 P2 的 JEOS 标志置 1 后写入第三个上下文 (P3)。这样可确保上一个上下文已被占用且队列未满。

上下文队列：队列变空时的操作

图 213 和*图 214* 显示了在 JQM = 0 或 1 这两种情况下队列变空时上下文队列的操作。

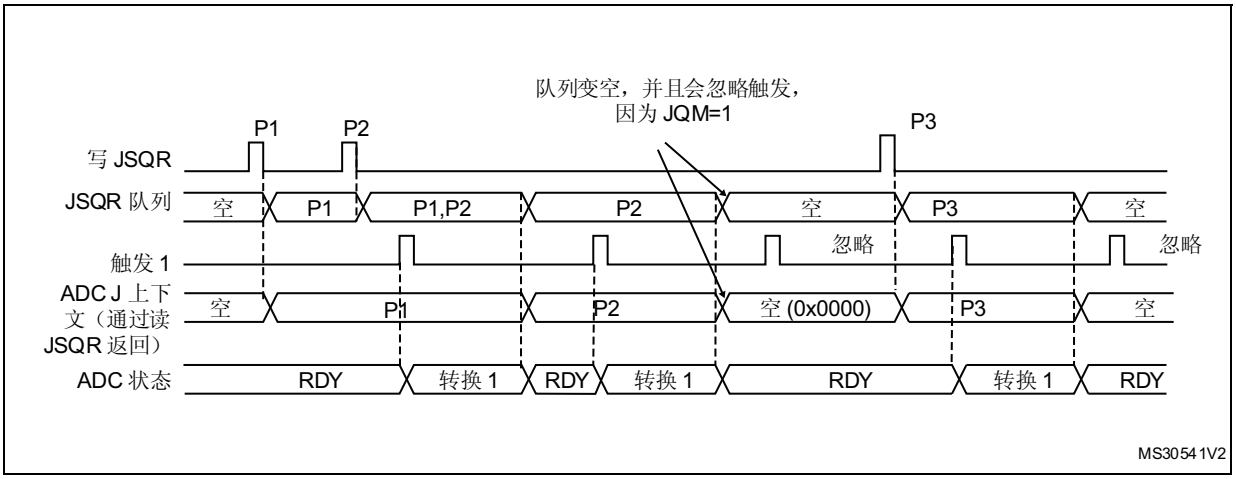
图 213. 队列为空时的 JSQR 上下文队列示例（JQM = 0 的情况）



1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

注： *写入 P3 时，上下文会立即改变。但由于内部重新同步的原因，会存在一定的延迟，如果刚好在写入 P3 之后或之前发生触发，可能会发生启动的转换被视为上下文 P2 的情况。为了避免出现这种情况，用户必须确保写入立即应用的新上下文时没有发生 ADC 触发。*

图 214. 队列为空时的 JSQR 上下文队列示例（JQM = 1 的情况）

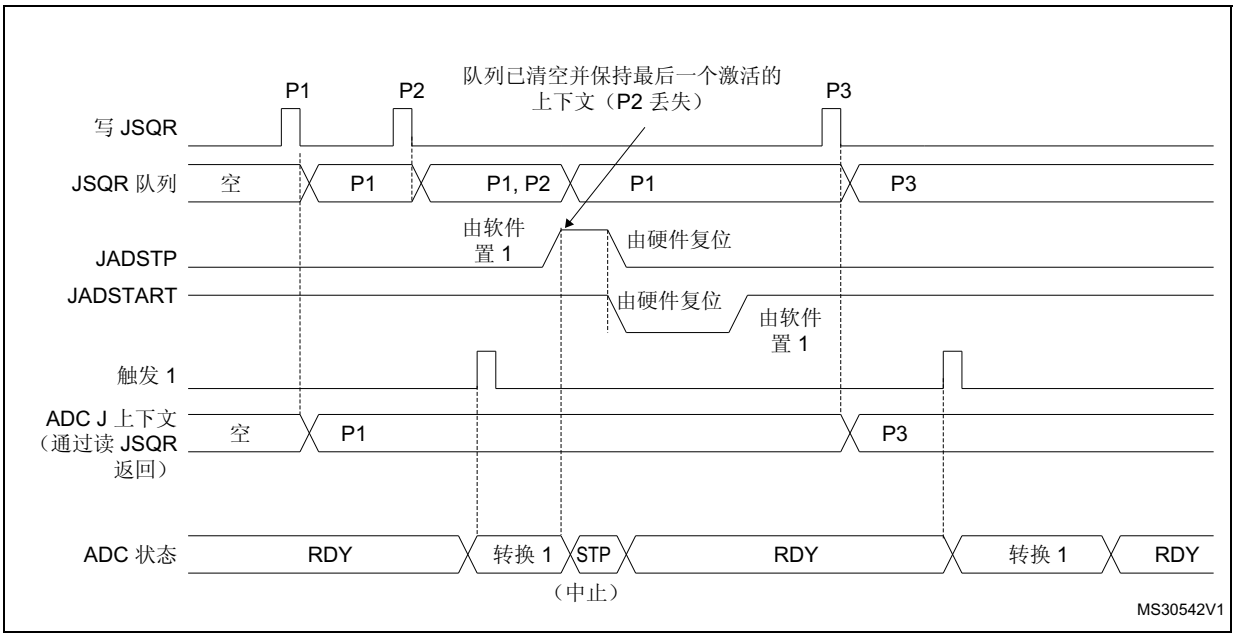


1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

清空上下文队列

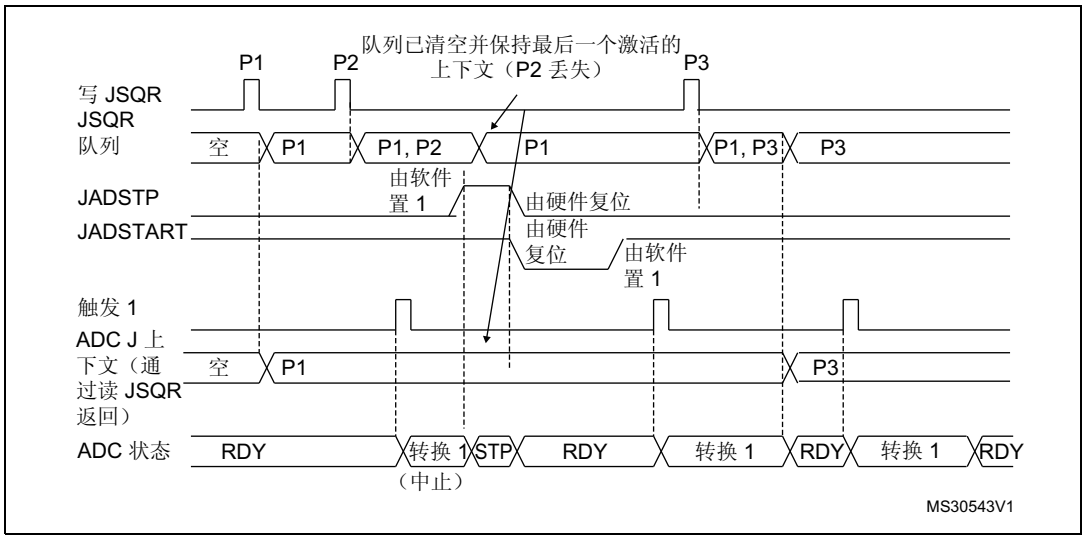
下图显示的是各种情况下清空队列时上下文队列的操作。

图 215. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM = 0) —— 正在进行转换时发生 JADSTP



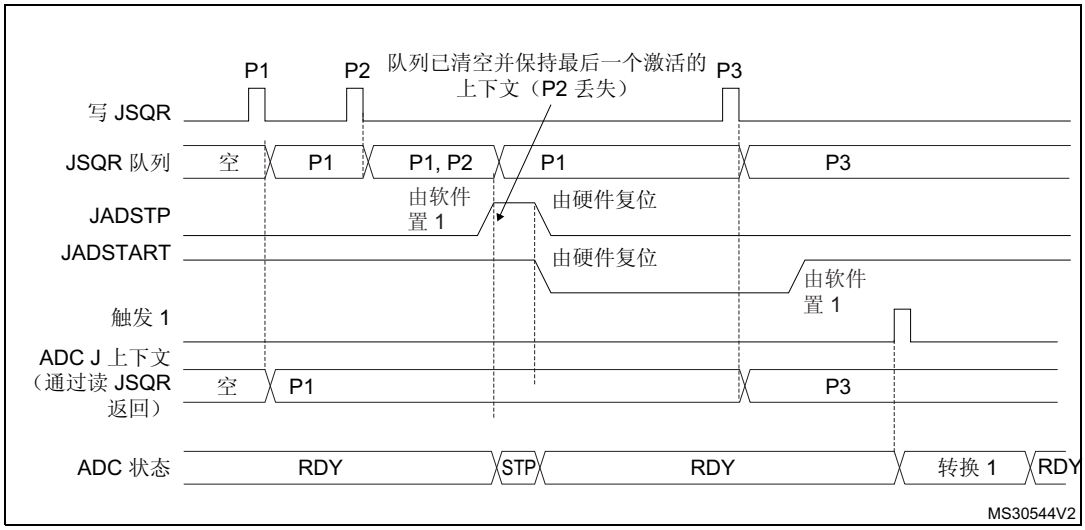
1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 216. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM = 0) —— 正在进行转换时发生 JADSTP 并出现新触发



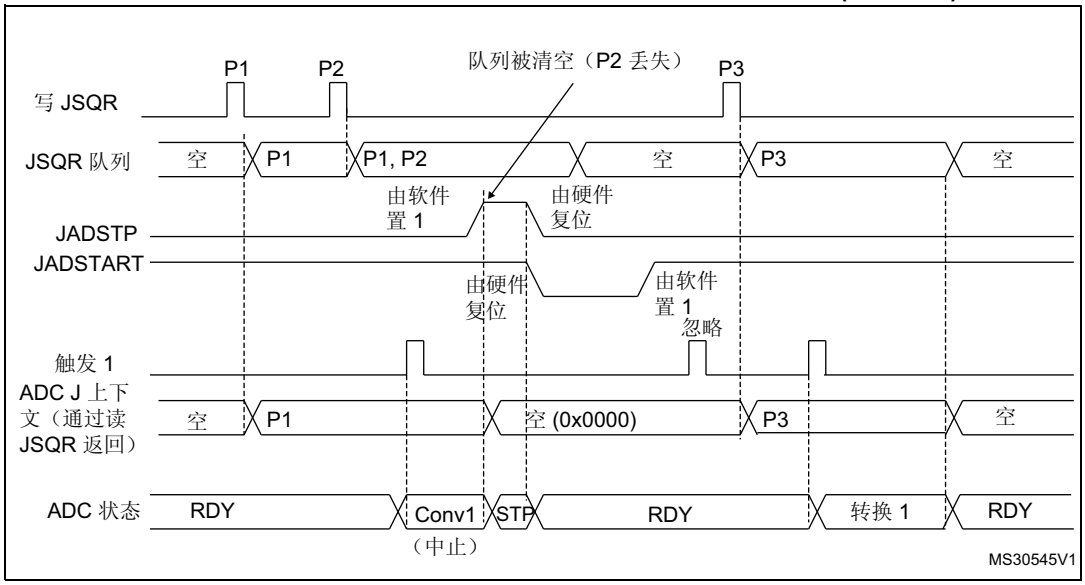
1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 217. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM = 0) —— 在正在进行转换的范围之外发生 JADSTP



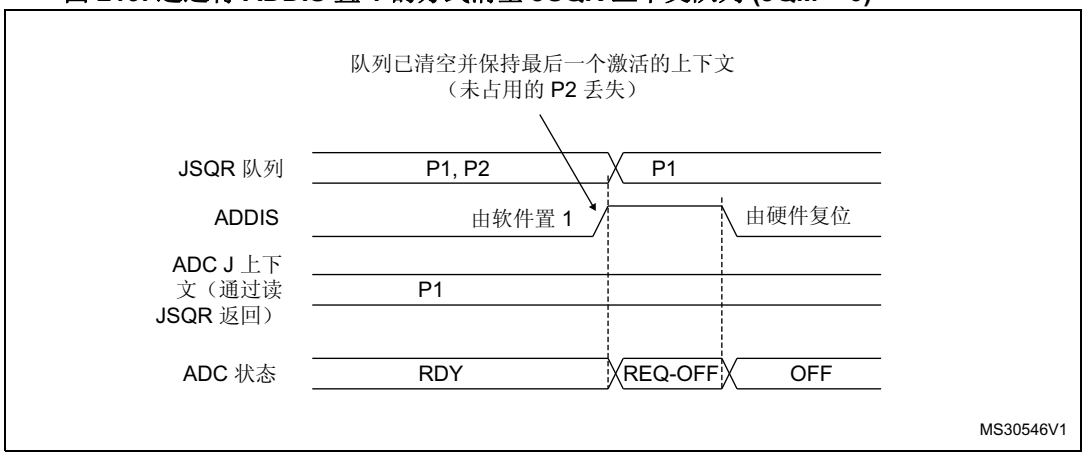
1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 218. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM = 1)



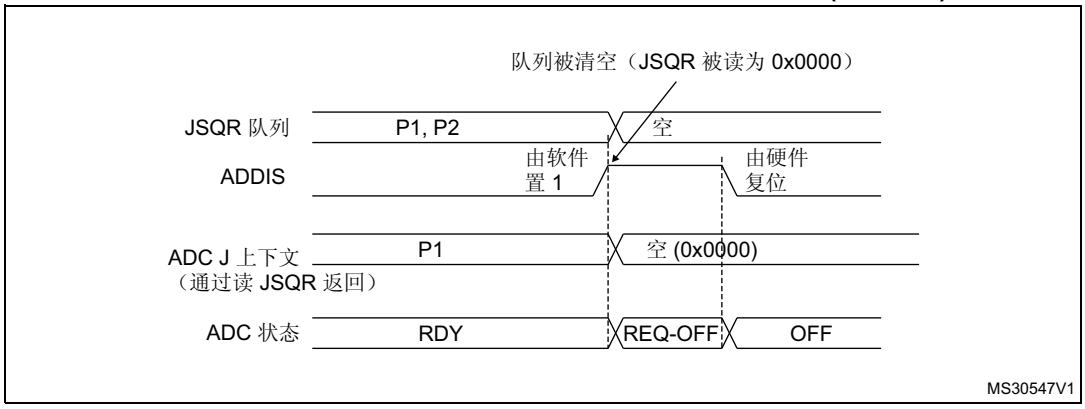
1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 219. 通过将 ADDIS 置 1 的方式清空 JSQR 上下文队列 (JQM = 0)



1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 220. 通过将 ADDIS 置 1 的方式清空 JSQR 上下文队列 (JQM = 1)



1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

上下文队列：在队列为空时启动 ADC

要在队列为空时启动 ADC 操作，必须按照以下程序进行操作，以免在 ADC 初始化时无法获悉第一个上下文。此程序仅在 JQM 位复位的情况下适用：

5. 写入空 JSQR，JEXTEN 不等于 0（否则会触发软件转换）
6. 将 JADSTART 置 1
7. 将 JADSTP 置 1
8. 等待，直至 JADSTART 复位
9. 将 JADSTART 置 1。

禁止队列

可通过将 ADC_CFGR 寄存器中的 JQDIS 位置 1 的方式禁止队列。

26.4.22 可编程分辨率 (RES)——快速转换模式

可通过降低 ADC 分辨率来执行快速转换。

通过对控制位 RES[1:0] 进行编程，可将分辨率配置为 12 位、10 位、8 位或 6 位。*图 225*、*图 226*、*图 227* 和*图 228* 显示了转换结果格式与分辨率和数据对齐方式的对应关系。

对于不要求使用高数据精度的应用，分辨率越低，转换时间越短，降低分辨率可缩短逐次逼近步骤所需的转换时间，如*表 209* 所示。

表 209. T_{SAR} 与分辨率的对应关系

RES（位）	T_{SAR}（ADC 时钟周期）	F_{ADC} = 30 MHz 时的 T_{SAR} (ns)	T_{CONV}（ADC 时钟周期）（采样时间 = 2.5 个 ADC 时钟周期）	F_{ADC} = 30 MHz 时的 T_{CONV} (ns)
12	12.5 个 ADC 时钟周期	416.67 ns	15 个 ADC 时钟周期	500.0 ns
10	10.5 个 ADC 时钟周期	350.0 ns	13 个 ADC 时钟周期	433.33 ns
8	8.5 个 ADC 时钟周期	203.33 ns	11 个 ADC 时钟周期	366.67 ns
6	6.5 个 ADC 时钟周期	216.67 ns	9 个 ADC 时钟周期	300.0 ns

26.4.23 转换结束、采样阶段结束（EOC、JEOC、EOSMP）

每次出现常规转换结束 (EOC) 事件和注入转换 (JEOC) 事件时，ADC 都会通知应用。

新的常规转换数据出现在 ADC_DR 寄存器中后，ADC 会立即将 EOC 标志置 1。如果 EOCIE 位置 1，可产生中断。EOC 标志可通过由软件向其写入 1 或读取 ADC_DR 的方式来清零。

新的注入转换数据出现在 ADC_JDRy 寄存器中后，ADC 会立即将 JEOC 标志置 1。如果 JEOCIE 位置 1，可产生中断。JEOC 标志可通过由软件向其写入 1 或读取相应 ADC_JDRy 寄存器的方式来清零。

ADC 还通过将状态位 EOSMP 置 1 来指示采样阶段结束（仅限常规转换）。EOSMP 标志可通过由软件向其写入 1 的方式来清零。如果 EOSMPIE 位置 1，可产生中断。

26.4.24 转换序列结束（EOS、JEOS）

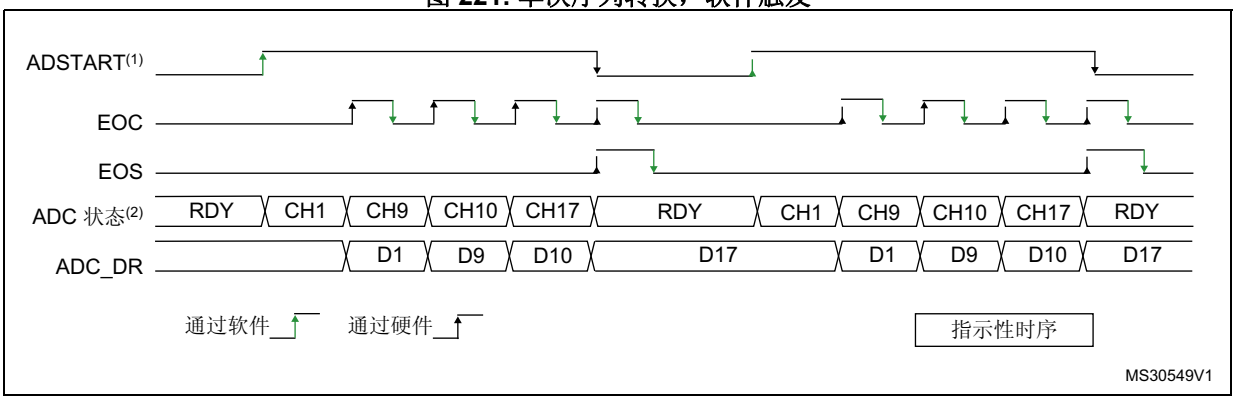
每次出现常规序列结束 (EOS) 事件和注入序列结束 (JEOS) 事件时，ADC 都会通知应用。

常规转换序列的最后一个数据出现在 ADC_DR 寄存器中时，ADC 会立即将 EOS 标志置 1。如果 EOSIE 位置 1，可产生中断。EOS 标志可通过由软件向其写入 1 的方式来清零。

注入转换序列的最后一数据完成时，ADC 会立即将 JEOS 标志置 1。如果 JEOSIE 位置 1，可产生中断。JEOS 标志可通过由软件向其写入 1 的方式来清零。

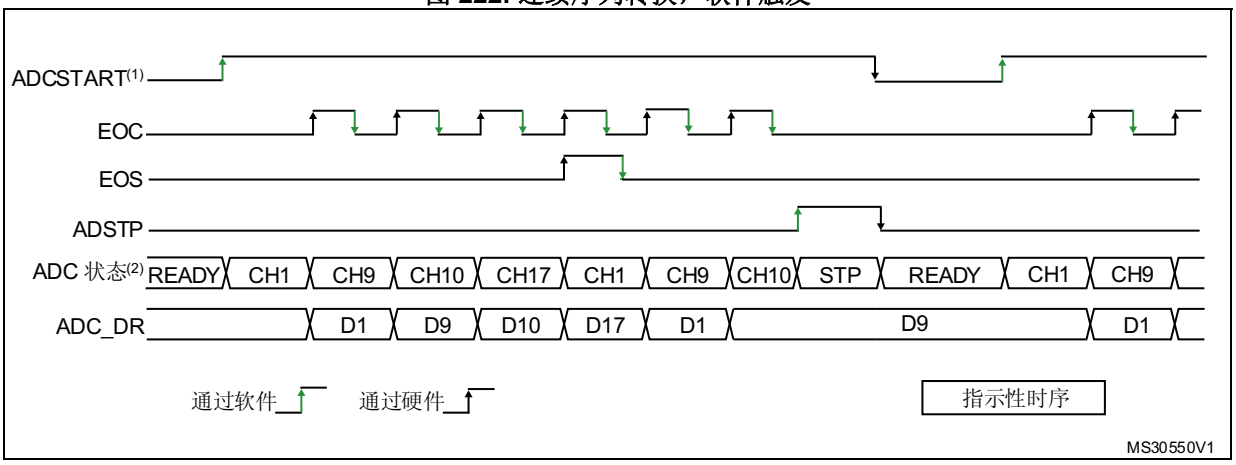
26.4.25 时序图示例（单次模式/连续模式，硬件/软件触发）

图 221. 单次序列转换，软件触发



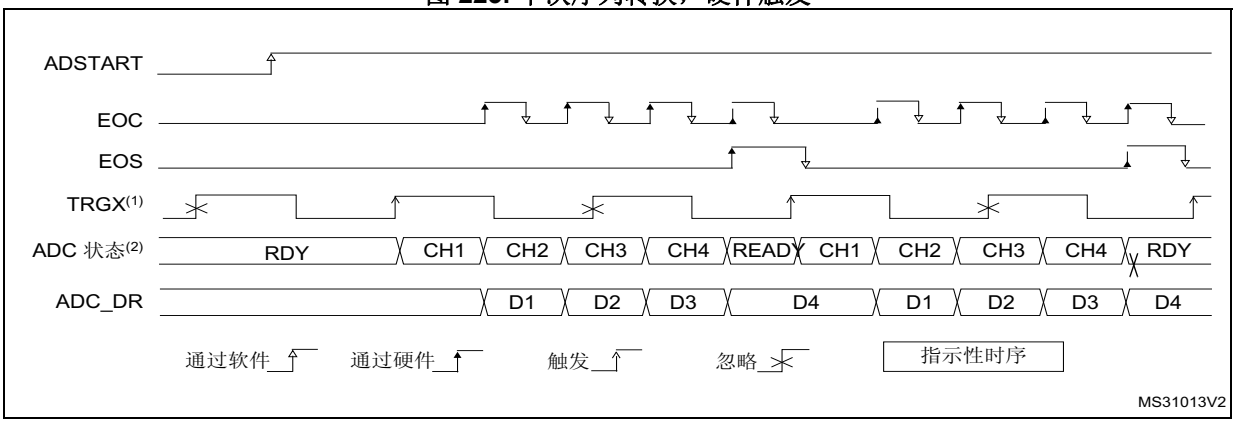
1. EXTEN = 0x0，CONT = 0
2. 所选通道 = 1、9、10、17；AUTDLY = 0。

图 222. 连续序列转换，软件触发



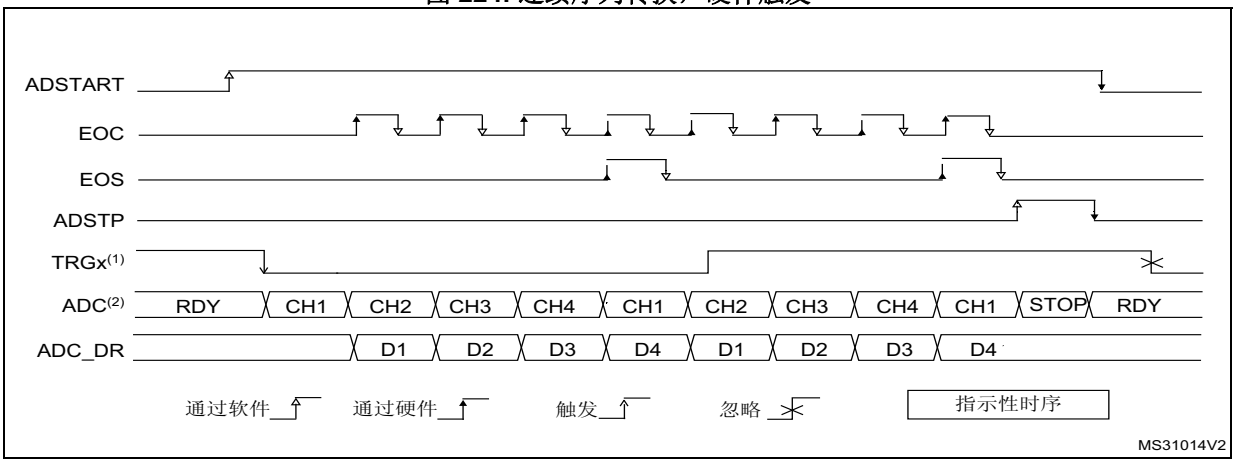
1. EXTEN = 0x0，CONT = 1
2. 所选通道 = 1、9、10、17；AUTDLY = 0。

图 223. 单次序列转换，硬件触发



1. 选择 TRGx（过频）作为触发源，EXTEN = 01，CONT = 0
2. 所选通道 = 1、2、3、4；AUTDLY = 0。

图 224. 连续序列转换，硬件触发



1. 选择 TRGx 作为触发源，EXTEN = 10，CONT = 1
2. 所选通道 = 1、2、3、4；AUTDLY = 0。

26.4.26 数据管理

数据寄存器、数据对齐和偏移（ADC_DR、OFFSET、OFFSET_CH、ALIGN）

数据和对齐

每次常规转换通道结束时（发生 EOC 事件时），转换后数据的结果都会存储在宽度为 16 位的 ADC_DR 数据寄存器中。

每次注入转换通道结束时（发生 JEOC 事件时），转换后数据的结果都会存储在宽度为 16 位的相应 ADC_JDRy 数据寄存器中。

ADC_CFGR 寄存器中的 ALIGN 位用于选择转换后存储的数据的对齐方式。可选择左对齐和右对齐两种方式，如*图 225*、*图 226*、*图 227* 和*图 228* 所示。

特例：采用左对齐时，数据基于半字对齐，分辨率设置为 6 位时除外。当分辨率设置为 6 位时，数据基于字节对齐，如*图 227* 和*图 228* 所示。

注： *过采样模式下不支持左对齐。当 ROVSE 和/或 JOVSE 位置 1 时，忽略 ALIGN 位的值，并且 ADC 仅提供右对齐数据。*

偏移

可以通过将 ADC_OFRy 寄存器中的 OFFSET_EN 置 1，对通道应用偏移 y（y = 1、2、3、4）。应用偏移的通道会编程到 ADC_OFRy 寄存器的 OFFSET_CH[4:0] 位中。这种情况下，转换后的值会减去写入到 OFFSET[11:0] 位中的用户自定义偏移。结果可能是负值，因此读取的数据为有符号值，SEXT 位代表扩展符号值。

注： *过采样模式下不支持偏移校正。如果 ROVSE 和/或 JOVSE 位置 1，ADC_OFRy 寄存器中 OFFSET_EN 位的值会被忽略（视为复位）。*

表 212 介绍了如何对模拟看门狗 1 的所有可能的分辨率进行比较。

表 210. 偏移计算与数据分辨率

分辨率（位 RES[1:0]）	原始转换数据与偏移相减		结果	备注
	原始转换数据，左对齐	偏移		
00：12 位	DATA[11:0]	OFFSET[11:0]	有符号 12 位数据	-
01：10 位	DATA[11:2]，00	OFFSET[11:0]	有符号 10 位数据	用户必须将 OFFSET[1:0] 配置为“00”
10：8 位	DATA[11:4]，0000	OFFSET[11:0]	有符号 8 位数据	用户必须将 OFFSET[3:0] 配置为“0000”
11：6 位	DATA[11:6]，000000	OFFSET[11:0]	有符号 6 位数据	用户必须将 OFFSET[5:0] 配置为“000000”

从对应于通道“i”的 ADC_DR（常规通道）或 ADC_JDRy（注入通道，y = 1、2、3、4）读取数据时：

- 如果相应通道的其中一个偏移已使能（位 OFFSET_EN = 1），则读取的数据为有符号数据。
- 如果此通道的四个偏移均未使能，则读取的数据为无符号数据。

图 225、*图 226*、*图 227* 和*图 228* 中显示了有符号数据和无符号数据的对齐方式。

图 225. 右对齐（偏移禁止，无符号值）

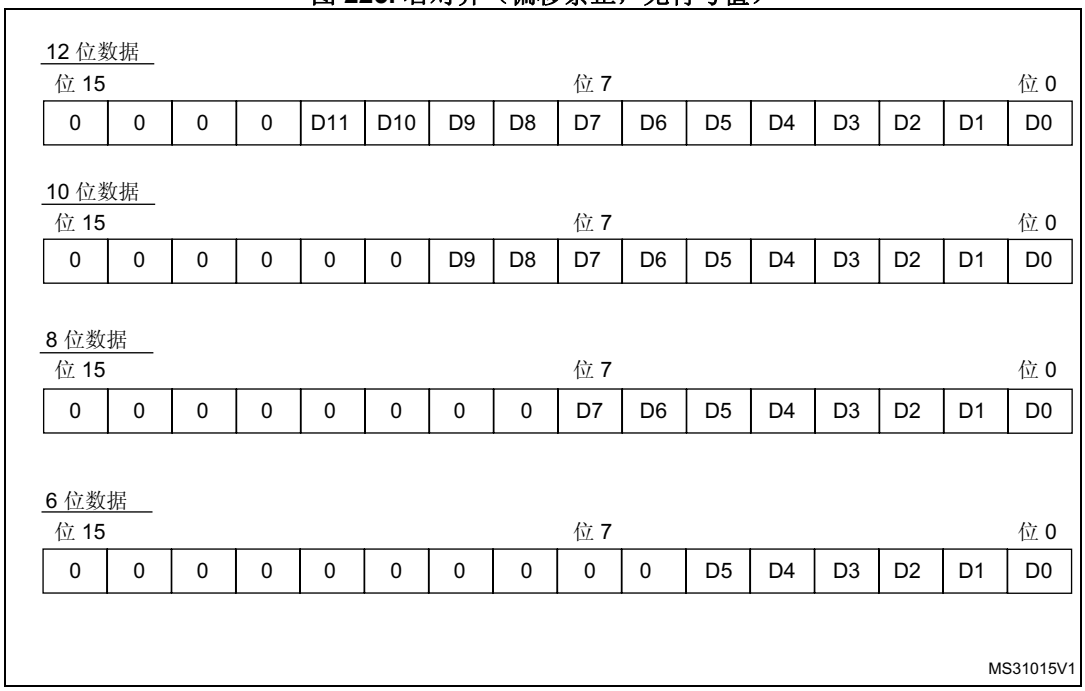


图 226. 右对齐（偏移使能，有符号值）

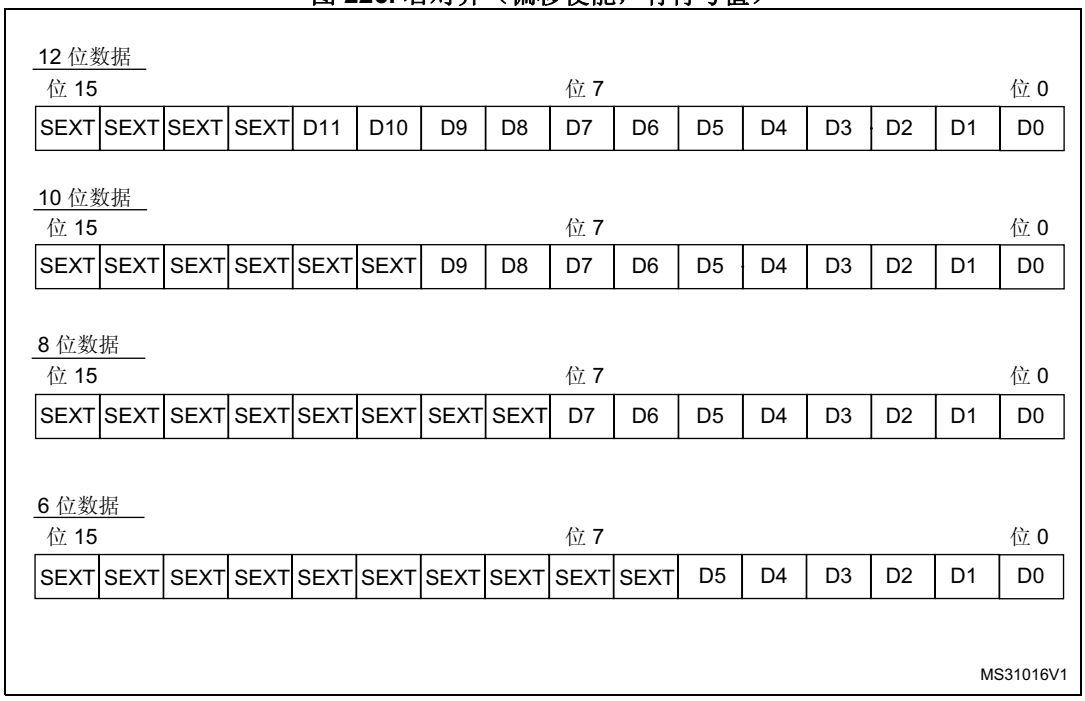


图 227. 左对齐（偏移禁止，无符号值）

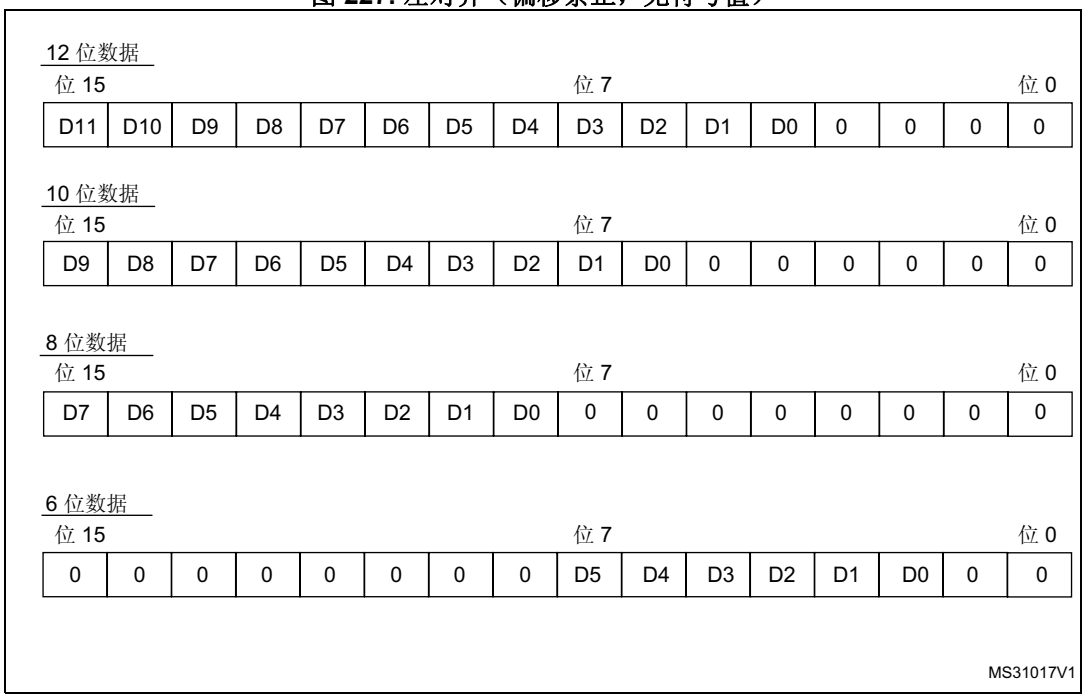
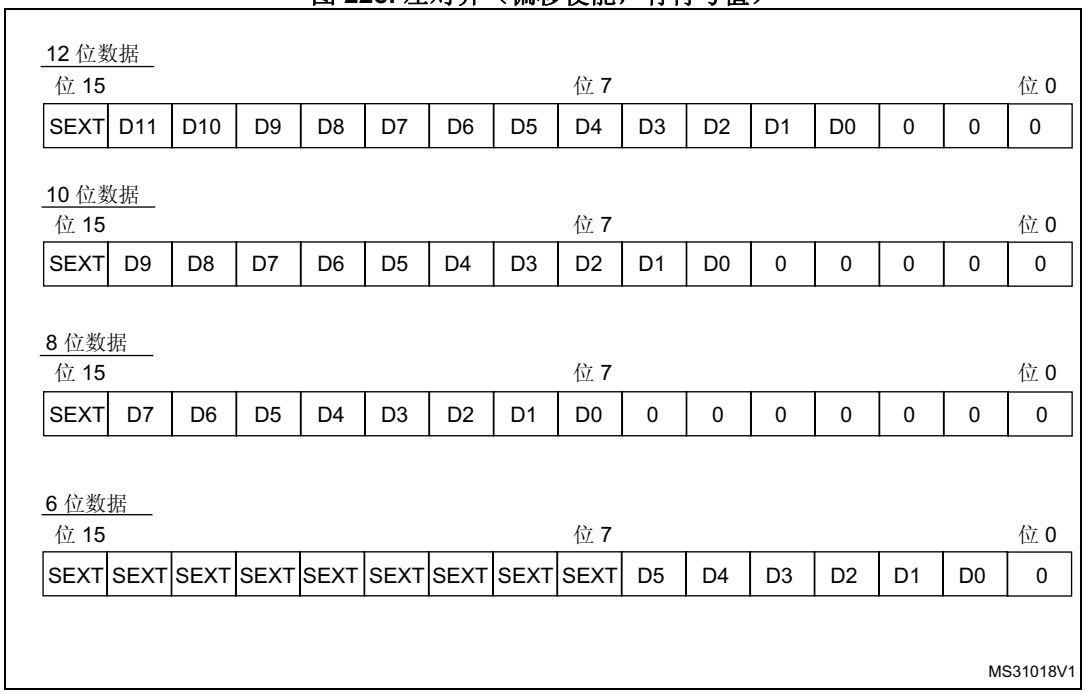


图 228. 左对齐（偏移使能，有符号值）



偏移补偿

如果偏移操作期间 ADC_OFRy 寄存器中的 SATEN 位置 1，则数据为无符号数据。所有偏移数据在 0x000 饱和（12 位模式下）。OFFSETPOS 位置 1 时，偏移方向为正向，数据在 0xFFF 饱和（12 位模式下）。在 8 位模式下，数据分别在 0x00 和 0xFF 饱和。

模拟看门狗比较在偏移补偿之前执行。

ADC 溢出（OVR、OVRMOD）

如果常规转换后的数据未在 ADC_DR FIFO（三个阶段）溢出之前（由 CPU 或 DMA）读取，会由溢出标志 (OVR) 进行指示。

如果新转换在 ADC_CR 寄存器 FIFO 已满的情况下完成，则 OVR 标志置 1。如果 OVRIE 位置 1，则会生成中断。

如果发生溢出情况，ADC 仍会保持工作状态并可继续进行转换，除非通过软件将 ADSTP 置 1，从而停止并复位序列。由于 ADC_DR FIFO 具有三个阶段，因此 FIFO 中最多存储三个数据。

OVR 标志可通过由软件向其写入 1 的方式来清零。

可对控制位 OVRMOD 进行编程，从而配置发生溢出事件时是要保留数据还是要覆盖数据：

- OVRMOD = 0：溢出事件会保留数据寄存器的数据，防止其被覆盖：会保留最多达到 ADC_DR FIFO 深度（三个阶段）的原数据，并会丢弃新的转换结果。在此模式下，将禁止 ADC_DR FIFO。如果 FIFO 已满，尽管可以继续执行转换，但生成的数据还是会被丢弃。EOC 通过读取 ADC_DR 寄存器的方式清零。但是，FIFO 仍可包含之前转换的数据。
- OVRMOD = 1：数据寄存器会由上一次转换结果覆盖，之前未读取的数据会丢失。在此模式下，将禁止 ADC_DR FIFO。如果 OVR 保持为 1，可继续正常执行转换，并且 ADC_DR 寄存器将始终包含最新的转换数据。

图 229. 溢出示例 (OVRMOD = 0)

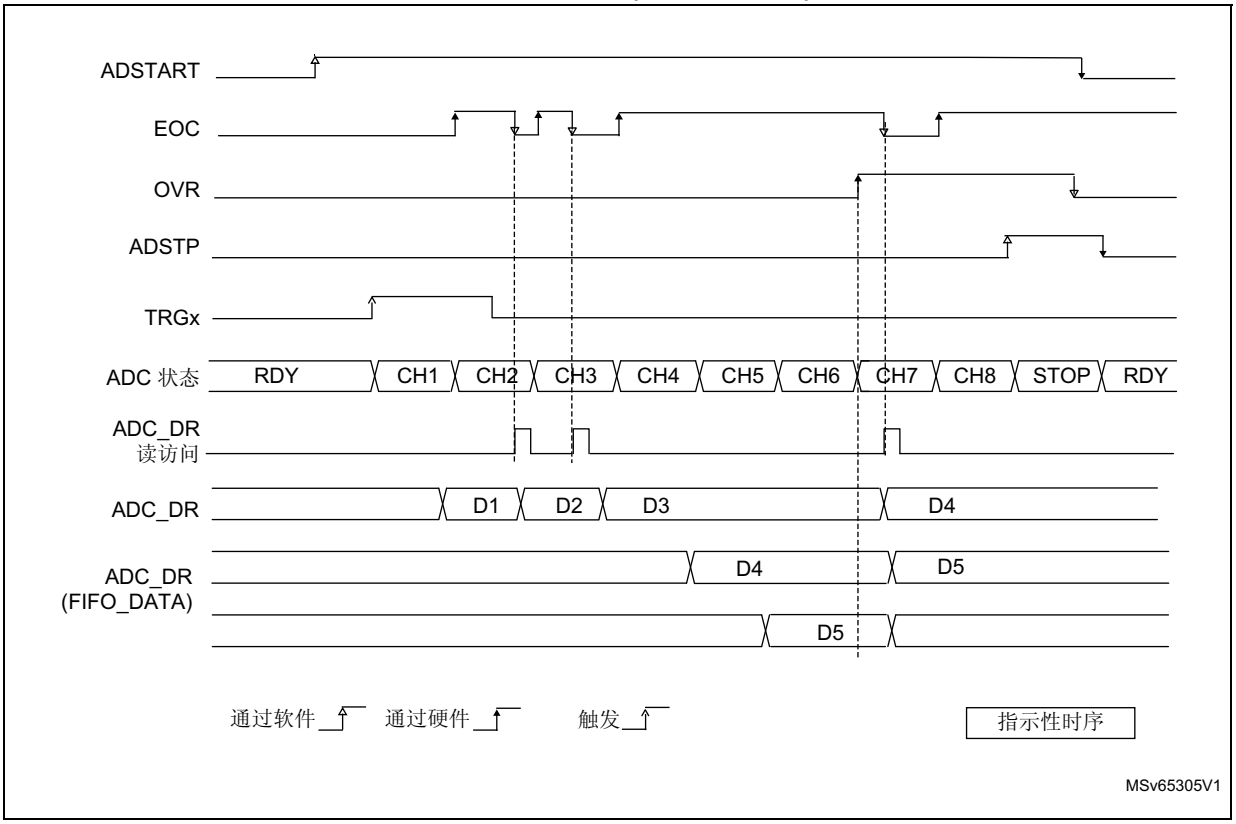
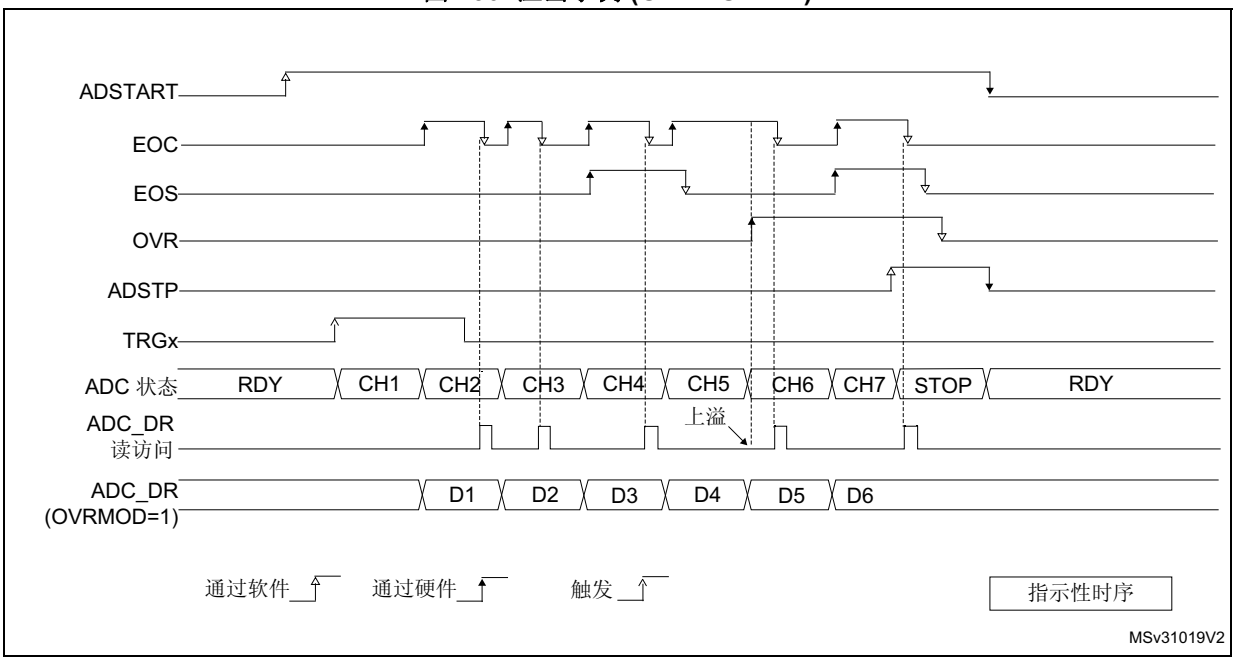


图 230. 溢出示例 (OVRMOD = 1)



注：　　由于四条注入通道均有专用的数据寄存器，因此不会对注入通道进行溢出检测。

在不使用 DMA 的情况下管理转换序列

如果转换过程足够慢，则可使用软件来处理转换序列。在这种情况下，软件必须使用 EOC 标志及其相关中断来处理各个数据。每当转换结束时，EOC 都会置 1，并且可以读取 ADC_DR 寄存器。OVRMOD 必须配置为 0 才能将溢出事件或 FIFO 上溢作为错误进行管理。

在不使用 DMA 且不发生溢出的情况下管理转换

ADC 在转换一个或多个通道时不是每次都读取数据的情况下，这可能会很有用（例如，存在模拟看门狗时）。在这种情况下，OVRMOD 位必须配置为 1，OVR 标志应被软件忽略。溢出事件不会阻止 ADC 继续进行转换，ADC_DR 寄存器始终包含最新的转换。

使用 DMA 管理转换

由于转换得出的通道值会存储到唯一的数据寄存器中，因此，对于多个通道的转换，使用 DMA 非常有帮助。这样可以避免丢失在下一次写入之前还未被读出的 ADC_DR 寄存器中的数据。

若 DMA 模式已使能（ADC_CFGR 寄存器中的 DMAEN 位置 1），则会在每个通道转换后生成 DMA 请求。这样便可将转换的数据从 ADC_DR 寄存器传输到用软件选择的目标位置。

尽管如此，如果因 DMA 无法及时处理 DMA 传输请求而发生溢出 (OVR = 1)，ADC 会停止生成 DMA 请求，新转换对应的数据不会通过 DMA 进行传输。这意味着可将传输到 RAM 的所有数据都视为有效数据。

根据 OVRMOD 位的配置，可保留或覆盖数据（请参见 *ADC 溢出（OVR、OVRMOD）* 一节）。

DMA 传输请求会禁止，直至软件将 OVR 位清零。

根据应用用途的不同，推荐使用两种不同的 DMA 模式，并使用 ADC_CFGR 寄存器的 DMACFG 位配置相应的模式：

- DMA 单次模式 (DMACFG = 0)。
  如果通过编程将 DMA 设置为传输固定数目的数据，应选择此模式。
- DMA 循环模式 (DMACFG = 1)
  如果在循环模式下对 DMA 进行编程，应选择此模式。

DMA 单次模式 (DMACFG = 0)

在该模式下，每次出现新的转换数据时，ADC 都会生成 DMA 传输请求，并且在 DMA 到达最后一个 DMA 传输操作时（发生传输完成中断，请参见 DMA 部分），即使转换已再次开始，ADC 也会停止生成 DMA 请求。

DMA 传输完成后（在 DMA 控制器中配置的所有传输操作均已完成后）：

- ADC 数据寄存器的内容会冻结。
- 任何正在进行的转换都会中止，其部分结果会被丢弃。
- 不会将任何新的 DMA 请求发送到 DMA 控制器。如果仍存在已开始的转换，这样可避免生成溢出错误。
- 扫描序列会停止并复位。
- DMA 会停止。

DMA 循环模式 (DMACFG = 1)

在该模式下，每次数据寄存器中出现新的转换数据时，ADC 都会生成 DMA 传输请求，即使 DMA 已到达最后一次 DMA 传输操作也不例外。这样可将处于循环模式的 DMA 配置为处理连续的模拟输入数据流。

26.4.27 使用 DFSDM 管理转换

ADC 转换结果可直接传送到 ΣΔ 滤波器 (DFSDM)。

在这种情况下，DFSDMCFG 位必须置 1，DMAEN 位必须清 0。

ADC 会将常规数据寄存器数据的所有 16 位传输到 DFSDM，传输完成后，会立即复位 EOC 标志。

数据必须为 16 位有符号格式：

ADC_DR[15:12] = 符号扩展

ADC_DR[11] = 符号

ADC_DR[11:0] = 数据

要在 12 位 ADC 模式下实现 16 位有符号格式，软件需要在 OFFSET_EN 置 1 后将 OFFSET[11:0] 配置为 0x800。

仅右对齐数据格式适用于 DFSDM 接口（请参见*图 226：右对齐（偏移使能，有符号值）*）。

26.4.28 动态低功耗特性

自动延迟转换模式 (AUTDLY)

ADC 会执行由 AUTDLY 配置位控制的自动延迟转换模式。自动延迟转换可用于简化软件，并可优化采用低频时钟的应用（此类应用可能存在 ADC 溢出的风险）的性能。

AUTDLY = 1 时，仅当同一组内所有之前的数据已处理完毕时，才能开始新的转换：

- 对于常规转换：ADC_DR 寄存器已读取或者 EOC 位已清零时（请参见*图 231*）。
- 对于注入转换：JEOS 位已清零时（请参见*图 232*）。

通过这种方式，可自动调整 ADC 的速度，使其适应系统读取数据的速度。

延时插入到每次常规转换后（无论 DISCEN = 0 还是 1）和每个注入转换序列后（无论 JDISCEN = 0 还是 1）。

注： *不会在注入转换序列之间插入延时，但会在最后一个序列之后插入延时。*

转换过程中，此延时期间发生的硬件触发事件（针对同一组转换）会被忽略。

注： *如果延时期间发生软件触发，则不会被忽略，仍可在该延时时间内通过将 ADSTART 或 JADSTART 位置 1 的方式重新启动转换；启动新转换之前，会由软件读取数据。*

不会在不同组的转换之间插入延时（常规转换后即进行注入转换，反之亦然）：

- 如果在常规转换的自动延时期间发生注入触发，注入转换会立即开始（请参见*图 232*）。
- 注入序列完成后，ADC 会等待上一常规转换的延时（如果未结束），然后才会启动新的常规转换（请参见*图 234*）。

自动注入模式 (JAUTO = 1) 下的操作略有不同，仅当上一注入转换序列的自动延时已结束时（JEOS 已清零时），才能开始进行新的常规转换。这是为了确保软件可在启动新序列之前读取给定序列的所有数据（请参见*图 235*）。

要在连续自动注入和自动延时组合模式下停止转换（JAUTO = 1、CONT = 1 且 AUTDLY = 1），请按照以下程序进行操作：

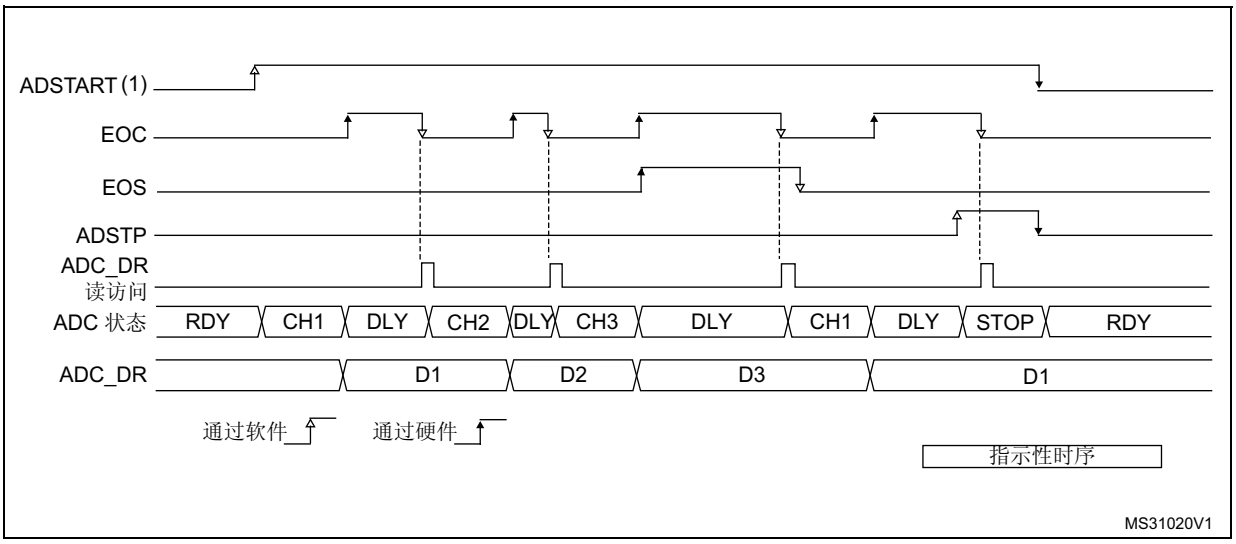
1. 等待，直至 JEOS = 1（不会重新启动任何转换）
2. 将 JEOS 清零
3. 将 ADSTP 置 1
4. 读取常规数据。

否则，如果 JEOS 在 ADSTP 已置 1 后清零，可能重新启动新的常规序列。

在 AUTDLY 模式下，如果在正在进行的常规序列期间或序列的最后一次常规转换之后的延时期间发生硬件常规触发事件，则会忽略该触发事件。但是，如果触发事件发生在该延时之后，即使发生在延时之后的注入序列期间，也会将该触发事件视为待处理事件。随后，会在注入序列的延时结束时开始转换。

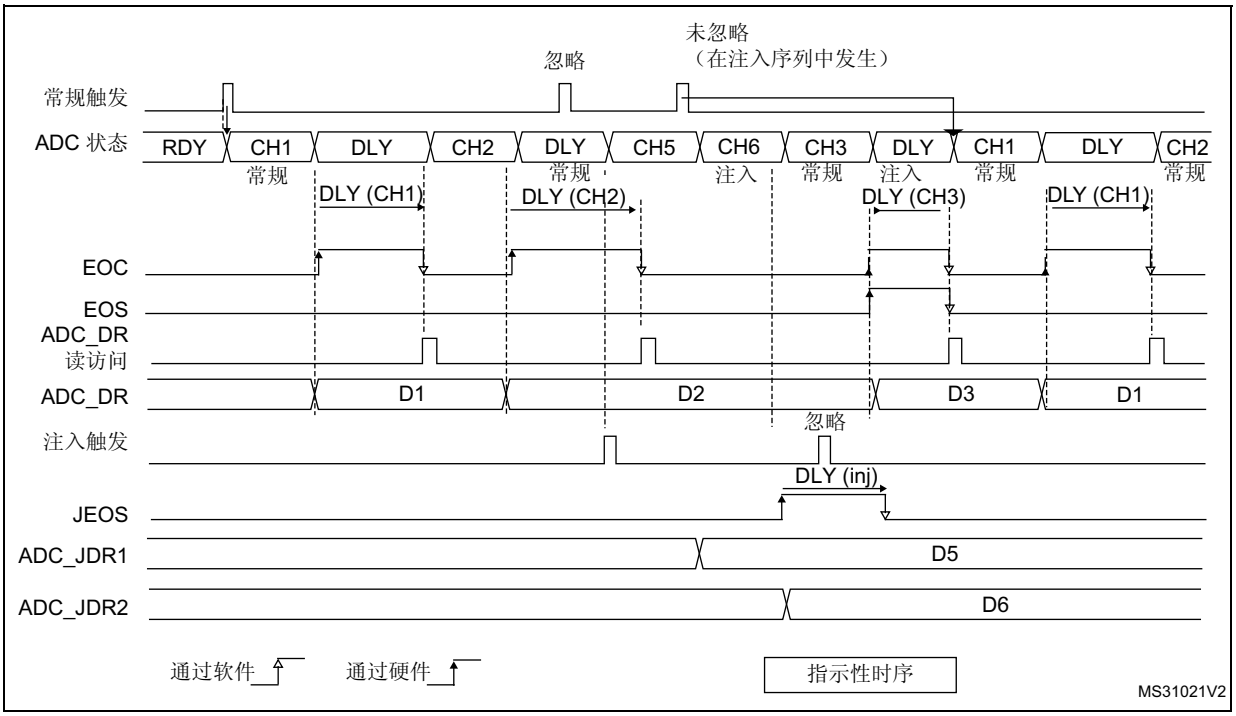
在 AUTDLY 模式下，如果在正在进行的注入序列期间或序列的最后一次注入转换之后的延时期间发生硬件注入触发事件，则会忽略该触发事件。

图 231. AUTODLY = 1，连续模式下的常规转换，软件触发



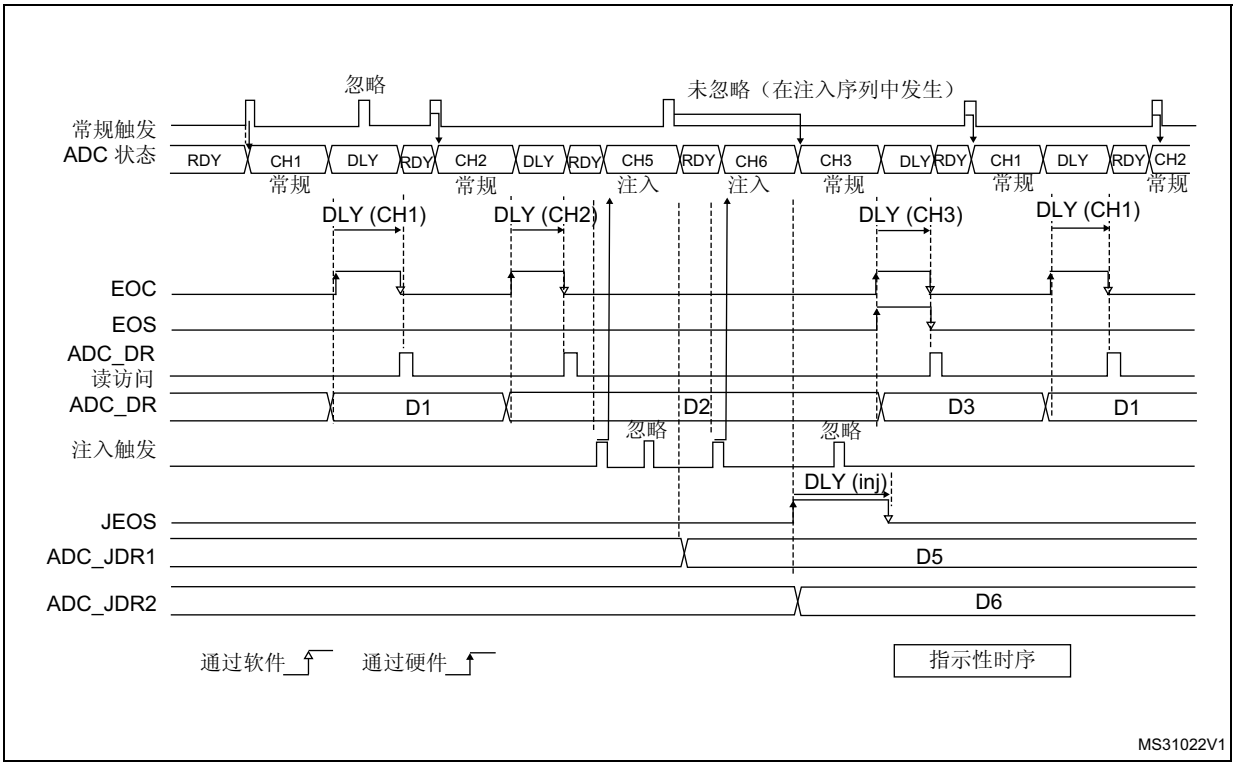
1. AUTDLY = 1
2. 常规配置：EXTEN = 0x0（软件触发），CONT = 1，CHANNELS = 1、2、3
3. 注入配置禁止

图 232. AUTODLY = 1，被注入转换中断的常规硬件转换（DISCEN = 0；JDISCEN = 0）



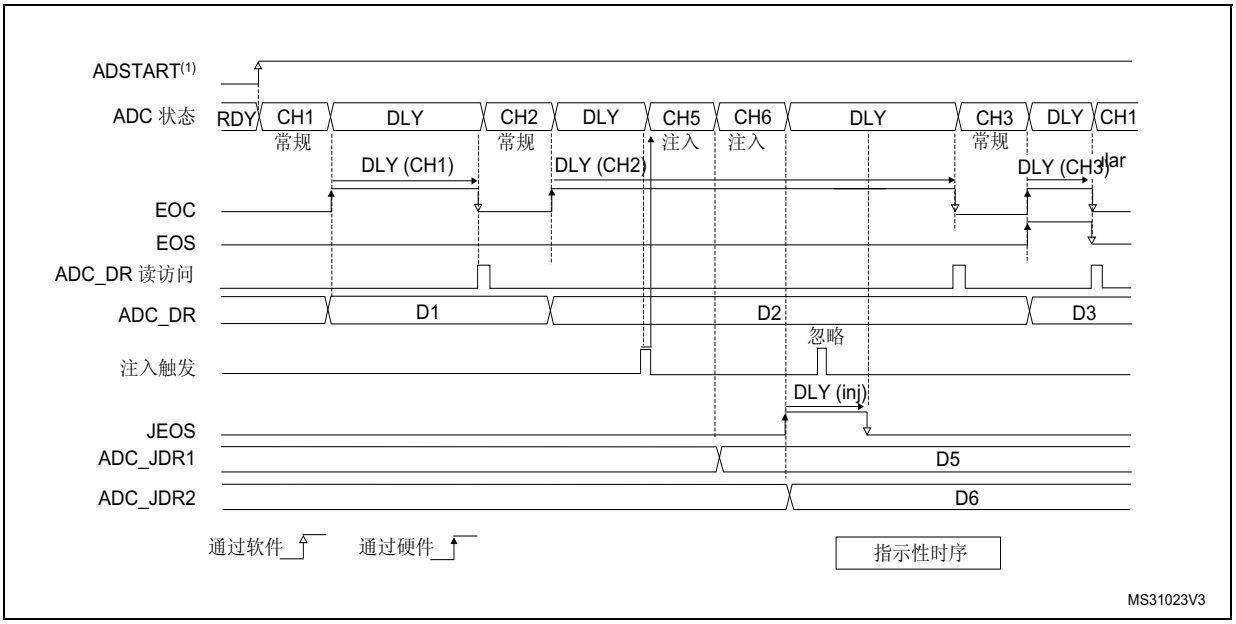
1. AUTDLY = 1
2. 常规配置：EXTEN = 0x1（硬件触发），CONT = 0，DISCEN = 0，CHANNELS = 1、2、3
3. 注入配置：JEXTEN = 0x1（硬件触发），JDISCEN = 0，CHANNELS = 5、6

图 233. AUTODLY = 1，被注入转换中断的常规硬件转换（DISCEN = 1，JDISCEN = 1）



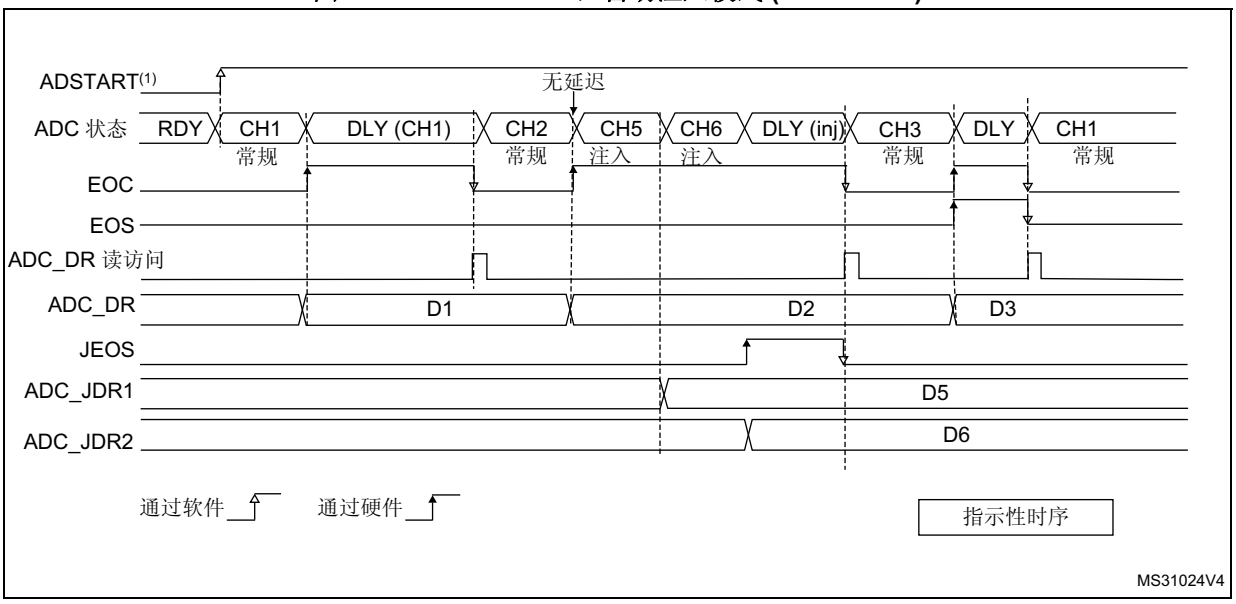
1. AUTDLY = 1
2. 常规配置：EXTEN = 0x1（硬件触发），CONT = 0，DISCEN = 1，DISCNUM = 1，CHANNELS = 1、2、3。
3. 注入配置：JEXTEN = 0x1（硬件触发），JDISCEN = 1，CHANNELS = 5、6

图 234. AUTODLY = 1，被注入转换中断的常规连续转换



1. AUTDLY = 1
2. 常规配置：EXTEN = 0x0（软件触发），CONT = 1，DISCEN = 0，CHANNELS = 1、2、3
3. 注入配置：JEXTEN = 0x1（硬件触发），JDISCEN = 0，CHANNELS = 5、6

图 235. AUTODLY = 1，自动注入模式 (JAUTO = 1)

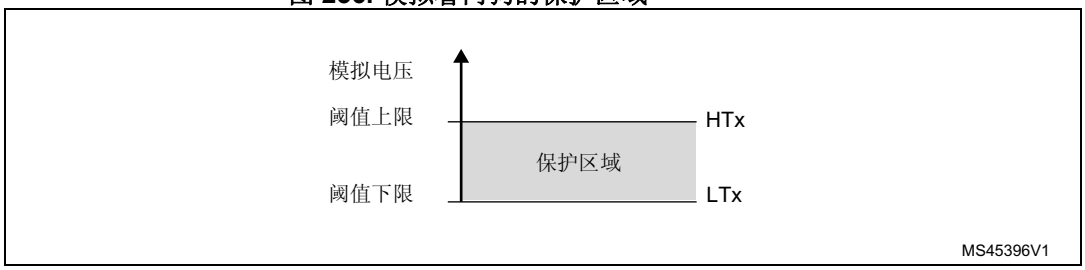


1. AUTDLY = 1
2. 常规配置：EXTEN = 0x0（软件触发），CONT = 1，DISCEN = 0，CHANNELS = 1、2
3. 注入配置：JAUTO = 1，CHANNELS = 5、6

26.4.29 模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）

三个 AWD 模拟看门狗会监测一些通道是否保持在配置的电压范围（窗口）内。

图 236. 模拟看门狗的保护区域



AWDx 标志和中断

可通过将 ADC_IER 寄存器中的 AWDxIE（x = 1、2、3）置 1 的方式分别为 3 个模拟看门狗使能中断。

AWDx（x = 1、2、3）标志可通过由软件向其写入 1 的方式来清零。

在对齐之前，会将 ADC 转换结果与阈值上限和下限进行比较。

模拟看门狗 1 说明

将 ADC_CFGR 寄存器中的 AWD1EN 位置 1，可使能 AWD 模拟看门狗 1。该看门狗监测一条已选通道或所有已使能通道(1)是否仍在配置的电压范围（窗口）内。

表 211 介绍了应如何配置 ADC_CFGR 寄存器才能在一条或多个通道上使能模拟看门狗。

表 211. 模拟看门狗通道选择

模拟看门狗保护的通道	AWD1SGL 位	AWD1EN 位	JAWD1EN 位
无	x	0	0
所有注入通道	0	0	1
所有常规通道	0	1	0
所有常规通道和注入通道	0	1	1
单个(1)注入通道	1	0	1
单个(1)常规通道	1	1	0
单个(1)常规通道或注入通道	1	1	1

1. 通过 AWD1CH[4:0] 位选择。此外，还必须将通道编程为在合适的常规序列或注入序列中进行转换。

如果 ADC 转换的模拟电压低于阈值下限或高于阈值上限，则 AWD1 模拟看门狗状态位会置 1。

这些阈值编程到模拟看门狗 1 的 ADC_TR1 寄存器的 HT1[11:0] 和 LT1[11:0] 位。如果转换的数据分辨率小于 12 位（取决于 RES[1:0] 位），由于在偏移补偿阶段前始终会在内部对完整的 12 位原始转换数据进行比较（左对齐），因此编程阈值的 LSB 必须保持清零状态。

表 212 介绍了如何对模拟看门狗 1 的所有可能的分辨率进行比较。

表 212. 模拟看门狗 1 比较

分辨率（RES[1:0] 位）	模拟看门狗比较对象：		备注
	原始转换数据，左对齐	阈值	
00：12 位	DATA[11:0]	LT1[11:0] 和 HT1[11:0]	-
01：10 位	DATA[11:2]，00	LT1[11:0] 和 HT1[11:0]	用户必须将 LT1[1:0] 和 HT1[1:0] 配置为 00
10：8 位	DATA[11:4]，0000	LT1[11:0] 和 HT1[11:0]	用户必须将 LT1[3:0] 和 HT1[3:0] 配置为 0000
11：6 位	DATA[11:6]，000000	LT1[11:0] 和 HT1[11:0]	用户必须将 LT1[5:0] 和 HT1[5:0] 配置为 000000

看门狗 1 的模拟看门狗滤波器

如果 ADC 配置为仅有一个输入通道（不允许在扫描模式下选择多个通道），则可以通过 ADC_TR1 寄存器配置一个有效的 ADC 转换数据区间：

- 如果转换后的数据属于 ADC_TR1 中定义的区间，会生成 DMA 请求。
- 否则，不会发出 DMA 请求。每次转换时都更新 RDATA 寄存器。如果数据超出区间的次数超出 ADC_TR1 的 AWDFILT 位所指定的值，则 AWDx 标志将置 1，并会发出相应的中断。

模拟看门狗 2 和 3 说明

第二个和第三个模拟看门狗更加灵活，可通过编程 AWDxCH[18:0]（x = 2、3）中的相应位来保护多条已选通道。

AWDxCH[18:0]（x = 2、3）的任何位置 1 时，会使能相应的看门狗。

其分辨率被限制为 8 位，仅高 8 位阈值可编程到 HTx[7:0] 和 LTx[7:0] 中。*表 213* 介绍了如何对所有可能的分辨率进行比较。

表 213. 模拟看门狗 2 和 3 比较

分辨率（RES[1:0] 位）	模拟看门狗比较对象：		备注
	原始转换数据，左对齐	阈值	
00：12 位	DATA[11:4]	LTx[7:0] 和 HTx[7:0]	DATA[3:0] 与比较无关
01：10 位	DATA[11:4]	LTx[7:0] 和 HTx[7:0]	DATA[3:2] 与比较无关
10：8 位	DATA[11:4]	LTx[7:0] 和 HTx[7:0]	-
11：6 位	DATA[11:6]，00	LTx[7:0] 和 HTx[7:0]	用户必须将 LTx[1:0] 和 HTx[1:0] 配置为 00

ADCy_AWDx_OUT 信号输出生成

每个模拟看门狗都关联到一个内部硬件信号 ADCy_AWDx_OUT（y = ADC 编号，x = 看门狗编号），该信号直接连接到一些片上定时器的 ETR 输入（外部触发）。请参见片上定时器部分以了解如何选择 ADCy_AWDx_OUT 信号作为 ETR。

当关联的模拟看门狗使能时，ADCy_AWDx_OUT 会激活：

- 当受保护的转换超出编程阈值时，ADCy_AWDx_OUT 会置 1。
- 在编程阈值范围内的下一受保护转换结束后，ADCy_AWDx_OUT 会复位（如果下一受保护转换仍超出编程阈值范围，该位仍保持置 1）。
- 禁止 ADC 时（将 ADDIS 置 1 时），ADCy_AWDx_OUT 也保持复位状态。请注意，停止常规转换或注入转换（将 ADSTP 或 JADSTP 置 1）对 ADCy_AWDx_OUT 的生成没有任何影响。

注： *AWDx 标志由硬件置 1，由软件复位。AWDx 标志对生成 ADCy_AWDx_OUT 无影响（例如：如果软件未将 AWDx 标志清零，则在该标志保持为 1 时 ADCy_AWDx_OUT 可以切换）。*

图 237. ADCy_AWDx_OUT 信号生成（所有常规通道上）

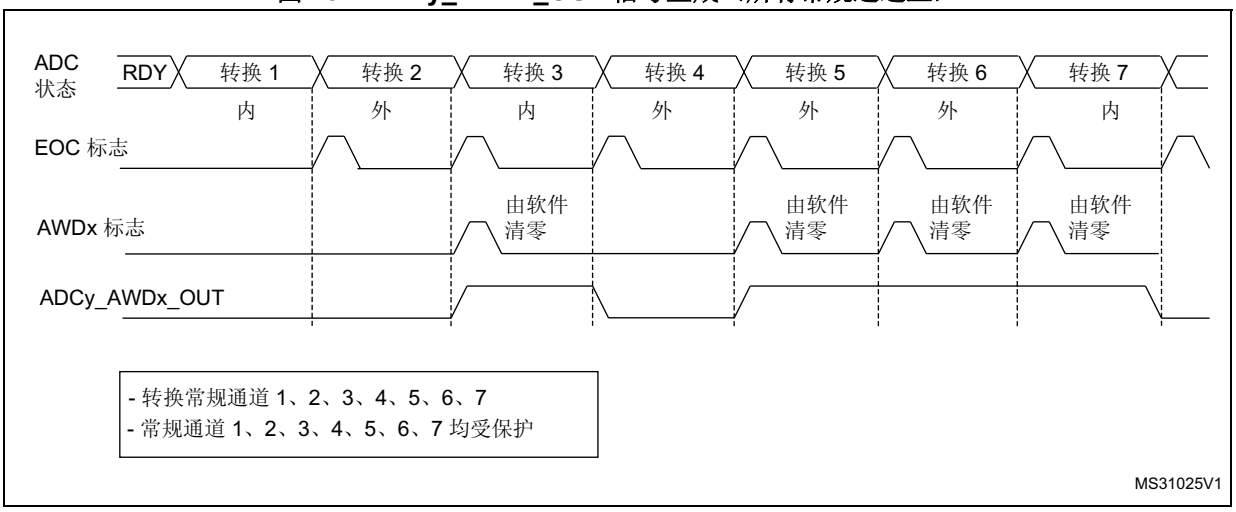


图 238. ADCy_AWDx_OUT 信号生成（AWDx 标志未通过软件清零）

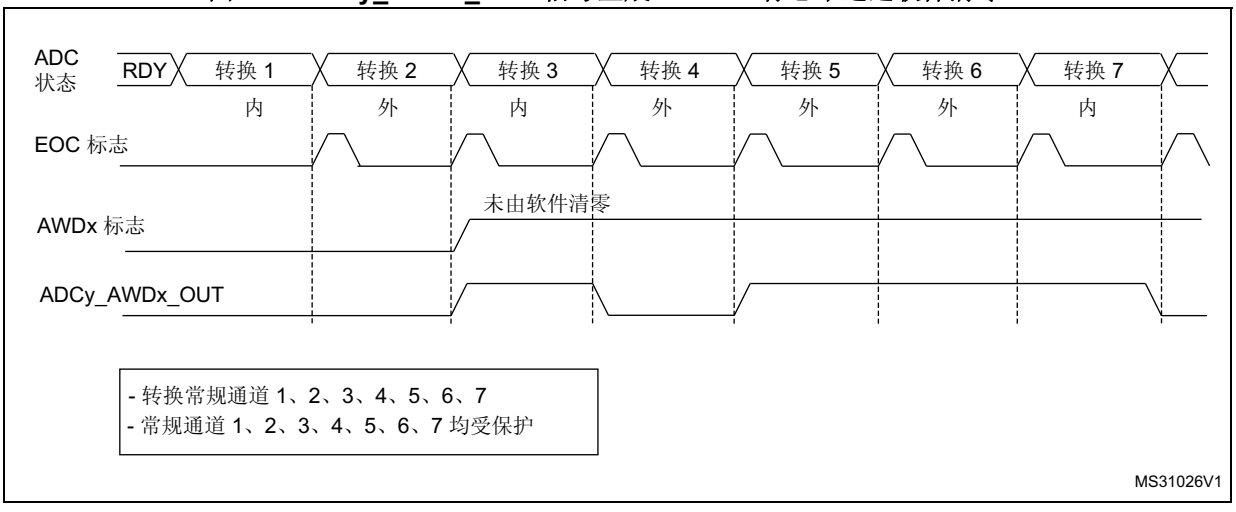


图 239. ADCy_AWDx_OUT 信号生成（单条常规通道上）

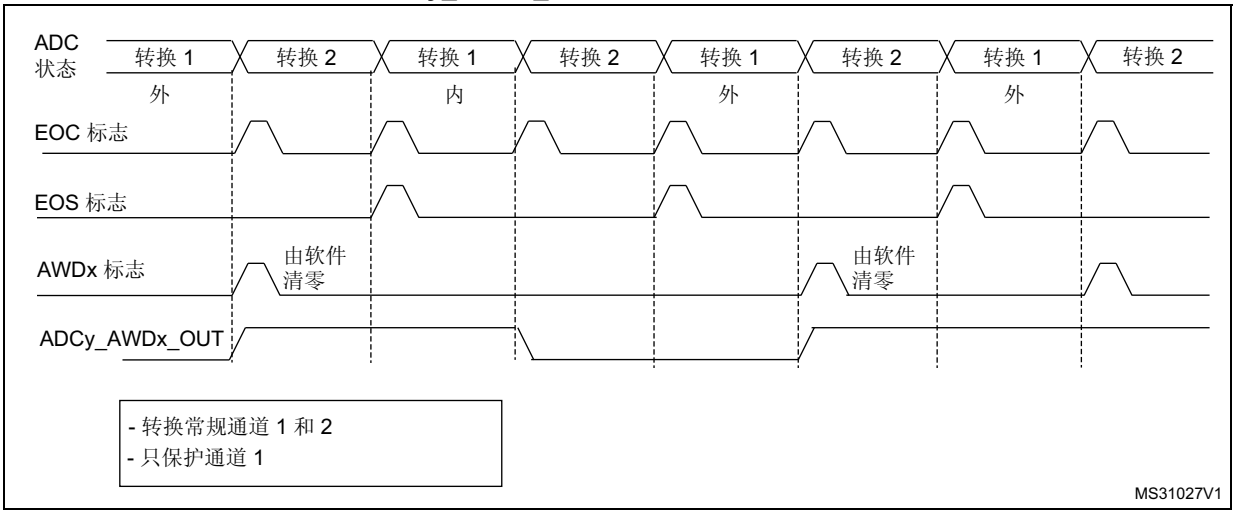
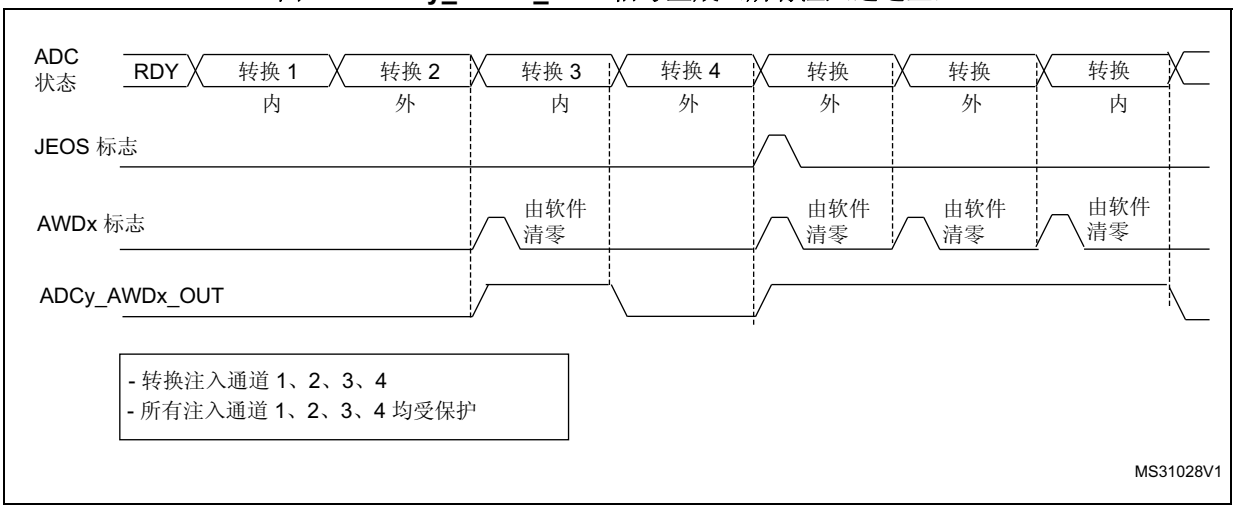


图 240. ADCy_AWDx_OUT 信号生成（所有注入通道上）



模拟看门狗阈值控制

LTx[11:0] 和 HTx[11:0] 可在进行模数转换时（即，从 ADC 内部状态转换开始到转换结束）更改。如果在 ADC 保护的通道上进行 ADC 转换期间 LTx[11:0] 和 HTx[11:0] 发生更新，则会对该转换屏蔽看门狗功能。该屏蔽在下一次转换开始时移除，从而通过下一次 ADC 转换应用模拟看门狗阈值。模拟看门狗比较在每次转换结束时执行。如果当前 ADC 数据超出新的区间，则不会发出中断和 AWDx_OUT 信号。只有在阈值更新后开始的转换结束时，才会生成中断和 AWD。如果 AWD_xOUT 已置为有效，则编程设定新的阈值不会将 AWDx_OUT 信号置为无效。

模拟看门狗与偏移补偿

如果使能了偏移补偿，则模拟看门狗会在数据补偿阶段之前比较阈值。

26.4.30 过采样器

过采样单元会进行数据预处理，以减轻 CPU 的负担。过采样单元能够处理多个转换，并计算多个转换结果的平均值，得到数据宽度增加（高达 16 位）的单个数据。

它提供的结果采用以下形式，其中的 N 和 M 可以进行调整：

$$\text{结果} = \frac{1}{M} \times \sum_{n=0}^{n=N-1} \text{转换}(t_n)$$

允许通过硬件执行以下功能：计算平均值、降低数据速率、改进 SNR 以及基本滤波。

过采样率 N 通过 ADC_CFGR2 寄存器中的 OVFS[2:0] 位进行定义，范围为 2x 到 256x。分频系数 M 通过向右移位来实现（最多可移 8 位），并且通过 ADC_CFGR2 寄存器中的 OVSS[3:0] 位定义。

求和单元可得出多达 20 位的结果（256x 12 位结果），结果会先右移。随后会截断为 16 个最低有效位，这些位使用移位后剩下的最低有效位四舍五入为最接近的数值，然后将最终得到的结果传输到 ADC_DR 数据寄存器中。

注： *如果移位后得到的中间结果超过 16 位，则会将结果按原样截断（无饱和）。*

图 241. 20 位到 16 位结果的截断过程

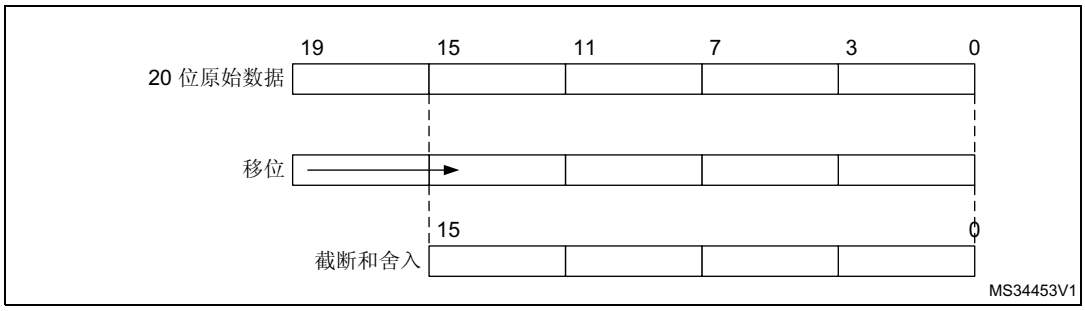


图 242 给出了从原始的 20 位累加数据到最终 16 位结果的数值处理过程示例。

图 242. 移 5 位并进行舍入的数值示例

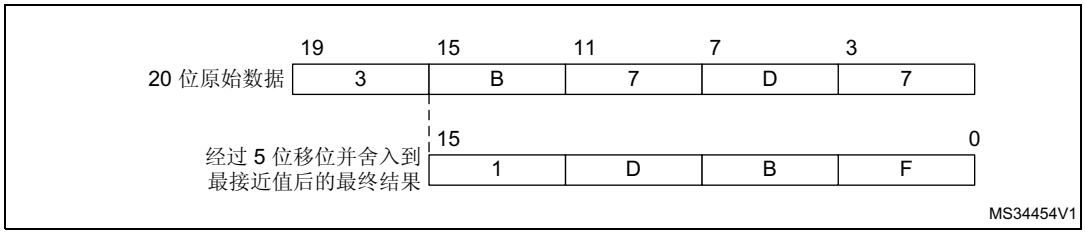


表 214 列出了原始转换数据等于 0xFFF 时对应的各种 N 和 M 组合的数据格式。

表 214. 最大输出结果与 N 和 M 的对应关系（灰色单元格表示截断的部分）

过采样率	最大原始数据	不移位 OVSS = 0000	移 1 位 OVSS = 0001	移 2 位 OVSS = 0010	移 3 位 OVSS = 0011	移 4 位 OVSS = 0100	移 5 位 OVSS = 0101	移 6 位 OVSS = 0110	移 7 位 OVSS = 0111	移 8 位 OVSS = 1000
2x	0x1FFE	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040	0x020
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080
16x	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100
32x	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800
256x	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

过采样模式下的转换时序不会发生变化：在整个过采样序列中，采样时间保持不变。每完成 N 次转换都会提供新数据，等效延迟等于 N x T_{CONV} = N x (t_{SMPL} + t_{SAR})。各标志的置 1 情况如下：

- 每个采样阶段后都会将采样阶段结束标志 (EOSMP) 置 1
- 如果过采样结果可用，每完成 N 次转换都会发生转换结束事件 (EOC)
- 过采样数据序列完成后（即 N x 序列长度次转换之后），会发生序列结束事件 (EOS)

过采样时支持的 ADC 工作模式

在过采样模式下，大部分 ADC 工作模式都会保留：

- 单次或连续转换模式
- 可由软件或触发器启动 ADC 转换
- ADC 在转换过程中停止（中止）
- 通过 CPU 或 DMA 在支持溢出检测的情况下读取数据
- 低功耗模式 (AUTDLY)
- 可编程分辨率：在这种情况下，会按照与 12 位转换相同的方式对分辨率降低的转换值（根据 ADC_CFGR 寄存器中的 RES[1:0] 位）进行累加、截断、四舍五入和移位

注： *处理过采样数据时，不可使用对齐模式。ADC_CFGR 中的 ALIGN 位会被忽略，且数据始终采用右对齐格式。*

过采样模式下不支持偏移校正。如果 ROVSE 和/或 JOVSE 位置 1，ADC_OFRy 寄存器中 OFFSET_EN 位的值会被忽略（视为复位）。

模拟看门狗

模拟看门狗功能保持不变，但存在以下区别：

- 会忽略 RES[1:0] 位，始终会使用完整的 12 位值 HT1[11:0] 和 LT1[11:0]（对于 AWD1）以及 HT2/HT3[7:0] 和 LT2/LT3[7:0] 的 8 个最高有效位值（对于 AWD2 和 AWD3）进行比较。
- 会比较 16 位过采样结果 ADC_DR[15:4]（对于 AWD1）以及 ADC_DR[15:8]（对于 AWD2 和 ADW3）的 12 个最高有效位

注： *移位位数较大时必须多加留意，因为这样会缩小比较范围。例如，如果过采样结果移了 4 位，得到 12 位右对齐数据，那么只能对 8 个数据位执行有效的模拟看门狗比较。比较是在 ADC_DR[11:4] 与 HTx[7:0] / LTx[7:0] (AWD1/2/3) 之间进行，HT1[11:8] 和 LT1[11:8] 保持复位（仅限 AWD1）。*

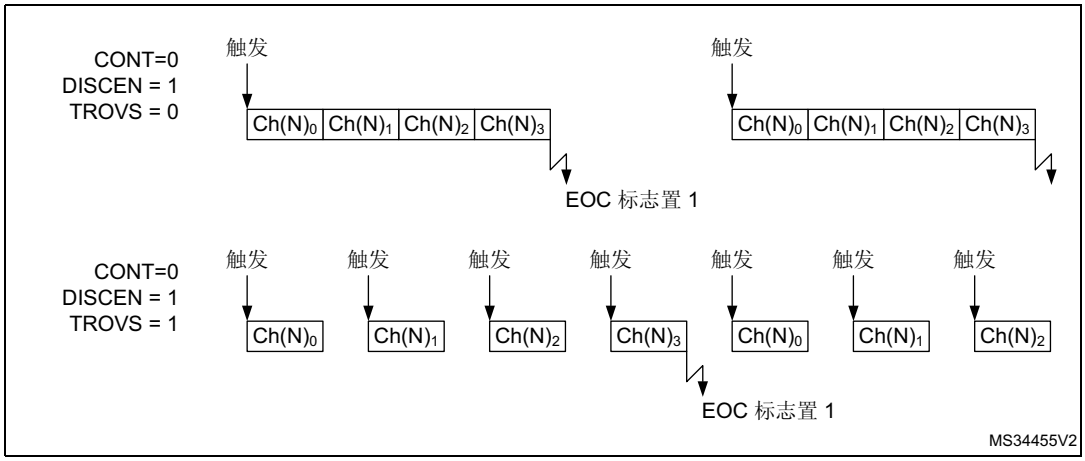
触发模式

平均值计算单元还可用于基本滤波，虽然它不是非常强大的滤波器（衰减缓慢、停止频段衰减受限），但可用作陷波滤波器，用于抑制恒定的寄生频率（通常来自电源或切换模式电源）。为此，可使用 ADC_CFGR2 中的 TROVS 位使能特定的不连续模式，以获得由用户定义、且与转换时间本身无关的过采样频率。

图 243 显示了在不连续模式下如何响应触发从而开始转换。

如果 TROVS 位置 1，则会忽略 DISCEN 位的内容，并将该位视为 1。

图 243. 已触发的常规过采样模式（TROVS 位 = 1）



过采样时的注入和常规定序器管理

在过采样模式下，注入定序器和常规定序器可以执行不同操作。如果两个定序器必须同时使用，则可为它们使能过采样并设定一些限制条件（与唯一的累加单元相关）。

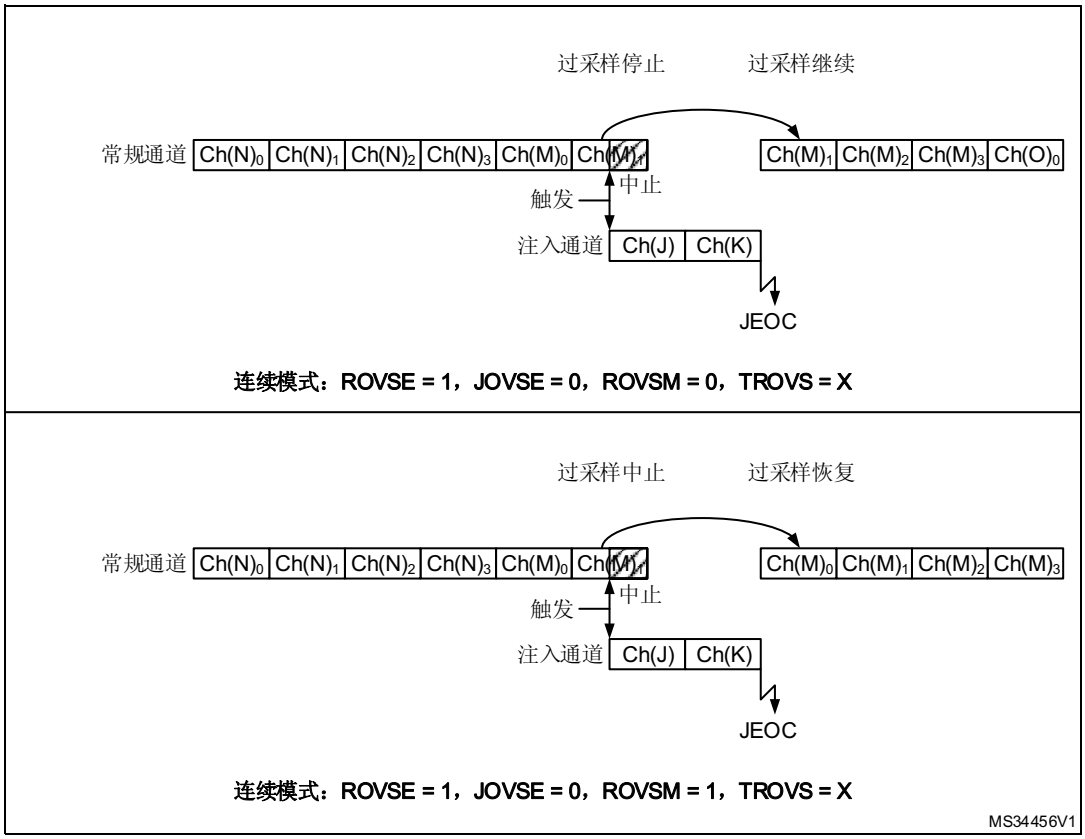
仅对常规通道进行过采样

常规过采样模式位 ROVSM 定义了常规过采样序列在被注入转换中断的情况下如何恢复。

- 在连续模式下，会从上一有效数据开始重新累加（在由于注入触发而发出的转换中止请求之前）。这样可确保在任何注入频率下均可而完成过采样（假设触发之间至少可完成一次常规转换）；
- 在恢复模式下，会从 0 开始重新累加（会忽略之前的转换结果）。该模式可确保所有用于过采样的数据在单个时隙内进行了背靠背转换。需要注意的是，注入触发周期必须超过过采样时长。如果该条件未得到满足，则无法完成过采样，常规定序器将被禁用。

图 244 以 4x 过采样率为例进行了说明。

图 244. 常规过采样模式（4x 过采样率）



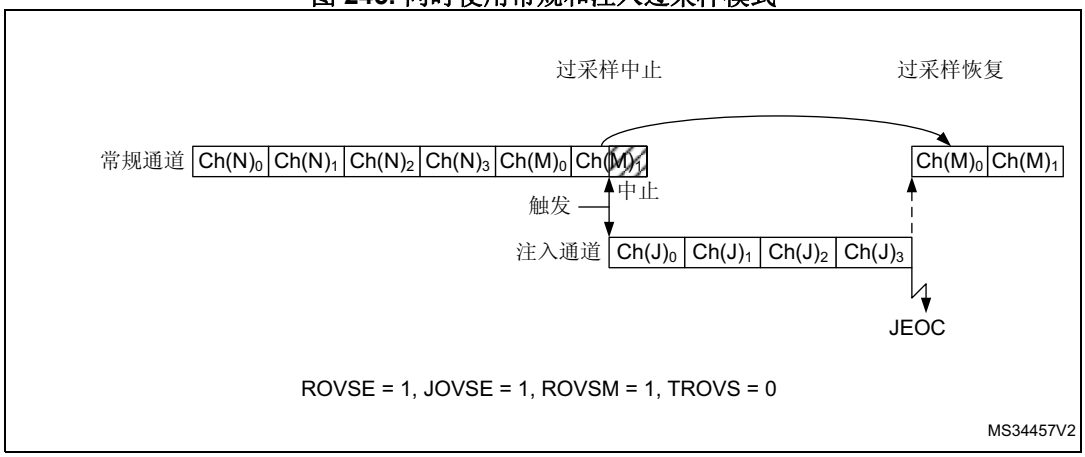
仅对注入通道进行过采样

注入过采样模式位 JOVSE 专为注入定序器中的转换使能过采样。

对常规通道和注入通道进行过采样

可以将 ROVSE 和 JOVSE 位都置 1。在这种情况下，常规过采样模式会强制进入恢复模式（忽略 ROVSM 位），如*图 245* 所示。

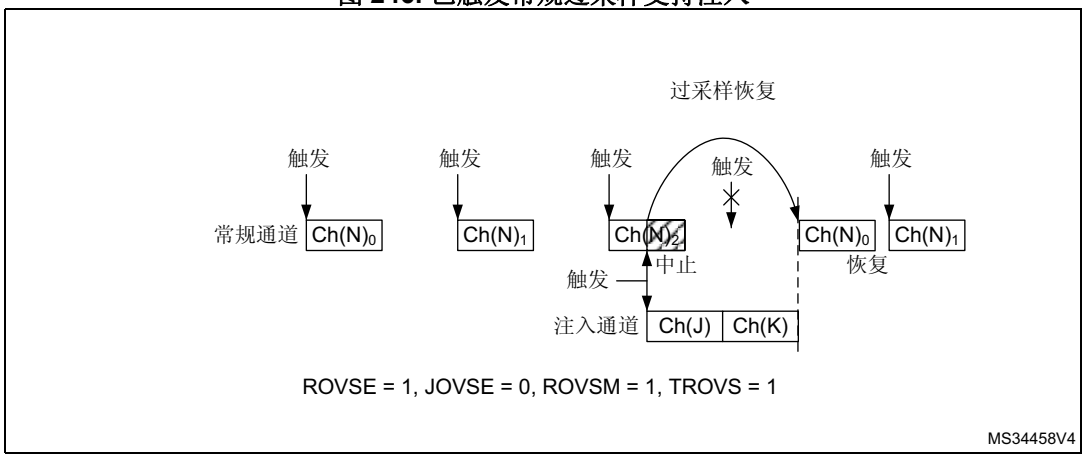
图 245. 同时使用常规和注入过采样模式



已触发常规过采样支持注入转换

已触发常规模式下支持注入转换。在这种情况下，必须禁止注入过采样模式，并且会忽略 ROVSM 位（强制进入恢复模式）。JOVSE 位必须复位。具体操作如*图 246* 所示。

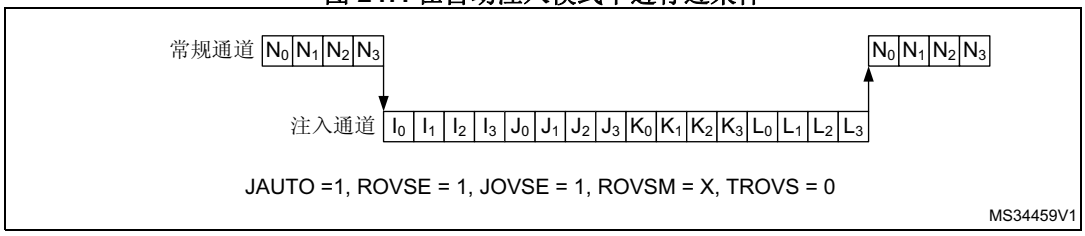
图 246. 已触发常规过采样支持注入



自动注入模式

可以对自动注入序列进行过采样，并将所有转换结果存储在寄存器中以保存 DMA 资源。使用该模式的前提条件是常规和注入过采样均激活：JAUTO = 1、ROVSE = 1 且 JOVSE = 1，不支持其他组合。在自动注入模式下，会忽略 ROVSM 位。*图 247* 显示了转换的排序方式。

图 247. 在自动注入模式下进行过采样



此外，还可以使用 TROVS 位使能触发模式。此时，ADC 必须如下配置：JAUTO = 1、DISCEN = 0、JDISCEN = 0、ROVSE = 1、JOVSE = 1 且 TROVSE = 1。

26.4.31 温度传感器

温度传感器可用于测量设备的结温 (Tj)。

温度传感器内部连接到 ADC 输入通道，这类通道用于将传感器输出电压转换为数字值（更多详细信息，请参见*表：ADC 互连*，该表在*第 26.4.2 节：ADC 引脚和内部信号*中）。不使用时可将传感器置于掉电模式。支持的温度范围：-40 °C 到 125 °C。

图 248 显示的是温度传感器与 ADC 之间连接的框图。

温度传感器的输出电压随温度线性变化。由于工艺不同，该线的偏移量取决于各个芯片（芯片之间的温度变化可达 45 °C）。

未校准的内部温度传感器更适用于对温度变量而非绝对温度进行测量的应用。为提高温度传感器测量的准确性，ST 在生产过程中将校准值存储在每个设备的系统内存中。

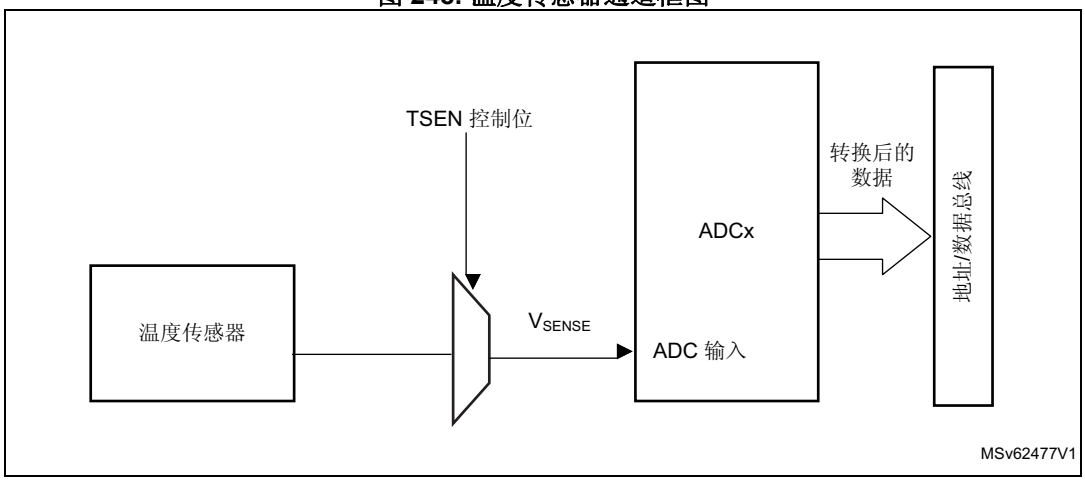
在制造过程中，会将温度传感器的校准数据和内部参考电压存储在系统存储区。随后，用户应用可读取这些数据，并使用这些数据提高温度传感器或内部参考的准确性（更多相关信息，请参见数据手册）。

温度传感器在内部连接到 ADC 输入通道，该通道用于将传感器输出电压转换为数字值。有关转换内部温度传感器时要应用的采样时间值，请参见设备数据手册的电气特性部分。

不使用时可将传感器置于掉电模式。

图 248 显示了温度传感器框图。

图 248. 温度传感器通道框图



读取温度

要使用传感器，请执行以下操作：

1. 选择连接到 V_{SENSE} 的 ADC 输入通道。
2. 设定合适的采样时间（请参见设备数据手册中的电气特性部分）。
3. 将 ADC_CCR 寄存器中的位置 1，以便将温度传感器从掉电模式中唤醒。
4. 开始 ADC 转换。
5. 读取 ADC 数据寄存器中生成的 V_{SENSE} 数据。
6. 使用以下公式计算实际温度：

$$\text{温度 (°C)} = \frac{\text{TS_CAL2_TEMP} - \text{TS_CAL1_TEMP}}{\text{TS_CAL2} - \text{TS_CAL1}} \times (\text{TS_DATA} - \text{TS_CAL1}) + \text{TS_CAL1_TEMP}$$

其中：

- TS_CAL2 是在 TS_CAL2_TEMP 下获得的温度传感器校准值。
- TS_CAL1 是在 TS_CAL1_TEMP 下获得的温度传感器校准值。
- TS_DATA 是由 ADC 转换得到的实际温度传感器输出值。

  更多关于 TS_CAL1 和 TS_CAL2 校准点的信息，请参见设备数据手册。

注： *传感器从掉电模式中唤醒需要一个启动时间，启动时间过后其才能正确输出 V_{SENSE}。ADC 在上电后同样需要一个启动时间，因此，为尽可能缩短延迟时间，应同时将 ADEN 和相关位置 1。*

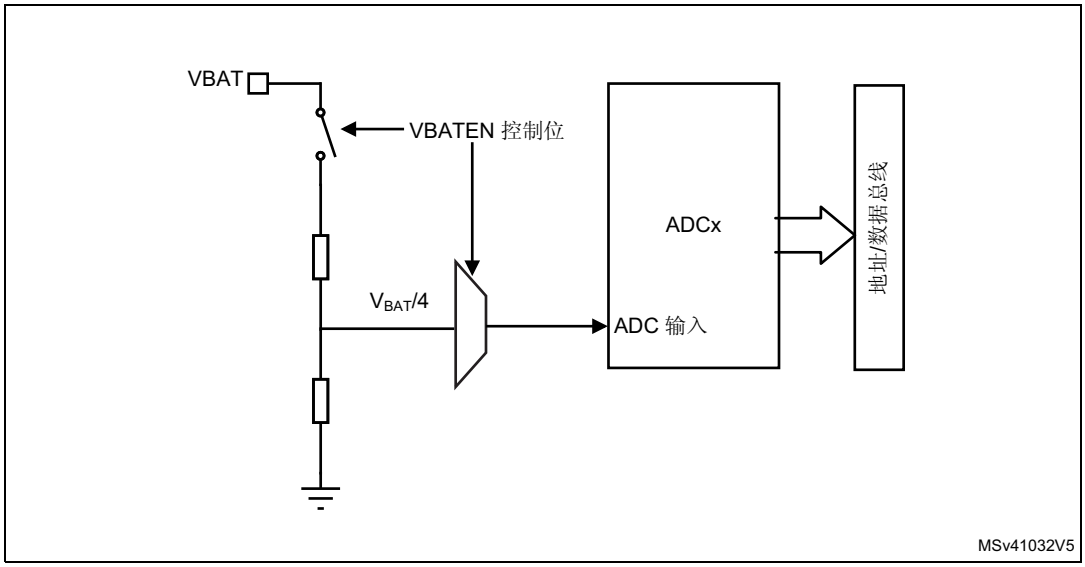
26.4.32 V_{BAT} 电源监测

ADC_CCR 寄存器中的 VBATEN 位用于切换到电池电压。由于 V_{BAT} 电压可能高于 V_{DDA}，因此 V_{BAT} 引脚需要从内部连接到桥接分配器（除以 4），以确保 ADC 正确运行。VBATEN 置 1 时，会自动使能此桥，以将 V_{BAT}/4 连接到 ADC 输入通道（更多详细信息，请参见*表：ADC 互连*，该表在*第 26.4.2 节：ADC 引脚和内部信号*中）。因此，转换出的数字值为 V_{BAT} 电压的三分之一。为防止电池出现意外的电能消耗，建议仅在必要时为 ADC 转换使能桥接分配器。

有关转换 V_{BAT}/4 电压时要应用的采样时间值，请参见设备数据手册的电气特性部分。

图 249 显示了 V_{BAT} 传感特性的框图。

图 249. V_{BAT} 通道框图



1. 必须将 VBATEN 位置 1 才能为 V_{BAT}/4 使能内部通道转换。

26.4.33 监测内部参考电压

可以通过监测内部参考电压 (V_{REFINT}) 来获得用于评估 ADC V_{REF+} 电压的参考值。

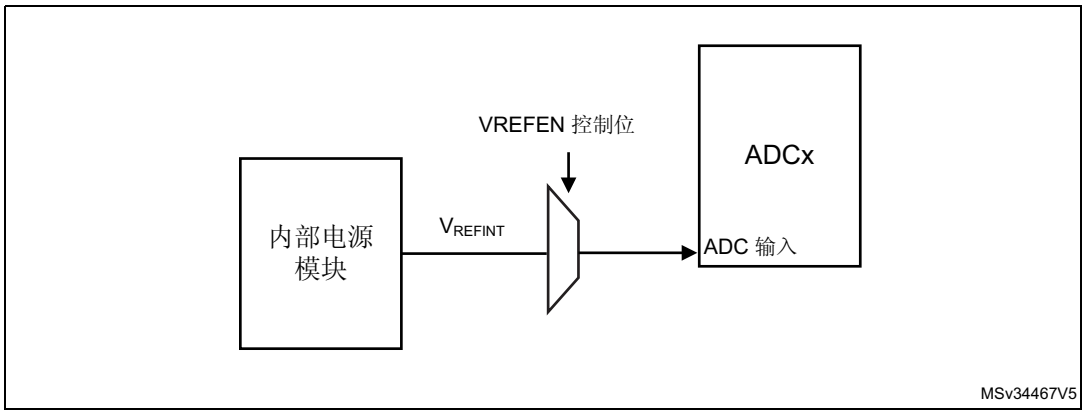
有关内部参考电压在内部连接的 ADC 输入通道的详细信息，请参见*表：ADC 互连*（在*第 26.4.2 节：ADC 引脚和内部信号*中）。

有关转换内部参考电压时要应用的采样时间值，请参见产品数据手册的电气特性部分。

图 250 显示了 REFINT 传感特性的框图。

注： *V_{REFINT} 不能同时由 ADC1 和 ADC2 转换。*

图 250. V_{REFINT} 通道框图



1. 必须将 ADC_CCR 寄存器中的 VREFEN 位置 1 才能使能内部通道 (V_{REFINT}) 的转换。

使用内部参考电压计算实际的 V_{REF+} 电压

V_{REF+} 电压可能会有变化，或无法获得准确值。在制造过程中由 ADC 在 V_{REF+_charac} 的条件下获得的内置内部参考电压 (V_{REFINT}) 及其校准数据可用于评估实际的 V_{REF+} 电压。

以下公式可求得为设备供电的实际 V_{REF+} 电压：

$$V_{REF+} = V_{REF+_Charac} \times VREFINT_CAL / VREFINT_DATA$$

其中：

- V_{REF+_Charac} 是制造过程中 V_{REF+} 电压在 V_{REFINT} 下的特性值。该值在设备数据手册中指定。
- VREFINT_CAL 是 VREFINT 校准值
- VREFINT_DATA 是由 ADC 转换得到的实际 VREFINT 输出值

将电源相关的 ADC 测量值转换为绝对电压值

ADC 用于提供对应于 V_{REF+} 与施加给转换通道的电压之比的数字值。

适用于 V_{REF+} 值未知并且 ADC 转换值为右对齐的应用。在这类应用中，需要将该比值转换为与 V_{REF+} 无关的电压：

$$V_{CHANNELx} = \frac{V_{REF+}}{FULL_SCALE} \times ADC_DATA$$

将 V_{REF+} 替换为上面的公式，就可以通过下面的公式求得绝对电压值

$$V_{CHANNELx} = \frac{V_{REF+_Charac} \times VREFINT_CAL \times ADC_DATA}{VREFINT_DATA \times FULL_SCALE}$$

其中：

- V_{REF+_Charac} 是制造过程中 V_{REF+} 电压在 V_{REFINT} 下的特性值。
- VREFINT_CAL 是 VREFINT 校准值
- ADC_DATA 是由 ADC 在通道 x 上测得的值（右对齐）
- VREFINT_DATA 是由 ADC 转换得到的实际 VREFINT 输出值
- FULL_SCALE 是 ADC 输出的最大数字值。例如，如果分辨率为 12 位，该值为 $2^{12} - 1 = 4095$，如果分辨率为 8 位，该值为 $2^8 - 1 = 255$。

注：　如果执行 ADC 测量时使用的是输出格式而非 12 位右对齐格式，那么必须先将所有参数转换为兼容格式，然后再进行计算。

26.4.34 监视电源电压

ADC2 连接到内部电源电压。若要使用 ADC 测量此电压，可通过 ADC 选项寄存器使能连接。

26.5 ADC 中断

在以下情况下会生成中断：

- ADC 就绪后，ADC 上电（ADRDY 标志）
- 常规组的任何转换结束时（EOC 标志）
- 常规组的转换序列结束时（EOS 标志）
- 注入组的任何转换结束时（JEOC 标志）
- 注入组的转换序列结束时（JEOS 标志）
- 发生模拟看门狗检测时（AWD1、AWD2 和 AWD3 标志）
- 采样阶段结束时（EOSMP 标志）
- 发生数据溢出时（OVR 标志）
- 注入序列上下文队列溢出时（JQOVF 标志）

可以使用单独的中断使能位以提高灵活性。

表 215. ADC 中断

中断向量	中断事件	事件标志	使能控制位	中断清除方法	退出睡眠模式	退出停止、待机模式
ADC	ADC 就绪	ADRDY	ADRDYIE	由硬件置 1，由软件清零	是	否
	常规组的转换结束	EOC	EOCIE			
	常规组的转换序列结束	EOS	EOSIE			
	注入组的转换结束	JEOC	JEOCIE			
	注入组的转换序列结束	JEOS	JEOSIE			
	模拟看门狗 1 状态位置 1	AWD1	AWD1IE			
	模拟看门狗 2 状态位置 1	AWD2	AWD2IE			
	模拟看门狗 3 状态位置 1	AWD3	AWD3IE			
	采样阶段结束	EOSMP	EOSMPIE			
	溢出	OVR	OVRIE			
	注入上下文队列溢出	JQOVF	JQOVFIE			

26.6 ADC 寄存器

有关寄存器说明中使用的缩写，请参见*第 109 页的第 1.2 节*。

26.6.1 ADC 中断和状态寄存器 (ADC_ISR)

ADC interrupt and status register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	JQOVF	AWD3	AWD2	AWD1	JEOS	JEOC	OVR	EOS	EOC	EOSMP	ADRDY
					rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:11 保留，必须保持复位值。

位 10 **JQOVF**：注入上下文队列溢出 (Injected context queue overflow)
当注入上下文队列溢出时，该位会由硬件置 1。通过软件写入 1 可将该位清零。更多信息，请参见*第 26.4.21 节：注入转换的上下文队列*。
0：未发生注入上下文队列溢出（或标志事件已通过软件确认并清零）
1：发生注入上下文队列溢出

位 9 **AWD3**：模拟看门狗 3 标志 (Analog watchdog 3 flag)
当转换电压超过在 ADC_TR3 寄存器的 LT3[7:0] 和 HT3[7:0] 位字段中编程的值时，硬件会将该位置 1。通过软件写入 1 可将该位清零。
0：未发生模拟看门狗 3 事件（或标志事件已通过软件确认并清零）。
1：发生模拟看门狗 3 事件

位 8 **AWD2**：模拟看门狗 2 标志 (Analog watchdog 2 flag)
当转换电压超过在 ADC_TR2 寄存器的 LT2[7:0] 和 HT2[7:0] 位字段中编程的值时，硬件会将该位置 1。通过软件写入 1 可将该位清零。
0：未发生模拟看门狗 2 事件（或标志事件已通过软件确认并清零）
1：发生模拟看门狗 2 事件

位 7 **AWD1**：模拟看门狗 1 标志 (Analog watchdog 1 flag)
当转换电压超过在 ADC_TR1 寄存器的 LT1[11:0] 和 HT1[11:0] 位字段中编程的值时，硬件会将该位置 1。通过软件写入 1 可将该位清零。
0：未发生模拟看门狗 1 事件（或标志事件已通过软件确认并清零）
1：发生模拟看门狗 1 事件

位 6 **JEOS**：注入通道序列结束标志 (Injected channel end of sequence flag)
组内所有注入通道转换结束时，硬件将该位置 1。通过软件写入 1 可将该位清零。
0：注入转换序列未完成（或标志事件已通过软件确认并清零）
1：注入转换已完成

位 5 **JEOC**：注入通道转换结束标志 (Injected channel end of conversion flag)
当通道的每次注入转换结束，新数据出现在相应的 ADC_JDRy 寄存器时，会通过硬件将该位置 1。通过软件向该位写入 1，或读取相应的 ADC_JDRy 寄存器都可将该位清零。
0：注入通道转换未完成（或标志事件已通过软件确认并清零）
1：注入通道转换已完成

位 4 **OVR**：ADC 溢出 (ADC overrun)

该位在常规通道上发生溢出事件时由硬件置 1，这意味着在 EOC 标志已置 1 时，新转换已完成。通过软件写入 1 可将该位清零。

0：未发生溢出（或标志事件已通过软件确认并清零）

1：发生溢出

位 3 **EOS**：常规序列结束标志 (End of regular sequence flag)

常规通道序列转换结束后，硬件将该位置 1。通过软件写入 1 可将该位清零。

0：常规转换序列未完成（或标志事件已通过软件确认并清零）

1：常规转换序列已完成

位 2 **EOC**：转换结束标志 (End of conversion flag)

当通道的每次常规转换结束，新数据出现在 ADC_DR 寄存器时，会通过硬件将该位置 1。通过软件向该位写入 1，或读取 ADC_DR 寄存器都可将该位清零。

0：常规通道转换未完成（或标志事件已通过软件确认并清零）

1：常规通道转换已完成

位 1 **EOSMP**：采样结束标志 (End of sampling flag)

在任何通道（仅限常规通道）的转换过程中，当采样阶段结束时，会通过硬件将该位置 1。

0：采样阶段未结束（或标志事件已通过软件确认并清零）

1：采样阶段已结束

位 0 **ADRDY**：ADC 就绪

ADC 使能后 (ADEN = 1) 以及 ADC 达到准备好接收转换请求的状态时，会通过硬件将该位置 1。通过软件写入 1 可将该位清零。

0：ADC 未准备好开始转换（或标志事件已通过软件确认并清零）

1：ADC 已准备好开始转换

26.6.2 ADC 中断使能寄存器 (ADC_IER)

ADC interrupt enable register

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	JQOVFIE	AWD3IE	AWD2IE	AWD1IE	JEOSIE	JEOCIE	OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10 **JQOVFIE**：注入上下文队列溢出中断使能 (Injected context queue overflow interrupt enable)

该位由软件置 1 和清零，用于使能/禁止注入上下文队列溢出中断。

0：禁止注入上下文队列溢出中断

1：使能注入上下文队列溢出中断JQOVF 位置 1 时产生中断。

注意：仅当 JADSTART = 0 时（这可确保当前未进行任何注入转换），才允许通过软件对该位执行写操作。

位 9 **AWD3IE**：模拟看门狗 3 中断使能 (Analog watchdog 3 interrupt enable)

该位由软件置 1 和清零，用于使能/禁止模拟看门狗 3 中断。

0：禁止模拟看门狗 3 中断

1：使能模拟看门狗 3 中断

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 8 **AWD2IE**：模拟看门狗 2 中断使能 (Analog watchdog 2 interrupt enable)

该位由软件置 1 和清零，用于使能/禁止模拟看门狗 2 中断。

0：禁止模拟看门狗 2 中断

1：使能模拟看门狗 2 中断

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 7 **AWD1IE**：模拟看门狗 1 中断使能 (Analog watchdog 1 interrupt enable)

该位由软件置 1 和清零，用于使能/禁止模拟看门狗 1 中断。

0：禁止模拟看门狗 1 中断

1：使能模拟看门狗 1 中断

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 6 **JEOSIE**：注入转换序列结束中断使能 (End of injected sequence of conversions interrupt enable)

该位由软件置 1 和清零，用于使能/禁止注入转换序列结束中断。

0：禁止 JEOS 中断

1：使能 JEOS 中断。JEOS 位置 1 时产生中断。

注意：仅当 JADSTART = 0 时（这可确保当前未进行任何注入转换），才允许通过软件对该位执行写操作。

位 5 **JEOCIE**：注入转换结束中断使能 (End of injected conversion interrupt enable)

该位由软件置 1 和清零，用于使能/禁止注入转换结束中断。

0：禁止 JEOC 中断。

1：使能 JEOC 中断。JEOC 位置 1 时产生中断。

注意：仅当 JADSTART = 0 时（这可确保当前未进行任何注入转换），才允许通过软件对该位执行写操作。

位 4 **OVRIE**：溢出中断使能 (Overrun interrupt enable)

该位由软件置 1 和清零，用于使能/禁止常规转换的溢出中断。

0：禁止溢出中断

1：使能溢出中断。OVR 位置 1 时产生中断。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对该位执行写操作。

位 3 **EOSIE**：常规转换序列结束中断使能 (End of regular sequence of conversions interrupt enable)

该位由软件置 1 和清零，用于使能/禁止常规转换序列结束中断。

0：禁止 EOS 中断

1：使能 EOS 中断EOS 位置 1 时产生中断。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对该位执行写操作。

位 2 **EOCIE**：常规转换结束中断使能 (End of regular conversion interrupt enable)

该位由软件置 1 和清零，用于使能/禁止常规转换结束中断。

0：禁止 EOC 中断。

1：使能 EOC 中断EOC 位置 1 时产生中断。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对该位执行写操作。

位 1 **EOSMPIE**：常规转换采样结束中断使能 (End of sampling flag interrupt enable for regular conversions)

该位由软件置 1 和清零，用于使能/禁止常规转换采样阶段结束中断。

0：禁止 EOSMP 中断。

1：使能 EOSMP 中断。EOSMP 位置 1 时产生中断。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对该位执行写操作。

位 0 **ADRDYIE**：ADC 就绪中断使能 (ADC ready interrupt enable)

该位由软件置 1 和清零，用于使能/禁止 ADC 就绪中断。

0：禁止 ADRDY 中断

1：使能 ADRDY 中断。ADRDY 位置 1 时产生中断。

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

26.6.3 ADC 控制寄存器 (ADC_CR)

ADC control register

偏移地址：0x08

复位值：0x2000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCAL	ADCALDIF	DEEPPWD	ADVREGEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JADSTP	ADSTP	JADSTART	ADSTART	ADDIS	ADEN
										rs	rs	rs	rs	rs	rs

位 31 **ADCAL**：ADC 校准

该位由软件置 1，用于开始 ADC 校准。先对 ADCALDIF 位编程，以确定此校准应用于单端输入模式还是差分输入模式。

校准完成后，该位由硬件清零。

0：校准已完成

1：写入 1 可校准 ADC。读取值为 1 表示正在进行校准。

注意：仅当 ADEN = 0 时，才允许通过软件将 ADCAL 置 1 的方式启动校准。

仅当 ADEN = 1、ADSTART = 0 且 JADSTART = 0（ADC 已使能，当前未进行任何转换）时，才允许通过软件对 ADC_CALFACT 执行写操作来更新校准系数。

位 30 **ADCALDIF**：校准的差分模式 (Differential mode for calibration)

该位由软件置 1 和清零，用于为校准配置单端输入模式或差分输入模式。

0：写入 ADCAL 会在单端输入模式下启动校准。

1：写入 ADCAL 会在差分输入模式下启动校准。

注意：仅当 ADC 已禁止、且当前未进行校准时（ADCAL = 0、JADSTART = 0、JADSTP = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0 且 ADEN = 0），才允许通过软件对该位执行写操作。

位 29 **DEEPPWD**：深度掉电使能 (Deep-power-down enable)

该位由软件置 1 和清零，用于使 ADC 进入深度掉电模式。

0：ADC 未处于深度掉电模式

1：ADC 处于深度掉电模式（默认复位状态）

注意：仅当 ADC 已禁止时（ADCAL = 0、JADSTART = 0、JADSTP = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0 且 ADEN = 0），才允许通过软件对该位执行写操作。

位 28 **ADVREGEN**：ADC 稳压器使能 (ADC voltage regulator enable)

该位由软件置 1，用于使能 ADC 稳压器。

执行启动校准或使能 ADC 等操作之前，必须先使能 ADC 稳压器，并且软件必须等待稳压器启动时间。

0：禁止 ADC 稳压器

1：使能 ADC 稳压器。

更多关于 ADC 稳压器使能和禁止序列的详细信息，请参见*第 26.4.6 节：ADC 深度掉电模式 (DEEPPWD) 和 ADC 稳压器 (ADVREGEN)*。

仅当 ADC 已禁止时（ADCAL = 0、JADSTART = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0 且 ADEN = 0），才能通过软件对该位字段进行编程。

位 27:6 保留，必须保持复位值。

位 5 **JADSTP**：ADC 停止注入转换命令 (ADC stop of injected conversion command)

该位由软件置 1，用于停止和丢弃正在进行的注入转换（JADSTP 命令）。

当转换已有效丢弃、并且可重新配置 ADC 注入序列和触发时，会通过硬件将该位清零。随后，ADC 会准备好接收新的开始注入转换命令（JADSTART 命令）。

0：当前未执行 ADC 停止注入转换命令

1：写入 1 可停止正在进行的注入转换。读取值为 1 表示正在执行 ADSTP 命令。

注意：仅当 JJADSTART = 1 且 ADDIS = 0 时（ADC 已使能、最终会进行注入转换、并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 JADSTP 置 1

在自动注入模式下 (JAUTO = 1)，将 ADSTP 位置 1 会中止常规转换和注入转换（不要使用 JADSTP）。

位 4 **ADSTP**：ADC 停止常规转换命令 (ADC stop of regular conversion command)

该位由软件置 1，用于停止和丢弃正在进行的常规转换（ADSTP 命令）。

当转换已有效丢弃、并且可重新配置 ADC 常规序列和触发时，会通过硬件将该位清零。随后，ADC 会准备好接收新的开始常规转换命令（ADSTART 命令）。

0：当前未执行 ADC 停止常规转换命令

1：写入 1 可停止正在进行的常规转换。读取值为 1 表示正在执行 ADSTP 命令。

注意：仅当 ADSTART = 1 且 ADDIS = 0 时（ADC 已使能、最终会进行常规转换、并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 ADSTP 置 1。

在自动注入模式下 (JAUTO = 1)，将 ADSTP 位置 1 会中止常规转换和注入转换（不要使用 JADSTP）。

位 3 **JADSTART**：ADC 开始注入转换 (ADC start of injected conversion)

该位由软件置 1，用于开始 ADC 的注入通道转换。根据配置位 JEXTEN 的值，可以立即开始转换（软件触发配置），也可以在发生注入硬件触发事件后开始转换（硬件触发配置）。

该位通过硬件清零：

– 在单次转换模式下，如果选择了软件触发 (JEXTSEL = 0x0)：注入转换序列结束标志 (JEOS) 置位时触发。

– 在所有情况下：执行完 JADSTP 命令后，由硬件将 JADSTP 位清零的同时清零。

0：当前未进行 ADC 注入转换。

1：写入 1 可开始注入转换。读取值为 1 表示 ADC 正在运行，最终会转换注入通道。

注意：仅当 ADEN = 1 且 ADDIS = 0 时（ADC 已使能，并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 JADSTART 置 1。

在自动注入模式下 (JAUTO = 1)，通过将 ADSTART 位置 1 开始常规转换和自动注入转换（JADSTART 必须保持清零）

位 2 **ADSTART**：ADC 开始常规转换 (ADC start of regular conversion)

该位由软件置 1，用于开始 ADC 的常规通道转换。根据配置位 EXTEN 的值，可以立即开始转换（软件触发配置），也可以在发生常规硬件触发事件后开始转换（硬件触发配置）。

该位通过硬件清零：

– 在单次转换模式下，如果选择了软件触发 (EXTSEL = 0x0)：出现常规转换序列结束 (EOS) 标志时清零。

– 在所有情况下：执行完 ADSTP 命令后，由硬件将 ADSTP 位清零的同时清零。

0：当前未进行 ADC 常规转换。

1：写入 1 可开始常规转换。读取值为 1 表示 ADC 正在运行，最终会转换常规通道。

注意：仅当 ADEN = 1 且 ADDIS = 0 时（ADC 已使能，并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 ADSTART 置 1。

在自动注入模式下 (JAUTO = 1)，通过将 ADSTART 位置 1 开始常规转换和自动注入转换（JADSTART 必须保持清零）

位 1 **ADDIS**：ADC 禁止命令 (ADC disable command)

该位由软件置 1，用于禁止 ADC（ADDIS 命令）并使其进入掉电状态（OFF 状态）。

ADC 已有效禁止后，会立即通过硬件将该位清零（此时也会通过硬件将 ADEN 清零）。

0：当前未执行 ADDIS 命令

1：写入 1 可禁止 ADC。读取值为 1 表示正在执行 ADDIS 命令。

注意：仅当 ADEN = 1、ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件将 ADDIS 置 1

位 0 **ADEN**：ADC 使能控制 (ADC enable control)

该位由软件置 1，用于使能 ADC。ADRDY 标志置 1 后，ADC 将立即准备好运行。

如果 ADC 已禁止，则执行 ADDIS 命令后，将通过硬件对该位清零。

0：禁止 ADC（OFF 状态）

1：写入 1 可使能 ADC。

注意：仅当 ADC_CR 寄存器的所有位均为 0 时（ADCAL = 0、JADSTART = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0 且 ADEN = 0，但 ADVREGEN 位除外，该位必须为 1），才允许通过软件将 ADEN 置 1（并且软件必须等待稳压器的启动时间）

26.6.4 ADC 配置寄存器 (ADC_CFGR)

ADC configuration register

偏移地址：0x0C

复位值：0x8000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JQDIS	AWD1CH[4:0]					JAUTO	JAWD1EN	AWD1EN	AWD1SGL	JQM	JDISCEN	DISCNUM[2:0]			DISCEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALIGN	AUTDLY	CONT	OVRMOD	EXTEN[1:0]		EXTSEL4	EXTSEL3	EXTSEL2	EXTSEL1	EXTSEL0	RES[1:0]		DFSDMCFG	DMACFG	DMAEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **JQDIS**：注入队列禁止 (Injected queue disable)

这些位由软件置 1 和清零，用于禁止注入队列机制：

0：使能注入队列

1：禁止注入队列

注意：仅当 AADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行常规转换、也未进行注入转换），才允许通过软件对该位执行写操作。

将 JQDIS 位置 1 或复位会导致注入队列被清空，JSQR 寄存器也会被清空。

位 30:26 **AWD1CH[4:0]**：模拟看门狗 1 通道选择 (Analog watchdog 1 channel selection)

这些位由软件置 1 和清零，用于选择由模拟看门狗监控的输入通道。

00000：通过 AWD1 监控 ADC 模拟输入通道 0

00001：通过 AWD1 监控 ADC 模拟输入通道 1

.....

10010：通过 AWD1 监控 ADC 模拟输入通道 18

其他：保留，不得使用

注意：一些通道未进行物理连接。将相应的 AWD1CH[4:0] 设置保持为复位值。

通过 AWD1CH 选择的通道必须也在 SQRi 或 JSQRi 寄存器中选择。

仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 25 **JAUTO**：注入组自动转换 (Automatic injected group conversion)

通过软件将该位置 1 和清零可在常规组转换后分别使能/禁止注入组自动转换。

0：禁止注入组自动转换

1：使能注入组自动转换

注意：仅当 AADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行常规转换、也未进行注入转换），才允许通过软件对该位执行写操作。

位 24 **JAWD1EN**：注入通道上的模拟看门狗 1 使能 (Analog watchdog 1 enable on injected channels)

该位由软件置 1 和清零

0：在注入通道上禁止模拟看门狗 1

1：在注入通道上使能模拟看门狗 1

注意：仅当 JADSTART = 0 时（这可确保当前未进行任何注入转换），才允许通过软件对该位执行写操作。

位 23 **AWD1EN**：常规通道上的模拟看门狗 1 使能 (Analog watchdog 1 enable on regular channels)

该位由软件置 1 和清零

0：在常规通道上禁止模拟看门狗 1

1：在常规通道上使能模拟看门狗 1

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对该位执行写操作。

位 22 **AWD1SGL**：在单一通道或所有通道上使能看门狗 1 (Enable the watchdog 1 on a single channel or on all channels)

该位由软件置 1 和清零，用于在通过 AWD1CH[4:0] 位确定的通道或所有通道上使能模拟看门狗。

0：在所有通道上使能模拟看门狗 1

1：在单一通道上使能模拟看门狗 1

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 21 **JQM**：JSQR 队列模式 (JSQR queue mode)

该位由软件置 1 和清零，

该位定义空队列的管理方式。

0：JSQR 模式 0：队列从不为空，并会保留上一次写入 JSQR 的配置。

1：JSQR 模式 1：队列可以为空，队列为空时，会在上一个有效注入序列完成后立即在内部禁止注入序列的软件和硬件触发。

更多信息，请参见*第 26.4.21 节：注入转换的上下文队列*。

注意：仅当 JADSTART = 0 时（这可确保当前未进行任何注入转换），才允许通过软件对该位执行写操作。

位 20 **JDISCEN**：注入通道的不连续采样模式 (Discontinuous mode on injected channels)

通过软件将该位置 1 和清零可使能/禁止注入通道的不连续采样模式。

0：禁止注入通道的不连续采样模式

1：使能注入通道的不连续采样模式

注意：仅当 JADSTART = 0 时（这可确保当前未进行任何注入转换），才允许通过软件对该位执行写操作。

不能同时使用自动注入模式和不连续模式：当 JAUTO 置 1 时，DISCEN 和 JDISCEN 位必须通过软件保持清零状态。

位 19:17 **DISCNUM[2:0]**：不连续采样模式通道计数 (Discontinuous mode channel count)

这些位由软件写入，用于定义在接收到外部触发后于不连续采样模式下转换的常规通道数。

000：1 个通道

001：2 个通道

...

111：8 个通道

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 16 **DISCEN**：常规通道的不连续模式 (Discontinuous mode for regular channels)

该位由软件置 1 和清零，用于使能/禁止常规通道的不连续模式。

0：禁止常规通道的不连续模式

1：使能常规通道的不连续模式

注意：不能同时使能不连续模式和连续模式：禁止同时将 DISCEN 和 CONT 位置 1。

不能同时使用自动注入模式和不连续模式：当 JAUTO 置 1 时，DISCEN 和 JDISCEN 位必须通过软件保持清零状态。

仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对该位执行写操作。

位 15 **ALIGN**：数据对齐 (Data alignment)

该位由软件置 1 和清零，用于选择右对齐或左对齐。请参见*数据寄存器、数据对齐和偏移（ADC_DR、OFFSET、OFFSET_CH、ALIGN）*一节。

0：右对齐

1：左对齐

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 14 **AUTDLY**：延迟转换模式 (Delayed conversion mode)

该位由软件置 1 和清零，用于使能/禁止自动延迟转换模式。

0：自动延迟转换模式关闭

1：自动延迟转换模式开启

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 13 **CONT**：常规转换的单次/连续转换模式 (Single / continuous conversion mode for regular conversions)

该位由软件置 1 和清零，该位置 1 时，常规转换将持续进行，直到该位清零。

0：单次转换模式

1：连续转换模式

注意：不能同时使能不连续模式和连续模式：禁止同时将 DISCEN 和 CONT 位置 1。

仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对该位执行写操作。

位 12 **OVRMOD**：溢出模式 (Overrun mode)

该位由软件置 1 和清零，用于配置数据溢出的管理方式。

0：如果检测到溢出，ADC_DR 寄存器会保留原有数据。

1：如果检测到溢出，ADC_DR 寄存器会被上一转换结果覆盖。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对该位执行写操作。

位 11:10 **EXTEN[1:0]**：常规通道的外部触发使能和极性选择 (External trigger enable and polarity selection for regular channels)

通过软件将这些位置 1 和清零可选择外部触发极性和使能常规组的触发。

00：禁止硬件触发检测（可通过软件启动转换）

01：在上升沿执行硬件触发检测

10：在下降沿执行硬件触发检测

11：在上升沿和下降沿都执行硬件触发检测

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 9:5 **EXTSEL[4:0]**：常规组的外部触发选择 (External trigger selection for regular group)

这些位可选择用于触发常规组转换的外部事件。

00000：adc_ext_trg0
00001：adc_ext_trg1
00010：adc_ext_trg2
00011：adc_ext_trg3
00100：adc_ext_trg4
00101：adc_ext_trg5
00110：adc_ext_trg6
00111：adc_ext_trg7
...
11111：adc_ext_trg31

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 4:3 **RES[1:0]**：数据分辨率 (Data resolution)

通过软件写入这些位可选择转换的分辨率。

00：12 位
01：10 位
10：8 位
11：6 位

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 2 **DFSDMCFG**：DFSDM 模式配置 (DFSDM mode configuration)

该位由软件置 1 和清零，用于使能 DFSDM 模式。仅当 DMAEN = 0 时有效。

0：禁止 DFSDM 模式
1：使能 DFSDM 模式

注意：为确保当前未执行任何转换，仅在 ADSTART = 0 且 JADSTART = 0 时才允许通过软件对该位执行写操作。

位 1 **DMACFG**：直接内存访问配置 (Direct memory access configuration)

该位由软件置 1 和清零，用于在 DMA 两种工作模式之间进行选择，仅当 DMAEN = 1 时，该位才有效。

0：选择 DMA 单次模式
1：选择 DMA 循环模式

更多详细信息，请参见*使用 DMA 管理转换*一节

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 0 **DMAEN**：直接内存访问使能 (Direct memory access enable)

该位由软件置 1 和清零，用于使能 DMA 请求的生成。这样便可使用 DMA 自动管理转换的数据。更多详细信息，请参见*使用 DMA 管理转换*一节。

0：禁止 DMA
1：使能 DMA

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

26.6.5 ADC 配置寄存器 2 (ADC_CFGR2)

ADC configuration register 2

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	SMPTRIG	BULB	SWTRIG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
				rw	rw	rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	ROVSM	TROVS	OVSS[3:0]				OVSR[2:0]			JOVSE	ROVSE
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27 **SMPTRIG**：采样时间控制触发模式 (Sampling time control trigger mode)

该位由软件置 1 和清零，用于使能采样时间控制触发模式。

0：禁止采样时间控制触发模式

1：使能采样时间控制触发模式

采样时间始于触发上升沿，转换始于触发下降沿。

EXTEN 位应设置为 01。当 SMPTRIG 位置 1 时，BULB 位不得置 1。

当 EXTEN 位设置为 00 时，将 SWTRIG 置 1 可以开始采样，将 SWTRIG 位清零可以开始转换。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 26 **BULB**：Bulb 采样模式 (Bulb sampling mode)

该位由软件置 1 和清零，用于使能 Bulb 采样模式。

0：禁止 Bulb 采样模式

1：使能 Bulb 采样模式。采样周期在上一次转换结束后立即开始。

如果 BULB 位置 1，则 SAMPTRIG 位不得置 1。

执行第一次 ADC 转换时，采样时间在 SMPx 位中设定。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 25 **SWTRIG**：采样时间控制触发模式的软件触发位 (Software trigger bit for sampling time control trigger mode)

该位由软件置 1 和清零，用于使能 Bulb 采样模式。

0：在采样时间控制触发模式下软件触发启动转换

1：在采样时间控制触发模式下软件触发启动采样

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 24:17 保留，必须保持复位值。

位 16:11 保留，必须保持复位值。

位 10 **ROVSM**：常规过采样模式 (Regular oversampling mode)

该位由软件置 1 和清零，用于选择常规过采样模式。

0：连续模式：如果触发注入转换，过采样会暂时停止，并会在注入序列完成后继续（注入序列过程中会保留过采样缓冲区）

1：恢复模式：如果触发注入转换，当前过采样会中止，并会在注入序列完成后从头开始进行过采样（注入序列开始后，过采样缓冲区会清零）

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 9 **TROVS**：已触发常规过采样 (Triggered Regular oversampling)

该位由软件置 1 和清零，以使能已触发过采样

0：会在触发后连续完成某一通道的所有过采样转换

1：某一通道的每个过采样转换都需要重新触发

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 8:5 **OVSS[3:0]**：过采样移位 (Oversampling shift)

该位字段由软件置 1 和清零，用于定义应用到原始过采样结果的右移位数。

0000：不移位

0001：移 1 位

0010：移 2 位

0011：移 3 位

0100：移 4 位

0101：移 5 位

0110：移 6 位

0111：移 7 位

1000：移 8 位

其他代码保留

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 4:2 **OVSR[2:0]**：过采样率 (Oversampling ratio)

该位字段由软件置 1 和清零，用于定义过采样率。

000：2x

001：4x

010：8x

011：16x

100：32x

101：64x

110：128x

111：256x

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 1 **JOVSE**：注入过采样使能 (Injected oversampling Enable)

该位由软件置 1 和清零，用于使能注入过采样。

0：禁止注入过采样

1：使能注入过采样

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作

位 0 **ROVSE**：常规过采样使能 (Regular oversampling Enable)

该位由软件置 1 和清零，用于使能常规过采样。

0：禁止常规过采样

1：使能常规过采样

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作

26.6.6 ADC 采样时间寄存器 1 (ADC_SMPR1)

ADC sample time register 1

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SMPPLUS	Res.	SMP9[2:0]			SMP8[2:0]			SMP7[2:0]			SMP6[2:0]			SMP5[2:1]	
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP5[0]	SMP4[2:0]			SMP3[2:0]			SMP2[2:0]			SMP1[2:0]			SMP0[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **SMPPLUS**：采样时间加一个时钟周期 (Addition of one clock cycle to the sampling time)

1：对于 ADC_SMPR1 和 ADC_SMPR2 寄存器，2.5 个 ADC 时钟周期的采样时间变为 3.5 个 ADC 时钟周期。

0：采样时间始终设置为 2.5 个 ADC 时钟周期

为确保当前未执行任何转换，仅在 ADSTART = 0 且 JADSTART = 0 时才允许通过软件对该位执行写操作。

位 30 保留，必须保持复位值。

位 29:0 **SMPx[2:0]**：通道 x 采样时间选择 (Channel x sampling time selection)（x = 9 到 0）

通过软件写入这些位可分别为各个通道选择采样时间。在采样周期期间，通道选择位必须保持不变。

000：2.5 个 ADC 时钟周期

001：6.5 个 ADC 时钟周期

010：12.5 个 ADC 时钟周期

011：24.5 个 ADC 时钟周期

100：47.5 个 ADC 时钟周期

101：92.5 个 ADC 时钟周期

110：247.5 个 ADC 时钟周期

111：640.5 个 ADC 时钟周期

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

一些通道未进行物理连接。将相应的 SMPx[2:0] 设置保持为复位值。

26.6.7 ADC 采样时间寄存器 2 (ADC_SMPR2)

ADC sample time register 2

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SMP18[2:0]			SMP17[2:0]			SMP16[2:0]			SMP15[2:1]	
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP15[0]	SMP14[2:0]			SMP13[2:0]			SMP12[2:0]			SMP11[2:0]			SMP10[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:27 保留，必须保持复位值。

位 26:0 **SMPx[2:0]**：通道 x 采样时间选择 (Channel x sampling time selection)（x = 18 到 10）

通过软件写入这些位可分别为各个通道选择采样时间。在采样周期期间，通道选择位必须保持不变。

000：2.5 个 ADC 时钟周期
001：6.5 个 ADC 时钟周期
010：12.5 个 ADC 时钟周期
011：24.5 个 ADC 时钟周期
100：47.5 个 ADC 时钟周期
101：92.5 个 ADC 时钟周期
110：247.5 个 ADC 时钟周期
111：640.5 个 ADC 时钟周期

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

一些通道未进行物理连接。将相应的 SMPx[2:0] 设置保持为复位值。

26.6.8 ADC 看门狗阈值寄存器 1 (ADC_TR1)

ADC watchdog threshold register 1

偏移地址：0x20

复位值：0x0FFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	HT1[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	AWDFILT[2:0]			LT1[11:0]											
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **HT1[11:0]**：模拟看门狗 1 阈值上限 (Analog watchdog 1 higher threshold)

这些位由软件写入，用于定义模拟看门狗 1 的阈值上限。

请参见 *模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*一节

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 15 保留，必须保持复位值。

位 14:12 **AWDFILT[2:0]**：模拟看门狗滤波参数 (Analog watchdog filtering parameter)

该位由软件置 1 和清零，

000：不滤波
001：两次连续检测会生成一个 AWDx 标志或中断
...
111：八次连续检测会生成一个 AWDx 标志或中断

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 11:0 **LT1[11:0]**：模拟看门狗 1 阈值下限 (Analog watchdog 1 lower threshold)
这些位由软件写入，用于定义模拟看门狗 1 的阈值下限。
请参见*第 26.4.29 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*
注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

26.6.9 ADC 看门狗阈值寄存器 2 (ADC_TR2)

ADC watchdog threshold register 2

偏移地址：0x24

复位值：0x00FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT2[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT2[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **HT2[7:0]**：模拟看门狗 2 阈值上限 (Analog watchdog 2 higher threshold)
这些位由软件写入，用于定义模拟看门狗 2 的阈值上限。
请参见*第 26.4.29 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*
注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 15:8 保留，必须保持复位值。

位 7:0 **LT2[7:0]**：模拟看门狗 2 阈值下限（Analog watchdog 2 lower threshold）
这些位由软件写入，用于定义模拟看门狗 2 的阈值下限。
请参见*第 26.4.29 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*
注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

26.6.10 ADC 看门狗阈值寄存器 3 (ADC_TR3)

ADC watchdog threshold register 3

偏移地址：0x28

复位值：0x00FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT3[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT3[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **HT3[7:0]**：模拟看门狗 3 阈值上限 (Analog watchdog 3 higher threshold)

这些位由软件写入，用于定义模拟看门狗 3 的阈值上限。

请参见*第 26.4.29 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 15:8 保留，必须保持复位值。

位 7:0 **LT3[7:0]**：模拟看门狗 3 阈值下限 (Analog watchdog 3 lower threshold)

这些位由软件写入，用于定义模拟看门狗 3 的阈值下限。

此看门狗将 LT3 的 8 位与转换后数据的高 8 位进行比较。

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

26.6.11 ADC 常规序列寄存器 1 (ADC_SQR1)

ADC regular sequence register 1

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ4[4:0]					Res.	SQ3[4:0]					Res.	SQ2[4]
			rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ2[3:0]				Res.	SQ1[4:0]					Res.	Res.	L[3:0]			
rw	rw	rw	rw		rw	rw	rw	rw	rw			rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **SQ4[4:0]**：常规序列中的第四次转换 (4th conversion in regular sequence)

通过软件写入这些位，并将通道编号（0 到 18）分配为常规转换序列中的第四次转换。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 23 保留，必须保持复位值。

位 22:18 **SQ3[4:0]**：常规序列中的第三次转换 (3rd conversion in regular sequence)

通过软件写入这些位，并将通道编号（0 到 18）分配为常规转换序列中的第三次转换。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 17 保留，必须保持复位值。

位 16:12 **SQ2[4:0]**：常规序列中的第二次转换 (2nd conversion in regular sequence)

通过软件写入这些位，并将通道编号（0 到 18）分配为常规转换序列中的第二次转换。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 11 保留，必须保持复位值。

位 10:6 **SQ1[4:0]**：常规序列中的第一次转换 (1st conversion in regular sequence)

通过软件写入这些位，并将通道编号（0 到 18）分配为常规转换序列中的第一次转换。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5:4 保留，必须保持复位值。

位 3:0 **L[3:0]**：常规通道序列长度 (Regular channel sequence length)

通过软件写入这些位可定义常规通道转换序列中的转换总数。

0000：1 次转换

0001：2 次转换

...

1111：16 次转换

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

注：　一些通道未进行物理连接，不得选择这样的通道进行转换。

26.6.12 ADC 常规序列寄存器 2 (ADC_SQR2)

ADC regular sequence register 2

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ9[4:0]					Res.	SQ8[4:0]					Res.	SQ7[4]
			rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ7[3:0]				Res.	SQ6[4:0]					Res.	SQ5[4:0]				
rw	rw	rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **SQ9[4:0]**：常规序列中的第九次转换 (9th conversion in regular sequence)

通过软件写入这些位，并将通道编号（0 到 18）分配为常规转换序列中的第九次转换。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 23 保留，必须保持复位值。

位 22:18 **SQ8[4:0]**：常规序列中的第八次转换 (8th conversion in regular sequence)
通过软件写入这些位，并将通道编号（0 到 18）分配为常规转换序列中的第八次转换。
注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 17 保留，必须保持复位值。

位 16:12 **SQ7[4:0]**：常规序列中的第七次转换 (7th conversion in regular sequence)
通过软件写入这些位，并将通道编号（0 到 18）分配为常规转换序列中的第七次转换。
注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 11 保留，必须保持复位值。

位 10:6 **SQ6[4:0]**：常规序列中的第六次转换 (6th conversion in regular sequence)
通过软件写入这些位，并将通道编号（0 到 18）分配为常规转换序列中的第六次转换。
注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5 保留，必须保持复位值。

位 4:0 **SQ5[4:0]**：常规序列中的第五次转换 (5th conversion in regular sequence)
通过软件写入这些位，并将通道编号（0 到 18）分配为常规转换序列中的第五次转换。
注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

注：一些通道未进行物理连接，不得选择这样的通道进行转换。

26.6.13 ADC 常规序列寄存器 3 (ADC_SQR3)

ADC regular sequence register 3

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ14[4:0]					Res.	SQ13[4:0]					Res.	SQ12[4]
			rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ12[3:0]				Res.	SQ11[4:0]					Res.	SQ10[4:0]				
rw	rw	rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **SQ14[4:0]**：常规序列中的第十四次转换 (14th conversion in regular sequence)
通过软件写入这些位，并将通道编号（0 到 18）分配为常规转换序列中的第十四次转换。
注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 23 保留，必须保持复位值。

位 22:18 **SQ13[4:0]**：常规序列中的第十三次转换 (13th conversion in regular sequence)
通过软件写入这些位，并将通道编号（0 到 18）分配为常规转换序列中的第十三次转换。
注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 17 保留，必须保持复位值。

位 16:12 **SQ12[4:0]**：常规序列中的第十二次转换 (12th conversion in regular sequence)

通过软件写入这些位，并将通道编号（0 到 18）分配为常规转换序列中的第十二次转换。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 11 保留，必须保持复位值。

位 10:6 **SQ11[4:0]**：常规序列中的第十一次转换 (11th conversion in regular sequence)

通过软件写入这些位，并将通道编号（0 到 18）分配为常规转换序列中的第十一次转换。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5 保留，必须保持复位值。

位 4:0 **SQ10[4:0]**：常规序列中的第十次转换 (10th conversion in regular sequence)

通过软件写入这些位，并将通道编号（0 到 18）分配为常规转换序列中的第十次转换。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

注：　一些通道未进行物理连接，不得选择这样的通道进行转换。

26.6.14 ADC 常规序列寄存器 4 (ADC_SQR4)

ADC regular sequence register 4

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	SQ16[4:0]					Res.	SQ15[4:0]				
					rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10:6 **SQ16[4:0]**：常规序列中的第十六次转换 (16th conversion in regular sequence)

通过软件写入这些位，并将通道编号（0 到 18）分配为常规转换序列中的第十六次转换。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5 保留，必须保持复位值。

位 4:0 **SQ15[4:0]**：常规序列中的第十五次转换 (15th conversion in regular sequence)

通过软件写入这些位，并将通道编号（0 到 18）分配为常规转换序列中的第十五次转换。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

注：　一些通道未进行物理连接，不得选择这样的通道进行转换。

26.6.15 ADC 常规数据寄存器 (ADC_DR)

ADC regular data register

偏移地址：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **RDATA[15:0]**：已转换的常规数据 (Regular data converted)

这些位为只读。其中包含上一转换常规通道的转换结果。数据有左对齐和右对齐两种方式，如*第 26.4.26 节：数据管理*所述。

26.6.16 ADC 注入序列寄存器 (ADC_JSQR)

ADC injected sequence register

偏移地址：0x4C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JSQ4[4:0]					Res.	JSQ3[4:0]					Res.	JSQ2[4:1]			
rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSQ2[0]	Res.	JSQ1[4:0]					JEXTEN[1:0]		JEXTSEL[4:0]					JL[1:0]	
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:27 **JSQ4[4:0]**：注入序列中的第四次转换 (4th conversion in the injected sequence)

通过软件写入这些位，并将通道编号（0 到 18）分配为注入转换序列中的第四次转换。

注意：仅当 JADSTART = 0 时（这可确保当前未进行任何注入转换），才允许通过软件对这些位执行写操作。

位 26 保留，必须保持复位值。

位 25:21 **JSQ3[4:0]**：注入序列中的第三次转换 (3rd conversion in the injected sequence)

通过软件写入这些位，并将通道编号（0 到 18）分配为注入转换序列中的第三次转换。

注意：仅当 JADSTART = 0 时（这可确保当前未进行任何注入转换），才允许通过软件对这些位执行写操作。

位 20 保留，必须保持复位值。

位 19:15 **JSQ2[4:0]**：注入序列中的第二次转换 (2nd conversion in the injected sequence)

通过软件写入这些位，并将通道编号（0 到 18）分配为注入转换序列中的第二次转换。

注意：仅当 JADSTART = 0 时（这可确保当前未进行任何注入转换），才允许通过软件对这些位执行写操作。

位 14 保留，必须保持复位值。

位 13:9 **JSQ1[4:0]**：注入序列中的第一次转换 (1st conversion in the injected sequence)

通过软件写入这些位，并将通道编号（0 到 18）分配为注入转换序列中的第一次转换。

注意：仅当 JADSTART = 0 时（这可确保当前未进行任何注入转换），才允许通过软件对这些位执行写操作。

位 8:7 **JEXTEN[1:0]**：注入通道的外部触发使能和极性选择 (External trigger enable and polarity selection for injected channels)

通过软件将这些位置 1 和清零可选择外部触发极性和使能注入组的触发。

00：如果 JQDIS = 0（队列使能），将同时禁止硬件和软件触发检测。否则，禁止队列以及硬件触发检测（可通过软件启动转换）

01：在上升沿执行硬件触发检测

10：在下降沿执行硬件触发检测

11：在上升沿和下降沿都执行硬件触发检测

注意：仅当 JADSTART = 0 时（这可确保当前未进行任何注入转换），才允许通过软件对这些位执行写操作。

如果 JQM = 1 且上下文队列为空，则会在内部禁止注入序列的软件和硬件触发（请参见第 26.4.21 节：注入转换的上下文队列）

位 6:2 **JEXTSEL[4:0]**：注入组的外部触发选择 (External Trigger Selection for injected group)

这些位可选择用于触发注入组转换的外部事件：

00000：adc_jext_trg0

00001：adc_jext_trg1

00010：adc_jext_trg2

00011：adc_jext_trg3

00100：adc_jext_trg4

00101：adc_jext_trg5

00110：adc_jext_trg6

00111：adc_jext_trg7

...

11111：adc_jext_trg31

注意：仅当 JADSTART = 0 时（这可确保当前未进行任何注入转换），才允许通过软件对这些位执行写操作。

位 1:0 **JL[1:0]**：注入通道序列长度 (Injected channel sequence length)

通过软件写入这些位可定义注入通道转换序列中的转换总数。

00：1 次转换

01：2 次转换

10：3 次转换

11：4 次转换

注意：仅当 JADSTART = 0 时（这可确保当前未进行任何注入转换），才允许通过软件对这些位执行写操作。

注：　一些通道未进行物理连接，不得选择这样的通道进行转换。

26.6.17 ADC 偏移 y 寄存器 (ADC_OFRy)

ADC offset y register

偏移地址：0x60 + 0x04 * (y -1)，（y = 1 到 4）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OFFSET_EN	OFFSET_CH[4:0]					SATEN	OFFSETPOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	OFFSET[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **OFFSET_EN**：偏移 y 使能 (Offset y enable)

该位由软件写入，用于使能或禁止编程到 OFFSET[11:0] 位中的偏移。

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 30:26 **OFFSET_CH[4:0]**：数据偏移 y 的通道选择 (Channel selection for the data offset y)

这些位由软件写入，用于定义编程到 OFFSET[11:0] 位中的偏移将应用于哪个通道。

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

一些通道未进行物理连接，不得选择这样的通道用于数据偏移 y。

如果 OFFSET_EN 置 1，则不允许为不同的 ADC_OFRy 寄存器选择同一个通道。

位 25 **SATEN**：饱和使能 (Saturation enable)

该位由软件置 1 和清零，用于使能偏移功能在 0x000 和 0xFFF 的饱和。

0：无饱和控制，偏移结果可以是有符号

1：使能饱和，偏移结果为无符号，在 0x000 和 0xFFF 饱和

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 24 **OFFSETPOS**：正偏移 (Positive offset)

该位由软件置 1 和清零，用于使能正偏移。

0：负偏移

1：正偏移 (Positive offset)

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 23:12 保留，必须保持复位值。

位 11:0 **OFFSET[11:0]**：编程到 OFFSET_CH[4:0] 位中的通道对应的数据偏移 y (Data offset y for the channel programmed into bits OFFSET_CH[4:0])

这些位由软件写入，用于定义在转换通道（可以是常规通道或注入通道）时从原始转换数据中减去的偏移。应用数据偏移的通道必须在 OFFSET_CH[4:0] 位中编程。转换结果可从 ADC_DR（常规转换）或 ADC_JDRyi 寄存器（注入转换）中读取。

注意：仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

如果多个偏移 (OFFSET) 指向同一通道，相减时只会考虑使用 x 值最小的偏移。

例如：如果 OFFSET1_CH[4:0] = 4 且 OFFSET2_CH[4:0] = 4，转换通道 4 时会减去 OFFSET1[11:0]。

26.6.18 ADC 注入通道 y 数据寄存器 (ADC_JDRy)

ADC injected channel y data register

偏移地址：0x80 + 0x04 * (y - 1)，（y = 1 到 4）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JDATA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **JDATA[15:0]**：注入数据（Injected data）

这些位为只读。它们包括来自注入通道 y 的转换结果。数据有左对齐和右对齐两种方式，如 *第 26.4.26 节：数据管理* 所述。

26.6.19 ADC 模拟看门狗 2 配置寄存器 (ADC_AWD2CR)

ADC analog watchdog 2 configuration register

偏移地址：0xA0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD2CH[18:16]		
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD2CH[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:19 保留，必须保持复位值。

位 18:0 **AWD2CH[18:0]**：模拟看门狗 2 通道选择 (Analog watchdog 2 channel selection)

这些位由软件置 1 和清零，用于使能和选择由模拟看门狗 2 监控的输入通道。

AWD2CH[i] = 0：不通过 AWD2 监控 ADC 模拟输入通道 i

AWD2CH[i] = 1：通过 AWD2 监控 ADC 模拟输入通道 i

当 AWD2CH[18:0] = 000..0 时，会禁止模拟看门狗 2

注意：通过 AWD2CH 选择的通道必须也在 SQRi 或 JSQRi 寄存器中选择。

仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

一些通道未进行物理连接，不得选择这样的通道用于模拟看门狗。

26.6.20 ADC 模拟看门狗 3 配置寄存器 (ADC_AWD3CR)

ADC analog watchdog 3 configuration register

偏移地址：0xA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD3CH[18:16]		
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD3CH[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:19 保留，必须保持复位值。

位 18:0 **AWD3CH[18:0]**：模拟看门狗 3 通道选择 (Analog watchdog 2 channel selection)

这些位由软件置 1 和清零，用于使能和选择由模拟看门狗 3 监控的输入通道。

AWD3CH[i] = 0：不通过 AWD3 监控 ADC 模拟输入通道 i

AWD3CH[i] = 1：通过 AWD3 监控 ADC 模拟输入通道 i

当 AWD3CH[18:0] 时，会禁止模拟看门狗 3

注意：通过 AWD3CH 选择的通道必须也在 SQRi 或 JSQRi 寄存器中选择。

仅当 ADSTART = 0 且 JADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

一些通道未进行物理连接，不得选择这样的通道用于模拟看门狗。

26.6.21 ADC 差分模式选择寄存器 (ADC_DIFSEL)

ADC Differential mode selection register

偏移地址：0xB0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIFSEL[18:16]		
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIFSEL[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r

位 31:19 保留，必须保持复位值。

位 18:0 **DIFSEL[18:0]**：通道 **18** 到 0 的差分模式 (Differential mode for channels 18 to 0)

这些位由软件置 1 和清零，它们用于选择将通道配置为单端模式或差分模式。

DIFSEL[i] = 0：将 ADC 模拟输入通道 i 配置为单端模式

DIFSEL[i] = 1：将 ADC 模拟输入通道 i 配置为差分模式

注意：与连接到单端 I/O 端口或内部通道的通道相对应的 DIFSEL 位必须保持复位值（单端输入模式）。

仅当 ADC 已禁止时（ADCAL = 0、JADSTART = 0、JADSTP = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0 且 ADEN = 0），才允许通过软件对这些位执行写操作。

26.6.22 ADC 校准系数 (ADC_CALFACT)

ADC calibration factors

偏移地址：0xB4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALFACT_D[6:0]						
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALFACT_S[6:0]						
									rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值。

位 22:16 **CALFACT_D[6:0]**：差分模式下的校准系数 (Calibration Factors in differential mode)

这些位可由硬件或软件写入。

差分输入校准完成后，会立即由硬件更新为校准系数。

软件可向这些位写入新的校准系数。如果新校准系数不同于当前存储于模拟 ADC 中的校准系数，启动新的差分校准后，会立即应用新校准系数。

注意：仅当 ADEN = 1、ADSTART = 0 且 JADSTART = 0 时（ADC 已使能、当前未执行任何校准和转换），才允许通过软件对这些位执行写操作。

位 15:7 保留，必须保持复位值。

位 6:0 **CALFACT_S[6:0]**：单端模式下的校准系数 (Calibration Factors In single-ended mode)

这些位可由硬件或软件写入。

单端输入校准完成后，会立即由硬件更新为校准系数。

软件可向这些位写入新的校准系数。如果新校准系数不同于当前存储于模拟 ADC 中的校准系数，启动新的单端校准后，会立即应用新校准系数。

注意：仅当 ADEN = 1、ADSTART = 0 且 JADSTART = 0 时（ADC 已使能、当前未执行任何校准和转换），才允许通过软件对这些位执行写操作。

26.6.23 ADC 选项寄存器 (ADC_OR)

ADC option register

偏移地址：0xC8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OP2	OP1	OP0
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2 **OP2**：选项位 2 (Option bit 2)
0：禁止 V_{DDQ_DDR} 通道
1：使能 V_{DDQ_DDR} 通道

位 1 **OP1**：选项位 1 (Option bit 1)
0：禁止 V_{DDCPU} 通道
1：使能 V_{DDCPU} 通道

位 0 **OP0**：选项位 0
0：禁止 V_{DDCORE} 通道
1：使能 V_{DDCORE} 通道

注： *在一些 ADC 实例上，ADC_OR 寄存器可能会被保留。请参见第 26.3 节：ADC 实现。*

26.7 ADC 通用寄存器

26.7.1 ADC 通用控制寄存器 (ADC_CCR)

ADC common control register

偏移地址：0x308

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBATEN	TSEN	VREFEN	PRESC[3:0]				CKMODE[1:0]	
							rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:25 保留，必须保持复位值。

位 24 **VBATEN**：VBAT 使能 (VBAT enable)
通过软件将该位置 1 和清零以进行控制。
0：禁止 V_{BAT} 通道
1：使能 V_{BAT} 通道

位 23 **TSEN**：V_{SENSE} 使能 (V_{SENSE} enable)
通过软件将该位置 1 和清零以控制 V_{SENSE}。
0：禁止温度传感器通道
1：使能温度传感器通道

位 22 **VREFEN**：V_{REFINT} 使能 (V_{REFINT} enable)
通过软件将该位置 1 和清零可使能/禁止 V_{REFINT} 通道。
0：禁止 V_{REFINT} 通道
1：使能 V_{REFINT} 通道

位 21:18 **PRESC[3:0]**：ADC 预分频器 (ADC prescaler)

这些位由软件置 1 和清零，用于选择 ADC 的时钟频率。该时钟为所有 ADC 所共用。

0000：输入 ADC 时钟未分频
0001：输入 ADC 时钟 2 分频
0010：输入 ADC 时钟 4 分频
0011：输入 ADC 时钟 6 分频
0100：输入 ADC 时钟 8 分频
0101：输入 ADC 时钟 10 分频
0110：输入 ADC 时钟 12 分频
0111：输入 ADC 时钟 16 分频
1000：输入 ADC 时钟 32 分频
1001：输入 ADC 时钟 64 分频
1010：输入 ADC 时钟 128 分频
1011：输入 ADC 时钟 256 分频
其他：保留

注意：仅当 ADC 已禁止时（ADCAL = 0、JADSTART = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0 且 ADEN = 0），才允许通过软件对这些位执行写操作。仅当 CKMODE[1:0] = 0b00 时，才会应用 ADC 预分频值。

位 17:16 **CKMODE[1:0]**：ADC 时钟模式 (ADC clock mode)

这些位由软件置 1 和清零，用于定义 ADC 时钟方案（主 ADC 和从 ADC 共用）：

00：adc_ker_ck (x = 1/2)（异步时钟模式），在产品级生成（请参见*第 6 节：复位和时钟控制 (RCC)*）

01：adc_hclk/1（同步时钟模式）。仅当 AHB 时钟预分频器设为 1（RCC_CFGR 寄存器中的 HPRE[3:0] = 0XXX）且系统时钟占空比为 50% 时，才必须使能此配置。

10：adc_hclk/2（同步时钟模式）

11：adc_hclk/4（同步时钟模式）

在所有同步时钟模式下，从定时器触发到转换开始的延迟过程中，不存在抖动。

注意：仅当 ADC 已禁止时（ADCAL = 0、JADSTART = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0 且 ADEN = 0），才允许通过软件对这些位执行写操作。

位 15:0 保留，必须保持复位值。

26.7.2 ADC 硬件配置寄存器 (ADC_HWCFGR0)

ADC hardware configuration register

偏移地址：0x3F0

复位值：0x0000 1411

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDLEVALUE[3:0]				OPBITS[3:0]				MULPIPE[3:0]				ADCNUM[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:12 **IDLEVALUE[3:0]**：未选通道的空闲值 (Idle value for non-selected channels)
0000：空闲通道选择为 0x13
0001：空闲通道选择为 0x1F

位 11:8 **OPBITS[3:0]**：选项位的数量 (Number of option bits)
0000：未实现选项寄存器
0100：在 ADC 选项寄存器 (ADC_OR) 中实现了 4 个选项位（偏移地址为 0xC8）

位 7:4 **MULPIPE[3:0]**：流水线级数 (Number of pipeline stages)
0001：一级流水线

位 3:0 **ADCNUM[3:0]**：实现的 ADC 数 (Number of ADCs implemented)
0001：实现了一个 ADC 实例
0010：实现了两个 ADC 实例
0011：实现了三个 ADC 实例

26.7.3 ADC 版本触发器 (ADC_VERR)

ADC version register

偏移地址：0x3F4

复位值：0x0000 0011

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本 (Major revision)
这些位返回 ADC IP 主版本
0001：主版本 = 1.X

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)
这些位返回 ADC IP 次版本
0001：次版本 = X.1
0002：次版本 = X.2
0003：次版本 = X.3

26.7.4 ADC 标识寄存器 (ADC_IPDR)

ADC identification register

偏移地址：0x3F8

复位值：0x0011 0006

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：外设标识符 (Peripheral identifier)

这些位返回 ADC 标识符。

ID[31:0] = 0x0011 0006：c7amba_aditf5_90_v1

26.7.5 ADC 大小标识寄存器 (ADC_SIDR)

ADC size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识 (Size Identification)

SID[31:8]：描述 ADC_SIDR 寄存器特性的固定代码。该位字段始终读为 0xA3C5DD。

SID[7:0]：只读数字字段，返回标识寄存器与 IP 基址之间的偏移地址 (KB)：

0x01：1 KB 偏移地址

0x02：2 KB 偏移地址

0x04：4 KB 偏移地址

0x08：8 KB 偏移地址

26.8 ADC 寄存器映射

表 216. ADC 全局寄存器映射

偏移	寄存器
0x000 - 0x0B4	主 ADC1/2
0x0B8 - 0x2FC	保留
0x300 - 0x30C	主 ADC 和从 ADC 通用寄存器

表 217. ADC 寄存器映射和复位值

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**ADC_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JQOVF	AWD3	AWD2	AWD1	JEOS	JEOC	OVR	EOS	EOC	EOSMP	ADRDY
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x04	**ADC_IER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JQOVFIE	AWD3IE	AWD2IE	AWD1IE	JEOSIE	JEOCIE	OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x08	**ADC_CR**	ADCAL	ADCALDIF	DEEPPWD	ADVREGEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JADSTP	ADSTP	JADSTART	ADSTART	ADDIS	ADEN
	Reset value	0	0	1	0																							0	0	0	0	0	0
0x0C	**ADC_CFGR**	JQDIS	AWD1CH[4:0]					JAUTO	JAWD1EN	AWD1EN	AWD1SGL	JQM	JDISCEN	DISCNUM[2:0]			DISCEN	ALIGN	AUTDLY	CONT	OVRMOD	EXTEN[1:0]		EXTSEL4	EXTSEL3	EXTSEL2	EXTSEL1	EXTSEL0	RES[1:0]		DFSDMCFG	DMACFG	DMAEN
	Reset value	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**ADC_CFGR2**	Res.	Res.	Res.	Res.	SMPTRIG	BULB	SWTRIG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ROVSM	TROVS	OVSS[3:0]					OVSR[2:0]		JOVSE	ROVSE
	Reset value					0	0	0															0	0	0	0	0	0	0	0	0	0	0
0x14	**ADC_SMPR1**	SMPPLUS	Res.	SMP9 [2:0]			SMP8 [2:0]			SMP7 [2:0]			SMP6 [2:0]			SMP5 [2:0]			SMP4 [2:0]			SMP3 [2:0]			SMP2 [2:0]			SMP1 [2:0]			SMP0 [2:0]		
	Reset value	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	**ADC_SMPR2**	Res.	Res.	Res.	Res.	Res.	SMP18 [2:0]			SMP17 [2:0]			SMP16 [2:0]			SMP15 [2:0]			SMP14 [2:0]			SMP13 [2:0]			SMP12 [2:0]			SMP11 [2:0]			SMP10 [2:0]		
	Reset value						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	Reserved	Res.																															
0x20	**ADC_TR1**	Res.	Res.	Res.	Res.	HT1[11:0]												Res.	AWDFILT [2:0]			LT1[11:0]												
	Reset value					1	1	1	1	1	1	1	1	1	1	1	1		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	**ADC_TR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT2[[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT2[7:0]							
	Reset value									1	1	1	1	1	1	1	1									0	0	0	0	0	0	0	0
0x28	**ADC_TR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT3[[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT3[7:0]							
	Reset value									1	1	1	1	1	1	1	1									0	0	0	0	0	0	0	0

表 217. ADC 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x2C	Reserved	Res.																															
0x30	**ADC_SQR1**	Res.	Res.	Res.	SQ4[4:0]					Res.	SQ3[4:0]					Res.	SQ2[4:0]					Res.	SQ1[4:0]					Res.	Res.	L[3:0]			
	Reset value				0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0			0	0	0	0
0x34	**ADC_SQR2**	Res.	Res.	Res.	SQ9[4:0]					Res.	SQ8[4:0]					Res.	SQ7[4:0]					Res.	SQ6[4:0]					Res.	SQ5[4:0]				
	Reset value				0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0	0
0x38	**ADC_SQR3**	Res.	Res.	Res.	SQ14[4:0]					Res.	SQ13[4:0]					Res.	SQ12[4:0]					Res.	SQ11[4:0]					Res.	SQ10[4:0]				
	Reset value				0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0	0
0x3C	**ADC_SQR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SQ16[4:0]					Res.	SQ15[4:0]				
	Reset value																						0	0	0	0	0		0	0	0	0	0
0x40	**ADC_DR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	regular RDATA[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44- 0x48	Reserved	Res.																															
0x4C	**ADC_JSQR**	JSQ4[4:0]					Res.	JSQ3[4:0]					Res.	JSQ2[4:0]				Res.	JSQ1[4:0]					JEXTEN[1:0]		JEXTSEL [4:0]				JL[1:0]			
	Reset value	0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50- 0x5C	Reserved	Res.																															
0x60	**ADC_OFR1**	OFFSET_EN	OFFSET_ CH[4:0]					SATEN	OFFSETPOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET[11:0]										
	Reset value	0	0	0	0	0	0	0	0														0	0	0	0	0	0	0	0	0	0	0
0x64	**ADC_OFR2**	OFFSET_EN	OFFSET_ CH[4:0]					SATEN	OFFSETPOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET[11:0]										
	Reset value	0	0	0	0	0	0	0	0														0	0	0	0	0	0	0	0	0	0	0
0x68	**ADC_OFR3**	OFFSET_EN	OFFSET_ CH[4:0]					SATEN	OFFSETPOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET[11:0]										
	Reset value	0	0	0	0	0	0	0	0														0	0	0	0	0	0	0	0	0	0	0
0x6C	**ADC_OFR4**	OFFSET_EN	OFFSET_ CH[4:0]					SATEN	OFFSETPOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET[11:0]										
	Reset value	0	0	0	0	0	0	0	0														0	0	0	0	0	0	0	0	0	0	0
0x70- 0x7C	Reserved	Res.																															
0x80	**ADC_JDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x84	**ADC_JDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 217. ADC 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x88	**ADC_JDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA3[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8C	**ADC_JDR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA4[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x90-0x9C	Reserved	Res.																															
0xA0	**ADC_AWD2CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD2CH[18:0]																		
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA4	**ADC_AWD3CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD3CH[18:0]																		
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA8-0xAC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xB0	**ADC_DIFSEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIFSEL[18:0]																		
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xB4	**ADC_CALFACT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALFACT_D[6:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALFACT_S[6:0]							
	Reset value									0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0xB8-0xC4	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xC8	**ADC_OR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OP2	OP1	OP0
	Reset value																														0	0	0
0xCC-0xFC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 218. ADC 寄存器映射和复位值（主 ADC 和从 ADC 通用寄存器）

偏移	寄存器名称 Reset value	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x304	Reserved	Res.																															
0x308	**ADC_CCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBATEN	TSEN	VREFEN	PRESC[3:0]				CKMODE[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value								0	0	0	0	0	0	0	0	0																
x30C-0x3EC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x3F0	**ADC_HWCFGR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDLEVALUE [3:0]				OPBITS[3:0]				MULPIPE[3:0]				ADCNUM[3:0]			
	Reset value	0x0000 1411																															
0x3F4	**ADC_VERR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value	0x0000 0011																															
0x3F8	**ADC_IPDR**	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0x3FC	**ADC_SIDR**	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

27 数字温度传感器 (DTS)

27.1 简介

设备内置一个可将温度转换为方波的传感器，其中方波的频率与温度成正比。该频率基于 PCLK 或 LSE 时钟测得。

27.2 DTS 主要特性

温度传感器模块的主要特性如下：

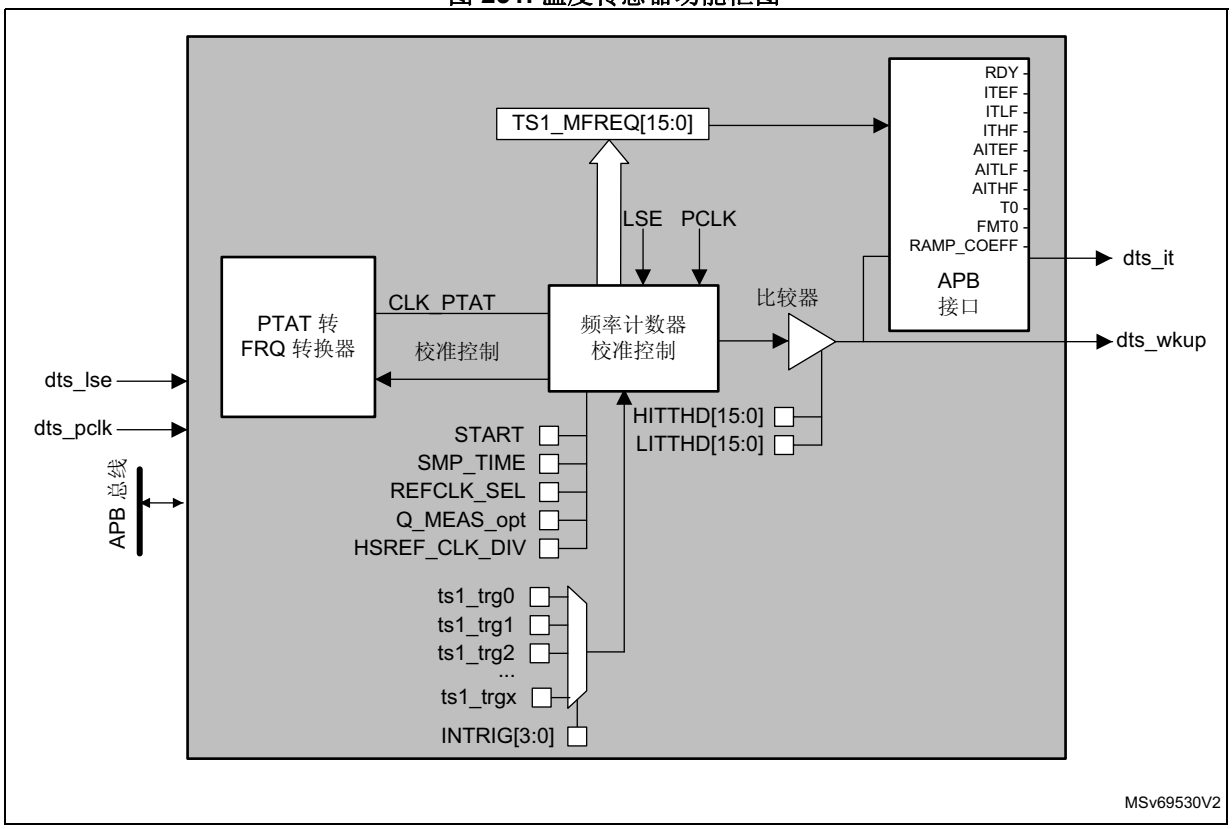
- 测量由软件或 4 个硬件源触发启动
- 可编程采样时间以提高温度测量精度
- 计数器基于 LSE 或 PCLK 时钟进行同步
- 基于阈值下限和阈值上限的温度看门狗
- 可在温度低于或高于预定义阈值时以及测量结束时生成中断
- 可在温度高于/低于预定义阈值时生成异步唤醒信号（仅限 LSE 模式）
- 使用 LSE 时钟实现快速测量

27.3 DTS 功能描述

27.3.1 DTS 框图

温度传感器框图如*图 251* 所示。

图 251. 温度传感器功能框图



27.3.2 DTS 内部信号

表 219. DTS 内部输入/输出信号

信号名称	信号类型	说明
dts_lse	数字输入	LSE 时钟
dts_pclk	数字输入	APB 时钟
dts_it	数字输出	温度传感器中断
dts_wkup	数字输出	温度传感器唤醒

27.3.3 DTS 模块操作

温度传感器的模拟部分用于输出频率，该频率与绝对温度 (CLK_PTAT) 成正比。该频率基于 PCLK 或 LSE 时钟测得。

每次测量前，温度传感器都会对频率生成模块执行校准。

27.3.4 操作模式

可通过设置*温度传感器配置寄存器 1 (DTS_CFGR1)* 中的 REFCLK_SEL 位来选择多种工作模式：

- 仅 PCLK (REFCLK_SEL = 0)
  可以访问温度传感器寄存器。因此，可重新配置接口并使用 PCLK 时钟执行测量序列。
- PCLK 和 LSE (REFCLK_SEL = 1)
  可以访问温度传感器寄存器。因此，可重新配置接口并使用 LSE 时钟执行测量序列。
- 仅 LSE (REFCLK_SEL = 1) 且 PCLK 关闭
  无法访问温度传感器寄存器。可使用 LSE 时钟执行测量。该模式用于通过硬件触发和异步中断线退出睡眠模式。

27.3.5 校准

温度传感器在执行任何频率测量前都必须运行校准。除快速测量模式（DTS_CFGR1 中的 Q_MEAS_OPT 设为 1）外，触发温度测量后都将自动执行校准。

27.3.6 预分频器

在校准过程中，计数器时钟必须低于 1 MHz。这可通过内置在温度传感器中的 PCLK 时钟预分频器来实现。

在温度测量期间，预分频器会被绕过。

- 当 PCLK 用作参考时钟（DTS_CFGR1 中的 REFCLK_SEL 设为 0）时，使用预分频器。分频比必须配置为 127（关于分频器设置，请参见 HSREF_CLK_DIV[6:0] 寄存器定义）。
- 当 LSE 用作参考时钟（DTS_CFGR1 中的 REFCLK_SEL 设为 1）时，时基等于 2 个 LSE 周期。在这种情况下，不使用预分频器。

27.3.7 温度测量原理

温度传感器的模拟部分用于输出信号 (CLK_PTAT)，其 FM(T) 频率取决于温度。

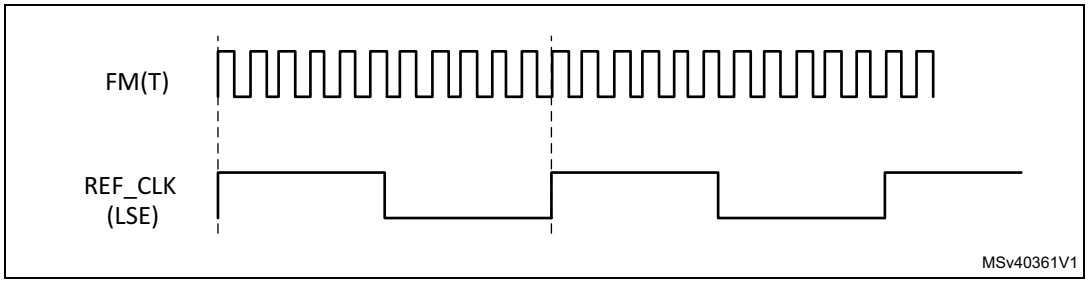
可以通过 DTS_CFGR1 中的 REFCLK_SEL 位将 PCLK 或 LSE 选作参考时钟 (REF_CLK)。

计数方法取决于 REF_CLK 频率。这是因为温度传感器模块中实现了两个计数器：

- 对于低 REF_CLK 频率，在一个或多个 REF_CLK 周期内对 FM(T) 周期进行计数。
- 对于高 REF_CLK 频率，在一个或多个 FM(T) 周期内对 REF_CLK 周期进行计数。

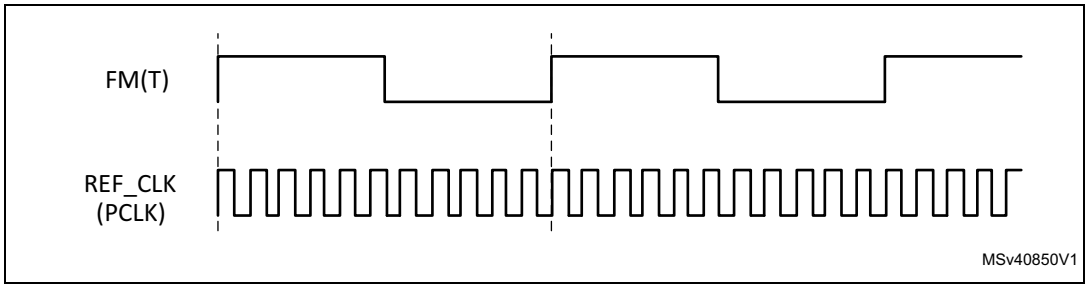
该计数器的行为如*图 252* 和*图 253* 所示。

图 252. 低 REF_CLK 频率时的方法



1. 为提高精度，可以在多个 LSE 周期内进行 FM(T) 测量。

图 253. 高 REF_CLK 频率时的方法



1. 为提高精度，可以在多个 FM(T) 周期内进行 PCLK 测量。

计数结果存储在 DTS_DR 寄存器中（请参见*温度传感器数据寄存器 (DTS_DR)*）。

获得 FM(T) 频率后，可使用以下公式通过软件计算相应的温度：

- 使用 PCLK 时：

$$T = T_0 + ((F_{PCLK} / TS1_MFREQ) \times TS1_SMP_TIME - 100 \times TS1_FMT0) / TS1_RAMP_COEFF$$

其中

T0（出厂校准温度）等于 30°C。

TS1_FMT0 为测量结果，存储在 DTS_T0VALR1 寄存器中。该值以 0.1 kHz 表示。

TS1_RAMP_COEFF 为工厂测试期间的测量结果，存储在 DTS_RAMPVALR 寄存器中。该值以 Hz/°C 表示。

- 使用 LSE 时钟时：

$$T = T_0 + ((F_{LSE} \times TS1_MFREQ / TS1_SMP_TIME) - (100 \times TS1_FMT0)) / TS1_RAMP_COEFF$$

27.3.8 采样时间

可以增加采样周期来提高测量精度。当参考频率 (REF_CLK) 接近 FM(T) 频率时，这种方法非常有用。LSE 模式下的默认值为 1 个 REF_CLK 周期，PCLK 模式下的默认值为 1 个 FM(T) 周期。

采样时间通过 DTS_CFGR1 寄存器中的 TS1_SMP_TIME 位进行配置（请参见*表 220*）。

表 220. 采样时间配置

TS1_SMP_TIME[3:0]	LSE 或 FM(T) 时钟周期
0000	1
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8

表 220. 采样时间配置（续）

TS1_SMP_TIME[3:0]	LSE 或 FM(T) 时钟周期
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

27.3.9 快速测量模式

如果不要求高精度，可以将 DTS_CFGR1 寄存器中的 Q_MEAS_OPT 设为 1 来跳过每个测量序列中所包含的校准步骤。该方法只能在选择 LSE 时钟作为参考时钟（LSREF_CLK 设为 1）的情况下使用。该模式可以缩短测量时间。

27.3.10 触发输入

温度测量可通过软件或外部事件触发。触发源可通过 DTS_CFGR1 中的 TS1_INTRIG[3:0] 位来选择。

表 221. 触发配置

名称	TS1_INTRIG[3:0]				备注
N.A	0	0	0	0	无硬件触发
ts1_trg0	0	0	0	1	lptim1_out
ts1_trg1	0	0	1	0	lptim2_out
ts1_trg2	0	0	1	1	lptim3_out
ts1_trg3	0	1	0	0	exti13
ts1_trg4	0	1	0	1	保留
ts1_trg5	0	1	1	0	
ts1_trg6	0	1	1	1	
ts1_trg7	1	0	0	0	
ts1_trg8	1	0	0	1	
ts1_trg9	1	0	1	0	
ts1_trg10	1	0	1	1	
ts1_trg11	1	1	0	0	
ts1_trg12	1	1	0	1	
ts1_trg13	1	1	1	0	
ts1_trg14	1	1	1	1	

注： *硬件触发仅在上升沿有效。*

温度传感器只能在 TS1_RDY 位置 1 时捕获硬件触发上升沿（请参见第 27.3.11 节：开关控制和就绪标志），其他情况下会忽略触发。

如果触发源动态更改，则新的触发源信号应为低电平。如果新的触发源信号为高电平，则温度传感器将检测上升沿，并据此启动测量序列。

27.3.11 开关控制和就绪标志

DTS 模块可通过将 DTS_CFGR1 寄存器中的 TS1_EN 位置 1 来使能。*温度传感器状态寄存器 (DTS_SR)* 中的 TS1_RDY 标志用于指示 DTS 模块是否准备好进行温度测量：当 TS1_RDY 位设为 1 时，可以开始测量。测量开始后，TS1_RDY 位将会复位。随后，新的测量请求将被忽略。测量完成后，TS1_RDY 位会再次置 1，以指示传感器已准备好开始新的测量。

27.3.12 温度测量序列

测量可通过软件或硬件触发启动。

软件触发

当 DTS_CFGR1 中的 TS1_INTRIG_SEL[3:0] 设为“0000”时，选择软件触发。

如果 TS1_RDY 设为 1，则将 1 写入 DTS_CFGR1 中的 TS1_START 位可启动测量。

如果 TS1_RDY 等于 0，则在 TS1_RDY 置 1 前，软件触发都不会启动测量。

如果 TS1_START 在测量完成后保持为 1，则 TS1_RDY 标志将变为 1，并重新启动测量。

硬件触发

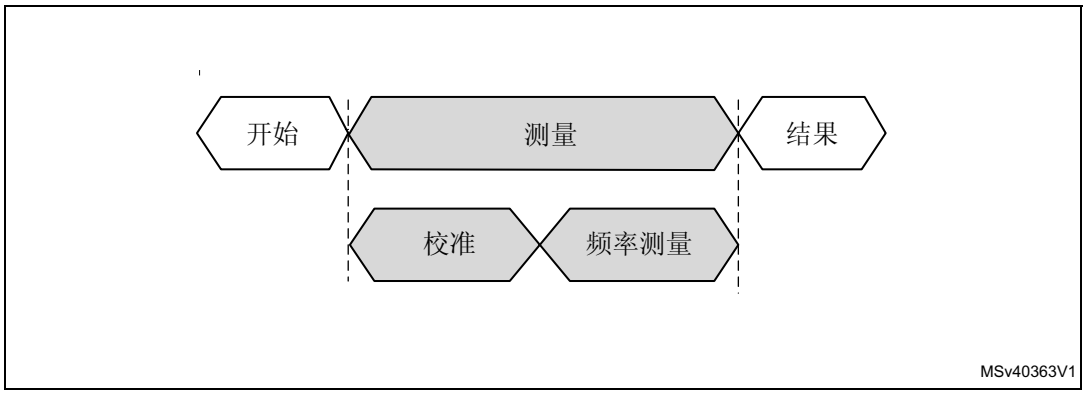
可通过 TS1_INTRIG_SEL[3:0] 位选择 4 个硬件触发信号之一。当 TS1_RDY 设为 1 时，在触发信号的上升沿启动测量。如果 TS1_RDY 为 0，则忽略上升沿。

温度测量序列

一次测量包含两个步骤：校准模拟模块和测量。触发测量后，将自动开始校准（请参见*第 27.3.5 节：校准*）。测量周期取决于以下 DTS_CFGR1 位：

- 参考时钟通过 REFCLK_SEL 位选择
- 分频比由 HSREF_CLK_DIV 位配置
- 采样时间由 TS1_SMP_TIME 位定义

图 254. 温度传感器序列



27.4 DTS 低功耗模式

表 222. 低功耗模式下的温度传感器行为

模式	说明
睡眠	仅适用于 LSE 模式。 DTS 中断可使设备退出睡眠模式。
停止	仅适用于 LSE 模式。 DTS 中断可使设备退出停止模式。

27.5 DTS 中断

可通过两种方式将 DTS 模块用作中断源。DTS 中断线可以连接到 EXTI 控制器（请参见*第 27.5.3 节：异步唤醒*）。

27.5.1 温度窗口比较器

可通过 DTS_ITR1 寄存器定义用于温度比较的阈值上限和阈值下限。如果温度数据等于/高于 TS1_HITTHD，或者等于/低于 TS1_LITTHD 位，则会生成中断，并且 DTS_SR 寄存器中的相应标志（TS1_ ITLF、TS1_ITHF、TS1_AITLF 和 TS1_AITHF）置 1（请参见*第 27.6.6 节*）。

27.5.2 同步中断

DTS 模块提供了一条全局中断输出线。在测量结束时和/或当测量结果等于/高于或等于/低于预定义阈值时，都会生成中断（请参见*第 27.5.1 节：温度窗口比较器*）。

三个中断事件可通过 DTS_ITENR 寄存器中的 3 位来选择（请参见*第 27.6.7 节*）。三个中断可以进行任意组合来使用。

DTS_SR 寄存器中的 TS1_ITEF、TS1_ITLF 和 TS1_ITHF 标志用于反映中断事件。这些位可通过 DTS_ICIFR 寄存器的相应位来复位（请参见*第 27.6.8 节*）。

27.5.3 异步唤醒

DTS 模块还提供了一条异步中断线。该中断线只能在选择 LSE 作为参考时钟 (REFCLK_SEL = 1) 的情况下使用。

该中断线可生成唤醒信号，以便在测量结束时和/或当测量结果等于/高于或等于/低于预定义阈值时，将系统从睡眠模式唤醒（请参见*第 27.5.1 节：温度窗口比较器*）。

三个异步唤醒事件可通过 DTS_ITENR 寄存器中的 3 位来选择。三个中断可以进行任意组合来使用。

DTS_SR 寄存器中的 TS1_AITEF、TS1_AITLF 和 TS1_AITHF 标志用于反映中断状态。它们可通过 DTS_ICIFR 寄存器的相应位来复位。

下表给出了各个中断位及其说明。

表 223. 中断控制位

中断事件	中断标志	使能控制位	中断清除位	退出睡眠模式	同步/异步
当测量结束时	DTS_SR 中的 TS1_ITEF	DTS_ITENR 中的 TS1_ITEEN	DTS_ICIFR 中的 TS1_CITEF	否	基于 PCLK 进行同步
当测量值等于/低于阈值下限时	DTS_SR 中的 TS1_ITLF	DTS_ITENR 中的 TS1_ITLEN	DTS_ICIFR 中的 TS1_CITLF	否	
当测量值等于/高于阈值上限时	DTS_SR 中的 TS1_ITHF	DTS_ITENR 中的 TS1_ITHEN	DTS_ICIFR 中的 TS1_CITHF	否	
当测量结束时	DTS_SR 中的 TS1_AITEF	DTS_ITENR 中的 TS1_AITEEN	DTS_ICIFR 中的 TS1_CAITEF	是	异步
当测量值等于/低于阈值下限时	DTS_SR 中的 TS1_AITLF	DTS_ITENR 中的 TS1_AITLEN	DTS_ICIFR 中的 TS1_CAITLF	是	
当测量值等于/高于阈值上限时	DTS_SR 中的 TS1_AITHF	DTS_ITENR 中的 TS1_AITHEN	DTS_ICIFR 中的 TS1_CAITHF	是	

27.6 DTS 寄存器

该外设的寄存器只能按字（32 位）进行访问。

27.6.1 温度传感器配置寄存器 1 (DTS_CFGR1)

Temperature sensor configuration register 1

DTS_CFGR1 是温度传感器 1 的配置寄存器。

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	HSREF_CLK_DIV[6:0]							Res.	Res.	Q_MEAS_OPT	REFCLK_SEL	TS1_SMP_TIME[3:0]			
	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TS1_INTRIG_SEL[3:0]				Res.	Res.	Res.	TS1_START	Res.	Res.	Res.	TS1_EN
				rw	rw	rw	rw				rw				rw

位 31 保留，必须保持复位值。

位 30:24 **HSREF_CLK_DIV[6:0]**：高速时钟分频比 (High speed clock division ratio)

这些位由软件置 1 和清零，可用于定义主时钟的分频比，以获得校准所需的低于 1 MHz 的内部频率。这些位仅在选择 PCLK 作为参考时钟 (REFCLK_SEL = 0) 的情况下才适用于校准。

0000000：无分频器

0000001：无分频器

0000010：1/2 分频比

...

1111111：1/127 分频比

位 23:22 保留，必须保持复位值。

位 21 **Q_MEAS_OPT**：快速测量选项位 (Quick measurement option bit)

该位由软件置 1 和清零，用于跳过校准步骤以提高测量速度。该位仅在将 LSE 时钟用作参考时钟 (REFCLK_SEL = 1) 的情况下生效。

0：含校准的测量

1：不含校准的测量

位 20 **REFCLK_SEL**：参考时钟选择位 (Reference clock selection bit)

该位由软件置 1 和清零，用于指示参考时钟是高速时钟 (PCLK) 还是低速时钟 (LSE)。

0：高速参考时钟 (PCLK)

1：低速参考时钟 (LSE)

位 19:16 **TS1_SMP_TIME[3:0]**：温度传感器 1 的采样时间 (Sampling time for temperature sensor 1)

这些位用于增加采样时间以提高测量精度。

当选择 PCLK 时钟作为参考时钟 (REFCLK_SEL = 0) 时，以 CLK_PTAT 的 TS1_SMP_TIME 周期执行测量。

当选择 LSE 作为参考时钟 (REFCLK_SEL = 1)时，以 LSE 的 TS1_SMP_TIME 周期执行测量。

位 15:12 保留，必须保持复位值。

位 11:8 **TS1_INTRIG_SEL[3:0]**：温度传感器 1 的输入触发选择位 (Input trigger selection bit for temperature sensor 1)

这些位由软件置 1 和清零，用于选择温度测量的输入触发。请参见*第 27.3.10 节：触发输入*。

位 7:5 保留，必须保持复位值。

位 4 **TS1_START**：启动温度传感器 1 的频率测量 (Start frequency measurement on temperature sensor 1)

该位由软件置 1 和清零，

0：无软件触发。

1：软件触发频率测量。（仅当 TS1 就绪时）。

位 3:1 保留，必须保持复位值。

位 0 **TS1_EN**：温度传感器 1 使能位 (Temperature sensor 1 enable bit)

该位由软件置 1 和清零，

0：禁止温度传感器 1

1：使能温度传感器 1

注意： 使能时，温度传感器将在一段特定的延时后激活。TS1_RDY 标志在传感器就绪时置 1。

27.6.2 温度传感器 T0 值寄存器 1 (DTS_T0VALR1)

Temperature sensor T0 value register 1

DTS_T0VALR1 包含温度传感器 1 的出厂校准温度值 (T0)。复位值经过出厂微调。

偏移地址：0x08

复位值：0x000X XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_T0[1:0]	
														r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS1_FMT0[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:18 保留，必须保持复位值。

位 17:16 **TS1_T0[1:0]**：温度传感器 1 的 T0 温度工程值 (Engineering value of the T0 temperature for temperature sensor 1)

00：30°C

01：130°C

其他值：保留，不得使用。

位 15:0 **TS1_FMT0[15:0]**：温度传感器 1 在 T0 测得的频率工程值 (Engineering value of the frequency measured at T0 for temperature sensor 1)

该值以 0.1 kHz 表示。

27.6.3 温度传感器斜坡值寄存器 (DTS_RAMPVALR)

Temperature sensor ramp value register

DTS_RAMPVALR 是温度传感器的斜坡系数。复位值经过出厂微调。

偏移地址：0x10

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS1_RAMP_COEFF[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **TS1_RAMP_COEFF[15:0]**：温度传感器 1 的斜坡系数工程值 (Engineering value of the ramp coefficient for the temperature sensor 1)

该值以 Hz/°C 表示。

27.6.4 温度传感器中断阈值寄存器 1 (DTS_ITR1)

Temperature sensor interrupt threshold register 1

DTS_ITR1 包含传感器 1 的阈值。

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TS1_HITTHD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS1_LITTHD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **TS1_HITTHD[15:0]**：温度传感器 1 的中断阈值上限 (High interrupt threshold for temperature sensor 1)

这些位由软件置 1 和清零，用于指示在发出中断信号之前可以达到的最高值。

位 15:0 **TS1_LITTHD[15:0]**：温度传感器 1 的中断阈值下限 (Low interrupt threshold for temperature sensor 1)

这些位由软件置 1 和清零，用于指示在发出中断信号之前可以达到的最低值。

27.6.5 温度传感器数据寄存器 (DTS_DR)

Temperature sensor data register

DTS_DR 包含用于计算 FM(T) 频率的 REF_CLK 周期数。

偏移地址：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS1_MFREQ[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **TS1_MFREQ[15:0]**：温度传感器 1 的计数器输出值 (Value of the counter output value for temperature sensor 1)

27.6.6 温度传感器状态寄存器 (DTS_SR)

Temperature sensor status register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS1_RDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_AITHF	TS1_AITLF	TS1_AITEF	Res.	TS1_ITHF	TS1_ITLF	TS1_ITEF
r									r	r	r		r	r	r

位 31:16 保留，必须保持复位值。

位 15 **TS1_RDY**：温度传感器 1 就绪标志 (Temperature sensor 1 ready flag)
该位由硬件置 1 和复位，
用于指示正在进行测量。
0：温度传感器 1 繁忙
1：温度传感器 1 就绪

位 14:7 保留，必须保持复位值。

位 6 **TS1_AITHF**：温度传感器 1 阈值上限异步中断标志 (Asynchronous interrupt flag for high threshold on temperature sensor 1)
当达到阈值上限时，该位由硬件置 1。
通过软件向 DTS_ICIFR 寄存器中的 TS1_CAITHF 位写入 1 可将该位清零。
0：温度传感器 1 未达到阈值上限
1：温度传感器 1 达到阈值上限
注意：该位只有在 TS1_AITHFEN 位置 1 时才有效

位 5 **TS1_AITLF**：温度传感器 1 阈值下限异步中断标志 (Asynchronous interrupt flag for low threshold on temperature sensor 1)
当达到阈值下限时，该位由硬件置 1。
通过软件向 DTS_ICIFR 寄存器中的 TS1_CAITLF 位写入 1 可将该位清零。
0：温度传感器 1 未达到阈值下限
1：温度传感器 1 达到阈值下限
注意：该位只有在 TS1_AITLFEN 位置 1 时才有效

位 4 **TS1_AITEF**：温度传感器 1 测量结束异步中断标志 (Asynchronous interrupt flag for end of measure on temperature sensor 1)
当温度测量完成时，该位由硬件置 1。
通过软件向 DTS_ICIFR 寄存器中的 TS1_CAITEF 位写入 1 可将该位清零。
0：未检测到温度传感器 1 测量结束
1：检测到温度传感器 1 测量结束
注意：该位只有在 TS1_AITEFEN 位置 1 时才有效

位 3 保留，必须保持复位值。

位 2 **TS1_ITHF**：温度传感器 1 阈值上限中断标志（基于 PCLK 进行同步）(Interrupt flag for high threshold on temperature sensor 1, synchronized on PCLK)
当设置并达到阈值上限时，该位由硬件置 1。
通过软件向 DTS_ICIFR 寄存器中的 TS1_CITHF 位写入 1 可将该位清零。
0：温度传感器 1 未达到阈值上限
1：温度传感器 1 达到阈值上限
注意：该位只有在 TS1_ITHFEN 位置 1 时才有效

位 1 **TS1_ITLF**：温度传感器 1 阈值下限中断标志（基于 PCLK 进行同步）(Interrupt flag for low threshold on temperature sensor 1, synchronized on PCLK)

当设置并达到阈值下限时，该位由硬件置 1。
通过软件向 DTS_ICIFR 寄存器中的 TS1_CITLF 位写入 1 可将该位清零。
0：温度传感器 1 未达到阈值下限
1：温度传感器 1 达到阈值下限
注意：该位只有在 TS1_ITLFEN 位置 1 时才有效

位 0 **TS1_ITEF**：温度传感器 1 测量结束中断标志（基于 PCLK 进行同步）(Interrupt flag for end of measurement on temperature sensor 1, synchronized on PCLK)

当温度测量完成时，该位由硬件置 1。
通过软件向 DTS_ICIFR 寄存器中的 TS2_CITEF 位写入 1 可将该位清零。
0：未检测到温度传感器 1 测量结束
1：检测到温度传感器 1 测量结束
注意：该位只有在 TS1_ITEFEN 位置 1 时才有效

27.6.7 温度传感器中断使能寄存器 (DTS_ITENR)

Temperature sensor interrupt enable register

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_ AITHEN	TS1_ AITLEN	TS1_ AITEEN	Res.	TS1_ ITHEN	TS1_ ITLEN	TS1_ ITEEN
									rw	rw	rw		rw	rw	rw

位 31:7 保留，必须保持复位值。

位 6 **TS1_AITHEN**：温度传感器 1 阈值上限异步中断使能标志 (Asynchronous interrupt enable flag on high threshold for temperature sensor 1)

该位由软件置 1 和清零，
用于使能在温度高于阈值上限时生成异步中断（仅当 REFCLK_SEL = 1 时）。
0：禁止温度传感器 1 阈值上限异步中断
1：使能温度传感器 1 阈值上限异步中断

位 5 **TS1_AITLEN**：温度传感器 1 阈值下限异步中断使能标志 (Asynchronous interrupt enable flag for low threshold on temperature sensor 1)

该位由软件置 1 和清零，
用于在温度低于阈值下限时使能异步中断（仅当 REFCLK_SEL = 1 时）。
0：禁止温度传感器 1 阈值下限异步中断
1：使能温度传感器 1 阈值下限异步中断

位 4 **TS1_AITEEN**：温度传感器 1 测量结束异步中断使能标志 (Asynchronous interrupt enable flag for end of measurement on temperature sensor 1)

该位由软件置 1 和清零，
用于使能测量结束异步中断（仅当 REFCLK_SEL = 1 时）。
0：禁止温度传感器 1 测量结束异步中断
1：使能温度传感器 1 测量结束异步中断

位 3 保留，必须保持复位值。

位 2 **TS1_ITHEN**：温度传感器 1 阈值上限中断使能标志（基于 PCLK 进行同步）(Interrupt enable flag for high threshold on temperature sensor 1, synchronized on PCLK)

该位由软件置 1 和清零，
用于使能在测量值等于/高于阈值上限时生成中断。
0：禁止温度传感器 1 阈值上限同步中断
1：使能温度传感器 1 阈值上限同步中断

位 1 **TS1_ITLEN**：温度传感器 1 阈值下限中断使能标志（基于 PCLK 进行同步）(Interrupt enable flag for low threshold on temperature sensor 1, synchronized on PCLK)

该位由软件置 1 和清零，
用于使能在测量值等于/低于阈值下限时生成同步中断。
0：禁止温度传感器 1 阈值下限同步中断
1：使能温度传感器 1 阈值下限同步中断

位 0 **TS1_ITEEN**：温度传感器 1 测量结束中断使能标志（基于 PCLK 进行同步）(Interrupt enable flag for end of measurement on temperature sensor 1, synchronized on PCLK)

该位由软件置 1 和清零，
用于使能测量结束同步中断。
0：禁止温度传感器 1 测量结束同步中断
1：使能温度传感器 1 测量结束同步中断

27.6.8 温度传感器清除中断标志寄存器 (DTS_ICIFR)

Temperature sensor clear interrupt flag register

DTS_ICIFR 是中断标志控制寄存器。

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_CAITHF	TS1_CAITLF	TS1_CAITEF	Res.	TS1_CITHF	TS1_CITLF	TS1_CITEF
									rc_w1	rc_w1	rc_w1		rc_w1	rc_w1	rc_w1

位 31:7 保留，必须保持复位值。

位 6 **TS1_CAITHF**：温度传感器 1 阈值上限异步中断清零标志 (Asynchronous interrupt clear flag for high threshold on temperature sensor 1)

向该位写入 1 可将 DTS_SR 寄存器中的 TS1_AITHF 标志清零。

位 5 **TS1_CAITLF**：温度传感器 1 阈值下限异步中断清零标志 (Asynchronous interrupt clear flag for low threshold on temperature sensor 1)

向该位写入 1 可将 DTS_SR 寄存器中的 TS1_AITLF 标志清零。

位 4 **TS1_CAITEF**：一次写入位，用于清零温度传感器 1 测量结束异步 IT 标志。

向该位写入 1 可将 DTS_SR 寄存器中的 TS1_AITEF 标志清零。

位 3 保留，必须保持复位值。

位 2 **TS1_CITHF**：温度传感器 1 阈值上限中断清零标志 (Interrupt clear flag for high threshold on temperature sensor 1)

向该位写入 1 可将 DTS_SR 寄存器中的 TS1_ITHF 标志清零。

位 1 **TS1_CITLF**：温度传感器 1 阈值下限中断清零标志 (Interrupt clear flag for low threshold on temperature sensor 1)

向该位写入 1 可将 DTS_SR 寄存器中的 TS1_ITLF 标志清零。

位 0 **TS1_CITEF**：温度传感器 1 测量结束中断清零标志 (Interrupt clear flag for end of measurement on temperature sensor 1)

向该位写入 1 可将 DTS_SR 寄存器中的 TS1_ITEF 标志清零。

27.6.9 温度传感器选项寄存器 (DTS_OR)

Temperature sensor option register

DTS_OR 包含通用选项位。

偏移地址：0x2C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TS_OP 31	TS_OP 30	TS_OP 29	TS_OP 28	TS_OP 27	TS_OP 26	TS_OP 25	TS_OP 24	TS_OP 23	TS_OP 22	TS_OP 21	TS_OP 20	TS_OP 19	TS_OP 18	TS_OP 17	TS_OP 16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS_OP 15	TS_OP 14	TS_OP 13	TS_OP 12	TS_OP 11	TS_OP 10	TS_OP 9	TS_OP 8	TS_OP 7	TS_OP 6	TS_OP 5	TS_OP 4	TS_OP 3	TS_OP 2	TS_OP 1	TS_OP 0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **TS_OP[31:0]**：通用选项位 (general purpose option bits)

27.6.10 DTS 寄存器映射

下表对温度传感器寄存器进行了汇总。

表 224. DTS 寄存器映射和复位值

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**DTS_ CFGR1**	Res.	HSREF_CLK_DIV [6:0]							Res.	Res.	Q_MEAS_OPT	REFCLK_SEL	TS1_ SMP_TIME [3:0]				Res.	Res.	Res.	Res.	TS1_ INTRIG_SEL [3:0]				Res.	Res.	Res.	TS1_START	Res.	Res.	Res.	TS1_EN
	Reset value		0	0	0	0	0	0	0			0	0	0	0	0	0					0	0	0	0				0				0
0x04	Reserved																																
0x08	**DTS_ T0VALR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_T0[1:0]		TS1_FMT0[15:0]															
	Reset value															X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x0C	Reserved																																
0x10	**DTS_ RAMPVALR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_RAMP_COEFF[15:0]															
	Reset value																	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x14	**DTS_ITR1**	TS1_HITTHD[15:0]																TS1_LITTHD[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	Reserved																																
0x1C	**DTS_DR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_MFREQ[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**DTS_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_RDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_AITHF	TS1_AITLF	TS1_AITEF	Res.	TS1_ITHF	TS1_ITLF	TS1_ITEF
	Reset value																	0									0	0	0		0	0	0
0x24	**DTS_ITENR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_AITHEN	TS1_AITLEN	TS1_AITEEN	Res.	TS1_ITHEN	TS1_ITLEN	TS1_ITEEN
	Reset value																										0	0	0		0	0	0
0x28	**DTS_ICIFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_CAITHF	TS1_CAITLF	TS1_CAITEF	Res.	TS1_CITHF	TS1_CITLF	TS1_CITEF
	Reset value																										0	0	0		0	0	0
0x2C	**DTS_OR**	TS_OP31	TS_OP30	TS_OP29	TS_OP28	TS_OP27	TS_OP26	TS_OP25	TS_OP24	TS_OP23	TS_OP22	TS_OP21	TS_OP20	TS_OP19	TS_OP18	TS_OP17	TS_OP16	TS_OP15	TS_OP14	TS_OP13	TS_OP12	TS_OP11	TS_OP10	TS_OP9	TS_OP8	TS_OP7	TS_OP6	TS_OP5	TS_OP4	TS_OP3	TS_OP2	TS_OP1	TS_OP0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

28 电压参考缓冲器 (VREFBUF)

28.1 简介

设备内置有电压参考缓冲器，既可用作 ADC 的参考电压，也可通过 VREF+ 引脚用作外部元件的参考电压。

28.2 VREFBUF 实现

下表列出了 VREFBUF 电压的典型值：

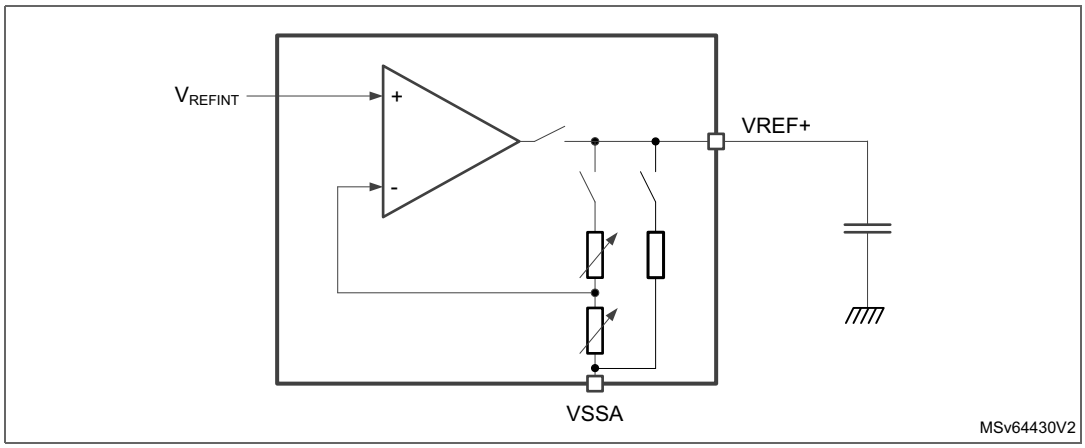
表 225. VREFBUF 典型值

符号	值
VREFBUF0	2.5 V
VREFBUF1	2.048 V
VREFBUF2	1.8 V
VREFBUF3	1.65 V

注：　更多详细信息，请参见产品数据手册。

28.3 VREFBUF 功能描述

图 255. VREFBUF 框图



内部电压参考缓冲器是一个具有可编程增益的运算放大器。放大器输入连接到内部参考电压 V_{REFINT}。VREFBUF 支持四个电压[(a)]，这些电压通过 VREFBUF_CSR 寄存器中的 VRS 位进行配置：

- VRS = 000：选择 VREFBUF0 电压。
- VRS = 001：选择 VREFBUF1 电压。
- VRS = 010：选择 VREFBUF2 电压。
- VRS = 011：选择 VREFBUF3 电压。

内部参考电压配置为四种不同模式，具体取决于 ENVR 和 HIZ 位的配置。下表列出了这些模式：

表 226. VREF 缓冲器模式

ENVR	HIZ	VREF 缓冲器配置
0	0	VREFBUF 缓冲器关闭模式： – V_{REF+} 引脚下拉到 V_{SSA}
0	1	外部参考电压模式（默认值）： – VREFBUF 缓冲器关闭 – V_{REF+} 引脚输入模式
1	0	内部参考电压模式： – VREFBUF 缓冲器开启 – V_{REF+} 引脚连接到 VREFBUF 缓冲器输出
1	1	保持模式： – VREF 在无输出缓冲器的情况下使能，VREF+ 引脚电压通过外部电容保持 – 禁止 VRR 检测，VRR 位保持最后一个状态

通过将 VREFBUF_CSR 寄存器中的 ENVR 位置 1 并将 HIZ 位清零使能 VREFBUF 后，用户必须等待至 VRR 位置 1，指示参考电压输出已达到预期值。

28.4 VREFBUF 校准

VREFBUF 输出电压由 ST 在工厂校准。复位时，以及每次更改 VRS 设置时，校准数据会自动加载到 TRIM 寄存器。

用户可以选择通过直接更改 TRIM 寄存器位来校准输出电压。在这种情况下，VRS 设置对 TRIM 寄存器没有其他影响，直到设备复位。

28.5 VREFBUF 寄存器

28.5.1 VREFBUF 控制和状态寄存器 (VREFBUF_CSR)

VREFBUF control and status register

偏移地址：0x00

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VRS[2:0]			VRR	Res.	HIZ	ENVR
									rw	rw	rw	r		rw	rw

a. 最小 V_{DDA} 电压取决于 VRS 设置，请参见产品数据手册。

位 31:7 保留，必须保持复位值。

位 6:4 **VRS[2:0]**：参考电压调节 (Voltage reference scale)
这些位选择由电压参考缓冲器生成的值。
VRS = 000：选择 VREFBUF0 电压。
VRS = 001：选择 VREFBUF1 电压。
VRS = 010：选择 VREFBUF2 电压。
VRS = 011：选择 VREFBUF3 电压。
其他值：保留
注意：关于每个 VREFBUFx 电压的设置值，请参见产品数据手册。
只有在 VREFBUF 处于禁止状态 (ENVR=0) 时，软件才能编程该位字段。

位 3 **VRR**：电压参考缓冲器就绪 (Voltage reference buffer ready)
0：参考电压缓冲器输出未就绪。
1：参考电压缓冲器输出已达到请求值。

位 2 保留，必须保持复位值。

位 1 **HIZ**：高阻抗模式 (High impedance mode)
该位控制模拟开关是否连接到 V_{REF+} 引脚。
0：V_{REF+} 从内部连接到电压参考缓冲器输出。
1：V_{REF+} 引脚为高阻抗。
关于 ENVR 位配置及模式说明，请参见*表 226：VREF 缓冲器模式*。

位 0 **ENVR**：电压参考缓冲器模式使能 (Voltage reference buffer mode enable)
该位用于使能电压参考缓冲器模式。
0：禁止内部参考电压模式（外部参考电压模式）。
1：使能内部参考电压模式（参考缓冲器使能或保持模式）。

28.5.2 VREFBUF 校准控制寄存器 (VREFBUF_CCR)

VREFBUF calibration control register

偏移地址：0x04

复位值：0x0000 00XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM[5:0]					
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持复位值。

位 5:0 **TRIM[5:0]**：校准代码 (Trimming code)

TRIM 代码是一个 6 位无符号数据（最小 000000，最大 111111），基于下述机制进行设置和更新。

复位：

TRIM[5:0] 将自动使用生产测试时存储于 OTP 内存中的校准值（对应于 VRS = 0）进行初始化。

更改 VRS：

TRIM[5:0] 将自动使用生产测试时存储于 OTP 内存中的校准值（对应于 VRS 设置）进行初始化。

写入 TRIM[5:0]：

用户可以将 TRIM[5:0] 修改为任意值。这将永久性地禁止使用 VRS 控制校准值（直到设备复位）。

注意：如果用户应用程序执行校准，则校准代码必须从 000000 开始，按升序一直到 111111。

28.5.3 VREFBUF 寄存器映射

下表列出了 VREFBUF 寄存器映射和复位值。

表 227. VREFBUF 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	VREFBUF_CSR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	VRS[2:0]			VRR	Res	HIZ	ENVR
	Reset value																										0	0	0	0		1	0
0x04	VREFBUF_CCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TRIM[5:0]					
	Reset value																											x	x	x	x	x	x

有关寄存器边界地址的信息，请参见*第 2.5 节*。

29 ∑Δ 调制器数字滤波器 (DFSDM)

29.1 简介

ΣΔ 调制器数字滤波器 (DFSDM) 是一种高性能模块，专用于连接外部 ΣΔ 调制器。它具有多达 4 个外部数字串行接口（通道）以及多达 2 个具有灵活 Σ - Δ 流数字处理功能的数字滤波器，可提供高达 24 位 ADC 最终分辨率。DFSM 还具有从内部 ADC 或外部设备输入并行数据的功能。

外部 ΣΔ 调制器将其模拟输入的模拟值转换为数字数据流。此数字数据流通过串行接口发送到 DFSDM 输入通道。DFSDM 支持多种标准，可连接不同的 ΣΔ 调制器输出：SPI 接口和曼彻斯特编码的 1 线接口（两者均具有可调参数）。DFSDM 模块支持连接多达 4 个复用输入数字串行通道，这些通道可与多达 2 个 DFSDM 模块共享。DFSDM 模块还支持来自多达 4 个内部 16 位数据通道（可选择来自内部 ADC 或来自设备内存）的并行数据输入。

DFSDM 将输入数据流转换为最终数字数据字，该数字字表示 ΣΔ 调制器模拟输入上的模拟输入值。该转换基于以下可配置的数字处理而完成：对输入串行数据流进行数字滤波和抽取。

转换速度和分辨率可根据以下可配置的数字处理参数进行调整：滤波器类型、滤波器阶数、滤波器长度和积分器长度。最大输出数据分辨率高达 24 位。有两种转换模式：单次转换模式和连续转换模式。数据可通过 DMA 自动存储在系统 RAM 缓冲区中，因此降低了软件开销。

灵活的定时器触发系统可用于控制 DFSDM 的转换开始。此时序控制能够触发同步转换或在转换之间插入可编程延迟。

DFSDM 具有模拟看门狗功能。模拟看门狗可以分配给任何输入通道数据流或最终输出数据。模拟看门狗本身可以对输入数据流进行数字滤波，以达到被监视数据的所需速度和分辨率。

为了检测控制应用中的短路情况，提供了一个短路检测器。该模块监视各个输入通道数据流是否在定义的持续时间内保持稳定（即输入数据流中有若干 0 或 1）。

极值检测器模块监视最终输出数据并存储最大和最小输出数据值。存储的极值可通过软件重新启动。

支持两种功率模式：正常模式和停止模式。

29.2 DFSDM主要特性

- 多达 4 个复用输入数字串行通道：
  - 可配置的 SPI 接口，用于连接各种 ΣΔ 调制器
  - 支持可配置的曼彻斯特编码单线接口
  - 用于 ΣΔ 调制器的时钟输出
- 来自多达 4 个内部数字并行通道的可选输入：
  - 分辨率高达 16 位的输入
  - 内部源：ADC 数据或内存（CPU/DMA 写）数据流
- 可调节的数字信号处理：
  - $Sinc^x$ 滤波器：滤波器阶数/类型 (1..5)，过采样率（高达 1..1024）
  - 积分器：过采样率 (1..256)
- 高达 24 位的输出数据分辨率：
  - 最终数据的右移位器（0..31 位）
- 有符号输出数据格式
- 自动数据偏移校正（偏移值由用户存储在寄存器中）
- 连续或单次转换
- 可通过以下途径同步启动转换：
  - 软件触发
  - 内部定时器
  - 外部事件
  - 使用第一个 DFSDM 滤波器 (DFSDM_FLT0) 同步启动转换
- 模拟看门狗功能：
  - 下限和上限数据阈值寄存器
  - 自带可配置的 $Sinc^x$ 数字滤波器（阶数 = 1..3，过采样率 = 1..32）
  - 输入来自输出数据寄存器或来自一个或多个输入数字串行通道
  - 独立于标准转换的连续监视
- 短路检测器，用于检测饱和的模拟输入值（下限和上限）：
  - 高达 8 位计数器，用于检测输入数据流中 1..256 个连续的 0 或 1
  - 连续监视各个通道（4 个串行通道收发器输出）
- 当发生模拟看门狗事件或短路检测器事件时，生成断路
- 极值检测器：
  - 存储最小和最大输出数据值
  - 由软件更新
- 脉冲跳跃功能支持波束形成应用（类似延迟线的行为）
- DMA 可用于读取转换数据
- 中断：结束转换，溢出，模拟看门狗，短路，通道时钟缺失
- “常规”或“注入”转换：
  - “常规”转换可随时进行请求，即使在连续模式下亦如此，且不会对“注入”转换的时序产生影响
  - “注入”转换可实现精确时序和高转换优先级

29.3 DFSDM 实现

本部分介绍在 DFSDMx 中实现的配置。

表 228. DFSDM1 实现

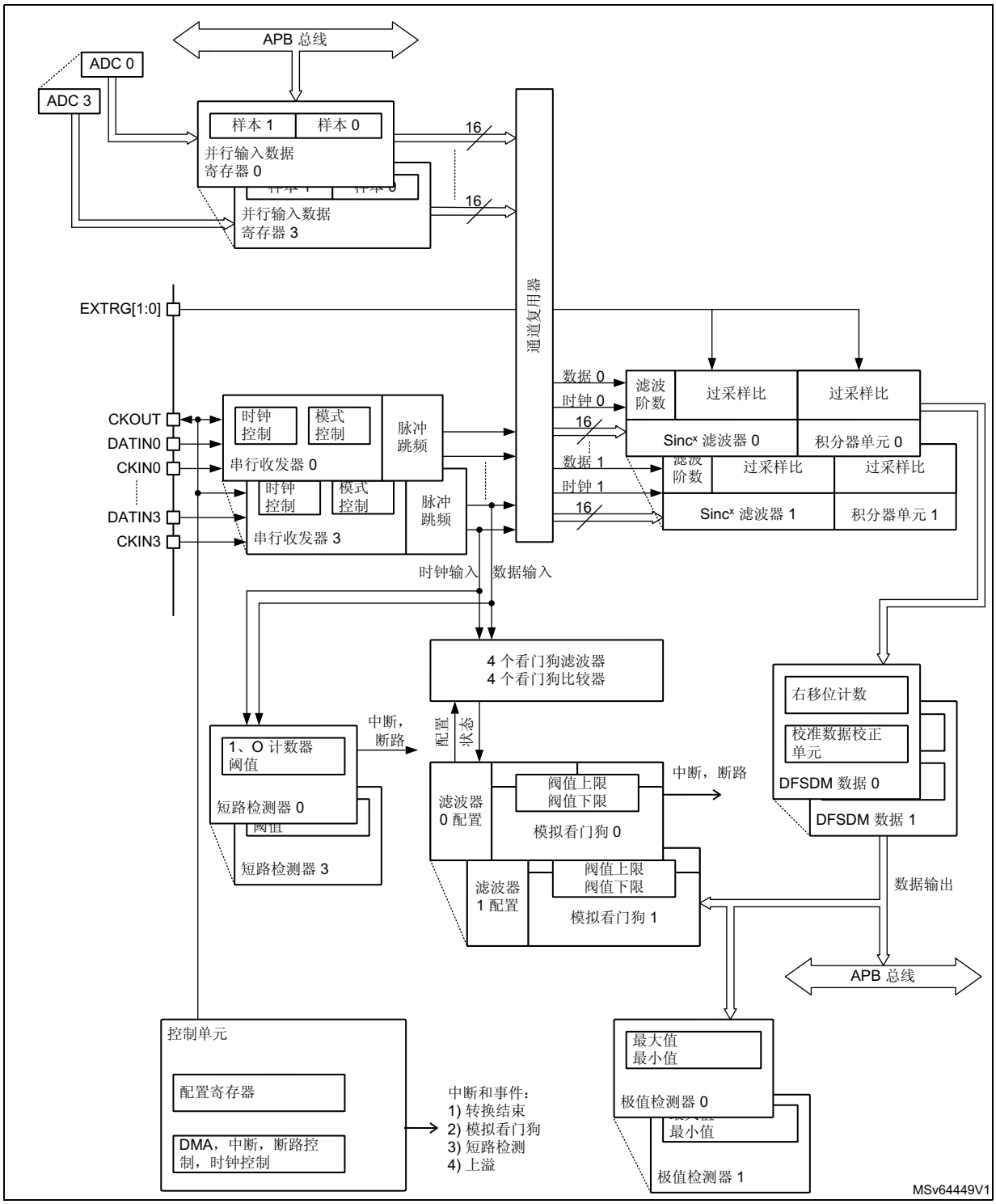
DFSDM 功能	DFSDM1
通道数	4
过滤器数	2
来自内部 ADC 的输入	X
支持的触发源	32(1)
脉冲跳跃	X

1. 有关可用的触发源，请参见*表 231：DFSDM 触发器连接*。

29.4 DFSDM 功能描述

29.4.1 DFSDM 框图

图 256. 单个 DFSDM 框图



1. 本示例显示了 *2* 个 DFSDM 滤波器和 4 个输入通道。

29.4.2 DFSDM 引脚和内部信号

表 229. DFSDM 外部引脚

名称	信号类型	注释
VDD	电源	数字电源。
VSS	电源	数字接地电源。
CKIN[3:0]	时钟输入	从外部 ΣΔ 调制器提供的时钟信号。FT input
DATIN[3:0]	数据输入	从外部 ΣΔ 调制器提供的数据信号。FT input
CKOUT	时钟输出	时钟输出，用于为外部 ΣΔ 调制器提供时钟信号。
EXTRG[1:0]	外部触发信号	来自两个 EXTI 信号的输入触发，用于启动模拟转换（来自 GPIO：EXTI11 和 EXTI15）。

表 230. DFSDM 内部信号

名称	信号类型	注释
dfsdm_jtrg[31:0]	内部/外部触发信号	来自内部/外部触发源的输入触发，用于启动模拟转换（来自内部源：同步输入，来自外部源：使用时钟同步的异步输入）。有关详细信息，请参见*表 231*。
dfsdm_break[3:0]	断路信号输出	由模拟看门狗或短路检测器触发的断路信号事件
dfsdm_dma[1:0]	DMA 请求信号	来自各个 DFSDM_FLTx (x=0..1) 的 DMA 请求信号：结束注入转换事件。
dfsdm_it[1:0]	中断请求信号	各个 DFSDM_FLTx (x=0..1) 的中断信号
dfsdm_dat_adc[15:0]	ADC输入数据	多达 4 个内部 ADC 数据总线用作并行输入。

表 231. DFSDM 触发器连接

触发信号名称	触发源
dfsdm_jtrg0	TIM1_TRGO
dfsdm_jtrg1	TIM1_TRGO2
dfsdm_jtrg2	TIM8_TRGO
dfsdm_jtrg3	TIM8_TRGO2
dfsdm_jtrg4	TIM3_TRGO
dfsdm_jtrg5	TIM4_TRGO
dfsdm_jtrg6	TIM16_OC1
dfsdm_jtrg7	TIM6_TRGO
dfsdm_jtrg8	TIM7_TRGO
dfsdm_jtrg[23:9]	保留
dfsdm_jtrg24	EXTI11
dfsdm_jtrg25	EXTI15
dfsdm_jtrg26	LPTIMER1

表 231. DFSDM 触发器连接（续）

触发信号名称	触发源
dfsdm_jtrg27	LPTIMER2
dfsdm_jtrg28	LPTIMER3
dfsdm_jtrg[31:29]	保留

表 232. DFSDM 断路连接

断路名称	断路目标
dfsdm_break[0]	TIM1/TIM15 断路
dfsdm_break[1]	TIM1 断路 2/TIM16 断路
dfsdm_break[2]	TIM8/TIM17 断路
dfsdm_break[3]	TIM8 断路 2

29.4.3 DFSDM 复位和时钟

DFSDM 开关控制

通过将 DFSDM_CH0CFGR1 寄存器的 DFSDMEN 置 1，全局使能 DFSDM 接口。DFSDM 被全局使能后，所有输入通道 y (y=0..3) 和数字滤波器 DFSDM_FLTx (x=0..1) 会在其使能位（DFSDM_CHyCFGR1 中的通道使能位 CHEN 和 DFSDM_FLTxCR1 中的 DFSDM_FLTx 使能位 DFEN）置 1 时开始工作。

数字滤波器 x DFSDM_FLTx (x=0..1) 通过将 DFSDM_FLTxCR1 寄存器的 DFEN 置 1 来使能。使能 DFSDM_FLTx (DFEN=1) 后，$Sinc^x$ 数字滤波器单元和积分器单元会重新初始化。

通过将 DFEN 清零，正在进行的所有转换都会立即停止，DFSDM_FLTx 会置于停止模式。除 DFSDM_FLTxAWSR 和 DFSDM_FLTxISR（均被复位）外，所有寄存器设置保持不变。

通过将 DFSDM_CHyCFGR1 寄存器的 CHEN 置 1 使能通道 y (y=0..3)。通道使能后，会从外部 ΣΔ 调制器或并行内部数据源（ADC 或内存的 CPU/DMA 写操作）接收串行数据。

在停止系统时钟，使设备进入停止模式之前，必须全局禁止 DFSDM（通过在 DFSDM_CH0CFGR1 中设置 DFSDMEN = 0）。

DFSDM 时钟

内部 DFSDM 时钟 $f_{DFSDMCLK}$ 用于驱动通道收发器、数字处理模块（数字滤波器、积分器）和后续附加模块（模拟看门狗、短路检测器、极值检测器、控制模块），此时钟由 RCC 模块生成，源自系统时钟 SYSCLK 或外设时钟 PCLK2（请参见*第 10.8.79 节：RCC SAI1 内核时钟选择寄存器 (RCC_SAI1CKSELR)*）。DFSDM 时钟在停止模式下自动停止（对于所有 DFSDM_FLTx，x=0..1，DFEN = 0 时）。

DFSDM 串行通道收发器可以接收外部串行时钟，以采样外部串行数据流。如果使用标准 SPI 编码，内部 DFSDM 时钟必须至少比外部串行时钟快 4 倍；如果使用曼彻斯特编码，则须比外部串行时钟快 6 倍。

DFSDM 可提供一个外部输出时钟信号，以驱动外部 ΣΔ 调制器时钟输入。它通过 CKOUT 引脚提供。该输出时钟信号必须在给定设备手册中的指定范围之内，可以通过分频器取自 DFSDM 时钟，也可以取自音频时钟（请参见 DFSDM_CH0CFGR1 寄存器 CKOUTSRC 位），分频器可以编程，范围为 2 - 256（DFSDM_CH0CFGR1 寄存器中的 CKOUTDIV）。音频时钟源为 SAI1 时钟，可通过 RCC 配置中 SAI1SEL[1:0] 字段进行选择（请参见）。

29.4.4 串行通道收发器

有 4 个复用串行数据通道，可以通过各个滤波器、模拟看门狗或者短路检测器将其选择用于转换。这些串行收发器接收来自外部 ΣΔ 调制器的数据流。数据流能以 SPI 格式或曼彻斯特编码格式进行发送（请参见 DFSDM_CHyCFGR1 寄存器的 SITP[1:0] 位）。
通过将 DFSDM_CHyCFGR1 寄存器的 CHEN 置 1 可以使能通道。

通道输入选择

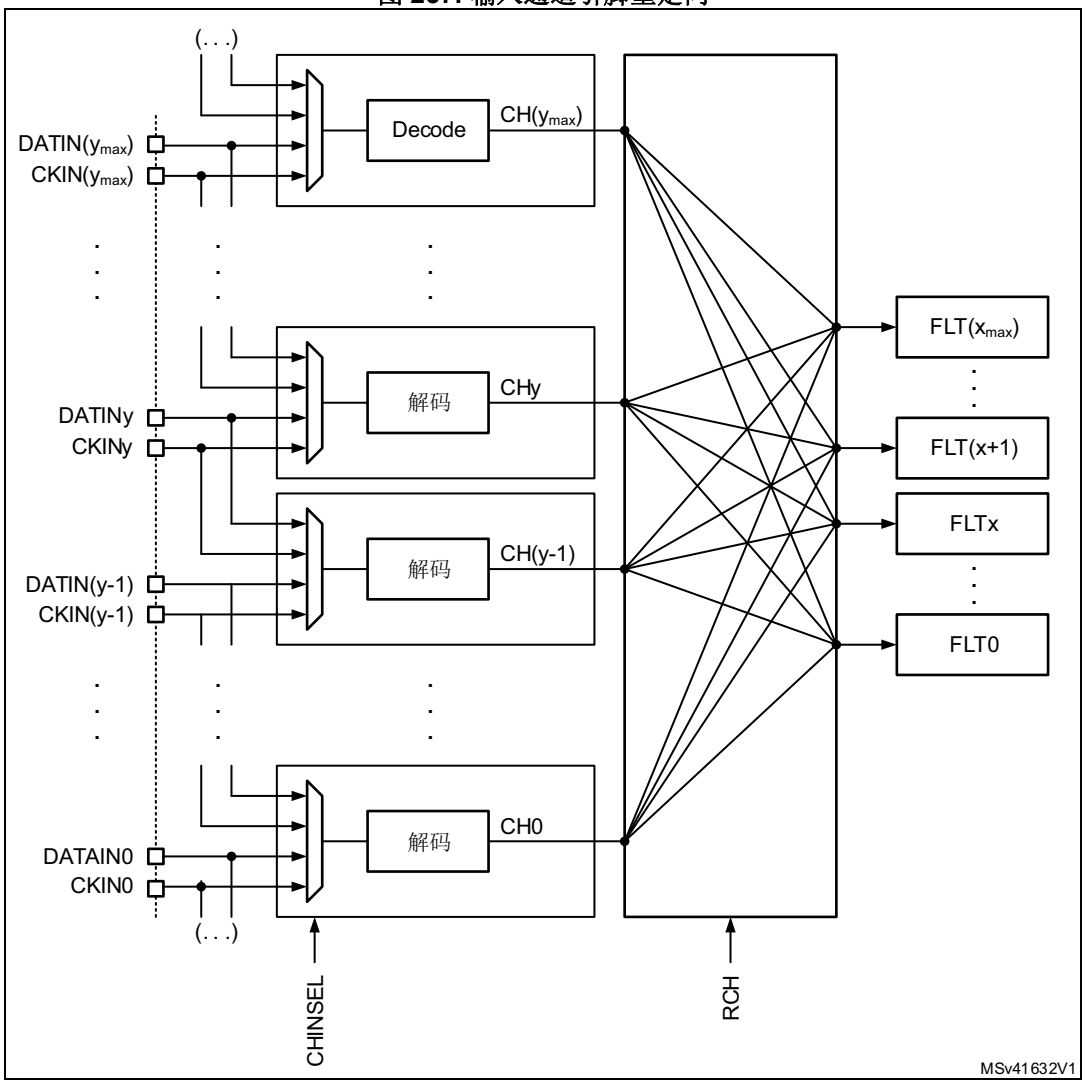
DATINy 和 CKINy 引脚的串行输入（数据和时钟信号）可从后续通道引脚重定向。此串行输入通道重定向可通过 DFSDM_CHyCFGR1 寄存器中的 CHINSEL 位设置。

通道重定向可用于从 PDM（脉冲密度调制）立体声麦克风中采集音频数据。PDM 立体声麦克风包含一个数据和一个时钟信号。数据信号为左右音频通道提供信息（时钟上升沿采样用于左通道，时钟下降沿采样用于右通道）。

PDM 麦克风输入串行通道的配置：

- PDM 麦克风信号（数据、时钟）将连接到 DFSDM 输入串行通道 y（DATINy，CKOUT）引脚。
- 通道 y 将被配置：CHINSEL = 0（来自给定通道引脚的输入：DATINy，CKINy）。
- 通道（y-1）（模 4）将被配置：CHINSEL = 1（输入来自后续通道 ((y-1)+1) 引脚：DATINy，CKINy）。
- 通道 y：SITP[1:0] = 0（上升沿选通数据）=> 通道 y 的左音频通道。
- 通道 (y-1)：SITP[1:0] = 1（下降沿选通数据）=> 通道 y-1 的右音频通道。
- 两个 DFSDM 滤波器将分配至通道 y 和通道 (y-1)（用于对 PDM 麦克风的左右通道进行滤波）。

图 257. 输入通道引脚重定向



输出时钟生成

可在 CKOUT 引脚上提供一个时钟信号，来驱动外部 ΣΔ 调制器时钟输入。此 CKOUT 信号的频率通过预分频器（请参见 DFSDM_CH0CFGR1 寄存器 CKOUTDIV 位）取自 DFSDM 时钟或音频时钟（请参见 DFSDM_CH0CFGR1 寄存器 CKOUTSRC 位）。如果输出时钟停止，则 CKOUT 信号会被置于低电平状态（通过在 DFSDM_CHyCFGR1 寄存器中设置 CKOUTDIV = 0 或 在 DFSDM_CH0CFGR1 寄存器中设置 DFSDMEN = 0，可以停止输出时钟）。在以下时间执行输出时钟停止：

- DFSDMEN 清零后 4 个系统时钟（当 CKOUTSRC =0 时）
- DFSDMEN 清零后 1 个系统时钟和 3 个音频时钟（CKOUTSRC = 1 时）

更改 CKOUTSRC 之前，软件必须等待 CKOUT 停止，以避免在 CKOUT 引脚上产生毛刺信号。输出时钟信号频率必须介于 0 - 20 MHz 范围内。

SPI 数据输入格式操作

在 SPI 格式下，数据流通过数据和时钟信号以串行格式进行发送。数据信号始终由 DATINy 引脚提供。时钟信号可通过 CKINy 引脚从外部提供，也可以通过取自 CKOUT 信号源的信号从内部提供。

如果选择外部时钟源 (SPICKSEL[1:0] = 0)，则根据（DFSDM_CHyCFGR1 寄存器中的）SITP[1:0] 位设置在（CKINy 引脚的）时钟上升沿或下降沿采样（DATINy 引脚上的）数据信号。

内部时钟源 —— 请参见 DFSDM_CHyCFGR1 寄存器中的 SPICKSEL[1:0]：

- CKOUT 信号：
  - 用于连接外部 ΣΔ 调制器，调制器直接使用其时钟输入（来自 CKOUT）来生成输出串行通信时钟。
  - 采样点：上升沿/下降沿，具体取决于 SITP[1:0] 设置。
- CKOUT/2 信号（在 CKOUT 上升沿生成）：
  - 用于连接外部 ΣΔ 调制器，调制器将时钟输入（来自 CKOUT）除以 2，来生成输出串行通信时钟（该输出时钟变化在各个时钟输入上升沿有效）。
  - 采样点：每第二个 CKOUT 下降沿。
- CKOUT/2 信号（在 CKOUT 下降沿生成）：
  - 用于连接外部 ΣΔ 调制器，调制器将时钟输入（来自 CKOUT）除以 2，来生成输出串行通信时钟（该输出时钟变化在各个时钟输入下降沿有效）。
  - 采样点：每第二个 CKOUT 上升沿。

注： *只有在外部 ΣΔ 调制器将 CKOUT 信号用作时钟输入（以实现同步时钟和数据操作）时，才能使用内部时钟源。*

使用内部时钟源可省去 CKINy 引脚连接（CKINy 引脚可用于其他用途）。

如果采用 SPI 编码，时钟源信号频率必须介于 0 - 20 MHz 范围内，且小于 $f_{DFSDMCLK}/4$。

曼彻斯特编码数据输入格式操作

如果采用曼彻斯特编码格式，数据流仅通过 DATINy 引脚以串行格式进行发送。经过曼彻斯特解码之后，数据和时钟信号从串行流中恢复。可以有两种曼彻斯特编码设置（请参见 DFSDM_CHyCFGR1 寄存器 SITP[1:0] 位）：

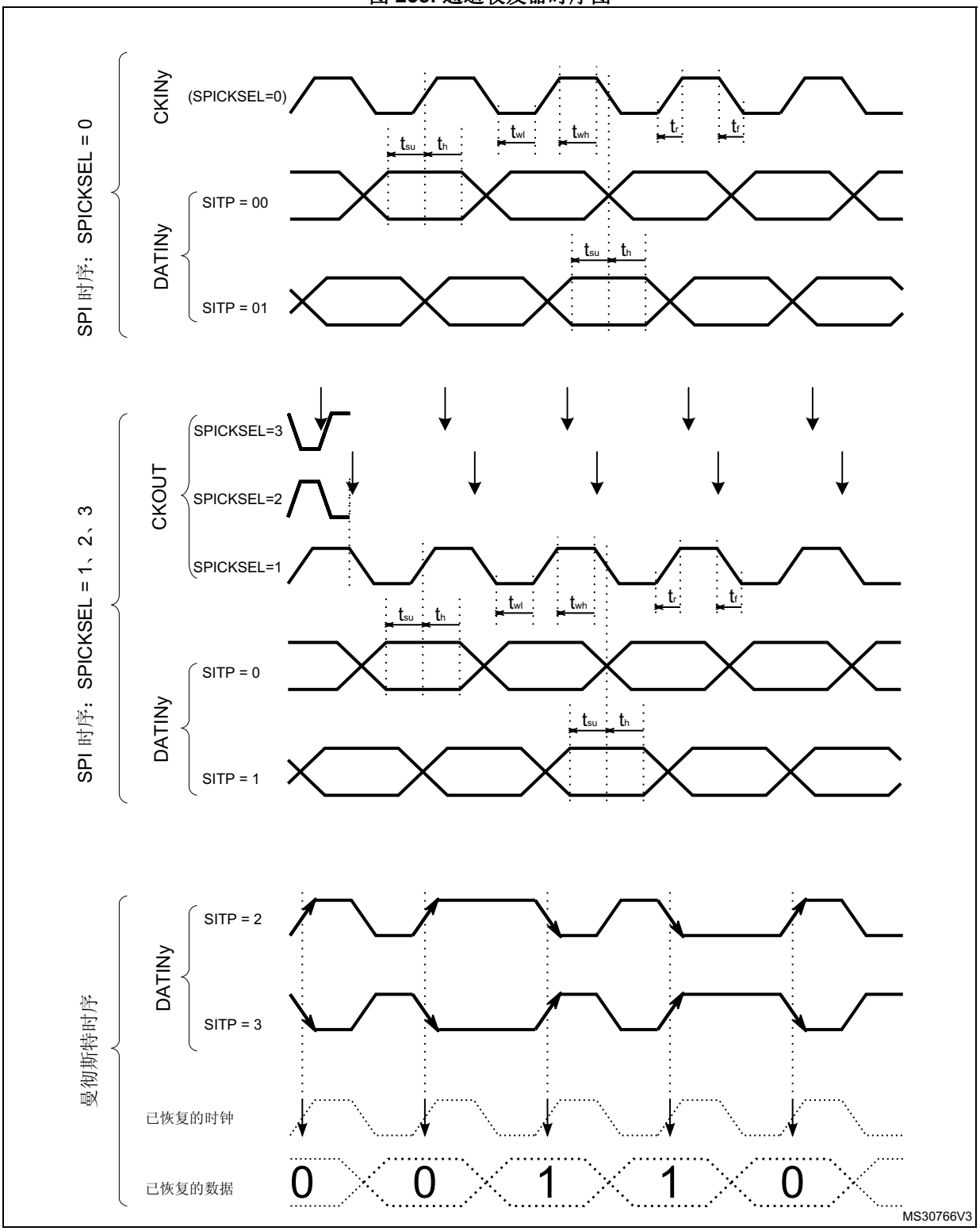
- 信号上升沿 = 逻辑 0；信号下降沿 = 逻辑 1
- 信号上升沿 = 逻辑 1；信号下降沿 = 逻辑 0

如果采用曼彻斯特编码，则恢复后的时钟信号频率必须介于 0 - 10 MHz 范围内，且小于 $f_{DFSDMCLK}/6$。

为了正确接收曼彻斯特编码数据，必须根据预期的曼彻斯特数据速率对（DFSDM_CH0CFGR1 寄存器中的）CKOUTDIV 分频器进行设置，设置公式为：

$$((CKOUTDIV + 1) \times T_{SYSCLK}) < T_{曼彻斯特时钟} < (2 \times CKOUTDIV \times T_{SYSCLK})$$

图 258. 通道收发器时序图



时钟缺失检测

可检查通道串行时钟输入的时钟缺失/存在情况，以确保转换和错误报告正常运行。可通过 DFSDM_CHyCFGR1 寄存器 CKABEN 位在各个输入通道 y 上使能或禁止时钟缺失检测。如果使能了给定通道的时钟缺失检测，则会对其连续执行时钟缺失检测。如果出现输入时钟错误，则时钟缺失标志会置 1 (CKABF[y] = 1)，并会调用中断（CKABIE =1 时）（请参见 DFSDM_FLT0ISR 寄存器的 CKABF[3:0] 和 DFSDM_CHyCFGR1 的 CKABEN）。时钟缺失标志清零（通过 DFSDM_FLT0ICR 寄存器的 CLRCKABF）后，会刷新时钟缺失标志。相应通道 y 禁止时，则时钟缺失状态位 CKABF[y] 还会由硬件置 1（如果 CHEN[y] = 0，则 CKABF[y] 保持在置 1 状态）。

如果发生了时钟缺失事件，则数据转换（和/或模拟看门狗和短路检测器）会提供错误的数据。当报告了时钟缺失时，用户应管理此事件并丢弃给定数据。

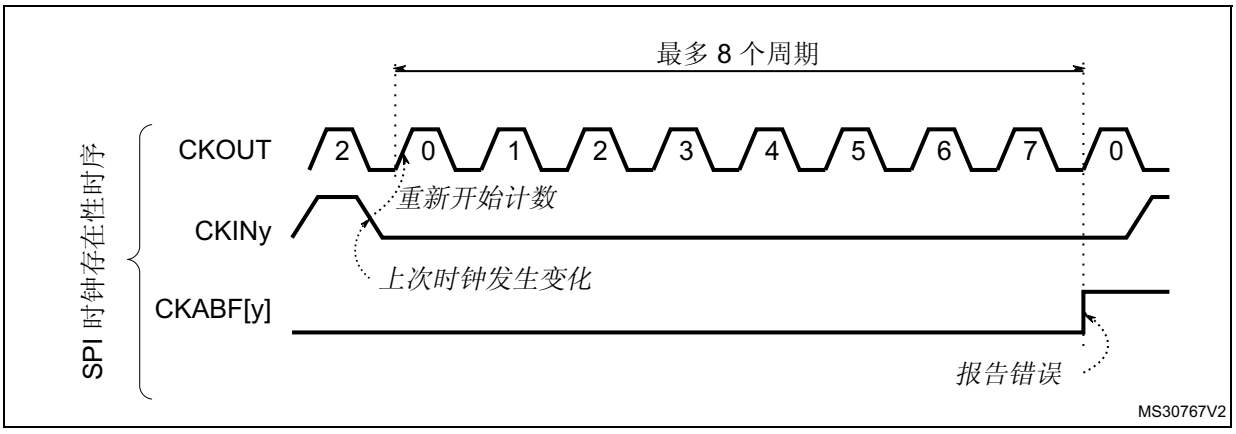
只有在将系统时钟用于 CKOUT 信号（DFSDM_CH0CFGR1 寄存器中的 CKOUTSRC = 0）时，时钟缺失功能才可用。

当收发器尚未被同步时，时钟缺失标志会置 1，且无法由（DFSDM_FLT0ICR 寄存器中）CLRCKABF[y] 位清零。与时钟缺失检测功能相关的软件序列应：

- 通过 CHEN = 1 使能给定通道
- 尝试将时钟缺失标志清零（通过 CLRCKABF = 1），直至时钟缺失标志已实际清零 (CKABF = 0)。此时，收发器被同步（信号时钟有效），能够接收数据。
- 使能时钟缺失功能 CKABEN = 1 且相关中断 CKABIE = 1，以检测 SPI 时钟是否丢失或曼彻斯特数据边沿是否缺失。

如果使用 SPI 数据格式，将基于外部输入时钟与输出时钟生成（CKOUT 信号）的比较进行时钟缺失检测。输入通道中的外部输入时钟信号必须每经过 CKOUT 信号（由 DFSDM_CH0CFGR1 寄存器 CKOUTDIV 字段控制）的 8 个信号周期更改一次。

图 259. SPI 时钟缺失时序图



如果使用曼彻斯特数据格式，则时钟缺失表示无法从曼彻斯特编码信号中恢复时钟。为了能够正常恢复时钟，首先必须接收带 1 - 0 或 0 - 1 转换的数据（有关曼彻斯特同步，请参见*图 261*）。

如果使用曼彻斯特编码格式，则时钟缺失检测（在首次成功同步之后）是基于编码串行数据输入信号与输出时钟生成（CKOUT 信号）之间的变化比较。在 CKOUT 信号（由 DFSDM_CH0CFGR1 寄存器 CKOUTDIV 位控制）的两个周期期间，DATINy 引脚电平必须发生变化。该条件还定义了能正确恢复曼彻斯特编码的数据和时钟信号的最小数据速率。

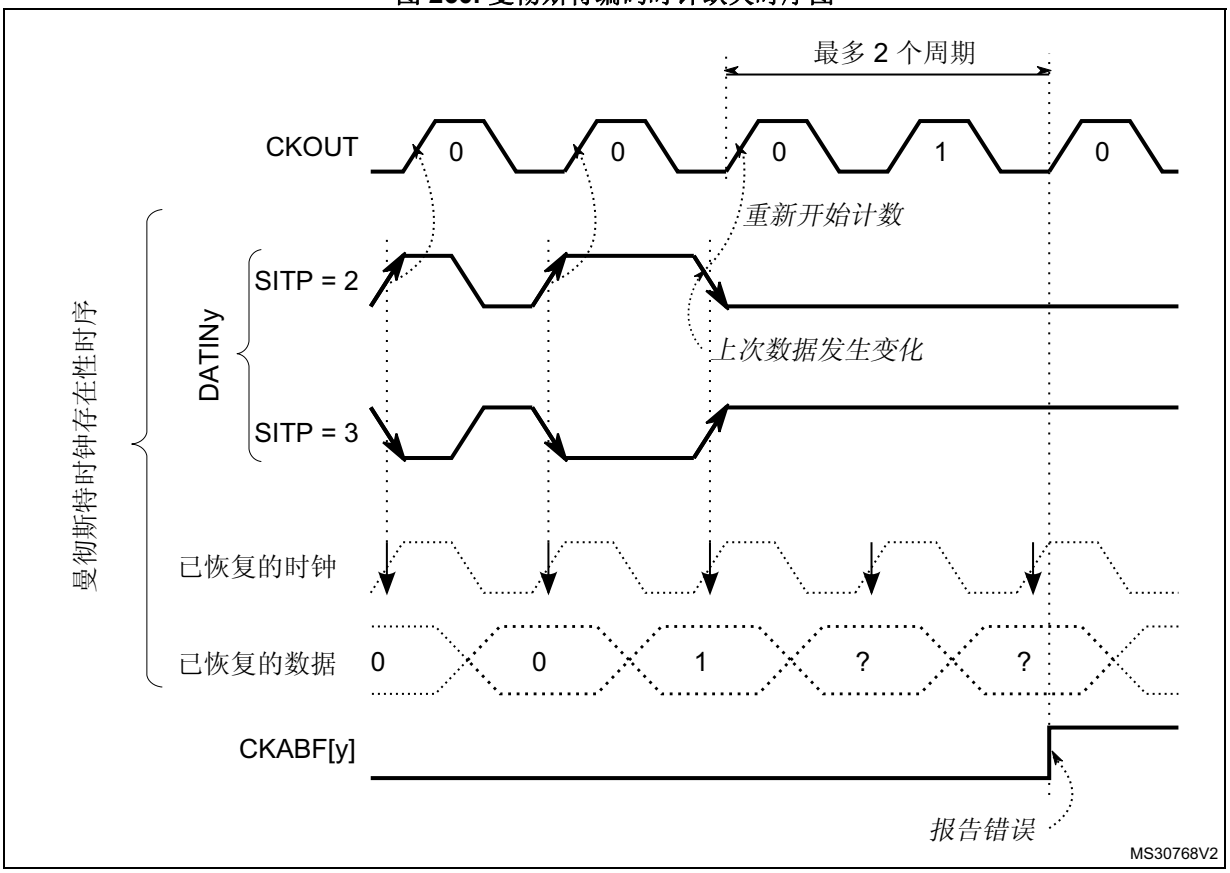
曼彻斯特编码数据的最大数据速率必须小于 CKOUT 信号。

因此，为正确接收曼彻斯特编码数据，必须根据以下公式设置 CKOUTDIV 分频器：

$$((CKOUTDIV + 1) \times T_{SYSCLK}) < T_{曼彻斯特时钟} < (2 \times CKOUTDIV \times T_{SYSCLK})$$

如果出现输入时钟恢复错误，则时钟缺失标志会置 1 (CKABF[y] = 1)，并会调用中断（CKABIE=1 时）（请参见 DFSDM_FLT0ISR 寄存器的 CKABF[3:0] 和 DFSDM_CHyCFGR1 的 CKABEN）。时钟缺失标志清零（通过 DFSDM_FLT0ICR 寄存器的 CLRCKABF）后，会刷新时钟缺失标志。

图 260. 曼彻斯特编码时钟缺失时序图



曼彻斯特/SPI 编码同步

必须在使能通道（DFSDM_CHyCFGR1 寄存器 CHEN = 1）后，首次同步曼彻斯特编码数据流。当接收到 0 - 1 或 1 - 0 数据转换（以便能检测有效数据边沿）时，同步结束。在通过 DFSDM_FLT0ICR 的 CLRCKABF [y] 将给定通道的 CKABF[y] 清零之后，可以按照以下详细描述的软件序列通过轮询 CKABF [y] = 0 来检查同步是否结束：

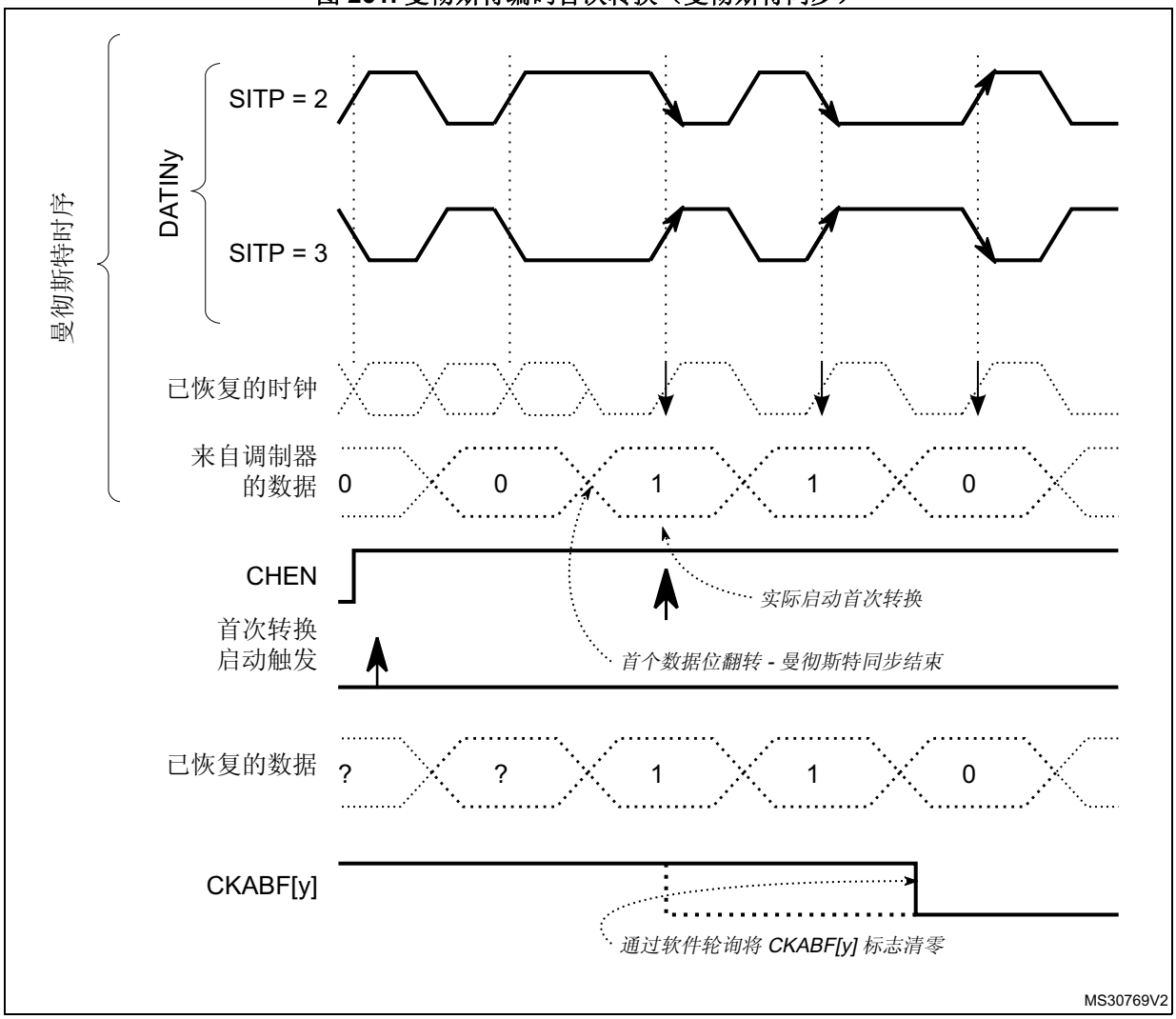
通过将 CLRCKABF[y] 位置 1 将 CKABF[y] 标志清零。如果通道 y 尚未同步，则硬件会立即将 CKABF[y] 标志置 1。软件随后将回读 CKABF[y] 标志，如果该标志置 1，则会再次通过将 CLRCKABF[y] 位置 1 来将其清零。此软件序列（CKABF[y] 标志的轮询）将继续进行，直至 CKABF[y] 标志置 1（表示曼彻斯特码流已同步）。为了能够同步/接收曼彻斯特编码数据，必须根据预期的曼彻斯特数据速率对（DFSDM_CH0CFGR1 寄存器中的）CKOUTDIV 分频器进行设置，以下为设置公式：

$$((CKOUTDIV + 1) \times T_{SYSCLK}) < T_{曼彻斯特时钟} < (2 \times CKOUTDIV \times T_{SYSCLK})$$

SPI 码流在第一次检测到时钟输入信号（有效的上升沿/下降沿）后同步。

注： *当收发器尚未被同步时，时钟缺失标志会置 1，且无法由（DFSDM_FLT0ICR 寄存器中）CLRCKABF[y] 位清零。*

图 261. 曼彻斯特编码首次转换（曼彻斯特同步）



外部串行时钟频率测量

通过测量通道串行时钟输入频率可提供来自外部 ΣΔ 调制器的实时数据速率，这对应用目的很重要。

外部串行时钟输入频率可以通过在一个转换持续时间内对 DFSDM 时钟（$f_{DFSDMCLK}$）进行计数的定时器来测量。计数在转换触发（常规或注入）后的第一个输入数据时钟处开始，并在转换结束（转换结束标志置 1）前的最后一个输入数据时钟处结束。转换完成（JEOCF =1 或 REOCF = 1）时，会在寄存器 DFSDM_FLTxCNVTIMR 的计数器 CNVCNT[27:0] 中更新每次转换的时间（第一个串行采样和最后一个串行采样之间的时间）。然后用户可以根据数字滤波器设置（FORD、FOSR、IOSR、FAST）计算数据速率。只有在滤波器被旁路（FOSR = 0，只有积分器有效，DFSDM_FLTxCNVTIMR 寄存器 CNVCNT[27:0] = 0）时，才会停止外部串行频率测量。

对于并行数据输入（*第 29.4.6 节：并行数据输入*），测量的频率为一次转换期间内的平均输入数据速率。

注： *转换被中断（例如，通过禁止/使能所选通道）时，还会在 CNVCNT[27:0] 中对中断时间进行计数。因此，为得到正确的转换时间结果，建议不要中断转换。*

转换时间：

FAST = 0 时的注入转换或常规转换（或者 FAST=1 时的首次转换）：

对于 $Sinc^x$ 滤波器 (x =1..5)：

$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * (I_{OSR}-1 + F_{ORD}) + F_{ORD}] / f_{CKIN}$

对于 FastSinc 滤波器：

$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * (I_{OSR}-1 + 4) + 2] / f_{CKIN}$

FAST = 1 时的常规转换（首次转换除外）：

对于 $Sinc^x$ 和 FastSinc 滤波器：

$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * I_{OSR}] / f_{CKIN}$

如果 F_{OSR} = FOSR[9:0] + 1 = 1（滤波器旁路，仅积分器有效），则：

$t = I_{OSR} / f_{CKIN}$（... 但是 CNVCNT = 0）

其中：

- f_{CKIN} 为（给定通道 CKINy 引脚上的）通道输入时钟频率或（并行数据输入时的）输入数据速率
- F_{OSR} 为滤波器过采样率：F_{OSR} = FOSR[9:0]+1（请参见 DFSDM_FLTxFCR 寄存器）
- I_{OSR} 为积分器过采样率：I_{OSR} = IOSR[7:0]+1（请参见 DFSDM_FLTxFCR 寄存器）
- F_{ORD} 为滤波器阶数：F_{ORD} = FORD[2:0]（请参见 DFSDM_FLTxFCR 寄存器）

通道偏移设置

每个通道都有自身的偏移设置（在寄存器中），该偏移值最终会从给定通道的各个转换结果（注入或常规）中减去。在数据右移位后执行偏移校正。偏移值以 24 位有符号值的形式存储在 DFSDM_CHyCFGR2 寄存器的 OFFSET[23:0] 字段中。

数据右移位

要使结果与 24 位值对齐，每个通道均定义了一系列右移位，这些右移位将应用于给定通道的各个转换结果（注入或常规）。右移位数存储在 DFSDM_CHyCFGR2 寄存器的 DTRBS[4:0] 位中。

右移位将结果四舍五入为最接近的整数值。移位结果的符号保留，以便得到有效 24 位有符号格式的结果数据。

脉冲跳跃

脉冲跳跃的目的在于实现指定输入通道的延迟线等特性。输入串行数据流（仅串行流）中指定数量的样品可在进入滤波器前被丢弃。这种数据丢弃操作是通过跳过一定数量的采样输入时钟脉冲来实现的（这样一来，给定的串行数据样本就不会被滤波器采样）。采样时钟由脉冲跳跃功能门控，以获得指定数量的时钟脉冲。指定的时钟脉冲被跳过时，输入数据的滤波会继续。与未跳过的数据流相比，该操作导致来自滤波器的最终输出样品（及下一个样品）从后面的输入数据计算得出。由于是从较新的输入样品而不是“未跳过”样品计算得出，因此，最终的样品看起来是向前移动的。最终的“跳过样品”将在稍后转换，因为跳过的输入数据样品必须由随后的输入数据样品替换。由于两个数据缓冲区都相移了，因此，最终的数据缓冲区特性（跳过和未跳过的输出数据缓冲区比较）就好像未跳过的数据流稍微延迟了。

要跳过的时钟脉冲数应写入 DFSDM_CHyDLYR 寄存器的 PLSSKP[5:0] 字段。写入 PLSSKP[5:0] 字段后，将开始在指定通道上执行脉冲跳过。可对 PLSSKP[5:0] 字段进行读取，来检查脉冲跳跃的进度。PLSSKP[5:0]=0 表示已执行脉冲跳跃。

对 PLSSKP[5:0] 执行单次写操作时，最多可跳过 63 个时钟脉冲。如果需要跳过更多脉冲，则用户必须多次写入 PLSSKP[5:0] 字段。应用程序软件应处理每个滤波器的累积跳过时钟数。

29.4.5 配置输入串行接口

必须为输入串行接口配置以下参数：

- **输出时钟预分频器**。提供了可编程预分频器，用来从 DFSDM 时钟生成输出时钟 (2 - 256)。它由 DFSDM_CH0CFGR1 寄存器中的 CKOUTDIV[7:0] 位定义。
- **串行接口类型和输入时钟相位。**选择 SPI 或曼彻斯特编码以及输入时钟的采样边沿。它由 DFSDM_CHyCFGR1 寄存器中的 SITP [1:0] 位定义。
- **输入时钟源。**来自 CKINy 引脚的外部源或来自 CKOUT 引脚的内部源。它由 DFSDM_CHyCFGR1 寄存器中的 SPICKSEL[1:0] 字段定义。
- **最终数据右移位。**定义最终数据右移位，以将结果与 24 位值对齐。它由 DFSDM_CHyCFGR2 寄存器中的 DTRBS[4:0] 定义。
- **每个通道的通道偏移。**定义给定串行通道的模拟偏移（所连接外部 ΣΔ 调制器的偏移）。它由 DFSDM_CHyCFGR2 寄存器中的 OFFSET[23:0] 位定义。
- **每个通道的短路检测器和时钟缺失使能。**用于在 DFSDM_CHyCFGR1 寄存器中使能或禁止给定串行通道的短路检测器（通过 SCDEN 位）和时钟缺失检测（通过 CKABEN 位）。
- **模拟看门狗滤波器和短路检测器阈值设置。**用于配置通道模拟看门狗滤波器参数和通道短路检测器参数。在 DFSDM_CHyAWSCDR 寄存器中定义这些配置。

29.4.6 并行数据输入

每个输入通道均为 16 位并行数据输入提供一个寄存器（除了串行数据输入外）。每个 16 位并行输入只能源自内部数据源：

- 内部 ADC 结果
- 直接 CPU/DMA 写操作。

通过 DFSDM_CHyCFGR1 寄存器的 DATMPX[1:0] 字段选择是对给定通道使用串行还是并行数据输入。DATMPX[1:0] 中还定义了并行数据源：内部 ADC 或直接由 CPU/DMA 进行的写操作。

每个通道均包含一个 32 位数据输入寄存器 DFSDM_CHyDATINR，在该寄存器中可写入 16 位数据。数据采用 16 位有符号格式。这些数据还可用作接受 16 位并行数据的数字滤波器的输入。

如果选择串行数据输入 (DATMPX[1:0] = 0)，则 DFSDM_CHyDATINR 寄存器被写保护。

来自内部 ADC 的输入

对于并行 ADC 数据输入 (DATMPX[1:0]=1)，ADC[y+1] 结果被分配至通道 y 输入（ADC1 填充 DFSDM_CHDATIN0R 寄存器，ADC2 填充 DFSDM_CHDATIN1R 寄存器，...，ADC8 填充 DFSDM_CHDATIN7R 寄存器）。来自 ADC[y+1] 的转换结束事件会导致更新通道 y 的数据（来自 ADC[y+1] 的并行数据被用作数字滤波器的下一个采样）。转换结束事件发生时，来自 ADC[y+1] 的数据被写入 DFSDM_CHyDATINR 寄存器（INDAT0[15:0] 字段）。

数据封装模式设置（DFSDM_CHyCFGR1 寄存器中的 DATPACK[1:0]）对 ADC 数据输入无影响。

注： *ADC 扩展规范：将内部 ADC 配置为交错模式时（例如，ADC1 与 ADC2 交错 - 请参见 ADC 规范），来自 ADC1 或 ADC2 的每个结果均会传送至同一 16 位总线（即 ADC1 的总线），该总线连至 DFSDM 通道 0（固定连接）。因此，DFSDM 通道 0 中将存在双倍输入数据速率（偶数采样来自 ADC1，奇数采样来自 ADC2）。与 ADC2 相关的通道 1 将处于空闲状态。*

来自内存的输入（直接由 CPU/DMA 写入）

对于由 CPU 或 DMA (DATMPX[1:0] = 2) 直接写入到 DFSDM_CHyDATINR 寄存器中的数据，可以将其用作数据输入，以处理来自内存或外设的数字数据流。

数据可由 CPU 或 DMA 写入到 DFSDM_CHyDATINR 寄存器中：

1. **CPU 数据写入：**

   输入数据直接由 CPU 写入到 DFSDM_CHyDATINR 寄存器中。

2. **DMA 数据写入：**

   DMA 应配置为内存到内存传输模式，以将内存缓冲区中的数据传输至 DFSDM_CHyDATINR 寄存器中。目标内存地址为 DFSDM_CHyDATINR 寄存器的地址。数据以 DMA 传输速度从内存传输至 DFSDM 并行输入。

   此 DMA 不同于读取 DFSDM 转换结果所使用的 DMA。这两个 DMA 可同时使用，第一个 DMA（配置为内存到内存传输）用于输入数据写入，第二个 DMA（配置为外设到内存传输）用于数据结果读取。

对 DFSDM_CHyDATINR 的访问可为 16 位或 32 位宽，允许分别在一次写操作中加载一个或两个采样。32 位输入数据寄存器 (DFSDM_CHyDATINR) 可用一个或两个 16 位数据采样进行填充，具体取决于在 DFSDM_CHyCFGR1 寄存器的 DATPACK[1:0] 字段中定义的数据封装操作模式：

1. **标准模式 (DATPACK[1:0] = 0)：**

   只有一个样本存储在 DFSDM_CHyDATINR 寄存器的 INDAT0[15:0] 字段中，其用作通道 y 的输入数据。高 16 位 (INDAT1[15:0]) 被忽略，且被写保护。数字滤波器必须执行一次输入采样（来自 INDAT0[15:0]）以在被 CPU/DMA 填充后清空数据寄存器。该模式与对 DFSDM_CHyDATINR 寄存器的 16 位 CPU/DMA 访问结合使用，以在每次写操作时加载一个采样。

2. **交错模式 (DATPACK[1:0] = 1)：**

   DFSDM_CHyDATINR 寄存器被用作两个采样缓冲区。第一个采样存储在 INDAT0[15:0] 中，第二个采样存储在 INDAT1[15:0] 中。数字滤波器必须通过通道 y 执行两次输入采样，以清空 DFSDM_CHyDATINR 寄存器。该模式与对 DFSDM_CHyDATINR 寄存器的 32 位 CPU/DMA 访问结合使用，以在每次写操作时加载两个采样。

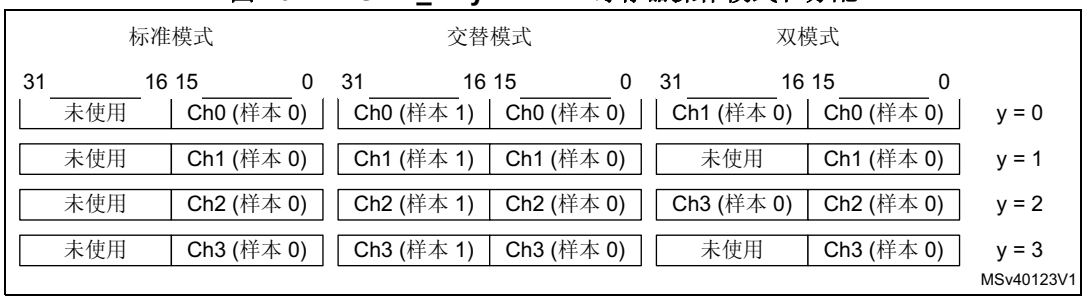
3. **双通道模式 (DATPACK[1:0] = 2)：**

   将两个采样写入到 DFSDM_CHyDATINR 寄存器中。INDAT0[15:0] 中的数据用于通道 y，INDAT1[15:0] 中的数据用于通道 y+1。INDAT1[15:0] 中的数据会被自动复制到下一 (y+1) 通道数据寄存器 (DFSDM_CH[y+1]DATINR) 的 INDAT0[15:0] 中。数字滤波器必须执行两次采样（一次来自通道 y，另一次通道 (y+1)），以清空 DFSDM_CHyDATINR 寄存器。

   只能针对偶数通道数（y = 0、2）进行双通道模式设置 (DATPACK[1:0]=2)。如果将奇数通道（y = 1、3）设为双通道模式，则对于该通道，INDAT0[15:0] 和 INDAT1[15:0] 部分均被写保护。如果偶数通道被设为双通道模式，则下一奇数通道必须设为标准模式 (DATPACK[1:0]=0)，以便与偶数通道正确配合。

有关 DFSDM_CHyDATINR 寄存器数据模式和通道数据样本分配，请参见*图 262*。

图 262. DFSDM_CHyDATINR 寄存器操作模式和分配



首先使能所选输入通道（通道 y），以便进行数据采集（启动通道 y 转换），然后必须向 DFSDM_CHyDATINR 寄存器进行写操作，以加载一个或两个采样。否则，在下次处理时会丢失写入的数据。

例如：对于单次转换和交错模式，在启动单次转换前，不要开始向 DFSDM_CHyDATINR 写入成对数据样本（开始转换前 DFSDM_CHyDATINR 中存在的数据均会被丢弃）。

29.4.7 通道选择

存在 4 个复用通道，可选择用于注入通道组转换和/或常规通道转换。

注入通道组可以选择 4 个通道中的任一通道或其全部。DFSDM_FLTxJCHGR 寄存器中的 JCHG[3:0] 选择注入组通道，其中 JCHG[y] =1 表示已选择通道 y。

注入转换可工作在扫描模式 (JSCAN = 1) 或单个模式 (JSCAN = 0) 下。在扫描模式下，每个所选通道都会依次转换。最低通道（通道 0，如果已选择）最先转换，紧接着转换下一个较高通道，直至 JCHG[3:0] 选择的所有通道均被转换。在单个模式 (JSCAN=0) 下，只会转换所选的一个通道，通道选择会移至下一通道。JSCAN = 0 时，对 JCHG[3:0] 进行写入操作会将通道选择复位为选择的最低通道。

注入转换可由软件或触发器启动。此类转换从不会被常规转换中断。

常规通道只选择 4 个通道的其中一个。DFSDM_FLTxCR1 寄存器中的 RCH[1:0] 指示所选通道。

常规转换只能由软件启动（不能由触发器启动）。请求注入转换时，会暂时中断连续常规转换序列。

对已禁止通道（DFSDM_CHyCFGR1 寄存器中的 CHEN = 0）执行转换会导致转换将永远不会结束，这是因为未提供输入数据的原因（无时钟信号）。在这种情况下，需要使能给定通道（DFSDM_CHyCFGR1 寄存器中的 CHEN = 1），或停止转换（通过 DFSDM_FLTxCR1 寄存器的 DFEN = 0）。

29.4.8 数字滤波器配置

DFSDM 包含 Sinc^x 类型数字滤波器实现。此 Sinc^x 滤波器执行输入数字数据流滤波，这会导致减小输出数据速率（抽取）以及增大输出数据分辨率。可对 Sinc^x 数字滤波器进行配置，以达到所需输出数据速率和所需输出数据分辨率。可配置参数有：

- 滤波器阶数/类型：（请参见 DFSDM_FLTxFCR 寄存器中的 FORD[2:0] 位）：
  - FastSinc
  - Sinc^1
  - Sinc^2
  - Sinc^3
  - Sinc^4
  - Sinc^5
- 滤波器过采样/抽取率：（请参见 DFSDM_FLTxFCR 寄存器中的 FOSR[9:0] 位）：
  - FOSR = 1-1024 —— 适用于 FastSinc 滤波器 和 Sinc^x 滤波器 x = F_{ORD} = 1..3
  - FOSR = 1-215 —— 适用于 Sinc^x 滤波器 x = F_{ORD} = 4
  - FOSR = 1-73 —— 适用于 Sinc^x 滤波器 x = F_{ORD} = 5

滤波器支持以下传递函数（H 域中的冲激响应）：

- Sinc^x 滤波器类型： $$高阻态 = \left(\frac{1-z^{-FOSR}}{1-z^{-1}}\right)^x$$
- FastSinc 滤波器类型： $$高阻态 = \left(\frac{1-z^{-FOSR}}{1-z^{-1}}\right)^2 Þ (1+z^{-(2\cdot FOSR)})$$

图 263. 示例：Sinc^3 滤波器类型

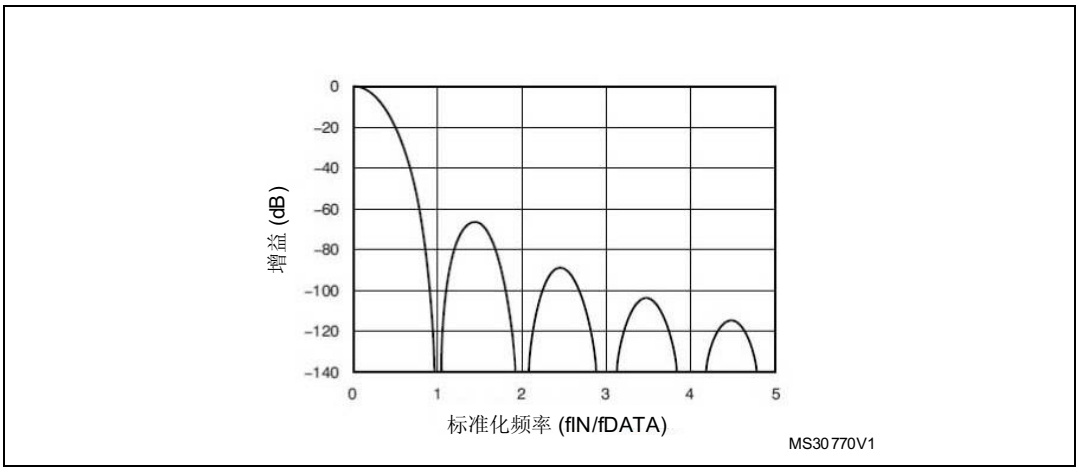


表 233. 滤波器最大输出分辨率（来自滤波器输出的峰值数据值），基于某些 FOSR 值

FOSR	$Sinc^1$	$Sinc^2$	FastSinc	$Sinc^3$	$Sinc^4$	$Sinc^5$
x	+/- x	+/- x^2	+/- $2x^2$	+/- x^3	+/- x^4	+/- x^5
4	+/- 4	+/- 16	+/- 32	+/- 64	+/- 256	+/- 1024
8	+/- 8	+/- 64	+/- 128	+/- 512	+/- 4096	-
32	+/- 32	+/- 1024	+/- 2048	+/- 32768	+/- 1048576	+/- 33554432
64	+/- 64	+/- 4096	+/- 8192	+/- 262144	+/- 16777216	+/- 1073741824
128	+/- 128	+/- 16384	+/- 32768	+/- 2097152	+/- 268435456	在满量程输入的条件下，结果会溢出（> 32 位有符号整数值）
256	+/- 256	+/- 65536	+/- 131072	+/- 16777216		
1024	+/- 1024	+/- 1048576	+/- 2097152	+/- 1073741824		

有关 Sinc 滤波器类型属性和使用的更多信息，建议了解下数字滤波器的理论知识（可从 Internet 上下载更多资源）。

29.4.9 积分器单元

积分器会对来自数字滤波器的数据进一步提升抽取率和分辨率。对于来自滤波器的给定数量的数据采样，积分器对其进行简单的求和。

积分器过采样率参数定义了将多少个数据的和作为积分器的一个数据输出。IOSR 可设置为介于 1-256 范围内（请参见 DFSDM_FLTxFCR 寄存器中的 IOSR[7:0] 位说明）。

表 234. 积分器最大输出分辨率（来自积分器输出的峰值数据值），基于某些 IOSR 值，FOSR = 256 以及 $Sinc^3$ 滤波器类型（最大数据）

IOSR	$Sinc^1$	$Sinc^2$	FastSinc	$Sinc^3$	$Sinc^4$	$Sinc^5$
x	+/- FOSR. x	+/- $FOSR^2$. x	+/- 2.$FOSR^2$. x	+/- $FOSR^3$. x	+/- $FOSR^4$. x	+/- $FOSR^5$. x
4	-	-	-	+/- 67 108 864	-	-
32	-	-	-	+/- 536 870 912	-	-
128	-	-	-	+/- 2 147 483 648	-	-
256	-	-	-	+/- 2^{32}	-	-

29.4.10 模拟看门狗

模拟看门狗用于在达到或超出给定阈值上限或者达到或低于给定阈值下限时，触发外部信号（断路或中断）。随即可调用中断/事件/断路生成。

每个模拟看门狗都会根据（DFSDM_FLTxCR1 寄存器中）AWFSEL 位设置监视串行数据接收器输出（在每个通道上模拟看门狗滤波器之后）或数据输出寄存器（当前注入或常规转换结果）。是否需要通过模拟看门狗 x 监视的输入通道可以通过 DFSDM_FLTxCR2 寄存器中的 AWDCH[3:0] 选择。

输入通道上的模拟看门狗转换与标准转换无关。在标准转换模式下，模拟看门狗在每个输入通道上使用自己的滤波器和信号处理，与注入或常规主转换无关。在连续转换模式下，对所选输入通道执行模拟看门狗转换，以便在注入或常规主转换暂停（RCIP = 0，JCIP = 0）时也仍能对通道进行监视。

阈值上限和阈值下限寄存器与给定数据值（通过 DFSDM_FLTxAWHTR 寄存器的 AWHT[23:0] 位和 DFSDM_FLTxAWLTR 寄存器的 AWLT[23:0] 位进行设置）进行比较。

可以选择 2 种情况将阈值寄存器与数据值进行比较：

- 选择 1：在这种情况下，输入数据取自最终输出数据寄存器 (AWFSEL = 0)。该选择具有以下特性：
  - 高输入数据分辨率（高达 24 位）
  - 响应时间慢——不适用于过流检测等快速响应应用
  - 为了进行比较，最终数据需要经过移位和偏移数据校正
  - 最终数据仅在执行常规或注入主转换后才可用
  - 可在并行输入数据源（DFSDM_CHyCFGR1 寄存器中的 DATMPX[1:0] ≠ 0）情况下使用
- 选择 2：在这种情况下，输入数据取自任一串行数据接收器输出 (AWFSEL = 1)。该选择具有以下特性：
  - 输入串行数据通过专用模拟看门狗 $Sinc^x$ 通道滤波器进行处理，滤波器的过采样率 (1..32) 和滤波器阶数 (1..3) 可以配置（请参见 DFSDM_CHyAWSCDR 寄存器中的 AWFOSR[4:0] 和 AWFORD[1:0] 位设置）
  - 较低分辨率（最高为 16 位）
  - 响应时间快——适用于需要快速响应的应用，如过流/过压检测
  - 数据以连续模式提供，与常规或注入主转换活动无关

进行输入通道监视 (AWFSEL = 1) 时，用于跟阈值进行比较的数据取自 AWDCH[3:0] 字段（DFSDM_FLTxCR2 寄存器）所选的通道。将每个所选通道滤波器结果与一个阈值对 (AWHT[23:0] / AWLT[23:0]) 进行比较。此时，只有高 16 位 (AWHT[23:8] / AWLT[23:8]) 会定义与模拟看门狗滤波器输出进行比较的 16 位阈值，因为来自模拟看门狗滤波器的数据的分辨率最高为 16 位。在这种情况下 (AWFSEL=1)，不对 AWHT[7:0] / AWLT[7:0] 位进行比较。

在 DFSDM_CHyAWSCDR 寄存器中设置各个输入通道的模拟看门狗滤波器配置的参数（滤波器阶数 AWFORD[1:0] 和滤波器过采样率 AWFOSR[4:0]）。

每个输入通道都配有自身的比较器，用于比较模拟看门狗数据（来自模拟看门狗滤波器）和模拟看门狗阈值 (AWHT/AWLT)。选择多个通道（DFSDM_FLTxCR2 寄存器的字段 AWDCH[3:0]）时，可同时接收多个比较请求。在这种情况下，首先管理编号最小的通道请求，然后继续管理编号较大的所选通道。对于每个通道，结果均可记录在单独的标志（DFSDM_FLTxAWSR 寄存器的字段 AWHTF[3:0] 和字段 AWLTF[3:0]）中。每个通道请求均在 8 个 DFSDM 时钟周期内执行。因此，各个通道的带宽被限制为 8 个 DFSDM 时钟周期（AWDCH[3:0] = 0x0F 时）。因为最大输入通道采样时钟频率为 DFSDM 时钟频率的四分之一，因此在该输入时钟速度下，不能对模拟看门狗配置 AWFOSR = 0（模拟看门狗滤波器被旁路）。因此，用户必须根据输入采样时钟速度和 DFSDM 频率正确配置监视的通道数和模拟看门狗滤波器参数。

给定通道 y 的模拟看门狗过滤器数据可通过固件从 DFSDM_CHyWDATR 寄存器 WDATA[15:0] 字段中读取。如果 DFSDM_CHyCFGR1 寄存器中的 CHEN = 1，则该模拟看门狗滤波器数据被连续转换，其数据速率由模拟看门狗滤波器设置和通道输入时钟频率给定。

模拟看门狗滤波器转换的工作方式类似于常规的快速连续转换（无积分器）。模拟看门狗滤波器输出（通道输入时钟频率为 f_{CKIN}）一个结果所需的串行采样数量：

首次转换：

对于 $Sinc^x$ 滤波器 (x = 1..5)：采样数 = [F_{OSR} * F_{ORD} + F_{ORD} + 1]

对于 FastSinc 滤波器：采样数 = [F_{OSR} * 4 + 2 + 1]

后续转换：

对于 $Sinc^x$ 和 FastSinc 滤波器：采样数 = [FOSR * IOSR]

其中：

F_{OSR} ..... 滤波器过采样率：F_{OSR} = AWFOSR[4:0]+1（请参见 DFSDM_CHyAWSCDR 寄存器）

F_{ORD} ..... 滤波器阶数：F_{ORD} = AWFORD[1:0]（请参见 DFSDM_CHyAWSCDR 寄存器）

对于输出数据寄存器监视 (AWFSEL = 0)，在完成最终数据的右移位和偏移校正（请参见 DFSDM_CHyCFGR2 寄存器的 OFFSET[23:0] 和 DTRBS[4:0] 字段）后进行比较。在由（DFSDM_FLTxCR2 寄存器中）AWDCH[3:0] 字段所选通道的每次注入或常规转换结束后，进行比较。

模拟看门狗事件的状态反映在 DFSDM_FLTxAWSR 寄存器中，该寄存器中锁定了给定事件。AWHTF[y] = 1 标志指示在通道 y 上是否超出了 AWHT[23:0] 值。AWLTF[y] = 1 标志指示在通道 y 上是否超出了 AWLT[23:0] 值。对于 DFSDM_FLTxAWSR 寄存器中锁存的事件，将通过向 DFSDM_FLTxAWCFR 寄存器中的相应清零位 CLRAWHTF[y] 或 CLRAWLTF[y] 写入“1”来清除。

模拟看门狗的全局状态通过 DFSDM_FLTxISR 寄存器中的 AWDF 标志位来反映（其用于快速检测中断源）。AWDF = 1 表示至少发生了一个看门狗事件（至少对于一个通道，AWHTF[y] = 1 或 AWLTF[y] = 1）。所有 AWHTF[3:0] 和 AWLTF[3:0] 均清零时，AWDF 位才清零。

可将模拟看门狗事件分配至断路输出信号。可将四个断路输出分配至超出阈值上限或低于阈值下限的事件 (dfsdm_break[3:0])。通过 DFSDM_FLTxAWHTR 和 DFSDM_FLTxAWLTR 寄存器中的 BKAWH[3:0] 和 BKAWL[3:0] 字段将断路信号分配至给定模拟看门狗事件。

29.4.11 短路检测器

使用短路检测器的目的在于，当模拟信号达到饱和值（超出满量程范围）并保持给定的时间时，短路检测器可以在极快的响应时间内发出信号。该特性可检测短路或断路故障（例如过流或过压）。可调用中断/事件/断路生成。

短路检测器的输入数据取自通道收发器输出。

各个输入通道上都有一个递增计数器，用于对串行数据接收器输出上的连续 0 或 1 进行计数。如果接收的数据流发生变化（数据信号从 1 到 0 或从 0 到 1），则会重新启动计数器。如果该计数器达到短路阈值寄存器值（DFSDM_CHyAWSCDR 寄存器中的 SCDT[7:0] 位），则会调用短路事件。每个输入通道都配有短路检测器。通过将 SCDEN 位（DFSDM_CHyCFGR1 寄存器中）置 1，可选择对任一通道进行连续监视，每个通道都有其自己的短路检测器设置（阈值位 SCDT[7:0]、状态位 SCDF[3:0]、状态清零位 CLRSCDF[3:0]。禁止相应通道 y (CHEN[y] = 0) 时，状态标志 SCDF[y] 还可由硬件清零。

对于各个通道，可将短路检测器事件分配至断路输出信号 dfsdm_break[3:0]。可将四个断路输出分配至短路检测器事件。通过 DFSDM_CHyAWSCDR 寄存器中的 BKSCD[3:0] 字段将断路信号分配至给定通道短路检测器事件。

如果选择了并行输入数据通道（DFSDM_CHyCFGR1 寄存器中的 DATMPX[1:0] ≠ 0），则无法使用短路检测器。

总共有四个断路输出（与模拟看门狗功能共享）。

29.4.12 极值检测器

极值检测器用于采集最终输出数据字的最小值和最大值（峰峰值）。

如果输出数据字高于极值检测器最大值寄存器（DFSDM_FLTxEXMAX 寄存器中的 EXMAX[23:0] 位）中存储的值，则该寄存器将用当前输出数据字值进行更新，且存储其数据的通道位于（DFSDM_FLTxEXMAX 寄存器的）EXMAXCH[1:0] 位中。

如果输出数据字低于极值检测器最小值寄存器（DFSDM_FLTxEXMIN 寄存器中的 EXMIN[23:0] 位）中存储的值，则该寄存器将用当前输出数据字值进行更新，且存储其数据的通道位于（DFSDM_FLTxEXMIN 寄存器的）EXMINCH[1:0] 位中。

最小值和最大值寄存器值可由软件（通过读取给定的 DFSDM_FLTxEXMAX 或 DFSDM_FLTxEXMIN 寄存器）进行刷新。刷新后，极值检测器最小值数据寄存器 DFSDM_FLTxEXMIN 用 0x7FFFFF（最大正值）进行填充，极值检测器最大值寄存器 DFSDM_FLTxEXMAX 用 0x800000（最小负值）进行填充。

在完成右移位和偏移数据校正之后，极值检测器才进行比较。对于每个极值检测器，在（DFSDM_FLTxCR2 寄存器中的）EXCH[3:0] 位中选择计算极值时所涉及的输入通道。

29.4.13 数据单元模块

数据单元模块是整个处理路径的最后一个模块：外部 ΣΔ 调制器 - 串行收发器 - Sinc 滤波器 - 积分器 - 数据单元模块。

输出数据速率取决于串行数据流速率以及滤波器和积分器设置。最大输出数据速率为：

$$\text{数据速率}[\text{采样}/\text{s}] = \frac{f_{CKIN}}{F_{OSR}\cdot(I_{OSR}-1+F_{ORD})+(F_{ORD}+1)} \quad \text{...FAST = 0，Sincx 滤波器}$$

$$\text{数据速率}[\text{采样}/\text{s}] = \frac{f_{CKIN}}{F_{OSR}\cdot(I_{OSR}-1+4)+(2+1)} \quad \text{...FAST = 0，FastSinc 滤波器}$$

或

$$\text{数据速率}[\text{采样}/\text{s}] = \frac{f_{CKIN}}{F_{OSR}\cdot I_{OSR}} \quad \text{...FAST = 1}$$

并行数据输入时的最大输出数据速率：

$$\text{数据速率}[\text{采样}/\text{s}] = \frac{f_{DATAIN_RATE}}{F_{OSR}\cdot(I_{OSR}-1+F_{ORD})+(F_{ORD}+1)} \quad \text{...FAST = 0，Sincx 滤波器}$$

或

$$\text{数据速率}[\text{采样}/\text{s}] = \frac{f_{DATAIN_RATE}}{F_{OSR}\cdot(I_{OSR}-1+4)+(2+1)} \quad \text{...FAST = 0，FastSinc 滤波器}$$

或

$$\text{数据速率}[\text{采样}/\text{s}] = \frac{f_{DATAIN_RATE}}{F_{OSR} \cdot I_{OSR}} \quad \text{...FAST=1 或任何滤波器旁路情况 } (F_{OSR} = 1)$$

其中：f_{DATAIN_RATE}...来自 ADC 或 CPU/DMA 的输入数据速率

在该模块中执行最终数据的右移位，因为最终数据宽度为 24 位，而来自处理路径的数据最高可为 32 位。对于每个所选输入通道，此右移位均可在 0-31 位的范围内进行配置（请参见 DFSDM_CHyCFGR2 寄存器的 DTRBS[4:0] 位）。右移位将结果四舍五入为最接近的整数值。移位结果的符号保留，以便得到有效 24 位有符号格式的结果数据。

接下来对结果进行偏移校正。从给定通道的输出数据中减去偏移校正值（存储在寄存器 DFSDM_CHyCFGR2 中的 OFFSET[23:0]）。OFFSET[23:0] 字段中的数据由软件通过相应的校准程序进行设置。

因数字处理过程中的所有操作均在 32 位有符号寄存器上执行，因此为保证结果不会发生溢出，必须满足以下条件：

$FOSR^{FORD} \cdot IOSR <= 2^{31}$ ... 适用于 $Sinc^x$ 滤波器 (x = 1..5)
$2 \cdot FOSR^2 \cdot IOSR <= 2^{31}$ ... 适用于 FastSinc 滤波器

注： *滤波器和积分器旁路时（IOSR[7:0] = 0，FOSR[9:0] = 0），输入数据速率 (f_{DATAIN_RATE}) 必须被限制为能够读取所有输出数据：*
$f_{DATAIN_RATE} \le f_{APB}$
其中，f_{APB} 为 DFSDM 外设连接的总线频率。

29.4.14 有符号数据格式

每个 DFSDM 输入串行通道均可连接至一个外部 ΣΔ 调制器。一个外部 ΣΔ 调制器可配有 2 个差分输入（正负），这两个输入可用于差分或单端信号测量。

始终假定 ΣΔ 调制器输出采用有符号格式（来自 ΣΔ 调制器的 0/1 数据流表示值 -1 和 +1）。

寄存器的有符号数据格式：在最终输出数据、模拟看门狗、极值检测器和偏移校正的寄存器中的数据是有符号格式的数据。输出数据字的 msb 表示值的符号（二进制补码格式）。

29.4.15 启动转换

注入转换可使用以下方法启动：

- 软件：向 DFSDM_FLTxCR1 寄存器中的 JSWSTART 写入“1”。
- 触发：JEXTSEL[4:0] 在 JEXTEN 激活触发时选择触发信号，同时选择有效边沿（请参见 DFSDM_FLTxCR1 寄存器）。
- 如果 JSYNC = 1，则使用 DFSDM_FLT0 同步启动：对于 DFSDM_FLTx (x > 0)，注入转换会在 DFSDM_FLT0 中自动启动；注入转换由软件启动（DFSDM_FLT0CR2 寄存器中的 JSWSTART = 1）。DFSDM_FLTx (x > 0) 中的每个注入转换都始终根据其本地配置设置（JSCAN 和 JCHG 等）执行。

如果使能扫描转换（位 JSCAN = 1），则每次触发注入转换时，注入组（DFSDM_FLTxJCHGR 寄存器中的 JCHG[3:0] 位）的全部所选通道将从最低通道（通道 0，如果已选择）开始，被依次转换。

如果禁止扫描转换（位 JSCAN = 0），则每次触发注入转换时，只会转换注入组（DFSDM_FLTxJCHGR 寄存器中的 JCHG[3:0] 位）的所选通道之一，通道选择随即会移至下一通道。JSCAN = 0 时，对 JCHG[3:0] 位进行写操作会将通道选择置为选择的最低注入通道。

在给定时间内只能进行一次注入转换。因此，如果已发出但尚未完成某个注入转换请求，则会忽略启动注入转换的任何请求。

常规转换可使用以下方法启动：

- 软件：向 DFSDM_FLTxCR1 寄存器中的 RSWSTART 写入“1”。
- 如果 RSYNC = 1，则使用 DFSDM_FLT0 同步启动：对于 DFSDM_FLTx (x > 0)，常规转换会在 DFSDM_FLT0 中自动启动；常规转换由软件启动（DFSDM_FLT0CR2 寄存器中的 RSWSTART = 1）。DFSDM_FLTx (x > 0) 中的每个常规转换都始终根据其本地配置设置（RCONT 和 RCH 等）执行。

在给定时间内只能有一个常规转换处于待处理或正在执行状态。因此，如果已发出但尚未完成某个常规转换请求，则会忽略启动常规转换的任何请求。如果常规转换被注入转换中断，或者常规转换在进行注入转换的过程中启动，则常规转换处于待处理状态。此待处理状态的常规转换随即被延迟，并在完成所有注入转换时执行。延迟的常规转换通过 DFSDM_FLTxRDATAR 寄存器的 RPEND 位进行表示。

29.4.16 连续和快速连续模式

如果将 DFSDM_FLTxCR1 寄存器中的 RCONT 置 1，则会在连续模式下执行常规转换。RCONT = 1 表示向 RSWSTART 写入“1”后，重复转换 RCH[1:0] 所选的通道。

可通过向 RCONT 写入“0”停止在连续模式下执行的常规转换。将 RCONT 清零后，会立即停止正在进行的转换。

在连续模式下，可通过将 DFSDM_FLTxCR1 寄存器的 FAST 位置 1 来提高数据速率。此时，如果对一个通道进行连续转换，则滤波器无需用新的刷新数据重新填充，因为滤波器内部数据有效，这些数据是先前通过对连续数据进行采样而得。速度提升与所选的滤波器阶数有关。通过 RSWSTART = 1 启动连续转换后，将充分进行快速模式 (FAST = 1) 下的首次转换（与 FAST = 0 时相同），之后会在较短的间隔内完成各个后续转换。

连续模式下的转换时间：

FAST = 0 时（或 FAST = 1 时的快速转换）：

对于 $Sinc^x$ 滤波器：

$$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * (I_{OSR}-1 + F_{ORD}) + F_{ORD}] / f_{CKIN}$$

对于 FastSinc 滤波器：

$$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * (I_{OSR}-1 + 4) + 2] / f_{CKIN}$$

如果 FAST = 1（首次转换除外）：

对于 $Sinc^x$ 和 FastSinc 滤波器：

$$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * I_{OSR}] / f_{CKIN}$$

如果 F_{OSR} = FOSR[9:0] + 1 = 1（滤波器旁路，仅积分器有效）：

$$t = I_{OSR} / f_{CKIN}$$（... 但是 CNVCNT = 0）

连续模式不适用于注入转换。注入转换可通过定时器触发进行启动，以便使用精确时序仿真连续模式。

如果常规连续转换正在进行 (RCONT = 1)，并且如果对请求常规连续转换 (RCONT = 1) 的 DFSDM_FLTxCR1 寄存器进行写访问，则会从下一转换周期重新启动常规连续转换（类似于新的常规连续转换应用于新通道选择，即使 DFSDM_FLTxCR1 寄存器中无更改，亦如此）。

29.4.17 请求优先级

注入转换的优先级高于常规转换。正在进行的常规转换会立即被注入转换请求中断；在注入转换完成后将重新启动此常规转换。

如果某个注入转换处于待处理状态或正在进行，则不能启动其他注入转换：只要（DFSDM_FLTxISR 寄存器中）位 JCIP 为“1”，就会忽略启动注入转换（通过 JSWSTART 或触发）的请求。

同理，如果某个常规转换处于待处理状态或正在进行，则不能启动其他常规转换：只要（DFSDM_FLTxISR 寄存器中）位 RCIP 为“1”，就会忽略启动常规转换（使用 RSWSTART）的请求。

不过，如果在进行某个常规转换时请求注入转换，则常规转换会立即停止，并会启动注入转换。常规转换之后会重新启动，并且此延迟的重新启动会通过位 RPEND 进行表示。

注入转换的优先级高于常规转换，因为注入转换可暂时中断连续的常规转换序列。完成注入转换序列时，如果 RCONT 仍置 1 则会重新启动连续常规转换（RPEND 位将发出基于首次常规转换结果的延迟启动信号）。

通过对 DFSDM 进行相同的写操作启动相关操作时，或者，如果在其他操作结束时有多个操作处于待处理状态，则优先级也至关重要。例如，假设注入转换正在进行 (JCIP = 1)，则对 DFSDM_FLTxCR1 进行单次写操作将向 RSWSTART 写入“1”，以请求常规转换。完成注入序列时，优先级指示接下来执行常规转换，并且通过 RPEND 位指示相应的延迟启动。

29.4.18 在运行模式下的功耗优化

为降低功耗，DFSDM 滤波器和积分器会在未用于转换时自动进入空闲状态（RCIP = 0，JCIP = 0）。

29.5 DFSDM 中断

为提升 CPU 性能，已实现了一组与 CPU 事件发生相关的中断：

- 注入转换结束中断：
  - 由 DFSDM_FLTxCR2 寄存器中的 JEOCIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 JEOCF 位指示
  - 通过读取 DFSDM_FLTxJDATAR 寄存器（注入数据）清除
  - 指示哪一个通道发生了转换结束，并在 DFSDM_FLTxJDATAR 寄存器的 JDATACH[1:0] 位中进行报告
- 常规转换结束中断：
  - 由 DFSDM_FLTxCR2 寄存器中的 REOCIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 REOCF 位指示
  - 通过读取 DFSDM_FLTxRDATAR 寄存器（常规数据）清除
  - 指示哪一个通道发生了转换结束，并在 DFSDM_FLTxRDATAR 寄存器的 RDATACH[1:0] 位中进行报告
- 注入转换数据溢出中断：
  - 当注入的转换数据未从 DFSDM_FLTxJDATAR 寄存器中读取（通过 CPU 或 DMA），并被新注入转换覆盖时，会发生此中断
  - 由 DFSDM_FLTxCR2 寄存器中的 JOVRIE 位使能

  - 由 DFSDM_FLTxISR 寄存器中的 JOVRF 位指示
  - 通过向 DFSDM_FLTxICR 寄存器中的 CLRJOVRF 位写入“1”进行清除
- 常规转换数据溢出中断：
  - 当常规转换数据未从 DFSDM_FLTxRDATAR 寄存器中读取（通过 CPU 或 DMA），并被新常规转换覆盖时，会发生此中断
  - 由 DFSDM_FLTxCR2 寄存器中的 ROVRIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 ROVRF 位指示
  - 通过向 DFSDM_FLTxICR 寄存器中的 CLRROVRF 位写入“1”进行清除
- 模拟看门狗中断：
  - 当转换数据（输出数据或模拟看门狗滤波器中的数据——具体取决于 DFSDM_FLTxCR1 寄存器中的 AWFSEL 位设置）超出 DFSDM_FLTxAWHTR/ DFSDM_FLTxAWLTR 寄存器中的阈值上限或低于相应阈值下限时，会发生此中断
  - 通过（所选通道 AWDCH[3:0] 的）DFSDM_FLTxCR2 寄存器 AWDIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 AWDF 位指示
  - 通过 DFSDM_FLTxAWSR 寄存器中的 AWHTF[3:0] 和 AWLTF[3:0] 字段单独指示模拟看门狗阈值上限或下限错误
  - 通过向 DFSDM_FLTxAWCFR 寄存器中的相应 CLRAWHTF[3:0] 或 CLRAWLTF[3:0] 位写入“1”进行清除
- 短路检测器中断：
  - 稳定的数据数超出 DFSDM_CHyAWSCDR 寄存器中的阈值时会发生此中断
  - 通过（通过 DFSDM_CHyCFGR1 寄存器 SCDEN 位选择的通道的）DFSDM_FLTxCR2 寄存器的 SCDIE 位使能
  - 由 DFSDM_FLTxISR 寄存器的 SCDF[3:0] 位指示（还会报告发生短路检测器事件的通道）
  - 通过向 DFSDM_FLTxICR 寄存器中的相应 CLRSCDF[3:0] 位写入“1”进行清除
- 通道时钟缺失中断：
  - CKINy 引脚上出现时钟缺失时会发生此中断（请参见*第 29.4.4 节：串行通道收发器*的*时钟缺失检测*）
  - 通过（通过 DFSDM_CHyCFGR1 寄存器 CKABEN 位选择的通道的）DFSDM_FLTxCR2 寄存器的 CKABIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 CKABF[y] 位指示
  - 通过向 DFSDM_FLTxICR 寄存器中的 CLRCKABF[y] 位写入“1”进行清除

表 235. DFSDM 中断请求

中断事件	事件标志	事件/中断清除方法	中断使能控制位
注入转换结束	JEOCF	读取 DFSDM_FLTxJDATAR	JEOCIE
常规转换结束	REOCF	读取 DFSDM_FLTxRDATAR	REOCIE
注入数据溢出	JOVRF	写入 CLRJOVRF = 1	JOVRIE
常规数据溢出	ROVRF	写入 CLRROVRF = 1	ROVRIE
模拟看门狗	AWDF、AWHTF[3:0], AWLTF[3:0]	写入 CLRAWHTF[3:0] = 1 写入 CLRAWLTF[3:0] = 1	AWDIE，(AWDCH[3:0])

表 235. DFSDM 中断请求（续）

中断事件	事件标志	事件/中断清除方法	中断使能控制位
短路检测器	SCDF[3:0]	写入 CLRSCDF[3:0] = 1	SCDIE、(SCDEN)
通道时钟缺失	CKABF[3:0]	写入 CLRCKABF[3:0] = 1	CKABIE、(CKABEN)

29.6 DFSDM DMA 传输

为降低 CPU 干预，可使用 DMA 传输将转换传输至内存。注入转换的 DMA 传输通过将 DFSDM_FLTxCR1 寄存器的 JDMAEN 位置 1 使能。常规转换的 DMA 传输通过将 DFSDM_FLTxCR1 寄存器的 RDMAEN 位置 1 使能。

注： *通过 DMA 传输，中断标志在注入或常规转换结束时自动清零（DFSDM_FLTxISR 寄存器中的 JEOCF 或 REOCF 位），因为 DMA 正在读取 DFSDM_FLTxJDATAR 或 DFSDM_FLTxRDATAR 寄存器。*

29.7 DFSDM 通道 y 寄存器 (y=0..3)

寄存器写访问必须使用字访问（32 位），DFSDM_CHyDATINR 寄存器除外。对 DFSDM_CHyDATINR 寄存器的写访问可以是字访问（32 位），也可以是半字访问（16 位）。

29.7.1 DFSDM 通道 y 配置寄存器 (DFSDM_CHyCFGR1)

DFSDM channel y configuration register

该寄存器指定通道 y 所使用的参数。

偏移地址：0x00 + 0x20 * y（y = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DFSDMEN	CKOUTSRC	Res.	Res.	Res.	Res.	Res.	Res.	CKOUTDIV[7:0]							
rw	rw							rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
rw	rw	rw	rw				rw	rw	rw	rw		rw	rw	rw	rw

位 31 **DFSDMEN**：DFSDM 接口全局使能 (Global enable for DFSDM interface)

0：禁止 DFSDM 接口

1：使能 DFSDM 接口

如果已使能 DFSDM 接口，则该接口将根据使能的 y 通道和使能的 x 滤波器设置（DFSDM_CHyCFGR1 的 CHEN 位和 DFSDM_FLTxCR1 的 DFEN 位）进行工作。通过设置 DFSDMEN = 0 清除数据：

–所有寄存器 DFSDM_FLTxISR 均被设为复位状态 (x = 0..1)

–所有寄存器 DFSDM_FLTxAWSR 均被设为复位状态 (x = 0..1)

注意：DFSDMEN 仅存在于 DFSDM_CH0CFGR1 寄存器（通道 y = 0）中

位 30 **CKOUTSRC**：输出串行时钟源选择 (Output serial clock source selection)

0：输出时钟的源来自系统时钟

1：输出时钟的源来自音频时钟

只有在（DFSDM_CH0CFGR1 寄存器中）DFSDMEN = 0 时，才能修改该值。

注意：CKOUTSRC 仅存在于 DFSDM_CH0CFGR1 寄存器（通道 y = 0）中

位 29:24 保留，必须保持复位值。

位 23:16 **CKOUTDIV[7:0]**：输出串行时钟分频器 (Output serial clock divider)

0：禁止输出时钟生成（CKOUT 信号被设为低电平状态）

1-255：为 CKOUT 信号定义串行时钟输出的系统时钟分频，范围为 2 - 256（分频器值 = CKOUTDIV+1）。

CKOUTDIV 还定义了时钟缺失检测的阈值。

只有在（DFSDM_CH0CFGR1 寄存器中）DFSDMEN = 0 时，才能修改该值。

如果 DFSDMEN = 0（DFSDM_CH0CFGR1 寄存器中），则 CKOUT 信号被置为低电平状态（在 DFSDMEN = 0 后的一个 DFSDM 时钟周期内执行设置）。

注意：CKOUTDIV 仅存在于 DFSDM_CH0CFGR1 寄存器（通道 y = 0）中

位 15:14 **DATPACK[1:0]**：DFSDM_CHyDATINR 寄存器数据封装模式 (Data packing mode in DFSDM_CHyDATINR register)

0：标准：DFSDM_CHyDATINR 寄存器中的输入数据仅存储在 INDAT0[15:0] 中。要清空 DFSDM_CHyDATINR 寄存器，必须由 DFSDM 滤波器从通道 y 中读取一个采样。

1：交错：DFSDM_CHyDATINR 寄存器中的输入数据存储为两个采样：

–第一个采样位于 INDAT0[15:0] 中（已分配至通道 y）

–第二个采样位于 INDAT1[15:0] 中（已分配至通道 y）

要清空 DFSDM_CHyDATINR 寄存器，必须由数字滤波器从通道 y 中读取两个采样（INDAT0[15:0] 部分被读为第一个采样，而 INDAT1[15:0] 部分则被读为下一个采样）。

2：双通道：DFSDM_CHyDATINR 寄存器中的输入数据存储为两个采样：

–第一个采样位于 INDAT0[15:0] 中（已分配至通道 y）

–第二个样采样于 INDAT1[15:0] 中（已分配至通道 y+1）

要清空 DFSDM_CHyDATINR 寄存器，必须由数字滤波器从通道 y 中读取第一个样本，且必须由另一数字滤波器从通道 y+1 中读取第二个样本。只有在通道数为偶数（y = 0、2）时才支持双通道模式，对于通道数为奇数（y = 1、3）的情况，DFSDM_CHyDATINR 被写保护。如果偶数通道被设为双通道模式，则下一奇数通道必须进入标准模式 (DATPACK[1:0] = 0)，以便与偶数通道正确配合。

3：保留

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该值。

位 13:12 **DATMPX[1:0]**：通道 y 输入数据复用器 (Input data multiplexer for channel y)

0：通道 y 输入数据取自外部串行输入，为 1 位值。DFSDM_CHyDATINR 寄存器被写保护。

1：通道 y 输入数据取自内部模数转换器 ADC_{y+1} 输出寄存器更新，为 16 位值（若 ADC_{y+1} 存在）。ADC 输出数据被写入 DFSDM_CHyDATINR 寄存器的 INDAT0[15:0] 部分。

2：通道 y 输入数据取自内部 DFSDM_CHyDATINR 寄存器（通过 CPU/DMA 直接写操作实现）。可写入一个或两个 16 位数据采样，具体取决于 DATPACK[1:0] 位字段设置。

3：保留

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该值。

位 11:9 保留，必须保持复位值。

位 8 **CHINSEL**：通道输入选择

0：通道输入取自同一通道 y 的引脚。

1：通道输入取自下一通道（通道 (y+1) 取 8 的模）的引脚。

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该值。

位 7 **CHEN**：使能通道 y

0：禁止通道 y

1：使能通道 y

如果使能了通道 y，则会根据给定通道设置开始接收串行数据。

位 6 **CKABEN**：使能通道 y 时钟缺失检测器

0：禁止通道 y 时钟缺失检测器

1：使能通道 y 时钟缺失检测器

位 5 **SCDEN**：通道 y 短路检测器使能 (Short-circuit detector enable on channel y)

0：输入通道 y 将不受短路检测器保护

1：输入通道 y 将持续受短路检测器保护

位 4 保留，必须保持复位值。

位 3:2 **SPICKSEL[1:0]**：通道 y SPI 时钟选择 (SPI clock select for channel y)

0：时钟来自外部 CKINy 输入——基于 SITP[1:0] 的采样点

1：时钟来自内部 CKOUT 输出——基于 SITP[1:0] 的采样点

2：时钟来自内部 CKOUT——在 CKOUT 每个第二个下降沿的采样点。
用于连接外部 ΣΔ 调制器，调制器将时钟输入（来自 CKOUT）除以 2，来生成输出串行通信时钟（该输出时钟变化在各个时钟输入上升沿有效）。

3：时钟来自内部 CKOUT 输出——在 CKOUT 每个第二个上升沿的采样点。
用于连接外部 ΣΔ 调制器，调制器将时钟输入（来自 CKOUT）除以 2，来生成输出串行通信时钟（该输出时钟变化在各个时钟输入下降沿有效）。

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该值。

位 1:0 **SITP[1:0]**：通道 y 串行接口类型 (Serial interface type for channel y)

00：上升沿选通数据 SPI

01：下降沿选通数据 SPI

10：DATINy 引脚曼彻斯特编码输入：上升沿 = 逻辑 0，下降沿 = 逻辑 1

11：DATINy 引脚曼彻斯特编码输入：上升沿 = 逻辑 1，下降沿 = 逻辑 0

只有在（DFSDM_CHyCFGR1 寄存器中的）CHEN = 0 时，才能修改该值。

29.7.2 DFSDM 通道 y 配置寄存器 (DFSDM_CHyCFGR2)

DFSDM channel y configuration register

该寄存器指定通道 y 所使用的参数。

偏移地址：0x04 + 0x20 * y（y = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OFFSET[23:8]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET[7:0]								DTRBS[4:0]					Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			

位 31:8 **OFFSET[23:0]**：通道 y 24 位校准偏移 (24-bit calibration offset for channel y)

对于通道 y 的每一次转换结果，都执行 OFFSET。

该值由软件设置。

位 7:3 **DTRBS[4:0]**：通道 y 数据右移位 (Data right bit-shift for channel y)

0-31：定义积分器输出数据结果的移位——将向右移多少位以获得最终结果。在进行偏移校正之前执行移位。数据移位将结果四舍五入为最接近的整数值。移位结果的符号保留（以便得到有效 24 位有符号格式的结果数据）。

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该值。

位 2:0 保留，必须保持复位值。

29.7.3 DFSDM 通道 y 模拟看门狗和短路检测器寄存器 (DFSDM_CHyAWSCDR)

DFSDM channel y analog watchdog and short-circuit detector register

通道 y 短路检测器和模拟看门狗设置

偏移地址：0x08 + 0x20 * y（y = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]				
								rw	rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
rw	rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:22 **AWFORD[1:0]**：通道 y 模拟看门狗 Sinc 滤波器阶数 (Analog watchdog Sinc filter order on channel y)

0：FastSinc 滤波器类型
1：$Sinc^1$ 滤波器类型
2：$Sinc^2$ 滤波器类型
3：$Sinc^3$ 滤波器类型

$Sinc^x$ 滤波器类型传递函数：

$$高阻态 = \left(\frac{1-z^{-FOSR}}{1-z^{-1}}\right)^x$$

FastSinc 滤波器类型传递函数：

$$高阻态 = \left(\frac{1-z^{-FOSR}}{1-z^{-1}}\right)^2 \cdot (1+z^{-(2\cdot FOSR)})$$

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该位。

位 21 保留，必须保持复位值。

位 20:16 **AWFOSR[4:0]**：通道 y 模拟看门狗滤波器过采样率（抽取率）(Analog watchdog filter oversampling ratio (decimation rate) on channel y)

0-31：定义 Sinc 类型滤波器的长度，长度范围为 1 - 32 (AWFOSR + 1)。该数字还表示模拟数据速率的抽取率。
只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该位。
注意：如果 AWFOSR = 0，则滤波器无影响（滤波器旁路）。

位 15:12 **BKSCD[3:0]**：通道 y 短路检测器断路信号分配 (Break signal assignment for short-circuit detector on channel y)

BKSCD[i] = 0：断路 i 信号未分配至通道 y 短路检测器
BKSCD[i] = 1：断路 i 信号已分配至通道 y 短路检测器

位 11:8 保留，必须保持复位值。

位 7:0 **SCDT[7:0]**：通道 y 的短路检测器阈值 (short-circuit detector threshold for channel y)

这些位可由软件写入，用来定义短路检测器的计数器阈值。如果达到该值，则在给定通道上发生短路检测器事件。

29.7.4 DFSDM 通道 y 看门狗滤波器数据寄存器 (DFSDM_CHyWDATR)

DFSDM channel y watchdog filter data register

该寄存器包含的数据来自于与输入通道 y 关联的模拟看门狗滤波器。

偏移地址：0x0C + 0x20 * y（y = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDATA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **WDATA[15:0]**：输入通道 y 看门狗数据 (Input channel y watchdog data)

由输入通道 y 模拟看门狗滤波器转换的数据。该通道进行连续的数据转换（无触发），转换分辨率受一定限制（OSR=1...32/sinc 阶数 = 1...3）。

29.7.5 DFSDM 通道 y 数据输入寄存器 (DFSDM_CHyDATINR)

DFSDM channel y data input register (DFSDM_CHyDATINR)

该寄存器包含的 16 位输入数据将通过 DFSDM 滤波器模块进行处理。写访问可以是字访问（32 位），也可以是半字访问（16 位）。

偏移地址：0x10 + 0x20 * y（y = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INDAT1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INDAT0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **INDAT1[15:0]**：通道 y 或通道 y+1 的输入数据 (Input data for channel y or channel y+1)

当 DATMPX[1:0] = 1 或 DATMPX[1:0] = 2 时，输入并行通道数据将通过数字滤波器进行处理。

数据可由 CPU/DMA 写入（当 DATMPX[1:0]=2 时），也可以由内部 ADC 直接写入（当 DATMPX[1:0] = 1 时）。

当 DATPACK[1:0] = 0（标准模式）时

INDAT0[15:0] 被写保护（不用于输入采样）。

当 DATPACK[1:0] = 1（交错模式）时

第二个通道 y 数据采样存储到 INDAT1[15:0] 中。第一个通道 y 数据采样存储到 INDAT0[15:0] 中。DFSDM_FLTx 滤波器依次读取该两个采样，作为两个通道 y 数据采样。

当 DATPACK[1:0] = 2（双模式）时

对于偶数 y 通道：INDAT1[15:0] 中的采样自动复制到通道 (y+1) 的 INDAT0[15:0] 中。

对于奇数 y 通道：INDAT1[15:0] 被写保护。

更多详细信息，请参见*第 29.4.6 节：并行数据输入*。

INDAT0[15:1] 采用 16 位带符号格式。

位 15:0 **INDAT0[15:0]**：通道 y 的输入数据

当 DATMPX[1:0] = 1 或 DATMPX[1:0] = 2 时，输入并行通道数据将通过数字滤波器进行处理。

数据可由 CPU/DMA 写入（当 DATMPX[1:0]=2 时），也可以由内部 ADC 直接写入（当 DATMPX[1:0] = 1 时）。

当 DATPACK[1:0] = 0（标准模式）时

通道 y 数据采样存储到 INDAT0[15:0] 中。

当 DATPACK[1:0] = 1（交错模式）时

第一个通道 y 数据采样存储到 INDAT0[15:0] 中。第二个通道 y 数据采样存储到 INDAT1[15:0] 中。DFSDM_FLTx 滤波器依次读取该两个采样，作为两个通道 y 数据采样。

当 DATPACK[1:0] = 2（双模式）时

对于偶数 y 通道：通道 y 数据采样存储到 INDAT0[15:0] 中。

对于奇数 y 通道：INDAT0[15:0] 被写保护。

更多详细信息，请参见*第 29.4.6 节：并行数据输入*。

INDAT0[15:0] 采用 16 位带符号格式。

29.7.6 DFSDM 通道 y 延迟寄存器 (DFSDM_CHyDLYR)

DFSDM channel y delay register

偏移地址：0x14 + 0x20 * y（y = 0 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLSSKP[5:0]					
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持复位值。

位 5:0 **PLSSKP[5:0]**：面向输入数据跳过功能的要跳过脉冲数 (Pulses to skip for input data skipping function)
0-63：定义要跳过的串行输入样品数。写入该字段后，立即应用跳过功能。读取 PLSSKP[5:0]，可返回要跳过的脉冲的当前值。如果 PLSSKP[5:0]=0，则已跳过全部所需数据样品。
注意：PLSSKP[5:0] 不为零时，用户也可更新 PLSSKP[5:0]。

29.8 DFSDM 滤波器 x 模块寄存器 (x=0..1)

寄存器写访问必须使用字访问（32 位），DFSDM_CHyDATINR 寄存器除外。

29.8.1 DFSDM 滤波器 x 控制寄存器 1 (DFSDM_FLTxCR1)

DFSDM filter x control register 1

偏移地址：0x100 + 0x80 * x（x = 0 到 1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AWFSEL	FAST	Res.	Res.	Res.	RCH[1:0]		Res.	Res.	RDMAEN	Res.	RSYNC	RCONT	RSWSTART	Res.
	rw	rw				rw	rw			rw		rw	rw	rt_w1	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	JEXTEN[1:0]		JEXTSEL[4:0]					Res.	Res.	JDMAEN	JSCAN	JSYNC	Res.	JSWSTART	DFEN
	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw		rt_w1	rw

位 31 保留，必须保持复位值。

位 30 **AWFSEL**：模拟看门狗快速模式选择 (Analog watchdog fast mode select)
0：数据输出值的模拟看门狗（在数字滤波器之后）。在偏移校正和移位后进行比较
1：通道收发器值的模拟看门狗（在看门狗滤波器之后）

位 29 **FAST**：常规转换的快速转换模式选择 (Fast conversion mode selection for regular conversions)

0：禁止快速转换模式

1：使能快速转换模式

在连续模式下进行常规转换时，如果使能了快速模式，则每次转换（第一次转换除外）都比在标准模式下的转换执行得快。该位对于不连续的转换没有影响。

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

如果 FAST=0（或者 FAST=1 连续模式下的第一次转换），则：

$t = [F_{OSR} * (I_{OSR}-1 + F_{ORD}) + F_{ORD}] / f_{CKIN}$..... 适用于 $Sinc^X$ 滤波器

$t = [F_{OSR} * (I_{OSR}-1 + 4) + 2] / f_{CKIN}$..... 适用于 FastSinc 滤波器

如果在连续模式下 FAST=1（第一次转换除外），则：

$t = [F_{OSR} * I_{OSR}] / f_{CKIN}$

如果 $F_{OSR} = F_{OSR}[9:0] + 1 = 1$（滤波器旁路，仅积分器有效），则：

$t = I_{OSR} / f_{CKIN}$（... 但是 CNVCNT = 0）

其中，f_{CKIN} 为（给定通道 CKINy 引脚上的）通道输入时钟频率，在并行数据输入的情况下，其表示输入数据速率。

位 28:26 保留，必须保持复位值。

位 25:24 **RCH[1:0]**：常规通道选择 (Regular channel selection)

0：将通道 0 选作常规通道

1：将通道 1 选作常规通道

...

3：将通道 3 选作常规通道

当 RCIP = 1 时写入此位，并在下一个常规转换开始时生效。这在连续模式下（当 RCONT = 1 时）特别有用。它还会影响待处理的常规转换（因正在进行注入转换而暂停）。

位 23:22 保留，必须保持复位值。

位 21 **RDMAEN**：使能 DMA 通道以读取常规转换数据 (DMA channel enabled to read data for the regular conversion)

0：未使能 DMA 通道来读取常规数据

1：已使能 DMA 通道来读取常规数据

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 20 保留，必须保持复位值。

位 19 **RSYNC**：使用 DFSDM_FLT0 同步启动常规转换 (Launch regular conversion synchronously with DFSDM_FLT0)

0：不使用 DFSDM_FLT0 同步启动常规转换

1：在 DFSDM_FLT0 中启动常规转换的同时，在该 DFSDM_FLTx 中启动常规转换

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 18 **RCONT**：常规转换的连续模式选择 (Continuous mode selection for regular conversions)

0：对于每个转换请求，只转换一次常规通道

1：在发出每个转换请求后，重复转换常规通道

在连续常规转换进行过程中，向该位写入“0”将立即停止连续模式。

位 17 **RSWSTART**：软件启动常规通道转换 (Software start of a conversion on the regular channel)

0：写入“0”无影响

1：写入“1”会发出启动常规通道转换的请求，并会使 RCIP 变为“1”。如果已经设置 RCIP=1，则对 RSWSTART 进行的写操作无效。如果 RSYNC = 1，则写入“1”无效。

此位始终读为“0”。

位 16:15 保留，必须保持复位值。

位 14:13 **JEXTEN[1:0]**：注入转换触发使能和触发边沿选择 (Trigger enable and trigger edge selection for injected conversions)

00：禁止触发检测
01：所选触发的每个上升沿都会发出启动注入转换的请求
10：所选触发的每个下降沿都会发出启动注入转换的请求
11：所选触发的上升沿和下降沿都会发出启动注入转换的请求
只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 12:8 **JEXTSEL[4:0]**：启动注入转换的触发信号选择 (Trigger signal selection for launching injected conversions)

0x0-0x1F：根据下表选择的触发输入（内部或外部触发）。
只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

注意：同步触发的延迟最大为一个 $f_{DFSDMCLK}$ 时钟周期（存在确定的抖动），异步触发的延迟为 2-3 个 $f_{DFSDMCLK}$ 时钟周期（抖动长达 1 个周期）。

DFSDM_FLTx
0x00dfsdm_jtrg0
0x01dfsdm_jtrg1
...
0x1Edfsdm_jtrg30
0x1Fdfsdm_jtrg31
请参见*表 231：DFSDM 触发器连接*。

位 7:6 保留，必须保持复位值。

位 5 **JDMAEN**：为注入通道组使能 DMA 以读取数据 (DMA channel enabled to read data for the injected channel group)

0：未使能 DMA 通道来读取注入数据
1：已使能 DMA 通道来读取注入数据
只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 4 **JSCAN**：注入转换的扫描转换模式 (Scanning conversion mode for injected conversions)

0：对注入通道组执行一次通道转换，然后选择该通道组下一个通道。
1：从选择的最低通道开始，对注入通道组执行连续转换。
只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。
如果 JSCAN=0，则对 JCHG 进行写入操作会将通道选择复位为最小所选通道。

位 3 **JSYNC**：使用 DFSDM_FLT0 JSWSTART 触发同步启动注入转换 (Launch an injected conversion synchronously with the DFSDM_FLT0 JSWSTART trigger)

0：不使用 DFSDM_FLT0 同步启动注入转换
1：在 DFSDM_FLT0 中通过 JSWSTART 触发启动注入转换的同时，在该 DFSDM_FLTx 中启动注入转换
只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 2 保留，必须保持复位值。

位 1 **JSWSTART**：启动注入通道组转换 (Start a conversion of the injected group of channels)

0：写入“0”无影响。
1：写入“1”会发出转换注入通道组请求，同时会使 JCIP 变为“1”。如果已经设置 JCIP = 1，则对 JSWSTART 进行的写操作无效。如果 JSYNC = 1，则写入“1”无效。
此位始终读为“0”。

位 0 **DFEN**：使能 DFSDM_FLTx

0：禁止 DFSDM_FLTx。给定 DFSDM_FLTx 的所有转换均立即停止，DFSDM_FLTx 的所有功能均停止。

1：使能 DFSDM_FLTx。如果使能了 DFSDM_FLTx，则 DFSDM_FLTx 会根据其设置开始工作。

通过设置 DFEN = 0 清零的数据：

– 寄存器 DFSDM_FLTxISR 被设为复位状态

– 寄存器 DFSDM_FLTxAWSR 被设为复位状态

29.8.2 DFSDM 滤波器 x 控制寄存器 2 (DFSDM_FLTxCR2)

DFSDM filter x control register 2

偏移地址：0x104 + 0x80 * x（x = 0 到 1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDCH[3:0]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	EXCH[3:0]				Res.	CKABIE	SCDIE	AWDIE	ROVRIE	JOVRIE	REOCIE	JEOCIE
				rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:20 保留，必须保持复位值。

位 19:16 **AWDCH[3:0]**：模拟看门狗通道选择

这些位用于选择由模拟看门狗持续监控的输入通道。

AWDCH[y] = 0：在通道 y 上禁止模拟看门狗

AWDCH[y] = 1：在通道 y 上使能模拟看门狗

位 15:12 保留，必须保持复位值。

位 11:8 **EXCH[3:0]**：极值检测器通道选择 (Extremes detector channel selection)

这些位选择极值检测器要采用的输入通道。

EXCH[y] = 0：极值检测器不接受通道 y 的数据

EXCH[y] = 1：极值检测器接受通道 y 的数据

位 7 保留，必须保持复位值。

位 6 **CKABIE**：使能时钟缺失中断

0：禁止检测通道输入时钟缺失中断

1：使能检测通道输入时钟缺失中断

请参见 DFSDM_FLTxISR 中的 CKABF[3:0] 说明。

注意：CKABIE 仅存在于 DFSDM_FLT0CR2 寄存器中（滤波器 x = 0）

位 5 **SCDIE**：短路检测器中断使能 (Short-circuit detector interrupt enable)

0：禁止短路检测器中断

1：使能短路检测器中断

请参见 DFSDM_FLTxISR 中的 SCDF[3:0] 说明。

注意：SCDIE 仅存在于 DFSDM_FLT0CR2 寄存器中（滤波器 x = 0）

位 4 **AWDIE**：模拟看门狗中断使能 (Analog watchdog interrupt enable)
0：禁止模拟看门狗中断
1：使能模拟看门狗中断
请参见 DFSDM_FLTxISR 中的 AWDF 说明。

位 3 **ROVRIE**：使能常规数据溢出中断
0：禁止常规数据溢出中断
1：使能常规数据溢出中断
请参见 DFSDM_FLTxISR 中的 ROVRF 说明。

位 2 **JOVRIE**：使能注入数据溢出中断
0：禁止注入数据溢出中断
1：使能注入数据溢出中断
请参见 DFSDM_FLTxISR 中的 JOVRF 说明。

位 1 **REOCIE**：使能常规转换结束中断
0：禁止常规转换结束中断
1：使能常规转换结束中断
请参见 DFSDM_FLTxISR 中的 REOCF 说明。

位 0 **JEOCIE**：使能注入转换结束中断
0：禁止注入转换结束中断
1：使能注入转换结束中断
请参见 DFSDM_FLTxISR 中的 JEOCF 说明。

29.8.3 DFSDM 滤波器 x 中断和状态寄存器 (DFSDM_FLTxISR)

DFSDM filter x interrupt and status register

偏移地址：0x108 + 0x80 * x（x = 0 到 1）

复位值：0x00FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	SCDF[3:0]				Res.	Res.	Res.	Res.	CKABF[3:0]			
				r	r	r	r					r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RCIP	JCIP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDF	ROVRF	JOVRF	REOCF	JEOCF
	r	r									r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27:24 **SCDF[3:0]**：短路检测器标志 (short-circuit detector flag)
SDCF[y]=0：通道 y 上未发生任何短路检测器事件
SDCF[y]=1：通道 y 上的短路检测器计数器达到在 DFSDM_CHyAWSCDR 寄存器中编程的值
该位由硬件置 1，该位可由软件使用 DFSDM_FLTxICR 寄存器中的相应 CLRSCDF[y] 位清零。当 CHEN[y] = 0（给定通道被禁止）时，SCDF[y] 还可以由硬件清零。
注意：SCDF[3:0] 仅存在于 DFSDM_FLT0ISR 寄存器中（滤波器 x = 0）

位 23:20 保留，必须保持复位值。

位 19:16 **CKABF[3:0]**：时钟缺失标志 (Clock absence flag)

CKABF[y]=0：通道 y 上存在时钟信号。

CKABF[y]=1：通道 y 上不存在时钟信号。

当检测到通道 y 上不存在时钟信号时，给定 y 位由硬件置 1。当 CHEN = 0 时（请参见 DFSDM_CHyCFGR1 寄存器），它由硬件保持为 CKABF[y] = 1 状态。当收发器尚未被同步时，它由硬件保持为 CKABF[y] = 1 状态。它可由软件使用 DFSDM_FLTxICR 寄存器中的相应 CLRCKABF[y] 位清零。

注意：CKABF[3:0] 仅存在于 DFSDM_FLT0ISR 寄存器中（滤波器 x = 0）

位 15 保留，必须保持复位值。

位 14 **RCIP**：常规转换正在进行状态 (Regular conversion in progress status)

0：未发出转换常规通道的请求

1：常规通道转换正在进行，或一个常规转换请求正在等待处理

当 RCIP = 1 时，忽略启动常规转换的请求。

位 13 **JCIP**：注入转换正在进行状态 (Injected conversion in progress status)

0：未发出转换注入通道组的请求（软件和触发器均未发出请求）

1：注入通道组转换处于正在进行状态（向 JSWSTART 写入了“1”）或者一个注入转换请求正在等待处理（由于触发检测）

当 JCIP = 1 时，忽略启动注入转换的请求。

位 12:5 保留，必须保持复位值。

位 4 **AWDF**：模拟看门狗

0：未发生模拟看门狗事件

1：模拟看门狗模块检测到电压超出在 DFSDM_FLTxAWLTR 或 DFSDM_FLTxAWHTR 寄存器中编程的值。

该位由硬件置 1，它由软件通过以下方式清零：将 DFSDM_FLTxAWSR 寄存器中的所有源标志位 AWHTF[3:0] 和 AWLTF[3:0] 清零（通过向 DFSDM_FLTxAWCFR 寄存器中的清零位写入“1”实现）。

位 3 **ROVRF**：常规转换溢出标志 (Regular conversion overrun flag)

0：未发生常规转换溢出

1：发生了常规转换溢出，也就是说，常规转换已经完成，但 REOCF 仍然为“1”。RDATAR 不受溢出影响

该位由硬件置 1，该位可由软件使用 DFSDM_FLTxICR 寄存器中的 CLRROVRF 位清零。

位 2 **JOVRF**：注入转换溢出标志 (Injected conversion overrun flag)

0：未发生注入转换溢出

1：发生了注入转换溢出，也就是说，注入转换已经完成，但 JEOCF 仍然为“1”。JDATAR 不受溢出影响

该位由硬件置 1，该位可由软件使用 DFSDM_FLTxICR 寄存器中的 CLRJOVRF 位清零。

位 1 **REOCF**：常规转换结束标志 (End of regular conversion flag)

0：未完成常规转换

1：已完成常规转换并可以读取其数据

该位由硬件置 1，当软件或 DMA 读取 DFSDM_FLTxRDATAR 时，该位清零。

位 0 **JEOCF**：注入转换结束标志 (End of injected conversion flag)

0：未完成注入转换

1：已完成注入转换并可以读取其数据

该位由硬件置 1，当软件或 DMA 读取 DFSDM_FLTxJDATAR 时，该位清零。

注： *对于每个标志位，可通过将 DFSDM_FLTxCR2 中的相应位置 1 来使能中断。如果调用了中断，则必须在退出中断服务程序之前将该标志清零。*

当 DFEN = 0 时，DFSDM_FLTxISR 的所有位都会自动复位。

29.8.4 DFSDM 滤波器 x 中断标志清零寄存器 (DFSDM_FLTxICR)

DFSDM filter x interrupt flag clear register

偏移地址：0x10C + 0x80 * x（x = 0 到 1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CLRSCDF[3:0]				Res.	Res.	Res.	Res.	CLRCKABF[3:0]			
				rc_w1	rc_w1	rc_w1	rc_w1					rc_w1	rc_w1	rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLRROVRF	CLRJOVRF	Res.	Res.
												rc_w1	rc_w1		

位 31:28 保留，必须保持复位值。

位 27:24 **CLRSCDF[3:0]**：清零短路检测器标志 (Clear the short-circuit detector flag)

CLRSCDF[y]=0：写入“0”无影响

CLRSCDF[y]=1：向位置 y 写入“1”会将 DFSDM_FLTxISR 寄存器中的相应 SCDF[y] 位清零

注意：CLRSCDF[3:0] 仅存在于 DFSDM_FLT0ICR 寄存器中（滤波器 x = 0）

位 23:20 保留，必须保持复位值。

位 19:16 **CLRCKABF[3:0]**：清零时钟缺失标志 (Clear the clock absence flag)

CLRCKABF[y]=0：写入“0”无影响

CLRCKABF[y]=1：向位置 y 写入“1”会将 DFSDM_FLTxISR 寄存器中的相应 CKABF[y] 位清零。当收发器尚未被同步时，时钟缺失标志会置 1，且无法由 CLRCKABF[y] 清零。

注意：CLRCKABF[3:0] 仅存在于 DFSDM_FLT0ICR 寄存器中（滤波器 x = 0）

位 15:4 保留，必须保持复位值。

位 3 **CLRROVRF**：将常规转换溢出标志清零 (Clear the regular conversion overrun flag)

0：写入“0”无影响

1：写入“1”会将 DFSDM_FLTxISR 寄存器中的 ROVRF 位清零

位 2 **CLRJOVRF**：将注入转换溢出标志清零 (Clear the injected conversion overrun flag)

0：写入“0”无影响

1：写入“1”会将 DFSDM_FLTxISR 寄存器中的 JOVRF 位清零

位 1:0 保留，必须保持复位值。

注： *DFSDM_FLTxICR 的位始终读为“0”。*

29.8.5 DFSDM 滤波器 x 注入通道组选择寄存器 (DFSDM_FLTxJCHGR)

DFSDM filter x injected channel group selection register

偏移地址：0x110 + 0x80 * x（x = 0 到 1）

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JCHG[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **JCHG[3:0]**：注入通道组选择

JCHG[y] = 0：通道 y 不属于注入组

JCHG[y] = 1：通道 y 属于注入组

如果 JSCAN = 1，则依次转换每一个所选通道。首先转换最小通道（通道 0，如果已选择），最后转换最大所选通道。

如果 JSCAN = 0，则只转换其中一个所选通道，然后通道选择移至下一个通道。如果 JSCAN=0，则对 JCHG 进行写入操作会将通道选择复位为最小所选通道。

必须始终至少为注入组选择一个通道。如果写操作导致所有 JCHG 位均变为零，则忽略写操作。

29.8.6 DFSDM 滤波器 x 控制寄存器 (DFSDM_FLTxFCR)

DFSDM filter x control register

偏移地址：0x114 + 0x80 * x（x = 0 到 1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FORD[2:0]			Res.	Res.	Res.	FOSR[9:0]									
rw	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOSR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 **FORD[2:0]**：Sinc 滤波器阶数

0：FastSinc 滤波器类型
1：$Sinc^1$ 滤波器类型
2：$Sinc^2$ 滤波器类型
3：$Sinc^3$ 滤波器类型
4：$Sinc^4$ 滤波器类型
5：$Sinc^5$ 滤波器类型
6-7：保留

$Sinc^x$ 滤波器类型传递函数：

$$\text{高阻态} = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}}\right)^x$$

FastSinc 滤波器类型传递函数：

$$\text{高阻态} = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}}\right)^2 \cdot (1 + z^{-(2 \cdot FOSR)})$$

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 28:26 保留，必须保持复位值。

位 25:16 **FOSR[9:0]**：Sinc 滤波器过采样率（抽取率）(Sinc filter oversampling ratio (decimation rate))

0-1023：定义 Sinc 类型滤波器的长度，长度范围为 1 - 1024 (F_{OSR} = FOSR[9:0] +1)。该数字还表示滤波器输出数据速率的抽取率。

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位

注意：如果 FOSR = 0，则滤波器无影响（滤波器旁路）。

位 15:8 保留，必须保持复位值。

位 7:0 **IOSR[7:0]**：积分器过采样率（平均长度）(Integrator oversampling ratio (averaging length))

0-255：积分器的长度，长度范围为 1 - 256 (IOSR + 1)。定义一个积分器输出数据采样是由多少个 Sinc 滤波器采样而合成的。积分器的输出数据速率将被减去该数值（额外数据抽取率）。

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位

注意：如果 IOSR = 0，则积分器无影响（积分器旁路）。

29.8.7 注入组的 DFSDM 滤波器 x 数据寄存器 (DFSDM_FLTxJDATAR)

DFSDM filter x data register for injected group

偏移地址：0x118 + 0x80 * x（x = 0 到 1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JDATA[23:8]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JDATA[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	JDATACH[1:0]	
r	r	r	r	r	r	r	r							r	r

位 31:8 **JDATA[23:0]**：注入组转换数据 (Injected group conversion data)

每次转换完注入组中的一个通道时，得到的数据都会存储在该字段中。JEOCF =1 时数据有效。读取该寄存器会将相应的 JEOCF 清零。

位 7:2 保留，必须保持复位值。

位 1:0 **JDATACH[1:0]**：最近转换的注入通道 (Injected channel most recently converted)
每次转换完注入组中的一个通道时，都会更新 JDATACH[1:0] 以指示转换了哪一个通道。因此，JDATA[23:0] 保持的数据对应于 JDATACH[1:0] 所指示的通道。

注： *可以使用 DMA 从该寄存器中读取数据。可以使用半字访问只读取转换数据的最高有效位 (MSB)。*
读取该寄存器还会将 DFSDM_FLTxISR 的 JEOCF 清零。因此，如果为从该寄存器中读取数据而激活了 DMA，则固件不能读取该寄存器。

29.8.8 常规通道的 DFSDM 滤波器 x 数据寄存器 (DFSDM_FLTxRDATAR)

DFSDM filter x data register for the regular channel

偏移地址：0x11C + 0x80 * x（x = 0 到 1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDATA[23:8]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATA[7:0]								Res.	Res.	Res.	RPEND	Res.	Res.	RDATACH[1:0]	
r	r	r	r	r	r	r	r				r			r	r

位 31:8 **RDATA[23:0]**：常规通道转换数据 (Regular channel conversion data)
每次完成常规转换时，相应的数据都会存储在该寄存器中。当 REOCF =1 时，数据有效。读取该寄存器会将相应的 REOCF 清零。

位 7:5 保留，必须保持复位值。

位 4 **RPEND**：常规通道待处理数据 (Regular channel pending data)
因在转换期间发生注入通道触发而导致 RDATA[23:0] 中的常规数据被延迟处理

位 3:2 保留，必须保持复位值。

位 1:0 **RDATACH[1:0]**：最近转换的常规通道 (Regular channel most recently converted)
每次完成常规转换时，都会更新 RDATACH[1:0] 以指示转换了哪一个通道（因为在常规转换期间，可更新 DFSDM_FLTxCR1 寄存器中的常规通道选择 RCH[1:0]）。因此，RDATA[23:0] 保持的数据对应于 RDATACH[1:0] 所指示的通道。

注： *可以使用半字访问只读取转换数据的最高有效位 (MSB)。*
读取该寄存器还会将 DFSDM_FLTxISR 的 REOCF 清零。

29.8.9 DFSDM 滤波器 x 模拟看门狗阈值上限寄存器 (DFSDM_FLTxAWHTR)

DFSDM filter x analog watchdog high threshold register

偏移地址：0x120 + 0x80 * x（x = 0 到 1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AWHT[23:8]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWHT[7:0]								Res.	Res.	Res.	Res.	BKAWH[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw

位 31:8 **AWHT[23:0]**：模拟看门狗上限阈值 (Analog watchdog high threshold)

这些位由软件写入，用于定义模拟看门狗的阈值上限。

注意：对于通道收发器监视器 (AWFSEL = 1)，高 16 位 (AWHT[23:8]) 会定义 16 位阈值，该阈值与模拟看门狗滤波器输出进行比较（因为来自模拟看门狗滤波器的数据的分辨率最高为 16 位）。在这种情况下，不对位 AWHT[7:0] 进行比较。

位 7:4 保留，必须保持复位值。

位 3:0 **BKAWH[3:0]**：模拟看门狗阈值上限事件的断路信号分配 (Break signal assignment to analog watchdog high threshold event)

BKAWH[i] = 0：断路 i 信号不分配到模拟看门狗阈值上限事件

BKAWH[i] = 1：断路 i 信号分配到模拟看门狗阈值上限事件

29.8.10 DFSDM 滤波器 x 模拟看门狗阈值下限寄存器 (DFSDM_FLTxAWLTR)

DFSDM filter x analog watchdog low threshold register

偏移地址：0x124 + 0x80 * x（x = 0 到 1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AWLT[23:8]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWLT[7:0]								Res.	Res.	Res.	Res.	BKAWL[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw

位 31:8 **AWLT[23:0]**：模拟看门狗下限阈值 (Analog watchdog low threshold)

这些位由软件写入，用于定义模拟看门狗的阈值下限。

注意：对于通道收发器监视 (AWFSEL = 1)，仅高 16 位 (AWLT[23:8]) 定义 16 位阈值，该阈值与模拟看门狗滤波器输出进行比较（因为来自模拟看门狗滤波器的数据的分辨率最高为 16 位）。在这种情况下，不对位 AWLT[7:0] 进行比较。

位 7:4 保留，必须保持复位值。

位 3:0 **BKAWL[3:0]**：模拟看门狗阈值下限事件的断路信号分配 (Break signal assignment to analog watchdog low threshold event)

BKAWL[i] = 0：断路 i 信号不分配到模拟看门狗阈值下限事件
BKAWL[i] = 1：断路 i 信号分配到模拟看门狗阈值下限事件

29.8.11 DFSDM 滤波器 x 模拟看门狗状态寄存器 (DFSDM_FLTxAWSR)

DFSDM filter x analog watchdog status register

偏移地址：0x128 + 0x80 * x（x = 0 到 1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	AWHTF[3:0]				Res.	Res.	Res.	Res.	AWLTF[3:0]			
				r	r	r	r					r	r	r	r

位 31:12 保留，必须保持复位值。

位 11:8 **AWHTF[3:0]**：模拟看门狗阈值上限标志 (Analog watchdog high threshold flag)
AWHTF[y] = 1 指示通道 y 上存在阈值上限错误。该位由硬件置 1。该位可由软件使用 DFSDM_FLTxAWCFR 寄存器中的相应 CLRAWHTF[y] 位清零。

位 7:4 保留，必须保持复位值。

位 3:0 **AWLTF[3:0]**：模拟看门狗阈值下限标志 (Analog watchdog low threshold flag)
AWLTF[y] = 1 指示通道 y 上存在阈值下限错误。该位由硬件置 1。该位可由软件使用 DFSDM_FLTxAWCFR 寄存器中的相应 CLRAWLTF[y] 位清零。

注： *DFEN = 0 时，DFSDM_FLTxAWSR 的所有位都会自动复位。*

29.8.12 DFSDM 滤波器 x 模拟看门狗清零标志寄存器 (DFSDM_FLTxAWCFR)

DFSDM filter x analog watchdog clear flag register

偏移地址：0x12C + 0x80 * x（x = 0 到 1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CLRAWHTF[3:0]				Res.	Res.	Res.	Res.	CLRAWLTF[3:0]			
				rc_w1	rc_w1	rc_w1	rc_w1					rc_w1	rc_w1	rc_w1	rc_w1

位 31:12 保留，必须保持复位值。

位 11:8 **CLRAWHTF[3:0]**：清零模拟看门狗阈值上限标志 (Clear the analog watchdog high threshold flag)
CLRAWHTF[y]=0：写入“0”无影响
CLRAWHTF[y]=1：向位置 y 写入“1”会将 DFSDM_FLTxAWSR 寄存器中的相应 AWHTF[y] 位清零。

位 7:4 保留，必须保持复位值。

位 3:0 **CLRAWLTF[3:0]**：将模拟看门狗阈值下限标志清零 (Clear the analog watchdog low threshold flag)
CLRAWLTF[y]=0：写入“0”无影响
CLRAWLTF[y]=1：向位置 y 写入“1”会将 DFSDM_FLTxAWSR 寄存器中的相应 AWLTF[y] 位清零。

29.8.13 DFSDM 滤波器 x 极值检测器最大值寄存器 (DFSDM_FLTxEXMAX)

DFSDM filter x extremes detector maximum register

偏移地址：0x130 + 0x80 * x（x = 0 到 1）

复位值：0x8000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXMAX[23:8]															
rs_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXMAX[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	EXMAXCH[1:0]	
rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r							r	r

位 31:8 **EXMAX[23:0]**：极值检测器最大值 (Extremes detector maximum value)
这些位由硬件置 1，用于指示 DFSDM_FLTx 所转换的最大值。通过读取该寄存器将 EXMAX[23:0] 位复位为值 (0x800000)。

位 7:2 保留，必须保持复位值。

位 1:0 **EXMAXCH[1:0]**：极值检测器最大值数据通道 (Extremes detector maximum data channel)
这些位包含的信息涉及哪一个通道的数据被存储到 EXMAX[23:0]。可通过读取该寄存器将这些位清零。

29.8.14 DFSDM 滤波器 x 极值检测器最小值寄存器 (DFSDM_FLTxEXMIN)

DFSDM filter x extremes detector minimum register

偏移地址：0x134 + 0x80 * x（x = 0 到 1）

复位值：0x7FFF FF00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXMIN[23:8]															
rc_r	rs_r	rs_r	rs_r	rs_r	rs_r	rs_r	rs_r	rs_r	rs_r	rs_r	rs_r	rs_r	rs_r	rs_r	rs_r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXMIN[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	EXMINCH[1:0]	
rs_r	rs_r	rs_r	rs_r	rs_r	rs_r	rs_r	rs_r							r	r

位 31:8 **EXMIN[23:0]**：极值检测器最小值 (Extremes detector minimum value)
这些位由硬件置 1，用于指示 DFSDM_FLTx 所转换的最小值。通过读取该寄存器将 EXMIN[23:0] 位复位为值 (0x7FFFFF)。

位 7:2 保留，必须保持复位值。

位 1:0 **EXMINCH[1:0]**：极值检测器最小值数据通道 (Extremes detector minimum data channel)
这些位包含的信息涉及哪一个通道的数据被存储到 EXMIN[23:0]。可通过读取该寄存器将这些位清零。

29.8.15 DFSDM 滤波器 x 转换定时器寄存器 (DFSDM_FLTxCNVTIMR)

DFSDM filter x conversion timer register

偏移地址：0x138 + 0x80 * x（x = 0 到 1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNVCNT[27:12]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNVCNT[11:0]												Res.	Res.	Res.	Res.
r	r	r	r	r	r	r	r	r	r	r	r				

位 31:4 **CNVCNT[27:0]**：28 位定时器计数转换时间 (28-bit timer counting conversion time) t = CNVCNT[27:0] / $f_{DFSDMCLK}$
定时器的输入时钟来自 DFSDM 时钟（系统时钟 $f_{DFSDMCLK}$）。转换时间测量始于每次转换开始，并止于每次转换结束（即第一次和最后一次串行采样之间的间隔）。只有在滤波器旁路 (FOSR[9:0] = 0) 的情况下，转换时间测量才会停止，且 CNVCNT[27:0] = 0。时间计时如下：
如果 FAST=0（或者 FAST=1 连续模式下的第一次转换），则：
$t = [F_{OSR} * (I_{OSR}-1 + F_{ORD}) + F_{ORD}] / f_{CKIN}$..... 适用于 $Sinc^x$ 滤波器
$t = [F_{OSR} * (I_{OSR}-1 + 4) + 2] / f_{CKIN}$..... 适用于 FastSinc 滤波器
如果在连续模式下 FAST=1（第一次转换除外），则：
$t = [F_{OSR} * I_{OSR}] / f_{CKIN}$
如果 F_{OSR} = FOSR[9:0] + 1 = 1（滤波器旁路，仅积分器有效），则：
CNVCNT = 0（时间计时停止，转换时间：$t = I_{OSR} / f_{CKIN}$）
其中，f_{CKIN} 为（给定通道 CKINy 引脚上的）通道输入时钟频率；在并行数据输入（来自内部 ADC 或来自 CPU/DMA 写操作）的情况下，其表示输入数据速率
注意：当转换被中断时（例如通过禁止/使能所选通道），定时器也将该中断时间计算在内。

位 3:0 保留，必须保持复位值。

29.9 DFSDM 版本寄存器

29.9.1 DFSDM 硬件配置寄存器 (DFSDM_HWCFGR)

DFSDM hardware configuration register

此寄存器指定 DFSDM 外设的硬件配置。

偏移地址：0x7F0

复位值：0x0000 0204

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NBF[7:0]								NBT[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:8 **NBF[7:0]**：实现的滤波器数 (Number of implemented filters)
定义在 DFSDM 外设中实现了多少个滤波器。

位 7:0 **NBT[7:0]**：实现的收发器数 (Number of implemented transceivers)
定义在 DFSDM 外设中实现了多少个收发器（输入通道）。

29.9.2 DFSDM 版本寄存器 (DFSDM_VERR)

DFSDM version register

此寄存器指定 DFSDM 外设的版本。

偏移地址：0x7F4

复位值：0x0000 0021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：DFSDM 外设的主版本 (Major revision of the DFSDM peripheral)
这些位返回 DFSDM 的主版本（范围 0..15）。

位 3:0 **MINREV[3:0]**：DFSDM 外设的次版本 (Minor revision of the DFSDM peripheral)
这些位返回 DFSDM 的次版本（范围 0..15）。

29.9.3 DFSDM 标识寄存器 (DFSDM_IPIDR)

DFSDM identification register

此寄存器指定 DFSDM 外设的标识。

偏移地址：0x7F8

复位值：0x0011 0031

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：外设标识符 (Peripheral identifier)
位 [31:0]：这些位返回 DFSDM 标识符 ID[31:0] = 0x0011 0031

29.9.4 DFSDM 大小标识寄存器 (DFSDM_SIDR)

DFSDM size identification register

此寄存器指定分配给 DFSDM 寄存器的大小。

偏移地址：0x7FC

复位值：0xA3C5 DD02

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：DFSDM 外设的大小标识 (Size identification of DFSDM peripheral)
位 [31:8]：固定代码 = 0xA3C5DD
位 [7:0]：这些位返回分配给 DFSDM 寄存器的存储区域的大小。
0x02：DFSDM 外设分配的 2 KB（固定值）。

29.9.5 DFSDM 寄存器映射

下表对 DFSDM 寄存器进行了汇总。

表 236. DFSDM 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**DFSDM_CH0CFGR1**	DFSDMEN	CKOUTSRC	Res.	Res.	Res.	Res.	Res.	Res.	CKOUTDIV[7:0]								DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL [1:0]		SITP[1:0]	
	reset value	0	0							0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0		0	0	0	0
0x04	**DFSDM_CH0CFGR2**	OFFSET[23:0]																								DTRBS[4:0]					Res.	Res.	Res.
	reset value	0																								0							
0x08	**DFSDM_CH0AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD [1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0
0x0C	**DFSDM_CH0WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**DFSDM_CH0DATINR**	INDAT1[15:0]																INDAT0[15:0]															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**DFSDM_CH0DLYR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLSSKP[5:0]					
	reset value																											0	0	0	0	0	0
0x18 - 0x1C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x20	**DFSDM_CH1CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL [1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0
0x24	**DFSDM_CH1CFGR2**	OFFSET[23:0]																								DTRBS[4:0]					Res.	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x28	**DFSDM_CH1AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0
0x2C	**DFSDM_CH1WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	**DFSDM_CH1DATINR**	INDAT1[15:0]																INDAT0[15:0]															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	**DFSDM_CH1DLYR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLSSKP[5:0]					
	reset value																											0	0	0	0	0	0

表 236. DFSDM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x38 - 0x3C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x40	**DFSDM_ CH2CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0
0x44	**DFSDM_ CH2CFGR2**	OFFSET[23:0]																								DTRBS[4:0]					Res.	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x48	**DFSDM_ CH2AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x4C	**DFSDM_ CH2WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	**DFSDM_ CH2DATINR**	INDAT1[15:0]																INDAT0[15:0]															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54	**DFSDM_ CH2DLYR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLSSKP[5:0]					
	reset value																											0	0	0	0	0	0
0x58 - 0x5C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x60	**DFSDM_ CH3CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0
0x64	**DFSDM_ CH3CFGR2**	OFFSET[23:0]																								DTRBS[4:0]					Res.	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x68	**DFSDM_ CH3AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x6C	**DFSDM_ CH3WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x70	**DFSDM_ CH3DATINR**	INDAT1[15:0]																INDAT0[15:0]															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x74	**DFSDM_ CH3DLYR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLSSKP[5:0]					
	reset value																											0	0	0	0	0	0

表 236. DFSDM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x78 - 0xFC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x100	**DFSDM_ FLT0CR1**	Res.	AWFSEL	FAST	Res.	Res.	Res.	RCH[1:0]		Res.	Res.	RDMAEN	Res.	RSYNC	RCONT	RSW START	Res.	Res.	JEXTEN[1:0]		JEXTSEL[4:0]					Res.	Res.	JDMAEN	JSCAN	JSYNC	Res.	JSW START	DFEN
	reset value		0	0				0	0			0		0	0	0			0	0	0	0	0	0	0			0	0	0		0	0
0x104	**DFSDM_ FLT0CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDCH[3:0]				Res.	Res.	Res.	Res.	EXCH[3:0]				Res.	CKABIE	SCDIE	AWDIE	ROVRIE	JOVRIE	REOCIE	JEOCIE
	reset value													0	0	0	0					0	0	0	0		0	0	0	0	0	0	0
0x108	**DFSDM_ FLT0ISR**	Res.	Res.	Res.	Res.	SCDF[3:0]				Res.	Res.	Res.	Res.	CKABF[3:0]				Res.	RCIP	JCIP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDF	ROVRF	JOVRF	REOCF	JEOCF
	reset value					0	0	0	0					1	1	1	1		0	0									0	0	0	0	0
0x10C	**DFSDM_ FLT0ICR**	Res.	Res.	Res.	Res.	CLRSCDF [3:0]				Res.	Res.	Res.	Res.	CLRCKABF [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR ROVRF	CLR JOVRF	Res.	Res.
	reset value					0	0	0	0					0	0	0	0													0	0		
0x110	**DFSDM_ FLT0JCHGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JCHG[3:0]			
	reset value																													0	0	0	1
0x114	**DFSDM_ FLT0FCR**	FORD[2:0]			Res.	Res.	Res.	FOSR[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOSR[7:0]								
	reset value	0	0	0				0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x118	**DFSDM_ FLT0JDATAR**	JDATA[23:0]																								Res.	Res.	Res.	Res.	Res.	Res.	JDATACH [1:0]	
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							0	0
0x11C	**DFSDM_ FLT0RDATAR**	RDATA[23:0]																								Res.	Res.	Res.	RPEND	Res.	Res.	RDATACH[1:0]	
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0			0	0
0x120	**DFSDM_ FLT0AWHTR**	AWHT[23:0]																								Res.	Res.	Res.	Res.	BKAWH[3:0]			
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0
0x124	**DFSDM_ FLT0AWLTR**	AWLT[23:0]																								Res.	Res.	Res.	Res.	BKAWL[3:0]			
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0
0x128	**DFSDM_ FLT0AWSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWHTF[3:0]				Res.	Res.	Res.	Res.	AWLTF[3:0]			
	reset value																					0	0	0	0					0	0	0	0
0x12C	**DFSDM_ FLT0AWCFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLRAWHTF [3:0]				Res.	Res.	Res.	Res.	CLRAWLTF [3:0]			
	reset value																					0	0	0	0					0	0	0	0

表 236. DFSDM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x130	**DFSDM_FLT0EXMAX**	EXMAX[23:0]																								Res.	Res.	Res.	Res.	Res.	Res.	EXMAXCH[1:0]	
	reset value	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							0	0
0x134	**DFSDM_FLT0EXMIN**	EXMIN[23:0]																								Res.	Res.	Res.	Res.	Res.	Res.	EXMINCH[1:0]	
	reset value	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1							0	0
0x138	**DFSDM_FLT0CNVTIMR**	CNVCNT[27:0]																												Res.	Res.	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x13C - 0x17C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x180	**DFSDM_FLT1CR1**	Res.	AWFSEL	FAST	Res.	Res.	Res.	RCH[1:0]		Res.	Res.	RDMAEN	Res.	RSYNC	RCONT	RSW START	Res.	Res.	JEXTEN[1:0]		JEXTSEL[4:0]					Res.	Res.	JDMAEN	JSCAN	JSYNC	Res.	JSW START	DFEN
	reset value		0	0				0	0			0		0	0	0			0	0	0	0	0	0	0			0	0	0		0	0
0x184	**DFSDM_FLT1CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDCH[3:0]				Res.	Res.	Res.	Res.	EXCH[3:0]				Res.	Res.	Res.	AWDIE	ROVRIE	JOVRIE	REOCIE	JEOCIE
	reset value													0	0	0	0					0	0	0	0				0	0	0	0	0
0x188	**DFSDM_FLT1ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RCIP	JCIP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDF	ROVRF	JOVRF	REOCF	JEOCF
	reset value																		0	0									0	0	0	0	0
0x18C	**DFSDM_FLT1ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR ROVRF	CLR JOVRF	Res.	Res.
	reset value																													0	0		
0x190	**DFSDM_FLT1JCHGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JCHG[3:0]			
	reset value																													0	0	0	1
0x194	**DFSDM_FLT1FCR**	FORD[2:0]			Res.	Res.	Res.	FOSR[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOSR[7:0]							
	reset value	0	0	0				0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x198	**DFSDM_FLT1JDATAR**	JDATA[23:0]																								Res.	Res.	Res.	Res.	Res.	Res.	JDATACH[1:0]	
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							0	0

表 236. DFSDM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x19C	**DFSDM_FLT1RDATAR**	RDATA[23:0]																								Res.	Res.	Res.	RPEND	Res.	Res.	RDATACH[1:0]	
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0			0	0
0x1A0	**DFSDM_FLT1AWHTR**	AWHT[23:0]																								Res.	Res.	Res.	Res.	BKAWH[3:0]			
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0
0x1A4	**DFSDM_FLT1AWLTR**	AWLT[23:0]																								Res.	Res.	Res.	Res.	BKAWL[3:0]			
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0
0x1A8	**DFSDM_FLT1AWSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWHTF[3:0]				Res.	Res.	Res.	Res.	AWLTF[3:0]			
	reset value																					0	0	0	0					0	0	0	0
0x1AC	**DFSDM_FLT1AWCFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLRAWHTF[3:0]				Res.	Res.	Res.	Res.	CLRAWLTF[3:0]			
	reset value																					0	0	0	0					0	0	0	0
0x1B0	**DFSDM_FLT1EXMAX**	EXMAX[23:0]																								Res.	Res.	Res.	Res.	Res.	Res.	EXMAXCH[1:0]	
	reset value	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							0	0
0x1B4	**DFSDM_FLT1EXMIN**	EXMIN[23:0]																								Res.	Res.	Res.	Res.	Res.	Res.	EXMINCH[1:0]	
	reset value	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1							0	0
0x1B8	**DFSDM_FLT1CNVTIMR**	CNVCNT[27:0]																												Res.	Res.	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x1BC - 0x2FC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x7F0	**DFSDM_HWCFGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBF[7:0]								NBT[7:0]							
	reset value																	2								4							
0x7F4	**DFSDM_VERR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	reset value																									0x2				0x1			
0x7F8	**DFSDM_IPIDR**	ID[31:0]																															
	reset value	0x00110031																															
0x7FC	**DFSDM_SIDR**	SID[31:0]																															
	reset value	0xA3C5DD02																															

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

30 数字摄像头接口像素流水线 (DCMIPP)

30.1 简介

DCMIPP 是高分辨率摄像头子系统的像素流水线部分：它通过并行接口获取像素，并在一些处理（例如抽取、裁剪）之后将它们转储到内存。

DCMIPP 支持多种类型的外部传感器，包括但不限于：

- 简单传感器（无内部 ISP），输出原始拜尔像素
- 智能传感器（具有内部 ISP），通常输出 RGB 或 YUV 像素
- 支持内部压缩功能的智能传感器，输出位流（例如 JPEG 或 H264）

DCMIPP 输入接口集成了一个并行接口（120 MHz 下最高 16 位，支持内部/外部同步）。

DCMIPP 的第一个通用部分专门从并行接口中选择输入。数据在发送到内存进行进一步处理或供显示之前，会先进入专用流水线。

表 237. 可用流水线

ID	功能	说明
Pipe0	转储流水线	用于原样转储接收到的数据，不进行任何处理（例如，不提取像素、不格式化），但有一个例外：它将 10/12/14 位的原始拜耳分量映射为未打包的 16 位内存字，以便于下游应用的处理。 在此流水线内可以进行 2D 裁剪操作和基本抽取。

图 264 显示了 DCMIPP 的主要功能，即集成的并行接口、应用后处理流水线与其裁剪、降采样和像素格式化。

图 264. DCMIPP 概述

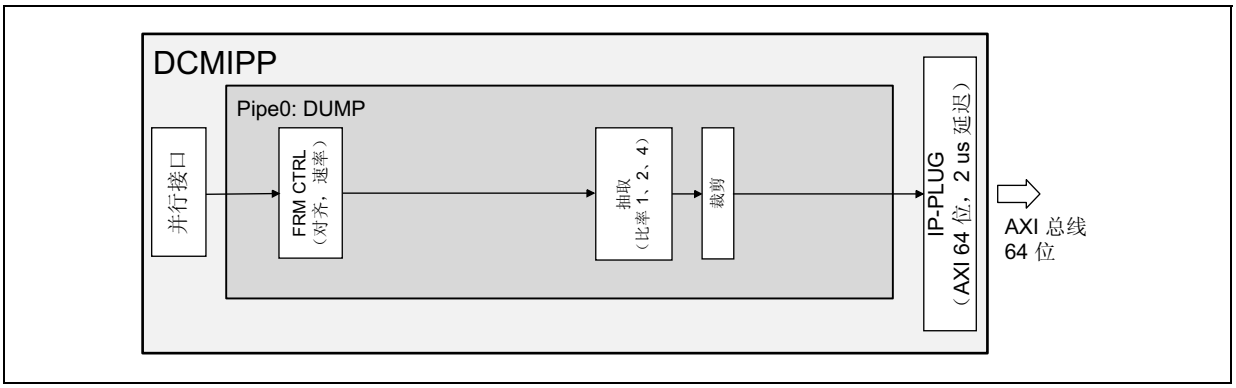


表 238. 词汇表

项目	定义
A	Alpha 分量，用于定义透明度，例如不透明 = 0xFF。ARGB。
BPC	每个分量的位数（例如 RGB565 的红色分量有 5 位，所以是 5 bpc）
BPP	每像素位数（例如，RGB565 每像素有 16 位，所以是 16 bpp）
DCMIPP_	数字摄像头接口 - 像素流水线

表 238. 词汇表（续）

项目	定义
交错	交错视频：首先传输带有奇数行的场，最后一个奇数行传输完之后，传输带有偶数行的场。这两个字段因此是连续但互斥的。
NPU	神经网络处理引擎（也称为 NN，神经网络）。
平面	定义像素缓冲区被划分为多少个子缓冲区：1（共面）。
YUV444	像素格式，YUV 色彩参考，每个像素给定全部三个分量（Y、U、V）。
YUV422	像素格式，YUV 色彩参考，色度分量（U、V）在 X 方向上进行 1/2 子采样
YUV420	像素格式，YUV 色彩参考，色度分量（U、V）在 X 和 Y 方向上进行 1/2 子采样。

30.2 DCMIPP 主要特性

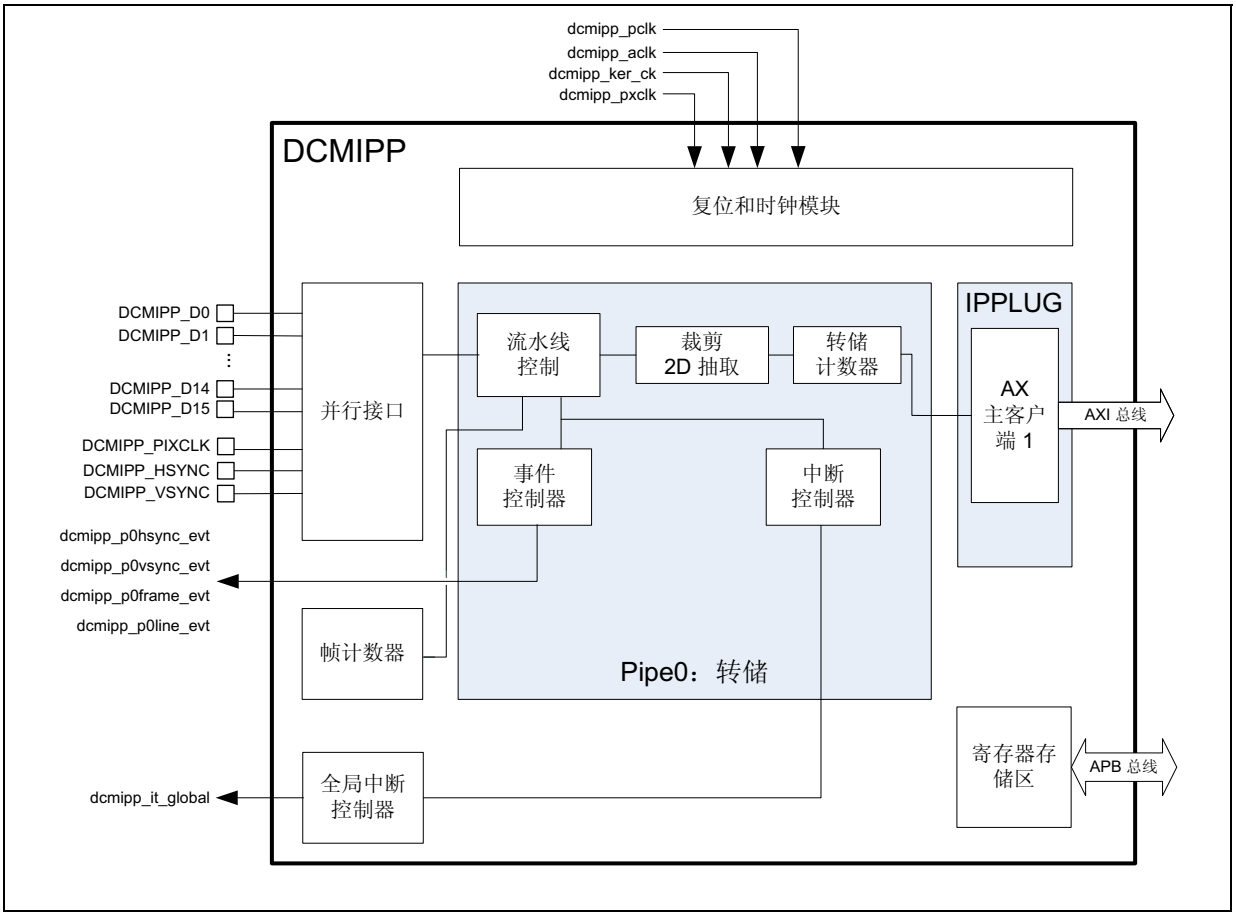
- 并行输入接口：
  - 像素率：最高 16 位并行，1 亿像素/秒（通常为 1080p30，抽取后处理流水线上最大 2048x2048）。
  - 像素格式：RGB565、RGB888、YUV422、原始拜耳/Mono 8/10/12/14 和 ByteStream (JPEG)
- 流水线最大分辨率：
  - 抽取前为 500 万像素（通常为 2560 x 2048 或 2688 x 1944）的尺寸
  - 最大流水线分辨率：4094 x 4094
- 最大帧率 (fps)：典型示例定义为 30 fps，可以通过降低图像分辨率来达到更高的帧率（例如 120 fps），以保证最大像素率的限制。
- 流选择和帧控制：
  - 连续模式与快照模式下的捕获模式
- Pipe0（转储流水线）上的抽取
  - 纯子采样，X 方向上上的比率为 1:1、1:2、1:4，Y 方向上的比率为 1:1、1:2
- 并行应用的并行流水线：
  - Pipe0（用于数据转储），无需处理即可直接转储
- 单个 Pipe0（用于数据转储），从摄像头直接转储到内存（只需少量操作，如裁剪和基本抽取）。
- 输出像素格式：
  - Pipe0：任何数据按原样，Y/Rb：8/10/12/14 统计数据，位流
- AXI 主设备：
  - 输出 FIFO 将像素/数据从流水线排出到内存（与内存延迟相关）
  - 独特的 AXI 主设备接口为每个流水线实现一个客户端
  - 每个流水线的双缓冲模式功能
- 上溢检测：
  - 在像素拥堵影响帧的情况下通知（如果解决了拥堵原因，则对后续帧无影响）

30.3 DCMIPP 功能描述

30.3.1 DCMIPP 框图

DCMIPP 的框图如*图 265* 所示。

图 265. DCMIPP 框图



30.3.2 DCMIPP 引脚和内部信号

表 239. DCMIPP 输入/输出引脚

引脚名称	信号类型	说明
DCMIPP_Dn （n = 0 到 15）	输入	来自摄像头传感器的并行数据总线的第 n 位
DCMIPP_PIXCLK	输入	由主设备（并行摄像头传感器模块）发送的像素时钟
DCMIPP_VSYNC	输入	来自摄像头传感器的 VSYNC 信号（垂直同步）
DCMIPP_HSYNC	输入	来自摄像头传感器的 HSYNC 信号（水平同步）

表 240. DCMIPP 输入/输出引脚

内部信号名称	信号类型	说明
dcmipp_pclk	输入	DCMIPP APB 时钟（APB 总线）
dcmipp_aclk	输入	DCMIPP AXI 时钟（AXI 总线）
dcmipp_ker_ck	输入	DCMIPP 内核时钟（ISB 字节标头）
dcmipp_it_global	输出	DCMIPP 中断（有关中断源的完整列表，请参见*表 257*）
dcmipp_p0hsync_evt	输出	Pipe0 Hsync 事件
dcmipp_p0vsync_evt		Pipe0 Vsync 事件
dcmipp_p0frame_evt		Pipe0 帧结束事件
dcmipp_p0line_evt		Pipe0 行事件

30.3.3 DCMIPP 复位和时钟

表 241. DCMIPP 时钟

域	时钟	最大频率	备注
并行	dcmipp_pxclk	120 MHz	在 16 位模式下高达 2x 120 MB/s
流水线	dcmipp_ker_ck	266 MHz	高达 2.66 亿像素/秒（不含消隐）
AXI 总线	dcmipp_aclk	350 MHz	SoC AXI 频率
APB 总线	dcmipp_pclk	133 MHz	SoC APB 频率

表 242. DCMIPP 复位

域	复位类型	备注
并行 + 流水线	异步	在 dcmipp_pxclk/dcmipp_ker_ck 上内部重新同步
AXI 总线	同步	直接复位 dcmipp_aclk 域
APB 总线	同步	直接复位 dcmipp_pclk 域

时钟和像素率限制

本段列出了 DCMIPP 的限制/瓶颈：

- 目标是维持平均像素速率 (PixRateAvg)：
  - 一个帧率为 15 fps 的 500 万像素传感器需要 7500 万像素/秒
- 垂直和水平消隐：
  - 整个系统（传感器、接口、流水线、AXI）内置有限容量的 FIFO。传感器在消隐期间插入的空闲时间会影响所有下游元素，包括接口、流水线和 AXI。必须使用高于平均值的像素速率 (PixRatePeak)。由于消隐导致的空闲时间可以达到帧持续时间的 33%，有时更多（请查阅传感器规范）。
  - PixRatePeak = 1.33 * PixRateAvg
- 并行接口时钟 (dcmipp_pxclk)：
  - 在并行接口上，像素是在一、二或三个周期内按顺序接收的。实现的最大时钟 (120 MHz) 限制了峰值像素率，具体取决于每像素周期的比率。
  - PixRatePeak ≤ dcmipp_pxclk / (1、2 或 3) = 120 / (1、2 或 3)
  - 时钟频率：
    - dcmipp_pxclk ≤ dcmipp_ker_ck（由于它们之间的重新同步接口）
    - dcmipp_pxclk ≤ dcmipp_aclk
  - 后果：请参见*表 243*。

表 243. 并行接口最大分辨率 (80 MHz)

像素		接口上的传输			分辨率 (30 fps)，包括消隐	
格式	宽度 (bpp)	宽度	周期/像素	像素率	百万像素/帧	示例
单色/原始	8	8	1	80	2.15	1080 p
	14	14	1	80	2.15	1080 p
RGB565	16	8	2	40	1.08	720 p
		16	1	80	2.15	1080 p
RGB888	24	8	3	27	0.72	qHD
		12	2	40	1.08	720 p
YUV422	16	16	1	80	2.15	1080 p
		16	2	40	1.08	720 p
YUV444	24	12	2	40	1.08	720 p
		16	2	40	1.08	720 p

  - 时钟设置：dcmipp_pxclk ≥ PixRatePeak * 比率(1,2,3)
- AXI 时钟：
  - 必须将像素的时钟从处理重新同步到 AXI 域，从而得出以下结果：
    - dcmipp_ker_ck ≤ dcmipp_aclk

30.3.4 DCMIPP 最大分辨率

与 DCMIPP 支持的最大分辨率相关的限制包括：

- 跨流水线的最大宽度和高度：4094 x 4094 像素
  - 寄存器和计数器限制为 12 位
- 最大性能：2.66 亿像素/秒
  - 可以使用更大的像素率，但需要更低的内存延迟
  - 每个流水线的最大像素速率必须低于像素时钟
- 最大行跨距：32767 字节
  - 寄存器和计数器，处理 4094 像素 x 32 位 x 2（预留给交错）

30.3.5 DCMIPP 对帧结构的最低要求

DCMIPP 对帧架构提出以下最低要求：

消隐阶段

- 转储流水线
  - 两帧之间的最小消隐时间必须相当于 666 个 32 位数据字

数据图像区域

- 垂直图像区域：最少一行
- 水平图像区域：最少两个像素
- 最小总帧大小固定为 36 像素

30.3.6 DCMIPP 像素格式支持的说明

表 244 列出了输入接口（并行接口）和输出 Pipe0 中支持的像素格式。

表 244. 支持的像素格式

索引(1)	像素格式	BPP	并行输入 (clk/pix)(2)	Pipe0 输出（转储）
13(3)	字节流（JPEG、压缩视频）	8	Y、1	Y
14(4)	其他数据	8 到 16(4)	Y、1	Y
5	Raw6	6	(5)	Y
5	Raw7	7	(5)	Y
5	Raw8	8	Y、1	Y
6	Raw10	10	Y、1	至 16 bpp
7	Raw12	12	Y、1	至 16 bpp
8	Raw14	14	Y、1	至 16 bpp
9	Mono8	8	Y、1	Y
10	Mono10	10	Y、1	至 16 bpp
11	Mono12	12	Y、1	至 16 bpp
12	Mono14	14	Y、1	至 16 bpp
2	RGB444	12	(5)	(5)
2	RGB555	15	(5)	(5)

表 244. 支持的像素格式（续）

索引(1)	像素格式	BPP	并行输入 (clk/pix)(2)	Pipe0 输出（转储）
2	RGB565	16	Y、1 和 2	Y
3	RGB666	18	(5)	(5)
4	RGB888/YUV444 (a)	24	Y、2 和 3	Y
1	YUV422-1 (YUYV)	16	Y、1 和 2	Y

1. 索引用于将输入像素格式与 DCMIPP_PRCR 寄存器链接。索引仅对连接并行接口的摄像头有意义
2. 16 bpp 上的像素格式（如 RGB565 或 YUV422-1）可以通过 8 位接口（使用 2 个周期）或 16 位输入接口（使用 1 个周期）接收。24 bpp 上的像素格式（如 RGB888）可以通过 8 位接口（使用三个周期）或 12 位输入接口（使用两个周期）接收。
3. 在字节流模式下，字节连续转储，仅在帧（或 JPEG 流）末尾添加 32 位填充（如果需要）。在 VSYNC 禁止（帧结束或 JPEG 流结束）之前，流内部没有 32 位填充操作。
4. 如果数据格式与此表中不同，用户可以选择在 DCMIPP_PRCR 寄存器的位字段 FORMAT 中选择“其他”配置，使用位字段 EDM 来选择在一个像素时钟内必须捕获多少数据位。可以在单个周期内捕获映射在 16 位并行输入数据上的两个 8 位数据（例如 FORMAT = 其他，EDM = 100，即每个像素时钟的 16 位捕获）。在行末添加 32 位填充，以使整行与 32 位宽对齐。
5. 在并行接口输入上，DCMIPP 支持某些传感器像素格式，通过在 DCMIPP 中选择更宽的像素格式，将传感器线映射到 DCMIPP 接口的 MSB 上。对于输入缺失的引脚，建议将传感器 MSB 引脚复制到缺失的输入 LSB 引脚上。这有助于在扩展像素格式上实现全动态范围。可以将解决方法应用于以下格式（左侧为传感器输出，右侧为 DCMIPP 格式映射）：
   - Raw6：具有 1 周期输入的 Raw8，按 8 bpp 处理
   - Raw7：具有 1 周期输入的 Raw8，按 8 bpp 处理
   - RGB444：具有 1 周期输入的 RGB565，按 16 bpp 处理
   - RGB555：具有 1 周期输入的 RGB565，按 16 bpp 处理
   - RGB666：具有 3 周期输入的 RGB888，按 24 bpp 处理

30.4 DCMIPP 输入和流控制

本节介绍 DCMIPP 的功能元素。对于每个元素，它详细说明了特性、软件配置，并提供了软件配置示例（需要时）。

30.4.1 并行输入接口

并行接口是一种灵活的像素接口，它利用外部提供的时钟以并行方式在 8 到 16 位总线上同步时钟，并增加了垂直和水平同步信号，同步信号可通过特定引脚（外部同步）提供或通过特定数据（如 CCIR601）标记。

接口（所有输入）

- Clk：提供最高 120 MHz 的时钟，内部名为 dcmipp_pxclk
- 数据：16 个 IO 引脚，每个引脚速率最高为 120 Mbps，8 到 16 位，像素可能被串行化
- HSync、VSync，但无使能 (EN)

特性

- 水平和垂直同步：
  - 外部，通过 Hsync 和 Vsync 引脚。
  - 内部，通过 CCIR601 标记。

  水平或垂直消隐期间提供的数据无法被捕获。
- 输入像素格式（请参见*第 30.3.6 节*）
  - 像 JPG 这样的二进制流
  - 单周期数据：
    - 原始拜尔和单色：并行的 8/10/12/14 位
    - YUV422、RGB565：并行 16 位上的 16 bpp
  - 双周期数据：
    - YUV422、RGB565：并行 8 位上的 16 bpp
    - RGB888/YUV444：并行 12 位上的 24 bpp
  - 三周期数据：
    - RGB888/YUV444：并行 8 位上的 24 bpp
- 灵活输入：
  - 可交换 MSB 与 LSB 位以便于 PCB 布线
  - 可交换周期 0 与周期 1（用于长度为 2 个周期的像素）。

硬件同步模式

在这种模式下，使用两个同步信号（DCMIPP_HSYNC 和 DCMIPP_VSYNC）。

根据摄像头模块/模式的不同，可能在水平/垂直同步期间内发送数据。DCMIPP_HSYNC/DCMIPP_VSYNC 充当消隐信号，因为在 DCMIPP_HSYNC/DCMIPP_VSYNC 活动期间接收的所有数据都会被忽略。

为了正确地将图像传输到 RAM 缓冲区，数据传输将与 DCMIPP_HSYNC/DCMIPP_VSYNC 信号同步。选择硬件同步模式并使能捕获（DCMIPP_PxFCTCR 中的 CPTREQ 位置 1）时，数据传输将与 DCMIPP_VSYNC 信号的禁止电平同步（开始下一帧时）。

之后传输便可以连续执行，由 IP-Plug 将连续帧传输到多个连续的缓冲区或一个具有循环特性的缓冲区。

内嵌码同步模式

在此同步模式下，数据流使用嵌入的 32 位代码进行同步，使用不再用于数据的 0x00/0xFF 值。有四种类型的代码，全部采用 0xFF00 00XY 格式。只有 8 位并行数据捕获支持内嵌码同步（在 DCMIPP_PRCR 寄存器中，必须将 EDM[2:0] 位编程为“000”）。对于其他数据宽度，此模式会产生不可预测的结果。

注： *当摄像头模块处于交错模式时，会生成多达八个同步代码，而 DCMIPP 只对一个单一代码作出响应。因此，带有内嵌码同步的交错流中每隔一帧就有一帧因未检测到而被丢弃。*

模式 2

四个内嵌码通知以下事件：

- 帧开始 (FS)
- 帧结束 (FE)
- 行开始 (LS)
- 行结束 (LE)

这些代码采用的格式 0xFF00 00XY 中的 XY 值可编程（请参见*第 30.10.9 节*）。

将 0xFF 编程为“帧结束”意味着所有未使用的编码均被解释为有效帧结束代码。

在此模式下，一旦使能摄像头接口，将在首次出现帧结束 (FE) 代码并且后接帧开始 (FS) 代码之后开始捕获帧。

模式 1

摄像头模式 1 是另一种编码。此模式与 ITU656 兼容。

这些代码通知另一组事件：

- SAV（有效行）——行开始
- EAV（有效行）——行结束
- SAV（消隐）——帧间消隐期内的行开始
- EAV（消隐）——帧间消隐期内的行结束

可通过对代码进行如下编程来支持此模式：

- FS ≤ 0xFF
- FE ≤ 0xFF
- LS ≤ SAV（有效）
- LE ≤ EAV（有效）

此外还针对帧/行开始和帧/行结束代码实现了非屏蔽位功能。这样可以仅使用已编程代码中未被屏蔽的位进行比较。因此，可以选择一个位用于内嵌码的比较，来检测帧/行起始和帧/行结束。这意味着可以多个代码表示帧/行的起始和结束，它们仅在未被屏蔽的位相同即可。

示例：

- FS = 0xA5
- FS 的非屏蔽码 = 0x10

这种情况下，只需要比较数据码的第 4 位来检测是否是 FS 信号。

注： *FEC 序列必须在首帧传输前发送，否则其它代码无法解码，并且在 DCMIPP 使能后首帧可能丢失。在第一个 FE 序列之后，将根据 FS 序列捕获后续帧。*

错误条件

使用内嵌码同步数据模式时，可以检测到错误状态。DCMIPP_CMSR2 中的 PRERRF 标志和 DCMIPP_PRSR 中的 ERRF 标志用于此功能。如果为全局中断将 DCMIPP_CMIER 中的 PRERRIE 位置 1（或为局部中断线将 DCMIPP_PRIER 中的 ERRIE 位置 1），则可以基于错误检测触发中断。

例如，此类错误序列或值会产生错误：

- 检测到 FF 00 00 XY 序列，其 XY 值与为 FS/LS/FE/LE 设置的值不同
- LS = 0xFF 或 LE = 0xFF
- 如果 LS 或 FS 序列跟在 LS 序列之后

软件配置

表 245. DCMIPP_PRCR 位功能

位 ID	功能	备注
ENABLE	使能并行接口	禁用时为低功耗模式。
PCKPOL	定义时钟极性	– 0：下降沿 – 1：上升沿
ESS	定义 VSync 和 HSync 同步是通过旁带（特定硬件 IO）还是通过带内（特定内嵌码）信号提供	– 0：硬件（外部）同步：使用 DCMIPP_HSYNC 和 DCMIPP_VSYNC 引脚。 – 1：内嵌码（内部）同步：使用寄存器 DCMIPP_PRESCR 和 DCMIPP_PRESURy 中的 FSC、FEC、LSC、LEC 代码和屏蔽。
VSPOL，HSPOL	分别定义用于垂直和水平同步的 IO DCMIPP_VSYNC 和 DCMIPP_HSYNC 的极性	– 0：低电平有效 – 1：高电平有效
FORMAT	定义在 IO DCMIPP_Dx 上使用的像素格式	从 0 到 15
EDM	定义每个周期接收的有效位数	8/10/12/14/16 位
	像素的持续时间由每像素位数（由 FORMAT 指定）和每周期位数（由 EDM 指定）的比率决定	1、2 或 3 个周期
SWAPBITS	交换并行接口的 16 位	15:0 变为 0:15
SWAPCYCLES	交换，在像素在两个周期内接收时，第一个周期的数据与第二个周期的数据	当像素在一个或三个周期内接收时，软件不得激活它。

通过 DCMIPP_PRESCR 和 DCMIPP_PRESUR 寄存器配置内嵌码同步代码（见*表 246*）。

表 246. DCMIPP_PRESCR 和 DCMIPP_PRESUR 位功能

位 ID	功能
FSC，FEC	内嵌码同步 ESS = 1 时的帧开始和帧结束同步
FSU，FEU	FSC 和 FEC 代码的屏蔽
LSC，LEC	内嵌码同步 ESS = 1 时的行开始和行结束同步
LSU，LEU	LSC 和 LEC 代码的屏蔽

交换操作序列

位、周期和 RB（红色与蓝色）上 SWAP 操作符的序列由其功能引起：

- 位上的交换（IO 上的 MSB 与 LSB）首先发生，因为它作用于 IO
- 周期上的交换随后发生，因为它会作用于仍处于原始状态时采样的字

30.4.2 帧计数器

帧计数器用于标记目的，可对选定流水线上接收的所有帧进行计数。

计数器为 32 位只读计数器。至少当流水线的 PIPEN 位使能 (PIPEN = 1) 时，它才有效。

它提供一个（几乎）唯一的帧号，循环一个周期时间为 4.5 年（如果是 30 fps）。

当摄像头传感器模块与并行接口连接时，计数器可以在流水线的 FrameStart 事件处递增。

通过在 DCMIPP_CMCR 寄存器中将 CFC 置 1 来清零帧计数器。

软件配置

将帧计数器用于流水线前，必须先对其进行配置。必须在 DCMIPP_CMCR 寄存器中执行软件操作：

- 一旦选定的流水线有效 (PIPEN = 1)，帧计数器即开始运行

注： *可以让帧计数器在流水线上运行，同时让流水线不流过任何像素：这是通过激活流水线使能 (PIPEN = 1) 以运行帧计数器来实现的，但是需要将 CPTREQ 设置为 0 来避免任何像素流。*

30.4.3 帧控制

该模块被复制用于转储流水线。它处理帧控制和流水线的捕获，将数据分组到帧中，并在请求时捕获它们。

定义

有两种帧事件的概念

- 硬件内部使用的事件（由输入接口提供）：
  - FrameStart：当帧在流水线输入处开始时（在帧控制中）
    - 并行接口：Vsync 下降
  - FrameEnd：当帧在流水线输入处结束时（在帧控制中）
    - 并行接口：Vsync 上升（帧结束）
- 通过 DCMIPP 事件和中断暴露给软件的事件：
  - VSYNC
    - 当帧开始时，在流水线输入处观察到（在帧控制中）
    - 它指示当前帧已开始并已采样其影子寄存器，并且软件可以开始配置下一帧
    - 建议使用该中断来触发重新配置 DCMIPP 的软件。
  - 帧结束
    - 当帧结束时，在流水线输出处观察到（当最后一个像素已写入 AXI 时）
    - 它指示当前帧已完成将所有数据写入内存，并且应用程序可以立即开始使用它们
    - 建议使用该中断来触发将内存缓冲区转发给其他应用程序的软件

特性

- 帧定界：允许以所需的帧粒度转储数据。
  - 根据输入的帧起始和帧结束，提取 FrameStart 和 FrameEnd 事件。
    - > FrameStart 立即生成 VSYNC 事件。
    - > FrameEnd 在像素打包器级别不断波动直至流水线末端，并在此处生成 FRAME 事件。
  - 基于在 *FrameStart 和 FrameEnd 的提取算法*中详述的算法。
- 捕获模式：可以捕获帧快照或连续捕获。
- 捕获率：可以捕获所有帧，或者每 2、4、8 帧捕获一帧。仅当捕获模式处于连续模式时才有效。

FrameStart 和 FrameEnd 的提取算法

FrameStart 和 FrameEnd 的提取简单直接。

用例

连续模式通常用于向显示屏发送或对连续帧流进行软件分析。

快照模式通常用于转储高分辨率帧。

帧率模式通常用于低功耗连续分析，其中每八帧只转储和分析一帧。

帧信息通常用于交错视频用例，其中传感器交替发送奇数和偶数场（顶部与底部），并且软件必须能够检索场的类型。根据传感器，可从以下位置检索：

- 该帧中计数的行数的 LSB（指示该数字是奇数还是偶数）

软件配置

帧控制支持两种不同的捕获模式：

- 快照模式：软件捕获单个帧
- 连续模式：软件捕获连续序列。

这两种模式的时序图如*图 266* 所示。

图 266. 快照 (CPTMODE = 1) 和连续 (CPTMODE = 0) 捕获模式

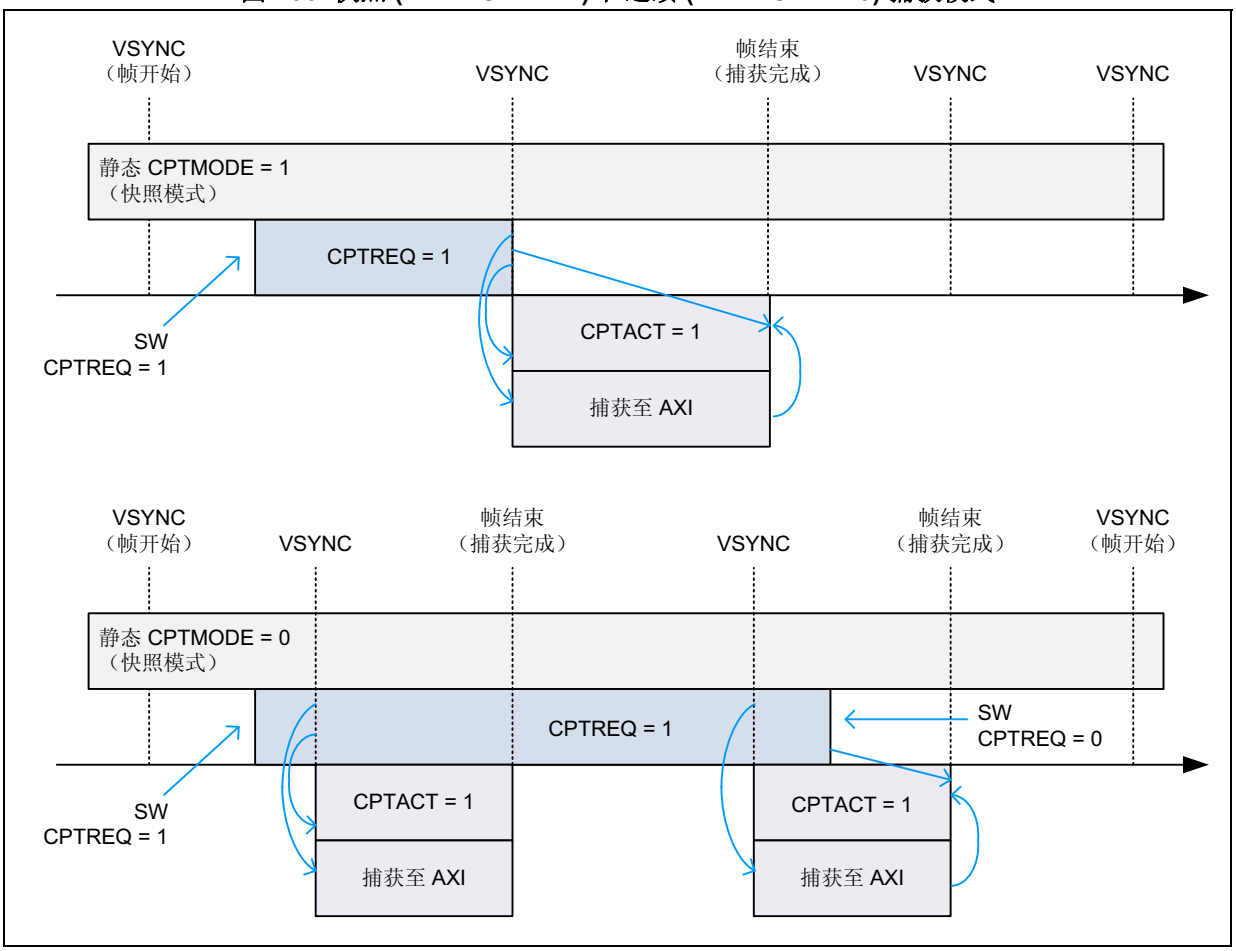


表 247. DCMIPP_PxFCTCR 位功能

位 ID	功能	值
CPTMODE	捕获模式	– 0：连续 – 1：快照
CPTREQ	捕获请求	– 1：已请求捕获
FRATE	帧率	– 0：全速率捕获 – 1：1/2 速率捕获 – 2：1/4 速率捕获 – 3：1/8 速率捕获
CPTACT(1)	捕获状态	– 1：捕获当前有效

1. 从 DCMIPP_PxSR 寄存器读取位。

配置示例

使用快照模式时的软件操作如下：

1. 软件设置 CPTMODE = 1，以使用快照模式。
2. 软件设置 PIPEN = 1，以让流选择将数据发送到 PipeN。
3. 软件设置 CPTREQ = 1，以请求捕获一帧。
4. 在接下来的第一个 Vsync，硬件将 CPTREQ 采样为 1，产生以下影响：
   a) 捕获有效开始：像素流入流水线并转储到内存中。
   b) CPTACT 置 1，表示当前正在进行捕获，并且最好不要修改流水线运算符的配置，除非是影子操作。
   c) CPTREQ 复位为 0，因此只会转储单个帧。
5. 在接下来的捕获完成中断中
   a) CPTACT 复位为 0，以表示捕获结束。
   b) 捕获完成，不再有像素流动，因此软件可以重新开始更新非影子寄存器的配置而不会引起任何问题，并且会使用内存中捕获的像素。根据传感器的消隐情况，此周期可能很短。
6. 在下一个 VSync，CPTREQ 通常被采样为 0，因此不再捕获帧
   – 一旦 CPTREQ 复位为 0，软件就可以再次将其设置为 1，以请求在下一帧进行捕获。这意味着可以通过连续设置 CPTREQ 为 1 来捕获连续的帧序列，一帧接一帧地连续捕获。

使用连续模式时，软件操作如下：

1. 软件设置 CPTMODE = 0，以使用连续模式。
2. 软件设置 PIPEN = 1，以让流选择将数据发送到 PipeN。
3. 软件设置 CPTREQ = 1，以请求连续捕获帧。
4. 在接下来的第一个 VSync，CPTREQ 被采样为 1，产生以下影响：
   – 与快照模式下一样，捕获开始，并且 CPTACT 置 1。
   – 与快照模式不同，CPTREQ 未被修改，仍然为 1。
5. 在接下来的捕获完成中断中：
   – 与快照模式下一样，CPTACT 复位为 0，捕获停止，软件可以短暂地重新配置流水线，并且软件可以使用捕获到的像素。
6. 在下一个 VSync，只要 CPTREQ 仍然为 1，捕获就会重新开始，过程如上所述。
7. 要停止捕获后续帧，软件会将 CPTREQ 复位为 0。
   – 正在进行的捕获将继续直到完成。
8. 在下一个 VSync，CPTREQ 被采样为 0，下一次捕获不会重新开始。

30.4.4 流水线禁用

可以中止流水线以停止任何帧采集，并且可能提供一种完全重编程该流水线的方法（包括非影子寄存器），或者让它在一段时间内完全禁用。必须考虑以下工作模式以正确停止流水线：

1. 禁用相应流水线的 DCMIPP_PxFCTCR 寄存器中的 CPTREQ 位
2. 在 DCMIPP_CMSR1 寄存器中轮询 CPTACT = 0 状态位，以检查流水线是否不再有效（空闲状态）
3. 在 DCMIPP_PxFCTCR 寄存器中禁用 PIPEN。

禁用流水线时，最后一帧将完全按照上述工作模式处理。

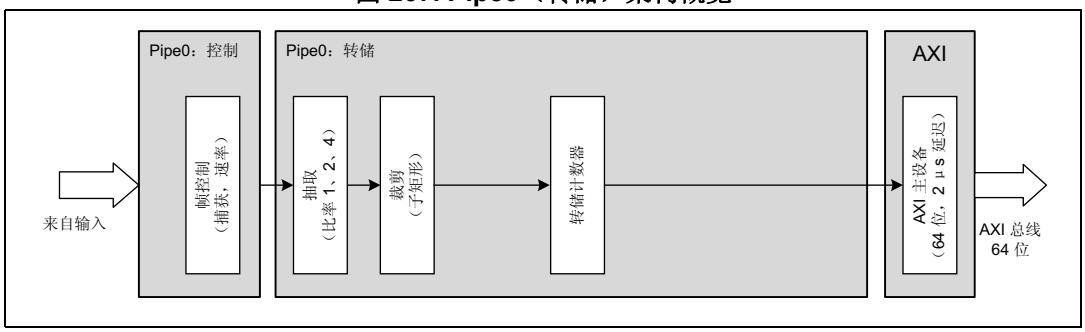
注： *要禁用并行接口（DCMIPP_PxFSCR 中的 PREN = 0），考虑上述到工作模式，建议在关闭并行接口之前首先禁用流水线。*

30.5 Pipe0（转储流水线）

30.5.1 概述

Pipe0 作为转储流水线工作：它从摄像头传感器模块提取数据，并将它们（按原样）转储到目标内存，过程中会进行一些基本的抽取和裁剪 2D 操作。

图 267. Pipe0（转储）架构概览



Pipe0 通过 16 位并行接口从连接到 DCMIPP 的摄像头传感器传感器模块检索数据。

帧控制器主要处理摄像头采集模式（连续或快照，帧速率），请参见*第 30.4.3 节*。

当摄像头的输入数据有效且应由 Pipe0 处理时，软件可以配置 2D 裁剪和抽取操作。提供了一个与要设置的数据量限制相结合的转储计数器，用于处理长度未知的数据或避免帧内数据量过大。

30.5.2 抽取

抽取允许以较低开销缩小帧。

基于并行接口摄像头模块：

- 根据 DCMIPP_P0PPCR 寄存器中位字段 BSM[1:0] 的配置，数据/字节级别的系数为 1x、2x、4x。抽取并非直接在像素级别起作用，而是基于字节或 16 位字的粒度（例如，对于填充的原始 10），以实现相当于每两个像素中抽取一个像素。位 OEBS 用于首先从奇数或偶数字节（或数据）开始捕获。
- 抽取也可以基于整个行，通过捕获所有行，或者通过位 LSM 捕获每两行中的一行。在这里，会再次向软件提供选择，通过配置位 OELS 拒绝帧内的奇数或偶数行。

表 248. DCMIPP_P0PPCR 位功能

位 ID	功能
BSM[1:0]	选择一行内需要捕获多少字节/数据。此功能仅适用于并行接口摄像头模块。
OEBS	允许用户选择滤波是从奇数还是偶数字节开始。
LSM	可滤除每两行中的一行。
OELS	软件可以选择滤波是否拒绝奇数行或偶数行。

30.5.3 裁剪/统计数据选择/抑制

转储流水线 (Pipe0) 有能力处理 2-D 裁剪过程。由于 HSYNC 事件，垂直裁剪基于行（来自物理 IO 或内嵌码检测，如果已选择）。

裁剪应用的水平区域基于数据（32 位宽）而不是像素，因为此流水线不直接考虑像素。它以 32 位粒度处理数据流。流水线传输 JPEG 格式时不支持裁剪功能。必须在 DCMIPP_P0SCSZR 中清零 ENABLE 位，以避免任何不可预测的行为。

必须通过配置寄存器 DCMIPP_P0SCSTR 和 DCMIPP_P0SCSZR 来指定要在帧内捕获的区域。两轴上的起点以及两个方向上的宽度均已相应设置。通过 DCMIPP_P0SCSZR 寄存器中的 POSNEG 位，可以获取该区域内部或外部的数据。一些传感器可以在帧的最初和最后行发送传感器配置数据和统计数据（直方图）。通过指定此窗口选择之外的活动区域 (POSNEG = 1)，仅会提取统计数据，并且软件可以决定对数据应用某些处理以进行校正，例如对比度或曝光。

特性

- 2D 裁剪操作及粒度：
  - 垂直轴上的行
  - 水平轴上的 32 位数据
- 可捕获已定义窗口内部或外部的数据（有效图像数据或统计数据）
- 为 HSIZE[11:0] 或 VSIZE[11:0] 使用 0 值进行裁剪不会对应用了 0 值的尺寸（垂直或水平轴）产生裁剪效果。

软件配置

必须配置寄存器 DCMIPP_P0SCSTR 和 DCMIPP_P0SCSZR，以将裁剪功能设置到转储流水线中：

1. 使用 HSTART[11:0] 配置裁剪后的水平起点，使用 HSIZE[11:0] 配置宽度（两者均使用 32 位数据粒度）。
2. 配置裁剪行，从 VSTART[11:0] 行开始，高度为 VSIZE[11:0]（两者均使用行粒度）
3. 如果 HSIZE 或 VSIZE 的任何值设置为 0，则硬件不会考虑在应用了 0 值的垂直或水平尺寸上进行裁剪操作，因此该尺寸中的所有像素都会被采样。
4. 通过配置 POSNEG 位，选择窗口的内部或外部进行数据捕获。
5. 使能裁剪功能会将 ENABLE 位置 1。选择 JPEG 作为 Pipe0 的输入格式时，必须清零此位。

注： *裁剪掉图片大小（HSTART 或 VSTART 过大）会导致处理帧无法获得保证。*

30.5.4 转储计数器

转储流水线上存在转储计数器。它用于对该帧中转储的数据量进行计数，并可能在转储量过大时限制转储的数据量。软件可利用该计数器了解已转储缓冲区的大小，并确保不会在预分配的缓冲区外进行转储。

在转储长度在接收前未知的内容时，它特别有用，例如编码的 JPG 流。当转储一个像素帧时，由于帧的宽度和高度已配置，因此大小已知。

该计数器可对几乎所有输入格式的 32 位数据进行计数，即使值以字节数表示。除 JPEG 字节流输入模式外，所有格式的计数器增量均为 4 字节粒度，而对于 JPEG 字节流输入模式，计数器粒度为 8 位，因为应用程序不知道已转储数据的量。

特性

- 转储计数器，用于对转储的字节数量进行计数。它会对转储的 32 位字的数量（增量为 4）进行计数（请参见 DCMIPP_PRCR 寄存器中的 FORMAT 位），除非连接到并行接口摄像头并以字节流模式转储 8 位数据，在这种情况下，它会按字节计数（增量为 1）。
  - 计数器在达到 0x03FF_FFFF 时饱和。
  - 在 FrameEnd，计数器的值会被重新复制到一个可读的阴影寄存器中，并且计数器本身会被复位，以开始对下一帧进行计数。
- 转储限制，该限制将转储的字数限制在最大值以下：
  - 如果计数器达到限制，所有后续字节将被删除，直到下一个 VSync。
  - 计数器会继续超过限制地计数，以表明接收了多少字节，如果这些字节未被删除，则必须已经被转储。
- 当一些字节因超过转储限制而被删除时，便会触发中断限制。
- 单色或原始拜耳填充（10/12/14 bpp 对齐到 16 bpp）也被计入数据量中。

软件配置

表 249. DCMIPP_P0DCCNTR 和 DCMIPP_P0DCLMTR 位功能

位 ID	功能
ENABLE	1：使能限制检查。
CNT	只读，对大多数输入格式的数据量进行 4 字节粒度的计数，或者在 DCMIPP_PRCR.FORMAT 中选择的 JPEG 字节流模式下以 8 位数据粒度对数据量进行计数。计数器增量基于流水线输出端点处的转储数据。无论输入格式如何，该值均以字节表示。
LIMIT	给出了可以转储的最大 32 位数据量。如果将 ENABLE 位置 1，则限制值 0 不一致，处理的帧无法得到保证。
LIMITIE	DCMIPP_PxIER.LIMITIE 位使能中断。

注： *10/12/14 bpc 的原始拜耳和单色像素填充到 16 位，因此略大于其 10/12/14 位大小。*

30.5.5 双缓冲区模式

转储流水线使用 AXI 主接口将数据从内部 FIFO 转储到外部内存。在应用程序中，可以逐帧处理帧数据交换内存区域。双缓冲区模式填补了这一功能。有两个内存地址寄存器用于初始化这些内存区域的基址。每个捕获帧事件的开始都会交换内存基址来处理双缓冲模式。

双缓冲模式可用于对一个缓冲区（帧缓冲区）进行后处理，同时读取另一个缓冲区以显示（显示缓冲区）。

软件配置

双缓冲模式需要激活，并进行地址配置，以定义数据在流水线输出端逐帧连续存储的两个内存区域。

- DCMIPP_P0PPM0AR1 和 DCMIPP_P0PPM0AR2 必须填入与双缓冲区存储单元相关的内存地址。
- 在 DCMIPP_P0PPCR 寄存器中将 DBM 位置 1 以使能双缓冲区模式。

30.5.6 像素封装

该模块在转储流水线 (Pipe0) 上工作，在内存的字中设置到达的像素。

特性

- DBM 地址：如果应用程序需要将一个帧的内存地址交换到下一个帧，则为双缓冲区模式及其相关的地址位置。
- 像素格式：所有常见格式，如*第 30.3.6 节*中所示。
- 填充：对 10/12/14 位进行 MSB 与 LSB 填充至 16 位。
- 多行事件：在发送行的每个倍数处生成的事件（适用于 DMA 和 INT）。

表 250. DCMIPP_PxPPCR 位功能

位 ID	功能	值	备注
M0A	内存中像素缓冲区的基址，32 位宽，按 16 字节对齐	-	像素缓冲区在内存中的行对齐假定为 16 字节。因此，缓冲区基址（即 M0A）必须是 16 字节的倍数。
FORMAT	输出像素格式	-	– Pipe0：未定义为用于转储的默认输入格式。它允许独特的格式 ByteData、ByteHeader 和单色/原始拜耳 8/10/12/14。
LINEMULT	行 (Hsync) 中断和事件的周期性，周期为 LINEMULT 配置的 2 的幂次（即每 1、2、...、128 行）。	-	-

30.5.7 上溢检测

此逻辑处理流控制风险：为 DCMIPP 提供了连续数据流（通过并行接口）且没有任何等待/保持功能，而在 DCMIPP 的下游，对内存的访问可能会暂时卡死。

在某些情况下，DCMIPP 的内部 FIFO 可能会变满，像素可能被删除。上溢检测逻辑会尽可能处理此类风险。

特性

- 出现风险时，软件会收到警告：
  - 当首次删除像素时，会触发中断 OVR，只要上溢风险存在，就会每帧重复。
  - 建议跳过/丢弃受影响的帧：
    - 溢出像素会被删除，因此当对内存的访问恢复时，行的剩余部分会被丢弃（不写出）。正常处理会在下一行开始时恢复。可能会丢弃几个整行。
- 出现风险时，没有帧间的影响：
  - VSync 会复位 DCMIPP 中所需的内容，以便前一帧的任何风险不影响后续帧中转存的像素。

软件配置

- 在出现风险时，会向中断 DCMIPP_PxSR.OVRF 发出警告。

30.6 像素格式说明

本节介绍了在并行接口和转储流水线中使用的像素格式。

表 251 中总结了各个接口（并行、转储输出）对像素格式的支持情况。

注： *在下面的表格中，使用象征性颜色表示相应的分量：红色和 Cr (= V) 用红色表示，蓝色和 Cb (= U) 用蓝色表示，绿色用绿色表示，而 Y、单色以及所有原始拜耳分量用灰色表示。*

30.6.1 并行接口格式

本段介绍了 DCMIPP 并行接口支持的输入像素格式和可能的交换组合。

注： *并行接口不会专门输入 RGB444/RGB555（及 RGB666）。然而，带有这些输出的传感器可以通过选择 RGB565（和 RGB888），以及通过将缺少的位与传感器输出的 MSB 连接或用跳线连接它们，来将它们连接到 DCMIPP。*

表 251. 并行接口输入像素格式

索引	IO 引脚		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
13	字节	周期 1/1	-	-	-	-	-	-	-	-	7	6	5	4	3	2	1	0
5	原始拜耳 8	周期 1/1	-	-	-	-	-	-	-	-	7	6	5	4	3	2	1	0
6	原始拜耳 10	周期 1/1	-	-	-	-	-	-	9	8	7	6	5	4	3	2	1	0
7	原始拜耳 12	周期 1/1	-	-	-	-	11	10	9	8	7	6	5	4	3	2	1	0
8	原始拜耳 14	周期 1/1	-	-	13	12	11	10	9	8	7	6	5	4	3	2	1	0
9	单色 8	周期 1/1	-	-	-	-	-	-	-	-	7	6	5	4	3	2	1	0
10	单色 10	周期 1/1	-	-	-	-	-	-	9	8	7	6	5	4	3	2	1	0
11	单色 12	周期 1/1	-	-	-	-	11	10	9	8	7	6	5	4	3	2	1	0
12	单色 14	周期 1/1	-	-	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2-1	RGB565 - 8 位	周期 1/2	-	-	-	-	-	-	-	-	R4	R3	R2	R1	R0	G5	G4	G3
		周期 2/2	-	-	-	-	-	-	-	-	G2	G1	G0	B4	B3	B2	B1	B0
2-2	RGB565 - 16 位	周期 1/1	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
2-2	PCB：Sensor444 到 DCMIPP-RGB565	周期 1/1	R3	R2	R1	R0	R3	G3	G2	G1	G0	G3	G2	B3	B2	B1	B0	B3
2-2	PCB：Sensor555 到 DCMIPP-RGB565	周期 1/1	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	G4	B4	B3	B2	B1	B0
4-1	RGB888 - 12 位	周期 1/2	-	-	-	-	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4
		周期 2/2	-	-	-	-	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
	YUV444 -12 位	周期 1/2	-	-	-	-	V7	V6	V5	V4	V3	V2	V1	V0	Y7	Y6	Y5	Y4
		周期 2/2	-	-	-	-	Y3	Y2	Y1	Y0	U7	U6	U5	U4	U3	U2	U1	U0

表 251. 并行接口输入像素格式（续）

索引	IO 引脚		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
4-2	RGB888 - 8 位	周期 1/3	-	-	-	-	-	-	-	-	R7	R6	R5	R4	R3	R2	R1	R0
		周期 2/3	-	-	-	-	-	-	-	-	G7	G6	G5	G4	G3	G2	G1	G0
		周期 3/3	-	-	-	-	-	-	-	-	B7	B6	B5	B4	B3	B2	B1	B0
	YUV444 - 8 位	周期 1/3	-	-	-	-	-	-	-	-	V7	V6	V5	V4	V3	V2	V1	V0
		周期 2/3	-	-	-	-	-	-	-	-	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
		周期 3/3	-	-	-	-	-	-	-	-	U7	U6	U5	U4	U3	U2	U1	U0
3	PCB：Sensor 666 到 RGB888	周期 1/3	-	-	-	-	-	-	-	-	R5	R4	R3	R2	R1	R0	R5	R4
		周期 2/3	-	-	-	-	-	-	-	-	G5	G4	G3	G2	G1	G0	G5	G4
		周期 3/3	-	-	-	-	-	-	-	-	B5	B4	B3	B2	B1	B0	B5	B4
1-1	YUV422 - 8 位 (YUY'V)	周期 1/4	-	-	-	-	-	-	-	-	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
		周期 2/4	-	-	-	-	-	-	-	-	U7	U6	U5	U4	U3	U2	U1	U0
		模式 3/4	-	-	-	-	-	-	-	-	Y7'	Y6'	Y5'	Y4'	Y3'	Y2'	Y1'	Y0'
		周期 4/4	-	-	-	-	-	-	-	-	V7	V6	V5	V4	V3	V2	V1	V0
1-2	YUV422 - 16 位 (YUY'V)	周期 1/2	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	U7	U6	U5	U4	U3	U2	U1	U0
		周期 2/2	Y7'	Y6'	Y5'	Y4'	Y3'	Y2'	Y1'	Y0'	V7	V6	V5	V4	V3	V2	V1	V0

表 252. 索引与 DCMIPP_PRCR 寄存器值之间的对应关系

索引	FORMAT[7:0]	EDM[2:0]
1-1	0x1E	0x00
1-2	0x1E	0x04
2	0x22	0x04
3	0x23	0x00
4-1	0x24	0x02
4-2	0x24	0x00
5	0x2A	0x00
6	0x2B	0x01
7	0x2C	0x02
8	0x2D	0x03
9	0x4A	0x00
10	0x4B	0x01
11	0x4C	0x02
12	0x4D	0x03
13	其他值	0x00

为了适应非标准传感器，并行接口十分灵活，支持交换位、周期和分量，即：

- 位：在整个 16 位 IO 引脚之间交换 LSB 与 MSB。
- 周期：交换周期 1/2 与周期 2/2。不影响 1 个或 3 个周期。

表 253 显示了基于 RGB565 格式的这些排列。它列出了周期 1 和周期 2 中在引脚 0 到引脚 7 上采样的位，并为每个采样位返回 RGB 输出分量的分配情况。

表 253. 并行接口输入像素格式

IO 引脚		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
原生 RGB565 8 位	周期 1/2	-	-	-	-	-	-	-	-	R4	R3	R2	R1	R0	G5	G4	G3
	周期 2/2	-	-	-	-	-	-	-	-	G2	G1	G0	B4	B3	B2	B1	B0
交换 LSB-MSB RGB565 - 8 位	周期 1/2	G3	G4	G5	R0	R1	R2	R3	R4	-	-	-	-	-	-	-	-
	周期 2/2	B0	B1	B2	B3	B4	G0	G1	G2	-	-	-	-	-	-	-	-
交换周期 RGB565 8 位	周期 1/2	-	-	-	-	-	-	-	-	G2	G1	G0	B4	B3	B2	B1	B0
	周期 2/2	-	-	-	-	-	-	-	-	R4	R3	R2	R1	R0	G5	G4	G3

30.6.2 转储流水线格式

由于转储流水线具有转储能力，因此支持：

- 10/12/14 位的单色和原始拜耳像素填充到 16 bpp。

表 254 详细说明了这些附加像素格式的支持。假定其他格式是宽度相同的像素。

表 254. 转储流水线输出像素格式(1)

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字节/单色/原始 8 bpp	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	PN + 3								PN + 2								PN + 1								PN + 0							
PAD = 0 单色/原始 10 bpp	0						9	8	7	6	5	4	3	2	1	0	0						9	8	7	6	5	4	3	2	1	0
	PN + 1																PN + 0															
PAD = 0 单色/原始 12 bpp	0				11	10	9	8	7	6	5	4	3	2	1	0	0				11	10	9	8	7	6	5	4	3	2	1	0
	PN + 1																PN + 0															
PAD = 0 单色/原始 14 bpp	0	13	12	11	10	9	8	7	6	5	4	3	2	1	0		0	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	PN + 1																PN + 0															
PAD = 1 单色/原始 10 bpp	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0
	PN + 1																PN + 0															
PAD = 1 单色/原始 12 bpp	11	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	11	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0
	PN + 1																PN + 0															
PAD = 1 单色/原始 14 bpp	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0
	PN + 1																PN + 0															

表 254. 转储流水线输出像素格式(1)（续）

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RGB 16 bpp	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PN + 1																PN + 0															
	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
RGB 24 bpp（字 0 到 2）	7	6	5	4	3	2	1	0	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PN + 1								PN + 0																							
	B7	B6	B5	B4	B3	B2	B1	B0	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
	PN + 2																PN + 1															
	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0
	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	23	22	21	20	19	18	17	16
	PN + 3																								PN + 2							
	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0	R7	R6	R5	R4	R3	R2	R1	R0
YUV422-1（1 个字）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PN + 1																PN + 0															
	Y'7	Y'6	Y'5	Y'4	Y'3	Y'2	Y'1	Y'0	V7	V6	V5	V4	V3	V2	V1	V0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	U7	U6	U5	U4	U3	U2	U1	U0

1. PN 表示像素 N。

30.6.3 AXI IP-Plug

AXI IP-Plug 通过 AXI 将流水线的数据转储到内存中。

特性

- 流水线 ID：单一 ID，因为部署了唯一的流水线转储 (Pipe0)
- FIFO：大小足以处理最高 2 µs 的延迟（1 µs 平均往返时间 + 1 µs 的内存减速）
  - Pipe0：200 MB/s，因此是 512 字节
- 突发长度：所有流水线 64 或 128 字节
- 未完成事务能力：四个，128 字节突发
- 带宽限制器（通过每个流水线的未完成限制）：
  - 避免最后一行转储导致内存带宽过载
- 唯一 AXI-QOS：DCMIPP 的唯一 QOS

通用软件配置

AXI IP-Plug 在 DCMIPP_IPGR1/2/3/8 寄存器中有一个通用置。

仅当 IP-Plug 处于空闲模式时，通用寄存器 R1 和 R3 才可写。

- MEMORYPAGE：用于有效处理 DRAM（突发不会穿过此模块）。
- PSTART：请求冻结插头，以安全地重新配置它。
- IDLE：此状态表示插头已冻结，可安全地重新配置。
- IPPID[7:0]、ARCHIID[4:0]、REVID[4:0]、DID[5:0]：IP-Plug 的版本 ID。

每个客户端的软件配置

AXI IP-Plug 在寄存器 DCMIPP_IPC0R1/2/3 中具有每个客户端的配置。

仅在 IP-Plug 处于空闲模式时，客户端寄存器 R1 和 R3 才可写（请参见上方通用寄存器中的 PSTART 以将其锁定，或参见下方示例）。

- Client1 = Pipe0

可配置的每个客户端特性如下：

- OTR：每个客户端的未完成事务数 (Oustd = OTR + 1)
- TRAFFIC[2:0]：突发大小：最佳选择 128 字节，因此 TRAFFIC = 4
- WLRU[3:0]：总带宽比例：WLRU = BWi/ 总和（所有带宽）
- SVC：系统虚拟通道。保持 SVC = 0
- DPREGSTART：此客户端 FIFO 起始的本地基址。
- DPREGEND：此客户端 FIFO 结束的本地基址（包含字）。

配置计算

IP-Plug 通用配置：

- PSTART：设置为 1，使 IP-Plug 处于空闲状态并配置其寄存器 G1/3、R1/3。
- IDLE：读取并等待至 = 1，使其处于空闲状态。
- MEMORYPAGE：4 个，用于优化 256 字节的内存页。

IP-Plug 每个客户端的配置：

- OTR（客户端）：每个客户端的未完成事务，总共四个未完成事务。
  - 提供的值 OTR 等于未完成事务数减 1。
- DPREG 大小（客户端）：RAM 共享大小，512 字节
  - DPREGSTART = 0（或在另一个客户端位置之后）
  - DPREGEND = 4，大小为 ((4 - 0) + 1) x 128 B = 512 字节（对于此客户端）。要编程的值为 3，因为包含了字。

其结果是 Client1 被分配给 Pipe0：

- IP-Plug 通用寄存器：
  - DCMIPP_IPGR1.MEMORYPAGE = 2（默认，256 字节）
  - DCMIPP_IPGR2.PSTART = 1（锁定 IP-Plug 以进行重新配置）
  - DCMIPP_IPGR3.IDLE 以等待至 = 1（继续之前一直等待）
- 视频静态映射到 IP-Plug Client1：
  - DCMIPP_IPC1R1.OTR = 4
  - DCMIPP_IPC1R1.TRAFFIC = 4（默认）
  - DCMIPP_IPC1R3.DPREGSTART = 0
  - DCMIPP_IPC1R3.DPREGEND = 4
- IP-Plug 通用寄存器：
  - DCMIPP_IPGR2.PSTART= 1（配置后解锁 IP-Plug）

30.7 影子寄存器

转储流水线必须具有在不停止的情况下重新配置的能力。某些寄存器可以在帧的中间写入，而不影响实际的帧采集。根据*表 255* 描述的触发事件，在写入相应的物理寄存器之前，值被存储在一些影子寄存器中。

寄存器映射中的每个物理寄存器的地址相对于其自己的影子寄存器地址增加了 0x200。

表 255. 影子和物理寄存器

影子寄存器	物理寄存器	触发事件
DCMIPP_P0FCTCR	DCMIPP_P0CFCTCR	Vsync 边界
DCMIPP_P0SCSTR	DCMIPP_P0CSCSTR	Vsync 边界
DCMIPP_P0SCSZR	DCMIPP_P0CSCSZR	Vsync 边界
DCMIPP_P0PPCR	DCMIPP_P0CPPCR	Vsync 边界
DCMIPP_P0PPM0AR1	DCMIPP_P0CPPxM0AR1	Vsync 边界
DCMIPP_P0PPM0AR2	DCMIPP_P0CPPM0AR2	Vsync 边界

必须刷新帧内的影子寄存器，以准备好下一帧开始时的上下文，从而避免在采集期间出现任何不可预测的行为。软件必须对以下中断源做出响应，以启动对影子寄存器的访问：

- 在 VSYNC 中断（流水线开始处的 FrameStart）中生成中断，软件可以在当前帧结束时更新加载到物理寄存器中的影子寄存器（从而在下一帧激活）。软件拥有整个帧周期来完成更新。
- 在 LINE 中断中，可使用线选择触发软件来更新考虑新帧上下文的物理寄存器。应用程序代码必须确保在帧结束事件（当前帧）之前进行更新。如果未更新，则可以在下一帧中部分考虑上下文，从而处理不一致的设置。

在任何情况下，软件都必须确保在当前帧结束前更新影子寄存器，以便在下一帧中激活。如果更新开始但在当前帧结束前未完成，则会导致下一帧的帧采集不一致（混合旧配置和新配置）。

建议在对应的流水线被禁止且处于空闲状态时更改非影子寄存器（请参见*第 30.4.4 节：流水线禁用*）。

30.8 DCMIPP 低功耗模式

本节介绍了 DCMIPP 与*表 256* 中列出的模式的行为。

表 256. DCMIPP 低功耗模式

模式	说明
停止	外设内容保持不变。 建议在进入前按照流水线禁止过程操作。
待机	掉电。 在退出此模式后，必须重新初始化外设。
睡眠	无影响。 DCMIPP 在此模式下仍然工作，外设中断会导致设备退出此模式。

30.9 DCMIPP 中断

本节介绍了 DCMIPP 中断，以及与实时操作相关、更具全局性的特性，即：

- 自由运行：DCMIPP 可以在无 CPU 干预的情况下运行
- 影子寄存器：缓解软件的时间约束
- 中断：中断源
- 事件引脚：用来驱动外部 IP，如 DMA

30.9.1 自由运行 DCMIPP

在配置并建立摄像头流之后，它可以永久运行，而无需任何软件干预。可以使用以下功能：

- Pipe0 的永久流水线激活：设置后，流水线会为无限数量的帧转储帧数据。
  - DCMIPP_PxFCTCR.CPTMODE = 0
  - DCMIPP_PxSR.CPTACT = 1
  - DCMIPP_PxFCTCR.FRATE = xx（具体取决于所需的帧二次采样率）
- 双缓冲帧：可使用两个 BaseAddress 指针为输出缓冲区定义 DCMIPP：
  - 软件可以设置这两个 BaseAddress 指针，使硬件响应 VSYNC 事件，从而轮流从一个缓冲区 BaseAddress 切换到另一个缓冲区 BaseAddress。

30.9.2 中断

DCMIPP 处理三个中断引脚，其中第三个是前两个的或运算。

每个中断都有相关联的寄存器组提供以下功能：

- 未屏蔽事件：报告已发生且尚未清除的事件
- 屏蔽：屏蔽不必要的事件
- 已屏蔽事件：不可读，任何激活的未屏蔽位都会触发中断引脚
- 清除寄存器：进行写操作以清除任何激活的事件

中断引脚在高电平时激活：它们在被未屏蔽事件触发时变为高电平，并保持高电平状态，直至该中断的所有已屏蔽事件都被清除。

中断线和寄存器如下：

1. 并行接口，处理以下中断事件：
   - ERR：检测到内嵌码同步错误。
     - > 并行接口中的意外行为，没有问题时通常不会触发。
     - > 这是由于意外的传感器行为或发送错误引起的。
2. Pipe0（转储），处理以下中断事件：
   - OVR：数据上溢：内存速度太慢，无法及时处理接收到的像素。
     - > Pipe0 中的意外行为，没有问题时通常不会触发。
     - > 与上述 OVR 的建议相同：跳过受影响的帧。
   - LIMIT：接收到的数据量大于允许的最大转储容量。
     - > Pipe0 中的意外行为，没有问题时通常不会触发。
     - > 发送的流过长。必须在内存中分配更多空间来存储下一帧的发送。

- VSYNC：主中断，大多数软件可依此执行。
  - > 触发：在 VSync（中间消隐）时，为永久性（即使没有转储有效）。
  - > 通常用于为下一帧缓慢重新配置流水线（至少是影子寄存器）
  - > 通常用于触发使用之前捕获的帧。
- FRAME：次要中断，作为快速软件的备份
  - > 触发：在此流水线的最后数据转储之后，如果未发生捕获则无效。
  - > 通常用于快速重新配置非影子寄存器（仅在垂直消隐期间）
  - > 通常用于触发使用之前捕获的帧。
- LINE：由基于条带的运算符的中断触发：在每 1/2/4/8/16/32/64/128 个转储行和最后一个行之后。它在流水线末端靠近内存输出的位置进行测量和提取。
  - > 通常用于触发快速/响应式软件或硬件基于条带的运算符。

注： *LINE 标志（LINEF 位）在帧结束时置 1，以在帧高度不是所选行数的倍数时触发软件事件，此时会触发 LINEF 事件。在这种情况下，帧的最后一部分可以少于选择的行数。即使行数不完整，也会生成事件/中断。软件必须考虑到，当到达帧结束（FRAME 事件）时，该最后事件的行数可能小于预期。*

3. 通用中断：将上述中断的事件分组到单个寄存器组中，并驱动其自身的中断行。
   - 第三个中断适用于单个驱动程序处理整个 DCMIPP 的系统：单次访问此中断状态可检索整个 DCMIPP 的状态。
   - 前两个中断适用于驱动程序处理流水线以及（可能的）另一个驱动程序处理并行接口的系统。在这种情况下，每个驱动程序处理其各自的中断寄存器组，无需信号量对冲突的访问进行仲裁。

   如果使能，则在从 IP-Plug 到内存的 AXI 传输过程中检测到传输错误时，还可以生成通用中断。DCMIPP_CMSR2 寄存器中的错误标志 ATXERR 置 1，可以生成中断来通知软件这一错误状态。此错误没有关联的特定硬件操作，由软件来处理传输错误情况。

表 257 中汇总了 DCMIPP 中断。生成中断的事件会在两个位置发生：

- 在通用中断寄存器组中发生一次（由单个 CPU 驱动时）
- 在本地中断寄存器组中发生一次（由多个 CPU 驱动时）。

表 257 提供了本地（寄存器和位）和全局（只有位，因为所有位都在同一个 DCMIPP_CMSR2 寄存器中）位置。

表 257. DCMIPP 中断

中断事件	本地寄存器	本地位	DCMIPP 中的通用位_CMSR2	事件标志/中断清除方法	中断使能控制位本地/通用寄存器
并行接口中的同步错误	DCMIPP_PRSR	ERRF	PRERRF	写入 CERRF = 1/CPRERRF = 1	ERRIE/PRERRFIE
AXI 传输错误	-	-	ATXERR	写入 ALERTCF = 1	ATXERRIE

表 257. DCMIPP 中断（续）

中断事件	本地寄存器	本地位	DCMIPP 中的通用位_CMSR2	事件标志/中断清除方法	中断使能控制位本地/通用寄存器
Pipe0 中的上溢	DCMIPP_P0SR	OVRF	P0OVRF	写入 COVRF = 1/ CP0OVRF = 1	OVRIE/P0OVRIE
Pipe0 中的限值违例		LIMITF	P0LIMITF	写入 CLIMITF = 1/ CP0LIMITF = 1	LIMITIE/P0LIMITIE
Pipe0 中的帧开始 (VSYNC)		VSYNC	P0VSYNC	写入 CVSYNCF = 1/ CP0VCSYNCF = 1	VSYNCIE/P0VSYNCIE
Pipe0 中的帧结束 (FRAME)		FRAMEF	P0FRAMEF	写入 CFRAMEF = 1/ CP0FRAMEF = 1	FRAMEIE/P0FRAMIE
Pipe0 中的多行 (LINE)		LINEF	P0LINEF	写入 CLINEF = 1/ CP0LINEF = 1	LINEFIE/P0LINEFIE

30.9.3 事件引脚

事件引脚被暴露出来，以便让辅助硬件 IP（如中央 DMA）与其同步。事件引脚是流水线的未屏蔽同步事件（VSYNC、FRAME、LINE、HSYNC）。

内部事件在其引脚上生成脉冲（长度为 15 个 APB 周期）。

表 258. 事件连接

流水线	事件名称	内部信号
Pipe0	HSYNC	dcmipp_p0hsync_evt
	VSYNC	dcmipp_p0vsync_evt
	帧	dcmipp_p0frame_evt
	行	dcmipp_p0line_evt

30.10 DCMIPP 寄存器

寄存器分为具有相同前缀的多个组，如*表 259* 所示。

表 259. DCMIPP 寄存器结构

寄存器名称	偏移	功能
DCMIPP_IP	0x000	IP-Plug 寄存器
DCMIPP_PR	0x100	并行接口
DCMIPP_CM	0x200	通用寄存器
DCMIPP_P0	0x500	使用影子寄存器的 Pipe0（转储流水线）
	0x700	使用物理寄存器的 Pipe0（转储流水线）
DCMIPP_	0xFF0	IP 版本和配置（无前缀）

30.10.1 DCMIPP IP-Plug 全局寄存器 1 (DCMIPP_IPGR1)

DCMIPP IP-Plug global register 1

偏移地址：0x000

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	QOS_MODE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEMORYPAGE[2:0]		
													rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24 **QOS_MODE**：服务质量 (Quality of service)
一组允许构建和配置能够满足带宽和延迟要求的架构的功能。

位 23:3 保留，必须保持复位值。

位 2:0 **MEMORYPAGE[2:0]**：以 64 字节为单位的 2 的幂次的内存页大小 (Memory page size, as power of 2 of 64-byte units)
0：64 字节
1：128 字节
2：256 字节
3：512 字节
4：1K 字节
5：2K 字节
6：4K 字节
7：8K 字节

30.10.2 DCMIPP IP-Plug 全局寄存器 2 (DCMIPP_IPGR2)

DCMIPP IP-Plug global register 2

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSTART
															rw

位 31:1 保留，必须保持复位值。

位 0 **PSTART**：请求锁定 IP-Plug，以允许重新配置 (Request to lock the IP-Plug, to allow reconfiguration)
0：未请求锁定，IP-Plug 根据后台硬件按需运行。
1：已请求锁定：IP-Plug 会短暂冻结（锁定激活时，请参见 IDLE 位）。
配置完成后必须将 PSTART 复位为 0，以重启 IP-Plug。

30.10.3 DCMIPP IP-Plug 全局寄存器 3 (DCMIPP_IPGR3)

DCMIPP IP-Plug global register 3

偏移地址：0x008

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDLE
															r

位 31:1 保留，必须保持复位值。

位 0 **IDLE**：IP-Plug 的状态 (Status of IP-Plug)

0：IP-Plug 正在运行（根据后台硬件按需运行）

1：IP-Plug 当前已锁定，可重新配置

IDLE 在请求后通过将 PSTART 置 1 来置 1，并通过将 PSTART 复位为 0 来复位。

30.10.4 DCMIPP IP-Plug 标识寄存器 (DCMIPP_IPGR8)

DCMIPP IP-Plug identification register

偏移地址：0x01C

复位值：0xAA04 0314

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPPID[7:0]								Res.	Res.	Res.	ARCHIID[4:0]				
r	r	r	r	r	r	r	r				r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	REVID[4:0]					Res.	Res.	DID[5:0]					
			r	r	r	r	r			r	r	r	r	r	r

位 31:24 **IPPID[7:0]**：IP 标识符 (IP identifier) (0xAA)

位 23:21 保留，必须保持复位值。

位 20:16 **ARCHIID[4:0]**：架构标识符 (Architecture identifier) (0x04)

位 15:13 保留，必须保持复位值。

位 12:8 **REVID[4:0]**：版本标识符 (Revision identifier) (0x03)

位 7:6 保留，必须保持复位值。

位 5:0 **DID[5:0]**：部门标识符 (Division identifier) (0x14)

30.10.5 DCMIPP IP-Plug 客户端 x 寄存器 1 (DCMIPP_IPCxR1)

DCMIPP IP-Plug Clientx register 1

偏移地址：0x020 + 0x10 * (x - 1)（x = 1 到 1）

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	OTR[1:0]		Res.	Res.	Res.	Res.	Res.	TRAFFIC[2:0]		
						rw	rw						rw	rw	rw

位 31:10 保留，必须保持复位值。

位 9:8 **OTR[1:0]**：最大未完成事务数 (Maximum outstanding transactions)

0：禁止。无未完成事务限制（FIFO 大小除外）

1：最多 2 个未完成事务正在进行。

...

3：最多 4 个未完成事务正在进行。

不允许其他值。

位 7:3 保留，必须保持复位值。

位 2:0 **TRAFFIC[2:0]**：以 8 字节为单位的 2 的幂次的突发大小 (Burst size as power of 2 of 8-byte units)

0：8 字节

1：16 字节

2：32 字节

3：64 字节

4：128 字节

其他值：保留

30.10.6 DCMIPP IP-Plug 客户端 x 寄存器 2 (DCMIPP_IPCxR2)

DCMIPP IP-Plug Clientx register 2

偏移地址：0x024 + 0x10 * (x - 1)（x = 1 到 1）

复位值：0x0001 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WLRU[3:0]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:20 保留，必须保持复位值。

位 19:16 **WLRU[3:0]**：WLRU[3:0] 仲裁比率 (Ratio for WLRU[3:0] arbitration)
客户端获取总带宽的一部分 = 比率（客户端）/总和（所有比率）
0：比率部分 = 1
1：比率部分 = 2
...
15：比率部分 = 16

位 15:12 保留，必须保持复位值。

位 11:8 保留，必须保持复位值。

位 7:0 保留，必须保持复位值。

30.10.7 DCMIPP IP-Plug 客户 x 寄存器 3 (DCMIPP_IPCxR3)

DCMIPP IP-Plug Clientx register 3

偏移地址：0x028 + 0x10 * (x - 1)（x = 1 到 1）

复位值：0x003F 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DPREGEND[5:0]					
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DPREGSTART[5:0]					
										rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:16 **DPREGEND[5:0]**：客户端 x FIFO 的结束字（AXI 宽度 = 64 位）(End word (AXI width = 64 bits) of the FIFO of Clientx)
地址字包含在 FIFO 中，因此下一个 DPREGSTART 为 DPREGEND + 1。

位 15:6 保留，必须保持复位值。

位 5:0 **DPREGSTART[5:0]**：客户端 x FIFO 的开始字（AXI 宽度 = 64 位）(Start word (AXI width = 64 bits) of the FIFO of Clientx)

30.10.8 DCMIPP 并行接口控制寄存器 (DCMIPP_PRCR)

DCMIPP parallel interface control register

偏移地址：0x104

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SWAP BITS	SWAP CYCLES	Res.	FORMAT[7:0]							
					rw	rw		rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	ENABLE	Res.	EDM[2:0]			Res.	Res.	VSPOL	HSPOL	PCKPOL	ESS	Res.	Res.	Res.	Res.
	rw		rw	rw	rw			rw	rw	rw	rw				

位 31:27 保留，必须保持复位值。

位 26 **SWAPBITS**：交换每个已接收分量内的 LSB 与 MSB (Swap LSB vs. MSB within each received component)

0：与接收相同
1：交换 MSB 与 LSB

位 25 **SWAPCYCLES**：交换两个周期接收到的像素数据（周期 0 与周期 1）(Swap data (cycle 0 vs. cycle 1) for pixels received on two cycles)

0：默认
1：交换有效：周期 1 的数据在周期 0 的数据之前使用。
对于一个或三个周期内接收到的像素，不得通过软件激活交换。

位 24 保留，必须保持复位值。

位 23:16 **FORMAT[7:0]**：

0x1E：YUV422
0x22：RGB565
0x24：RGB888 (= YUV444)
0x2A：RAW8
0x2B：RAW10
0x2C：RAW12
0x2D：RAW14
0x4A：单色 8 位
0x4B：单色 10 位
0x4C：单色 12 位
0x4D：单色 14 位
0x5A：字节流（JPEG，压缩视频）
其它值：数据只能通过数据/转储流水线按原样捕获和输出（例如 JPEG 或字节输入格式）。
单色 Y 输入作为 YUV 像素插入流水线，U 和 V 分量设为中性色来表示灰色。

位 15 保留，必须保持复位值。

位 14 **ENABLE**：并行接口使能 (Parallel interface enable)

0：禁止并行接口以降低功耗
1：使能并行接口
使能该位之前，必须正确编程并行接口配置寄存器。

位 13 保留，必须保持复位值。

位 12:10 **EDM[2:0]**：扩展数据模式 (Extended data mode)

000：接口每个像素时钟捕获 8 位数据
001：接口每个像素时钟捕获 10 位数据
010：接口每个像素时钟捕获 12 位数据
011：接口每个像素时钟捕获 14 位数据
100：接口每个像素时钟捕获 14 位数据
其他值：保留

位 9:8 保留，必须保持复位值。

位 7 **VSPOL**：垂直同步极性 (Vertical synchronization polarity)

此位指示数据在并行接口上无效时 VSYNC 引脚的电平。
0：VSYNC 低电平有效
1：VSYNC 高电平有效

位 6 **HSPOL**：水平同步极性 (Horizontal synchronization polarity)
此位指示数据在并行接口上无效时 HSYNC 引脚的电平。
0：HSYNC 低电平有效
1：HSYNC 高电平有效

位 5 **PCKPOL**：像素时钟极性 (Pixel clock polarity)
此位用来配置像素时钟的捕获沿
0：下降沿有效
1：上升沿有效

位 4 **ESS**：内嵌码同步选择 (Embedded synchronization select)
0：硬件同步：数据捕获（帧/行开始/停止）由 HSYNC/VSYNC 信号同步。
1：内嵌码同步：数据捕获由数据流中嵌入的同步码同步。
仅对 8 位并行数据有效。此位置 1 时，将忽略 HSPOL/VSPOL。

位 3:0 保留，必须保持复位值。

30.10.9 DCMIPP 并行接口内嵌同步码寄存器 (DCMIPP_PRESCR)

DCMIPP parallel interface embedded synchronization code register

偏移地址：0x108

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FEC[7:0]								LEC[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LSC[7:0]								FSC[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **FEC[7:0]**：帧结束分隔码 (Frame end delimiter code)
此字节指定帧结束分隔码。由 4 个字节组成，格式为 0xFF 0x00 0x00 FEC。
如果将 FEC 编程为 0xFF，所有未使用的其他代码 (0xFF00 00XY) 都将视为帧结束分隔符。

位 23:16 **LEC[7:0]**：行结束分隔码 (Line end delimiter code)
此字节指定行结束分隔码。由 4 个字节组成，格式为 0xFF 0x00 0x00 LEC。

位 15:8 **LSC[7:0]**：行开始分隔码 (Line start delimiter code)
此字节指定行开始分隔码。由 4 个字节组成，格式为 0xFF 0x00 0x00 LSC。

位 7:0 **FSC[7:0]**：帧开始分隔码 (Frame start delimiter code)
此字节指定帧开始分隔码。由 4 个字节组成，格式为 0xFF 0x00 0x00 FSC。
如果将 FSC 编程为 0xFF，将检测不到任何帧开始分隔符，但在 FEC 代码后第一次出现 LSC 时将视为帧分隔符的开始。

30.10.10 DCMIPP 并行接口内嵌码同步取消屏蔽寄存器 (DCMIPP_PRESUR)

DCMIPP parallel interface embedded synchronization unmask register

偏移地址：0x10C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FEU[7:0]								LEU[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LSU[7:0]								FSU[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **FEU[7:0]**：帧结束分隔符取消屏蔽 (Frame end delimiter unmask)

此字节指定对帧结束分隔码的屏蔽。

0：将帧结束分隔符与收到的数据进行比较时，将屏蔽 DCMIPP_ESCR 中 FEC 字节中的相应位

1：将帧结束分隔符与收到的数据进行比较时，将比较 DCMIPP_ESCR 中 FEC 字节中的相应位

位 23:16 **LEU[7:0]**：行结束分隔符取消屏蔽 (Line end delimiter unmask)

此字节指定对行结束分隔码的屏蔽。

0：将行结束分隔符与收到的数据进行比较时，将屏蔽 DCMIPP_ESCR 中 LEC 字节中的相应位

1：将行结束分隔符与收到的数据进行比较时，将比较 DCMIPP_ESCR 中 LEC 字节中的相应位

位 15:8 **LSU[7:0]**：行开始分隔符取消屏蔽 (Line start delimiter unmask)

此字节指定对行开始分隔码的屏蔽。

0：将行开始分隔符与收到的数据进行比较时，将屏蔽 DCMIPP_ESCR 中 LSC 字节中的相应位

1：将行开始分隔符与收到的数据进行比较时，将比较 DCMIPP_ESCR 中 LSC 字节中的相应位

位 7:0 **FSU[7:0]**：帧开始分隔符取消屏蔽 (Frame start delimiter unmask)

此字节指定对帧开始分隔码的屏蔽。

0：将帧开始分隔符与收到的数据进行比较时，将屏蔽 DCMIPP_ESCR 中 FSC 字节中的相应位

1：将帧开始分隔符与收到的数据进行比较时，将比较 DCMIPP_ESCR 中 FSC 字节中的相应位

30.10.11 DCMIPP 并行接口中断使能寄存器 (DCMIPP_PRIER)

DCMIPP parallel interface interrupt enable register

偏移地址：0x1F4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERRIE	Res.	Res.	Res.	Res.	Res.	Res.
									rw						

位 31:7 保留，必须保持复位值。

位 6 **ERRIE**：同步错误中断使能 (Synchronization error interrupt enable)

0：不生成中断

1：如果未按正确顺序接收内嵌同步码，则生成中断。

该位仅适用于内嵌码同步模式。

位 5:0 保留，必须保持复位值。

30.10.12 DCMIPP 并行接口状态寄存器 (DCMIPP_PRSR)

DCMIPP parallel interface status register

偏移地址：0x1F8

复位值：0x0003 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VSYNC	HSYNC
														r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERRF	Res.	Res.	Res.	Res.	Res.	Res.
									r						

位 31:18 保留，必须保持复位值。

位 17 **VSYNC**：

如果在 DCMIPP_PRCR 寄存器中将 ENABLE 位置 1，并且接收到像素时钟，则此位会以正确编程的极性给出 VSYNC 引脚的状态。无论在 VPOL 位中选择了何种极性，在消隐期间均会将此位置 1，其它情况则清零。

使用内嵌同步码时：

0：有效帧

1：在帧之间同步

如果使用内嵌码同步，则仅当 DCMIPP_CR 中的 CAPTURE 位置 1 时，该位才有意义。

位 16 **HSYNC**：

如果在 DCMIPP_PRCR 寄存器中将 ENABLE 位置 1，并且接收到像素时钟，则此位提供 HSYNC 引脚的状态，且具有正确编程的极性。无论在 HPOL 位中选择了何种极性，在消隐期间均会将此位置 1，其它情况则清零。

使用内嵌同步码时：

0：有效行

1：在行之间同步

如果使用内嵌码同步，则仅当 DCMIPP_CR 中的 CAPTURE 位置 1 时，该位才有意义。

位 15:7 保留，必须保持复位值。

位 6 **ERRF**：同步错误原始中断状态 (Synchronization error raw interrupt status)

0：未检测到同步错误

1：未按正确顺序接收内嵌码同步字符。

此位仅在内嵌码同步模式下有效。向 DCMIPP_PRFCR 中的 CERRF 位写入“1”即可将此位清零。

该位仅适用于内嵌码同步模式。

位 5:0 保留，必须保持复位值。

30.10.13 DCMIPP 并行接口中断清零寄存器 (DCMIPP_PRFCR)

DCMIPP parallel interface interrupt clear register

偏移地址：0x1FC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CERRF	Res.	Res.	Res.	Res.	Res.	Res.
									w						

位 31:7 保留，必须保持复位值。

位 6 **CERRF**：同步错误中断状态清零 (Synchronization error interrupt status clear)
在此位中写入“1”可将 DCMIPP_RIS 中的 ERRF 位清零。
该位仅适用于内嵌码同步模式。

位 5:0 保留，必须保持复位值。

30.10.14 DCMIPP 通用 IP 硬件配置寄存器 (DCMIPP_CMHWCFGR)

DCMIPP common IP HW configuration register

偏移地址：0x200

复位值：0x0000 0014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCSI	CPAR	CPIPES[1:0]		Res.	Res.
										r	r	r	r		

位 31:6 保留，必须保持复位值。

位 5 **CCSI**：CSI-2 主机接口的存在情况 (Presence of the CSI-2 host interface)：
0：没有来自 CSI-2 主机的输入。
1：已实现来自 CSI-2 主机的输入（与 CSI-2 主机的存在无关）。

位 4 **CPAR**：并行接口的存在情况 (Presence of the parallel interface)：
0：未实现并行接口输入。
1：已实现并行接口输入。

位 3:2 **CPIPES[1:0]**：已实例化流水线的数量（转储或像素流水线）(Amount of instantiated pipes (either dump or pixel pipes))

位 1:0 保留，必须保持复位值。

30.10.15 DCMIPP 通用配置寄存器 (DCMIPP_CMCR)

DCMIPP common configuration register

偏移地址：0x204

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFC	Res.	Res.	Res.	Res.
											w				

位 31:5 保留，必须保持复位值。

位 4 **CFC**：清零帧计数器 (Clear frame counter)
将此位置 1 时，与流水线相关的帧计数器将被清零。它会复位 DCMIPP_CMFRCR 寄存器。该位始终读为 0。

位 3:1 保留，必须保持复位值。

位 0 保留，必须保持复位值。

30.10.16 DCMIPP 通用帧计数器寄存器 (DCMIPP_CMFRCR)

DCMIPP common frame counter register

偏移地址：0x208

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRMCNT[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRMCNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **FRMCNT[31:0]**：帧计数器，只读，循环 (Frame counter, read-only, loops around)
在映射到在 DCMIPP_CMCR 寄存器的位 PSFC[1:0] 中配置的流水线的 VSYNC 检测后递增。使用该寄存器的 CRC 位清零计数器。

30.10.17 DCMIPP 通用中断使能寄存器 (DCMIPP_CMIER)

DCMIPP common interrupt enable register

偏移地址：0x3F0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P0 OVR IE	P0 LIMIT IE	Res.	Res.	Res.	P0 VSYNC IE	P0 FRAME IE	P0 LINE IE	Res.	PR ERR IE	ATX ERR IE	Res.	Res.	Res.	Res.	Res.
rw	rw				rw	rw	rw		rw	rw					

位 31 保留，必须保持复位值。

位 30:28 保留，必须保持复位值。

位 27 保留，必须保持复位值。

位 26:24 保留，必须保持复位值。

位 23 保留，必须保持复位值。

位 22:20 保留，必须保持复位值。

位 19 保留，必须保持复位值。

位 18:16 保留，必须保持复位值。

位 15 **P0OVRIE**：Pipe0 的上溢中断使能 (Overrun interrupt enable for Pipe0)

0：不生成中断

1：生成中断

位 14 **P0LIMITIE**：Pipe0 的限制中断使能 (Limit interrupt enable for Pipe0)

0：不生成中断

1：生成中断

位 13:11 保留，必须保持复位值。

位 10 **P0VSYNCIE**：Pipe0 的垂直同步中断使能 (Vertical sync interrupt enable for Pipe0)

0：不生成中断

1：生成中断

位 9 **P0FRAMEIE**：Pipe0 的帧捕获完成中断使能 (Frame capture complete interrupt enable for Pipe0)

0：不生成中断

1：生成中断

位 8 **P0LINEIE**：Pipe0 的多行捕获完成中断使能 (Multi-line capture complete interrupt enable for Pipe0)

0：不生成中断

1：生成中断

位 7 保留，必须保持复位值。

位 6 **PRERRIE**：并行接口的限制中断使能 (Limit interrupt enable for the parallel Interface)

0：不生成中断

1：生成中断

位 5 **ATXERRIE**：IP-Plug 的 AXI 传输错误中断使能 (AXI transfer error interrupt enable for IP-Plug)

0：不生成中断

1：生成中断

位 4:0 保留，必须保持复位值。

30.10.18 DCMIPP 通用状态寄存器 1 (DCMIPP_CMSR1)

DCMIPP common status register 1

偏移地址：0x3F4

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P0CPTACT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRVSYNC	PRHSYNC
r														r	r

位 31:24 保留，必须保持复位值。

位 23:16 保留，必须保持复位值。

位 15 **P0CPTACT**：Pipe0 的有效帧捕获（从帧开始到帧完成的过程中有效）(Active frame capture (active from start-of-frame to frame complete) for Pipe0)

0：当前没有有效的捕获
1：捕获当前有效

位 14:12 保留，必须保持复位值。

位 11:10 保留，必须保持复位值。

位 9:2 保留，必须保持复位值。

位 1 **PRVSYNC**：

如果在 DCMIPP_PRCR 寄存器中将 ENABLE 位置 1，并且接收到像素时钟，则此位会在并行接口上以正确编程的极性给出 VSYNC 引脚的状态。无论在 DCMIPP_PRCR 寄存器的 VPOL 位中选择了何种极性，在消隐期间均会将此位置 1，其它情况则清零。
使用内嵌码同步时，此位的含义如下：
0：有效帧
1：在帧之间同步
如果使用内嵌码同步，则仅当 DCMIPP_PRCR 寄存器中的 CAPTURE 位置 1 时，该位才有意义。

位 0 **PRHSYNC**：

如果在 DCMIPP_PRCR 寄存器中将 ENABLE 位置 1，并且接收到像素时钟，则此位会在并行接口上以正确编程的极性给出 HSYNC 引脚的状态。无论在 DCMIPP_PRCR 寄存器的 HPOL 位中选择了何种极性，在消隐期间均会将此位置 1，其它情况则清零。
使用内嵌码同步时，此位的含义如下：
0：有效行
1：在行之间同步
如果使用内嵌码同步，则仅当 DCMIPP_PRCR 寄存器中的 CAPTURE 位置 1 时，该位才有意义。

30.10.19 DCMIPP 通用状态寄存器 2 (DCMIPP_CMSR2)

DCMIPP common status register 2

偏移地址：0x3F8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P0 OVRF	P0 LIMITF	Res.	Res.	Res.	P0 VSYNC F	P0 FRAME F	P0 LINEF	Res.	PR ERRF	ATX ERRF	Res.	Res.	Res.	Res.	Res.
r	r				r	r	r		r	r					

位 31:28 保留，必须保持复位值。

位 27 保留，必须保持复位值。

位 26:24 保留，必须保持复位值。

位 23 保留，必须保持复位值。

位 22:20 保留，必须保持复位值。

位 19 保留，必须保持复位值。

位 18:16 保留，必须保持复位值。

位 15 **P0OVRF**：Pipe0 的上溢原始中断状态 (Overrun raw interrupt status for Pipe0)

0：未发生数据缓冲区上溢

1：发生数据缓冲区上溢，此帧数据损坏

此位通过向 DCMIPP_CMFCR 寄存器中的 CP0OVRF 位写入 1 来清零。

位 14 **P0LIMITF**：Pipe0 的限制原始中断状态 (Limit raw interrupt status for Pipe0)

当数据计数器 DCMIPP_P0DCCNT 达到其最大值 DCMIPP_P0DCLIMIT 时，此位置 1。通过向 DCMIPP_CMFCR 寄存器中的 CP0LIMITF 位写入 1 进行清零。

位 13:11 保留，必须保持复位值。

位 10 **P0VSYNCF**：Pipe0 的 VSYNC 原始中断状态 (VSYNC raw interrupt status for Pipe0)

当 VSYNC 信号从无效状态更改为有效状态时，此位置 1。

如果使用内嵌码同步，则仅当 DCMIPP_CR 中的 CAPTURE 位置 1 时，此位才置 1。通过向 DCMIPP_CMFCR 寄存器中的 CP0VSYNCF 位写入 1 进行清零。

位 9 **P0FRAMEF**：Pipe0 的帧捕获完成原始中断状态 (Frame capture completed raw interrupt status for Pipe0)

0：无捕获或正在进行的捕获

1：已捕获一帧的所有数据

对一帧数据或裁剪窗口内的所有数据捕获完毕后，此位置 1。

如果捕获裁剪窗口，则将在裁剪窗口最后一行的行结束时将此位置 1，即使捕获的帧为空（例如，裁剪窗口超出帧范围）。

此位通过向 DCMIPP_CMFCR 寄存器中的 CP0FRAMEF 位写入 1 来清零。

位 8 **P0LINEF**：Pipe0 的多行捕获完成原始中断状态 (Multi-line capture completed raw interrupt status for Pipe0)

当完成一行或多行时，此位置 1。LINEF 事件的周期性由 DCMIPP_P0PPCR 寄存器的 LINEMULT 位配置。当到达帧结束时，即使设置的 LINEMULT 值不是帧总行数的倍数，也会触发此事件以允许软件操作。

如果使用内嵌码同步，则仅当 DCMIPP_CR 寄存器中的 CAPTURE 位置 1 时，此位才置 1。通过向 DCMIPP_CMFCR 寄存器中的 CP0LINEF 位写入 1 进行清零。

位 7 保留，必须保持复位值。

位 6 **PRERRF**：并行接口的同步错误原始中断状态 (Synchronization error raw interrupt status for the parallel interface)

0：未检测到同步错误

1：未按正确顺序接收内嵌码同步字符。

此位仅在内嵌码同步模式下有效。通过向 DCMIPP_CMFCR 寄存器中的 CPRERRF 位写入 1 进行清零。

该位仅适用于内嵌码同步模式。

位 5 **ATXERRF**：IP-Plug 的 AXI 传输错误中断状态标志 (AXI transfer error interrupt status flag for the IP-Plug)

0：未检测到 AXI 传输错误

1：AXI 客户端上发生 AXI 传输错误。此位指示客户端上的错误，但不进行任何特定的硬件操作，因此软件必须处理这种情况（通常在调试软件应用程序代码时使用）。

此位通过向 DCMIPP_CMFCR 寄存器中的 CATXERRF 位写入 1 来清零。

位 4:0 保留，必须保持复位值。

30.10.20 DCMIPP 通用中断清零寄存器 (DCMIPP_CMFCR)

DCMIPP common interrupt clear register

偏移地址：0x3FC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP0 OVRF	CP0 LIMITF	Res.	Res.	Res.	CP0 VSYNCF	CP0 FRAMEF	CP0 LINEF	Res.	CPR ERRF	CATX ERRF	Res.	Res.	Res.	Res.	Res.
w	w				w	w	w		w	w					

位 31 保留，必须保持复位值。

位 30:28 保留，必须保持复位值。

位 27 保留，必须保持复位值。

位 26:24 保留，必须保持复位值。

位 23 保留，必须保持复位值。

位 22:20 保留，必须保持复位值。

位 19:16 保留，必须保持复位值。

位 15 **CP0OVRF**：上溢中断状态清零 (Overrun interrupt status clear)
向此位写入 1 可清零 DCMIPP_CMSR2 寄存器中的 P0OVRF 位

位 14 **CP0LIMITF**：限制中断状态清零 (limit interrupt status clear)
向此位写入 1 可清零 DCMIPP_CMSR2 寄存器中的 P0LIMITF 位

位 13:11 保留，必须保持复位值。

位 10 **CP0VSYNCF**：垂直同步中断状态清零 (Vertical Synchronization interrupt status clear)
向此位写入 1 可清零 DCMIPP_CMSR2 寄存器中的 P0VSYNCF 位。

位 9 **CP0FRAMEF**：帧捕获完成中断状态清零 (Frame capture complete interrupt status clear)
向此位写入 1 可清零 DCMIPP_CMSR2 寄存器中的 P0FRAMEF 位。

位 8 **CP0LINEF**：多行捕获完成中断状态清零 (Multi-line capture complete interrupt status clear)
向此位写入 1 可清零 DCMIPP_CMSR2 寄存器中的 P0LINEF 位

位 7 保留，必须保持复位值。

位 6 **CPRERRF**：同步错误中断状态清零 (Synchronization error interrupt status clear)
向此位写入 1 可清零 DCMIPP_CMSR2 寄存器中的 PRERRF 位。
该位仅适用于内嵌码同步模式。

位 5 **CATXERRF**：AXI 传输错误中断状态清零 (AXI transfer error interrupt status clear)
向此位写入 1 可清零 DCMIPP_CMSR2 寄存器中的 ATXERRF 位。

位 4:0 保留，必须保持复位值。

30.10.21 DCMIPP Pipe0 硬件配置寄存器 (DCMIPP_P0HWCFGR)

DCMIPP Pipe0 HW configuration register

偏移地址：0x400

复位值：0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CVP	Res.	CROI[3:0]				CGM[1:0]		CRB[1:0]		DBM	CDS	CPLA[1:0]	
		r		r	r	r	r	r	r	r	r	r	r	r	r

位 31:14 保留，必须保持复位值。

位 13 **CVP**：流水线虚拟化能力 (Capability for pipeline virtualization)
0：流水线无虚拟化
1：为流水线实现虚拟化机制

位 12 保留，必须保持复位值。

位 11:8 **CROI[3:0]**：ROI 能力 (Capability for ROI)：定义流水线中的 ROI 数量

位 7:6 **CGM[1:0]**：伽马转换能力 (Capability for gamma conversions)
00：无伽马转换
01：包含三个插值段的伽玛转换
10, 11：保留。

位 5:4 **CRB[1:0]**：去马赛克功能 (Capability for demosaicing)
00：无去马赛克功能
01：使用 2018-03 算法的去马赛克（3 x 3 矩阵，边缘/LineHV/峰值）
10, 11：保留

位 3 **DBM**：双缓冲区模式 (Double buffer mode)
0：不支持双缓冲区模式
1：支持双缓冲区模式

位 2 **CDS**：降采样滤波器能力 (Capability for downsize filter)
0：无降采样滤波器，而是使用基本抽取
1：基于 BoxFilter 的降采样滤波器

位 1:0 **CPLA[1:0]**：平面缓冲区能力 (Capability for planar buffers)
00：仅具有转储到共面缓冲区的能力
01：具有转储到半平面缓冲区的能力（例如 YUV420-2）
10：具有转储到完全平面缓冲区的能力（例如 YUV420-3）
11：保留

30.10.22 DCMIPP Pipe0 流选择配置寄存器 (DCMIPP_P0FSCR)

DCMIPP Pipe0 flow selection configuration register

偏移地址：0x404

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31 **PIPEN**：激活 PipeN (Activation of PipeN)
0：禁止流水线
1：使能流水线，可使用 CPTMODE、CPTREQ、CPTACK 开始捕获

注：　此位不会进行影子操作，与此寄存器中的所有其他位不同。

位 30:0 保留，必须保持复位值。

30.10.23 DCMIPP Pipe0 流控制配置寄存器 (DCMIPP_P0FCTCR)

DCMIPP Pipe0 flow control configuration register

偏移地址：0x500

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPT REQ	CPT MODE	FRATE[1:0]	
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3 **CPTREQ**：已请求捕获 (Capture requested)

0：未请求捕获下一帧

1：请求捕获下一帧

当 PIPEN = 1 且 CPTREQ = 1 时，流水线等待第一个 VSync，随即自动开始捕获，并将 CPTACT 置 1 来表示。

在快照模式下，CPTREQ 位在第一个接收到的帧开始时清零。

在连续采集模式下，捕获保持有效且 CPTREQ = 1，直到软件清零 CPTREQ：捕获在进行中的帧结束时停止，CPTACT 会被复位。

使能此位之前，必须正确编程 DCMI 和流水线配置寄存器。

位 2 **CPTMODE**：捕获模式 (Capture mode)

0：连续采集模式——收到的数据将通过 AXI 主机传输到目标存储区。

1：快照模式（单帧）——一旦激活，接口将等待帧开始，然后通过 AXI 主机传输单帧。帧结束时将自动复位 CPTACT 位。

位 1:0 **FRATE[1:0]**：帧捕获率控制 (Frame capture rate control)

这些位定义了帧捕获频率。仅在连续采集模式下有效，快照模式下会被忽略。

00：捕获所有帧

01：每两帧捕获一帧（带宽降低 50%）

10：每四帧捕获一帧（带宽降低 75%）

11：每八帧捕获一帧（带宽降低 87%）

30.10.24 DCMIPP Pipe0 统计数据/裁剪开始寄存器 (DCMIPP_P0SCSTR)

DCMIPP Pipe0 statistic/crop start register

偏移地址：0x504

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	VSTART[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	HSTART[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **VSTART[11:0]**：垂直开始，从 0 到 4094 像素高 (Vertical start, from 0 to 4094 pixels high)

位 15:12 保留，必须保持复位值。

位 11:0 **HSTART[11:0]**：水平开始，从 0 到 4094 字宽 (Horizontal start, from 0 to 4094 words wide)

30.10.25 DCMIPP Pipe0 统计数据/裁剪大小寄存器 (DCMIPP_P0SCSZR)

DCMIPP Pipe0 statistic/crop size register

偏移地址：0x508

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ENABLE	POSNEG	Res.	Res.	VSIZE[11:0]											
rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	HSIZE[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **ENABLE**：

该位由软件置 1 和清零，

0：旁路。所有数据都会被计算，如果统计数据在帧内发送，则它们作为像素数据发送到处理流水线。

1：使能。根据 POSNEG 位的值，由 VSIZE、HSIZE、VSTART 和 HSTART 定义的矩形可以用来提取或移除某些数据（统计提取或移除，或基本的 2D 裁剪功能）。

如果 POSNEG = 0，发送矩形区域内的数据（它可对应于统计数据移除，或作为数据有效图像区域的裁剪功能）。

如果 POSNEG = 1，发送矩形区域外的数据（它可对应于统计数据提取，拒绝窗口内的所有数据）。

如果输入格式为 JPEG，则此位必须保持清零，以避免流水线发生不可预测的行为。

位 30 **POSNEG**：

该位由软件置 1 和清零，仅当 ENABLE 位置 1 时才有意义。

0：正区域，由 VSIZE、HSIZE、VSTART、HSTART 定义的矩形

1：负区域，排除由 VSIZE、HSIZE、VSTART、HSTART 定义的矩形的区域

位 29:28 保留，必须保持复位值。

位 27:16 **VSIZE[11:0]**：垂直大小，从 0 到 4094 像素高 (Vertical size, from 0 to 4094 pixels high)

如果在通过 ENABLE 位使能裁剪时值保持为 0，则不会在垂直方向上执行裁剪操作。

位 15:12 保留，必须保持复位值。

位 11:0 **HSIZE[11:0]**：水平大小，从 0 到 4094 字宽（数据 32 位）(Horizontal size, from 0 to 4094 word wide (data 32-bit))

如果在通过 ENABLE 位使能裁剪时值保持为 0，则不会在水平方向上执行裁剪操作。

30.10.26 DCMIPP Pipe0 转储计数器寄存器 (DCMIPP_P0DCCNTR)

DCMIPP Pipe0 dump counter register

偏移地址：0x5B0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	CNT[25:16]									
						r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:26 保留，必须保持复位值。

位 25:0 **CNT[25:0]**：在帧期间转储的数据数量 (Number of data dumped during the frame)
数据大小以字节表示。它仅计算通过 CROP 2D 功能选择的数据。计数器在达到 0x3FFFFFF 时饱和。除了具有字节粒度的字节流格式（例如 JPEG）外，所有格式的粒度均为 32 位。

30.10.27 DCMIPP Pipe0 转储限制寄存器 (DCMIPP_P0DCLMTR)

DCMIPP Pipe0 dump limit register

偏移地址：0x5B4

复位值：0x00FF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ENABLE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LIMIT[23:16]							
rw								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LIMIT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **ENABLE**：
0：禁止，不检查发送的 32 位字数
1：使能，根据限制进行检查

位 30:24 保留，必须保持复位值。

位 23:0 **LIMIT[23:0]**：进行 2D 裁剪操作后，在一帧中能转储的最大 32 位数据量 (Maximum number of 32-bit data that can be dumped during a frame, after the crop 2D operation) 最大值不得超过 0x003F FFFF。

30.10.28 DCMIPP Pipe0 像素打包器配置寄存器 (DCMIPP_P0PPCR)

DCMIPP Pipe0 pixel packer configuration register

偏移地址：0x5C0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBM
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINEMULT[2:0]			Res.	OELS	LSM	OEBS	BSM[1:0]		Res.	PAD	Res.	Res.	Res.	Res.	Res.
rw	rw	rw		rw	rw	rw	rw	rw		rw					

位 31:17 保留，必须保持复位值。

位 16 **DBM**：双缓冲区模式 (Double buffer mode)

该位由软件置 1 和清零。

0：未激活双缓冲区模式。Pipe0 始终将数据转储到由 DCMIPP_P0PPM0AR1 设置的内存地址。

1：激活双缓冲区模式。转储地址位置在每一帧上从 DCMIPP_P0PPM0AR1 切换到 DCMIPP_P0PPM0AR2。

位 15:13 **LINEMULT[2:0]**：LINE 事件和中断的捕获完成行数 (Amount of capture completed lines for LINE event and interrupt)

000：一行后的事件
001：两行后的事件
010：四行后的事件
011：八行后的事件
100：十六行后的事件
101：32 行后的事件
110：64 行后的事件
111：128 行后的事件

位 12 保留，必须保持复位值。

位 11 **OELS**：奇数/偶数行选择（行选择开始）(Odd/even line select (line select start))

此位与 LSM 字段 (LSM = 1) 一起使用。
0：接口在帧开始后捕获第一行，丢弃第二行
1：接口在帧开始后捕获第二行，丢弃第一行

位 10 **LSM**：行选择模式 (Line select mode)

0：接口捕获所有接收到的行
1：接口每两行捕获一行

位 9 **OEBS**：奇数/偶数字节选择（字节选择开始）(Odd/even byte select (byte select start))

此位与 BSM 字段 (BSM ≠ 00) 一起使用
0：接口从帧/行开始捕获第一个数据（字节或双字节），丢弃第二个数据
1：接口从帧/行开始捕获第二个数据（字节或双字节），丢弃第一个数据

位 8:7 **BSM[1:0]**：字节选择模式 (Byte select mode)

00：接口捕获所有接收到的数据
01：接口每 2 个数据捕获 1 个数据
10：接口每四个字节捕获一个字节
11：接口每四个字节捕获两个字节
模式 10 和 11 仅适用于 DCMIPP_PRCR 寄存器中的 EDM [2:0] = 000。

位 6 保留，必须保持复位值。

位 5 **PAD**：单色和原始拜耳 10/12/14 bpp 的填充模式：MSB 与 LSB 对齐 (Pad mode for monochrome and raw Bayer 10/12/14 bpp: MSB vs. LSB alignment)

0：按 LSB 对齐（在 MSB 上填充空位），以便向后兼容以前的 DCMI。

1：按 MSB 对齐（在 LSB 上填充空位），以便软件或 GPU 更轻松地处理。

位 4:0 保留，必须保持复位值。

30.10.29 DCMIPP Pipe0 像素打包器内存 0 地址寄存器 1 (DCMIPP_P0PPM0AR1)

DCMIPP Pipe0 pixel packer Memory0 address register 1

偏移地址：0x5C4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M0A[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M0A[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	r	r

位 31:0 **M0A[31:0]**：内存 0 地址 (Memory0 address)

数据写入的内存区域 0 的基址。假定它是 16 的倍数，因此其位 3:0 始终为 0x0。

30.10.30 DCMIPP Pipe0 像素打包器内存 0 地址寄存器 2 (DCMIPP_P0PPM0AR2)

DCMIPP Pipe0 pixel packer Memory0 address register 2

偏移地址：0x5C8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M0A[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M0A[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	r	r

位 31:0 **M0A[31:0]**：内存 0 地址 (Memory0 address)

数据写入的内存区域 0 的基址。假定它是 16 的倍数，因此其位 3:0 始终为 0x0。

30.10.31 DCMIPP Pipe0 中断使能寄存器 (DCMIPP_P0IER)

DCMIPP Pipe0 interrupt enable register

偏移地址：0x5F4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVRIE	LIMITIE	Res.	Res.	Res.	VSYNCIE	FRAMEIE	LINEIE
								rw	rw				rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7 **OVRIE**：溢出中断使能 (Overrun interrupt enable)
0：不生成中断
1：如果 AXI 主设备无法在收到新数据（32 位）之前传输上一个数据，则生成中断。

位 6 **LIMITIE**：限制中断使能 (Limit interrupt enable)
0：达到限制时不生成中断
1：达到限制时生成中断。

位 5:3 保留，必须保持复位值。

位 2 **VSYNCIE**：VSYNC 中断使能 (VSYNC interrupt enable)
0：不生成中断
1：每个 VSYNC（无论是否捕获）都会生成中断

位 1 **FRAMEIE**：帧捕获完成中断使能 (Frame capture completed interrupt enable)
0：不生成中断
1：在完全捕获裁剪的帧后，将产生中断

位 0 **LINEIE**：多行捕获完成中断使能 (Multi-line capture completed interrupt enable)
0：接收到行时不生成中断
1：完全捕获一组行（或到达最后一行）后生成中断

30.10.32 DCMIPP Pipe0 状态寄存器 (DCMIPP_P0SR)

DCMIPP Pipe0 status register

偏移地址：0x5F8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPTACT	Res.	Res.	Res.	Res.	Res.	Res.	Res.
								r							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVRF	LIMITF	Res.	Res.	Res.	VSYNCF	FRAMEF	LINEF
								r	r				r	r	r

位 31:24 保留，必须保持复位值。

位 23 **CPTACT**：捕获立即状态 (Capture immediate status)

0：捕获当前无效

1：捕获当前有效

该位会在帧捕获完成事件结束时自动复位（在该帧的所有数据被捕获且 IP-Plug 已开始在 AXI 上发出最后一次突发传输后，通常在下一个 VSync 之前）。

位 22:17 保留，必须保持复位值。

位 16:8 保留，必须保持复位值。

位 7 **OVRF**：上溢原始中断状态 (Overrun raw interrupt status)

0：未发生数据缓冲区上溢

1：发生数据缓冲区上溢，此帧数据损坏

此位通过向 DCMIPP_P0FCR 寄存器中的 COVRF 位写入 1 来清零。

位 6 **LIMITF**：限制原始中断状态 (Limit raw interrupt status)

当数据计数器 DCMIPP_PxDCCNTR 达到其最大值 DCMIPP_PxDCLIMITR 时，此位置 1。

通过向 DCMIPP_P0FCR 寄存器中的 CLIMITF 位写入 1 进行清零。

位 5:3 保留，必须保持复位值。

位 2 **VSYNCF**：VSYNC 原始中断状态 (VSYNC raw interrupt status)

当 VSYNC 信号从无效状态更改为有效状态时，此位置 1。如果使用内嵌码同步，则仅当 DCMIPP_CR 中的 CAPTURE 位置 1 时，此位才置 1。

通过向 DCMIPP_P0FCR 寄存器中的 CVSYNCF 位写入 1 进行清零。

位 1 **FRAMEF**：帧捕获完成原始中断状态 (Frame capture completed raw interrupt status)

0：无捕获或正在进行的捕获

1：已捕获一帧的所有数据

对一帧数据或裁剪窗口内的所有数据捕获完毕后，此位置 1。如果捕获裁剪窗口，则将在裁剪窗口最后一行的行结束时将此位置 1。即使捕获的帧为空（例如，窗口裁剪超出帧范围），该位也将置 1。

此位通过向 DCMIPP_P0FCR 中的 CFRAMEF 位写入 1 来清零。

位 0 **LINEF**：多行捕获完成原始中断状态 (Multi-line capture completed raw interrupt status)

当完成一行或多行时，此位置 1。对于 JPEG 模式，此位在帧结束时置 1。

LINEF 事件的周期性由 DCMIPP_P0PPCR 寄存器的 LINEMULT 位配置。当到达帧结束时，即使设置的 LINEMULT 值不是帧总行数的倍数，也会触发此事件以允许软件操作。

如果使用内嵌码同步，则仅当 DCMIPP_CR 寄存器中的 CAPTURE 位置 1 时，此位才置 1。通过向 DCMIPP_P0FCR 寄存器中的 CLINEF 位写入 1 进行清零。

30.10.33 DCMIPP Pipe0 中断清零寄存器 (DCMIPP_P0FCR)

DCMIPP Pipe0 interrupt clear register

偏移地址：0x5FC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COVRF	CLIMITF	Res.	Res.	Res.	CVSYNCF	CFRAMEF	CLINEF
								w	w				w	w	w

位 31:8 保留，必须保持复位值。

位 7 **COVRF**：上溢中断状态清零 (Overrun interrupt status clear)
向此位写入 1 可清零 DCMIPP_P0SR 寄存器中的 OVRF 位。

位 6 **CLIMITF**：限制中断状态清零 (limit interrupt status clear)
向此位写入 1 可清零 DCMIPP_P0SR 寄存器中的 LIMITF。

位 5:3 保留，必须保持复位值。

位 2 **CVSYNCF**：垂直同步中断状态清零 (Vertical Synchronization interrupt status clear)
向此位写入 1 可清零 DCMIPP_P0SR 寄存器中的 VSYNCF 位。

位 1 **CFRAMEF**：帧捕获完成中断状态清零 (Frame capture complete interrupt status clear)
向此位写入 1 可清零 DCMIPP_P0SR 寄存器中的 FRAMEF 位。

位 0 **CLINEF**：多行捕获完成中断状态清零 (Multi-line capture complete interrupt status clear)
向此位写入 1 可清零 DCMIPP_P0SR 寄存器中的 LINEF。

30.10.34 DCMIPP Pipe0 当前流控制配置寄存器 (DCMIPP_P0CFCTCR)

DCMIPP Pipe0 current flow control configuration register

偏移地址：0x700

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPTREQ	CPTMODE	FRATE[1:0]	
												r	r	r	r

位 31:4 保留，必须保持复位值。

位 3 **CPTREQ**：已请求捕获 (Capture requested)

0：未请求捕获下一帧。

1：请求捕获下一帧。

当 PIPEN = 1 且 CPTREQ 设置为 1 时，流水线等待第一个 VSync，随即自动开始捕获，并将 CPTACT 置 1 来表示。

在快照模式下，CPTREQ 位在接收到的第一帧开始时自动清零。

在连续采集模式下，捕获保持有效且 CPTREQ = 1，直到软件清零 CPTREQ：捕获在进行中的帧结束时停止，CPTACT 会被复位。

使能此位之前，必须正确编程 DCMI 和流水线配置寄存器。

位 2 **CPTMODE**：捕获模式 (Capture mode)

0：连续采集模式——收到的数据将通过 AXI 主机传输到目标存储区。

1：快照模式（单帧）——一旦激活，接口将等待帧开始，然后通过 AXI 主机传输单帧。帧结束时将自动复位 CAPTURE 位。

位 1:0 **FRATE[1:0]**：帧捕获率控制 (Frame capture rate control)

这些位定义了帧捕获频率。仅在连续采集模式下有效，快照模式下会被忽略。

00：捕获所有帧

01：每两帧捕获一帧（带宽降低 50%）

10：每四帧捕获一帧（带宽降低 75%）

11：每八帧捕获一帧（带宽降低 87%）

30.10.35 DCMIPP Pipe0 当前统计数据/剪裁开始寄存器 (DCMIPP_P0CSCSTR)

DCMIPP Pipe0 current statistic/crop start register

偏移地址：0x704

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	VSTART[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	HSTART[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27:16 **VSTART[11:0]**：当前垂直开始，从 0 到 4094 像素高 (Current vertical start, from 0 to 4094 pixels high)

位 15:12 保留，必须保持复位值。

位 11:0 **HSTART[11:0]**：当前水平开始，从 0 到 4094 字宽 (Current horizontal start, from 0 to 4094 words wide)

30.10.36 DCMIPP Pipe0 当前统计数据/剪裁大小寄存器 (DCMIPP_P0CSCSZR)

DCMIPP Pipe0 current statistic/crop size register

偏移地址：0x708

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ENABLE	POSNEG	Res.	Res.	VSIZE[11:0]											
r	r			r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	HSIZE[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31 **ENABLE**：ENABLE 位的当前值 (Current value of the ENABLE bit)

0：旁路。所有都会被计算，如果统计数据在帧内发送，则它们作为像素数据发送到处理流水线。

1：使能：根据 POSNEG 位的值，由 VSIZE、HSIZE、VSTART 和 HSTART 定义的矩形可以用来提取或移除特定数量的数据（统计提取或移除，或基本的 2D 裁剪功能）

如果 POSNEG = 0，发送矩形区域内的数据（可对应于统计数据移除，或作为数据有效图像区域的裁剪功能）。

如果 POSNEG = 1，发送矩形区域外的数据（可对应于统计数据提取，拒绝窗口内的所有数据）

位 30 **POSNEG**：POSNEG 位的当前值 (Current value of the POSNEG bit)

此位仅在 ENABLE 位置 1 时才有意义。

0：正区域。由 VSIZE、HSIZE 和 VSTART、HSTART 定义的矩形是有效区域。

1：负区域。有效区域是排除由 VSIZE、HSIZE 和 VSTART、HSTART 定义的矩形的区域。

位 29:28 保留，必须保持复位值。

位 27:16 **VSIZE[11:0]**：当前垂直大小，从 0 到 4094 像素高 (Current vertical size, from 0 to 4094 pixels high)

如果在通过 ENABLE 位使能剪裁时值保持为 0，则该值会内部强制为最大值 0xFFE。

位 15:12 保留，必须保持复位值。

位 11:0 **HSIZE[11:0]**：当前水平大小，从 0 到 4094 字宽（数据 32 位）(Current horizontal size, from 0 to 4094 word wide (data 32-bit))

如果在通过 ENABLE 位使能剪裁时值保持为 0，则该值会内部强制为最大值 0xFFE。

30.10.37 DCMIPP Pipe0 当前像素打包器配置寄存器 (DCMIPP_P0CPPCR)

DCMIPP Pipe0 current pixel packer configuration register

偏移地址：0x7C0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBM
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINEMULT[2:0]			Res.	OELS	LSM	OEBS	BSM[1:0]		Res.	PAD	Res.	Res.	Res.	Res.	Res.
r	r	r		r	r	r	r	r		r					

位 31:17 保留，必须保持复位值。

位 16 **DBM**：双缓冲区模式 (Double buffer mode)

0：未激活双缓冲区模式。Pipe0 始终将数据转储到由 DCMIPP_P0PPM0AR1 寄存器设置的内存地址

1：激活双缓冲区模式。转储地址位置在每一帧上从 DCMIPP_P0PPM0AR1 寄存器切换到 DCMIPP_P0PPM0AR2 寄存器。

位 15:13 **LINEMULT[2:0]**：LINE 事件和中断的当前捕获完成行数 (Current amount of capture completed lines for LINE event and interrupt)

000：每行后的事件
001：两行后的事件
010：四行后的事件
011：八行后的事件
100：十六行后的事件
101：32 行后的事件
110：64 行后的事件
111：128 行后的事件

位 12 保留，必须保持复位值。

位 11 **OELS**：当前奇数/偶数行选择（行选择开始）(Current odd/even line select (line select start))

此位与 LSM 字段 (LSM = 1) 一起使用
0：接口在帧开始后捕获第一行，丢弃第二行
1：接口在帧开始后捕获第二行，丢弃第一行

位 10 **LSM**：当前行选择模式 (Current Line select mode)

0：接口捕获所有接收到的行
1：接口每两行捕获一行

位 9 **OEBS**：当前奇数/偶数字节选择（字节选择开始）(Current odd/even byte select (byte select start))

此位与 BSM 字段 (BSM ≠ 00) 一起使用
0：接口从帧/行开始捕获第一个数据（字节或双字节），丢弃第二个数据
1：接口从帧/行开始捕获第二个数据（字节或双字节），丢弃第一个数据

位 8:7 **BSM[1:0]**：当前字节选择模式 (Current Byte select mode)

00：接口捕获所有接收到的数据
01：接口每两个数据捕获一个数据
10：接口每四个字节捕获一个字节
11：接口每四个字节捕获两个字节
模式 10 和 11 仅适用于 DCMIPP_PRCR 寄存器中的 EDM [2:0] = 000。

位 6 保留，必须保持复位值。

位 5 **PAD**：单色和原始拜耳 10/12/14 bpp 的当前填充模式：MSB 与 LSB 对齐 (Current Pad mode for monochrome and raw Bayer 10/12/14 bpp: MSB vs. LSB alignment)

0：按 LSB 对齐（在 MSB 上填充空位），以便向后兼容以前的 DCMI
1：按 MSB 对齐（在 LSB 上填充空位），以便软件或 GPU 更轻松地处理

位 4:0 保留，必须保持复位值。

30.10.38 DCMIPP Pipe0 当前像素打包器内存 0 地址寄存器 1 (DCMIPP_P0CPPM0AR1)

DCMIPP Pipe0 current pixel packer Memory0 address register 1

偏移地址：0x7C4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M0A[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M0A[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **M0A[31:0]**：内存 0 地址 (Memory0 address)
数据写入的当前内存区域 0 的基址。假定它是 16 的倍数，因此其位 3:0 始终为 0x0。

30.10.39 DCMIPP Pipe0 当前像素打包器内存 0 地址寄存器 2 (DCMIPP_P0CPPM0AR2)

DCMIPP Pipe0 current pixel packer Memory0 address register 2

偏移地址：0x7C8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M0A[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M0A[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **M0A[31:0]**：内存 0 地址 (Memory0 address)
数据写入的当前内存区域 0 的基址。假定它是 16 的倍数，因此其位 3:0 始终为 0x0。

30.10.40 DCMIPP 硬件配置寄存器 2 (DCMIPP_HWCFGR2)

DCMIPP hardware configuration register 2

偏移地址：0xFEC

复位值：0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	STV	Res.	Res.	Res.	TPG	Res.	Res.	Res.	MCU	Res.	Res.	CDEC2	DEC1
			r				r				r			r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	ADD MOD	Res.	Res.	Res.	PROC CLK	Res.	Res.	Res.	DBMFT	Res.	VPFT[2:0]		
			r				r				r		r	r	r

位 31:29 保留，必须保持复位值。

位 28 **STV**：统计版本 (Statistic version)
0：已实现统计提取（带三个累加器）
1：已实现统计和直方图提取

位 27:25 保留，必须保持复位值。

位 24 **TPG**：测试模式发生器 (Test pattern generator)
0：未实现 TPG
1：已实现 TPG

位 23:21 保留，必须保持复位值。

位 20 **MCU**：宏块单元作为像素格式 (Macroblock unit as pixel format)
0：未实现 MCU
1：已实现 MCU

位 19:18 保留，必须保持复位值。

位 17 **DEC2**：降采样前在 Pipe2 上抽取 (Decimation on Pipe2 before downsize)
0：未实现抽取
1：已实现抽取

位 16 **DEC1**：降采样前在 Pipe1 上抽取 (Decimation on Pipe1 before downsize)
0：未实现抽取
1：已实现抽取

位 15:13 保留，必须保持复位值。

位 12 **ADDMOD**：支持通过地址取模计算来访问流式传输中的小缓冲区 (Address modulo computation to access a small buffer in streaming featured)
0：未实现地址取模
1：已实现地址取模

位 11:9 保留，必须保持复位值。

位 8 **PROCCLK**：支持将处理时钟与 AXI 时钟相连 (Processing clock linked to AXI clock featured)
0：独立处理时钟未连接到 AXI
1：处理时钟直接为 AXI 时钟

位 7:5 保留，必须保持复位值。

位 4 **DBMFT**：支持双缓冲区模式 (Double buffer mode featured)

位 3 保留，必须保持复位值。

位 2:0 **VPFT[2:0]**：虚拟流水线功能

30.10.41 DCMIPP 硬件配置寄存器 1 (DCMIPP_HWCFGR1)

DCMIPP hardware configuration register 1

偏移地址：0xFF0

复位值：0x0000 0110

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ROI2NB[3:0]				ROI1NB[3:0]				Res.	Res.	PLANARFT[1:0]		Res.	Res.	Res.	RB2 RGB
r	r	r	r	r	r	r	r			r	r				r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DSP2 FT	DSP1 FT	Res.	Res.	Res.	IPPLUG CFG	Res.	Res.	PIPENB[1:0]		Res.	Res.	Res.	CSIFT
		r	r				r			r	r				r

位 31:28 **ROI2NB[3:0]**：Pipe2 的 ROI 数量 (Number of ROIs for Pipe2)

位 27:24 **ROI1NB[3:0]**：Pipe1 的 ROI 数量 (Number of ROIs for Pipe1)

位 23:22 保留，必须保持复位值。

位 21:20 **PLANARFT[1:0]**：Pipe1 的缓冲区功能 (Buffer features for Pipe1)

00：共面
01：半平面
10：保留
11：完全平面（每个分量一个缓冲区）

位 19:17 保留，必须保持复位值。

位 16 **RB2RGB**：原始拜耳转 RGB 功能（去马赛克）(Raw Bayer to RGB feature (demosaicer))

位 15:14 保留，必须保持复位值。

位 13 **DSP2FT**：像素 Pipe2 的下采样功能 (Down-sampling feature for the pixel Pipe2)

位 12 **DSP1FT**：像素 Pipe1 的下采样功能 (Down-sampling feature for the pixel Pipe1)

位 11:9 保留，必须保持复位值。

位 8 **IPPLUGCFG**：IP-Plug 配置 (IP-Plug configuration)

位 7:6 保留，必须保持复位值。

位 5:4 **PIPENB[1:0]**：流水线数量 (Number of pipes)

位 3:1 保留，必须保持复位值。

位 0 **CSIFT**：符合 CSI2 主机协议 (CSI2 host protocol compliant)

30.10.42 DCMIPP 版本寄存器 (DCMIPP_VERR)

DCMIPP version register

偏移地址：0xFF4

复位值：0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：DCMIPP 主版本 (DMA major revision)

位 3:0 **MINREV[3:0]**：DCMIPP 次版本 (DMA minor revision)

30.10.43 DCMIPP 标识寄存器 (DCMIPP_IPIDR)

DCMIPP identification register

偏移地址：0xFF8

复位值：0x0016 2021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **IDR[31:0]**：并行摄像头接口 (DCMI) 和可选像素处理 (PP) (Paralle camera interface (DCMI) and optional pixel processing (PP))

30.10.44 DCMIPP 大小标识寄存器 (DCMIPP_SIDR)

DCMIPP size identification register

偏移地址：0xFFC

复位值：0xA3C5 DD04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：4 KB 解码空间 (4-Kbyte decoding space)

30.11 DCMIPP 寄存器映射

表 260. DCMIPP 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	DCMIPP_IPGR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	QOS_MODE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEMORYPAGE[2:0]	
	Reset value								0																						0	1	0
0x004	DCMIPP_IPGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSTART
	Reset value																																0
0x008	DCMIPP_IPGR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDLE
	Reset value																																1
0x00C-0x018	Reserved	Reserved																															
0x01C	DCMIPP_IPGR8	IPPID[7:0]								Res.	Res.	Res.	ARCHIID[4:0]					Res.	Res.	Res.	REVID[4:0]					Res.	Res.	DID[5:0]					
	Reset value	1	0	1	0	1	0	1	0				0	0	1	0	0				0	0	0	1	1			0	1	0	1	0	0
0x020	DCMIPP_IPC1R1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OTR[1:0]		Res.	Res.	Res.	Res.	Res.	TRAFFIC[2:0]		
	Reset value																							0	0						0	1	1
0x024	DCMIPP_IPC1R2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WLRU[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value													0	0	0	1																
0x028	DCMIPP_IPC1R3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DPREGEND[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DPREGSTART[5:0]					
	Reset value											1	1	1	1	1	1											0	0	0	0	0	0
0x02C to 0x0FC	Reserved	Reserved																															
0x038	DCMIPP_IPC2R3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DPREGEND[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DPREGSTART[5:0]					
	Reset value											1	1	1	1	1	1											0	0	0	0	0	0
0x048	DCMIPP_IPC3R3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DPREGEND[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DPREGSTART[5:0]					
	Reset value											1	1	1	1	1	1											0	0	0	0	0	0
0x058	DCMIPP_IPC4R3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DPREGEND[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DPREGSTART[5:0]					
	Reset value											1	1	1	1	1	1											0	0	0	0	0	0
0x104	DCMIPP_PRCR	Res.	Res.	Res.	Res.	Res.	SWAPBITS	SWAPCYCLES	Res.	FORMAT[7:0]								Res.	ENABLE	Res.	EDM[2:0]			Res.	Res.	VSPOL	HSPOL	PCKPOL	ESS	Res.	Res.	Res.	Res.
	Reset value						0	0		0	0	0	0	0	0	0	0		0		0	0	0			0	0	0	0				
0x108	DCMIPP_PRESCR	FEC[7:0]								LEC[7:0]								LSC[7:0]								FSC[7:0]					
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10C	DCMIPP_PRESUR	FEU[7:0]								LEU[7:0]								LSU[7:0]								FSU[7:0]					
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x110 to 0x1F0	Reserved	Reserved																															

表 260. DCMIPP 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1F4	DCMIPP_PRIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERRIE	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																										0						
0x1F8	DCMIPP_PRSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VSYNC	HSYNC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERRF	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value															1	1										0						
0x1FC	DCMIPP_PRFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CERRF	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																										0						
0x200	DCMIPP_CMHWCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCSI	CPAR	CPIPES [1:0]		Res.	Res.
	Reset value																											0	1	0	1		
0x204	DCMIPP_CMCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFC	Res.	Res.	Res.	Res.
	Reset value																												0				
0x208	DCMIPP_CMFRCR	FRMCNT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20C-0x3EC	Reserved	Reserved																															
0x3F0	DCMIPP_CMIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	P0OVRIE	P0LIMITIE	Res.	Res.	Res.	Res.	P0VSYNCIE	P0FRAMEIE	P0LINEIE	Res.	PRERRIE	ATXERRIE	Res.	Res.	Res.	Res.	Res.
	Reset value																0	0					0	0	0		0	0					
0x3F4	DCMIPP_CMSR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	P0CPTACT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRVSYNC	PRHSYNC
	Reset value																0															1	1
0x3F8	DCMIPP_CMSR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	P0OVRF	P0LIMITF	Res.	Res.	Res.	Res.	P0VSYNCF	P0FRAMEF	P0LINEF	Res.	PRERRF	ATXERRF	Res.	Res.	Res.	Res.	Res.
	Reset value																0	0					0	0	0		0	0					
0x3FC	DCMIPP_CMFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CP0OVRF	CP0LIMITF	Res.	Res.	Res.	Res.	CP0VSYNCF	CP0FRAMEF	CP0LINEF	Res.	CPRERRF	CATXERRF	Res.	Res.	Res.	Res.	Res.
	Reset value																0	0					0	0	0		0	0					
0x400	DCMIPP_P0HWCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CVP	Res.	CROI [3:0]				CGM[1:0]		CRB [1:0]		DBM	CDS	CPLA[1:0]	
	Reset value																			0		0	0	0	0	0	0	0	0	1	0	0	0
0x404	DCMIPP_P0FSCR	PIPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0																															
0x408-0x4FC	Reserved																																

表 260. DCMIPP 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x500	DCMIPP_P0FCTCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPTREQ	CPTMODE	FRATE [1:0]	
	Reset value																													0	0	0	0
0x504	DCMIPP_P0SCSTR	Res.	Res.	Res.	Res.	VSTART[11:0]												Res.	Res.	Res.	Res.	HSTART[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x508	DCMIPP_P0SCSZR	ENABLE	POSNEG	Res.	Res.	VSIZE[11:0]												Res.	Res.	Res.	Res.	HSIZE[11:0]											
	Reset value	0	0			0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x50C-0x5AC	Reserved	Reserved																															
0x5B0	DCMIPP_P0DCCNTR	Res.	Res.	Res.	Res.	Res.	Res.	CNT[25:0]																									
	Reset value							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5B4	DCMIPP_P0DCLMTR	ENABLE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LIMIT[23:0]																							
	Reset value	0								1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x5B8 - 0x5BC	Reserved	Reserved																															
0x5C0	DCMIPP_P0PPCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBM	LINEMULT [2:0]			Res.	OELS	LSM	OEBS	BSM[1:0]		Res.	PAD	Res.	Res.	Res.	Res.	Res.
	Reset value																0	0	0	0		0	0	0	0	0		0					
0x5C4	DCMIPP_P0PPM0AR1	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5C8	DCMIPP_P0PPM0AR2	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5CC-0x5F0	Reserved	Reserved																															
0x5F4	DCMIPP_P0IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVRIE	LIMITIE	Res.	Res.	Res.	VSYNCIE	FRAMEIE	LINEIE
	Reset value																									0	0				0	0	0
0x5F8	DCMIPP_P0SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPTACT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVRF	LIMITF	Res.	Res.	Res.	VSYNCF	FRAMEF	LINEF
	Reset value								0																	0	0				0	0	0
0x5FC	DCMIPP_P0FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COVRF	CLIMITF	Res.	Res.	Res.	CVSYNCF	CFRAMEF	CLINEF
	Reset value																									0	0				0	0	0
0x600-0x6FC	Reserved	Reserved																															
0x700	DCMIPP_P0CFCTCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPTREQ	CPTMODE	FRATE [1:0]	
	Reset value																													0	0	0	0

表 260. DCMIPP 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x704	DCMIPP_P0CSCSTR	Res.	Res.	Res.	Res.	VSTART[11:0]												Res.	Res.	Res.	Res.	HSTART[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x708	DCMIPP_P0CSCSZR	ENABLE	POSNEG	Res.	Res.	VSIZE[11:0]												Res.	Res.	Res.	Res.	HSIZE[11:0]											
	Reset value	0	0			0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x70C-0x7BC	Reserved	Reserved																															
0x7C0	DCMIPP_P0CPPCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBM	LINEMULT[2:0]			Res.	OELS	LSM	OEBS	BSM[1:0]		Res.	PAD	Res.	Res.	Res.	Res.	Res.
	Reset value																0	0	0	0		0	0	0	0	0		0					
0x7C4	DCMIPP_P0CPPM0AR1	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x7C8	DCMIPP_P0CPPM0AR2	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x7CC - 0xFE8	Reserved	Reserved																															
0xFEC	DCMIPP_HWCFGR2	Res.	Res.	Res.	STV	Res.	Res.	Res.	TPG	Res.	Res.	Res.	MCU	Res.	Res.	DEC2	DEC1	Res.	Res.	Res.	ADDMOD	Res.	Res.	Res.	PROCCLK	Res.	Res.	Res.	DBMFT	Res.	VPFT[2:0]		
	Reset value				0				0				0			0	0				0				0				1		0	0	0
0xFF0	DCMIPP_HWCFGR1	ROI2NB[3:0]				ROI1NB[3:0]				Res.	Res.	PLANARTF[1:0]		Res.	Res.	Res.	RB2RGB	Res.	Res.	DSP2FT	DSP1FT	Res.	Res.	Res.	IPPLUGCFG	Res.	Res.	PIPENB[1:0]		Res.	Res.	Res.	CSIFT
	Reset value	0	0	0	0	0	0	0	0			0	0				0			0	0				1			0	1				0
0xFF4	DCMIPP_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	0	0	1	0	0	0	0
0xFF8	DCMIPP_IPIDR	IDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1
0xFFC	DCMIPP_SIDR	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	1	0	0

31 LCD-TFT 显示控制器 (LTDC)

31.1 LTDC 简介

LTDC（液晶显示器——薄膜晶体管）显示器控制器提供并行数字 RGB（红色、绿色、蓝色）以及水平同步、垂直同步、像素时钟和数据使能信号，这些信号直接输出到不同接口的 LCD-TFT 面板。

31.2 LTDC 主要特性

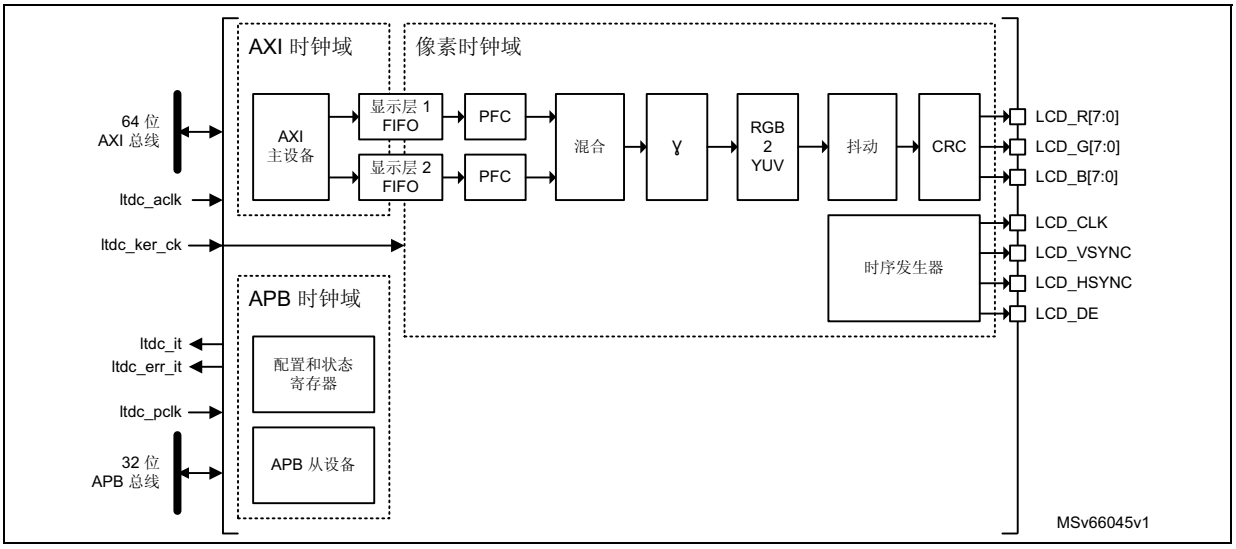
- 2 个输入图层混合在一起构成显示画面
- 从任意输入尺寸和位置裁剪图层
- 多种输入像素格式：
  - 预定义 ARGB，有 7 种格式：ARGB8888、ABGR8888、RGBA8888、BGRA8888、RGB565、BGR565、RGB888packed
  - 灵活的 ARGB 允许任意宽度和位置的 A、R、G、B 分量
  - 预定义 YUV，有 3 种格式：YUV422-1L（FourCC：YUYV，交错），YUV420-2L（FourCC：NV12，半平面），YUV420-3L（FourCC：Yxx，全平面），分量顺序灵活
- 支持查色表 (CLUT)，每图层高达 256 种颜色（256x24 位）
- 透明色键
- 合成窗口位置和尺寸灵活的输出显示
- 以灵活的图层顺序和透明度值（每像素或恒定）进行混合
- 底层背景色
- 使用非线性可配置表进行伽马校正
- 支持抖动，输出时每分量位数更少（2 位伪随机抖动）
- 支持极性反转，实现 HSync、VSync、和DataEnable 输出
- 以 RGB888 24 bpp 或 YUV422 16 bpp 格式输出
- 基于 7 个不同事件的中断
- AXI 主接口，支持有效的长字节突发传输（64 或 128 字节）

注： *上述 FourCC 像素格式在 www.fourcc.org 中有介绍。*

31.3 LTDC 功能描述

31.3.1 LTDC 框图

图 268. LTDC 框图



31.3.2 LTDC 引脚和内部信号

表 261. LTDC 引脚

引脚名称	引脚类型	说明
LCD_CLK	输出	时钟输出
LCD_HSYNC		水平同步
LCD_VSYNC		垂直同步
LCD_DE		非数据使能
LCD_R[7:0]		8 位红色数据（如果输出格式为 YUV422，则为 U/V 分量）
LCD_G[7:0]		8 位绿色数据（如果输出格式为 YUV422，则为 Y 分量）
LCD_B[7:0]		8 位蓝色数据

必须通过用户应用配置 LTDC 引脚。未使用的引脚可用于其他用途。

使用比默认 24 bpp 更少的引脚

LTDC 默认在 24 个引脚上输出像素 (RGB888)。但用户使用更少的线就可以连接显示面板，例如 16 根线 (RGB565) 或 18 根线 (RGB666)。这种情况下，R（或 G、B）线必须连接到 LTDC R（或 G、B）输出引脚的 MSB 上（参见*第 31.3.2 节*）。

示例：LTDC 与 RGB565 16 位显示面板连接。显示面板 R[4:0]、G[5:0] 和 B[4:0] 线输入引脚必须连接到 LTDC_R[7:3]、LCD_G[7:2] 和 LCD_B[7:3] 输出引脚。

31.3.3 LTDC 复位和时钟

LTDC 使用以下时钟域：

- AXI 时钟域
  该时钟域包含 LTDC AXI 主接口，该接口用于将数据从内存传输到显示层的 FIFO 和帧缓冲区配置寄存器中。
- APB 时钟域
  该时钟域包含全局配置寄存器和中断寄存器。
- 像素时钟域
  该时钟域包含像素数据生成、层配置寄存器以及 LTDC 接口信号发生器。LCD_CLK 输出必须按照面板要求配置。

注： *要正常工作，必须提供所有时钟以允许读写操作的时钟域交叉。*

下表列出了用于存储寄存器值的时钟域。

表 262. 各个寄存器的时钟域

LTDC 寄存器	时钟域
LTDC_LxCR	ltdc_aclk
LTDC_LxCFBAR	
LTDC_LxCFBLR	
LTDC_LxCFBLNR	
LTDC_SRCR	ltdc_pclk
LTDC_IER	
LTDC_ISR	
LTDC_ICR	
LTDC_SSCR	像素时钟 (LCD_CLK)
LTDC_BPCR	
LTDC_AWCR	
LTDC_TWCR	
LTDC_GCR	
LTDC_BCCR	
LTDC_LIPCR	
LTDC_CPSR	
LTDC_CDSR	
LTDC_LxWHPCR	
LTDC_LxWVPCR	

表 262. 各个寄存器的时钟域（续）

LTDC 寄存器	时钟域
LTDC_LxCKCR	像素时钟 (LCD_CLK)
LTDC_LxPFCR	
LTDC_LxCACR	
LTDC_LxDCCR	
LTDC_LxBFCR	
LTDC_LxCLUTWR	

访问 LTDC 寄存器时，APB 总线在访问期间会停止一段指定的时间，如下表所详述。

表 263. LTDC 寄存器访问和更新持续时间

访问类型	AXI 寄存器时钟域	APB 寄存器时钟域	像素寄存器时钟域
读	7 x ltdc_pclk + 5 x ltdc_aclk	7 x ltdc_pclk	7 x ltdc_pclk + 5 x ltdc_ker_clk
写	6 x ltdc_pclk + 5 x ltdc_aclk	6 x ltdc_pclk	6 x ltdc_pclk + 5 x ltdc_ker_clk

LTDC 可通过设置 RCC 中的相应位来复位，这可以复位三个时钟域。

31.4 LTDC 配置参数

LTDC 配置参数按照*图 268* 所示的像素流顺序排列，从输入（读取像素的 AXI 主设备）开始，到像素输出结束。

通用控制（例如中断）列在最后。

31.4.1 AXI 主设备

AXI 主设备总线执行对内存的访问，并由内部逻辑驱动。LTDC 内部的 FIFO 用于两个容量相当大的图层，以应对相当长的外部内存延迟。

LDTC 生成长突发传输（128 字节），可提高外部内存效率。为了提高细粒度，可以将最大突发传输的大小减小到 64 字节，但这种做法并不推荐。

LTDC_LxBLCR 中的参数

BL：突发传输长度，以 64 位字为单位，默认为 0，表示最大大小

默认配置

复位后，在 BL = 0 的情况下，默认的 AXI 突发传输长度会设置为其最大值。

31.4.2 输入图层定义和裁剪

LTDC 支持两个图层。每个图层的定义方式为：

- 通过其在内存中的基址定义

  如果缓冲区是 YUV 半平面或全平面，则分别有两个和三个基址。第一个缓冲区称为主缓冲区，存储亮度数据，其他缓冲区则称为辅助缓冲区，存储色度数据。
- 通过内存中的一个维度来定义
  - 宽度定义为其间距（或地址跳转或跨距）：一个行的开始与下一行的开始之间的字节数
  - 不需要定义高度
- 通过从内存中裁剪并读取的像素矩形的维度来定义
  - 左上角起始位置：根据基址设置为左上角像素所在位置的确切字节
  - 宽度，每行要读取的字节数

    - 如果字节数少于所需数量，将生成 FIFO 下溢中断。

    - 如果字节数多于所需数量，将丢弃无用数据。
  - 高度，每帧要读取的行数

注： *如果帧的子部分被裁剪以进行显示，则间距和每行字节数可以不同。如果显示完整帧，则间距和行数相同。*

参数

- LTDC_LxCR 中的 LEN：激活该图层。
- LTDC_LxCFBAR 中的 CFBADD：主缓冲区基址，用于 ARGB 或 Y 缓冲区
- LTDC_LxCFBLR 中的 CFBLL：主缓冲区每行的字节数
- LTDC_LxCFBLR 中的 CFBP：主缓冲区连续两行之间的间距
- LTDC_LxCFBLNR 中的 CFBLNBR：主缓冲区每帧的行数

特别是对于多平面图层，必须配置以下附加寄存器（仅用于 YUV420 半平面和全平面像素格式）：

- LTDC_L1AFBA0R 中的 AFBADD0：辅助地址 0，用于 UV（半平面时）或 V（全平面时）
- LTDC_L1AFBA1R 中的 AFBADD1：辅助地址 1，用于 U（全平面时）
- LTDC_L1AFBLR 中的 AFBLL：辅助地址 0 和 1，每行的字节数
- LTDC_L1AFBLR 中的 AFBP：辅助 0 和 1 缓冲区的间距
- LTDC_L1AFBLNR 中的 AFBLNBR：辅助 0 和 1 每帧行数

配置示例

- 采用 RGB565 格式（每像素 2 个字节）且宽度为 256 像素的帧缓冲区（每行总字节数为 256 * 2 = 512）需要向此寄存器写入值 0x02000207（其中，间距 = 行长）。
- 采用 RGB888 格式（每像素 3 个字节）且宽度为 320 像素的帧缓冲区（每行总字节数为 320 * 3 = 960）需要向此寄存器写入值 0x03C003C7（其中，间距 = 行长）。
- 采用 CbCr 格式（每像素 2 个字节）且宽度为 256 像素的帧缓冲区（每行总字节数为 256 * 2 = 512）需要向此寄存器写入值 0x02000207（其中，间距 = 行长）。
- 采用 Cb 格式（每像素 1 个字节）且宽度为 320 像素的帧缓冲区（每行总字节数为 320 * 1 = 320）需要向此寄存器写入值 0x01400147（其中，间距 = 行长）。

31.4.3 输入像素格式

LTDC 从内存缓冲区读取像素数据，并根据每层定义的输入像素格式，将其映射到内部 LTDC 格式。

像素格式分组为：

- 预定义的 ARGB 格式

  ARGB8888、ABGR8888、RGBA8888、BGRA8888、RGB565、BGR565、RGB888pack

- 灵活 ARGB 格式

  允许配置每个 A、R、G、B 分量的宽度和位置，并用于将上述列表扩展到任意多分量共面 ARGB 像素格式。

- YUV 格式

  用于读入 422 共面、420 半平面或 420 全平面的 YUV 格式，并将 YUV 颜色转换为 RGB 颜色

  YUV 选项仅适用于图层 1，只有 422 共面适用于图层 2。

注： *LTDC 硬件流水线使用每分量 8 位：当读取和映射像素格式时，每个分量少于 8 位。相应分量通过重新复制其 MSB 来填充 LSB，从而扩展为 8 位。这主要适用于 RGB565。*

例如，一个 5 位分量（命名为：43210）会产生以下映射到流水线中的 8 位：43210432。一个 3 位输入（命名为 210）则会产生 21021021。

使用预定义 ARGB 格式时的参数

- LTDC_LxPFCR 中的 PF：0 到 6，从七种预定义像素格式中选择一种

使用灵活 ARGB 格式时的参数

- LTDC_LxPFCR 中的 PR：7 可激活灵活像素格式类型
- LTDC_LxFPF1R 中的 PSIZE：定义每像素字节数（通常为 1、2、3 或 4）
- LTDC_LxFPF0R 中的 APOS/RPOS：定义 32 位字中的 Alpha/Red 位置
- LTDC_LxFPF1R 中的 BPOS/GPOS：定义 32 位字中的 Blue/Green 位置
- LTDC_LxFPF0R 中的 ALEN/RLEN：定义透明度/红色分量的位数
- LTDC_LxFPF1R 中的 BLEN/GLEN：定义蓝色/绿色分量的位数

  示例：

  RGB565 像素格式定义如下：

  – 透明度：APOS = 0，ALEN = 0
  – 红色：RPOS = 11，RLEN = 5
  – 绿色：GPOS = 5，GLEN = 6
  – 蓝色：BPOS = 0，BLEN = 5

  BGRA8888 32 bpp 像素格式设置如下（LTDC_LxFPF0R 和 LTDC_LxFPF1R 的默认复位值）：

  – 透明度：APOS = 0，ALEN = 8
  – 红色：RPOS = 8，RLEN = 8
  – 绿色：GPOS = 16，GLEN = 8
  – 蓝色：BPOS = 24，BLEN = 8

使用 YUV 格式时的参数

YUV 像素格式在*第 31.4.4 节：YUV 平面*中定义并详细进行了介绍，取代此处的任何设置。

下表描述了预定义 ARGB 像素格式的内存映射。

表 264. 像素数据映射与颜色格式的关系

ARGB8888 (PF = 000)			
@+3 A_x[7:0]	@+2 R_x[7:0]	@+1 G_x[7:0]	@ B_x[7:0]
@+7 A_{x+1}[7:0]	@+6 R_{x+1}[7:0]	@+5 G_{x+1}[7:0]	@+4 B_{x+1}[7:0]
ABGR8888 (PF = 001)			
@+3 A_x[7:0]	@+2 B_x[7:0]	@+1 G_x[7:0]	@ R_x[7:0]
@+7 A_{x+1}[7:0]	@+6 B_{x+1}[7:0]	@+5 G_{x+1}[7:0]	@+4 R_{x+1}[7:0]
RGBA8888 (PF = 010)			
@+3 R_x[7:0]	@+2 G_x[7:0]	@+1 B_x[7:0]	@ A_x[7:0]
@+7 R_{x+1}[7:0]	@+6 G_{x+1}[7:0]	@+5 B_{x+1}[7:0]	@+4 A_{x+1}[7:0]
BGRA8888 (PF = 011)			
@+3 B_x[7:0]	@+2 G_x[7:0]	@+1 R_x[7:0]	@ A_x[7:0]
@+7 B_{x+1}[7:0]	@+6 G_{x+1}[7:0]	@+5 R_{x+1}[7:0]	@+4 A_{x+1}[7:0]
RGB565 (PF = 100)			
@+3 R_{x+1}[4:0] G_{x+1}[5:3]	@+2 G_{x+1}[2:0] B_{x+1}[4:0]	@+1 R_x[4:0] G_x[5:3]	@ G_x[2:0] B_x[4:0]
@+7 R_{x+3}[4:0] G_{x+3}[5:3]	@+6 G_{x+3}[2:0] B_{x+3}[4:0]	@+5 R_{x+2}[4:0] G_{x+2}[5:3]	@+4 G_{x+2}[2:0] B_{x+2}[4:0]
BGR565 (PF = 101)			
@+3 B_{x+1}[4:0] G_{x+1}[5:3]	@+2 G_{x+1}[2:0] R_{x+1}[4:0]	@+1 B_x[4:0] G_x[5:3]	@ G_x[2:0] R_x[4:0]
@+7 B_{x+3}[4:0] G_{x+3}[5:3]	@+6 G_{x+3}[2:0] R_{x+3}[4:0]	@+5 B_{x+2}[4:0] G_{x+2}[5:3]	@+4 G_{x+2}[2:0] R_{x+2}[4:0]
RGB888 (PF = 110)			
@+3 B_{x+1}[7:0]	@+2 R_x[7:0]	@+1 G_x[7:0]	@ B_x[7:0]
@+7 G_{x+2}[7:0]	@+6 B_{x+2}[7:0]	@+5 R_{x+1}[7:0]	@+4 G_{x+1}[7:0]
灵活格式 (PF = 111) 未在表中列出，该格式允许以上任何映射。			

31.4.4 YUV 平面

该操作处理 YUV 平面格式，从可能的多个不同缓冲区（Y、Cb、Cr）读取数据到单个像素信息。该功能适用于图层 1，仅共面时可用于图层 2。

支持的平面格式有：

- YUV422 共面（一个缓冲区）
- YUV420 半平面（两个缓冲区）
- YUV420 全平面（三个缓冲区）

根据其配置，LTDC 读取相关的亮度和色度缓冲区，将所需的色度分量放大到 YUV444 像素，并将颜色从 YUV 转换为 RGB（转换在下一段中描述）。

LTDC 允许内存中 Y 与 Cb/Cr 数据位置交换、Cb 与 Cr 交换或亮度字节序交换，从而允许上述三种格式的变体。它扩展了支持的像素格式数量。

参数

像素格式参数存储在 LTDC_LxPCR 中：

- YCEN：激活 YUV 模式（平面提取和颜色转换）。
- YCM：选择输入方案（共面为 422，半平面为 420，全平面为 420）。
- YF：定义是先提供 Y 还是 Cb/Cr（从 LSB）。
- CBF：定义是先提供 Cb 还是 Cr（从 LSB）。
- OF：定义是先提供奇数 Y 还是偶数 Y（从 LSB）。

使用半平面或全平面缓冲区时，必须为辅助（色度）缓冲区配置一些附加的基址，以及这些缓冲区的间距（请参见*第 31.4.2 节*）。

配置示例

YCbCr 422 像素格式的 OF、CBF、YF 位设置：

- OF = 0，CBF = 0，YF = 0 定义的缓冲字包含 Word[31:0] = Y1-Cb-Y0-Cr。
- OF = 1，CBF = 0，YF = 0 定义的缓冲字包含 Word[31:0] = Y0-Cb-Y1-Cr。
- OF = 0，CBF = 1，YF = 0 定义的缓冲字包含 Word[31:0] = Y0-Cr-Y1-Cb。
- OF = 0，CBF = 0，YF = 1 定义的缓冲字包含 Word[31:0] = Cb-Y1-Cr-Y0。
- OF = 1，CBF = 1，YF = 1 定义的缓冲字包含 Word[31:0] = Cr-Y0-Cb-Y1。

  这些设置也适用于 YCbCr 420 像素格式。

31.4.5 YUV 到 RGB 颜色转换

该操作处理 YUV 平面像素的 YUV 到 RGB 颜色转换。颜色转换会自动与 YUV 平面处理同时激活。

颜色转换基于半灵活 3x3 矩阵乘法：

- 由于一些固定系数（见下文），它仅限于 YUV 到 RGB 转换。
- 一些系数可配置，因而它可处理最常用的 YUV 参考标准（ITU-R BT.601 和 ITU-R BT.709）。
- 它将可能范围缩小的输入扩展到完整范围，从而支持输入像素的全动态与缩减动态（亮度定义为在 0 到 255 或在 16 到 235 之间）。

参数

- LTDC_LxPCR 中的 YCEN：激活颜色转换，与平面时的位相同。
- LTDC_LxPCR 中的 YREN：如果激活，则将缩减范围扩展至完整范围。
- LTDC_LxCYR0R 中的 CR2R：Cr 到红色的矩阵系数
- LTDC_LxCYR0R 中的 CB2B：Cb 到蓝色的矩阵系数。
- LTDC_LxCYR0R 中的 CR2G：Cr 到绿色的矩阵系数。
- LTDC_LxCYR0R 中的 CB2G：Cb 到绿色的矩阵系数

完整矩阵计算（将 YCbCr 转换为 RGB）定义如下：

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} CR2R & 256 & 0 \\ -CR2G & 256 & -CB2G \\ 0 & 256 & CB2B \end{pmatrix} \times \begin{pmatrix} Y \\ Cr - 128 \\ Cb - 128 \end{pmatrix}$$

CR2R、CB2B、CR2G 和 CB2G 系数以 10 位无符号定义：由于该操作处理 8 位分量，因此这些系数相当于定点的 2 位整数和 8 位小数。

CR2G 和 CB2G 系数配置为无符号正数，但在硬件计算中设为负数，如上面提及“-”的矩阵。

当亮度调整激活时（YREN = 1，此时假定 YCbCr 输入范围在 16 到 235/240 之间，RGB 在 0 到 255 之间），则 Y 输入值首先通过硬件调整为 Y’ = 255/219 * (Y - 16)。然后将这第一个结果作为输入插入到上述矩阵乘法中。

配置示例

要读取 ITU-R BT.601 图层并将其 YCbCr（假定默认的缩减动态范围在 16 到 235/240）转换为 RGB（范围在 0 到 255 的全动态），推荐系数为：

- YCEN = 1：激活转换
- YREN = 1：选择缩减动态范围
- CR2R = 409：为 ITU-R BT.601 设置的特定系数
- CR2G = 208
- CB2G = 100
- CB2B = 516

要读取 ITU-R BT.709 图层并将其 YCbCr（假定默认的缩减动态范围在 16 到 235/240）转换为 RGB（范围在 0 到 255 的全动态），建议将系数值配置为：

- YCEN = 1：激活转换
- YREN = 1：选择缩减动态范围
- CR2R = 459：以下特定系数集适用于 ITU-R BT.709
- CR2G = 136
- CB2G = 55
- CB2B = 541

31.4.6 水平或垂直镜像

LTDC 能够水平或垂直镜像其输入图层，因而可以从右向左进行读取，以及/或者从下向上进行读取。

这适用于共面（ARGB、YUV422）和多平面（YUV420）像素格式。

水平镜像的参数

要水平镜像一个图层，LTDC 必须配置为从右向左（反向）提取像素，并将图层基址设定在行的右端。

配置位字段如下：

- LTDC_LxCR 中的 HMEN：激活水平镜像。
- LTDC_LxCFBAR 中的 CFBADD：基址 + (宽度 - 1)（水平镜像时增加）
- LTDC_L1AFBA0R 中的 AFBADD0：基址 + (宽度 - 1)
- LTDC_L1AFBA1R 中的 AFBADD1：基址 + (宽度 - 1)

垂直镜像的参数

要垂直镜像一个图层，LTDC 必须配置为从底部开始读取该图层，并逐行向上读取。图层基址必须设定为最底行的起点，并且必须提供负的逐行间距。

基址的配置位字段为（全部设定在最底行的起点）：

- LTDC_LxCFBAR 中的 CFBADD：主基址
- LTDC_L1AFBA0R 中的 AFBADD0：辅助地址 0
- LTDC_L1AFBA1R 中的 AFBADD1：辅助地址 1
- LTDC_LxCFBAR 中的 CFBADD：基址 + (行数 - 1) * 间距（垂直镜像时增加）
- LTDC_L1AFBA0R 中的 AFBADD0：基址 + (行数 - 1) * 间距
- LTDC_L1AFBA1R 中的 AFBADD1：基址 + (行数 - 1) * 间距

间距的配置位字段为（全部设为负间距）：

- LTDC_LxCFBLR 中的 CFBP：-间距（垂直镜像的负的行间距）
- LTDC_L1AFBLR 中的 AFBP：-间距

31.4.7 默认图层颜色

每层具有 ARGB8888 格式的默认颜色，该颜色在定义的层窗口外使用或在层禁止时使用。该默认颜色可以配置为激活或不存在。

参数

默认图层颜色参数包括：

- LTDC_LxCR 中的 DCBEN：默认颜色激活
- LTDC_LxDCCR 中的 DCALPHA：默认 Alpha 透明度设置
- LTDC_LxDCCR 中的 DCRED：默认红色分量
- LTDC_LxDCCR 中的 DCGREEN：默认绿色分量
- LTDC_LxDCCR 中的 DCBLUE：默认蓝色分量

默认配置

默认情况下，在复位之后，默认图层颜色无效（DCBEN = 0 且 DCALPHA = 0）。

注： *即使层被禁止，也始终会执行图层混合。为避免在图层被禁止时显示默认颜色，必须将默认 Alpha 设置为 null (DCALPHA = 0)，或者禁止整个默认颜色（使用 DCBEN）。*

31.4.8 查色表 (CLUT)

LTDC 支持帧缓冲区，其中每个像素定义为颜色表的索引，从而实现一组 256 色的显示图像，但只有 8 bpp 图像存储在内存中。

CLUT 定义了 256 种颜色，并且必须使用最大 8 位像素格式 (通常是 L8 或 AL44) 索引。可以使用*第 31.4.3 节*中所述的灵活 RGB 方法提取此类格式。

CLUT 表对每个图层而言有所不同。它必须使用每个颜色索引一个 RGB888 值的方式进行初始化，且颜色索引对应于 CLUT 地址：

- 对于 L8，CLUT 必须加载 256 种颜色。
- 对于 AL44，CLUT 只能加载 16 种颜色。每个颜色地址必须通过将 4 位 L 通道重复为 8 位进行定义，具体如下：
  - 颜色 L4 = 0：从索引 = 0x00 查找其 RGB888。
  - 颜色 L4 = 1：从索引 = 0x11 查找其 RGB888。
  - 颜色 L4 = 2：从索引 = 0x22 查找其 RGB888。
  - ...
  - 颜色 L4 = 15：从索引 = 0xFF 查找其 RGB888。
- 对于 ARGB，蓝色分量用作 CLUT 索引。

参数

RGB888 值和颜色索引通过 LTDC_LxCLUTWR 中的以下位字段进行配置：

- CLUTADD 定义要更新的颜色索引。
- RED 定义该颜色索引的 8 位红色分量。
- GREEN 定义该颜色索引的 8 位绿色分量。
- BLUE 定义该颜色索引的 8 位蓝色分量。

31.4.9 透明色键

可以使用色键将像素定义为透明。如果像素的 RGB 分量等于定义的 RGB 色键，则该像素被视为透明，并且它的 Alpha 通道强制为 null。

强制设为透明的像素在进行到后续的混合阶段时为透明，并显示底层的像素。

色键定义为 RGB888 格式的值，在经过输入格式操作的处理（LSB 通过复制填充）和 CLUT 处理后，可与流动像素媲美。恒定的色键参数必须将这些预先处理考虑在内。

参数

色键参数如下：

- LTDC_LxCR 中的 CKEN 使能色键。
- LTDC_LxCKCR 中的 CKRED 定义 8 位 红色色键。
- LTDC_LxCKCR 中的 CKGREEN 定义 8 位绿色色键。
- LTDC_LxCKCR 中的 CKBLUE 定义 8 位蓝色色键。

配置示例

将中黄色（50% 红 + 50% 绿）用作透明色键。色键参数将以下内容纳入考量：

- ARGB8888 输入：中黄色输入为 0x808000。将 LTDC_LxCKCR 设置为 0x808000。
- RGB565 输入：中黄色输入为 0x8400。将 LTDC_LxCKCR 设置为 0x848200（由于输入 MSB 复制到缺少位的 LSB）。
- L8、AL88 或 AL44 的 CLUT 输入：将其中一个调色板条目设置为中黄色 0x808000，并将 LTDC_LxCKCR 设置为 0x808000。

31.4.10 显示合成 - 窗口化

在执行显示合成时，每个图层都可以被裁剪并定位到有效显示区域。窗口位置和大小是通过定义其左上角和右下角来配置的。

注： *位置是相对于内部时序发生器来定义的，必须考虑潜在的同步头、后沿尺寸和所选有效像素区域（如图 269 所示）。*

输入图层与可见窗口之间的映射按如下方式执行：

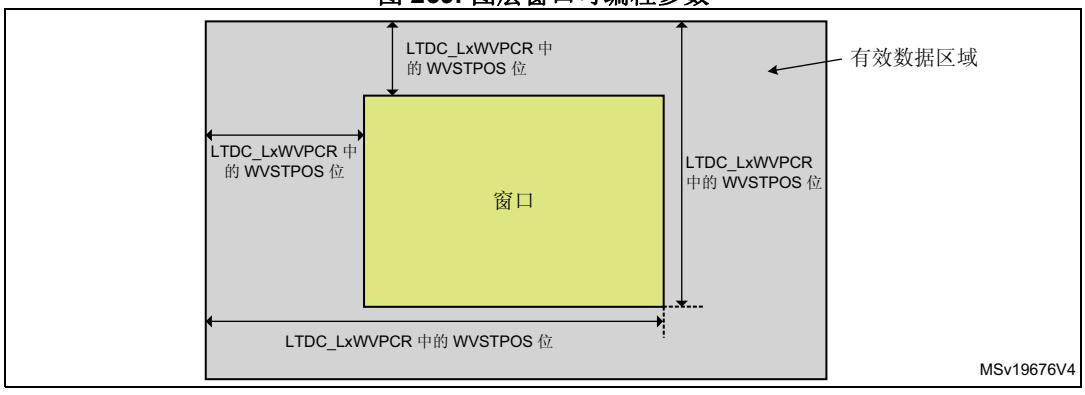
- 输入图层左上角的像素映射到该图层可见窗口的左上角。
- 如果可见窗口比输入图层小（宽度和高度均小于图层），则超出部分的输入像素会自动从流水线中丢弃。

参数：

以下参数是相对于同步的左上角来定义的：

- LTDC_LxWHPCR 中的 WHSTPOS：图层的第一个（左侧）可见像素
- LTDC_LxWHPCR 中的 WHSPPOS：图层的最后一个（右侧）可见像素
- LTDC_LxWVPCR 中的 WVSTPOS：图层的第一条（顶部）可见行
- LTDC_LxWVPCR 中的 WVSPPOS：图层的最后一条（底部）可见行

图 269. 图层窗口可编程参数



配置示例

- LTDC_BPCR 配置为 0x000E0005（AHBP[11:0] 为 0xE），LTDC_AWCR 配置为 0x028E01E5（AAW[11:0] 为 0x28E）。要配置大小为 630x460 的窗口的水平位置（有效数据区域中的水平起始偏移为 5 个像素）：
  - 图层窗口第一个像素：WHSTPOS[11:0] 必须编程为 0x14 (0xE + 1 + 0x5)。
  - 图层窗口最后一个像素：WHSPPOS[11:0] 必须编程为 0x28A。

- LTDC_BPCR 配置为 0x000E0005（AVBP[11:0] 为 0x5），LTDC_AWCR 配置为 0x028E01E5（AAH[11:0] 为 0x1E5）。要配置大小为 630x460 的窗口的垂直位置（有效数据区域中的垂直起始偏移为八行）：
  - 图层窗口第一行：WVSTPOS[11:0] 必须编程为 0xE (0x5 + 1 + 0x8)。
  - 图层窗口最后一行：WVSTPOS[11:0] 必须编程为 0x1DA。

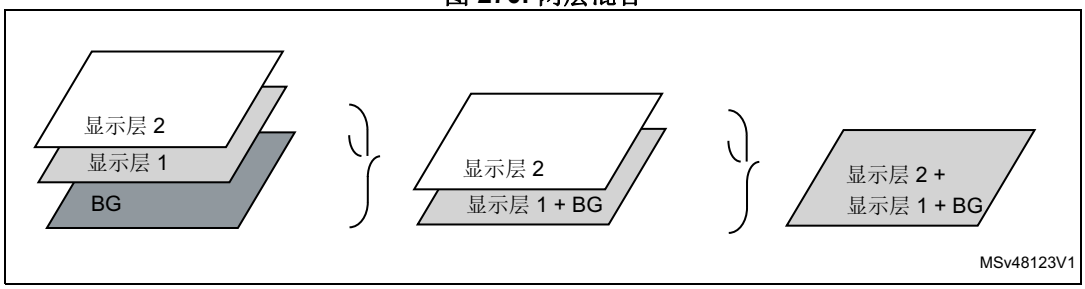
31.4.11 显示组成——混合

Alpha 混合阶段用于将 2 个输入图层在背景色上混合。

可以选择图层顺序（前景、中景和背景），对每层应用全局乘法 Alpha，并定义底层背景色。

混合计算从后向前顺序进行，如下图所示。

图 270. 两层混合



2 合 1 混合的一般公式为 CR = (BF1 x C1) + (BF2 x C2)，其中：

- CR：结果混合色
- BF1：当前图层的混合系数
- C1：当前图层的层色
- BF2：底层像素的混合系数
- C2：底层像素的层色

混合系数 BF1 和 BF2 可以定义为以下之一：

- 纯每像素 Alpha 值
- 每像素 Alpha 与全局每层 Alpha 的混合（混合为乘积）

参数

- LTDC_LxCACR 中的 CONSTA：当前层的全局乘积 Alpha 值
- LTDC_LxBFCR 中的 BOR：当前层的混合顺序
- LTDC_LxBFCR 中的 BF1：当前层的混合系数
- LTDC_LxBFCR 中的 BF2：底层的混合系数

背景色由 LTDC_BCCR 中的以下位字段定义：

- BCRED：背景红色分量
- BCGREEN：背景绿色分量
- BCBLUE：背景蓝色分量

注： *为安全起见，如果配置不一致，混合顺序将回退到固定顺序，图层 2 在前景，图层 1 在背景，然后是底层背景色。*

配置示例

仅使能层 1，BF1 配置为恒定 alpha。BF2 配置为 1——恒定 Alpha 在 LTDC_LxCACR 中编程的恒定 alpha 为 240 (0xF0)。因此，恒定 alpha 值为 240 / 255 = 0.94。C：当前层色为 128。Cs：背景色为 48。层 1 与背景色混合。
BC = 恒定 alpha x C + (1 - 恒定 Alpha) x Cs = 0.94 x 128 + (1 - 0.94) x 48 = 123。

默认配置

保持复位后的以下配置值，可以使所有有效图层可见并进行混合：

- CONSTA：保持复位后的 CONSTA = 255。
- BOR：保持复位后的 BOR（层）= LayerID-1。
- BF1：保持复位后的 BF1 = 6（默认混合 Alpha）。
- BF12：保持复位后的 BF2 = 7（默认混合 1-Alpha）。
- 背景色：保持为复位后的颜色（黑色），尽管任何颜色都可以。

31.4.12 伽马校正

伽马校正是一个非线性操作，可采用补偿预失真技术对像素进行调整，使之适合非线性显示。可以使用可配置的非线性转换独立调整 R、G、B 分量。

非线性转换在 LTDC 中实现为八个独立的线性插值段：所需的非线性转换必须映射到这八个段上，且八个段的形状在 LTDC 寄存器中配置。

参数

- LTDC_GCR 中的 GAMEN：激活伽马操作
- LTDC_GCCR 中的 ADDR：表配置，要写入的索引（256 个）
- LTDC_GCCR 中的 COMP：表配置，要写入的 8 位分量
- LTDC_GCCR 中的 REN：表配置，激活以写入红表
- LTDC_GCCR 中的 GEN：表配置，激活以写入绿表
- LTDC_GCCR 中的 BEN：表配置，激活以写入蓝表

注： *伽马校正使用八个插值段。唯一要写入的表索引位于以下索引：0, 32, 64, 96,...,224, 255。要为索引 255 提供的值包含伪索引 256 的值。*

配置示例

建议将伽马校正设置为预补偿伽马指数为 2.2 的像素：伽马校正以相同的方式补偿红、绿和蓝分量。这样一来，红、绿和蓝表的配置通常便会相同。

默认配置

伽马校正将通过复位后的默认配置和 GAMEN = 0 来禁止。

31.4.13 YUV 输出转换

默认的 RGB888 24 bpp 输出可转换为 YUV422 12 bpp 输出（共面），以减少比特率或接口宽度（如有必要）。

该操作可将 RGB 转换为 YUV（使用 ITU-R BT.601 或 BT.709 标准）并降低色度，以作为 YUV422 接口输出像素。

当使能转换时，Y 分量通过 LCD_G（绿色位）输出，而 Cb/Cr 分量通过 LCD_R（红色位）输出。

LTDC_EDCR 中的参数

- OCYEN：激活输出转换到 YUV422。
- OCSEL：选择颜色转换为 ITU-R BT.601 还是转换为 BT.709。
- OCYCO：定义先输出 Cb 还是 Cr。

配置必须遵循外部显示面板的预期。

默认配置

YUV 输出将通过复位后的默认配置和 OCYEN = 0 来禁止，从而允许输出默认格式 RGB888。

31.4.14 抖动

抖动操作允许将 RGB888 输出清晰地转换为缩减像素格式，如 RGB565 或 RGB666，从而避免没有抖动时通常会出现的平坦区域失真。

抖动基于 LFSR（多项式生成随机抖动）在每个显示分量的 LSB（在 RGB666 情况下为第 2 位）上增加伪随机噪声，使局部像素的平均值代表了完整的 8 分量值的信息内容。

使能 LTDC 后，LFSR 以第一个有效像素开始运行，并且即使在消隐周期内和抖动关闭时也保持运行状态。如果禁止 LTDC，LFSR 将复位。

LTDC_GCR 中的参数

- DEN：抖动使能
- DRW：显示的红色分量 LSB 的位标识（8 位）
- DGW：显示的绿色分量 LSB 的位标识（8 位）
- DBW：显示的蓝色分量 LSB 的位标识（8 位）

示例

从 RGB888 抖动为 RGB565（假设使用 RGB565 显示面板）：

- LTDC 通过引脚 LCD_R[7:0] 输出红色分量。
- 显示面板探测五个 MSB 位 LCD_R[7:3]。
- 要抖动的红色位是位 3（使用其后的两个 LSB 作为小数位）。
- DRW 必须配置为 3。
- DGW = 2，DBW = 3。

同样，抖动为 RGB666 需要 DRW = DGW = DBW = 2。

默认配置

抖动将通过复位后的默认配置和 DEN = 0 来禁止，从而允许输出默认格式 RGB888。

31.4.15 CRC 哈希

CRC 操作允许通过哈希所有帧像素的 R、G、B 分量来为每个帧提取一个 16 位的签名。

计算出的 CRC 可在 LTDC_CCRCR 中读取。

可配置预期的 CRC 值：如果计算的 CRC 与预期的 CRC 不匹配，则触发 CRC 中断。可进行系统测试，让 LTDC 运行数小时，只在发生帧失败时才发出警告。

参数

- LTDC_GCR 中的 CRCEN：激活 CRC 计算
- LTDC_ECRCR 中的 ECRC：预期的 16 位 CRC 值
- LTDC_CCRCR 中的 CCRC：计算出的 16 位 CRC 值

默认配置

CRC 哈希将通过复位后的默认配置和 CRCEN = 0 来禁止。

31.4.16 显示时序

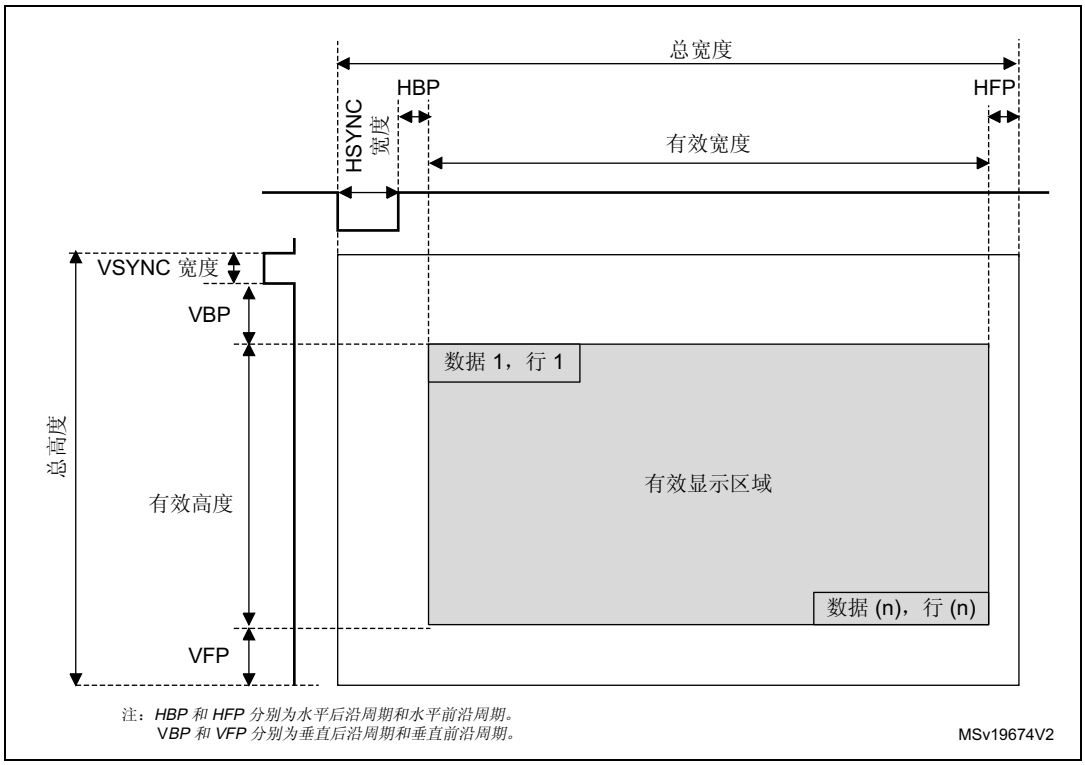
本节描述了显示时序的配置，以使 LTDC 符合外部显示面板的预期。

配置中定义了以下各项的水平和垂直持续时间：

- 同步周期（HSYNC 和 VSYNC）
- 后沿（HBP、VBP）
- 有效显示
- 前沿（HFP、VFP）

这些时序在下图中进行了描述。

图 271. LTDC 同步时序



LTDC 通过这些时序在有效显示周期以像素为单位进行采样，并生成输出同步信号（HSYNC、VSYNC）、像素数据和数据使能信号。

参数

以下配置的所有时序均为绝对时序，从显示开始位置计算，而不是每个区域的相对增量（或宽度或高度）。

所有这些位字段均表示每个区域的最后一个像素，因此要减 1。

- LTDC_SSCR 中的 HSYNC 和 VSYNC 宽度（水平和垂直同步宽度）：
  - HSW = HSYNC 宽度 - 1
  - VSW = VSYNC 宽度 - 1
- LTDC_BPCR 中的 HBP 和 VBP（水平和垂直同步后沿的最后一个像素/行）：
  - AHBP = HSYNC 宽度 + HBP - 1
  - AVBP = VSYNC 宽度 + VBP - 1
- LTDC_AWCR 中的有效宽度和有效高度（有效显示区的最后一个像素/行）：
  - AAW = HSYNC 宽度 + HBP + 有效宽度 - 1
  - AAW = VSYNC 宽度 + VBP + 有效高度 - 1
- LTDC_TWCR 中的总显示宽度（总显示时序的最后一个像素/行）：
  - TOTALW = HSYNC 宽度 + HBP + 有效宽度 + HFP - 1
  - TOTALH = VSYNC 高度 + VBP + 有效高度 + VFP - 1

注： *当 LTDC 使能时，生成的时序从位置 (X,Y) = (0,0) 开始，并立即生成垂直同步中的第一个水平同步像素，接着是后沿，然后是有效显示区域，最后是前沿。*

当 LTDC 禁止时，时序发生器模块复位为 X = 总宽度 - 1，Y = 总高度 - 1。它保存垂直同步阶段和 FIFO 清空之前的最后一个像素。因此，只有消隐数据会连续输出。

同步时序配置示例

假设有以下显示面板时序（从面板数据表中提取）：

- 水平和垂直同步宽度：0xA 像素，0x2 行
- 水平和垂直后沿：0x14 像素，0x2 行
- 有效宽度和有效高度 (320x240)：0x140 像素，0x0F0 行
- 水平和垂直前沿：0xA 像素，0x4 行

配置的位字段值变为：

- LTDC_SSCR 中的 HSW[11:0] = 0x9（水平同步 -1）。
- LTDC_SSCR 中的 VSH[11:0] = 0x1（垂直同步 -1）。
- LTDC_BPCR 中的 AHBP[11:0] = 0x1D（水平同步 + 后沿 -1）。
- LTDC_BPCR 中的 AVBP[11:0] = 0x3（垂直同步 + 后沿 -1）。
- LTDC_AWCR 中的 AAW[11:0] = 0x15D（水平同步 + 后沿 + 宽度 -1）。
- LTDC_AWCR 中的 AAH[11:0] = 0xF3（垂直同步 + 后沿 + 高度 -1）。
- LTDC_TWCR 中的 TOTALW[11:0] = 0x167（水平同步 + 后沿 + 宽度 + 前沿 -1）。
- LTDC_TWCR 中的 TOTALH[11:0] = 0xF7（垂直同步 + 后沿 + 高度 + 前沿 -1）。

默认配置

没有推荐的默认配置。显示时序取决于特定的显示面板及其所需的时序，因此所有这些参数必须从显示面板规格中获取。

注： *以上所有配置均是必需配置，以确保 LTDC 正确驱动外部显示面板。*

31.4.17 输出接口极性

输出接口的控制信号极性包括时钟极性、垂直/水平同步和数据使能信号。

LTDC_GCR 中的参数

- CKPOL：输出时钟的极性
- VSPOL：垂直同步的极性
- HSPOL：水平同步的极性
- DEPOL：数据使能的极性

默认配置

极性配置取决于显示面板规格，因此无默认配置。

以下配置可用于使所有项目以正极性工作：

- CKPOL = 1（上升沿）
- VSPOL = 1（高电平有效）
- HSPOL = 1（高电平有效）
- DEPOL = 1（高电平有效）

31.4.18 影子寄存器

激活后，LTDC 预期将连续地逐帧显示数据。

用户可能想要修改一些设置：例如，添加新图层、修改图层窗口位置，或通过更新指向显示缓冲区的指针来更新显示内容。

LTDC 为其大多数配置寄存器提供影子寄存器，以允许在短时垂直同步期间（没有显示内容时）发生快速彻底地同步转换：当 LTDC 忙于使用其配置寄存器的副本来控制正在进行的显示（有效集）时，可通过软件对新版寄存器逐步更新（影子集）。准备就绪后，恰好发生 VSync 事件时，影子集将被复制到有效集中，为下一帧做好准备。

影子集完全更新后，有两种方式将其传输到有效集中：

- 立即复制：立即进行复制。
- 延迟到下一个 VSync 事件：发生下一个 VSync 事件时复制。

注： *如果是延迟复制，则在重载完成前，不得再次修改影子寄存器。*

读取影子寄存器将返回当前有效值。新写入的值只能在重载发生后读到。

由于不能在影子寄存器中读取临时值，因此无法执行读-改-写操作来将影子寄存器中的单个位置 1/复位。

如果在 LTDC_IER 中使能，则可产生寄存器重载中断。

影子寄存器

- 非图层寄存器（不带 LTDC_Lx 前缀）不会进行影子操作：
  这类寄存器被视为包含静态配置（主要依赖于静态连接的显示面板），并且显示期间无需更新。
- 图层寄存器（带 LTDC_Lx 前缀）会进行影子操作（以下列出的部分寄存器除外）：
  它们可以逐帧更新，而且必须快速更新。
  下列图层寄存器因成本过高或需要静态配置（无需影子操作）而不会进行影子操作：
  - LTDC_LxRCR 重载控制（不自行进行影子操作）
  - LTDC_LxCKCR 色键（需要静态配置）
  - LTDC_LxCLUTWR CLUT（间接访问，进行影子操作成本过高）
  - LTDC_LxCYR0R/R1R YUV 到 RGB 颜色转换（视为静态）
  - LTDC_LxFPF0R/LTDC_LxFPF1R 灵活像素格式（视为静态）

参数

影子寄存器的集中重载通过 LTDC_SRCR 中的位字段控制：

- IMR：写入 1 会触发立即复制。
- VBR：写入 1 会触发延迟到下次消隐时复制。

每图层重载控制可以通过 LTDC_LxRCR 中的位字段进行：

- GRMSK 将每图层重载托管给集中重载（上文所述）。
- IMR：写入 1 会触发立即复制该图层。
- VBR：写入 1 会触发延迟到下次消隐时复制。

默认配置

对于初始 LTDC 配置，当显示处于非有效状态时，必须使用 IMR 模式将初始配置复制到有效集。

当显示处于有效状态时，建议使用 VBR 来更新任意参数：参数转换逐帧进行，而不是在进行中的帧内执行。在帧内立即更改 LTDC 配置会导致不可预测的行为。

31.4.19 常规控制

完成所有上述配置之后，最后的必要步骤将触发显示。LTDC 有两种工作模式：

- 连续模式：LTDC 持续向显示屏发送帧。
- 单帧模式：LTDC 在软件触发后发送单个帧（通过软件访问寄存器生成）

LTDC_GCR 中的参数

- LTDCEN：常规 LTDC 激活
- SFEN：从连续模式切换到单帧模式。
- SFSWTR：软件触发（处于单帧模式时）

31.5 LTDC 中断

LTDC 提供各种可屏蔽中断，这些中断分为两组全局中断引脚：一组是工作中断引脚，一组是错误中断引脚。

以下用于同步软件的工作中断通过“或”逻辑合并为全局工作中断：

- 行中断：达到配置的行数时产生。其位置在 LTDC_LIPCR 中配置。
- 寄存器重载中断：在垂直消隐周期内执行影子寄存器重载时产生。

以下错误中断发出关于一些内部处理错误的警告，并通过逻辑“或”合并为全局错误中断：

- FIFO 下溢警告中断：从空图层 FIFO 中请求像素时产生。
- FIFO 下溢错误中断：从空图层 FIFO 中请求更多像素时产生。
- 传输错误中断：数据传输期间出现 AXI 总线错误时产生。
- CRC 错误中断：在计算的 CRC 与预期的 CRC 不同时产生。

LTDC 还提供了一种观测方法，用于检索当前显示像素的 X 和 Y 位置。

参数

以下寄存器用于处理中断：

- LTDC_LIPCR：定义发生行号中断的行号。
- LTDC_IER 解除掩码位：将每个位置 1 可使能相应的中断。
- LTDC_ISR：事件位，用于读取是否发生了事件
- LTDC_ICR：清除位，用于写入以清除中断

以下复制的寄存器用于处理中断：LTDC_LIPCR2、LTDC_IER2、LTDC_ISR2、LTDC_ICR2。

下表列出了可用中断的位字段。

表 265. LTDC 中断请求

中断事件	事件位	使能控制位
行号	LIF	LIE
寄存器重载	RRIF	RRIEN
FIFO 下溢警告	FUWIF	FUWIE
FIFO 下溢错误	FUIF	FUIE
总线传输错误	TERRIF	TERRIE
CRC 错误	CRCIF	CRCIE

31.6 LTDC 编程步骤

LTDC 编程步骤中包括激活时钟，遵循从开始（AXI 主设备）到结束（输出接口极性）的所有操作序列（如*第 31.4 节*中所述），以及根据描述配置所需操作。

必须执行的最低配置如下：

1. 在 RCC 寄存器中使能 LTDC 时钟：
   - APB 从设备
   - AXI 主设备
   - 符合显示面板要求的像素时钟（分辨率、消隐、帧速率）
2. 定义至少一个输入图层的缓冲区（参见*第 31.4.2 节*，包括图层基址、大小或缓冲区间距）。
3. 定义此图层像素格式（参见*第 31.4.3 节*）。根据存储的缓冲区，可以使用 ARGB8888。
4. 定义此图层在最终显示中的插入位置（参见*第 31.4.10 节*）：通常是显示画面的左上角和右下角。
5. 定义该图层如何在最终显示中混合（参见*第 31.4.11 节*）。
6. 定义驱动显示面板的时序（参见*第 31.4.16 节*）。
7. 定义输出信号的极性（参见*第 31.4.17 节*）。
8. 使用立即全局重载一次性复制上述所有设置（参见*第 31.4.18 节*）。
9. 通过激活主使能开始进行显示（参见*第 31.4.18 节*）。

注： *大多数图层寄存器均执行影子操作。一旦对某个寄存器执行写操作，在重载完成前不得再次进行修改。如果在尚未重载时对同一寄存器执行新的写操作，则将覆盖之前的配置。影子寄存器的临时值也无法读取，因而也无法执行读-改-写来将该寄存器中的单个位置 1/ 复位。*

31.7 LTDC 寄存器

31.7.1 LTDC 标识寄存器 (LTDC_IDR)

LTDC identification register

偏移地址：0x000

复位值：0x0004 0100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJVER[7:0]							
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MINVER[7:0]								REV[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 保留，必须保持复位值。

位 23:16 **MAJVER[7:0]**：LTDC 主版本 (LTDC major version)

位 15:8 **MINVER[7:0]**：LTDC 次版本 (LTDC minor version)

位 7:0 **REV[7:0]**：版本 (Revision)

31.7.2 LDTC 图层计数寄存器 (LTDC_LCR)

LDTC layer count register

偏移地址：0x004

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LNBR[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **LNBR[7:0]**：图层数 (Number of layers)

31.7.3 LTDC 同步大小配置寄存器 (LTDC_SSCR)

LTDC synchronization size configuration register

偏移地址：0x008

复位值：0x0000 0000

该寄存器定义水平同步像素数减 1 以及垂直同步行数减 1。关于配置示例，请参见*图 271* 和*第 31.4 节*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	HSW[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	VSH[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **HSW[11:0]**：水平同步宽度 (Horizontal synchronization width)（以像素时钟周期为单位）
该位字段定义水平同步像素数减 1。

位 15:12 保留，必须保持复位值。

位 11:0 **VSH[11:0]**：垂直同步高度 (Vertical synchronization height)（以水平扫描行为单位）
该位字段定义垂直同步高度减 1。它表示水平同步行数。

31.7.4 LTDC 后沿配置寄存器 (LTDC_BPCR)

LTDC back porch configuration register

偏移地址：0x00C

复位值：0x0000 0000

该寄存器定义水平同步像素加水平后沿像素的累加数减 1（HSYNC 宽度 + HBP - 1）以及垂直同步行加垂直后沿行的累加数减 1（VSYNC 高度 + VBP - 1）。关于配置示例，请参见*图 271* 和*第 31.4 节*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	AHBP[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	AVBP[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **AHBP[11:0]**：累加水平后沿 (Accumulated horizontal back porch)（以像素时钟周期为单位）
该位字段定义累加水平后沿宽度（水平同步像素加水平后沿像素减 1）。
水平后沿是水平同步信号变为无效到下一扫描行的有效显示开始之间的间隔。

位 15:12 保留，必须保持复位值。

位 11:0 **AVBP[11:0]**：累加垂直后沿 (Accumulated Vertical back porch)（以水平扫描行为单位）
该位字段定义累加垂直后沿宽度（垂直同步行加垂直后沿行减 1）。
垂直后沿是帧开始到下一帧的首个有效扫描行开始所包含的水平扫描行的数量。

31.7.5 LTDC 有效宽度配置寄存器 (LTDC_AWCR)

LTDC active width configuration register

偏移地址：0x010

复位值：0x0000 0000

该寄存器定义水平同步加水平后沿加有效像素的累加数减 1（HSYNC 宽度 + HBP + 有效宽度 - 1）以及垂直同步行加垂直后沿行加有效行的累加数减 1（VSYNC 高度 + BVBP + 有效高度 - 1）。关于配置示例，请参见*图 271* 和*第 31.4 节*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	AAW[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	AAH[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **AAW[11:0]**：累加有效宽度 (Accumulated active width)（以像素时钟周期为单位）
该位字段定义累加有效宽度（水平同步像素加水平后沿像素加有效像素减 1）。有效宽度是面板扫描行的有效显示区中的像素数。
有关最大像素时钟支持的最大有效宽度，请参见设备数据手册。

位 15:12 保留，必须保持复位值。

位 11:0 **AAH[11:0]**：累加有效高度 (Accumulated active height)（以水平扫描行为单位）
该位字段定义累加高度（垂直同步行加垂直后沿行加有效高度减 1）。有效高度是面板中的有效行数。
有关最大像素时钟支持的最大有效高度，请参见设备数据手册。

31.7.6 LTDC 总宽度配置寄存器 (LTDC_TWCR)

LTDC total width configuration register

偏移地址：0x014

复位值：0x0000 0000

该寄存器定义水平同步加水平后沿加有效像素加水平前沿的像素累加数减 1（HSYNC 宽度 + HBP + 有效宽度 + HFP - 1）以及垂直同步行加垂直后沿行加有效行加垂直前沿行的累加数减 1（VSYNC 高度 + BVBP + 有效高度 + VFP - 1）。关于配置示例，请参见*图 271* 和*第 31.4 节*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TOTALW[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TOTALH[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **TOTALW[11:0]**：总宽度 (Total width)（以像素时钟周期为单位）
该位字段定义累加总宽度（水平同步像素加水平后沿像素加有效宽度加水平前沿像素减 1）。

位 15:12 保留，必须保持复位值。

位 11:0 **TOTALH[11:0]**：总高度 (Total height)（以水平扫描行为单位）
该位字段定义累加高度（垂直同步行加垂直后沿行加有效高度加垂直前沿行减 1）。

31.7.7 LTDC 全局控制寄存器 (LTDC_GCR)

LTDC global control register

偏移地址：0x018

复位值：0x0000 2220

该寄存器定义 LTDC 全局配置。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HSPOL	VSPOL	DEPOL	PCPOL	Res.	Res.	SFSWTR	SFEN	Res.	Res.	Res.	Res.	CRCEN	Res.	Res.	DEN
rw	rw	rw	rw			w	rw					rw			rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DRW[2:0]			Res.	DGW[2:0]			Res.	DBW[2:0]			Res.	Res.	GAMEN	LTDCEN
	r	r	r		r	r	r		r	r	r			rw	rw

位 31 **HSPOL**：水平同步极性 (Horizontal synchronization polarity)
该位由软件置 1 和清零，
0：水平同步极性低电平有效。
1：水平同步极性高电平有效。

位 30 **VSPOL**：垂直同步极性 (Vertical synchronization polarity)
该位由软件置 1 和清零，
0：垂直同步低电平有效。
1：垂直同步高电平有效。

位 29 **DEPOL**：消隐（非数据/像素）极性 (Blanking (no data/pixel) polarity)
该位由软件置 1 和清零，
0：消隐（非数据/像素）极性低电平有效。
1：消隐（非数据/像素）极性高电平有效。

位 28 **PCPOL**：像素时钟极性 (Pixel clock polarity)
该位由软件置 1 和清零，
0：像素和同步数据在输出 LCD_CLK 时钟的上升沿生成。
1：像素和同步数据在输出 LCD_CLK 时钟的下降沿生成。

位 27:26 保留，必须保持复位值。

位 25 **SFSWTR**：单帧模式：软件触发 (Single-frame mode: software trigger)
该位通过软件置 1，通过硬件清零。
0：无操作
1：触发一个帧

位 24 **SFEN**：单帧模式：模式使能 (Single-frame mode: mode enable)
该位由软件置 1 和清零，
0：禁止单帧：触发（在 SFSWTR 上）生成连续流。
1：使能单帧：触发（在 SFSWTR 上）生成单个帧。

位 23:20 保留，必须保持复位值。

位 19 **CRCEN**：CRC 使能 (CRC enable)
该位由软件置 1 和清零，
0：禁止 CRC
1：CRC 已使能

位 18:17 保留，必须保持复位值。

位 16 **DEN**：抖动使能 (Dither enable)
该位由软件置 1 和清零，
0：禁止抖动
1：使能抖动

位 15 保留，必须保持复位值。

位 14:12 **DRW[2:0]**：抖动红色宽度 (Dither red width)
该位字段返回抖动红色位。

位 11 保留，必须保持复位值。

位 10:8 **DGW[2:0]**：抖动绿色宽度 (Dither green width)
该位字段返回抖动绿色位。

位 7 保留，必须保持复位值。

位 6:4 **DBW[2:0]**：抖动蓝色宽度 (Dither blue width)
该位字段返回抖动蓝色位。

位 3:2 保留，必须保持复位值。

位 1 **GAMEN**：伽马校正使能 (Gamma correction enable)
该位由软件置 1 和清零，
0：禁止伽马校正（像素绕过伽马操作）
1：使能伽马校正

位 0 **LTDCEN**：LTDC 全局使能 (LTDC global enable)
该位由软件置 1 和清零，
0：禁止 LTDC
1：使能 LTDC

31.7.8 LTDC 全局配置 1 寄存器 (LTDC_GC1R)

LTDC global configuration 1 register

偏移地址：0x01C

复位值：0x6BE4 D888

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BMA	CRMA	STRA	DWP	SPP	Res.	TP	LNIP	BBA	BCP	SHRA	Res.	GCT[2:0]			Res.
r	r	r	r	r		r	r	r	r	r		r	r	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DT[1:0]		Res.	PRBA	WRCH[3:0]				WGCH[3:0]				WBCH[3:0]			
r	r		r	r	r	r	r	r	r	r	r	r	r	r	r

位 31 **BMA**：盲模式能力 (Blind mode ability)

位 30 **CRMA**：配置读取模式能力 (Configuration reading mode ability)

位 29 **STRA**：状态寄存器能力 (Status register ability)

位 28 **DWP**：抖动宽度可编程性 (Dither width programmability)

位 27 **SPP**：同步极性可编程性 (Sync polarity programmability)

位 26 保留，必须保持复位值。

位 25 **TP**：时序可编程性 (Timing programmability)

位 24 **LNIP**：Line-IRQ：线位置可编程性 (Line-IRQ: line position programmability)

位 23 **BBA**：背景混合能力 (Background blending ability)

位 22 **BCP**：背景色可编程性 (Background color programmability)（混合为背景的独特颜色）

位 21 **SHRA**：影子寄存器能力 (Shadow register ability)

位 20 保留，必须保持复位值。

位 19:17 **GCT[2:0]**：伽马校正技术已实现 (Gamma correction technique implemented)

000：无伽马
001：256 个样本伽马
010：8 个插值段伽马
其他值：保留

位 16 保留，必须保持复位值。

位 15:14 **DT[1:0]**：抖动技术已实现 (Dithering technique implemented)

00：无抖动
01：有序 4x4 拜尔
10：有序 4x4 自定义
11：伪随机 LFSR

位 13 保留，必须保持复位值。

位 12 **PRBA**：支持精确混合 (Precise blending ability)

位 11:8 **WRCH[3:0]**：红色通道输出的宽度 (Width of red channel output)

位 7:4 **WGCH[3:0]**：绿色通道输出的宽度 (Width of green channel output)

位 3:0 **WBCH[3:0]**：蓝色通道输出的宽度 (Width of blue channel output)

31.7.9 LTDC 全局配置 2 寄存器 (LTDC_GC2R)

LTDC global configuration 2 register

偏移地址：0x020

复位值：0x0000 BB30

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOA	Res.	CRCA	SFA	SISA	ROTA	AXIIDA	OCA	EDCA	BW[2:0]			DPA	DVA	STSA	BLA
r		r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15 **BOA**：混合顺序能力 (Blending order ability)
0：混合顺序固定
1：混合顺序可配置

位 14 保留，必须保持复位值。

位 13 **CRCA**：CRC 能力 (CRC ability)
0：无 CRC 计算
1：有 CRC 计算

位 12 **SFA**：单帧模式能力 (Single frame mode ability)
0：单帧不可用
1：单帧可用

位 11 **SISA**：秒中断集能力 (Second interrupt set ability)
0：秒中断集不可用
1：秒中断集可用

位 10 **ROTA**：旋转支持能力 (Rotation support ability)

位 9 **AXIIDA**：AXIID 能力 (AXIID ability)

位 8 **OCA**：输出转换能力 (Output conversion ability)（RGB 到 YCbCr）

位 7 **EDCA**：外部显示控制能力 (External display control ability)

位 6:4 **BW[2:0]**：总线宽度 (Bus width)（以 2 为底字节数的对数）
010：32 位总线
011：64 位总线
100：128 位总线
其他值：保留

位 3 **DPA**：次级 RGB 输出端口能力 (Secondary RGB output port ability)

位 2 **DVA**：双视图能力 (Dual-view ability)

位 1 **STSA**：从设备时序同步能力 (Slave timings synchronization ability)

位 0 **BLA**：背景层能力 (Background layer ability)（背景层的像素从内存读取）

31.7.10 LTDC 影子重载配置寄存器 (LTDC_SRCR)

LTDC shadow reload configuration register

偏移地址：0x024

复位值：0x0000 0000

该寄存器允许立即或在垂直消隐周期内将影子寄存器的值重载到活动寄存器中。影子寄存器为图层 x 寄存器，但 LTDC_L1CLUTWR 和 LTDC_L2CLUTWR 除外。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBR	IMR
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **VBR**：垂直消隐重载请求 (Vertical blanking reload request)
该位由软件置 1，只有重载后才由硬件清零（一旦置 1，便无法通过寄存器写操作清零）。
0：无影响
1：影子寄存器在垂直消隐周期（有效显示区之后的第一行开始时）内重载。

位 0 **IMR**：立即重载触发 (Immediate reload trigger)
该位由软件置 1，只有重载后才由硬件清零。
0：无影响
1：影子寄存器立即被重载。

31.7.11 LTDC 伽马校正配置寄存器 (LTDC_GCCR)

LTDC gamma correction configuration register

偏移地址：0x028

复位值：0x0000 0000

该寄存器对伽马校正 R、G、B 表进行间接访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REN	GEN	BEN
													w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP[7:0]								ADDR[7:0]							
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:19 保留，必须保持复位值。

位 18 **REN**：写触发红表 (Write trigger to the red table)
0：无操作
1：在红表的 ADDR 写入 COMP。

位 17 **GEN**：写触发绿表 (Write trigger to the green table)
0：无操作
1：在绿表的 ADDR 写入 COMP。

位 16 **BEN**：写触发蓝表 (Write trigger to the blue table)
0：无操作
1：在蓝表的 ADDR 写入 COMP。

位 15:8 **COMP[7:0]**：要写入的颜色分量（在一个或全部 R、G、B 表中）

位 7:0 **ADDR[7:0]**：写入 COMP 分量的 R、G、B 表的地址
当 LTDC_GC1R.GCT=2 时，伽马使用八个插值段实现。在这种情况下，有效的 ADDR 地址是：0, 32, 64, 96, 128, 160, 192, 224, 255。
注意：如果 ADDR = 255，则伽马插值硬件认为地址为 256。

31.7.12 LTDC 背景色配置寄存器 (LTDC_BCCR)

LTDC background color configuration register

偏移地址：0x02C

复位值：0x0000 0000

该寄存器定义背景色 (RGB888)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BCRED[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCGREEN[7:0]								BCBLUE[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **BCRED[7:0]**：背景红色值 (Background color red value)
该位字段配置背景红色值。

位 15:8 **BCGREEN[7:0]**：背景绿色值 (Background color green value)
该位字段配置背景绿色值。

位 7:0 **BCBLUE[7:0]**：背景蓝色值 (Background color blue value)
该位字段配置背景蓝色值。

31.7.13 LTDC 中断使能寄存器 (LTDC_IER)

LTDC interrupt enable register

偏移地址：0x034

复位值：0x0000 0000

该寄存器通过将对应的位置 1 来确定哪一个状态标志位产生中断请求。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCIE	FUIE	Res.	Res.	RRIE	TERRIE	FUWIE	LIE
								rw	rw			rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7 **CRCIE**：CRC 错误中断使能 (CRC error interrupt enable)
该位由软件置 1 和清零，
0：禁止 CRC 错误中断
1：使能 CRC 错误中断

位 6 **FUIE**：FIFO 下溢中断使能 (FIFO underrun interrupt enable)
该位由软件置 1 和清零，
0：禁止 FIFO 下溢中断
1：使能 FIFO 下溢中断

位 5:4 保留，必须保持复位值。

位 3 **RRIE**：寄存器重载中断使能 (Register reload interrupt enable)
该位由软件置 1 和清零，
0：禁止寄存器重载中断
1：使能寄存器重载中断

位 2 **TERRIE**：传输错误中断使能 (Transfer error interrupt enable)
该位由软件置 1 和清零，
0：禁止传输错误中断
1：使能传输错误中断

位 1 **FUWIE**：FIFO 下溢警告中断使能 (FIFO underrun warning interrupt enable)
该位由软件置 1 和清零，
0：禁止 FIFO 下溢中断
1：使能 FIFO 下溢中断

位 0 **LIE**：行中断使能 (Line interrupt enable)
该位由软件置 1 和清零，
0：禁止行中断
1：使能行中断

31.7.14 LTDC 中断状态寄存器 (LTDC_ISR)

LTDC interrupt status register

偏移地址：0x038

复位值：0x0000 0000

该寄存器返回中断状态标志。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCIF	FUIF	Res.	Res.	RRIF	TERRIF	FUWIF	LIF
								r	r			r	r	r	r

位 31:8 保留，必须保持复位值。

位 7 **CRCIF**：CRC 错误中断标志 (CRC error interrupt flag)
0：未产生 CRC 错误中断
1：出现总线错误时产生 CRC 错误中断

位 6 **FUIF**：FIFO 下溢中断标志 (FIFO underrun interrupt flag)
0：未产生 FIFO 下溢中断
1：当其中一个层 FIFO 为空并从 FIFO 读取大量像素数据时，将产生 FIFO 下溢中断

位 5:4 保留，必须保持复位值。

位 3 **RRIF**：寄存器重载中断标志 (Register reload interrupt flag)
0：未产生寄存器重载中断
1：发生垂直消隐重载时（以及到达有效区域后的第一行时）产生寄存器重载中断

位 2 **TERRIF**：传输错误中断标志 (Transfer error interrupt flag)
0：未产生传输错误中断
1：出现总线错误时产生传输错误中断

位 1 **FUWIF**：FIFO 下溢警告中断标志 (FIFO underrun warning interrupt flag)
0：未产生 FIFO 下溢警告中断
1：当其中一个层 FIFO 为空并从 FIFO 读取像素数据时，将产生 FIFO 下溢警告中断

位 0 **LIF**：行中断标志 (Line interrupt flag)
0：未产生行中断
1：达到编程的行时产生行中断

31.7.15 LTDC 中断清零寄存器 (LTDC_ICR)

LTDC interrupt clear register

偏移地址：0x03C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCRCIF	CFUIF	Res.	Res.	CRRIF	CTERRIF	CFUWIF	CLIF
								w	w			w	w	w	w

位 31:8 保留，必须保持复位值。

位 7 **CCRCIF**：CRC 错误中断标志清零 (CRC error interrupt flag clear)
0：无影响
1：将 LTDC_ISR 中的 CRCIF 标志清零。

位 6 **CFUIF**：FIFO 下溢中断标志清零 (FIFO underrun interrupt flag clear)
0：无影响
1：将 LTDC_ISR 中的 FUIF 标志清零。

位 5:4 保留，必须保持复位值。

位 3 **CRRIF**：寄存器重载中断标志清零 (Register reload interrupt flag clear)
0：无影响
1：将 LTDC_ISR 中的 RRIF 标志清零。

位 2 **CTERRIF**：传输错误中断标志清零 (Transfer error interrupt flag clear)
0：无影响
1：将 LTDC_ISR 中的 TERRIF 标志清零。

位 1 **CFUWIF**：FIFO 下溢警告中断标志清零 (FIFO underrun warning interrupt flag clear)
0：无影响
1：将 LTDC_ISR 中的 FUWIF 标志清零。

位 0 **CLIF**：行中断标志清零 (Line-interrupt flag clear)
0：无影响
1：将 LTDC_ISR 中的 LIF 标志清零。

31.7.16 LTDC 行中断位置配置寄存器 (LTDC_LIPCR)

LTDC line interrupt position configuration register

偏移地址：0x040

复位值：0x0000 0000

该寄存器定义行中断位置。要编程的行值取决于时序参数（参见*图 271*）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	LIPOS[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:0 **LIPOS[11:0]**：行中断位置 (Line interrupt position)
该位字段配置行中断位置。

31.7.17 LTDC 当前位置状态寄存器 (LTDC_CPSR)

LTDC current position status register

偏移地址：0x044

复位值：0x0000 0000

该寄存器返回当前显示像素的 X、Y 位置。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CXPOS[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CYPOS[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27:16 **CXPOS[11:0]**：当前 X 位置 (Current X position)

位 15:12 保留，必须保持复位值。

位 11:0 **CYPOS[11:0]**：当前 Y 位置 (Current Y position)

31.7.18 LTDC 当前显示状态寄存器 (LTDC_CDSR)

LTDC current display status register

偏移地址：0x048

复位值：0x0000 0003

该寄存器返回由 HSYNC、VSYNC 和水平/垂直数据使能信号控制的当前显示阶段的状态。

示例：如果当前显示阶段为垂直同步阶段，则 VSYNCS 高电平有效。示例：如果当前显示阶段为水平同步阶段，则 HSYNCS 高电平有效。

返回的状态不取决于 LTDC_GCR 中配置的极性：返回当前有效显示阶段。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSYNCS	VSYNCS	HDES	VDES
												r	r	r	r

位 31:4 保留，必须保持复位值。

位 3 **HSYNCS**：水平同步显示状态 (Horizontal synchronization display status)
0：低电平有效
1：高电平有效

位 2 **VSYNCS**：垂直同步显示状态 (Vertical synchronization display status)
0：低电平有效
1：高电平有效

位 1 **HDES**：水平数据使能显示状态 (Horizontal data enable display status)
0：低电平有效
1：高电平有效

位 0 **VDES**：垂直数据使能显示状态 (Vertical data enable display status)
0：低电平有效
1：高电平有效

31.7.19 LTDC 外部显示控制寄存器 (LTDC_EDCR)

LTDC external display control register

偏移地址：0x060

复位值：0x0000 0000

该寄存器控制外部显示接口。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	OCYCO	OCYSEL	OCYEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
				rw	rw	rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:28 保留，必须保持复位值。

位 27 **OCYCO**：输出转换为 YCbCr 422
该位定义色度顺序（先输出 Cb 还是 Cr）。
0：先输出 Cb（Y0Cb，然后是 Y1Cr、Y2Cb 等等）。
1：先输出 Cr（Y0Cr，然后是 Y1Cb、Y2Cr 等等）。

位 26 **OCYSEL**：输出转换为 YCbCr 422
该位选择 CCIR 硬连线系数集（ITU-R BT.601 或 709）。
0：使用 ITU-R BT.601 集（适用于典型的 SDTV 类模拟显示）。
1：使用 ITU-R BT.709 集（适用于典型的 HDTV 类数字显示）。

位 25 **OCYEN**：输出转换为 YCbCr 422 使能
0：禁止转换
1：使能转换

位 24:0 保留，必须保持复位值。

31.7.20 LTDC 中断使能寄存器 2 (LTDC_IER2)

LTDC interrupt enable register 2

偏移地址：0x064

复位值：0x0000 0000

该寄存器通过将对应的位置 1 来确定哪一个状态标志位产生中断请求。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCIE	FUIE	Res.	Res.	RRIE	TERRIE	FUWIE	LIE
								rw	rw			rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7 **CRCIE**：CRC 错误中断使能 (CRC error interrupt enable)
该位由软件置 1 和清零，
0：禁止 CRC 错误中断
1：使能 CRC 错误中断

位 6 **FUIE**：FIFO 下溢中断使能 (FIFO underrun interrupt enable)
该位由软件置 1 和清零，
0：禁止 FIFO 下溢中断
1：使能 FIFO 下溢中断

位 5:4 保留，必须保持复位值。

位 3 **RRIE**：寄存器重载中断使能 (Register reload interrupt enable)
该位由软件置 1 和清零，
0：禁止寄存器重载中断
1：使能寄存器重载中断

位 2 **TERRIE**：传输错误中断使能 (Transfer error interrupt enable)
该位由软件置 1 和清零，
0：禁止传输错误中断
1：使能传输错误中断

位 1 **FUWIE**：FIFO 下溢警告中断使能 (FIFO underrun warning interrupt enable)
该位由软件置 1 和清零，
0：禁止 FIFO 下溢中断
1：使能 FIFO 下溢中断

位 0 **LIE**：行中断使能 (Line interrupt enable)
该位由软件置 1 和清零，
0：禁止行中断
1：使能行中断

31.7.21 LTDC 中断状态寄存器 2 (LTDC_ISR2)

LTDC interrupt status register 2

偏移地址：0x068

复位值：0x0000 0000

该寄存器返回中断状态标志。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCIF	FUIF	Res.	Res.	RRIF	TERRIF	FUWIF	LIF
								r	r			r	r	r	r

位 31:8 保留，必须保持复位值。

位 7 **CRCIF**：CRC 错误中断标志 (CRC error interrupt flag)
0：未产生 CRC 错误中断
1：出现总线错误时产生 CRC 错误中断

位 6 **FUIF**：FIFO 下溢中断标志 (FIFO underrun interrupt flag)
0：未产生 FIFO 下溢中断。
1：当其中一个层 FIFO 为空并从 FIFO 读取大量像素数据时，将产生 FIFO 下溢中断

位 5:4 保留，必须保持复位值。

位 3 **RRIF**：寄存器重载中断标志 (Register reload interrupt flag)
0：未产生寄存器重载中断
1：发生垂直消隐重载时（以及到达有效区域后的第一行时）产生寄存器重载中断

位 2 **TERRIF**：传输错误中断标志 (Transfer error interrupt flag)
0：未产生传输错误中断
1：出现总线错误时产生传输错误中断

位 1 **FUWIF**：FIFO 下溢警告中断标志 (FIFO underrun warning interrupt flag)
0：未产生 FIFO 下溢警告中断。
1：当其中一个层 FIFO 为空并从 FIFO 读取像素数据时，将产生 FIFO 下溢警告中断

位 0 **LIF**：行中断标志 (Line interrupt flag)
0：未产生行中断
1：达到编程的行时产生行中断

31.7.22 LTDC 中断清零寄存器 2 (LTDC_ICR2)

LTDC interrupt clear register 2

偏移地址：0x06C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCRCIF	CFUIF	Res.	Res.	CRRIF	CTERRIF	CFUWIF	CLIF
								w	w			w	w	w	w

位 31:8 保留，必须保持复位值。

位 7 **CCRCIF**：清零 CRC 错误中断标志 (Clear the CRC error interrupt flag)
0：无影响
1：将 LTDC_ISR2 中的 CRCIF 标志清零。

位 6 **CFUIF**：清除 FIFO 下溢中断标志 (Clear the FIFO underrun interrupt flag)
0：无影响
1：将 LTDC_ISR2 中的 FUIF 标志清零。

位 5:4 保留，必须保持复位值。

位 3 **CRRIF**：清零寄存器重载中断标志 (Clear register reload interrupt flag)
0：无影响
1：将 LTDC_ISR2 中的 RRIF 标志清零。

位 2 **CTERRIF**：清零传输错误中断标志 (Clear the transfer error interrupt flag)
0：无影响
1：将 LTDC_ISR2 中的 TERRIF 标志清零。

位 1 **CFUWIF**：清零 FIFO 下溢警告中断标志 (Clear the FIFO underrun warning interrupt flag)
0：无影响
1：将 LTDC_ISR2 中的 FUWIF 标志清零。

位 0 **CLIF**：清零行中断标志 (Clear the line interrupt flag)
0：无影响
1：将 LTDC_ISR2 中的 LIF 标志清零。

31.7.23 LTDC 行中断位置配置寄存器 2 (LTDC_LIPCR2)

LTDC line interrupt position configuration register 2

偏移地址：0x070

复位值：0x0000 0000

该寄存器定义行中断位置。要编程的行值取决于时序参数（参见*图 271*）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	LIPOS[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:0 **LIPOS[11:0]**：行中断位置 (Line interrupt position)

31.7.24 LTDC 预期 CRC 寄存器 (LTDC_ECRCR)

LTDC expected CRC register

偏移地址：0x078

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECRC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **ECRC[15:0]**：预期的帧 CRC (Expected CRC of frame)

31.7.25 LTDC 计算 CRC 寄存器 (LTDC_CCRCR)

LTDC computed CRC register

偏移地址：0x07C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **CCRC[15:0]**：计算的帧 CRC

31.7.26 LTDC FIFO 下溢阈值寄存器 (LTDC_FUTR)

LTDC FIFO underrun threshold register

偏移地址：0x090

复位值：0x0000 0010

THRE[15:0] 的默认值为 0x80，代表 128 个 64 位字，即 256 像素 32bpp 或 YUV420 像素 1024 亮度。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THRE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **THRE[15:0]**：触发 FIFO 下溢中断的阈值 (Threshold to trigger a FIFO underrun interrupt)（每个 FIFO 字，64 位）

FIFO 下溢警告变成 FIFO 下溢错误的阈值。

31.7.27 LTDC 第 x 层配置 0 寄存器 (LTDC_LxC0R)

LTDC layerx configuration 0 register

偏移地址：0x100 + 0x100 * (x - 1)（x = 1 到 2）

复位值：0xFF50 A075

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARGB8888	ABGR8888	RGBA8888	BGRA888	RGB565	BGR565	RGB888	FF	F11PC	F1PC	F11C	F1C	F11P	F1P	F10	F11
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F21PC	F2PC	F21C	F2C	F21P	F2P	F20	F21	CKRA	CLUTA	WINA	DCP	APA	CFBPA	CFBDA	CKTA
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31 **ARGB8888**：支持像素格式 argb8888 (Pixel format, ability for argb8888)

位 30 **ABGR8888**：支持像素格式 abgr8888 (Pixel format, ability for abgr8888)

位 29 **RGBA8888**：支持像素格式 rgba8888 (Pixel format, ability for rgba8888)

位 28 **BGRA888**：支持像素格式 bgra8888 (Pixel format, ability for bgra8888)

位 27 **RGB565**：支持像素格式 rgb565 (Pixel format, ability for rgb565)

位 26 **BGR565**：支持像素格式 bgr565 (Pixel format, ability for bgr565)

位 25 **RGB888**：支持像素格式 rgb888 (Pixel format, ability for rgb888)

位 24 **FF**：支持灵活像素格式 (Flexible pixel format, ability)

位 23 **F11PC**：支持混合系数 1，适用于：1.0 - (pixel_alpha * constant_alpha)

位 22 **F1PC**：支持混合系数 1，适用于：pixel_alpha * constant_alpha

位 21 **F11C**：支持混合系数 1，适用于：1.0 - constant_alpha

位 20 **F1C**：支持混合系数 1，适用于：constant_alpha

位 19 **F11P**：支持混合系数 1，适用于：1.0 - pixel_alpha

位 18 **F1P**：支持混合系数 1，适用于：pixel_alpha

位 17 **F10**：支持混合系数 1，适用于：0.0

位 16 **F11**：支持混合系数 1，适用于：1.0

位 15 **F21PC**：支持混合系数 2，适用于：1.0 - (pixel_alpha * constant_alpha)

位 14 **F2PC**：支持混合系数 2，适用于：pixel_alpha * constant_alpha

位 13 **F21C**：支持混合系数 2，适用于：1.0 - constant_alpha

位 12 **F2C**：支持混合系数 2，适用于：constant_alpha

位 11 **F21P**：支持混合系数 2，适用于：1.0 - pixel_alpha

位 10 **F2P**：支持混合系数 2，适用于：pixel_alpha

位 9 **F20**：支持混合系数 2，适用于：0.0

位 8 **F21**：支持混合系数 2，适用于：1.0

位 7 **CKRA**：色键替换能力 (Color key replace ability)

位 6 **CLUTA**：CLUT 能力 (CLUT ability)

位 5 **WINA**：窗口化能力 (Windowing ability)

位 4 **DCP**：默认颜色可编程性 (Default color programmability)

位 3 **APA**：Alpha 平面能力 (Alpha plane ability)

位 2 **CFBPA**：颜色帧缓冲区间距能力 (Color frame buffer pitch ability)

位 1 **CFBDA**：颜色帧缓冲区复制能力 (Color frame buffer duplication ability)

位 0 **CKTA**：色键透明能力 (Color key transparency ability)

31.7.28 LTDC 第 x 层配置 1 寄存器 (LTDC_LxC1R)

LTDC layerx configuration 1 register

偏移地址：0x104 + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 0007, 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YFPA	YSPA	YIA
													r	r	r

位 31 **SCA**：该层的调整能力 (Scaling ability for this layer)
0：不可调整
1：可进行调整

位 30:3 保留，必须保持复位值。

位 2 **YFPA**：该层支持 YCbCr 420 全平面格式 (YCbCr 420 full-planar ability for this layer)
0：不支持全平面
1：支持全平面

位 1 **YSPA**：该层支持 YCbCr 420 半平面格式 (YCbCr 420 semi-planar ability for this layer)
0：不支持半平面格式
1：支持半平面格式

位 0 **YIA**：该层支持 YCbCr 422 交错格式 (YCbCr 422 interleaved ability for this layer)
0：不支持交错格式
1：支持交错格式

31.7.29 LTDC 第 x 层重载控制寄存器 (LTDC_LxRCR)

LTDC layerx reload control register

偏移地址：0x108 + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GRMSK	VBR	IMR
													rw	rw	w

位 31:3 保留，必须保持复位值。

位 2 **GRMSK**：屏蔽了全局（集中）重载的影子重载控制 (Shadow reload control, global (centralized) reload masked)
该位由软件置 1 和清零，
0：对该层屏蔽了全局重载（禁止通过 LTDC_SRCR 进行控制）
1：全局重载对该层有效（使能通过 LTDC_SRCR 进行控制）

位 1 **VBR**：垂直消隐重载请求 (Vertical blanking reload request)
该位由软件置 1，只有重载后才由硬件清零（一旦置 1，便无法通过寄存器写操作清零）。
0：无影响
1：影子寄存器在垂直消隐周期（有效显示区之后的第一行开始时）内重载。

位 0 **IMR**：立即重载触发 (Immediate reload trigger)
该位由软件置 1，只有重载后才由硬件清零。
0：无影响
1：影子寄存器立即重载。

31.7.30 LTDC 第 x 层控制寄存器 (LTDC_LxCR)

LTDC layerx control register

偏移地址：0x10C + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	DCBEN	HMEN	Res.	Res.	Res.	CLUTEN	Res.	Res.	CKEN	LEN
						rw	rw				rw			rw	rw

位 31:10 保留，必须保持复位值。

位 9 **DCBEN**：默认颜色混合使能 (Default color blending enable)
该位由软件置 1 和清零，
0：禁止混合
1：使能混合

位 8 **HMEN**：水平镜像使能 (Horizontal mirroring enable)
该位由软件置 1 和清零，
0：禁止镜像
1：使能镜像（如果使能，则颜色帧缓冲区起始地址必须设置为第一行的最后一个字节，例如：如果行是 100 像素，24 bpp，则地址设置为 299）

位 7:5 保留，必须保持复位值。

位 4 **CLUTEN**：查色表使能 (Color look-up table enable)
该位由软件置 1 和清零，
0：禁止查色表
1：使能查色表
CLUT 仅对 L8、AL44 和 AL88 像素格式有意义。

位 3:2 保留，必须保持复位值。

位 1 **CKEN**：使能色键
该位由软件置 1 和清零，
0：禁止色键
1：使能色键：如果 RGB 匹配，则 ARGB 设置为 0。

位 0 **LEN**：层使能 (Layer enable)
该位用于使能/禁止整个图层的存在。由软件置 1 和清零。
0：禁止层
1：使能层

31.7.31 LTDC 第 x 层窗口水平位置配置寄存器 (LTDC_LxWHPCR)

LTDC layerx window horizontal position configuration register

偏移地址：0x110 + 0x100 * (x -1)（x = 1 到 2）

复位值：0x0000 0000

该寄存器定义第 1/2 层窗口的水平位置（第一个和最后一个像素）。

行的第一个可见像素是在 LTDC_BPCR 中编程的 AHBP[11:0] + 1 的值。

行的最后一个可见像素是在 LTDC_AWCR 中编程的 AAW[11:0] 的值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	WHSPPOS[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	WHSTPOS[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **WHSPPOS[11:0]**：窗口水平结束位置 (Window horizontal stop position)
该位字段配置层窗口的一个行的最后一个可见像素。
WHSPPOS[11:0] 必须 ≥ AHBP[11:0] 位 + 1（在 LTDC_BPCR 中编程）。

位 15:12 保留，必须保持复位值。

位 11:0 **WHSTPOS[11:0]**：窗口水平起始位置 (Window horizontal start position)
该位字段配置层窗口的一个行的第一个可见像素。
WHSTPOS[11:0] 必须 ≤ AAW[11:0] 位（在 LTDC_AWCR 中编程）。

31.7.32 LTDC 第 x 层窗口垂直位置配置寄存器 (LTDC_LxWVPCR)

LTDC layerx window vertical position configuration register

偏移地址：0x114 + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

该寄存器定义第 1 层或第 2 层窗口的垂直位置（第一行或最后一行）。

一个帧的第一个可见行是在 LTDC_BPCR 中编程的 AVBP[11:0] + 1 的值。

一个帧的最后一个可见行是在 LTDC_AWCR 中编程的 AAH[11:0] 的值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	WVSPPOS[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	WVSTPOS[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **WVSPPOS[11:0]**：窗口垂直结束位置 (Window vertical stop position)
该位字段配置层窗口的最后一个可见行。
WVSPPOS[11:0] 必须 ≥ AVBP[11:0] + 1（在 LTDC_BPCR 中编程）。

位 15:12 保留，必须保持复位值。

位 11:0 **WVSTPOS[11:0]**：窗口垂直起始位置 (Window vertical start position)
该位字段配置层窗口的第一个可见行。
WVSTPOS[11:0] 必须 ≤ AAH[11:0]（在 LTDC_AWCR 中编程）。

31.7.33 LTDC 第 x 层色键配置寄存器 (LTDC_LxCKCR)

LTDC layerx color keying configuration register

偏移地址：0x118 + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

该寄存器定义色键使用的色键值 (RGB)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKRED[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKGREEN[7:0]								CKBLUE[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **CKRED[7:0]**：色键红色值 (Color key red value)

位 15:8 **CKGREEN[7:0]**：色键绿色值 (Color key green value)

位 7:0 **CKBLUE[7:0]**：色键蓝色值 (Color key blue value)

31.7.34 LTDC 第 x 层像素格式配置寄存器 (LTDC_LxPFCR)

LTDC layerx pixel format configuration register

偏移地址：0x11C + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

该寄存器定义层的帧缓冲区中存储的数据所使用的像素格式。像素数据从帧缓冲区读取，随后转换为内部格式 8888 (ARGB)。如果每个分量小于 8 位，则通过将其 MSB 复制到 LSB，将其扩展到 8 位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PF[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **PF[2:0]**：像素格式

该位字段配置像素格式。

000：ARGB8888 (32 bpp)
001：ABGR888 (32 bpp)
010：RGBA888 (32 bpp)
011：BGRA8888 (32 bpp)
100：RGB565（16 bpp，A = 255）
101：BGR565（16 bpp，A = 255）
110：RGB888（24 bpp 封装，A = 255）
111：选择灵活像素格式（参见*第 31.7.50 节*和*第 31.7.51 节*）

31.7.35 LTDC 第 x 层恒定 alpha 配置寄存器 (LTDC_LxCACR)

LTDC layerx constant alpha configuration register

偏移地址：0x120 + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 00FF

该寄存器定义在 alpha 混合中使用的恒定 alpha 值（由硬件实现 255 等分）（参见 LTDC_LxBFCR）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CONSTA[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **CONSTA[7:0]**：恒定 alpha (Constant alpha)

该位字段配置混合时使用的恒定 alpha。恒定 alpha 由硬件实现 255 等分。
示例：如果编程的恒定 alpha 为 0xFF，则浮点 alpha 值为 255/255 = 1。

31.7.36 LTDC 第 x 层默认颜色配置寄存器 (LTDC_LxDCCR)

LTDC layerx default color configuration register

偏移地址：0x124 + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

该寄存器定义采用 ARGB 格式的层的默认颜色。默认颜色在定义的层窗口外使用或在层禁止时使用。复位值 0x00000000 定义了透明黑色。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DCALPHA[7:0]								DCRED[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCGREEN[7:0]								DCBLUE[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **DCALPHA[7:0]**：默认颜色 alpha (Default color alpha)
该位字段配置默认 alpha 值。

位 23:16 **DCRED[7:0]**：默认颜色红色 (Default color red)
该位字段配置默认红色值。

位 15:8 **DCGREEN[7:0]**：默认颜色绿色 (Default color green)
该位字段配置默认绿色值。

位 7:0 **DCBLUE[7:0]**：默认颜色蓝色 (Default color blue)
该位字段配置默认蓝色值。

31.7.37 LTDC 第 x 层混合系数配置寄存器 (LTDC_LxBFCR)

LTDC layerx blending factors configuration register

偏移地址：0x128 + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 0607, 0x0001 0607,

该寄存器定义混合顺序和系数 BF1 和 BF2。恒定 alpha 值是在 LTDC_LxCACR 中编程的值，由硬件实现 255 等分。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOR
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	BF1[2:0]			Res.	Res.	Res.	Res.	Res.	BF2[2:0]		
					rw	rw	rw						rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **BOR**：默认顺序 (Default order)
该位定义层的混合顺序。如果两层有相同的 BOR 值，则索引最高的层位于前景。
0：层设置在背景中
1：层设置在前景中

位 15:11 保留，必须保持复位值。

位 10:8 **BF1[2:0]**：默认系数 1 (Default factor 1)
该位字段选择混合系数 F1。
100：恒定 alpha (Constant alpha)
110：像素 alpha x 恒定 alpha
其他值：保留

位 7:3 保留，必须保持复位值。

位 2:0 **BF2[2:0]**：混合系数 2 (Blending factor 2)
该位字段选择混合系数 F2
101：1 - 恒定 alpha
111：1 -（像素 alpha x 恒定 alpha）
其他值：保留

31.7.38 LTDC 第 x 层突发传输长度配置寄存器 (LTDC_LxBLCR)

LTDC layerx burst length configuration register

偏移地址：0x12C + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

该寄存器定义第 x 层请求的突发传输长度。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BL[4:0]				
											rw	rw	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4:0 **BL[4:0]**：突发传输长度 (Burst length)
0x00：最大突发传输长度（16 字 64 位，即 128 字节）
0x01：每突发传输 1 个字（64 位）
0x10：每突发传输 16 个字（每个字 64 位）
其他值：保留

31.7.39 LTDC 第 x 层平面配置寄存器 (LTDC_LxPCR)

LTDC layerx planar configuration register

偏移地址：0x130 + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

该寄存器定义访问平面或半平面缓冲区时第 x 层的配置。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	YREN	OF	CBF	YF	YCM[1:0]		YCEN	Res.	Res.	Res.
						rw	rw	rw	rw	rw	rw	rw			

位 31:10 保留，必须保持复位值。

位 9 **YREN**：Y 重新调整颜色动态范围使能 (Y rescale enable for the color dynamic range)
如果使能，则输入的范围为 16 到 235 的 Y 值重新调整为 0 到 255 的范围。
0：禁止重新调整（假设输入分量为 0 到 255）
1：使能重新调整（假设输入分量为 16 到 235）。

位 8 **OF**：奇数像素在先 (Odd pixel first)
该位定义字的字节 0（LSB 中）是否包含奇数像素。
0：禁止奇数像素（字节 0 为偶数像素）
1：使能奇数像素（字节 0 为奇数像素）

位 7 **CBF**：Cb 分量在先 (Cb component first)

该位定义字的字节 0 和 1（LSB 中）是否包含 Cb 分量。该设置仅对交错和半平面模式有影响，在全平面模式中无意义。

0：禁止 Cb（Cr 分量在字节 0 和 1 中）

1：使能 Cb（Cb 分量在字节 0 和 1 中）

位 6 **YF**：Y 分量在先 (Y component first)

该位定义字的字节 0（LSB 中）是否包含 Y 分量。

0：禁止 Y 分量（Cr 或 Cb 分量在字节 0 中）

1：使能 Y 分量（Y 分量在字节 0 中）

位 5:4 **YCM[1:0]**：YCbCr 转换模式 (YCbCr conversion mode)

该位字段定义应考虑并转换为 YCbCr 444 的输入类型。

00：交错 422（水平复制 Cb 和 Cr 分量以形成像素 P 和 P+1）

01：半平面 420：（水平和垂直复制 Cb 和 Cr 分量。图层的主要配置定义了对 Y 缓冲区的访问，辅助寄存器定义了对 Cb 和 Cr 缓冲区的访问）。

10：全平面 420：（水平和垂直复制 Cb 和 Cr 分量。图层的主要配置定义了对 Y 缓冲区的访问，辅助寄存器定义了对 Cb 和 Cr 缓冲区的访问）。

11：保留。

位 3 **YCEN**：YCbCr 到 RGB 转换使能 (YCbCr-to-RGB conversion enable)

0：禁止转换

1：使能 YCbCr 转换（使用上述 YCM 设置）

位 2:0 保留，必须保持复位值。

31.7.40 LTDC 第 x 层颜色帧缓冲区地址寄存器 (LTDC_LxCFBAR)

LTDC layerx color frame buffer address register

偏移地址：0x134 + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

该寄存器定义颜色帧缓冲区的起始地址，该地址必须指向帧缓冲区中存储的层的左上角像素的像素数据地址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CFBADD[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFBADD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CFBADD[31:0]**：颜色帧缓冲区起始地址 (Color frame buffer start address)

该位字段定义颜色帧缓冲区的起始地址。

31.7.41 LTDC 第 x 层颜色帧缓冲区长度寄存器 (LTDC_LxCFBLR)

LTDC layerx color frame buffer length register

偏移地址：0x138 + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

该寄存器定义颜色帧缓冲区的行长和行间距。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CFBP[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CFBLL[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **CFBP[11:0]**：颜色帧缓冲区间距（以字节为单位）(Color frame buffer pitch in bytes)
该位字段定义从像素某行的起始处到下一行的起始处的增量（以字节为单位）。
允许负值（MSB 位 = 1），以从底部向顶部读取缓冲区，从而进行垂直翻转。垂直翻转时，地址寄存器必须提供要读取的第一行的地址，地址寄存器必须指向缓冲区底部行的起始位置。

位 15:12 保留，必须保持复位值。

位 11:0 **CFBLL[11:0]**：颜色帧缓冲区行长度 (Color frame buffer line length)
该位字段定义一行像素的长度（字节数）+ 7。
行长的计算方法为：有效宽度 x 每像素的字节数 + 7。

31.7.42 LTDC 第 x 层颜色帧缓冲区行数寄存器 (LTDC_LxCFBLNR)

LTDC layerx color frame buffer line number register

偏移地址：0x13C + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

该寄存器定义颜色帧缓冲区中的行数。行数和行长设置定义了每层从每帧中获取的数据量。如果配置为低于所需字节数，则会产生 FIFO 下溢中断（如果使能）。起始地址和间距设置定义内存中每行的正确起始位置。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CFBLNBR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:0 **CFBLNBR[11:0]**：帧缓冲区行数 (Frame buffer line number)
该位字段定义帧缓冲区中的行数，行数跟有效高度对应。

31.7.43 LTDC 第 1 层辅助帧缓冲区地址 0 寄存器 (LTDC_L1AFBA0R)

LTDC layer1 auxiliary frame buffer address 0 register

偏移地址：0x140

复位值：0x0000 0000

该寄存器定义辅助 0 帧缓冲区的起始地址。该地址必须指向帧缓冲区中存储的层的左上角像素的像素数据地址。辅助 0 和 1 帧缓冲区用于 YCbCr 半平面或全平面像素格式：颜色帧缓冲区用于存储 Y 分量，辅助缓冲区 0 和 1 用于存储另一个（半平面）或两个（全平面）平面分量。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFBADD0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFBADD0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **AFBADD0[31:0]**：帧缓冲区起始地址 (Frame buffer start address)
该位字段定义颜色帧缓冲区的起始地址。

31.7.44 LTDC 第 1 层辅助帧缓冲区地址 1 寄存器 (LTDC_L1AFBA1R)

LTDC layer1 auxiliary frame buffer address 1 register

偏移地址：0x144

复位值：0x0000 0000

该寄存器定义辅助 1 帧缓冲区的起始地址。该地址必须指向帧缓冲区中存储的层的左上角像素的像素数据地址（辅助 0 和 1 帧缓冲区用于 YCbCr 半平面或全平面像素格式）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFBADD1[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFBADD1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **AFBADD1[31:0]**：辅助帧缓冲区起始地址 (Auxiliary frame buffer start address)
该位字段定义颜色帧缓冲区的起始地址。

31.7.45 LTDC 第 1 层辅助帧缓冲区长度寄存器 (LTDC_L1AFBLR)

LTDC layer1 auxiliary frame buffer length register

偏移地址：0x148

复位值：0x0000 0000

该寄存器定义辅助 0 和 1 帧缓冲区的行长度和间距（辅助 0 和 1 帧缓冲区用于 YCbCr 半平面或全平面像素格式）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	AFBP[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	AFBLL[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **AFBP[11:0]**：辅助帧缓冲区间距（以字节为单位）(Auxiliary frame buffer pitch in bytes)
该位字段定义从像素某行的起始处到下一行的起始处的增量（以字节为单位）。
允许负值（MSB 位 = 1），以从底部向顶部读取缓冲区，从而进行垂直翻转。垂直翻转时，地址寄存器必须提供要读取的第一行的地址，地址寄存器必须指向缓冲区底部行的起始位置。

位 15:12 保留，必须保持复位值。

位 11:0 **AFBLL[11:0]**：辅助帧缓冲区行长 (Auxiliary frame buffer line length)
该位字段定义一行像素的长度（字节数）+ 7。
行长的计算方法为：
有效宽度 * 每像素的字节数 + 7。

31.7.46 LTDC 第 1 层辅助帧缓冲区行数寄存器 (LTDC_L1AFBLNR)

LTDC layer1 auxiliary frame buffer line number register

偏移地址：0x14C

复位值：0x0000 0000

该寄存器定义辅助 0 和 1 帧缓冲区中的行数（辅助 0 和 1 帧缓冲区用于 YCbCr 半平面或全平面像素格式）。行数和行长设置定义了每层从每帧中获取的数据量。如果配置为低于所需字节数，则会产生 FIFO 下溢中断（如果使能）。起始地址和间距设置定义内存中每行的正确起始位置。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	AFBLNBR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:0 **AFBLNBR[11:0]**：辅助帧缓冲区行数 (Auxiliary frame buffer line number)
该位字段定义辅助帧缓冲区中的行数，行数与有效高度对应。

31.7.47 LTDC 第 x 层 CLUT 写寄存器 (LTDC_LxCLUTWR)

LTDC layerx CLUT write register

偏移地址：0x150 + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

该寄存器定义 CLUT 地址和 RGB 值。CLUT 写寄存器必须只在消隐周期内或在层禁止时配置。CLUT 可通过 LTDC_LxCR 使能和禁止。CLUT 仅对 L8、AL44 和 AL88 像素格式有意义。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLUTADD[7:0]								RED[7:0]							
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GREEN[7:0]								BLUE[7:0]							
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:24 **CLUTADD[7:0]**：CLUT 地址 (CLUT address)
该位字段配置每个 RGB 值的 CLUT 地址（CLUT 内的颜色位置）。

位 23:16 **RED[7:0]**：红色值 (Red value)
该位字段配置红色值。

位 15:8 **GREEN[7:0]**：绿色值 (Green value)
该位字段配置绿色值。

位 7:0 **BLUE[7:0]**：蓝色值 (Blue value)
该位字段配置蓝色值。

31.7.48 LTDC 第 x 层转换 YCbCr RGB 0 寄存器 (LTDC_LxCYR0R)

LTDC layerx conversion YCbCr RGB 0 register (LTDC_LxCYR0R)

偏移地址：0x16C + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

该寄存器定义从 YCbCr 转换为 RGB 的系数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	CB2B[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CR2R[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25:16 **CB2B[9:0]**：Cb 到蓝色的转换系数 (Cb-to-Blue coefficient)，位 9:8 为正整数，位 7:0 为小数。

位 15:10 保留，必须保持复位值。

位 9:0 **CR2R[9:0]**：Cr 到红色的转换系数 (Cr-to-Red coefficient)，位 9:8 为正整数，位 7:0 为小数。

31.7.49 LTDC 第 x 层转换 YCbCr RGB 1 寄存器 (LTDC_LxCYR1R)

LTDC layerx conversion YCbCr RGB 1 register (LTDC_LxCYR1R)

偏移地址：0x170 + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

该寄存器定义从 YCbCr 转换为 RGB 的系数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	CB2G[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CR2G[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25:16 **CB2G[9:0]**：Cb 到绿色的转换系数 (Cb-to-Green coefficient)，位 9:8 为正整数，位 7:0 为小数。

位 15:10 保留，必须保持复位值。

位 9:0 **CR2G[9:0]**：Cr 到绿色的转换系数，位 9:8 为正整数，位 7:0 为小数。

31.7.50 LTDC 第 x 层灵活像素格式 0 寄存器 (LTDC_LxFPF0R)

LTDC layerx flexible pixel format 0 register

偏移地址：0x174 + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0002 1100

该寄存器是定义灵活颜色格式的寄存器之一，定义方法是指定内存字中所有 A、R、G、B 分量的位置和宽度。A 和 R 分量在该特定寄存器中定义。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RLEN[3:2]	
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RLEN[1:0]		RPOS[4:0]					ALEN[3:0]				APOS[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:18 保留，必须保持复位值。

位 17:14 **RLEN[3:0]**：红色分量的宽度 (Width of the red component)（以位为单位）

位 13:9 **RPOS[4:0]**：像素内存字中红色分量的位置 (Location of the red component inside the pixel memory word)（以位为单位）

位 8:5 **ALEN[3:0]**：Alpha 分量的宽度 (Width of the alpha component)（以位为单位）

位 4:0 **APOS[4:0]**：像素内存字中 alpha 分量的位置 (Location of the alpha component inside the pixel memory word)（以位为单位）

31.7.51 LTDC 第 x 层灵活像素格式 1 寄存器 (LTDC_LxFPF1R)

LTDC layerx flexible pixel format 1 register

偏移地址：0x178 + 0x100 * (x - 1)（x = 1 到 2）

复位值：0x0012 3110

该寄存器是定义灵活颜色格式的寄存器之一，定义方法是指定内存字中所有 A、R、G、B 分量的位置和宽度。G 和 B 分量在此特定寄存器中定义。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSIZE[2:0]			BLEN[3:2]	
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLEN[1:0]		BPOS[4:0]					GLEN[3:0]				GPOS[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值。

位 20:18 **PSIZE[2:0]**：像素大小 (Pixel size)（以字节为单位）

位 17:14 **BLEN[3:0]**：蓝色分量的宽度 (Width of the blue component)（以位为单位）

位 13:9 **BPOS[4:0]**：像素内存字中蓝色分量的位置 (Location of the blue component inside the pixel memory word)（以位为单位）

位 8:5 **GLEN[3:0]**：绿色分量的宽度 (Width of the green component)（以位为单位）

位 4:0 **GPOS[4:0]**：像素内存字中绿色分量的位置 (Location of the green component inside the pixel memory word)（以位为单位）

31.7.52 LTDC 寄存器映射

表 266. LTDC 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	LTDC_IDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJVER[7:0]								MINVER[7:0]								REV[7:0]							
	Reset value									0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	LTDC_LCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LNBR7:0]							
	Reset value																									0	0	0	0	0	0	1	1
	Reset value																									0	0	0	0	0	0	1	0
0x008	LTDC_SSCR	Res.	Res.	Res.	Res.	Res.	HSW[11:0]											Res.	Res.	Res.	Res.	VSH[11:0]											
	Reset value						0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x00C	LTDC_BPCR	Res.	Res.	Res.	Res.	AHBP[11:0]												Res.	Res.	Res.	Res.	AVBP[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x010	LTDC_AWCR	Res.	Res.	Res.	Res.	AAW[11:0]												Res.	Res.	Res.	Res.	AAH[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x014	LTDC_TWCR	Res.	Res.	Res.	Res.	TOTALW[11:0]												Res.	Res.	Res.	Res.	TOTALH[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0

表 266. LTDC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x018	LTDC_GCR	HSPOL	VSPOL	DEPOL	PCPOL	Res.	Res.	SFSWTR	SFEN	Res.	Res.	Res.	Res.	CRCEN	Res.	Res.	DEN	Res.	DRW[2:0]			Res.	DGW[2:0]			Res.	DBW[2:0]			Res.	Res.	GAMEN	LTDCEN
	Reset value	0	0	0	0			0	0					0			0		0	1	0		0	1	0		0	1	0			0	0
0x01C	LTDC_GC1R	BMA	CRMA	STRA	DWP	SPP	Res.	TP	LNIP	BBA	BCP	SHRA	Res.	GCT[2:0]			Res.	DT [1:0]		Res.	PRBA	WRCH[3:0]				WGCH[3:0]				WBCH[3:0]			
	Reset value	0	1	1	0	1		1	1	1	1	1		0	1	0		1	1		1	1	0	0	0	1	0	0	0	1	0	0	0
0x020	LTDC_GC2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOA	Res.	CRCA	SFA	SISA	ROTA	AXIDA	OCA	EDCA	BW [2:0]			DPA	DVA	STSA	BLA
	Reset value	0x0000 BB30																															
0x024	LTDC_SRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBR	IMR
	Reset value																															0	0
0x028	LTDC_GCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REN	GEN	BEN	COMP[7:0]								ADDR[7:0]							
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x02C	LTDC_BCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BCRED[7:0]								BCGREEN[7:0]								BCBLUE[7:0]							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x030	Reserved	Reserved																															
0x034	LTDC_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCIE	FUIE	Res.	Res.	RRIE	TERRIE	FUWIE	LIE
	Reset value																									0	0			0	0	0	0
0x038	LTDC_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCIF	FUIF	Res.	Res.	RRIF	TERRIF	FUWIF	LIF
	Reset value																									0	0			0	0	0	0
0x03C	LTDC_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCRCIF	CFUIF	Res.	Res.	CRRIF	CTERRIF	CFUWIF	CLIF
	Reset value																									0	0			0	0	0	0
0x040	LTDC_LIPCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LIPOS[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x044	LTDC_CPSR	Res.	Res.	Res.	Res.	CXPOS[11:0]												Res.	Res.	Res.	Res.	CYPOS[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x048	LTDC_CDSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSYNCS	VSYNCS	HDES	VDES
	Reset value																													0	0	1	1
0x04C-0x05C	Reserved	Reserved																															
0x060	LTDC_EDCR	Res.	Res.	Res.	Res.	OCYCO	OCYSEL	OCYEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value					0	0	0																									
0x064	LTDC_IE2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCIE	FUIE	Res.	Res.	RRIE	TERRIE	FUWIE	LIE
	Reset value																									0	0			0	0	0	0

表 266. LTDC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x068	LTDC_ISR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCIF	FUIF	Res.	Res.	RRIF	TERRIF	FUWIF	LIF
	Reset value																									0	0			0	0	0	0
0x06C	LTDC_ICR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCRCIF	CFUIF	Res.	Res.	CRRIF	CTERRIF	CFUWIF	CLIF
	Reset value																									0	0			0	0	0	0
0x070	LTDC_LIPCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LIPOS[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x074	Reserved	Reserved																															
0x078	LTDC_ECRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECRC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07C	LTDC_CCRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCRC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x080-0x08C	Reserved	Reserved																															
0x090	LTDC_FUTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	THRE[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x094-0x0FC	Reserved	Reserved																															
0x100	LTDC_L1C0R	ARGB8888	ABGR8888	RGBA8888	BGRA888	RGB565	BGR565	RGB888	FF	F11PC	F1PC	F11C	F1C	F11P	F1P	F10	F11	F21PC	F2PC	F21C	F2C	F21P	F2P	F20	F21	CKRA	CLUTA	WINA	DCP	APA	CFBPA	CFBDA	CKTA
	Reset value	1	1	1	1	1	1	1	1	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1	0	1	0	1
0x104	LTDC_L1C1R	SCA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YFPA	YSPA	YIA
	Reset value	0																													1	1	1
0x108	LTDC_L1RCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GRMSK	VBR	IMR
	Reset value																														1	0	0
0x10C	LTDC_L1CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCBEN	HMEN	Res.	Res.	Res.	CLUTEN	HPDEN	VPDEN	CKEN	LEN
	Reset value																							0	0				0	0	0	0	0
0x110	LTDC_L1WHPCR	Res.	Res.	Res.	Res.	WHSPPOS[11:0]												Res.	Res.	Res.	Res.	Res.	WHSTPOS[11:0]										
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x114	LTDC_L1WVPCR	Res.	Res.	Res.	Res.	WVSPPOS[11:0]												Res.	Res.	Res.	Res.	WVSTPOS[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x118	LTDC_L1CKCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKRED[7:0]								CKGREEN[7:0]								CKBLUE[7:0]							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x11C	LTDC_L1PFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PF[2:0]		
	Reset value																														0	0	0
0x120	LTDC_L1CACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CONSTA[7:0]							
	Reset value																									1	1	1	1	1	1	1	1

表 266. LTDC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x124	LTDC_L1DCCR	DCALPHA[7:0]								DCRED[7:0]								DCGREEN[7:0]								DCBLUE[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x128	LTDC_L1BFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOR	Res.	Res.	Res.	Res.	Res.	BF1[2:0]			Res.	Res.	Res.	Res.	Res.	BF2[2:0]		
	Reset value																0						1	1	0						1	1	1
0x12C	LTDC_L1BLCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BL[4:0]				
	Reset value																												0	0	0	0	0
0x130	LTDC_L1PCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YREN	OF	CBF	YF	YCM [1:0]		YCEN	Res.	Res.	Res.
	Reset value																							0	0	0	0	0	0	0			
0x134	LTDC_L1CFBAR	CFBADD[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x138	LTDC_L1CFBLR	Res.	Res.	Res.	Res.	CFBP[11:0]												Res.	Res.	Res.	Res.	CFBLL[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x13C	LTDC_L1CFBLNR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFBLNBR[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x140	LTDC_L1AFBA0R	AFBADD0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x144	LTDC_L1AFBA1R	AFBADD1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x148	LTDC_L1AFBLR	Res.	Res.	Res.	Res.	AFBP[11:0]												Res.	Res.	Res.	Res.	AFBLL[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x14C	LTDC_L1AFBLNR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AFBLNBR[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x150	LTDC_ L1CLUTWR	CLUTADD[7:0]								RED[7:0]								GREEN[7:0]								BLUE[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x154-0x168	Reserved	Reserved																															
0x16C	LTDC_L1CYR0R	Res.	Res.	Res.	Res.	Res.	Res.	CB2B[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	CR2R[9:0]									
	Reset value							0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x170	LTDC_L1CYR1R	Res.	Res.	Res.	Res.	Res.	Res.	CB2G[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	CR2G[9:0]									
	Reset value							0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x174	LTDC_L1FPF0R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RLEN[3:0]				RPOS[4:0]					ALEN[3:0]				APOS[4:0]				
	Reset value															1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
0x178	LTDC_L1FPF1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSIZE [2:0]			BLEN[3:0]				BPOS[4:0]					GLEN[3:0]				GPOS[4:0]				
	Reset value												1	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	1	0	0	0	0
0x17C-0x1FC	Reserved	Reserved																															
0x200	LTDC_L2C0R	ARGB8888	ABGR8888	RGBA8888	BGRA888	RGB565	BGR565	RGB888	FF	F11PC	F1PC	F11C	F1C	F11P	F1P	F10	F11	F21PC	F2PC	F21C	F2C	F21P	F2P	F20	F21	CKRA	CLUTA	WINA	DCP	APA	CFBPA	CFBDA	CKTA
	Reset value	1	1	1	1	1	1	1	1	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1	0	1	0	1

表 266. LTDC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x204	LTDC_L2C1R	SCA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YFPA	YSPA	YIA
	Reset value	0																													0	0	1
0x208	LTDC_L2RCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GRMSK	VBR	IMR
	Reset value																														1	0	0
0x20C	LTDC_L2CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCBEN	HMEN	Res.	Res.	Res.	CLUTEN	HPDEN	VPDEN	CKEN	LEN
	Reset value																							0	0				0	0	0	0	0
0x210	LTDC_L2WHPCR	Res.	Res.	Res.	Res.	WHSPPOS[11:0]												Res.	Res.	Res.	Res.	Res.	WHSTPOS[11:0]										
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x214	LTDC_L2WVPCR	Res.	Res.	Res.	Res.	WVSPPOS[11:0]												Res.	Res.	Res.	Res.	WVSTPOS[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x218	LTDC_L2CKCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKRED[7:0]								CKGREEN[7:0]								CKBLUE[7:0]							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x21C	LTDC_L2PFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PF[2:0]		
	Reset value																														0	0	0
0x220	LTDC_L2CACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CONSTA[7:0]							
	Reset value																									1	1	1	1	1	1	1	1
0x224	LTDC_L2DCCR	DCALPHA[7:0]								DCRED[7:0]								DCGREEN[7:0]								DCBLUE[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x228	LTDC_L2BFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOR	Res.	Res.	Res.	Res.	Res.	BF1[2:0]			Res.	Res.	Res.	Res.	Res.	BF2[2:0]		
	Reset value																0						1	1	0						1	1	1
0x22C	LTDC_L2BLCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BL[4:0]				
	Reset value																												0	0	0	0	0
0x230	LTDC_L2PCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YREN	OF	CBF	YF	YCM[1:0]	YCEN	Res.	Res.	Res.	
	Reset value																							0	0	0	0	0	0	0			
0x234	LTDC_L2CFBAR	CFBADD[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x238	LTDC_L2CFBLR	Res.	Res.	Res.	Res.	CFBP[11:0]												Res.	Res.	Res.	Res.	CFBLL[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x23C	LTDC_L2CFBLNR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFBLNBR[11:0]												Res.
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	
0x240-0x24C	Reserved	Reserved																															
0x250	LTDC_L2CLUTWR	CLUTADD[7:0]								RED[7:0]								GREEN[7:0]								BLUE[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x254-0x268	Reserved	Reserved																															
0x26C	LTDC_L2CYR0R	Res.	Res.	Res.	Res.	Res.	Res.	CB2B[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	CR2R[9:0]									
	Reset value							0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0

表 266. LTDC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x270	LTDC_L2CYR1R	Res.	Res.	Res.	Res.	Res.	Res.	CB2G[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	CR2G[9:0]									
	Reset value							0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x274	LTDC_L2FPF0R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RLEN[3:0]				RPOS[4:0]				ALEN[3:0]				APOS[4:0]					
	Reset value															1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
0x278	LTDC_L2FPF1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSIZE [2:0]			BLEN[3:0]				BPOS[4:0]				GLEN[3:0]				GPOS[4:0]					
	Reset value												1	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	1	0	0	0	0

有关寄存器边界地址的信息，请参见*第 2.5 节*。

32 真随机数发生器 (RNG)

32.1 简介

RNG 是一个真随机数发生器，它以 32 位样本的形式向应用程序提供全熵输出。它由一个实时熵源（模拟）和一个内部调节组件构成。

RNG 是符合 NIST SP 800-90B 标准的熵源，可用于构建非确定性随机位发生器 (NDRBG)。

RNG 真随机数发生器已通过 NIST SP800-90B 标准预先认证，也已按照德国 BSI AIS-31（T0 到 T8）统计测试进行了测试。

32.2 RNG 主要特性

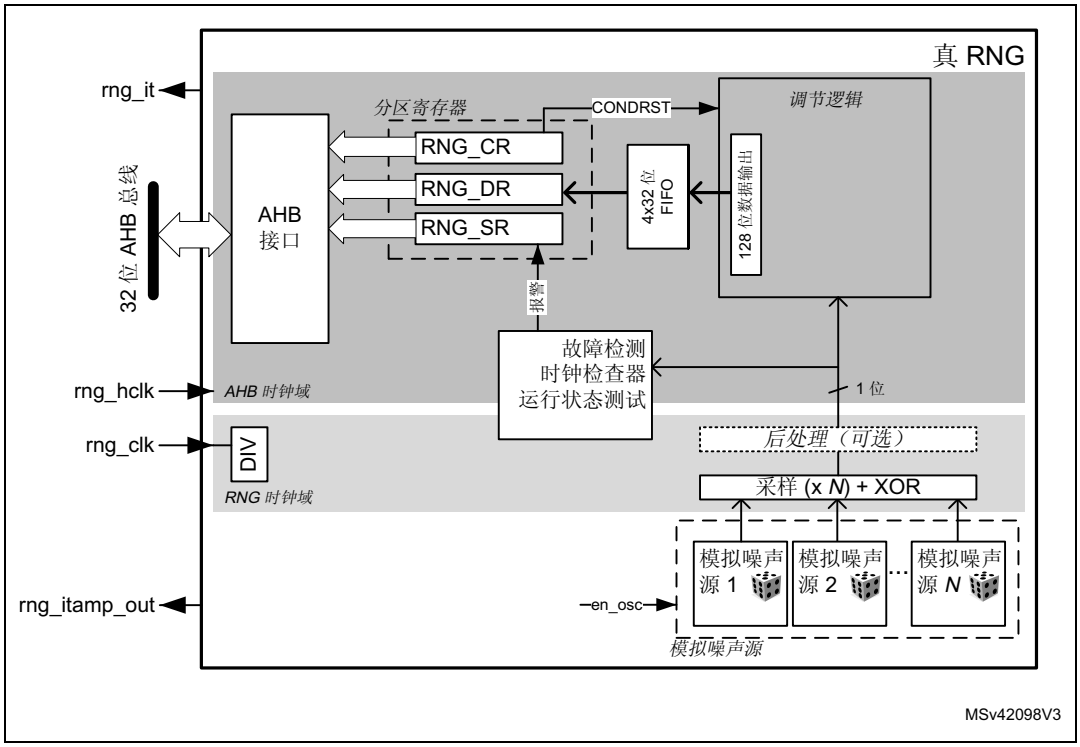
- RNG 提供的 32 位真随机数由模拟熵源生成，并通过符合 NIST SP800-90B 标准要求的调节级进行调节。
- 可用作熵源以构建非确定性随机位发生器 (NDRBG)。
- 在 NIST 配置中，如果 $f_{AHB} < f_{threshold}$，则 RNG 每 412 个 AHB 时钟周期会生成四个 32 位随机样本（否则为 256 个 RNG 时钟周期）。
- 内置启动和符合 NIST SP800-90B 标准认证的连续运行状态测试（重复计数和自适应比例测试）及相关的特定错误管理。
- 可被禁止以降低功耗，或以自动低功耗模式使能（默认配置）。
- 具有一个 AMBA® AHB 从设备，只能通过 32 位字进行单次访问（否则会生成 AHB 总线错误，并会忽略写访问）。

32.3 RNG 功能描述

32.3.1 RNG 框图

图 272 显示了 RNG 框图。

图 272. RNG 框图



32.3.2 RNG 内部信号

表 267 中列出了 RNG 级而非 STM32 产品级（焊盘上）所提供的内部信号，对这些信号有所了解会很有用。

表 267. RNG 内部输入/输出信号

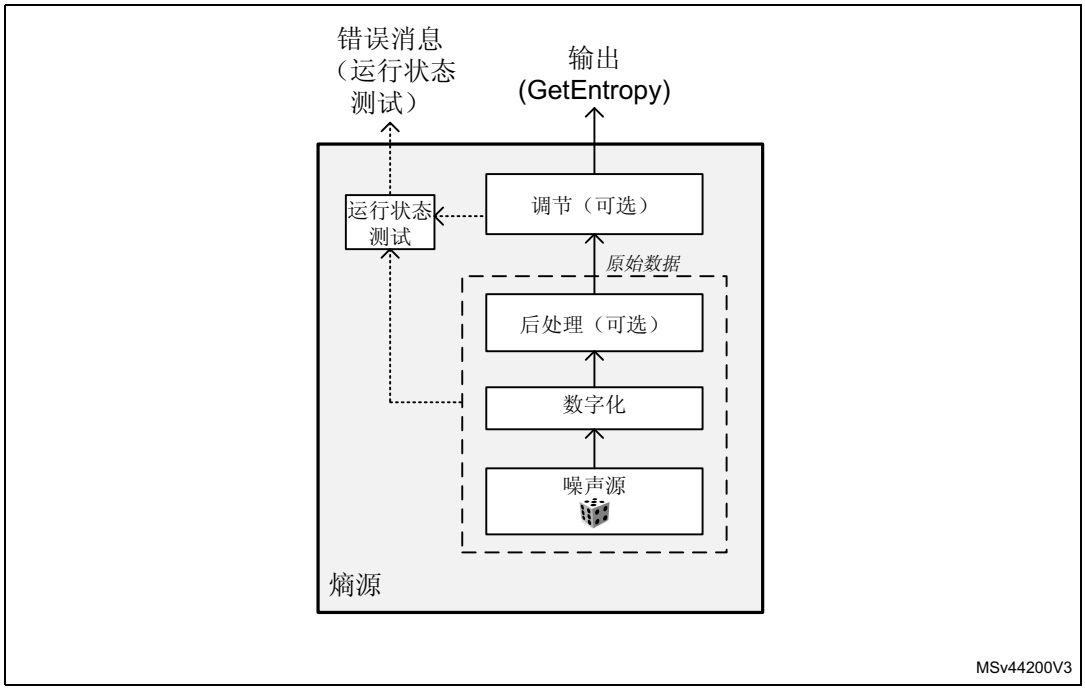
信号名称	信号类型	说明
rng_it	数字输出	RNG 全局中断请求
rng_hclk	数字输入	AHB 时钟
rng_clk	数字输入	RNG 专用时钟，与 rng_hclk 异步
rng_itamp_out	数字输出	发送到 TAMP 的 RNG 内部 Tamper 事件信号（异或运算），在发生意外硬件故障时触发。该信号被触发后，RNG 将停止提供随机样本，需要复位并重新初始化才能再次使用。

32.3.3 随机数生成

真随机数发生器 (RNG) 按确定的间隔通过其 AHB 接口提供真随机数。

RNG 在其边界内集成了所有必需的 NIST 组件（如*图 273* 所示）。这些组件包括模拟噪声源、数字化处理级、调节算法、运行状态监视模块以及两个用于与熵源交互的接口：GetEntropy 和 HealthTest。

图 273. NIST SP800-90B 熵源模型



下文将详细介绍上图所示的组件。

噪声源

噪声源组件包含不确定性的、能够提供熵值的活动，其最终决定了与熵源位串输出相关联的不确定性。该噪声源提供 1 位样本。它包括：

- 多个模拟噪声源 (x6)，每个噪声源由三个自由运行的环形振荡器输出异或 (XOR) 而成。可以禁止这些模拟振荡器以实现节能，如*第 32.3.8 节：RNG 低功耗使用*所述。
- 将所有噪声源的异或运算结果作为单个模拟输出。
- 该输出的采样级由专用时钟输入（rng_clk 通过集成分频器）提供时钟信号，输出 1 位原始数据。

该噪声源采样与 AHB 接口时钟频率 (rng_hclk) 无关，软件可通过使用集成分频器来降低采样频率。

注： *第 32.6 节：RNG 熵源验证中提供了建议使用的 RNG 时钟频率及相关分频值。*

后期处理

在 NIST 配置中，不会对采样的噪声源进行后期处理。在非 NIST 配置 B 中（如*第 32.6.2 节*中定义），会进行归一化去偏置处理，即一半的位取自采样噪声源，一半的位是对取自采样噪声源的值取反。

调节

RNG 中的调节组件是确定性函数，可以提高生成固定长度位串输出（128 位）的熵率。NIST SP800-90B 的目标是提供全熵输出（128 位）。

*第 32.5 节：RNG 处理时间*中给出了生成两个随机数之间和 RNG 初始化之间所需的时间，以及第一个样本的可用性的说明。

输出缓冲区

数据输出缓冲区最多可存储四个从调节组件输出的 32 位字。如果已通过 RNG_DR 寄存器从输出 FIFO 读取了四个字，则 128 位调节输出寄存器的内容会被推送到输出 FIFO 中，同时自动启动新的一轮调节。经过*第 32.5 节：RNG 处理时间*中指定的时钟周期数后，会将四个新字添加到调节输出寄存器。

只要通过 RNG_DR 寄存器获取随机数时，DRDY 标志就从 0 变为 1。该标志会保持在置 1 状态，直至从 RNG_DR 寄存器读取四个字后输出缓冲区变空。

注： *如果使能了中断，则当该数据就绪标志从 0 变为 1 时，会产生中断。随后，中断会自动由 RNG 清零，如上所述。*

运行状态检查

该组件可确保整个熵源（包括其噪声源）正常启动并按预期运行，能够快速准确地定位故障，从而提供了高可靠性保证。

根据 NIST SP800-90B 标准，RNG 可实现以下运行状态检查功能。规定的阈值对应于寄存器 RNG_HTCR 的建议值（*第 32.6.2 节*中的配置 A）。

1. 启动运行状态测试（在复位后和在首次将 RNG 用作熵源前执行）：
   - 重复计数测试，在噪声源持续不变地输出了 37 位或以上的 0 或 1 时指示错误。
   - 在 1024 个连续位窗口上运行的自适应比例测试：RNG 验证噪声源输出上的第一个位重复次数是否超过 669 次。
   - 已知答案测试，用于验证调节级。
2. 连续运行状态测试（无限期地在噪声源输出端运行）：
   - 重复计数测试，类似于启动测试中运行的测试。
   - 自适应比例测试，类似于启动测试中运行的测试。
3. 厂商专用连续测试
   - 转换计数测试，当噪声源持续不变地输出了 32 位或以上的 01 或 10 时指示错误。
   - 实时“过慢”采样时钟检测器，如果一个 RNG 时钟周期（分频前）小于 AHB 时钟周期的 32 分之一，则该检测器会指示错误。
4. 数字化噪声源按需测试（原始数据）
   - 通过重新启动熵源并重新运行启动测试来提供支持（请参见*第 32.3.4 节：RNG 初始化*中的软件复位序列）。*不支持*其他类型的按需测试（基于软件）。

RNG_SR 寄存器中的 CECS 和 SECS 状态位会在检测到错误条件时进行指示，*第 32.3.7 节：错误管理*对此进行了详细说明。

注： *检测到错误时可产生中断。*

可通过更改 RNG_HTCR 寄存器中的值来修改超出运行状态测试阈值的值。有关详细信息，请参见第 32.6 节：RNG 熵源验证。

32.3.4 RNG 初始化

RNG 简化状态机如*图 274* 所示

在使能 RNG（RNG_CR 中的 RNGEN = 1）后，会发生以下一系列事件：

1. 使能模拟噪声源，默认情况下，RNG 在等待 16 个 RNG 时钟周期（分频前）后开始采样模拟输出并填充 128 位调节移位寄存器。
2. 调节硬件初始化，自动触发对原始数据样本的启动行为测试和已知答案测试。
3. 启动运行状态测试完成。在此期间，使用三个 128 位噪声源样本。
4. 再次使用 128 位数据填充调节级内部输入数据缓冲区并执行调节，调节轮数由 RNG 配置（NIST 或非 NIST）定义。然后使用后期处理结果填充输出缓冲区。
5. 输出缓冲区会按照 RNG 使用情况自动重填。

相关初始化时间请参见*第 32.5 节：RNG 处理时间*。

图 274. RNG 初始化概述

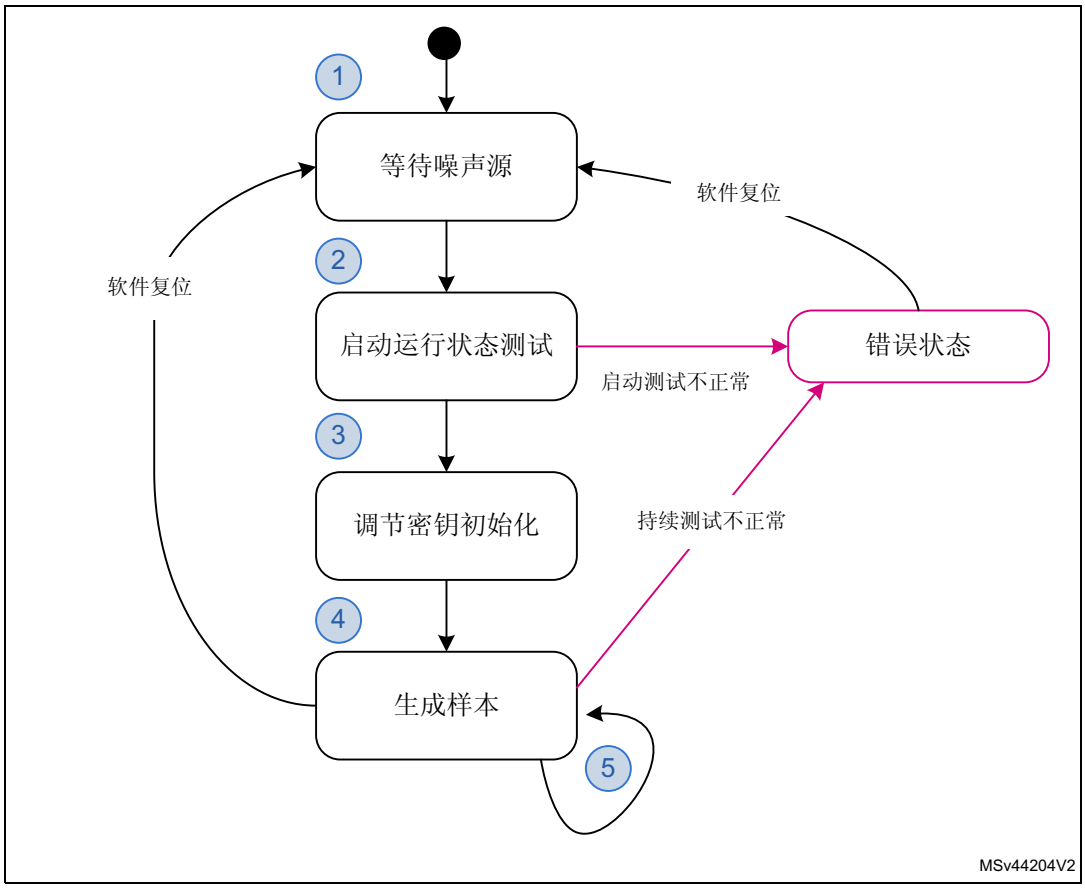


图 274 还重点说明了可能的软件复位序列，其实现方式如下：

1. 向 RNG_CR 寄存器中写入 RNGEN = 0 和 CONDRST = 1，并使用相同的 RNG 配置和新的 CLKDIV（如果需要）。
2. 然后，向 RNG_CR 寄存器中写入 RNGEN = 1 和 CONDRST = 0。
3. 初始化完成后，等待随机数就绪。

注： *当 RNG 外设通过 RCC（硬件复位）复位时，RNG 寄存器中可实现最佳随机性的 RNG 配置将丢失。通过将 CONFIGLOCK 置 1 来执行软件复位时可保留 RNG 配置。*

32.3.5 RNG 操作

正常工作

要使用中断运行 RNG，建议执行以下步骤：

1. 查阅*第 32.6 节：RNG 熵源验证*并验证应用是否需要特定的 RNG 配置。
   - 如果需要，则向 RNG_CR 寄存器中写入 CONDRST = 1 以及正确的 RNG 配置。然后，对 RNG_CR 寄存器执行第二次写操作：CONDRST = 0、中断使能位 IE = 1 且 RNG 使能位 RNGEN = 1。
   - 如果不需要，则对 RNG_CR 寄存器执行写操作：中断使能位 IE = 1 且 RNG 使能位 RNGEN = 1。
2. 这以后，准备好随机数时或出现错误时就产生中断。因此，每次发生中断时，应检查：
   - 未发生错误。RNG_SR 寄存器中的 SEIS 和 CEIS 位必须设为 0。
   - 随机数准备就绪。RNG_SR 寄存器中的 DRDY 位必须置 1。
   - 如果满足以上两个条件，最多可连续四次读取 RNG_DR 寄存器的内容。如果调节输出缓冲区中存在有效数据，则应用程序可以额外读取四个字（此时 DRDY 位仍保持为 1）。如果上述条件中有一个或两个都无法满足，则不得读取 RNG_DR 寄存器。如果发生错误，则必须使用*第 32.3.7 节*中介绍的错误恢复序列。

要在轮询模式下运行 RNG，建议执行以下步骤：

1. 查阅*第 32.6 节：RNG 熵源验证*并验证应用是否需要特定的 RNG 配置。
   - 如果需要，则向 RNG_CR 寄存器中写入 CONDRST = 1 以及相应的 RNG 配置。然后，对 RNG_CR 寄存器执行第二次写操作：CONDRST = 0 且 RNG 使能位 RNGEN = 1。
   - 如果不需要，则只需将 RNG_CR 寄存器中的 RNGEN 位置 1 即可使能 RNG。
2. 读取 RNG_SR 寄存器并检查：
   - 未发生错误（SEIS 和 CEIS 位必须设为 0）
   - 随机数准备就绪（DRDY 位必须置 1）
3. 如果满足以上两个条件，最多可连续四次读取 RNG_DR 寄存器的内容。如果调节输出缓冲区中存在有效数据，则应用程序可以额外读取四个字（此时 DRDY 位仍保持为 1）。如果上述条件中有一个或两个都无法满足，则不得读取 RNG_DR 寄存器。如果发生错误，则必须使用*第 32.3.7 节*中介绍的错误恢复序列。

注： *如果数据未就绪 (DRDY = 0)，则 RNG_DR 会返回 0。*
建议始终验证 RNG_DR 是否不为零。因为如果不为零，在轮询 RNG_SR 与读取 RND_DR 输出之间可能会发生种子错误（罕见事件）。

如果应用程序比较关注随机数生成周期，并且不需要符合 NIST 标准，则可以使用 RNG 配置“B”来选择更快的 RNG 配置，如*第 32.6 节：RNG 熵源验证*中所述。*第 32.5 节：RNG 处理时间*中总结了随机数生成的速度响应。

低功耗运行

如果应用程序比较关注功耗，则可以使用低功耗策略，如*第 32.3.8 节：RNG 低功耗使用*所述。

软件后期处理

无需进行特定的软件后期处理/调节即可满足 AIS-31 或 NIST SP800-90B 认证。

*第 32.3.3 节：随机数生成*介绍了内置的运行状态检查功能。

32.3.6 RNG 时钟

RNG 在两个不同的时钟上运行：AHB 总线时钟和专用 RNG 时钟。

AHB 时钟用于为 AHB 分组寄存器和调节组件提供时钟信号。RNG 时钟与可编程分频器搭配使用（请参见 RNG_CR 寄存器中的 CLKDIV 位字段），用于噪声源采样。*第 32.6 节：RNG 熵源验证*中详细介绍了建议使用的时钟配置。

注： *如果 RNG_CR 寄存器中的 CED 位置 0，那么 RNG 时钟频率（内部分频前）必须大于 AHB 时钟频率的 32 分之一，否则时钟检查器会始终指示时钟错误（RNG_SR 寄存器中的 CECS = 1）。*

有关 AHB 和 RNG 时钟域的详细信息，请参见*第 32.3.1 节：RNG 框图*。

32.3.7 错误管理

运行状态检查模块在生成随机数的同时还验证噪声源行为以及 RNG 源时钟频率是否正常，具体说明见本部分。另外，还描述了相关的错误状态。

时钟错误检测

如果时钟错误检测已使能 (CED = 0) 且 RNG 时钟频率过低，RNG 会将 CEIS 和 CECS 位都置 1，以指示发生了时钟错误。在这种情况下，应用程序必须检查 RNG 时钟是否配置正确（参见*第 32.3.6 节：RNG 时钟*），随后必须将中断标志 CEIS 位清零。当时钟状态正常时，CECS 位自动清零。

注： *时钟错误对生成的随机数没有影响，即应用程序仍然可以读取 RNG_DR 寄存器。*

仅当 RNG 将 CECS 置 1 时，CEIS 才会置 1。

噪声源错误检测

如果发生噪声源（或种子）错误，RNG 会停止生成随机数，并会将 SEIS 和 SECS 位都置 1，以指示发生了种子错误。如果 RNG_DR 寄存器中有可用值，不得使用该值，因为它可能没有足够的熵。

要从种子错误中完全恢复，必须使用以下序列：

1. 先后向 CONDRST 中写入 1 和 0 以执行软件复位（有关详细信息，请参见位字段说明）。仅当 SECS 置 1 时才需要执行此步骤。事实上，当 SEIS 置 1 且 SECS 清零时，意味着 RNG 会自动执行复位（自动复位）。在这种情况下，应用程序必须将中断标志位 SEIS 清零。
2. 如果在步骤 1 中将 SECS 置 1（不自动复位），则需等待 RNG_CR 寄存器中的 CONDRST 清零，然后确认 RNG_SR 寄存器中的 SEIS 是否清零。否则，只需将 RNG_SR 寄存器中的 SEIS 位清零即可。
3. 如果在步骤 1 中将 SECS 置 1（不自动复位），则需等待 RNG 将 SECS 清零。随机数生成于是恢复正常。

注： *发生种子错误后，当 SECS 清零时，RNG 会重新开始生成随机数。*

应用程序将 RNG_CR 寄存器中的 ARDIS 位置 1 后，将禁止自动复位。必须在步骤 1 中使用 CONDRST。

RNG Tamper 错误

当 RNG 发现意外错误时，TAMP 外设中将会触发内部 Tamper 事件，并且 RNG 将停止提供随机数据。

该事件发生后，安全应用程序需使用中央复位管理或全局 SoC 复位来复位 RNG 外设。之后，需要再次正确初始化 RNG。

32.3.8 RNG 低功耗使用

考虑到功耗问题，可在 DRDY 位置 1 后立即禁止 RNG，具体方法为将 RNG_CR 寄存器中的 RNGEN 位设为 0。由于 RNGEN = 0 时后期处理逻辑和输出缓冲区仍处于工作状态，因此软件可使用以下功能：

- 如果输出缓冲区中存在有效字，仍可从 RNG_DR 寄存器中读取四个随机数。
- 如果调节输出内部寄存器中存在有效位，仍可从 RNG_DR 寄存器中额外读取四个随机数。如果不属于以上两种情况，则必须由应用程序重新使能 RNG，直至达到预期的新噪声源位阈值（NIST 模式下为 128 位）并且完成了一轮完整的调节。
  只有达到预期的调节轮数时（NISTC = 0 时为两轮），四个新的随机字才可用。相关总时间请参见*第 1678 页的第 32.5 节：RNG 处理时间*。

禁止 RNG 时，用户会禁用所有模拟种子发生器，各发生器的功耗列于数据手册中的电气特性部分。用户还会对所有由 RNG 时钟提供时钟信号的逻辑系统进行门控。请注意，由于 RNG 需要一定的时间进行初始化，因此该策略导致在一段延时后才可在 RNG_DR 寄存器中提供随机样本。

如果 RNG 模块在初始化期间（也就是刚好在 DRDY 位首次置 1 之前）被禁止，则初始化序列将从 RNGEN 位置 1 时停止的位置恢复运行，但应用程序使用 RNG_CR 寄存器中的 CONDRST 位复位调节逻辑时除外。

当应用程序希望禁止 RNG 时钟时，建议在 RNGEN 位清零后等待两个 RNG 内核时钟周期，然后再使用 RCC 禁止 RNG 内核时钟。

此外，当应用程序需要进入 RNG 停用的功耗模式时，也建议在 RNGEN 位清零后等待两个 RNG 内核时钟周期，然后再使用 PWR 进入低功耗模式。

在上述两种情况下，为避免在 RNG 模拟振荡器保持工作状态时产生意外消耗，应用程序可将 RNG_CR 寄存器中的位 13 置 1。当 RNGEN 位置 1（RNG 激活）时，将该位置 1 会增加一些边际功耗。

注： *有关 RNG 停用（保留或不可用）的功耗模式，请参见 PWR 部分。*

32.4 RNG 中断

在 RNG 中，发生以下事件时会产生中断：

- 数据就绪标志
- 种子错误，请参见 *第 32.3.7 节：错误管理*
- 时钟错误，请参见 *第 32.3.7 节：错误管理*

可以使用专用的中断使能控制位，如*表 268* 所示。

表 268. RNG 中断请求

中断缩写	中断事件	事件标志	使能控制位	中断清除方法
RNG	数据就绪标志	DRDY	IE	无（自动）
	种子错误标志	SEIS	IE	除非 ARDIS 清零（参见*第 32.3.7 节：错误管理*），否则向 CONDRST 写入 1 然后写入 0
	时钟错误标志	CEIS	IE	向 CEIS 写入 0

用户可以通过更改 RNG_CR 寄存器中的屏蔽位或通用中断控制位 IE 的方式分别使能或禁止上述中断源。可以从 RNG_SR 寄存器中读取各个中断源的状态。

注： *仅当 RNG 已使能时，才会产生中断。*

32.5 RNG 处理时间

在*表 269* 中介绍的建议配置 A 或 C 中，两组四个 32 位数据之间的时间为：

- 206 x N 个 AHB 周期（如果 $f_{AHB} < f_{threshold}$（调节级实施限制）），或者
- 128 x N 个 RNG 周期（如果 $f_{AHB} \geq f_{threshold}$（噪声源级实施限制））。

其中 $f_{threshold} = 1.6 \times f_{RNG}$，例如 f_{RNG}= 48 MHz 时，该频率为 77 MHz。对于配置 A 和配置 C，值 N 均为 2。

注： *当 CLKDIV 不等于 0 时，f_{RNG} 必须考虑内部分频比。*

如果选择配置 B，则性能数据将变为：

- 206 个 AHB 周期（如果 $f_{AHB} < f_{threshold}$），或者
- 32 个 RNG 周期（如果 $f_{AHB} \geq f_{threshold}$）

其中 $f_{threshold} = 6.5 \times f_{RNG}$。

32.6 RNG 熵源验证

32.6.1 简介

为了评估 RNG 可提供的熵大小，意法半导体通过德国 BSI AIS-31 统计测试（T0 到 T8）以及 NIST SP800-90B 测试套件对外设进行了测试。

32.6.2 验证条件

意法半导体在以下条件下对 RNG 真随机数发生器进行了测试：

- RNG 时钟 rng_clk = 48 MHz
- RNG 配置如*表 269：RNG 配置*所述。请注意，只有配置 A 可通过 NIST SP800-90B 标准认证。有关选择应用最佳配置的信息，请参见*表 270：配置选择*。

表 269. RNG 配置

RNG 配置	RNG_CR 位						轮数 (N)	RNG_HTCR 寄存器	RNG_NCSR 寄存器
	NISTC 位	RNG_CONFIG1 [5:0]	CLKDIV [3:0]	RNG_CONFIG2 [2:0]	RNG_CONFIG3 [3:0]	CED 位			
A	0	0x0F	0x0(1)	0x0(2)	0xD	0	2	0x0000 969D(3)	0x0002 B5BB
B	1	0x18	0x0	0x0	0x0	0	1	0x0000 AAC7(4)	0x0003 FFFF
C	0	0x0F	0x0	0x0	0xD	0	2	0x0000 AAC7(4)	0x0003 FFFF

1. 对于 NIST 认证，噪声源采样频率必须为 48 MHz 或更低。因此，如果 RNG 时钟频率不等于 48 MHz，则必须调整 CLKDIV 的值。有关详细信息，请参见*第 32.7.1 节*中的 CLKDIV 位字段说明。
2. 对于 RNG 功耗极为敏感的情况，建议使用值 0x1。有关详细信息，请参见*第 32.3.8 节：RNG 低功耗使用*结尾部分。
3. 对应于 37（重复计数测试）和 669（自适应比例测试）。有关详细信息，请参见*第 1673 页的运行状态检查*。
4. 该值在 RNG 驱动程序中可以是固定值（不取决于 STM32 系列）。

表 270. 配置选择

部分标准	配置 A	配置 B	配置 C
适用于生成符合 NIST 标准的密钥	是	否	
熵(1)	通过认证	好	非常好
速度(2)	基线	快速	基线

1. 对于配置 B 和 C，使用 AIS-31 测试套件（T0 到 T8）对熵进行验证。
2. 当速度不能满足应用需求时，可以使用符合 NIST 标准的 DRBG 来提高吞吐量。

32.6.3 数据采集

为了运行统计测试，需要对熵源的原始数据和输出进行采样。有关数据收集和统计测试套件运行的详细信息，请参见“使用 NIST 统计测试套件验证 STM32 微控制器随机数生成”应用笔记 (AN4230)（可访问 *www.st.com* 获取）。

32.7 RNG 寄存器

RNG 与控制寄存器、数据寄存器和状态寄存器相关联。

32.7.1 RNG 控制寄存器 (RNG_CR)

RNG control register

偏移地址：0x000

复位值：0x0080 0D00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CONFIGLOCK	CONDRST	Res.	Res.	Res.	Res.	RNG_CONFIG1[5:0]						CLKDIV[3:0]			
rs	rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RNG_CONFIG2[2:0]			NISTC	RNG_CONFIG3[3:0]				ARDIS	Res.	CED	Res.	IE	RNGEN	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw		rw		rw	rw		

位 31 **CONFIGLOCK**：RNG 配置锁定 (RNG Config lock)

0：允许写入 RNG_NSCR、RNG_HTCR 和 RNG_CR 配置位 [29:4]。

1：忽略对 RNG_NSCR、RNG_HTCR 和 RNG_CR 配置位 [29:4] 的写操作，直到下一次 RNG 复位。

该位一旦置 1，只能在 RNG 复位时清零（单次设置）。

位 30 **CONDRST**：调节软复位 (Conditioning soft reset)

写入 1 然后写入 0，可复位调节逻辑、清空所有 FIFO 并开始重新初始化 RNG（RNG_SR 寄存器清零）。CONDRST 不会更改寄存器 RNG_CR 和 RNG_HTCR。

在设置任何配置位 [29:4] 的同一访问中，必须将该位置 1。换句话说，当 CONDRST 位置 1 时，也必须在位 [29:4] 中写入正确配置。

当 CONDRST 由软件设置为 0 时，其值在复位过程完成后将变为 0。大约需要 2 个 AHB 时钟周期 + 2 个 RNG 时钟周期。

位 29:26 保留，必须保持复位值。

位 25:20 **RNG_CONFIG1[5:0]**：RNG 配置 1 (RNG configuration 1)

保留用于 RNG 配置（位字段 1）。必须使用*第 32.6 节：RNG 熵源验证*中记录的建议值进行初始化。

仅当在同一访问中 CONDRST 位置 1，而 CONFIGLOCK 保持为 0 时，才允许写入 RNG_CONFIG1 的任意位。如果 CONFIGLOCK = 1，则忽略对该位的写操作。

位 19:16 **CLKDIV[3:0]**：时钟分频系数 (Clock divider factor)

该值用于配置输入 RNG 时钟的内部可编程分频器（从 1 到 16）。这些位只能在内核已禁止 (RNGEN = 0) 时写入。

0x0：分频后的内部 RNG 时钟等同于输入 RNG 时钟。

0x1：每个内部 RNG 时钟周期为两个 RNG 时钟周期。

0x2：每个内部 RNG 时钟周期为 2^2 (= 4) 个 RNG 时钟周期。

...

0xF：每个内部时钟周期为 2^{15} 个 RNG 时钟周期（例如，48 MHz 输入 RNG 时钟变为 1.5 kHz 内部 RNG 时钟）

仅当在同一访问中 CONDRST 位置 1，而 CONFIGLOCK 保持为 0 时，才允许写入这些位。如果 CONFIGLOCK = 1，则忽略对该位的写操作。

位 15:13 **RNG_CONFIG2[2:0]**：RNG 配置 2 (RNG configuration 2)
保留用于 RNG 配置（位字段 2）。对于 RNG 功耗极为敏感的情况，可以将位 13 置 1。请参见*第 32.3.8 节：RNG 低功耗使用*。有关详细信息，请参见 RNG_CONFIG1 位字段。

位 12 **NISTC**：NIST 自定义 (NIST custom)
0：符合 NIST 标准的 RNG 的硬件默认值。在此配置中，每 128 位输出执行两轮调节并使用 256 位噪声源。
1：符合 NIST 标准的 RNG 的自定义值。关于建议使用的配置，请参见*第 32.6 节：RNG 熵源验证*。
仅当在同一访问中 CONDRST 位置 1，而 CONFIGLOCK 保持为 0 时，才允许写入该位。如果 CONFIGLOCK = 1，则忽略对该位的写操作。

位 11:8 **RNG_CONFIG3[3:0]**：RNG 配置 3 (RNG configuration 3)
保留用于 RNG 配置（位字段 3）。有关详细信息，请参见 RNG_CONFIG1 位字段。
如果该寄存器中的 NISTC 位清零，RNG 会忽略 RNG_CONFIG3 位字段值。

位 7 **ARDIS**：自动复位禁止 (Auto reset disable)
将该位置 1 可停用自动复位功能。
0：使能自动复位
1：禁止自动复位
使自动复位保持使能（自动清零 SECS 位）可简化对噪声源错误的管理，如*第 32.3.7 节：错误管理*所述。
仅当在同一访问中 CONDRST 位置 1，而 CONFIGLOCK 保持为 0 时，才允许写入该位。如果 CONFIGLOCK = 1，则忽略对该位的写操作。

位 6 保留，必须保持复位值。

位 5 **CED**：时钟错误检测 (Clock error detection)
0：使能时钟错误检测
1：禁止时钟错误检测
当使能 RNG 时，不能实时使能或禁止时钟错误检测，即必须禁止 RNG 才能使能或禁止 CED。
仅当在同一访问中 CONDRST 位置 1，而 CONFIGLOCK 保持为 0 时，才允许写入该位。如果 CONFIGLOCK = 1，则忽略对该位的写操作。

位 4 保留，必须保持复位值。

位 3 **IE**：中断使能
0：禁止 RNG 中断
1：使能 RNG 中断。只要 RNG_SR 寄存器中的 DRDY、SEIS 或 CEIS 置 1，就会挂起中断。

位 2 **RNGEN**：真随机数发生器使能 (True random number generator enable)
0：禁止真随机数发生器。模拟噪声源关闭，并会对由 RNG 时钟提供时钟信号的逻辑系统进行门控。
1：使能真随机数发生器。

位 1:0 保留，必须保持复位值。

32.7.2 RNG 状态寄存器 (RNG_SR)

RNG status register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEIS	CEIS	Res.	Res.	SECS	CECS	DRDY
									rc_w0	rc_w0			r	r	r

位 31:7 保留，必须保持复位值。

位 6 **SEIS**：种子错误中断状态 (Seed error interrupt status)

该位与 SECS 同时置 1，通过写入 0 来清零（除非使用 CONDRST）。写入 1 无效。

0：未检测到错误序列

1：至少检测到一个错误序列。有关详细信息，请参见 SECS 位说明。

如果 RNG_CR 寄存器中 IE = 1，则会挂起中断。

位 5 **CEIS**：时钟错误中断状态 (Clock error interrupt status)

该位与 CECS 同时置 1，通过写入 0 来清零。写入 1 无效。

0：RNG 时钟正确 ($f_{RNGCLK} > f_{HCLK}/32$)

1：检测到 RNG 时钟在内部分频前过慢 ($f_{RNGCLK} < f_{HCLK}/32$)

如果 RNG_CR 寄存器中 IE = 1，则会挂起中断。

位 4:3 保留，必须保持复位值。

位 2 **SECS**：种子错误当前状态 (Seed error current status)

0：目前未检测到错误序列。如果 SEIS 位置 1，则意味着已检测到错误序列并已恢复正常。

1：至少检测到一个以下错误序列：

– 运行时重复计数测试失败（噪声源持续不变地输出了 24 位或以上的 0 或 1，或者 32 个或以上的 01 或 10）。

– 噪声源启动测试或连续自适应比例测试失败。

– 启动后期处理/调节完整性检查失败。

位 1 **CECS**：时钟错误当前状态 (Clock error current status)

0：RNG 时钟正确 ($f_{RNGCLK} > f_{HCLK}/32$)。如果 CEIS 位置 1，则意味着已检测到时钟过慢并已恢复正常。

1：RNG 时钟过慢 ($f_{RNGCLK} < f_{HCLK}/32$)。

注意：只有 RNG_CR 寄存器中的 CED 位置 0 时，CECS 位才有效。

位 0 **DRDY**：数据就绪 (Data ready)

0：RNG_DR 寄存器尚未有效，无可用随机数据。

1：RNG_DR 寄存器包含有效随机数据。

在输出缓冲区变空之后（读取 RNG_DR 寄存器之后），在生成新的随机值之前，该位返回 0。

注意：当禁止外设时（RNG_CR 寄存器中的 RNGEN = 0），DRDY 位可能会被置 1。

如果 RNG_CR 寄存器中的 IE = 1，则在 DRDY = 1 时生成中断。

32.7.3 RNG 数据寄存器 (RNG_DR)

RNG data register

偏移地址：0x008

复位值：0x0000 0000

RNG_DR 寄存器是只读寄存器，在读取时提供 32 位随机数值。当 DRDY = 1 并且该值不为 0x0 时，即使 RNGEN = 0，该寄存器的内容也有效。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RNDATA[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RNDATA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RNDATA[31:0]**：随机数据 (Random data)

DRDY = 1 时有效的 32 位随机数据。DRDY = 0 时，RNDATA 值为 0。
当 DRDY 置 1 时，建议始终验证 RNG_DR 是否不为零。值为零表示在轮询 RNG_SR 与读取 RND_DR 输出之间可能发生了种子错误（罕见事件）。

32.7.4 RNG 噪声源控制寄存器 (RNG_NSCR)

RNG noise source control register

偏移地址：0x00C

复位值：0x0003 FFFF

仅当 RNG_CR 中的 CONDRST 位置 1 且 CONFIGLOCK 位清零时，才允许写入 RNG_NSCR。如果 CONFIGLOCK = 1，则忽略对该寄存器的写操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN_OSC6[2:1]	
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN_OSC6[0]	EN_OSC5[2:0]			EN_OSC4[2:0]			EN_OSC3[2:0]			EN_OSC2[2:0]			EN_OSC1[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:18 保留，必须保持复位值。

位 17:15 **EN_OSC6[2:0]**：

当 RNG 被使能时 (RNGEN=1)，每个位使能振荡器 6 三个输入中的一路。其他情况下，该位字段无效。

位 14:12 **EN_OSC5[2:0]**：

当 RNG 被使能时 (RNGEN=1)，每个位使能振荡器 5 三个输入中的一路。其他情况下，该位字段无效。

位 11:9 **EN_OSC4[2:0]**：

当 RNG 被使能时 (RNGEN=1)，每个位使能振荡器 4 三个输入中的一路。其他情况下，该位字段无效。

位 8:6 **EN_OSC3[2:0]**：
当 RNG 被使能时 (RNGEN=1)，每个位使能振荡器 3 三个输入中的一路。其他情况下，该位字段无效。

位 5:3 **EN_OSC2[2:0]**：
当 RNG 被使能时 (RNGEN=1)，每个位使能振荡器 2 三个输入中的一路。其他情况下，该位字段无效。

位 2:0 **EN_OSC1[2:0]**：
当 RNG 被使能时 (RNGEN=1)，每个位使能振荡器 1 三个输入中的一路。其他情况下，该位字段无效。

32.7.5 RNG 运行状态测试控制寄存器 (RNG_HTCR)

RNG health test control register

偏移地址：0x010

复位值：0x0000 72AC

仅当 RNG_CR 中的 CONDRST 位置 1 且 CONFIGLOCK 位清零时，才允许写入 RNG_HTCR。如果 CONFIGLOCK = 1，则忽略对该寄存器的写操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HTCFG[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HTCFG[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **HTCFG[31:0]**：运行状态测试配置 (health test configuration)
RNG 使用该配置来配置运行状态测试。有关建议值，请参见*第 32.6 节：RNG 熵源验证*。
注意：如果写入的值与建议值不同，则无法保证 RNG 行为（包括对该寄存器的读操作）。

32.7.6 RNG 硬件配置寄存器 (RNG_HWCFGR)

RNG hardware configuration register

偏移地址：0x3F0

复位值：0x0000 0206

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG2[7:0]								CFG1[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:8 **CFG2[7:0]**：硬件通用 2 (Hardware generic 2)
该位字段返回 RNG 总线数量 (rngbus_nb)。

位 7:0 **CFG1[7:0]**：硬件通用 1 (Hardware generic 1)
该位字段返回 triple_oscillator_nb 通用值 (0x6)。

32.7.7 RNG 版本寄存器 (RNG_VERR)

RNG version register

偏移地址：0x3F4

复位值：0x0000 0041

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：RNG 主版本 (DMA major revision)

位 3:0 **MINREV[3:0]**：RNG 次版本 (DMA minor revision)

32.7.8 RNG 标识寄存器 (RNG_IPIDR)

RNG identification register

偏移地址：0x3F8

复位值：0x0017 0041

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：RNG 标识码 (RNG identification code)

32.7.9 RNG 大小标识寄存器 (RNG_SIDR)

RNG size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：RNG 大小标识码 (RNG size identification code)

位[31:8] = 0xA3C5DD（固定代码）

位[7:0] = 0x01（1 KB 地址解码）

32.7.10 RNG 寄存器映射

表 271. RNG 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	RNG_CR	CONFIGLOCK	CONDRST	Res.	Res.	Res.	Res.	RNG_CONFIG1[5:0]						.CLKDIV [3:0]				RNG_ CONFIG2 [2:0]			NISTC	RNG_ CONFIG3 [3:0]				ARDIS	Res.	CED	Res.	IE	RNGEN	Res.	Res.
	Reset value	0	0					0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0		0		0	0		
0x004	RNG_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEIS	CEIS	Res.	Res.	SECS	CECS	DRDY
	Reset value																										0	0			0	0	0
0x008	RNG_DR	RNDATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	RNG_NSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN_OSC6[2:0]			EN_OSC5[2:0]			EN_OSC4[2:0]			EN_OSC3[2:0]			EN_OSC2[2:0]			EN_OSC1[2:0]		
	Reset value															1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x010	RNG_HTCR	HTCFG[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	1	1	0	0
0x3EC 0x20C	Reserved	Reserved																															
0x3F0	RNG_ HWCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFG2[7:0].								CFG1[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3F4	RNG_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV [3:0]				MINREV [3:0]			
	Reset value																									0	1	0	0	0	0	0	1
0x3F8	RNG_IPIDR	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
0x3FC	RNG_SIDR	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 2.5 节*。

33 安全 AES 协处理器 (SAES)

33.1 简介

安全 AES 协处理器 (SAES) 使用完全符合联邦信息处理标准 (FIPS) 第 197 号出版物中规定的高级加密标准 (AES) 的算法和实现来对数据进行加密或解密。它集成了针对侧信道攻击 (SCA) 的保护，包括差分功耗分析 (DPA)，通过了 SESIP 和 PSA 安全保证的级别 3 的认证。

该外设支持 128 或 256 位密钥大小的 CTR、GCM、GMAC、CCM、ECB 和 CBC 链接模式，也支持一些特殊模式，如硬件密钥加/解密（封装密钥模式），与快速 CRYP 共享密钥（共享密钥模式）等。

对于 STM32 硬件密钥（启动硬件密钥 BHK 和派生的硬件唯一密钥 DHUK），SAES 有能力对它们进行硬件加载，而应用程序可以使用这些密钥但无法读取它们。

SAES 是 AMBA AHB 从外设，仅支持通过 32 位单次访问的方式进行访问。其他访问类型会生成 AHB 错误，非 32 位的写入可能会破坏寄存器内容。

该外设支持对传入和传出数据进行 DMA 单次传输（需要两个 DMA 通道）。它与真随机数发生器 (TRNG) 和 CRYP 外设进行硬件链接。

33.2 SAES 主要特性

- 符合自 2001 年 11 月起的 NIST *“高级加密标准 (AES)，FIPS 第 197 号出版物”*
- 128 位数据块处理
- 支持 128 位和 256 位密钥长度
- 基于多种链接模式的加密和解密：
  - 电子密码本 (ECB) 模式
  - 密码块链接 (CBC) 模式
  - 计数器 (CTR) 模式
  - Galois 计数器模式 (GCM)
  - Galois 消息认证码 (GMAC) 模式
  - CBC-MAC 计数器 (CCM) 模式
- 在 ECB 模式下，使用 128 位密钥或 256 位密钥处理一个 128 位数据块时，分别会有 480 或 680 个时钟周期的延迟
- 集成轮密钥调度程序，用于计算 ECB/CBC 解密的最后一轮密钥
- AMBA AHB 从外设，仅支持通过 32 位字单次访问的方式进行访问。
- 256 位只写寄存器，用于存储密钥（8 个 32 位寄存器）
  - 也可以由硬件加载两个可通过异或运算连接在一起的硬件密钥（BHK、DHUK）
- 128 位寄存器，用于存储初始化向量（4 个 32 位寄存器）
- 32 位缓冲器，用于数据输入和输出
- 采用自动数据流控制，支持单次传输直接内存访问 (DMA)（使用两个通道，分别用于传入数据和已处理数据）
- 采用数据交换逻辑，支持 1 位、8 位、16 位或 32 位数据
- 在 SAES 需要处理优先级更高的消息时，可通过软件将当前消息挂起，然后恢复原始消息
- 密钥的安全上下文增强

- 硬件密钥加密/解密（封装密钥模式）
- 针对侧信道攻击 (SCA) 的保护，包括差分功耗分析 (DPA)，通过了 SESIP 和 PSA 安全保证的级别 3 的认证。
- 在 SAES 控制下，与速度更快的 CRYP 外设共享硬件密钥（共享密钥模式）

33.3 SAES 实现

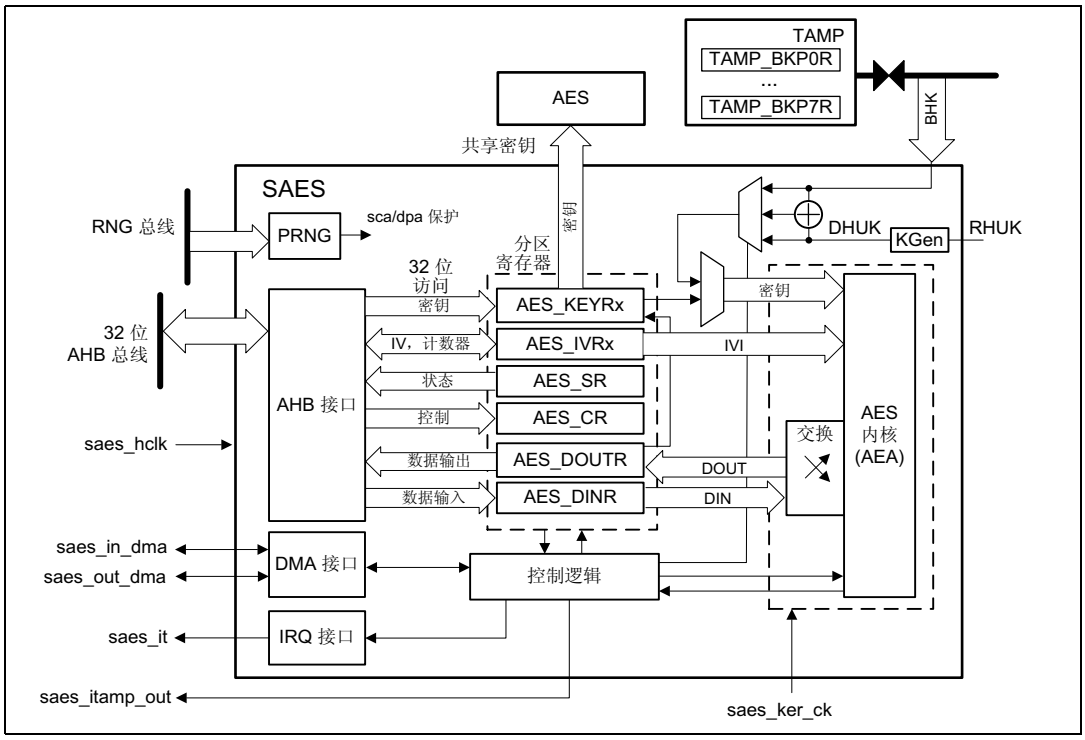
设备具有一个连接到 MDMA 的 SAES 外设。

33.4 SAES 功能描述

33.4.1 SAES 框图

图 275 显示了 SAES 的框图。

图 275. SAES 框图



注： *AES 表示 CRYP 外设。*

33.4.2 SAES 内部信号

表 272 描述了连接 SAES 外设的用户相关内部信号。

表 272. SAES 内部输入/输出信号

信号名称	信号类型	说明
saes_hclk	输入	AHB 总线时钟
saes_it	输出	SAES 中断请求
saes_in_dma	输入/输出	输入 DMA 单次请求/确认
saes_out_dma	输入/输出	输出 DMA 单次请求/确认
saes_itamp_out	输出	发送到 TAMP（异或运算）的 Tamper 事件信号，在发生意外硬件故障时触发。当该信号被触发时，SAES 自动将密钥寄存器清零。需要进行复位才能再次使用 SAES。
saes_ker_ck	输入	SAES 内核时钟。
RHUK	输入	256 位根硬件唯一密钥（非易失性、每个设备唯一且对软件保密），用于内部计算派生硬件唯一密钥 (DHUK)
BHK(1)	输入	256 位启动硬件密钥 (BHK)，存储在防篡改安全备份寄存器中，在启动期间通过安全代码写入。一旦写入，在下一次产品复位之前，任何应用都无法读取或写入该密钥。

1. 连接到 TAMP 外设中的一组备份寄存器，这些寄存器由应用软件写入，然后读/写锁定（详细信息请参见 *第 33.4.16 节*）。

33.4.3 SAES 密码内核

概述

SAES 密码内核由以下部分组成：

- AES 内核算法 (AEA)
- 基于二元 Galois 域的乘法器 (GF2mul)
- 密钥输入
- 初始化向量 (IV) 输入
- 链接算法逻辑（XOR、反馈/计数器、屏蔽）

SAES 内核适用于密钥长度为 128 位或 256 位的 128 位数据块（四字）。根据链接模式的不同，SAES 可能需要一个 128 位初始化向量 IV，也可能不需要。

SAES 具有以下工作模式：

- **模式 1：**
  使用存储在 SAES_KEYRx 寄存器中，或者使用专用硬件总线读取的密钥，进行明文加密。
- **模式 2：**
  ECB 或 CBC 解密密钥准备。在选择 ECB 或 CBC 链接模式的模式 3 之前，必须先使用此模式。准备解密的密钥自动存储在 SAES_KEYRx 寄存器中。于是，SAES 外设准备好切换到模式 3 以执行数据解密。

- **模式 3：**
  使用存储在 SAES_KEYRx 寄存器中的密钥实现密文解密。当选择 ECB 和 CBC 链接模式时，必须通过模式 2 预先准备好密钥。

注： *仅在执行 ECB 和 CBC 解密时使用模式 2。*

通过对 SAES_CR 寄存器中的 MODE[1:0] 位字段进行编程来选择工作模式。只能在 SAES 外设已禁止时进行。

使用 SAES_CR 寄存器中的 KMOD[1:0] 位字段来选择特殊密钥操作。详细信息，请参见*第 33.4.14 节*和*第 33.4.13 节*。

典型数据处理

SAES 的典型用法如*第 1696 页的第 33.4.4 节：执行密码操作的 SAES 过程*中所述。

注： *除了 IVI 位字段，AEA 算法中间阶段的输出通常不会在密码边界外公开。*

链接模式

SAES 支持以下链接模式，可通过 SAES_CR 寄存器的 CHMOD[2:0] 位字段选择：

- 电子密码本 (ECB)
- 密码块链接 (CBC)
- 计数器 (CTR)
- Galois 计数器模式 (GCM)
- Galois 消息认证码 (GMAC)
- CBC-MAC 计数器模式 (CCM)

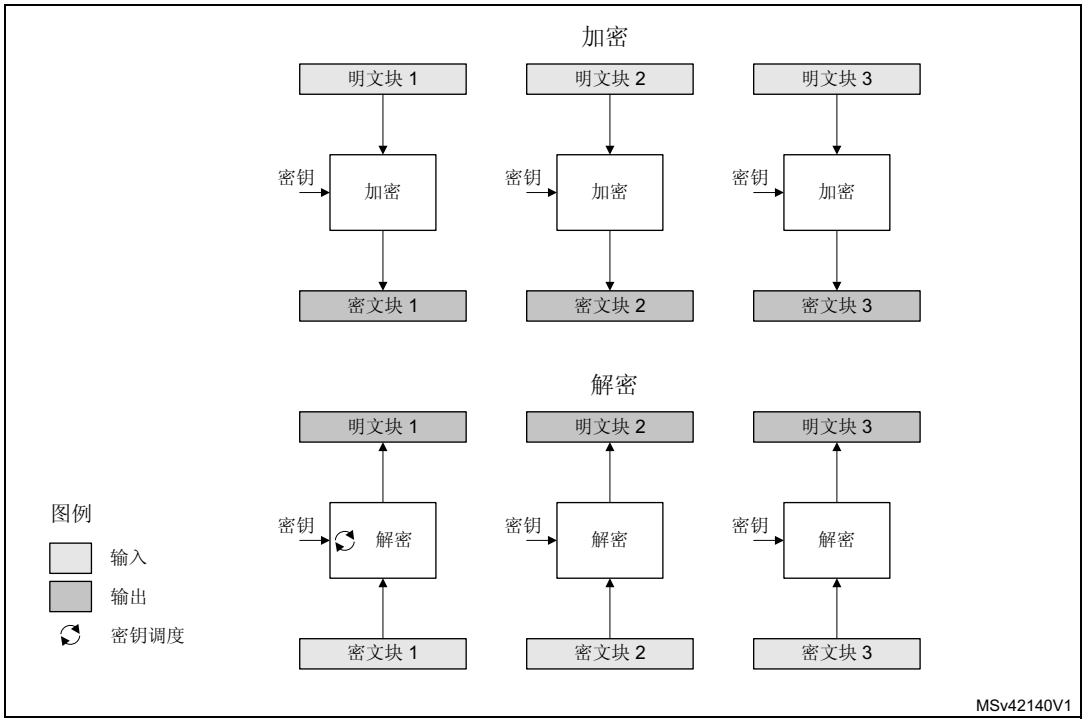
注： *只有在 SAES 禁止（SAES_CR 寄存器的位 EN 清零）时，才能更改链接模式。*

后续子章节对每个 SAES 链接模式的原理进行了介绍。

详细信息请参见从*第 33.4.8 节：SAES 基本链接模式（ECB 和 CBC）*开始的专用章节。

电子密码本 (ECB) 模式

图 276. ECB 加密和解密原理

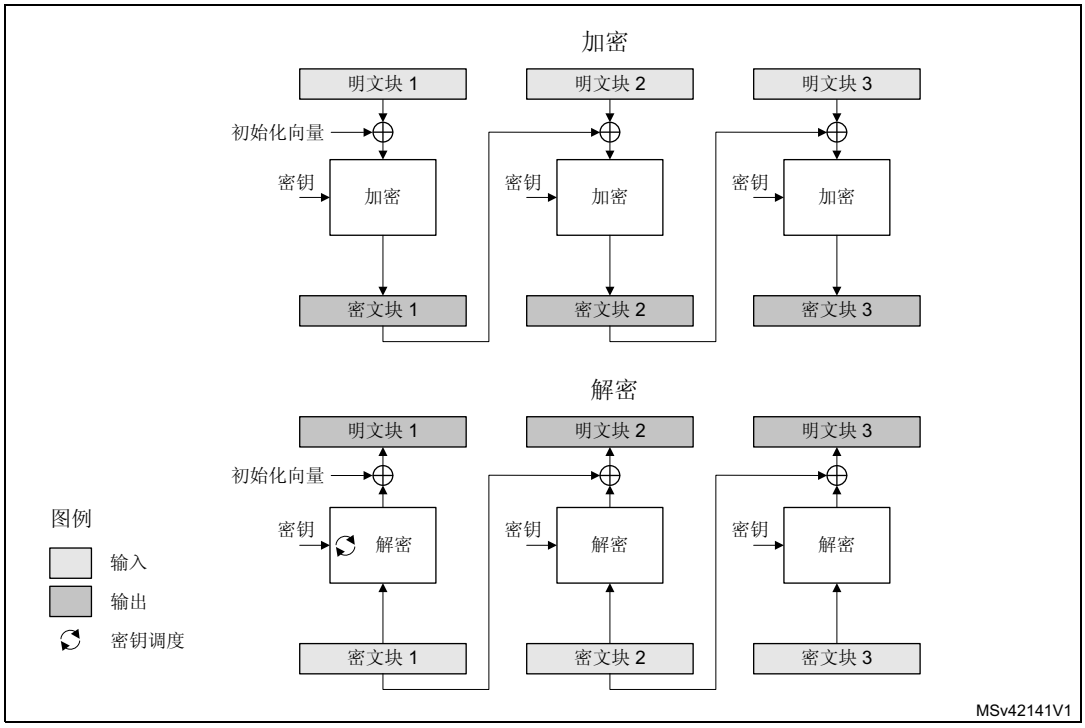


ECB 是最简单的工作模式。无链接操作，且无特殊初始化阶段。消息会划分到各个块中，并对各个块分别进行加密或解密。

注： *对于解密，在处理第一个块之前需要特殊的密钥调度。*

密码块链接 (CBC) 模式

图 277. CBC 加密和解密原理

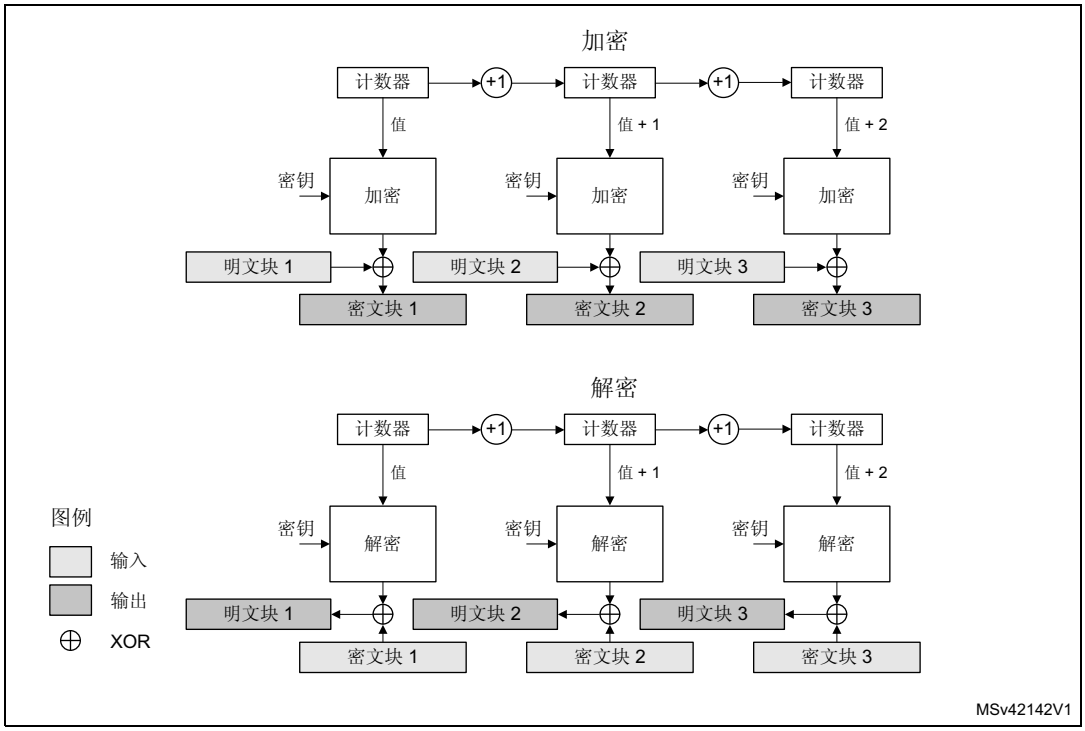


在 CBC 模式下，每个块的输出均与下一个块的输入相连。为了确保每条消息都是唯一的，会在第一个块处理过程中使用初始化向量。

注： *对于解密，在处理第一个块之前需要特殊的密钥调度。*

计数器 (CTR) 模式

图 278. CTR 加密和解密原理

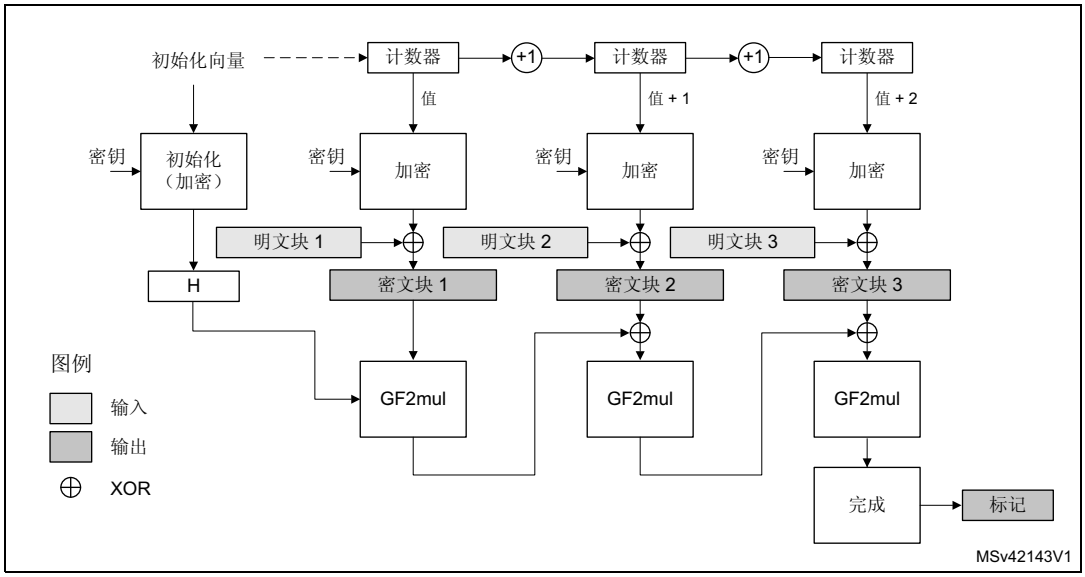


CTR 模式使用 SAES 内核生成密匙流。这些密钥将与明文进行逻辑异或运算，以获得密文（如 NIST *特别出版物 800-38A，块密码工作模式的建议*中所述）。

注： *与 ECB 和 CBC 模式不同，CTR 解密无需密钥调度，因为在该链接方案中，SAES 内核始终在加密模式下用于生成密钥流或计数器块。*

Galois/计数器模式 (GCM)

图 279. GCM 加密和认证原理

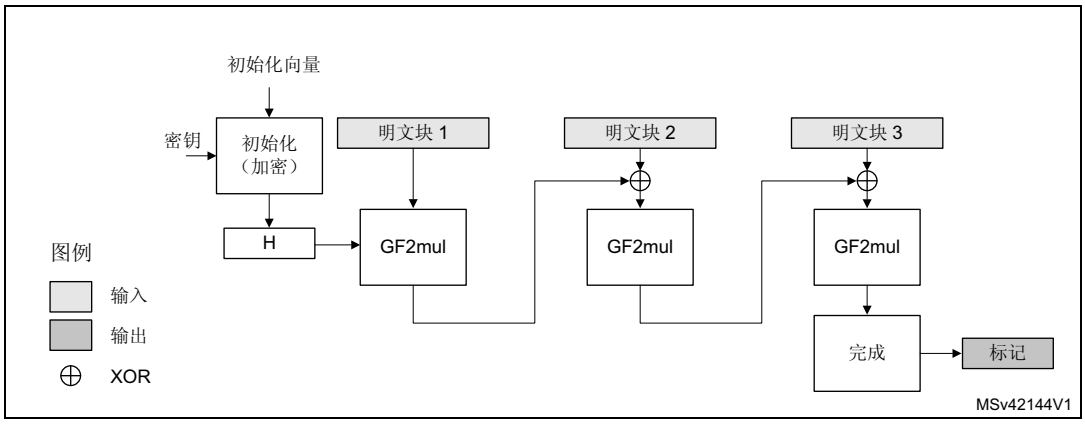


在 Galois/计数器模式 (GCM) 下，将对明文消息进行加密，同时会对消息认证码 (MAC) 进行计算，从而生成相应的密文及其 MAC（也称为认证标记）。NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/ 计数器模式 (GCM) 和 GMAC* 中对此进行了相关定义。

GCM 模式基于 SAES 计数器模式，可实现机密性。它使用固定有限域乘法器来计算消息认证码。消息末尾处需要一个初始值和一个特定的 128 位块。

Galois 消息认证码 (GMAC) 原理

图 280. GMAC 认证原理

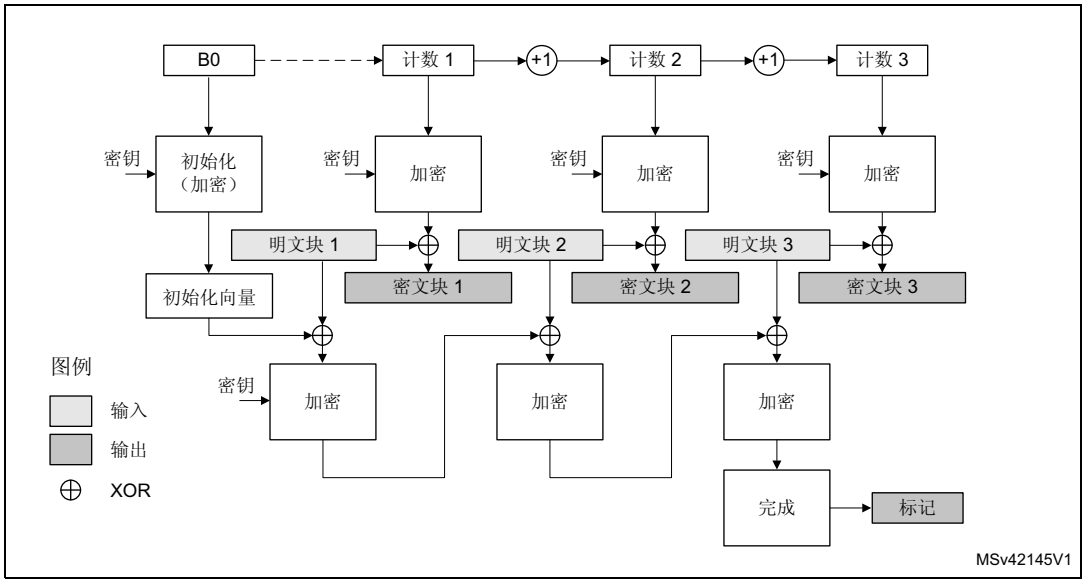


Galois 消息认证码 (GMAC) 允许认证消息并生成相应的消息认证码 (MAC)。NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/ 计数器模式 (GCM) 和 GMAC* 中对此进行了相关定义。

GMAC 与 GCM 类似，只不过它应用于仅由明文认证数据组成的消息（即只有标头，无有效负载）。

CBC-MAC 计数器 (CCM) 原理

图 281. CCM 加密和认证原理



在密码块链接-消息认证码计数器 (CCM) 模式下，将对明文消息进行加密，同时会对消息认证码 (MAC) 进行计算，从而生成相应的密文以及相应的 MAC（也称为标记）。NIST *特别出版物 800-38C，块密码工作模式的建议 - CCM 模式实现认证和保密性*中对此进行了介绍。

CCM 模式基于 SAES 计数器模式，可实现机密性，并且使用 CBC 来计算消息认证码。它需要一个初始值。

与 GCM 类似，CCM 链接模式可应用于仅由明文认证数据组成的消息（即只有标头，无有效负载）。请注意，这种 CCM 使用方式不被称为 CMAC（它并非类似于 GCM/GMAC），NIST 不推荐这种用途。

33.4.4 执行密码操作的 SAES 过程

简介

下面对典型密码操作进行了介绍。详细信息请参见从*第 33.4.8 节：SAES 基本链接模式（ECB 和 CBC）*开始的章节。

初始化 SAES

要初始化 SAES，首先通过将 SAES_CR 寄存器中的 EN 位清零来禁止它。然后以任意顺序执行以下步骤（KEYSIZE 除外）：

- 确认 SAES_SR 中的 BUSY = 0（未在进行 RNG 随机数获取），并且 SAES_ISR 中的 RNGEIF = 0（未标记随机数获取错误）。
- 通过对 SAES_CR 寄存器中的 MODE[1:0] 位字段进行编程来配置 SAES 模式。
  - 对于加密，选择模式 1 (MODE[1:0] = 00)。
  - 对于解密，选择模式 3 (MODE[1:0] = 10)，除非使用 ECB 或 CBC 链接模式。在后一种情况下，执行加密密钥的初始密钥派生，如*第 33.4.5 节：SAES 解密轮密钥准备*中所述。

- 通过对 SAES_CR 寄存器中的 CHMOD[2:0] 位字段进行编程来选择链接模式。通过将 KMOD[1:0] 设置为 00 选择正常密钥模式。有关其他 KMOD[1:0] 值，请参见*第 33.4.13 节*和*第 33.4.14 节*。
- 使用 SAES_CR 寄存器中的 DATATYPE[1:0] 位字段配置数据类型（1 位、8 位、16 位或 32 位）
- 必要时（例如，在 CBC 或 CTR 链接模式下），向 SAES_IVRx 寄存器中写入初始化向量。
- 使用 SAES_CR 寄存器中的 KEYSIZE 位字段配置密钥大小（128 位或 256 位）该步骤必须在写入密钥寄存器或使用 KEYSEL 选择不同的密钥源之前完成。如果密钥不能与不同的安全上下文（不同的安全属性）共享，还必须将 SAES_CR 寄存器的 KEYPROT 位置 1。
- 将对称密钥写入 SAES_KEYRx 寄存器（4 或 8 个寄存器，具体取决于密钥大小）。或者，通过将 KEYSEL[2:0] 设置为不同于 0x0 的值来选择与密钥寄存器不同的密钥源。有关详细信息，请参见*第 33.4.16 节*。

注： *当 KEYSIZE 定义的密钥信息载入 SAES_KEYRx 时，SAES 会将 SAES_SR 中的 KEYVALID 置 1。*

附加数据

本节介绍，当 KMOD[1:0] = 00 时，要处理数据的大小不是 128 位的倍数，处理附加数据的不同方式。有关其他 KMOD[1:0] 值，请参见*第 33.4.14 节*和*第 33.4.13 节*。

有关 ECB 或 CBC 模式，请参见*第 33.4.6 节：SAES 密文窃取和数据填充*。这些情况下的最后一个块管理比本节描述的流程更复杂。

通过轮询附加数据

此方法按照以下步骤使用标志轮询来控制附加数据：

1. 通过将 SAES_CR 寄存器中的 EN 位置 1 来使能 SAES 外设。
2. 重复以下流程，直到全部有效数据处理完成：
   a) 将四个输入数据字写入 SAES_DINR 寄存器。
   b) SAES_SR 中的状态标志 CCF 置 1 后，从 SAES_DOUTR 寄存器读取四个数据字。
   c) 通过将 SAES_ICR 寄存器中的 CCF 位置 1，将 CCF 标志清零。
   d) 如果刚刚处理的数据块是消息的倒数第二个块，并且要处理的最后一个块中的有效数据不足 128 位，则用零填充最后一个块的其余部分，并且在 GCM 有效负载加密或 CCM 有效负载解密的情况下，使用 SAES_CR 寄存器的 NPBLB 位字段指定无效字节的数量，以便 SAES 计算正确的标记；请参见*第 33.4.6 节：SAES 密文窃取和数据填充*。
3. 由于是最后一个块，按照*第 33.4.6 节：SAES 密文窃取和数据填充*中的说明操作，丢弃无效数据，然后通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设。

注： *在对 SAES_DINR 寄存器的两次连续写入之间自动插入最多三个等待周期，以便将密钥发送到 AES 协处理器。*

NPBLB 位不用于 GCM、GMAC 和 CCM 链接模式的标头阶段。

使用中断附加数据

此方法通过以下步骤，使用 SAES 外设中断来控制附加数据：

1. 通过将 SAES_IER 寄存器的 CCFIE 位置 1 使能来自 SAES 的中断。
2. 通过将 SAES_CR 寄存器中的 EN 位置 1 来使能 SAES 外设。
3. 将前四个输入数据字写入 SAES_DINR 寄存器。
4. 收到中断时在 SAES 中断服务程序中处理数据：
   a) 从 SAES_DOUTR 寄存器读取四个输出数据字。
   b) 通过将 SAES_ICR 寄存器中的 CCF 位置 1，将 CCF 标志清零，从而将挂起中断清零。
   c) 如果刚刚处理的数据块是消息的倒数第二个块，并且要处理的最后一个块中的有效数据不足 128 位，则用零填充最后一个块的其余部分，并且在 GCM 有效负载加密或 CCM 有效负载解密的情况下，使用 SAES_CR 寄存器的 NPBLB 位字段指定无效字节的数量，以便 SAES 计算正确的标记；请参见*第 33.4.6 节：SAES 密文窃取和数据填充*。然后继续进行 4e)。
   d) 如果刚刚处理的数据块是消息的最后一块，则丢弃无效数据（根据需要按照*第 33.4.6 节：SAES 密文窃取和数据填充*中的说明操作），然后通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设，并退出中断服务程序。
   e) 将接下来的四个输入数据字写入 SAES_DINR 寄存器，并退出中断服务程序。

注： *SAES 允许连续读取或写入操作之间存在延迟，例如在两次 SAES 计算之间要响应的另一个外设的中断。*

NPBLB 位不用于 GCM、GMAC 和 CCM 链接模式的标头阶段。

使用 DMA 附加数据

借助此方法，所有传输和处理过程均由 DMA 和 SAES 管理。要使用此方法，请按照以下步骤操作：

1. 如果要处理的最后一个块小于 128 位，请参见*第 33.4.6 节：SAES 密文窃取和数据填充*准备最后一个四字数据块。
2. 将 DMA 控制器配置为将来自内存的要处理的数据传输到 SAES 外设输入以及将来自 SAES 外设输出的已处理数据传输到内存，如*第 33.4.18 节：SAES DMA 接口*所述。配置 DMA 控制器，以在传输完成时产生中断。在 GCM 有效负载加密或 CCM 有效负载解密的情况下，如果用零填充了最后一个四字块，则 DMA 传输**不得**包括最后一个四字块。对于这个最后的四字块，必须改用*通过轮询附加数据*中描述的步骤，因为处理该块之前，必须设置 NPBLB 位，以便 SAES 计算正确的标记。
3. 通过将 SAES_CR 寄存器中的 EN 位置 1 来使能 SAES 外设
4. 将 SAES_CR 寄存器中的 DMAINEN 和 DMAOUTEN 位置 1，以使能 DMA 请求。
5. 当 DMA 中断指示传输完成时，从内存中获取 SAES 处理过的数据。

注： *CCF 标志在这种方法中没有作用，因为读取 SAES_DOUTR 寄存器是通过 DMA 自动管理的，在计算过程结束时没有任何软件操作。*

NPBLB 位不用于 GCM、GMAC 和 CCM 链接模式的标头阶段。

33.4.5 SAES 解密轮密钥准备

内部密钥调度用于生成 AES 轮密钥。在 AES 加密中，第 0 轮密钥是存储在密钥寄存器中的密钥。AES 解密必须使用最后一个轮密钥开始。由于加密密钥存储在内存中，因此必须执行特殊的密钥调度，以获得解密密钥。仅 ECB 和 CBC 模式下的 AES 解密需要此密钥调度。

推荐方法是通过将 SAES_CR 的 MODE[1:0] 位字段设置为 01 来选择模式 2（仅密钥处理），然后通过将 MODE[1:0] 设置为 10 来继续解密（模式 3，仅解密）。模式 2 的使用说明如下：

1. 确认 SAES_SR 中的 BUSY = 0（未在进行 RNG 随机数获取）。
2. 通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设。
3. 通过将 SAES_CR 寄存器中的 MODE[1:0] 位字段设置为 01 来选择模式 2。CHMOD[2:0] 位字段在这种情况下没有意义，因为该密钥派生模式与所选链接算法无关。通过将 KMOD[1:0] 设置为 00 选择正常密钥模式。有关使用其他 KMOD[1:0] 值进行解密的信息，请参见*第 33.4.14 节*和*第 33.4.13 节*。
4. 通过 SAES_CR 寄存器的 KEYSIZE 位，将密钥长度设置为 128 或 256 位。如果密钥不能与不同的安全上下文（不同的安全属性）共享，还必须将 SAES_CR 寄存器的 KEYPROT 位置 1。
5. 将加密密钥写入 SAES_KEYRx 寄存器（128 或 256 位），或者，通过 KEYSEL[2:0] 选择与密钥寄存器不同的密钥源。有关详细信息，请参见*第 33.4.16 节：SAES 密钥寄存器*。对 SAES_IVRx 寄存器执行写入操作不起作用。
6. 通过将 SAES_CR 寄存器中的 EN 位置 1 来使能 SAES 外设。
7. 等待 SAES_SR 寄存器中的 CCF 标志置 1。
8. 将 CCF 标志清零。派生密钥在 AES 内核中，可以用于解密。

注： *派生密钥可用时，将由硬件禁止 SAES。*

要重新开始计算派生密钥，请重复步骤 5、6、7 和 8。

注： *密钥准备操作持续 145 或 230 个时钟周期，具体取决于密钥大小（128 位或 256 位）。*

33.4.6 SAES 密文窃取和数据填充

在 ECB 或 CBC 模式下使用 SAES 来管理不是块大小（128 位）倍数的消息时，请使用密文窃取技术，如 NIST *第 800-38A 号特别出版物，块密码工作模式的建议：CBC 模式密文窃取的三种变型*中介绍的窃取技术。由于 SAES 外设不支持此类技术，应用程序必须使用倒数第二个块中的数据补全输入数据的最后一个块。

注： *本参考手册中未对密文窃取技术进行介绍。*

类似地，在除 ECB 或 CBC 之外的模式下使用 SAES 时，必须先用零填充不完整的输入数据块（即，输入数据短于 128 位的块），然后再加密（即，必须在数据串的尾端附加额外的位）。解密后必须丢弃额外填充的位。由于 SAES 不会对**最后一个块**执行自动数据填充操作，因此应用程序必须遵循*第 1696 页的第 33.4.4 节：执行密码操作的 SAES 过程*中给出的建议来管理大小不是 128 位倍数的消息。

注： *填充数据根据 SAES_CR 寄存器中的 DATATYPE[1:0] 位字段，以类似于正常数据的方式进行交换（有关详细信息，请参见第 33.4.15 节：SAES 数据寄存器和数据交换）。*

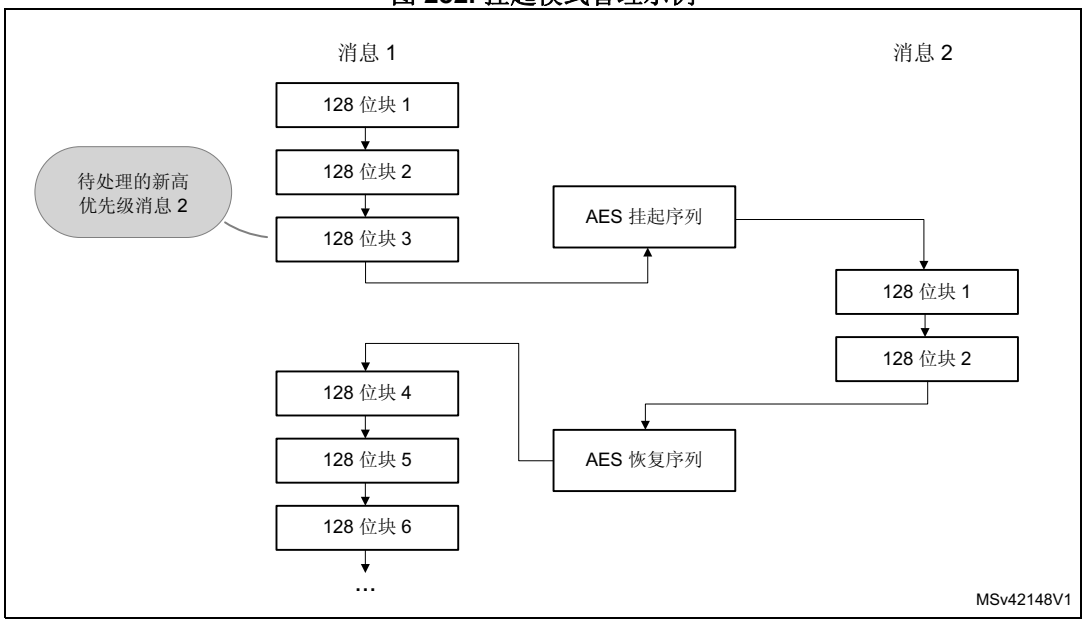
33.4.7 SAES 任务挂起和恢复

如果必须处理另一条优先级较高的消息，则可将消息挂起。该最高优先级消息完成发送后，可在加密或解密模式下恢复挂起的消息。

挂起/恢复操作不会破坏链接运算，并且再次使能 SAES 以接收下一个数据块时，可立即恢复消息处理。

图 282 举例说明了挂起/恢复操作：为发送长度更短、优先级更高的消息 2 而将消息 1 挂起。

图 282. 挂起模式管理示例



有关挂起/恢复操作的详细说明，请参见每个 SAES 模式的相应章节。

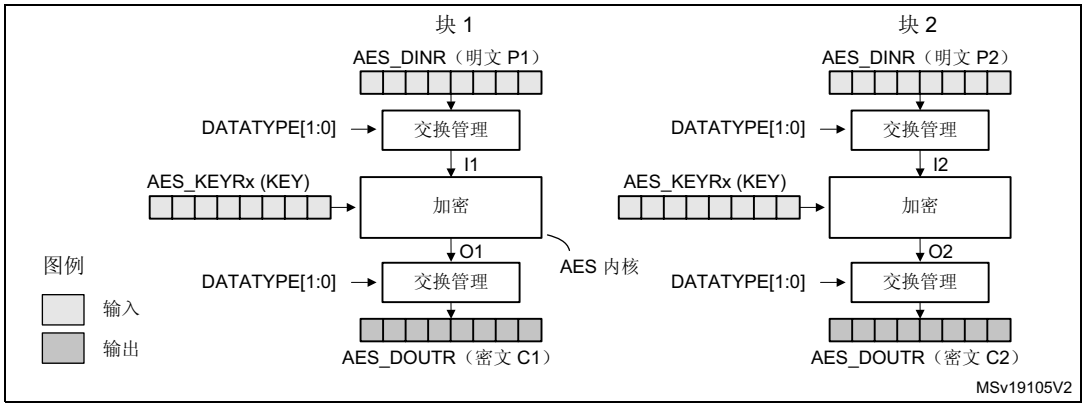
33.4.8 SAES 基本链接模式（ECB 和 CBC）

概述

本节简要介绍了 SAES 内核提供的四种基本操作模式：ECB 加密、ECB 解密、CBC 加密和 CBC 解密。有关详细信息，请参见 2001 年 11 月 26 日的 FIPS 第 197 号出版物。

图 283 介绍了电子密码本 (ECB) 加密。

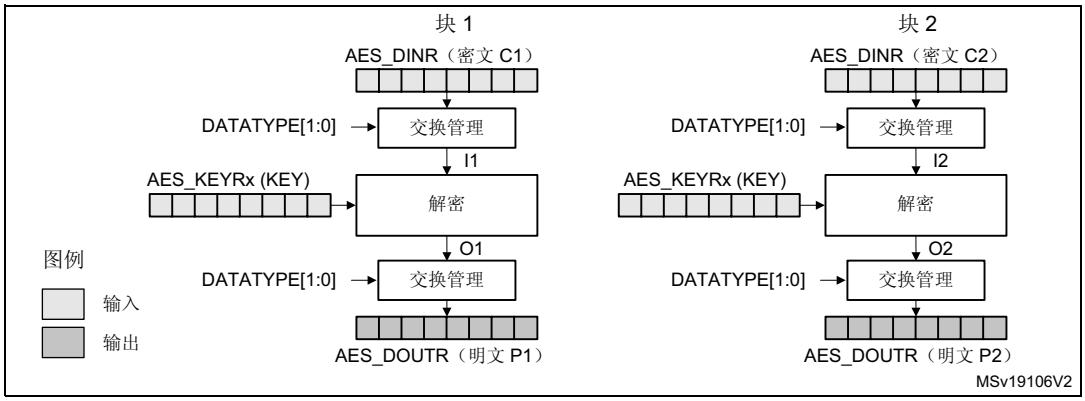
图 283. ECB 加密



在 ECB 加密模式下，SAES_DINR 寄存器中的 128 位明文输入数据块 Px 首先进行位/字节/半字交换。使用 128 位或 256 位密钥，通过在 AES 加密模式下设置的 AES 内核对交换结果 Ix 进行处理。加密结果 Ox 经过位/字节/半字交换，然后作为 128 位密文输出数据块 Cx 存储在 SAES_DOUTR 寄存器中。ECB 加密以这种方式继续，直到最后一个完整的明文块被加密。

图 284 介绍了电子密码本 (ECB) 解密。

图 284. ECB 解密

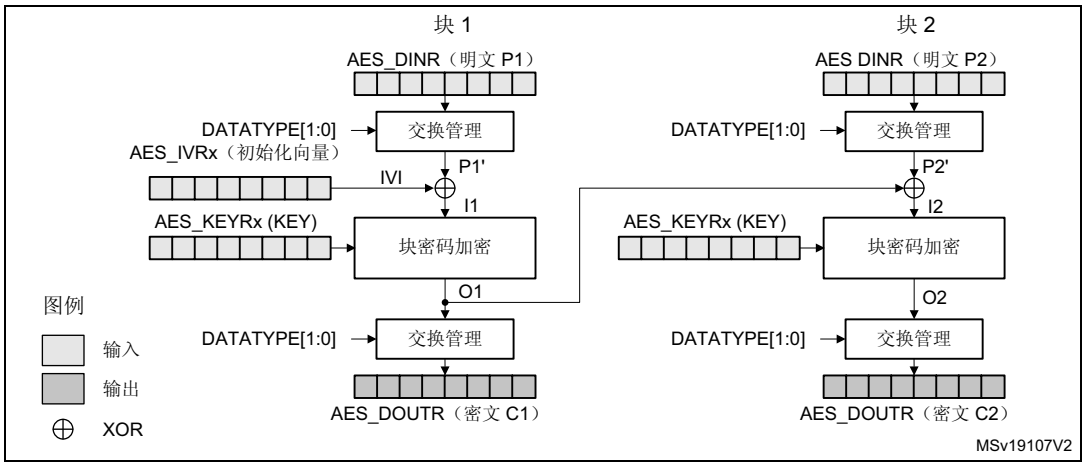


要在 ECB 模式下执行 AES 解密，需要收集最后一轮加密密钥（针对加密首先执行完整的密钥调度），并将其用作解密密文的第一轮密钥。该准备由 AES 内核提供支持。

在 ECB 解密模式下，SAES_DINR 寄存器中的 128 位密文输入数据块 C1 首先进行位/字节/半字交换。该密钥序列与 ECB 加密中的密钥序列相反。使用先前准备的解密密钥，通过在解密模式下设置的 AES 内核对交换结果 I1 进行处理。解密结果经过位/字节/半字交换，然后作为 128 位明文输出数据块 P1 存储在 SAES_DOUTR 寄存器中。ECB 解密以这种方式继续，直到最后一个完整的密文块被解密。

图 285 说明了密码块链接 (CBC) 解密。

图 285. CBC 加密

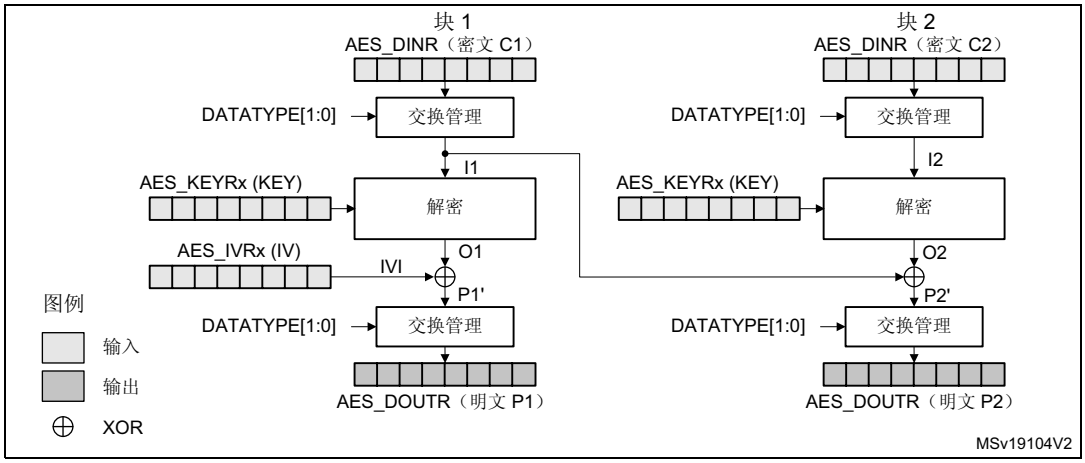


在 CBC 加密模式下，执行位/字节/半字交换 (P1') 后的第一个明文输入块与一个 128 位 IVI 位字段（初始化向量和计数器）进行逻辑异或运算，生成使用 128 位或 256 位密钥通过 AES 内核进行加密的 I1 输入数据。在交换操作之后，得到的 128 位输出块 O1 用作密文 C1。然后将 O1 数据与第二块明文数据 P2' 进行逻辑异或运算，生成用于 AES 内核的 I2 输入数据以生成第二块密文数据。数据块的链接以这种方式继续，直到消息中的最后一个明文块被加密。

如果消息大小不是 128 位的倍数，则将按照*第 33.4.6 节：SAES 密文窃取和数据填充*中说明的方式对最后的不完整数据块进行加密。

图 286 说明了密码块链接 (CBC) 解密。

图 286. CBC 解密



在 CBC 解密模式下（如 ECB 解密模式），必须准备密钥才可以执行 AES 解密。

完成密钥准备过程后，按下述方式执行解密：使用 128 位或 256 位密钥将第一个 128 位密文块（在交换操作之后）直接用作用于解密操作的 AES 核心输入块 I1。其输出 O1 与 128 位 IVI 字段（必须与加密期间使用的字段相同）进行逻辑异或运算以生成第一个明文块 P1。

除了使用来自第一个块的 I1 数据代替初始化向量外，第二个密文块的处理方式与第一个块相同。

解密以这种方式继续，直到最后一个完整的密文块被解密。

如果消息大小不是 128 位的倍数，则将按照*第 33.4.6 节：SAES 密文窃取和数据填充*中说明的方式对最后的不完整数据块进行解密。

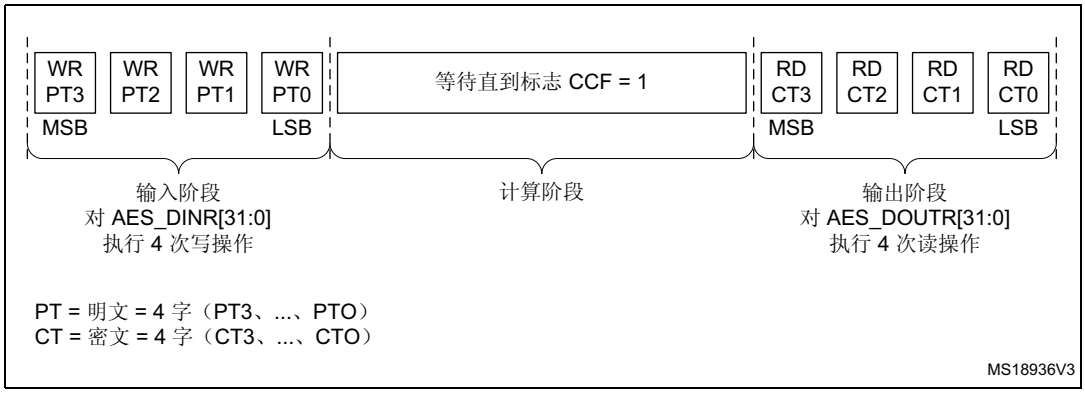
有关数据交换的更多信息，请参见*第 33.4.15 节：SAES 数据寄存器和数据交换*。

ECB/CBC 加密序列

执行 ECB/CBC 加密的事件序列（详见*第 33.4.4 节*）：

1. 确认 SAES_SR 中的 BUSY = 0（未在进行 RNG 随机数获取）。
2. 通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设。
3. 通过将 SAES_CR 寄存器的 MODE[1:0] 位字段设置为 00 来选择模式 1，通过将 SAES_CR 寄存器的 CHMOD[2:0] 位字段设置为 000 或 001 来分别选择 ECB 或 CBC 链接模式。也可以使用 DATATYPE[1:0] 位字段定义数据类型。通过将 KMOD[1:0] 设置为 00 选择正常密钥模式。有关使用其他 KMOD[1:0] 值的加密，请参见*第 33.4.14 节*和*第 33.4.13 节*。
4. 通过 SAES_CR 寄存器中的 KEYSIZE 位选择 128 位或 256 位密钥长度。如果密钥不能与不同的安全上下文（不同的安全属性）共享，还必须将 SAES_CR 寄存器的 KEYPROT 位置 1。
5. 将加密密钥写入 SAES_KEYRx 寄存器（128 或 256 位）。或者，通过 KEYSEL[2:0] 选择与密钥寄存器不同的密钥源。有关详细信息，请参见*第 33.4.16 节：SAES 密钥寄存器*。如果选择了 CBC 模式，则用初始化向量数据填充 SAES_IVRx 寄存器。
6. 通过将 SAES_CR 寄存器中的 EN 位置 1 来使能 SAES 外设。
7. 对 SAES_DINR 寄存器执行四次写入操作，以输入明文（MSB 优先），如*图 287* 所示。
8. 等待 SAES_SR 寄存器中的 CCF 标志置 1。
9. 对 SAES_DOUTR 寄存器进行四次读取操作，以获取密文（MSB 优先），如*图 287* 所示。然后通过将 SAES_ICR 寄存器中的 CCF 位置 1，将 CCF 标志清零。
10. 重复执行步骤 *7*-*8*-*9*，使用相同加密密钥处理所有块。

图 287. ECB/CBC 加密（模式 1）

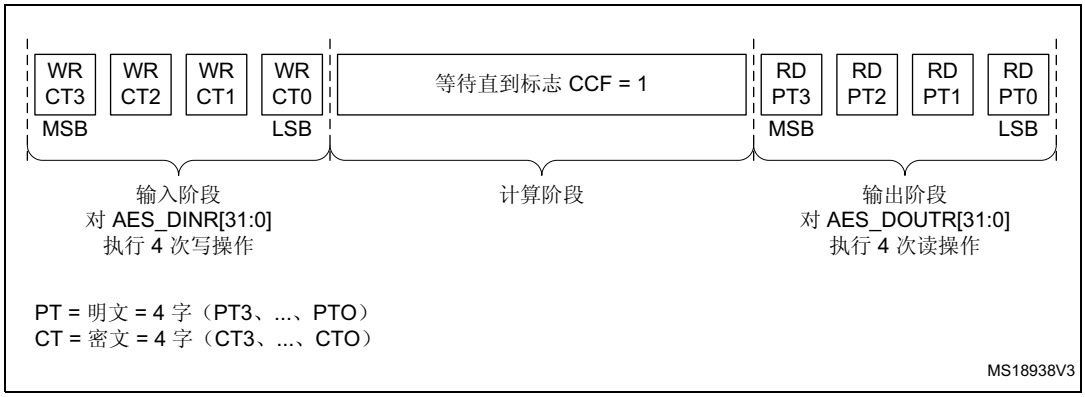


ECB/CBC 解密序列

执行 AES ECB/CBC 解密的事件序列如下所述（详见*第 33.4.4 节*）.通过将 KMOD[1:0] 设置为 00 选择正常密钥模式。有关使用其他 KMOD[1:0] 值进行解密的信息，请参见*第 33.4.14 节*和*第 33.4.13 节*。

1. 按照*第 33.4.5 节：SAES 解密轮密钥准备*中所述的步骤，准备 AES 内核中的解密密钥。
2. 通过将 SAES_CR 寄存器的 MODE[1:0] 位字段设置为 10 来选择模式 3，并通过将 SAES_CR 寄存器的 CHMOD[2:0] 位字段设置为 000 或 001 来分别选择 ECB 或 CBC 链接模式。也可以使用 DATATYPE[1:0] 位字段定义数据类型。KEYSIZE 和 KMOD[1:0] 位字段必须保持不变。
3. 将初始化向量写入 SAES_IVRx 寄存器（仅在 CBC 模式下需要此步骤）。
4. 通过将 SAES_CR 寄存器中的 EN 位置 1 来使能 SAES。
5. 对 SAES_DINR 寄存器执行四次写入操作，以输入密文（MSB 优先），如*图 288* 所示。
6. 等待 SAES_SR 寄存器中的 CCF 标志置 1。
7. 对 SAES_DOUTR 寄存器进行四次读取操作，以获取明文（MSB 优先），如*图 288* 所示。然后通过将 SAES_ICR 寄存器中的 CCF 位置 1，将 CCF 标志清零。
8. 重复执行步骤 *5*-*6*-*7*，使用相同密钥解密所有块。

图 288. ECB/CBC 解密（模式 3）



ECB/CBC 模式下的挂起/恢复操作

以下序列对于正常密钥模式 (KMOD[1:0] = 00) 有效。

要挂起消息处理，请执行以下操作：

1. 如果使用 DMA，则通过将 SAES_CR 寄存器中的 DMAINEN 位清零来停止 SAES DMA 对 IN FIFO 的数据传输。
2. 如果未使用 DMA，则读取四次 SAES_DOUTR 寄存器以保存最后处理的块。如果使用 DMA，等待直到 SAES_SR 寄存器中的 CCF 标志置 1，然后通过将 SAES_CR 寄存器中的 DMAOUTEN 输出位清零来停止 DMA 对 OUT FIFO 的数据传输。
3. 如果未使用 DMA，轮询 SAES_SR 寄存器的 CCF 标志，直到它变为 1（计算完成）。
4. 通过将 SAES_ICR 寄存器中的 CCF 位置 1，将 CCF 标志清零。
5. 保存初始化向量寄存器（仅在 CBC 模式下需要此步骤，因为 SAES_IVRx 寄存器在数据处理过程中已发生更改）。
6. 通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设。
7. 保存 SAES_CR 寄存器并将密钥寄存器清零（如果不再需要），以处理优先级更高的消息。

8. 如果使用 DMA，则保存 DMA 控制器状态（指向 IN 和 OUT 数据传输的指针以及剩余字节数等）。

要恢复消息处理，请执行以下操作：

1. 如果使用 DMA，则配置 DMA 控制器以完成 FIFO IN 和 FIFO OUT 传输的其余部分。
2. 通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设。
3. 恢复 SAES_CR 寄存器（使用正确的 KEYSIZE），然后恢复 SAES_KEYRx 寄存器。或者，使用 KEYSEL[2:0] 选择与密钥寄存器不同的密钥源。有关详细信息，请参见*第 33.4.16 节：SAES 密钥寄存器*。
4. 按*第 33.4.5 节：SAES 解密轮密钥准备*所述准备解密密钥（仅 ECB 或 CBC 解密需要此步骤）。
5. 使用保存的配置恢复 SAES_IVRx 寄存器（仅在 CBC 模式下需要此步骤）。
6. 通过将 SAES_CR 寄存器中的 EN 位置 1 来使能 SAES 外设。
7. 如果使用 DMA，将 SAES_CR 寄存器中的 DMAINEN 和 DMAOUTEN 位置 1，以使能 SAES DMA 传输。

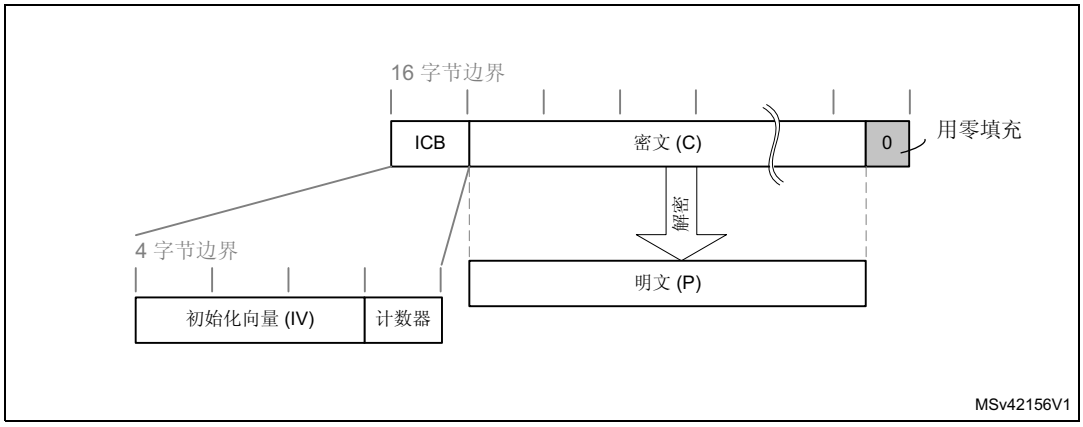
33.4.9 SAES 计数器 (CTR) 模式

概述

计数器模式 (CTR) 使用 SAES 作为密钥流生成器。然后，生成的密钥与明文进行逻辑异或运算来获得密文。

NIST *特别出版物 800-38A，块密码工作模式的建议*对 CTR 链接进行了相关定义。*图 289* 给出了 CTR 模式下的典型消息结构。

图 289. CTR 模式下的消息结构



该消息的结构为：

- 16 字节的初始计数器块 (ICB)，它由两个不同的字段组成：
  - **初始化向量** (IV)：96 位值，对于给定密钥下每个加密周期，该值都必须唯一。
  - **计数器**：32 位大端模式的整数，随着块的处理次数而递增。计数器的初始值必须设为 1。
- 明文 P 被加密为密文 C（长度已知）。该长度可以不是 16 字节的倍数，在这种情况下需要明文填充。

CTR 加密和解密

图 290 和*图 291* 分别描述了 SAES 外设中实现的 CTR 加密和解密过程。通过向 SAES_CR 寄存器中的 CHMOD[2:0] 位字段写入 010 来选择 CTR 模式。

图 290. CTR 加密

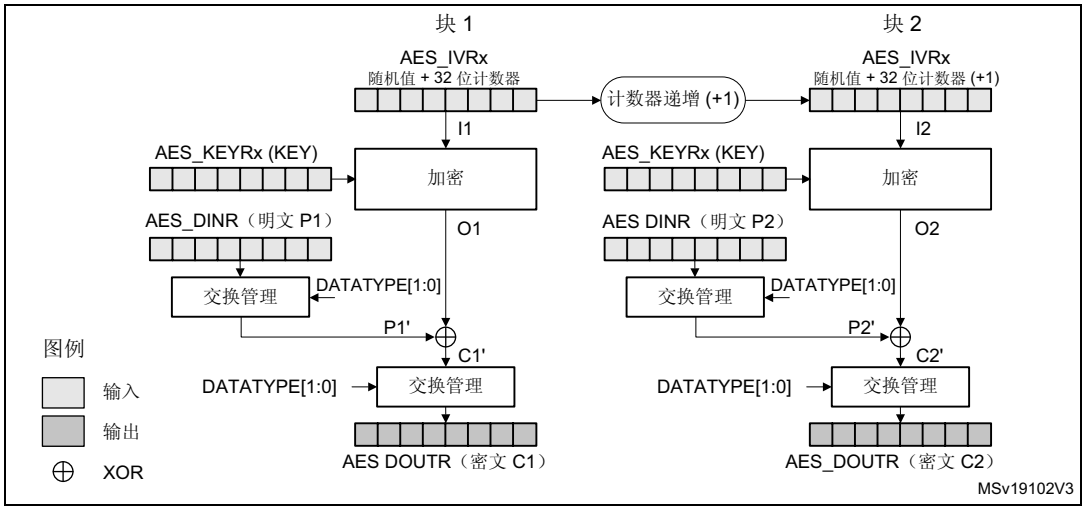
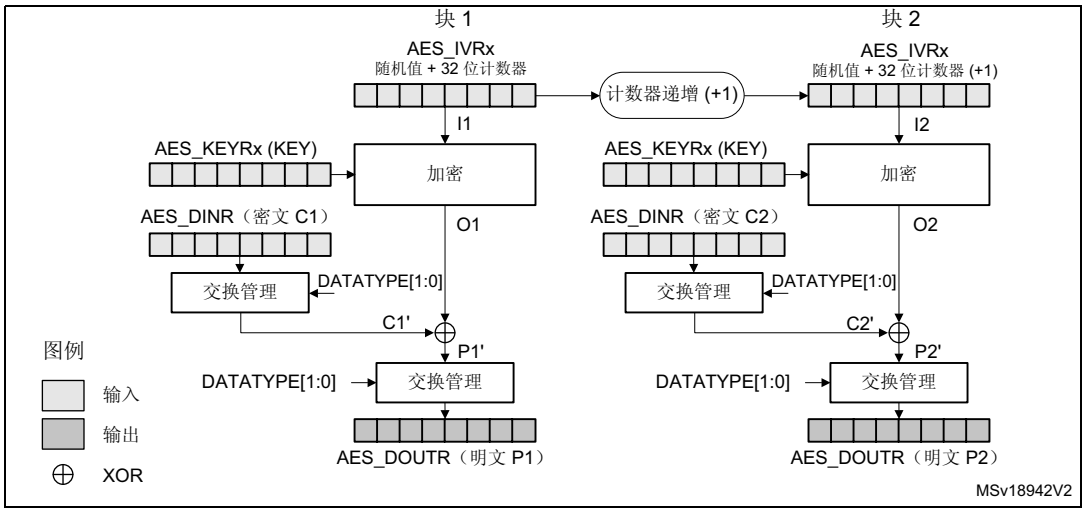


图 291. CTR 解密



在 CTR 模式中，将密码内核输出（也称为密钥流）Ox 与相关输入块（用于加密的 Px'，用于解密的 Cx'）进行逻辑异或运算，以生成正确的输出块（用于加密的 Cx'，用于解密的 Px'）。SAES 中的初始化向量必须按*表 273* 所示进行初始化。

表 273. CTR 模式初始化向量定义

SAES_IVR3[31:0]	SAES_IVR2[31:0]	SAES_IVR1[31:0]	SAES_IVR0[31:0]
IVI[127:96]	IVI[95:64]	IVI[63:32]	IVI[31:0] 32 位计数器 = 0x0001

在 CBC 模式下，处理第一个数据块时仅使用 SAES_IVRx 寄存器一次，而在 CTR 模式下则有所不同，处理每个数据块都使用 SAES_IVRx 寄存器，并且 SAES 外设会使初始化向量的计数器位递增（随机值位保持不变）。

CTR 解密与 CTR 加密并无不同，因为内核始终会对当前的计数器块加密以产生密钥数据流，该密钥数据流与明文（CTR 加密）或密文（CTR 解密）输入进行异或运算。在 CTR 模式下，禁止 MODE[1:0] 位字段设置 01（密钥派生），所有其他设置默认为加密模式。

在 CTR 链接模式下执行加密或解密的事件序列：

1. 确认 SAES_SR 中的 BUSY = 0（未在进行 RNG 随机数获取）。
2. 通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设。
3. 通过将 SAES_CR 寄存器中的 CHMOD[2:0] 位字段设置为 010 来选择 CTR 链接模式。将 MODE[1:0] 位字段设置为除 01 之外的任何值。
4. 初始化 SAES_KEYRx 寄存器，并按*表 273* 所述加载 SAES_IVRx 寄存器。
5. 将 SAES_CR 寄存器的 EN 位置 1，开始加密当前计数器（计算完成时 EN 自动复位）。
6. 如果是最后一个块，可在需要时用零填充数据以获得完整的块。
7. 在 SAES 中附加数据并读取结果。*第 33.4.4 节：执行密码操作的 SAES 过程*介绍了三种可能的情况。
8. 重复上一步，直到处理完倒数第二个块。对于最后一个块，执行前两步并将不属于有效负载的一部分的位丢弃（如果最后一个输入块中有效数据的大小小于 16 字节）。

CTR 模式下的挂起/恢复操作

该模式与 CBC 模式相似，可以中断消息、发送优先级更高的消息，并可恢复已中断的消息。有关 CBC 挂起/恢复序列的详细信息，请参见*第 33.4.8 节：SAES 基本链接模式（ECB 和 CBC）*。

注： *与 CBC 模式类似，恢复操作期间必须重新加载 SAES_IVRx 寄存器。*

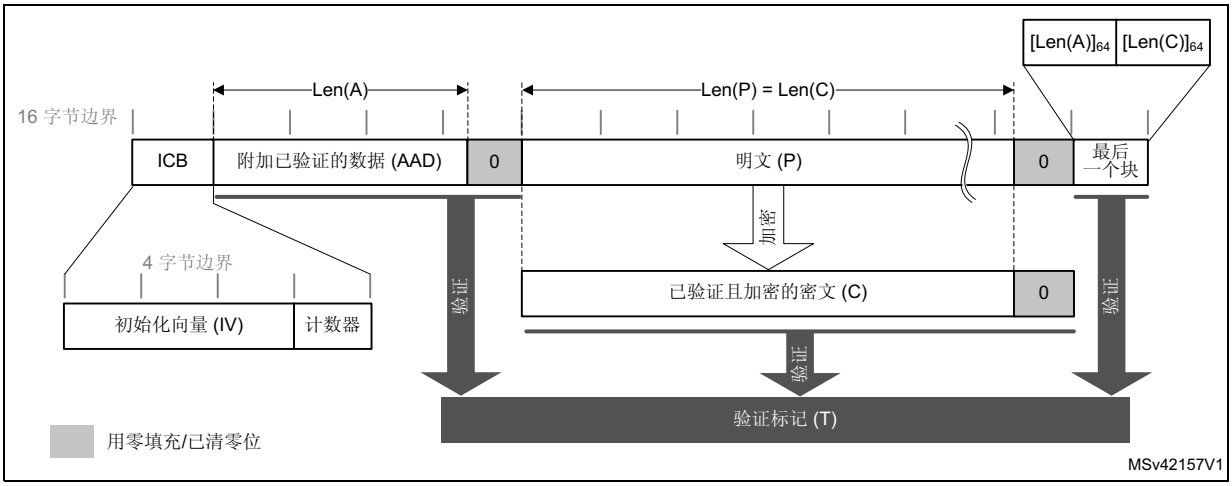
33.4.10 SAES Galois/计数器模式 (GCM)

概述

AES Galois/计数器模式 (GCM) 允许将明文消息加密和验证为对应的密文和标记（也称为消息认证码）。为实现保密性，GCM 算法基于 AES 计数器模式。它使用固定有限域乘法器来生成标记。

NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/计数器模式 (GCM) 和 GMAC* 中对 GCM 链接进行了定义。*图 292* 给出了 GCM 模式下的典型消息结构。

图 292. GCM 中的消息结构



消息具有以下结构：

- **16 字节的初始计数器块 (ICB)**，它由两个不同的位字段组成：
  - **初始化向量 (IV)**：96 位值，对于给定密钥下每个加密周期，该值都必须唯一。请注意，GCM 标准支持短于 96 位的 IV，在这种情况下应遵循严格的规则。
  - **计数器**：32 位大端模式的整数，随着块的处理次数而递增。根据 NIST 规范，处理有效负载的第一个块时，计数器值为 0x2。
- **认证的标头 AAD**（也称为附加认证数据）具有已知长度 Len(A)，此长度值可以不是 16 字节的倍数，但不能超过 $2^{64} – 1$ 位。消息的这一部分仅经过认证，未被加密。
- **明文消息 P** 将经过认证且被加密为密文 C，其具有已知长度 Len(P)，该长度值可以不是 16 字节的倍数，但不能超过 $2^{32} - 2$ 个 128 位块。
- **最后一个块**包含 AAD 标头长度（位 [32:63]）和有效负载长度（位 [96:127]）信息，如 *表 274* 所示。

GCM 标准规定密文 C 的位长度与明文 P 的位长度相同。

当消息某一部分（AAD 或 P）的长度不是 16 字节的倍数时，需要采取特殊填充方案。

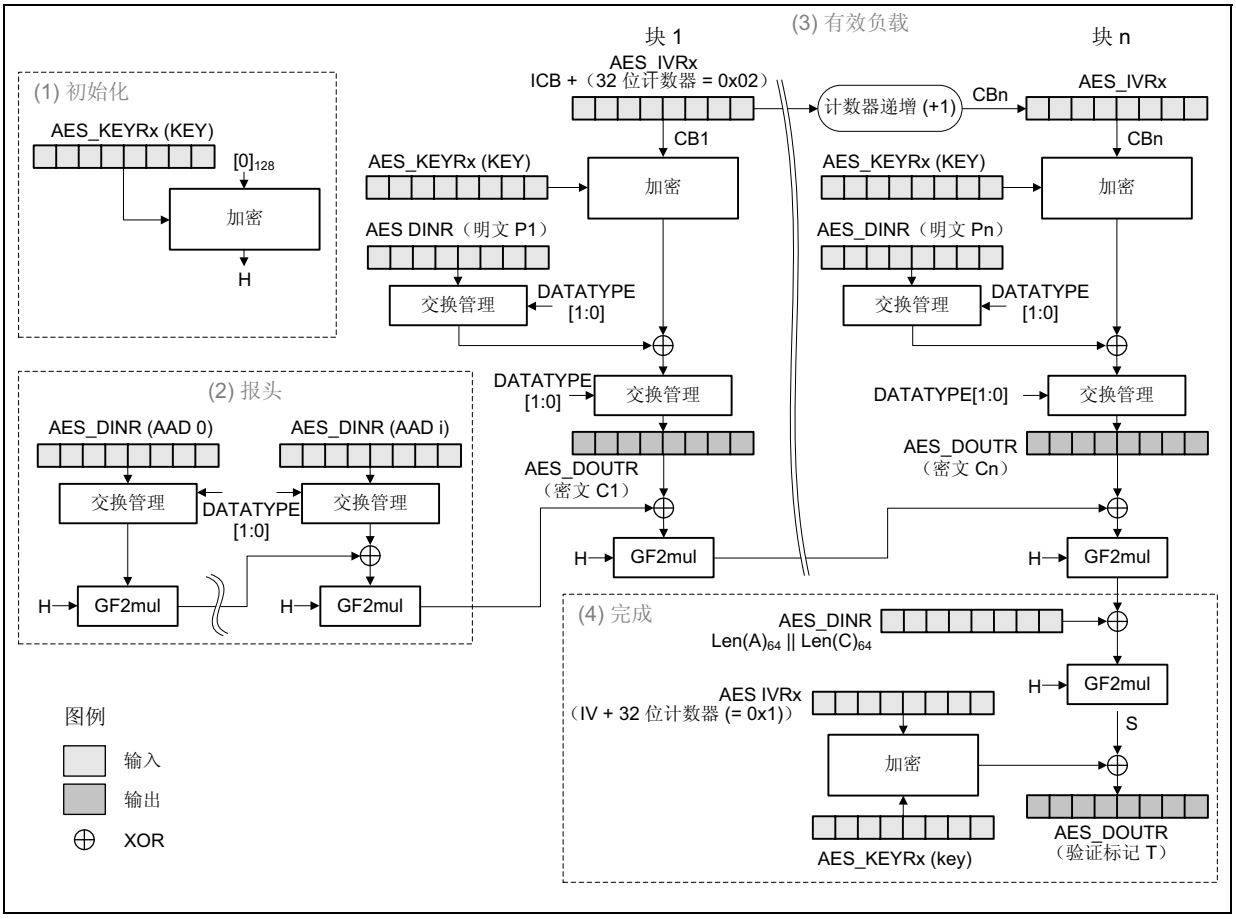
表 274. GCM 最后一个块定义

字节序	位 [0] ---------- 位 [31]	位 [32]---------- 位 [63]	位 [64] -------- 位 [95]	位 [96] --------- 位 [127]
输入数据	0x0	AAD 长度 [31:0]	0x0	有效负载长度 [31:0]

GCM 处理

图 293 列出了 SAES 外设中的 GCM 实现。通过向 SAES_CR 寄存器中的 CHMOD[2:0] 位字段写入 011 来选择 GCM。

图 293. GCM 已验证加密



GCM 模式下明文的保密性机制与计数器模式下的类似，其具有特定递增函数（表示为 32 位递增），可生成输入计数器块序列。

保持数据的**计数器块**的 SAES_IVRx 寄存器用于处理每个数据块。SAES 外设自动递增计数器 [31:0] 位字段。第一个计数器块 (CB1) 由应用软件从初始计数器块 ICB 分散（请参见 *表 275*）。

表 275. 在 GCM 模式下初始化 SAES_IVRx 寄存器

SAES_IVR3[31:0]	SAES_IVR2[31:0]	SAES_IVR1[31:0]	SAES_IVR0[31:0]
ICB[127:96]	ICB[95:64]	ICB[63:32]	ICB[31:0] 32 位计数器 = 0x0002

注：　在该模式下，禁止将 MODE[1:0] 位字段设置为 01（密钥派生）。

GCM 模式下的认证机制基于散列函数（称为 **GF2mul**），其在二元 Galois 域内执行与固定参数（称为散列子密钥 (H)）的乘法。

GCM 消息通过以下阶段进行处理（将在下一小节中对此进行进一步介绍）：

- **初始化阶段**：SAES 准备 GCM 散列子密钥 (H)。
- **标头阶段**：SAES 处理附加认证数据 (AAD)（仅使用散列计算）。
- **有效负载阶段**：SAES 基于散列计算、计数器块加密和数据异或运算处理明文 (P)。其操作方式与密文 (C) 类似。
- **最后阶段**：SAES 使用消息的最后一个块生成认证标记 (T)。

GCM 初始化阶段

在此第一步中，将在内部计算和保存 GCM 散列子密钥 (H)，供处理所有块使用。建议流程为：

1. 确认 SAES_SR 中的 BUSY = 0（未在进行 RNG 随机数获取）。
2. 通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设。
3. 通过将 SAES_CR 寄存器的 CHMOD[2:0] 位字段设置为 011 选择 GCM 链接模式，并可选择将 DATATYPE[1:0] 位字段进行相应的设置。
4. 通过将 SAES_CR 寄存器中的 GCMPH[1:0] 位字段设置为 00 来指示初始化阶段。
5. 将 SAES_CR 寄存器的 MODE[1:0] 位字段设置为 00 或 10。尽管该位字段仅在有效负载阶段使用，但建议在初始化阶段对其进行设置，并在所有后续阶段保持不变。
6. 使用密钥初始化 SAES_KEYRx 寄存器，使用*表 275* 中定义的信息初始化 SAES_IVRx 寄存器。
7. 通过将 SAES_CR 寄存器的 EN 位设置为 1 来开始计算散列密钥（计算完成时 EN 自动复位）。
8. 等待计算完成，通过 SAES_SR 的 CCF 标志转换为 1 获取指示。或者，使用相应的中断。
9. 通过将 SAES_ICR 寄存器中的 CCF 位置 1，将 SAES_SR 寄存器中的 CCF 标志清零。

GCM 标头阶段

GCM 初始化阶段之后的这个阶段必须在有效负载阶段之前完成。要执行的流程为（加密与解密完全相同）：

1. 通过将 SAES_CR 寄存器中的 GCMPH[1:0] 位字段设置为 01 来指示标头阶段。请勿修改初始化阶段中设置的 MODE[1:0] 位字段。
2. 通过将 SAES_CR 寄存器中的 EN 位置 1 来使能 SAES 外设。
3. 如果它是最后一个块，并且块中的 AAD 大小低于 128 位，则用零填充块的其余部分。然后按*第 33.4.4 节：执行密码操作的 SAES 过程*中所述的其中一种方式将数据块附加到 SAES。该阶段不读取任何数据。
4. 重复步骤 *3*，直到处理完最后一个附加认证数据块。

注： *如无 AAD（即 Len(A) = 0），则可以跳过标头阶段。*

GCM 有效负载阶段

在 GCM 标头阶段之后执行此阶段（加密和解密相同）。在此阶段，加密/解密的有效负载将存储在 SAES_DOUTR 寄存器中。要执行的流程为：

1. 通过将 SAES_CR 寄存器中的 GCMPH[1:0] 位字段设置为 10 来指示有效负载阶段。请勿修改初始化阶段中设置的 MODE[1:0] 位字段。
2. 如果跳过了标头阶段，则通过将 SAES_CR 寄存器中的 EN 位置 1 来使能 SAES 外设。
3. 如果它是最后一个块，并且块中的明文（加密）或密文（解密）大小低于 128 位，则用零填充块的其余部分。
4. 按*第 1696 页的第 33.4.4 节：执行密码操作的 SAES 过程*中所述的其中一种方式将数据块附加到 SAES，并读取结果。
5. 重复上一步，直到倒数第二个明文块被加密或直到最后一个密文块被解密。对于最后一个明文块（仅限加密），执行前两步。对于最后一个块，当其大小小于 16 字节时，将不属于有效负载的一部分的位丢弃。

注： *如无有效负载数据（即 Len(C) = 0），则可以跳过有效负载阶段（请参见 GMAC 模式）。*

GCM 最后阶段

在此最后一个阶段中，SAES 外设会生成 GCM 认证标记并将其存储在 SAES_DOUTR 寄存器中。要执行的步骤为：

1. 仅加密：等待直到 SAES_SR 寄存器中的 BUSY 标志清零。
2. 通过将 SAES_CR 寄存器中的 GCMPH[1:0] 位字段设置为 11 来指示最后阶段。
3. 通过连接 AAD 位长度和有效负载位长度来组成块的数据，如*表 274* 中所示。将块写入 SAES_DINR 寄存器。
4. 等待计算完成，通过 SAES_SR 的 CCF 标志转换为 1 获取指示。
5. 对 SAES_DOUTR 寄存器进行四次读取操作，以获取 GCM 认证标记。
6. 通过将 SAES_ICR 寄存器中的 CCF 位置 1，将 SAES_SR 寄存器中的 CCF 标志清零。
7. 通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设。如果它是经验证的解密，请将生成的标记与通过消息传递的预期标记进行比较。

注： *在最后阶段，数据通常写入 SAES_DINR（无交换），而交换则应用于从 SAES_DOUTR 读取的标签数据。*

当从标头或有效负载阶段过渡到最后阶段时，不得禁止 SAES 外设，否则结果会是错误的。

GCM 模式下的挂起/恢复操作

GCM 模式不支持挂起/恢复操作。

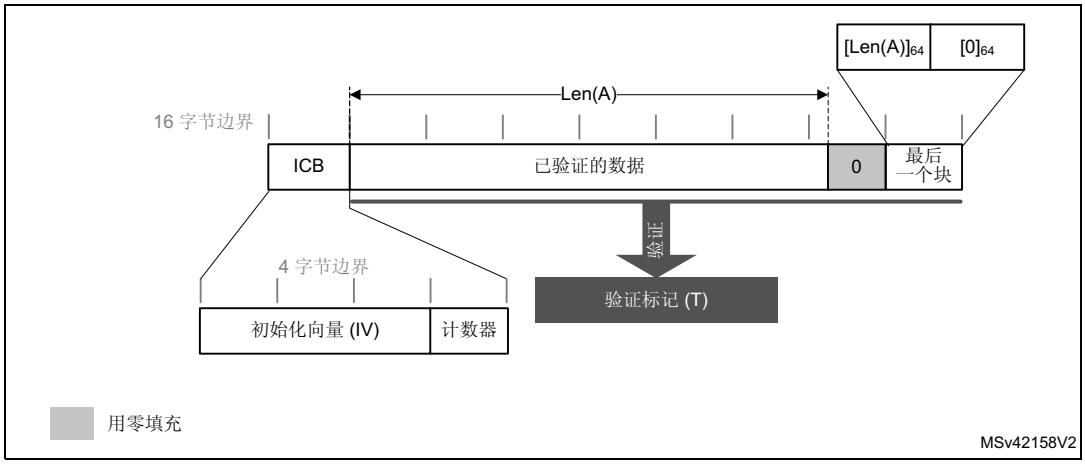
33.4.11 SAES Galois 消息认证码 (GMAC)

概述

Galois 消息认证码 (GMAC) 支持认证明文，生成相应的标记信息（也称为消息认证码）。它基于 GCM 算法，NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/ 计数器模式 (GCM) 和 GMAC* 中对此进行了相关定义。

图 294 给出了 GMAC 下的典型消息结构。

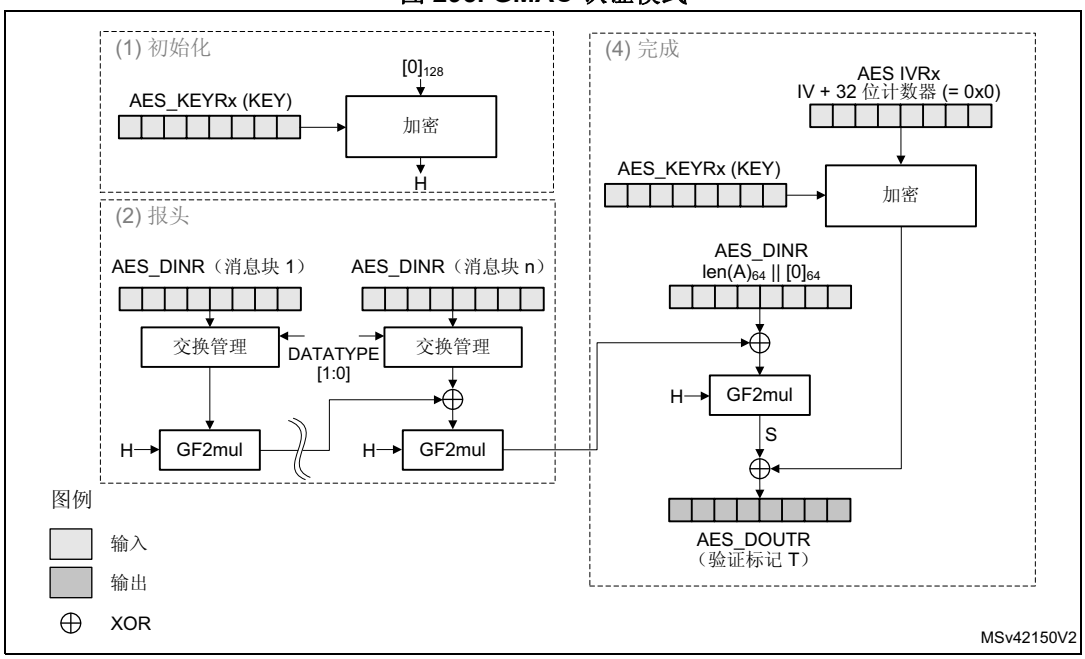
图 294. GMAC 模式下的消息结构



AES GMAC 处理

图 295 列出了 SAES 外设中的 GMAC 模式实现。通过向 SAES_CR 寄存器中的 CHMOD[2:0] 位字段写入 011 来选择此模式。

图 295. GMAC 认证模式



GMAC 算法相当于应用在仅包含标头的消息上的 GCM 算法。因此，除了忽略有效负载阶段之外，所有步骤和设置均与 GCM 相同。

GMAC 模式下的挂起/恢复操作

GMAC 模式不支持挂起/恢复操作。

33.4.12 SAES CBC-MAC 计数器模式 (CCM)

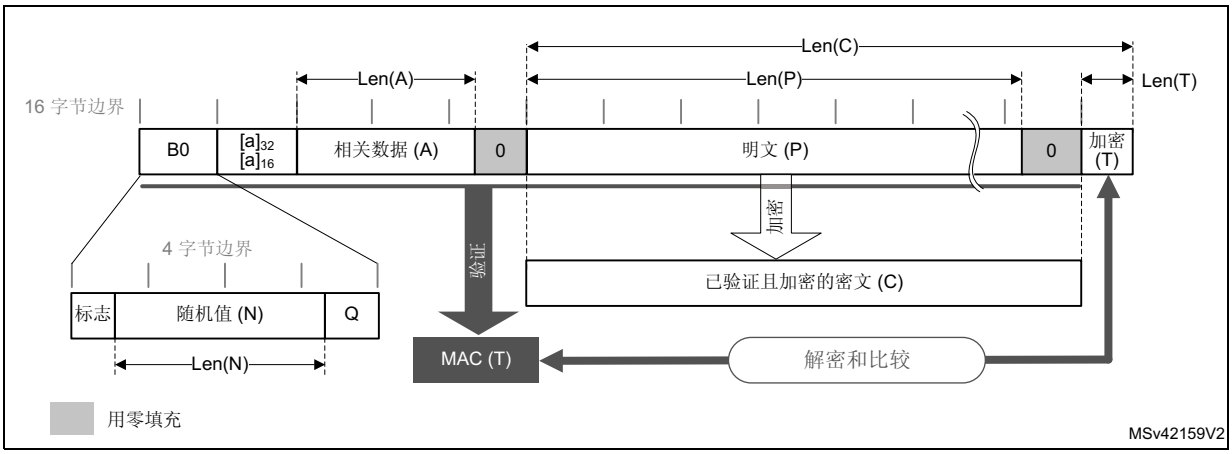
概述

AES **密码块链接-消息认证码计数器模式** (CCM) 算法可用于加密和认证明文，生成对应的密文和标记（也称为消息认证码）。为实现保密性，CCM 算法基于 SAES 计数器模式。它使用密码块链接技术来生成消息认证码。这通常被称为 CBC-MAC。

注： *NIST 不允许将该 CBC-MAC 用作 CCM 规范上下文之外的认证模式。*

NIST *特别出版物 800-38C，块密码工作模式的建议 - CCM 模式实现认证和保密性*中对 CCM 链接进行了介绍。*图 296* 给出了 CCM 下的典型消息结构。

图 296. CCM 模式下的消息结构



消息的结构为：

- **16 字节的首次认证块 (B0)**，它由三个不同的字段组成：
  - **Q**：位字符串，表示 P 的八位字节长度 (Len(P))
  - **随机值 (N)**：大小为 Len(N) 的一次性值（即，必须为每一次新的通信分配新的随机值）。Len(N) + Len(P) 的和必须等于 15 个字节。
  - **标志**：最高有效八位字节，包含四个控制信息标志（根据标准规定）。它包含两个 3 位字符串，用于编码值 **t**（MAC 长度，以字节表示）和 **Q**（明文长度，例如 Len(P) < 2^{8q} 个字节）。与 **Q** 相关的计数器块范围为 2^{8Q-4}，即，如果 **Q** 的最大值为 8，则密码中使用的计数器块必须为 60 位。

- 与相关数据 (A) 关联的 **16 字节块 (B)**。
  消息的这一部分仅经过认证，未被加密。这部分具有已知长度 Len(A)，此长度值可以不是 16 字节的倍数（请参见*图 296*）。该标准还规定：对于第一个消息块 (B1) 的 MSB 位，以字节表示的相关数据长度 (a) 必须按如下所述进行编码：
  - 如果 $0 < a < 2^{16} - 2^8$，则将其编码为 $[a]_{16}$，即两个字节。
  - 如果 $2^{16} - 2^8 < a < 2^{32}$，则将其编码为 0xff || 0xfe || $[a]_{32}$，即六个字节。
  - 如果 $2^{32} < a < 2^{64}$，则将其编码为 0xff || 0xff || $[a]_{64}$，即十个字节。
- 与明文消息 P 相关的 **16 字节块 (B)**，此明文消息将经过认证并被加密为密文 C（具有已知长度 Len(P)）。此长度可以不是 16 字节的倍数（请参见*图 296*）。
- 长度为 Len(T) 的**加密 MAC (T)** 被附加到总长度为 Len(C) 的密文 C。

当消息某一部分（A 或 P）的长度不是 16 字节的倍数时，需要采取特殊填充方案。

注： *CCM 链接模式同样只能与相关数据搭配使用（即，无有效负载）。*

例如，NIST 特别出版物 800-38C 的 C.1 部分给出了以下值（十六进制数）：

N：10111213 141516（Len(N) = 56 位或 7 字节）

A：00010203 04050607（Len(A) = 64 位或 8 字节）

P：20212223（Len(P) = 32 位或 4 字节）

T：6084341B（Len(T) = 32 位或 t = 4）

B0：4F101112 13141516 00000000 00000004
B1：00080001 02030405 06070000 00000000
B2：20212223 00000000 00000000 00000000

CTR0：0710111213 141516 00000000 00000000

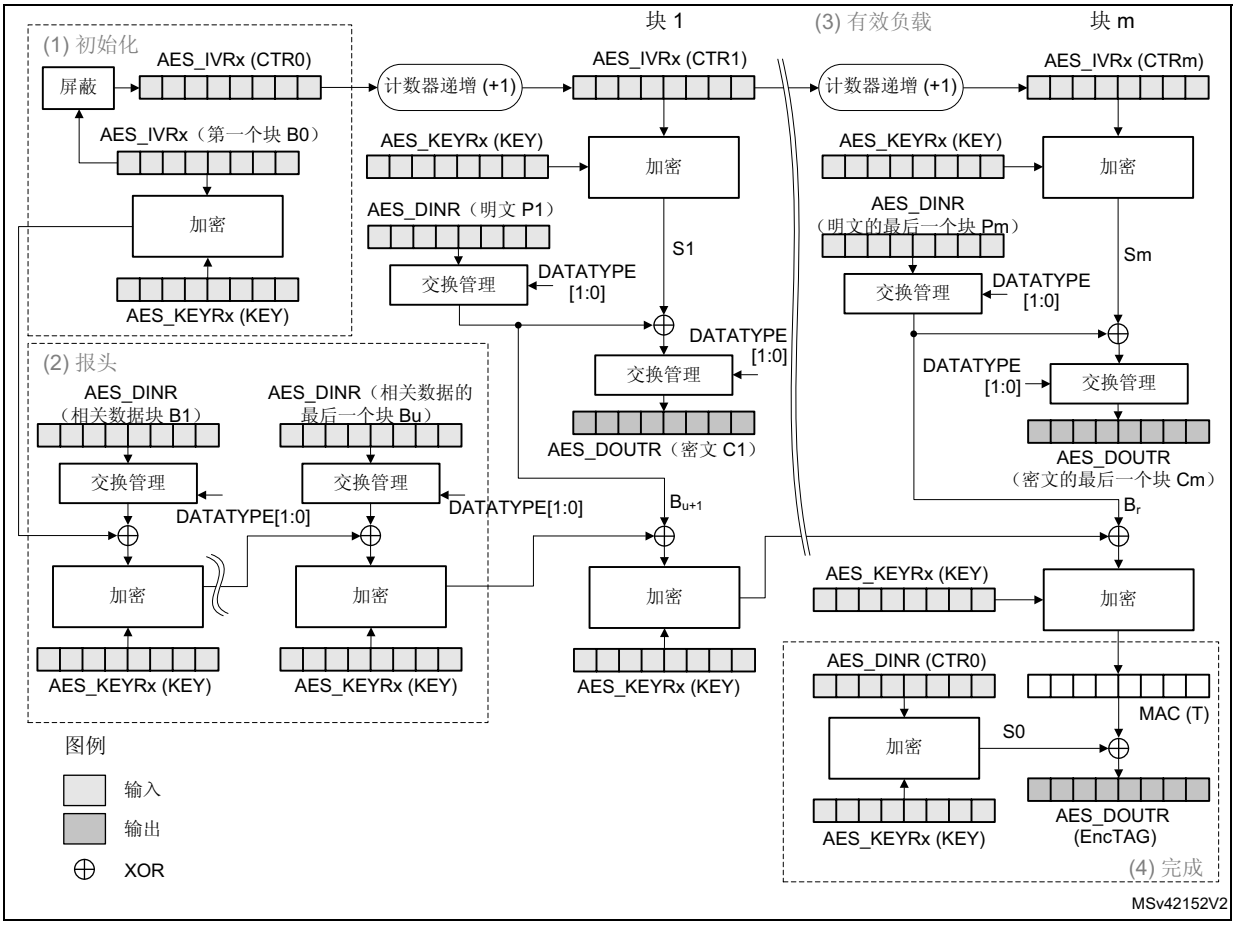
CTR1：0710111213 141516 00000000 00000001

格式化输入数据块 Bx（特别是 B0 和 B1）的生成必须由应用程序管理。

CCM 处理

图 297 列出了 SAES 外设中的 CCM 实现（加密示例）。通过向 SAES_CR 寄存器中的 CHMOD[2:0] 位字段写入 100 来选择此模式。

图 297. CCM 模式认证加密



生成-加密过程的数据输入包括一个有效随机值、一个有效的有效负载字符串和一个有效的相关数据字符串，这些数据均经过正确格式化。对经过格式化的明文数据应用 CBC 链接机制可生成长度已知的 MAC。计数器模式加密需要足够长的计数器块序列作为输入，应用于有效负载字符串，并单独应用于 MAC。生成的密文 C 是明文 P 加密生成过程的输出。

SAES_IVRx 寄存器用于处理每个数据块，SAES 以第一个块 B0 定义的位长度为增量，自动使 CTR 计数器递增*表 276*。显示了应用程序必须如何加载 B0 数据。

注： *根据 NIST 规定，CCM 模式下的 SAES 外设最多支持 64 位计数器。*

表 276. 在 CCM 模式下初始化 SAES_IVRx 寄存器

SAES_IVR3[31:0]	SAES_IVR2[31:0]	SAES_IVR1[31:0]	SAES_IVR0[31:0]
B0[127:96]	B0[95:64]	B0[63:32]	B0[31:0]

注： *在该模式下，禁止将 MODE[1:0] 位字段设置为 01（密钥派生）。*

CCM 消息通过以下阶段进行处理（将在下一小节中对此进行进一步介绍）：

- **初始化阶段**：SAES 处理第一个块并准备第一个计数器块。
- **标头阶段**：SAES 仅使用标记计算处理相关数据 (A)。
- **有效负载阶段**：IP 基于标记计算、计数器块加密和数据异或运算处理明文 (P)。其操作方式与密文 (C) 类似。
- **最后阶段**：SAES 生成消息认证码 (MAC)。

CCM 初始化阶段

在此阶段中，将 CCM 消息的第一个块 B0 写入 SAES_IVRx 寄存器中。SAES_DOUTR 寄存器不包含任何输出数据。建议流程为：

1. 确认 SAES_SR 中的 BUSY = 0（未在进行 RNG 随机数获取）。
2. 通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设。
3. 通过将 SAES_CR 寄存器的 CHMOD[2:0] 位字段设置为 100 选择 CCM 链接模式，并可选择将 DATATYPE[1:0] 位字段进行相应的设置。
4. 通过将 SAES_CR 寄存器中的 GCMPH[1:0] 位字段设置为 00 来指示初始化阶段。
5. 将 SAES_CR 寄存器的 MODE[1:0] 位字段设置为 00 或 10。尽管该位字段仅在有效负载阶段使用，但建议在初始化阶段对其进行设置，并在所有后续阶段保持不变。
6. 使用密钥初始化 SAES_KEYRx 寄存器，并如*表 276* 所述使用 B0 数据初始化 SAES_IVRx 寄存器。
7. 通过将 SAES_CR 寄存器的 EN 位设置为 1 来启动计数器计算（计算完成时 EN 自动复位）。
8. 等待计算完成，通过 SAES_SR 的 CCF 标志转换为 1 获取指示。或者，使用相应的中断。
9. 通过将 SAES_ICR 寄存器中的 CCF 位置 1，将 SAES_SR 寄存器中的 CCF 标志清零。

CCM 标头阶段

GCM 初始化阶段之后的这个阶段必须在有效负载阶段之前完成。在此阶段期间，SAES_DOUTR 寄存器不包含任何输出数据。

要执行的流程为（加密与解密完全相同）：

1. 通过将 SAES_CR 寄存器中的 GCMPH[1:0] 位字段设置为 01 来指示标头阶段。请勿修改初始化阶段中设置的 MODE[1:0] 位字段。
2. 通过将 SAES_CR 寄存器中的 EN 位置 1 来使能 SAES 外设。
3. 如果它是最后一个块，并且块中的 AAD 大小低于 128 位，则用零填充块的其余部分。然后按*第 33.4.4 节：执行密码操作的 SAES 过程*中所述的其中一种方式将数据块附加到 SAES。该阶段不读取任何数据。
4. 重复步骤 *3*，直到处理完最后一个附加认证数据块。

注： *如无相关数据（即 Len(A) = 0），则可以跳过标头阶段。*

必须由软件使用相关数据长度格式化相关数据的首个块 (B1)。

CCM 有效负载阶段（加密或解密）

在 CCM 标头阶段之后执行此阶段（加密和解密相同）。在此阶段，加密/解密的有效负载将存储在 SAES_DOUTR 寄存器中。要执行的流程为：

1. 通过将 SAES_CR 寄存器中的 GCMPH[1:0] 位字段设置为 10 来指示有效负载阶段。请勿修改初始化阶段中设置的 MODE[1:0] 位字段。
2. 如果跳过了标头阶段，则通过将 SAES_CR 寄存器中的 EN 位置 1 来使能 SAES 外设。
3. 如果它是最后一个待加密的数据块，并且块中明文大小低于 128 位，则用零填充块的其余部分。
4. 按*第 1696 页的第 33.4.4 节：执行密码操作的 SAES 过程*中所述的其中一种方式将数据块附加到 SAES，并读取结果。
5. 重复上一步，直到倒数第二个明文块被加密或直到最后一个密文块被解密。对于最后一个明文块（仅限加密），应用之前的两步。对于最后一个块，当其大小小于 16 字节时，将不属于有效负载的一部分的位丢弃。

注： *如果没有有效负载数据（即 Len(P) = 0 或 Len(C) = Len(T)），可以跳过有效负载阶段。*

解密密文 C 时，移除 $LSB_{Len(T)}(C)$ 加密标记信息。

CCM 最后阶段

在此最后一个阶段中，SAES 外设会生成 GCM 认证标记并将其存储在 SAES_DOUTR 寄存器中。要执行的流程为：

1. 通过将 SAES_CR 寄存器中的 GCMPH[1:0] 位字段设置为 11 来指示最后阶段。
2. 等待直到 SAES_SR 寄存器的计算完成标志 CCF 置 1。
3. 读取 SAES_DOUTR 寄存器四次：此输出即为 CCM 认证标记。
4. 通过将 SAES_ICR 寄存器中的 CCF 位置 1，将 SAES_SR 寄存器中的 CCF 标志清零。
5. 通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设。
6. 对于认证解密，将生成的加密标记与密文中填充的加密标记进行比较。

注： *在这个最后阶段，交换应用于从 SAES_DOUTR 寄存器读取的标签数据。*

当从标头阶段过渡到最后阶段时，不得禁止 SAES 外设，否则结果会是错误的。

应用程序必须使用标记长度屏蔽认证标记输出以获取有效标记。

CCM 模式下的挂起/恢复操作

要在标头或有效负载阶段挂起消息处理，请执行以下操作：

1. 如果使用 DMA，则通过将 SAES_CR 寄存器中的 DMAINEN 位清零来停止 SAES DMA 对 IN FIFO 的数据传输。如果未使用 DMA，确保当前计算已完成，完成由 SAES_SR 寄存器的 CCF 标志设置为 1 指示。
2. 在有效负载阶段，如果未使用 DMA，则读取四次 SAES_DOUTR 寄存器以保存最后处理的块。如果使用 DMA，等待 SAES_SR 寄存器中的 CCF 标志置 1，然后通过将 SAES_CR 寄存器中的 DMAOUTEN 输出位清零来停止 DMA 对 OUT FIFO 的数据传输。
3. 通过将 SAES_ICR 寄存器的 CCF 位置 1，将 SAES_SR 寄存器的 CCF 标志清零。
4. 将 SAES_SUSPxR 寄存器（其中 x 等于 0 到 7）保存到内存中。
5. 保存 SAES_IVRx 寄存器，因为在数据处理过程中，它们相较于初始值发生更改。
6. 通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设。
7. 将当前 SAES 配置保存到内存中（初始化向量寄存器 SAES_IVRx 除外）。无需保存密钥寄存器，因为应用程序已知初始密钥值。

8. 如果使用 DMA，则保存 DMA 控制器状态（指向 IN 数据传输的指针以及剩余字节数等）。在有效负载阶段，也必须保存指向 OUT 数据传输的指针。

要恢复消息处理，请执行以下操作：

1. 如果使用 DMA，则重新配置 DMA 控制器以完成 FIFO IN 传输的其余部分。在有效负载阶段，也必须在 DMA 控制器中配置 FIFO OUT 的其余部分。
2. 通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设。
3. 将先前保存在内存中的挂起寄存器值回写到其相应的 SAES_SUSPxR 寄存器中（其中 x 等于 0 到 7）。
4. 将先前保存在内存中的初始化向量寄存器值写回到其相应的 SAES_IVRx 寄存器中。
5. 恢复 SAES_CR 和 SAES_KEYRx 寄存器中的初始化设置值。
6. 通过将 SAES_CR 寄存器中的 EN 位置 1 来使能 SAES 外设。
7. 如果使用 DMA，将 SAES_CR 寄存器中的 DMAINEN 位置 1（如果在有效负载阶段，还要将 DMAOUTEN 位置 1），以使能 SAES DMA 请求。

33.4.13 SAES 使用封装密钥进行操作

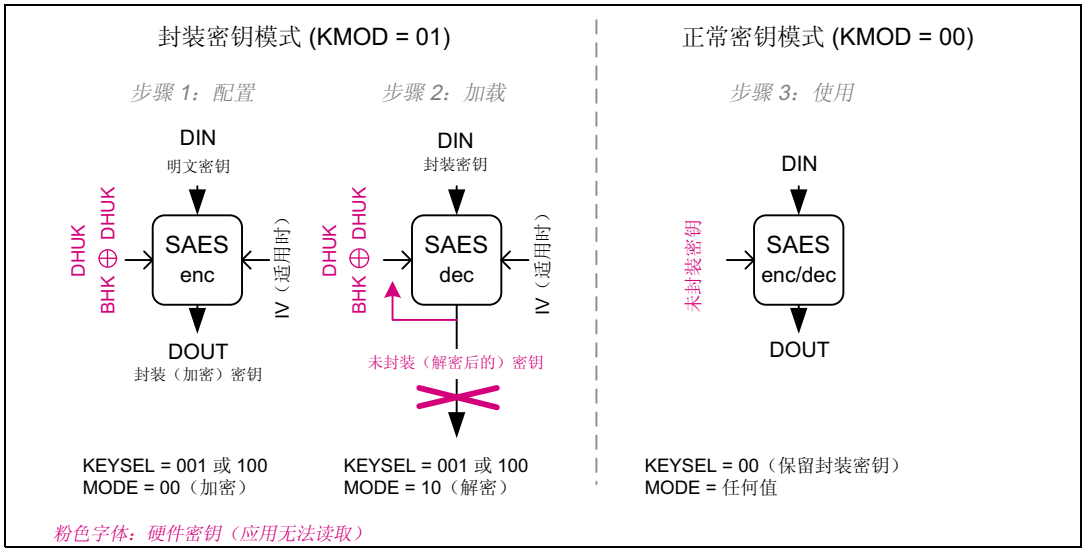
SAES 外设可以使用硬件密钥 DHUK（与应用密钥 BHK 进行或不进行异或运算），来封装（加密）和解封（解密）应用密钥。借助此功能，应用软件可以使用 AES 密钥，而无需以明文形式（未加密）公开。

封装密钥序列太小而无法挂起/恢复。SAES 无法使用未封装的密钥将密钥解封。

ECB 和 CBC 模式下的封装密钥操作

图 298 总结了 ECB 和 CBC 模式下的封装密钥操作。为了保护封装密钥，可通过将 KEYSEL[2:0] 设置为 001 或 100 来选择 DHUK。或者，如果 TAMP 外设中相应的寄存器被读/写锁定，可以通过将 KEYSEL 设置为 010 来选择 BHK。

图 298. ECB 和 CBC 模式下的封装密钥操作



注： *DHUK 值取决于权限、KMOD[1:0]、KEYSEL、CHMOD[2:0]，以及 SAES 外设是安全的还是非安全的。*

封装密钥模式下的加密

推荐的封装（即加密）密钥的顺序如下所述：

1. 确认 SAES_SR 中的 BUSY = 0（未在进行 RNG 随机数获取）。
2. 通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设。
3. 将 SAES_CR 寄存器中的 MODE[1:0] 位字段设置为 00 可选择模式 1，将 CHMOD[2:0] 位字段设置为 000 或 001 可分别选择 ECB 或 CBC 链接模式。DATATYPE[1:0] 位字段设置为 00 时，数据类型可以定义为 32 位。密钥大小必须使用 KEYSIZE 位来正确配置，并且 KMOD[1:0] 位字段必须设置为 01（封装密钥）。KEYSIZE 信息既可用于加密密钥，也可用于要加密的密钥。
4. 如果在上一步选择了 CBC 模式，则在 SAES_IVRx 寄存器中写入初始化向量数据。
5. 将 SAES_CR 寄存器的 KEYSEL[2:0] 位字段设置为 001 或 100 可选择 DHUK。成功加载密钥后，BUSY 位清零，并且 SAES_SR 寄存器中的 KEYVALID 位置 1。有关 KEYSEL = 100 用法的详细信息，请参见*第 33.4.16 节：SAES 密钥寄存器*。
6. 通过将 SAES_CR 寄存器中的 EN 位置 1 来使能 SAES。
7. 对 SAES_DINR 寄存器执行四次写入操作，以输入要加密的密钥（MSB 优先，参见*第 1725 页的表 277*）。
8. 等待 SAES_SR 寄存器中的 CCF 标志置 1。
9. 对 SAES_DOUTR 寄存器进行四次读取操作，以获取加密后的密钥（MSB 优先）。然后通过将 SAES_ICR 寄存器中的 CCF 位置 1，将 CCF 标志清零。
10. 如果 KEYSIZE 为 256 位，则重复步骤 6 至 8。

注： *仅当通过 CHMOD[2:0] 选择了 ECB 或 CBC 时，才支持以封装密钥模式加密。*

封装密钥模式下的解密

推荐的将封装密钥解封（即解密）的顺序如下所述：

1. 确认 SAES_SR 中的 BUSY = 0（未在进行 RNG 随机数获取）。
2. 通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设。
3. 将 SAES_CR 寄存器中的 MODE[1:0] 位字段设置为 01 可选择模式 2，将 CHMOD[2:0] 设置为 000 或 001 可分别选择 ECB 或 CBC 链接模式。密钥大小必须使用 KEYSIZE 位来正确配置，并且 KMOD[1:0] 位字段必须设置为 01（封装密钥）。选择数据类型时，DATATYPE[1:0] 位字段设置必须与加密期间使用的设置相同（即 0x0）。KEYSIZE 信息既可用于解密密钥，也可用于要解密的密钥。

   如果解密后的密钥不与不同的安全上下文（不同的安全属性）共享，还必须将 SAES_CR 寄存器的 KEYPROT 位置 1。
4. 将 SAES_CR 寄存器的 KEYSEL[2:0] 位字段设置为 001 或 100 可选择 DHUK。成功加载密钥后，SAES_SR 寄存器 BUSY 位清零，KEYVALID 位置 1。有关 KEYSEL = 100 用法的详细信息，请参见*第 33.4.16 节：SAES 密钥寄存器*。
5. 通过将 SAES_CR 寄存器中的 EN 位置 1 来使能 SAES。
6. 等待 SAES_SR 寄存器中的 CCF 标志置 1。
7. 将 CCF 标志清零。解密密钥在 AES 内核中可用，SAES 自动被禁止。
8. 将 SAES_CR 寄存器中的 MODE[1:0] 位字段设置为 10 可选择模式 3。
9. 如果在上一步选择了 CBC 模式，则在 SAES_IVRx 寄存器中写入初始化向量数据。
10. 通过将 SAES_CR 寄存器中的 EN 位置 1 来使能 SAES。
11. 对 SAES_DINR 寄存器执行四次写入操作，以输入加密的随机密钥（MSB 优先，参见*第 1725 页的表 277*）。
12. 等待 SAES_SR 寄存器中的 CCF 标志置 1。

13. 将 CCF 标志清零，如果 KEYSIZE 为 256 位，则重复步骤 10 和 11。

当解密后的密钥加载到密钥寄存器中时，SAES_CR 寄存器的 KEYSEL[2:0] 自动清零。因此，在该序列之后，解密的封装密钥在 SAES_KEYRx 寄存器中可用，应用软件可立即将其用于任何 AES 操作（正常密钥模式）。

如果在步骤 2 时将 KEYPROT 位清零，则解密的封装密钥可以与在不同安全上下文（不同安全属性）中运行的应用共享。

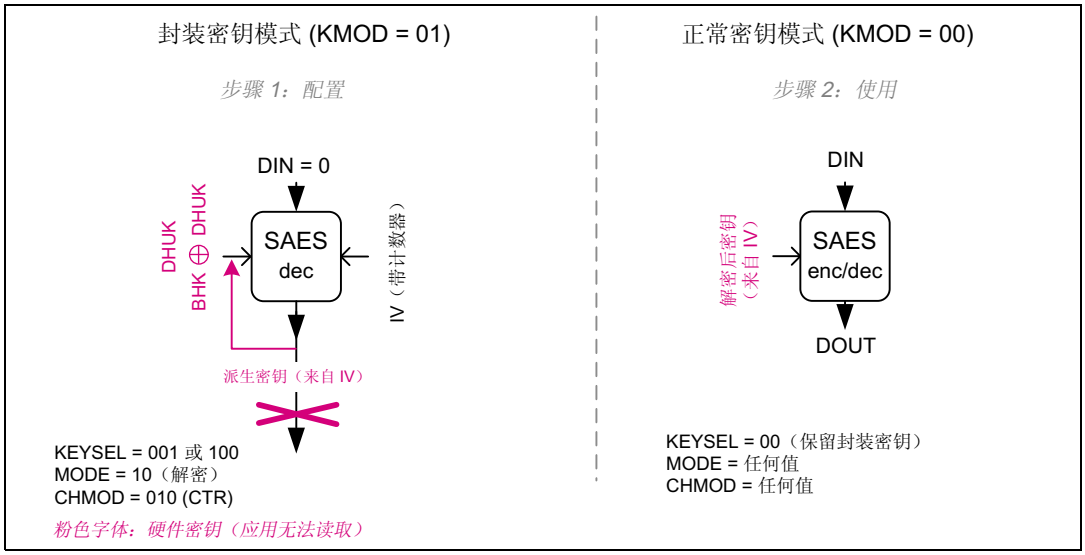
注： *当 KMOD[1:0] = 01（封装密钥）且 MODE[1:0] = 10（解密）时，对 SAES_DOUTR 寄存器进行读访问会触发读取错误 (RDERR)。*

当 KEYSEL[2:0] = 001 (DHUK) 或 100 (DHUK XOR BHK) 时，应用软件必须对加密和解密使用相同的特权、安全、KMOD[1:0]、CHMOD[2:0] 和 KEYSIZE 上下文。否则会导致结果不正确。

CTR 模式下的封装密钥操作

图 299 总结了 CTR 模式下的封装密钥操作。为了保护派生密钥，可通过将 KEYSEL[2:0] 设置为 001 或 100 来选择 DHUK。或者，如果 TAMP 外设中相应的寄存器被读/写锁定，可以通过将 KEYSEL 设置为 010 来选择 BHK。

图 299. CTR 模式下的封装密钥操作



注： *DHUK 值取决于权限、KMOD[1:0]、KEYSEL、CHMOD[2:0]，以及 SAES 外设是安全的还是非安全的。*

将 SAES 封装密钥模式与流密码 (CTR) 配合使用的推荐步骤如下：

1. 确认 SAES_SR 中的 BUSY = 0（未在进行 RNG 随机数获取）。
2. 通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设。
3. 将 SAES_CR 寄存器的 KMOD[1:0] 设置为 01（封装密钥）并通过 CHMOD[2:0] 选择 CTR 链接模式。使 DATATYPE[1:0] 保持为 00，并将 MODE 设置为 10（不支持值 00）。正确配置 KEYSIZE 位以设置密钥大小。它既用于加密密钥，也用于加载密钥。
4. 通过将 SAES_CR 寄存器中的 KEYSEL[2:0] 设置为 001 或 100 来选择 DHUK。成功加载密钥后，BUSY 位清零，并且 SAES_SR 寄存器中的 KEYVALID 位置 1。有关 KEYSEL = 100 用法的详细信息，请参见*图 33.4.16*。

5. 加载 SAES_IVRx 寄存器，并保持 SAES_IVR0 的两个最低有效位为 00
6. 通过将 SAES_CR 寄存器中的 EN 位置 1 来使能 SAES。
7. 等待 SAES_SR 寄存器中的 CCF 标志置 1。
8. 将 CCF 标志清零。派生的硬件密钥存储在 SAES_KEYRx 寄存器中。
9. 通过将 SAES_CR 寄存器中的 EN 位清零来禁止 SAES 外设。

基于 SAES_IVRx 寄存器中的公共数据派生的硬件密钥会自动加载到 SAES_KEYRx 寄存器中，同时也会清零 SAES_CR 寄存器的 KEYSEL[2:0] 位字段。之后，应用软件即可使用该硬件密钥进行所有 AES 操作（正常密钥模式）。

注： *设置 KMOD[1:0] = 01（封装密钥），CHMOD[2:0] = 010（CTR 链接）和 MODE = 00（加密）会通过自动清零 SAES_CR 寄存器的 EN 位禁止外设。*

33.4.14 使用共享密钥进行 SAES 操作

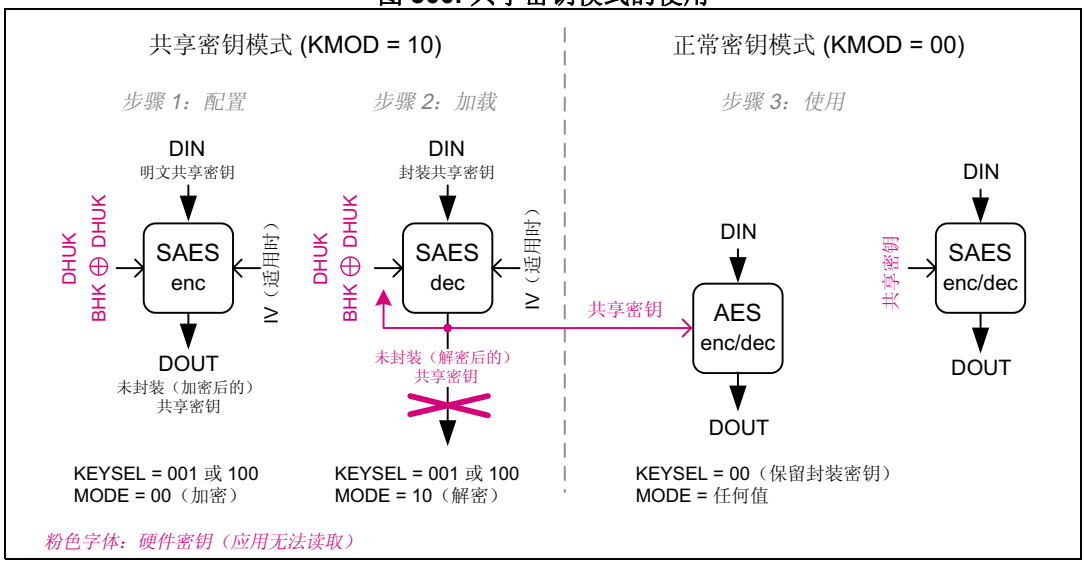
SAES 外设可以共享通过硬件加密密钥 DHUK（与应用密钥 BHK 进行或不进行异或运算）封装的应用密钥。应用软件可以通过该功能使 AES 密钥可用于 CRYP 外设，而不是以明文形式（未加密）公开这些密钥。

共享密钥序列太短而无法挂起/恢复。SAES 无法使用未封装密钥来解封共享密钥。

注： *当存储在 SAES 中的密钥与 CRYP 共享时，通过 KEYPROT 位提供的保护将丢失。有关保护的详细信息，请参见第 33.4.16 节：SAES 密钥寄存器。*

图 300 总结了如何使用共享密钥模式。为了保护共享密钥，必须通过将 KEYSEL[2:0] 设置为 001 或 100 来选择 DHUK。或者，如果 TAMP 外设中相应的寄存器被读/写锁定，可以通过将 KEYSEL 设置为 010 来选择 BHK。

图 300. 共享密钥模式的使用



注： *DHUK 值取决于特权、KMOD[1:0]、KSHAREID、KEYSEL、CHMOD[2:0]，以及 SAES 外设是安全的还是非安全的。*

在步骤 3 中，AES 表示 CRYP 外设。

共享密钥模式下的加密

必须先将密钥加密一次，SAES 才能与 CRYP 外设共享该密钥。KMOD[1:0] 设置为 10（共享密钥），KSHAREID[1:0] 保持为 00（*图 300* 的步骤 3 中）时，共享密钥的加密顺序与封装密钥相同。有关详细信息，请参见*封装密钥模式下的加密*。

注： *仅当通过 CHMOD[2:0] 选择了 ECB 或 CBC 时，才支持共享密钥模式下的加密。*

共享密钥模式下的解密

每次 SAES 需要与 CRYP 外设共享密钥时，加密后的共享密钥必须在 SAES 中解密，然后由 CRYP 加载。下文说明了整体顺序。

SAES 外设中的顺序

KMOD[1:0] 设置为 10（共享密钥），KSHAREID[1:0] 保持为 00（*图 300* 的步骤 3 中）时，共享密钥的解密顺序与封装密钥相同。有关详细信息，请参见*封装密钥模式下的解密*。

注： *由于 KEYSEL[2:0] 位字段会自动清零，解密的共享密钥可以直接用于 SAES，而不必共享。在这种情况下，KMOD[1:0] 必须设置为 00（正常密钥模式）。*

CRYP 外设中的顺序

一旦 SAES 密钥寄存器中的共享密钥被解密，就可以与 CRYP 外设共享，而 SAES 外设依然保持密钥共享状态，即，SAES_SR 寄存器中的 KMOD[1:0] = 10 并且 KEYVALID = 1。CRYP 密钥共享目标外设中的顺序如下所述：

1. 初始化 AES 处理器以处理一些数据
2. 如果必须加载密钥，则将 KEYSIZE 设置为与 SAES 外设的密钥大小相同，并向 CRYP_CR 寄存器的 KMOD[1:0] 位字段写入 10（共享密钥）。如果 SAES 外设中的上一个序列已成功完成，并且 KMOD[1:0] 保持为 10，KSHAREID[1:0] 保持为 00，则 SAES_KEYRx 寄存器被自动复制到 CRYP_KEYRx 寄存器中，同时 CRYP_SR 中的 BUSY 置 1。
3. 传输完成后，BUSY 标志立即清零，并且 CRYP_SR 寄存器中的 KEYVALID 置 1。如果在 BUSY 位清零时 KEYVALID 未置 1，或者密钥错误标志 (KEIF) 置 1，则传输期间已经发生意外事件，例如 DPA 错误、Tamper 事件或 KEYVALID SAES 标志在传输结束前被清零。当出现此类错误时，从 SAES 外设开始的整个密钥共享过程必须通过两个控制寄存器的 IPRST 位重新启动。
4. 由于 KEYVALID 置 1，密钥共享目标外设将使用有效的共享密钥进行初始化。应用可以继续处理数据，这时 KMOD[1:0] 设置为 00。

该步骤可以运行多次（例如，管理挂起/恢复情况），前提是未使用 SAES 并适当地保持在密钥共享状态。

注： *在 SAES 外设中，当 KMOD[1:0] = 10（共享密钥）且 MODE[1:0] = 10（解密）时，对 SAES_DOUTR 寄存器进行读访问会触发读取错误 (RDERR)。*

当 KEYSEL[2:0] = 001 (DHUK) 或 100 (DHUK XOR BHK) 时，应用软件必须对加密和解密使用相同的特权、安全、KMOD[1:0]/KSHAREID[1:0]、CHMOD[2:0]，和 KEYSIZE 上下文。否则会导致结果不正确。

当 CRYP 外设中的 KMOD[1:0] = 10 且 BUSY = 1 并且 CRYP 和 SAES 的 KEYSIZE 值不同时，密钥共享失败，并且两个外设中的 KEIF 标志均会置 1。

33.4.15 SAES 数据寄存器和数据交换

数据输入和输出

通过将四个连续的 32 位字写入 SAES_DINR 寄存器（位字段 DIN[127:0]），将 128 位数据块输入到 SAES 外设，先最高有效字（位 [127:96]），后最低有效字（位 [31:0]）。

通过从 SAES_DOUTR 寄存器（位字段 DOUT[127:0]）读取四个连续的 32 位字，从 SAES 外设检索 128 位数据块，先最高有效字（位 [127:96]），后最低有效字（位 [31:0]）。

写入 SAES_DINR 寄存器或从 SAES_DOUTR 寄存器读取的 32 位数据字使用大端模式进行组织，即：

- 要写入 SAES_DINR 的字的最高有效字节必须存储在保留要写入的字的四个相邻存储单元的最低地址中，或
- 从 SAES_DOUTR 读取的字的最高有效字节存储在接收字的四个相邻存储单元的最低地址中

要使用 DMA 将输入数据块写入 SAES，输入块的四个字必须以大端模式连续存储在内存中，即最高有效字存储在最低地址中。请参见*第 33.4.18 节：SAES DMA 接口*。

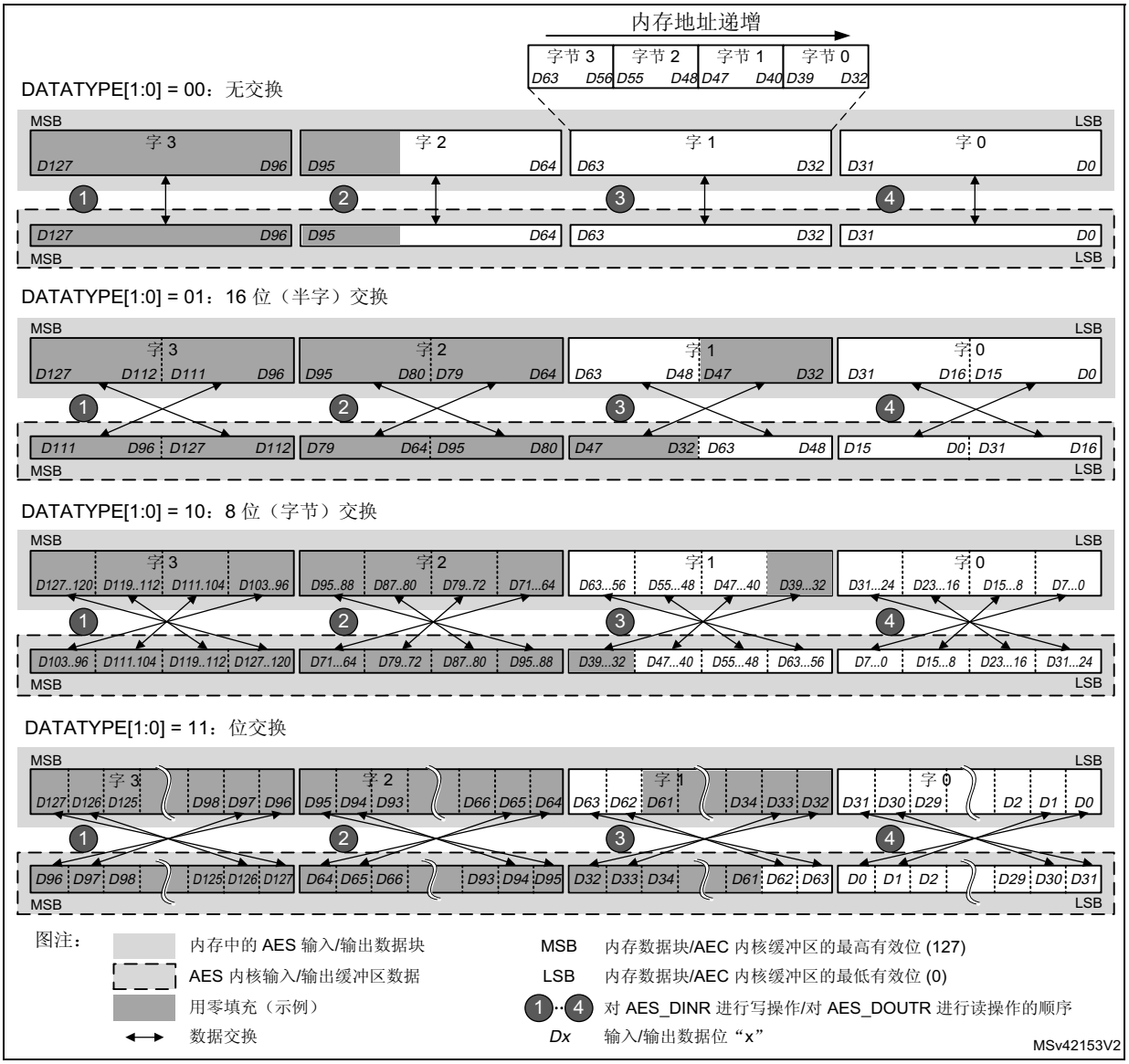
数据交换

可将 SAES 外设配置为在将其加载到 AES 处理内核前对 SAES_DINR 寄存器中的输入数据字，以及在将其发送到 SAES_DOUTR 寄存器前对 AES 处理内核的输出数据，执行位、字节、半字或无交换。选择取决于数据类型。例如，对 ASCII 文本流使用字节交换。

通过 SAES_CR 寄存器中的 DATATYPE[1:0] 位字段来选择数据交换类型。该选择适用于 AES 内核的输入和输出。

对于不同的数据交换类型，*图 301* 显示了通过 SAES_DINR 寄存器输入的输入数据的 AES 处理内核输入缓冲区数据 P127 到 P0 的结构，或通过 SAES_DOUTR 寄存器的输出数据的 AES 处理内核输出缓冲区数据 P127 到 P0 的结构。

图 301. 根据数据交换构建 128 位块



注： *SAES 密钥寄存器 (SAES_KEYRx) 和初始化寄存器 (SAES_IVRx) 中的数据对所选交换模式不敏感。*

数据填充

图 301 还给出了一个用零填充内存数据块的示例，这样数据交换后的补零位在 AES 内核输入缓冲区的 MSB 端形成一个连续区域。该示例显示了输入数据块的填充，包含：

- 48 个消息位，DATATYPE[1:0] = 01
- 56 个消息位，DATATYPE[1:0] = 10
- 34 个消息位，DATATYPE[1:0] = 11

33.4.16 SAES 密钥寄存器

SAES_KEYRx 只读寄存器存储加密或解密密钥位字段 KEY[127:0] 或 KEY[255:0]。要写入每个寄存器的数据以小端模式在内存中排列，即最高有效字节存储在最高地址中（出于安全原因，不允许读取）。

密钥分布在八个寄存器中，如*表 277* 所示。

表 277. SAES_KEYRx 寄存器中的密钥字节序（128 位或 256 位密钥长度）

SAES_KEYR7 [31:0]	SAES_KEYR6 [31:0]	SAES_KEYR5 [31:0]	SAES_KEYR4 [31:0]	SAES_KEYR3 [31:0]	SAES_KEYR2 [31:0]	SAES_KEYR1 [31:0]	SAES_KEYR0 [31:0]
-	-	-	-	KEY[127:96]	KEY[95:64]	KEY[63:32]	KEY[31:0]
KEY[255:224]	KEY[223:192]	KEY[191:160]	KEY[159:128]	KEY[127:96]	KEY[95:64]	KEY[63:32]	KEY[31:0]
TAMP_BKP7R [31:0]	TAMP_BKP6R [31:0]	TAMP_BKP5R [31:0]	TAMP_BKP4R [31:0]	TAMP_BKP3R [31:0]	TAMP_BKP2R [31:0]	TAMP_BKP1R [31:0]	TAMP_BKP0R [31:0]

如果通过将 SAES_CR 寄存器中的 EN 位和 KEYSEL[2:0] 位字段清零而禁止 SAES 外设，则用于加密或解密的密钥可以写入这些寄存器中。

密钥寄存器不受由 SAES_CR 寄存器的 DATATYPE[1:0] 位字段控制的数据交换影响。

必须在开始进行 AES 计算之前写入整个密钥。在正常密钥模式下 (KMOD[1:0] = 00)，且 KEYSEL[2:0] = 000，SAES_KEYRx（KEYSIZE = 0 时 x = 0 到 3；KEYSIZE = 1 时 x = 0 到 7）寄存器必须始终按升序或降序写入。

当 KEYSEL[2:0] 设置为 001 时，派生硬件唯一密钥 (DHUK)（在 SAES 引擎内从基于非易失性 OTP 的根硬件唯一密钥计算得出）将根据 KEYSIZE 信息直接加载到密钥寄存器中。

将 KEYSEL[2:0] 设置为 010 时，存储在防篡改安全备份寄存器中的启动硬件密钥 (BHK) 将在安全应用按升序或降序对所有 TAMP_BKPxR 寄存器（KEYSIZE = 0 时 x = 0 到 3；KEYSIZE = 1 时 x = 0 到 7）执行单次读取时，完整传输到密钥寄存器中。请参见*表 277*。

当 KEYSEL[2:0] 设置为 100 时，DHUK 和 BHK 的 XOR 组合将在安全应用按升序或降序对所有 TAMP_BKPxR 寄存器（KEYSIZE = 0 时 x = 0 到 3；KEYSIZE = 1 时 x = 0 到 7）执行单次读取时，完整传输到密钥寄存器中。请参见*表 277*。

如果 KEYVALID = 0，则使用相同的非零值重复写入 KEYSEL[2:0] 只会触发 DHUK 或 BHK 的加载。清零 KEYVALID 的推荐方法是将 SAES_CR 寄存器中的 IPRST 位置 1。例如，当从 ECB 解密切换到 ECB 加密、选择相同的 BHK (KEYSEL[2:0] = 010) 时，需要使用这种方法。

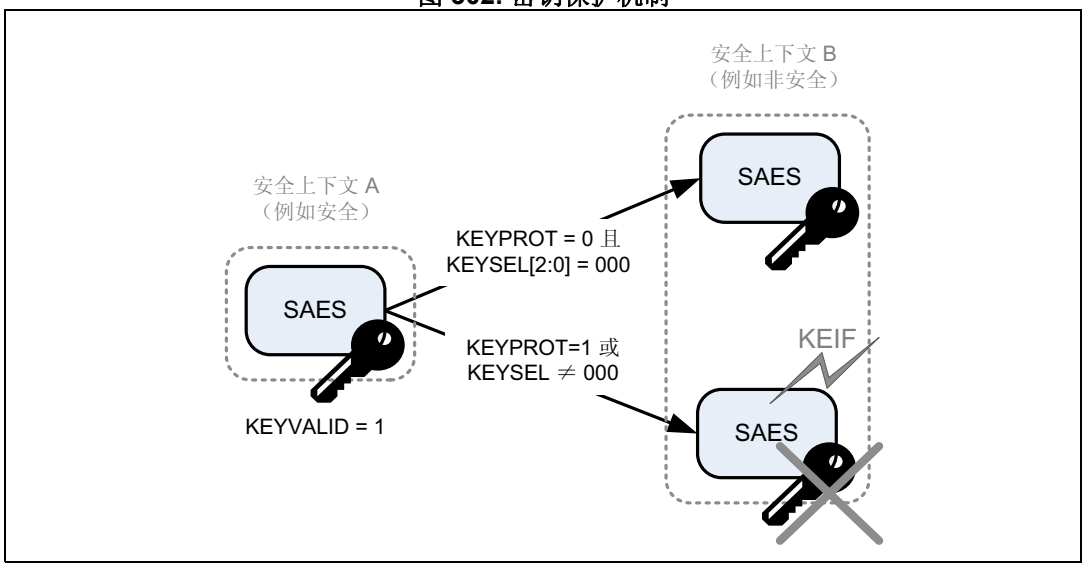
如果要载入到密钥寄存器中的密钥不得与在不同安全上下文（即不同的安全属性）中执行的应用共享，则 SAES_CR 寄存器的 KEYPROT 位必须置 1。如果将 KEYPROT 和 KEYVALID 置 1，那么，与加载密钥不同安全环境中的应用程序尝试 AES 访问时，KEIF 将标记错误。如*图 302* 所示。

注： *如果 KEYSEL[2:0] 的值不为零（正常密钥），会自动保护密钥寄存器。*

DHUK、BHK 及其 XOR 组合无法由任何软件（即使是安全的）读取。

安全 SAES 使用安全 DHUK。非安全 SAES 使用非安全 DHUK。

图 302. 密钥保护机制



注： *如果启动密钥加载序列，会将 BUSY 标志置 1，将 KEYVALID 标志清零。向 SAES_KEYRx 寄存器传输 KEYSIZE 定义的位数后，BUSY 将立即清零，KEYVALID 置 1，EN 位变为可写。如果发生错误，BUSY 和 KEYVALID 将被清零，KEIF 将置 1（详细信息，请参见第 33.4.19 节：SAES 错误管理）。这适用于所有 KEYSEL 值。*
有关密钥模式的更多信息，请参见第 33.4.14 节和第 33.4.13 节。

33.4.17 SAES 初始化向量寄存器

四个 SAES_IVRx 寄存器保持初始化向量输入位字段 IVI[127:0]。要写入每个寄存器或从中读取的数据以小端模式组织在内存中，即最高有效字节存储在最高地址中。寄存器也以最低地址 (SAES_IVR0) 到最高地址 (SAES_IVR3) 的顺序排序。

位字段中数据的意义取决于选择的链接模式。使用时，位字段在 AES 内核的每个计算周期内都会更新。

SAES 外设使能时，对 SAES_IVRx 寄存器的写入操作对寄存器内容没有影响。要修改 SAES_IVRx 寄存器的内容，必须首先清零 SAES_CR 寄存器的 EN 位。

读取 SAES_IVRx 寄存器会返回最新计数器值（用于管理挂起模式）。

SAES_IVRx 寄存器不受由 SAES_CR 寄存器的 DATATYPE[1:0] 位字段控制的数据交换功能的影响。

33.4.18 SAES DMA 接口

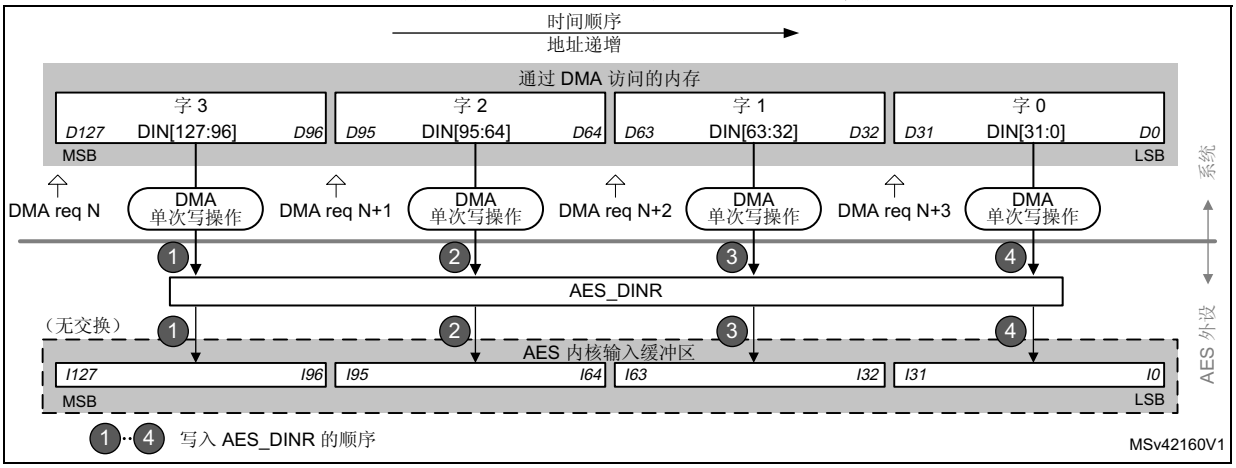
SAES 外设使用一个接口连接 DMA（直接内存访问）控制器。DMA 操作通过 SAES_CR 寄存器进行控制。

使用 DMA 的数据输入

通过将 SAES_CR 寄存器的 DMAINEN 位置 1，可使能 DMA 写入 SAES。然后，SAES 外设每次需要向 SAES_DINR 寄存器中写入一个 128 位块（四字）时，都会在输入阶段发出 DMA 请求，如*图 303* 所示。

注： *根据所选算法和模式，可能需要采用特殊填充/密文窃取技术。例如在 AES GCM 加密或 AES CCM 解密情况下，DMA 传输不得包括最后一个块。有关详细信息，请参见第 33.4.4 节：执行密码操作的 SAES 过程。*

图 303. 输入阶段 128 位数据块的 DMA 传输

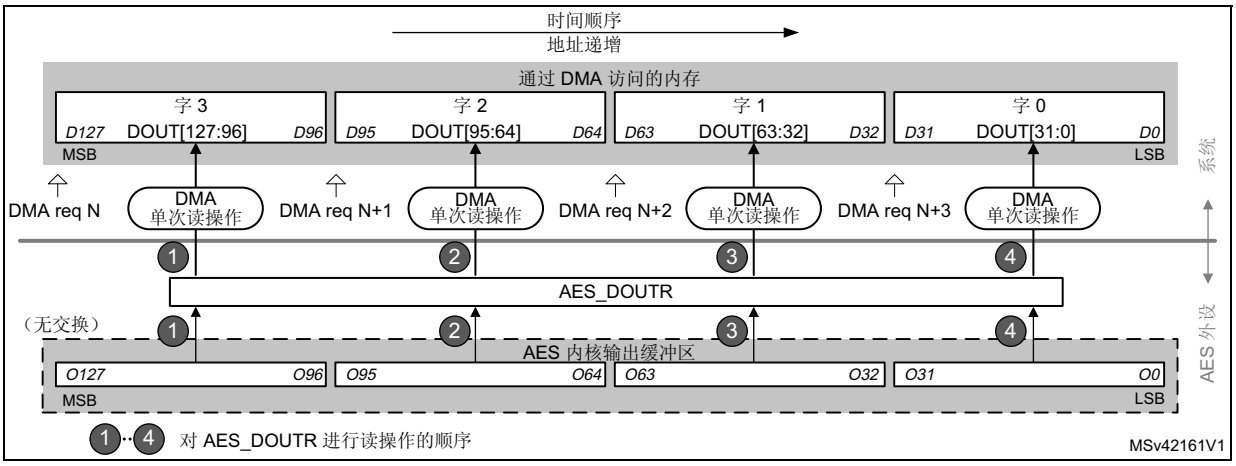


使用 DMA 的数据输出

通过将 SAES_CR 寄存器的 DMAOUTEN 位置 1，可使能 DMA 从 SAES 读取。然后，SAES 外设每次需要向 SAES_DINR 寄存器中读入一个 128 位块（四字）时，都会在输出阶段发出 DMA 请求，如*图 304* 所示。

注： *根据消息大小，应用程序可能需要丢弃最后一个块中的多余字节。*

图 304. 输出阶段 128 位数据块的 DMA 传输



不同工作模式下的 DMA 操作

当通过寄存器 SAES_CR 的 MODE[1:0] 位字段选择模式 1（加密）或模式 3（解密）时，可以使用 DMA 操作。与模式 2（密钥派生）一样，SAES_KEYRx 寄存器必须由软件写入，通过 SAES_CR 寄存器的 DMAINEN 和 DMAOUTEN 位使能 DMA 传输在该模式下无效。

SAES 会一直产生 DMA 单次请求，直到被禁止。因此，128 位数据块处理结束时的数据输出阶段之后，SAES 会自动切换到下一个数据块的新数据输入阶段（如果存在）。

当 SAES 与内存之间的数据传输由 DMA 管理时，CCF 标志没有用，因为在计算阶段结束时，SAES_DOUTR 寄存器的读取操作由 DMA 自动管理。只有数据传输转换回由软件管理时，才能将 CCF 标志清零。有关详细信息，请参见*第 33.4.4 节：执行密码操作的 SAES 过程*中的*附加数据*小节。

33.4.19 SAES 错误管理

除非另有说明，否则 SAES 配置可通过将 SAES_CR 寄存器的 EN 位清零随时进行更改。

读取错误标志 (RDERR)

如果意外尝试读取 SAES_DOUTR 寄存器，会将 SAES_SR 寄存器的 RDERR 标志和 SAES_ISR 寄存器的 RWEIF 标志置 1，并返回零。

在计算阶段或输入阶段会触发 RDERR。

注： *除非另有说明，否则在检测到 RDERR 错误时 SAES 不会被禁止，而会继续进行处理。*

如果 SAES_IER 寄存器中的 RWEIE 位置 1，会产生中断。更多详细信息，请参见*第 33.5 节：SAES 中断*。

RDERR 和 RWEIF 标志可通过将 SAES_ICR 寄存器的 RWEIE 位置 1 来清零。

写入错误标志 (WDERR)

如果意外尝试写入 SAES_DINR 寄存器，会将 SAES_SR 寄存器的 WRERR 标志和 SAES_ISR 寄存器的 RWEIF 标志置 1，而对 SAES_DINR 寄存器无效。在计算阶段或输出阶段会触发 WRERR。

注： *除非另有说明，否则在检测到 WRERR 错误后 SAES 不会被禁止，而会继续进行处理。*

如果 SAES_IER 寄存器中的 RWEIE 位置 1，会产生中断。更多详细信息，请参见*第 33.5 节：SAES 中断*。

WRERR 和 RWEIF 标志可通过将 SAES_ICR 寄存器的 RWEIF 位置 1 来清零。

密钥错误中断标志 (KEIF)

如果将密钥加载到密钥寄存器中失败，或在密钥受保护时尝试加载密钥，会将 SAES_ISR 寄存器的 KEIF 标志置 1，并将 SAES_SR 寄存器的 KEYVALID 位清零。

KEIF 标志通过 SAES_ICR 寄存器的相应位清零。如果 SAES_IER 寄存器的 KEIE 位置 1，会产生中断。更多详细信息，请参见*第 33.5 节：SAES 中断*。

密钥错误的可能来源包括：

- 密钥保护错误：KEYVALID 置 1 时，如果 KEYPROT = 1 或 KEYSEL 不为零，则当检测到在与用于加载密钥的安全上下文不同的安全上下文中执行的应用程序正在访问 SAES（即不同的安全属性）时，则会触发该错误。
- 密钥写入序列错误：写入密钥寄存器时检测到不正确的序列。有关详细信息，请参见*第 33.4.16 节：SAES 密钥寄存器*。
- 密钥共享大小不匹配：当 KMOD[1:0] = 10 并且应用在 CRYP 外设中设置的 KEYSIZE 信息与存储在 SAES 外设中 KEYSIZE 不匹配时，将触发错误
- 密钥共享错误：将密钥寄存器从 SAES 外设复制到 CRYP 失败。有关详细信息，请参见*第 33.4.14 节：使用共享密钥进行 SAES 操作*。

- 硬件密钥加载错误：DHUK 或 BHK 加载到 SAES 失败。KEYSEL = 001 (DHUK)、010 (BHK) 或 100 (DHUK XOR BHK) 时不起作用。

出现密钥共享错误时，通过 CRYP 和 SAES 外设的相应控制寄存器的 IPRST 位将这两个外设复位，然后重新启动密钥共享序列。

出现密钥选择错误时，将 KEIF 标志清零会自动重新启动密钥选择过程。对于持续存在的问题（例如，RHUK 加载失败），可能需要上电复位。

注： *对于任何关键错误，在重新配置 SAES 之前将 KEIF 标志清零。*

RNG 错误中断标志 (RNGEIF)

在 RCC 中触发了 IP 复位后，SAES 自动从 RNG 外设提取随机数。RNGEIF 置 1 时无法使用 SAES。

如果从 RNG 外设获取随机数时检测到错误（例如，由于熵错误），会将 SAES_ISR 寄存器的 RNGEIF 标志置 1。通过将 SAES_ICR 寄存器中的相应位置 1 可清零该标志。如果 SAES_IER 寄存器中的 RNGEIE 位置 1，则会产生中断。更多详细信息，请参见*第 33.5 节：SAES 中断*。

出现 RNG 错误时：

- 确认 RNG 外设 AHB 时钟已使能，并且该外设中没有挂起的噪声源（或种子）错误。
- 将 SAES_ISR 寄存器的 RNGEIF 位清零，或者通过将 SAES_CR 寄存器的 IPRST 位置 1 来复位外设。SAES_SR 寄存器的 BUSY 标志清零表示从 RNG 获取随机数已完成。

注： *为避免出现 RNGEIF 错误，建议每次 SAES AHB 时钟激活时，都激活 RNG AHB 时钟。*

关于 DPA 错误

如果出现意外错误，将在 TAMP 外设中触发一个 SAES 内部 Tamper 事件，并停止任何 SAES 协处理器处理。

要恢复正常操作，应用必须通过 RCC 或全局复位重置 SAES 外设。

33.5 SAES 中断

SAES 外设可生成独立可屏蔽中断源，用来对以下事件发送信号通知：

- 计算完成
- 读取错误
- 写入错误
- 密钥错误
- RNG 错误

这些源组合成一个公共中断信号，由连接 Arm® Cortex® 中断控制器的 SAES 外设发出。每个源都可以通过将 SAES_IER 寄存器的相应使能位置 1/清零来分别使能/禁止，并通过将 SAES_ICR 寄存器的相应位置 1 来清零。

每个源的状态可以从 SAES_SR 和 SAES_ISR 寄存器读取。

表 278 对中断源及其事件标志和使能位进行了总结。

表 278. SAES 中断请求

中断缩写	SAES 中断事件	事件标志	使能位	中断清除方法
SAES	计算完成标志	CCF	CCFIE	将 CCF 置 1(1)
	读取错误标志	RDERR(2)	RWEIE	将 RWEIF 置 1(1)
	写入错误标志	WRERR(2)		
	密钥错误标志	KEIF	KEIE	将 KEIF(1) 置 1
	RNG 错误标志	RNGEIF	RNGEIE	将 RNGEIF(1) 置 1

1. SAES_ICR 寄存器位
2. SAES_SR 寄存器的标志，通过 SAES_ISR 寄存器的 RWEIF 标志镜像。

33.6 SAES 处理延迟

下表列出了每种工作模式下处理 128 位块的延迟。

表 279. ECB、CBC 和 CTR 模式下的处理延迟

密钥大小	工作模式	算法	时钟周期(1)
128 位	模式 1：加密	ECB、CBC、CTR	**480**
	模式 2：密钥派生	-	**145**
	模式 3：解密	ECB、CBC、CTR	**480**
256 位	模式 1：加密	ECB、CBC、CTR	**680**
	模式 2：密钥派生	-	**230**
	模式 3：解密	ECB、CBC、CTR	**680**

1. SAES 内核时钟

表 280. GCM 和 CCM 模式下的处理延迟（SAES 内核时钟周期数）

密钥大小	工作模式	算法	初始化阶段	标头阶段(1)	有效负载阶段(1)	标记阶段(1)
128 位	模式 1：加密/ 模式 3：解密	GCM	490	72(2)	480(3)	490
		CCM	490	490	800	490
256 位	模式 1：加密/ 模式 3：解密	GCM	650	72(2)	690(3)	650
		CCM	650	680	1350	650

1. 数据插入可以包括 SAES 在 AHB 总线上强制的等待状态（最多 3 个周期，通常为 1 个周期）。
2. SAES AHB 时钟周期而不是内核时钟周期（仅限 Galois 乘法器）。
3. 作为加密模式下最坏的情况，为最后一个块计算添加额外 72 个 AHB 时钟周期。

33.7 SAES 寄存器

33.7.1 SAES 控制寄存器 (SAES_CR)

SAES control register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPRST	KEYSEL[2:0]			KSHAREID[1:0]		KMOD[1:0]		NPBLB[3:0]				KEYPROT	KEYSIZE	Res.	CHMOD[2]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	GCMPH[1:0]		DMAOUTEN	DMAINEN	Res.	Res.	Res.	Res.	CHMOD[1:0]		MODE[1:0]		DATATYPE[1:0]		EN
	rw	rw	rw	rw					rw	rw	rw	rw	rw	rw	rw

位 31 **IPRST**：SAES 外设软件复位 (SAES peripheral software reset)

将该位置 1 会将 SAES 外设复位,从而将所有寄存器设置为其默认值（IPRST 位本身 和 SAES_DPACFG 寄存器除外）。任何与密钥相关的数据都会丢失。因此，建议在将 SAES 交给安全性较低的应用之前，将该位置 1。

写入任何配置寄存器时，该位必须为低电平。

位 30:28 **KEYSEL[2:0]**：密钥选择 (Key selection)

该位字段定义 AES 密码内核中使用的密钥信息的来源。

000：软件密钥，加载到密钥寄存器 SAES_KEYx 中

001：派生硬件唯一密钥 (DHUK)

010：启动硬件密钥 (BHK)

100：DHUK 和 BHK 的异或运算

111：测试模式密钥（256 位硬件常数 0xA5A5...A5A5）

其他值：保留（如果使用，请通过 IPRST 解冻 SAES）

如果 KEYSEL 不为零，则当 SAES_SR 寄存器中的 BUSY 位清零且 KEYVALID 置 1 时，所选密钥值在密钥寄存器中可用。否则，密钥错误标志 KEIF 置 1。如果 KEYVALID = 0，则使用相同的非零值重复写入 KEYSEL[2:0] 只会触发 DHUK 或 BHK 的加载。

当应用软件通过写入 KEYSEL[2:0] 位字段来更改密钥选择时，密钥寄存器会立即被擦除，并且 KEYVALID 标志会清零。

在解密过程结束时，如果 KMOD[1:0] 不为零，则 KEYSEL[2:0] 会清零。

如果位字段值不为零并且 KEYVALID 置 1，则应用无法使用加载到在其他安全上下文（例如安全、非安全）中运行的应用的密钥，来转移 SAES 的所有权。更具体来讲，当对任何寄存器进行访问时的安全性与 SAES 记录的信息不匹配时，KEIF 标志被置 1。

当 SAES_SR 寄存器的 BUSY 标志置 1 时，以及当 SAES_CR 寄存器的 EN 位在写访问之前置 1 并且不会被该写访问清零时，对该位字段的写入将被忽略。

位 27:26 **KSHAREID[1:0]**：密钥共享标识 (Key share identification)

该位字段定义在解密过程结束并且 KMOD[1:0] = 10（共享密钥）时，可以使用专用硬件总线读取 SAES 密钥寄存器的目标。

00：CRYP 外设

其他值：保留

当 SAES_SR 寄存器的 BUSY 标志置 1 时，以及当 SAES_CR 寄存器的 EN 位在写访问之前置 1 并且不会被该写访问清零时，对该位段的写入将被忽略。

位 25:24 **KMOD[1:0]**：密钥模式选择 (Key mode selection)

该位字段定义应用如何使用 SAES 密钥：

00：正常密钥

01：封装密钥

10：共享密钥

其他值：保留

如果选择正常密钥，则可以自由使用密钥寄存器，不对 SAES_DIN 和 SAES_DOUT 寄存器应用特殊用途或保护。

如果选择封装密钥，则加载到密钥寄存器中的密钥只能用于加密或解密 AES 密钥。因此，当在封装密钥模式下选择解密时，在成功完成解密过程后，读为零 SAES_DOUT 寄存器自动加载到 SAES 密钥寄存器中。

如果选择共享密钥，则在成功完成解密过程后，SAES 密钥寄存器将共享给 KSHAREID[1:0] 位字段中描述的外设。仅当 KMOD[1:0] = 10 且 KEYVALID = 1 时，此共享才有效。当选择解密时，在成功完成解密过程后，读为零 SAES_DOUT 寄存器自动加载到 SAES 密钥寄存器中。

当 KMOD[1:0] 不为零时，任何将 SAES 外设配置为不同安全域（安全或非安全）的应用来使用的操作，都会导致自动擦除密钥和 KEIF 标志置 1。

当 SAES_SR 寄存器的 BUSY 标志置 1 时，以及当 SAES_CR 寄存器的 EN 位在写访问之前置 1 并且不会被该写访问清零时，对该位字段的写入将被忽略。

位 23:20 **NPBLB[3:0]**：最后一个块中填充字节的数量 (Number of padding bytes in last block)

该位字段用于设置有效负载的最后一个块中填充字节的数量：

0000：所有字节均有效（无填充）

0001：填充最后一个块的一个最低有效字节

...

1111：填充最后一个块的 15 个最低有效字节

位 19 **KEYPROT**：密钥保护 (Key protection)

置 1 后，将使能基于硬件的密钥保护。

0：当 KEYVALID 置 1 并且 KEYSEL = 0 时，应用可以将 SAES 的所有权及其加载的密钥转给在另一个安全上下文（例如非安全、安全）中运行的应用。

1：当 KEYVALID 被置位时，如果检测到对任何寄存器的访问，并且这个访问的安全上下文（例如安全、非安全）与加载密钥的应用程序的安全上下文不匹配，则密钥错误标志 (KEIF) 置 1。

当 SAES_SR 寄存器的 BUSY 标志置 1 时，以及当 SAES_CR 寄存器的 EN 位在写访问之前置 1 并且不会被该写访问清零时，对该位字段的写入将被忽略。

位 18 **KEYSIZE**：密钥大小选择 (Key size selection)

该位字段定义了 SAES 密码内核中使用的密钥长度（以位为单位）。

0：128

1：256

当 KMOD[1:0] = 01 或 10时，KEYSIZE 还将定义要加密或解密的密钥的长度。

当 SAES_SR 寄存器的 BUSY 标志置 1 时，以及当 SAES_CR 寄存器的 EN 位在写访问之前置 1 并且不会被该写访问清零时，对该位的写入将被忽略。

位 17 保留，必须保持复位值。

位 15 保留，必须保持复位值。

位 14:13 **GCMPH[1:0]**：GCM 或 CCM 阶段选择 (GCM or CCM Phase selection)

该位字段选择 GCM、GMAC 或 CCM 算法的阶段：

00：初始化阶段

01：标头阶段

10：有效负载阶段

11：最后阶段

如果选择除 GCM、GMAC 或 CCM 以外的算法（通过 ALGOMODE 位字段），则该位字段不起作用。

位 12 **DMAOUTEN**：DMA 输出使能 (DMA output enable)

该位在输出阶段使能/禁止使用 DMA 进行数据传输：

0：禁止

1：使能

该位置 1 时，SAES 会在输出数据阶段自动生成 DMA 请求。该功能仅在通过 MODE[1:0] 位字段选择了模式 1 或模式 3 时才有效。它不适用于模式 2（密钥派生）。

位 11 **DMAINEN**：DMA 输入使能 (DMA input enable)

该位在输入阶段使能/禁止使用 DMA 进行数据传输：

0：禁止

1：使能

该位置 1 时，SAES 会在输入数据阶段自动生成 DMA 请求。该功能仅在通过 MODE[1:0] 位字段选择了模式 1 或模式 3 时才有效。它不适用于模式 2（密钥派生）。

位 10:7 保留，必须保持复位值。

位 16、6:5 **CHMOD[2:0]**：链接模式选择 (Chaining mode selection)

该位字段选择 AES 链接模式：

000：电子密码本 (ECB)

001：密码块链接 (CBC)

010：计数器模式 (CTR)

011：Galois 计数器模式 (GCM) 和 Galois 消息认证码 (GMAC)

100：CBC-MAC 计数器模式 (CCM)

其他值：保留

当 SAES_SR 寄存器的 BUSY 标志置 1 时，以及当 SAES_CR 寄存器的 EN 位在写访问之前置 1 并且不会被该写访问清零时，对该位字段的写入将被忽略。

位 4:3 **MODE[1:0]**：SAES 工作模式 (SAES operating mode)

该位字段选择 SAES 工作模式：

00：模式 1：加密

01：模式 2：密钥派生（或针对 ECB/CBC 解密的密钥准备）

10：模式 3：解密

11：保留

当 SAES_SR 寄存器的 BUSY 标志置 1 时，以及当 SAES_CR 寄存器的 EN 位在写访问之前置 1 并且不会被该写访问清零时，对该位段的写入将被忽略。

位 2:1 **DATATYPE[1:0]**：数据类型选择 (Data type selection)

该位字段通过选择数据交换模式，定义了写入 SAES_DINR 寄存器或从 SAES_DOUTR 寄存器中读取的数据的格式：

00：无

01：半字（16 位）

10：字节（8 位）

11：位

更多详细信息，请参见*第 33.4.15 节：SAES 数据寄存器和数据交换*。

当 SAES_SR 寄存器的 BUSY 标志置 1 时，以及当 SAES_CR 寄存器的 EN 位在写访问之前置 1 并且不会被该写访问清零时，对该位段的写入将被忽略。

位 0 **EN**：SAES 使能 (SAES enable)

该位用于使能/禁止 SAES 外设：

0：禁止

1：使能

可随时通过将该位清零再置 1 来重新初始化 SAES 外设。

密钥准备完成后以及 GCM/GMAC/CCM 初始阶段完成后，该位由硬件自动清零（模式 2）。

只要 KEYVALID = 0，就无法将该位置 1，并且无法使用以下设置：KMOD[1:0] = 01 + CHMOD[2:0] = 011 和 KMOD[1:0] = 01 + CHMOD[2:0] = 010 + MODE[1:0] = 00。

注意：对于 KMOD[1:0] 不为 00 的情况，使用 IPRST 位而不是 EN 位。

33.7.2 SAES 状态寄存器 (SAES_SR)

SAES status register

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEYVALID	Res.	Res.	Res.	BUSY	WRERR	RDERR	CCF
								r				r	r	r	r

位 31:8 保留，必须保持复位值。

位 7 **KEYVALID**：密钥有效标志 (Key Valid flag)

当 SAES_KEYx 密钥寄存器中载入了由 SAES_CR 中的 KEYSIZE 所定义的密钥信息量时，该位由硬件置 1。

0：密钥寄存器中没有有效的密钥信息可用。SAES_CR 中的 EN 位无法置 1。

1：由 SAES_CR 中的 KEYSIZE 所定义的有效密钥信息加载到密钥寄存器中。

在正常模式下，当 KEYSEL 等于 0 时，应用必须按正确的顺序写入密钥寄存器，否则 SAES_ISR 寄存器的 KEIF 标志置 1，KEYVALID 保持为零。

当 KEYSEL 不为零时，BUSY 标志由 SAES 自动置 1。成功加载密钥后，BUSY 标志清零，KEYVALID 置 1。出现错误时，SAES_ISR 寄存器的 KEIF 标志置 1，BUSY 标志清零，KEYVALID 保持为零。

当 KEIF 标志置 1 时，应用必须通过 SAES_ICR 寄存器将其清零，否则无法将 KEYVALID 置 1。有关详细信息，请参见 KEIF 位描述。

有关密钥加载的更多信息，请参见*第 33.4.16 节：SAES 密钥寄存器*。

位 6:4 保留，必须保持复位值。

位 3 **BUSY**：忙

该标志指示 SAES 在 GCM 有效负载**加密**阶段是空闲还是繁忙：

0：空闲

1：忙

在 SAES 初始化、从 RNG 获取随机数或在向目标外设传送共享密钥时，该标志也会置 1。

当选择 GCM 加密时，必须在选择 GCM 最终阶段之前将标志清零。

位 2 **WRERR**：写入错误 (Write error)

该标志指示检测到 SAES_DINR 寄存器中发生了未预期的写入操作（计算阶段或数据输出阶段）：

0：未检测到

1：检测到

该标志由硬件置 1。将 SAES_ICR 寄存器中的 RWEIF 位置 1 时，该标志由软件清零。

该标志置 1 时，如果通过 SAES_ICR 寄存器的 RWEIE 位使能，则会产生中断。

标志置 1 对 SAES 操作没有影响。将忽略意外写入。

位 1 **RDERR**：读取错误标志 (Read error flag)

该标志指示检测到 SAES_DOUTR 寄存器中发生了未预期的读取操作（计算阶段或数据输入阶段）：

0：未检测到

1：检测到

该标志由硬件置 1。将 SAES_ICR 寄存器中的 RWEIF 位置 1 时，该标志由软件清零。

该标志置 1 时，如果通过 SAES_ICR 寄存器的 RWEIE 位使能，则会产生中断。

标志置 1 对 SAES 操作没有影响。意外读取返回零。

位 0 **CCF**：计算完成标志 (Computation completed flag)

该位镜像 SAES_ISR 寄存器的 CCF 位。

33.7.3 SAES 数据输入寄存器 (SAES_DINR)

SAES data input register

偏移地址：0x08

复位值：0x0000 0000

仅支持 32 位写入访问类型。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIN[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **DIN[31:0]**：输入数据字 (Input data word)

在输入阶段，对该位字段进行 4 倍连续写入，可将一个完整的 128 位输入数据块写入 SAES 外设。从第一次写入到第四次写入，对应的数据权重分别为 [127:96]、[95:64]、[63:32] 和 [31:0]。在每次写入时，数据交换块根据 DATATYPE[1:0] 位字段对 32 位输入缓冲区的数据进行处理，然后写入 AES 内核 128 位输入缓冲区。

输入数据块的数据的意义取决于 SAES 工作模式：

- **模式 1**（加密）：明文
- **模式 2**（密钥派生）：未使用该位字段（如果 KEYSEL = 0，将 SAES_KEYRx 寄存器用于输入）
- **模式 3**（解密）：密文

*第 1723 页的第 33.4.15 节：SAES 数据寄存器和数据交换*对数据交换操作进行了介绍。

33.7.4 SAES 数据输出寄存器 (SAES_DOUTR)

SAES data output register

偏移地址：0x0C

复位值：0x0000 0000

仅支持 32 位读访问类型。如果在 KMOD[1:0] = 01 或 10，而 MODE[1:0] = 10 且 EN = 1 时读取，会触发读取错误。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DOUT[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOUT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **DOUT[31:0]**：输出数据字 (Output data word)

该只读位字段获取一个 32 位输出缓冲区。计算完成后（CCF 置 1），对该位字段进行 4 倍连续读取，可从 SAES 外设读取一个完整的 128 位输出数据块。在到达输出缓冲区之前，数据交换块根据 DATATYPE[1:0] 位字段对 AES 内核产生的数据进行处理。

从第一次到第四次读取操作的数据权重分别为：[127:96]、[95:64]、[63:32] 和 [31:0]。

输出数据块的数据的意义取决于 SAES 工作模式：

- **模式 1**（加密）：密文
- **模式 2**（密钥派生）：不使用该位字段
- **模式 3**（解密）：纯文本

*第 1723 页的第 33.4.15 节：SAES 数据寄存器和数据交换*对数据交换操作进行了介绍。

33.7.5 SAES 密钥寄存器 0 (SAES_KEYR0)

SAES key register 0

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **KEY[31:0]**：密钥 (Cryptographic key)，位 [31:0]

该只写位字段包含 AES 加密或解密密钥的位 [31:0]，具体取决于工作模式：

- 在**模式 1**（加密）、**模式 2**（密钥派生）中：写入该位字段的值是加密密钥。
- 在**模式 3**（解密）中：写入该位字段的值是在用于解密之前进行派生的加密密钥。

仅当 KEYSIZE 值正确并且 SAES 外设已禁止（SAES_CR 寄存器的 EN 位清零）时，才可写入 SAES_KEYRx 寄存器。还需要特殊的写入序列，如 SAES_SR 寄存器的 KEYVALID 位所描述。请注意，如果 KEYSEL 不等于 0 并且 KEYVALID = 0，则密钥将直接加载 SAES_KEYRx 寄存器（因此将忽略密钥寄存器写入，并且 KEIF 置 1）。

更多详细信息，请参见*第 1725 页的第 33.4.16 节：SAES 密钥寄存器*。

33.7.6 SAES 密钥寄存器 1 (SAES_KEYR1)

SAES key register 1

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[63:48]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[47:32]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **KEY[63:32]**：密钥 (Cryptographic key)，位 [63:32]

有关 KEY[255:0] 位字段的说明，请参见 SAES_KEYR0 寄存器。

33.7.7 SAES 密钥寄存器 2 (SAES_KEYR2)

SAES key register 2

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[95:80]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[79:64]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **KEY[95:64]**：密钥 (Cryptographic key)，位[95:64]
有关 KEY[255:0] 位字段的说明，请参见 SAES_KEYR0 寄存器。

33.7.8 SAES 密钥寄存器 3 (SAES_KEYR3)

SAES key register 3

偏移地址：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[127:112]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[111:96]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **KEY[127:96]**：密钥 (Cryptographic key)，位[127:96]
有关 KEY[255:0] 位字段的说明，请参见 SAES_KEYR0 寄存器。

33.7.9 SAES 初始化向量寄存器 0 (SAES_IVR0)

SAES initialization vector register 0

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVI[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IVI[31:0]**：初始化向量输入 (Initialization vector input)，位 [31:0]
有关 IVI[127:0] 位字段的说明，请参见*第 1726 页的第 33.4.17 节：SAES 初始化向量寄存器*。
初始化向量仅用于除 ECB 以外的链接模式。
只有在禁止 SAES 外设时，才能写入 SAES_IVRx 寄存器。

33.7.10 SAES 初始化向量寄存器 1 (SAES_IVR1)

SAES initialization vector register 1

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[63:48]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVI[47:32]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IVI[63:32]**：初始化向量输入 (Initialization vector input)，位 [63:32]
有关 IVI[128:0] 位字段的说明，请参见 SAES_IVR0 寄存器。

33.7.11 SAES 初始化向量寄存器 2 (SAES_IVR2)

SAES initialization vector register 2

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[95:80]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVI[79:64]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IVI[95:64]**：初始化向量输入 (Initialization vector input)，位 [95:64]
有关 IVI[128:0] 位字段的说明，请参见 SAES_IVR0 寄存器。

33.7.12 SAES 初始化向量寄存器 3 (SAES_IVR3)

SAES initialization vector register 3

偏移地址：0x2C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[127:112]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVI[111:96]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IVI[127:96]**：初始化向量输入 (Initialization vector input)，位 [127:96]
有关 IVI[128:0] 位字段的说明，请参见 SAES_IVR0 寄存器。

33.7.13 SAES 密钥寄存器 4 (SAES_KEYR4)

SAES key register 4

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[159:144]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[143:128]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **KEY[159:128]**：密钥 (Cryptographic key)，位[159:128]
有关 KEY[255:0] 位字段的说明，请参见 SAES_KEYR0 寄存器。

33.7.14 SAES 密钥寄存器 5 (SAES_KEYR5)

SAES key register 5

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[191:176]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[175:160]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **KEY[191:160]**：密钥 (Cryptographic key)，位[191:160]
有关 KEY[255:0] 位字段的说明，请参见 SAES_KEYR0 寄存器。

33.7.15 SAES 密钥寄存器 6 (SAES_KEYR6)

SAES key register 6

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[223:208]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[207:192]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **KEY[223:192]**：密钥 (Cryptographic key)，位[223:192]
有关 KEY[255:0] 位字段的说明，请参见 SAES_KEYR0 寄存器。

33.7.16 SAES 密钥寄存器 7 (SAES_KEYR7)

SAES key register 7

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[255:240]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[239:224]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **KEY[255:224]**：密钥 (Cryptographic key)，位[255:224]
有关 KEY[255:0] 位字段的说明，请参见 SAES_KEYR0 寄存器。

注： *仅当选择 256 位密钥长度时，才使用密钥寄存器 4 到密钥寄存器 7。当选择 128 位密钥长度时，它们不起作用（此时仅使用密钥寄存器 0 到密钥寄存器 3）。*

33.7.17 SAES 挂起寄存器 (SAES_SUSPxR)

SAES suspend registers

偏移地址：0x040 + 0x4 * x（x = 0 到 7）

复位值：0x0000 0000

当前任务的 AES 处理被挂起以处理更高优先级的任务时，这些寄存器包含 AES 处理器的完整内部寄存器状态。

挂起后，软件在将 AES 处理器用于更高优先级的任务前，读取 SAES_SUSPxR 寄存器内容（其中 x 等于 0 到 7）并将其保存到内存中。完成后，软件在恢复原始任务前将保存的内容恢复到相应的挂起寄存器中。

注： *这些寄存器仅在选择了 CCM 链接模式时才可使用。*

仅当 SAES 使能时才能读取这些寄存器。当 SAES 禁止时读取这些寄存器将返回 0x0000 0000。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SUSP[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUSP[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **SUSP[31:0]**：SAES 挂起 (SAES suspend)

挂起操作后，相应 SAES_SUSPxR 寄存器的该位字段将从内部 SAES 寄存器之一取值。

33.7.18 SAES 中断使能寄存器 (SAES_IER)

SAES interrupt enable register

偏移地址：0x300

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNGEIE	KEIE	RWEIE	CCFIE
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3 **RNGEIE**：RNG 错误中断使能 (RNG error interrupt enable)

当 RNGEIF（RNG 错误标志）置 1 时，该位使能或禁止（屏蔽）SAES 中断生成。

0：禁止（屏蔽）

1：使能（未屏蔽）

位 2 **KEIE**：密钥错误中断使能 (Key error interrupt enable)

当 KEIF（密钥错误标志）置 1 时，该位使能或禁止（屏蔽）SAES 中断生成。

0：禁止（屏蔽）

1：使能（未屏蔽）

位 1 **RWEIE**：读取或写入错误中断使能 (Read or write error interrupt enable)

当 RWEIF（读取和/或写入错误标志）置 1 时，该位使能或禁止（屏蔽）SAES 中断生成。

0：禁止（屏蔽）

1：使能（未屏蔽）

位 0 **CCFIE**：计算完成标志中断使能 (Computation complete flag interrupt enable)
当 CCF（计算完成标志）置 1 时，该位使能或禁止（屏蔽）SAES 中断生成。
0：禁止（屏蔽）
1：使能（未屏蔽）

33.7.19 SAES 中断状态寄存器 (SAES_ISR)

SAES interrupt status register

偏移地址：0x304

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNGEIF	KEIF	RWEIF	CCF
												r	r	r	r

位 31:4 保留，必须保持复位值。

位 3 **RNGEIF**：RNG 错误中断标志 (RNG error interrupt flag)
当在 RNG 总线接口上检测到错误（例如熵错误）时，该只读位由硬件置 1。
0：RNG 总线正常运行
1：RNG 总线接口上检测到错误（随机种子获取错误）
当应用将 SAES_ICR 寄存器的相应位置 1 时，即会将 RNGEIE 位清零。如果 SAES_IER 寄存器中的 RNGEIE 位之前已置 1，将生成中断。将该位清零会触发从 RNG 外设重新加载新的随机数。

位 2 **KEIF**：密钥错误中断标志 (Key error interrupt flag)
当密钥信息加载到密钥寄存器失败或禁止使用密钥寄存器时，该只读位将由硬件置 1。
0：未检测到密钥错误
1：密钥信息加载到密钥寄存器失败，或者禁止使用密钥寄存器
如果将 SAES_ICR 寄存器的相应位置 1，会将 KEIF 清零，并且当 SAES_IER 寄存器的 KEIE 位置 1 时会生成中断。
出现以下任何错误时都会触发 KEIF：
–SAES 加载 DHUK（KEYSEL = 001 或 100）失败。
–SAES 按正确顺序加载 BHK（KEYSEL = 010 或 100）失败。
–CRYP 加载 SAES 外设共享的密钥 (KMOD = 10) 失败。
–当 KEYVALID = 1 且（KEYPROT = 1 或 KEYSEL 不为 0x0）时，如果应用程序加载密钥的安全上下文（安全或非安全）与访问 SAES_CR 或 SAES_DOUT 的安全属性不匹配。在这种情况下，KEYVALID 和 EN 位将被清零。
–SAES_KEYRx 寄存器写入未遵循正确的顺序。（KEYSIZE = 0 时，寄存器顺序为 SAES_KEYR0、SAES_KEYR1、SAES_KEYR2、SAES_KEYR3，或相反顺序。KEYSIZE = 1 时，寄存器顺序为 SAES_KEYR0、SAES_KEYR1、SAES_KEYR2、SAES_KEYR3、SAES_KEYR4、SAES_KEYR5、SAES_KEYR6、SAES_KEYR7，或相反顺序）。
KEIF 必须由应用软件清零，否则无法将 KEYVALID 置 1。

位 1 **RWEIF**：读取或写入错误中断标志 (Read or write error interrupt flag)

当 SAES_SR 寄存器中的 RDERR 或 WRERR 错误标志置 1 时，该只读位由硬件置 1。

0：未检测到读取或写入错误

1：检测到读取或写入错误（详细信息请参见 SAES_SR 寄存器）

当应用将 SAES_ICR 寄存器的相应位置 1 时，RWEIF 即会清零。如果 SAES_IER 寄存器中的 RWEIE 位之前已置 1，将生成中断。

如果选择了密钥派生模式，则该标志没有意义。

位 0 **CCF**：计算完成标志 (Computation complete flag)

该标志指示计算是否完成：

0：未完成

1：已完成

计算完成后，该标志由硬件置 1。该位在 CCF 位（在SAES_ICR 寄存器中）置 1 时由软件清零。

该标志置 1 时，如果通过 SAES_IER 寄存器的 CCFIE 位使能，则会产生中断。

仅当 DMAOUTEN 位为 0 时，该标志才有效。DMA_EN 为 1 时，该位可保持为高电平。

33.7.20 SAES 中断清零寄存器 (SAES_ICR)

SAES interrupt clear register

偏移地址：0x308

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNGEIF	KEIF	RWEIF	CCF
												w	w	w	w

位 31:4 保留，必须保持复位值。

位 3 **RNGEIF**：RNG 错误中断标志清零 (RNG error interrupt flag clear)

应用必须将该位置 1 才能将 SAES_ISR 寄存器的 RNGEIF 状态位清零。

位 2 **KEIF**：密钥错误中断标志清零 (Key error interrupt flag clear)

将该位置 1 会将 SAES_ISR 寄存器的 KEIF 状态位清零。

位 1 **RWEIF**：读取或写入错误中断标志清零 (Read or write error interrupt flag clear)

将该位置 1 会将 SAES_ISR 寄存器的 RWEIF 状态位以及 SAES_SR 寄存器中的 RDERR 和 WRERR 标志清零。

位 0 **CCF**：计算完成标志清零 (Computation complete flag clear)

将该位置 1 会将 SAES_SR 和 SAES_ISR 寄存器的 CCF 状态位清零。

33.7.21 SAES 硬件配置寄存器 (SAES_HWCFGR)

SAES hardware configuration register

偏移地址：0x3F0

复位值：0x0000 0112

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CFG3[3:0]				CFG2[3:0]				CFG1[3:0]			
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12 保留，必须保持为零。

位 11:8 **CFG3[3:0]**：HW 通用 3 (HW Generic 3)
该位字段返回 0x1。

位 7:4 **CFG2[3:0]**：HW 通用 2 (HW Generic 2)
该位字段返回 0x1。

位 3:0 **CFG1[3:0]**：HW 通用 1 (HW Generic 1)
该位字段返回 0x2。

33.7.22 SAES 版本寄存器 (SAES_VERR)

SAES version register

偏移地址：0x3F4

复位值：0x0000 0030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持为零

位 7:4 **MAJREV[3:0]**：主版本 (Major revision)
该位字段返回 SAES 外设主版本 (0x3)。

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)
该位字段返回 SAES 外设次版本 (0x0)。

33.7.23 SAES 标识寄存器 (SAES_IPIDR)

SAES identification register

偏移地址：0x3F8

复位值：0x0017 0023

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：外设的标识码 (Identification code of the peripheral)
该位字段返回 SAES 外设的标识码。

33.7.24 SAES 尺寸 ID 寄存器 (SAES_SIDR)

SAES size ID register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识码 (size identification code)
该位字段返回 SAES 外设的大小标识码，定义如下：
位[31:8] = 0xA3C5DD（固定代码）
位[7:0] = 0x01（1 KB 地址解码）

33.7.25 SAES 寄存器映射

表 281. SAES 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	SAES_CR	IPRST	KEYSEL[2:0]			KSHAREID[1:0]		KMOD[1:0]		NPBLB[3:0]				KEYPROT	KEYSIZE	Res.	CHMOD[2]	Res.	GCMPH[1:0]		DMAOUTEN	DMAINEN	Res.	Res.	Res.	Res.	CHMOD[1:0]		MODE[1:0]		DATATYPE[1:0]		EN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0		0	0	0	0					0	0	0	0	0	0	0

表 281. SAES 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x004	SAES_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEYVALID	Res.	Res.	Res.	BUSY	WRERR	RDERR	CCF
	Reset value																									0				0	0	0	0
0x008	SAES_DINR	DIN[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	SAES_DOUTR	DOUT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	SAES_KEYR0	KEY[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014	SAES_KEYR1	KEY[63:32]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	SAES_KEYR2	KEY[95:64]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x01C	SAES_KEYR3	KEY[127:96]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	SAES_IVR0	IVI[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	SAES_IVR1	IVI[63:32]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x028	SAES_IVR2	IVI[95:64]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x02C	SAES_IVR3	IVI[127:96]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x030	SAES_KEYR4	KEY[159:128]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x034	SAES_KEYR5	KEY[191:160]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x038	SAES_KEYR6	KEY[223:192]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03C	SAES_KEYR7	KEY[255:224]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x040	SAES_SUSP0R	SUSP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x044	SAES_SUSP1R	SUSP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x048	SAES_SUSP2R	SUSP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 281. SAES 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x04C	SAES_SUSP3R	SUSP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x050	SAES_SUSP4R	SUSP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x054	SAES_SUSP5R	SUSP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x058	SAES_SUSP6R	SUSP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x05C	SAES_SUSP7R	SUSP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x060-0x2FF	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x300	SAES_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNGEIE	KEIE	RWEIE	CCFIE
	Reset value																													0	0	0	0
0x304	SAES_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNGEIF	KEIF	RWEIF	CCF
	Reset value																													0	0	0	0
0x308	SAES_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNGEIF	KEIF	RWEIF	CCF
	Reset value																													0	0	0	0
0x3F0	SAES_HWCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFG3[3:0]				CFG2[3:0]				CFG1[3:0]			
	Reset value																					0	0	0	1	0	0	0	1	0	0	1	0
0x3F4	SAES_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	0	1	1	0	0	0	0
0x3F8	SAES_IPIDR	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
0x3FC	SAES_SIDR	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

34 密码处理器 (CRYP)

34.1 简介

借助密码处理器 (CRYP)，可使用 DES、Triple-DES (3DES) 或 AES 算法对数据进行加密和解密。密码处理器完全兼容下列标准：

- 联邦信息处理标准出版物（FIPS PUB 46-3，1999 年 10 月）以及美国国家标准协会 (ANSI X9.52) 规定的数据加密标准 (DES) 和 3DES (TDES)。
- 联邦信息处理标准出版物（FIPS PUB 197，2001 年 11 月）规定的高级加密标准 (AES)

支持多种密钥大小和链接模式：

- DES/TDES 链接模式 ECB 和 CBC，支持标准 56 位密钥（每个密钥带 8 位奇偶校验）
- AES 链接模式 ECB、CBC、CTR、GCM、GMAC、CCM，支持 128、192 或 256 位密钥大小

CRYP 处理器为 32 位 AHB 外设。它支持对传入和传出数据进行 DMA 传输（需要两个 DMA 通道）。该外设还包含输入和输出 FIFO（分别为 8 个字的深度），有助于实现更佳性能。

在 AES 模式下，CRYP 处理器支持与 SAES 协处理器共享密钥。

34.2 CRYP 主要特性

- 兼容下列标准：
  - NIST *FIPS 出版物 46-3，数据加密标准 (DES)*
  - ANSI *X9.52，三重数据加密算法工作模式*
  - NIST *FIPS 出版物 197，高级加密标准 (AES)*
- AES 对称块密码实现
  - 128 位数据块处理
  - 支持 128 位、192 位和 256 位密钥长度
  - 基于多种链接模式的加密和解密：电子密码本 (ECB)、密码块链接 (CBC)、计数器模式 (CTR)、Galois 计数器模式 (GCM)、Galois 消息认证码模式 (GMAC) 和 CBC-MAC 计数器模式 (CCM)
  - 14（或 18）个时钟周期，用于在 AES-ECB 模式下处理一个包含 128 位（或 256 位）密钥的 128 位数据块
  - 集成的密钥调度器，并包含密钥派生（仅限 ECB 或 CBC 解密）
  - 在 AES 模式下，SAES 协处理器与 CRYP 共享密钥
- DES/TDES 加密/解密实现
  - 64 位数据块处理
  - 支持 64 位、128 位和 192 位密钥长度（包括奇偶校验）
  - 基于 ECB 和 CBC 链接模式的加密和解密
  - 直接执行简单 DES 算法（使用单一密钥 K1）
  - 16（或 48）个时钟周期，用于在 DES（或 TDES）ECB 模式下处理一个 64 位数据块
  - 密文窃取的软件实现

- DES/TDES 和 AES 的共同特性
  - AMBA AHB 从外设，仅支持 32 位字单次访问（否则，会生成 AHB 总线错误，并会忽略写访问）
  - 256 位寄存器，用于存储密钥（8 个 32 位寄存器）
  - 128 位寄存器，用于存储初始化向量（4 个 32 位寄存器）
  - 1 个 32 位输入缓冲器，它与八个 32 位字的内部 IN FIFO 相关联，对应于四个传入 DES 块或两个 AES 块
  - 1 个 32 位输出缓冲器，它与八个 32 位字的内部 OUT FIFO 相关联，对应于四个已处理的 DES 块或两个 AES 块
  - 采用自动数据流控制，支持直接内存访问 (DMA)（使用两个通道，分别用于传入数据和已处理数据）。OUT FIFO 支持单次传输和突发传输，而 IN FIFO 仅支持突发传输。
  - 采用数据交换逻辑，支持 1 位、8 位、16 位或 32 位数据
  - 在密码处理器需要处理优先级更高的消息时，可通过软件将当前消息挂起（挂起/恢复操作）

34.3 CRYP 实现

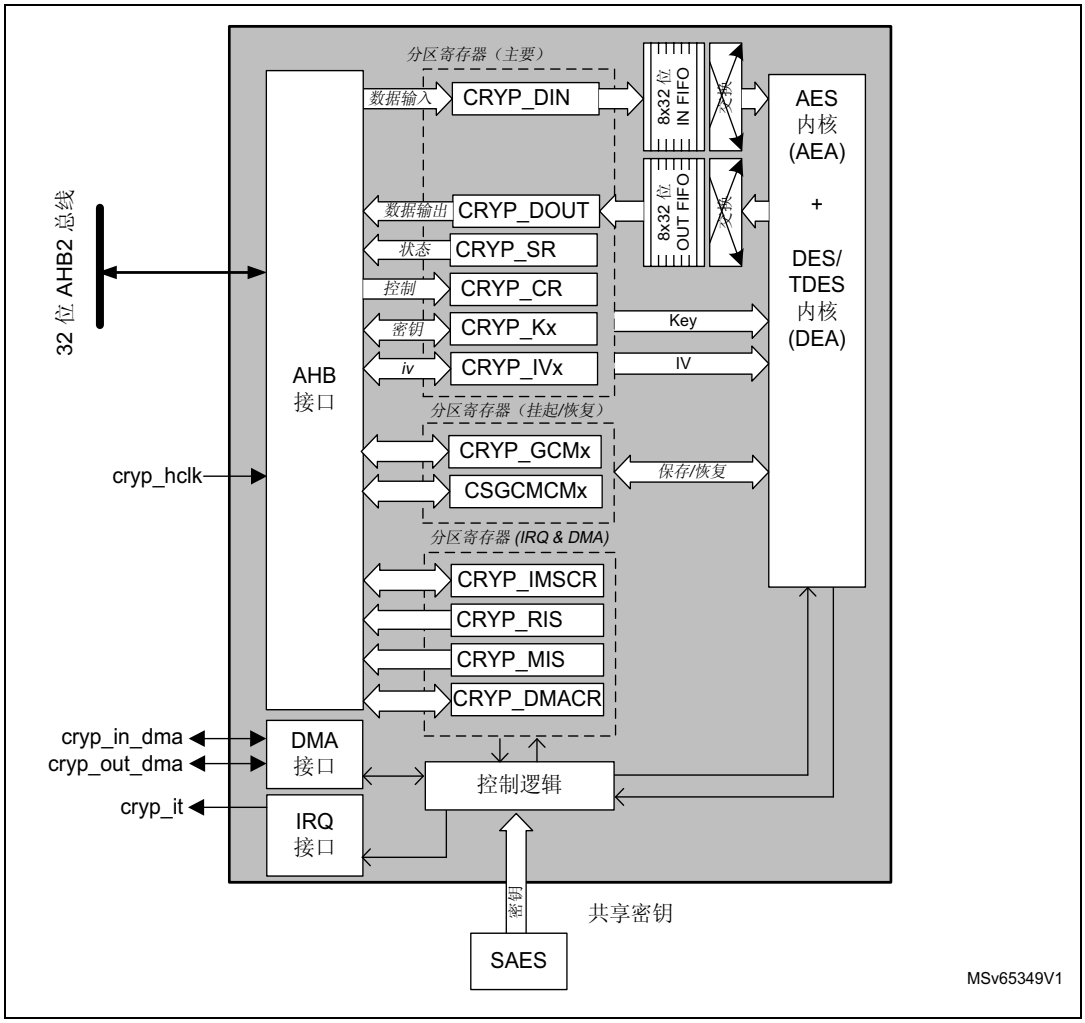
该设备有一个 CRYP 实例连接到一个 SAES 外设（共享密钥）和一个 MDMA 外设。

34.4 CRYP 功能描述

34.4.1 CRYP 框图

下图显示了密码处理器的框图。

图 305. CRYP 框图



34.4.2 CRYP 内部信号

表 282 列出了密码处理器级而非 STM32 产品级（焊盘上）所提供的内部信号，对这些信号有所了解会很有用。

表 282. CRYP 内部输入/输出信号

信号名称	信号类型	说明
cryp_hclk	数字输入	AHB 总线时钟
cryp_it	数字输出	密码处理器全局中断请求
cryp_in_dma	数字输入/输出	IN FIFO DMA 突发请求/确认
cryp_out_dma	数字输入/输出	OUT FIFO DMA 突发请求/确认

34.4.3 CRYP DES/TDES 密码内核

概述

DES/Triple-DES (3DES) 密码内核由三部分组成：

- DES 算法（DEA 内核）
- 多个密钥（DES 算法使用一个密钥，TDES 算法使用一到三个密钥）
- 初始化向量（仅在 CBC 模式中使用）

DES/Triple-DES 密码内核支持两种工作模式：

- **ALGODIR = 0**：使用存储在 CRYP_Kx 寄存器中的密钥实现明文加密。
- **ALGODIR = 1**：使用存储在 CRYP_Kx 寄存器中的密钥实现密文解密。

通过对 CRYP_CR 寄存器中的 ALGODIR 位进行编程来选择工作模式。

典型数据处理

*第 34.4.10 节：CRYP DES/TDES 基本链接模式（ECB 和 CBC）*对 DES 模式下密码处理器的典型用法进行了介绍。

注： *中间 DEA 阶段的输出始终不会在密码边界外暴露，CBC 模式下的 IV 寄存器除外。*

DES/TDES 密钥和链接模式

DES/TDES 允许使用三种不同的密钥选项：

- *三个独立的密钥*

  第一个选项指定所有的密钥均是独立的，即 K1、K2 和 K3 均是独立的。FIPS PUB 46-3 – 1999（以及 ANSI X9.52 – 1998）将该选项称为“密钥选项 1”，并将这种 TDES 称为“3 密钥 TDES”。

- *两个独立的密钥*

  第二个选项指定 K1 和 K2 是独立的，而 K3 等于 K1，即 K1 和 K2 独立，而 K3 = K1。FIPS PUB 46-3 – 1999（以及 ANSI X9.52 – 1998）将该选项称为“密钥选项 2”，并将这种 TDES 称为“2 密钥 TDES”。

- *三个相同的密钥*

  第三个选项指定 K1、K2 和 K3 相等，即：

  K1 = K2 = K3

  FIPS PUB 46-3 – 1999（以及 ANSI X9.52 – 1998）将第三个选项称为“密钥选项 3”。这种“1 密钥”TDES 与单独 DES 相同。

以下链接算法受 DES 硬件支持，可通过 CRYP_CR 寄存器中的 ALGOMODE 位进行选择：

- 电子密码本 (ECB)
- 密码块链接 (CBC)

*第 34.4.10 节：CRYP DES/TDES 基本链接模式（ECB 和 CBC）*详细介绍了这些模式。

34.4.4 CRYP AES 密码内核

概述

AES 密码内核由以下部分组成：

- AES 算法（AEA 内核）
- 基于二元 Galois 域的乘法器 (GF2mul)
- 密钥信息
- 初始化向量 (IV) 或随机值信息
- 链接算法逻辑（XOR、反馈/计数器、屏蔽）

AES 内核适用于密钥长度为 128 位、192 位或 256 位的 128 位数据块（四字）。根据链接模式的不同，外设可能需要一个 128 位初始化向量 (IV)，也可能不需要。

密码外设支持以下两种工作模式：

- **ALGODIR = 0**：使用存储在 CRYP_Kx 寄存器中的密钥实现明文加密。
- **ALGODIR = 1**：使用存储在 CRYP_Kx 寄存器中的密钥实现密文解密。选择 ECB 和 CBC 链接模式时，密码外设会自动执行初始密钥生成过程。

通过对 CRYP_CR 寄存器中的 ALGODIR 位进行编程来选择工作模式。

典型数据处理

*第 34.4.11 节：CRYP AES 基本链接模式（ECB 和 CBC）*介绍了 AES 模式下密码处理器的典型用法。

注： *中间 AEA 阶段的输出始终不会在密码边界外显示，IV 寄存器除外。*

AES 链接模式

以下链接算法受密码处理器支持，可通过 CRYP_CR 寄存器中的 ALGOMODE 位进行选择：

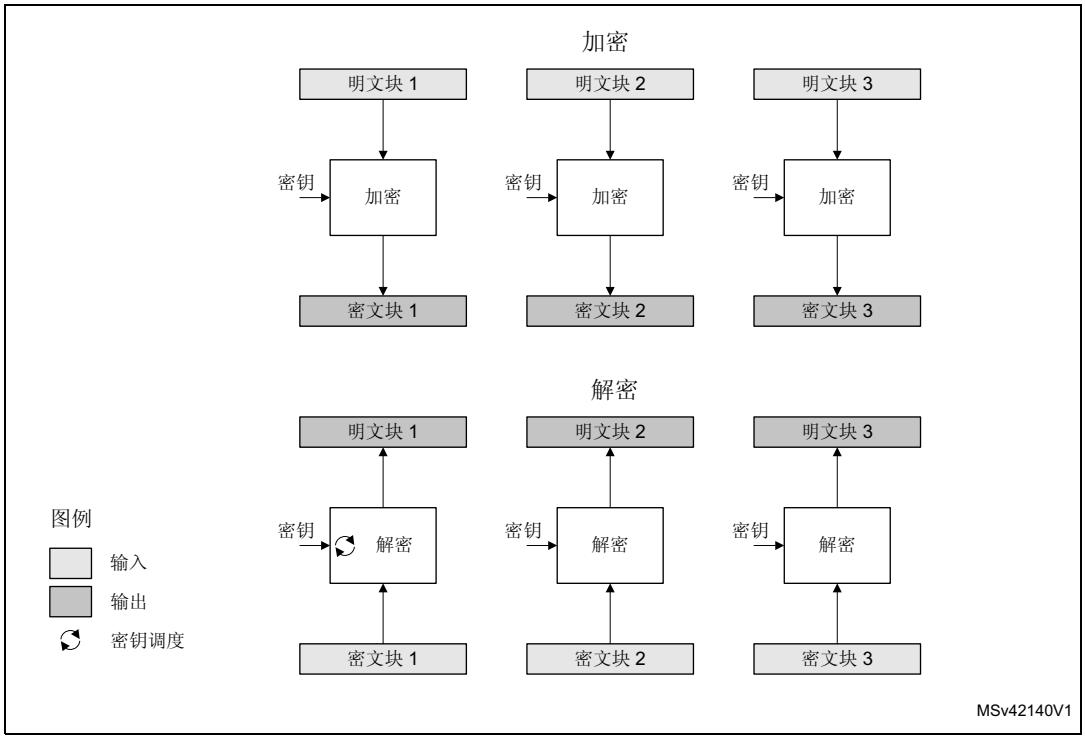
- 电子密码本 (ECB)
- 密码块链接 (CBC)
- 计数器模式 (CTR)
- Galois/计数器模式 (GCM)
- Galois 消息认证码模式 (GMAC)
- CBC-MAC 计数器模式 (CCM)

后续子章节对这些链接模式进行了简单介绍。

有关详细说明，请参见*第 34.4.11 节：CRYP AES 基本链接模式（ECB 和 CBC）*以及之前的章节。

AES 电子密码本 (ECB)

图 306. AES-ECB 模式概览

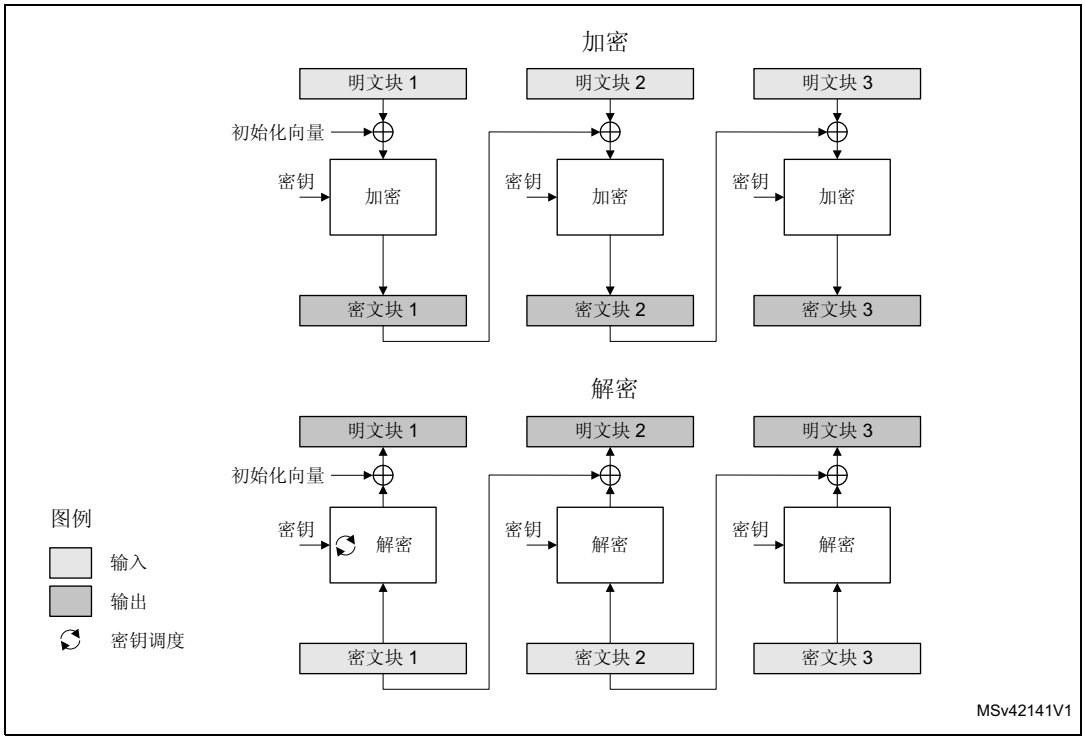


ECB 是最简单的工作模式。无链接操作，且无特殊初始化阶段。消息会划分到各个块中，并对各个块分别进行加密或解密。

注： *对于解密，在处理第一个块之前需要特殊的密钥调度。*

AES 密码块链接 (CBC)

图 307. AES-CBC 模式概览

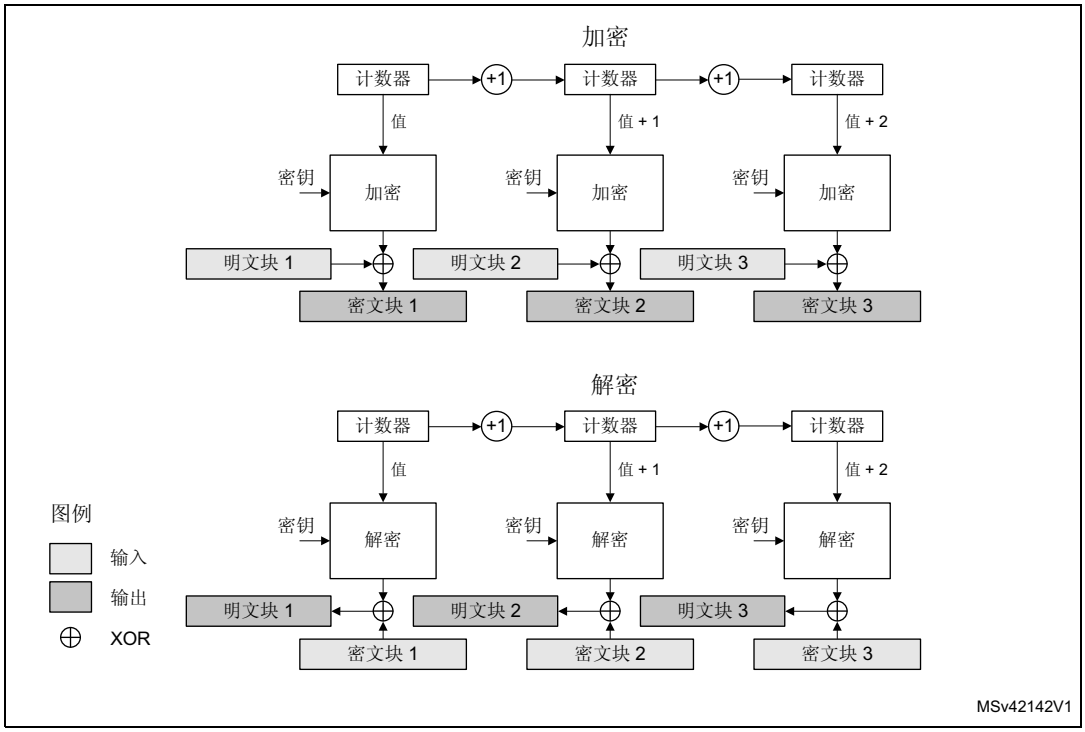


在 CBC 工作模式下，每个块的输出均与下一个块的输入相连。为了确保每条消息都是唯一的，会在第一个块处理过程中使用初始化向量。

注： *对于解密，在处理第一个块之前需要特殊的密钥调度。*

AES 计数器模式 (CTR)

图 308. AES-CTR 模式概览

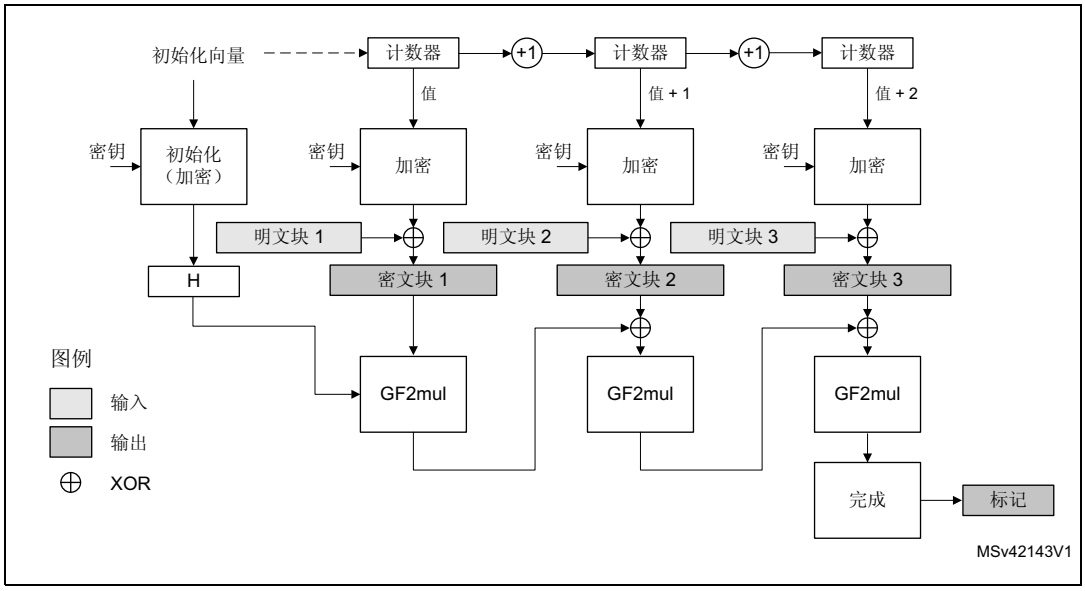


CTR 模式使用 AES 内核生成密钥流；这些密钥将与明文进行异或运算，以获得密文（如 NIST *特别出版物 800-38A，块密码工作模式的建议*中所述）。

注： *与 ECB 和 CBC 模式不同，CTR 解密无需密钥调度，因为在该链接方案中，AES 内核始终在加密模式下用于生成计数器块。*

AES Galois/计数器模式 (GCM)

图 309. AES-GCM 模式概览

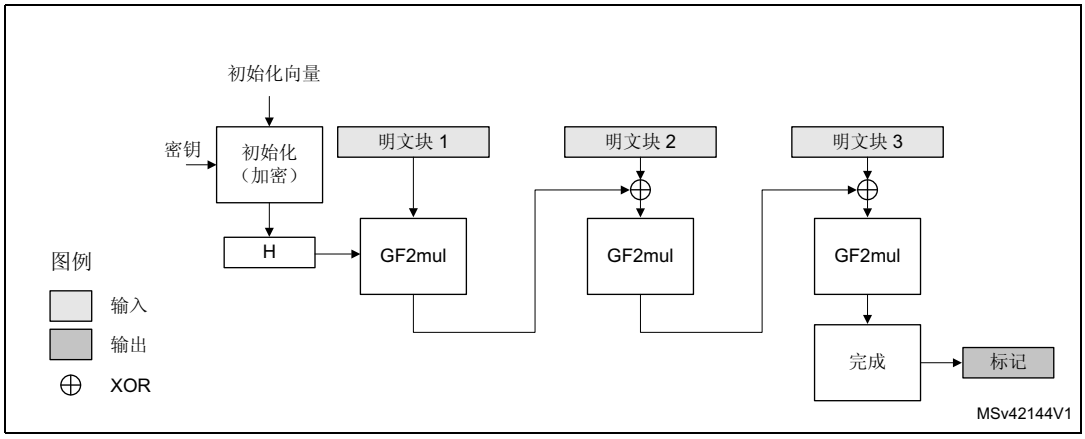


在 Galois/计数器模式 (GCM) 下，将对明文消息进行加密，同时会对消息认证码 (MAC) 进行计算，从而生成相应的密文及其 MAC（也称为认证标记）。NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/计数器模式 (GCM) 和 GMAC* 中对此进行了相关定义。

GCM 模式基于 AES 计数器模式，可实现保密性。它使用固定有限域乘法器来计算消息认证码。消息末尾处需要一个初始值和一个特定的 128 位块。

AES Galois 消息认证代码 (GMAC)

图 310. AES-GMAC 模式概览

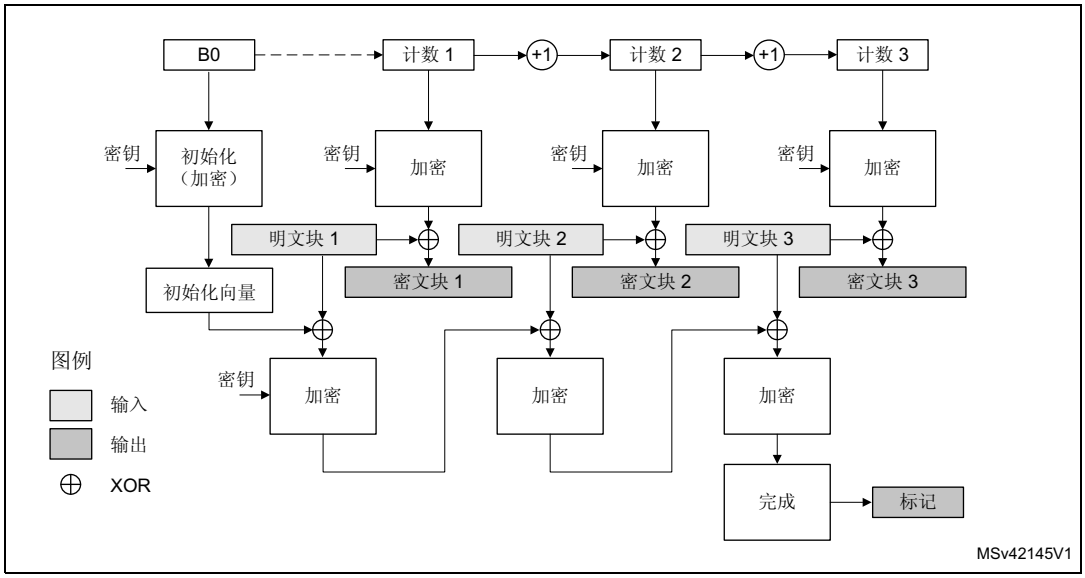


Galois 消息认证码 (GMAC) 允许认证消息并生成相应的消息认证码 (MAC)。NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/计数器模式 (GCM) 和 GMAC* 中对此进行了相关定义。

GMAC 与 Galois/计数器模式 (GCM) 类似，只不过它应用于仅由明文认证数据组成的消息（即只有标头，无有效负载）。

AES CBC-MAC 计数器模式 (CCM)

图 311. AES-CCM 模式概览



在密码块链接-消息认证码计数器模式 (CCM) 下，将对明文消息进行加密，同时会对消息认证码 (MAC) 进行计算，从而生成相应的密文以及相应的 MAC（也称为标记）。NIST *特别出版物 800-38C，块密码工作模式的建议 - CCM 模式实现认证和保密性*中对此进行了介绍。

CCM 模式基于 AES 计数器模式，可实现保密性，并且使用 CBC 来计算消息认证码。它需要一个初始值。

与 GCM CCM 链接模式一样，AES-CCM 模式可应用于仅由明文认证数据组成的消息（即只有标头，无有效负载）。请注意，这种 CCM 使用方式不被称为 CMAC（它并非类似于 GCM/GMAC），NIST 不推荐这种用途。

34.4.5 执行加密操作的 CRYP 过程

简介

为了理解密码外设的工作原理，下面对典型密码操作进行了介绍。有关不同密码模式下外设使用方法的详细信息，请参见具体章节，例如*第 34.4.11 节：CRYP AES 基本链接模式（ECB 和 CBC）*。

CRYP 初始化

若要初始化密码处理器，首先应通过将 CRYP_CR 寄存器中的 CRYPEN 位清零来禁止密码处理器。然后执行以下步骤：

1. 通过 CRYP_CR 寄存器中的 ALGOMODE 和 ALGODIR 位配置算法和链接模式。另外，使用 KEYSIZE 位配置密钥大小。
   - 通过将 KMOD[1:0] 设置为 00 选择正常密钥模式。如果选择了 AES 算法，并且密钥不是由用户应用程序写入而必须由 SAES 传入，请参见*第 34.4.16 节*。
   - 当 ALGODIR 设置为 1（解密）并且选定的算法/链接模式为 AES-ECB 或 AES-CBC 时，必须执行加密密钥的初始密钥派生，如*第 34.4.7 节：为解密准备 CRYP AES 密钥*所述。
2. 完成前一步骤后，在适用的情况下将对称密钥写入 CRYP_KxL/R 寄存器。*第 34.4.18 节：CRYP 密钥寄存器*定义了写入密钥寄存器的方法（仅适用于 2.2 以上的 CRYP 版本）。
3. 通过 CRYP_CR 寄存器的 DATATYPE 位配置数据类型（1 位、8 位、16 位或 32 位）。
4. 必要时（例如，CBC 或 CTR 链接模式），向 CRYP_IVx(L/R)R 寄存器中写入初始化向量。
5. 向 CRYP_CR 寄存器中的 FFLUSH 位写入 1，刷新 IN 和 OUT FIFO。

注： *如上所述，必须在写入密钥之前写入 CRYP_CR 中的 KEYSIZE 和 ALGOMODE。*

必须在写入 ALGOMODE 的同时或之后写入 KMOD = 10（共享密钥），否则如果选择了 DES 或 TDES（例如 ALGOMODE = 0x0），将触发 KERF 错误。

针对所有情况的初始警告

如果选择 ECB 或 CBC 模式，且数据不是 64 位（对于 DES）或 128 位（对于 AES）的倍数，则第二块和最后一块的数据管理要比下述步骤更为复杂。更多详细信息，请参见*第 34.4.8 节：CRYP 窃取和数据填充*。

在轮询模式下使用 CPU 附加数据

1. 必须将 CRYP_SR 寄存器中的 KEYVALID 置 1。将 CRYP_CR 寄存器中的 CRYPEN 位置 1，以使能密码处理器。
2. 向 IN FIFO 中写入数据（一个块或直到 FIFO 已满）。
3. 重复以下步骤，直到处理完数据的倒数第二个块：
   a) 等待至非空标志 OFNE 置 1，然后读取 OUT FIFO（一个块或直到 FIFO 为空）。
   b) 等待至非满标志 IFNF 置 1，然后写入 IN FIFO（一个块或直到 FIFO 已满），但是最后一个块的情况除外。
4. 密码处理器自动将 BUSY 位置 1。处理结束时，BUSY 位会返回至 0，两个 FIFO 均为空（IN FIFO 空标志 IFEM = 1 且 OUT FIFO 非空标志 OFNE = 0）。
5. 如果下一个处理块为最后一个块，则 CPU 必须用零填充（如适用）数据以获得完整块，并且在 AES GCM 有效负载加密或 AES CCM 有效负载解密的情况下，使用 CRYP_CR 寄存器中的 NPBLB 位指定无效字节的数量（否则标签计算将会出现错误）。必须在确定 CRYP_CR 寄存器中的 BUSY 位设置为 0 之后再执行该操作。

注： *NPBLB 位不用于 AES GCM、GMAC 和 CCM 模式的标头阶段。*

6. 操作完成后，可通过将 CRYP_CR 寄存器中的 CRYPEN 位清零来关闭密码处理器。

在中断模式下使用 CPU 附加数据

1. 将 CRYP_IMSCR 寄存器中的 INIM 和 OUTIM 位置 1，以使能中断。
2. 将 CRYP_CR 寄存器中的 CRYPEN 位置 1，以使能密码处理器。
3. 在管理输入数据的中断服务程序中：
   a) 如果正在加载的是最后一个块，则 CPU 必须用零填充（如适用）数据以获得完整块，并且在 AES GCM 有效负载加密或 AES CCM 有效负载解密的情况下，使用 CRYP_CR 寄存器中的 NPBLB 位指定无效字节的数量（否则标签计算将会出现错误）。必须在确定 CRYP_CR 寄存器中的 BUSY 位设置为 0 之后再执行该操作。然后将该块加载到 IN FIFO 中。

注： *NPBLB 位不用于 AES GCM、GMAC 和 CCM 的标头阶段。*

   b) 如果不是最后一个块，则将数据加载到 IN FIFO 中。只能加载一个块（对于DES 为 2 个字，对于 AES 为 4 个字），或加载数据直至 FIFO 已满。
   c) 在所有情况下，写入数据的最后一个字后，通过清除 INIM 中断屏蔽禁止中断。
4. 在管理输入数据的中断服务程序中：
   a) 从 OUT FIFO 中读取输出数据。只能读取一个块（对于 DES 为 2 个字，对于 AES 为 4 个字），或读取数据直至 FIFO 为空。
   b) 读取最后一个字后，INIM 和 BUSY 位均设为 0，两个 FIFO 均为空（IFEM = 1 且 OFNE = 0）。将 OUTIM 位清零可禁止中断，而将 CRYPEN 位清零可禁止外设。
   c) 如果读取明文数据的最后一个块（即解密），则可选择丢弃不属于消息 / 有效负载一部分的数据。

使用 DMA 附加数据

1. 可选择用零进行填充的方式来准备最后一个数据块，以获得完整块。
2. 将 DMA 控制器配置为传输来自内存的输入数据以及将来自外设的输出数据传输到内存，如*第 34.4.20 节：CRYP DMA 接口*所述。应将 DMA 配置为在完成传输时设置一个中断，以指示处理过程已完成。在 AES GCM 有效负载加密或 AES CCM 有效负载解密情况下，DMA 传输不得包括最后一个块。对于最后一个块，必须改用上述使用 CPU 的流程，因为在处理该块之前需要设置 NPBLB 位（否则标签计算将会出现错误）。

注： *NPBLB 位不用于 AES GCM、GMAC 和 CCM 的标头阶段。*

3. 通过将 CRYP_CR 寄存器中的 CRYPEN 位置 1 使能密码处理器，然后通过将 CRYP_DMACR 寄存器中的 DIEN 和 DOEN 位置 1 来使能 DMA IN 和 OUT 请求。
4. 所有传输和处理过程均由 DMA 和密码处理器管理。DMA 中断表示处理过程已完成。两个 FIFO 通常均为空，且 BUSY 标志设为 0。

注： *当选择 DES/TDES 算法时，无法使用 DMA。*

小心： 在填满密码处理器输入 FIFO 之前，由 DMA 控制器清空密码处理器输出 FIFO 至关重要。为此，应对 DMA 控制器进行相应配置，以使从密码外设到内存的传输优先级高于从内存到密码外设的传输。

34.4.6 CRYP 繁忙状态

满足以下所有条件时，即表示密码处理器繁忙，正在处理数据（CRYP_SR 寄存器中的 BUSY 设为 1）：

- CRYP_CR 寄存器中的 CRYPEN = 1。
- 输入 FIFO 中有足够的数据（对于 DES 或 TDES 算法模式至少有两个字，对于 AES 算法模式至少有四个字）。
- 输出 FIFO 中有足够的可用空间（对于 DES 至少有两个字单元，对于 AES 至少有四个字单元）。

当密码处理器繁忙时，会忽略针对 CRYP_Kx(L/R)R 密钥寄存器、CRYP_IVx(L/R)R 初始化寄存器或 CRYP_CR 寄存器的位 [9:2] 的写操作（即，不会修改这些寄存器）。因此，不能在密码处理器处理数据块时修改其配置。

在 BUSY 位置 1 时，可将 CRYPEN 位清零。在这种情况下，在硬件清零 BUSY 位之前，会先完成正在进行的 DES/TDES 或 AES 处理过程（即，字结果被写入输出 FIFO）。

注： *如果应用程序需要挂起一条消息，以处理另一条优先级更高的消息，请参见第 34.4.9 节：CRYP 挂起/恢复操作*

在 DES 或 TDES 模式下处理某个块时，如果输出 FIFO 已满并且输入 FIFO 至少包含一个新块，则输入 FIFO 会弹出新块且 BUSY 位保持置 1，直到有足够的空间可将这个新块存储到输出 FIFO。

34.4.7 为解密准备 CRYP AES 密钥

执行 AES **ECB 或 CBC 解密**时，需要预先准备 AES 密钥。实际上，在 AES 加密中，第 0 轮密钥是存储在密钥寄存器中的密钥，而 AES 解密必须使用最后一轮密钥开始。由于加密密钥存储在内存中，因此必须执行特殊的密钥调度，以获得解密密钥。在除 ECB 或 CBC 解密外的所有其他 AES 模式下，均无需此准备过程。

当禁止密码处理器 (CRYPEN = 0) 时，按照下列步骤执行 CRYP 密钥准备过程：

1. 将 CRYP_CR 中的 ALGOMODE 位编程为 0x7，ALGODIR 位编程为 0x0。另外，使用 KEYSIZE 位配置密钥长度。KMOD 也必须设为 00。对于使用 KMOD[1:0] = 10（共享密钥）的解密，请参见*第 34.4.16 节*。
2. 将对称密钥写入 CRYP_KxL/R 寄存器，如*第 34.4.18 节：CRYP 密钥寄存器*所述（如果适用）。
3. 将 CRYP_CR 寄存器中的 CRYPEN 位置 1，以使能密码处理器。这会立即启动 AES 轮来准备密钥 (BUSY = 1)。
4. 等待直到 CRYP_SR 寄存器中的 BUSY 位清零。然后更新 CRYP_CR 寄存器中的 ALGOMODE 位以选择正确的链接模式，即对于 ECB 为 0x4 或对于 CBC 为 0x5。
5. AES 密钥位于 CRYP 密钥寄存器中，随时可用于解密。

注： *由于 CRYPEN 位字段在密钥准备结束后由硬件复位，因此应用软件必须再次将其置 1 以进行下一次操作。*

密钥准备操作的延迟为 14、16 或 18 个时钟周期，具体取决于密钥大小（128、192 或 256 位）。

34.4.8 CRYP 窃取和数据填充

在 **ECB** 或 **CBC** 模式下使用 DES 或 AES 算法来管理不是块大小（对于 DES 为 64 位，对于 AES 为 128 位）倍数的消息时，请使用 NIST *特别出版物 800-38A，块密码工作模式的建议：CBC 模式密文窃取的三种变型*中介绍的窃取技术。由于密码处理器未实现这类技术，因此**最后两个块**必须由应用程序以特殊方式处理。

注： *本参考手册中未对密文窃取技术进行介绍。*

类似地，在除 ECB 或 CBC 之外的模式下使用 AES 算法时，应用程序必须先用零填充不完整的输入数据块（即，短于 128 位的块）后再进行加密（即，应在数据串的尾端附加额外的位）。解密后需要丢弃额外填充的位。密码处理器不会对**最后一个块**执行自动数据填充操作，因此应用程序应遵循*第 34.4.5 节：执行加密操作的 CRYP 过程*中给出的建议来管理不是 128 位倍数的消息。

注： *填充数据根据 CRYP_CR 寄存器中的 DATATYPE 位字段，以类似于正常数据的方式进行交换（有关详细信息，请参见第 34.4.17 节：CRYP 数据寄存器和数据交换）。*

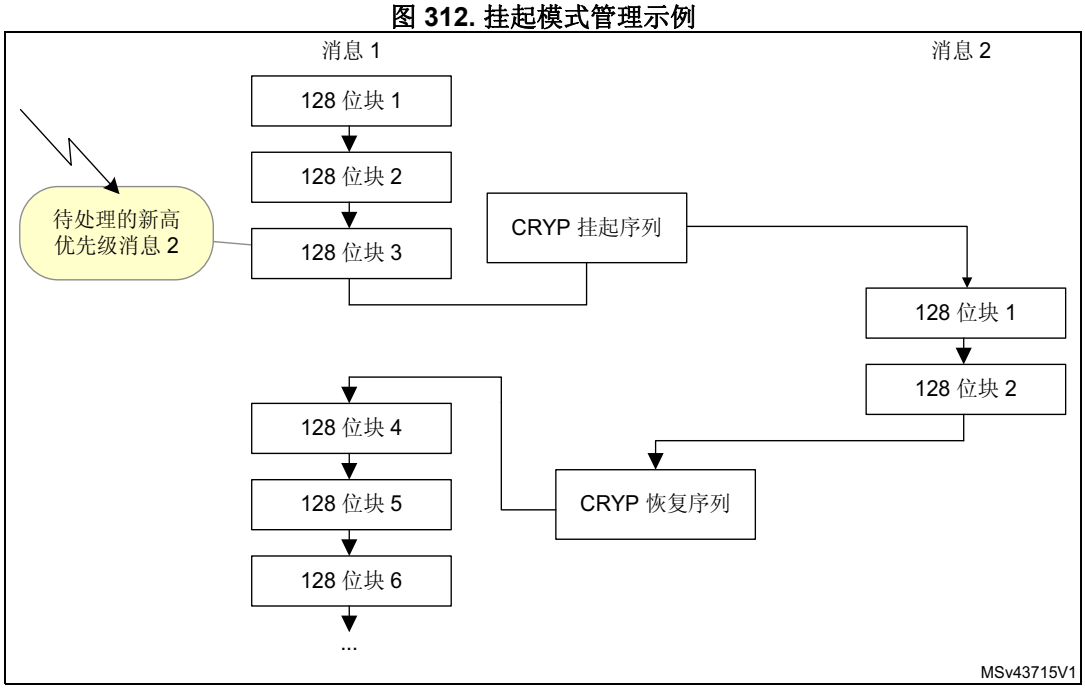
34.4.9 CRYP 挂起/恢复操作

如果需要处理另一条优先级较高的消息，则可将消息挂起。该最高优先级消息完成发送后，可在加密或解密模式下恢复挂起的消息。

挂起/恢复操作不会破坏链接运算，并且再次使能密码处理器以接收下一个数据块时，可立即恢复消息处理。

图 312 所示为挂起/恢复操作示例：为发送优先级更高、长度更短的消息 2 而将消息 1 挂起（AES 算法）。

图 312. 挂起模式管理示例



各 AES 模式部分对挂起/恢复操作进行了详细说明。

34.4.10 CRYP DES/TDES 基本链接模式（ECB 和 CBC）

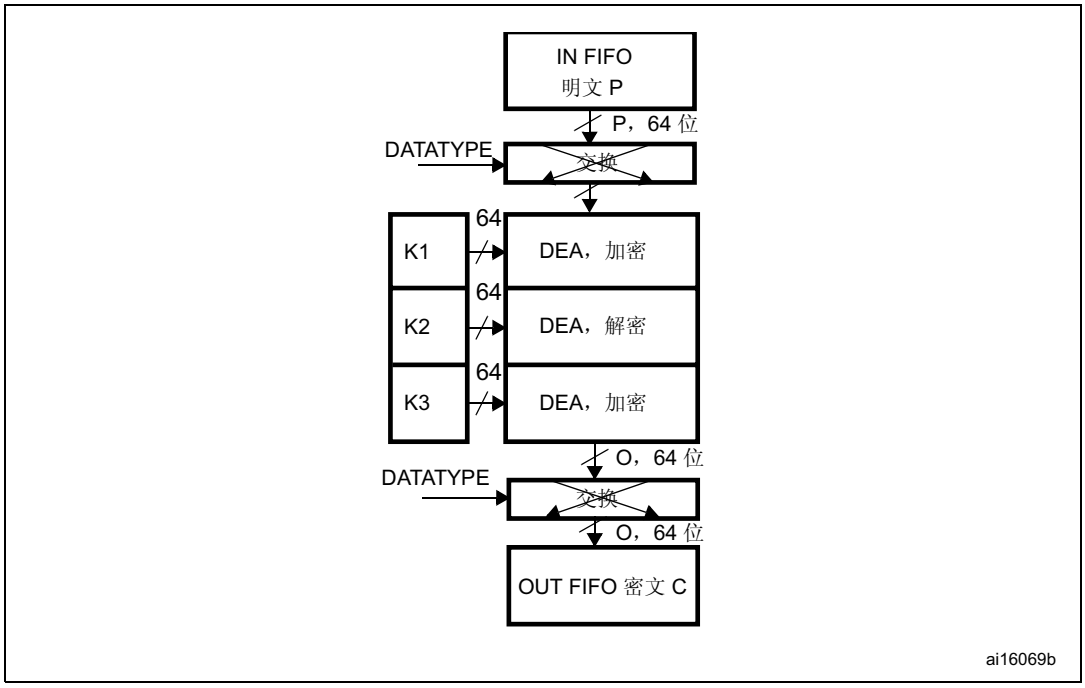
概述

FIPS PUB 46-3 – 1999（以及 ANSI X9.52-1998）对 DES 计算内核所提供的四种操作模式中涉及的处理过程进行了详尽的解释，这四种模式分别为：TDES-ECB 加密、TDES-ECB 解密、TDES-CBC 加密和 TDES-CBC 解密。本部分仅对每种模式进行了简要解释。

DES/TDES-ECB 加密

图 313 介绍了 DES 和 TDES 电子密码本 (DES/TDES-ECB) 模式中的加密。通过将 CRYP_CR 中的 ALGOMODE 和 ALGODIR 分别编程为 0x0 和 0 来选择该模式。

图 313. DES/TDES-ECB 模式加密



1. K：密钥；C：密文；I：输入块；O：输出块；P：明文。

64 位明文数据块 (P) 经过位/字节/半字交换后用作输入块 (I)。输入块通过 DEA 在加密状态下使用 K1 进行加密处理。上述处理过程的输出会直接反馈到 DEA 的输入，在解密状态下使用 K2 执行 DES。上述处理过程的输出会直接反馈到 DEA 的输入，在加密状态下使用 K3 执行 DES。生成的 64 位输出块 (O) 在执行位/字节/半字交换之后，以密文 (C) 形式推入 OUT FIFO。

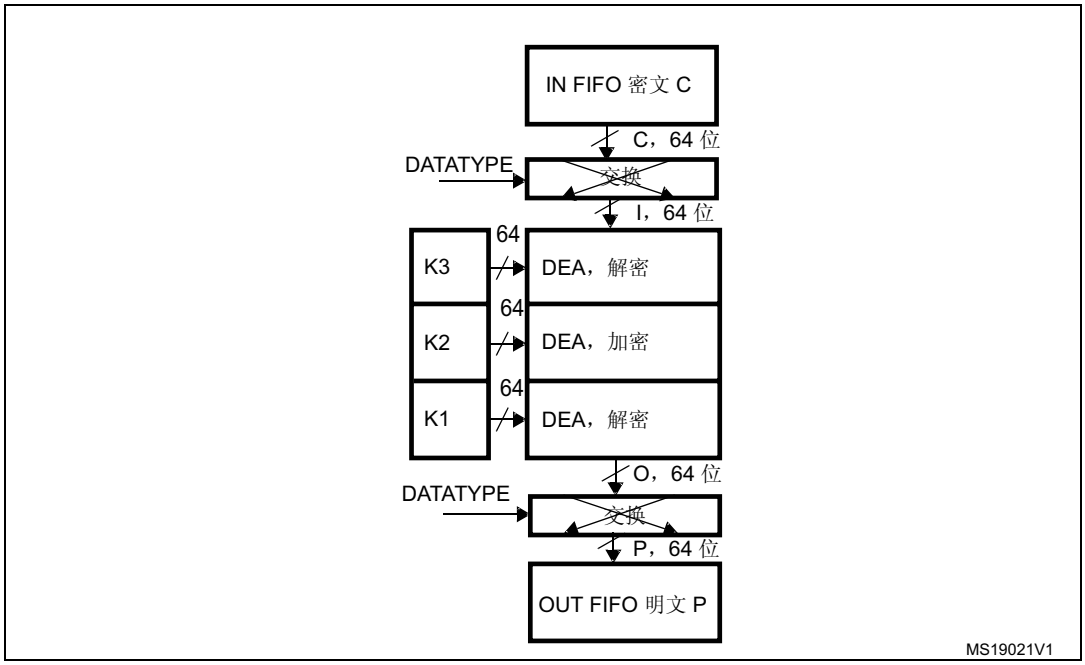
注： *有关数据交换的更多信息，请参见第 34.4.17 节：CRYP 数据寄存器和数据交换。*

有关详细的 DES/TDES 加密序列，请参见第 34.4.5 节：执行加密操作的 CRYP 过程。

DES/TDES-ECB 模式解密

图 314 介绍了 DES 和 TDES 电子密码本 (DES/TDES-ECB) 模式中的解密。通过将 CRYP_CR 中的 ALGOMODE 和 ALGODIR 分别编程为 0x0 和 1 来选择该模式。

图 314. DES/TDES-ECB 模式解密



1. K：密钥；C：密文；I：输入块；O：输出块；P：明文。

64 位官方块 (C) 经过位/字节/半字交换后，作为输入块 (I)。该密钥序列与加密过程中使用的密钥序列相反。输入块通过 DEA 在解密状态下使用 K3 进行解密处理。上述处理过程的输出会直接反馈到 DEA 的输入，在加密状态下使用 K2 执行 DES。新结果会直接反馈到 DEA 的输入，在解密状态下使用 K1 执行 DES。生成的 64 位输出块 (O) 在进行位/字节/半字交换后，产生明文 (P)。

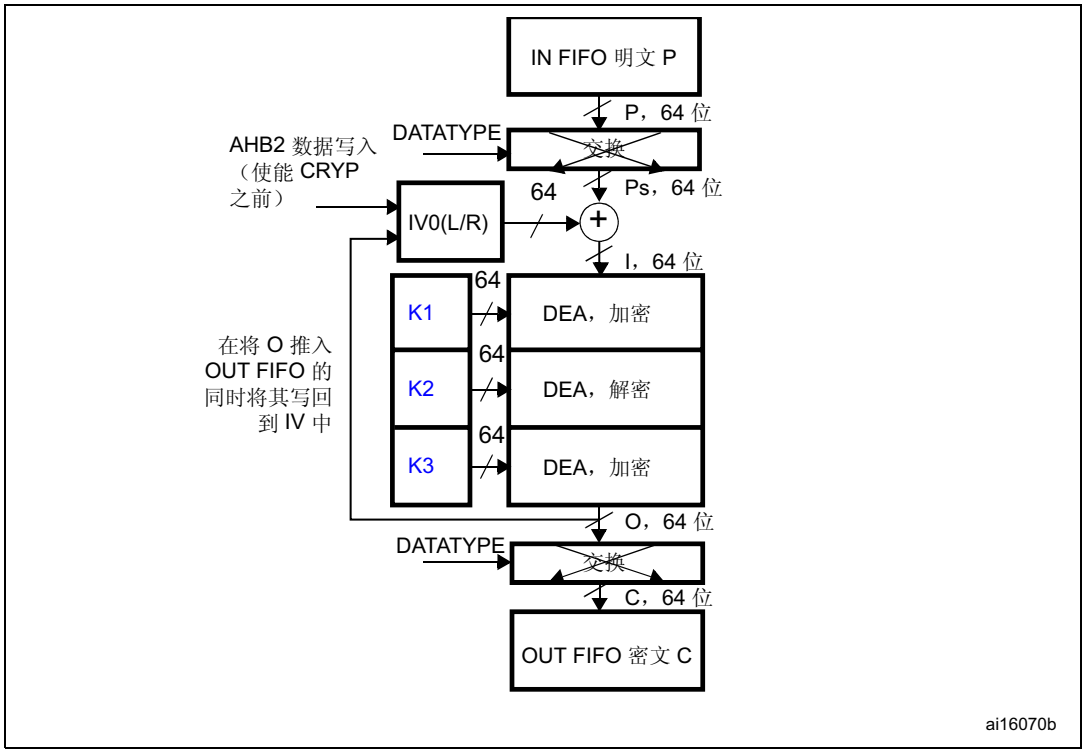
注： *有关数据交换的更多信息，请参见第 34.4.17 节：CRYP 数据寄存器和数据交换。*

有关详细的 DES/TDES 加密序列，请参见第 34.4.5 节：执行加密操作的 CRYP 过程。

DES/TDES-CBC 加密

图 315 介绍了 DES 和 TDES 密码块链接 (DES/TDES-ECB) 模式中的加密。通过将 CRYP_CR 中的 ALGOMODE 和 ALGODIR 分别编程为 0x1 和 0 来选择该模式。

图 315. DES/TDES-CBC 模式加密



K：密钥；C：密文；I：输入块；O：输出块；Ps：交换前（解码时）或交换后（编码时）的明文；P：明文；IV：初始化向量。

该模式首先将明文消息分成多个 64 位数据块。在 TCBC 加密中，执行位/字节/半字交换后获得的第一个输入块 (I_1) 是通过一个 64 位初始化向量 IV 与第一个明文数据块 (P_1) 进行异或运算形成的 (I_1 = IV ⊕ P_1)。输入块通过 DEA 在加密状态下使用 K1 进行加密处理。上述处理过程的输出会直接反馈到 DEA 的输入，在解密状态下使用 K2 执行 DES。上述处理过程的输出会直接反馈到 DEA 的输入，在加密状态下使用 K3 执行 DES。生成的 64 位输出块 (O_1) 将直接用作密文 (C_1)，即 C_1 = O_1。

然后，第一个密文块与第二个明文数据块进行异或运算，从而生成第二个输入块 (I_2) = (C_1 ⊕ P_2)。注意，此时的 I_2 和 P_2 指的是第二个块。第二个输入块通过 TDEA 处理而生成第二个密文块。

此加密处理会不断将后续密文块和明文块链接到一起，直到消息中最后一个明文块得到加密为止。

如果消息中包含的数据块数不是整数，则应按照应用程序指定的方式对最后的不完整数据块进行加密。

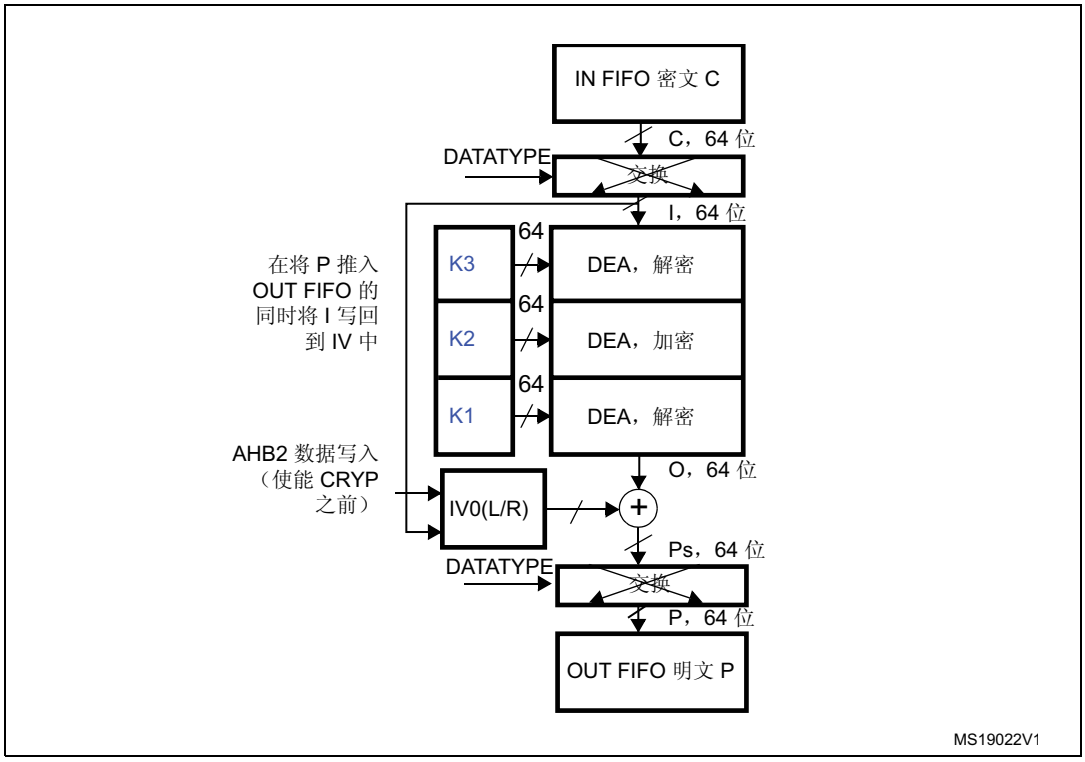
注： *有关数据交换的更多信息，请参见第 34.4.17 节：CRYP 数据寄存器和数据交换。*

有关详细的 DES/TDES 加密序列，请参见第 34.4.5 节：执行加密操作的 CRYP 过程。

DES/TDES-CBC 解密

图 315 介绍了 DES 和 TDES 密码块链接 (DES/TDES-ECB) 模式中的解密。通过向 CRYP_CR 中的 ALGOMODE 和 ALGODIR 分别写入 0x1 和 1 来选择该模式。

图 316. DES/TDES-CBC 模式解密



1. K：密钥；C：密文；I：输入块；O：输出块；Ps：交换前（解码时）或交换后（编码时）的明文；P：明文；IV：初始化向量。

在该模式下，第一个密文块 (C_1) 将直接用作输入块 (I_1)。该密钥序列与加密过程中使用的密钥序列相反。输入块通过 DEA 在解密状态下使用 K3 进行解密处理。上述处理过程的输出会直接反馈到 DEA 的输入，在加密状态下使用 K2 执行 DES。生成值会直接反馈到 DEA 的输入，在解密状态下使用 K1 处理 DES。生成的输出块与 IV（必须与加密期间使用的相同）进行异或运算，从而生成第一个明文块 ($P_1 = O_1 \oplus IV$)。

然后，第二个密文块将用作下一个输入块，并由 TDEA 进行处理。生成的输出块与第一个密文块进行异或运算，从而生成第二个明文数据块 ($P_2 = O_2 \oplus C_1$)。（注意，P_2 和 O_2 指的是第二个数据块。）

DES/TDES-CBC 解密过程将以此方式继续进行，直到最后一个完整密文块得到解密为止。

必须按照应用程序指定的方式对不完整数据块密文进行解密。

注： *有关数据交换的更多信息，请参见第 34.4.17 节：CRYP 数据寄存器和数据交换。*

有关详细的 DES/TDES 加密序列，请参见第 34.4.5 节：执行加密操作的 CRYP 过程。

ECB/CBC 模式下的 DES/TDES 挂起/恢复操作

在中断当前消息之前，用户应用程序必须遵循以下步骤：

1. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DIEN 位清零来停止 DMA 对 IN FIFO 的数据传输。
2. 等待 IN 和 OUT FIFO 均为空（CRYP_SR 中的 IFEM = 1 且 OFNE = 0），并且 BUSY 位清零。或者，由于输入 FIFO 最多可包含四个未处理的 DES 块，因此应用程序可根据实际情况决定在不等待 IN FIFO 变空的情况下中断密码处理过程。此时，还可以：
   a) 等待至 OUT FIFO 为空 (OFNE = 0)。
   b) 读回 IN FIFO 中加载的尚未处理的数据，并将其保存到内存中，直到 IN FIFO 为空。
3. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DOEN 位清零来停止 DMA 对 OUT FIFO 的数据传输。
4. 通过将 CRYP_CR 中的 CRYPEN 位置 0 来禁止密码处理器，然后保存当前配置（CRYP_CR 寄存器中的位 [9:2]）。如果选择 CBC 模式，则保存初始化向量寄存器，因为 CRYP_IVx(L/R)R 寄存器在数据处理过程中已发生更改，不再是初始值。

注： *无需保存密钥寄存器，因为应用程序已知初始密钥值。*

5. 如果使用 DMA，则保存 DMA 控制器状态（例如，指向 IN 和 OUT 数据传输的指针以及剩余字节数）。

要恢复消息处理，用户应用程序必须遵循以下步骤：

1. 如果使用 DMA，则重新配置 DMA 控制器以完成 FIFO IN 和 FIFO OUT 传输的剩余部分。
2. 确保通过读取 CRYP_CR 中的 CRYPEN 位（必须为 0）来禁止密码处理器。
3. 使用 CRYP_CR 中的初始设置再次配置密码处理器，以及使用所保存的配置再次配置密钥寄存器。
4. 如果选择 CBC 模式，则使用保存的配置恢复 CRYP_IVx(L/R)R 寄存器。
5. 可选择将上下文保存期间保存的数据写入 IN FIFO。
6. 通过将 CRYPEN 位置 1 来使能密码处理器。
7. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DIEN 和 DOEN 位置 1 再次使能对密码处理器的 DMA 请求。

小心： 当选择 DES/TDES 算法时，无法使用 DMA。

34.4.11 CRYP AES 基本链接模式（ECB 和 CBC）

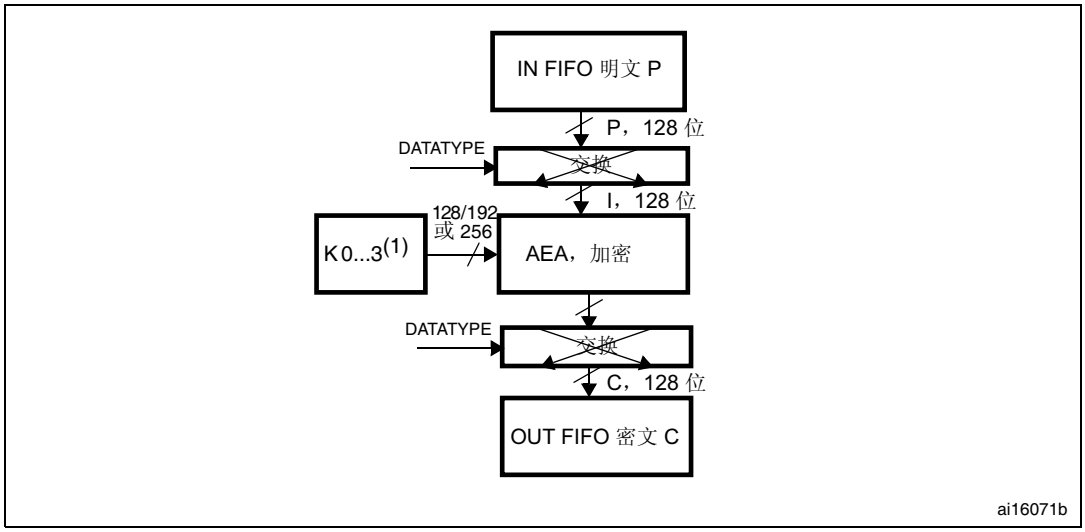
概述

FIPS PUB 197（2001 年 11 月 26 日）对 AES 计算内核提供的四种基本操作模式中涉及的处理过程进行了详尽的解释，这四种模式为：AES-ECB 加密、AES-ECB 解密、AES-CBC 加密和 AES-CBC 解密。本部分仅对每种模式进行了简要解释。

AES ECB 加密

图 317 介绍了 AES 电子密码本 (AES-ECB) 模式加密。通过向 CRYP_CR 中的 ALGOMODE 和 ALGODIR 分别写入 0x4 和 0 来选择该模式。

图 317. AES-ECB 模式加密



1. K：密钥；C：密文；I：输入块；O：输出块；P：明文。
2. 如果密钥大小 = 128：密钥 = [K3 K2]。
   如果密钥大小 = 192：密钥 = [K3 K2 K1]
   如果密钥大小 = 256：密钥 = [K3 K2 K1 K0]。

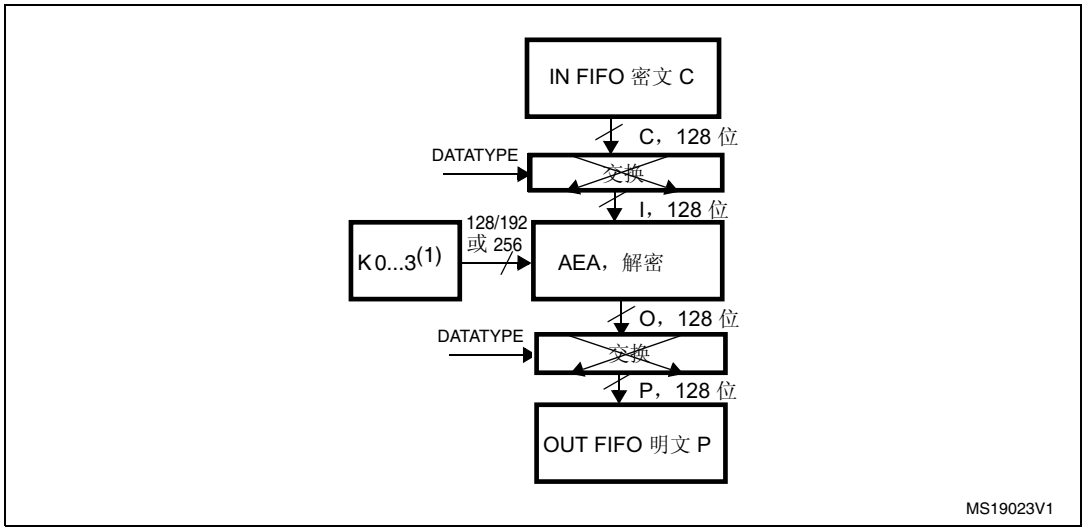
在该模式下，128 位明文数据块 (P) 经过位/字节/半字交换后用作输入块 (I)。输入块通过 AEA 在加密状态下使用 128 位、192 位或 256 位密钥进行处理。执行位/字节/半字交换后，生成的 128 位输出块 (O) 将用作密文 (C)。该密文随后推入 OUT FIFO。

有关数据交换的更多信息，请参见*第 34.4.17 节：CRYP 数据寄存器和数据交换*。

AES ECB 解密

图 318 介绍了 AES 电子密码本 (AES-ECB) 模式解密。通过将 CRYP_CR 中的 ALGOMODE 和 ALGODIR 分别编程为 0x4 和 1 来选择该模式。

图 318. AES-ECB 模式解密



1. K：密钥；C：密文；I：输入块；O：输出块；P：明文。
2. 如果密钥大小 = 128 => 密钥 = [K3 K2]。
   如果密钥大小 = 192 => 密钥 = [K3 K2 K1]
   如果密钥大小 = 256 => 密钥 = [K3 K2 K1 K0]。

要在 ECB 模式下执行 AES 解密，需要准备密钥（需要针对加密执行完整的密钥计划），方法为：收集最后一个轮密钥并将其用作解密密文的第一个轮密钥。该准备阶段通过 AES 内核计算完成。有关如何准备密钥的详细信息，请参见*第 34.4.7 节：为解密准备 CRYP AES 密钥*。

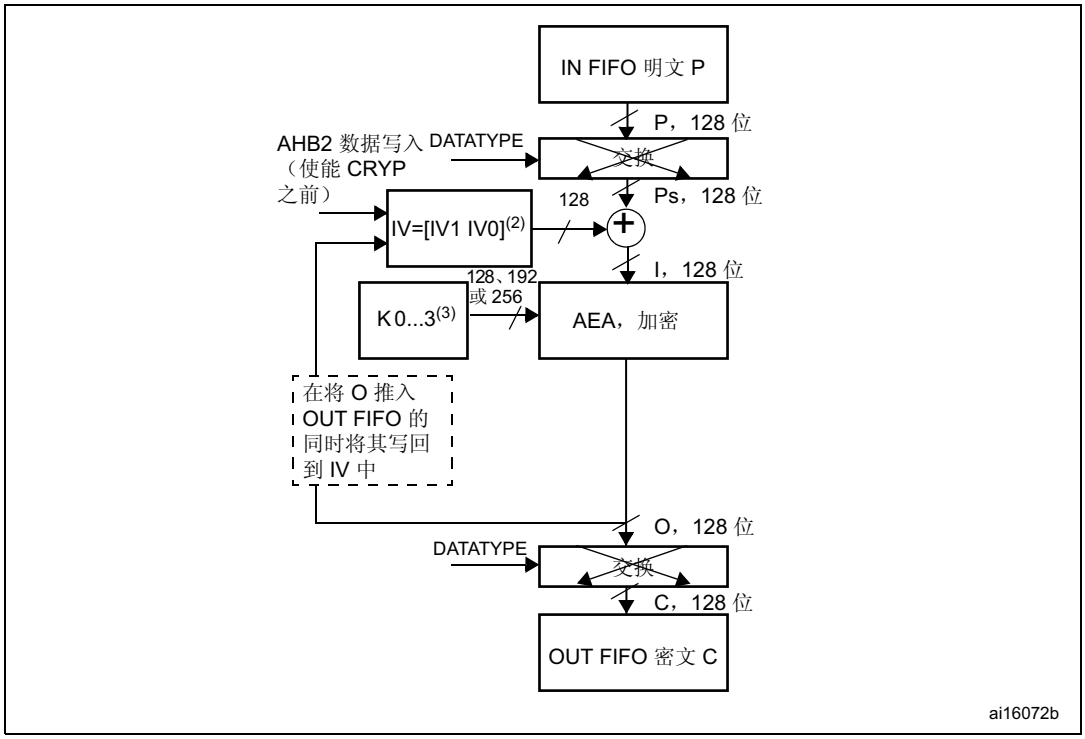
完成密钥准备后，按以下所述执行解密：对 128 位密文块 (C) 执行位/字节/半字交换后将其用作输入块 (I)。该密钥序列与加密处理中的密钥序列相反。执行位/字节/半字交换，生成的 128 位输出块 (O) 将产生明文 (P)。AES-CBC 解密过程将以此方式继续进行，直到最后一个完整密文块得到解密为止。

有关数据交换的更多信息，请参见*第 34.4.17 节：CRYP 数据寄存器和数据交换*。

AES CBC 加密

图 319 所示为 AES 密码分组链接 (AES-CBC) 模式加密。通过向 CRYP_CR 中的 ALGOMODE 和 ALGODIR 分别写入 0x5 和 0 来选择该模式。

图 319. AES-CBC 模式加密



1. K：密钥；C：密文；I：输入块；O：输出块；Ps：交换前（解码时）或交换后（编码时）的明文；P：明文；IV：初始化向量。
2. IVx = [IVxR IVxL]，R = 右，L = 左。
3. 如果密钥大小 = 128 => 密钥 = [K3 K2]。
   如果密钥大小 = 192 => 密钥 = [K3 K2 K1]
   如果密钥大小 = 256 => 密钥 = [K3 K2 K1 K0]。

在该模式下，执行位/字节/半字交换后获得的第一个输入块 (I_1) 是通过一个 128 位初始化向量 IV 与第一个明文数据块 (P_1) 进行异或运算形成的 (I_1 = IV ⊕ P_1)。输入块通过 AEA 在加密状态下使用 128 位、192 位或 256 位密钥 (K0...K3) 进行处理。生成的 128 位输出块 (O_1) 将直接用作密文 (C_1)，即 C_1 = O_1。然后，第一个密文块与第二个明文数据块进行异或运算，从而生成第二个输入块 (I_2) = (C_1 ⊕ P_2)。注意，此时的 I_2 和 P_2 指的是第二个块。第二个输入块通过 AEA 处理而生成第二个密文块。此加密处理会不断将后续密文块和明文块链接到一起，直到消息中最后一个明文块得到加密为止。

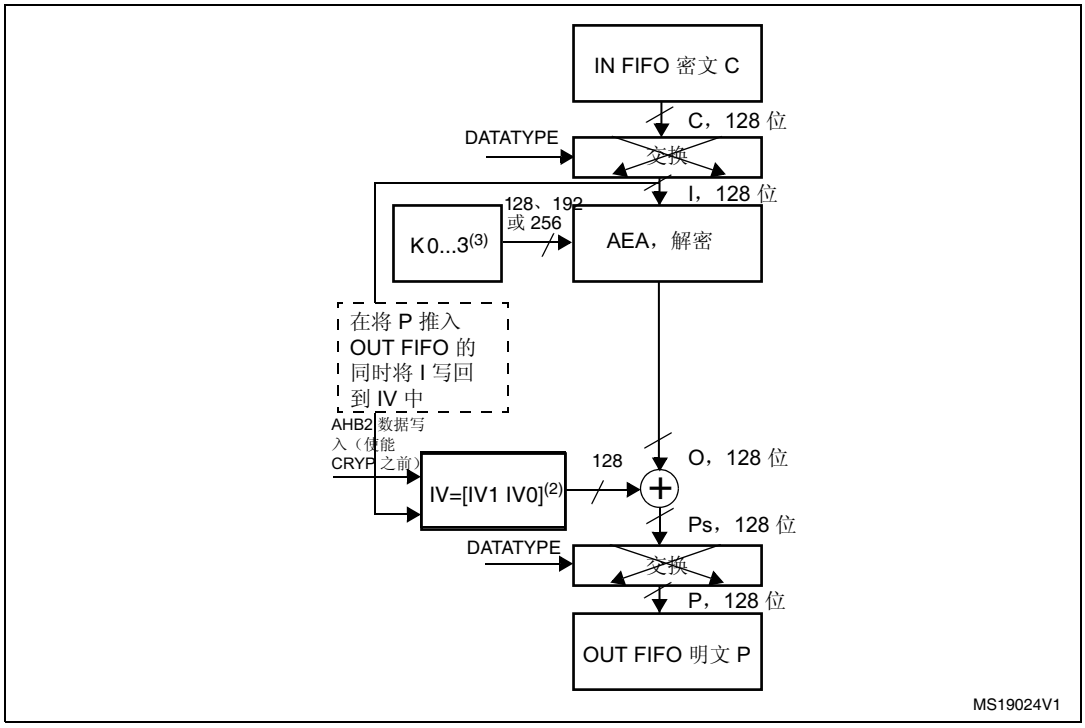
如果消息中包含的数据块数不是整数，则应按照应用程序指定的方式对最后的不完整数据块进行加密，如*第 34.4.8 节：CRYP 窃取和数据填充*所述。

有关数据交换的更多信息，请参见*第 34.4.17 节：CRYP 数据寄存器和数据交换*。

AES CBC 解密

图 320 所示为 AES 密码块链接 (AES-CBC) 模式解密。通过向 CRYP_CR 中的 ALGOMODE 和 ALGODIR 分别写入 0x5 和 1 来选择该模式。

图 320. AES-CBC 模式解密



1. K：密钥；C：密文；I：输入块；O：输出块；Ps：交换前（解码时）或交换后（编码时）的明文；P：明文；IV：初始化向量。
2. IVx = [IVxR IVxL]，R = 右，L = 左。
3. 如果密钥大小 = 128 => 密钥 = [K3 K2]。
   如果密钥大小 = 192 => 密钥 = [K3 K2 K1]
   如果密钥大小 = 256 => 密钥 = [K3 K2 K1 K0]。

在 CBC 模式下（如 ECB 模式），必须准备密钥才可以执行 AES 解密。有关如何准备密钥的详细信息，请参见*第 34.4.7 节：为解密准备 CRYP AES 密钥*。

完成密钥准备过程后，按以下所述执行解密：直接将第一个 128 位密文块 (C_1) 用作输入块 (I_1)。输入块通过 AEA 在解密状态下使用 128 位、192 位或 256 位密钥进行处理。生成的输出块与 128 位初始化向量 IV（必须与加密期间使用的相同）进行异或运算，从而生成第一个明文块 ($P_1 = O_1 \oplus IV$)。

然后，第二个密文块将用作下一个输入块，并由 AEA 进行处理。生成的输出块与第一个密文块进行异或运算，从而生成第二个明文数据块 ($P_2 = O_2 \oplus C_1$)。（注意，P_2 和 O_2 指的是第二个数据块。）AES-CBC 解密过程将以此方式继续进行，直到最后一个完整密文块得到解密为止。

必须按照应用程序指定的方式对不完整数据块密文进行解密，如*第 34.4.8 节：CRYP 窃取和数据填充*所述。

有关数据交换的更多信息，请参见*第 34.4.17 节：CRYP 数据寄存器和数据交换*。

ECB/CBC 模式下的 AES 挂起/恢复操作

在中断当前消息之前，用户应用程序必须遵循以下步骤：

1. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DIEN 位清零来停止 DMA 对 IN FIFO 的数据传输。
2. 等待 IN 和 OUT FIFO 均为空（CRYP_SR 中的 IFEM = 1 且 OFNE = 0），并且 BUSY 位清零。
3. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DOEN 位清零来停止 DMA 对 OUT FIFO 的数据传输。
4. 通过将 CRYP_CR 中的 CRYPEN 位置 0 来禁止 CRYP，然后保存当前配置（CRYP_CR 寄存器中的位 [9:2]）。如果未选择 ECB 模式，则保存初始化向量寄存器，因为 CRYP_IVx(L/R)R 寄存器在数据处理过程中已发生更改，不再是初始值。

注： *无需保存密钥寄存器，因为应用程序已知初始密钥值。*

5. 如果使用 DMA，则保存 DMA 控制器状态（例如，指向 IN 和 OUT 数据传输的指针以及剩余字节数）。

要恢复消息处理，用户应用程序必须遵循以下步骤：

1. 如果使用 DMA，则重新配置 DMA 控制器以完成 FIFO IN 和 FIFO OUT 传输的剩余部分。
2. 确保通过读取 CRYP_CR 中的 CRYPEN 位（必须设为 0）来禁止密码处理器。
3. 使用 CRYP_CR 中的初始设置再次配置密码处理器，以及使用所保存的配置再次配置密钥寄存器。如果之前已与 SAES 外设共享密钥，将需要再次与之共享。
4. 针对 AES-ECB 或 AES-CBC 解密，必须再次准备密钥，如*第 34.4.7 节：为解密准备 CRYP AES 密钥*所述。
5. 如果未选择 ECB 模式，则使用保存的配置恢复 CRYP_IVx(L/R)R 寄存器。
6. 通过将 CRYPEN 位置 1 来使能密码处理器。
7. 如果使用 DMA，通过将 CRYP_DMACR 寄存器中的 DIEN 和 DOEN 位置 1 再次使能来自密码处理器的 DMA 请求。

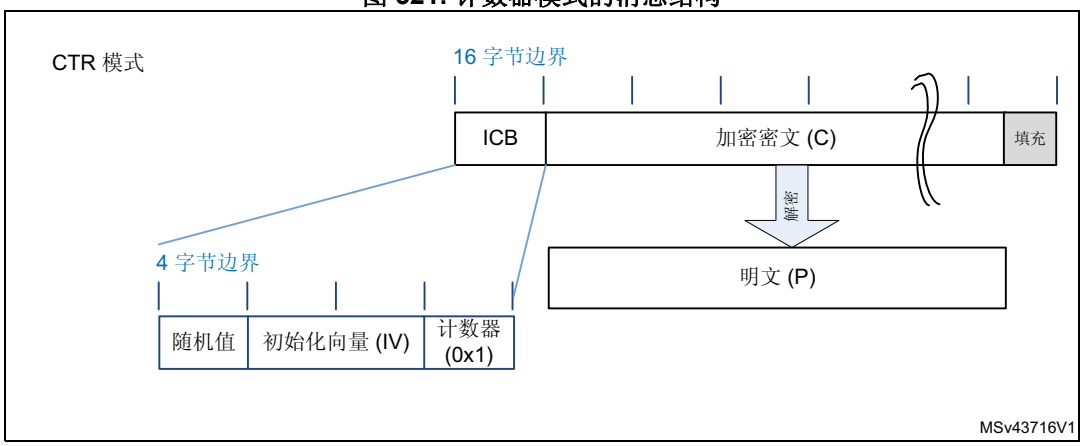
34.4.12 CRYP AES 计数器模式 (AES-CTR)

概述

AES 计数器模式 (CTR) 使用 AES 块作为密钥流生成器。然后，生成的密钥与明文进行异或运算来获得密文。

NIST *特别出版物 800-38A，块密码工作模式的建议*对 CTR 链接进行了相关定义。*图 321* 给出了 CTR 模式下的典型消息结构。

图 321. 计数器模式的消息结构



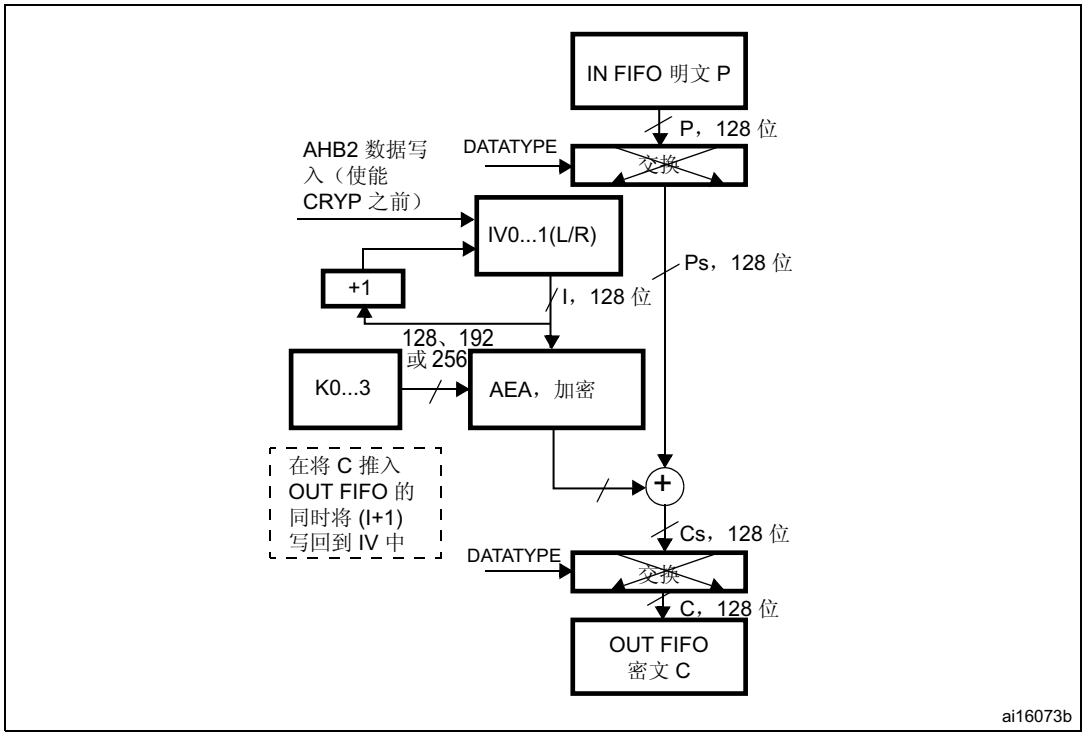
该消息的结构如下：

- 16 字节的初始计数器块 (ICB)，它由三个不同的字段组成：
  - *随机值*：32 位一次性值（即，应为每一次新的通信分配新的随机值）。
  - *初始化向量* (IV)：64 位值，对于给定密钥模式的每次执行，该值都必须唯一。
  - *计数器*：32 位大端模式的整数，每当处理完一个块，该值递增。应将计数器的初始值设为 1。
- 明文 (P) 将经过认证并被加密为密文 C（长度已知）。该长度可以不是 16 字节的倍数，在这种情况下需要明文填充。

AES CTR 处理

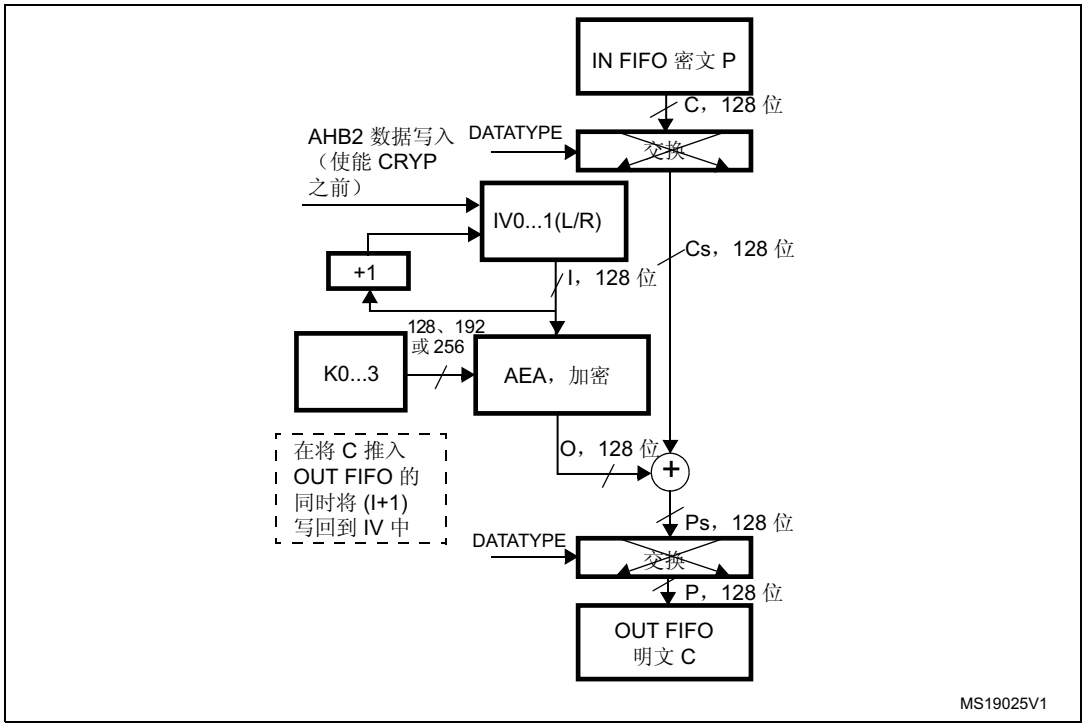
图 322（*图 323*）介绍了该外设中实现的 AES-CTR 加密（解密）过程。通过将 CRYP_CR 中的 ALGOMODE 位字段编程为 0x6 来选择该模式。

图 322. AES-CTR 模式加密



1. K：密钥；C：密文；I：输入块；O：输出块；Ps：交换前（解码时）或交换后（编码时）的明文；Cs：交换后（解码时）或交换前（编码时）的密文；P：明文；IV：初始化向量。

图 323. AES-CTR 模式解密



1. K：密钥；C：密文；I：输入块；O：输出块；Ps：交换前（解码时）或交换后（编码时）的明文；Cs：交换后（解码时）或交换前（编码时）的密文；P：明文；IV：初始化向量。

在 CTR 模式下，输出块会与后续输入块先进行异或运算，然后再输入到相应算法中。外设中的初始化向量必须按*表 283* 所示进行初始化。

表 283. 计数器模式初始化向量

CRYP_IV1RR[31:0]	CRYP_IV1LR[31:0]	CRYP_IV0RR[31:0]	CRYP_IV0LR[31:0]
IVI[127:96]	IVI[95:64]	IVI[63:32]	IVI[31:0] 32 位计数器 = 0x1

在 CBC 模式下，处理第一个数据块时仅使用 CRYP_IVx(L/R)R 寄存器一次，而在 CTR 模式下则有所不同，处理每个数据块都使用 IV 寄存器，并且外设会使 32 个最低有效位递增（其他 96 个最高有效位保持不变）。

CTR 解密与 CTR 加密并无不同，因为内核始终会对当前的计数器块加密以产生密钥数据流，该密钥数据流与输入中的明文或密码进行异或运算。因此，将 ALGOMODE 设为 0x6 时，ALGODIR 无关紧要。

注： *在该模式下，不得为解密准备密钥。*

必须使用以下步骤在 CTR 链接模式下执行加密或解密：

1. 确保通过将 CRYP_CR 寄存器中的 CRYPEN 位清零来禁止密码处理器。
2. 按如下所述配置 CRYP_CR：
   a) 将 ALGOMODE 位编程为 0x6 来选择 CTR 模式。ALGODIR 可设置为任何值。
   b) 通过 DATATYPE 位配置数据类型（1 位、8 位、16 位或 32 位）。
   c) 使用 KEYSIZE 位定义密钥长度。
   d) 如果未通过软件写入密钥，而是使用共享密钥模式从 SAES 外设传入密钥，则将 KMOD 位设置为 10，否则保持为 00。有关详细信息，请参见*第 34.4.16 节*。
3. 按*表 283* 所述初始化 CRYP_KEYRx 中的密钥寄存器（128 位、192 位或 256 位）（如果需要）以及初始化向量 (IV)。
4. 向 CRYP_CR 寄存器中的 FFLUSH 位写入 1，刷新 IN 和 OUT FIFO。
5. 如果是最后一个块，则可选择用零填充数据以获得完整的块。
6. 在密码处理器中附加数据并读取结果。*第 34.4.5 节：执行加密操作的 CRYP 过程*介绍了三种可能的情况。
7. 重复上一步，直到处理完倒数第二个块。对于最后一个块，执行前两步。对于这最后一个块，当其大小小于 16 字节时，驱动程序必须将不属于数据的一部分的位丢弃。

CTR 模式下的挂起/恢复操作

该模式与 CBC 模式相似，可以中断消息、发送优先级更高的消息，并可恢复已中断的消息。有关详细的 CBC 流程，请参见*第 34.4.11 节：CRYP AES 基本链接模式（ECB 和 CBC）*。

注： *与 CBC 模式类似，恢复操作期间必须重新加载 IV 寄存器。*

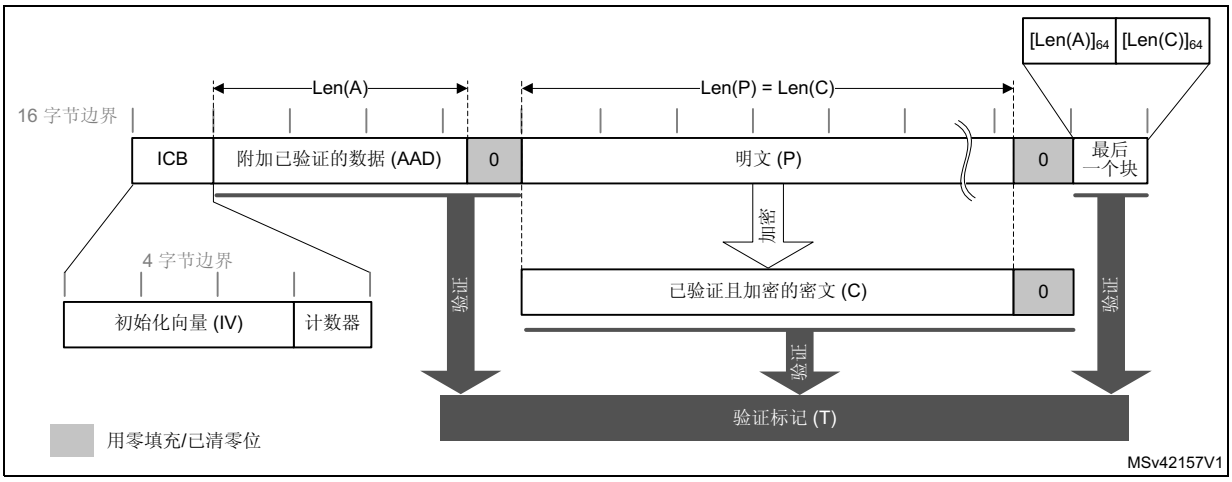
34.4.13 CRYP AES Galois/计数器模式 (GCM)

概述

AES Galois/计数器模式 (GCM) 允许加密和验证明文，以及生成对应的密文和标记（也称为消息认证码）。为实现保密性，GCM 算法基于 AES 计数器模式。它使用固定有限域乘法器来生成标记。

NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/计数器模式 (GCM) 和 GMAC* 中对 GCM 链接进行了定义。*图 324* 给出了 GCM 模式下的典型消息结构。

图 324. Galois/计数器模式的消息结构



该消息结构的定义如下：

- 16 字节的初始计数器块 (ICB)，它由两个不同的字段组成：
  - *初始化向量* (IV)：96 位值，对于给定密钥模式的条件下，每次执行，该值都必须唯一。请注意，GCM 标准支持短于 96 位的 IV，在这种情况下应遵循严格的规则。
  - *计数器*：32 位大端模式的整数，每当处理完一个块，该值递增。根据 NIST 规范，处理有效负载的第一个块时，计数器值为 0x2。
- 认证的标头 A（也称为附加认证数据）具有已知长度 *Len(A)*，此长度值可以不是 16 字节的倍数，但不能超过 2^{64}-1 位。消息的这一部分仅经过认证，未被加密。
- 明文消息 (P) 将经过认证且被加密为密文 C，其具有已知长度 *Len(P)*，该长度值可以不是 16 字节的倍数，但不能超过 2^{32} -2 个 128 位块。

注： *GCM 标准规定密文 C 的位长度与明文 P 的位长度相同。*

- 当消息某一部分（AAD 或 P）的长度不是 16 字节的倍数时，需要采取特殊填充方案。
- 最后一个块由 A 的长度（64 位）和密文 C 的长度（64 位）组成，如*表 284* 所示。

表 284. GCM 最后一个块定义

字节序	位[0]	位[32]	位[64]	位[96]
输入数据	0x0	标头长度 [31:0]	0x0	有效负载长度 [31:0]

AES GCM 处理

通过向 CRYP_CR 中的 ALGOMODE 位字段写入 0x6 来选择该模式。

GCM 模式下明文的保密性机制是计数器模式的变型，其具有特定 32 位递增函数，可生成必要的计数器块序列。

CRYP_IVx(L/R)R 寄存器用于处理每个数据块。密码处理器自动递增计数器块的 32 个最低有效位。应用程序写入的第一个计数器块 (CB1) 等于初始计数器块递增 1（请参见*表 285*）。

表 285. GCM 模式 IV 寄存器初始化

寄存器	CRYP_IV0LR[31:0]	CRYP_IV0RR[31:0]	CRYP_IV1LR[31:0]	CRYP_IV1RR[31:0]
输入数据	ICB[127:96]	ICB[95:64]	ICB[63:32]	ICB[31:0] 32 位计数器 = 0x2

注： *在该模式下，不得为解密准备密钥。*

GCM 模式下的认证机制基于散列函数（称为 *GF2mul*），其在二元 Galois 域内执行与固定参数（称为散列子密钥 (H)）的乘法。

要处理 GCM 消息，驱动程序必须经过四个阶段，后续子章节对此进行了介绍。

- 初始化阶段：外设准备 GCM 散列子密钥 (H) 并执行 IV 处理
- 标头阶段：外设处理附加认证数据 (AAD)（仅使用散列计算）。
- 有效负载阶段：外设基于散列计算、密钥流加密和数据异或运算处理明文 (P)。其操作方式与密文 (C) 类似。
- 最后阶段：外设使用数据的最后一个块生成认证标记 (T)。

1. GCM 初始化阶段

在此第一步中，将在内部计算和保存 GCM 散列子密钥 (H)，供处理所有块使用。建议遵守下列顺序：

a) 确保通过将 CRYP_CR 寄存器中的 CRYPEN 位清零来禁止密码处理器。

b) 通过将 CRYP_CR 中的 ALGOMODE 位编程为 0x8 来选择 GCM 链接模式。在同一寄存器中使用 KEYSIZE 位定义密钥长度，并使用 DATATYPE 位定义数据类型。如果未通过软件写入密钥，而是使用共享密钥模式从 SAES 外设传入密钥，则将 KMOD 位设置为 10，否则保持为 00。有关详细信息，请参见*第 34.4.16 节*。

c) 将 CRYP_CR 中的 GCM_CCMPH 位配置为 0b00 来指示正在进行初始化阶段。

d) 按照*表 285* 的定义初始化 CRYP_KEYRx 中的密钥寄存器（128 位、192 位和 256 位）（如果需要）以及初始化向量 (IV)。

e) 将 CRYPEN 位置 1 以开始计算散列密钥。

f) 等待 CRYPEN 位由密码处理器清零后进入下一个阶段。

2. GCM 标头阶段

在 GCM 初始化阶段后方能执行以下步骤：在跳转到有效负载阶段之前，必须先执行完该步骤。加密与解密的顺序完全相同。

g) 将 CRYP_CR 中的 GCM_CCMPH 位置为 0b01 来指示正在进行标头阶段。

h) 将 CRYPEN 位置 1 开始接受数据。

i) 如果是附加认证数据的最后一个块，则可选择用零填充数据以获得完整的块。

j) 在密码处理器中添加附加认证数据。*第 34.4.5 节：执行加密操作的 CRYP 过程*介绍了三种可能的情况。

k) 重复上一步，直到处理完倒数第二个附加认证数据块。对于最后一个块，执行前两步。提供所有附加认证数据后，请等待 BUSY 标志清零后进入下一阶段。

注： *如无附加认证数据（即 Len(A) = 0），则可以跳过该阶段。*

在标头和有效负载阶段，CRYPEN 位不会自动由密码处理器清零。

3. GCM 有效负载阶段（加密或解密）

如果有效负载大小不为空，则必须在 GCM 标头阶段后执行以下步骤。在此阶段期间，加密/解密的有效负载将存储在 CRYP_DOUT 寄存器中。

l) 将 CRYPEN 位置 0。

m) 将 CRYP_CR 寄存器中的 GCM_CCMPH 位配置为 0b10 来指示正在进行有效负载阶段。

n) 通过 CRYP_CR 中的 ALGODIR 位来选择算法方向（0 表示加密，1 表示解密）。

o) 将 CRYPEN 位置 1 开始接受数据。

p) 如果是最后一个明文块，则可选择用零填充数据以获得完整的块。有关加密的更多详细信息，请参见*第 34.4.8 节：CRYP 窃取和数据填充*。

q) 在密码处理器中附加有效负载数据并读取结果。*第 34.4.5 节：执行加密操作的 CRYP 过程*介绍了三种可能的情况。

r) 重复上一步，直到倒数第二个明文块被加密或直到最后一个密文块被解密。对于最后一个明文块（仅限加密），执行前两步。对于最后一个块，当其大小小于 16 字节时，驱动程序必须将不属于明文或密文的一部分的位丢弃。提供完所有有效负载数据后，等待 BUSY 标志清零。

注： *如无有效负载数据（即 Len(C) = 0），则可以跳过该阶段（请参见 GMAC 模式）。*

4. GCM 最后阶段

在此最后一步中，密码处理器会生成 GCM 认证标记并将其存储在 CRYP_DOUT 寄存器中。

s) 将 CRYP_CR 中的 GCM_CCMPH[1:0] 位配置为 0b11 来指示正在进行最后阶段。将同一寄存器中的 ALGODIR 位置 0。

t) 将输入写入 CRYP_DIN 寄存器四次。输入必须包含附加认证数据的长度（以位为单位，64 位编码）和有效负载的长度（以位为单位，64 位编码），如*表 284* 所示。

注： *在最后阶段，必须正常插入数据（无交换）。*

u) 等待 CRYP_SR 寄存器中的 OFNE 标志（FIFO 输出非空）置 1。

v) 读取 CRYP_DOUT 寄存器四次：此输出即为认证标记。

w) 禁止密码处理器（CRYP_CR 中的 CRYPEN 位 = 0）

x) 如果正在执行经验证的解密，请将生成的标记与通过消息传递的预期标记进行比较。

GCM 模式下的挂起/恢复操作

在标头或有效负载阶段中断当前消息之前，用户应用程序必须遵循以下步骤：

1. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DIEN 位清零来停止 DMA 对 IN FIFO 的数据传输。
2. 等待 IN 和 OUT FIFO 均为空（CRYP_SR 寄存器中的 IFEM = 1 且 OFNE = 0），并且 BUSY 位清零。
3. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DOEN 位清零来停止 DMA 对 OUT FIFO 的数据传输。
4. 通过将 CRYP_CR 中的 CRYPEN 位置 0 来禁止密码处理器，然后保存当前配置（CRYP_CR 寄存器中的位 [9:2]、位 [17:16] 和位 19）。此外，还应保存初始化向量寄存器，因为 CRYP_IVx(L/R)R 寄存器在数据处理过程中已发生更改，不再是初始值。

注： *无需保存密钥寄存器，因为应用程序已知其初始密钥值。*

5. 保存上下文交换寄存器：CRYP_CSGCMCCM0..7R 和 CRYP_CSGCM0..7R
6. 如果使用 DMA，则保存 DMA 控制器状态（指向 IN 和 OUT 数据传输的指针以及剩余字节数等）。

要恢复消息处理，用户必须遵循以下步骤：

1. 如果使用 DMA，则重新配置 DMA 控制器以完成 FIFO IN 和 FIFO OUT 传输的剩余部分。
2. 确保通过读取 CRYP_CR 中的 CRYPEN 位（必须为 0）来禁止密码处理器。
3. 使用 CRYP_CR 中的初始设置再次配置密码处理器，以及使用所保存的配置再次配置密钥寄存器。如果之前已与 SAES 外设共享密钥，将需要再次与之共享。
4. 恢复上下文交换寄存器：CRYP_CSGCMCCM0..7R 和 CRYP_CSGCM0..7R
5. 使用保存的配置恢复 CRYP_IVx(L/R)R 寄存器。
6. 通过将 CRYPEN 位置 1 来使能密码处理器。
7. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DIEN 和 DOEN 位置 1 再次使能对密码处理器的 DMA 请求。

注： *在标头阶段，不使用 DMA OUT FIFO 传输。*

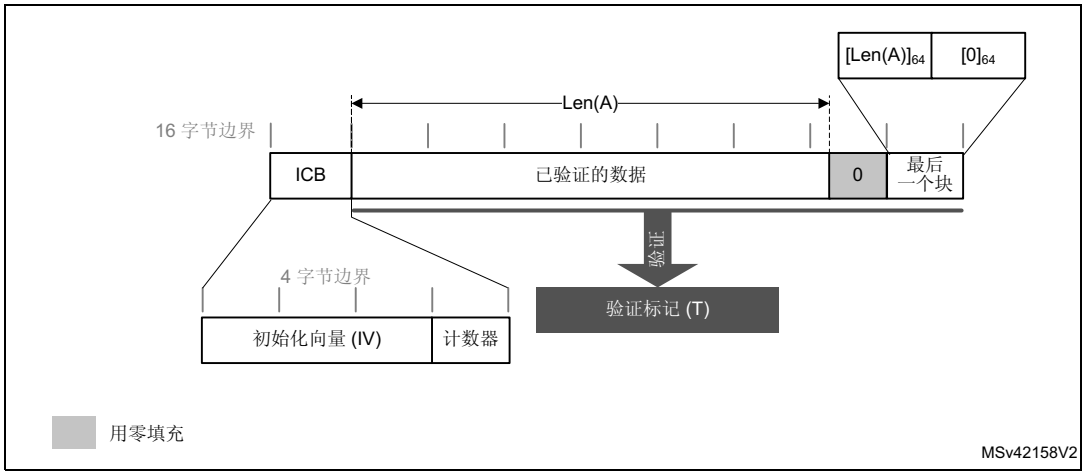
34.4.14 CRYP AES Galois 消息认证码 (GMAC)

概述

Galois 消息认证码 (GMAC) 支持认证明文以及生成相应的标记信息（也称为消息认证码）。它基于 GCM 算法，NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/计数器模式 (GCM) 和 GMAC* 中对此进行了相关定义。

图 325 给出了 GMAC 模式下的典型消息结构。

图 325. Galois 消息认证码模式的消息结构



AES GMAC 处理

通过向 CRYP_CR 中的 ALGOMODE 位字段写入 0x6 来选择该模式。

GMAC 算法相当于应用程序在仅由标头构成的消息上的 GCM 算法。因此，除了不使用有效负载阶段 (3) 之外，所有步骤和设置均与 GCM 模式相同。

GMAC 模式下的挂起/恢复操作

除了只能中断标头阶段 (2) 外，GMAC 与 GCM 算法完全相同。

34.4.15 CRYP AES CBC-MAC 计数器模式 (CCM)

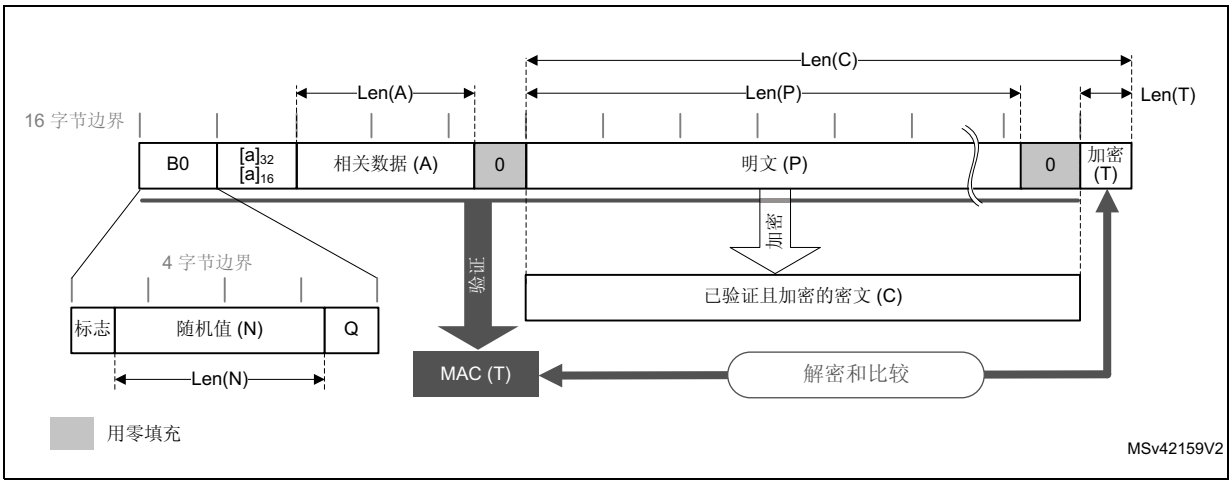
概述

AES 密码块链接-消息认证码计数器模式 (CCM) 算法可用于加密和认证明文，以及生成对应的密文和标记（也称为消息认证码）。为实现保密性，CCM 算法基于 AES 计数器模式。它使用密码块链接技术来生成消息认证码。这通常被称为 CBC-MAC

注： *NIST 不允许将该 CBC-MAC 用作 CCM 规范上下文之外的认证模式。*

NIST *特别出版物 800-38C，块密码工作模式的建议 - CCM 模式实现认证和保密性*中对 CCM 链接进行了介绍。*图 326* 给出了 CCM 模式下的典型消息结构

图 326. CBC-MAC 计数器模式的消息结构



该消息的结构如下：

- 一个 16 字节的首个认证块（标准将其定义为 B0），它由三个不同的字段组成：
  - *Q*：位字符串，表示 P 的字节长度 (Plen)
  - *随机值 (N)*：一次性值（即，应为每一次新的通信分配新的随机值）。随机值 *Nlen* 的大小 + *Plen* 的大小应等于 15 个字节。
  - 标志：最高有效字节，包含四个控制信息标志（根据标准规定）。它包含两个 3 位字符串，用于编码值 t（MAC 长度，以字节表示）和 q（明文长度，例如 Plen<2^{8q} 个字节）。请注意，与 q 相关的计数器块范围为 2^{8q-4}，即，如果 q 的最大值为 8，则密码中使用的计数器块应为 60 位。
- 与相关数据 (A) 关联的 16 字节块 (B)。

  消息的这一部分仅经过认证，未被加密。这部分具有已知长度 *ALen*，此长度值可以不是 16 字节的倍数（请参见*图 326*）。标准还具有以下规定：对于第一个消息块 (B1) 的 MSB 位，以字节表示的相关数据长度 (*a*) 必须按如下所述进行编码：
  - 如果 $0 < a < 2^{16}-2^8$，则将其编码为 $[a]_{16}$，即两个字节。
  - 如果 $2^{16}-2^8 < a < 2^{32}$，则将其编码为 0xff || 0xfe || $[a]_{32}$，即六个字节。
  - 如果 $2^{32} < a < 2^{64}$，则将其编码为 0xff || 0xff || $[a]_{64}$，即十个字节。
- 与明文消息 (P) 相关的 16 字节块 (B)，此明文消息将经过认证并被加密为密文 C（具有已知长度 *Plen*）。此长度可以不是 16 字节的倍数（请参见*图 326*）。
- 长度为 *Tlen* 的加密 MAC (T) 被附加到总长度为 *Clen* 的密文 C。
- 当消息某一部分（A 或 P）的长度不是 16 字节的倍数时，需要采取特殊填充方案。

注： *CCM 链接模式同样只能与相关数据搭配使用（即，无有效负载）。*

例如，*NIST 特别出版物 800-38C* 的 C.1 部分给出了以下内容：

N：**10111213 141516**（Nlen = 56 位或 0x7 字节）

A：***00010203 04050607***（Alen = 64 位或 0x8 字节）

P：*20212223*（Plen = 32 位，即 Q = 0x*4* 字节）

T：6084341b（Tlen = 32 位或 t = 4）

B0：4f**101112 13141516** 00000000 00000004
B1：0008***0001 02030405 0607***0000 00000000
B2：*20212223* 00000000 00000000 00000000

CTR0：07**10111213 141516** 00000000 00000000

CTR1：07**10111213 141516** 00000000 00000001

以下章节对控制块 CTRx 的使用进行了介绍。必须通过软件来管理第一个块 (B0) 生成 CTR0 的过程。

AES CCM 处理

通过向 CRYP_CR 中的 ALGOMODE 位字段写入 0x9 来选择该模式。

生成-加密过程的数据输入包括一个有效随机值、一个有效的有效负载字符串和一个有效的相关数据字符串，这些数据均经过正确格式化。对经过格式化的数据应用 CBC 链接机制可生成长度已知的 MAC。计数器模式加密需要足够长的计数器块序列作为输入，应用于有效负载字符串，并单独应用于 MAC。生成的数据（称为密文 C）是明文 P 加密生成过程的输出。

CRYP_IVx(L/R)R 寄存器用于处理每个数据块。密码处理器以第一个块 (B0) 定义的位长度自动使 CTR 计数器递增。由应用程序写入的第一个计数器 CTR1 相当于 B0（前 5 位清零，并且包含 P 字节长度的最高有效位也清零），然后按 1 递增（请参见*表 286*）。

表 286. CCM 模式 IV 寄存器初始化

寄存器	CRYP_IV0LR[31:0]	CRYP_IV0RR[31:0]	CRYP_IV1LR[31:0]	CRYP_IV1RR[31:0]
输入数据	B0[127:96]，其中的 5 个最高有效位置 0（标志位）	B0[95:64]	B0[63:32]	B0[31:0]，其中除了位 0 置 1 外，Q 长度位均置 0

注： *在该模式下，不得为解密准备密钥。*

要处理 CCM 消息，驱动程序必须经过四个阶段，下文对此进行了说明。

- **初始化阶段**：外设处理第一个块并准备第一个计数器块。
- **标头阶段**：外设处理相关数据 (A)（仅使用哈希计算）。
- **有效负载阶段**：外设基于哈希计算、计数器块加密和数据异或运算处理明文 (P)。其操作方式与密文 (C) 类似。
- **最后阶段**：外设生成消息认证码 (MAC)。

1. **CCM 初始化阶段**

   在此第一步中，将 CCM 消息的第一个块 (B0) 编程到 CRYP_DIN 寄存器中。在此阶段期间，CRYP_DOUT 寄存器不包含任何输出数据。建议遵守下列顺序：

   a) 确保通过将 CRYP_CR 寄存器中的 CRYPEN 位清零来禁止密码处理器。

   b) 通过将 CRYP_CR 寄存器中的 ALGOMODE 位编程为 0x9 来选择 CCM 链接模式。在同一寄存器中使用 KEYSIZE 位定义密钥长度，并使用 DATATYPE 位定义数据类型。如果未通过软件写入密钥，而是使用共享密钥模式从 SAES 外设传入密钥，则将 KMOD 位设置为 10，否则保持为 00。有关详细信息，请参见*第 34.4.16 节*。

   c) 将 CRYP_CR 中的 GCM_CCMPH 位配置为 0b00 来指示正在进行初始化阶段。

   d) 按照*表 286* 的定义初始化 CRYP_KEYRx 中的密钥寄存器（128 位、192 位和 256 位）（如果需要）以及包含 CTR1 信息的初始化向量 (IV)。

   e) 将 CRYP_CR 中的 CRYPEN 位置 1 开始接受数据。

   f) 向 CRYP_DIN 寄存器中写入 B0 数据包，然后等待 CRYPEN 位由密码处理器清零后进入下一个阶段。

注： *在最后阶段，必须正常插入数据（无交换）。*

2. **CCM 标头阶段**

   在 CCM 初始化阶段后方能执行以下步骤：在跳转到有效负载阶段之前，必须先执行完该步骤。加密与解密的顺序完全相同。在此阶段期间，CRYP_DOUT 寄存器不包含任何输出数据。

   g) 将 CRYP_CR 中的 GCM_CCMPH 位置为 0b01 来指示正在进行标头阶段。

   h) 将 CRYPEN 位置 1 开始接受数据。

   i) 如果是相关数据的最后一个块，则可选择用零填充数据以获得完整的块。

   j) 在密码处理器中附加相关数据。*第 34.4.5 节：执行加密操作的 CRYP 过程*介绍了三种可能的情况。

   k) 重复上一步，直到处理完倒数第二个相关数据块。对于最后一个块，执行前两步。提供完所有附加认证数据后，等待 BUSY 标志清零。

注： *如无相关数据 (AIen = 0)，则可以跳过该阶段。*

必须使用相关数据长度格式化相关数据的首个块 B1。此任务必须由驱动程序进行管理。

3. **CCM 有效负载阶段（加密或解密）**

   如果有效负载大小不为空，则必须在 CCM 标头阶段后执行以下步骤。在此阶段期间，加密/解密的有效负载将存储在 CRYP_DOUT 寄存器中。

   l) 将 CRYPEN 位置 0。

   m) 将 CRYP_CR 中的 GCM_CCMPH 位配置为 0b10 来指示正在进行有效负载阶段。

   n) 通过 CRYP_CR 中的 ALGODIR 位来选择算法方向（0 表示加密，1 表示解密）。

   o) 将 CRYPEN 位置 1 开始接受数据。

   p) 如果是最后一个明文块，则可选择用零填充数据以获得完整的块（仅限加密）。有关解密的更多详细信息，请参见*第 34.4.8 节：CRYP 窃取和数据填充*。

   q) 在密码处理器中附加有效负载数据并读取结果。*第 34.4.5 节：执行加密操作的 CRYP 过程*介绍了三种可能的情况。

   r) 重复上一步，直到倒数第二个明文块被加密或直到最后一个密文块被解密。对于最后一个明文块（仅限加密），执行前两步。对于最后一个密文块（仅限解密），当其大小小于 16 字节时，驱动程序必须将不属于明文的一部分的数据丢弃。提供完所有有效负载数据后，等待 BUSY 标志清零。

注： *如无有效负载数据（即，Plen = 0 或 Clen = Tlen），则可以跳过该阶段。*

注： *解密密文 C 时，不要忘了移除 LSB_{Tlen}(C) 加密标记信息。*

4. CCM 最后阶段

在此最后一步中，密码处理器会生成 CCM 认证标记并将其存储在 CRYP_DOUT 寄存器中。

s) 将 CRYP_CR 中的 GCM_CCMPH[1:0] 位配置为 0b11，以指示正在进行最后阶段，并将同一寄存器中的 ALGODIR 位置 0。

t) 在 CRYP_DIN 中加载*表 286* 给出的 CTR0 信息（位 [0] 置 0）。

注： *在最后阶段，必须正常插入数据（无交换）。*

u) 等待 CRYP_SR 寄存器中的 OFNE 标志（FIFO 输出非空）置 1。

v) 读取 CRYP_DOUT 寄存器四次：此输出即为加密的 CCM 标记。

w) 禁止密码处理器（CRYP_CR 中的 CRYPEN 位置 0）

x) 如果正在执行认证解密，则将生成的加密标记与在密文中填充的加密标记进行比较，即，LSB_{Tlen}(C) = MSB_{Tlen}（CRYP_DOUT 数据）。

CCM 模式下的挂起/恢复操作

在有效负载阶段中断当前消息之前，用户应用程序必须遵循以下步骤：

1. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DIEN 位清零来停止 DMA 对 IN FIFO 的数据传输。
2. 等待 IN 和 OUT FIFO 均为空（CRYP_SR 寄存器中的 IFEM = 1 且 OFNE = 0），并且 BUSY 位清零。
3. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DOEN 位清零来停止 DMA 对 OUT FIFO 的数据传输。
4. 通过将 CRYP_CR 中的 CRYPEN 位置 0 来禁止密码处理器，然后保存当前配置（CRYP_CR 寄存器中的位 [9:2]、位 [17:16] 和位 19）。此外，还应保存初始化向量寄存器，因为 CRYP_IVx 寄存器在数据处理过程中已发生更改，不再是初始值。

注： *无需保存密钥寄存器，因为应用程序已知其初始密钥值。*

5. 保存上下文交换寄存器：CRYP_CSGCMCCM0..7R
6. 如果使用 DMA，则保存 DMA 控制器状态（指向 IN 和 OUT 数据传输的指针以及剩余字节数等）。

要恢复消息处理，用户应用程序必须遵循以下步骤：

1. 如果使用 DMA，则重新配置 DMA 控制器以完成 FIFO IN 和 FIFO OUT 传输的剩余部分。
2. 确保通过读取 CRYP_CR 中的 CRYPEN 位（必须为 0）来禁止密码处理器。
3. 使用 CRYP_CR 中的初始设置再次配置密码处理器，以及使用所保存的配置再次配置密钥寄存器。如果之前已与 SAES 外设共享密钥，将需要再次与之共享。
4. 恢复上下文交换寄存器：CRYP_CSGCMCCM0..7R
5. 使用保存的配置恢复 CRYP_IVx(L/R)R 寄存器。
6. 通过将 CRYPEN 位置 1 来使能密码处理器。
7. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DIEN 和 DOEN 位置 1 再次使能对密码处理器的 DMA 请求。

注： *在标头阶段，不使用 DMA OUT FIFO 传输。*

34.4.16 与安全 AES 协处理器共享 AES 密钥

密码处理器可以使用 SAES 外设作为安全协处理器。在这种情况下，安全应用在稳健的 SAES 外设中准备密钥，当密钥就绪后，CRYP 应用通过专用的硬件密钥总线加载这一 AES 共享密钥。

以下部分以及 SAES 外设的*使用共享密钥的 SAES 操作*部分介绍了推荐的步骤。

1. 用户应用在共享密钥模式 (KMOD[1:0] = 10) 下对要在 SAES 中共享的密钥进行一次加密。然后，必须通过 SAES_CR 中的 KSHAREID[1:0] 选择 CRYP 实例。
2. 每次需要共享密钥时，CRYP 外设应用都需要选择相同的共享密钥模式 (KMOD[1:0] = 10)，以在 SAES 中对其进行解密。
3. 一旦共享密钥解密并载入到 SAES_KEYRx 寄存器中，就可以与 CRYP 共用（仅在 AES 模式下）。要在 CRYP 中加载共享密钥，应用程序必须遵循初始化过程（如*第 34.4.5 节*的 ECB 和 CBC 链接模式部分所述），但存在以下区别：
   - 必须使用 ALGOMODE 选择 AES 算法。当 KMOD = 10 时，选择 DES/TDES 算法会触发 KERF 错误。
   - KEYSIZE 必须与 SAES 值（128 或 256 位）相同。
   - 必须选择共享密钥模式，而不是正常密钥模式 (KMOD = 10)。当 KEYVALID 清零时，向 CRYP_CR 中写入该值会触发自动将密钥从 SAES 密钥寄存器传输到相应的 CRYP_KEYRx 寄存器，同时 CRYP_SR 中的 BUSY 位置 1。
4. 密钥传输完成后，CRYP_SR 寄存器中的 BUSY 位清零，KEYVALID 位置 1。如果在 BUSY 位清零时 KEYVALID 未置 1，或者 CRYP_SR 中的密钥错误标志 (KERF) 置 1，则表示尚未使用 ALGOMODE 选择 AES 算法、已选择错误的 KEYSIZE 或者传输期间已经发生意外事件（DPA 错误、Tamper 事件或 SAES_SR 中的 KEYVALID 位在传输结束前被清零）。当出现此类错误时，从 SAES 外设开始的整个密钥共享过程必须通过两个控制寄存器的 IPRST 位重新启动。有关密钥错误标志的更多信息，请参见*第 34.4.21 节*。

当该步骤完成时，CRYP 将完成初始化，并拥有一个有效的共享密钥。应用随后可在 CRYP_CR 寄存器中的 KMOD 位设为 00 后继续处理数据。

小心： 当密钥寄存器已初始化为共享密钥后，应避免将 KEYSIZE 或 ALGOMODE 更改为 DES 或 TDES 算法，否则将导致密钥数据丢失并引发 KERF 错误。

34.4.17 CRYP 数据寄存器和数据交换

CRYP_DIN 寄存器是外设的 32 位宽数据输入寄存器。最多可将四个 64 位 (TDES) 或两个 128 位 (AES) 明文（加密时）或密文（解密时）块输入到输入 FIFO 中，一次输入一个 32 位字。

AES（或 DES/TDES）数据块的四个相邻（或两个）字使用大端模式进行组织，即字的最高有效字节存储在最低地址中。

可将密码加速器配置为在将其加载到 CRYP 处理内核前对 CRYP_DINR 寄存器中的输入数据字，以及在将其发送到 CRYP_DOUTR 寄存器前对 CRYP 处理内核的输出数据，执行位、字节、半字交换或不交换。选择取决于数据类型。例如，对 ASCII 文本流使用字节交换。

通过 AES_CR 寄存器中的 DATATYPE[1:0] 位字段来选择数据交换类型。该选择适用于 CRYP 处理内核的输入和输出。

注： *CRYP 密钥寄存器 (CRYP_Kx(L/R)) 和初始化寄存器 (CRYP_IVx(L/R)) 对所选交换模式不敏感。*

有关数据输入和数据交换的更多信息，可参见下一小节。

DES/TDES 数据输入和输出

通过将两个连续的 32 位字写入 CRYP_DINR 寄存器（DIN[31:0] 位字段），将 64 位数据块输入到密码处理器，先最高有效字（位 [64:33]），后最低有效字（位 [32:1]）。

通过从 CRYP_DOUTR 寄存器（DOUT[31:0] 位字段）读取两个连续的 32 位字，从密码处理器检索 64 位数据块，先最高有效字（位 [64:33]），后最低有效字（位 [32:1]）。

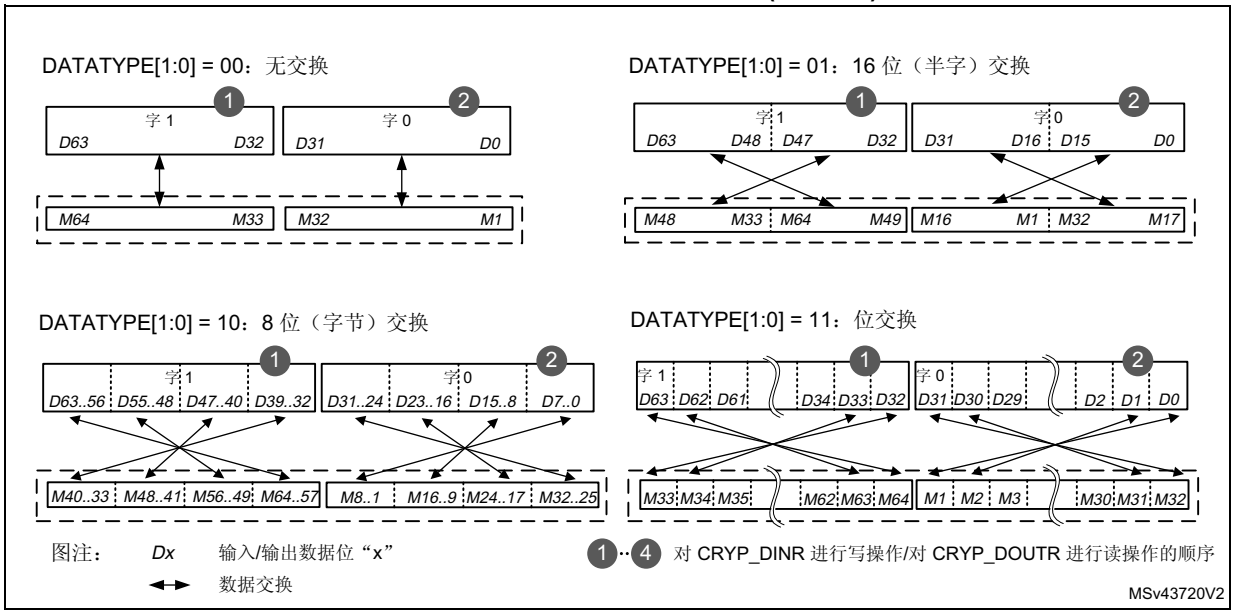
DES/TDES 数据交换功能

表 287 和*图 327* 总结了 DES/TDES 的密码处理器数据交换功能。

表 287. DES/TDES 数据交换示例

CRYP_CR 中的 DATATYPE	要执行的交换	数据块表示（64 位） 系统内存数据（大端）
00	无交换	块[64..1]：0xABCD7720 6973FE01
		地址 @, 字[63..32]：0xABCD 7720 地址 @+4, 字[31..0]：0x6973 FE01
01	半字（16 位）交换	块[64..1]：0x**ABCD** 7720 **6973** FE01
		地址 @, 字[63..32]：0x7720 **ABCD** 地址 @+4, 字[31..0]：0xFE01 **6973**
10	字节（8 位）交换	块[64..1]：0x**AB** CD **77** 20 **69** 73 **FE** 01
		地址 @, 字[63..32]：0x20**77** CD**AB** 地址 @+4, 字[31..0]：0x01**FE** 73**69**
11	32 位字中的位交换	块[64..33]：0xABCD7720 1010 1011 1100 1101 0111 0111 0010 0000 块[32..1]：0x6973FE01 0110 1001 0111 0011 1111 1110 0000 0001
		地址 @, 字[63..32]：0x04EEB3D5 0000 0100 1110 1110 1011 0011 1101 0101 地址 @+4, 字[31..0]：0x807FCE96 1000 0000 0111 1111 1100 1110 1001 0110

图 327. 根据数据类型构建 64 位块 (IN FIFO)



AES 数据输入和输出

通过将四个连续的 32 位字写入 CRYP_DINR 寄存器（位字段 DIN[31:0]），将 128 位数据块输入到密码处理器，先最高有效字（位 [127:96]），后最低有效字（位 [31:0]）。

通过从 CRYP_DOUTR 寄存器（位字段 DOUT[31:0]）读取四个连续的 32 位字，从密码处理器检索 128 位数据块，先最高有效字（位 [127:96]），后最低有效字（位 [31:0]）。

AES 数据交换功能

表 288 和*图 328* 总结了 AES 的密码处理器数据交换功能。

注： *数据交换不适用于 GCM 和 CCM 最后阶段。应用程序可以正常插入数据。*

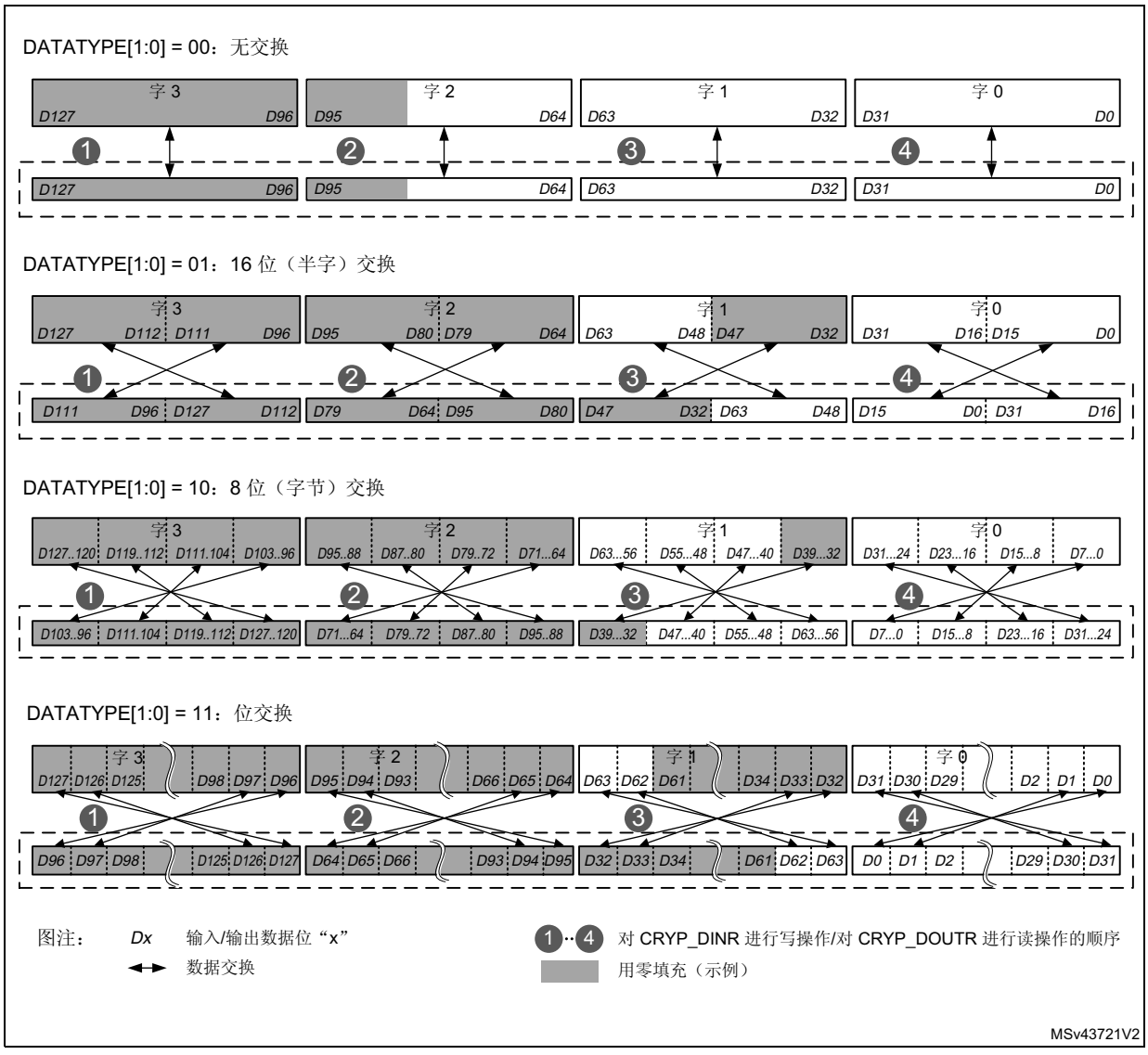
表 288. AES 数据交换示例

CRYP_CR 中的 DATATYPE	要执行的交换	前半部分数据块（64 位）
		系统内存数据（大端）
00	无交换	块[63..0]：0x4E6F7720 69732074
		地址 @, 字[63..32]：0x4E6F7720 地址 @+4, 字[31..0]：0x69732074
01	半字（16 位）交换	块[63..0]：0x**4E6F** 7720 **6973** 2074
		地址 @, 字[63..32]：0x7720 **4E6F** 地址 @+4, 字[31..0]：0x2074 **6973**
10	字节（8 位）交换	块[63..0]：0x**4E** 6F **77** 20 **69** 73 **20** 74
		地址 @, 字[63..32]：0x20**77** 6F**4E** 地址 @+4, 字[31..0]：0x74**20** 73**69**

表 288. AES 数据交换示例（续）

CRYP_CR 中的 DATATYPE	要执行的交换	前半部分数据块（64 位）
		系统内存数据（大端）
11	位交换	块[63..32]：0x4E6F7720 0100 1110 0110 1111 0111 0111 0010 0000 块[31..0]：0x69732074 0110 1001 0111 0011 0010 0000 0111 0100
		地址 @, 字[63..32]：0x04EE F672 0000 0100 1110 1110 1111 0110 0111 0010 地址 @+4, 字[31..0]：0x2E04 CE96 0010 1110 0000 0100 1100 1110 1001 0110

图 328. 根据数据类型构建 128 位块



34.4.18 CRYP 密钥寄存器

CRYP_Kx 寄存器为只写寄存器，用于存储加密或解密密钥。总共划分为四个 64 位寄存器，如*表 289* 和*表 290* 所示。

注： *在内存和 CRYP 密钥寄存器中，AES 和 DES/TDES 密钥以大端模式进行存储，即最高有效字节存储在最低地址中。*

表 289. CRYP_KxR/LR 寄存器中的密钥字节序（AES 128/192/256 位密钥）

K0LR[31:0]	K0RR[31:0]	K1LR[31:0]	K1RR[31:0]	K2LR[31:0]	K2RR[31:0]	K3LR[31:0]	K3RR[31:0]
-	-	-	-	k[127:96]	k[95:64]	k[63:32]	k[31:0]
K0LR[31:0]	**K0RR[31:0]**	**K1LR[31:0]**	**K1RR[31:0]**	**K2LR[31:0]**	**K2RR[31:0]**	**K3LR[31:0]**	**K3RR[31:0]**
-	-	k[191:160]	k[159:128]	k[127:96]	k[95:64]	k[63:32]	k[31:0]
K0LR[31:0]	**K0RR[31:0]**	**K1LR[31:0]**	**K1RR[31:0]**	**K2LR[31:0]**	**K2RR[31:0]**	**K3LR[31:0]**	**K3RR[31:0]**
k[255:224]	k[223:192]	k[191:160]	k[159:128]	k[127:96]	k[95:64]	k[63:32]	k[31:0]

表 290. CRYP_KxR/LR 寄存器中的密钥字节序（DES K1 和 TDES K1/2/3）

K0LR[31:0]	K0RR[31:0]	K1LR[31:0]	K1RR[31:0]	K2LR[31:0]	K2RR[31:0]	K3LR[31:0]	K3RR[31:0]
-	-	K1[64:33]	K1[32:1]	-	-	-	-
-	-	K1[64:33]	K1[32:1]	K2[64:33]	K2[32:1]	K3[64:33]	K3[32:1]

如*表 290* 所示，当选择 TDES（ALGOMODE[2:0] = 000 或 001）时，使用 3 密钥向量（K1、K2 和 K3）。当选择 DES（ALGOMODE[2:0] = 010 或 011）时，仅使用 1 密钥向量 (K1)。

注： *DES/TDES 密钥包含密码处理器不使用的 8 位奇偶校验信息。换言之，不使用各个 64 位密钥值 Kx[1:64] 的位 8、16、24、32、40、48、56 和 64。*

当 CRYP_SR 寄存器的 BUSY 位置 1 时，针对 CRYP_Kx(L/R) 寄存器的写操作会被忽略（即，寄存器内容不会被修改）。因此，应用程序必须先确定 BUSY 位已清零后再修改密钥寄存器。

密钥寄存器不受数据交换功能（由 CRYP_CR 寄存器中 DATATYPE 值控制）的影响。

如果使用 CRYP_SR 寄存器中的 KEYVALID 状态位，应用程序始终需要先写入整个密钥，再通过将 CRYP_CR 中的 CRYPEN 位置 1 使能密码处理器开始计算。更具体来讲，在正常密钥模式 (KMOD = 00) 下，正确的密钥写入序列如下：

- 对于 DES/TDES 算法：
  - 如果 ALGOMODE = 0000 或 0001：必须依序写入 K1LR、K1RR、K2LR、K2RR、K3LR 和 K3RR 寄存器，或者逆序写入这些寄存器。
  - 如果 ALGOMODE = 0010 或 0011：必须先写入 K1LR 再写入 K1RR，或者先写入 K1RR 再写入 K1LR。

- 对于与 AES 算法相对应的任何 ALGOMODE：
  - 如果 KEYSIZE = 00：必须依序写入 K2LR、K2RR、K3LR 和 K3RR 寄存器，或者逆序写入这些寄存器。
  - 如果 KEYSIZE = 01：必须依序写入 K1LR、K1RR、K2LR、K2RR、K3LR 和 K3RR 寄存器，或者逆序写入这些寄存器。
  - 如果 KEYSIZE = 10：必须依序写入 K0LR、K0RR、K1LR、K1RR、K2LR、K2RR、K3LR 和 K3RR 寄存器，或者逆序写入这些寄存器。

注： *必须在写入密钥之前写入 CRYP_CR 中的 KEYSIZE 和 ALGOMODE。*

对于上述所有流程，一旦写入第一个寄存器，KEYVALID 位就会清零。当写入流程完成后，KEYVALID 置 1，CRYPEN 位变为可写。如果发生错误，KEYVALID 清零，KERF 标志置 1（有关详细信息，请参见*第 34.4.21 节：CRYP 错误管理*）。

有关 CRYP_Kx(L/R) 寄存器的详细说明，请参见*第 34.7 节：CRYP 寄存器*。

34.4.19 CRYP 初始化向量寄存器

CRYP_IVxL/IVxR 寄存器用于存储初始化向量或随机值，具体取决于所选链接模式。在使用过程中，完成每一轮的 TDES 或 AES 内核计算后，内核都会更新这些寄存器。

总共划分为四个 64 位寄存器，如*表 291* 和*表 292* 所示。在 DES/TDES 模式下，仅使用 CRYP_IV0x。

注： *在内存和 CRYP IV 寄存器中，AES 和 DES/TDES 初始化向量以大端模式进行存储，即最高有效字节存储在最低地址中。*

表 291. CRYP_IVx(L/R)R 寄存器中的初始化向量字节序 (AES)

CRYP_IV0LR[31:0]	CRYP_IV0RR[31:0]	CRYP_IV1LR[31:0]	CRYP_IV1RR[31:0]
IVI[127:96]	IVI[95:64]	IVI[63:32]	IVI[31:0]

表 292. CRYP_IVx(L/R)R 寄存器中的初始化向量字节序 (DES/TDES)

CRYP_IV0LR[31:0]	CRYP_IV0RR[31:0]	CRYP_IV1LR[31:0]	CRYP_IV1RR[31:0]
IVI[64:32]	IVI[31:0]	-	-

当 CRYP_SR 寄存器的 BUSY 位置 1 时，针对 CRYP_IV0...1(L/R)R 寄存器的任何写操作都会被忽略（即，寄存器内容不会被修改）。因此，应用程序必须先确定 CRYP_SR 寄存器中的 BUSY 位已清零后再修改初始化向量。

读取 CRYP_IV0...1(L/R)R 寄存器会返回最新计数器值（对于管理挂起模式很有用，CCM/GCM 除外）。

初始化向量寄存器不受数据交换功能（由 CRYP_CR 寄存器中 DATATYPE 值控制）的影响。

有关 CRYP_IVxL/IVxR 寄存器的详细说明，请参见*第 34.7 节：CRYP 寄存器*。

34.4.20 CRYP DMA 接口

密码处理器可使用一个接口连接 DMA（直接内存访问）控制器。DMA 操作通过 CRYP DMA 控制寄存器 (CRYP_DMACR) 进行控制。

注： *当选择 DES/TDES 算法时，无法使用 DMA。*

使用 DMA 的数据输入

可通过将 CRYP_DMACR 寄存器中的 DIEN 位置 1 使能 DMA，从而向密码外设中写入数据。如果该位置 1，则密码处理器每次需要向 CRYP_DIN 寄存器中写入字时，都会在 INPUT 阶段发出 DMA 请求。

表 293 给出了通过 DMA 控制器将数据从内存传输到密码处理器时的推荐配置。

表 293. 针对内存到外设的 DMA 传输的密码处理器配置

DMA 通道控制寄存器字段	编程建议
传输数据量大小	消息长度，AES 密码块大小（4 个字）的倍数。 *注意：DMA 不支持 DES/TDES。* 根据所选模式，可能需要采用特殊填充/密文窃取技术。例如，在 AES GCM 加密或 AES CCM 解密情况下，DMA 传输不得包括最后一个块。有关详细信息，请参见*第 34.4.5 节：执行加密操作的 CRYP 过程*。
源突发大小（内存）	如果使用高延迟内存 (DDR)： CRYP FIFO_size x2 /transfer_width = **16** （FIFO_size = 8x32 位，transfer_width = 32 位） 如果使用低延迟内存 (SRAM)： CRYP FIFO_size /2 /transfer_width = **4**
目标突发大小（外设）	CRYP FIFO_size /2 /transfer_width = **4** （FIFO_size = 8x32 位，transfer_width = 32 位）
DMA FIFO 大小	如果使用高延迟内存 (DDR)：CRYP FIFO_size x2 = 64 字节 如果使用低延迟内存 (SRAM)：CRYP FIFO_size /2 = 16 字节
源传输宽度（内存）	32 位字
目标传输宽度（外设）	32 位字
源地址递增（内存）	是，每次 32 位传输后。
目标地址递增（外设）	应使用 CRYP_DIN 的固定地址（无递增）。

使用 DMA 的数据输出

要使能 DMA 以从 CRYP 处理器读取数据，请将 CRYP_DMACR 寄存器中的 DOEN 位置 1。如果该位置 1，则密码处理器每次需要从 CRYP_DOUT 寄存器中读取字时，都会在 OUTPUT 阶段发出 DMA 请求。

表 294 给出了通过 DMA 控制器将数据从密码处理器传输到内存时的推荐配置。

表 294. 针对外设到内存的 DMA 传输的密码处理器配置

DMA 通道控制寄存器字段	编程建议
传输数据量大小	消息长度，AES 密码块大小（4 个字）的倍数。 *注意：DMA 不支持 DES/TDES。* 根据所选的链接模式，必须丢弃额外的位。
源突发大小（外设）	CRYP FIFO_size /2 /transfer_width = **4** （FIFO_size = 8x32 位，transfer_width = 32 位）
目标突发大小（内存）	如果使用*高延迟*内存 (DDR)： CRYP FIFO_size x2 /transfer_width = **16** （FIFO_size = 8x32 位，transfer_width = 32 位） 如果使用*低延迟*内存 (SRAM)： CRYP FIFO_size /2 /transfer_width = **4**
DMA FIFO 大小	如果使用高延迟内存 (DDR)：CRYP FIFO_size x2 = 64 字节 如果使用低延迟内存 (SRAM)：CRYP FIFO_size /2 = 16 字节
源传输宽度（外设）	32 位字
内存传输宽度（内存）	32 位字
源地址递增（外设）	应使用 CRYP_DOUT 的固定地址（无递增）。
目标地址递增（内存）	是，每次 32 位传输后。

DMA 模式

使用 AES 时，密码处理器通过 `cryp_in_dma` 和 `cryp_out_dma` 内部输入/输出信号管理两个 DMA 传输请求，请求信号的有效条件如下：

- 对于 IN FIFO：每次 CRYP 从 FIFO 中读取一个块时，
- 对于 OUT FIFO：每次密码处理器将一个块写入 FIFO 中时。

使用 DES/TDES 时，密码处理器通过 `cryp_in_dma` 和 `cryp_out_dma` 内部输入/输出信号管理两个 DMA 传输请求，请求信号的有效条件如下：

- 对于 IN FIFO：每次密码处理器从 FIFO 中读取两个块时，

对于 OUT FIFO：每次密码处理器将一个字写入 FIFO 中（单次传输）时。请注意，如果已将两个块写入 FIFO，则也会触发突发传输。

注：　　当选择 DES/TDES 算法时，无法使用 DMA。

如果已禁止密码外设或已将 DMA 使能位清零（CRYP_DMACR 寄存器中的 DIEN 位用于 IN FIFO，DOEN 位用于 OUT FIFO），则会禁止所有请求信号。

小心： 在填满 CRYP 输入 FIFO 之前，由 DMA 控制器清空密码外设输出 FIFO 至关重要。为此，应对 DMA 控制器进行相应配置，以使从外设到内存的传输优先级高于从内存到外设的传输。

有关 DMA 操作的更多详细信息，请参见*第 34.4.5 节：执行加密操作的 CRYP 过程*。

34.4.21 CRYP 错误管理

密钥错误标志 (KERF)

可能导致 CRYP_SR 寄存器中的密钥错误标志 (KERF) 置 1 的错误来源有以下三种：

1. 密钥写入序列错误：写入密钥寄存器时检测到不正确的序列。有关详细信息，请参见*第 34.4.18 节：CRYP 密钥寄存器*。
2. 密钥共享错误：将密钥寄存器从 SAES 外设复制到 CRYP 失败。有关详细信息，请参见*第 34.4.16 节*。
3. 密钥共享大小不匹配：当 KMOD 设置为 10 并且用户应用程序在 CRYP 处理器中设置的 KEYSIZE 信息与存储在 SAES 中的 KEYSIZE 不匹配时，或者向 CRYP_CR 中的 KEYSIZE 写入 01（192 位密钥）时，将触发 KERF。
4. 在 KMOD = 10（共享密钥模式）时选择了 DES 或 TDES

在第一种情况下，当正确的密钥写入流程开始时，KERF 位清零。在第二种和第三种情况下，通过将 CRYP_CR 寄存器中的 IPRST 位先置 1 再清零来清零 KERF 位。

注： *必须清零 KERF 错误标志才能将 CRYP_CR 中的 CRYPEN 置 1。*

当 KERF 置 1 时，会自动清零 KEYVALID 位。

34.5 CRYP 中断

概述

密码处理器可生成两个独立的可屏蔽中断源，用以通知发生以下事件：

- 输入 FIFO 为空或未满
- 输出 FIFO 已满或非空

这两个中断源合为同一个中断信号，而该中断信号是 CRYP 处理器发出的唯一中断信号，用于驱动 Cortex® CPU 中断控制器。通过更改 CRYP_IMSCR 寄存器中的屏蔽位，可单独使能或禁止各个 CRYP 中断源。将相应的屏蔽位置 1 可使能中断。

关于各个可屏蔽中断源的状态，通过 CRYP_RISR 寄存器可以读取原始中断状态，通过 CRYP_MISR 寄存器可以读取屏蔽中断状态。可以从 CRYP_SR 寄存器中读取各个事件标志源的状态。

表 295 对可用功能进行了总结。

表 295. CRYP 中断请求

中断缩写	中断事件	事件标志		使能位	中断清除方法
		在 CRYP_xISR 中(1)	在 CRYP_SR 中		
CRYP	输出 FIFO 已满	OUTRIS（未屏蔽） OUTMIS（已屏蔽）	OFFU	OUTIM(2) 和 CRYPEN	从输出 FIFO 中读取一个数据
	输出 FIFO 非空		OFNE		从输出 FIFO 中读取所有数据
	输入 FIFO 未满	INRIS（未屏蔽） INMIS（已屏蔽）	IFNF	INIM(2) 和 CRYPEN	写入数据直到输入 FIFO 已满
	输入 FIFO 为空		IFEM		在输入 FIFO 中至少写入一个字

1. 这些标志属于 CRYP_RISR 寄存器（未屏蔽或原始）或 CRYP_MISR 寄存器（已屏蔽）。
2. 这些标志属于 CRYP_IMSCR 寄存器。

输出 FIFO 服务中断 - OUTMIS

当输出 FIFO 中存在一个或多个（32 位字）数据项时，即会产生输出 FIFO 服务中断。通过读取输出 FIFO 的数据，直到读完所有有效（32 位）字（即，该中断的状态与输出 FIFO 非空标志 OFNE 一致时），即可将此中断清除。

输出 FIFO 服务中断 OUTMIS 不是通过 CRYP 处理器使能位使能。因此，如果输出 FIFO 非空，即使禁止 CRYP 处理器之后也不会强制 OUTMIS 信号为低电平。

输入 FIFO 服务中断 - INMIS

当输入 FIFO 中少于四个字时，会产生输入 FIFO 服务中断。对输入 FIFO 执行写操作直到其中所含字不小于四字，这样即可将此中断清除。

输入 FIFO 服务中断 INMIS 通过 CRYP 使能位使能。因此，禁止 CRYP 之后，即使输入 FIFO 为空，INMIS 信号也为低（无效）。

34.6 CRYP 处理时间

下表汇总了每种工作模式下处理一个块需要的时间。块大小为 128 位（对于 AES）或 64 位（对于 DES/TDES）。

表 296. ECB、CBC 和 CTR 模式下的处理延迟

密钥大小	工作模式	链接算法	时钟周期
64 位	DES 加密或解密	ECD、CBC	16
3x64 位	TDES 加密或解密		48
128 位	AES 加密或解密(1)	ECD、CBC、CTR	14
	AES 密钥准备	-	12

表 296. ECB、CBC 和 CTR 模式下的处理延迟（续）

192 位	AES 加密或解密(1)	ECD、CBC、CTR	16
	AES 密钥准备	-	14
256 位	AES 加密或解密(1)	ECD、CBC、CTR	18
	AES 密钥准备	-	16

1. 不包括密钥准备时间（仅限 ECB 和 CBC）。

表 297. GCM 和 CCM 模式下每个 128 位块的处理时间（以时钟周期计）

密钥大小	工作模式	链接算法	初始化阶段	标头阶段	有效负载阶段	标记阶段
128 位	AES 加密或解密	GCM	24	10	14	14
		CCM	12	14	25	14
192 位		GCM	28	10	16	16
		CCM	14	16	29	16
256 位		GCM	32	10	18	18
		CCM	16	18	33	18

34.7 CRYP 寄存器

密码内核连接若干控制和状态寄存器、八个密钥寄存器和四个初始化向量寄存器。

34.7.1 CRYP 控制寄存器 (CRYP_CR)

CRYP control register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPRST	Res.	Res.	Res.	Res.	Res.	KMOD1	KMOD0	NPBLB[3:0]				ALGOMODE3	Res.	GCM_CCMPH[1:0]	
rw						rw	rw	rw	rw	rw	rw	rw		rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRYPEN	FFLUSH	Res.	Res.	Res.	Res.	KEYSIZE[1:0]		DATATYPE[1:0]		ALGOMODE[2:0]			ALGODIR	Res.	Res.
rw	rw					rw	rw	rw	rw	rw	rw	rw	rw		

位 31 **IPRST**：CRYP 软件复位 (CRYP software reset)

将该位置 1 会将 CRYP 处理器复位，从而将所有寄存器复位为其默认值（IPRST 位本身除外），

写入任何配置寄存器时，该位必须保持清零。

位 30:26 保留，必须保持复位值。

位 25:24 **KMOD[1:0]**：密钥模式选择 (Key mode selection)

该位字段定义应用程序如何使用 CRYP 密钥：

00：正常密钥模式。可以自由使用密钥寄存器。

10：共享 AES 密钥模式。在 AES 模式下，CRYP 自动将 SAES 协处理器的密钥寄存器中存储的数据加载到其密钥寄存器中。

其他值：保留，不得使用。

当 KMOD[1:0] = 10 时，所选的密钥值会在 SAES_SR 寄存器中的 KEYVALID 置 1 时在密钥寄存器中可用。否则，CRYP_SR 寄存器中的密钥错误标志 KERF 置 1。

仅当 KEYSIZE 正确并且 SAES 协处理器已成功完成共享密钥解密流程时，才必须将 KMOD[1:0] 设置为 10。否则，CRYP_SR 中的密钥错误标志 KERF 置 1。

BUSY = 1 时写入 KMOD 位无影响。

位 23:20 **NPBLB[3:0]**：有效负载的最后一个块中填充字节的数量 (Number of Padding Bytes in Last Block of payload)。

在加密 GCM 有效负载的最后一个块或解密 CCM 有效负载的最后一个块之前，必须通过软件填充该填充信息，否则验证标记计算不正确。

0000：所有字节均有效（无填充）

0001：填充最后一个 LSB 字节

...

1111：填充最后一个块的 15 个 LSB 字节

BUSY = 1 时写入 NPBLB 位无影响。

位 18 保留，必须保持复位值。

位 17:16 **GCM_CCMPH[1:0]**：GCM 或 CCM 阶段选择 (GCM or CCM Phase selection)

如果未在 ALGOMODE 位字段中选择 GCM、GMAC 或 CCM 算法，则该位字段无影响。

00：初始化阶段

01：标头阶段

10：有效负载阶段

11：最后阶段

BUSY = 1 时写入 GCM_CCMPH 位无影响。

位 15 **CRYPEN**：CRYP 处理器使能 (CRYP processor Enable)

0：禁止密码处理器外设

1：使能密码处理器外设

当密钥准备过程结束 (ALGOMODE= 0111) 时或者在 GCM/GMAC 或 CCM 初始化阶段后，硬件会自动将该位清零。

位 14 **FFLUSH**：CRYP FIFO 刷新 (CRYP FIFO Flush)

0：禁止 FIFO 刷新

1：使能 FIFO 刷新

CRYPEN = 0 时，在该位中写入 1 可刷新 IN 和 OUT FIFO（即，将复位 FIFO 的读指针和写指针）。若在该位中写入 0 则不会产生影响。CRYPEN = 1 时，在该位中写入 0 或 1 都不会产生影响。

读取该位时，始终返回 0。

FFLUSH 位必须在 BUSY = 0 时才能置 1。否则，虽然 FIFO 会进行刷新，但在刷新操作后可能会将所处理的块压入输出 FIFO，从而产生 FIFO 非空的情况。

位 13:10 保留，必须保持复位值。

位 9:8 **KEYSIZE[1:0]**：密钥大小选择 (Key size selection)（仅限 AES 模式）

该位字段定义了 AES 密码内核使用的密钥位长度。该位字段在 DES 或 TDES 模式中可忽略。

00：128 位密钥长度

01：192 位密钥长度

10：256 位密钥长度

11：保留，不使用该值

BUSY = 1 时写入 KEYSIZE 位无影响。

当 KEYSIZE 发生变化时，KEYVALID 清零。

位 7:6 **DATATYPE[1:0]**：数据类型选择 (Data Type selection)

该位字段定义写入 CRYP_DIN 寄存器或从 CRYP_DOUT 寄存器读取的数据的格式。更多详细信息，请参见*第 34.4.17 节：CRYP 数据寄存器和数据交换*。

00：32 位数据。不交换各个字。压入 IN FIFO（或 OUT FIFO 弹出）的首个字形成数据块的位 1...32，第二个字形成数据块的位 33...64。

01：16 位数据或半字。可将压入 IN FIFO（或 OUT FIFO 弹出）的每个字视为 2 个已相互交换的半字。

10：8 位数据或字节。可将压入 IN FIFO（或 OUT FIFO 弹出）的每个字视为 4 个已相互交换的字。

11：位数据或位串。可将压入 IN FIFO（或 OUT FIFO 弹出）的每个字视为 32 个已相互交换的位（字符串的首个位在位置 0）。

BUSY = 1 时写入 DATATYPE 位无影响。

位 19 和 5:3 **ALGOMODE[3:0]**：算法模式 (Algorithm mode)

以下定义包括位 19：

0000：TDES-ECB（TDES 电子密码本）。

0001：TDES-CBC（TDES 密码分组链接）。

0010：DES-ECB（简易 DES 电子密码本）。

0011：DES-CBC（简易 DES 密码分组链接）。

0100：AES-ECB（AES 电子密码本）。

0101：AES-CBC（AES 密码分组链接）。

0110：AES-CTR（AES 计数器模式）。

0111：针对 ECB 或 CBC 解密准备 AES 密钥。

1000：AES-GCM（Galois 计数器模式）和 AES-GMAC（Galois 消息认证码模式）。

1001：AES-CCM（CBC-MAC 计数器模式）。

BUSY = 1 时写入 ALGOMODE 位无影响。

其他值：保留，不得使用

将 ALGOMODE 从 AES 算法更改为 DES 或 TDES（或反之）会清零 KEYVALID。当 ALGOMODE 更改前后的算法类型保持不变时，即在 TDES-ECB 与 TDES-CBC 之间或 AES-ECB 与 AES_CTR 之间更改时，KEYVALID 不会清零。

位 2 **ALGODIR**：算法方向 (Algorithm Direction)

0：加密

1：解密

BUSY = 1 时写入 ALGODIR 位无影响。

位 1:0 保留，必须保持复位值。

34.7.2 CRYP 状态寄存器 (CRYP_SR)

CRYP status register

偏移地址：0x04

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY VALID	KERF	Res.	BUSY	OFFU	OFNE	IFNF	IFEM
								r	r		r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7 **KEYVALID**：密钥有效 (Key valid)

KEYVALID 为只读位。当 CRYP_KEYx 密钥寄存器中载入了预期的密钥信息时，该位由硬件置 1。该位对 AES 和 DES/TDES 算法均有效。

0：密钥寄存器中没有有效的密钥信息可用。CRYP_CR 中的 CRYPEN 位无法置 1。

1：由 CRYP_CR 中的 KEYSIZE 所定义的有效密钥信息加载到密钥寄存器中。

当 KMOD = 00 时，用户应用程序必须按正确的顺序写入密钥寄存器，否则 KERF 标志会置 1，KEYVALID 保持为零。

当 KMOD = 10（AES 共享密钥模式）时，KEYVALID 置 1 表示共享密钥成功从 SAES 加载到 CRYP 中。发生错误时，CRYP_SR 中的 KERF 标志置 1，KEYVALID 保持为零。

位 6 **KERF**：密钥错误标志 (Key error flag)

KERF 为只读位。如果密钥信息加载到密钥寄存器失败，该位由硬件置 1。

0：未检测到密钥错误

1：密钥信息加载到密钥寄存器失败

如果发生以下错误，KERF 会置 1：

- 用户应用程序未按正确的顺序写入密钥寄存器（有关详细信息，请参见*第 34.4.18 节：CRYP 密钥寄存器*）。
- CRYP 加载 SAES 外设共享的 AES 密钥 (KMOD = 10) 失败。
- 当 KMOD = 10 时，使用 ALGOMODE 选择 DES 或 TDES 算法。

KERF 必须由应用软件清零，否则无法将 KEYVALID 置 1。KERF 位可通过 CRYP_CR 的 IPRST 位清零，或者在正确的密钥写入流程开始时清零。

位 5 保留，必须保持复位值。

位 4 **BUSY**：繁忙位 (Busy bit)

0：CRYP 内核当前没有处理任何数据。原因是：

- CRYP 内核已禁止（CRYP_CR 寄存器的 CRYPEN = 0）且最后的处理过程已完成，
- 或者 CRYP 内核正在等待输入 FIFO 的数据足够多或输出 FIFO 的自由空间足够大（即，在这两种情况下，DES 中至少有 2 个字，AES 中至少有 4 个字）。

1：CRYP 内核目前正在处理一个数据块或正在准备密钥（仅限 AES ECB 或 CBC 解密）。当 AES 密钥正在从 SAES 传输到 CRYP 时，BUSY 也会置 1。

位 3 **OFFU**：输出 FIFO 已满标志 (Output FIFO full flag)

0：输出 FIFO 未满

1：输出 FIFO 已满

位 2 **OFNE**：输出 FIFO 非空标志 (Output FIFO not empty flag)
0：输出 FIFO 为空
1：输出 FIFO 非空

位 1 **IFNF**：输入 FIFO 未满标志 (Input FIFO not full flag)
0：输入 FIFO 已满
1：输入 FIFO 未满

位 0 **IFEM**：输入 FIFO 为空标志 (Input FIFO empty flag)
0：输入 FIFO 非空
1：输入 FIFO 为空

34.7.3 CRYP 数据输入寄存器 (CRYP_DIN)

CRYP data input register

偏移地址：0x08

复位值：0x0000 0000

CRYP_DIN 寄存器为数据输入寄存器。其宽度为 32 位。最多可将四个 64 位 (TDES) 或两个 128 位 (AES) 明文（加密时）或密文（解密时）块输入到输入 FIFO 中，一次输入一个 32 位字。

为了满足不同的数据大小，通过配置 CRYP_CR 寄存器中的 DATATYPE 位，数据可在处理之后进行交换。更多详细信息，请参见*第 34.4.17 节：CRYP 数据寄存器和数据交换*。

写入 CRYP_DIN 寄存器时，数据会压入输入 FIFO。

- 如果 CRYPEN = 1，则在 AES 模式下至少有四个 32 位字（或者 DES/TDES 模式下至少有两个字）已压入输入 FIFO 中，并且输出 FIFO 中至少有四个字的自由空间时（或者 DES/DTES 模式下，至少有两个字的自由空间），CRYP 引擎会启动加密或解密过程。

读取 CRYP_DIN 寄存器时：

- 如果 CRYPEN = 0，则 FIFO 会弹出数据，之后，返回输入 FIFO 中的数据，从最早的数据（首先读取）到最新的数据（最后读取）。在执行每次读取操作前必须检查 IFEM 标志，以确保 FIFO 非空。
- 如果 CRYPEN = 1，则会返回一个未定义值。

注： *读取 CRYP_DIN 寄存器一次或多次之后，在处理新数据之前必须通过将 FFLUSH 位置 1 来刷新 FIFO。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATAIN[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAIN[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DATAIN[31:0]**：数据输入
读取时，FIFO 会弹出数据（返回最后写入的值），如果 CRYPEN = 0 则会返回相应值。如果 CRYPEN = 1，则 DATAIN 寄存器会返回一个未定义值。
写入时，会将当前寄存器内容压入 FIFO 内。

34.7.4 CRYP 数据输出寄存器 (CRYP_DOUT)

CRYP data output register

偏移地址：0x0C

复位值：0x0000 0000

CRYP_DOUT 寄存器为数据输出寄存器。该寄存器为 32 位宽度的只读寄存器。最多可接收来自输出 FIFO 的四个 64 位 (TDES) 或两个 128 位 (AES) 明文（加密时）或密文（解密时）块，一次接收一个 32 位字。

为了满足不同的数据大小，通过配置 CRYP_CR 寄存器中的 DATATYPE 位，数据可在处理之后进行交换。更多详细信息，请参见*第 34.4.17 节：CRYP 数据寄存器和数据交换*。

读取 CRYP_DOUT 寄存器时，会返回最后输入输出 FIFO 的数据（由读指针指定）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATAOUT[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAOUT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **DATAOUT[31:0]**：数据输出 (Data output)
读取时，返回输出 FIFO 的内容（读指针所指向的内容），否则会返回一个未定义的值。
写入时，无影响。

34.7.5 CRYP DMA 控制寄存器 (CRYP_DMACR)

CRYP DMA control register

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOEN	DIEN
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **DOEN**：DMA 输出使能 (DMA output enable)
该位置 1 时，外设会在输出数据阶段自动生成 DMA 请求。
0：禁止通过 DMA 从 CRYP 传输传出数据
1：允许通过 DMA 从 CRYP 传输传出数据

位 0 **DIEN**：DMA 输入使能 (DMA input enable)
该位置 1 时，外设会在输入数据阶段自动生成 DMA 请求。
0：禁止通过 DMA 向 CRYP 传输传入数据
1：允许通过 DMA 向 CRYP 传输传入数据

34.7.6 CRYP 中断屏蔽置 1/清零寄存器 (CRYP_IMSCR)

CRYP interrupt mask set/clear register

偏移地址：0x14

复位值：0x0000 0000

CRYP_IMSCR 寄存器为中断屏蔽置 1/清零寄存器。该寄存器为读/写寄存器。执行读取操作时，此寄存器会给出应用于相关中断屏蔽的当前值。向该特定位写入 1 时会将屏蔽置 1，从而使能要读取的中断。向该位写入 0 时会清除相应的屏蔽。外设复位后，所有位均清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OUTIM	INIM
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **OUTIM**：输出 FIFO 服务中断屏蔽 (Output FIFO service interrupt mask)
0：屏蔽输出 FIFO 服务中断
1：不屏蔽输出 FIFO 服务中断

位 0 **INIM**：输入 FIFO 服务中断屏蔽 (Input FIFO service interrupt mask)
0：屏蔽输入 FIFO 服务中断
1：不屏蔽输入 FIFO 服务中断

34.7.7 CRYP 原始中断状态寄存器 (CRYP_RISR)

CRYP raw interrupt status register

偏移地址：0x18

复位值：0x0000 0001

CRYP_RISR 寄存器为原始中断状态寄存器。该寄存器为只读寄存器。执行读操作时，该寄存器会给出相应中断的当前原始状态，即中断信息（不考虑 CRYP_IMSCR 屏蔽）。写操作无影响。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OUTRIS	INRIS
														r	r

位 31:2 保留，必须保持复位值。

位 1 **OUTRIS**：输出 FIFO 服务原始中断状态 (Output FIFO service raw interrupt status)
该位提供输出 FIFO 中断信息（不考虑相应 CRYP_IMSCR 屏蔽）。
0：原始中断未挂起
1：原始中断挂起

位 0 **INRIS**：输入 FIFO 服务原始中断状态 (Input FIFO service raw interrupt status)
该位提供输入 FIFO 中断信息（不考虑相应 CRYP_IMSCR 屏蔽）。
0：原始中断未挂起
1：原始中断挂起

34.7.8 CRYP 屏蔽中断状态寄存器 (CRYP_MISR)

CRYP masked interrupt status register

偏移地址：0x1C

复位值：0x0000 0000

CRYP_MISR 寄存器为屏蔽中断状态寄存器。该寄存器为只读寄存器。执行读操作时，该寄存器会给出相应中断的当前屏蔽状态，即中断信息（考虑 CRYP_IMSCR 屏蔽）。写操作无影响。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OUTMIS	INMIS
														r	r

位 31:2 保留，必须保持复位值。

位 1 **OUTMIS**：输出 FIFO 服务屏蔽中断状态 (Output FIFO service masked interrupt status)
该位提供输出 FIFO 中断信息（考虑相应 CRYP_IMSCR 屏蔽）。
0：输出 FIFO 中断未挂起
1：输出 FIFO 中断挂起

位 0 **INMIS**：输入 FIFO 服务屏蔽中断状态 (Input FIFO service masked interrupt status)
该位提供输入 FIFO 中断信息（考虑相应 CRYP_IMSCR 屏蔽）。
0：输入 FIFO 中断未挂起
1：CRYPEN = 1 时输入 FIFO 中断挂起

34.7.9 CRYP 密钥寄存器 0L (CRYP_K0LR)

CRYP key register 0L

偏移地址：0x20

复位值：0x0000 0000

CRYP 密钥寄存器包含密码密钥。

有关详细信息，请参见*第 34.4.18 节：CRYP 密钥寄存器*。

注： *当密码处理器繁忙（CRYP_SR 寄存器中的位 BUSY = 1）时或 KMOD = 10（共享 AES 密钥）时，会忽略对这些寄存器的写访问。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
K[255:240]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K[239:224]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **K[255:224]**：密钥位 x (Key bit x)（x = 255 到 224）

该只写位字段包含 AES 加密或解密密钥的位 [255:224]，具体取决于工作模式。该寄存器不适用于 DES/TDES 模式。

34.7.10 CRYP 密钥寄存器 0R (CRYP_K0RR)

CRYP key register 0R

偏移地址：0x24

复位值：0x0000 0000

有关详细信息，请参见*第 34.7.9 节：CRYP 密钥寄存器 0L (CRYP_K0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
K[223:208]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K[207:192]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **K[223:192]**：密钥位 x (Key bit x)（x = 223 到 192）

该只写位字段包含 AES 加密或解密密钥的位 [223:192]，具体取决于工作模式。该寄存器不适用于 DES/TDES 模式。

34.7.11 CRYP 密钥寄存器 1L (CRYP_K1LR)

CRYP key register 1L

偏移地址：0x28

复位值：0x0000 0000

有关详细信息，请参见*第 34.7.9 节：CRYP 密钥寄存器 0L (CRYP_K0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
K[191:176]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K[175:160]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **K[191:160]**：密钥位 x (Key bit x)（x = 191 到 160）

该只写位字段包含 AES 加密或解密密钥的位 [191:160]，具体取决于工作模式。在 DES/TDES 模式下，该位字段包含密钥 K1 的位[64:33]，奇偶校验位未使用。

34.7.12 CRYP 密钥寄存器 1R (CRYP_K1RR)

CRYP key register 1R

偏移地址：0x2C

复位值：0x0000 0000

有关详细信息，请参见*第 34.7.9 节：CRYP 密钥寄存器 0L (CRYP_K0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
K[159:144]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K[143:128]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **K[159:128]**：密钥位 x (Key bit x)（x = 159 到 128）

该只写位字段包含 AES 加密或解密密钥的位 [159:128]，具体取决于工作模式。在 DES/TDES 模式下，该位字段包含密钥 K1 的位[32:1]，奇偶校验位未使用。

34.7.13 CRYP 密钥寄存器 2L (CRYP_K2LR)

CRYP key register 2L

偏移地址：0x30

复位值：0x0000 0000

有关详细信息，请参见*第 34.7.9 节：CRYP 密钥寄存器 0L (CRYP_K0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
K[127:112]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K[111:96]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **K[127:96]**：密钥位 x (Key bit x)（x = 127 到 96）

该只写位字段包含 AES 加密或解密密钥的位 [127:96]，具体取决于工作模式。在 DES/TDES 模式下，该位字段包含密钥 K2 的位[64:33]，奇偶校验位未使用。

34.7.14 CRYP 密钥寄存器 2R (CRYP_K2RR)

CRYP key register 2R

偏移地址：0x34

复位值：0x0000 0000

有关详细信息，请参见*第 34.7.9 节：CRYP 密钥寄存器 0L (CRYP_K0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
K[95:80]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K[79:64]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **K[95:64]**：密钥位 x (Key bit x)（x = 95 到 64）

该只写位字段包含 AES 加密或解密密钥的位 [95:64]，具体取决于工作模式。在 DES/TDES 模式下，该位字段包含密钥 K2 的位[32:1]，奇偶校验位未使用。

34.7.15 CRYP 密钥寄存器 3L (CRYP_K3LR)

CRYP key register 3L

偏移地址：0x38

复位值：0x0000 0000

有关详细信息，请参见*第 34.7.9 节：CRYP 密钥寄存器 0L (CRYP_K0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
K[63:48]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K[47:32]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **K[63:32]**：密钥位 x (Key bit x)（x = 63 到 32）

该只写位字段包含 AES 加密或解密密钥的位 [63:32]，具体取决于工作模式。在 DES/TDES 模式下，该位字段包含密钥 K3 的位[64:33]，奇偶校验位未使用。

34.7.16 CRYP 密钥寄存器 3R (CRYP_K3RR)

CRYP key register 3R

偏移地址：0x3C

复位值：0x0000 0000

有关详细信息，请参见*第 34.7.9 节：CRYP 密钥寄存器 0L (CRYP_K0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
K[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **K[31:0]**：密钥位 x (Key bit x)（x = 31 到 0）

该只写位字段包含 AES 加密或解密密钥的位 [31:0]，具体取决于工作模式。在 DES/TDES 模式下，该位字段包含密钥 K3 的位[32:1]，奇偶校验位未使用。

34.7.17 CRYP 初始化向量寄存器 0L (CRYP_IV0LR)

CRYP initialization vector register 0L

偏移地址：0x40

复位值：0x0000 0000

CRYP_IVx(L/R)R 寄存器用于存储初始化向量或随机值，具体取决于所选链接模式。IV 数据的大小为 64 位（对于 DES/TDES）或128 位（对于 AES）。有关详细信息，请参见*第 34.4.19 节：CRYP 初始化向量寄存器*。

注：　当密码处理器繁忙时（CRYP_SR 寄存器中的 BUSY = 1），会忽略对这些寄存器的写访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[127:112]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVI[111:96]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IVI[127:96]**：初始化向量位 x (Initialization vector bit x)（x = 127 到 96）

该位字段存储除 ECB 外的 AES 链接模式的初始化向量位 [127:96]。在 DES/TDES 模式下，该位字段对应于 IVI 位 [63:32]。

完成每一轮的 DES/TDES 或 AES 内核计算后，内核都会更新 IV 寄存器。

34.7.18 CRYP 初始化向量寄存器 0R (CRYP_IV0RR)

CRYP initialization vector register 0R

偏移地址：0x44

复位值：0x0000 0000

有关详细信息，请参见*第 34.7.17 节：CRYP 初始化向量寄存器 0L (CRYP_IV0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[95:80]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVI[79:64]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IVI[95:64]**：初始化向量位 x (Initialization vector bit x)（x = 95 到 64）

该位字段存储除 ECB 外的 AES 链接模式的初始化向量位 [95:64]。在 DES/TDES 模式下，该位字段对应于 IVI 位 [31:0]。

完成每一轮的 DES/TDES 或 AES 内核计算后，内核都会更新 IV 寄存器。

34.7.19 CRYP 初始化向量寄存器 1L (CRYP_IV1LR)

CRYP initialization vector register 1L

偏移地址：0x48

复位值：0x0000 0000

有关详细信息，请参见*第 34.7.17 节：CRYP 初始化向量寄存器 0L (CRYP_IV0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[63:48]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVI[47:32]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IVI[63:32]**：初始化向量位 x (Initialization vector bit x)（x = 63 到 32）

该位字段存储除 ECB 外的 AES 链接模式的初始化向量位 [63:32]。*该寄存器不适用于 DES 模式*。

完成每一轮的 AES 内核计算后，内核都会更新 IV 寄存器。

34.7.20 CRYP 初始化向量寄存器 1R (CRYP_IV1RR)

CRYP initialization vector register 1R

偏移地址：0x4C

复位值：0x0000 0000

有关详细信息，请参见*第 34.7.17 节：CRYP 初始化向量寄存器 0L (CRYP_IV0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVI[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IVI[31:0]**：初始化向量位 x (Initialization vector bit x)（x = 31 到 0）

该位字段存储除 ECB 外的 AES 链接模式的初始化向量位 [31:0]。*该寄存器不适用于 DES 模式*。

完成每一轮的 AES 内核计算后，内核都会更新 IV 寄存器。

34.7.21 CRYP 上下文交换 GCM-CCM 寄存器 (CRYP_CSGCMCCMxR)

CRYP context swap GCM-CCM registers

偏移地址：0x050 + x* 0x4（x=0 到 7）

复位值：0x0000 0000

在选择 GCM/GMAC 或 CCM 算法后，这些寄存器含有 CRYP 处理器的全部内部寄存器状态。如果高优先级的任务需要使用密码处理器，而该处理器正在执行其他任务，则需要使用上述寄存器执行上下文交换。

出现此类事件时，需要对 CRYP_CSGCMCCM0..7R 和 CRYP_CSGCM0..7R（GCM/GMAC 模式）或 CRYP_CSGCMCCM0..7R（CCM 模式）寄存器进行读操作，并且需要将读出的值保存到系统内存空间。之后，可使用密码处理器执行高优先级的任务。完成密码计算之后，可从内存中读出保存的上下文并将其写回相应的上下文交换寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSGCMCCMx[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSGCMCCMx[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CSGCMCCMx[31:0]**：GCM、GMAC 和 CCM 模式的 CRYP 处理器内部寄存器状态 (CRYP processor internal register states for GCM, GMAC and CCM modes)

注意：该寄存器不用于 DES/TDES 或其他未指定的 AES 模式

34.7.22 CRYP 上下文交换 GCM 寄存器 (CRYP_CSGCMxR)

CRYP context swap GCM registers

偏移地址：0x070 + x* 0x4（x=0 到 7）

复位值：0x0000 0000

有关详细信息，请参见*第 34.7.21 节：CRYP 上下文交换 GCM-CCM 寄存器 (CRYP_CSGCMCCMxR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSGCMx[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSGCMx[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CSGCMx[31:0]**：GCM 和 GMAC 模式的 CRYP 处理器内部寄存器状态 (CRYP processor internal register states for GCM and GMAC modes)。

注意：该寄存器不用于 DES/TDES 或其他未指定的 AES 模式

34.7.23 CRYP 硬件配置寄存器 (CRYP_HWCFGR)

CRYP hardware configuration register

偏移地址：0x3F0

复位值：0x0000 0131

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG4[3:0]				CFG3[3:0]				CFG2[3:0]				CFG1[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:12 **CFG4[3:0]**：硬件通用 4 (Hardware Generic 4)
保留，必须保持复位值。

位 11:8 **CFG3[3:0]**：硬件通用 3 (Hardware Generic 3)
该位字段返回 `dma_version` 通用值 (0x1)。

位 7:4 **CFG2[3:0]**：硬件通用 2 (Hardware Generic 2)
该位字段返回 `aes_impl` 通用值 (0x3)。

位 3:0 **CFG1[3:0]**：硬件通用 1 (Hardware Generic 1)
该位字段返回 `tdes_impl` 通用值 (0x1)。

34.7.24 CRYP 版本寄存器 (CRYP_VERR)

CRYP version register

偏移地址：0x3F4

复位值：0x0000 0023

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJVER[3:0]				MINVER[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJVER[3:0]**：CRYP 处理器主版本 (CRYP processor major version)
该位字段返回 CRYP 处理器主版本。

位 3:0 **MINVER[3:0]**：CRYP 处理器次版本 (CRYP processor minor version)
该位字段返回 CRYP 处理器次版本。

34.7.25 CRYP 标识 (CRYP_IPIDR)

CRYP identification

偏移地址：0x3F8

复位值：0x0017 0011

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：标识码 (Identification code)
该位字段返回 CRYP 处理器的标识码。

34.7.26 CRYP 大小标识寄存器 (CRYP_SIDR)

CRYP size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识码 (size identification code)
该位字段返回 CRYP 处理器的大小标识码，定义如下：
SID[31:8] = 0xA3C5DD（固定代码）
SID[7:0] = 0x01（1 KB 地址解码）

34.7.27 CRYP 寄存器映射

表 298. CRYP 寄存器映射和复位值

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CRYP_CR	IPRST	Res.	Res.	Res.	Res.	Res.	KMOD[1:0]		NPBLB				ALGOMODE[3]	Res.	GCM_CCMPH		CRYPEN	FFLUSH	Res.	Res.	Res.	Res.	KEYSIZE		DATATYPE		ALGOMODE[2:0]			ALGODIR	Res.	Res.
	Reset value	0						0	0	0	0	0	0	0		0	0	0	0					0	0	0	0	0	0	0	0		
0x04	CRYP_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEYVALID	KERF	Res.	BUSY	OFFU	OFNE	IFNF	IFEM
	Reset value																									0	0		0	0	0	1	1
0x08	CRYP_DIN	DATAIN																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	CRYP_DOUT	DATAOUT																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	CRYP_DMACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOEN	DIEN
	Reset value																															0	0
0x14	CRYP_IMSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OUTIM	INIM
	Reset value																															0	0
0x18	CRYP_RISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OUTRIS	INRIS
	Reset value																															0	1
0x1C	CRYP_MISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OUTMIS	INMIS
	Reset value																															0	0
0x20	CRYP_K0LR	K[255:224]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	CRYP_K0RR	K[223:192]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...																																	
0x38	CRYP_K3LR	K[63:32]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	CRYP_K3RR	K[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	CRYP_IV0LR	IVI[127:96]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 298. CRYP 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x44	CRYP_IV0RR	IVI[95:64]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	CRYP_IV1LR	IVI[63:32]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	CRYP_IV1RR	IVI[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	CRYP_ CSGCMCCM0R	CSGCMCCM0																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54	CRYP_ CSGCMCCM1R	CSGCMCCM1																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x58	CRYP_ CSGCMCCM2R	CSGCMCCM2																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5C	CRYP_ CSGCMCCM3R	CSGCMCCM3																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	CRYP_ CSGCMCCM4R	CSGCMCCM4																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x64	CRYP_ CSGCMCCM5R	CSGCMCCM5																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x68	CRYP_ CSGCMCCM6R	CSGCMCCM6																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x6C	CRYP_ CSGCMCCM7R	CSGCMCCM7																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x70	CRYP_CSGCM 0R	CSGCM0																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x74	CRYP_CSGCM 1R	CSGCM1																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x78	CRYP_CSGCM 2R	CSGCM2																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x7C	CRYP_CSGCM 3R	CSGCM3																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x80	CRYP_CSGCM 4R	CSGCM4																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 298. CRYP 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x84	CRYP_CSGCM5R	CSGCM5																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x88	CRYP_CSGCM6R	CSGCM6																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8C	CRYP_CSGCM7R	CSGCM7																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x90 to 0x3EC	Reserved	Reserved																															
0x03F0	CRYP_HWCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFG3[3:0]				CFG2[3:0]				CFG1[3:0]			
	Reset value																					0	0	0	1	0	0	1	1	0	0	0	1
0x03F4	CRYP_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJVER [3:0]				MINVER [3:0]			
	Reset value																									0	0	1	0	0	0	1	1
0x03F8	CRYP_IPIDR	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
0x03FC	CRYP_SIDR	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

35 哈希处理器 (HASH)

35.1 简介

哈希处理器完全兼容安全哈希算法（SHA-1、SHA-2 系列、SHA-3 系列）和 HMAC（密钥哈希消息认证码）算法。HMAC 适用于需要进行消息验证的应用。

哈希处理器可以为任何长度小于 2^{64} 位（对于 SHA-1、SHA-224 和 SHA-256）或小于 2^{128} 位（对于 SHA-384 和 SHA-512）的消息计算经 FIPS（联邦信息处理标准）认可的 160、224、256、384 和 512 位长度的哈希值。

35.2 HASH 主要特性

- 适合于数据验证应用，符合以下标准：
  - 联邦信息处理标准出版物 FIPS PUB 180-4 *安全哈希标准*（SHA-1 和 SHA-2 系列）
  - 联邦信息处理标准出版物 FIPS PUB 186-4 *数字签名标准 (DSS)*
  - 互联网工程任务组 (IETF) 征求意见 RFC 2104 *HMAC：密钥哈希消息认证*和联邦信息处理标准出版物 FIPS PUB 198-1 *密钥哈希消息认证码 (HMAC)*
  - 联邦信息处理标准出版物 FIPS PUB 202 *安全哈希标准*（SHA-3 系列）
- 快速计算 SHA-1、SHA2-224、SHA2-256、SHA2-384 和 SHA2-512
  - 使用 SHA-1（或 SHA-256）算法处理一个 512 位数据块需要 82（或 66）个时钟周期
  - 使用 SHA2-384 或 SHA2-512 算法处理一个 1024 位数据块需要 98 个时钟周期
  - 支持 SHA-2 截断输出（SHA2-512/224、SHA2-512/256）
- 支持 FIPS 202 标准中定义的六个基于 Keccak 的函数
  - 固定输出长度：SHA3-224、SHA3-256、SHA3-384 和 SHA3-512
  - 可扩展输出函数：SHAKE128 和 SHAKE256
  - 每一 Keccak 轮为一个周期，因此 SHA3-256 的每个 1088 位块需要 58 个周期
- 支持 HMAC 模式及其所有支持的算法
- 连续消息块中哈希值的对应 32 位字添加到彼此之中，以构成整个消息的哈希值
  - 可自动进行 32 位字交换，以兼容输入位串的内部小端模式表示法
  - 支持的字交换格式：位、字节、半字和 32 位字
- 单个 32 位只写输入寄存器，与内部输入 FIFO 相关联，对应于一个 64 字节块的大小（16 x 32 位）
- 可自动填充来完成输入位串，从而适应哈希值最小块大小
- AHB 从外设，仅支持 32 位字访问（否则会产生 AHB 错误）
- 50 × 32 位字（HR0 到 HR41）用于输出消息哈希值和通用 SHA3 输入
- 数据流自动控制，支持使用一个通道进行直接内存访问 (DMA)。
- 支持 固定四字 DMA 突发传输。
- 按块计算可中断消息的哈希值
  - 可重载哈希寄存器
  - 哈希计算挂起/恢复机制，包括 DMA

35.3 HASH 实现

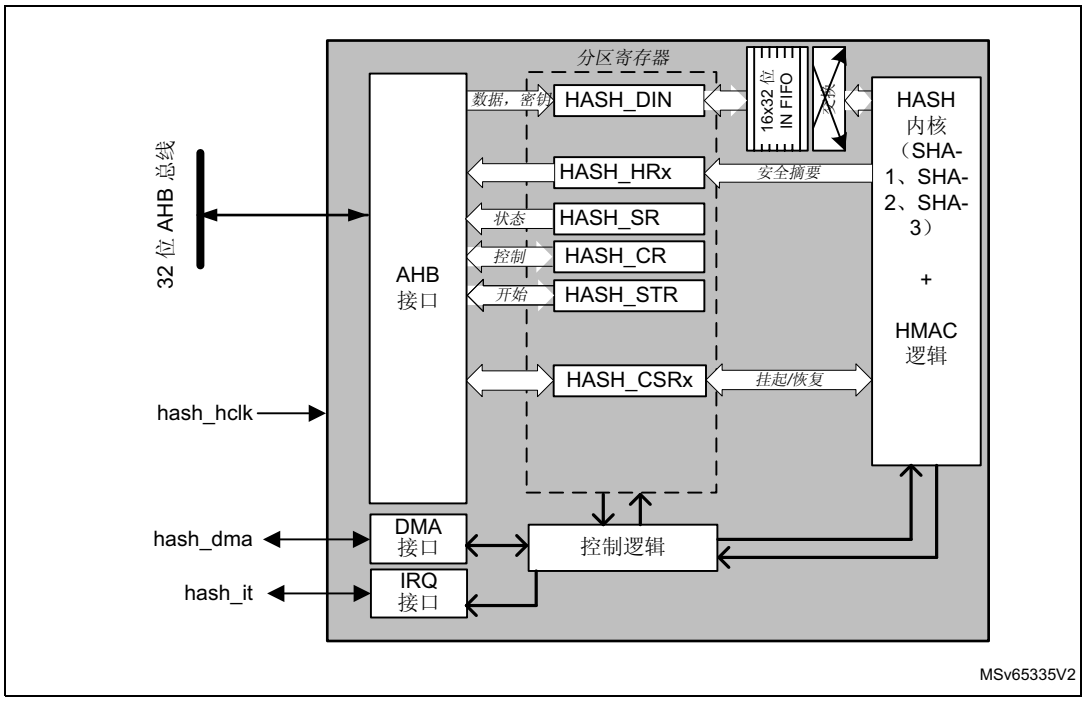
该设备拥有一个 HASH 外设实例，连接到 MDMA。

35.4 HASH 功能描述

35.4.1 HASH 框图

图 329 显示了哈希处理器的框图。

图 329. HASH 框图



35.4.2 HASH 内部信号

表 299 列出了 HASH 级而非产品级（焊盘上）所提供的内部信号，对这些信号有所了解会很有用。

表 299. HASH 内部输入/输出信号

信号名称	信号类型	说明
hash_hclk	数字输入	AHB 总线时钟
hash_it	数字输出	哈希处理器全局中断请求
hash_dma	数字输入/输出	DMA 突发请求/确认

35.4.3 关于安全哈希算法

哈希处理器完全兼容由 FIPS PUB 180-4 和 FIPS PUB 202 标准定义的安全哈希算法。

对于每个算法，HASH 计算消息或数据文件的压缩表示。更具体来讲，当输入中出现消息时，HASH 处理内核将生成一个固定长度的输出位串，称为哈希值（请参见*表 300*）。

表 300. 所支持哈希算法的相关信息

算法	哈希值大小（位）	块大小（字节）(1)	消息长度	位串消息
SHA-1	160	64	$< 2^{64}$ 位	是
SHA2-224	224			
SHA2-256	256			
SHA2-384	384	128	$< 2^{128}$ 位	是
SHA2-512	512(2)			
SHA3-224	224	144	$< 2^{64}$ 位	否(3)
SHA3-256	256	136		
SHA3-384	384	104	$< 2^{128}$ 位	
SHA3-512	512	72		
SHAKE128	可变输出(4)	168	$< 2^{64}$ 位	
SHAKE256		136		

1. 块大小 = (NBWE-1) * 4 字节。写入 HASH_CR 寄存器中的 ALGO 和 INIT 位后，可以从 HASH_SR 寄存器读取 NBWE[5:0] 位字段。
2. SHA2-512/224 算法的哈希值大小为 224 位，SHA2-512/256 算法的哈希值大小为 256 位（截断模式）。
3. 数据以字节串而非位串的形式输送。
4. 当未达到预期的哈希长度（无边界输出）时，需通过将 HASH_STR 寄存器中的 DCAL 置 1 来进行新的计算。当哈希长度大于 1600 位的块大小时，就会出现这种情况。请注意，输出长度不能大于块大小（或输入速率）。

然后可以通过数字签名算法来处理此哈希值，以便生成或验证消息的签名。

对哈希值而不是对消息签名通常可提高流程的效率，因为哈希值通常比消息要小得多。数字签名的验证程序和数字签名的创建程序必须使用相同哈希算法。

哈希处理器支持的 SHA-2 和 SHA-3 函数由 NIST 评定为“安全可靠”，因为要找出与某个给定哈希值对应的消息，或找出两个生成相同哈希值的不同消息（自 2017 年 2 月起 SHA-1 不再安全可靠），在计算层面无法实现。对传输中的消息进行任何更改都极有可能产生不同的哈希值，从而导致签名验证失败。

可扩展输出函数 (XOF) SHAKE 可通过其海绵函数应请求生成任意位数。例如，根据 NIST 标准，对于输出长度为 d 的消息 M：

- SHAKE128(M, d=256) 可用作哈希函数以提供安全强度为 128 位的 256 位比特流。任意大的长度都可以用作伪随机数发生器。
- SHAKE256(M, d=128) 可用作哈希函数以提供安全强度为 128 位的 128 位比特流。

35.4.4 消息数据输送

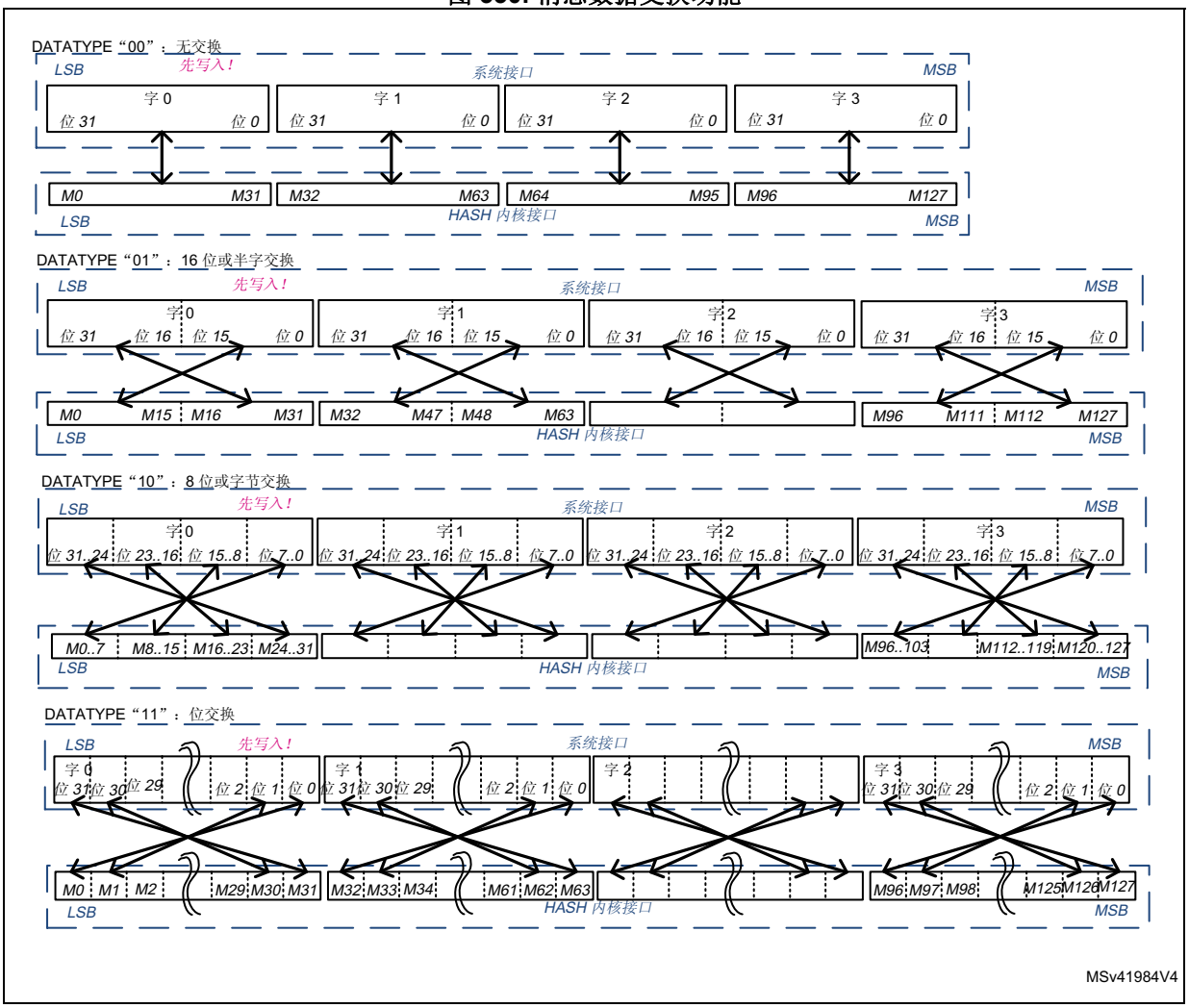
要由 HASH 处理的信息（或数据文件）必须视为位串，但 SHA-3 模式（见*表 300*）除外，在该模式下消息为字节串。根据 FIPS PUB 180-4 标准，该消息位串从左到右增长（十六进制字采用“大端”约定来表示），从而使得在每个字内，最高有效位存储在最左侧位的位置上。以位串表示为“**01100001 01100010 01100011**”的消息串“abc”为例，其 32 位字表示为 **0x00636261**，8 位字表示为 **0x61626300**。

通过将数据写入到 HASH_DIN 寄存器，数据将一次性输入到 HASH 的一个 32 位字中。每次在寄存器中写入新数据时，HASH_DIN 寄存器的当前内容都将传输到 16 字输入 FIFO。因此，HASH_DIN 和 FIFO 构成了一个长度为十七个 32 位字的 FIFO（名为 IN 缓冲器）。

根据要处理的数据类型（例如，当数据是 ASCII 文本流时的字节交换），必须先对从输入 FIFO 读取的数据进行位、字节、半字或无交换操作，然后再将其输入小端哈希处理内核。*图 330* 显示了如何根据 HASH 控制寄存器 (HASH_CR) 中的 DATATYPE 位字段，使用由驱动程序弹出到输入 FIFO 中的一个 32 位字构建哈希处理内核 32 位数据块 M0...31。

当禁止位交换时 (DATATYPE = 00)，HASH_DIN 数据字节序如下所述：消息的最低有效位必须在输入到哈希处理器的第一个字中的 MSB 位置，位串的第 32 位必须在输入到哈希处理器的第二个字中的 MSB 位置，依此类推。

图 330. 消息数据交换功能



35.4.5 哈希值计算

当计算哈希值时，哈希处理器会按顺序处理多个块。有关块大小的详细信息，可参见 *表 300：所支持哈希算法的相关信息*。

DMA 或 CPU 每次将块写入到哈希处理器时，HASH 都会自动开始计算哈希值。此操作称为部分哈希计算。

如*第 35.4.4 节：消息数据输送*中所述，待处理的消息将一次性输入到 HASH 32 位字中，从而写入 HASH_DIN 寄存器以填充输入 FIFO。应用程序必须使用以下流程对该数据执行哈希计算。

1. 使用 HASH_CR 寄存器初始化哈希处理器：
   - 写入 HASH_CR 寄存器中的 ALGO 位字段以选择正确的算法，并将 INIT 位置 1（其他位保持为零）。然后从 HASH_SR 寄存器读取 NBWE 位字段以推断算法块大小，其等于 **(NBWE-1) * 4**。如果已知块大小，则不需要此步骤（有关详细信息，请参见*表 300*）。
   - 使用 ALGO[3:0] 位字段选择正确的算法。如果需要，可使用 DATATYPE[1:0] 位字段为消息输入字编程正确的交换操作。
   - 当需要使用 HMAC 模式时，如果 HMAC 密钥大小大于算法已知的块大小，则将 MODE 位以及 LKEY 位置 1（否则将 LKEY 位保持清零状态）。有关详细信息，请参见*第 35.4.7 节：HMAC 运算*。
   - 如果消息的最后一个字中的有效位数不是 32 位，则更新 HASH_STR 寄存器中的 NBLW[4:0] 以定义最后一个字中的有效位数。NBLW 信息用于在最终哈希值计算之前正确执行自动消息填充操作。
2. 通过将 HASH_CR 寄存器中的 INIT 位置 1 来完成初始化。如果通过 DMA 传输数据，也将 DMAE 位置 1。

小心： 编程步骤 2 时，务必事先或同时设置正确的配置值（ALGO、DATATYPE、HMAC 模式、密钥长度、NBLW）。

3. 开始通过写入 HASH_DIN 寄存器来填充数据（自动通过 MDMA 传输数据的情况除外）。请注意，只有当块的最后一个值进入输入 FIFO 后，才能开始处理块。部分或最终哈希计算的管理方式取决于数据输送到处理器的方式：
   - 通过软件填充数据：

     每次应用程序写入下一个块的第一个字时，都会触发部分哈希计算，块大小由 HASH_SR 中的 NBWE 位定义。一旦处理器再次准备就绪（HASH_SR 中的 DINIS = 1），软件即可将新数据写入 HASH_DIN。这种机制可防止 HASH 引入等待状态。

     当输入最后一个块并且软件将 DCAL 位置 1 时，将触发最终哈希计算。如果消息长度不是块大小的整数倍，则必须在写入 DCAL 位之前写入 HASH_STR 寄存器中的 NBLW 位字段（有关详细信息，请参见*第 35.4.6 节*）。
   - 通过单次 DMA 传输 (MDMAT = 0) 填充数据：

     每次 FIFO 已满时，都将自动触发部分哈希计算。

     当最后一个块通过 DMA 传输到 HASH_DIN 寄存器后，将自动触发最终哈希计算（硬件将 DCAL 位置 1）。如果消息长度不是块大小的整数倍，则必须在使能 DMA 之前写入 HASH_STR 寄存器中的 NBLW 位字段（有关详细信息，请参见*第 35.4.6 节*）。
   - 使用多次 DMA 传输 (MDMAT = 1) 填充数据：

     可针对单次 DMA 传输触发部分哈希计算（请参见上述说明）。但是，当最后一个块通过 MDMA 传输到 HASH_DIN 寄存器后，不会自动触发最终哈希计算（硬件不将 DCAL 位置 1）。当执行倒数第二个块的 4 字传输时，软件必须先将 DMA 中断位（DMAA）置 1，然后再将最后的至多 4 个字传输到 HASH_DIN（实际字数取决于消息的总长度）。一旦输入最后一个字后，软件会立即将 DCAL 位置 1 以开始最终哈希计算过程。如果消息长度不是块大小的整数倍，则必须在写入 DCAL 位之前写入 HASH_STR 寄存器中的 NBLW 位字段（有关详细信息，请参见*第 35.4.6 节*）。

4. 哈希计算完成 (DCIS = 1) 后，可以从输出寄存器读取生成的哈希值，如*表 301* 中所述。
5. 对于 SHAKE128 或 SHAKE256，计算一次可能不够（实际计算次数 = 预期输出长度除以块大小）。每多执行一次计算，就要在步骤 4 之后执行一次以下步骤：
   a) 将 HASH_STR 寄存器中的 DCAL 置 1。
   b) 等待 DCIS = 1，然后从相同的输出寄存器中读取新生成的哈希值。如果还要再执行一次计算，则转至步骤 a。

表 301. 哈希处理器输出

算法	有效输出寄存器	最高有效位	哈希大小（位）
SHA-1	HASH_H0 到 HASH_H4	HASH_H0[31]	160
SHA2-224	HASH_H0 到 HASH_H6	HASH_H0[31]	224
SHA2-256	HASH_H0 到 HASH_H7		256
SHA2-384	HASH_H0 到 HASH_H11	HASH_H0[31]	384
SHA2-512	HASH_H0 到 HASH_H15		512(1)
SHA3-224	HASH_H0 到 HASH_H6	HASH_H0[31]	224
SHA3-256	HASH_H0 到 HASH_H7		256
SHA3-384	HASH_H0 到 HASH_H11		384
SHA3-512	HASH_H0 到 HASH_H15		512
SHAKE128	HASH_H0 到 HASH_H41		可变输出(2)
SHAKE256	HASH_H0 到 HASH_H33		

1. SHA2-512/224 算法的哈希大小为 224 位，SHA2-512/256 算法的哈希大小为 256 位（截断模式）
2. 上限无边界（可扩展输出功能），请参见*表 300*。

关于 HMAC 详细指令的更多信息，请参见*第 35.4.7 节：HMAC 运算*。

35.4.6 消息填充

概述

当计算消息的压缩表示时，循环执行将数据输送到哈希处理器（每个块大小传输执行一次自动部分哈希计算）的过程，直到原始消息的最后几位被写入 HASH_DIN 寄存器。

因为消息的长度（位数）可以是任何整数值，因此写入到哈希处理器的最后一个字的有效位数在 1 到 32 之间。这个最后一个字的有效位数必须写到 HASH_STR 寄存器的 NBLW 位字段中，以便消息填充能够在最终哈希计算之前得到正确处理。

注： *当在 HASH_CR 寄存器中选择 SHA-3 模式时，消息被定义为多个字节，最后一个字有 1 到 4 个有效字节数。因此，HASH_STR 寄存器中的 NBLW 位 0 到 2 始终为零。*

填充处理

*第 35.4.5 节：哈希值计算*中介绍了使能或禁止 DMA 时的详细填充步骤。

填充示例

根据联邦信息处理标准 PUB 180-4 规定，消息填充包括添加一个“1”，后跟 *k* 个“0”以及一个 64 位整数（长度与消息长度 *L*（位数）相等）。经过这三个填充操作后生成的消息长度为 *L* + 1 + *k* + 64（512 位的整数倍）。

对于哈希处理器，“1”将添加到 HASH_DIN 寄存器中写入的最后一个字，位位置由 NBLW 位字段定义，而其余更高的位将被清零（“0”）。

FIPS PUB180-4 提供的示例

我们假定原始消息是采用 ASCII 二进制编码格式的“abc”，长度 L=24：

```
字节 0     字节 1     字节 2     字节 3
01100001 01100010 01100011 uuuuuuuu
<-- 写入 HASH_DIN 的第一个字 -->
```

NBLW 必须以值 24 加载：“**1**”在位串中的位 24 处追加（在以上位串中从左到右开始计数），这对应于 HASH_DIN 寄存器中的位 31（小端模式约定）：

```
01100001 01100010 01100011 1uuuuuuu
```

由于 L = 24，因此以上位串中的位数是 25，并将追加 423 个“0”位，现在为 448 位。

这将生成十六进制（大端模式的字节字）：

```
61626380 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000
```

将追加双字格式的消息长度值 **L**，即 00000000 00000018。因此，最后填充的消息为十六进制（大端模式的字节字）：

```
61626380 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000018
```

如果哈希处理器被编程为在 HASH_DIN 输入寄存器（HASH_CR 中的 DATATYPE = 10）中交换字节，则必须使用以下步骤输入上述消息：

1. 将 **0xUU636261** 写入到 HASH_DIN 寄存器（其中的“**U**”意味着无关）
2. 将 **0x18** 写入到 HASH_STR 寄存器（写入到 HASH_DIN 寄存器的最后一个字中的有效位数是 24，因为原始消息长度为 24 位）
3. 将 **0x10** 写入到 HASH_STR 寄存器以启动消息填充（如上文所述）以及随后执行哈希计算
4. 完成哈希计算后，HASH_HRx 寄存器 (x = 0...4) 中的哈希值可供 SHA-1 算法使用。对于此 FIPS 示例，预期值如下：

```
HASH_HR0 = 0xA9993E36
HASH_HR1 = 0x4706816A
HASH_HR2 = 0xBA3E2571
HASH_HR3 = 0x7850C26C
HASH_HR4 = 0x9CD0D89D
```

35.4.7 HMAC 运算

概述

根据互联网工程任务组 RFC2104 和 NIST FIPS PUB 198-1 的规定，HMAC 算法以不可逆方式将正在处理的消息与用户所选的密钥进行绑定，从而用于消息验证。该算法由两个嵌套的哈希运算组成：

```
HMAC(message)= Hash((Key | pad) XOR opad |
                    Hash((Key | pad) XOR ipad | message))
```

其中：

- **Opad** = $[0x5C]_n$（外 pad），**ipad** = $[0x36]_n$（内 pad）。
- $[X]_n$ 表示 X 重复 *n* 次，其中的 *n* 等于底层哈希函数数据块的字节大小（对于 512 位块大小，n = 64）。
- **pad** 是将密钥扩展到上面定义的长度 *n* 所需的零序列。如果密钥长度大于 *n*，应用程序必须首先使用 Hash() 函数对密钥进行哈希处理，然后将得到的字节串用作 HMAC 的实际密钥。
- | 表示串联运算符。

注： *哈希处理器的 HMAC 模式可与所有支持的算法一起使用。*

HMAC 处理

计算 HMAC 需要下列四个步骤：

1. 软件将 INIT 位置 1，同时将 MODE 位置 1，并使用 ALGO 位选择所需的算法。如果正在使用的密钥的长度超过块大小，则还必须将 LKEY 位置 1。在这种情况下，根据 HMAC 规范的要求，哈希处理器将使用密钥的哈希代替真实密钥。
2. 软件提供用于内部哈希函数的密钥，该操作与消息字符串加载操作遵循相同的机制：将密钥数据写入 HASH_DIN 寄存器，然后通过将 DCAL 位置 1 以及将正确的 NBLW 写入 HASH_STR 寄存器来完成传输。

3. 一旦处理器再次准备就绪（HASH_SR 中的 DINIS = 1），软件即可将消息字符串写入 HASH_DIN。当输入最后一个块的最后一个字并且软件将 HASH_STR 寄存器中的 DCAL 位置 1 时，如果消息长度不是块大小的整数倍，则必须同时将 NBLW 位字段编程为非零值。请注意，DMA 也可用于输送消息字符串，如*第 35.4.5 节：哈希值计算*中所述。
4. 一旦处理器再次准备就绪（HASH_SR 中的 DINIS = 1），软件就会提供用于外部哈希函数的密钥，也就是将密钥数据写入 HASH_DIN 寄存器，然后通过将 DCAL 位置 1 以及将正确的 NBLW 编程到 HASH_STR 寄存器来完成传输。DCIS 位置 1 后，HMAC 结果就会存储到有效输出寄存器 (HASH_HRx) 中。

注： *HMAC 本原的计算延迟取决于密钥和消息的长度，如第 35.6 节：HASH 处理时间中所述。*
有关字节序管理的详细信息，可参见第 35.4.4 节：消息数据输送。

HMAC 示例

下面是一个由 NIST 指定的 HMAC SHA-1 算法示例（HASH_CR 中的 ALGO = 00 且 MODE = 1）。SHA-1 块大小为 64 字节。

假定原始消息是采用 ASCII 二进制编码格式的“**Sample message for keylen = blocklen**”（keylen = blocklen 时的示例消息），长度 L = 34 字节。如果 HASH 编程为无交换模式（HASH_CR 中的 DATATYPE = 00），则必须将以下数据按顺序加载至 HASH_DIN 寄存器：

1. 由 NIST 指定的**内部哈希密钥**输入（长度 = 64，即不填充）。由于密钥长度 = 64，HASH_CR 寄存器中的 LKEY 位清零

```
00010203 04050607 08090A0B 0C0D0E0F 10111213 14151617
18191A1B 1C1D1E1F 20212223 24252627 28292A2B 2C2D2E2F
30313233 34353637 38393A3B 3C3D3E3F
```

2. **消息**输入（长度 = 34，即需要填充）。必须将 HASH_STR 设置为 **0x20**，以开始消息填充和内部哈希计算（将“**U**”视为无关）

```
53616D70 6C65206D 65737361 67652066 6F72206B 65796C65
6E3D626C 6F636B6C 656EUUUU
```

3. **外部哈希密钥**输入（长度 = 64，即不填充）。在此输入与内部哈希密钥相同的密钥。
4. **最终外部哈希计算**由 HASH 执行。HMAC-SHA1 哈希结果存储于 HASH_HRx 寄存器中，如下所示：

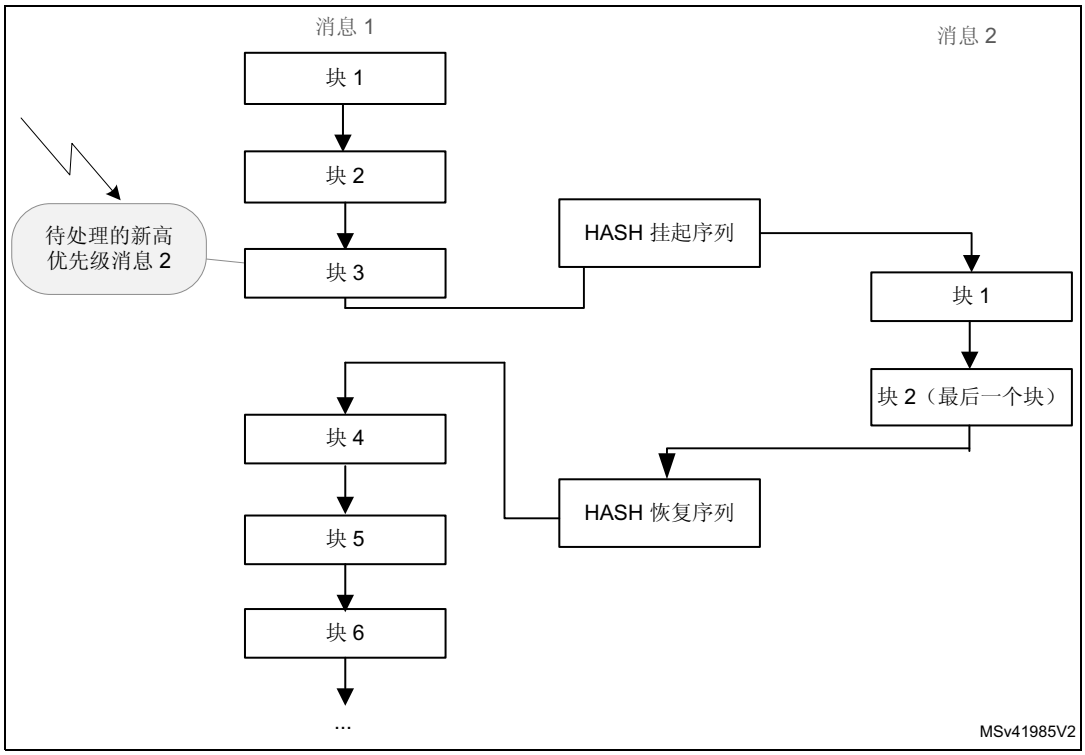
```
HASH_HR0 = 0x5FD596EE
HASH_HR1 = 0x78D5553C
HASH_HR2 = 0x8FF4E72D
HASH_HR3 = 0x266DFD19
HASH_HR4 = 0x2366DA29
```

35.4.8 HASH 挂起/恢复操作

概述

可以中断哈希/HMAC 操作以执行另一个优先级较高的处理。处理完优先级较高的任务后，可以继续完成之前中断的过程，如*图 331* 所示。

图 331. HASH 挂起/恢复机制



要这样做，必须将被中断的任务的上下文从 HASH 寄存器保存到内存，然后从内存恢复到 HASH 寄存器。

以下说明由软件或 DMA 控制数据流的过程。

通过软件加载数据

如果不使用 DMA 将消息加载到哈希处理器，则只有在当前未处理任何块时才能保存上下文。

要挂起消息处理，请在写入 NBWE 中定义的字数后执行以下操作：

1. 在轮询模式下，等待 BUSY = 0，然后轮询 DINIS 状态位是否置 1。
   在中断模式下，通过 DINIS 中断处理程序执行下一步（建议使用）。
2. 将以下寄存器的内容存储到内存中：
   - HASH_IMR
   - HASH_STR
   - HASH_CR
   - HASH_CSR0 到 HASH_CSR37（当选择 SHA-1 或 SHA2-256 时），如果正在进行 HMAC 操作，则加上 HASH_CSR38 到 HASH_CSR53
   - HASH_CSR0 到 HASH_CSR90（当选择 SHA2-384 或 SHA2-512（无论是否截断）时），如果正在进行 HMAC 操作，则加上 HASH_CSR91 到 HASH_CSR102。
   - HASH_CSR0 到 HASH_CSR71（当选择 SHA-3 时），如果正在进行 SHA3-384 或 SHA3-512 HMAC 操作，则加上 HASH_CSR72 到 HASH_CSR87。

要恢复消息处理，请执行以下操作：

1. 将内存中保存的值写入以下寄存器中：HASH_IMR、HASH_STR 和 HASH_CR。
2. 通过将 HASH_CR 寄存器中的 INIT 位置 1 来初始化哈希处理器
3. 将内存中保存的值写入 HASH_CSRx 寄存器。
4. 从之前的中断点重新开始处理。

通过 DMA 加载数据

如果使用 DMA 将消息加载到哈希处理器中，建议先挂起再恢复安全哈希计算，如下所述。

在此流程中，分配给哈希外设的 DMA 通道仍会分配用于处理消息 1（请参见*图 331*）。

要挂起使用 DMA 进行的消息处理，请执行以下操作：

1. 将 DMAE 位清零以禁止 DMA 接口。哈希外设通过 DMA 自动获取足够的数据来完成当前的突发传输。
2. 等待至最后一次 DMA 传输完成（HASH_SR 中的 DMAS = 0）。
3. 禁止 DMA 通道。
4. 在轮询模式或中断模式（建议使用）下，等待至哈希处理器就绪（未在处理任何块），即等待 HASH_SR 中的 DINIS = 1 如果 HASH_SR 中的 DCIS 也置 1，则哈希结果可用，无需进行上下文交换操作。否则转至步骤 5。

5. 保存 HASH_IMR、HASH_STR 和 HASH_CR 寄存器。此外，还保存多个 HASH_CSRx 寄存器，寄存器的具体数量取决于使用的 SHA 算法：
   - 当选择 SHA-1 或 SHA2-256 时，保存 HASH_CSR0 到 HASH_CSR37，如果正在进行 HMAC 操作，则加上 HASH_CSR38 到 HASH_CSR53。
   - 当选择 SHA2-384 或 SHA2-512 时，保存 HASH_CSR0 到 HASH_CSR90，如果正在进行 HMAC 操作，则加上 HASH_CSR91 到 HASH_CSR102。
   - 当选择 SHA3 时，保存 HASH_CSR0 到 HASH_CSR71，如果正在进行 SHA3-384 或 SHA3-512 HMAC 操作，则加上 HASH_CSR72 到 HASH_CSR87。

要恢复使用 DMA 进行的消息处理，请执行以下操作：

1. 重新配置 DMA 控制器，以便其在没有再次被中断的情况下继续传输消息直到结束。
2. 将内存中保存的值编程到 HASH_IMR、HASH_STR 和 HASH_CR 寄存器。
3. 通过将 HASH_CR 寄存器中的 INIT 位置 1 来初始化哈希处理器。
4. 将内存中保存的值编程到 HASH_CSRx 寄存器。
5. 通过将 DMAE 位置 1，从之前的中断点重新开始处理。

35.4.9 HASH DMA 接口

HASH 支持固定四字 DMA 突发传输。

哈希处理器可使用一个接口连接 MDMA 控制器。通过将 HASH_CR 寄存器中的 DMAE 位置 1，可以使用此 DMA 向 HASH 中写入数据。如果该位置 1，HASH 每次需要向 HASH_DIN 寄存器写入块时，都会发出 DMA 请求。

接收到四个 32 位字后，HASH 会自动触发新的 DMA 请求。有关详细信息，请参见*第 35.4.5 节：哈希值计算*。

在开始 DMA 传输之前，软件必须对将要复制到 HASH_DIN 寄存器的最后一个字中的有效位数进行编程。这可通过向 HASH_STR 寄存器中写入以下值来完成：

NBLW = Len(Message)% 32，其中“**x%32**”表示 x 除以 32 的余数。

注： *当在 HASH_CR 寄存器中选择 SHA-3 算法时，NBLW 的位 0 到 2 始终为零（处理的是字节，不是位）。*

HASH_SR 寄存器的 DMAS 位提供有关 DMA 接口活动的信息。该位与 DMAE 一起置 1，并且在 DMAE 清零且未在进行任何 DMA 传输时清零。

注： *没有任何中断与 DMAS 位相关联。*

35.4.10 HASH 错误管理

哈希处理器不生成任何错误标志。

35.5 HASH 中断

哈希处理器可生成两个单独的可屏蔽中断源，用以通知发生以下事件：

- 哈希计算完成 (DCIS)
- 数据输入缓冲器就绪 (DINIS)

这两个中断源连接到同一全局中断请求信号 (hash_it)，该信号又连接到设备中断控制器。通过更改 HASH_IMR 寄存器中的屏蔽位，可以单独使能或禁止各个中断源。将相应的屏蔽位置 1 可使能中断。

可以从 HASH_SR 寄存器中读取各个可屏蔽中断源的状态。*表 302* 对可用功能进行了总结。

表 302. HASH 中断请求

中断缩写	中断事件	事件标志	使能控制位	中断清除方法
HASH	哈希计算完成	DCIS	DCIE	将 DCIS 清零或将 INIT 置 1
	数据输入缓冲器准备好获取新的块	DINIS	DINIE	清零 DINIS 或写入 HASH_DIN

35.6 HASH 处理时间

表 303 汇总了每种工作模式下处理中间块需要的时间。

表 303. 处理时间（时钟周期数）

工作模式	块大小（以字节为单位）	FIFO 加载(1)	计算阶段	总计
SHA-1	64	16	66	**82**
SHA2-224		16	50	**66**
SHA2-256				
SHA2-384	128	32	66	**98**
SHA2-512(2)				
SHA3-224	144	36	24(3)	**60**
SHA3-256	136	34		**58**
SHA3-384	104	26		**50**
SHA3-512	72	18		**42**

1. 包括向处理器中加载块所需的时间。
2. SHA2-512 包括 SHA2-512/224 和 SHA2-512/256 模式。
3. 每轮一个周期，这些操作需要执行 24 轮。

处理消息的最后一个块（或者 HMAC 中某个密钥的最后一个块）所需要的时间可能会更长。此时间取决于最后一个块的长度以及密钥的大小（在 HMAC 模式下）。

与处理中间块相比，此时间可能按以下系数增加：

- 对于哈希消息，该系数为 **1 到 2.5**
- 对于 HMAC 输入密钥，该系数在 **2.5** 左右
- 对于 HMAC 消息，该系数为 **1 到 2.5**
- 对于 HMAC 输出密钥，如果是短密钥，则该系数在 **2.5** 左右
- 对于 HMAC 输出密钥，如果是长密钥，则该系数为 **3.5 到 5**

35.7 HASH 寄存器

哈希内核与多个控制和状态寄存器以及多个哈希值寄存器相关联。所有这些寄存器都只能通过 32 位字来访问，否则将生成 AHB 错误。

35.7.1 HASH 控制寄存器 (HASH_CR)

HASH control register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALGO[3:0]				LKEY
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DMAA	MDMAT	DINNE	NBW[3:0]				Res.	MODE	DATATYPE[1:0]		DMAE	INIT	Res.	Res.
	rw	rw	r	r	r	r	r		rw	rw	rw	rw	rw		

位 31:21 保留，必须保持复位值。

位 20:17 **ALGO[3:0]**：算法选择 (Algorithm selection)

这些位用于选择哈希算法：
0000：SHA-1
0001：保留
0010：SHA2-224
0011：SHA2-256
1100：SHA2-384
1101：SHA2-512/224
1110：SHA2-512/256
1111：SHA2-512
0100：SHA3-224
0101：SHA3-256
0110：SHA3-384
0111：SHA3-512
1000：SHAKE128
1001：SHAKE256
其他：保留，不得使用
仅当 INIT 位置 1 时，此选择才有效。在计算期间更改该位字段不起作用。
在 ALGO 位字段更新并且 INIT 位置 1 后，HASH_SR 中的 NBWE 会自动更新为 0x11（当 ALGO = 0x00XX 时）。

位 16 **LKEY**：长密钥选择 (Long key selection)

如果 HMAC 密钥大于哈希算法（有关详细信息，请参见*表 300：所支持哈希算法的相关信息*）对应的块大小，应用程序必须将该位置 1。例如，对于 SHA2-256，块大小为 64 字节。
0：HMAC 密钥大小小于或等于块大小（短密钥）。在 HMAC 计算过程中，使用 HASH_DIN 中写入的实际密钥值。
1：HMAC 密钥大小大于块大小（长密钥）。在 HMAC 计算过程中，使用密钥的哈希代替真实密钥。
仅当 INIT 位置 1 并且 MODE 位置 1（选择 HMAC 模式）时，此选择才有效。在计算期间更改该位不起作用。

位 15 保留，必须保持复位值。

位 14 **DMAA**：DMA 中止 (DMA Abort)

该位用于中止使用 DMA 来计算哈希值。当 DMAA 置 1 时，不再向 DMA 发出 DMA 请求，HASH_CR 寄存器中的 DMAE 和 HASH_SR 寄存器中的 DMAS 均清零。
如果读取 DMAA，则 DMAA 始终返回 0。向该位写入 0 不起作用。

位 13 **MDMAT**：多个 DMA 传输 (Multiple DMA transfers)

如果在需要多个 DMA 传输时对大型文件进行哈希运算，则将该位置 1。
0：在 DMA 传输结束时自动将 DCAL 置 1。
1：在 DMA 传输结束时不自动将 DCAL 置 1。

位 12 **DINNE**：DIN 非空 (DIN not empty)

有关 DINNE 位的说明，请参见 HASH_SR 的 DINNE 位。
该位为只读。

位 11:8 **NBW[3:0]**：已推入的字数 (Number of words already pushed)

有关 NBW[3:0] 位字段的说明，请参见 HASH_SR 的 NBWP[3:0] 位字段。
该位为只读。

位 7 保留，必须保持复位值。

位 6 **MODE**：模式选择 (Mode selection)

该位针对所选算法选择正常或密钥 HMAC 模式：

0：选择哈希模式

1：选择 HMAC 模式。如果正在使用的密钥的长度超过算法块大小，则必须将 LKEY 位置 1。

仅当 INIT 位置 1 时，此选择才有效。在计算期间更改该位不起作用。

位 5:4 **DATATYPE[1:0]**：数据类型选择 (Data type selection)

该位字段用于定义在 HASH_DIN 寄存器中输入的数据格式：

00：32 位数据。写入到 HASH_DIN 的数据将由哈希处理直接使用，不会重新排序。

01：16 位数据或半字。写入到 HASH_DIN 的数据被视为两个半字，并且在交换之后由 HASH 处理使用。

10：8 位数据或字节。写入到 HASH_DIN 的数据被视为四个字节，并且在交换之后由 HASH 处理使用。

11：位数据或位串。写入到 HASH_DIN 的数据被视为 32 位（字符串的第一位处于位置 0），并且在交换之后由 HASH 处理使用（字符串的第一位处于位置 31）。

位 3 **DMAE**：DMA 使能 (DMA enable)

0：禁止 DMA 传输

1：使能 DMA 传输。哈希内核可以接收数据时，便发送 DMA 请求。

该位置 1 后，将在消息的最后一个数据被写入哈希处理器时由硬件清零。

当正在进行 DMA 传输时，将该位清零不会中止当前的传输。HASH 的 DMA 接口在内部继续保持使能状态，直到传输完成或 INIT 置 1。

将 INIT 位置 1 不会清零 DMAE 位。

位 2 **INIT**：初始化哈希计算 (Initialize message digest calculation)

将该位置 1 将会复位哈希处理器内核，以使 HASH 可以计算新消息的哈希值。

将该位清零不会产生影响。读取该位时，始终返回 0。

位 1:0 保留，必须保持复位值。

35.7.2 HASH 数据输入寄存器 (HASH_DIN)

HASH data input register

偏移地址：0x04

复位值：0x0000 0000

HASH_DIN 是数据输入寄存器。其宽度为 32 位。该寄存器用于以哈希算法（有关详细信息，请参见*表 300：所支持哈希算法的相关信息*）定义的块输入消息。例如，对于 SHA2-256，块大小为 64 字节。

编程 HASH_DIN 寄存器时，在 AHB 数据总线上呈现的值将被“推入”哈希内核，并且寄存器获取 AHB 数据总线上呈现的新值。要获得正确的消息格式，必须先对 HASH_CR 寄存器中的 DATATYPE 位进行配置。

完整的块写入到 HASH_DIN 寄存器后，将启动中间哈希计算。有两种启动方式：

- 通过将下一个块的第一个数据写入 HASH_DIN 寄存器来启动（未使用 DMA 时）
- 自动启动（使用 DMA 时）

最后一个块写入到 HASH_DIN 寄存器之后，可通过将 HASH_STR 寄存器中的 DCAL 位置 1（最终哈希计算）来启动最终哈希计算（包括填充）。使用 DMA 时，如果 MDMAT 位清零且 DMAA 未置 1，则会自动执行该操作。

读取 HASH_DIN 寄存器将返回零。

注： *当 HASH 繁忙时，如果哈希计算（中间或最终）未完成，对 HASH_DIN 寄存器进行写访问可能会导致 AHB 总线阻塞。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATAIN[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAIN[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **DATAIN[31:0]**：数据输入

写入该寄存器会将当前寄存器内容推入到 FIFO，并且寄存器获取在 AHB 数据总线上呈现的新值。

读取该寄存器将返回零。

35.7.3 HASH 启动寄存器 (HASH_STR)

HASH start register

偏移地址：0x08

复位值：0x0000 0000

HASH_STR 寄存器有两个功能：

- 用于定义哈希处理器中输入的消息的最后一个字的有效位数（也就是写入到 HASH_DIN 寄存器的最后一个数据中有效的最低有效位的数量）
- 用于通过将 DCAL 位置 1 来启动对消息中最后一个块的处理。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCAL	Res.	Res.	Res.	NBLW[4:0]				
							rw				rw	rw	rw	rw	rw

位 31:9 保留，必须保持复位值。

位 8 **DCAL**：哈希计算 (Digest calculation)

将该位置 1 将开始使用先前写入的 NBLW 值来填充消息，并且，由于 INIT 位已置 1，因此将使用写入到输入 FIFO 的所有数据字开始计算最终哈希值。

读取该位将返回 0。

位 7:5 保留，必须保持复位值。

位 4:0 **NBLW[4:0]**：最后一个字中的有效位数 (Number of valid bits in the last word)

当消息位串的最后一个字被写入 HASH_DIN 寄存器并且经过内部数据交换后，哈希处理器会只接收如下指定的有效位：

0x00：写入的最后一个数据的全部 32 位均是有效消息位，即 M[31:0]

0x01：写入的最后一个数据中只有一位（在交换后）是有效消息位，即 M[0]

0x02：写入的最后一个数据中只有两位（在交换后）是有效消息位，即 M[1:0]

0x03：最后写入的数据中只有三位（在交换后）有效，即 M[2:0]

...

0x1F：最后写入的数据中只有 31 位（在交换后）有效，即 M[30:0]

注意：当选择 SHA-3 时，位 0 到 2 保持为零（面向字节的输入）。

上述机制仅在 DCAL = 0 时有效。如果在 DCAL 置 1 时写入 NBLW 位，则 NBLW 位字段保持不变。换句话说，配置 NBLW 和设置 DCAL 不能同时进行。

读取 NBLW 位将返回写入到 NBLW 的最后一个值。

35.7.4 哈希结果寄存器

HASH digest registers

这些寄存器包含哈希结果，其定义如下：

- HASH_HR0、HASH_HR1、HASH_HR2、HASH_HR3 和 HASH_HR4 寄存器返回 SHA-1 哈希结果。
- HASH_HR0 到 HASH_HR6 寄存器返回 SHA2-224、SHA2-512/224 或 SHA3-224 哈希结果。
- HASH_HR0 到 HASH_HR7 寄存器返回 SHA2-256、SHA2-512/256 或 SHA3-256 哈希结果。
- HASH_HR0 到 HASH_HR11 寄存器返回 SHA2-384 或 SHA3-384 哈希结果。
- HASH_HR0 到 HASH_HR15 寄存器返回 SHA2-512 或 SHA3-512 哈希结果。
- HASH_HR0 到 HASH_HR41 寄存器返回 SHAKE128 哈希结果，而 HASH_HR0 到 HASH_HR33 寄存器返回 SHAKE256 哈希结果（有关详细信息，请参见*表 301：哈希处理器输出*）。

在所有情况下，哈希值的最高有效位均存储在 HASH_HR0[31] 中，并且未使用的 HASH_HRx 寄存器读为零。

如果当哈希内核在计算中间值或最终哈希值（DCIS 位等于 0）时，对这些寄存器之一进行读访问，则读取操作将返回零。

注： *当启动新消息的哈希计算时（通过将 INIT 位置 1），HASH_HRx 寄存器将被强制设为其复位值。*

HASH_HR0 到 HASH_HR4 寄存器可以通过两个不同的地址进行访问（寄存器别名）。

哈希值别名寄存器 x (HASH_HRAx)

HASH aliased digest register x

偏移地址：0x0C + x * 0x4（x = 0 到 4）

复位值：0x0000 0000

HASH_HRAx 寄存器的内容与位于偏移地址 0x310 处的 HASH_HRx 寄存器之一的内容相同。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Hx[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Hx[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **Hx[31:0]**：哈希数据 x (Hash data x)

请参见*第 35.7.4 节：哈希结果寄存器*简介。

哈希结果寄存器 x (HASH_HRx)

HASH digest register x

偏移地址：0x310 + x * 0x4（x = 0 到 4）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Hx[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Hx[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **Hx[31:0]**：哈希数据 x (Hash data x)

请参见*第 35.7.4 节：哈希结果寄存器*简介。

哈希值补充寄存器 x (HASH_HRx)

HASH supplementary digest register x

偏移地址：0x310 + x * 0x4（x = 5 到 41）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Hx[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Hx[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **Hx[31:0]**：哈希数据 x (Hash data x)
请参见*第 35.7.4 节：哈希结果寄存器*简介。

35.7.5 HASH 中断使能寄存器 (HASH_IMR)

HASH interrupt enable register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCIE	DINIE
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **DCIE**：哈希计算完成中断使能 (Digest calculation completion interrupt enable)
0：禁止哈希计算完成中断
1：使能哈希计算完成中断

位 0 **DINIE**：数据输入中断使能 (Data input interrupt enable)
0：禁止数据输入中断
1：使能数据输入中断

35.7.6 HASH 状态寄存器 (HASH_SR)

HASH status register

偏移地址：0x24

复位值：0x0011 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBWE[5:0]					
										r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DINNE	NBWP[5:0]						Res.	Res.	Res.	Res.	Res.	BUSY	DMAS	DCIS	DINIS
r	r	r	r	r	r	r						r	r	rc_w0	rc_w0

位 31:22 保留，必须保持复位值。

位 21:16 **NBWE[5:0]**：预期的字数 (Number of words expected)

该位字段用于反映消息中必须推入到 FIFO 以触发部分计算的字数。当对 HASH_DIN 寄存器进行写访问时，NBWE 递减 1。

如果 HASH_CR 中的 INIT 位置 1，则 NBWE 将设置为预期的块大小 + 1（以字为单位）。当部分哈希计算结束后，该位将被设置为预期的块大小。

位 15 **DINNE**：DIN 非空 (DIN not empty)

当 HASH_DIN 寄存器包含有效数据时（也就是在至少写入一次之后），该位将置 1。当 INIT 位（初始化）或 DCAL 位（上一个消息处理已完成）置 1 时，该位将清零。

0：数据输入缓冲器中不存在任何数据

1：输入缓冲器至少包含一个字的数据

位 14:9 **NBWP[5:0]**：已推入的字数 (Number of words already pushed)

该位字段用于反映消息中已推入到 FIFO 的准确字数。当对 HASH_DIN 寄存器进行写访问时，NBWP 递增 1。

开始哈希计算后，NBWP 更新为 NBWP 块大小（以字为单位），并且当 INIT 位置 1 时，NBWP 将变为零。

位 8:4 保留，必须保持复位值。

位 3 **BUSY**：繁忙位 (Busy bit)

0：当前未在处理任何块

1：哈希内核正在处理某个数据块

位 2 **DMAS**：DMA 状态 (DMA Status)

该位提供有关 DMA 接口活动的信息。该位与 DMAE 一起置 1，并且在 DMAE = 0 且未在进行任何 DMA 传输时清零。没有任何中断与该位相关联。

0：禁止 DMA 接口 (DMAE = 0) 并且未在进行任何传输

1：使能 DMA 接口 (DMAE = 1) 或者某个传输在进行

位 1 **DCIS**：哈希计算完成中断状态 (Digest calculation completion interrupt status)

该位是在哈希值就绪时（也就是整个消息已处理完毕时）由硬件设置。要将该位清零，可写入 0，或者将 HASH_CR 寄存器中的 INIT 位置 1。

0：HASH_HRx 寄存器中无任何计算结果（返回零）

1：哈希计算已完成，结果存储于 HASH_HRx 寄存器中。如果 HASH_IMR 寄存器中将 DCIE 位置 1，则产生中断。

位 0 **DINIS**：数据输入中断状态 (Data input interrupt status)

该位可在 FIFO 准备好获取新块时（16 个位置空闲）由硬件置 1。要将该位清零，可写入 0，或者写入 HASH_DIN 寄存器。

0：输入缓冲器中的空闲位置少于 16 个

1：可以将一个新块输入到输入缓冲器中。如果 HASH_IMR 寄存器中将 DINIE 位置 1，则产生中断。

当 DINIS = 0 时，HASH_CSRx 寄存器读为零。

35.7.7 HASH 上下文交换寄存器

HASH context swap registers

这些寄存器包含哈希处理器的完整内部寄存器状态。如果高优先级的任务需要使用哈希处理器，而该处理器正在执行其他任务，则必须使用上述寄存器执行挂起/恢复操作。

发生此类事件时，必须读取 HASH_CSRx 寄存器，并且读取值必须保存到系统内存空间中。随后，可以使用哈希处理器执行高优先级的任务。完成哈希计算之后，可从内存中读取保存的上下文并将其写回 HASH_CSRx 寄存器。

仅当 DINIS 等于 1 时才能读取 HASH_CSRx 寄存器，否则读操作将返回零。

HASH 上下文交换寄存器 x (HASH_CSRx)

HASH context swap register x

偏移地址：0x0F8 + x * 0x4（x = 0 到 102）

复位值：0x0022 0002 (HASH_CSR0)

复位值：0x0020 0000 (HASH_CSR2)

复位值：0x0000 0000（其他）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSx[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSx[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CSx[31:0]**：上下文交换 x (Context swap x)
请参见*第 35.7.7 节：HASH 上下文交换寄存器*简介。

35.7.8 HASH 硬件配置寄存器 (HASH_HWCFGR)

HASH hardware configuration register

偏移地址：0x3F0

复位值：0x0001 0111

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFG5[3:0]			
												r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG4[3:0]				CFG3[3:0]				CFG2[3:0]				CFG1[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:20 保留，必须保持复位值。

位 19:16 **CFG5[3:0]**：HW 通用 5 (HW Generic 5)
该位字段反映支持 SHA384 和 SHA512 算法。

位 15:12 **CFG4[3:0]**：HW 通用 4 (HW Generic 4)
该位字段读为零。

位 11:8 **CFG3[3:0]**：HW 通用 3 (HW Generic 3)
该位字段反映支持 SHA3 算法。

位 7:4 **CFG2[3:0]**：HW 通用 2 (HW Generic 2)
该位字段反映支持 HMAC。

位 3:0 **CFG1[3:0]**：HW 通用 1 (HW Generic 1)
该位字段反映连接到该 HASH 外设的 DMA 类型。

35.7.9 HASH 版本寄存器 (HASH_VERR)

HASH version register

偏移地址：0x3F4

复位值：0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本 (Major revision)
这些位返回 HASH 主版本。

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)
这些位返回 HASH 次版本。

35.7.10 HASH 标识寄存器 (HASH_IPIDR)

HASH identification register

偏移地址：0x3F8

复位值：0x0017 0031

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：标识符 (Identifier)
这些位返回 HASH 外设的唯一标识符。

35.7.11 HASH 大小识别寄存器 (HASH_SIDR)

HASH size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识码 (size identification code)

该位字段返回 HASH 外设的大小标识码，定义如下：

位 [31:8] = 0xA3C5DD（固定代码）

位 [7:0] = 0x01（1 KB 地址解码）

35.7.12 HASH 寄存器映射

表 304. HASH1 寄存器映射和复位值

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	HASH_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALGO[3:2]		ALGO[1:0]		LKEY	Res.	DMAA	MDMAT	DINNE	NBW[3:0]				Res.	MODE	DATATYPE[1:0]		DMAE	INIT	Res.	Res.
	Reset value												0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0		
0x04	HASH_DIN	DATAIN[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	HASH_STR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCAL	Res.	Res.	Res.	NBLW[4:0]				
	Reset value																								0				0	0	0	0	0
0x0C	HASH_HR0	H0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	HASH_HR1	H1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	HASH_HR2	H2[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	HASH_HR3	H3[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	HASH_HR4	H4[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	HASH_IMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCIE	DINIE
	Reset value																															0	0
0x24	HASH_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBWE[5]	NBWE[4:0]				DINNE	NBWP[5]	NBWP[4:0]						Res.	Res.	Res.	Res.	Res.	BUSY	DMAS	DCIS	DINIS
	Reset value											0	1	0	0	0	1	0	0	0	0	0	0	0						0	0	0	1
Reserved																																	
0xF8	HASH_CSR0	CSR0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0xFC	HASH_CSR1	CSR1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x100	HASH_CSR2	CSR2[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...																																	
0x1CC	HASH_CSR53	CSR53[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...																																	
0x290	HASH_CSR102	CSR102[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Reserved																																	
0x310	HASH_HR0	H0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x314	HASH_HR1	H1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x318	HASH_HR2	H2[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x31C	HASH_HR3	H3[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x320	HASH_HR4	H4[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 304. HASH1 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x324	HASH_HR5	H5[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x328	HASH_HR6	H6[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x32C	HASH_HR7	H7[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...																																	
0x34C	HASH_HR15	H15[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		...																															
0x3B0	HASH_HR40	H40[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3B4	HASH_HR41	H41[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03F0	HASH_ HWCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFG5[3:0]				CFG4[3:0]				CFG3[3:0]				CFG2[3:0]				CFG1[3:0]			
	Reset value	0x0001 0111																															
0x03F4	HASH_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	1	0	0	0	0	0	0
0x03F8	HASH_IPIDR	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1
0x03FC	HASH_SIDR	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

36 公钥加速器 (PKA)

36.1 简介

PKA（公钥加速器）用于计算公钥加解密原语，特别是与 RSA、Diffie-Hellmann 或 GF(p)（Galois 域）上的 ECC（椭圆曲线加密）相关的公钥加解密原语。为了能够以合理的成本实现高性能，上述运算将在 Montgomery 域内执行。

对于给定操作，所有必要计算均在此加速器内执行，因此对于输入或输出，无需更多的软硬件上的处理。

操作密钥时，PKA 会加入一个防止侧通道攻击 (SCA) 的保护功能，包括差分功耗分析 (DPA)、且经 SESIP 和 PSA 安全保证级别 3 的认证。

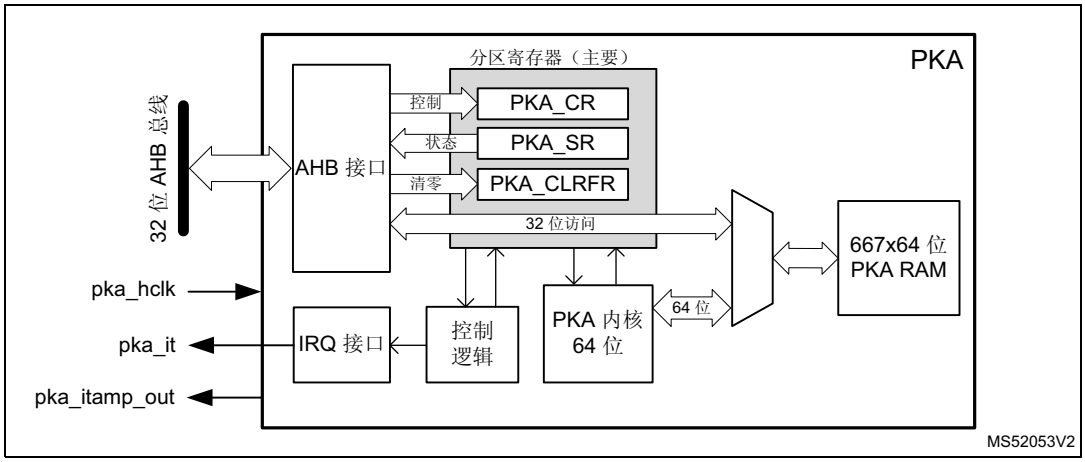
36.2 PKA 主要特性

- 加速 RSA、DH 和 GF(p) 上的 ECC 运算，基于 Montgomery 方法实现快速模乘。具体包括：
  - RSA 模幂、RSA 中国剩余定理 (CRT) 求幂
  - ECC 标量乘法、检查点是否在曲线上、完整加法、双基阶梯运算、仿射投影运算
  - ECDSA 签名生成与验证
- 能够处理多达 4160 位（针对 RSA/DH）和 640 位（针对 ECC）的操作数。
- 操作密钥时：防止侧通道攻击 (SCA) 的保护功能，包括差分功耗分析 (DPA)、且经 SESIP 和 PSA 安全保证级别 3 的认证
  - 适用于模幂、ECC 标量乘法和 ECDSA 签名生成
- 支持算术运算和模运算，例如加法、减法、乘法、模约简、模逆、比较和 Montgomery 乘法
- 内置 Montgomery 域内向和外向转换
- AMBA AHB 从外设，仅支持 32 位字单次访问（否则，会生成 AHB 总线错误，并会忽略写访问）

36.3 PKA 功能描述

36.3.1 PKA 框图

图 332. PKA 框图



36.3.2 PKA 内部信号

表 305 列出了 PKA 级所提供的内部信号（产品焊盘上不一定提供）。

表 305. 内部输入/输出信号

信号名称	信号类型	说明
pka_hclk	数字输入	AHB 总线时钟
pka_it	数字输出	公钥加速器 IP 全局中断请求
pka_itamp_out	数字输出	PKA 内部 Tamper 事件发送至 TAMP（异或），如果在 PKA 操作密钥时发生意外故障，或者在输入曲线上找不到编程输入点时，即会触发该事件（仅 ECDSA 签名和 ECC 标量乘法）。 一旦检测到故障，该信号即会被置为有效。置为有效后，对 PKA 寄存器的读访问为 0，写访问忽略。 当 PKA 内存清空时，该信号即置为无效。

36.3.3 PKA 复位和时钟

PKA 时钟由 AHB 总线时钟提供。PKA 外设复位信号释放后，PKA_RAM 在 667 个时钟周期后自动清空。在此期间，若设置 PKA_CR 寄存器的 EN 位，则忽略。

根据应用于设备的安全策略，PKA RAM 也可以在 Tamper 事件后复位。请参见系统安全部分的 *Tamper 检测和响应*（如果适用于本产品）。

36.3.4 PKA 公钥加速

概述

公钥加速器 (PKA) 用于加速 Rivest、Shamir 和 Adleman (RSA)、Diffie-Hellman (DH) 以及质数域上的 ECC 运算。支持的操作数大小多达 4160 位（针对 RSA 和 DH）和 640 位（针对 ECC）。

PKA 支持在质数域上定义的所有非奇异椭圆曲线，可以用简短的 Weierstrass 方程 $y^2 = x^3 + ax + b$（模 p）来描述。更多信息，请参见*第 36.5.1 节：支持的椭圆曲线*。

注： *PKA 不支持二元曲线、Edwards 曲线和 Curve25519。*

使用一个名为 PKA RAM 的 5336 字节（667 字，每个字 64 位）内存向 PKA 提供初始数据，并在计算完成后保存结果。通过 PKA AHB 接口进行访问。

PKA 运算模式

表 306 和*表 307* 分别针对整数运算函数和素域 (Fp) 椭圆曲线函数详细列出了 PKA 可执行的运算。

表 306. PKA 整数算术函数列表

PKA_CR.MODE[5:0]		执行的运算	参考
十六进制	二进制		
0x01	000001	Montgomery 参数计算 R2 mod n	*第 36.4.2 节*
0x0E	001110	模加 (A+B) mod n	*第 36.4.3 节*
0x0F	001111	模减 (A-B) mod n	*第 36.4.4 节*
0x10	010000	Montgomery 乘法 (AxB) mod n	*第 36.4.5 节*
0x00	000000	模幂 A^e mod n	*第 36.4.6 节*
0x02	000010	模幂 A^e mod n（快速模式）	
0x03	000011	模幂 A^e mod n（受保护）	*第 36.4.6 节*
0x08	001000	模逆 A^{-1} mod n	*第 36.4.7 节*
0x0D	001101	模约简 A mod n	*第 36.4.8 节*
0x09	001001	算术加法 A+B	*第 36.4.9 节*
0x0A	001010	算术减法 A-B	*第 36.4.10 节*
0x0B	001011	算术乘法 AxB	*第 36.4.11 节*
0x0C	001100	算术比较（A=B、A>B、A<B）	*第 36.4.12 节*
0x07	000111	RSA CRT 求幂	*第 36.4.13 节*

表 307. PKA 素域 (Fp) 椭圆曲线函数列表

PKA_CR.MODE[5:0]		执行的运算	参考
十六进制	二进制		
0x28	101000	检查点是否在椭圆曲线 Fp 上	*第 36.4.14 节*
0x20	100000	ECC 标量乘法 kP（受保护）	*第 36.4.15 节*
0x23	100011	ECC 完整加法	*第 36.4.18 节*
0x24	100100	ECDSA 签名（受保护）	*第 36.4.16 节*
0x26	100110	ECDSA 验证	*第 36.4.17 节*
0x27	100111	ECC 双基阶梯运算	*第 36.4.19 节*
0x2F	101111	ECC 仿射投影运算	*第 36.4.20 节*

每个操作都具有一个关联码，并由代码写入 PKA_CR 寄存器的 MODE 位字段。如果此值并未在列表中，则直接忽略对 MODE 位字段的写操作，并触发运算错误 (OPERRF)。发生这种情况时，在选择新操作之前，必须先清除错误。

表 306 和 *表 307* 中有一些运算标记为受保护。表示这些运算在操作密钥期间将受到保护（RSA 解密的模幂、ECC 的标量乘法和签名）。这些密钥（防止侧信道攻击）会在操作结束时自动从 PKA RAM 中擦除（BUSY 变为低电平）。它们同样受到保护，以防侧信道攻击。

小心： 出于安全原因，在执行 RSA 解密时选择受保护的模幂运算 (MODE = 0x3) 非常重要。

Montgomery 空间与快速模式运算

为了提高效率，PKA 在 Montgomery 域内部执行模乘运算，以便自动执行内向和外向转换。

由于 Montgomery 参数计算十分耗时，因此应用程序可决定使用快速运算模式，在此运算模式下会提供预先计算好的 Montgomery 参数。*第 36.5.2 节：计算时间*对性能改进情况进行了详细介绍。

唯一可使用快速模幂运算的模式为：MODE = 0x02。

36.3.5 PKA 的典型应用

简介

PKA 可用于加速多种公钥加密函数。尤其是：

- RSA 加密和解密
- RSA 密钥完成
- CRT-RSA 解密
- DSA 和 ECDSA 签名生成与验证
- DH 和 ECDH 密钥协商

以下出版物给出了上述函数的相关规范：

- FIPS PUB 186-4，数字签名标准 (DSS)，2013 年 7 月，NIST 著
- PKCS #1，RSA 加密标准，v1.5、v2.1 和 v2.2。RSA 实验室提供
- IEEE1363-2000，公钥加密的 IEEE 标准规范，2000 年 1 月
- ANSI X9.62-2005，金融服务业用公钥加密：椭圆曲线数字签名算法 (ECDSA)，2005 年 11 月

本节介绍主要函数的原理，有关更详细的说明，请参见上面引用的文档。

RSA 密钥对

对于以下 RSA 运算，公钥和私钥信息定义如下：

- Alice 将其公钥 (*n*, *e*) 传送给 Bob。数字 *n* 和 *e* 均为非常大的正整数。
- Alice 对其私钥 d（也是一个非常大的正整数）进行保密。或者，该私钥也可以由一个五元组 (p, q, dp, dq, qInv) 表示。

有关上述说明的更多信息，请参见 RSA 规范。

RSA 加密/解密原理

根据 PKCS#1 规范的建议，如果 Bob 要使用 Alice 的公钥 (*n*, *e*) 加密消息 M，必须经过以下步骤：

1. 计算编码消息 EM = ENCODE(M)，其中 ENCODE 为编码方法。
2. 将 EM 转换为整数 *m*，$0 \le m < n$，且 (*m*, *n*) 互质。
3. 计算密文 $c = m^e \bmod n$。
4. 将整数 c 转换为字符串密文 C。

如果 Alice 要使用其私钥 *d* 来解密密文 c，则应按以下步骤操作：

1. 将密文 C 转换为整数密文 c。
2. 如有必要，使用 (n, e, d) 信息检索质数因子 (p, q)，然后计算 phi = (p - 1) * (q - 1)。有关详细信息，请参见 NIST SP800-56B 附录 C。
3. 恢复明文 $m = c^d \bmod n = (m^e)^d \bmod n$。如果私钥是五元组 (p, q, dp, dq, qInv)，则明文 *M* 将通过执行以下运算获得：
   a) $m_1 = c^{dp} \bmod p$
   b) $m_2 = c^{dq} \bmod q$
   c) $h = qInv\ (m_1 - m_2) \bmod p$
   d) $m = m_2 + h\ q$
4. 将整数消息 *m* 转换为编码的消息 EM。
5. 恢复消息 M = DECODE(EM)，其中 DECODE 为解码方法。

上述操作可使用 PKA 来加速运算：当私钥为 *d* 时，使用*模幂* $A^e \bmod n$，当私钥为一个五元组 (p, q, dp, dq, qInv) 时，使用 *RSA CRT 求幂*。

注： *PKCS#1 标准中规定了消息与整数之间的解码运算和转换运算。*

注： *对于解密过程，出于安全原因，强烈建议使用受保护的模幂运算 (MODE = 0x3)。对于加密过程，不能使用 MODE = 0x3，因为需要知道私钥。*

椭圆曲线选择

对于以下 ECC 运算，曲线参数定义如下：

- 曲线对应于参与方（Alice 和 Bob）之间约定的椭圆曲线域。*第 36.5.1 节：支持的椭圆曲线*汇总了支持的曲线参数。
- G 是选择的椭圆曲线基点（也称为发生器），具有较大质数阶 n（即，*n* x G = 无限远点元素 O）。

ECDSA 消息签名生成

ECDSA（椭圆曲线数字签名算法）签名生成函数的原理如下：Alice 要使用她的私钥 d_A（整数）签名消息 *m*，所以执行以下步骤。

1. 计算 e = HASH(*m*)，其中 HASH 为密码学算法哈希函数。
2. 使 z 成为 e 最左边的 L_n 个位，其中 L_n 为阶数 *n* 的位长度。
3. 选择一个随机数，整数 k，其中 $0 < k < n$。
4. 计算曲线点 $(x_1,y_1) = k \times G$。
5. 计算 $r = x_1 \bmod n$。如果 $r = 0$，返回第 3 步。
6. 计算 $s = k^{-1} (z + rd_A) \bmod n$。如果 $s = 0$，返回第 3 步。
7. 最终签名结果即为 (*r*, *s*) 这个参数对。

PKA 可通过以下方式加速步骤 4 到 7：

- *ECDSA 签名*或
- 以下所有运算：
  - *ECC Fp 标量乘法* k x P
  - *模约简* A mod *n*
  - *模逆* A^{-1} mod *n*
  - *模加* 和 *模数和 Montgomery 乘法*

ECDSA 签名验证

ECDSA（椭圆曲线数字签名算法）签名验证函数原理如下：Bob 想要验证 Alice 的签名，他必须拥有她的公钥曲线点 Q_A 的副本。

Bob 可通过以下步骤验证 Q_A 是否为有效的曲线点：

1. 确认 Q_A 不等于无限远点元素 O
2. 确认 Q_A 位于约定的曲线上
3. 确认 $n \times Q_A = O$。

然后，Bob 执行下面详述的步骤：

1. 验证 *r* 和 *s* 是否为 [1, n-1] 内的整数
2. 计算 *e* = HASH(*m*)，其中 HASH 为约定的密码学算法哈希函数
3. 使 z 成为 *e* 最左边的 L_n 个位
4. 计算 $w = s^{-1} \bmod n$
5. 计算 $u_1 = zw \bmod n$ 以及 $u_2 = rw \bmod n$
6. 计算曲线点 $(x_1, y_1) = u_1 \times G + u_2 \times Q_A$
7. 如果 $r = x_1 \pmod n$，则签名有效，否则无效。

PKA 可通过 *ECDSA 验证*加速步骤 4 到 7。

36.3.6 PKA 执行过程

使能/禁止 PKA

将 PKA_CR 寄存器中的 EN 位置 1 可使能 PKA 外设。当 PKA_SR 中的 INITOK 位置 1 时，PKA 变为可用。当 EN=0 时，PKA 外设保持复位状态，应用程序仍可通过 AHB 接口访问 PKA 内存。

注： *在 PKA 清空内存期间，EN 位无法置 1。*

注： *PKA_CR 中的 EN 位置 1 时，应确保 MODE 位字段的值对应于合法的 PKA 运算（请参见第 36.3.7 节 的 OPERRF 相关内容）。*

在计算过程中将 EN 位清零会导致运算中止。在这种情况下，PKA 内存的内容无法得以保证，但 PKA 模式 0x03、0x20 和 0x24 除外。对于这些运算，PKA 内存在中止后将被清空，从而使内存持续 667 个周期不可用。在清空期间，只能访问 PKA 寄存器，将忽略对 EN 位的写入。

如果 INITOK 位保持为 0，应确保 RNG 外设已计时并正确初始化，然后尝试再次使能 PKA。

数据格式

*第 36.4 节*为每种运算规定了 PKA RAM 中的输入数据和结果的格式。

执行 PKA 运算

按照以下步骤执行支持的每种 PKA 运算：

1. 将初始数据装载至 PKA 内部 RAM（位于偏移地址 0x400 处）。
2. 通过对 PKA_CR 寄存器的 MODE 位字段进行写操作来指定要执行的运算，然后将同样位于 PKA_CR 寄存器中的 START 位置为有效。
3. 等待至 PKA_SR 寄存器中的 PROCENDF 位置 1，表示计算已完成。
4. 从 PKA 内部 RAM 读取结果数据，然后通过将 PKA_CLRFR 中的 PROCENDFC 位置 1 来清零 PROCENDF 位。

注： *PKA 忙 (BUSY = 1) 时，将忽略应用程序对 PKA RAM 进行的所有访问，PKA_SR 中的标志 RAMERRF 将置 1。*

如果在 *2* 中选择了非法或未知运算，将会触发 OPERRF 错误，并且永远不会执行步骤 *3* (PROCENDF = 1)。有关详细信息，请参见*第 36.3.7 节*。

使用预先计算的 Montgomery 参数（PKA 快速模式）

如*第 36.3.4 节*所述，当执行多个具有相同模数的运算时，如果应用程序只需计算一次相应的 Montgomery 参数，则会十分有益（例如，请参见*第 36.4.5 节*）。这便是所谓的“快速模式”。

要管理快速模式的使用，建议遵循下述步骤：

1. 在 PKA RAM 中装入模数大小和值信息。相关信息请参见*第 36.5.1 节*。
2. 在 PKA_CR 寄存器中将 PKA 编程为 *Montgomery 参数计算*模式（MODE=“0x1”），然后将 START 位置 1。
3. 等待至 PKA_SR 寄存器中的 PROCENDF 位置 1，然后从 PKA 内存读回相应的 Montgomery 参数，接着通过将 PKA_CLRFR 中的 PROCENDFC 位置 1 来清零 PROCENDF 位。
4. PKA 执行计算 Montgomery 参数（R2 mod m），其输入数据所涉及到的所有地址信息请参考*第 36.4 节*。

36.3.7 PKA 错误管理

使用 PKA 时，可能会出现以下错误：

- 访问 PKA RAM 时超出了预期范围。此时，PKA_SR 寄存器中的地址错误标志 (ADDRERRF) 置 1。
- 对正在被 PKA 内核使用的 PKA RAM 发起 AHB 访问。此时，PKA_SR 寄存器中的 RAM 错误标志 (RAMERRF) 置 1，读取 PKA RAM 将返回零，而写操作将被忽略。
- 使用 MODE 位字段选择的运算模式未在 PKA 运算模式（或在位字段说明）中列出，或者 PKA 正在受限模式下运行（请参见 PKA_SR 中的 LMF 位）。在这种情况下，PKA_SR 寄存器中的操作错误标志 (OPERRF) 置 1，并将忽略对 MODE 位字段的写入。

对于上述每个错误标志，如果应用程序将 PKA_CR 寄存器中的相应位置 1，则 PKA 会产生一个中断（有关详细信息，请参见*第 36.6 节*）。

通过将 PKA_CLRFR 中的相应位置 1 来清除 ADDRERRF、OPERRF 和 RAMERRF 错误。

可随时通过将 PKA_CR 寄存器中的 EN 位置 0 来重新初始化 PKA。

必须先使用 PKA_CLRFR 中的 OPERRFC 位清除 OPERRF 错误，然后才能在 PKA_CR 寄存器中开启新运算。

36.4 PKA 运算模式

36.4.1 简介

以下各小节介绍 PKA 支持的各种运算及其相关输入数据和结果的格式（均存储在 PKA RAM 中）。

警告： **必须先检查所有输入参数对于 PKA 的有效性，然后才能开始运算，因为 PKA 假定所有参数均有效并且相互一致。输入参数不得超过运算表中指定的操作数大小。**

以下信息适用于所有 PKA 运算。

- PKA 内核在其 RAM 中处理 64 位字。因此后文中所有字长均为 64 位
- 当一个元素作为输入写入 PKA RAM 时，必须在最高有效输入字之后添加一个所有位均为 0 的附加字。如果操作数的大小固定为 1，则此规则不适用。
- 所有报告的 RAM 存储地址均指数据的最低有效字，要获得使用的实际地址，应用程序必须将 PKA 的基址加上所指示的偏移。
- 支持的操作数“大小”如下：
  - ROS（RSA 操作数大小）：数据大小为 (*rsa_size*/64 + 1) 个字，其中 *rsa_size* 等于所选模数长度（位）。例如，当计算操作数大小为 1024 位的 RSA 时，ROS 等于 17 个字或 1088 位。
  - EOS（ECC 操作数大小）：数据大小为 (*ecc_size*/64 + 1) 个字，其中 *ecc_size* 等于所选质数模长度（位）。例如，当计算操作数大小为 192 位的 ECC 时，EOS 等于 4 个字或 256 位。
  - ROS 和 EOS 值包括所需的额外的全 0 字。
- 除非另有说明，否则表中的所有操作数均为整数。

注： *由于 PKA 内核处理 64 位字，如果上述公式的结果为小数，则会舍入为最接近的整数。*

注： *最大 ROS 为 66 个字（最大指数大小为 4160 位），而最大 EOS 为 11 个字（最大操作数大小为 640 位）。*

作为第一个示例（为了更好地理解 PKA 内存中的字节顺序），准备 *ECC Fp 标量乘法*运算，当应用程序写入 ECC P256 曲线的 P 点的 x 坐标（EOS = 5 个字）时，最低有效位必须放在位 0（偏移地址 0x578）中，最高有效位必须放在位 63（偏移地址 0x590）。然后，如上所述，应用程序必须在偏移地址 0x598 的位置入空字 0x0000000000000000。

作为第二个示例，仍然要准备 ECC Fp 标量乘法运算，当应用程序需要在模数长度为 224 位（即四个 64 位字舍入整数加上 1）的曲线上写入信息 a = -3 时，PKA 内存中必须写入以下数据：

```
@RAM+410  0x0000000000000001  /* 曲线系数 'a' 签名，没有额外的字 */
@RAM+418  0x0000000000000011  /* |a| 的值的最低有效位
@RAM+420  0x0000000000000000  ...
@RAM+428  0x0000000000000000  ...
@RAM+430  0x0000000000000000  |a| 的值的最高有效位 */
@RAM+438  0x0000000000000000  /* 附加全 0 字*/
```

36.4.2 Montgomery 参数计算

此函数用于计算 PKA 使用的 Montgomery 参数 (R^2 mod *n*)，以将操作数转换为 Montgomery 剩余系统表示形式。当使用快速模式运算时，该运算非常有用，因为在这种情况下，Montgomery 参数作为输入传递，从而节省了计算时间。

注： *此运算也可与 ECC 曲线搭配使用。此时，必须使用质数模长度和 EOS 大小。*

表 308 汇总了 Montgomery 参数计算的运算说明。

表 308. Montgomery 参数计算

带方向的参数		值（注）	存储	大小
IN	MODE	0x01	PKA_CR	6 位
	模数长度	（以位为单位，0 ≤ 值 < 4160）	RAM@0x408	64 位
	模数值 n	（仅限奇数，n < 2^{4160}）	RAM@0x1088	ROS
OUT	结果：R^2 mod n	-	RAM@0x620	

36.4.3 模加

模加运算计算的是 A + B mod *n*。*表 309* 汇总了运算说明。

表 309. 模加

带方向的参数		值（注）	存储	大小
IN	MODE	0x0E	PKA_CR	6 位
	操作数长度	（以位为单位，非空）	RAM@0x408	64 位
	操作数 A	(0 ≤ A < n)	RAM@0xA50	ROS
	操作数 B	(0 ≤ B < n)	RAM@0xC68	
	模数值 n	($n < 2^{4160}$)	RAM@0x1088	
OUT	结果：A+B mod n	（0 ≤ 结果 < n）	RAM@0xE78	

36.4.4 模减

模减运算的计算如下：

- 如果 A ≥ B，则结果等于 A - B mod *n*
- 如果 A < B，则结果等于 A + n - B mod *n*

表 310 汇总了运算说明。

表 310. 模减

带方向的参数		值（注）	存储	大小
IN	MODE	0x0F	PKA_CR	6 位
	操作数长度	（以位为单位，非空）	RAM@0x408	64 位
	操作数 A	(0 ≤ A < n)	RAM@0xA50	ROS
	操作数 B	(0 ≤ B < n)	RAM@0xC68	
	模数值 n	($n < 2^{4160}$)	RAM@0x1088	
OUT	结果：A-B mod n	（0 ≤ 结果 < n）	RAM@0xE78	

36.4.5 模数和 Montgomery 乘法

为提高乘法序列的执行效率，PKA 会对使用 Montgomery 域中的至少一个输入的乘法进行加速。此运算的两个主要用途如下：

- 将值从自然域映射到 Montgomery 域，或反之
- 执行模乘运算 A x B mod *n*

下面介绍了执行上述运算的方法。请注意，“x”函数表示此运算，A、B、C 操作数处于自然域中。

1. Montgomery 域的内向和外向转换
   a) 假设 A 是自然域中的一个整数：
      – 使用 *Montgomery 参数计算*计算 *r2modn*。
      – 结果 AR = A x *r2modn* mod *n* 是 Montgomery 域中的 A。
   b) 假设 BR 是 Montgomery 域中的一个整数：
      – 结果 B = BR x 1 mod *n* 为自然域中的 B。
      – 类似地，通过计算 A = AR x 1 mod *n*，可以将 a) 中计算出的上述值 AR 转换为自然域。
2. 单模乘法运算 A x B mod *n*
   a) 使用 *Montgomery 参数计算*计算 *r2modn*。
   b) 计算 AR = A x *r2modn* mod *n*。输出位于 Montgomery 域中。
   c) 计算 AB= AR x B mod *n*。输出位于自然域中。
3. 多模乘法运算 A x B x C mod *n*
   a) 使用 *Montgomery 参数计算*计算 *r2modn*。
   b) 计算 AR = A x *r2modn* mod *n*。输出位于 Montgomery 域中。
   c) 计算 BR = B x *r2modn* mod *n*。输出位于 Montgomery 域中。
   d) 计算 ABR= AR x BR mod *n*。输出位于 Montgomery 域中。
   e) 计算 CR = C x *r2modn* mod *n*。输出位于 Montgomery 域中。
   f) 计算 ABCR= ABR x CR mod *n*。输出位于 Montgomery 域中。
   g) （可选）如果需要对更多操作数进行乘法运算，请重复上述两个步骤。
   h) 计算 ABC = ABCR x 1 mod *n* 以获取自然域中的结果。

表 311 汇总了 Montgomery 乘法的运算说明。

表 311. Montgomery 乘法

带方向的参数		值（注）	存储	大小
IN	MODE	0x10	PKA_CR	6 位
	操作数长度	（以位为单位，非空）	RAM@0x408	64 位
	操作数 A	(0 ≤ A < n)	RAM@0xA50	ROS
	操作数 B	(0 ≤ B < n)	RAM@0xC68	
	模数值 n	（仅限奇数，$n < 2^{4160}$）	RAM@0x1088	
OUT	结果：AxB mod n(1)	-	RAM@0xE78	

1. 结果在 Montgomery 域或自然域中，具体取决于输入性质（参见示例 *2* 和 *3*）。

36.4.6 模幂

模幂运算通常用于执行单步 RSA 运算。其计算公式为 A^e mod *n*。

涉及公共信息的 RSA 运算（RSA 加密）可以使用普通或快速模式，详细信息请参见*表 312*和*表 313*。出于安全原因，涉及机密信息的 RSA 运算（RSA 解密）必须使用*表 314* 中详述的受保护模式。

注： *一旦该运算开始，PKA 控制寄存器和 PKA 内存就不再可用。BUSY 位由 PKA 置 0 后，即可恢复访问。*

当该运算因意外硬件事件而发生错误时，对于 TAMP 外设，将触发 PKA Tamper 事件，并会阻止对 PKA RAM 的访问，直到该事件由硬件擦除。

注： *当 MODE = 0x03 时，如果错误输出与 0xD60D 不同，则 PKA 将清空所有内存内容，以避免泄漏有关私钥的信息。*

表 312（正常模式）、*表 313*（快速模式）和*表 314*（受保护模式）中汇总了模幂运算的运算说明。*第 36.3.6 节*介绍了快速模式的使用方法。

表 312. 模幂运算（正常模式）

带方向的参数		值（注）	存储	大小
IN	MODE	0x00	PKA_CR	6 位
IN	指数长度	（以位为单位，非空）	RAM@0x400	64 位
	操作数长度	（以位为单位，非空）	RAM@0x408	
IN/OUT	操作数 A（幂的基数）	(0 ≤ A < n)	RAM@0xC68	ROS
IN	指数 e	(0 ≤ e < n)	RAM@0xE78	
	模数值 n	（仅限奇数，$n < 2^{4160}$）	RAM@0x1088	
OUT	结果：A^e mod n	（0 ≤ 结果 < n）	RAM@0x838	

表 313. 模幂运算（快速模式）

带方向的参数		值（注）	存储	大小
IN	MODE	0x02	PKA_CR	6 位
IN	指数长度	（以位为单位，非空）	RAM@0x400	64 位
	操作数长度	（以位为单位，非空）	RAM@0x408	
IN/OUT	操作数 A（幂的基数）	(0 ≤ A < n)	RAM@0xC68	ROS
IN	指数 e	(0 ≤ e < n)	RAM@0xE78	
	模数值 n	（仅限奇数，$n < 2^{4160}$）	RAM@0x1088	
IN/OUT	Montgomery 参数 R2 mod n	（强制）	RAM@0x620	
OUT	结果：A^e mod n	（0 ≤ 结果 < n）	RAM@0x838	

表 314. 模幂运算（受保护模式）

带方向的参数		值（注）	存储	大小
IN	MODE	0x03	PKA_CR	6 位
	指数 e 长度	（以位为单位，非空）	RAM@0x400	64 位
	模数或操作数长度	（以位为单位，非空）	RAM@0x408	
	操作数 A（幂的基数）	(0 ≤ A < n)	RAM@0x16C8	ROS
	指数 e	(0 ≤ e < n)	RAM@0x14B8	
	模数值 n	（仅限奇数，$n < 2^{4096}$）	RAM@0x0838	
	Phi 值(1)	-	RAM@0x0C68	
OUT	结果：$A^e \bmod n$	（0 ≤ 结果 < n）	RAM@0x838	
ERROR	错误 $A^e \bmod n$	– 无错误：0xD60D – 错误：0xCBC9	RAM@0x1298	64 位

1. n^1 的欧拉函数，phi = (p - 1) * (q - 1)，其中 p 和 q 是模数 n 的质数因数（详细信息请参见 NIST SP800-56B 附录 C 或 *RSA 加密/解密原理*）。作为一种优化手段，建议将 phi 信息作为密钥对生成的一部分予以保留。另一种方法是将密钥存储为 (*p*, *q*, *e*, *d*) 而不是 (*N*, *e*, *d*, *phi*)，此时，使用 PKA 算术乘法器得出 *N* 和 *phi*：*N* = *p* * *q*，*phi* = (*p* - 1) * (*q* - 1)。

36.4.7 模逆

模逆运算计算的是乘法逆元 $A^{-1} \bmod n$。如果模数 *n* 为质数，则对于 A (1 ≤ A < *n*) 的所有值，模数逆运算输出均有效。如果模数 *n* 不是素数，则只有当 A 与 *n* 之间的最大公约数为 1 时，A 才可逆。

如果操作数 A 是模数 *n* 的约数，则结果将是因子 *n* 的因数的倍数。

表 315 汇总了模逆运算的运算说明。

表 315. 模逆

带方向的参数		值（注）	存储	大小
IN	MODE	0x08	PKA_CR	6 位
	操作数长度	（以位为单位，非空）	RAM@0x408	64 位
	操作数 A	(0 ≤ A < n)	RAM@0xA50	ROS
	模数值 n	（仅限奇数，$n < 2^{4160}$）	RAM@0xC68	
OUT	结果：$A^{-1} \bmod n$	0 < 结果 < n	RAM@0xE78	

36.4.8 模约简

模约简运算计算的是 A 除以 n 所得的余数。*表 316* 汇总了运算说明。

表 316. 模约简

带方向的参数		值（注）	存储	大小
IN	MODE	0x0D	PKA_CR	6 位
	操作数长度	（以位为单位，非空）	RAM@0x400	64 位
	模数长度	（以位为单位，8 < 值 < 4160）	RAM@0x408	
	操作数 A	$(0 \le A < 2n < 2^{4160})$	RAM@0xA50	ROS
	模数值 n	（仅限奇数，$n < 2^{4160}$）	RAM@0xC68	
OUT	结果：A mod n	（0 < 结果 < n）	RAM@0xE78	

36.4.9 算术加法

算术加法运算的计算公式为 A + B。*表 317* 汇总了运算说明。

表 317. 算术加法

带方向的参数		值（注）	存储	大小
IN	MODE	0x09	PKA_CR	6 位
	操作数长度 M	（以位为单位，非空）	RAM@0x408	64 位
	操作数 A	$(0 \le A < 2^{M})$	RAM@0xA50	ROS
	操作数 B	$(0 \le B < 2^{M})$	RAM@0xC68	
OUT	结果：A+B	（$0 \le$ 结果 $< 2^{M+1}$）	RAM@0xE78	ROS + 1

36.4.10 算术减法

算术减法运算的计算如下：

- 如果 $A \ge B$，则结果等于 A - B
- 如果 A < B 且 M/32 余数 > 0，则结果等于 $A + 2^{int(M/32)*32+1} - B$
- 如果 A < B 且 M/32 余数为 0，则结果等于 $A + 2^{int(M/32)*32} - B$

对于后两点，输出的最高有效字后面的 64 位字等于 0xFFFF FFFF FFFF FFFF，因为结果为负。

表 318 汇总了运算说明。

表 318. 算术减法

带方向的参数		值（注）	存储	大小
IN	MODE	0x0A	PKA_CR	6 位
	操作数长度 M	（以位为单位，非空）	RAM@0x408	64 位
	操作数 A	$(0 \leq < A < 2^M)$	RAM@0xA50	ROS
	操作数 B	$(0 \leq < B < 2^M)$	RAM@0xC68	
OUT	结果：A-B	（$0 \leq$ 结果 $< 2^M$）	RAM@0xE78	

36.4.11 算术乘法

算术乘法运算计算 AxB。*表 319* 汇总了运算说明。

表 319. 算术乘法

带方向的参数		值（注）	存储	大小
IN	MODE	0x0B	PKA_CR	6 位
	操作数长度 M	（以位为单位，非空）	RAM@0x408	64 位
	操作数 A	$(0 \leq A < 2^M)$	RAM@0xA50	ROS
	操作数 B	$(0 \leq B < 2^M)$	RAM@0xC68	
OUT	结果：AxB	（$0 \leq$ 结果 $< 2^M$）	RAM@0xE78	2xROS

36.4.12 算术比较

算术比较运算的计算如下：

- 如果 A = B，则结果 = 0xED2C
- 如果 A > B，则结果 = 0x7AF8
- 如果 A < B，则结果 = 0x916A

表 320 汇总了算术比较的运算说明。

表 320. 算术比较

带方向的参数		值（注）	存储	大小
IN	MODE	0x0C	PKA_CR	6 位
	操作数长度 M	（以位为单位，非空）	RAM@0x408	64 位
	操作数 A	$(0 \leq A < 2^M)$	RAM@0xA50	ROS
	操作数 B	$(0 \leq B < 2^M)$	RAM@0xC68	
OUT	结果：A?B	0xED2C、0x7AF8 或 0x916A	RAM@0xE78	64 位

36.4.13 RSA CRT 求幂

为了提高效率，许多常用加密库（如 OpenSSL RSA）均根据中国剩余定理 (CRT) 在解密和签名方面进行了以下优化：

- *p* 和 *q* 是预先计算的素数，作为私钥的一部分进行存储
- $d_P = d \bmod (p-1)$
- $d_Q = d \bmod (q-1)$，
- $q_{inv} = q^{-1} \bmod p$

这些值有助于接收方更加高效地求幂 $m = A^d \pmod{pq}$，具体如下：

- $m_1 = A^{dP} \bmod p$
- $m_2 = A^{dQ} \bmod q$
- $h = q_{inv} (m_1 - m_2) \bmod p$，其中 $m_1 > m_2$
- $m = m_2 + hq \bmod pq$

表 321 汇总了用于计算 CRT 求幂 $A^d \bmod pq$ 的运算说明。

表 321. CRT 求幂

带方向的参数		值（注）	存储	大小
IN	MODE	0x07	PKA_CR	6 位
IN	操作数长度	（以位为单位，非空）	RAM@0x408	64 位
IN	操作数 d_P	$(0 < d_P < 2^{M/2})$	RAM@0x730	ROS/2
	操作数 d_Q	$(0 < d_Q < 2^{M/2})$	RAM@0xE78	
	操作数 q_{inv}	$(0 < q_{inv} < 2^{M/2})$	RAM@0x948	
	素数 *p*(1)	$(0 < p < 2^{M/2})$	RAM@0xB60	
	素数 *q*(1)	$(0 < q < 2^{M/2})$	RAM@0x1088	
IN	操作数 A	$(0 < A < 2^M)$	RAM@0x12A0	ROS
OUT	结果：$A^d \bmod pq$	（0 ≤ 结果 < *pq*）	RAM@0x838	

1. 不得为 2。

36.4.14 检查点是否在椭圆曲线 Fp 上

此运算检查给定点 P (x, y) 是否在素域曲线上，计算公式为 $y^2 = (x^3 + ax + b) \bmod p$，其中 *a* 和 *b* 是曲线的元素。

表 322 汇总了关于检查点是否在椭圆曲线 Fp 上的运算说明。

表 322. 检查点是否在椭圆曲线 Fp 上

带方向的参数		值（注）	存储	大小
IN	MODE	0x28	PKA_CR	6 位
	模数长度	（以位为单位，非空， 8 < 值 < 640）	RAM@0x408	64 位
	曲线系数 *a* 符号	0x0：正 0x1：负	RAM@0x410	
	曲线系数 |a|	（绝对值，|a| < p）	RAM@0x418	EOS
	曲线系数 *b*	(|b| < p)	RAM@0x520	
	曲线模数值 *p*	（奇素数，0 < p < 2640）	RAM@0x470	
	点 P 坐标 *x*	(x < *p*)	RAM@0x578	
	点 P 坐标 *y*	(y < *p*)	RAM@0x5D0	
	Montgomery 参数 R2 mod n	-	RAM@0x4C8	
OUT	结果：点 P 在曲线上	– 0xD60D：点在曲线上 – 0xA3B7：点不在曲线上 – 0xF946：x 或 y 坐标不小于模 p	RAM@0x680	64 位

36.4.15 ECC Fp 标量乘法

此运算的计算公式为 k x P (x_P, y_P)，其中 P 表示素域曲线上的一个点，“x”表示椭圆曲线标量点乘法。计算结果为属于同一曲线的点或无穷远的点。

表 323 汇总了 ECC Fp 标量乘法的运算说明。

注： *一旦该运算开始，PKA 控制寄存器和 PKA 内存就不再可用。BUSY 位由 PKA 置 0 后，即可恢复访问。*

当该运算因意外硬件事件而发生错误时，对于 TAMP 外设，将触发 PKA Tamper 事件，并会阻止对 PKA RAM 的访问，直到该事件由硬件擦除。如果在输入 ECC 曲线上找不到编程设定的输入点，也会触发 PKA Tamper 事件。可以使用 PKA 运算“检查点是否在椭圆曲线上”来避免这种情况。

表 323. ECC Fp 标量乘法

带方向的参数		值（注）	存储	大小
IN	MODE	0x20	PKA_CR	6 位
IN	曲线素数阶 *n* 长度	（以位为单位，非空）	RAM@0x400	64 位
	曲线模数 *p* 长度	（以位为单位，非空， 8 < 值 < 640）	RAM@0x408	
	曲线系数 *a* 符号	– 0x0：正 – 0x1：负	RAM@0x410	

表 323. ECC Fp 标量乘法（续）

带方向的参数		值（注）	存储	大小
IN	曲线系数 |*a*|	（绝对值，|a| < p）	RAM@0x418	EOS
	曲线系数 *b*	（正整数）	RAM@0x520	
	曲线模数值 *p*	（奇素数，0 < p < 2^{640}）	RAM@0x1088	
	标量乘数 k	(0 ≤ k < 2^{640})	RAM@0x12A0	
	点 P 坐标 x_P	(x < *p*)	RAM@0x578	
	点 P 坐标 y_P	(y < *p*)	RAM@0x470	
	曲线素数阶 *n*	（素数）	RAM@0xF88	
OUT	结果：k x P 坐标 *x'*	（结果 < *p*）	RAM@0x578	
	结果：k x P 坐标 *y'*	（结果 < *p*）	RAM@0x5D0	
ERROR	k x P 错误	– 无错误：0xD60D – 错误：0xCBC9	RAM@0x680	64 位

执行此运算时，应注意以下特殊情况：

- 对于 k = 0 的情况，此函数返回无穷远处的点 (0, 0)，如果曲线参数 b 非零则返回 (0, 1)。对于 k 不为 0 的情况，可能会返回无穷远处的点。当应用程序检测到此行为时，必须重新执行计算。
- 对于 k < 0（即，需要负标量乘法）的情况，应 k = |-k| 必须提供给 PKA。计算完成后，公式 -P = (x, -y) 计算有效最终结果的 y 坐标（x 坐标保持不变）。

注： *如果错误输出与 0xD60D 不同，则 PKA 将清空所有内存内容，以避免泄漏有关私钥的信息。*

36.4.16 ECDSA 签名

表 324（输入参数）和*表 325*（输出参数）汇总了 ECDSA 签名运算（在*第 36.3.5 节*进行了概要介绍）。

应用程序必须检查输出错误是否等于 0xD60D，如果不等于 0xD60D，则必须生成新的 k 并重新执行 ECDSA 签名运算。

注： *一旦该运算开始，PKA 控制寄存器和 PKA 内存就不再可用。BUSY 位由 PKA 置 0 后，即可恢复访问。*

当该运算因意外硬件事件而发生错误时，对于 TAMP 外设，将触发 PKA Tamper 事件，并会阻止对 PKA RAM 的访问，直到该事件由硬件擦除。如果在输入 ECC 曲线上找不到编程设定的输入点，也会触发 PKA Tamper 事件。可以使用 PKA 运算“检查点是否在椭圆曲线上”来避免这种情况。

表 324. ECDSA 签名——输入

带方向的参数		值（注）	存储	大小
IN	MODE	0x24	PKA_CR	6 位
	曲线素数阶 n 长度 (*nlen*)	（以位为单位，非空）	RAM@0x400	64 位
	曲线模数 p 长度	（以位为单位，8 < 值 < 640）	RAM@0x408	
	曲线系数 a 符号	0x0：正 0x1：负	RAM@0x410	
	曲线系数 |a|	（绝对值，|a| < p）	RAM@0x418	EOS
	曲线系数 b	（正整数）	RAM@0x520	
	曲线模数值 *p*	（奇素数，0< p < 2^{640}）	RAM@0x1088	
	整数 k(1)	(0 ≤ k < 2^{640})	RAM@0x12A0	
	曲线基点 G 坐标 x	(x < *p*)	RAM@0x578	
	曲线基点 G 坐标 y	(y < *p*)	RAM@0x470	
	消息 z 的散列值	（散列大小等于 *nlen*）(2)	RAM@0xFE8	
	私钥 d	(0 < d)	RAM@0xF28	
	曲线素数阶 n	（素数）	RAM@0xF88	

1. 此整数通常是加解密级别的安全的随机数，但在某些情况下也可以是生成的确定值 k。
2. 必须使用零填充或散列截断来使散列参数大小等于曲线素数阶 n 长度。

表 325. ECDSA 签名——输出

带方向的参数		值（注）	存储	大小
OUT	签名部分 *r*	(0 < *r* < n)	RAM@0x730	EOS
	签名部分 *s*	(0 < *s* < n)	RAM@0x788	
ERROR	签名结果	– 0xD60D：计算成功，无错误 – 0xCBC9：计算失败 – 0xA3B7：签名部分 r 等于 0 – 0xF946：签名部分 s 等于 0	RAM@0xFE0	64 位

注： *如果错误输出等于 0xD60D 或 0xCBC9，则 PKA 将清空所有内存内容，以避免泄漏有关私钥的信息。如果错误输出等于 0xA3B7 或 0xF946，则 PKA 内存内容会被部分擦除，从而保持错误代码可读。*

扩展 ECDSA 支持

PKA 还支持扩展 ECDSA 签名，对应的输入和输出与 ECDSA 签名的输入输出相同（分别参见*表 324* 和*表 325*），此外还添加了点 kG 的坐标。此额外输出在*表 326* 中定义。

表 326. 扩展 ECDSA 签名——额外输出

带方向的参数		值（注）	存储	大小
OUT	曲线点 kG 坐标 x_1	(0 ≤ x_1 < p)	RAM@0x1400	EOS
	曲线点 kG 坐标 y_1	(0 ≤ y_1 < p)	RAM@0x1458	

36.4.17 ECDSA 验证

表 327（输入参数）和*表 328*（输出参数）汇总了 ECDSA 验证运算（在*第 36.3.5 节*进行了概要介绍）。

应用程序必须检查输出错误是否等于 0xD60D，如果不等于 0xD60D，则不会验证签名。

表 327. ECDSA 验证——输入

带方向的参数		值（注）	存储	大小
IN	MODE	0x26	PKA_CR	6 位
	曲线素数阶 n 长度 (*nlen*)	（以位为单位，非空）	RAM@0x408	64 位
	曲线模数 p 长度	（以位为单位，非空， 8 < 值 < 640）	RAM@0x4C8	
	曲线系数 a 符号	0x0：正 0x1：负	RAM@0x468	
	曲线系数 |a|	（绝对值，|a| < p）	RAM@0x470	EOS
	曲线模数值 *p*	（奇素数，0< p < 2^{640}）	RAM@0x4D0	
	曲线基点 G 坐标 x	(x < *p*)	RAM@0x678	
	曲线基点 G 坐标 y	(y < *p*)	RAM@0x6D0	
	公钥曲线点 Q 坐标 x_Q	(x_Q < *p*)	RAM@0x12F8	
	公钥曲线点 Q 坐标 y_Q	(y_Q < *p*)	RAM@0x1350	
	签名部分 *r*	(0 < *r* < n)	RAM@0x10E0	
	签名部分 *s*	(0 < *s* < n)	RAM@0xC68	
	消息 *z* 的散列值	（散列大小等于 *nlen*）(1)	RAM@0x13A8	
	曲线素数阶 *n*	（素数）	RAM@0x1088	

1. 必须使用零填充或散列截断来使散列参数大小等于曲线素数阶 n 长度。

表 328. ECDSA 验证——输出

带方向的参数		值（注）	存储	大小
OUT	结果：ECDSA 验证	– 0xD60D：有效签名 – 0xA3B7：无效签名	RAM@0x5D0	64 位
	已计算的签名部分 *r*	– (0 < *r* < n)	RAM@0x578	EOS

36.4.18 ECC 完整加法

ECC 完整加法计算椭圆曲线上两个给定点的加法。

表 329 汇总了运算说明。

注： *两个输入点和计算结果的点以 Jacobian 坐标 (X,Y,Z) 表示。要在仿射坐标 (x,y) 中输入一个点，可以使用 (X,Y,Z) = (x,y,1) 转换。要将生成的点转换为雅可比坐标，可以使用 (x,y)= $(X/Z^2, Y/Z^3)$ 转换。*

表 329. ECC 完整加法

带方向的参数		值（注）	存储	大小
IN	MODE	0x23	PKA_CR	6 位
	曲线模数 *p* 长度	（以位为单位，非空， 8 < 值 < 640）	RAM@0x408	64 位
	曲线系数 *a* 符号	– 0x0：正 0x1：负	RAM@0x410	
	曲线模数值 *p*	（奇素数，$0< p < 2^{640}$）	RAM@0x470	EOS
	曲线系数 |*a*|	（绝对值，|a| < p）	RAM@0x418	
	第一点 P 坐标 X	(x < *p*)	RAM@0x628	
	第一点 P 坐标 Y	(y < *p*)	RAM@0x680	
	第一点 P 坐标 Z	(z < *p*)	RAM@0x6D8	
	第二点 Q 坐标 X	(x < *p*)	RAM@0x730	
	第二点 Q 坐标 Y	(y < *p*)	RAM@0x788	
	第二点 Q 坐标 Z	(z < *p*)	RAM@0x7E0	
OUT	结果坐标 X	(x < *p*)	RAM@0xD60	
	结果坐标 Y	(y < *p*)	RAM@0xDB8	
	结果坐标 Z	(z < *p*)	RAM@0xE10	

36.4.19 ECC 双基阶梯运算

ECC 双基阶梯运算计算 k*P+m*Q，其中 (P,Q) 是椭圆曲线上的两个点，(k,m) 是两个标量。*表 330* 汇总了运算说明。

如果计算出来的点是无穷远点（错误代码 0xA3B7），则结果坐标等于 (0, 0)。

注： *两个输入点以雅可比坐标 (X、Y、Z) 表示。要在仿射坐标 (x, y) 中输入一个点，可以使用 (X, Y, Z) = (x, y, 1) 转换。结果以仿射坐标 (x ,y) 表示*

表 330. ECC 双基阶梯运算

带方向的参数		值（注）	存储	大小
IN	MODE	0x27	PKA_CR	6 位
	曲线素数阶 *n* 长度	（以位为单位，非空）	RAM@0x400	64 位
	曲线模数 *p* 长度	（以位为单位，非空， 8 < 值 < 640）	RAM@0x408	
	曲线系数 *a* 符号	– 0x0：正 – 0x1：负	RAM@0x410	EOS
	曲线系数 |*a*|	（绝对值，|a| < p）	RAM@0x418	
	曲线模数值 *p*	（奇素数，0< p < 2^{640}）	RAM@0x470	
	整数 k	(0 < k < 2^{640})	RAM@0x520	
	整数 m	(0 < m < 2^{640})	RAM@0x578	
	第一点 P 坐标 X	(x < *p*)	RAM@0x628	
	第一点 P 坐标 Y	(y < *p*)	RAM@0x680	
	第一点 P 坐标 Z	(z < *p*)	RAM@0x6D8	
	第二点 Q 坐标 X	(x < *p*)	RAM@0x730	
	第二点 Q 坐标 Y	(y < *p*)	RAM@0x788	
	第二点 Q 坐标 Z	(z < *p*)	RAM@0x7E0	
OUT	结果坐标 x	(x < *p*)	RAM@0x578	EOS
	结果坐标 y	(y < *p*)	RAM@0x5D0	
	错误代码	– 点不在无穷远：0xD60D – 点在无穷远：0xA3B7	RAM@0x520	64 位

36.4.20 ECC 仿射投影运算

ECC 仿射投影运算计算齐次投影坐标中的点 P 的表示与仿射坐标中的点 P 的表示之间的转换。即，如果该点由三元坐标 (X, Y, Z) 表示，则计算仿射坐标 (x, y) = (X/Z, Y/Z)。

所有运算均以该点所属曲线的模 p 为模进行。如果结果点是无穷远的点（错误代码 0xA3B7），则结果坐标等于 (0,0)。

表 331 汇总了运算说明。

表 331. ECC 仿射投影运算

带方向的参数		值（注）	存储	大小
IN	MODE	0x2F	PKA_CR	6 位
	曲线模数 *p* 长度	（以位为单位，8 < 值 < 640）	RAM@0x408	64 位
	曲线模数值 p	（奇素数，0< $p < 2^{640}$）	RAM@0x470	EOS
	点 P 坐标 X（投影）	(x < *p*)	RAM@0xD60	
	点 P 坐标 Y（投影）	(y < *p*)	RAM@0xDB8	
	点 P 坐标 Z（投影）	(z < *p*)	RAM@0xE10	
	Montgomery 参数 R2 mod n	-	RAM@0x4C8	
OUT	点 P 坐标 x（仿射）	(x < *p*)	RAM@0x578	
	点 P 坐标 y（仿射）	(y < *p*)	RAM@0x5D0	
ERROR	错误代码	– 点不在无穷远：0xD60D – 点在无穷远：0xA3B7	RAM@0x680	64 位

36.5 配置和处理时间示例

36.5.1 支持的椭圆曲线

PKA 支持在素数域上定义的所有非奇异椭圆曲线。这些曲线可以用一个简短的 Weierstrass 方程 $y^2 = x^3 + ax + b \pmod p$ 来描述。

注： *PKA 不支持二元曲线、Edwards 曲线和 Curve25519。*
ECC 运算支持的最大操作数大小为 640 位。

当发布这些椭圆曲线的 ECC 域参数时，标准化组织定义了以下参数：

- 素数 *p*，用作有限域 GF(*p*) 中所有点算术的模
- 整数（通常是质数）*n*，由 G 生成的群的阶数，定义如下
- 曲线 G 的基点，由其坐标 (*Gx*, *Gy*) 定义
- 整数 *a* 和 *b*，简短 Weierstrass 方程的系数。

对于最后一点，当标准机构将 *a* 定义为负值时，PKA 支持两种表示：

1. 在有限域 GF(*p*) 中，**a 定义为 *p*-|*a*|**，例如**p-3**：
   曲线系数 p = 0xFFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFF**F**
   曲线系数 a 符号 = 0x0（正）
   曲线系数 a = 0xFFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFF**C**
2. ***a* 定义为负数**，例如 **-3**：
   曲线系数 p = 0xFFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFFF
   曲线系数 a 符号 = 0x**1**（负）
   曲线系数 a = 0x00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 0000000**3**

表 332 总结了针对 ECC 运算 PKA 所支持的曲线族。

表 332. 针对 ECC 运算而支持的曲线族

曲线名称	标准	参考	
P-192	NIST	*数字签名标准 (DSS)*，NIST FIPS 186-4	
P-224			
P-256			
P-384			
P-521			
brainpoolP224r1、brainpoolP224t1	IETF	– *Brainpool 椭圆曲线*，IETF RFC 5639 – *用于互联网密钥交换 (IKE) 组描述注册表的 Brainpool 椭圆曲线*，IETF RFC 6932	https://tools.ietf.org
brainpoolP256r1、brainpoolP256t1			
brainpoolP320r1、brainpoolP320t1			
brainpoolP384r1、brainpoolP384t1			
brainpoolP512r1、brainpoolP512t1			
secp192k1、secp192r1	SEC	*高效密码学标准 SEC 2 曲线*	https://www.secg.org
secp224k1、secp224r1			
secp256k1、secp256r1			
secp384r1			
secp521r1			
公钥密码算法 SM2 推荐曲线参数	OSCCA	– *Sm2 椭圆曲线公钥密码算法，中华人民共和国商务部国家密码管理局*，中国国家商务部国家密码管理局 SM2 标准，2010 年 12 月 – *数字签名 - 第 3 部分 基于离散对数的机制*，ISO/IEC 14888-3，2018 年 11 月	

36.5.2 计算时间

下表总结了 PKA 计算时间（以 AHB 时钟周期数表示）。

表 333. 模幂

指数长度（位）	模式	模数长度（位）			
		1024	2048	3072	4096
3	正常	66000	242000	310000	503000
	快速	7000	23000	48000	82000
17	正常	69000	253000	333000	543000
	快速	24000	34000	72000	123000
$2^{16}+1$	正常	82000	296000	427000	705000
	快速	23000	78000	165000	284000
1024	受保护	2960000	-	-	-
	正常	1725000	-	-	-
	快速	1666000	-	-	-
	CRT(1)	595000	-	-	-
2048	受保护	-	17643000	-	-
	正常	-	11553000	-	-
	快速	-	11334000	-	-
	CRT(1)	-	3454000	-	-
3072	受保护	-	-	53390000	-
	正常	-	-	36280000	-
	快速	-	-	36019000	-
	CRT(1)	-	-	10328000	-
4096	受保护	-	-	-	119824000
	正常	-	-	-	82619000
	快速	-	-	-	82198000
	CRT(1)	-	-	-	23061000

1. CRT 代表中国余数定理优化（MODE 位字段 = 0x07）。

表 334. ECC 标量乘法(1)

模数长度（位）							
160	192	256	320	384	512	521	640
-	742000	1220000	2116000	3031000	5981000	7844000	10200000

1. 这些时间取决于标量参数中包含的 1 的数量，并包括 Montgomery 参数 R2 的计算。

表 335. ECDSA 签名平均计算时间(1) (2)

模数长度（位）							
160	**192**	**256**	**320**	**384**	**512**	**521**	**640**
-	633000	1087000	1708000	2548000	4856000	5978000	8030000

1. 这些值为给定长度的随机模数的平均执行时间，因为它们取决于模数的长度和值。
2. 定义 NIST 椭圆曲线有限域的模数所需的执行时间比用于 Brainpool 椭圆曲线的模数或相同大小的随机模数所需的执行时间短。

表 336. ECDSA 验证平均计算时间

模数长度（位）							
160	**192**	**256**	**320**	**384**	**512**	**521**	**640**
566000	676000	1221000	1957000	2967000	5825000	7096000	9773000

表 337. ECC 双基阶梯运算平均计算时间

模数长度（位）							
160	**192**	**256**	**320**	**384**	**512**	**521**	**640**
541000	644000	1152000	1873000	2814000	5506000	6753000	9350000

表 338. ECC 仿射投影平均计算时间

模数长度（位）						
160	**192**	**256**	**320**	**384**	**512**	**640**
24700	32000	56000	90000	144000	271000	326000

表 339. ECC 完整加法平均计算时间

模数长度（位）						
160	**192**	**256**	**320**	**384**	**512**	**640**
8000	8200	10500	13000	19400	29600	39000

表 340. 检查点是否在椭圆曲线 Fp 上平均计算时间

模数长度（位）							
160	**192**	**256**	**320**	**384**	**512**	**521**	**640**
2200	2240	2870	3550	4400	6260	7460	8500

表 341. Montgomery 参数平均计算时间(1)

模数长度（位）							
192	256	320	512	1024	2048	3072	4096
4200	5200	5900	12600	35500	126500	290000	420000

1. 计算时间取决于模数的长度和值，因此这些值为给定长度的随机模数的平均执行时间。

36.6 PKA 中断

公钥加速器可生成四个独立的可屏蔽中断源，用以通知发生以下事件：

1. PKA 不支持的运算错误 (OPERRF)，请参见*第 36.3.7 节*
2. 访问未映射的地址 (ADDRERRF)，请参见*第 36.3.7 节*
3. 在执行 PKA 运算时访问 PKA RAM (RAMERRF)，请参见*第 36.3.7 节*
4. PKA 运算结束 (PROCENDF)

这四个中断源连接到同一全局中断请求信号 pka_it。

用户可通过更改 *PKA 控制寄存器 (PKA_CR)* 中的屏蔽位，单独使能或禁止上述中断源。将相应的屏蔽位置 1 可使能中断。各个中断事件的状态可从 PKA 状态寄存器 (PKA_SR) 中读取，并在 PKA_CLRFR 寄存器中清除。

表 342 对可用功能进行了总结。

表 342. PKA 中断请求

缩略语	事件	事件标志	使能控制位	清除方法
PKA	不支持的运算	OPERRF	OPERRIE	将 OPERRFC 位置 1
	访问未映射的地址错误	ADDRERRF	ADDRERRIE	将 ADDRERRFC 位置 1
	PKA RAM 访问错误	RAMERRF	RAMERRIE	将 RAMERRFC 位置 1
	PKA 运算结束	PROCENDF	PROCENDIE	将 PROCENDFC 位置 1

36.7 PKA 寄存器

36.7.1 PKA 控制寄存器 (PKA_CR)

PKA control register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OP ERRIE	ADDR ERRIE	RAM ERRIE	Res.	PROC ENDIE	Res.
										rw	rw	rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	MODE[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	START	EN
		rw	rw	rw	rw	rw	rw							rw	rw

位 31:22 保留，必须保持复位值。

位 21 **OPERRIE**：运算错误中断使能 (Operation error interrupt enable)
0：PKA_SR 中的 OPERRF 标志置 1 时，不生成中断。
1：PKA_SR 中的 OPERRF 标志置 1 时，生成中断。

位 20 **ADDRERRIE**：地址错误中断使能 (Address error interrupt enable)
0：如果 PKA_SR 中的 ADDRERRF 标志置 1，则不生成中断。
1：如果 PKA_SR 中的 ADDRERRF 标志置 1，则生成中断。

位 19 **RAMERRIE**：RAM 错误中断使能 (RAM error interrupt enable)
0：PKA_SR 中的 RAMERRF 标志置 1 时，不生成中断。
1：PKA_SR 中的 RAMERRF 标志置 1 时，生成中断。

位 18 保留，必须保持复位值。

位 17 **PROCENDIE**：操作结束中断使能 (End of operation interrupt enable)
0：PKA_SR 中的 PROCENDF 标志置 1 时，不生成中断。
1：PKA_SR 中的 PROCENDF 标志置 1 时，生成中断。

位 16:14 保留，必须保持复位值。

位 13:8 **MODE[5:0]**：PKA 操作代码 (PKA operation code)

000000：先进行 Montgomery 参数计算，然后进行模幂运算
000001：仅进行 Montgomery 参数计算
000010：仅进行模幂运算（必须先载入 Montgomery 参数）
000011：模幂（受保护，在操作秘密时使用）
100000：先进行 Montgomery 参数计算，然后进行 ECC 标量乘法（受保护）
100100：ECDSA 签名（受保护）
100110：ECDSA 验证
101000：检查点是否在椭圆曲线 Fp 上
000111：RSA CRT 求幂
001000：模逆
001001：算术加法
001010：算术减法
001011：算术乘法
001100：算术比较
001101：模约简
001110：模加
001111：模减
010000：Montgomery 乘法
100011：ECC 完整加法
100111：ECC 双基阶梯运算
101111：ECC 仿射投影运算

如果应用程序在 EN 位置 1 的情况下写入此处未列出的运算，则 PKA_SR 寄存器中的 OPERRF 位将置 1，并会忽略对 MODE 位字段的写入。当 PKA 配置为受限模式（PKA_SR 中 LMF = 1）时，如果写入 MODE 的值不为 0x26 且 EN 位置 1，则会触发 OPERRF 位置 1，并忽略对 MODE 位的写操作。

位 7:2 保留，必须保持复位值。

位 1 **START**：开始运算 (start the operation)

向该位写入 1 后，将使用已写入 PKA RAM 中的操作数和数据，开始进行 MODE[5:0] 选择的运算。该位始终读为 0。

如果在 START 位置 1 时选择了非法操作，则不会开始任何运算，并且 PKA_SR 中的 OPERRF 位置 1。

注意：如果 PKA 忙，则忽略 START。

位 0 **EN**：PKA 使能 (PKA enable)

0：禁止 PKA
1：使能 PKA当 PKA_SR 中的 INITOK 由硬件置 1 时，PKA 开始起作用。
如果当 EN = 1 时选择非法操作，PKA_SR 中的 OPERRF 位将置 1。有关详细信息，请参阅 PKA_CR.MODE 位字段。

注意：EN = 0 时，应用仍可访问 PKA RAM。

36.7.2 PKA 状态寄存器 (PKA_SR)

PKA status register

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OP ERRF	ADDR ERRF	RAM ERRF	Res.	PROC ENDF	BUSY
										r	r	r		r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LMF：	INITOK
														r	r

位 31:22 保留，必须保持复位值。

位 21 **OPERRF**：运算错误标志 (Operation error flag)

0：无事件错误

1：在 PKA_CR 寄存器中选择了非法或未知运算

该位使用 PKA_CLRFR 中的 OPERRFC 位清零。

位 20 **ADDRERRF**：地址错误标志 (Address error flag)

0：无地址错误

1：地址访问超出范围（未映射的地址）

该位使用 PKA_CLRFR 中的 ADDRERRFC 位清零。

位 19 **RAMERRF**：PKA RAM 错误标志 (PKA RAM error flag)

0：无 PKA RAM 访问错误

1：当 PKA 内核正在计算和使用其内部 RAM 时对 PKA RAM 进行了 AHB 访问（在进行 PKA 运算时不允许 AHB PKA_RAM 访问）。

该位使用 PKA_CLRFR 中的 RAMERRFC 位清零。

位 18 保留，必须保持复位值。

位 17 **PROCENDF**：PKA 运算结束标志 (PKA end of operation flag)

0：正在进行运算

1：PKA 运算已完成。当 BUSY 位清零时，该标志置 1。

位 16 **BUSY**：忙标志 (Busy flag)

只要 PKA 运算正在进行（PKA_CR 中的 START = 1），该位就会置 1。计算完成后该位会自动清请，使 PKA RAM 再次可访问。

0：当前未进行运算（默认）

1：正在进行运算

如果 PKA 以错误的操作码开始，将有若干个周期处于忙状态，然后将自动中止运算并返回到就绪状态 (BUSY = 0)。

位 15:2 保留，必须保持复位值。

位 1 **LMF**：受限模式标志 (Limited mode flag)

当 PKA_CR 中的 EN 位置 1 时，该位更新

0：可以使用 MODE 位字段中记录的所有值。

1：PKA 仅支持 ECDSA 验证 (MODE = 0x26)。

位 0 **INITOK**：PKA 初始化正常 (PKA initialization OK)
该位在 PKA 完成初始化时置为有效。当 RNG 无法输出正确的随机数时，INITOK 保持为 0。
0：PKA 未正确初始化。START 位无法置 1。
1：PKA 已正确初始化，可以正常使用。

36.7.3 PKA 清零标志寄存器 (PKA_CLRFR)

PKA clear flag register

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OP ERRFC	ADDR ERRFC	RAM ERRFC	Res.	PROC ENDFC	Res.
										w	w	w		w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:22 保留，必须保持复位值。

位 21 **OPERRFC**：清零运算错误标志 (Clear operation error flag)
0：无操作
1：将 PKA_SR 中的 OPERRF 标志清零

位 20 **ADDRERRFC**：清零地址错误标志 (Clear Address error flag)
0：无操作
1：将 PKA_SR 中的 ADDRERRF 标志清零

位 19 **RAMERRFC**：清零 PKA RAM 错误标志 (Clear PKA RAM error flag)
0：无操作
1：将 PKA_SR 中的 RAMERRF 标志清零

位 18 保留，必须保持复位值。

位 17 **PROCENDFC**：清零 PKA 运算结束标志 (Clear PKA End of Operation flag)
0：无操作
1：将 PKA_SR 中的 PROCENDF 标志清零

位 16:0 保留，必须保持复位值。

注： *读取 PKA_CLRFR 将返回全零。*

36.7.4 PKA RAM

PKA RAM 映射到偏移地址 0x0400（相对于 PKA 基址）。仅支持 32 位字单次访问（通过 PKA.AHB 接口）。

RAM 大小为 5336 字节（最大字偏移：0x14D0）

注： *PKA RAM 不能在 PKA 复位或产品复位后立即使用，如第 36.3.3 节：PKA 复位和时钟中所述。*

36.7.5 PKA 硬件配置寄存器 (PKA_HWCFGR)

PKA hardware configuration register

偏移地址：0x1FF0

复位值：0x0000 0011

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFG2[3:0]				CFG1[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **CFG2[3:0]**：HW 通用 2 (HW Generic 2)
该位字段返回 DPAEN 通用值 (0x1)

位 3:0 **CFG1[3:0]**：HW 通用 2 (HW Generic 2)
该位字段返回 PKA64_SEL 通用值 (0x1)

36.7.6 PKA 版本寄存器 (PKA_VERR)

PKA version register

偏移地址：0x1FF4

复位值：0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本 (Major revision)
该位字段返回 PKA 主版本。

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)
该位字段返回 PKA 次版本。

36.7.7 PKA 标识寄存器 (PKA_IPIDR)

PKA identification register

偏移地址：0x1FF8

复位值：0x0017 0072

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：标识码 (Identification code)
该位字段返回 PKA 标识码。

36.7.8 PKA 大小标识寄存器 (PKA_SIDR)

PKA size identification register

偏移地址：0x1FFC

复位值：0xA3C5 DD08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识码 (size identification code)
该位字段返回 PKA 的大小标识码，定义如下：
位[31:8] = 0xA3C5DD（固定代码）
位[7:0] = 0x08（8 KB 地址解码）

36.7.9 PKA 寄存器映射

表 343. PKA 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	PKA_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPERRIE	ADDRERRIE	RAMERRIE	Res.	PROCENDIE	Res.	Res.	Res.	MODE[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	START	EN
	Reset value											0	0	0		0				0	0	0	0	0	0							0	0

表 343. PKA 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x004	PKA_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPERRF	ADDRERRF	RAMERRF	Res.	PROCENDF	BUSY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LMF;	INITOK
	Reset value											0	0	0		0	0															0	0
0x008	PKA_CLRFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPERRFC	ADDRERRFC	RAMERRFC	Res.	PROCENDFC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value											0	0	0		0																	
0x1FF0	PKA_HWCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFG2[3:0]				CFG1[3:0]			
	Reset value																									0	0	0	1	0	0	0	1
0x1FF4	PKA_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV [3:0]				MINREV [3:0]			
	Reset value																									0	0	0	1	0	0	0	0
0x1FF8	PKA_IPIDR	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0
0x1FFC	PKA_SIDR	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

37 DDR 内存加密引擎 (DDRMCE)

37.1 简介

DDR 内存加密引擎 (DDRMCE) 在给定的地址空间（DDR-SDRAM 中的一个区域）定义了特定的加密区域。当系统总线通信通过这个加密区域时，DDRMCE 会自动管理这些数据，在经过授权的情况下自动执行解密读操作和加密写操作。

在本节以及本文档中，使用 MCE 缩写代替 DDRMCE。

37.2 MCE 主要特性

- 系统总线实时加密（对于写操作）和解密（对于读操作），基于嵌入式防火墙编程
  - 每个区域有两种加密模式（最多 1 个区域可用）：不加密（旁路模式）和块密码模式
  - 按照 64 KB 的粒度定义区域的起始地址和结束地址
  - 默认过滤（区域 0）：允许所有访问权限
  - 区域访问过滤条件：无
- 支持的块密码：AES
- 支持的链接模式：
  - 采用 AES 密码的块模式与 *NIST FIPS 出版物 197 高级加密标准 (AES)* 中指定的 ECB 模式兼容，相关的密钥派生函数基于发表在 *https://keccak.team* 网站上的 Keccak-400 算法。
  - 包括为防御侧信道攻击 (SCA) 的泄露弹性操作模式。
- 一组只写且可锁定的主密钥寄存器
- 读-写仲裁方案，可提高读取性能
- AHB 配置端口，特权感知
- AXI 系统总线主/从接口（64 位）
  - 支持所有 AXI-64 位读取事务
  - 使能加密时，支持 AXI-64 位 INCRx（x = 1 到 8）和 WRAPx (x = 4) 写入事务

37.3 MCE 实现

本产品包括一个与 DDR 控制器相关联的 MCE 实例。通过 MCE 的所有通信均由用于 DDR 的 TrustZone® 地址空间控制器 (TZC) 进行过滤。有关详细信息，请参见 *TZC*。

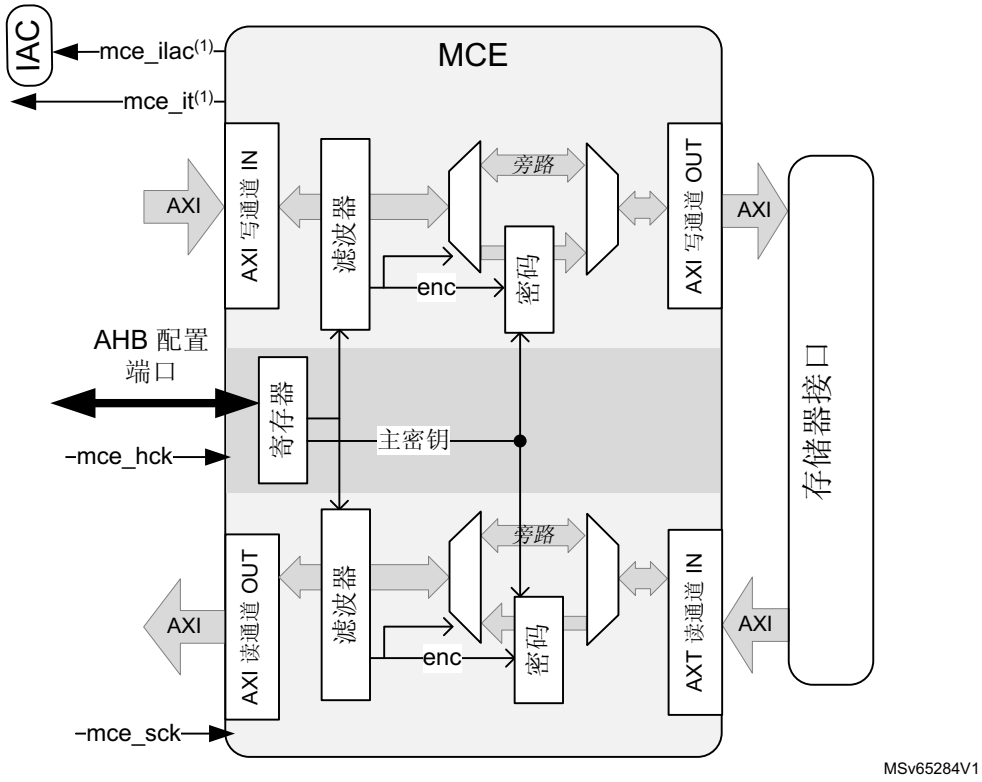
MCE 外设向 Cortex® CPU 中断控制器产生中断 (mce_it)。

37.4 MCE 功能描述

37.4.1 MCE 框图

图 333 显示了内存加密引擎的框图，与一个或多个内存接口外设相关。

图 333. MCE 框图



1. 请参见*第 37.3 节：MCE 实现*。

37.4.2 MCE 内部信号

表 344 描述了与用户相关的 MCE 外设内部信号。

表 344. MCE 内部输入/输出信号

信号名称	信号类型(1)	说明
mce_sck	输入	AXI 输入时钟
mce_hck	输入	MCE 的 AHB 输入时钟
mce_it	输出	MCE 中断请求

1. 全部为数字信号。

37.4.3 MCE 编程

当退出复位状态时，或者当访问一个不属于已使能区域的空间时，将应用默认区域过滤（即，授予所有访问权限）。

当应用程序需要使用加密区域时，可以使用以下步骤（例如在启动时）。

1. 应用程序使用 MCE_SADDR 和 MCE_EADDR 寄存器设置起始地址和结束地址。
2. 应用程序通过配置 MCE_REGCRx 设置安全参数（加密）来完成编程。有关初始化加密的详细信息，请参见*第 37.4.5 节*。
3. 当上述配置完成最终检查后，应用程序通过设置 MCE_REGCR 寄存器中的 BREN 位来启用加密区域。

当 MCE_CR 寄存器中的 GLOCK 置 1 时，MCE 配置可以保持固定，直到下一次 SoC 复位为止。

如果 MCE_PRIVCFGR 寄存器中的 PRIV 位置 1，则必须为上述应用程序授予特权访问。

37.4.4 MCE 复位和时钟

MCE 配置端口由 AHB 总线时钟提供时钟。

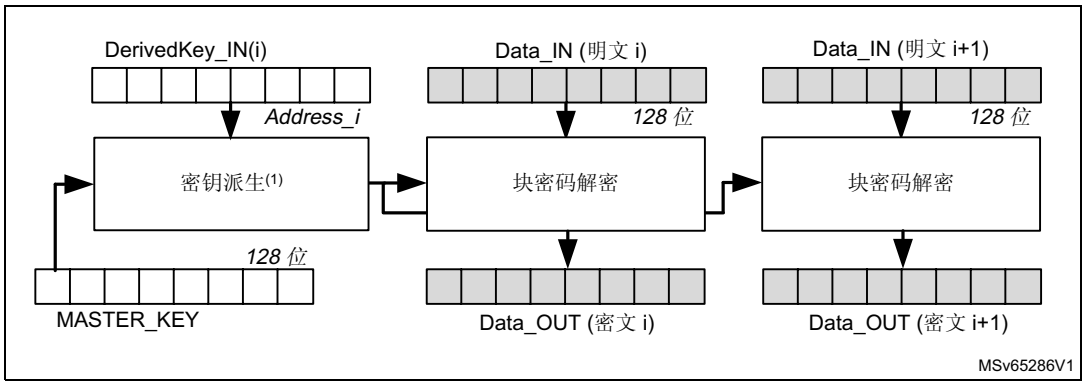
MCE 系统总线接口由 AXI 时钟提供时钟。无需为 AHB 接口提供时钟即可使用系统总线。

当 MCE 处于复位状态时，明文通信可以通过系统总线。退出复位状态时将应用默认防火墙配置，如*第 37.4.3 节*所述。

37.4.5 MCE 块密码加密模式

图 334 详细给出了 MCE 外设中的块密码实现。

图 334. MCE 块密码实现



1. 该操作从捕获地址开始。

Address_i 是块 (i) + 块 (i+1) 物理地址的最高有效位。密钥派生原语的结果可用于两个连续的 16 字节数据字。由于正常密钥派生功能为弹性泄漏，因此主密钥信息可防止侧信道攻击 (SCA)。

MASTER_KEY 必须存储在只写 MCE_MKEYRx 寄存器中。

每次密钥信息成功写入 MCE 后，以下标志将置 1：

- 当整个密钥写入 MCE_MKEYR 寄存器后，MCE_SR 中的 MKVALID 位将置 1。

如果上述标志未将所选密钥标记为有效，则选择旁路模式而不是密码模式。有关详细信息，请参见 ENC 位字段说明。

一轮密钥派生需要 15 个周期。一轮块密码加密需要 11 个周期。

注： *使用 AES 块密码时，写操作（加密）需要额外的 11 个周期，以执行所需的密钥调度来获取加密密钥。*

37.4.6 MCE AXI 通信管理

MCE 包含一种特殊的 AXI-64 读-写仲裁方案，旨在加速执行多次读操作。更具体来讲，当 MCE 中未进行写操作时，可以分配两个密码内核来并行解密两次读操作。在这种情况下，传入的写请求会延迟到有空闲的密码内核可以分配给写操作时再进行处理。

注： *当事务在两个 64 位字上对齐时，达到最佳性能。有关详细信息，请参见第 37.4.5 节。*

37.4.7 MCE 加密禁止选项

当产品中没有 MCE 功能时，无法使用 MCE 的加密/解密功能。这意味着通过 MCE 的读操作（或写操作）不会被解密（或加密）。

37.4.8 MCE 错误管理

当发生对*配置寄存器*的非法访问（权限不足）时，MCE_IASR 中相应的 CAEF 位将置 1。应用程序使用 MCE_IACR 中的 CAEF 位清零该标志。

读取只写寄存器或写入只读寄存器不会触发中断。读取任何 MCE 密钥寄存器都将返回零。

37.5 MCE 中断

MCE 可生成一个独立的可屏蔽中断源，用于通知发生以下错误事件：

- 通过配置总线进行非法访问 (CAEF)，请参见*第 37.4.8 节*

该中断源连接到全局中断请求信号 signal mce_it。

将 MCE_IAIER 寄存器中相应的屏蔽位置 1 可使能中断。各个中断事件的状态可从 MCE_IASR 寄存器中读取，并在 MCE_IACR 寄存器中清除。

表 345 对可用功能进行了总结。

表 345. MCE 中断请求

中断缩写	中断事件	事件标志(1)	使能控制位	中断清除方法
DDRMCE	配置访问错误	CAEF	CAEIE	将 MCE_IACR 中的 CAEF 位置 1

1. 在 MCE_IASR 寄存器中读取

37.6 MCE 寄存器

37.6.1 MCE 配置寄存器 (MCE_CR)

MCE configuration register

偏移地址：0x000

复位值：0x0000 0000

若 MCE_PRIVCFGR 中的 PRIV 置 1，则只允许特权读写访问。

如果 MCE_CR 寄存器中的 GLOCK 置 1，则忽略写操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MK LOCK	GLOCK
														rs	rs

位 31:2 保留，必须保持复位值。

位 1 **MKLOCK**：主密钥锁定 (Master keys lock)

锁定主密钥配置，直到下一次复位为止。该位默认清零，一旦置 1，便只能等到 MCE 复位时才会复位。

0：允许对 MCE_MKEYRx 寄存器进行写操作

1：忽略对 MCE_MKEYRx 寄存器的写操作，直到下一次 MCE 复位为止。

位 0 **GLOCK**：全局锁定 (Global lock)

锁定大多数寄存器的配置，直到下一次复位为止。该位默认清零，一旦置 1，便只能等到 MCE 复位时才会复位。

0：MCE 寄存器可写

1：忽略对 MCE 寄存器的所有写操作，但 MCE_IACR 和 MCE_IAIER 寄存器除外。

37.6.2 MCE 状态寄存器 (MCE_SR)

MCE status register

偏移地址：0x004

复位值：0x0000 0000

若 MCE_PRIVCFGR 中的 PRIV 置 1，则只允许特权读访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MK VALID
															r

位 31:5 保留，必须保持复位值。

位 4 保留，必须保持复位值。

位 3 保留，必须保持复位值。

位 2 保留，必须保持复位值。

位 1 保留，必须保持复位值。

位 0 **MKVALID**：主密钥有效 (Master key valid)

0：MCE_MKEYRx 寄存器中尚未写入有效密钥

1：MCE_MKEYRx 寄存器中已写入有效密钥（按照 R0 -> R1 -> R2 -> R3 顺序写入或逆序写入）

37.6.3 MCE 非法访问状态寄存器 (MCE_IASR)

MCE illegal access status register

偏移地址：0x008

复位值：0x0000 0000

若 MCE_PRIVCFGR 中的 PRIV 置 1，则只允许特权读访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CAEF
															r

位 31:1 保留，必须保持复位值。

位 0 **CAEF**：配置访问错误标志 (Configuration access error flag)

当检测到对任何 MCE 配置寄存器的非法访问时，该位置 1。该位通过将 MCE_IACR 寄存器中相应的位置 1 来清零。关于该错误，没有更多详细信息。

37.6.4 MCE 非法访问清零寄存器 (MCE_IACR)

MCE illegal access clear register

偏移地址：0x00C

复位值：0x0000 0000

若 MCE_PRIVCFGR 中的 PRIV 置 1，则只允许特权写访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CAEF
															w

位 31:1 保留，必须保持复位值。

位 0 **CAEF**：配置访问错误标志清零 (Configuration access error flag clear)
将该位置 1 会清零 MCE_IASR 寄存器中的 CAEF 位。

37.6.5 MCE 非法访问中断使能寄存器 (MCE_IAIER)

MCE illegal access interrupt enable register

偏移地址：0x010

复位值：0x0000 0000

若 MCE_PRIVCFGR 中的 PRIV 置 1，则只允许特权读写访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CAEIE
															rw

位 31:1 保留，必须保持复位值。

位 0 **CAEIE**：配置访问错误中断使能 (Configuration access error interrupt enable)
0：禁止在发生配置访问错误时产生中断
1：允许在发生配置访问错误（CAEF = 1）时产生中断

37.6.6 MCE 特权配置寄存器 (MCE_PRIVCFGR)

MCE privileged configuration register

偏移地址：0x1C

复位值：0x0000 0000

仅限特权写访问。

允许任意读访问。如果 MCE_CR 寄存器中的 GLOCK 置 1，则忽略写操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIV
															rw

位 31:1 保留，必须保持复位值。

位 0 **PRIV**：特权配置 (Privileged configuration)
0：授予对 MCE 寄存器的特权和非特权访问权限。
1：仅授予对 MCE 寄存器的特权访问，但 ID 寄存器（MCE_HWCFGR1/2、MCE_VERR、MCE_IPIDR 和 MCE_SIDR）除外

37.6.7 MCE 区域配置寄存器 (MCE_REGCR)

MCE region configuration register

偏移地址：0x040

复位值：0x0000 0000

若 MCE_PRIVCFGR 中的 PRIV 置 1，则只允许特权读写访问。如果 MCE_CR 寄存器中的 GLOCK 置 1，则忽略写操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BREN
rw															rw

位 31:16 保留，必须保持复位值。

位 15 **ENC**：加密区域 (Encrypted region)

仅当 BREN 置 1 时，才考虑该位。

0：无影响

1：使用块密码解密（或加密）所有允许的读取（或写入）请求。如果 MKVALID = 0，则改选旁路模式。

位 14:9 保留，必须保持复位值。

位 8:1 保留，必须保持复位值。

位 0 **BREN**：基础区域使能 (Base region enable)

0：区域禁用。该区域起始地址和结束地址之间的所有访问，遵循主要区域（特权或非特权，不加密）的访问控制规则。

1：区域启用。该区域起始地址和结束地址之间的所有访问，遵循该区域自定义的访问控制规则和加密选项。

如果 BADDRSTART > BADDREND，则无法将 BREN 置 1。

37.6.8 MCE 起始地址寄存器 (MCE_SADDR)

MCE start address register

偏移地址：0x044

复位值：0x0000 0000

若 MCE_PRIVCFGR 中的 PRIV 置 1，则只允许特权读写访问。如果 MCE_CR 寄存器中的 GLOCK 置 1，或者 MCE_REGCR 中的 BREN 置 1，则忽略写操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BADDSTART[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:16 **BADDSTART[31:16]**：区域起始地址 (Region address start)
该位字段定义区域在 64 KB 边界（含）上的绝对起始地址。
如果 BADDRSTART > BADDREND，则无法将 BREN 置 1。
当 MCE 确定区域时，忽略该寄存器中的前 16 位 (LSB) 和后 2 位 (MSB)，当读取该寄存器时，2 个 MSB 位和 16 个 LSB 位返回 0（MCE 中的参考值）。

位 15:0 保留，必须保持复位值。

37.6.9 MCE 结束地址寄存器 (MCE_EADDR)

MCE end address register

偏移地址：0x048

复位值：0x0000 FFFF

若 MCE_PRIVCFGR 中的 PRIV 置 1，则只允许特权读写访问。如果 MCE_CR 寄存器中的 GLOCK 置 1，或者 MCE_REGCR 中的 BREN 置 1，则忽略写操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BADDEND[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:16 **BADDEND[31:16]**：区域结束地址 (Region address end)
该位字段定义区域在 64 KB 边界（含）上的绝对结束地址。
如果 BADDRSTART > BADDREND，则无法将 BREN 置 1。
当 MCE 确定区域时，忽略该寄存器中的前 16 位 (LSB) 和后 2 位 (MSB)，当读取该寄存器时，2 个 MSB 位返回 0，16 个 LSB 位返回 1（MCE 中的参考值）。

位 15:0 保留，必须保持复位值。

37.6.10 MCE 主密钥 x (MCE_MKEYRx)

MCE master key x

偏移地址：0x200 + 0x4 * x（x = 0 到 3）

复位值：0x0000 0000

若 MCE_PRIVCFGR 中的 PRIV 置 1，则只允许特权写访问。

如果 MCE_CR 寄存器中的 MKLOCK 或 GLOCK 位置 1，则忽略写操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MKEY{i + 32 * x}															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MKEY{i + 32 * x}															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **MKEY{i +32 * x}**：主密钥位 {i + 32 * x} (Master key bit {i + 32 * x})（i = 31 到 0）
MCE 块密码使用该密钥

37.6.11 MCE 版本寄存器 (MCE_HWCFGR3)

MCE version register

偏移地址：0x3E8

复位值：0x3B00 0102

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CFG8[3:0]				CFG7[3:0]				CFG6[3:0]				CFG5[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG4[3:0]				CFG3[3:0]				CFG2[3:0]				CFG1[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **CFG8[3:0]**：HW 通用 8 (HW Generic 8)
该位字段返回通用值 *ooo_enable* | *interconnect_enable*

位 27:24 **CFG7[3:0]**：HW 通用 7 (HW Generic 7)
该位字段返回通用值 *id_size*

位 23:20 **CFG6[3:0]**：HW 通用 6 (HW Generic 6)
该位字段返回通用值 *priv_enable_axi*

位 19:16 **CFG5[3:0]**：HW 通用 5 (HW Generic 5)
该位字段返回通用值 *rif_enable*

位 15:12 **CFG4[3:0]**：HW 通用 4 (HW Generic 4)
该位字段返回通用值 *sec_enable*

位 11:8 **CFG3[3:0]**：HW 通用 3 (HW Generic 3)
该位字段返回通用值 *priv_enable*

位 7:4 **CFG2[3:0]**：HW 通用 2 (HW Generic 2)
该位字段返回通用值 *nk_core*

位 3:0 **CFG1[3:0]**：HW 通用 1 (HW Generic 1)
该位字段返回通用值 *aes_core*

37.6.12 MCE 版本寄存器 (MCE_HWCFGR2)

MCE version register

偏移地址：0x3EC

复位值：0x1107 1444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CFG8[3:0]				CFG7[3:0]				CFG6[3:0]				CFG5[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG4[3:0]				CFG3[3:0]				CFG2[3:0]				CFG1[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **CFG8[3:0]**：HW 通用 8 (HW Generic 8)
该位字段返回通用值 *nb_breg*

位 27:24 **CFG7[3:0]**：HW 通用 7 (HW Generic 7)
该位字段返回通用值 *nb_mkey*

位 23:20 **CFG6[3:0]**：HW 通用 6 (HW Generic 6)
该位字段返回通用值 *nb_cctx*

位 19:16 **CFG5[3:0]**：HW 通用 5 (HW Generic 5)
该位字段返回通用值 *nb_dpa_cycles*

位 15:12 **CFG4[3:0]**：HW 通用 4 (HW Generic 4)
该位字段返回通用值 *ilac_is_int*

位 11:8 **CFG3[3:0]**：HW 通用 3 (HW Generic 3)
该位字段返回通用值 *mask_fifo_depth*

位 7:4 **CFG2[3:0]**：HW 通用 2 (HW Generic 2)
该位字段返回通用值 *info_fifo_depth*

位 3:0 **CFG1[3:0]**：HW 通用 1 (HW Generic 1)
该位字段返回通用值 *burst_fifo_depth*

37.6.13 MCE 版本寄存器 (MCE_HWCFGR1)

MCE version register

偏移地址：0x3F0

复位值：0x0000 1D10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFG3[7:0]							
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG2[7:0]								CFG1[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 保留，必须保持复位值。

位 23:16 **CFG3[7:0]**：HW 通用 3 (HW Generic 3)
该位字段返回通用值 *opbits_gen*

位 15:8 **CFG2[7:0]**：HW 通用 2 (HW Generic 2)
该位字段返回通用值 *address_region_msb_max_bits*

位 7:0 **CFG1[7:0]**：HW 通用 1 (HW Generic 1)
该位字段返回通用值 *address_region_lsb_max_bits*

37.6.14 MCE 版本寄存器 (MCE_VERR)

MCE version register

偏移地址：0x3F4

复位值：0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本 (Major revision)
该位字段返回 MCE IP 主版本。

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)
该位字段返回 MCE IP 次版本。

37.6.15 MCE 标识寄存器 (MCE_IPIDR)

MCE identification register

偏移地址：0x3F8

复位值：0x0017 0081

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：标识码 (Identification Code)
该位字段返回 MCE IP 的标识码。

37.6.16 MCE 大小 ID 寄存器 (MCE_SIDR)

MCE size ID register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识码 (Size Identification Code)

该位字段返回 MCE IP 的大小标识码，定义如下：

位[31:8] = 0xA3C5DD（固定代码）

位[7:0] = 0x01（1KB 地址解码）

37.6.17 MCE 寄存器映射

表 346. MCE 寄存器映射和复位值

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**MCE_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MKLOCK	GLOCK
	Reset value																															0	0
0x004	MCE_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MKVALID
	Reset value																																0
0x008	MCE_IASR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CAEF
	Reset value																																0
0x00C	MCE_IACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CAEF
	Reset value																																0
0x010	MCE_IAIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CAEIE
	Reset value																																
0x01C	**MCE_ PRIVCFGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIV
	Reset value																																0
0x040	MCE_REGCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BREN
	Reset value																	0															0
0x044	**MCE_SADDR**	BADDSTART[31:16]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																
0x048	**MCE_EADDR**	BADDEND[31:16]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																

表 346. MCE 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x200+0x4 * x, (x = 0 to 3)	MCE_MKEYRx	MKEY{i + 32 * x} (i = 31 to 0)																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3E8	**MCE_HWCFGR3**	CFG8[3:0]				CFG7[3:0]				CFG6[3:0]				CFG5[3:0]				CFG4[3:0]				CFG3[3:0]				CFG2[3:0]				CFG1[3:0]			
	Reset value	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
0x3EC	**MCE_HWCFGR2**	CFG8[3:0]				CFG7[3:0]				CFG6[3:0]				CFG5[3:0]				CFG4[3:0]				CFG3[3:0]				CFG2[3:0]				CFG1[3:0]			
	Reset value	0	0	0	1	0	0	0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0
0x3F0	**MCE_HWCFGR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFG3[7:0]								CFG2[7:0]								CFG1[7:0]							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	1	0	0	0	0
0x3F4	**MCE_VERR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	0	0	1	0	0	0	0
0x3F8	**MCE_IPIDR**	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
0x3FC	**MCE_SIDR**	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

38 高级控制定时器 (TIM1/TIM8)

38.1 TIM1/TIM8 简介

高级控制定时器 (TIM1/TIM8) 包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

此类定时器可用于各种用途，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（输出比较、PWM 和带死区插入的互补 PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

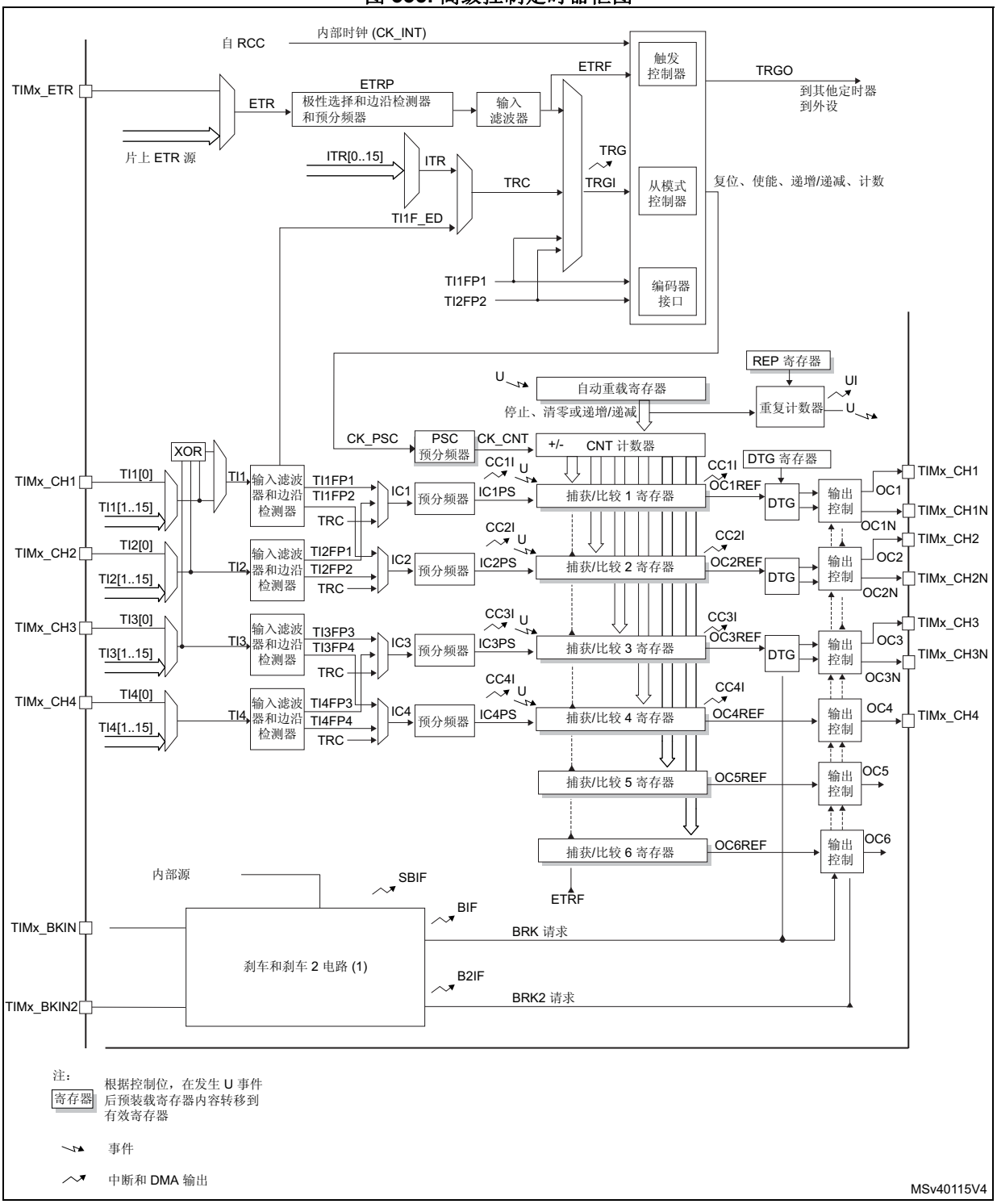
高级控制定时器 (TIM1/TIM8) 和通用定时器 (TIMy) 彼此完全独立，不共享任何资源。如*第 38.3.26 节：定时器同步*中所述，它们可以一起同步操作。

38.2 TIM1/TIM8 主要特性

TIM1/TIM8 定时器主要特性：

- 16 位递增、递减、递增/递减自动重载计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 到 65536 之间。
- 多达 6 个独立通道，可用于：
  - 输入捕获（但通道 5 和通道 6 除外）
  - 输出比较
  - PWM 生成（边沿和中心对齐模式）
  - 单脉冲模式输出
- 带可编程死区的互补输出
- 同步电路，用于实现外部信号对定时器的控制以及多个定时器互联。
- 重复计数器，用于仅在给定数目的计数周期后更新定时器寄存器。
- 2 个刹车输入，用于将定时器的输出信号置于用户可选的安全配置中。
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - 输入捕获
  - 输出比较
- 支持定位用增量（正交）编码器和霍尔传感器电路
- 触发输入可用作外部时钟或用于逐周期电流管理

图 335. 高级控制定时器框图



1. 详细信息，请参见*图 378：刹车事件和刹车 2 电路概述*。

38.3 TIM1/TIM8 功能描述

38.3.1 时基单元

可编程高级控制定时器的主要模块是一个 16 位计数器及其相关的自动重载寄存器。计数器可递增计数、递减计数或进行递增递减计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)
- 重复计数器寄存器 (TIMx_RCR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以随时直接写入影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值（或者在递减计数时达到下溢值）并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将产生更新事件。该更新事件也可由软件产生。下文将详细介绍每个配置中更新事件的产生方式。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

注意，计数器将在 TIMx_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。

预分频器说明

预分频器可对计数器时钟进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 TIMx_PSC 寄存器中的 16 位寄存器所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 336 和 *图 337* 以一些示例说明在预分频比实时变化时计数器的行为：

图 336. 预分频器分频由 1 变为 2 时的计数器时序图

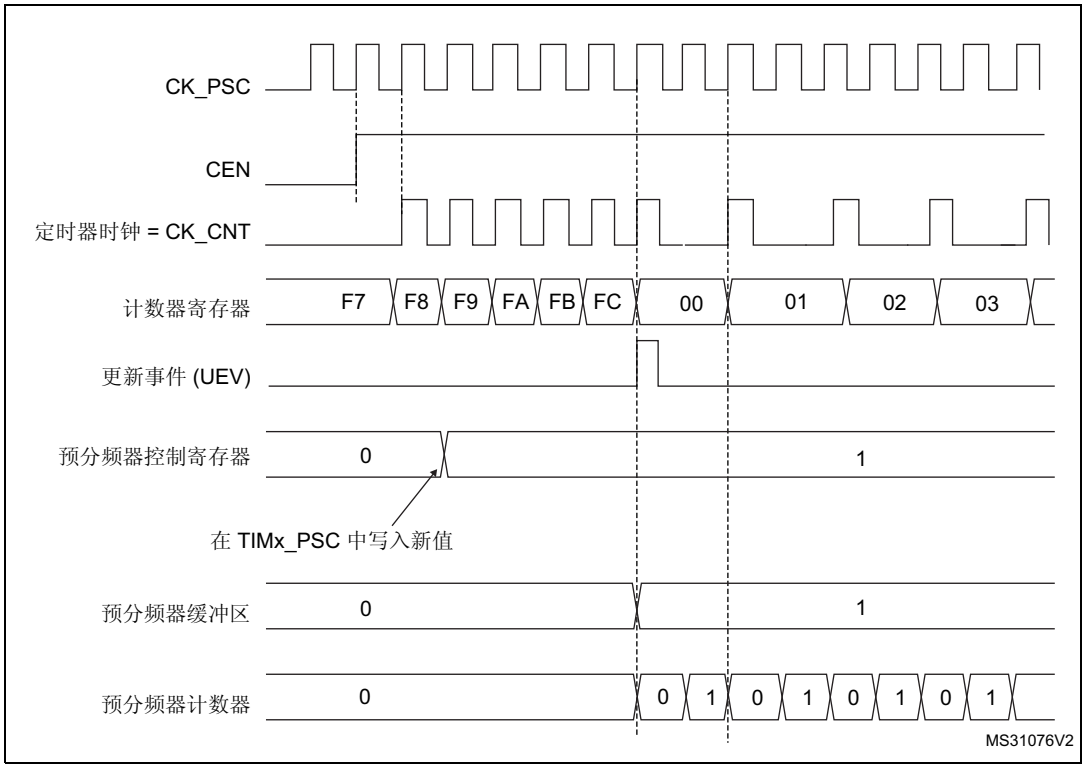
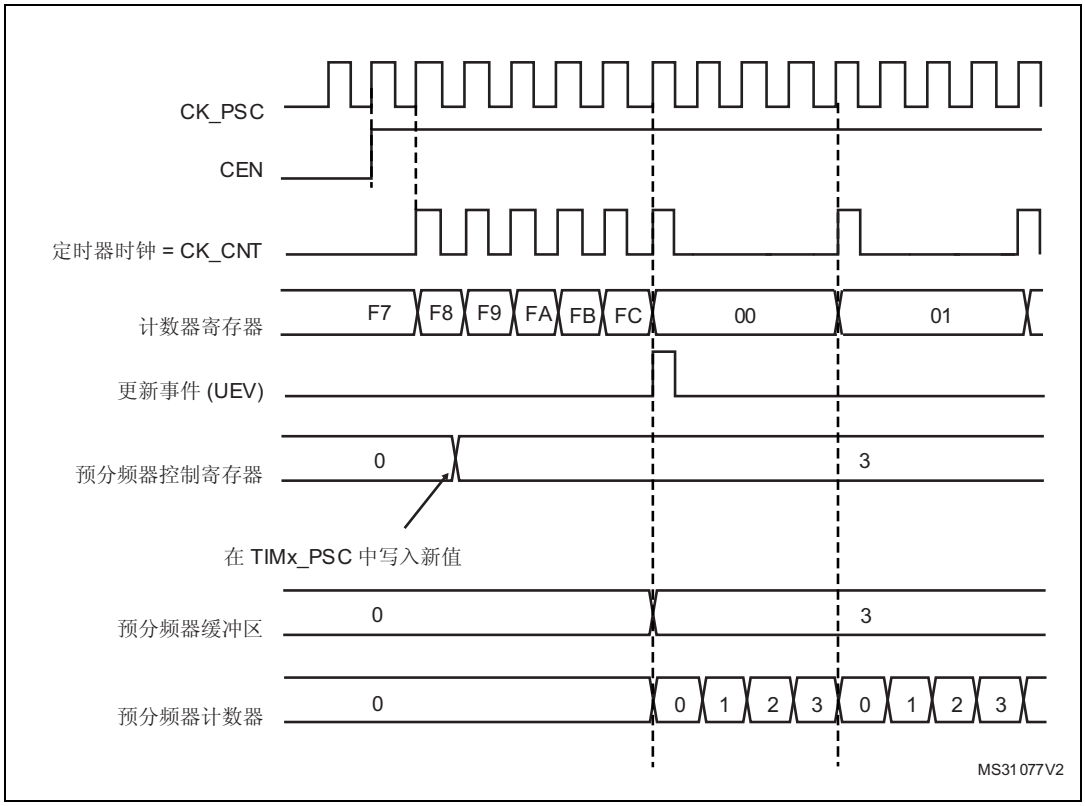


图 337. 预分频器分频由 1 变为 4 时的计数器时序图



38.3.2 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中设定的次数加一次（(TIMx_RCR) + 1）后，将生成更新事件 (UEV)。否则，将在每次计数器上溢时产生更新事件。

将 TIMx_EGR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。利用这一特性，在定时器的输入捕获操作时可以通过对 UG 位置 1 让计数器复位而不至于触发更新中断，也就避免了在捕获中断里触发更新中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容，
- 自动重载影子寄存器将以预装载值 (TIMx_ARR) 进行更新，
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 338. 计数器时序图，1 分频内部时钟

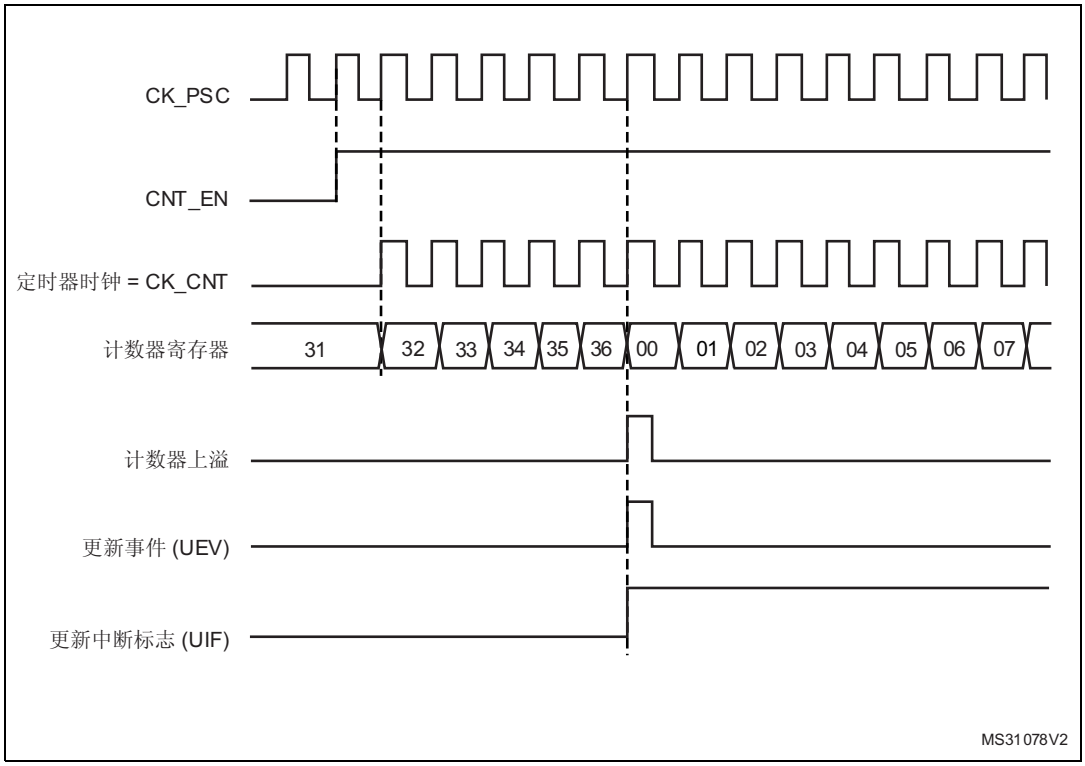


图 339. 计数器时序图，2 分频内部时钟

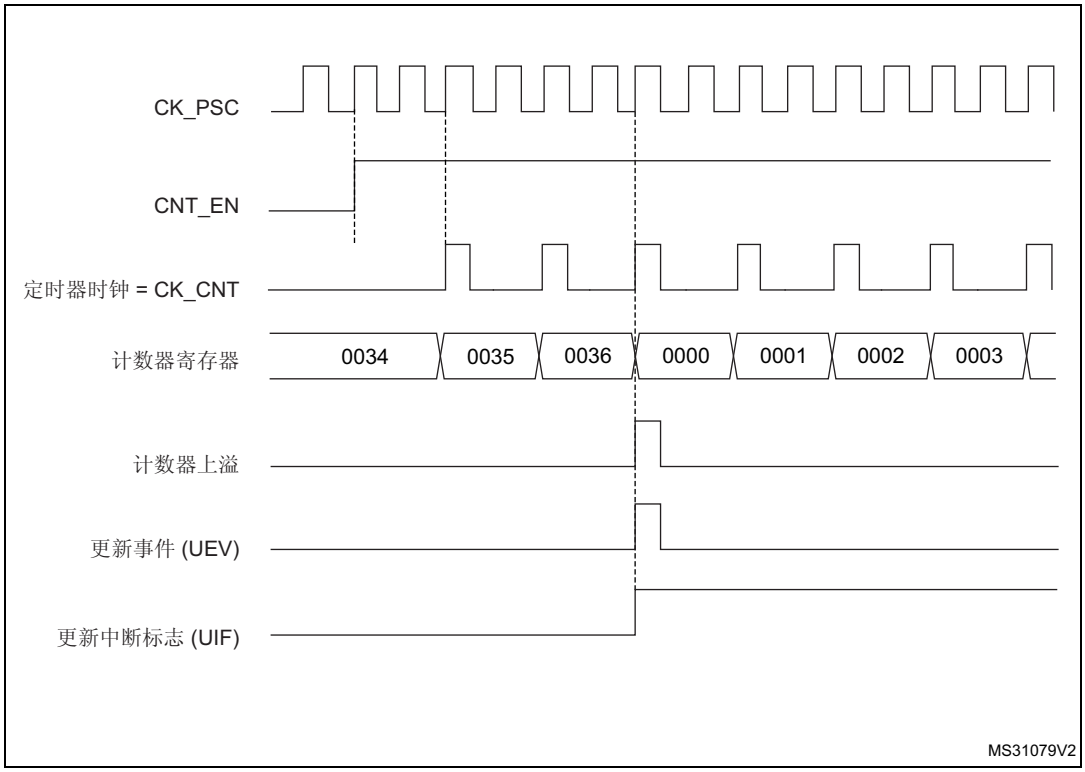


图 340. 计数器时序图，4 分频内部时钟

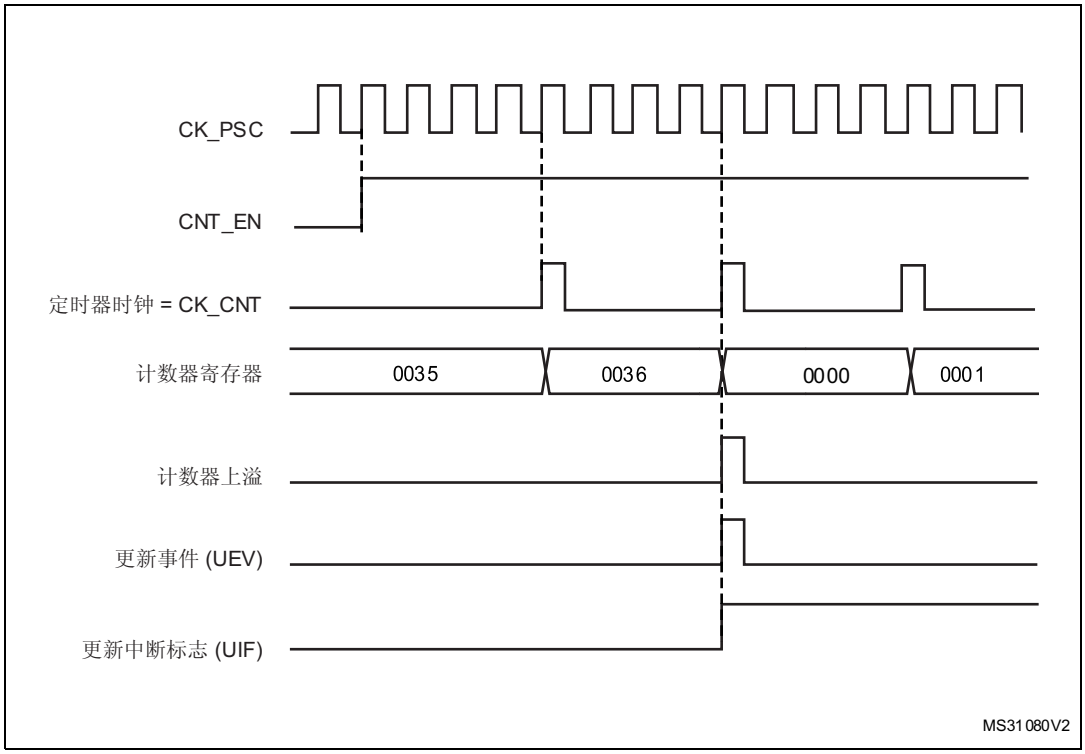


图 341. 计数器时序图，N 分频内部时钟

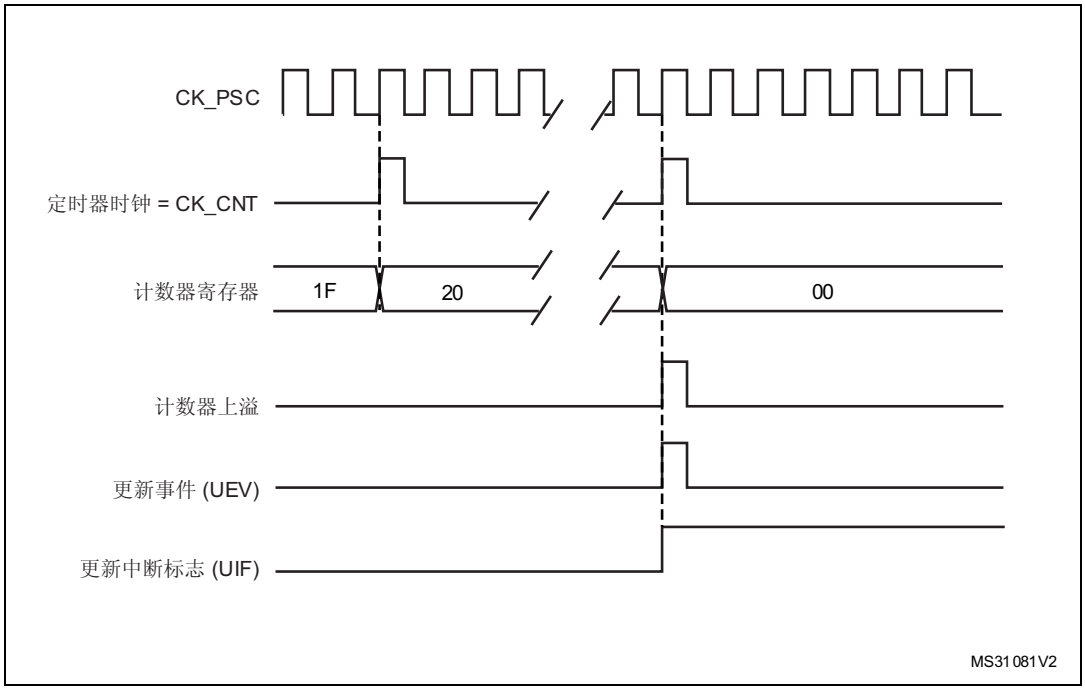


图 342. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

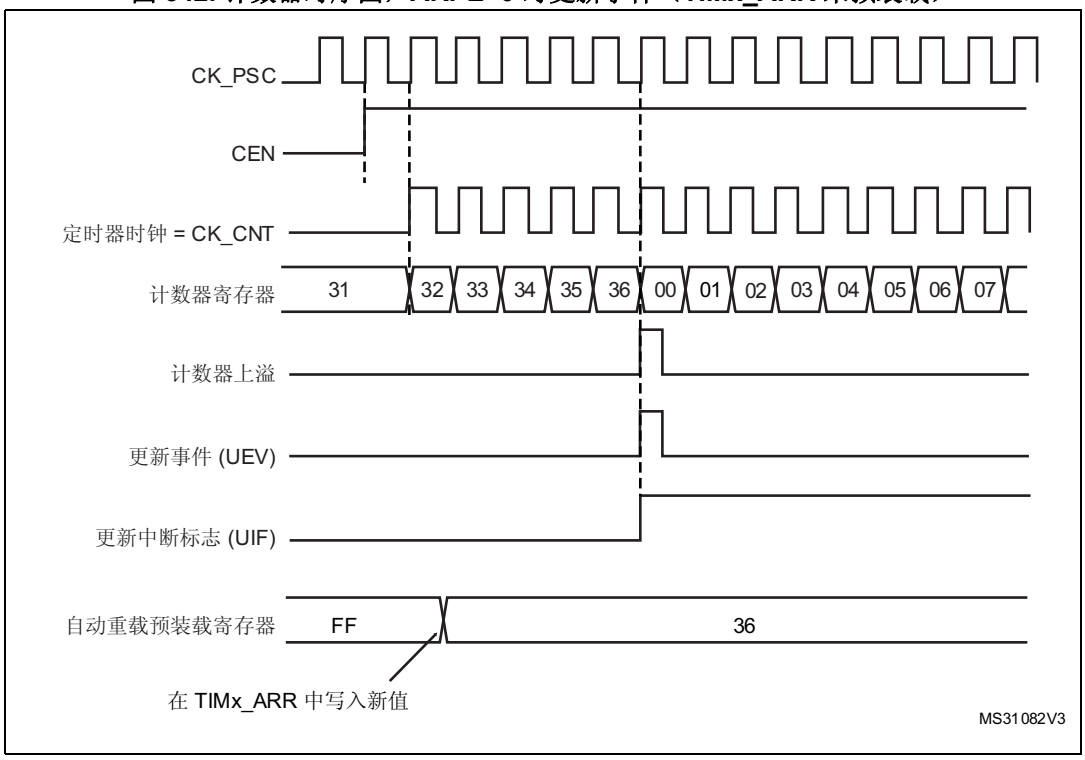
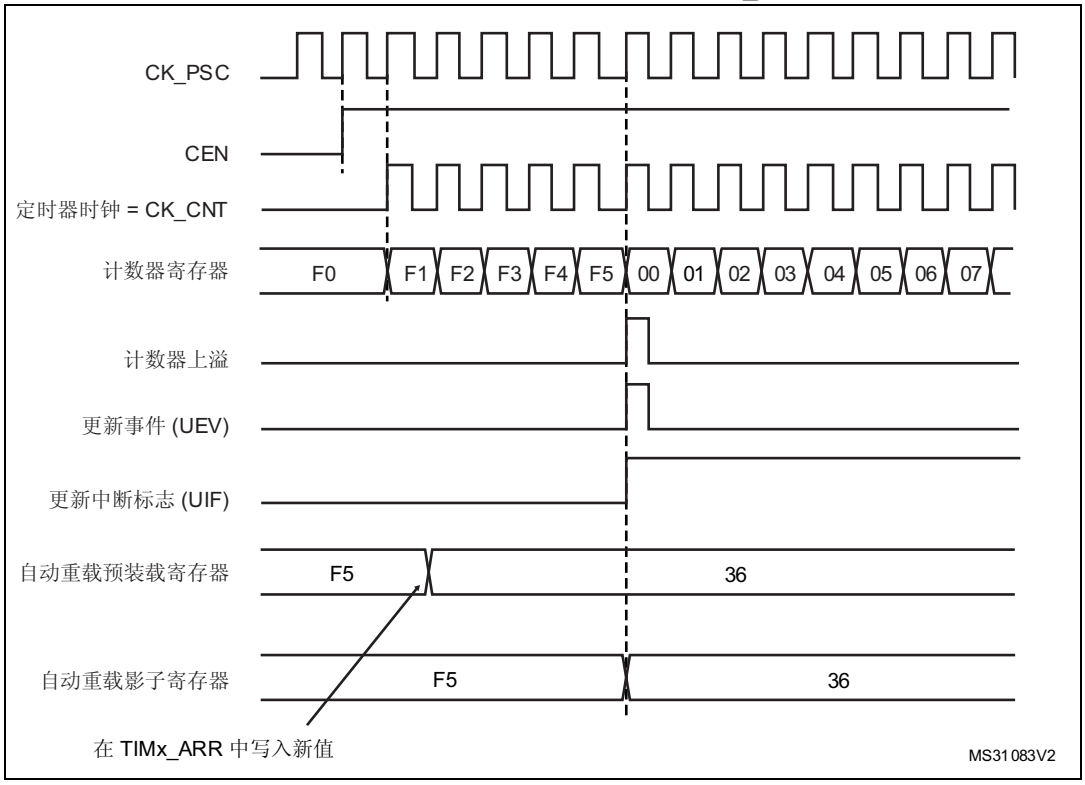


图 343. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载）



递减计数模式

在递减计数模式下，计数器从自动重载值（TIMx_ARR 寄存器的内容）开始递减计数到 0，然后重新从自动重载值开始计数并生成计数器下溢事件。

如果使用重复计数器，则当递减计数的重复次数达到重复计数器寄存器中编程的次数加一次（(TIMx_RCR) + 1）后，将产生更新事件 (UEV)。否则，将在每次计数器下溢时产生更新事件。

将 TIMx_EGR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会从当前自动重载值开始重新计数，而预分频器计数器则从 0 开始重新计数（但预分频比保持不变）。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。利用这一特性，在定时器的输入捕获操作时可以通过对 UG 位置 1 让计数器复位而不至于触发更新中断，也就避免了在捕获中断里触发更新中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容。
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载影子寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，ARR 寄存器更新在计数器重载之前被更新，因此下一个周期就是预期的值。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 344. 计数器时序图，1 分频内部时钟

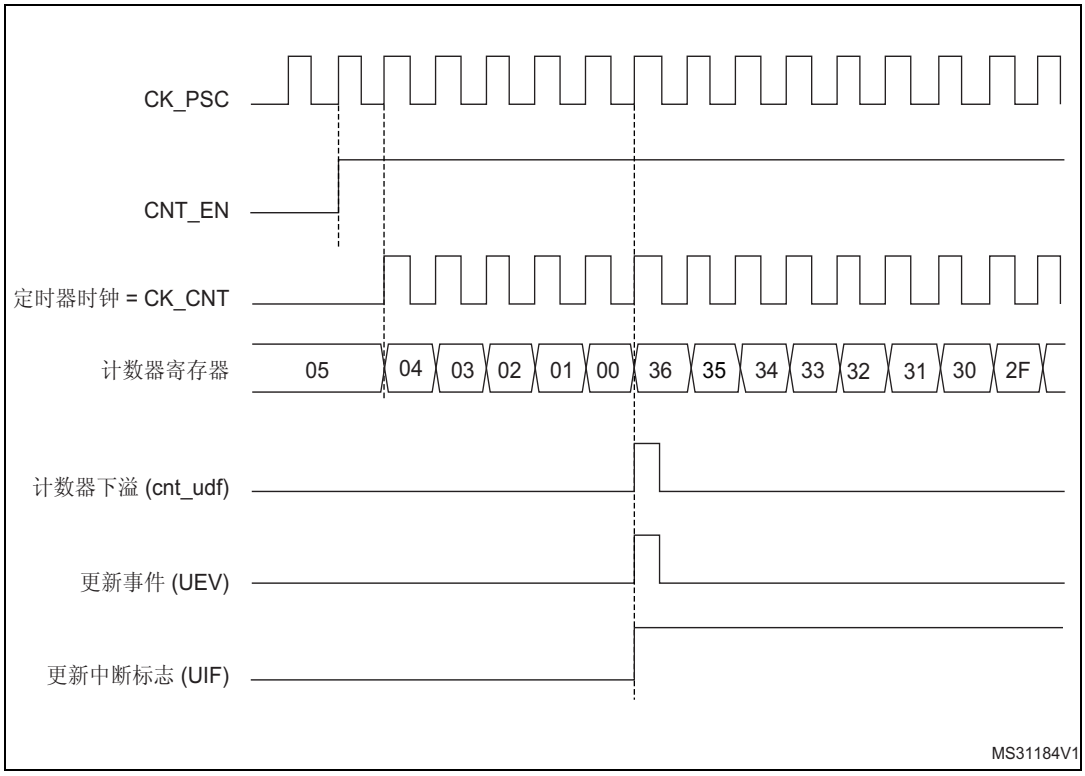


图 345. 计数器时序图，2 分频内部时钟

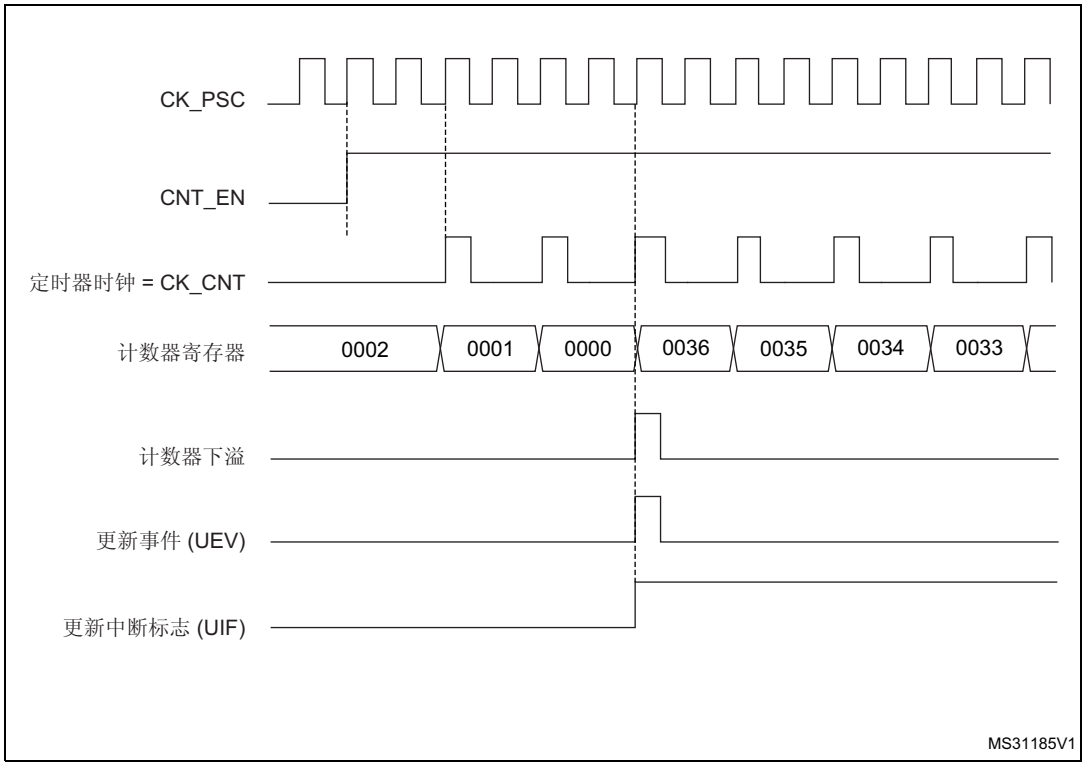


图 346. 计数器时序图，4 分频内部时钟

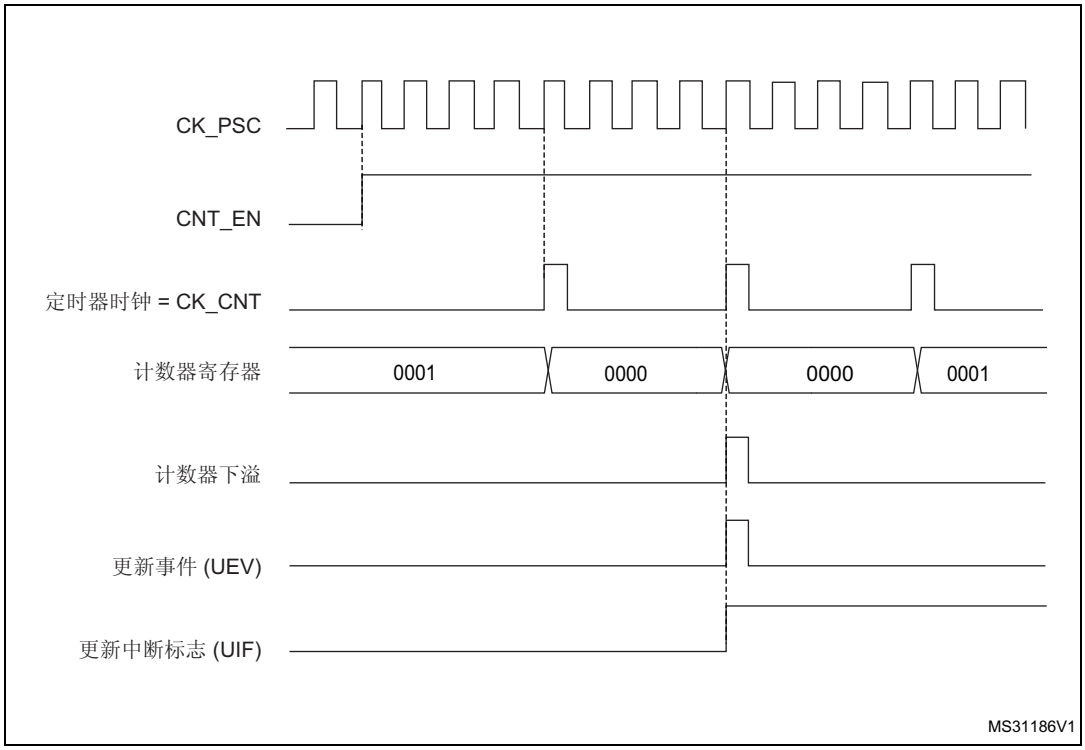


图 347. 计数器时序图，N 分频内部时钟

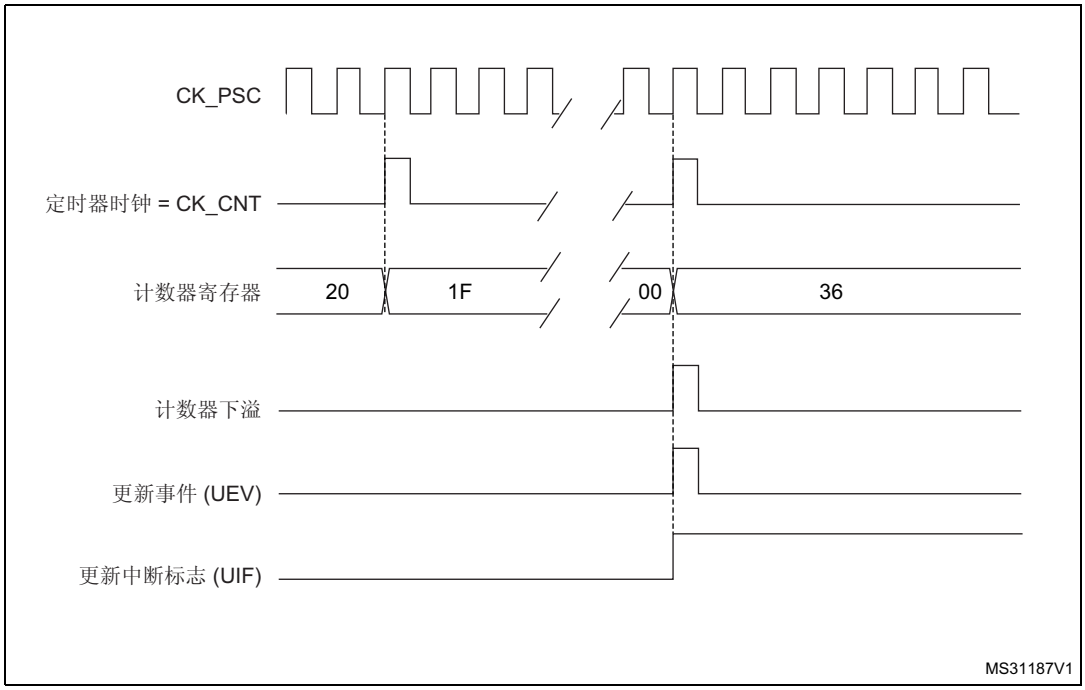
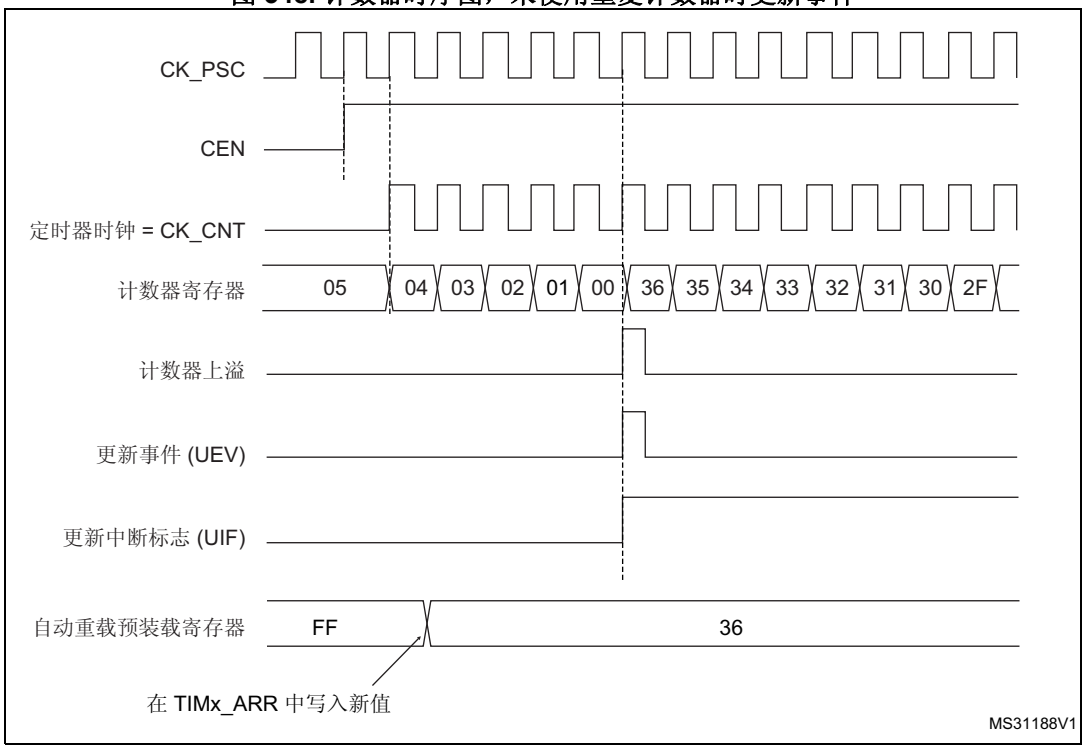


图 348. 计数器时序图，未使用重复计数器时更新事件



中心对齐模式（递增/递减计数）

在中心对齐模式下，计数器从 0 开始计数到自动重载值（TIMx_ARR 寄存器的内容）– 1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数（中心对齐模式 1，CMS = “01”）、计数器递增计数（中心对齐模式 2，CMS = “10”）以及计数器递增/递减计数（中心对齐模式 3，CMS = “11”）。

在此模式下，TIMx_CR1 寄存器的 DIR 方向位不可写入值，而是由硬件更新并指示当前计数器方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。

UEV 更新事件可通过软件禁止，只需将 TIMx_CR1 寄存器中的 UDIS 位置 1。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

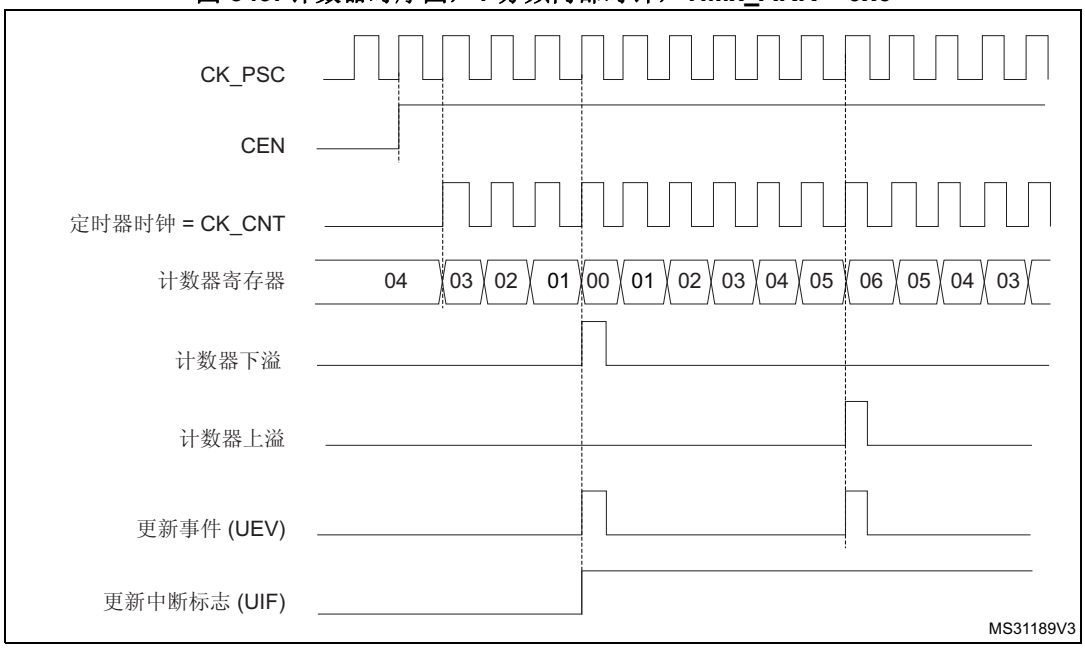
此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成 UEV 更新事件，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。利用这一特性，在定时器的输入捕获操作时可以通过对 UG 位置 1 让计数器复位而不至于触发更新中断，也就避免了在捕获中断里触发更新中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）
- 自动重载影子寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，如果更新操作是由计数器上溢触发的，则 ARR 寄存器在计数器重载之前更新，因此，下一个计数周期就是我们所希望的新的周期长度（计数器被重载新的值）。

以下各图以一些示例说明不同时钟频率下计数器的行为。

图 349. 计数器时序图，1 分频内部时钟，TIMx_ARR = 0x6



1. 此处使用中心对齐模式 1（有关详细信息，请参见*第 38.4 节：TIM1/TIM8 寄存器*）。

图 350. 计数器时序图，2 分频内部时钟

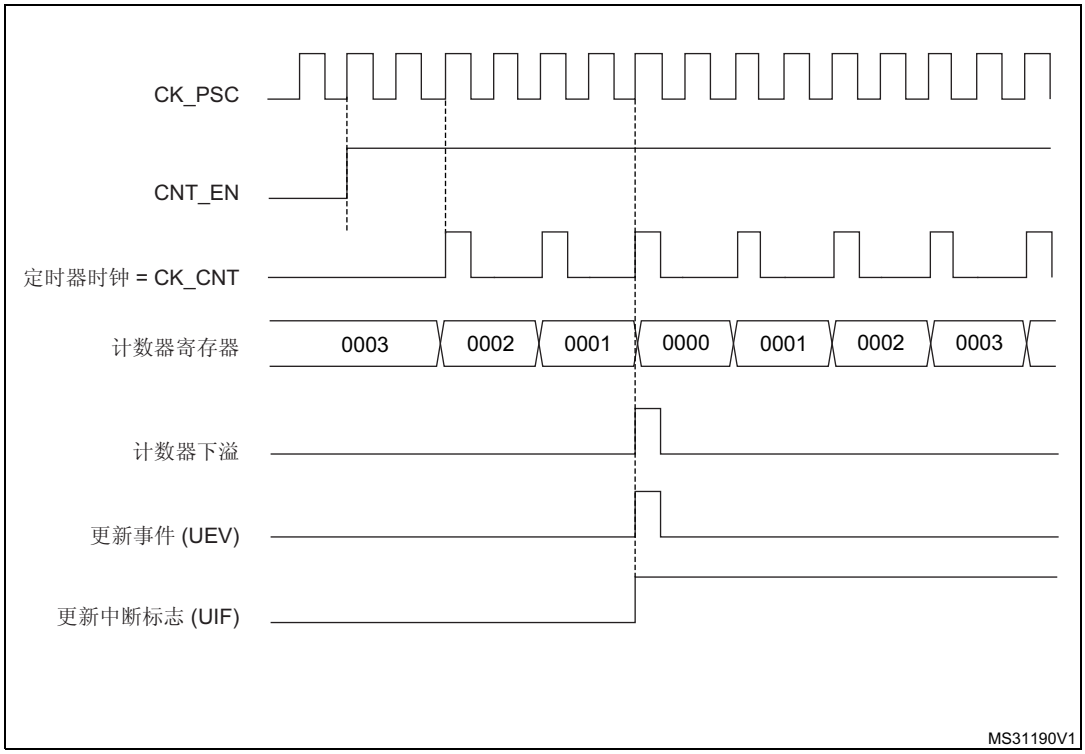


图 351. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36

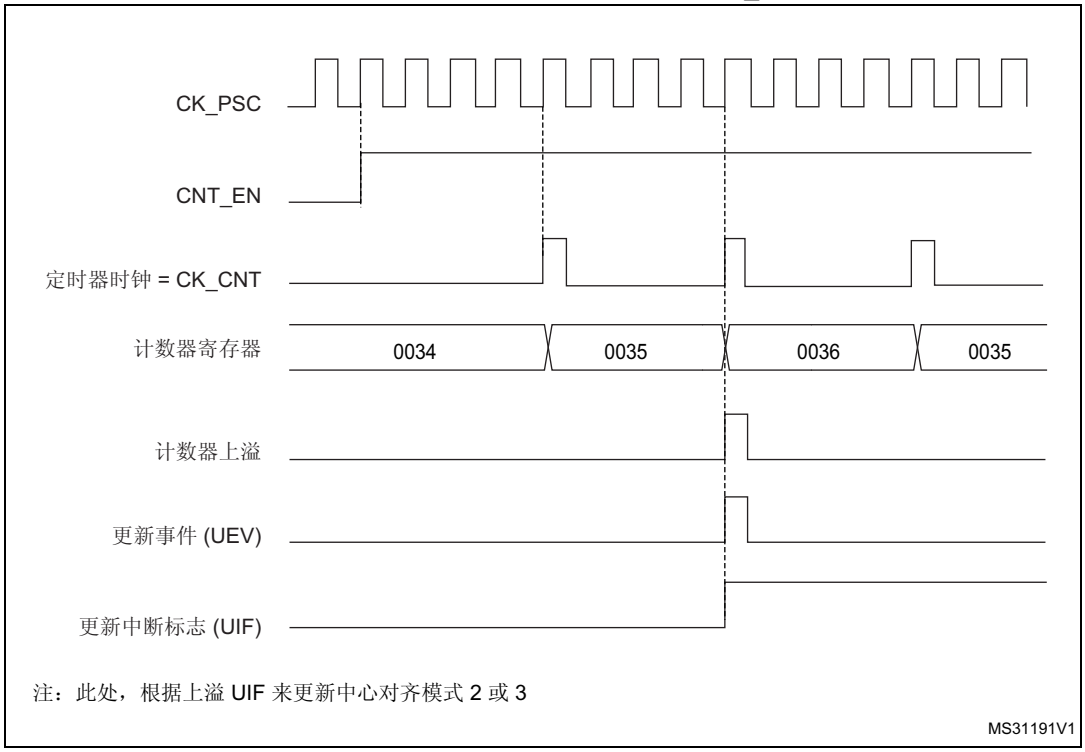


图 352. 计数器时序图，N 分频内部时钟

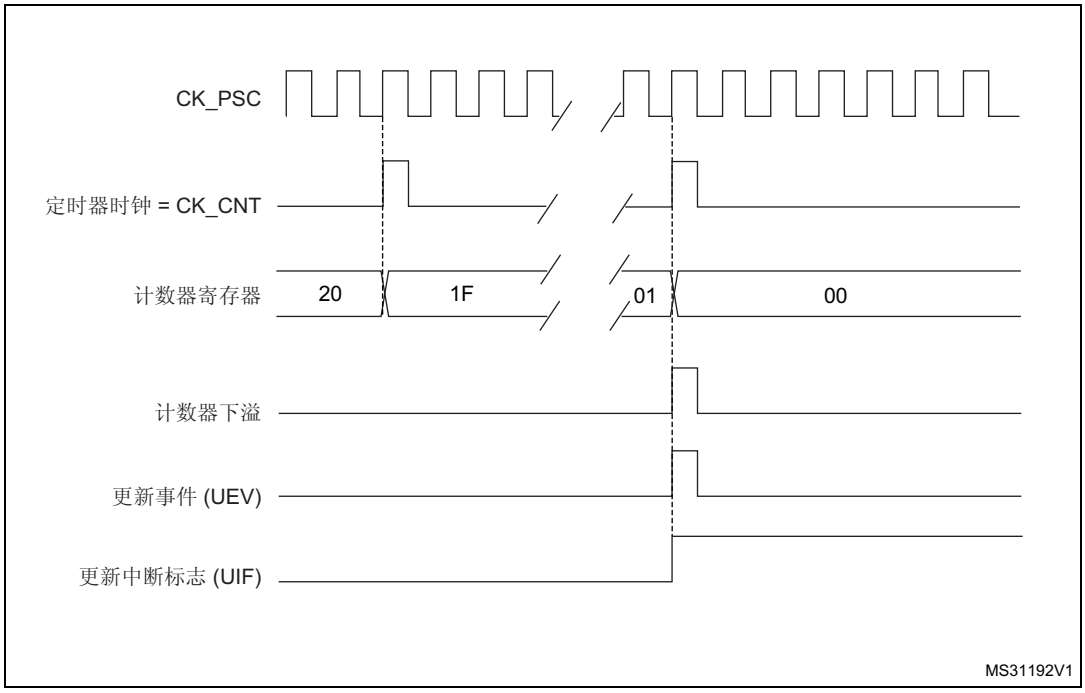


图 353. 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

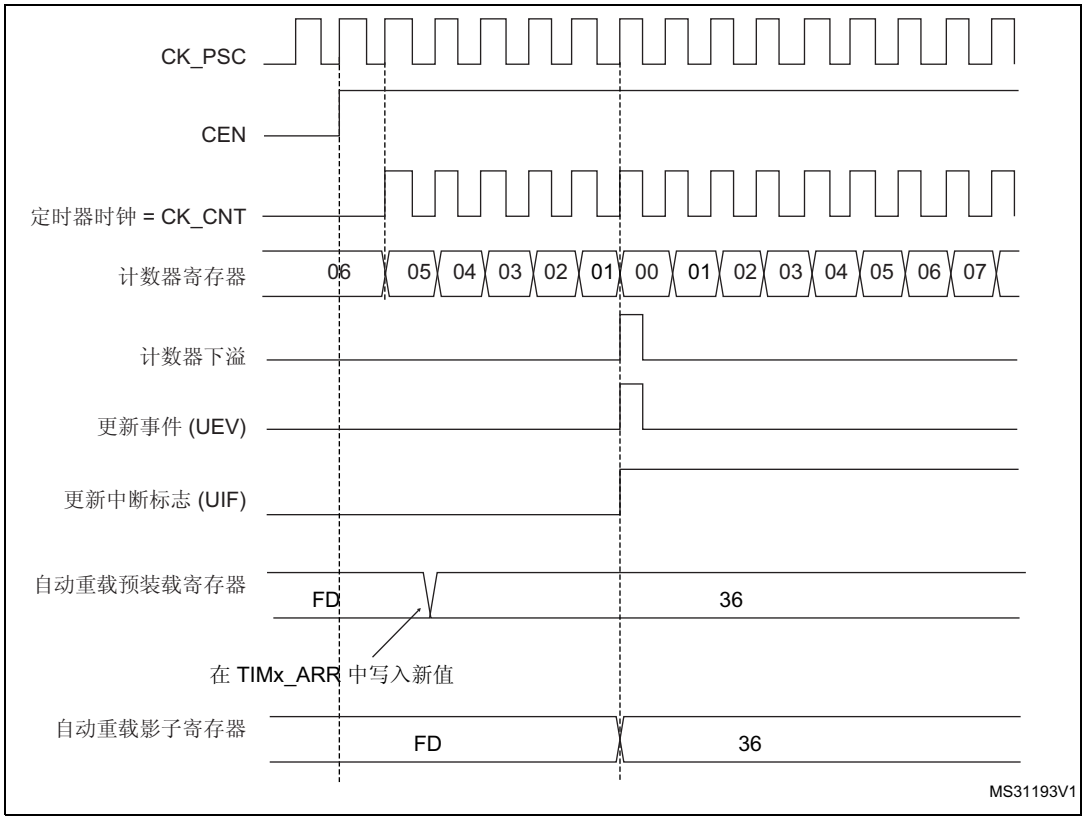
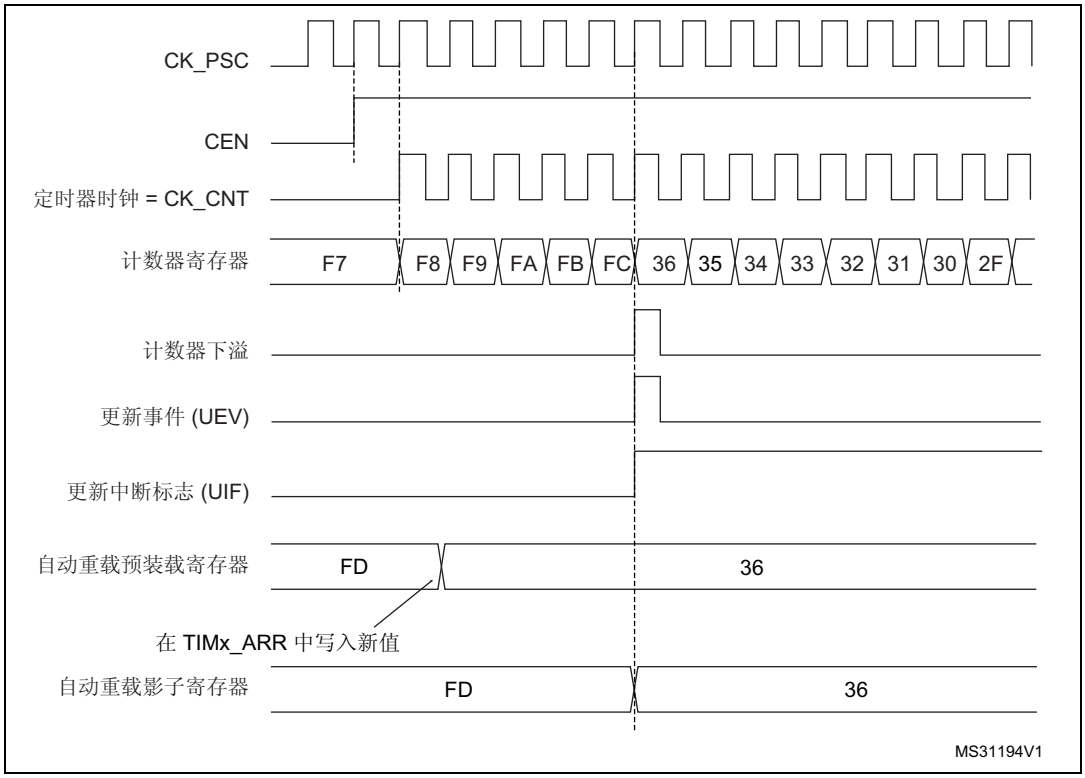


图 354. 计数器时序图，ARPE=1 时的更新事件（计数器上溢）



38.3.3 重复计数器

*第 38.3.1 节：时基单元*介绍如何因计数器上溢/下溢而生成更新事件 (UEV)。实际上，只有当重复计数器达到零时，才会生成更新事件。这在生成 PWM 信号时很有用。

这意味着，每当发生 N+1 个计数器上溢或下溢（其中，N 是 TIMx_RCR 重复计数器寄存器中的值），数据就将从预装载寄存器转移到影子寄存器（TIMx_ARR 自动重载寄存器、TIMx_PSC 预分频器寄存器以及比较模式下的 TIMx_CCRx 捕获/比较寄存器）。

重复计数器在下列情况下递减：

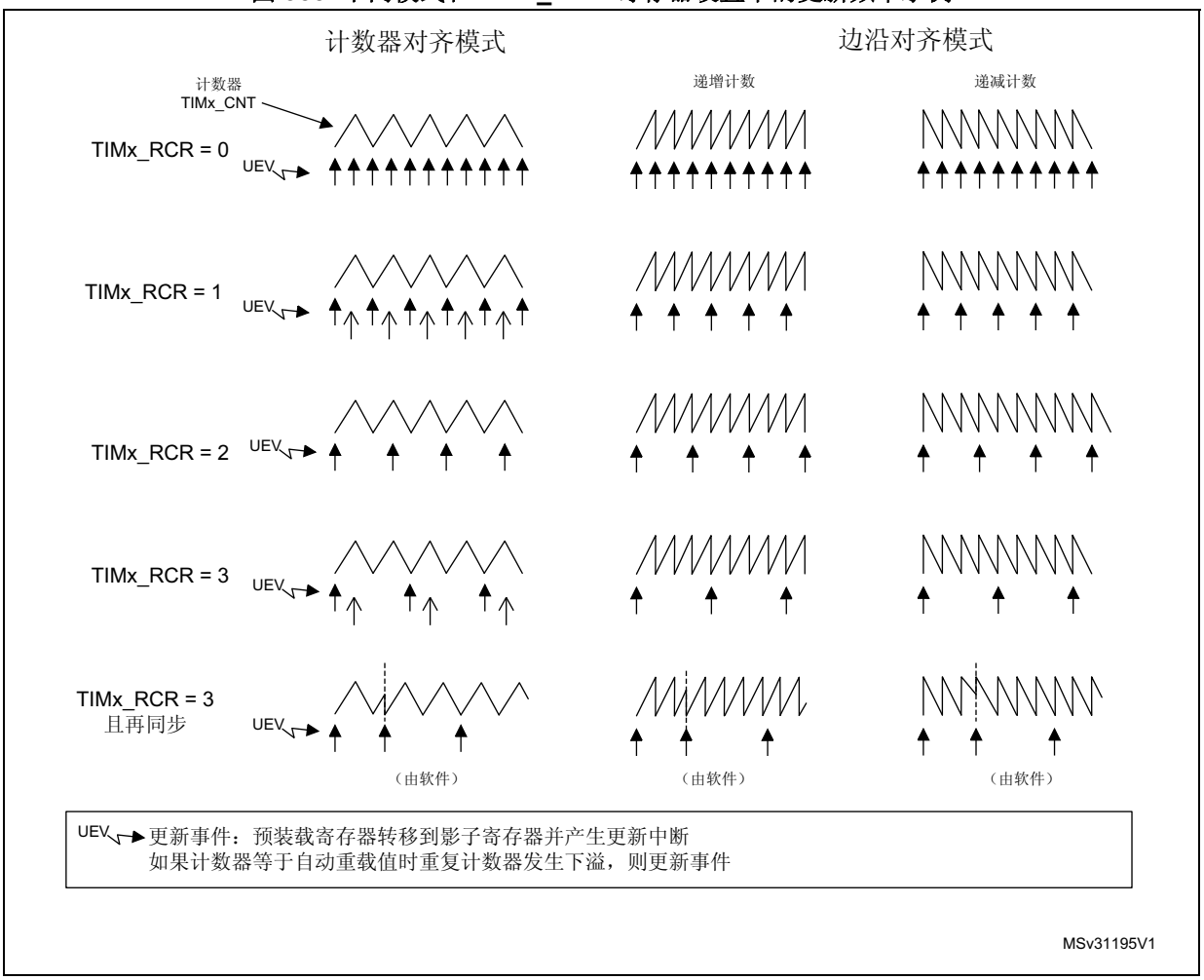
- 递增计数模式下的每个计数器上溢，
- 递减计数模式下的每个计数器下溢，
- 中心对齐模式下每个计数器上溢和计数器下溢。尽管这使得最大重复次数不超过 32768 个 PWM 周期，但在每个 PWM 周期内可更新占空比两次。当在中心对齐模式下，每个 PWM 周期仅刷新一次比较寄存器时，由于模式的对称性，最大分辨率为 $2xT_{ck}$。

重复计数器是自动重载类型；其重复率为 TIMx_RCR 寄存器所定义的值（请参见*图 355*）。当更新事件由软件（通过将 TIMx_EGR 寄存器的 UG 位置 1）或硬件（通过从模式控制器）生成时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装载 TIMx_RCR 寄存器的内容。

在中心对齐模式下，如果 RCR 值为奇数，更新事件将在上溢或下溢时发生，这取决于何时写入 RCR 寄存器以及何时启动计数器：如果在启动计数器前写入 RCR，则 UEV 在下溢时发生。如果在启动计数器后写入 RCR，则 UEV 在上溢时发生。

例如，如果 RCR = 3，UEV 将在每个周期的第四个上溢或下溢事件时产生（取决于何时写入 RCR）。

图 355. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例



38.3.4 外部触发输入

定时器具有一个外部触发输入 ETR。它的用途是：

- 外部时钟（外部时钟模式 2，请参见*第 38.3.5 节*）
- 用于从模式的触发信号（请参见*第 38.3.26 节*）
- 用于逐周期电流调节的 PWM 复位输入（请参见*第 38.3.7 节*）

下面的*图 356* 介绍了 ETR 输入的调节过程。输入极性通过 TIMx_SMCR 寄存器中的 ETP 位定义。触发信号可通过 ETPS[1:0] 位字段编程的分频比进行预分频，然后通过 ETF[3:0] 位字段设置的滤波方式进行数字滤波。

图 356. 外部触发输入模块

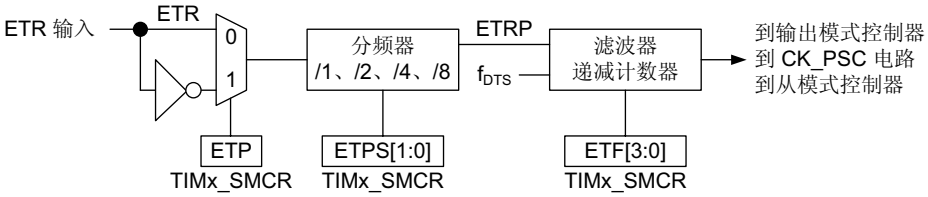
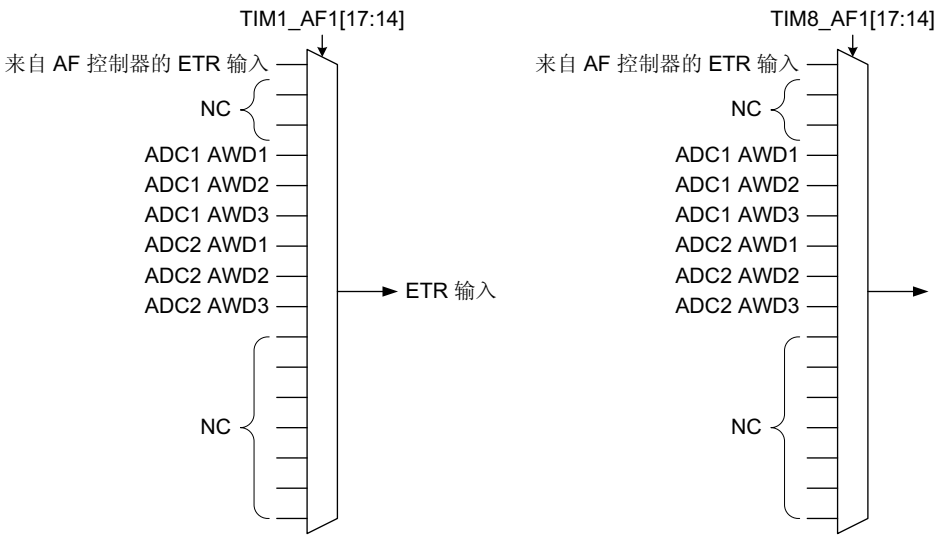


图 357. TIM1/TIM8 ETR 输入电路



38.3.5 时钟选择

计数器时钟可由下列时钟源提供：

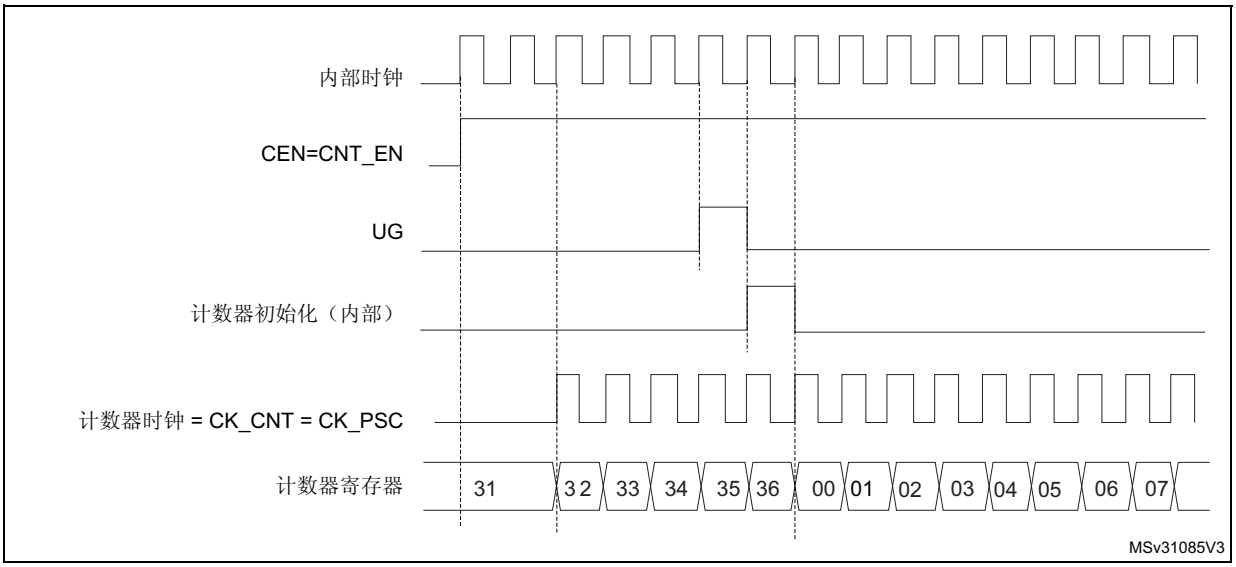
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚
- 外部时钟模式 2：外部触发输入 ETR
- 编码器模式

内部时钟源 (CK_INT)

如果禁止从模式控制器 (SMS=000)，则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍保持自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 358 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

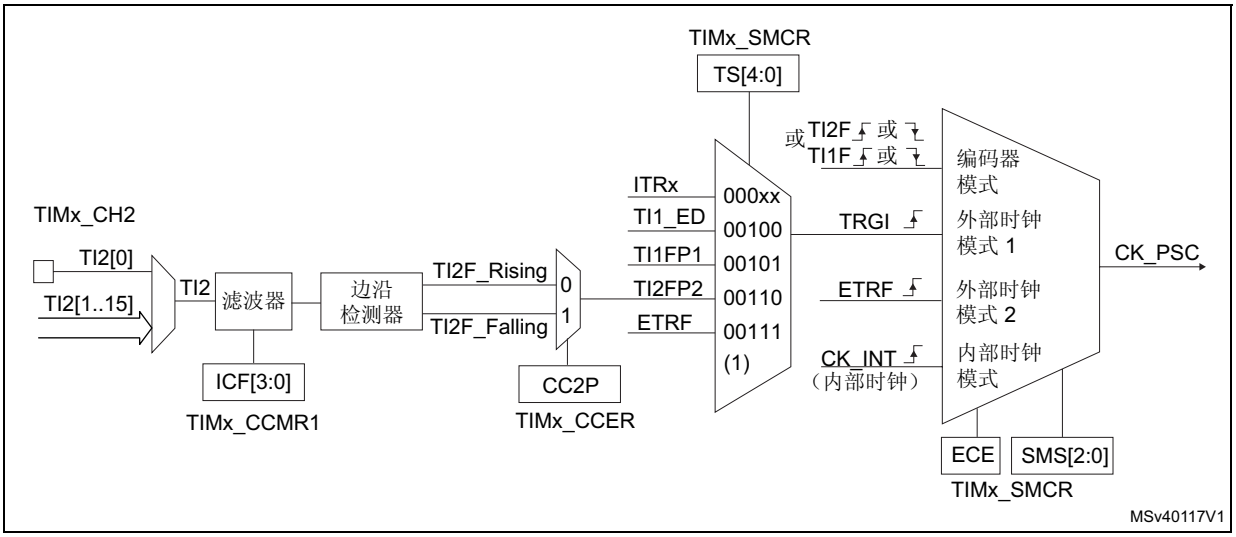
图 358. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 359. TI2 外部时钟连接示例



1. 保留从 01000 到 11111 的代码

例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

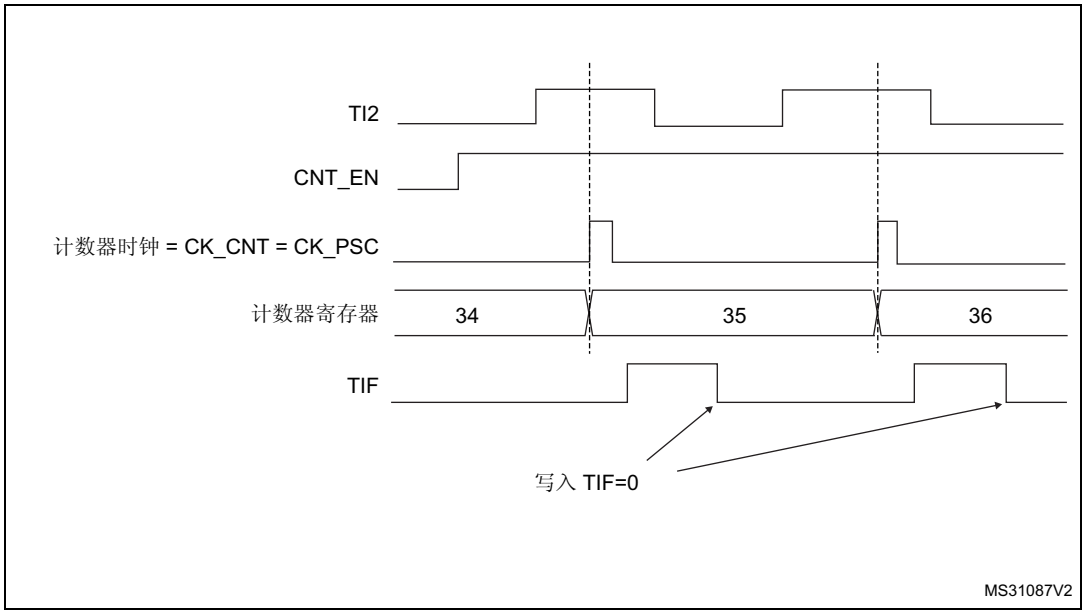
1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2x 源（内部或外部）。
2. 通过在 TIMx_CCMR1 寄存器中写入 CC2S = “01” 来配置通道 2，使其能够检测 TI2 输入的上升沿。
3. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。
4. 通过在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
5. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
6. 通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为触发输入源。
7. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

注： *由于捕获预分频器不用于触发操作，因此用户无需对其进行配置。*

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 360. 外部时钟模式 1 下的控制电路



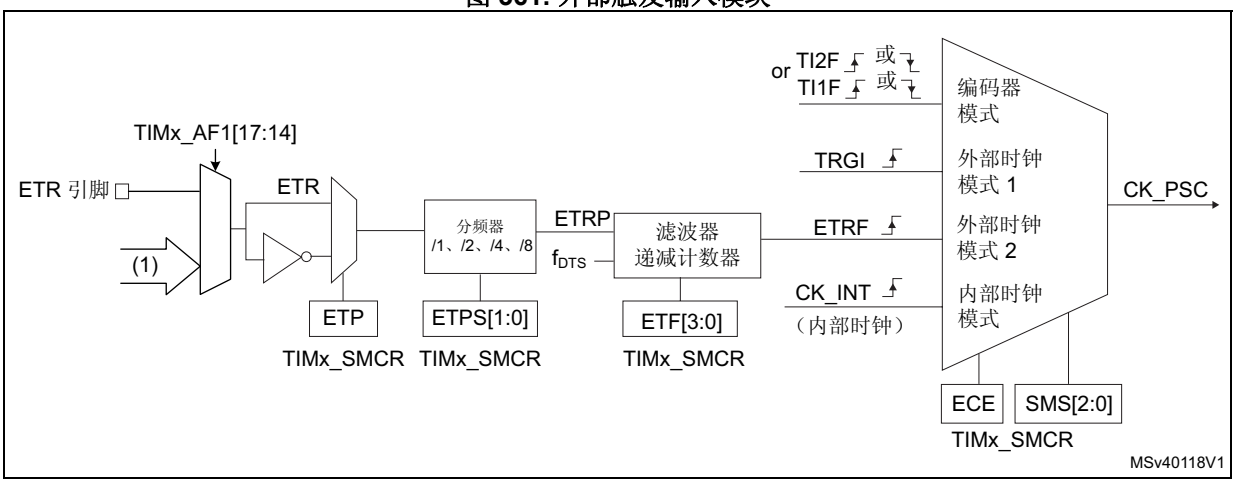
外部时钟源模式 2

通过在 TIMx_SMCR 寄存器中写入 ECE=1 可选择此模式。

计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。

图 361 简要介绍了外部触发输入模块。

图 361. 外部触发输入模块



1. 请参见*图 357：TIM1/TIM8 ETR 输入电路*。

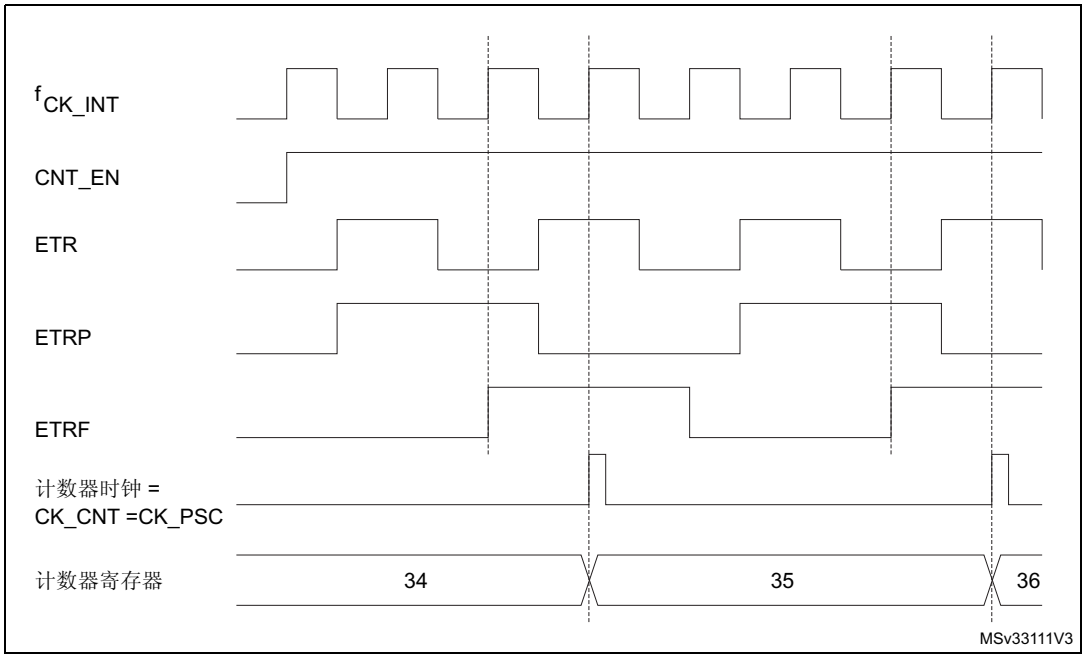
例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

1. 由于此例中不需滤波器，因此在 TIMx_SMCR 寄存器中写入 ETF[3:0]=0000。
2. 通过在 TIMx_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器
3. 通过在 TIMx_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测
4. 通过在 TIMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
5. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。

ETR 的上升沿与实际计数器时钟之间的延迟是由于 ETRP 信号的重新同步电路引起的。因此，计数器能够正确捕获的最大频率不超过 TIMxCLK 频率的 ¼。当 ETRP 信号较快时，用户应通过适当的 ETPS 预分频器设置对外部信号进行分频。

图 362. 外部时钟模式 2 下的控制电路



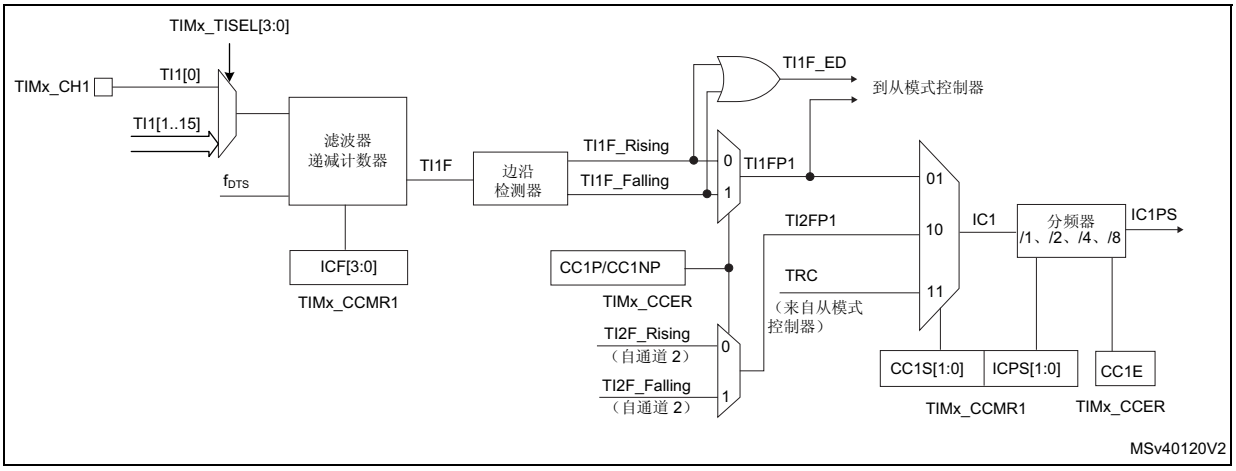
38.3.6 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入单元（数字滤波、多路复用和预分频器，通道 5 和通道 6 除外）和一个输出单元（比较器和输出控制）构建而成。

图 363 到*图 366* 简要介绍了一个捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 363. 捕获/比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 364. 捕获/比较通道 1 主电路

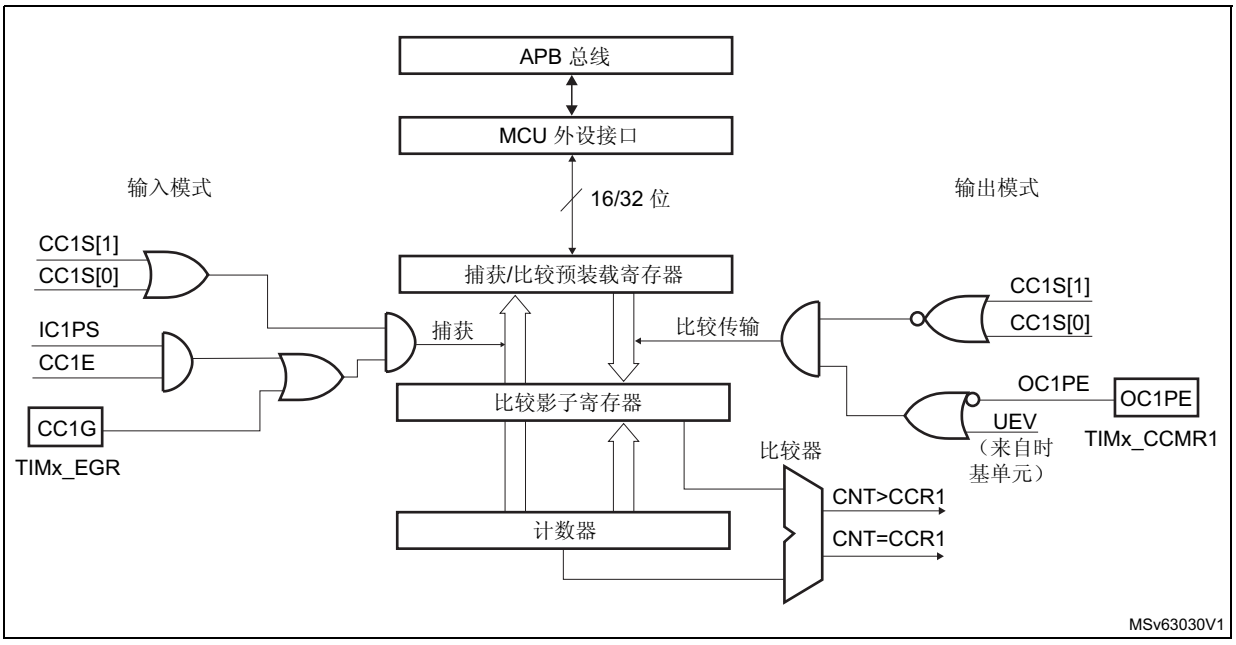
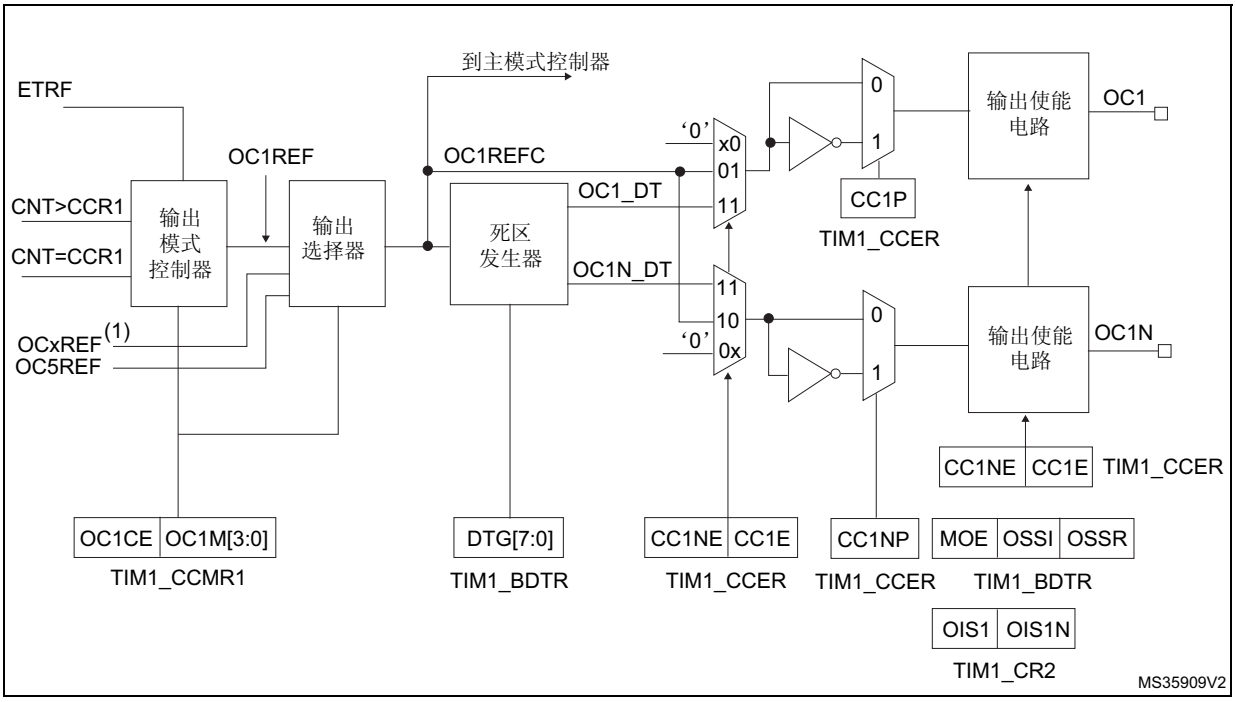


图 365. 捕获/比较通道的输出阶段（通道 1、通道 2 和通道 3）



1. OCxREF，其中 x 为互补通道的序号

图 366. 捕获/比较通道的输出阶段（通道 4）

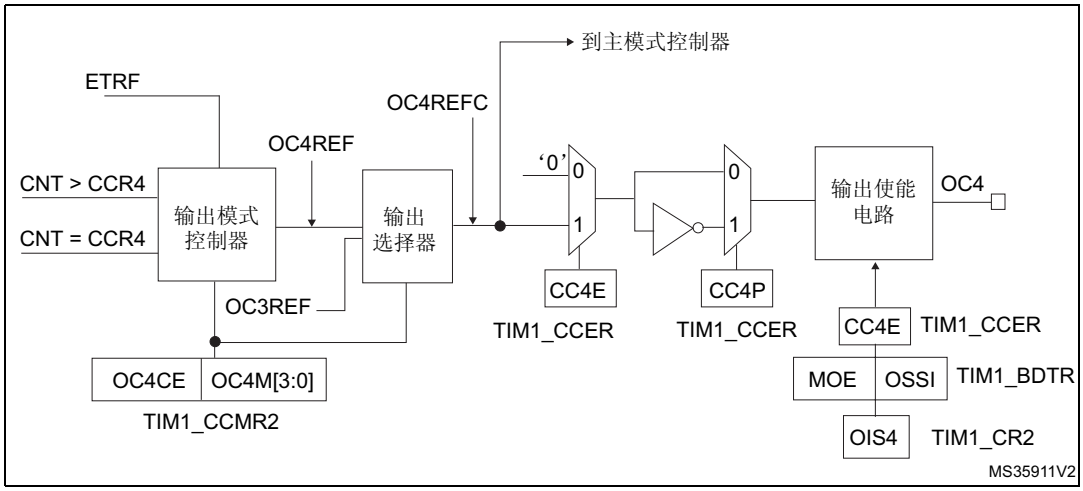
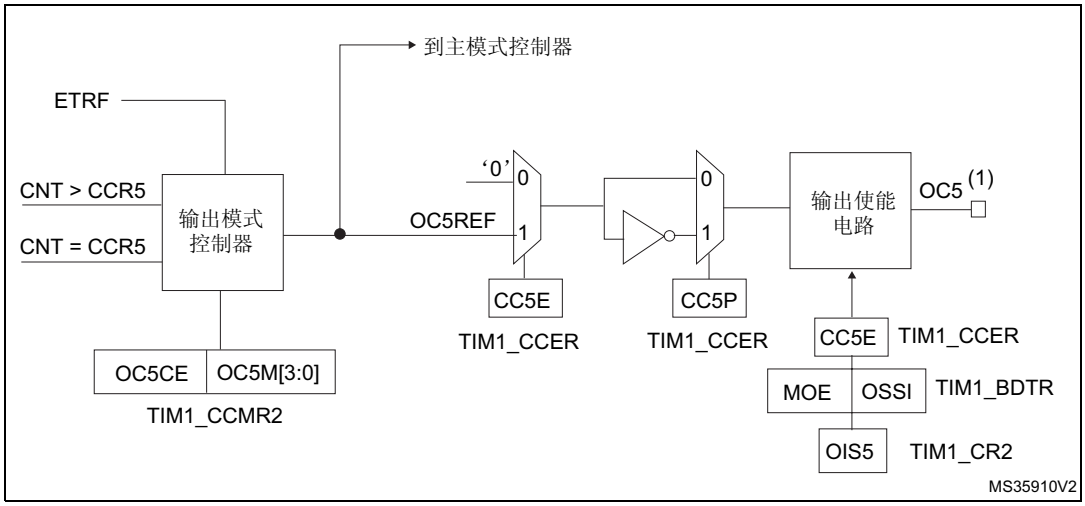


图 367. 捕获/比较通道的输出阶段（通道 5 和通道 6）



1. 不适用于外部。

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。读写访问的始终是预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

38.3.7 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将过捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件方法向 CCxIF 写入 0 来给 CCxIF 清零，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择有效输入：TIMx_CCR1 必须关联到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
3. 根据关联到定时器的信号，对相关输入滤波参数进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设输入信号发生翻转时，信号需要 5 个内部时钟周期才能稳定，则必须设置滤波时间大于 5 个内部时钟周期。若在 TI1 上连续 8 次检测到新电平采样值（以 f_{DTS} 频率采样）判定沿跳变合法，则向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
4. 通过在 TIMx_CCER 寄存器中将 CC1P 位和 CC1NP 位写入 0，选择 TI1 上的有效转换边沿（本例中为上升沿）。
5. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。
6. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。

7. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理过捕获，建议在读取捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的过捕获信息。

注： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。*

38.3.8 PWM 输入模式

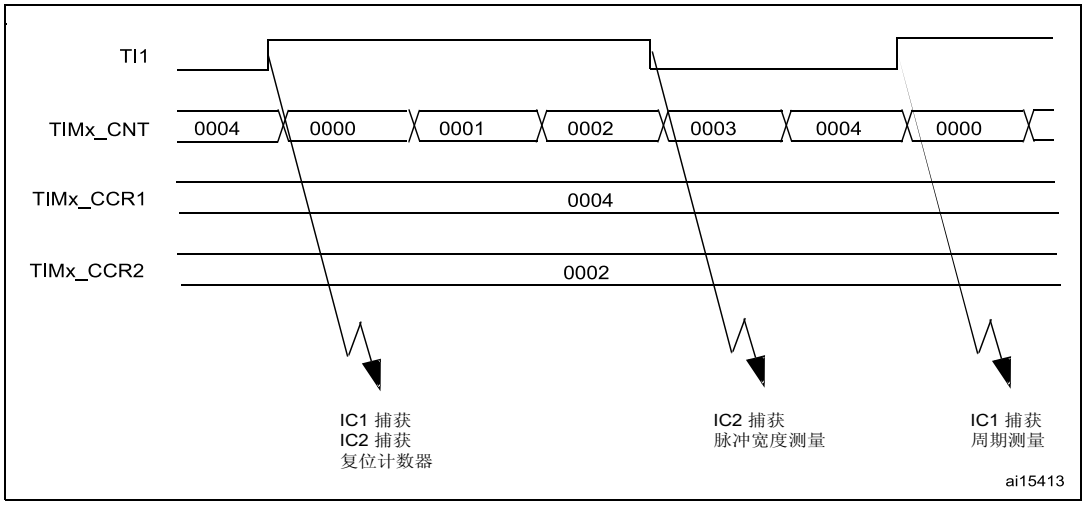
此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，用户可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
3. 选择 TI1FP1 的有效极性（用于在 TIMx_CCR1 中捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
4. 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10（选择 TI1）。
5. 选择 TI1FP2 的有效极性（用于在 TIMx_CCR2 中捕获）：向 CC2P 位和 CC2NP 位写入 CC2P/CC2NP=“10”（下降沿有效）。
6. 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入 00101（选择 TI1FP1）。
7. 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入 0100。
8. 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 368. PWM 输入模式时序



38.3.9 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（OCxREF 和 OCx/OCxN）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号 (OCxREF/OCx) 强制设置为有效电平，用户只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入 0101。OCxREF 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如：CCxP=0（OCx 高电平有效）=> OCx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 0100，可将 OCxREF 信号强制设置为低电平。

当然，即使强制输出模式下，TIMx_CCRx 影子寄存器与计数器之间的比较仍然会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。下面的输出比较模式一节对此进行了介绍。

38.3.10 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。通道 1 到通道 4 可用作输出，而通道 5 和通道 6 只能在设备内部使用（例如，用于产生混合波形或触发 ADC）。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平 (OCxM=0000)，也可设置为有效电平 (OCxM=0001)、无效电平 (OCxM=0010) 或进行翻转 (OCxM=0011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应使能位（TIMx_DIER 寄存器的 CCxDE 位，TIMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

使用 TIMx_CCMRx 寄存器中的 OCxPE 位，可将 TIMx_CCRx 寄存器配置为带或不带预装功能。

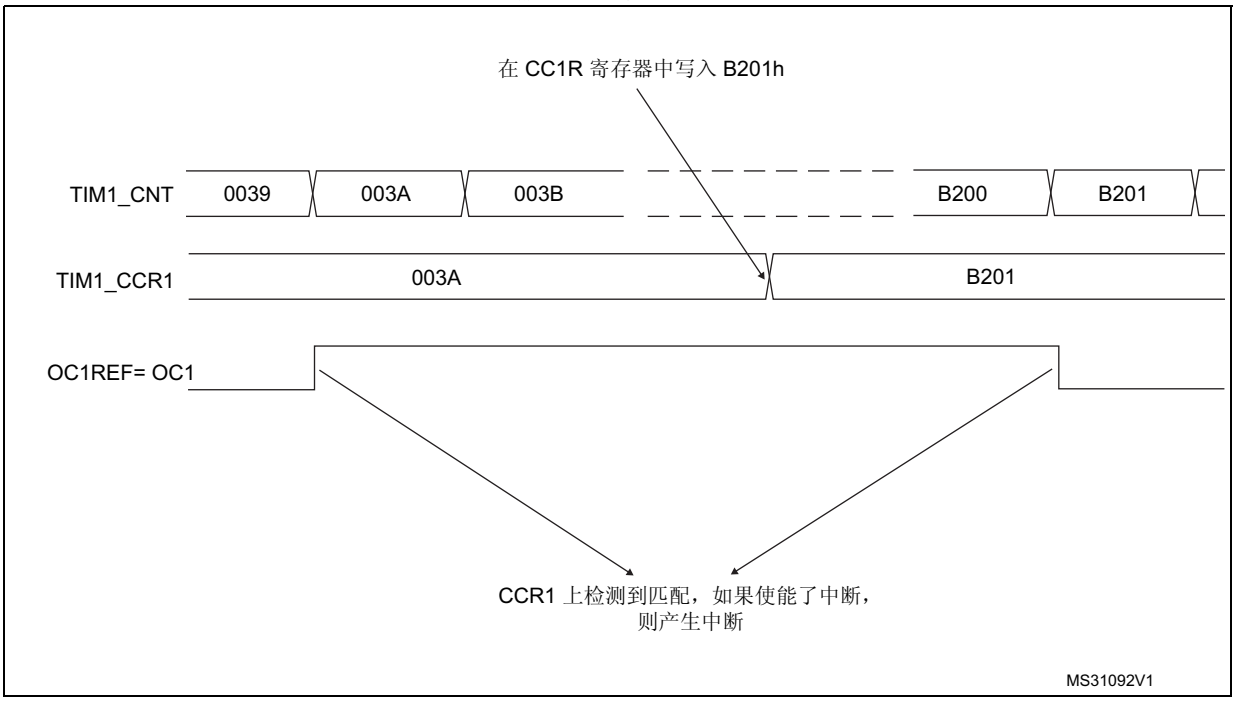
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。时间分辨率可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，则需将 CCxIE 位置 1。
4. 选择输出模式。例如：
   - 当 CNT 与 CCRx 匹配时，写入 OCxM = 0011 以翻转 OCx 输出引脚
   - 写入 OCxPE = 0 以禁止预装功能
   - 写入 CCxP = 0 以选择高电平有效极性
   - 写入 CCxE = 1 以使能输出
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装功能（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIMx_CCRx 影子寄存器）。*图 369* 给出了一个示例。

图 369. 输出比较模式，翻转 OC1



38.3.11 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIMx_ARR 寄存器值决定，其占空比则由 TIMx_CCRx 寄存器值决定。

各通道可以独立选择 PWM 模式（每个 OCx 输出对应一个 PWM），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“0110”（PWM 模式 1）或“0111”（PWM 模式 2）。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

由于只有在发生更新事件时预装载寄存器内容才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。通过 CCxE、CCxNE、MOE、OSSI 和 OSSR 位（TIMx_CCER 和 TIMx_BDTR 寄存器）的组合使能 OCx 输出。有关详细信息，请参见 TIMx_CCER 寄存器说明。

在 PWM 模式（1 或 2）下，TIMx_CNT 总是与 TIMx_CCRx 进行比较，以确定是 TIMx_CCRx ≤ TIMx_CNT 还是 TIMx_CNT ≤ TIMx_CCRx（取决于计数器计数方向）。

根据 TIMx_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

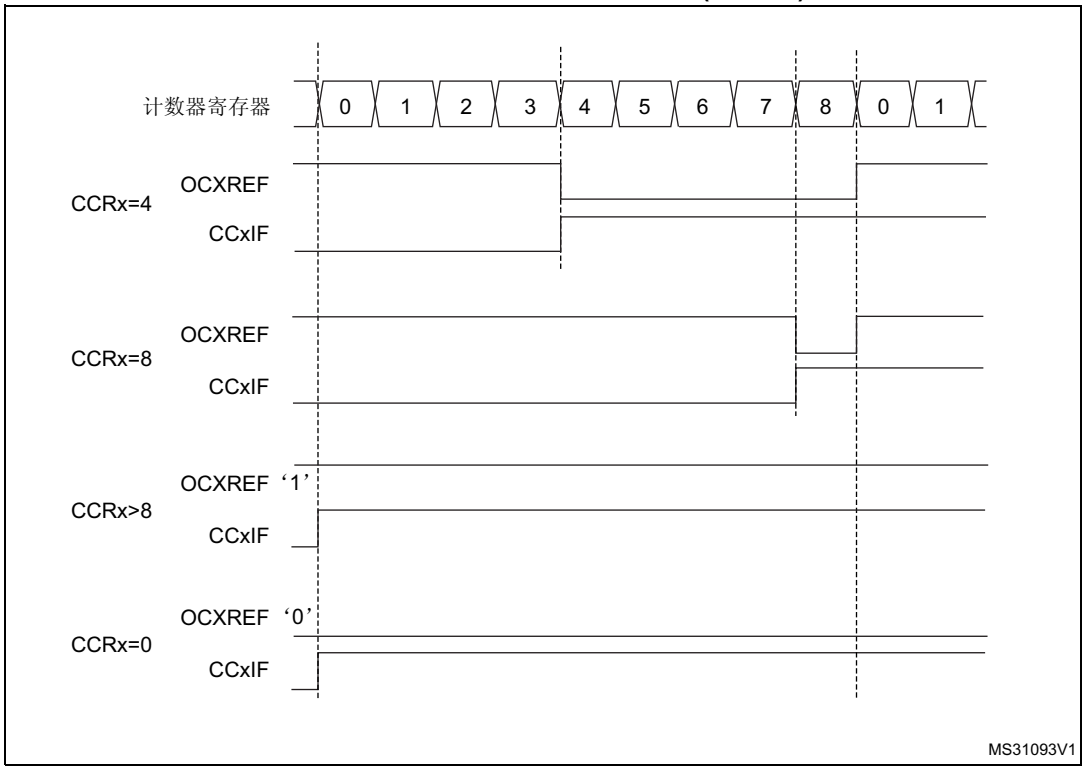
PWM 边沿对齐模式

- 递增计数配置

  当 TIMx_CR1 寄存器中的 DIR 位为低时执行递增计数。请参见*第 1895 页的递增计数模式*。

  以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxRef 保持为“0”。*图 370* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 370. 边沿对齐模式的 PWM 波形 (ARR=8)



- 递减计数配置

  当 TIMx_CR1 寄存器中的 DIR 位为高时执行递减计数。请参见*第 1898 页的递减计数模式*。

  在 PWM 模式 1 下，只要 TIMx_CNT > TIMx_CCRx，参考信号 OCxRef 即为低电平，否则其为高电平。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重载值，则 OCxREF 保持为“1”。此模式下不可能产生 0% 的 PWM 波形。

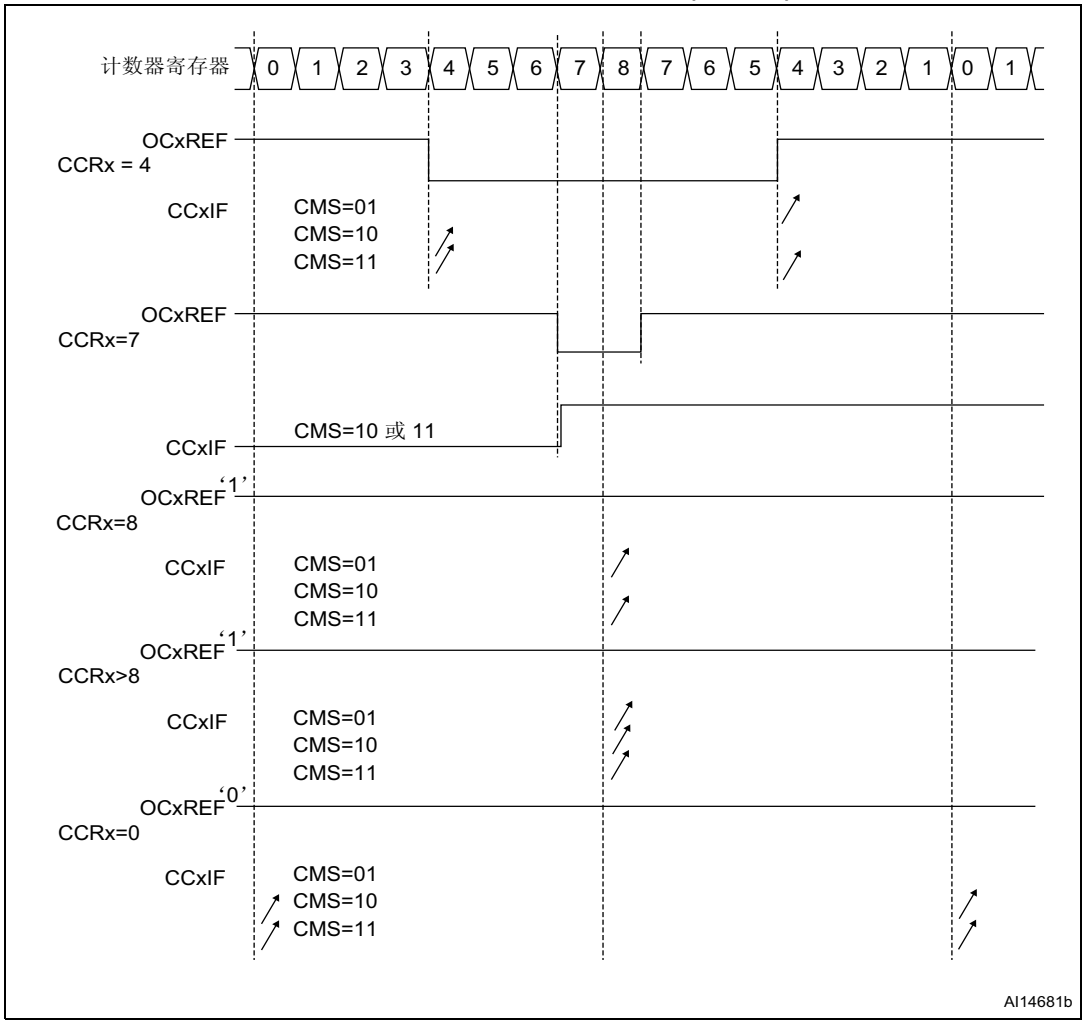
PWM 中心对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为“00”（其余所有配置对 OCxRef/OCx 信号具有相同的作用），中心对齐模式生效。根据 CMS 位的配置，可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIMx_CR1 寄存器中的方向位 (DIR) 由硬件更新，不得通过软件更改。请参见*第 1901 页的中心对齐模式（递增/递减计数）*。

图 371 显示了中心对齐模式的 PWM 波形，在此例中：

- TIMx_ARR=8，
- PWM 模式为 PWM 模式 1，
- 在根据 TIMx_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1。

图 371. 中心对齐模式 PWM 波形 (ARR=8)



中心对齐模式使用建议

- 启动中心对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 TIMx_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得通过软件同时修改 DIR 和 CMS 位。
- 不建议在中心对齐模式下对计数器执行写操作，否则将发生意外错误。尤其是：
  - 如果写入计数器的值大于自动重载值 (TIMx_CNT>TIMx_ARR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
  - 如果向计数器写入 0 或 TIMx_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIMx_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

38.3.12 不对称 PWM 模式

在不对称模式下，生成的两个中心对齐 PWM 信号间允许存在可编程相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和相移则由一对 TIMx_CCRx 寄存器确定。两个寄存器分别控制递增计数和递减计数期间的 PWM，这样每半个 PWM 周期便会调节一次 PWM：

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

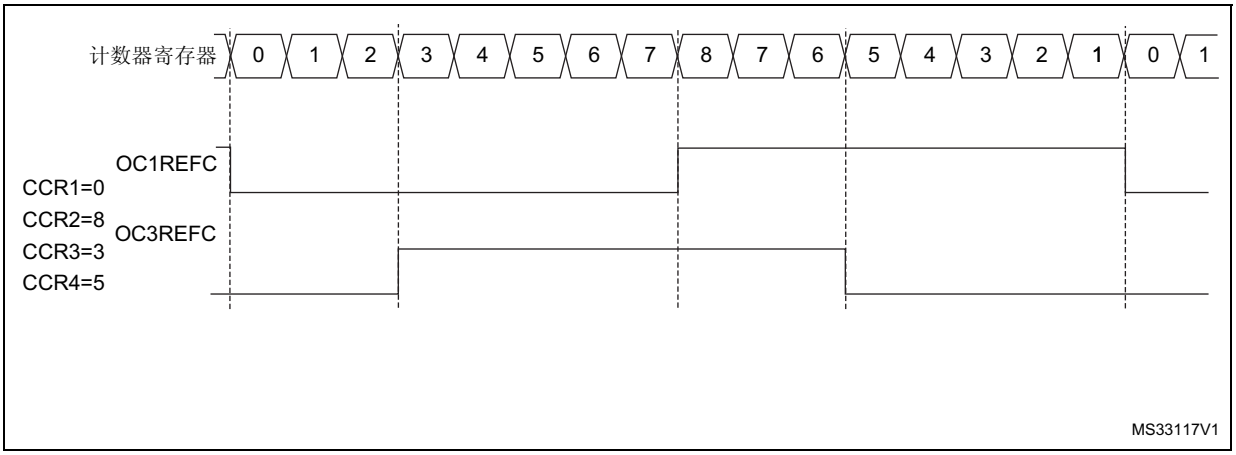
两个通道可以独立选择不对称 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1110”（不对称 PWM 模式 1）或“1111”（不对称 PWM 模式 2）。

注： *出于兼容性原因，OCxM[3:0] 位字段分为两部分，最高有效位与最低有效的 3 位不相邻。*

给定通道用作不对称 PWM 通道时，也可使用其互补通道。例如，如果通道 1 上产生 OC1REFC 信号（不对称 PWM 模式 1），则由于不对称 PWM 模式 1 的原因，通道 2 上可输出 OC2REF 信号或 OC2REFC 信号。

图 372 显示了不对称 PWM 模式下可以产生的信号示例（通道 1 到通道 4 在不对称 PWM 模式 1 下配置）。与死区发生器配合使用时，这可控制相移全桥直流到直流转换器。

图 372. 50% 占空比时产生的 2 个相移 PWM 信号



38.3.13 组合 PWM 模式

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的“逻辑或”运算或者“逻辑与”运算组合组成。

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

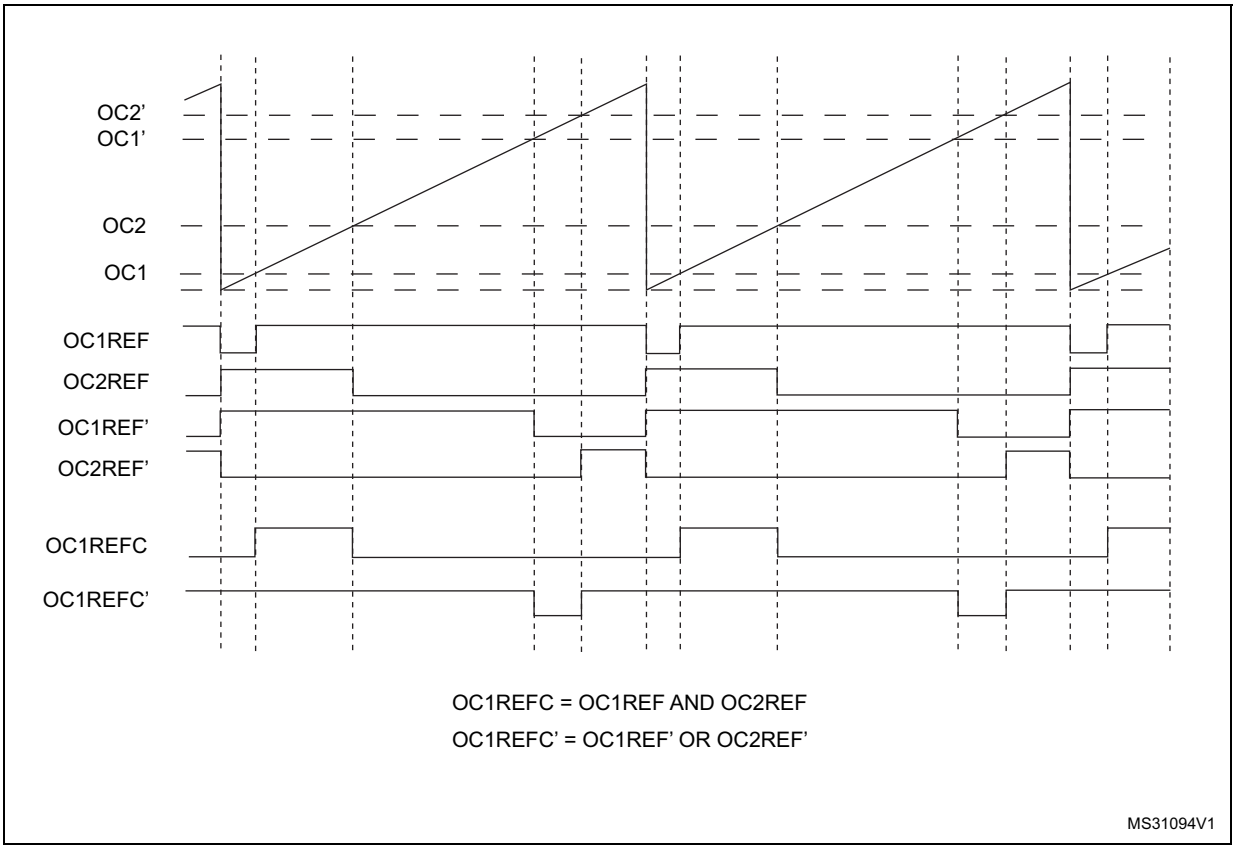
当给定通道用作组合 PWM 通道时，其互补通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注： *出于兼容性原因，OCxM[3:0] 位字段分为两部分，最高有效位与最低有效的 3 位不相邻。*

图 373 显示了不对称 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 配置为组合 PWM 模式 2，
- 通道 2 配置为 PWM 模式 1，
- 通道 3 配置为组合 PWM 模式 2，
- 通道 4 在 PWM 模式 1 下配置。

图 373. 通道 1 和通道 3 上的组合 PWM 模式



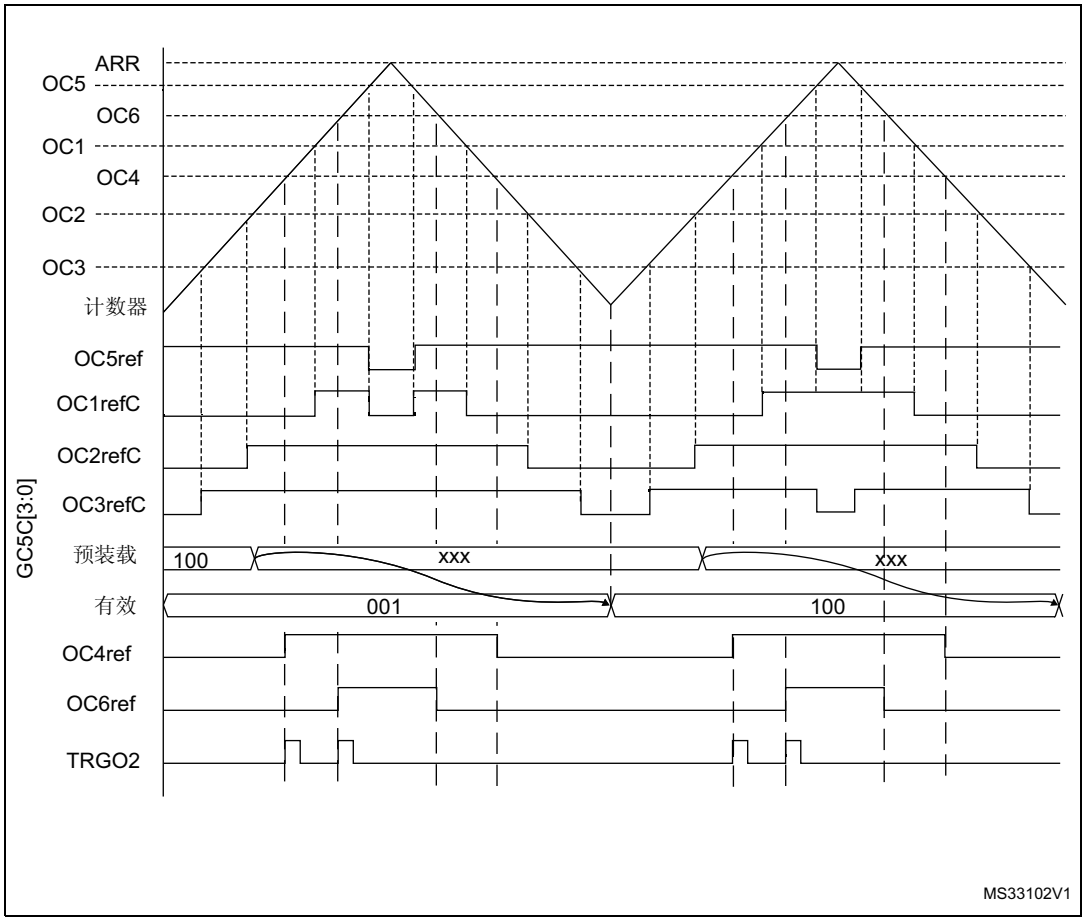
38.3.14 组合三相 PWM 模式

在组合三相 PWM 模式下，产生的一至三个中心对齐 PWM 信号与一个可编程信号间允许在脉冲中间进行逻辑与运算。OC5REF 信号用于定义产生的组合信号。通过 TIMx_CCR5 中的 3 位 GC5C[3:1]，可以选择 OC5REF 与哪个参考信号组合。产生的信号 OCxREFC 由两个参考 PWM 的逻辑与运算组合组成：

- 如果 GC5C1 置 1，则 OC1REFC 由 TIMx_CCR1 和 TIMx_CCR5 控制
- 如果 GC5C2 置 1，则 OC2REFC 由 TIMx_CCR2 和 TIMx_CCR5 控制
- 如果 GC5C3 置 1，则 OC3REFC 由 TIMx_CCR3 和 TIMx_CCR5 控制

通道 1 到通道 3 可独立选择组合三相 PWM 模式，只需将 3 位 GC5C[3:1] 中的至少一位置 1。

图 374. 三相组合 PWM 信号（每个周期多个触发脉冲）



TRGO2 波形说明了如何根据给定的三相 PWM 信号同步 ADC。更多详细信息，请参见*第 38.3.27 节：ADC 同步*。

38.3.15 互补输出和死区插入

高级控制定时器 (TIM1/TIM8) 可以输出两路互补信号，并管理输出的关断与接通瞬间。

这段时间通常称为死区，用户必须根据与输出相连接的设备及其特性（电平转换器的固有延迟、开关设备产生的延迟...）来调整死区时间

每路输出可以独立选择输出极性（主输出 OCx 或互补输出 OCxN）。可通过对 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位执行写操作来完成极性选择。

互补信号 OCx 和 OCxN 通过以下多个控制位的组合进行激活：TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位以及 TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。更多详细信息，请参见*第 1966 节的表 351：具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位*。应当注意，切换至空闲状态（MOE 位变为 0）的时刻，死区仍然有效。

CCxE 和 CCxNE 位同时置 1 并且 MOE 位置 1（若存在刹车电路）时，将使能死区插入。每个通道有一个 10 位死区发生器。将基于参考波形 OCxREF 生成 2 个输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高电平有效：

- 输出信号 OCx 与参考信号相同，只是其上升沿相对参考上升沿存在延迟。
- 输出信号 OCxN 与参考信号相反，并且其上升沿相对参考下降沿存在延迟。

如果延迟时间大于有效输出（OCx 或 OCxN）的宽度，则不会产生相应的脉冲。

下图所示为死区发生器的输出信号与参考信号 OCxREF 之间的关系。（在这些示例中，假定 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1）

图 375. 带死区插入的互补输出

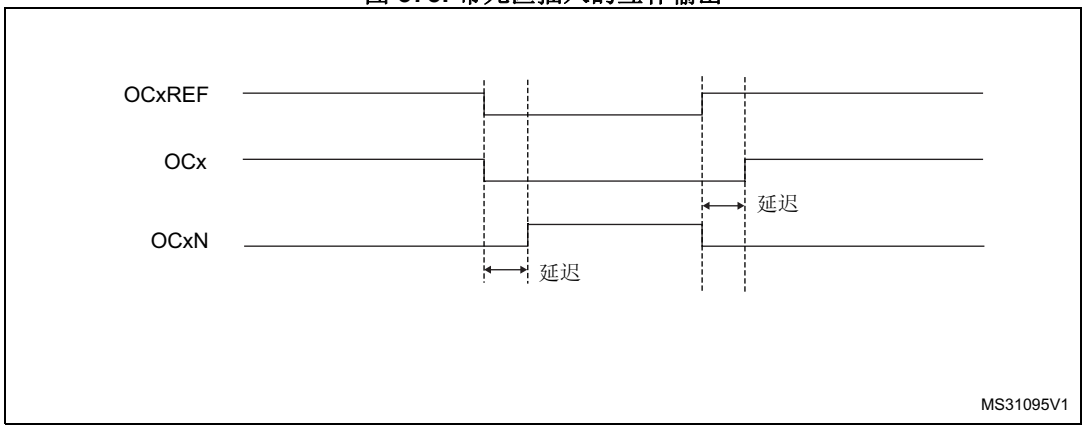


图 376. 延迟时间大于负脉冲宽度的死区波形

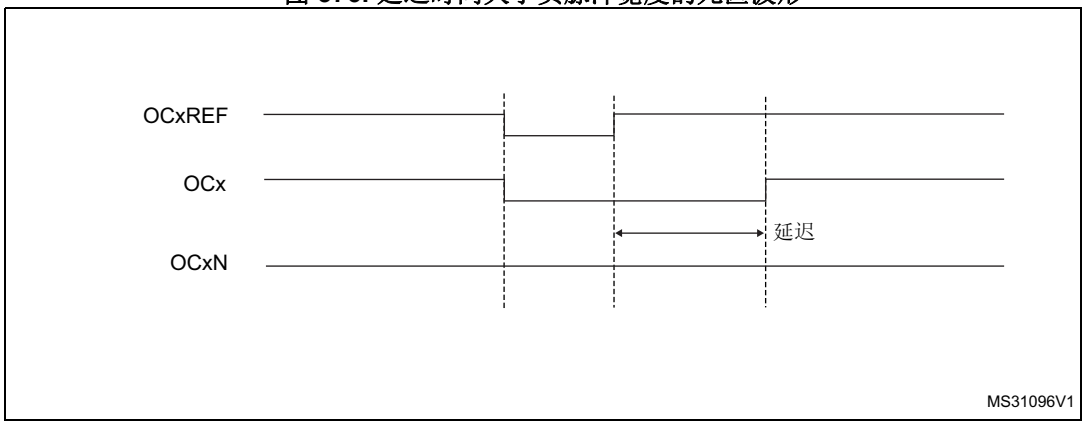
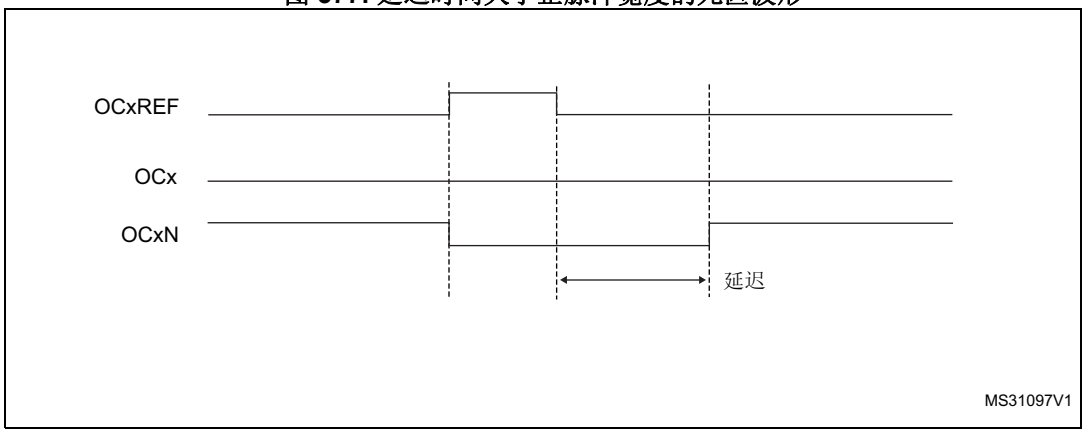


图 377. 延迟时间大于正脉冲宽度的死区波形



死区延迟对于所有通道均相同，可通过 TIMx_BDTR 寄存器中的 DTG 位进行编程。有关延迟时间计算的信息，请参见*第 38.4.20 节：TIMx 刹车和死区寄存器 (TIMx_BDTR)（x = 1、8）*。

将 OCxREF 重定向到 OCx 或 OCxN

在输出模式（强制输出模式、输出比较模式或 PWM 模式）下，通过配置 TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位，可将 OCxREF 重定向到 OCx 输出或 OCxN 输出。

通过此功能，可以在一个输出上发送特定波形（如 PWM 或静态有效电平），而同时使互补输出保持其无效电平。或者，使两个输出同时保持无效电平，或者两个输出同时处于有效电平，两者互补并且带死区。

注： *如果仅使能 OCxN (CCxE=0, CCxNE=1)，两者不互补，一旦 OCxREF 为高电平，OCxN 即变为有效。例如，如果 CCxNP=0，则 OCxN=OCxRef。另一方面，如果同时使能 OCx 和 OCxN (CCxE=CCxNE=1)，OCx 在 OCxREF 为高电平时变为有效，而 OCxN 则与之互补，在 OCxREF 为低电平时变为有效。*

38.3.16 使用刹车功能

刹车功能的用途是保护由 TIM1 和 TIM8 定时器产生的 PWM 信号所驱动的功率开关。两个刹车输入通常连接到功率级和三相逆变器的故障输出。激活时，刹车电路会关闭 PWM 输出，并将其强制为预定义的安全状态。也可选择一些内部 MCU 事件来触发输出关断。

有两个刹车通道。一个刹车通道收集系统级故障（时钟失效和奇偶校验错误等）和应用故障（来自输入引脚），可以在死区持续时间后将输出强制为预定义的电平（有效或无效）。刹车 2 通道只包括应用故障，能够将输出强制为无效状态。

刹车期间的输出使能信号和输出电平取决于多个控制位：

- TIMx_BDTR 寄存器中的 MOE 位，允许通过软件使能/禁止输出，在发生刹车事件和刹车 2 事件时复位。
- TIMx_BDTR 寄存器中的 OSSI 位，定义定时器将输出控制在无效状态下，还是将输出控制释放给 GPIO 控制器（通常使其处于高阻态模式）
- TIMx_CR2 寄存器中的 OISx 和 OISxN 位，将输出设置为关断电平（有效或无效）。无论 OISx 和 OISxN 的值为何，均无法在给定时间将 OCx 和 OCxN 输出同时设置为有效电平。更多详细信息，请参见*第 1966 节的表 351：具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位*。

退出复位状态后，刹车功能处于禁止状态，MOE 位处于低电平。通过设置 TIMx_BDTR 寄存器中的 BKE 位和 BK2E 位来使能刹车功能。可通过配置同一寄存器中的 BKP 位和 BK2P 位来选择刹车输入的极性。BKE/BK2E 和 BKP/BK2P 位可同时修改。对 BKE/BK2E 和

BKP/BK2P 位执行写操作时，写操作会在 1 个 APB 时钟周期的延迟后生效。因此，执行写操作后，需要等待 1 个 APB 时钟周期，才能准确回读此位。

由于 MOE 下降沿可能是异步信号，因此在实际信号（作用于输出）与同步控制位（位于 TIMx_BDTR 寄存器中）之间插入了再同步电路，从而在异步信号与同步信号之间产生延迟。具体而言，如果在 MOE 处于低电平时将其置 1，则必须首先插入延迟（空指令），才能准确进行读取。这是因为写入的是异步信号，而读取的却是同步信号。

刹车 (BRK) 通道的源为：

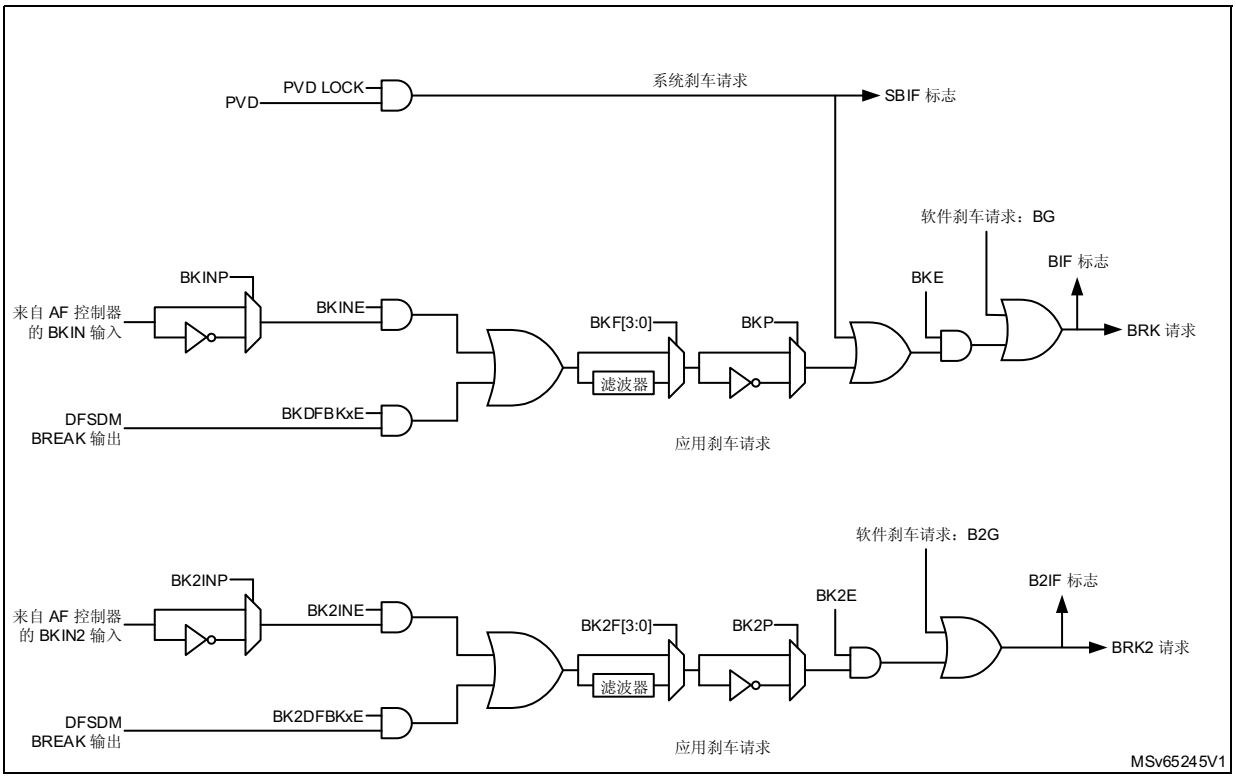
- 连接到 BKIN 引脚的外部源（由 GPIO 复用功能寄存器选定），具有极性选择和可选的数字滤波
- 内部源：
  - DFSDM1 外设的模拟看门狗输出
  - PVD 输出

刹车 2 (BRK2) 的源是连接到 BKIN 引脚的外部源（由 GPIO 复用功能寄存器选定），具有极性选择和可选的数字滤波。

也可由软件通过 TIMx_EGR 寄存器中的 BG 和 B2G 位产生刹车事件。无论 BKE 和 BK2E 使能位的值如何，都可以使用 BG 和 B2G 通过软件生成刹车。

在所有源进入定时器 BRK 或 BRK2 输入之前，对其进行 OR 运算，如以下*图 378* 所示。

图 378. 刹车事件和刹车 2 电路概述



注： *只有禁止可编程滤波器时才能保证异步（无时钟）操作。如果使能可编程滤波器，必须使用故障安全时钟模式（例如，使用内部 PLL 和/或 CSS）来保证能够处理刹车事件。*

发生刹车之一（其中一个刹车输入上出现所选电平）时：

- MOE 位异步清零，使输出处于无效状态、空闲状态甚至释放控制权给 GPIO 控制器（通过 OSSI 位进行选择）。即使 MCU 振荡器关闭，该功能仍然使能。
- MOE=0 时，将以 TIMx_CR2 寄存器 OISx 位中设定的电平驱动每个输出通道。如果 OSSI=0，定时器将释放输出控制（由 GPIO 控制器接管），否则使能输出保持高电平。
- 使用互补输出时：
  - 输出首先置于无效状态（取决于极性）。这是异步操作，因此即使没有为定时器提供时钟，该操作仍有效。
  - 如果定时器时钟仍存在，则将重新激活死区发生器，进而在死区后以 OISx 和 OISxN 位中编程的电平驱动输出。即使在这种情况下，也不能同时将 OCx 和 OCxN 驱动至其有效电平。请注意，MOE 进行再同步，因此死区的持续时间会比通常情况稍长一些（约 2 个 ck_tim 时钟周期）。
  - 如果 OSSI=0，定时器将释放输出控制（由强制高阻态的 GPIO 控制器接管），否则使能输出将保持高电平或在 CCxE 或 CCxNE 位之一为高电平时立即变为高电平。
- 将刹车状态标志（TIMx_SR 寄存器中的 SBIF、BIF 和 B2IF 位）置 1。如果 TIMx_DIER 寄存器中的 BIE 位置 1，则会产生中断。
- 如果 TIMx_BDTR 寄存器中的 AOE 位置 1，则 MOE 位会在发生下一更新事件 (UEV) 时自动再次置 1。例如，这可用于执行调节。否则，MOE 将始终保持低电平，直到应用将其再次置 1。这种情况下，这一特性可用于确保安全。可以将刹车输入连接到功率驱动器的警报、温度传感器或任何安全元件。

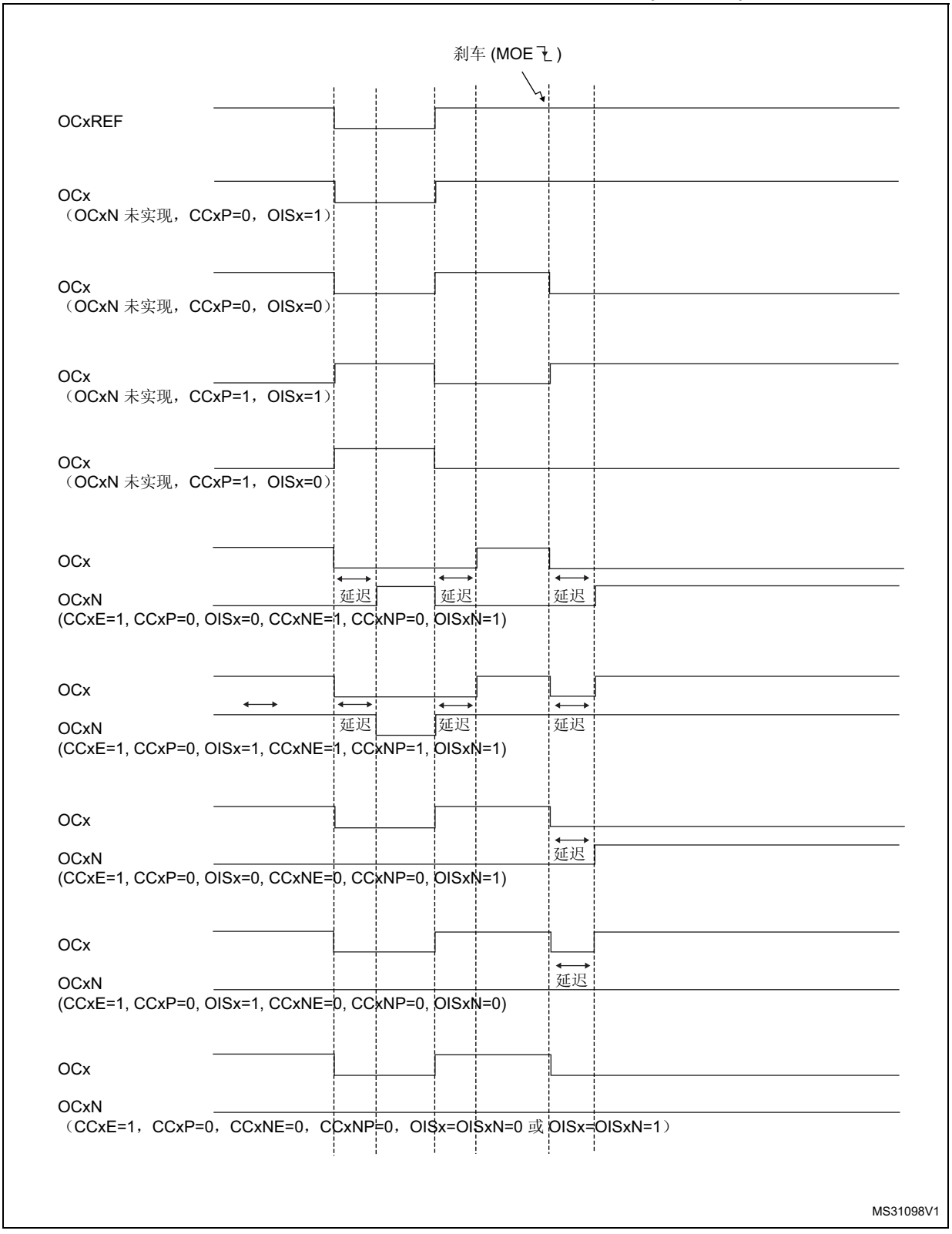
注： *当 AOE 位置 1 时，如果 CPU 将 MOE 复位，则输出处于空闲状态并根据 OSSI 值被强制设为无效电平或高阻态。*
如果 CPU 同时将 MOE 和 AOE 位复位，则输出处于禁止状态并由 TIMx_CR2 寄存器中 OISx 位设定的电平驱动。

注： *刹车输入为电平有效。因此，当刹车输入为有效电平时，不能将 MOE 位置 1（自动或通过软件）。同时，不能将状态标志 BIF 和 B2IF 清零。*

除刹车输入和输出管理外，刹车电路内部还实施了写保护，用以保护应用的安全。通过该功能，用户可冻结多个参数配置（死区持续时间、OCx/OCxN 极性和禁止时的状态、OCxM 配置、刹车使能和极性）。应用可以通过 TIMx_BDTR 寄存器中的 LOCK 位，从 3 种保护级别中进行选择。请参见*第 38.4.20 节：TIMx 刹车和死区寄存器 (TIMx_BDTR)（x = 1、8）*。MCU 复位后只能对 LOCK 位执行一次写操作。

图 379 所示为输出对刹车响应行为的示例。

图 379. 响应 BRK 上的刹车事件的不同输出行为 (OSSI = 1)



两个刹车输入针对定时器输出具有不同的行为：

- BRK 输入可禁止（无效状态）PWM 输出，也可将 PWM 输出强制为预定义的安全状态。
- BRK2 只能禁止（无效状态）PWM 输出。

BRK 输入的优先级高于 BRK2 输入，如*表 347* 所示。

注： *BRK2 只能在 OSSR = OSSI = 1 时使用。*

表 347. 定时器输出行为与 BRK/BRK2 输入

BRK	BRK2	定时器输出状态	典型用例	
			OCxN 输出（下桥臂开关）	OCx 输出（上桥臂开关）
激活	X	– 无效，之后强制为输出状态（死区后） – 如果 OSSI = 0，则禁止输出（由 GPIO 逻辑接管控制）	死区插入后开启	关闭
无效	激活	无效	关闭	关闭

图 380 给出了 BRK 和 BRK2 输入上出现有效信号时 OCx 和 OCxN 输出行为示例。在这种情况下，两个输出的极性均为高电平有效（TIMx_CCER 寄存器中的 CCxP = CCxNP = 0）。

图 380. BRK 和 BRK2 引脚使能后的 PWM 输出状态 (OSSI=1)

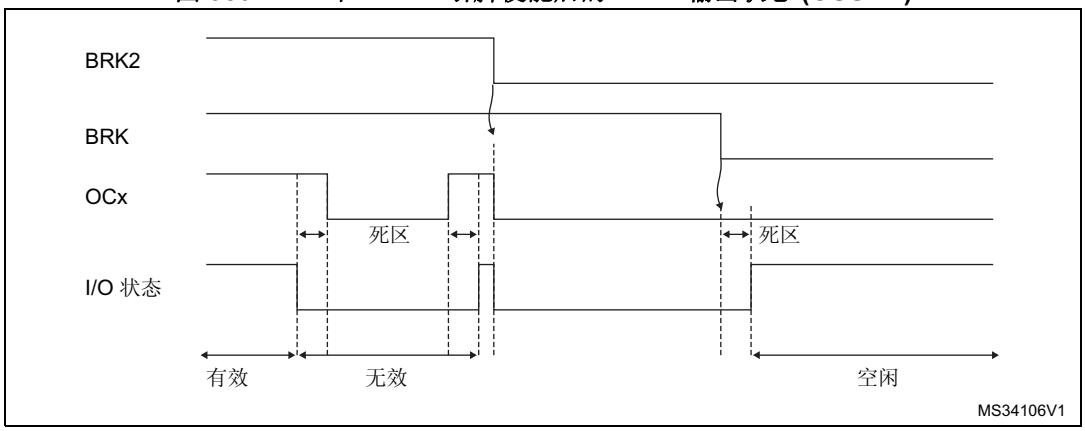
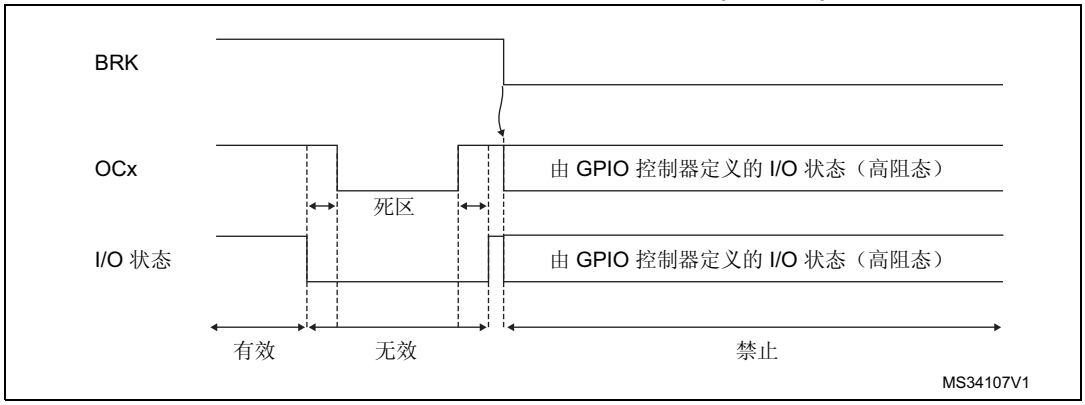


图 381. BRK 使能后的 PWM 输出状态 (OSSI=0)



38.3.17 双向刹车输入

TIM1/TIM8 具有双向刹车 I/O，如*图 382* 所示。

它们可以实现以下用途：

- 将板级全局刹车信号用于向外部 MCU 或栅极驱动器发送故障信号，唯一的特定引脚作为输入输出双向状态脚。
- 在必须将多个内部和外部刹车输入合并时，将内部刹车源和多个外部开漏比较器输出“或”连接在一起，触发唯一刹车事件。

使用 TIMxBDTR 寄存器的 BKBID 和 BK2BID 位，将刹车事件和刹车 2 输入配置为双向模式。可以使用 TIMxBDTR 寄存器中的 LOCK 位，将 BKBID 编程位锁定在只读模式（锁定级别 1 或更高）。

双向模式可以用于刹车事件和刹车 2 输入，需要将 I/O 配置为低电平有效极性的开漏模式（使用 BKINP、BKP、BK2INP 和 BK2P 位进行配置）。任何来自系统（例如 CSS）、片上外设或刹车输入的刹车请求都会强制将刹车输入置为低电平，以通知发生了故障事件。如果未正确设置极性位（高电平有效极性），则出于安全目的禁止双向模式。

软件刹车事件（BG 和 B2G）也会导致刹车 I/O 被强制置为“0”，从而向外部组件指示定时器已进入刹车状态。但是仅在使能刹车 (BK(2)E = 1) 时有效。当生成软件刹车事件，并且 BK(2)E = 0 时，输出会置于安全状态，并将刹车标志置 1，但对刹车 (2) I/O 无影响。

安全解除机制可防止系统最终锁定（刹车输入上的低电平会触发刹车，进而将相同输入强制置为低电平）。

当 BKDSRM (BK2DSRM) 位置 1 时，会释放刹车输出以清除故障信号，从而使系统能够重新获得保护。

在任何情况下都不能禁止刹车保护电路：

- 刹车输入路径始终有效：即使 BKDSRM (BK2DSRM) 位置 1 且释放开漏控制，刹车事件也仍然有效。这样可以在发生刹车期间防止 PWM 输出重新启动。
- 使能输出（MOE 位置 1）后，BK(2)DSRM 位不能解除刹车保护（请参见*表 348*）

表 348. 刹车保护解除条件

MOE	BKDIR (BK2DIR)	BKDSRM (BK2DSRM)	刹车保护状态
0	0	X	启动
0	1	0	启动
0	1	1	解除
1	X	X	启动

启动和重新启动刹车电路

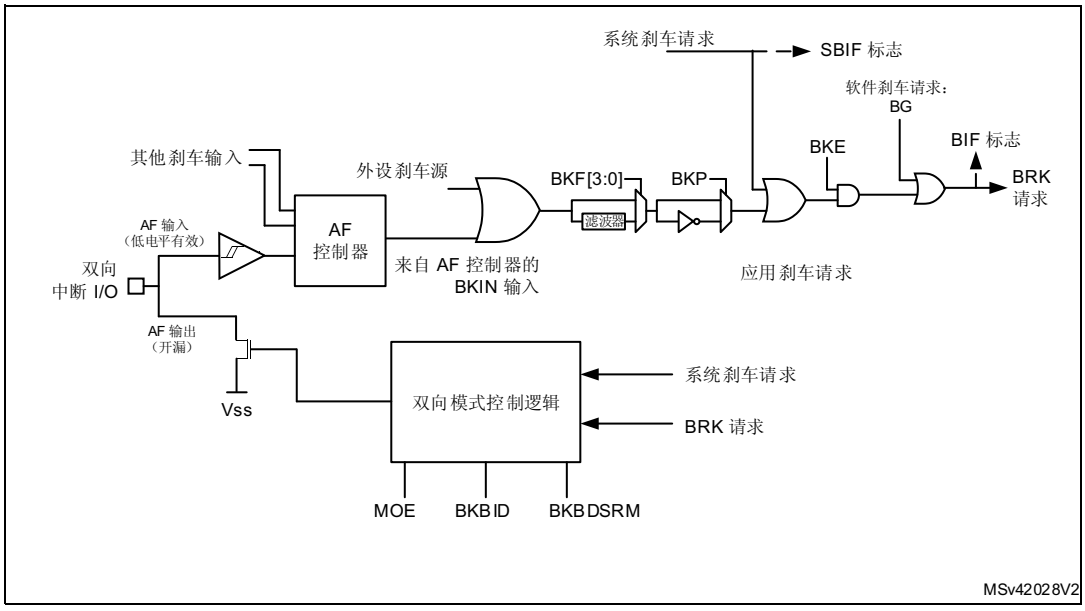
默认情况下（外设复位配置）会启动刹车电路（在输入或双向模式下）。

发生刹车（刹车 2）事件后，必须按照以下步骤重新启动保护：

- 必须将 BKDSRM (BK2DSRM) 位置 1，以释放输出控制
- 软件必须等待系统刹车条件消失（如果有），并清零 SBIF 状态标志（或在重新启动前由系统清零）
- 软件必须轮询 BKDSRM (BK2DSRM) 位，直到该位由硬件清零（当应用程序刹车条件消失时）

此后，刹车电路即启动并激活，可以通过将 MOE 位置 1 来重新使能 PWM 输出。

图 382. 输出重定向（图中未显示 BRK2 请求）



38.3.18 发生外部事件时清除 OCxREF 信号

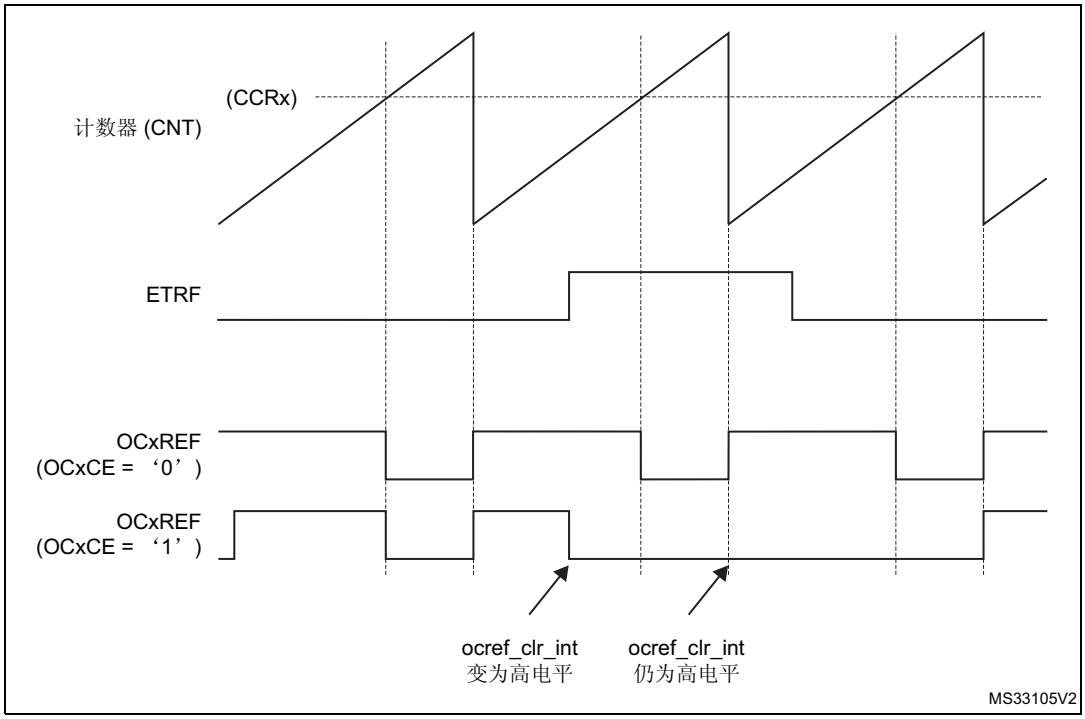
对于给定通道，在 ocref_clr_int 输入上施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置 1），可将 OCxREF 信号清零。OCxREF 信号将保持低电平，直到在下一 PWM 周期再次变为有效状态。该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。ocref_clr_int 连接到 ETRF 信号（滤波后的 ETR）。

选择 ETRF 时，ETR 必须配置如下：

1. 必须关闭外部触发预分频器：TIMx_SMCR 寄存器中的 ETPS[1:0] 位置“00”。
2. 必须禁止外部时钟模式 2：TIMx_SMCR 寄存器中的 ECE 位置“0”。
3. 外部触发极性 (ETP) 和外部触发滤波器 (ETF) 可根据用户需要进行配置。

图 383 对比了使能位 OCxCE 在不同值下的情况，显示了当 ETRF 输入变为高电平时 OCxREF 信号的行为。在本例中，定时器 TIMx 编程为 PWM 模式。

图 383. 清除 TIMx 的 OCxREF



注： *如果 PWM 的占空比为 100% (CCRx>ARR)，则下次计数器溢出时会再次使能 OCxREF。*

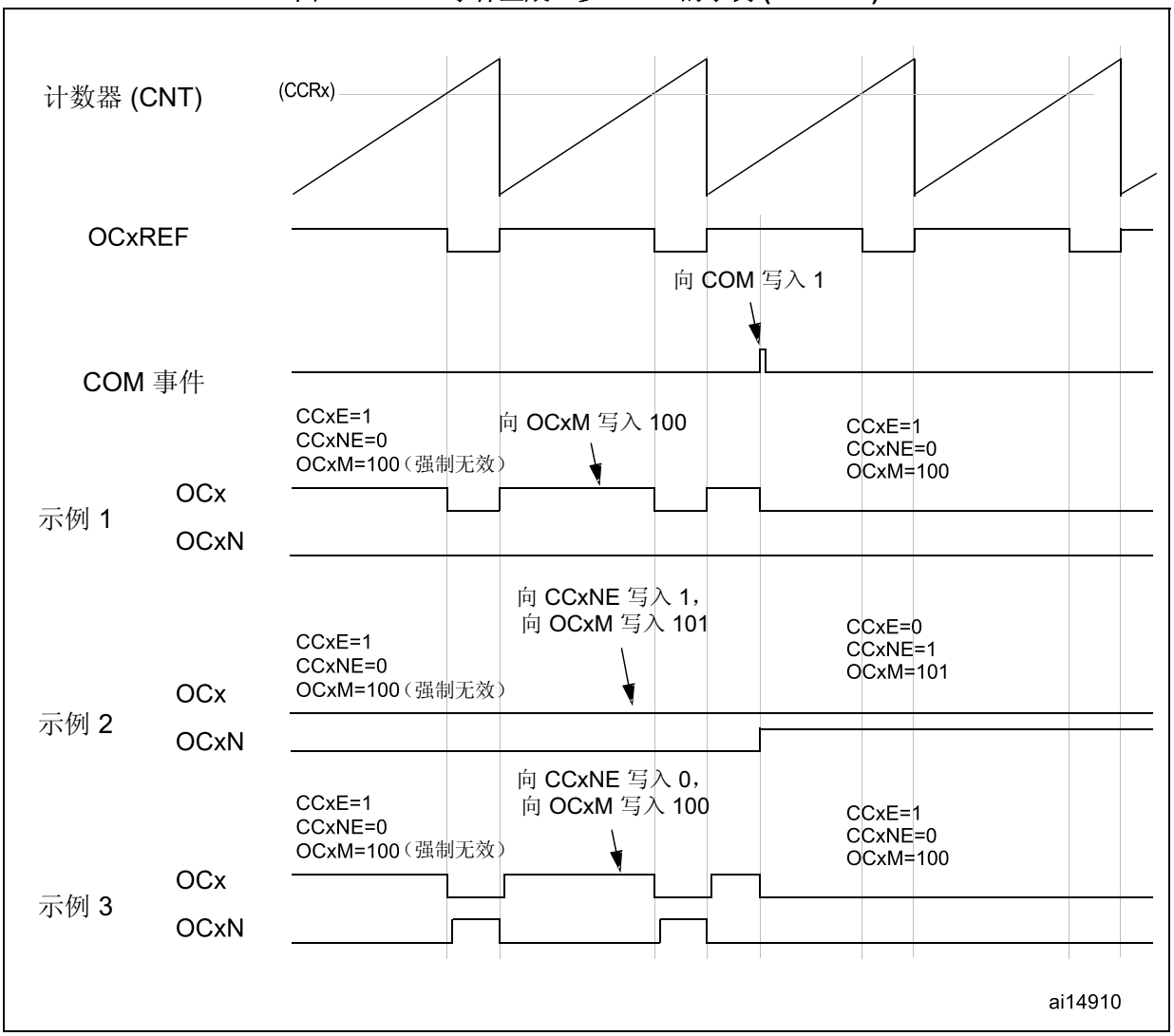
38.3.19 生成 6 步 PWM

当通道使用互补输出时，OCxM、CCxE 和 CCxNE 位上提供预装载位。发生 COM 换向事件时，这些预装载位将传输到影子位。因此，用户可以预先编程下一步骤的配置，并同时更改所有通道的配置。COM 可由软件通过将 TIMx_EGR 寄存器中的 COM 位置 1 而生成，也可以由硬件在 TRGI 上升沿生成。

发生 COM 事件时，某个标志位（TIMx_SR 寄存器中的 COMIF 位）将会置 1。这时，如果 TIMx_DIER 寄存器中的 COMIE 位置 1，将产生中断；如果 TIMx_DIER 寄存器中的 COMDE 位置 1，则将产生 DMA 请求。

图 384 以 3 种不同的编程配置为例，显示了发生 COM 事件时 OCx 和 OCxN 输出的行为。

图 384. COM 事件生成 6 步 PWM 的示例 (OSSR=1)



38.3.20 单脉冲模式

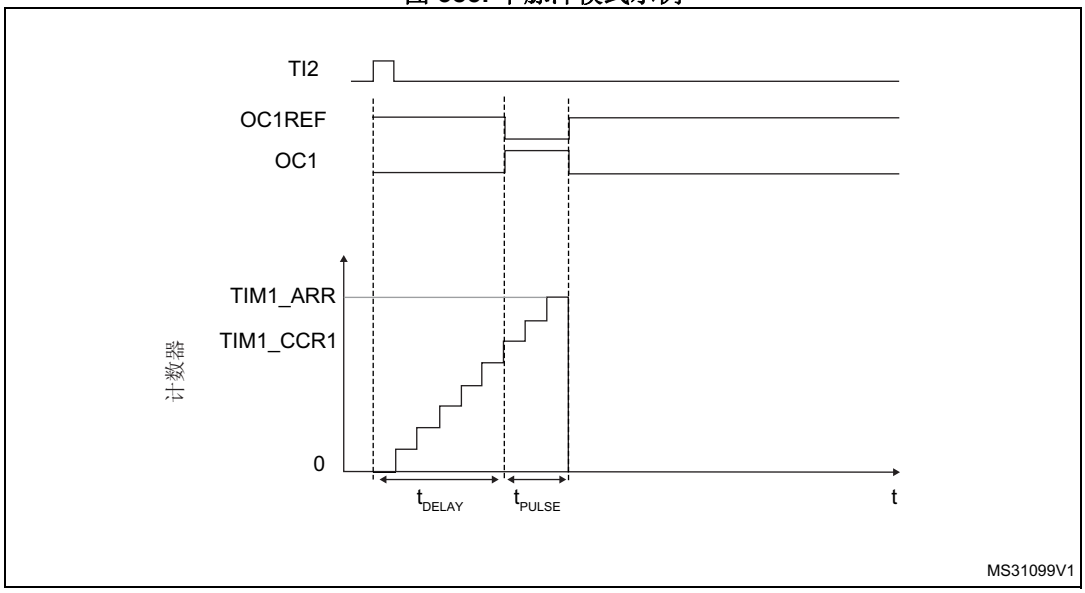
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。通过将 TIMx_CR1 寄存器中的 OPM 位置 1 选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

- 递增计数模式下：CNT < CCRx ≤ ARR（特别注意，0 < CCRx）
- 递减计数模式下：CNT > CCRx

图 385. 单脉冲模式示例



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2x 源（内部或外部）。
2. 在 TIMx_CCMR1 寄存器中写入 CC2S=“01”，以将 TI2FP2 映射到 TI2。
3. 在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
4. 在 TIMx_SMCR 寄存器中写入 TS=00110，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
5. 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），以使用 TI2FP2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，必须通过在 TIMx_CCMR1 寄存器中写入 OC1M=111，来使能 PWM 模式 2。可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

在本例中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此必须向 TIMx_CR1 寄存器的 OPM 位写入 1，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIMx_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特例：OCx 快速使能：

在单脉冲模式下，TIx 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟（t_{DELAY} 最小值）。

如果要输出延迟时间最短的波形，可以将 TIMx_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 OCxRef（和 OCx）对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

38.3.21 可再触发单脉冲模式

该模式可以启动计数器来响应激励信号，并且能产生长度可编程的脉冲，但与*第 38.3.20 节*所描述的不可再触发单脉冲模式间存在以下差别：

- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长

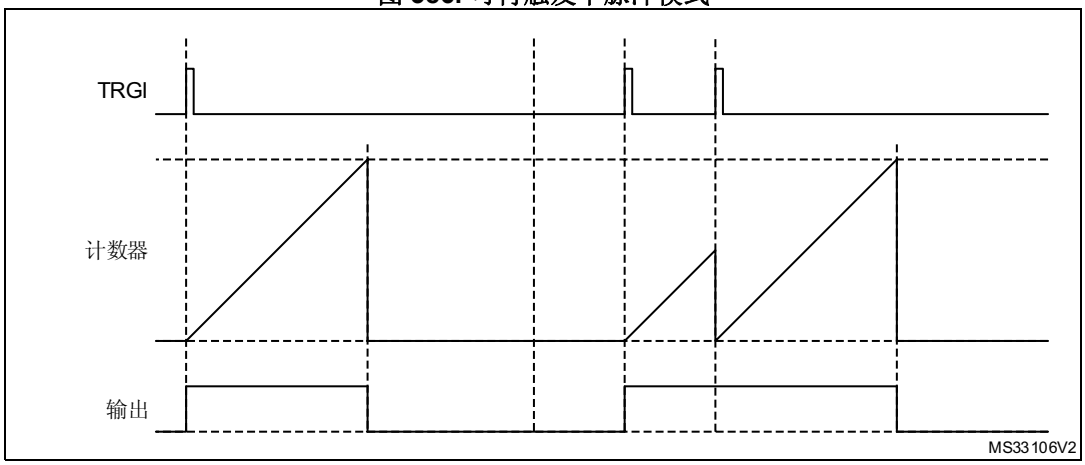
定时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0] = “1000”（组合复位 + 触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0] 位设置为“1000”或“1001”。

定时器配置为递增计数模式时，相应的 CCRx 必须置 0（ARR 寄存器设置脉冲长度）。如果定时器配置为递减计数模式，CCRx 必须高于或等于 ARR。

注： *出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位字段分为两部分，最高有效位与最低有效的 3 位不相邻。*

此模式不能与中心对齐 PWM 模式一起使用。在 TIMx_CR1 中必须设置 CMS[1:0] = 00。

图 386. 可再触发单脉冲模式



38.3.22 编码器接口模式

选择编码器接口模式时，如果计数器仅在 TI2 边沿处计数，在 TIMx_SMCR 寄存器中写入 SMS=“001”；如果计数器仅在 TI1 边沿处计数，写入 SMS=“010”；如果计数器在 TI1 和 TI2 边沿处均计数，则写入 SMS=“011”。

通过编程 TIMx_CCER 寄存器的 CC1P 和 CC2P 位，选择 TI1 和 TI2 极性。如果需要，还可对输入滤波器进行编程。CC1NP 和 CC2NP 必须保持低电平。

TI1 和 TI2 两个输入用于连接正交编码器。请参见 *表 349*。如果使能计数器（在 TIMx_CR1 寄存器的 CEN 位中写入“1”），则计数器的时钟由 TI1FP1 或 TI2FP2 上的每次有效信号转换提供。TI1FP1 和 TI2FP2 是 TI1 和 TI2 进行输入滤波和极性选择后的信号，如果不进行滤波和反相，则 TI1FP1=TI1，TI2FP2=TI2。将根据两个输入的信号转换序列，产生计数脉冲和方向信号。根据该信号转换序列，计数器相应递增或递减计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应修改。任何输入（TI1 或 TI2）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 TI1 或 TI2 边沿处计数，还是同时在 TI1 和 TI2 处计数。

编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 TIMx_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIMx_ARR。同样，捕获、比较、重复计数器和触发输出功能继续正常工作。编码器模式和外部时钟模式 2 不兼容，因此不能同时选择。

注： *使能编码器模式时，预分频器必须设置为零*

在此模式下，计数器会根据正交编码器的速度和方向自动进行修改，因此，其内容始终表示编码器的位置。计数方向对应于所连传感器的旋转方向。下表汇总了可能的组合（假设 TI1 和 TI2 不同时切换）。

表 349. 计数方向与编码器信号的关系

有效边沿	相反信号的电平（TI1FP1 对应 TI2，TI2FP2 对应 TI1）	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降
仅在 TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在 TI2 处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在 TI1 和 TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

正交编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将编码器的差分输出转换为数字信号。这样大幅提高了抗噪声性能。用于指示机械零位的第三个编码器输出可与外部中断输入相连，用以触发计数器复位。

图 387 以计数器工作为例，说明了计数信号的产生和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S=“01”（TIMx_CCMR1 寄存器，TI1FP1 映射到 TI1 上）。
- CC2S=“01”（TIMx_CCMR1 寄存器，TI2FP2 映射到 TI2 上）
- CC1P=“0”，CC1NP=“0”（TIMx_CCER 寄存器，TI1FP1 未反相，TI1FP1=TI1）。
- CC2P=“0”，CC2NP=“0”（TIMx_CCER 寄存器，TI1FP2 未反相，TI1FP2= TI2）。
- SMS=“011”（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）。
- CEN=“1”（TIMx_CR1 寄存器，使能计数器）。

图 387. 编码器接口模式下的计数器工作示例

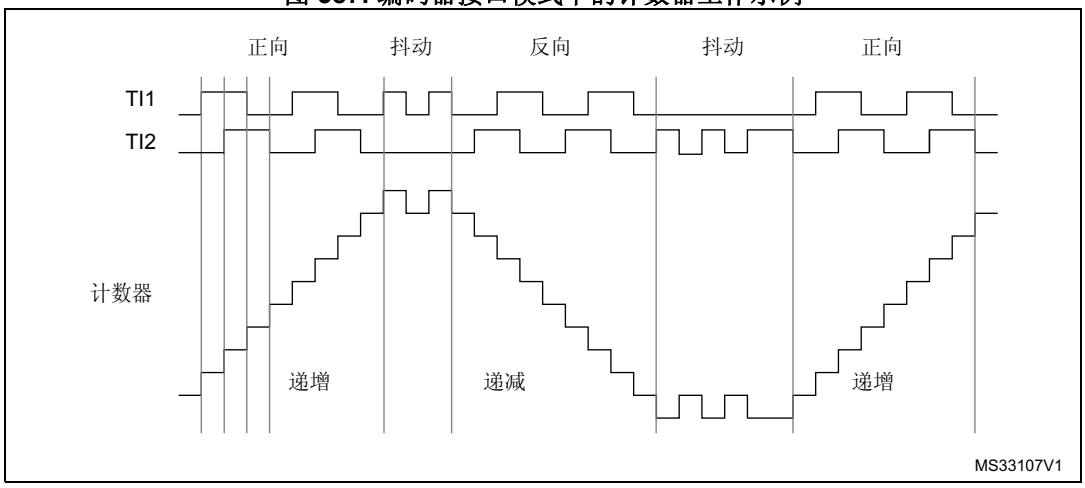
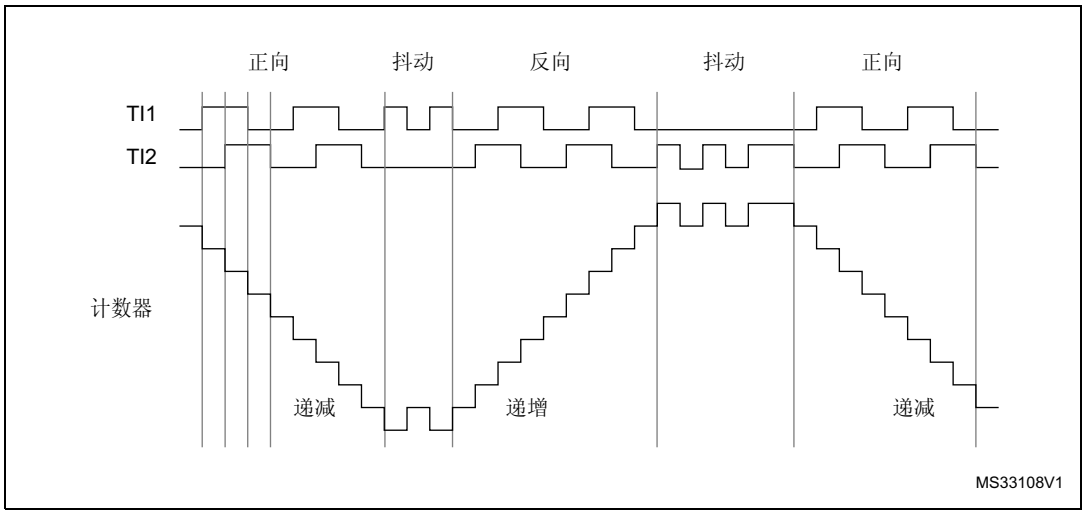


图 388 举例说明 TI1FP1 极性反相时计数器的行为（除 CC1P=“1”外，其他配置与上例相同）。

图 388. TI1FP1 极性反相时的编码器接口模式示例



定时器配置为编码器接口模式时，会提供传感器当前位置的相关信息。使用另一个配置为捕获模式的定时器测量两个编码器事件之间的周期，可获得动态信息（速度、加速度和减速度）。指示机械零位的编码器输出即可用于此目的。根据两个事件之间的时间间隔，还可定期读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕获信号必须为周期性信号，可以由另一个定时器产生）；还可以通过 DMA 请求读取计数器值。

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 (UIF) 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可按原子方式读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化角速度的计算。

UIF 和 UIFCPY 标志之间的置位没有延迟。

在 32 位定时器实现中，当 IUFREMAP 位置 1 时，计数器的位 31 在读访问时由 UIFCPY 标志覆盖（计数器的最高有效位只能在写模式下访问）。

38.3.23 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时器计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可按原子方式读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

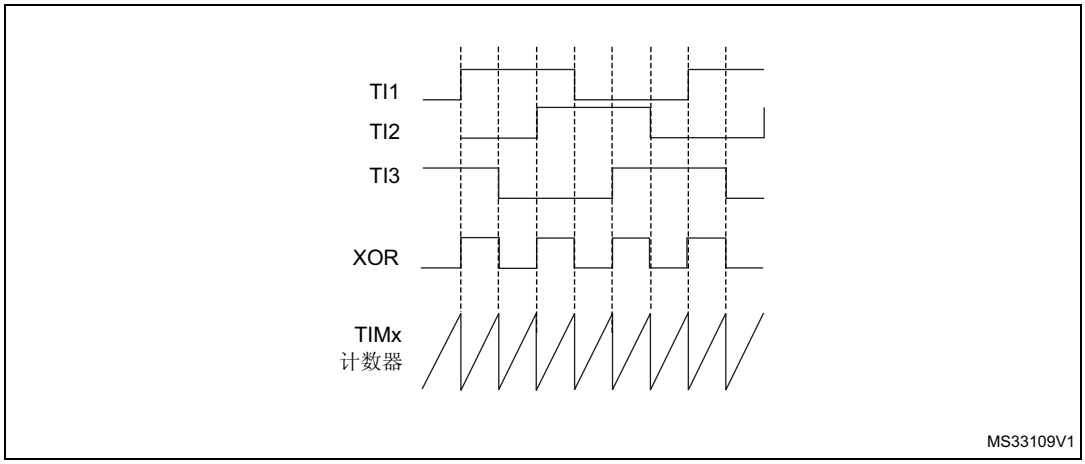
UIF 和 UIFCPY 标志之间的置位没有延迟。

38.3.24 定时器输入异或功能

通过 TIMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 到 TIMx_CH3 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。这样便于测量两个输入信号上边沿之间的间隔（如下面的*图 389* 所示）。

图 389. 测量 3 个信号上边沿之间的时间间隔



38.3.25 连接霍尔传感器

可通过用于产生电机驱动 PWM 信号的高级控制定时器（TIM1 或 TIM8）以及*图 390* 中称为“接口定时器”的另一个定时器 TIMx（TIM2、TIM3 和 TIM4），实现与霍尔传感器的连接。3 个定时器输入引脚（CC1、CC2 和 CC3）通过异或门连接到 TI1 输入通道（通过将 TIMx_CR2 寄存器中的 TI1S 位置 1 来选择），并由“接口定时器”进行捕获。

从模式控制器配置为复位模式；从输入为 TI1F_ED。这样，每当 3 个输入中有一个输入发生切换时，计数器会从 0 开始重新计数。这样将产生由霍尔输入的任何变化而触发的时基。

在“接口定时器”上，捕获/比较通道 1 配置为捕获模式，捕获信号为 TRC（请参见*第 1912 页的图 363：捕获/比较通道（例如：通道 1 输入阶段）*）。捕获值对应于输入上两次变化的间隔时间，可提供与电机转速相关的信息。

“接口定时器”可用于在输出模式下产生脉冲，以通过触发 COM 事件更改高级控制定时器（TIM1 或 TIM8）各个通道的配置。TIM1 定时器用于生成电机驱动 PWM 信号。为此，必须对接口定时器通道进行编程，以便在编程的延迟过后产生正脉冲（在输出比较或 PWM 模式中）。该脉冲通过 TRGO 输出发送到高级控制定时器（TIM1 或 TIM8）。

示例：霍尔输入与一个 TIMx 定时器相连接，每当霍尔输入发生更改，需要在所编程的延迟过后更改高级控制定时器 TIM1 的 PWM 配置。

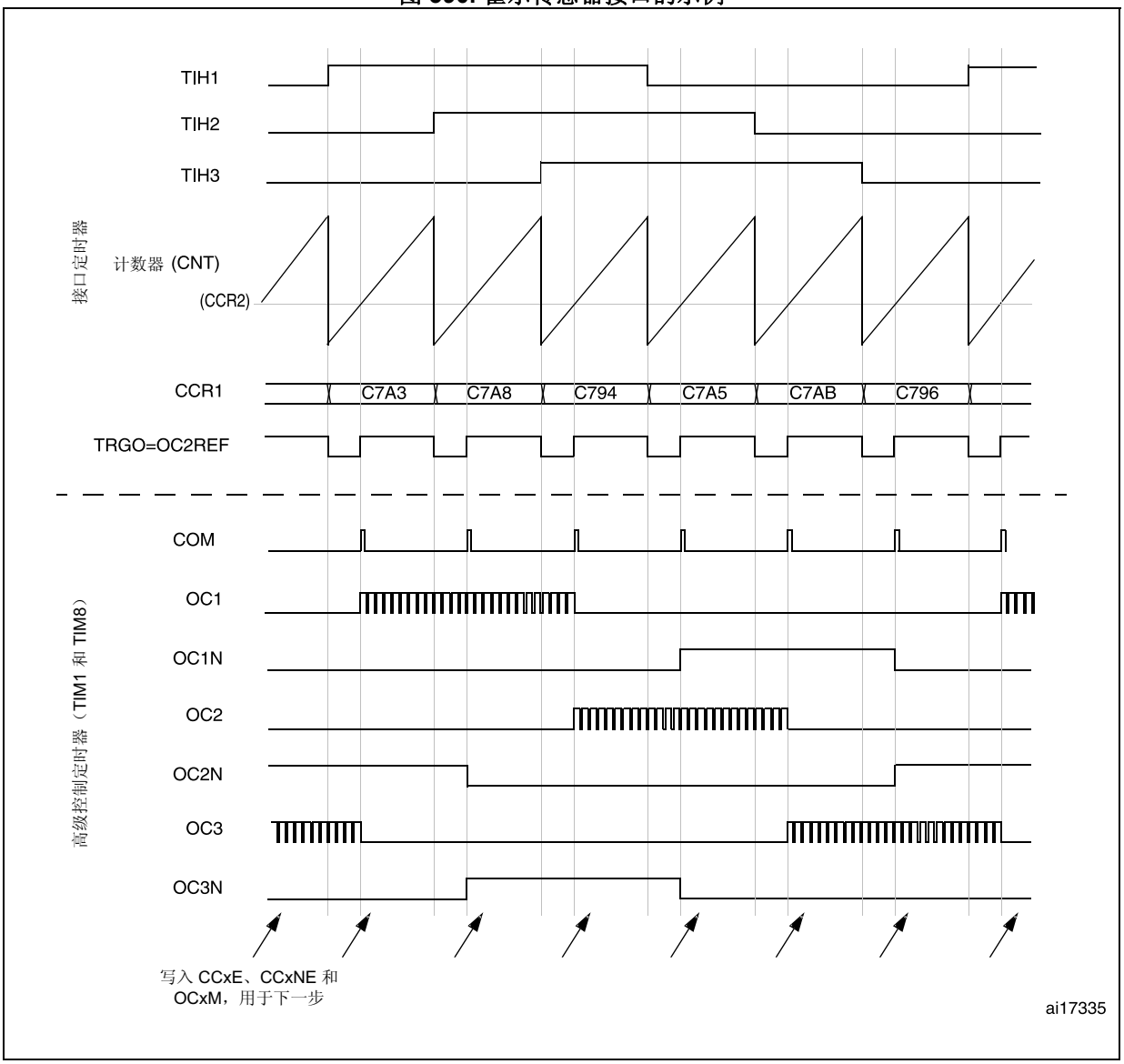
- 向 TIMx_CR2 寄存器的 TI1S 位写入“1”，使 3 个定时器输入经过异或运算后进入 TI1 输入通道。
- 时基编程：向 TIMx_ARR 写入其最大值（计数器必须通过 TI1 的变化清零）。设置预分频器，以得到最大计数器周期，该周期长于传感器上两次变化的间隔时间。
- 将通道 1 编程为捕获模式（选择 TRC）：向 TIMx_CCMR1 寄存器的 CC1S 位写入“01”。如果需要，还可以编程数字滤波器。
- 将通道 2 编程为 PWM 2 模式，并具有所需延迟：向 TIMx_CCMR1 寄存器的 OC2M 位写入“111”，CC2S 位写入“00”。

- 选择 OC2REF 作为 TRGO 上的触发输出：向 TIMx_CR2 寄存器的 MMS 位写入“101”。

在高级控制定时器 TIM1 中，必须选择正确的 ITR 输入作为触发输入，定时器编程为可产生 PWM 信号，捕获/比较控制信号进行预装载（TIMx_CR2 寄存器的 CCPC=1），并且 COM 事件由触发输入控制（TIMx_CR2 寄存器中 CCUS=1）。发生 COM 事件后，在 PWM 控制位（CCxE、OCxM）中写入下一步的配置，此操作可在由 OC2REF 上升沿产生的中断子程序中完成。

图 390 为本示例的示意图。

图 390. 霍尔传感器接口的示例



38.3.26 定时器同步

TIMx 定时器从内部连接在一起，以实现定时器同步或链接。有关详细信息，请参见*第 39.3.19 节：定时器同步*。它们可在以下几种模式下实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

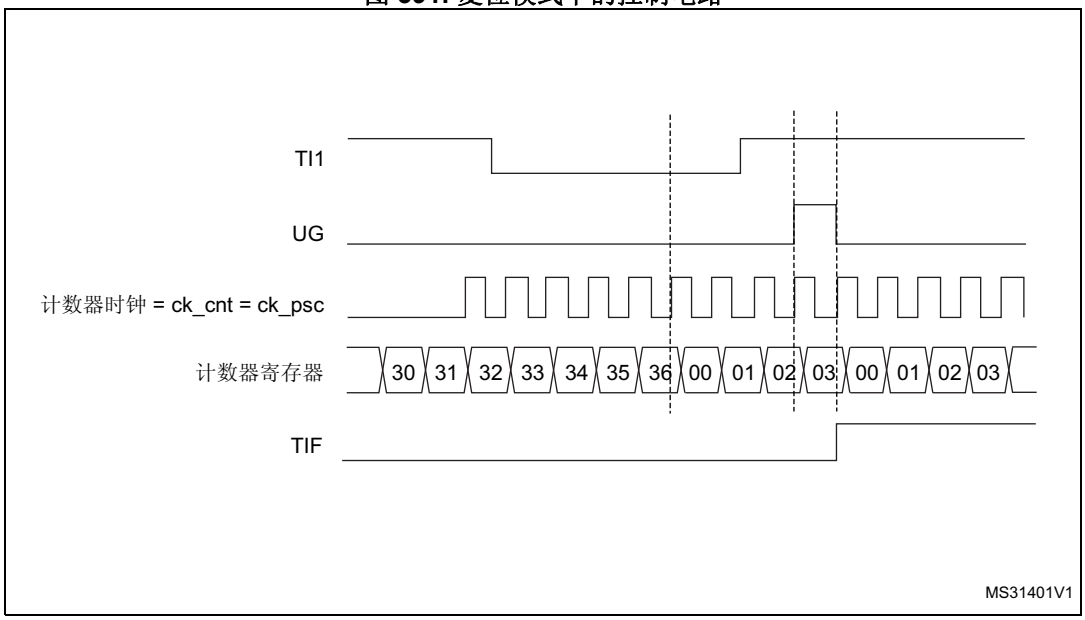
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

- 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S = 01。在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP=“0”，验证极性（仅检测上升沿）。
- 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。通过在 TIMx_SMCR 寄存器中写入 TS= 00101 来选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 391. 复位模式下的控制电路



从模式：门控模式

输入信号的电平可用来使能计数器。

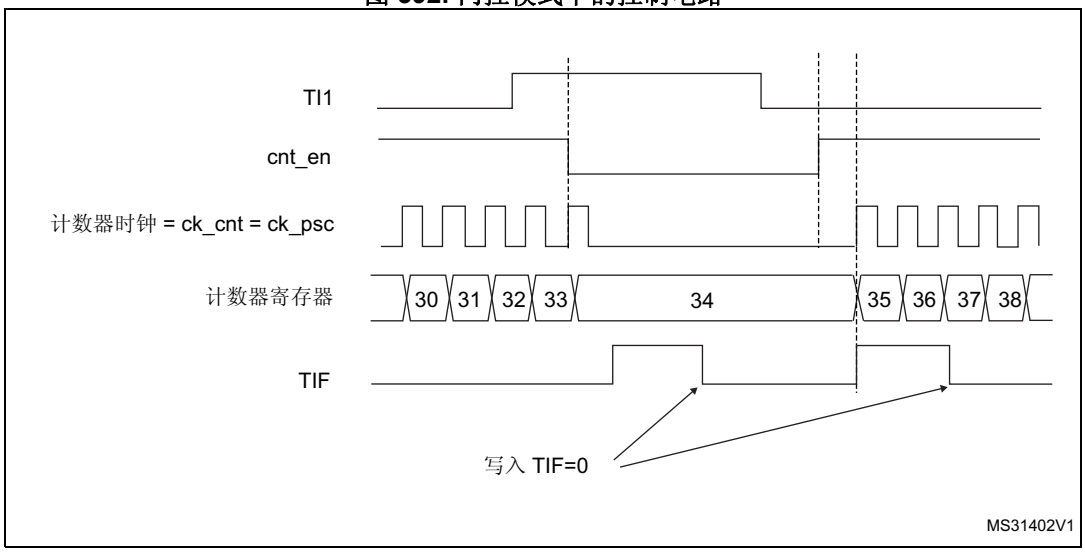
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

- 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=1 和 CC1NP=“0”，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。通过在 TIMx_SMCR 寄存器中写入 TS= 00101 来选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 392. 门控模式下的控制电路



从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

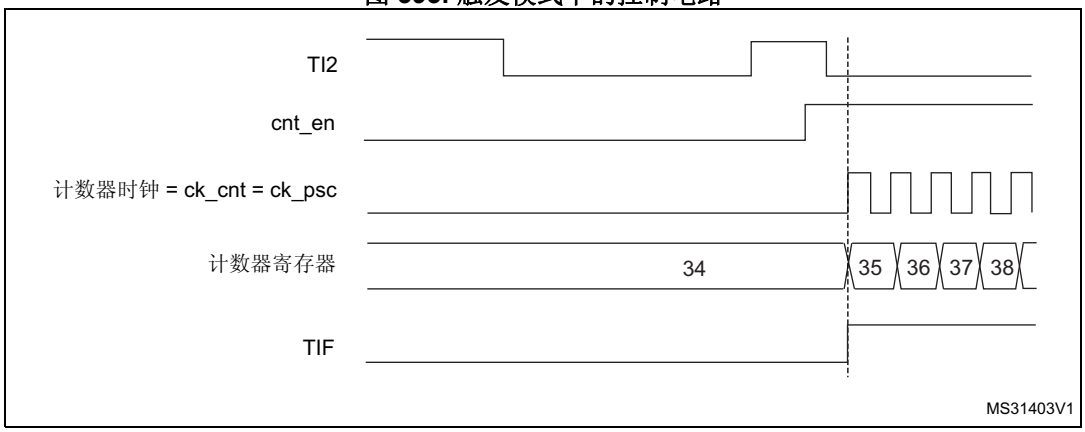
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

- 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位配置为只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 393. 触发模式下的控制电路



从模式：复位 + 触发组合模式

在这种情况下，在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个定时器更新事件，并启动计数器。

该模式用于单脉冲模式。

从模式：外部时钟模式 2 + 触发模式

外部时钟模式 2 可与另一种从模式（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下，ETR 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另一个输入作为触发输入。不建议通过 TIMx_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。

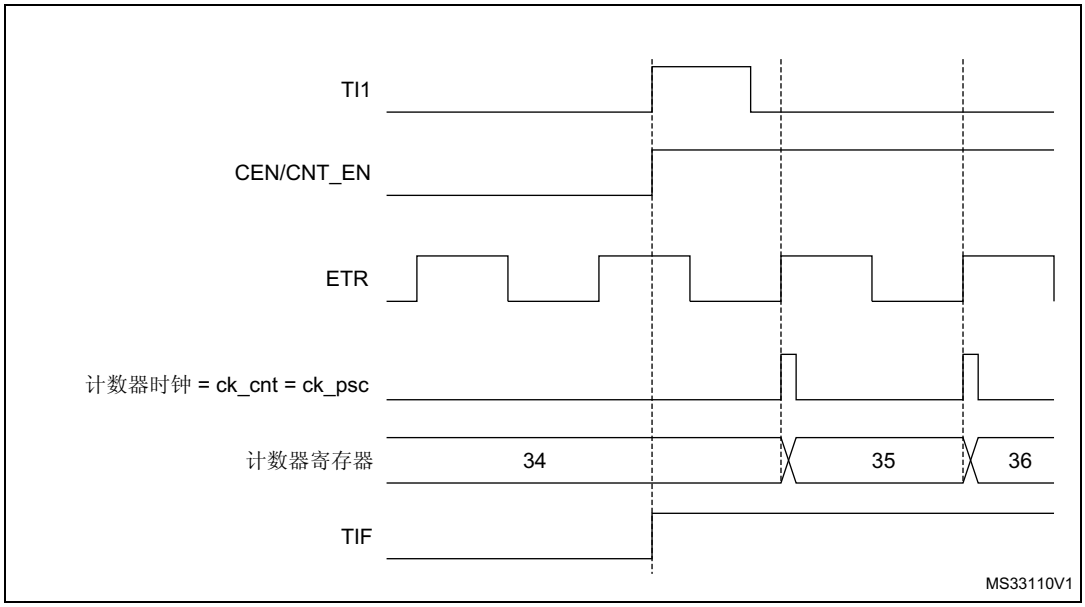
在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处递增：

1. 通过对 TIMx_SMCR 寄存器进行如下编程，配置外部触发输入电路：
   - ETF = 0000：无滤波器
   - ETPS = 00：禁止预分频器
   - ETP = 0：检测 ETR 的上升沿，并写入 ECE=1，以使能外部时钟模式 2。
2. 如下配置通道 1，以检测 TI 的上升沿：
   - IC1F = 0000：无滤波器
   - 由于捕获预分频器不用于触发操作，因此无需对其进行配置。
   - TIMx_CCMR1 寄存器中 CC1S = 01，只选择输入捕获源。
   - TIMx_CCER 寄存器中 CC1P = 0 且 CC1NP = 0，以确定极性（仅检测上升沿）。
3. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。通过在 TIMx_SMCR 寄存器中写入 TS= 00101 来选择 TI1 作为输入源。

TI1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 ETR 出现上升沿时计数。

ETR 信号的上升沿与实际计数器复位之间的延迟是由于 ETRP 输入的重新同步电路引起的。

图 394. 外部时钟模式 2 + 触发模式下的控制电路



注：必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。

38.3.27 ADC 同步

定时器可通过多种内部信号产生 ADC 触发事件，例如复位、使能或比较事件。也可生成由内部边沿检测器发出的脉冲，例如：

- OC4ref 的上升沿和下降沿
- OC5ref 上的上升沿或 OC6ref 上的下降沿

在重定向到 ADC 的 TRGO2 内部线路上发出触发信号。共有 16 个可能的事件，它们可通过 TIMx_CR2 寄存器中的 MMS2[3:0] 位选择。

第 1923 页的图 374 给出了三相电机驱动的应用示例。

注：必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。

注：必须先使能 ADC 时钟，才能从主定时器接收事件；从定时器接收触发信号时，不得实时更改 ADC 时钟。

38.3.28 DMA 分组传送模式

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对定时器的一部分多次重新编程而无需软件开销，但也可以用于定期地一次性地批量读取定时器的多个寄存器的内容。

DMA 控制器访问目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（分组）。每次写入 TIMx_DMAR 寄存器都会重定向到定时器其中的一个寄存器。

TIMx_DCR 寄存器中的 DBL[4:0] 位设置 DMA 突发传输的长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次分组传送，即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA[4:0] 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量：

示例：

00000：TIMx_CR1

00001：TIMx_CR2

00010：TIMx_SMCR

例如，定时器 DMA 分组传送功能用于在发生更新事件后将 CCRx 寄存器（x = 2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

1. 将相应的 DMA 通道配置如下：
   - DMA 通道外设地址为 DMAR 寄存器地址。
   - DMA 通道内存地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
   - 要传输的数据量 = 3（参见下文注释）。
   - 禁止循环模式。
2. 通过将 DBA 和 DBL 位字段配置如下来配置 DCR 寄存器：
   DBL = 3 次传输，DBA = 0xE。
3. 使能 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 TIMx
5. 使能 DMA 通道

本例适用于每个 CCRx 寄存器只更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间，data1 传输到 CCR2，data2 传输到 CCR3，data3 传输到 CCR4；在第二个更新 DMA 请求期间，data4 传输到 CCR2，data5 传输到 CCR3，data6 传输到 CCR4。

注： *可以将空值写入保留的寄存器中。*

38.3.29 调试模式

当系统进入调试模式（处理器内核停止）时，TIMx 计数器会根据 DBG 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。

为了安全起见，当计数器停止时，输出被禁止（就像 MOE 位被复位一样）。可以将输出强制变为无效状态（OSSI 位 = 1），或者通过 GPIO 控制器（OSSI 位 = 0）来控制输出，通常将其强制为高阻态。

更多详细信息，请参见调试支持 (DBG) 部分。

38.4 TIM1/TIM8 寄存器

有关寄存器说明中使用的缩写，请参见相应列表。

外设寄存器可按半字（16 位）或字（32 位）访问。

38.4.1 TIMx 控制寄存器 1 (TIMx_CR1)（x = 1、8）

TIMx control register 1

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
				rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)

0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD[1:0]**：时钟分频 (Clock division)

该位字段指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器（ETR、TIx）所使用的采样时钟 (t_{DTS}) 之间的分频比：

00：$t_{DTS}=t_{CK_INT}$
01：$t_{DTS}=2*t_{CK_INT}$
10：$t_{DTS}=4*t_{CK_INT}$
11：保留，不要设置成此值

注意：$t_{DTS} = 1/f_{DTS}$，$t_{CK_INT} = 1/f_{CK_INT}$。

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)

0：TIMx_ARR 寄存器不进行缓冲
1：TIMx_ARR 寄存器进行缓冲

位 6:5 **CMS[1:0]**：中心对齐模式选择 (Center-aligned mode selection)

00：边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。
01：中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。
10：中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。
11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志都会置 1。

注意：只要计数器处于使能状态 (CEN=1)，就不得从边沿对齐模式切换为中心对齐模式。

位 4 **DIR**：方向

0：计数器递增计数
1：计数器递减计数

注意：当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。

位 3 **OPM**：单脉冲模式 (One pulse mode)

0：计数器在发生更新事件时不会停止计数

1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

位 2 **URS**：更新请求源 (Update request source)

该位由软件置 1 和清零，用以选择 UEV 事件源。

0：使能时，所有以下事件都会生成更新中断或 DMA 请求。此类事件包括：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。

位 1 **UDIS**：更新禁止 (Update disable)

该位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

带缓冲的寄存器被加载为预加载数值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注意：只有事先通过软件将 CEN 位置 1，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将 CEN 位置 1。

38.4.2 TIMx 控制寄存器 2 (TIMx_CR2)（x = 1、8）

TIMx control register 2

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS2[3:0]				Res.	OIS6	Res.	OIS5
								rw	rw	rw	rw		rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位 31:24 保留，必须保持复位值。

位 23:20 **MMS2[3:0]**：主模式选择 2 (Master mode selection 2)

这些位可选择将发送到 ADC 以实现同步的信息 (TRGO2)。这些位的组合如下：

0000： **复位**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO2)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO2 上的信号相比实际复位会有延迟。

0001： **使能**——计数器使能信号 CNT_EN 用作触发输出 (TRGO2)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑与运算组合而成。当计数器使能信号由触发输入控制时，TRGO2 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。

0010： **更新**——选择更新事件作为触发输出 (TRGO2)。例如，主定时器可用作从定时器的预分频器。

0011： **比较脉冲**——CC1IF 标志置 1 时（即使已为高），只要发生捕获或比较匹配，触发输出 (TRGO2) 都会发送一个正脉冲。

0100： **比较**——OC1REFC 信号用作触发输出 (TRGO2)

0101： **比较**——OC2REFC 信号用作触发输出 (TRGO2)

0110： **比较**——OC3REFC 信号用作触发输出 (TRGO2)

0111： **比较**——OC4REFC 信号用作触发输出 (TRGO2)

1000： **比较**——OC5REFC 信号用作触发输出 (TRGO2)

1001: **比较**——OC6REFC 信号用作触发输出 (TRGO2)

1010: **比较脉冲**——OC4REFC 上升沿或下降沿时，TRGO2 上生成脉冲

1011: **比较脉冲**——OC6REFC 上升沿或下降沿时，TRGO2 上生成脉冲

1100: **比较脉冲**——OC4REFC 或 OC6REFC 上升沿时，TRGO2 上生成脉冲

1101: **比较脉冲**——OC4REFC 上升沿或 OC6REFC 下降沿时，TRGO2 上生成脉冲

1110: **比较脉冲**——OC5REFC 或 OC6REFC 上升沿时，TRGO2 上生成脉冲

1111： **比较脉冲**——OC5REFC 上升沿或 OC6REFC 下降沿时，TRGO2 上生成脉冲

注意：必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

位 19 保留，必须保持复位值。

位 18 **OIS6**：输出空闲状态 6（OC6 输出）(Output Idle state 6 (OC6 output))
请参见 OIS1 位

位 17 保留，必须保持复位值。

位 16 **OIS5**：输出空闲状态 5（OC5 输出）(Output Idle state 5 (OC5 output))
请参见 OIS1 位

位 15 保留，必须保持复位值。

位 14 **OIS4**：输出空闲状态 4（OC4 输出）(Output Idle state 4 (OC4 output))
请参见 OIS1 位

位 13 **OIS3N**：输出空闲状态 3（OC3N 输出）(Output Idle state 3 (OC3N output))
请参见 OIS1N 位

位 12 **OIS3**：输出空闲状态 3（OC3 输出）(Output Idle state 3 (OC3 output))
请参见 OIS1 位

位 11 **OIS2N**：输出空闲状态 2（OC2N 输出）(Output Idle state 2 (OC2N output))
请参见 OIS1N 位

位 10 **OIS2**：输出空闲状态 2（OC2 输出）(Output Idle state 2 (OC2 output))
请参见 OIS1 位

位 9 **OIS1N**：输出空闲状态 1（OC1N 输出）(Output Idle state 1 (OC1N output))

0：当 MOE=0 时，经过死区时间后 OC1N=0

1：当 MOE=0 时，经过死区时间后 OC1N=1

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，该位即无法修改。

位 8 **OIS1**：输出空闲状态 1（OC1 输出）(Output Idle state 1 (OC1 output))

0：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0

1：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=1

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，该位即无法修改。

位 7 **TI1S**：TI1 选择 (TI1 selection)

0：TIMx_CH1 引脚连接到 TI1 输入

1：TIMx_CH1、CH2 和 CH3 引脚连接到 TI1 输入（异或组合）

位 6:4 **MMS[2:0]**：主模式选择 (Master mode selection)

这些位允许在主模式下将选定的信息发送到从定时器以实现同步 (TRGO)。这些位的组合如下：

000： **复位**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。

001： **使能**——计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑与运算组合而成。当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。

010： **更新**——选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器。

011： **比较脉冲**——CC1IF 标志置 1 时（即使已为高电平），只要发生捕获或比较匹配，触发输出都会发送一个正脉冲。(TRGO)。

100： **比较**——OC1REFC 信号用作触发输出 (TRGO)

101： **比较**——OC2REFC 信号用作触发输出 (TRGO)

110： **比较**——OC3REFC 信号用作触发输出 (TRGO)

111： **比较**——OC4REFC 信号用作触发输出 (TRGO)

注意：必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

位 3 **CCDS**：捕获/比较 DMA 选择 (Capture/compare DMA selection)

0：发生 CCx 事件时发送 CCx DMA 请求

1：发生更新事件时发送 CCx DMA 请求

位 2 **CCUS**：捕获/比较控制更新选择 (Capture/compare control update selection)

0：如果捕获/比较控制位 (CCPC=1) 进行预装载，仅通过将 COMG 位置 1 来对这些位进行更新

1：如果捕获/比较控制位 (CCPC=1) 进行预装载，可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新。

注意：该位仅对具有互补输出的通道有效。

位 1 保留，必须保持复位值。

位 0 **CCPC**：捕获/比较预装载控制 (Capture/compare preloaded control)

0：CCxE、CCxNE 和 OCxM 位未进行预装载

1：CCxE、CCxNE 和 OCxM 位进行了预装载，写入这些位后，仅当发生换向事件 (COM)（COMG 位置 1 或在 TRGI 上检测到上升沿，取决于 CCUS 位）时才会对这些位进行更新。

注意：该位仅对具有互补输出的通道有效。

38.4.3 TIMx 从模式控制寄存器 (TIMx_SMCR)（x = 1、8）

TIMx slave mode control register

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS[4:3]		Res.	Res.	Res.	SMS[3]
										rw	rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

位 31:22 保留，必须保持复位值。

位 19:17 保留，必须保持复位值。

位 15 **ETP**：外部触发极性 (External trigger polarity)

该位可选择将 ETR 还是 $\overline{\text{ETR}}$ 用于触发操作

0：ETR 未反相，高电平或上升沿有效。

1：ETR 反相，低电平或下降沿有效。

位 14 **ECE**：外部时钟使能 (External clock enable)

该位可使能外部时钟模式 2。

0：禁止外部时钟模式 2

1：使能外部时钟模式 2。计数器时钟由 ETRF 信号的任意有效边沿提供。

注意：将 ECE 位置 1 与选择外部时钟模式 1 并将 TRGI 连接到 ETRF（SMS=111 且 TS=00111）具有相同效果。

外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF（TS 位不得为 00111）。

如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 ETRF。

位 13:12 **ETPS[1:0]**：外部触发预分频器 (External trigger prescaler)

外部触发信号 ETRP 频率不得超过 f_{CK_INT} 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。

00：预分频器关闭

01：2 分频 ETRP 频率

10：4 分频 ETRP 频率

11：8 分频 ETRP 频率

位 11:8 **ETF[3:0]**：外部触发滤波器 (External trigger filter)

该位字段可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 7 **MSM**：主/从模式 (Master/slave mode)

0：无操作
1：当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于由单个外部事件对多个定时器进行同步的情况。

位 21、20、6、5、4 **TS[4:0]**：触发选择

该位字段可选择将要用于同步计数器的触发输入。

00000：内部触发 0 (ITR0)
00001：内部触发 1 (ITR1)
00010：内部触发 2 (ITR2)
00011：内部触发 3 (ITR3)
00100：TI1 边沿检测器 (TI1F_ED)
00101：滤波后的定时器输入 1 (TI1FP1)
00110：滤波后的定时器输入 2 (TI2FP2)
00111：外部触发输入 (ETRF)
其他值：保留

有关各定时器 ITRx 含义的详细信息，请参见*第 1952 页的表 350：TIMx 内部触发连接*。

注意：这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。

位 3 保留，必须保持复位值。

位 16、2、1、0 **SMS[3:0]**：从模式选择 (Slave mode selection)

选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。

0000：禁止从模式——如果 CEN = “1”，预分频器时钟直接由内部时钟提供。

0001：编码器模式 1——计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。

0010：编码器模式 2——计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。

0011：编码器模式 3——计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个信号的电平。

0100：复位模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。

0101：门控模式——触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。

0110：触发模式——触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。

0111：外部时钟模式 1——由所选触发信号 (TRGI) 的上升沿提供计数器时钟。

1000：组合复位 + 触发模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。

1000 以上的代码：保留。

注意：如果将 TI1F_ED 选作触发输入 (TS=00100)，则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。

注意：必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。

表 350. TIMx 内部触发连接

从 TIM	ITR0 (TS = 00000)	ITR1 (TS = 00001)	ITR2 (TS = 00010)	ITR3 (TS = 00011)
TIM1	TIM15	TIM2	TIM3	TIM4
TIM8	TIM1	TIM2	TIM4	TIM5

38.4.4 TIMx DMA/中断使能寄存器 (TIMx_DIER)（x = 1、8）

TIMx DMA/interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15 保留，必须保持复位值。

位 14 **TDE**：触发信号 DMA 请求使能 (Trigger DMA request enable)

0：禁止触发信号 DMA 请求

1：使能触发信号 DMA 请求

位 13 **COMDE**：COM DMA 请求使能 (COM DMA request enable)

0：禁止 COM DMA 请求

1：使能 COM DMA 请求

位 12 **CC4DE**：捕获/比较 4 DMA 请求使能 (Capture/Compare 4 DMA request enable)
0：禁止 CC4 DMA 请求
1：使能 CC4 DMA 请求

位 11 **CC3DE**：捕获/比较 3 DMA 请求使能 (Capture/Compare 3 DMA request enable)
0：禁止 CC3 DMA 请求
1：使能 CC3 DMA 请求

位 10 **CC2DE**：捕获/比较 2 DMA 请求使能 (Capture/Compare 2 DMA request enable)
0：禁止 CC2 DMA 请求
1：使能 CC2 DMA 请求

位 9 **CC1DE**：捕获/比较 1 DMA 请求使能 (Capture/Compare 1 DMA request enable)
0：禁止 CC1 DMA 请求
1：使能 CC1 DMA 请求

位 8 **UDE**：更新 DMA 请求使能 (Update DMA request enable)
0：禁止更新 DMA 请求
1：使能更新 DMA 请求

位 7 **BIE**：刹车中断使能 (Break interrupt enable)
0：禁止刹车中断
1：使能刹车中断

位 6 **TIE**：触发信号中断使能 (Trigger interrupt enable)
0：禁止触发信号中断
1：使能触发信号中断

位 5 **COMIE**：COM 中断使能 (COM interrupt enable)
0：禁止 COM 中断
1：使能 COM 中断

位 4 **CC4IE**：捕获/比较 4 中断使能 (Capture/Compare 4 interrupt enable)
0：禁止 CC4 中断
1：使能 CC4 中断

位 3 **CC3IE**：捕获/比较 3 中断使能 (Capture/Compare 3 interrupt enable)
0：禁止 CC3 中断
1：使能 CC3 中断

位 2 **CC2IE**：捕获/比较 2 中断使能 (Capture/Compare 2 interrupt enable)
0：禁止 CC2 中断
1：使能 CC2 中断

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断
1：使能 CC1 中断

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断
1：使能更新中断

38.4.5 TIMx 状态寄存器 (TIMx_SR)（x = 1、8）

TIMx status register

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6IF	CC5IF
														rc_w0	rc_w0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位 31:18 保留，必须保持复位值。

位 17 **CC6IF**：比较 6 中断标志 (Compare 6 interrupt flag)
请参见 CC1IF 说明（注：通道 5 和通道 6 只能配置为输出）

位 16 **CC5IF**：比较 5 中断标志 (Compare 5 interrupt flag)
请参见 CC1IF 说明（注：通道 5 只能配置为输出）

位 15:14 保留，必须保持复位值。

位 13 **SBIF**：系统刹车中断标志 (System Break interrupt flag)
只要系统刹车输入变为有效状态，此标志便由硬件置 1。系统刹车输入无效后可通过软件对其清零。
此标志必须复位以使 PWM 重新开始工作。
0：未发生刹车事件。
1：在系统刹车输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。

位 12 **CC4OF**：捕获/比较 4 过捕获标志 (Capture/Compare 4 overcapture flag)
请参见 CC1OF 说明

位 11 **CC3OF**：捕获/比较 3 过捕获标志 (Capture/Compare 3 overcapture flag)
请参见 CC1OF 说明

位 10 **CC2OF**：捕获/比较 2 过捕获标志 (Capture/compare 2 overcapture flag)
请参见 CC1OF 说明

位 9 **CC1OF**：捕获/比较 1 过捕获标志 (Capture/Compare 1 overcapture flag)
仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。
0：未检测到过捕获。
1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8 **B2IF**：刹车 2 中断标志 (Break 2 interrupt flag)
只要刹车 2 输入变为有效状态，此标志便由硬件置 1。刹车 2 输入无效后可通过软件对其清零。
0：未发生刹车事件。
1：在刹车 2 输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。

位 7 **BIF**：刹车中断标志 (Break interrupt flag)
只要刹车输入变为有效状态，此标志便由硬件置 1。刹车输入无效后可通过软件对其清零。
0：未发生刹车事件。
1：在刹车输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。

位 6 **TIF**：触发中断标志 (Trigger interrupt flag)

在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到 TRG 触发事件有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。

0：未发生触发事件。

1：触发中断挂起。

位 5 **COMIF**：COM 中断标志 (COM interrupt flag)

此标志在发生 COM 事件时（捕获/比较控制位 CCxE、CCxNE 和 OCxM 已更新时）由硬件置 1。但需要通过软件清零。

0：未发生 COM 事件。

1：COM 中断挂起。

位 4 **CC4IF**：捕获/比较 4 中断标志 (Capture/Compare 4 interrupt flag)

请参见 CC1IF 说明

位 3 **CC3IF**：捕获/比较 3 中断标志 (Capture/Compare 3 interrupt flag)

请参见 CC1IF 说明

位 2 **CC2IF**：捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag)

请参见 CC1IF 说明

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/Compare 1 interrupt flag)

该标志由硬件置 1，但需要通过软件清零（输入捕获或输出比较模式）或通过读取 TIMx_CCR1 寄存器清零（仅限输入捕获模式）。

0：未发生比较匹配/输入捕获

1：发生了比较匹配/输入捕获

如果通道 CC1 配置为输出：当计数器 TIMx_CNT 的值与 TIMx_CCR1 的值匹配时，此标志置 1。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。在中心对齐模式下，可通过 3 种方式将此标志置 1，完整说明请参见 TIMx_CR1 寄存器中的 CMS 位。

如果通道 CC1 配置为输入：当 TIMx_CCR1 寄存器中捕获到计数器值时（如果通过设置 TIMx_CCER 中的 CC1P 和 CC1NP 位定义为边沿有效，则改为在 IC1 上检测到边沿时），该位置 1。

位 0 **UIF**：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

– TIMx_CR1 寄存器中的 UDIS=0，并且重复计数器值上溢或下溢时（重复计数器 = 0 时更新）。

– TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

– TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（请参见 *第 38.4.3 节：TIMx 从模式控制寄存器 (TIMx_SMCR)（x = 1、8）*）。

38.4.6 TIMx 事件生成寄存器 (TIMx_EGR)（x = 1、8）

TIMx event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
							w	w	w	w	w	w	w	w	w

位 15:9 保留，必须保持复位值。

位 8 **B2G**：刹车 2 生成 (Break 2 generation)
该位由软件置 1 以生成事件，并由硬件自动清零。
0：无操作
1：生成刹车 2 事件。MOE 位清零且 B2IF 标志置 1。使能后可发生相关中断。

位 7 **BG**：刹车生成 (Break generation)
该位由软件置 1 以生成事件，并由硬件自动清零。
0：无操作
1：生成刹车事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 6 **TG**：触发生成 (Trigger generation)
该位由软件置 1 以生成事件，并由硬件自动清零。
0：无操作
1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 5 **COMG**：捕获/比较控制更新生成 (Capture/Compare control update generation)
该位可通过软件置 1，并由硬件自动清零
0：无操作
1：CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位。
注意：该位仅对具有互补输出的通道有效。

位 4 **CC4G**：捕获/比较 4 生成 (Capture/Compare 4 generation)
请参见 CC1G 说明

位 3 **CC3G**：捕获/比较 3 生成 (Capture/Compare 3 generation)
请参见 CC1G 说明

位 2 **CC2G**：捕获/比较 2 生成 (Capture/Compare 2 generation)
请参见 CC1G 说明

位 1 **CC1G**：捕获/比较 1 生成 (Capture/Compare 1 generation)
该位由软件置 1 以生成事件，并由硬件自动清零。
0：无操作
1：通道 1 上生成捕获/比较事件：
如果通道 CC1 配置为输出：
使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。
如果通道 CC1 配置为输入：
TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)

该位可通过软件置 1，并由硬件自动清零。

0：无操作

1：重新初始化计数器并生成一个寄存器更新事件。预分频器内部计数器也将清零（预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值 (TIMx_ARR)。

38.4.7 TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1)（x = 1、8）

TIMx capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获模式和输出比较模式下的功能均不同。可独立组合两种模式（例如，输入捕获模式下的通道 1 和输出比较模式下的通道 2）。

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:12 **IC2F[3:0]**：输入捕获 2 滤波器 (Input capture 2 filter)

请参见 IC1F[3:0] 说明。

位 11:10 **IC2PSC[1:0]**：输入捕获 2 预分频器 (Input capture 2 prescaler)

请参见 IC1PSC[1:0] 说明。

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/Compare 2 selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出

01：CC2 通道配置为输入，IC2 映射到 TI2 上

10：CC2 通道配置为输入，IC2 映射到 TI1 上

11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据。

位 7:4 **IC1F[3:0]**：输入捕获 1 滤波器 (Input capture 1 filter)

该位字段可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC[1:0]**：输入捕获 1 预分频器 (Input capture 1 prescaler)

该位字段定义 CC1 输入 (IC1) 的预分频比。只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
10：CC1 通道配置为输入，IC1 映射到 TI2 上
11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

38.4.8 TIMx 捕获/比较模式寄存器 1 [复用] (TIMx_CCMR1)（x = 1、8）

TIMx capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获模式和输出比较模式下的功能均不同。可独立组合两种模式（例如，输入捕获模式下的通道 1 和输出比较模式下的通道 2）。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2 CE	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		OC1 CE	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 23:17 保留，必须保持复位值。

位 15 **OC2CE**：输出比较 2 清零使能 (Output Compare 2 clear enable)
请参见 OC1CE 说明。

位 24、14:12 **OC2M[3:0]**：输出比较 2 模式 (Output Compare 2 mode)
请参见 OC1M[3:0] 说明。

位 11 **OC2PE**：输出比较 2 预装载使能 (Output Compare 2 preload enable)
请参见 OC1PE 说明。

位 10 **OC2FE**：输出比较 2 快速使能 (Output Compare 2 fast enable)
请参见 OC1FE 说明。

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/Compare 2 selection)
该位字段定义通道方向（输入/输出）以及所使用的输入。
00：CC2 通道配置为输出
01：CC2 通道配置为输入，IC2 映射到 TI2 上
10：CC2 通道配置为输入，IC2 映射到 TI1 上
11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（TIMx_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据。

位 7 **OC1CE**：输出比较 1 清零使能 (Output Compare 1 clear enable)
0：OC1Ref 不受 ETRF 输入影响
1：ETRF 输入上检测到高电平时，OC1Ref 立即清零。

位 16，6:4 **OC1M[3:0]**：输出比较 1 模式 (Output Compare 1 mode)

这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。当定时器用作软件时基时，可以使用该模式。在定时器操作期间使能冻结模式时，输出保持进入冻结状态前的状态（有效或无效）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便输出有效电平，否则输出无效电平。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便输出无效电平（OC1REF=“0”），否则输出有效电平（OC1REF=“1”）。

0111：PWM 模式 2——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便输出无效电平，否则输出有效电平。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便输出有效电平，否则输出无效电平。

1000：可再触发 OPM 模式 1——在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。

1001: 可再触发 OPM 模式 2——在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。

1010: 保留。

1011: 保留。

1100: 组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。

1101: 组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。

1110: 不对称 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

1111：不对称 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。

注意：在 PWM 模式下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。

注意：该位字段将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成 COM 事件时，OC1M 有效位才会从预装载位获取新值。

注意：OC1M[3] 位不是连续的，而是在位 16 中。

位 3 **OC1PE**：输出比较 1 预装载使能 (Output Compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到影子寄存器中。

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。

位 2 **OC1FE**：输出比较 1 快速使能 (Output Compare 1 fast enable)

该位可降低触发事件与定时器输出跳变之间的延迟。单脉冲模式下必须使用（TIMx_CR1 寄存器中的 OPM 位置 1），以便在启动触发后快速发送输出脉冲。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

10：CC1 通道配置为输入，IC1 映射到 TI2 上

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

38.4.9 TIMx 捕获/比较模式寄存器 2 (TIMx_CCMR2)（x = 1、8）

TIMx capture/compare mode register 2

偏移地址：0x1C

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获模式和输出比较模式下的功能均不同。可独立组合两种模式（例如，输入捕获模式下的通道 1 和输出比较模式下的通道 2）。

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F[3:0]				IC4PSC[1:0]		CC4S[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:12 **IC4F[3:0]**：输入捕获 4 滤波器 (Input capture 4 filter)

请参见 IC1F[3:0] 说明。

位 11:10 **IC4PSC[1:0]**：输入捕获 4 预分频器 (Input capture 4 prescaler)

请参见 IC1PSC[1:0] 说明。

位 9:8 **CC4S[1:0]**：捕获/比较 4 选择 (Capture/Compare 4 selection)
该位字段定义通道方向（输入/输出）以及所使用的输入。
00：CC4 通道配置为输出
01：CC4 通道配置为输入，IC4 映射到 TI4 上
10：CC4 通道配置为输入，IC4 映射到 TI3 上
11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（TIMx_CCER 中的 CC4E = "0"），才可向 CC4S 位写入数据。

位 7:4 **IC3F[3:0]**：输入捕获 3 滤波器 (Input capture 3 filter)
请参见 IC1F[3:0] 说明。

位 3:2 **IC3PSC[1:0]**：输入捕获 3 预分频器 (Input capture 3 prescaler)
请参见 IC1PSC[1:0] 说明。

位 1:0 **CC3S[1:0]**：捕获/比较 3 选择 (Capture/compare 3 selection)
该位字段定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（TIMx_CCER 中的 CC3E = "0"），才可向 CC3S 位写入数据。

38.4.10 TIMx 捕获/比较模式寄存器 2 [复用] (TIMx_CCMR2)（x = 1、8）

TIMx capture/compare mode register 2

偏移地址：0x1C

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获模式和输出比较模式下的功能均不同。可独立组合两种模式（例如，输入捕获模式下的通道 1 和输出比较模式下的通道 2）。

输出比较模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4 CE	OC4M[2:0]			OC4 PE	OC4 FE	CC4S[1:0]		OC3 CE	OC3M[2:0]			OC3 PE	OC3 FE	CC3S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 23:17 保留，必须保持复位值。

位 15 **OC4CE**：输出比较 4 清零使能 (Output compare 4 clear enable)
请参见 OC1CE 说明。

位 24、14:12 **OC4M[3:0]**：输出比较 4 模式 (Output compare 4 mode)
请参见 OC3M[3:0] 说明。

位 11 **OC4PE**：输出比较 4 预装载使能 (Output compare 4 preload enable)
请参见 OC1PE 说明。

位 10 **OC4FE**：输出比较 4 快速使能 (Output compare 4 fast enable)
请参见 OC1FE 说明。

位 9:8 **CC4S[1:0]**：捕获/比较 4 选择 (Capture/Compare 4 selection)
该位字段定义通道方向（输入/输出）以及所使用的输入。
00：CC4 通道配置为输出
01：CC4 通道配置为输入，IC4 映射到 TI4 上
10：CC4 通道配置为输入，IC4 映射到 TI3 上
11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（TIMx_CCER 中的 CC4E = “0”），才可向 CC4S 位写入数据。

位 7 **OC3CE**：输出比较 3 清零使能 (Output compare 3 clear enable)
请参见 OC1CE 说明。

位 16，6:4 **OC3M[3:0]**：输出比较 3 模式 (Output compare 3 mode)
请参见 OC1M[3:0] 说明。

位 3 **OC3PE**：输出比较 3 预装载使能 (Output compare 3 preload enable)
请参见 OC1PE 说明。

位 2 **OC3FE**：输出比较 3 快速使能 (Output compare 3 fast enable)
请参见 OC1FE 说明。

位 1:0 **CC3S[1:0]**：捕获/比较 3 选择 (Capture/Compare 3 selection)
该位字段定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（TIMx_CCER 中的 CC3E = “0”），才可向 CC3S 位写入数据。

38.4.11 TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 1、8）

TIMx capture/compare enable register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E
										rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21 **CC6P**：捕获/比较 6 输出极性 (Capture/Compare 6 output polarity)
请参见 CC1P 说明

位 20 **CC6E**：捕获/比较 6 输出使能 (Capture/Compare 6 output enable)
请参见 CC1E 说明

位 19:18 保留，必须保持复位值。

位 17 **CC5P**：捕获/比较 5 输出极性 (Capture/Compare 5 output polarity)
请参见 CC1P 说明

位 16 **CC5E**：捕获/比较 5 输出使能 (Capture/Compare 5 output enable)
请参见 CC1E 说明

位 15 **CC4NP**：捕获/比较 4 互补输出极性 (Capture/Compare 4 complementary output polarity)
请参见 CC1NP 说明

位 14 保留，必须保持复位值。

位 13 **CC4P**：捕获/比较 4 输出极性 (Capture/Compare 4 output polarity)
请参见 CC1P 说明

位 12 **CC4E**：捕获/比较 4 输出使能 (Capture/Compare 4 output enable)
请参见 CC1E 说明

位 11 **CC3NP**：捕获/比较 3 互补输出极性 (Capture/Compare 3 complementary output polarity)
请参见 CC1NP 说明

位 10 **CC3NE**：捕获/比较 3 互补输出使能 (Capture/Compare 3 complementary output enable)
请参见 CC1NE 说明

位 9 **CC3P**：捕获/比较 3 输出极性 (Capture/Compare 3 output polarity)。
请参见 CC1P 说明

位 8 **CC3E**：捕获/比较 3 输出使能 (Capture/Compare 3 output enable)
请参见 CC1E 说明

位 7 **CC2NP**：捕获/比较 2 互补输出极性 (Capture/Compare 2 complementary output polarity)
请参见 CC1NP 说明

位 6 **CC2NE**：捕获/比较 2 互补输出使能 (Capture/Compare 2 complementary output enable)
请参见 CC1NE 说明

位 5 **CC2P**：捕获/比较 2 输出极性 (Capture/Compare 2 output polarity)
请参见 CC1P 说明

位 4 **CC2E**：捕获/比较 2 输出使能 (Capture/Compare 2 output enable)
请参见 CC1E 说明

位 3 **CC1NP**：捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output polarity)
CC1 通道配置为输出：
0：OC1N 高电平有效。
1：OC1N 低电平有效。
CC1 通道配置为输入：
该位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1P 说明。
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3 且 CC1S="00"（通道配置为输出），该位立即变为不可写状态。
该位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NP 有效位才会从预装载位获取新值。

位 2 **CC1NE**：捕获/比较 1 互补输出使能 (Capture/Compare 1 complementary output enable)
0：关闭——OC1N 未激活。OC1N 电平与 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位存在函数关系。
1：开启——基于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值，在相应输出引脚上输出 OC1N 信号。
该位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NE 有效位才会从预装载位获取新值。

位 1 **CC1P**：捕获/比较 1 输出极性 (Capture/Compare 1 output polarity)
0：OC1 高电平有效（输出模式）/边沿有效性选择（输入模式，见下文）
1：OC1 低电平有效（输出模式）/边沿有效性选择（输入模式，见下文）
CC1 通道配置为输入时，CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。
CC1NP=0，CC1P=0：非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。
CC1NP=0，CC1P=1：反相/下降沿触发。电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。
CC1NP=1，CC1P=1：非反相/上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。
CC1NP=1，CC1P=0：该配置保留，不得使用。
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3，该位立即变为不可写状态。
该位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。

位 0 **CC1E**：捕获/比较 1 输出使能 (Capture/Compare 1 output enable)
0：禁止捕获模式/OC1 无效（见下文）
1：使能捕获模式/在相应输出引脚上输出 OC1 信号
当 CC1 通道配置为输出时，OC1 电平取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位，与 CC1E 位的状态无关。有关详细信息，请参见*表 351*。
注意：该位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1E 有效位才会从预装载位获取新值。

表 351. 具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	X	0	0	禁止输出（不由定时器驱动：高阻态） OCx=0、OCxN=0	
		0	0	1	禁止输出（不由定时器驱动：高阻态） OCx=0	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		0	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	禁止输出（不由定时器驱动：高阻态） OCxN=0
		X	1	1	OCREF + 极性 + 死区	OCREF 互补项（对 OCREF 进行“非”运算）+ 极性 + 死区
		1	0	1	关闭状态（输出为无效状态） OCx=CCxP	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		1	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	关闭状态（输出为无效状态） OCxN=CCxNP
0	0	X	X	X	禁止输出（不由定时器驱动：高阻态）。	
	1		0	0		
			0	1	关闭状态（输出为无效状态） 异步：OCx=CCxP、OCxN=CCxNP（如果触发 BRK 或 BRK2）。 随后（仅当触发 BRK 时才有效），如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，假定 OISx 和 OISxN 并没有都设置成 OCx 及 OCxN 的有效电平（否则在半桥配置下驱动开关时可能导致短路）。 **注意**：BRK2 只能在 OSSI = OSSR = 1 时使用。	
			1	0		
			1	1		

1. 如果一个通道的两个输出均未被使用（由 GPIO 接管控制），则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

注： *与互补通道 OCx 和 OCxN 相连的外部 I/O 引脚的状态取决于通道 OCx 和 OCxN 的状态以及 GPIO 寄存器。*

38.4.12 TIMx 计数器 (TIMx_CNT)（x = 1、8）

TIMx counter

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **UIFCPY**：UIF 副本 (UIF copy)

该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。如果 TIMxCR1 中的 UIFREMAP 位复位，则位 31 保留，读数 0。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

38.4.13 TIMx 预分频器 (TIMx_PSC)（x = 1、8）

TIMx prescaler

偏移地址：0x28

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)

计数器时钟频率 (CK_CNT) 等于 f_{CK_PSC} / (PSC[15:0] + 1)。

PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到实际预分频器寄存器的值。

38.4.14 TIMx 自动重载寄存器 (TIMx_ARR)（x = 1、8）

TIMx auto-reload register

偏移地址：0x2C

复位值：0xFFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)

ARR 为要装载到实际自动重载寄存器的值。

有关 ARR 更新和行为的更多详细信息，请参见*第 1893 页的第 38.3.1 节：时基单元*。

当自动重载值为空时，计数器不工作。

38.4.15 TIMx 重复计数器寄存器 (TIMx_RCR)（x = 1、8）

TIMx repetition counter register

偏移地址：0x30

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REP[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **REP[15:0]**：重复计数器值 (Repetition counter value)

使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即，从预装载寄存器向影子寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。

与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP 值，因此在生成下一重复更新事件之前，向 TIMx_RCR 寄存器写入任何值都无影响。

这意味着 PWM 模式下 (REP+1) 相当于：

边沿对齐模式下的 PWM 周期数

中心对齐模式下的 PWM 半周期数。

38.4.16 TIMx 捕获/比较寄存器 1 (TIMx_CCR1)（x = 1、8）

TIMx capture/compare register 1

偏移地址：0x34

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR1[15:0]**：捕获/比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：CCR1 是捕获/比较寄存器 1 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 1）。

实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出相应电平的值。

如果通道 CC1 配置为输入：CR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。只能读取 TIMx_CCR1 寄存器，无法对其进行编程。

38.4.17 TIMx 捕获/比较寄存器 2 (TIMx_CCR2)（x = 1、8）

TIMx capture/compare register 2

偏移地址：0x38

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR2[15:0]**：捕获/比较 2 值 (Capture/Compare 2 value)

如果通道 CC2 配置为输出：CCR2 是捕获/比较寄存器 2 的预装载值。
如果没有通过 TIMx_CCMR1 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 2）。
有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出相应电平的值。
如果通道 CC2 配置为输入：CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。只能读取 TIMx_CCR2 寄存器，无法对其进行编程。

38.4.18 TIMx 捕获/比较模式寄存器 3 (TIMx_CCR3)（x = 1、8）

TIMx capture/compare register 3

偏移地址：0x3C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR3[15:0]**：捕获/比较值 (Capture/Compare value)

如果通道 CC3 配置为输出：CCR3 是捕获/比较寄存器 3 的预装载值。
如果没有通过 TIMx_CCMR2 寄存器中的 OC3PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 3）。
有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC3 输出上发出信号的值。

如果通道 CC3 配置为输入：CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。只能读取 TIMx_CCR3 寄存器，无法对其进行编程。

38.4.19 TIMx 捕获/比较寄存器 4 (TIMx_CCR4)（x = 1、8）

TIMx capture/compare register 4

偏移地址：0x40

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR4[15:0]**：捕获/比较值 (Capture/Compare value)

如果通道 CC4 配置为输出：CCR4 是捕获/比较寄存器 4 的预装载值。

如果没有通过 TIMx_CCMR2 寄存器中的 OC4PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 4）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC4 输出上发出信号的值。

如果通道 CC4 配置为输入：CCR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值。只能读取 TIMx_CCR4 寄存器，无法对其进行编程。

38.4.20 TIMx 刹车和死区寄存器 (TIMx_BDTR)（x = 1、8）

TIMx break and dead-time register

偏移地址：0x44

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	BK2BID	BKBID	BK2 DSRM	BK DSRM	BK2P	BK2E	BK2F[3:0]				BKF[3:0]			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注： *由于可以根据 LOCK 配置锁定位 BK2BID、BKBID、BK2DSRM、BKDSRM、BK2P、BK2E、BK2F[3:0]、BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0] 的写操作，因此必须在第一次对 TIMx_BDTR 寄存器执行写访问时对这些位进行配置。*

位 31:30 保留，必须保持复位值。

位 29 **BK2BID**：刹车 2 双向 (Break2 bidirectional)

请参见 BKBID 说明

位 28 **BKBID**：双向刹车 (Break bidirectional)

0：刹车输入 BRK 为输入模式

1：刹车输入 BRK 为双向模式

在双向模式下（BKBID 位置 1），刹车输入配置为输入模式和开漏输出模式。任何激活的刹车事件都将使刹车输入上呈逻辑低电平，以向外部设备指示发生了内部刹车事件。

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 27 **BK2DSRM**：刹车 2 解除 (Break2 Disarm)
请参见 BKDSRM 说明

位 26 **BKDSRM**：刹车解除 (Break Disarm)
0：启动刹车输入 BRK
1：解除刹车输入 BRK
当没有刹车源被激活时，该位由硬件清零。
必须通过软件将 BKDSRM 位置 1 以释放双向输出控制（开漏输出处于高阻态），然后不断轮询该位，直到其由硬件复位，指示故障条件已消失。
注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 25 **BK2P**：刹车 2 极性 (Break2 polarity)
0：刹车输入 BRK2 为低电平有效
1：刹车输入 BRK2 为高电平有效
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。
注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 24 **BK2E**：刹车 2 使能 (Break2 enable)
该位可使能完整的刹车 2 保护（包括连接到 bk_acth 的所有源和相应的 BKIN 源，如*图 378：刹车事件和刹车 2 电路概述*所示）。
0：禁止刹车 2 功能
1：使能刹车 2 功能
注意：BKIN2 必须只在 OSSR = OSSI = 1 时使用。
注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。
注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 23:20 **BK2F[3:0]**：刹车 2 滤波器 (Break 2 filter)
该位字段可定义 BRK2 输入的采样频率和适用于 BRK2 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：
0000：无滤波器，BRK2 异步工作
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8
注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

位 19:16 **BKF[3:0]**：刹车滤波器 (Break filter)

该位字段可定义 BRK 输入的采样频率和适用于 BRK 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，BRK 异步工作
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

位 15 **MOE**：主输出使能 (Main output enable)

只要刹车输入（BRK 或 BRK2）为有效状态，该位便由硬件异步清零。该位由软件置 1，也可根据 AOE 位状态自动置 1。该位仅对配置为输出的通道有效。

0：响应刹车事件（2 个）。禁止 OC 和 OCN 输出
响应刹车事件或向 MOE 写入 0 时：OC 和 OCN 输出被禁止或被强制为空闲状态，具体取决于 OSSI 位。

1：如果 OC 和 OCN 输出的相应使能位（TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位）均置 1，则使能 OC 和 OCN 输出。

有关详细信息，请参见 OC/OCN 使能说明（*第 38.4.11 节：TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 1、8）*）。

位 14 **AOE**：自动输出使能 (Automatic output enable)

0：MOE 只能由软件置 1
1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1（如果刹车输入 BRK 和 BRK2 均无效）

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 13 **BKP**：刹车极性 (Break polarity)

0：刹车输入 BRK 为低电平有效
1：刹车输入 BRK 为高电平有效

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。
注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 12 **BKE**：刹车使能 (Break enable)

该位可使能完整的刹车保护（包括连接到 bk_acth 的所有源和相应的 BRK 源，如*图 378：刹车事件和刹车 2 电路概述*所示）。

0：禁止刹车功能
1：使能刹车功能

注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。
注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 11 **OSSR**：运行模式下的关闭状态选择 (Off-state selection for Run mode)

该位在 MOE=1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。

有关详细信息，请参见 OC/OCN 使能说明（*第 38.4.11 节：TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 1、8）*）。

0：处于无效状态时，禁止 OC/OCN 输出（输出控制器呈高阻态，相应输出由 GPIO 接管）。

1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平（输出仍由定时器控制）。

注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，该位即无法修改。

位 10 **OSSI**：空闲模式下的关闭状态选择 (Off-state selection for Idle mode)

当由于刹车事件或软件写操作而使 MOE=0 时，该位作用于配置为输出的通道。

有关详细信息，请参见 OC/OCN 使能说明（*第 38.4.11 节：TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 1、8）*）。

0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 GPIO 逻辑接管）。

1：处于无效状态时，首先将 OC/OCN 输出强制为其无效电平，然后在死区后将其强制为空闲电平。定时器始终控制输出。

注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，该位即无法修改。

位 9:8 **LOCK[1:0]**：锁定配置 (Lock configuration)

这些位用于针对软件错误提供写保护。

00：关闭锁定——不对任何位提供写保护。

01：锁定级别 1，此时无法对 TIMx_BDTR 寄存器中的 DTG 位、TIMx_CR2 寄存器中的 OISx 和 OISxN 位以及 TIMx_BDTR 寄存器中的 B*K2BID*、*BKBID*、*BK2DSRM*、*BKDSRM*、*BK2P*、*BK2E*、*BK2F[3:0]*、*BKF[3:0]*、*AOE*、*BKP*、*BKE*、*OSSI*、*OSSR* 和 *DTG[7:0]* 位执行写操作。

10：锁定级别 2，此时无法对锁定级别 1 中适用的各位、CC 极性位（TIMx_CCER 寄存器中的 CCxP/CCxNP 位，只要通过 CCxS 位将相关通道配置为输出）以及 OSSR 和 OSSI 位执行写操作。

11：锁定级别 3，此时无法对锁定级别 2 中适用的各位、CC 控制位（TIMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。

注意：复位后只能对 LOCK 位执行一次写操作。对 TIMx_BDTR 寄存器执行写操作后其中的内容将冻结，直到下一次复位。

位 7:0 **DTG[7:0]**：配置死区发生器 (Dead-time generator setup)

该位字段定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。

DTG[7:5] = 0xx => DT = DTG[7:0] x t_{DTG}，其中 $t_{DTG} = t_{DTS}$。

DTG[7:5] = 10x => DT = (64 + DTG[5:0]) x t_{DTG}，其中 $t_{DTG} = 2 \times t_{DTS}$。

DTG[7:5] = 110 => DT = (32 + DTG[4:0]) x t_{DTG}，其中 $t_{DTG} = 8 \times t_{DTS}$。

DTG[7:5] = 111 => DT = (32 + DTG[4:0]) x t_{DTG}，其中 $t_{DTG} = 16 \times t_{DTS}$。

示例：如果 t_{DTS} = 125 ns (8 MHz)，则可能的死区值为：

0 到 15875 ns（步长为 125 ns），

16 us 到 31750 ns（步长为 250 ns），

32 µs 到 63 µs（步长为 1 µs），

64 us 到 126 us（步长为 2 us）

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，该位字段即无法修改。

38.4.21 TIMxDMA 控制寄存器 (TIMx_DCR)（x = 1、8）

TIMx DMA control register

偏移地址：0x48

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 15:13 保留，必须保持复位值。

位 12:8 **DBL[4:0]**：DMA 分组传送长度 (DMA burst length)

该 5 位向量定义了 DMA 的传送长度（当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次分组传送），即传送次数。可按半字或字节进行传送（请参见下面的示例）。

00000：1 次传输
00001：2 次传输
00010：3 次传输
...
10001：18 次传输

示例：考虑以下传输：DBL = 7 字节，DBA = TIMx_CR1。

– 如果 DBL = 7 字节且 DBA = TIMx_CR1 表示待传送字节的地址，应通过以下公式给出传送的地址：

(TIMx_CR1 地址) + DBA + (DMA 索引)，其中 DMA 索引 = DBL

在本例中，将为 (TIMx_CR1 地址) + DBA 加上 7 个字节，得到将要复制数据的源/目标地址。这种情况下将向自以下地址开始的 7 个寄存器传输数据：(TIMx_CR1 地址) + DBA

根据 DMA 数据大小的配置，可能发生下面几种情况：

– 如果按半字配置 DMA 数据大小，则将向 7 个寄存器中的每一个传送 16 位数据。

– 如果按字节配置 DMA 数据大小，也将向 7 个寄存器传送数据：第一个寄存器包含第一个 MSB 字节，第二个寄存器包含第一个 LSB 字节，依此类推。因此，使用传送定时器时，还必须指定 DMA 传送的数据大小。

位 7:5 保留，必须保持复位值。

位 4:0 **DBA[4:0]**：DMA 基址 (DMA base address)

该 5 位向量定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：

00000：TIMx_CR1，
00001：TIMx_CR2，
00010：TIMx_SMCR，
...

38.4.22 TIMx 全传输 DMA 地址 (TIMx_DMAR)（x = 1、8）

TIMx DMA address for full transfer

偏移地址：0x4C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAB[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DMAB[31:0]**：DMA 分组传送寄存器 (DMA register for burst accesses)

对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器：(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。

38.4.23 TIMx 捕获/比较模式寄存器 3 (TIMx_CCMR3)（x = 1、8）

TIMx capture/compare mode register 3

偏移地址：0x54

复位值：0x0000 0000

通道 5 和通道 6 只能配置为输出。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC6 CE	OC6M[2:0]			OC6 PE	OC6FE	Res.	Res.	OC5 CE	OC5M[2:0]			OC5PE	OC5FE	Res.	Res.
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

位 31:25 保留，必须保持复位值。

位 23:17 保留，必须保持复位值。

位 15 **OC6CE**：输出比较 6 清零使能 (Output compare 6 clear enable)

请参见 OC1CE 说明。

位 24、14、13、12 **OC6M[3:0]**：输出比较 6 模式 (Output compare 6 mode)

请参见 OC1M 说明。

位 11 **OC6PE**：输出比较 6 预装载使能 (Output compare 6 preload enable)

请参见 OC1PE 说明。

位 10 **OC6FE**：输出比较 6 快速使能 (Output compare 6 fast enable)
请参见 OC1FE 说明。

位 9:8 保留，必须保持复位值。

位 7 **OC5CE**：输出比较 5 清零使能 (Output compare 5 clear enable)
请参见 OC1CE 说明。

位 16、6、5、4 **OC5M[3:0]**：输出比较 5 模式 (Output compare 5 mode)
请参见 OC1M 说明。

位 3 **OC5PE**：输出比较 5 预装载使能 (Output compare 5 preload enable)
请参见 OC1PE 说明。

位 2 **OC5FE**：输出比较 5 快速使能 (Output compare 5 fast enable)
请参见 OC1FE 说明。

位 1:0 保留，必须保持复位值。

38.4.24 TIMx 捕获/比较寄存器 5 (TIMx_CCR5)（x = 1、8）

TIMx capture/compare register 5

偏移地址：0x58

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR5[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **GC5C3**：通道 5 和通道 3 组合 (Group Channel 5 and Channel 3)
通道 3 输出上失真：
0：OC5REF 对 OC3REFC 无影响
1：OC3REFC 是 OC3REFC 和 OC5REF 的逻辑与运算结果
该位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR2 中选择了预装载功能）。
注意：也可在组合 PWM 信号上应用此失真。

位 30 **GC5C2**：通道 5 和通道 2 组合 (Group Channel 5 and Channel 2)
通道 2 输出上失真：
0：OC5REF 对 OC2REFC 无影响
1：OC2REFC 是 OC2REFC 和 OC5REF 的逻辑与运算结果
该位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR1 中选择了预装载功能）。
注意：也可在组合 PWM 信号上应用此失真。

位 29 **GC5C1**：通道 5 和通道 1 组合 (Group Channel 5 and Channel 1)

通道 1 输出上失真：

0：OC5REF 对 OC1REFC5 无影响

1：OC1REFC 是 OC1REFC 和 OC5REF 的逻辑与运算结果

该位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR1 中选择了预装载功能）。

注意：也可在组合 PWM 信号上应用此失真。

位 28:16 保留，必须保持复位值。

位 15:0 **CCR5[15:0]**：捕获/比较 5 值 (Capture/Compare 5 value)

CCR5 是捕获/比较寄存器 5 的预装载值。

如果没有通过 TIMx_CCMR3 寄存器中的 OC5PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 5）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC5 输出上发出信号的值。

38.4.25 TIMx 捕获/比较寄存器 6 (TIMx_CCR6)（x = 1、8）

TIMx capture/compare register 6

偏移地址：0x5C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR6[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR6[15:0]**：捕获/比较 6 值 (Capture/Compare 6 value)

CCR6 是捕获/比较寄存器 6 的预装载值。

如果没有通过 TIMx_CCMR3 寄存器中的 OC6PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 6）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC6 输出上发出信号的值。

38.4.26 TIM1 复用功能选项寄存器 1 (TIM1_AF1)

TIM1 alternate function option register 1

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL[3:2]	
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETRSEL[1:0]		Res.	Res.	Res.	Res.	BKINP	BKDF1 BK0E	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINE
rw	rw					rw	rw								rw

位 31:18 保留，必须保持复位值。

位 17:14 **ETRSEL[3:0]**：ETR 源选择 (ETR source selection)
这些位选择 ETR 输入源。
0000：ETR 输入连接到 I/O
0001：保留
0010：保留
0011：ADC1 AWD1
0100：ADC1 AWD2
0101：ADC1 AWD3
0110：ADC2 AWD1
0111：ADC2 AWD2
1000：ADC2 AWD3
其他值：保留
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，这些位即无法修改。

位 13:10 保留，必须保持复位值。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
该位选择 BKIN 复用功能输入有效电平，必须与 BKP 极性位一起编程。
0：BKIN 输入极性不反相（BKP=0 时低电平有效，BKP=1 时高电平有效）
1：BKIN 输入极性反相（BKP=0 时高电平有效，BKP=1 时低电平有效）
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 8 **BKDF1BK0E**：BRK dfsdm1_break[0] 使能 (BRK dfsdm1_break[0] enable)
该位使能定时器 BRK 输入的 dfsdm1_break[0]。dfsdm1_break[0] 输出与其他 BRK 源进行“或”运算。
0：禁止 dfsdm1_break[0] 输入
1：使能 dfsdm1_break[0] 输入
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 7:1 保留，必须保持复位值。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
该位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

注： *请参见图 357：TIM1/TIM8 ETR 输入电路和图 378：刹车事件和刹车 2 电路概述。*

38.4.27 TIM1 复用功能寄存器 2 (TIM1_AF2)

TIM1 Alternate function register 2

偏移地址：0x64

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	BK2 INP	BK2DF1 BK1E	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BK2INE
						rw	rw								rw

位 31:10 保留，必须保持复位值。

位 9 **BK2INP**：BRK2 BKIN2 输入极性 (BRK2 BKIN2 input polarity)

该位选择 BKIN2 复用功能输入有效电平，必须与 BK2P 极性位一起编程。

0：BKIN2 输入极性不反相（BK2P=0 时低电平有效，BK2P=1 时高电平有效）

1：BKIN2 输入极性反相（BK2P=0 时高电平有效，BK2P=1 时低电平有效）

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 8 **BK2DF1BK1E**：BRK2 dfsdm1_break[1] 使能 (BRK2 dfsdm1_break[1] enable)

该位使能定时器 BRK2 输入的 dfsdm1_break[1]。dfsdm1_break[1] 输出与其他 BRK2 源进行“或”运算。

0：禁止 dfsdm1_break[1] 输入

1：使能 dfsdm1_break[1] 输入

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 7:1 保留，必须保持复位值。

位 0 **BK2INE**：BRK2 BKIN 输入使能 (BRK2 BKIN input enable)

该位使能定时器 BRK2 输入的 BKIN2 复用功能。BKIN2 输入与其他 BRK2 源进行“或”运算。

0：禁止 BKIN2 输入

1：使能 BKIN2 输入

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

注： *请参见图 378：刹车事件和刹车 2 电路概述。*

38.4.28 TIM8 复用功能选项寄存器 1 (TIM8_AF1)

TIM8 Alternate function option register 1

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL[3:2]	
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETRSEL[1:0]		Res.	Res.	Res.	Res.	BKINP	BKDF1 BK2E	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINE
rw	rw					rw	rw								rw

位 31:18 保留，必须保持复位值。

位 17:14 **ETRSEL[3:0]**：ETR 源选择 (ETR source selection)
这些位选择 ETR 输入源。
0000：ETR 输入连接到 I/O
0001：保留
0010：保留
0011：ADC1 AWD1
0100：ADC1 AWD2
0101：ADC1 AWD3
0110：ADC2 AWD1
0111：ADC2 AWD2
1000：ADC2 AWD3
其他值：保留
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，这些位即无法修改。

位 13:10 保留，必须保持复位值。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
该位选择 BKIN 复用功能输入有效电平，必须与 BKP 极性位一起编程。
0：BKIN 输入极性不反相（BKP=0 时低电平有效，BKP=1 时高电平有效）
1：BKIN 输入极性反相（BKP=0 时高电平有效，BKP=1 时低电平有效）
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 8 **BKDF1BK2E**：BRK dfsdm1_break[2] 使能 (BRK dfsdm1_break[2] enable)
该位使能定时器 BRK 输入的 dfsdm1_break[2]。dfsdm1_break[2] 输出与其他 BRK 源进行“或”运算。
0：禁止 dfsdm1_break[2] 输入
1：使能 dfsdm1_break[2] 输入
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 7:1 保留，必须保持复位值。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
该位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

注： *请参见图 357：TIM1/TIM8 ETR 输入电路和图 378：刹车事件和刹车 2 电路概述。*

38.4.29 TIM8 复用功能选项寄存器 2 (TIM8_AF2)

TIM8 Alternate function option register 2

偏移地址：0x64

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	BK2 INP	BK2DF1 BK3E	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BK2INE
						rw	rw								rw

位 31:10 保留，必须保持复位值。

位 9 **BK2INP**：BRK2 BKIN2 输入极性 (BRK2 BKIN2 input polarity)
该位选择 BKIN2 复用功能输入有效电平，必须与 BK2P 极性位一起编程。
0：BKIN2 输入极性不反相（BK2P=0 时低电平有效，BK2P=1 时高电平有效）
1：BKIN2 输入极性反相（BK2P=0 时高电平有效，BK2P=1 时低电平有效）
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 8 **BK2DF1BK3E**：BRK2 dfsdm1_break[3] 使能 (BRK2 dfsdm1_break[3] enable)
该位使能定时器 BRK2 输入的 dfsdm1_break[3]。dfsdm1_break[3] 输出与其他 BRK2 源进行“或”运算。
0：禁止 dfsdm1_break[3] 输入
1: dfsdm1_break[3] input enabled
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 7:1 保留，必须保持复位值。

位 0 **BK2INE**：BRK2 BKIN 输入使能 (BRK2 BKIN input enable)
该位使能定时器 BRK2 输入的 BKIN2 复用功能。BKIN2 输入与其他 BRK2 源进行“或”运算。
0：禁止 BKIN2 输入
1：使能 BKIN2 输入
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

注： *请参见图 378：刹车事件和刹车 2 电路概述。*

38.4.30 TIM1 定时器输入选择寄存器 (TIM1_TISEL)

TIM1 timer input selection register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **TI4SEL[3:0]**：选择 TI4[0] 到 TI4[15] 输入 (selects TI4[0] to TI4[15] input)
0000：TIM1_CH4 输入
其他值：保留

位 23:20 保留，必须保持复位值。

位 19:16 **TI3SEL[3:0]**：选择 TI3[0] 到 TI3[15] 输入 (selects TI3[0] to TI3[15] input)
0000：TIM1_CH3 输入
其他值：保留

位 15:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：选择 TI2[0] 到 TI2[15] 输入 (selects TI2[0] to TI2[15] input)
0000：TIM1_CH2 输入
其他值：保留

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)
0000：TIM1_CH1 输入
其他值：保留

38.4.31 TIM8 定时器输入选择寄存器 (TIM8_TISEL)

TIM8 timer input selection register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **TI4SEL[3:0]**：选择 TI4[0] 到 TI4[15] 输入 (selects TI4[0] to TI4[15] input)
0000：TIM8_CH4 输入
其他值：保留

位 23:20 保留，必须保持复位值。

位 19:16 **TI3SEL[3:0]**：选择 TI3[0] 到 TI3[15] 输入 (selects TI3[0] to TI3[15] input)
0000：TIM8_CH3 输入
其他值：保留

位 15:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：选择 TI2[0] 到 TI2[15] 输入 (selects TI2[0] to TI2[15] input)
0000：TIM8_CH2 输入
其他值：保留

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)
0000：TIM8_CH1 输入
其他值：保留

38.4.32 TIM1 寄存器映射

TIM1 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 352. TIM1 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIM1_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0	0	0	0	0	0	0	0
0x04	**TIM1_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS2[3:0]				Res.	OIS6	Res.	OIS5	Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS [2:0]			CCDS	CCUS	Res.	CCPC
	Reset value									0	0	0	0		0		0		0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x08	**TIM1_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS [4:3]		Res.	Res.	Res.	SMS[3]	ETP	ECE	ETPS [1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value											0	0				0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0x0C	**TIM1_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**TIM1_SR**	Res.	Res.	Res	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6IF	CC5IF	Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value															0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**TIM1_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																								0	0	0	0	0	0	0	0	0
0x18	**TIM1_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	OC2CE	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM1_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**TIM1_CCMR2** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]	OC4CE	OC4M [2:0]			OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM1_CCMR2 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]				IC4PSC [1:0]		CC4S [1:0]		IC3F[3:0]				IC3PSC [1:0]		CC3S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**TIM1_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E	CC4NP	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	Reset value											0	0			0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 352. TIM1 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x24	**TIM1_CNT**	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIM1_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIM1_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	**TIM1_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	**TIM1_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIM1_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**TIM1_CCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	**TIM1_CCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIM1_BDTR**	Res.	Res.	BK2BID	BKBID	BK2DSRM	BKDSRM	BK2P	BK2E	BK2F[3:0]				BKF[3:0]				MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DT[7:0]							
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**TIM1_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIM1_DMAR**	DMAB[31:0]																															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54	**TIM1_CCMR3** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]	OC6CE	OC6M [2:0]			OC6PE	OC6FE	Res.	Res.	OC5CE	OC5M [2:0]			OC5PE	OC5FE	Res.	Res.
	Reset value								0								0	0	0	0	0	0	0			0	0	0	0	0	0		
0x58	**TIM1_CCR5**	GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR5[15:0]															
	Reset value	0	0	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 352. TIM1 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x5C	**TIM1_CCR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR6[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**TIM1_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res ETRSEL [3:0]		Res.	Res.	Res.	Res.	Res.	Res.	BKINP	BKDF1BK0E	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINE
	Reset value															0	0	0	0					0	0								1
0x64	**TIM1_AF2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BK2INP	BK2DF1BK1E	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BK2INE
	Reset value																							0	0								1
0x68	**TIM1_TISEL**	Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]				Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

38.4.33 TIM8 寄存器映射

TIM8 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 353. TIM8 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIM8_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0	0	0	0	0	0	0	0
0x04	**TIM8_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS2[3:0]				Res.	OIS6	Res.	OIS5	Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS [2:0]			CCDS	CCUS	Res.	CCPC
	Reset value									0	0	0	0		0		0		0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x08	**TIM8_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS [4:3]		Res.	Res.	Res.	SMS[3]	ETP	ECE	ETP S [1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value											0	0				0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0x0C	**TIM8_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**TIM8_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6IF	CC5IF	Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value															0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 353. TIM8 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x14	**TIM8_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																								0	0	0	0	0	0	0	0	0
0x18	**TIM8_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	OC2CE	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM8_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**TIM8_CCMR2** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]	OC4CE	OC4M [2:0]			OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM8_CCMR2 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]				IC4PSC [1:0]		CC4S [1:0]		IC3F[3:0]				IC3PSC [1:0]		CC3S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**TIM8_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E	CC4NP	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	Reset value											0	0			0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	**TIM8_CNT**	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIM8_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIM8_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	**TIM8_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	**TIMx_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIM8_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**TIM8_CCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	**TIM8_CCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 353. TIM8 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x44	**TIM8_BDTR**	Res.	Res.	BK2BID	BKBID	BK2DSRM	BKDSRM	BK2P	BK2E	BK2F[3:0]				BKF[3:0]				MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DT[7:0]							
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**TIM8_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIM8_DMAR**	DMAB[31:0]																															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54	**TIM8_CCMR3** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]	OC6CE	OC6M [2:0]			OC6PE	OC6FE	Res.	Res.	OC5CE	OC5M [2:0]			OC5PE	OC5FE	Res.	Res.
	Reset value								0								0	0	0	0	0	0	0			0	0	0	0	0	0		
0x58	**TIM8_CCR5**	GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR5[15:0]															
	Reset value	0	0	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5C	**TIM8_CCR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR6[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**TIM8_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res ETRSEL [3:0]								BKINP	BKDF1BK2E								BKINE
	Reset value															0	0	0	0					0	0								1
0x64	**TIM8_AF2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BK2INP	BK2DF1BK3E								BK2INE
	Reset value																							0	0								1
0x68	**TIM8_TISEL**	Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]				Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

39 通用定时器 (TIM2/TIM3/TIM4/TIM5)

39.1 TIM2/TIM3/TIM4/TIM5 简介

通用定时器包含一个 16 位/32 位自动重载计数器，该计数器由可编程预分频器驱动。

它们可用于多种用途，包括测量输入信号的脉冲宽度（*输入捕获*）或生成输出波形（*输出比较和 PWM*）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

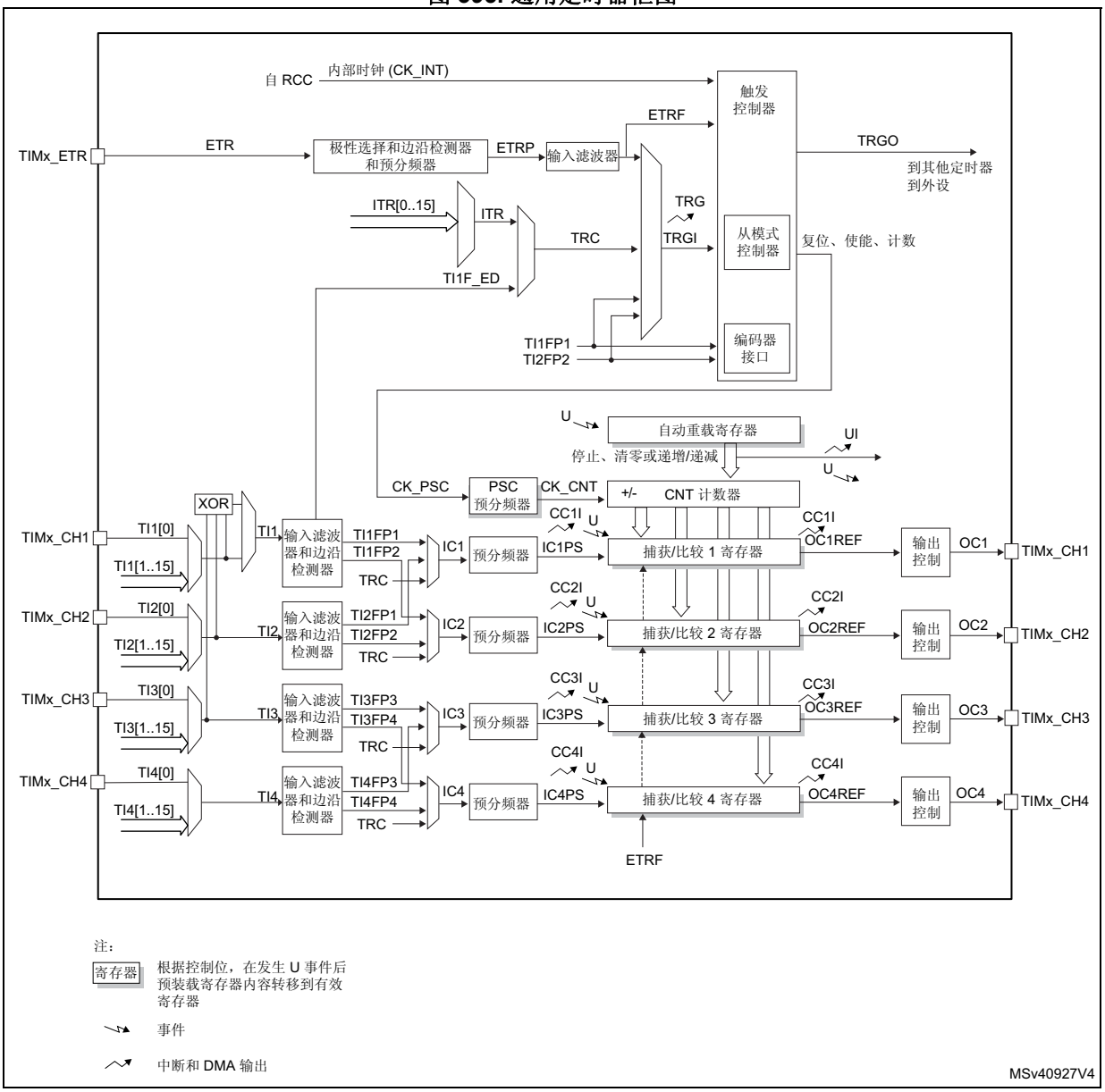
这些定时器彼此完全独立，不共享任何资源。如*第 39.3.19 节：定时器同步*中所述，它们可以一起同步操作。

39.2 TIM2/TIM3/TIM4/TIM5 主要特性

通用 TIMx 定时器具有以下特性：

- 16 位（TIM3 和 TIM4）或 32 位（TIM2 和 TIM5）递增、递减、递增/递减自动重载计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 到 65535 之间。
- 多达 4 个独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿对齐模式和中心对齐模式）
  - 单脉冲模式输出
- 同步电路可实现外部信号对定时器的控制以及多个定时器互联。
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - 输入捕获
  - 输出比较
- 支持定位用增量（正交）编码器和霍尔传感器电路
- 触发输入可用作外部时钟或用于逐周期电流管理

图 395. 通用定时器框图



39.3 TIM2/TIM3/TIM4/TIM5 功能描述

39.3.1 时基单元

可编程定时器的主要模块由一个 16 位/32 位 计数器及其相关的自动重载寄存器组成。计数器可递增计数、递减计数或进行递增递减计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以随时直接写入影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值（或者在递减计数时达到下溢值）并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将产生更新事件。该更新事件也可由软件产生。下文将详细介绍各配置的更新事件的产生过程。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

请注意，实际的计数器使能信号 CNT_EN 在 CEN 置 1 的一个时钟周期后被置 1。

预分频器说明

预分频器可对计数器时钟进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 16 位/32 位寄存器（TIMx_PSC 寄存器）所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 396 和*图 397* 以一些示例说明在预分频比实时变化时计数器的行为：

图 396. 预分频器分频由 1 变为 2 时的计数器时序图

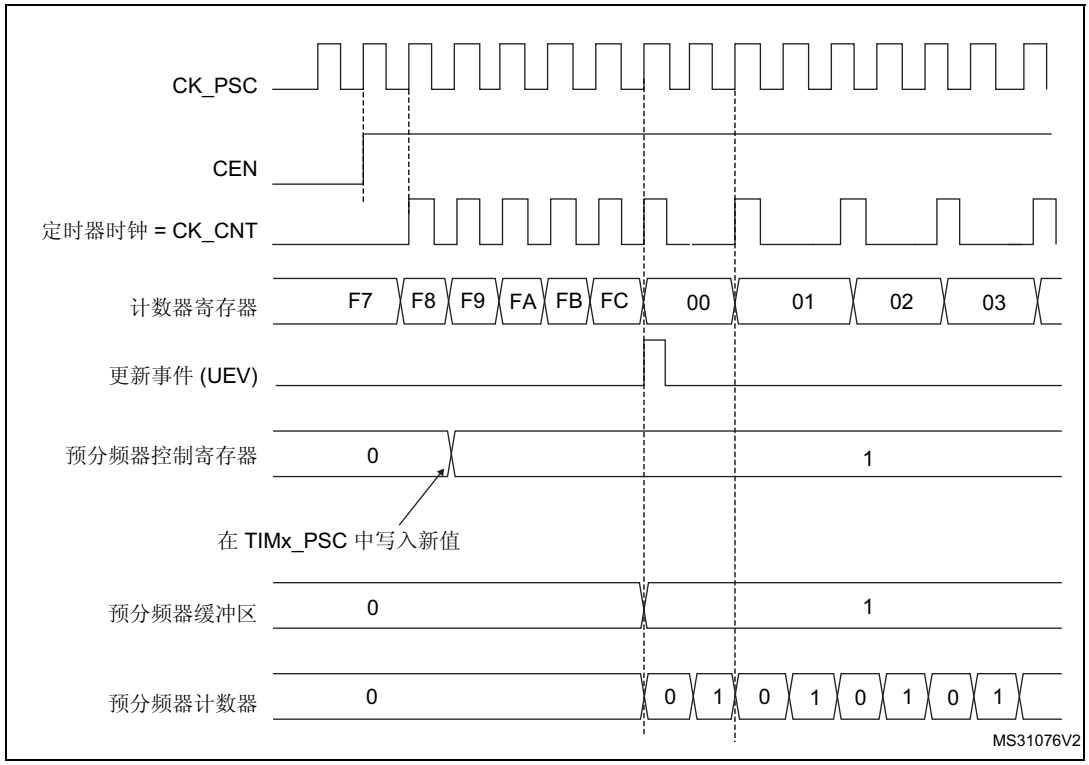
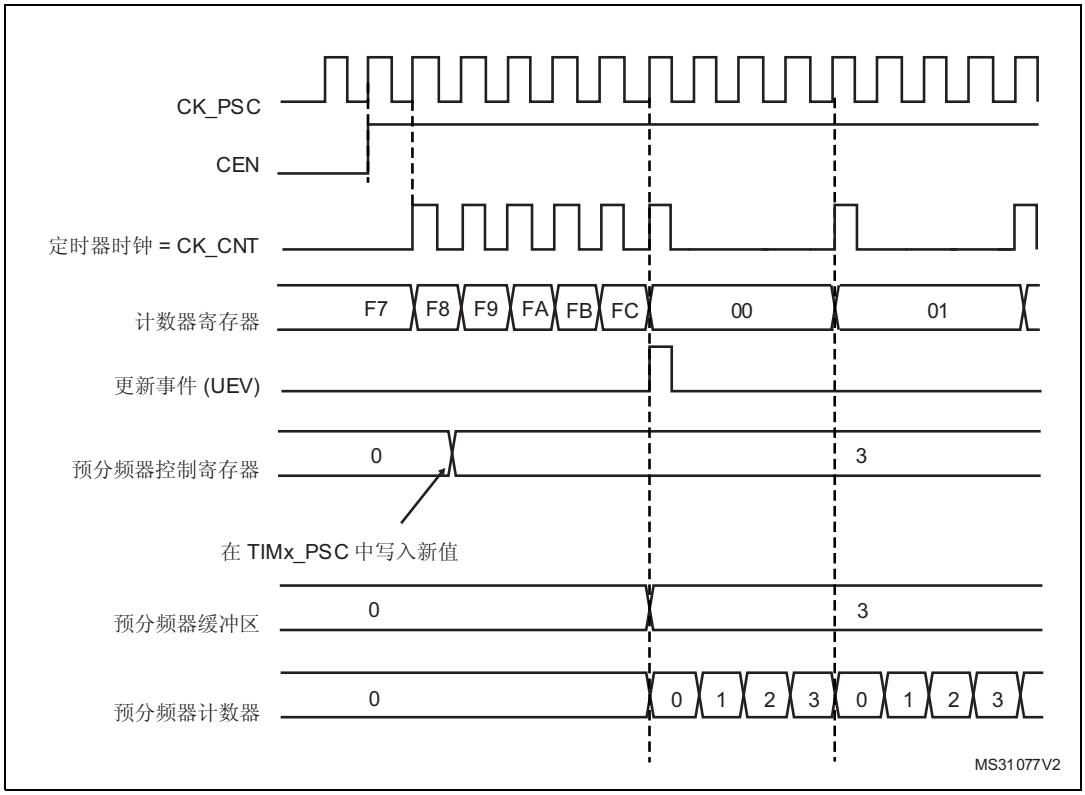


图 397. 预分频器分频由 1 变为 4 时的计数器时序图



39.3.2 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

每次发生计数器上溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。利用这一特性，在定时器的输入捕获操作时可以通过对 UG 位置 1 让计数器复位而不至于触发更新中断，也就避免了捕获中断和更新中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）
- 使用预装载值 (TIMx_ARR) 更新自动重载影子寄存器

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 398. 计数器时序图，1 分频内部时钟

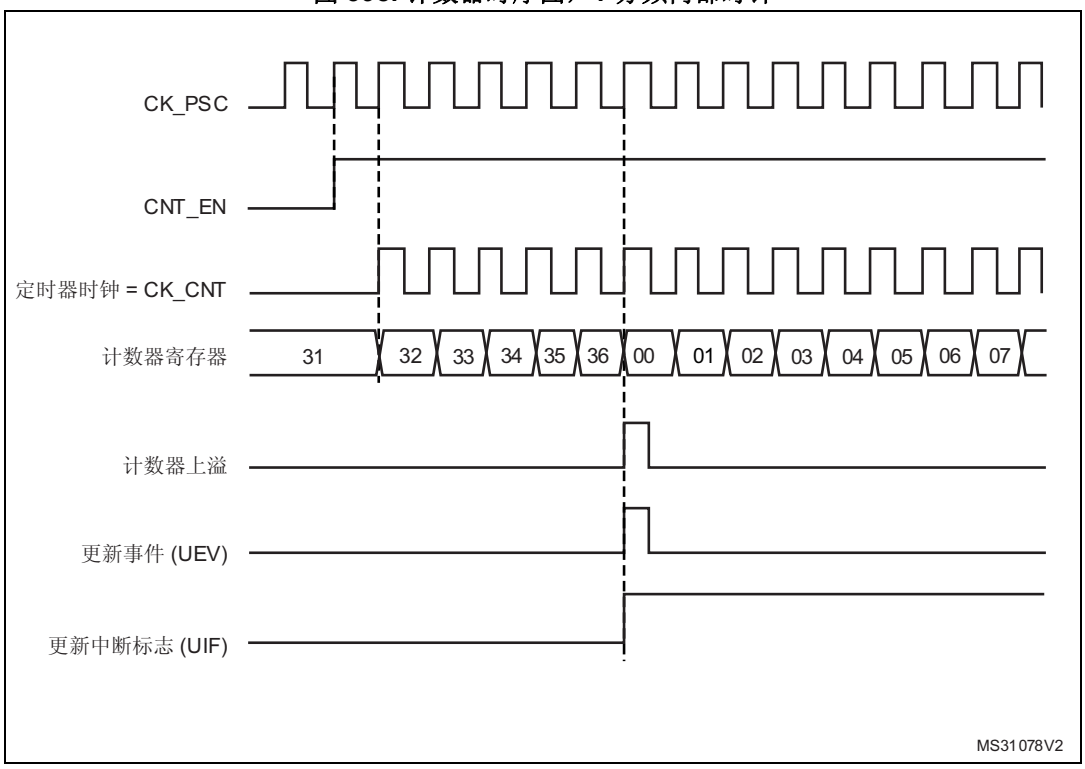


图 399. 计数器时序图，2 分频内部时钟

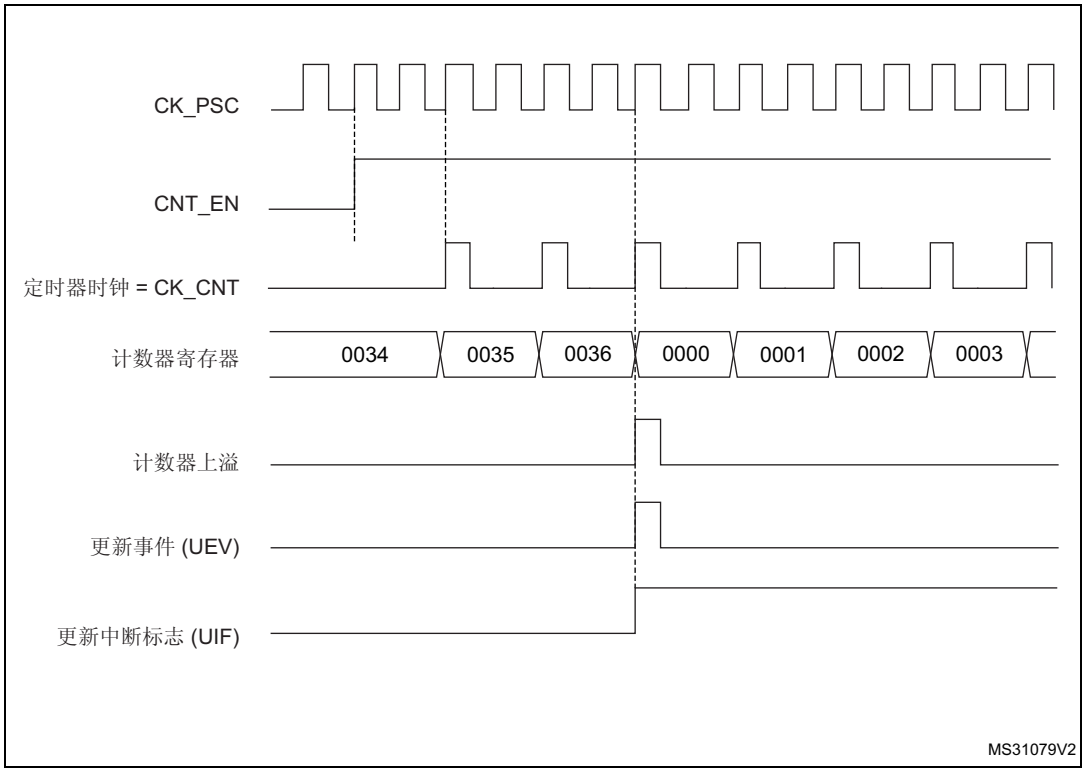


图 400. 计数器时序图，4 分频内部时钟

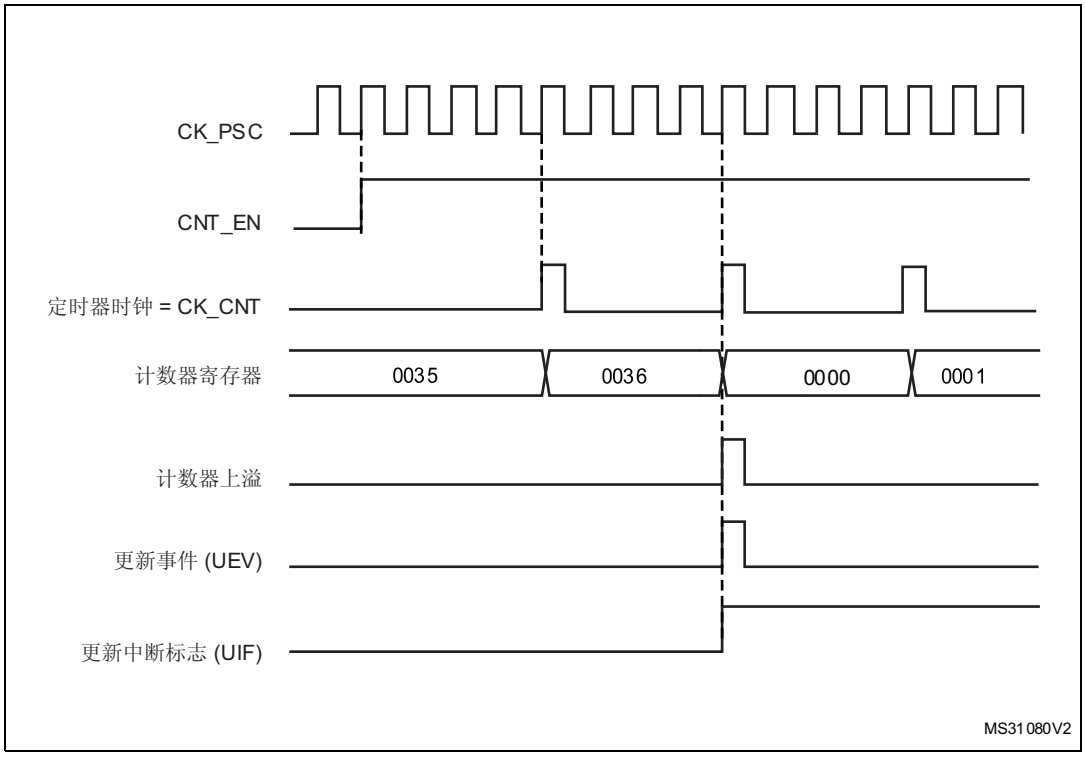


图 401. 计数器时序图，N 分频内部时钟

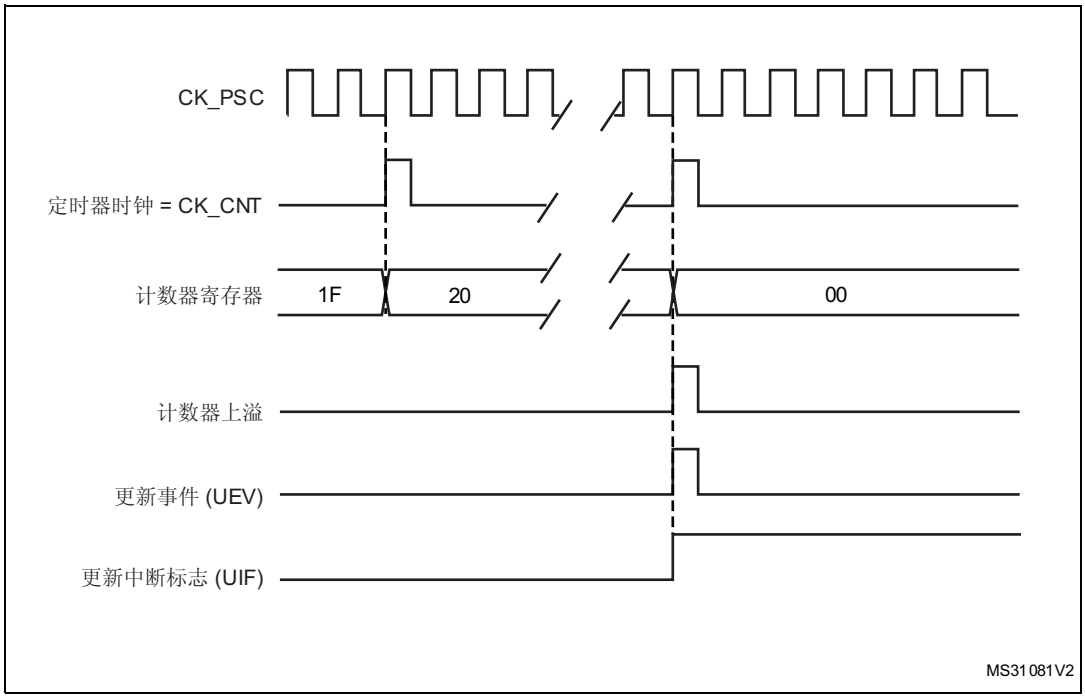


图 402. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

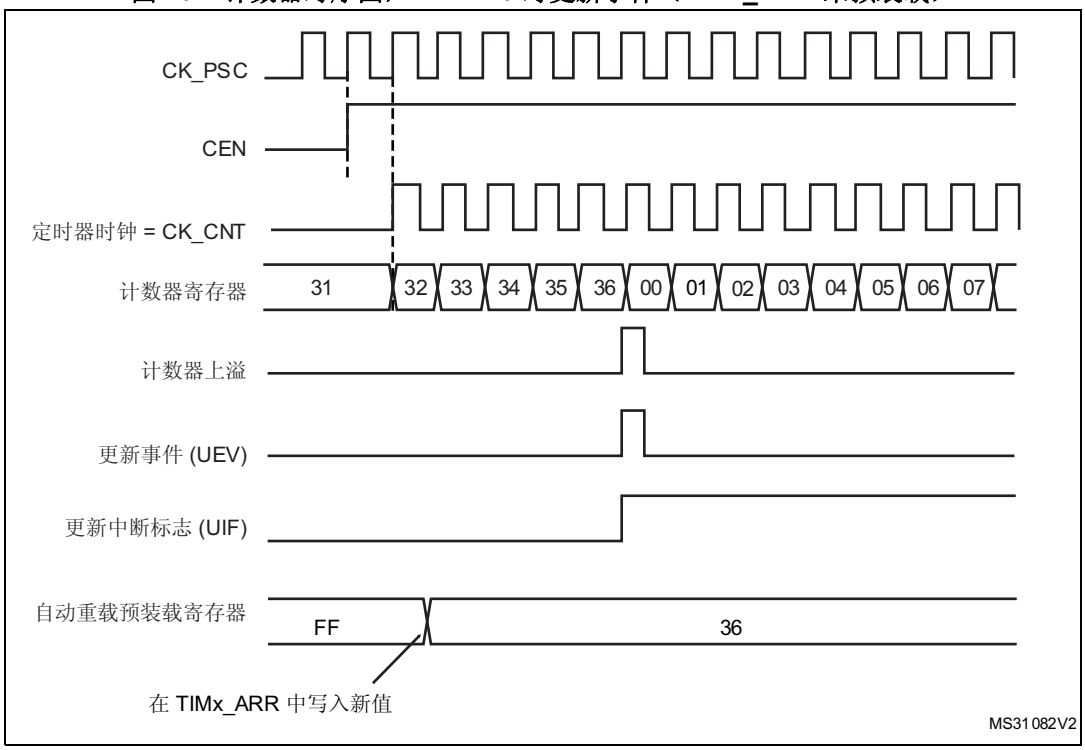
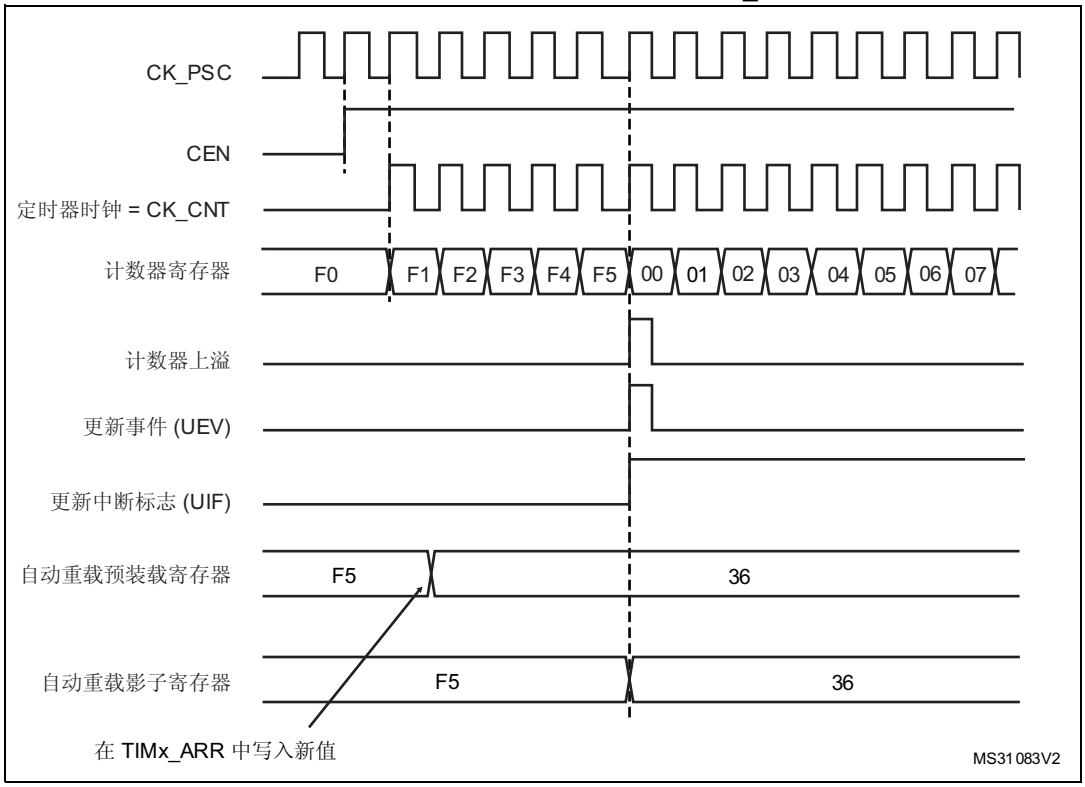


图 403. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载）



递减计数模式

在递减计数模式下，计数器从自动重载值（TIMx_ARR 寄存器的内容）开始递减计数到 0，然后重新从自动重载值开始计数并生成计数器下溢事件。

每次发生计数器下溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会从当前自动重载值开始重新计数，而预分频器计数器则从 0 开始重新计数（但预分频比保持不变）。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。利用这一特性，在定时器的输入捕获操作时可以通过对 UG 位置 1 让计数器复位而不至于触发更新中断，也就避免了在捕获中断里触发更新中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载影子寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，ARR 寄存器更新在计数器重载之前被更新，因此下一个周期就是预期的值。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 404. 计数器时序图，1 分频内部时钟

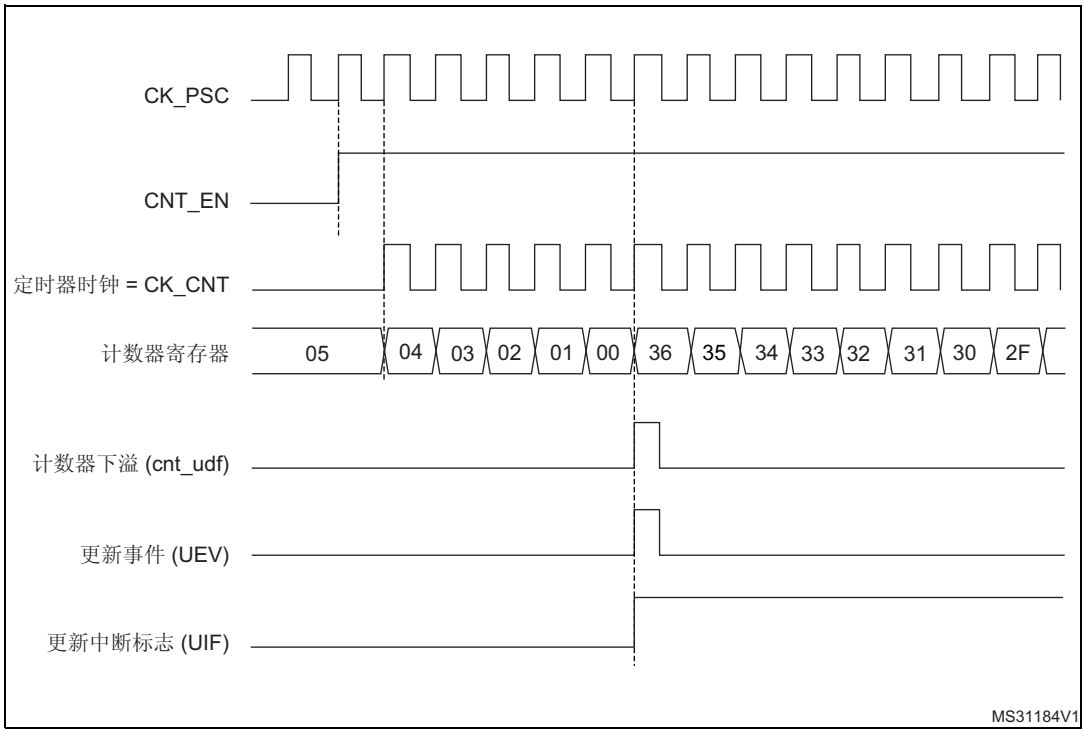


图 405. 计数器时序图，2 分频内部时钟

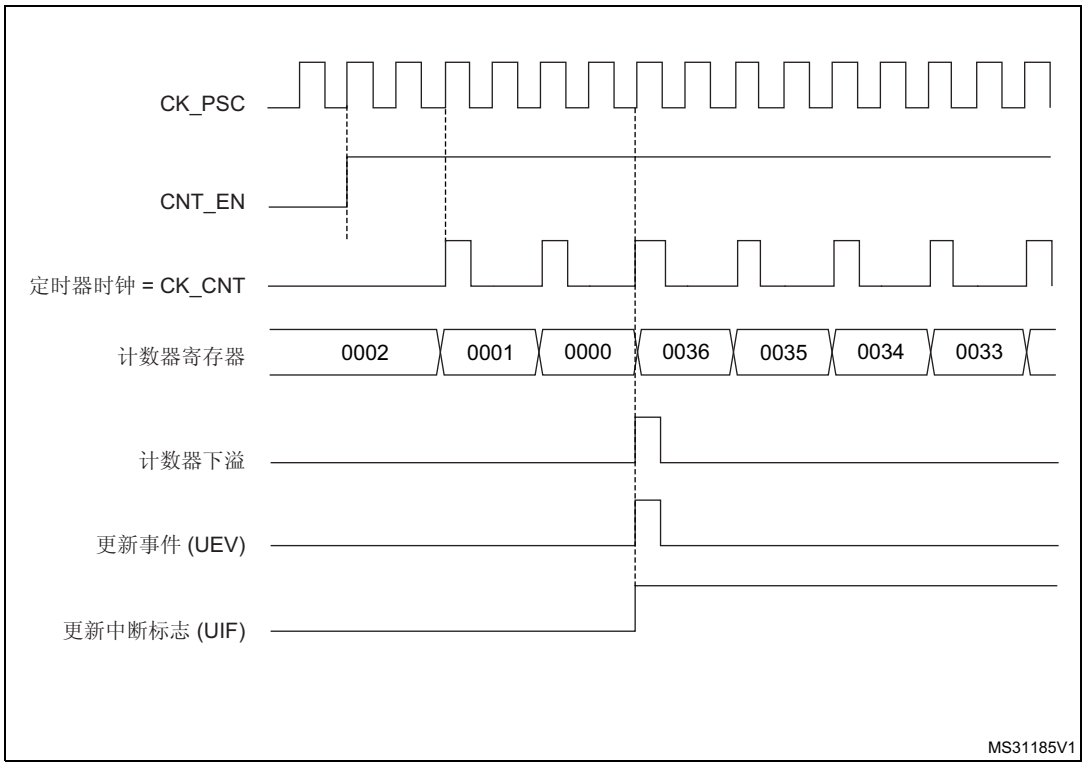


图 406. 计数器时序图，4 分频内部时钟

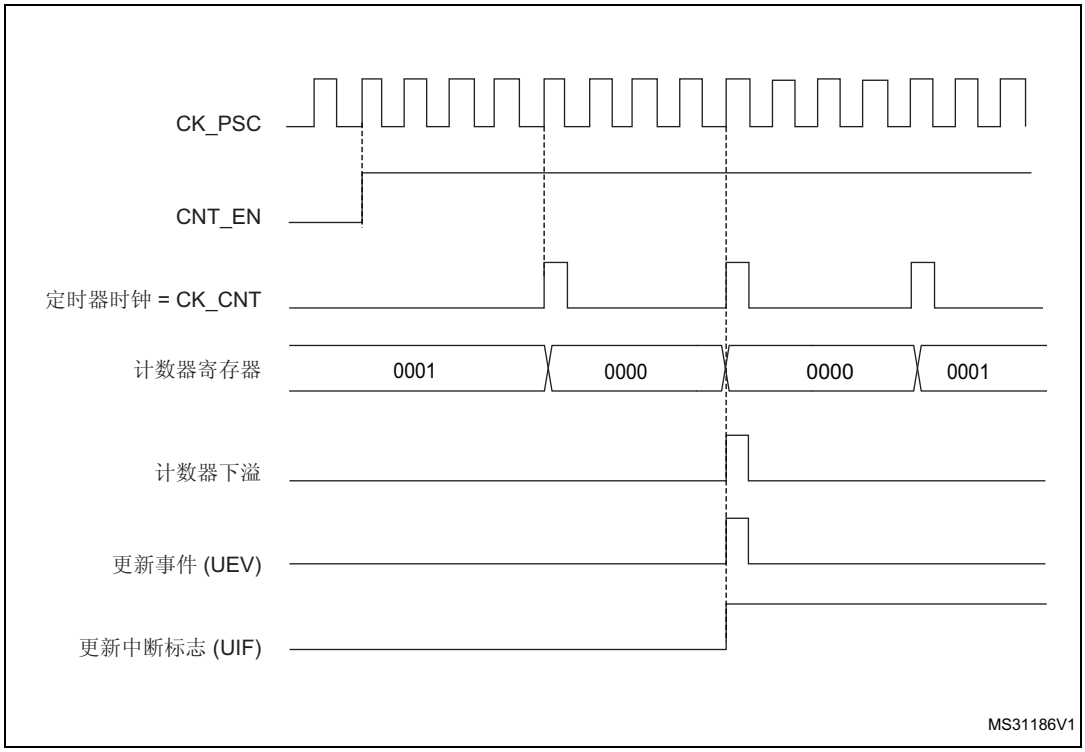


图 407. 计数器时序图，N 分频内部时钟

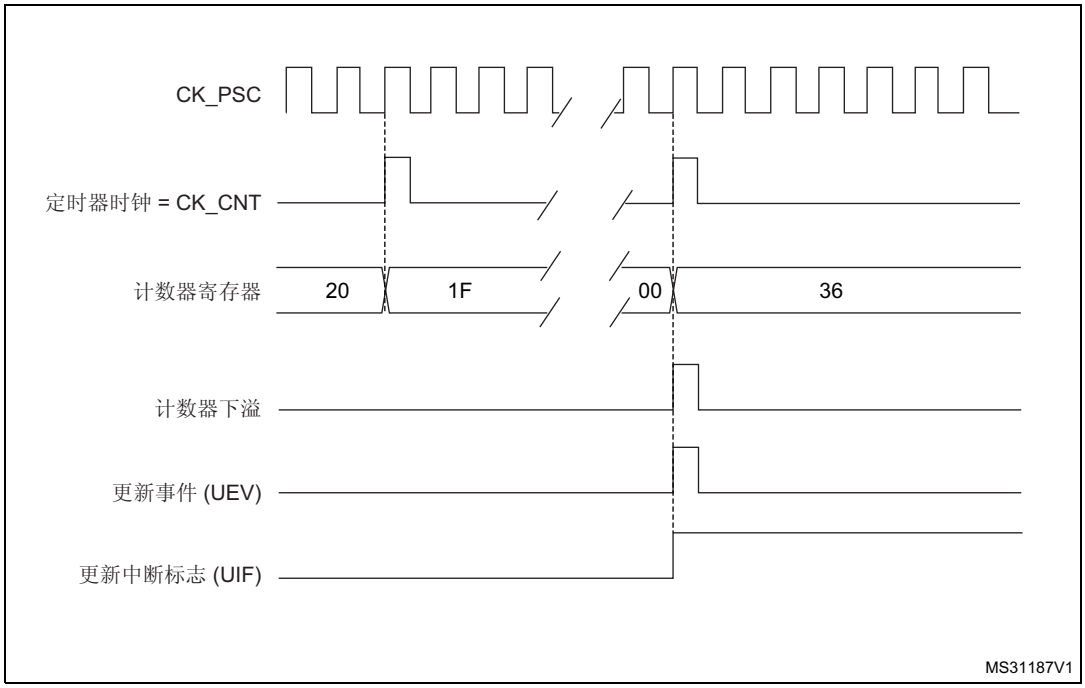
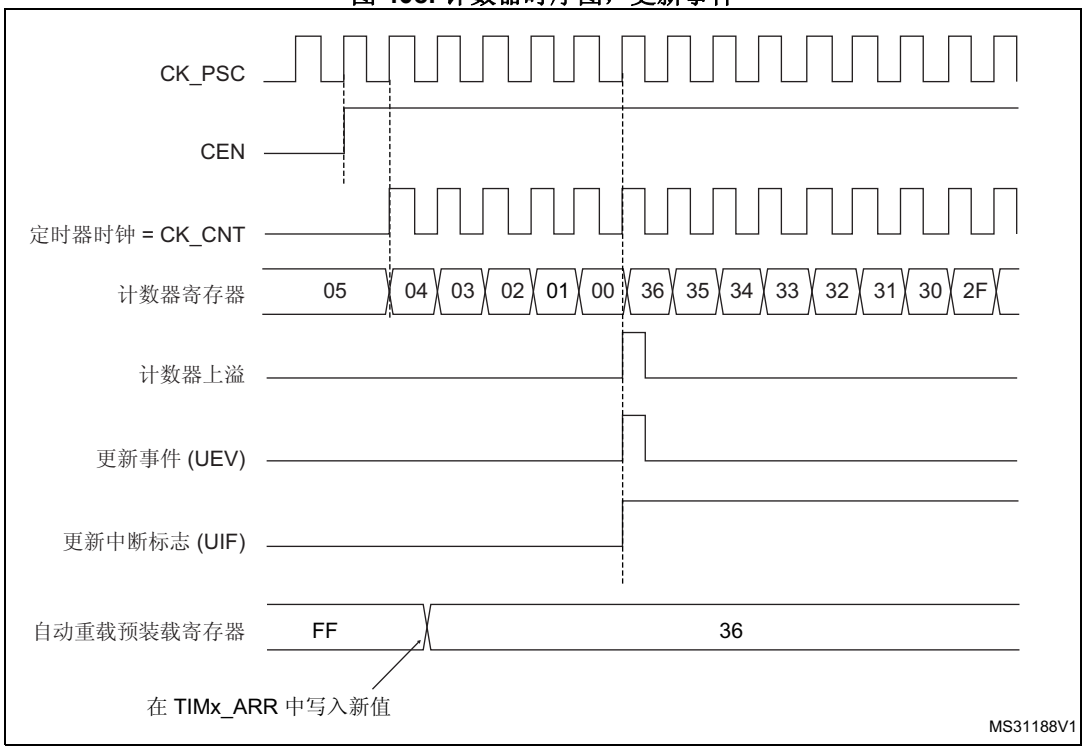


图 408. 计数器时序图，更新事件



中心对齐模式（递增/递减计数）

在中心对齐模式下，计数器从 0 开始计数到自动重载值（TIMx_ARR 寄存器的内容）– 1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数（中心对齐模式 1，CMS =“01”）、计数器递增计数（中心对齐模式 2，CMS =“10”）以及计数器递增/递减计数（中心对齐模式 3，CMS =“11”）。

此模式下无法写入方向位（TIMx_CR1 寄存器中的 DIR 位）。而是由硬件更新并指示当前计数器方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

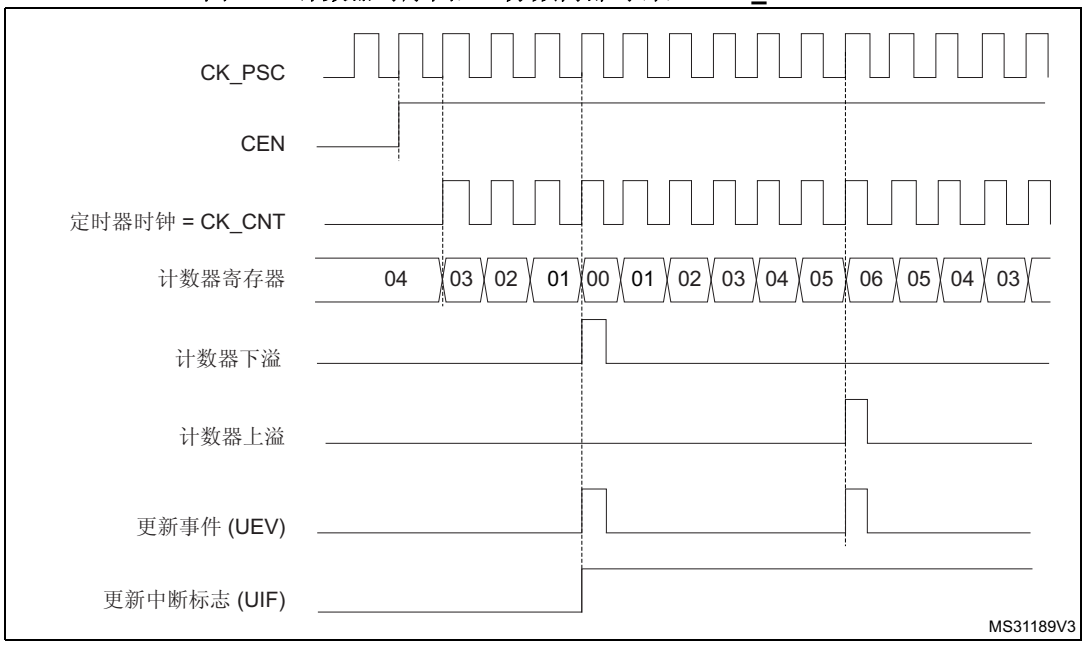
此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。利用这一特性，在定时器的输入捕获操作时可以通过对 UG 位置 1 让计数器清零而不至于触发更新中断，也就避免了在捕获中断里触发更新中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载影子寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，如果更新操作是由计数器上溢触发的，则 ARR 寄存器在计数器重载之前更新，因此，下一个计数周期就是我们所希望的新的周期长度（计数器被重载新的值）。

以下各图以一些示例说明不同时钟频率下计数器的行为。

图 409. 计数器时序图，1 分频内部时钟，TIMx_ARR=0x6



1. 此处使用中心对齐模式 1（有关详细信息，请参见*第 2032 页的第 39.4.1 节：TIMx 控制寄存器 1 (TIMx_CR1)（x = 2 到 5）*）。

图 410. 计数器时序图，2 分频内部时钟

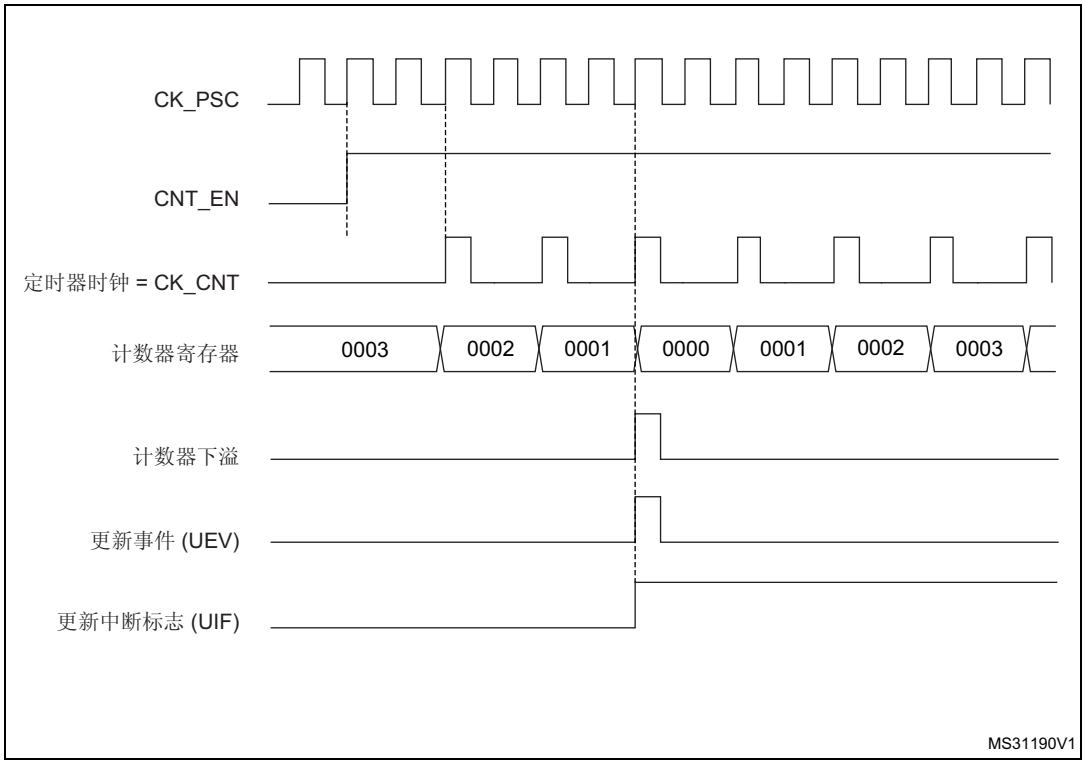
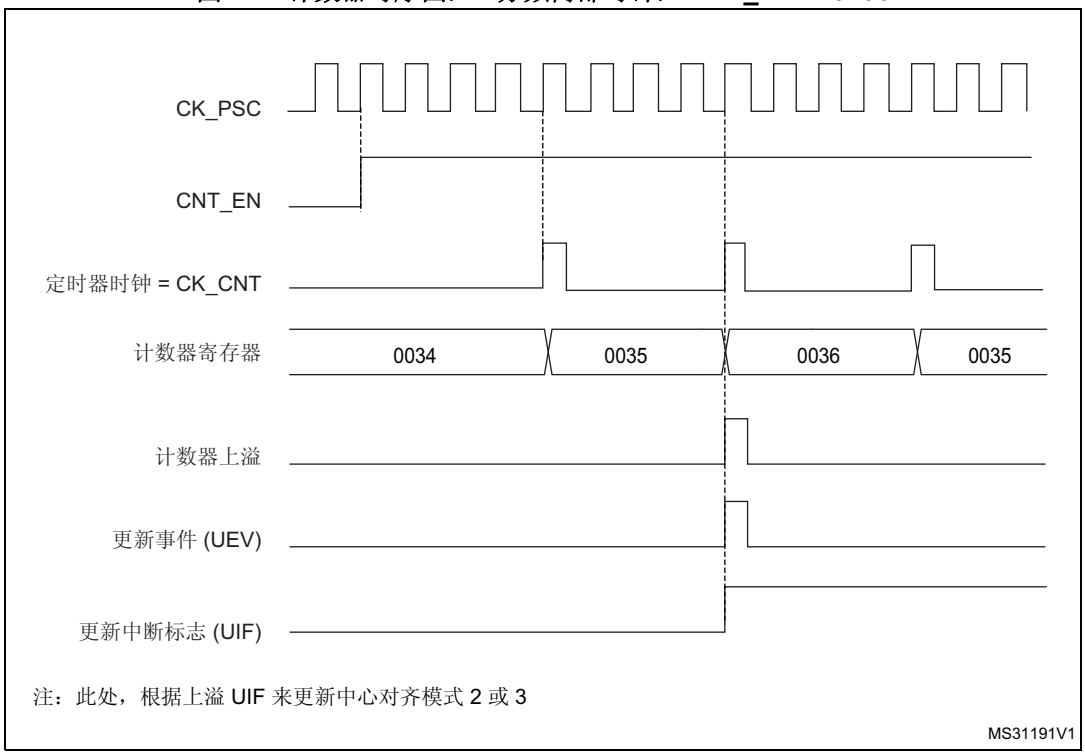


图 411. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36



1. 中心对齐模式 2 或模式 3 与上溢 UIF 结合使用。

图 412. 计数器时序图，N 分频内部时钟

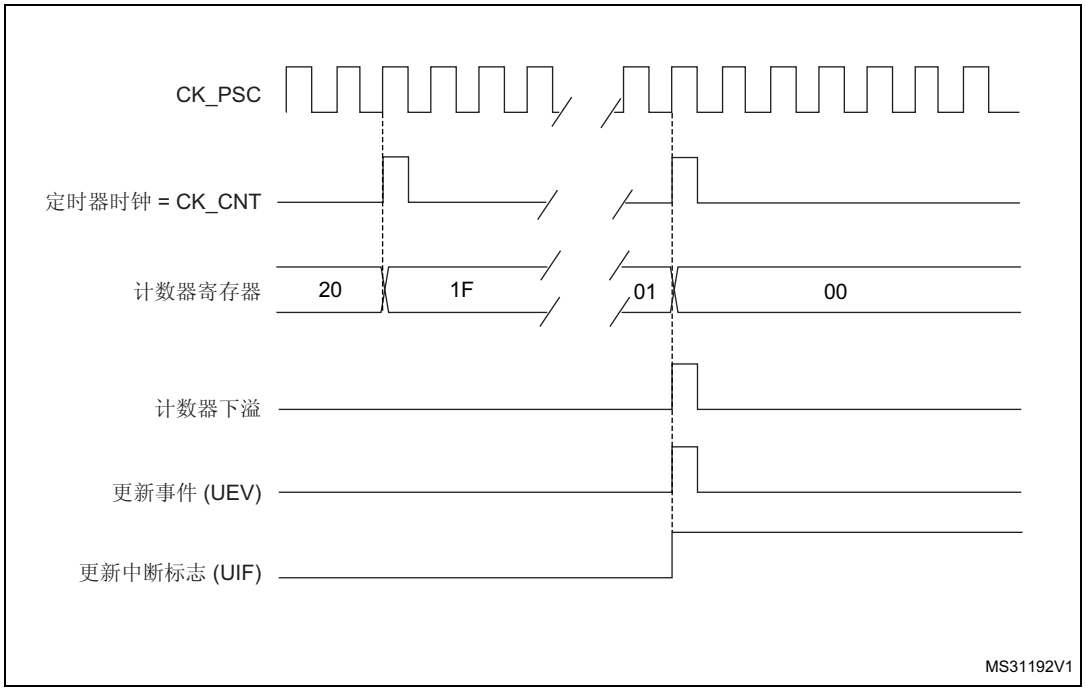


图 413. 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

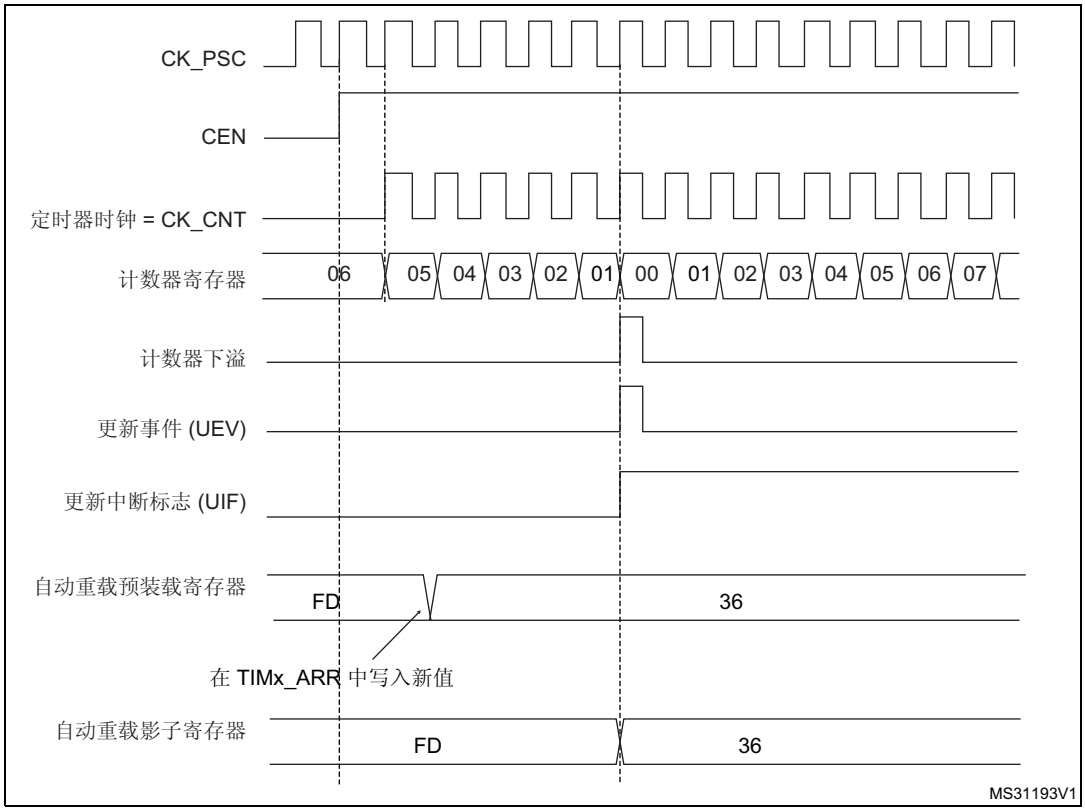
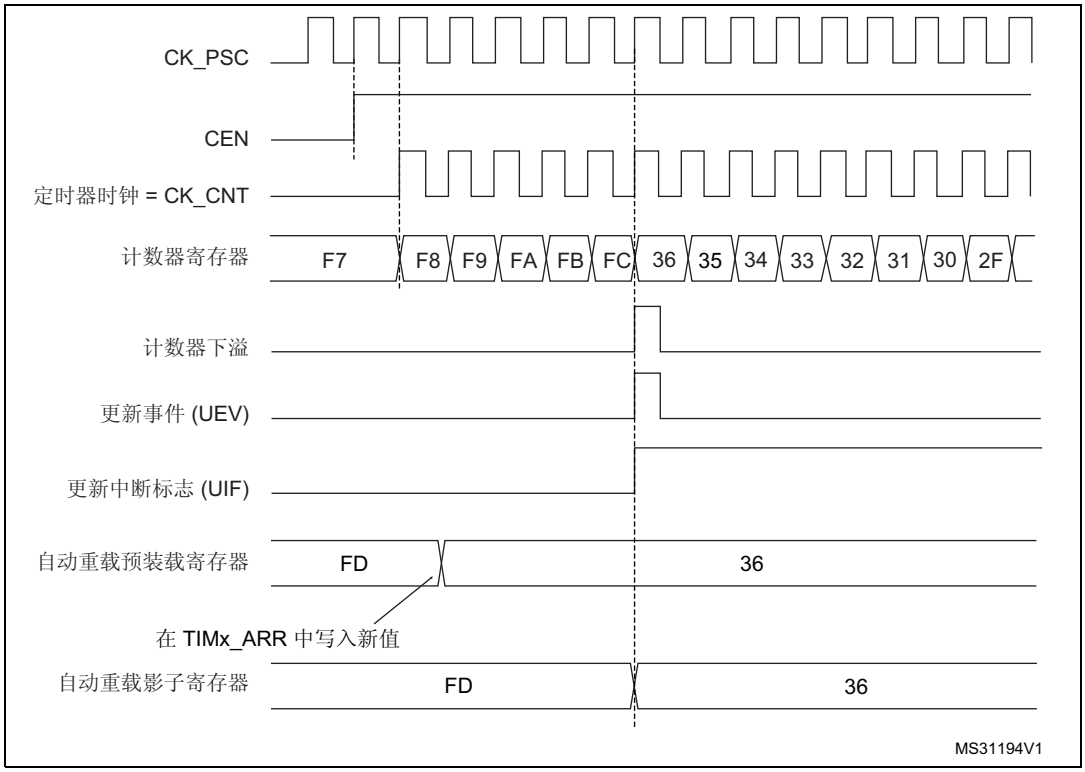


图 414. 计数器时序图，ARPE=1 时的更新事件（计数器上溢）



39.3.3 时钟选择

计数器时钟可由下列时钟源提供：

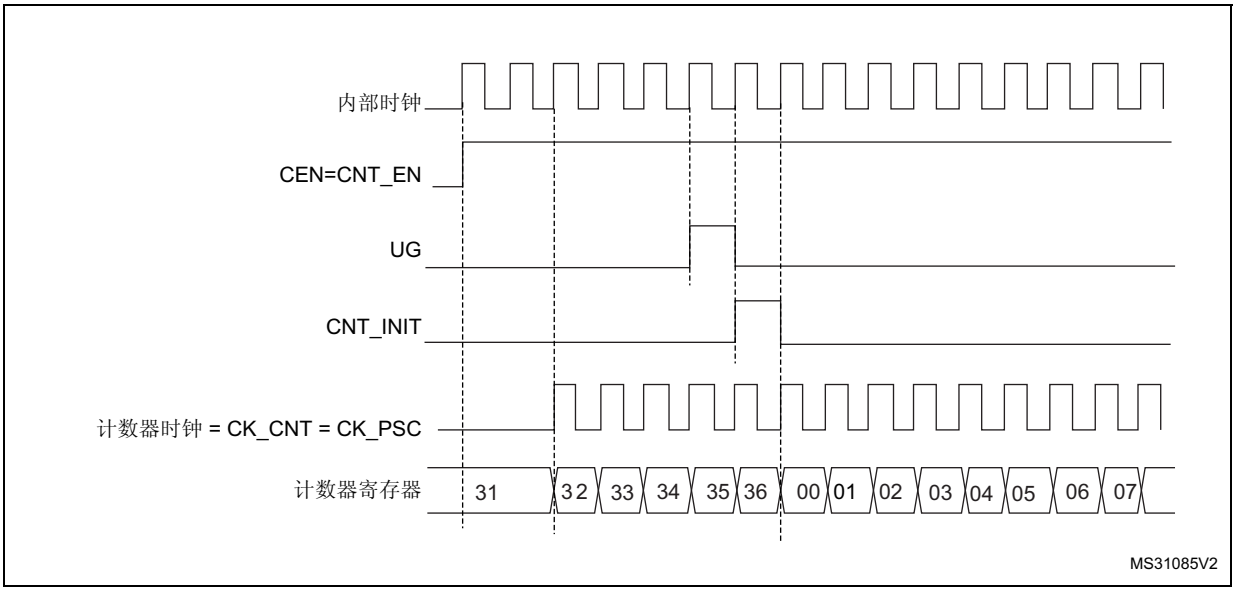
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚 (TIx)
- 外部时钟模式 2：外部触发输入 (ETR)
- 内部触发输入 (ITRx)：使用一个定时器作为另一个定时器的预分频器，例如可以将定时器 X 配置为定时器 Y 的预分频器。更多详细信息，请参见*第 2026 页的：将一个定时器用作另一个定时器的预分频器*。

内部时钟源 (CK_INT)

如果禁止从模式控制器（TIMx_SMCR 寄存器中 SMS=000），则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 415 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

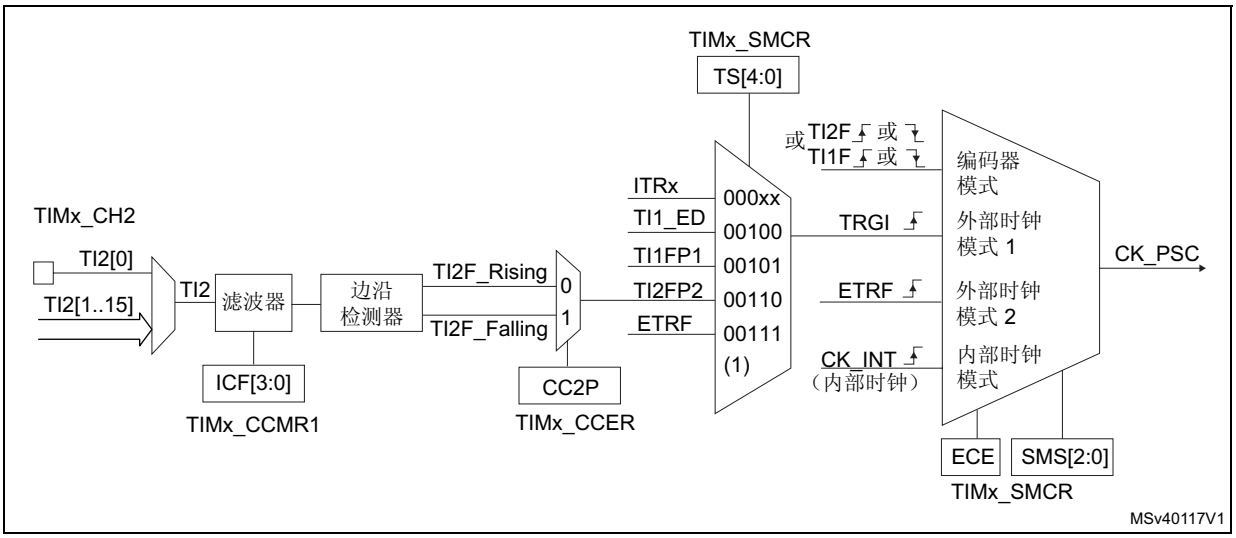
图 415. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 416. TI2 外部时钟连接示例



1. 代码范围是从 01000 到 11111：ITRy。

例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2x 源（内部或外部）。
2. 通过在 TIMx_CCMR1 寄存器中写入 CC2S=“01”来配置通道 2，使其能够检测 TI2 输入的上升沿。
3. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。

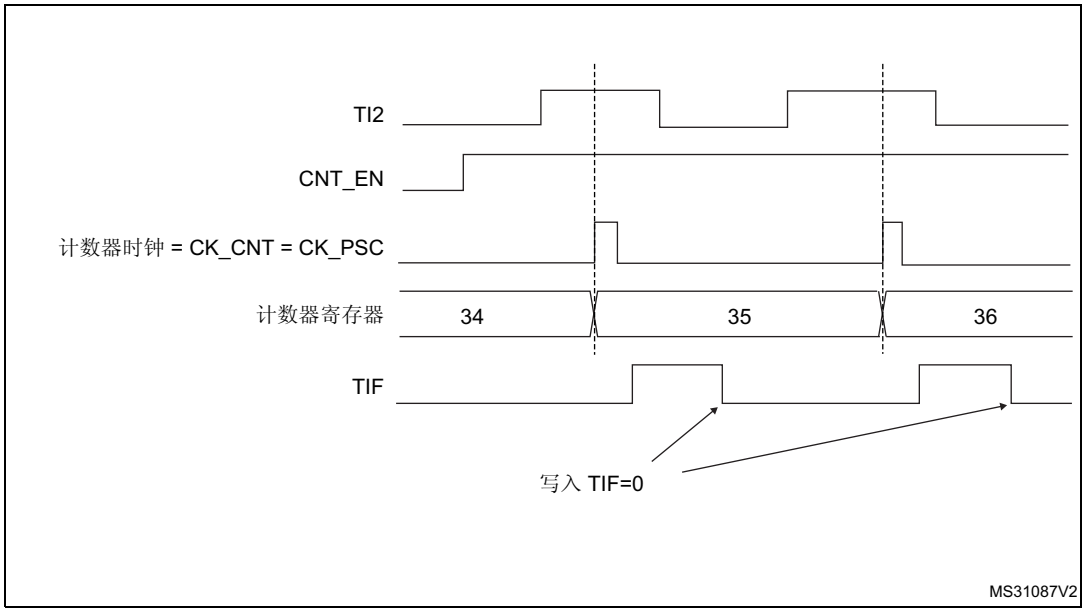
注： *由于捕获预分频器不用于触发操作，因此无需对其进行配置。*

4. 通过在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
5. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
6. 通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为输入源。
7. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 417. 外部时钟模式 1 下的控制电路



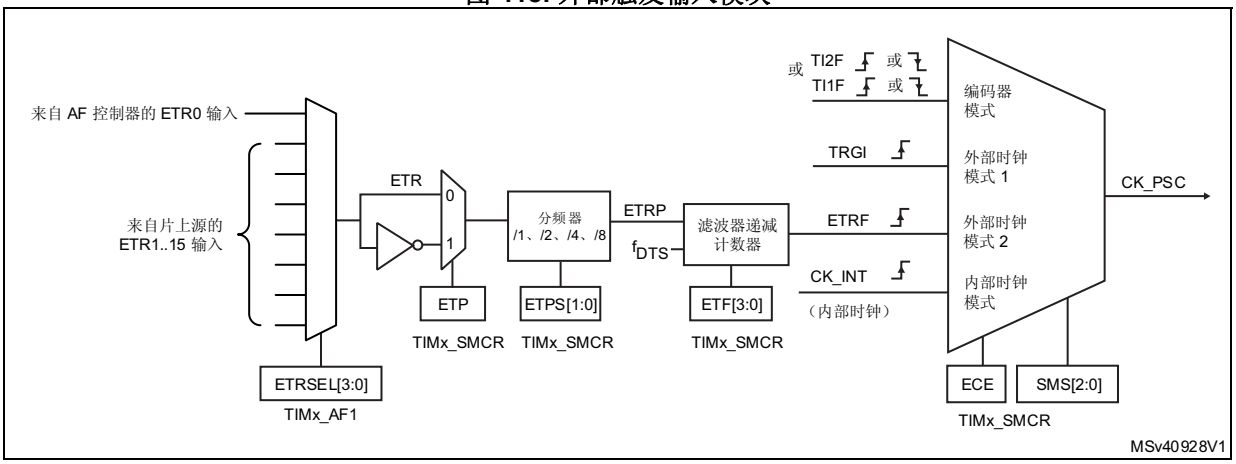
外部时钟源模式 2

通过在 TIMx_SMCR 寄存器中写入 ECE=1 可选择此模式。

计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。

图 418 简要介绍了外部触发输入模块。

图 418. 外部触发输入模块



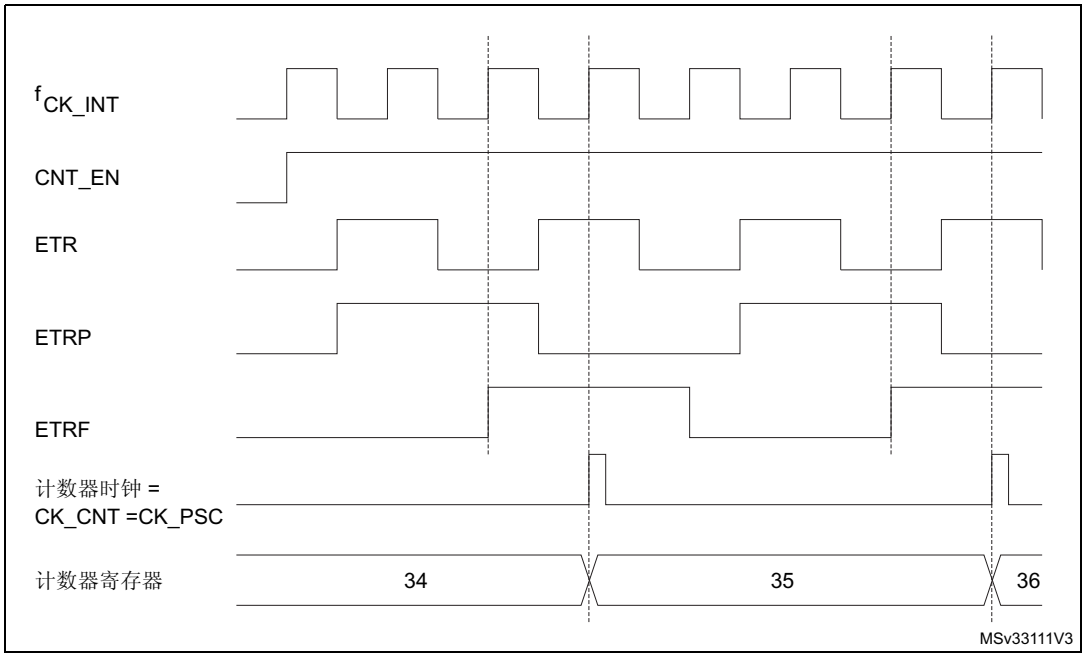
例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

1. 使用 TIMx_AF1 寄存器中的 ETRSEL[3:0] 位选择正确的 ETR 源（内部或外部）。
2. 由于此例中不需滤波器，因此在 TIMx_SMCR 寄存器中写入 ETF[3:0]=0000。
3. 通过在 TIMx_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
4. 通过在 TIMx_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
5. 通过在 TIMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
6. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。

ETR 的上升沿与实际计数器时钟之间的延迟是由于 ETRP 信号的重新同步电路引起的。因此，计数器能够正确捕获的最大频率不超过 TIMxCLK 频率的 ¼。当 ETRP 信号较快时，用户应通过适当的 ETPS 预分频器设置对外部信号进行分频。

图 419. 外部时钟模式 2 下的控制电路



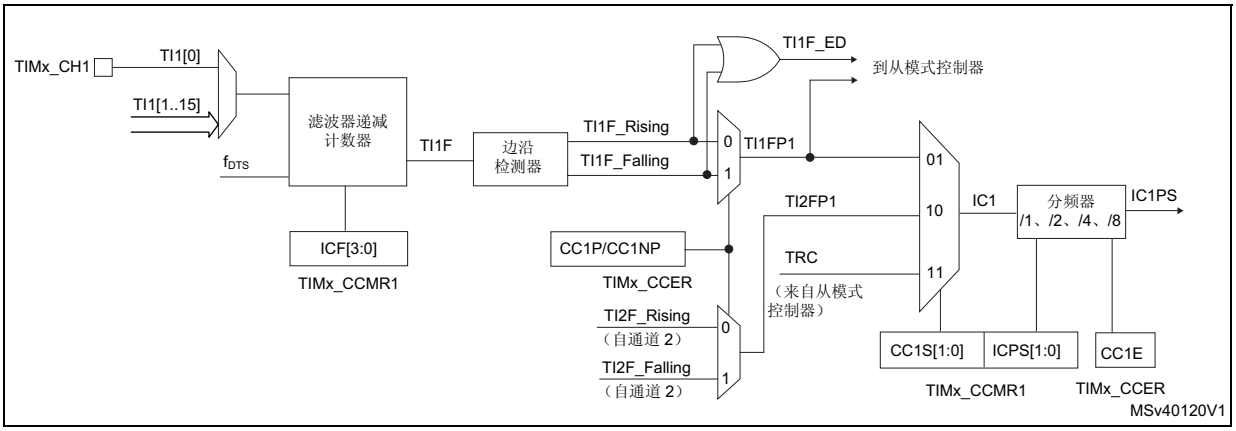
39.3.4 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入单元（数字滤波、多路复用和预分频器）和一个输出单元（比较器和输出控制）构建而成。

下图简要介绍了一路捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 420. 捕获/比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 421. 捕获/比较通道 1 主电路

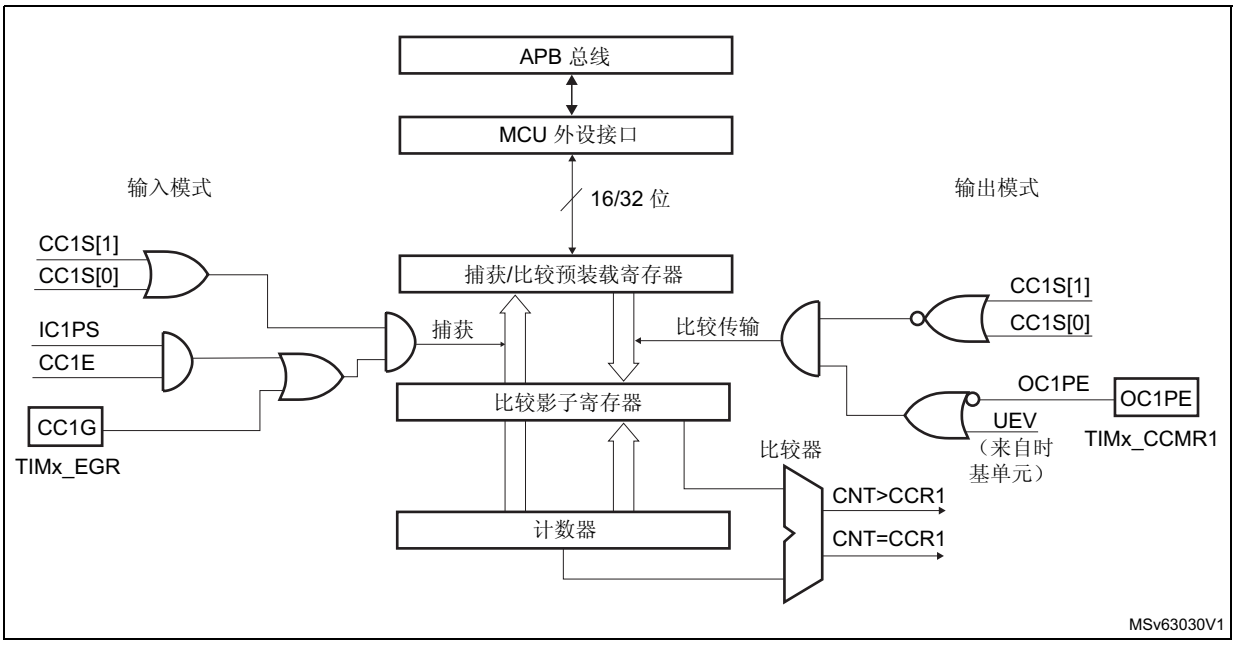
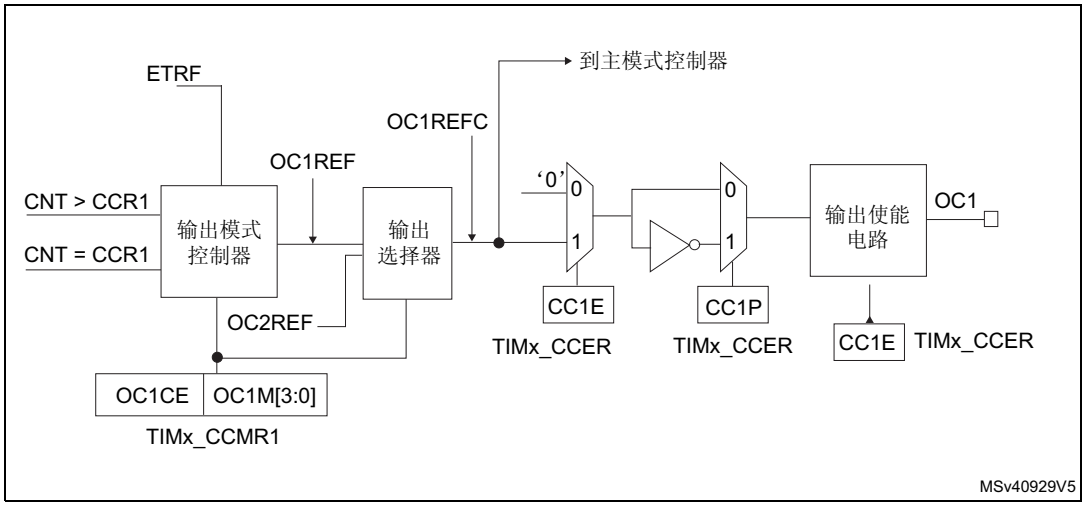


图 422. 捕获/比较通道的输出阶段（通道 1）



捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。读写访问的始终是预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

39.3.5 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将过捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件向 CCxIF 写入 0 使其清零，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择有效输入：TIMx_CCR1 必须关联到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
3. 根据关联到定时器的信号，对相关输入滤波参数进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设输入信号发生翻转时，信号需要 5 个内部时钟周期才能稳定，则必须设置滤波时间大于 5 个内部时钟周期。若在 TI1 上连续 8 次检测到新电平采样值（以 f_{DTS} 频率采样）判定沿跳变合法，则向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
4. 通过向 TIMx_CCER 寄存器中的 CC1P 位和 CC1NP 位写入 000，选择 TI1 通道的有效跳变沿（本例中为上升沿）。
5. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入 00）。
6. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
7. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理过捕获，建议在读取捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的过捕获信息。

注： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。*

39.3.6 PWM 输入模式

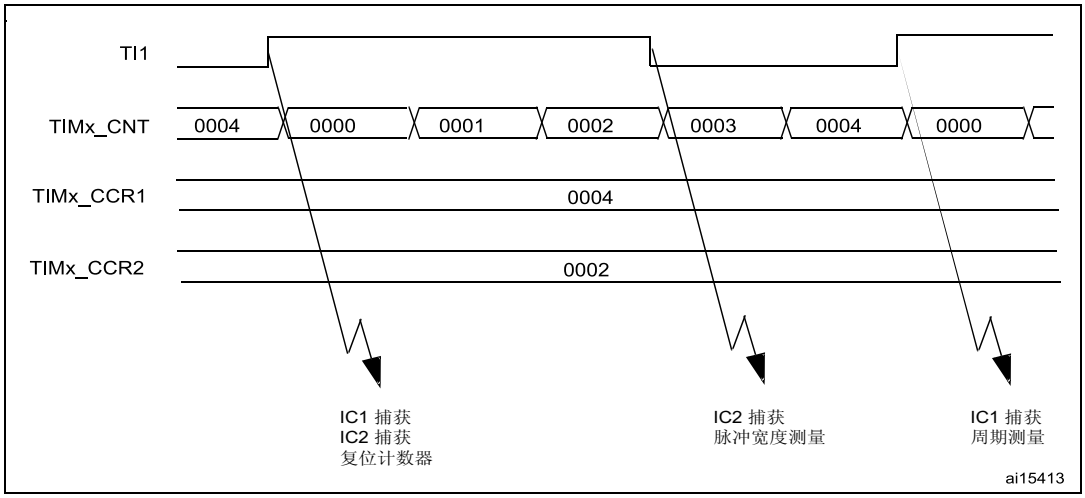
此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
3. 选择 TI1FP1 的有效极性（同时用于 TIMx_CCR1 中的捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
4. 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10（选择 TI1）。
5. 选择 TI1FP2 的有效极性（用于 TIMx_CCR2 中的捕获）：向 CC2P 位写入“1”，向 CC2NP 位写入“0”（下降沿有效）。
6. 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入 00101（选择 TI1FP1）。
7. 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入 100。
8. 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 423. PWM 输入模式时序



1. PWM 输入模式只能与 TIMx_CH1/TIMx_CH2 信号配合使用，因为只有 TI1FP1 和 TI2FP2 与从模式控制器相连。

39.3.7 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（OCxREF 和 OCx）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号 (ocxref/OCx) 强制设置为有效电平，只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入 101。ocxref 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如：CCxP=0（OCx 高电平有效）=> OCx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 100，可将 ocxref 信号强制设置为低电平。

当然，即使强制输出模式下，TIMx_CCRx 影子寄存器与计数器之间的比较仍然会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。输出比较模式一节对此进行了介绍。

39.3.8 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平 (OCXM=000)，也可设置为有效电平 (OCXM=001)、无效电平 (OCXM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应使能位（TIMx_DIER 寄存器的 CCxDE 位，TIMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

使用 TIMx_CCMRx 寄存器中的 OCxPE 位，可将 TIMx_CCRx 寄存器配置为带或不带预装功能。

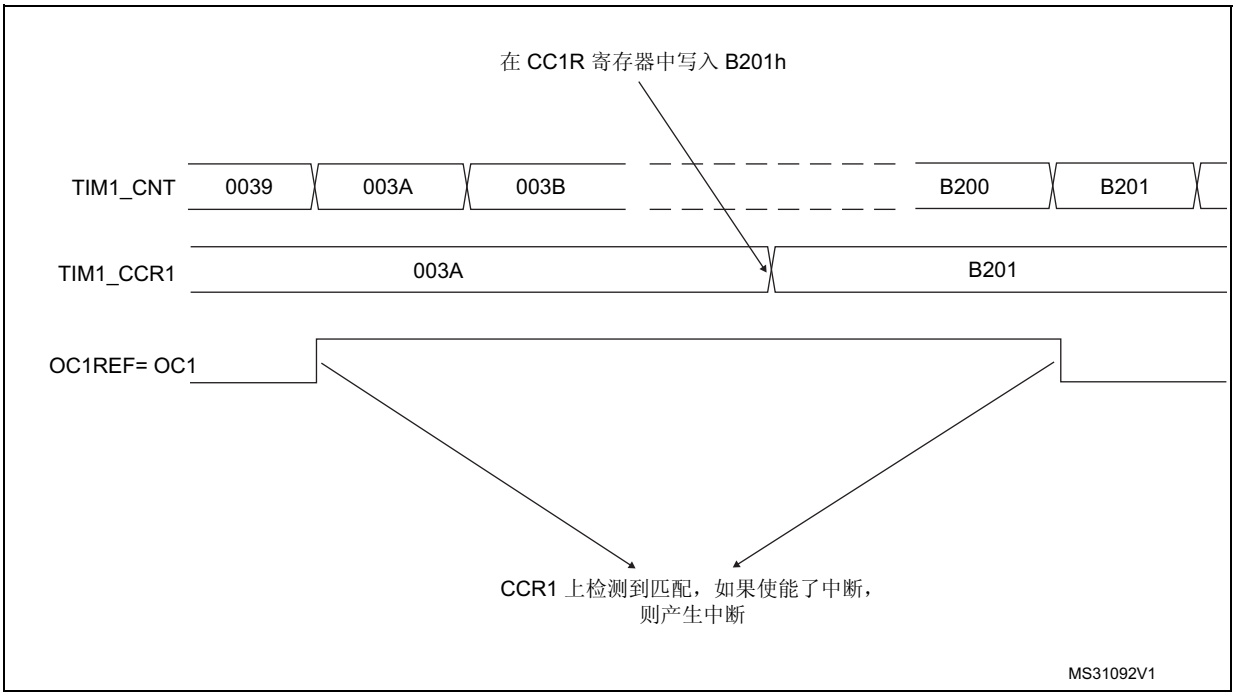
在输出比较模式下，更新事件 UEV 对 ocxref 和 OCx 输出毫无影响。时间分辨率可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断和/或 DMA 请求，将 CCxIE 位和/或 CCxDE 位置 1。
4. 选择输出模式。例如，当 CNT 与 CCRx 匹配、未使用预装载 CCRx 并且 OCx 使能且为高电平有效时，必须写入 OCxM=011、OCxPE=0、CCxP=0 和 CCxE=1 来翻转 OCx 输出引脚。
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装功能（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIMx_CCRx 影子寄存器）。*图 424* 给出了一个示例。

图 424. 输出比较模式，翻转 OC1



39.3.9 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIMx_ARR 寄存器值决定，其占空比则由 TIMx_CCRx 寄存器值决定。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 110（PWM 模式 1）或 111（PWM 模式 2），可以独立选择各通道（每个 OCx 输出对应一个 PWM）的 PWM 模式。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

由于只有在发生更新事件时预装载寄存器内容才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。OCx 输出通过将 TIMx_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息，请参见 TIMx_CCERx 寄存器说明。

在 PWM 模式（1 或 2）下，TIMx_CNT 与 TIMx_CCRx 进行实时比较，以确定是 TIMx_CCRx ≤ TIMx_CNT 还是 TIMx_CNT ≤ TIMx_CCRx（取决于计数器计数方向）。不过，为符合 OCREF_CLR 功能（在下一个 PWM 周期之前，ETR 信号上的一个外部事件能够清除 OCREF），OCREF 信号仅在以下情况下变为有效状态：

- 比较结果发生改变，或
- 输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）从“冻结”配置（不进行比较，OCxM=“000”）切换为任一 PWM 模式（OCxM=“110”或“111”）。

定时器运行期间，可以通过软件强制 PWM 输出。

根据 TIMx_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

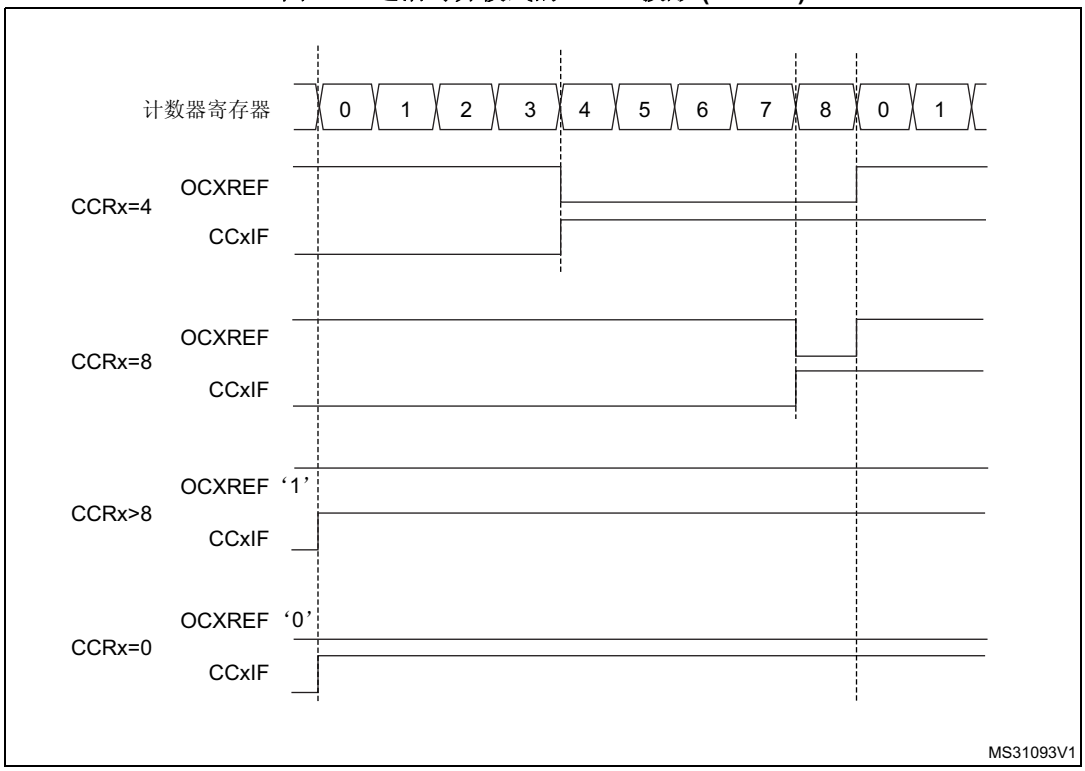
PWM 边沿对齐模式

递增计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低时执行递增计数。请参见*第 1992 页的递增计数模式*。

以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxREF 保持为“0”。*图 425* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 425. 边沿对齐模式的 PWM 波形 (ARR=8)



递减计数配置

当 TIMx_CR1 寄存器中的 DIR 位为高时执行递减计数。请参见*第 1995 页的递减计数模式*。

在 PWM 模式 1 下，只要 TIMx_CNT>TIMx_CCRx，参考信号 ocxref 便为低电平，否则为高电平。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重载值，则 ocxref 保持为 100%。此模式下不可能产生 PWM。

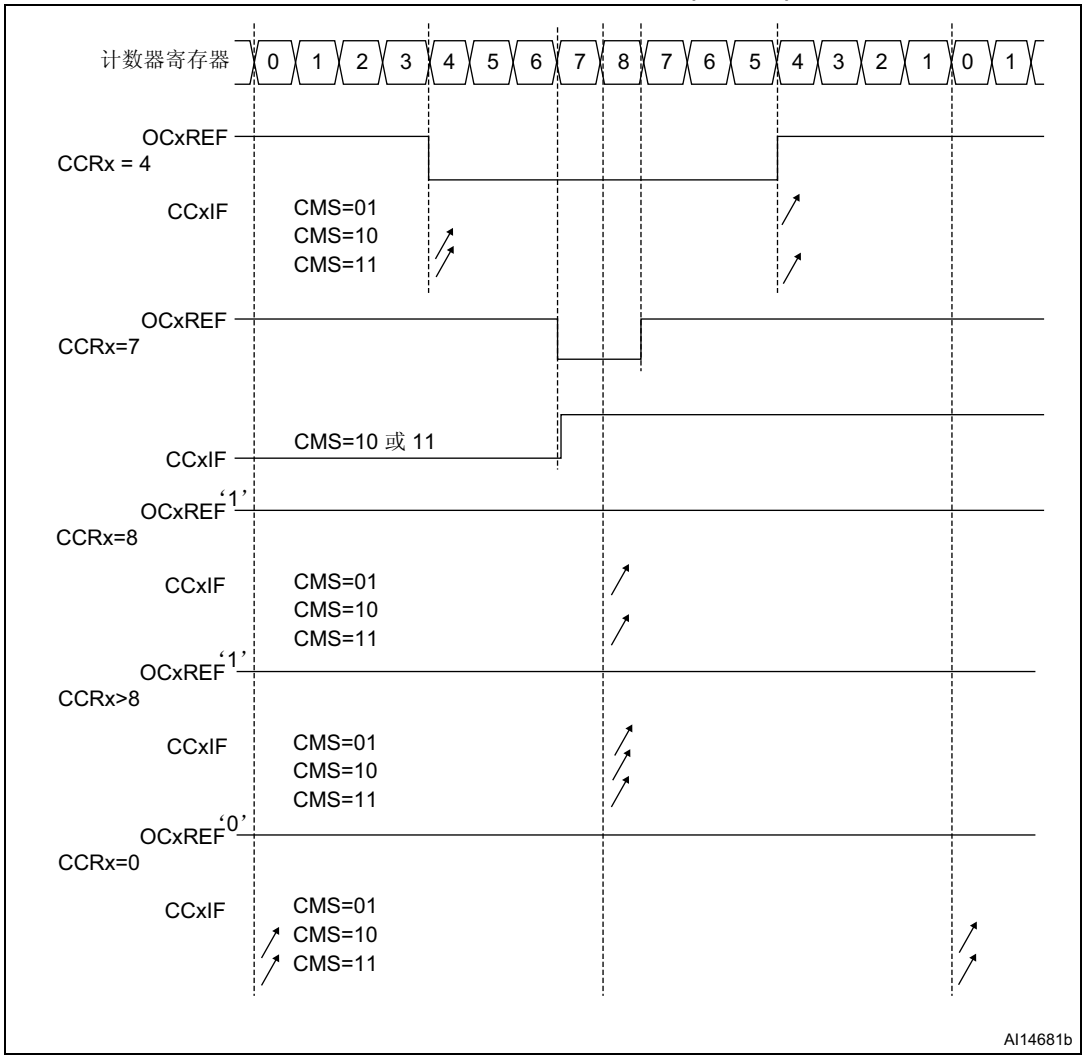
PWM 中心对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时（其余所有配置对 ocxref/OCx 信号具有相同的作用），中心对齐模式生效。根据 CMS 位的配置，可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIMx_CR1 寄存器中的方向位 (DIR) 由硬件更新，不得通过软件更改。请参见*第 1998 页的中心对齐模式（递增/递减计数）*。

图 426 显示了中心对齐模式的 PWM 波形，在此例中：

- TIMx_ARR=8，
- PWM 模式为 PWM 模式 1，
- 在根据 TIMx_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1。

图 426. 中心对齐模式 PWM 波形 (ARR=8)



中心对齐模式使用建议：

- 启动中心对齐模式时将使用当前的递增 / 递减计数配置。这意味着计数器将根据写入 TIMx_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得通过软件同时修改 DIR 和 CMS 位。
- 不建议在中心对齐模式下对计数器执行写操作，否则将发生意外错误。尤其是：
  - 如果写入计数器的值大于自动重载值 (TIMx_CNT>TIMx_ARR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
  - 如果向计数器写入 0 或 TIMx_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIMx_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

39.3.10 不对称 PWM 模式

在不对称模式下，生成的两个中心对齐 PWM 信号间允许存在可编程相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和相移则由一对 TIMx_CCRx 寄存器确定。两个寄存器分别控制递增计数和递减计数期间的 PWM，这样每半个 PWM 周期便会调节一次 PWM：

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

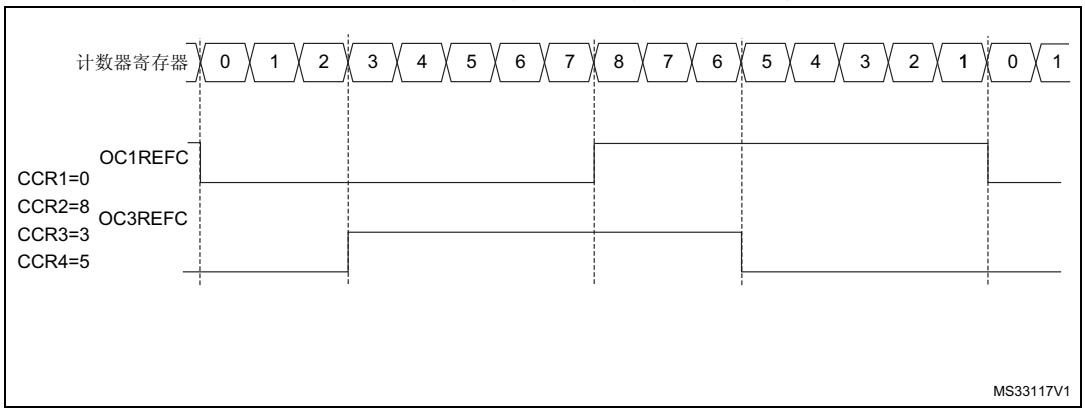
两个通道可以独立选择不对称 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1110”（不对称 PWM 模式 1）或“1111”（不对称 PWM 模式 2）。

注： *出于兼容性原因，OCxM[3:0] 位字段分为两部分，最高有效位与最低有效的 3 位不相邻。*

给定通道用作不对称 PWM 通道时，也可使用其辅助通道。例如，如果通道 1 上产生 OC1REFC 信号（不对称 PWM 模式 1），则由于不对称 PWM 模式 2 的原因，通道 2 上可输出 OC2REF 信号或 OC2REFC 信号。

图 427 显示了不对称 PWM 模式下可以产生的信号示例（通道 1 到通道 4 在不对称 PWM 模式 1 下配置）。

图 427. 50% 占空比时产生的 2 个相移 PWM 信号



39.3.11 组合 PWM 模式

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的“逻辑或”运算或者“逻辑与”运算组合组成。

——OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制

——OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

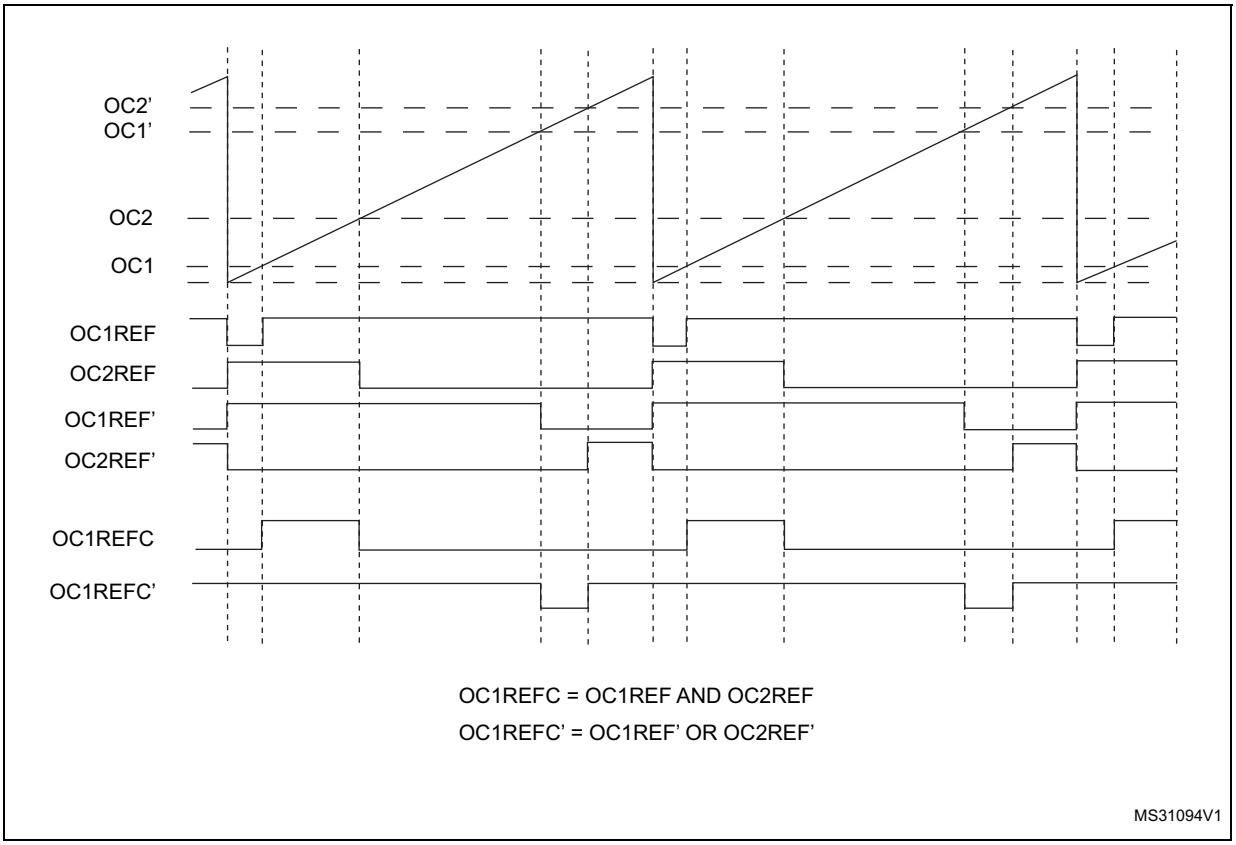
当给定通道用作组合 PWM 通道时，其辅助通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注： *出于兼容性原因，OCxM[3:0] 位字段分为两部分，最高有效位与最低有效的 3 位不相邻。*

图 428 显示了不对称 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 配置为组合 PWM 模式 2，
- 通道 2 配置为 PWM 模式 1，
- 通道 3 配置为组合 PWM 模式 2，
- 通道 4 配置为 PWM 模式 1

图 428. 通道 1 和通道 3 上的组合 PWM 模式



39.3.12 发生外部事件时清除 OCxREF 信号

对于给定通道，在 ocref_clr_int 输入上施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置 1），可将 OCxREF 信号清零。OCxREF 信号将保持低电平，直到在下一 PWM 周期再次变为有效状态。该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。

ocref_clr_int 连接到 ETRF 信号（滤波后的 ETR）。

对于给定通道，在 ETRF 输入施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置 1），可将 OCxREF 信号复位。OCxREF 信号将保持低电平，直到在下一 PWM 周期再次变为有效状态。

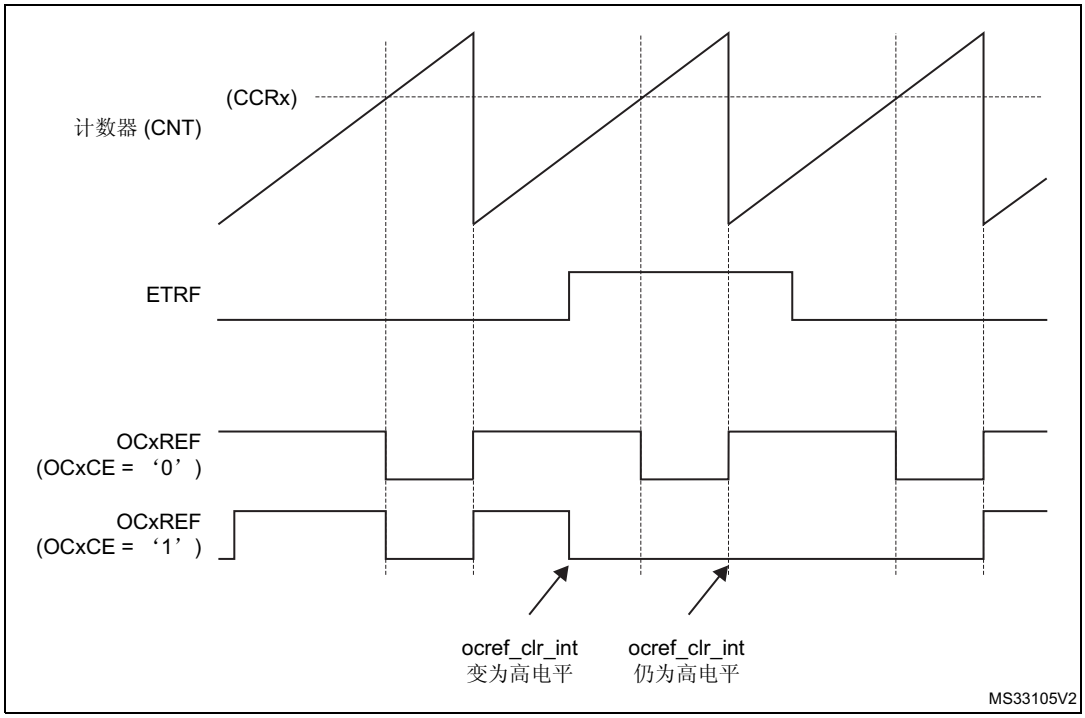
该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。

例如，OCxREF 信号可以连接到比较器的输出，用于控制电流。此时，ETR 必须配置如下：

1. 必须关闭外部触发预分频器：TIMx_SMCR 寄存器中的 ETPS[1:0] 位清为 00。
2. 必须禁止外部时钟模式 2：TIM1_SMCR 寄存器中的 ECE 位清为 0。
3. 外部触发极性 (ETP) 和外部触发滤波器 (ETF) 可根据应用需要进行配置。

图 429 对比了不同使能位 OCxCE 值的情况，显示了当 ETRF 输入变为高电平时 OCxREF 信号的行为。在本例中，定时器 TIMx 编程为 PWM 模式。

图 429. 清除 TIMx 的 OCxREF



注： *如果 PWM 的占空比为 100% (CCRx>ARR)，则下次计数器上溢时会再次使能 OCxREF。*

39.3.13 单脉冲模式

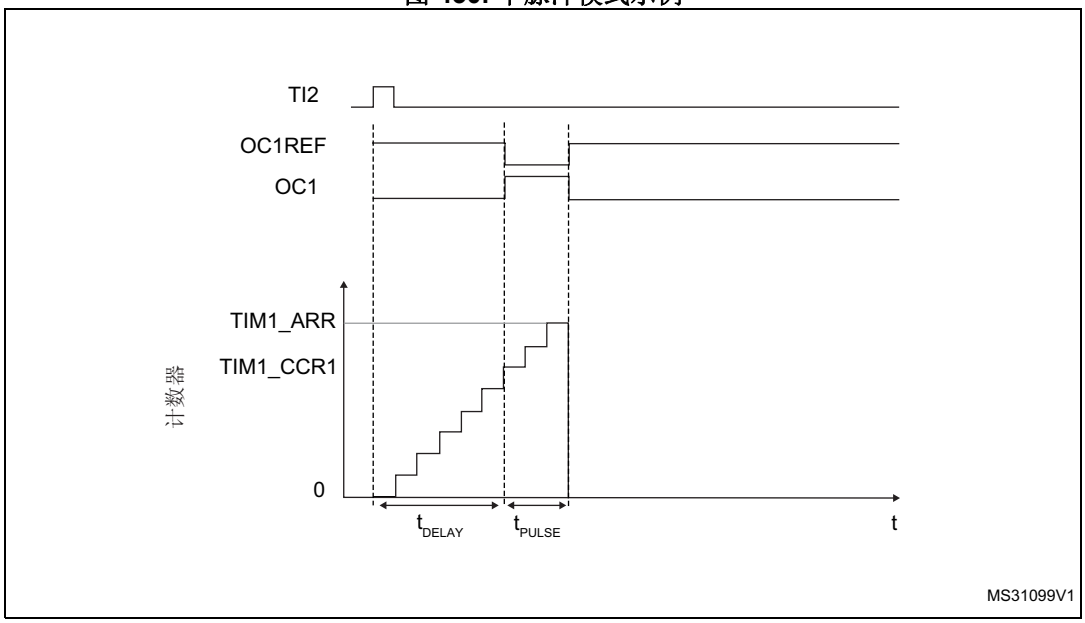
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。通过将 TIMx_CR1 寄存器中的 OPM 位置 1 选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

- CNT < CCRx ≤ ARR（特别注意，0 < CCRx），

图 430. 单脉冲模式示例



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2x 源（内部或外部）。
2. 在 TIMx_CCMR1 寄存器中写入 CC2S=01，将 TI2FP2 映射到 TI2。
3. 在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
4. 在 TIMx_SMCR 寄存器中写入 TS=00110，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
5. 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），使用 TI2FP2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，必须通过在 TIMx_CCMR1 寄存器中写入 OC1M=111，来使能 PWM 模式 2。可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

在本例中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此必须向 TIMx_CR1 寄存器的 OPM 位写入 1，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIMx_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特例：OCx 快速使能：

在单脉冲模式下，TIx 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟（t_{DELAY} 最小值）。

如果要输出延迟时间最短的波形，可以将 TIMx_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 OCxRef（和 OCx）对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

39.3.14 可再触发单脉冲模式

该模式可以启动计数器来响应激励信号，并且能产生长度可编程的脉冲，但与*第 39.3.13 节*所描述的不可再触发单脉冲模式间存在以下差别：

- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长

定时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0] = “1000”（组合复位 + 触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0] 位设置为“1000”或“1001”。

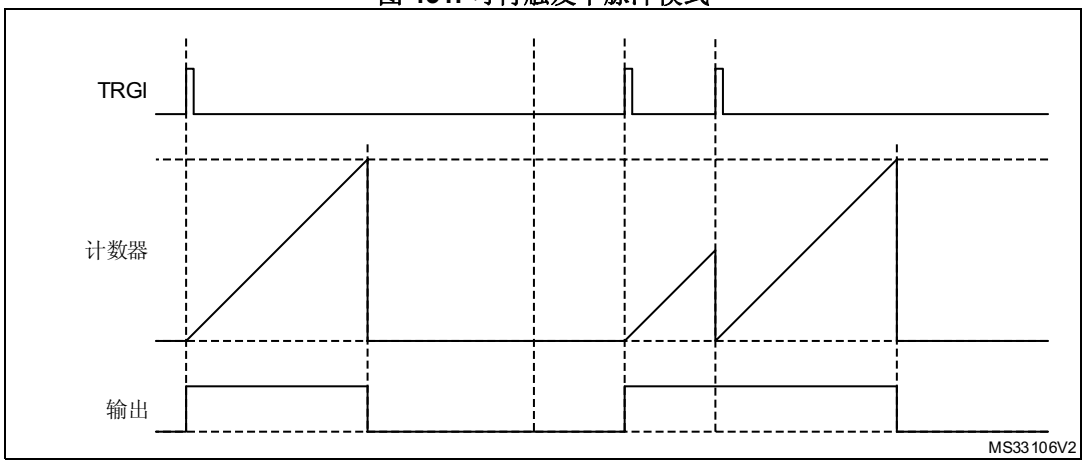
定时器配置为递增计数模式时，相应的 CCRx 必须置 0（ARR 寄存器设置脉冲长度）。如果定时器配置为递减计数模式，CCRx 必须高于或等于 ARR。

注： *在可再触发单脉冲模式下，CCxIF 标志没有意义。*

出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位字段分为两部分，最高有效位与最低有效的 3 位不相邻。

此模式不能与中心对齐 PWM 模式一起使用。在 TIMx_CR1 中必须设置 CMS[1:0] = 00。

图 431. 可再触发单脉冲模式



39.3.15 编码器接口模式

选择编码器接口模式时，如果计数器仅在 TI2 边沿处计数，在 TIMx_SMCR 寄存器中写入 SMS=001；如果计数器仅在 TI1 边沿处计数，写入 SMS=010；如果计数器在 TI1 和 TI2 边沿处均计数，则写入 SMS=011。

通过编程 TIMx_CCER 寄存器的 CC1P 和 CC2P 位，选择 TI1 和 TI2 极性。CC1NP 和 CC2NP 必须保持清零。如果需要，还可对输入滤波器进行编程。CC1NP 和 CC2NP 必须保持低电平。

TI1 和 TI2 两个输入用于连接增量编码器。请参见 *表 354*。如果使能计数器（在 TIMx_CR1 寄存器的 CEN 位中写入“1”），则计数器的时钟由 TI1FP1 或 TI2FP2 上的每次有效信号转换提供。TI1FP1 和 TI2FP2 是 TI1 和 TI2 进行输入滤波和极性选择后的信号，如果不进行滤波和反相，则 TI1FP1=TI1，TI2FP2=TI2。将根据两个输入的信号转换序列，产生计数脉冲和方向信号。根据该信号转换序列，计数器相应递增或递减计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应修改。任何输入（TI1 或 TI2）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 TI1 或 TI2 边沿处计数，还是同时在 TI1 和 TI2 处计数。

编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 TIMx_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIMx_ARR。同样，捕获、比较、预分频器、触发输出功能继续正常工作。

在此模式下，计数器会根据正交编码器的速度和方向自动进行修改，因此，其内容始终表示编码器的位置。计数方向对应于所连传感器的旋转方向。下表汇总了可能的组合（假设 TI1 和 TI2 不同时切换）。

表 354. 计数方向与编码器信号的关系

有效边沿	相反信号的电平（TI1FP1 对应 TI2，TI2FP2 对应 TI1）	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降
仅在 TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在 TI2 处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在 TI1 和 TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

外部增量编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将编码器的差分输出转换为数字信号。这样大幅提高了抗噪声性能。用于指示机械零位的第三个编码器输出可与外部中断输入相连，用以触发计数器复位。

图 432 以计数器工作为例，说明了计数信号的生成和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S=01（TIMx_CCMR1 寄存器，TI1FP1 映射到 TI1 上）
- CC2S=01（TIMx_CCMR2 寄存器，TI2FP2 映射到 TI2 上）
- CC1P 和 CC1NP = “0”（TIMx_CCER 寄存器，TI1FP1 未反相，TI1FP1=TI1）
- CC2P 和 CC2NP = “0”（TIMx_CCER 寄存器，TI2FP2 未反相，TI2FP2= TI2）
- SMS=011（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）
- CEN = 1（TIMx_CR1 寄存器，使能计数器）

图 432. 编码器接口模式下的计数器工作示例

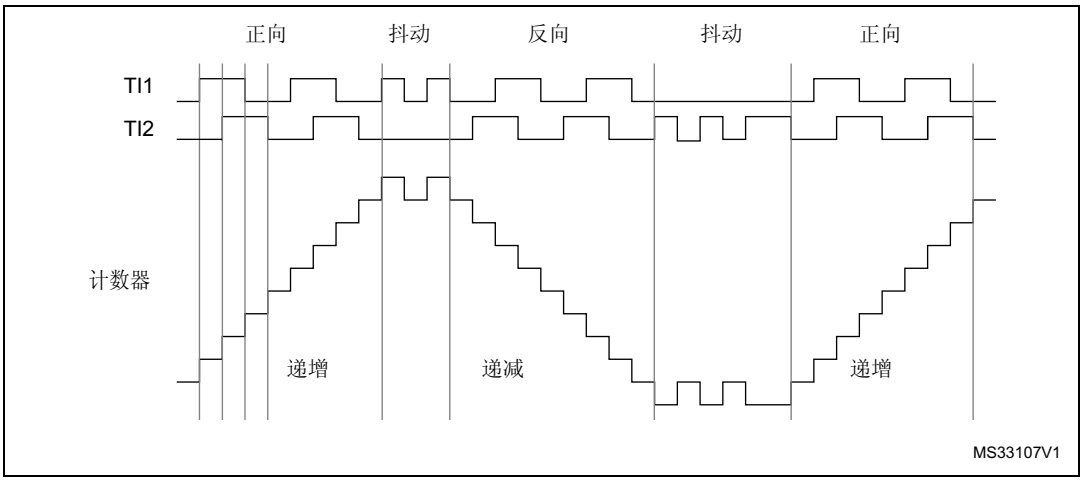
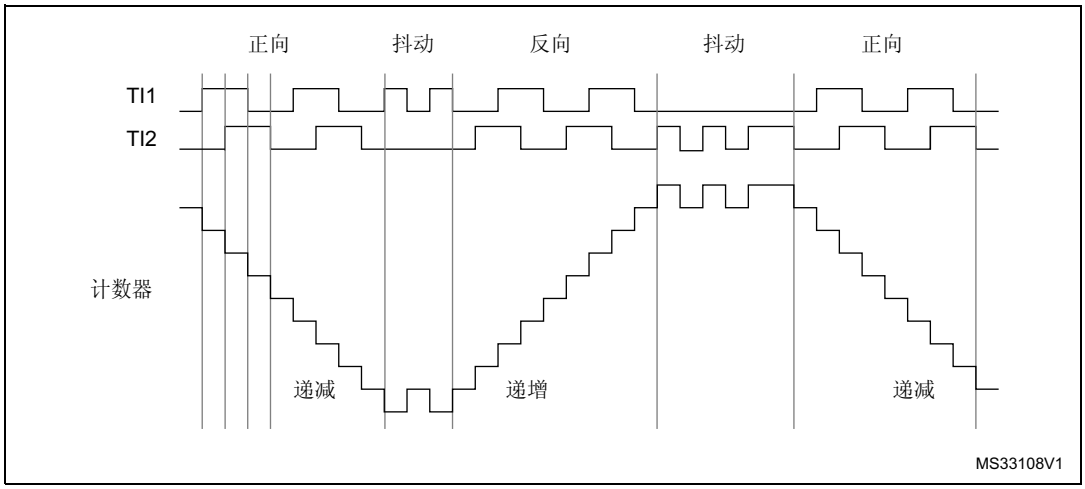


图 433 举例说明了 TI1FP1 极性反相时的计数器行为（除 CC1P=1 外，其他配置与上例相同）。

图 433. TI1FP1 极性反相时的编码器接口模式示例



定时器配置为编码器接口模式时，会提供传感器当前位置的相关信息。使用另一个配置为捕获模式的定时器测量两个编码器事件之间的周期，可获得动态信息（速度、加速度和减速度）。指示机械零位的编码器输出即可用于此目的。根据两个事件之间的时间间隔，还可定期读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕获信号必须为周期性信号，可以由另一个定时器产生）；还可以通过由实时时钟生成的 DMA 请求读取计数器值。

39.3.16 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 (UIF) 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可按原子方式读取计数器值以及由 UIFCPY 标志发出的更新事件情况。这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化角速度的计算。

UIF 和 UIFCPY 标志之间的置位没有延迟。

在 32 位定时器实现中，当 IUFREMAP 位置 1 时，计数器的位 31 在读访问时由 UIFCPY 标志覆盖（计数器的最高有效位只能在写模式下访问）。

39.3.17 定时器输入异或功能

借助 TIM1xx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 到 TIMx_CH3 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。

*第 1939 页的第 38.3.25 节：连接霍尔传感器*以连接霍尔传感器为例介绍了此功能。

39.3.18 定时器与外部触发同步

TIMx 定时器可在多种模式下与外部触发实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

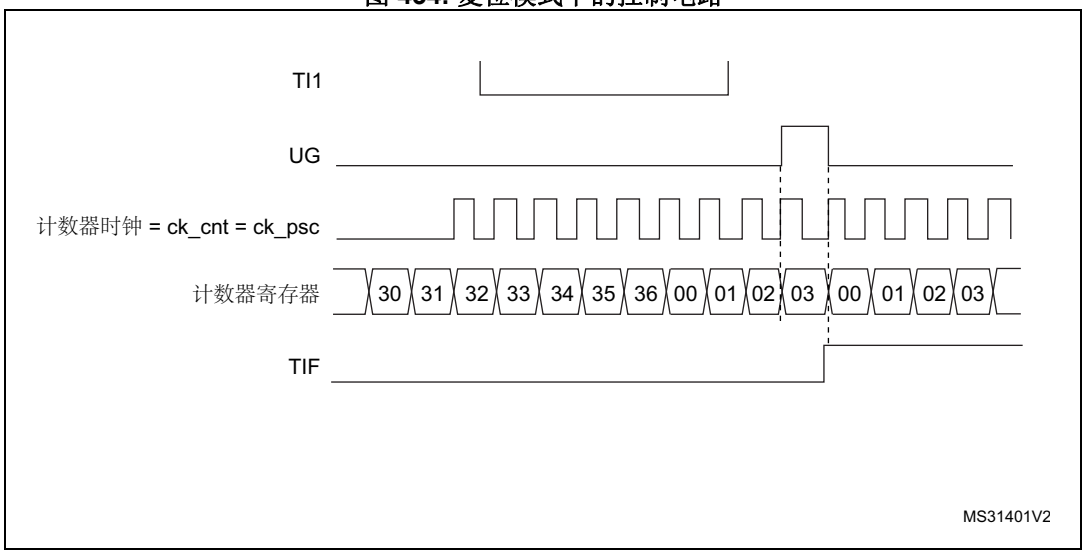
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

1. 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S = 01。在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP=0，验证极性（仅检测上升沿）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。通过在 TIMx_SMCR 寄存器中写入 TS= 00101 来选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 434. 复位模式下的控制电路



从模式：门控模式

输入信号的电平可用来使能计数器。

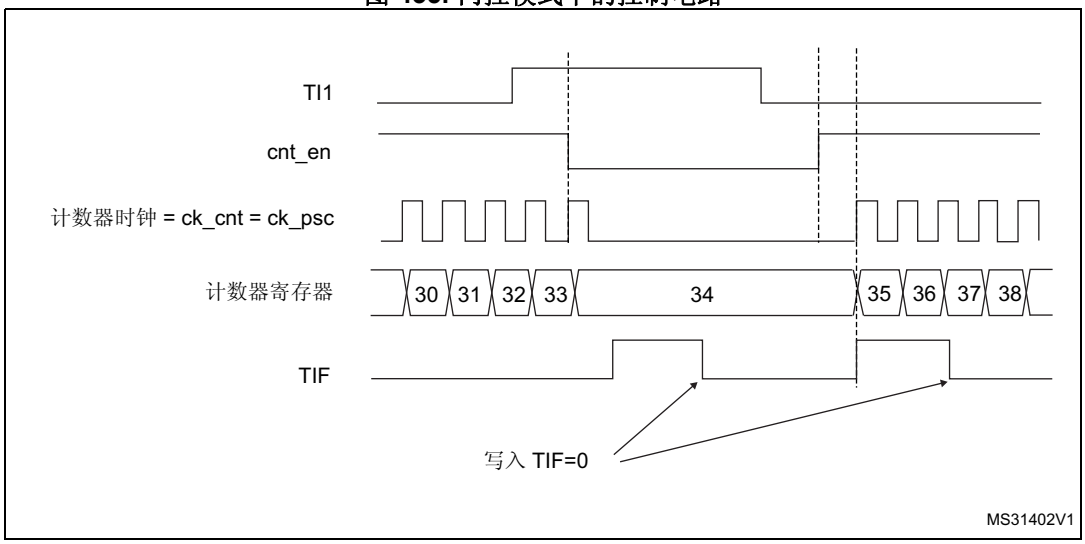
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

1. 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=1 和 CC1NP=0，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。通过在 TIMx_SMCR 寄存器中写入 TS= 00101 来选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 435. 门控模式下的控制电路



1. 由于门控模式作用于电平而非边沿，因此在门控模式下，“CCxP=CCxNP=1”（同时检测上升沿和下降沿）不发挥任何作用。

注： *由于门控模式作用于电平而非边沿，因此在门控模式下，“CCxP=CCxNP=1”（同时检测上升沿和下降沿）不发挥任何作用。*

从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

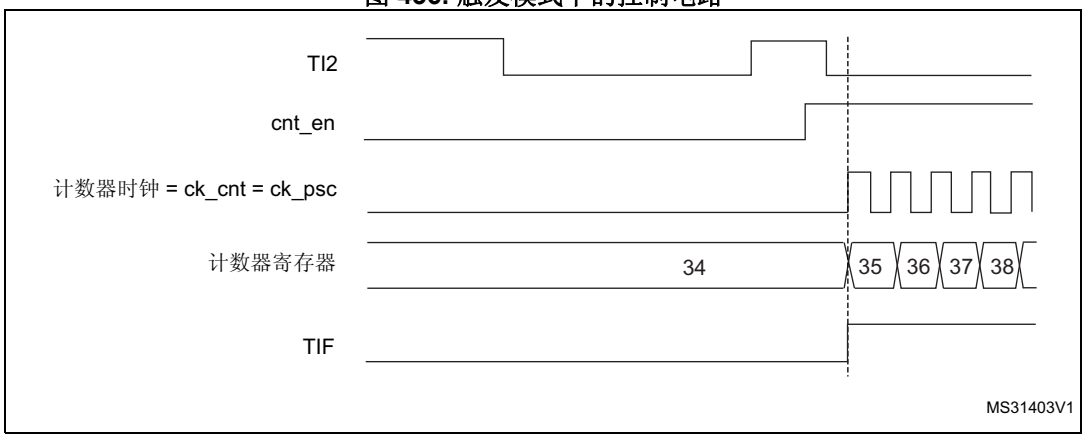
1. 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。

2. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 436. 触发模式下的控制电路



从模式：外部时钟模式 2 + 触发模式

外部时钟模式 2 可与另一种从模式（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下，ETR 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另一个输入作为触发输入。不建议通过 TIMx_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。

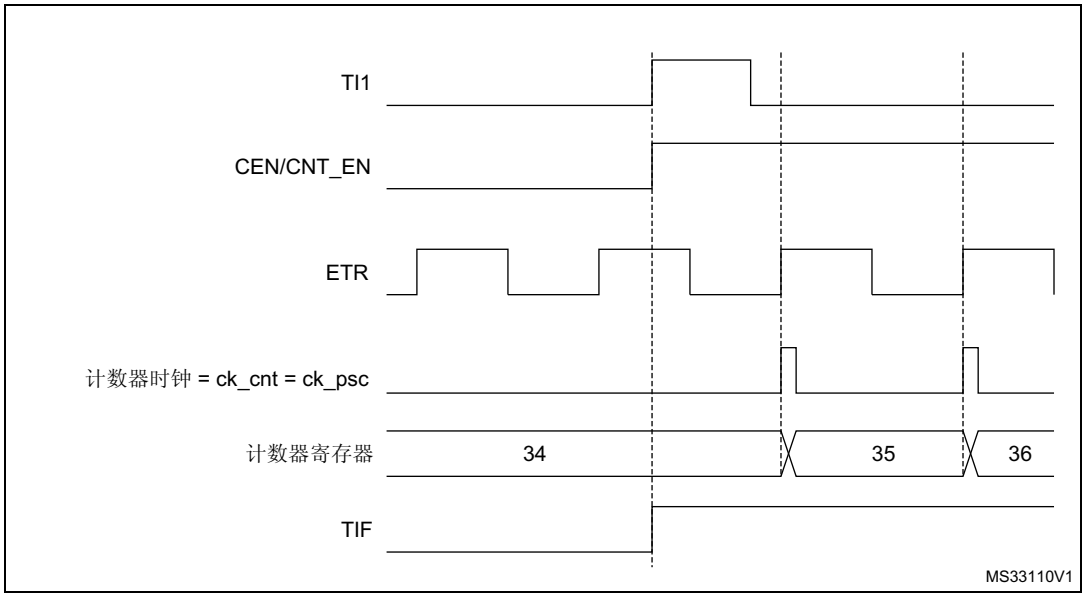
在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处递增：

1. 通过对 TIMx_SMCR 寄存器进行如下编程，配置外部触发输入电路：
   - ETF = 0000：无滤波器
   - ETPS = 00：禁止预分频器
   - ETP = 0：检测 ETR 的上升沿，并写入 ECE=1，以使能外部时钟模式 2。
2. 如下配置通道 1，以检测 TI 的上升沿：
   - IC1F=0000：无滤波器。
   - 由于捕获预分频器不用于触发操作，因此无需对其进行配置。
   - TIMx_CCMR1 寄存器中 CC1S=01，只选择输入捕获源。
   - TIMx_CCER 寄存器中 CC1P=0 且 CC1NP= 0，以确定极性（仅检测上升沿）。
3. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。通过在 TIMx_SMCR 寄存器中写入 TS= 00101 来选择 TI1 作为输入源。

TI1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 ETR 出现上升沿时计数。

ETR 信号的上升沿与实际计数器复位之间的延迟是由于 ETRP 输入的重新同步电路引起的。

图 437. 外部时钟模式 2 + 触发模式下的控制电路



39.3.19 定时器同步

TIMx 定时器从内部连接在一起，以实现定时器同步或链接。当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。

*图 438：主/从定时器示例*和*图 439：单通道定时器的主/从连接示例*简要介绍了触发选择和主模式选择框图。

图 438. 主/从定时器示例

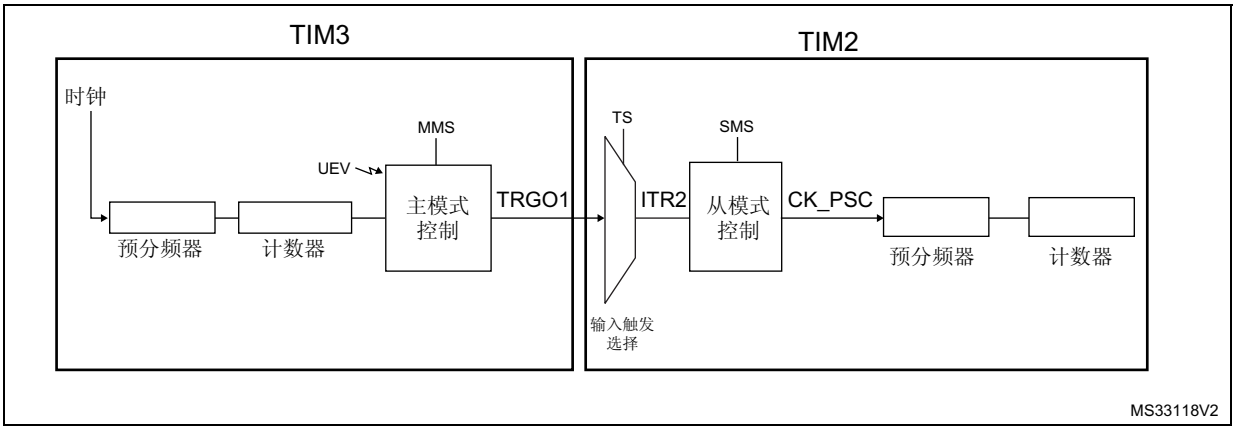
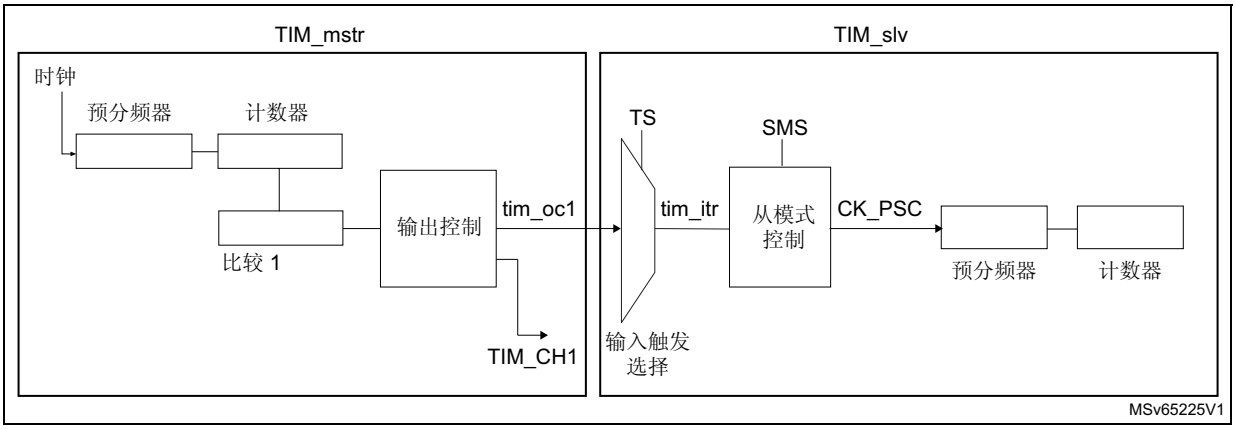


图 439. 单通道定时器的主/从连接示例



注： *单通道定时器（请参见图 439）没有主模式。但是，可以使用 OC1 输出信号来触发其他定时器（包括本文档其他部分所述的定时器）。通过查看关于设备上任何 TIMx_SMCR 寄存器的“TIMx 内部触发连接”表，可以确定哪些定时器可以作为目标从定时器。*
OC1 信号的脉冲宽度必须至少设为目标定时器时钟周期的 2 倍，以便确保从定时器能够检测到触发。
例如，如果目标定时器 CK_INT 时钟频率是源定时器的 1/4，则 OC1 的脉冲宽度必须至少是 8 个源定时器时钟周期。

将一个定时器用作另一个定时器的预分频器

例如，可以将 TIM3 配置为 TIM2 的预分频器。请参见图 438。为此：

1. 将 TIM3 配置为主模式，以便每次发生更新事件 UEV 时都输出一个周期性触发信号。如果在 TIM3_CR2 寄存器中写入 MMS=010，则每次生成更新事件时，TRGO 都会输出一个上升沿。
2. 要将 TIM3 的 TRGO 输出连接到 TIM2，必须将 TIM2 配置为从模式，使用 ITR2 作为内部触发。通过 TIM2_SMCR 寄存器中的 TS 位（写入 TS=00010）可对此进行选择。
3. 然后必须将从模式控制器设为外部时钟模式 1（在 TIM2_SMCR 寄存器中写入 SMS=111）。这样一来，TIM2 的时钟将由 TIM3 周期性触发信号的上升沿（与 TIM3 的计数器上溢对应）提供。
4. 最后必须通过将这两个定时器的相应 CEN 位（TIMx_CR1 寄存器）置 1 同时使能二者。

注： *如果选择 TIM3 的 OCx 信号作为触发输出 (MMS=1xx)，该信号的上升沿将用于驱动 TIM2 的计数器。*

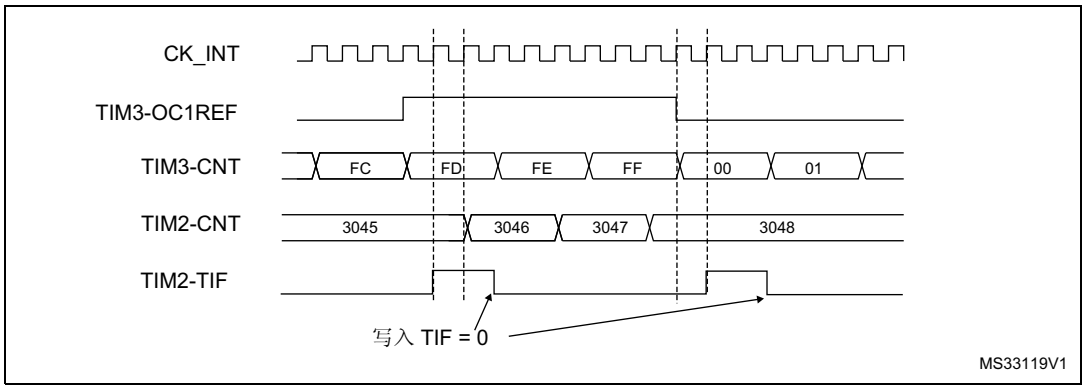
使用一个定时器使能另一个定时器

本例中通过定时器 3 的输出比较 1 来控制 TIM2 的使能。相关连接图，请参见图 438。仅当 TIM3 的 OC1REF 为高电平时，TIM2 才根据分频后的内部时钟进行计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频 ($f_{CK_CNT} = f_{CK_INT}/3$)。

1. 将 TIM3 配置为主模式，发送其输出比较 1 参考信号 (OC1REF) 作为触发输出（TIM3_CR2 寄存器中的 MMS=100）。
2. 配置 TIM3 的 OC1REF 波形（TIM3_CCMR1 寄存器）。
3. 配置 TIM2 以接收来自 TIM3 的输入触发（TIM2_SMCR 寄存器中的 TS=00010）。
4. 将 TIM2 配置为门控模式（TIM2_SMCR 寄存器中的 SMS=101）。
5. 通过向 CEN 位（TIM2_CR1 寄存器）写入“1”使能 TIM2。
6. 通过向 CEN 位（TIM3_CR1 寄存器）写入“1”启动 TIM3。

注：　计数器 2 的时钟与计数器 1 不同步，此模式仅影响 TIM2 的计数器使能信号。

图 440. 使用 TIM3 的 OC1REF 对 TIM2 实施门控控制

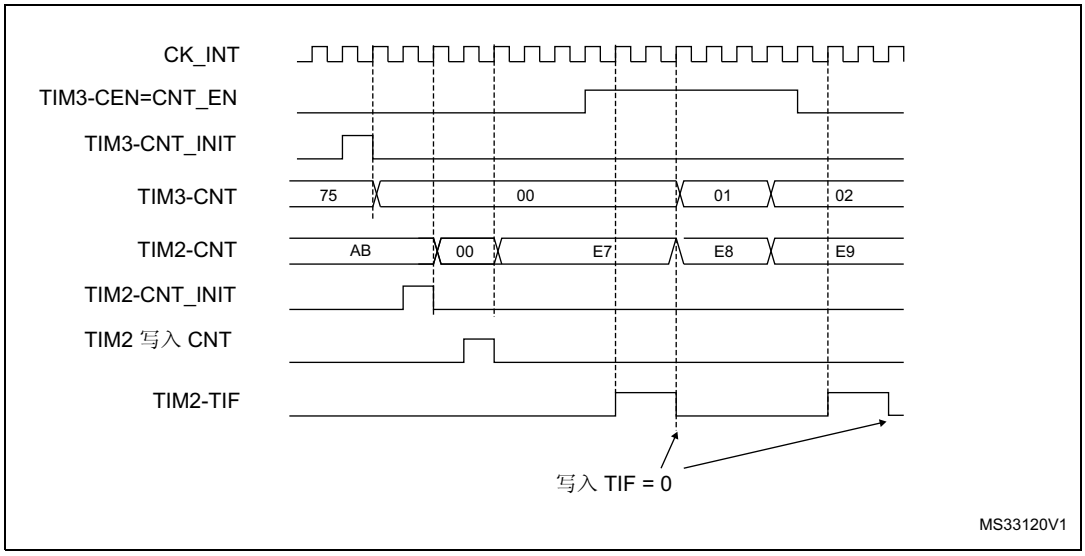


在*图 440* 的示例中，TIM2 的计数器和预分频器在启动前未进行初始化。因此从各自的当前值开始计数。启动定时器 TIM3 之前，通过复位这两个定时器可以从指定值开始计数。然后便可以在定时器计数器中写入任意值。两个定时器都可通过软件使用 TIMx_EGR 寄存器中的 UG 位轻松复位。

在下一示例中（请参见*图 441*），TIM3 与 TIM2 同步。TIM3 为主模式，从 0 开始计数。TIM2 为从模式，从 0xE7 开始计数。两个定时器的预分频比相同。通过向 TIM3_CR1 寄存器中的 CEN 位写入“0”来禁止 TIM3 时，TIM2 将停止：

1. 将 TIM3 配置为主模式，发送其输出比较 1 参考信号 (OC1REF) 作为触发输出（TIM3_CR2 寄存器中的 MMS=100）。
2. 配置 TIM3 的 OC1REF 波形（TIM3_CCMR1 寄存器）。
3. 配置 TIM2 以接收来自 TIM3 的输入触发（TIM2_SMCR 寄存器中的 TS=00010）。
4. 将 TIM2 配置为门控模式（TIM2_SMCR 寄存器中的 SMS=101）。
5. 通过向 UG 位（TIM3_EGR 寄存器）写入“1”复位 TIM3。
6. 通过向 UG 位（TIM2_EGR 寄存器）写入“1”复位 TIM2。
7. 通过在 TIM2 的计数器 (TIM2_CNTL) 中写入“0xE7”使 TIM2 初始化为 0xE7。
8. 通过向 CEN 位（TIM2_CR1 寄存器）写入“1”使能 TIM2。
9. 通过向 CEN 位（TIM3_CR1 寄存器）写入“1”启动 TIM3。
10. 通过向 CEN 位（TIM3_CR1 寄存器）写入“0”停止 TIM3。

图 441. 使用 TIM3 的使能信号对 TIM2 实施门控控制

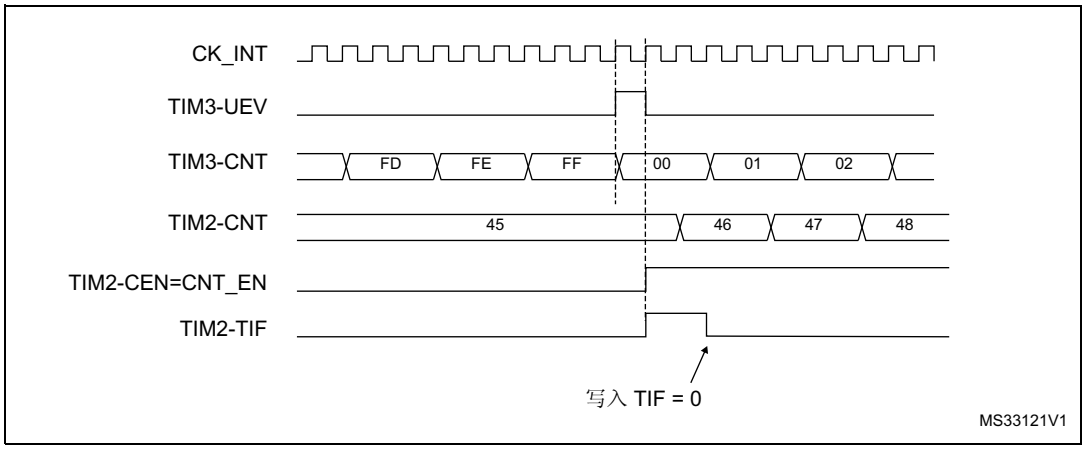


使用一个定时器启动另一个定时器

本例中，使用定时器 3 的更新事件使能定时器 2。相关连接图，请参见*图 438*。只要定时器 1 生成更新事件，定时器 2 便根据分频后的内部时钟从当前值（可以不为 0）开始计数。定时器 2 收到触发信号时，其 CEN 位自动置 1，并且计数器开始计数，直到向 TIM2_CR1 寄存器的 CEN 位写入“0”后停止计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频 ($f_{CK_CNT} = f_{CK_INT}/3$)。

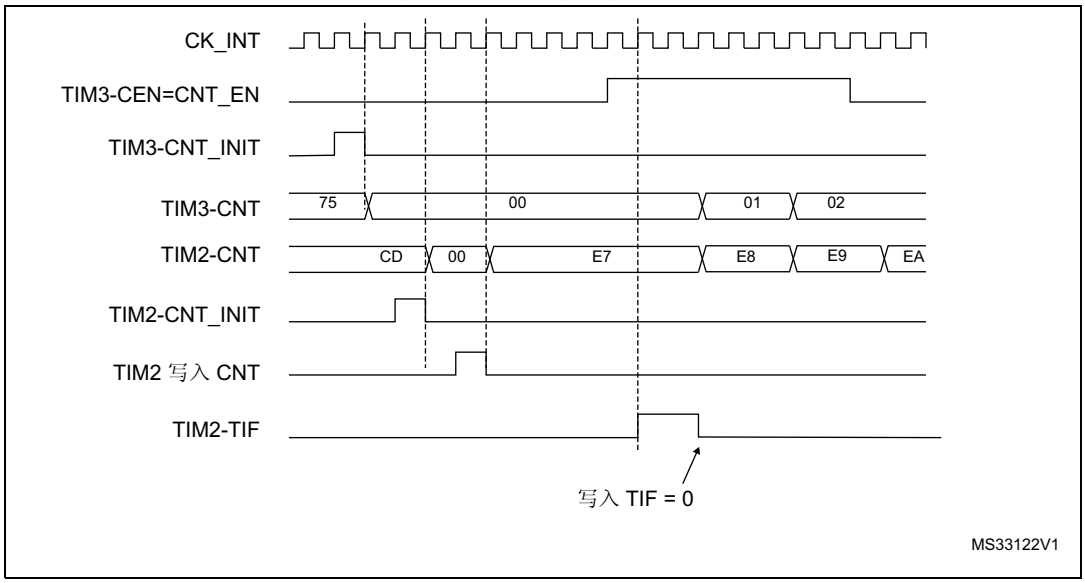
1. 将 TIM3 配置为主模式，发送其更新事件 (UEV) 作为触发输出（TIM3_CR2 寄存器中的 MMS=010）。
2. 配置 TIM3 的周期（TIM3_ARR 寄存器）。
3. 配置 TIM2 以接收来自 TIM3 的输入触发（TIM2_SMCR 寄存器中的 TS=00010）。
4. 将 TIM2 配置为触发模式（TIM2_SMCR 寄存器中的 SMS=110）。
5. 通过向 CEN 位（TIM3_CR1 寄存器）写入“1”启动 TIM3。

图 442. 使用 TIM3 的更新事件触发 TIM2



如上述示例所示，用户可以在开始计数之前初始化两个计数器。*图 443* 显示了与*图 442* 具有相同配置，只不过处于触发模式（TIM2_SMCR 寄存器中的 SMS=110）而非门控模式的计数行为。

图 443. 使用 TIM3 的使能信号触发 TIM2



使用一个外部触发信号同步启动 2 个定时器

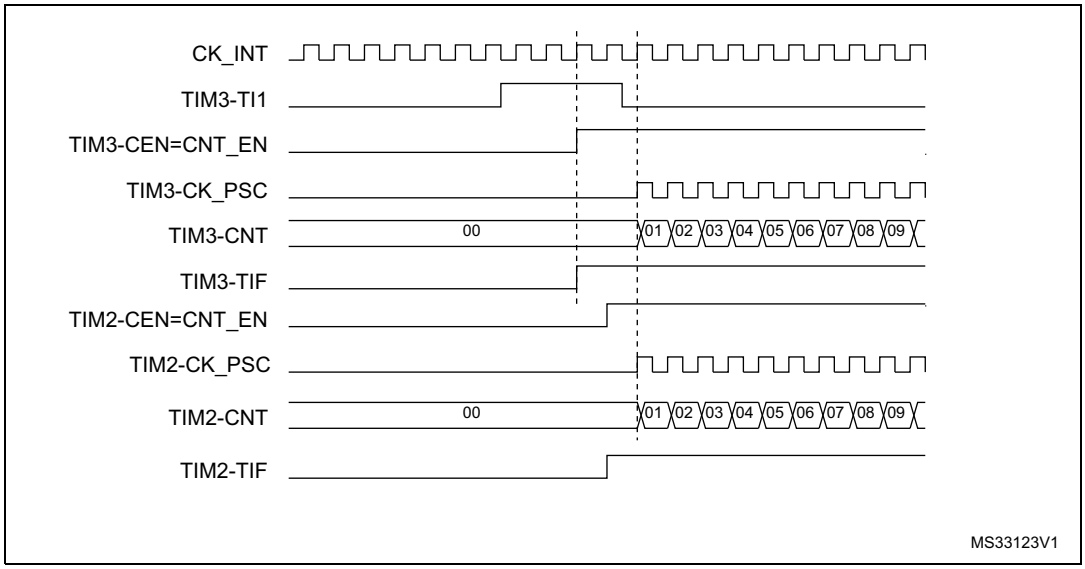
本例中，TIM3 的 TI1 输入出现上升沿时使能 TIM3，使能 TIM3 的同时使能 TIM2。相关连接图，请参见*图 438*。要确保计数器对齐，TIM3 必须配置为主/从模式（对于 TI1 而言，TIM3 为从模式；对于 TIM2 而言，TIM3 为主模式）：

1. 将 TIM3 配置为主模式，发送其使能信号作为触发输出（TIM3_CR2 寄存器中的 MMS=001）。
2. 将 TIM3 配置为从模式以接收来自 TI1 的输入触发（TIM3_SMCR 寄存器中的 TS=00100）。
3. 将 TIM3 配置为触发模式（TIM3_SMCR 寄存器中的 SMS=110）。
4. 通过写入 MSM=1（TIM3_SMCR 寄存器）将 TIM3 配置为主/从模式。
5. 配置 TIM2 以接收来自 TIM3 的输入触发（TIM2_SMCR 寄存器中的 TS=00000）。
6. 将 TIM2 配置为触发模式（TIM2_SMCR 寄存器中的 SMS=110）。

当 TI1 (TIM3) 出现上升沿时，两个计数器开始根据内部时钟同步计数，并且两个 TIF 标志都置 1。

注： *本例中，两个定时器都在启动之前进行了初始化（通过将各自的 UG 位置 1）。两个计数器都从 0 开始计数，但可以通过对任意一个计数器寄存器 (TIMx_CNT) 进行写操作，在二者之间轻松插入一个偏移量。可注意到主/从模式在 TIM3 的 CNT_EN 与 CK_PSC 之间产生了延迟。*

图 444. 使用 TIM3 的 TI1 输入触发 TIM3 和 TIM2



注： *必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。*

39.3.20 DMA 分组传送模式

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对定时器的一部分多次重新编程而无需软件开销，但也可以用于定期地一次性地批量读取定时器的多个寄存器的内容。

DMA 控制器访问目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（分组）。每次写入 TIMx_DMAR 寄存器都会重定向到定时器其中的一个寄存器。

TIMx_DCR 寄存器中的 DBL[4:0] 位设置 DMA 突发传输的长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次分组传送，即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA[4:0] 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量：

示例：

00000：TIMx_CR1

00001：TIMx_CR2

00010：TIMx_SMCR

例如，定时器 DMA 分组传送功能用于在发生更新事件后将 CCRx 寄存器（x = 2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

1. 将相应的 DMA 通道配置如下：
   - DMA 通道外设地址为 DMAR 寄存器地址。
   - DMA 通道内存地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
   - 要传输的数据量 = 3（参见下文注释）。
   - 禁止循环模式。
2. 通过将 DBA 和 DBL 位字段配置如下来配置 DCR 寄存器：
   DBL = 3 次传输，DBA = 0xE。
3. 使能 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 TIMx
5. 使能 DMA 通道

本例适用于每个 CCRx 寄存器必须更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间，data1 传输到 CCR2，data2 传输到 CCR3，data3 传输到 CCR4；在第二个更新 DMA 请求期间，data4 传输到 CCR2，data5 传输到 CCR3，data6 传输到 CCR4。

注： *可以将空值写入保留的寄存器中。*

39.3.21 调试模式

当系统进入调试模式（处理器内核停止）时，TIMx 计数器会根据 DBGMCU 模块中的 TIMx 配置位选择继续正常工作或者停止工作。更多详细信息，请参见*第 59.10.6 节：微处理器调试单元 (DBGMCU)*。

为了安全起见，当计数器停止（DBGMCU_APB1FZ2 中的 TIMx = 1）时，输出被禁止。

39.4 TIM2/TIM3/TIM4/TIM5 寄存器

有关寄存器说明中使用的缩写，请参见*第 1.2 节*。

外设寄存器可按半字（16 位）或字（32 位）访问。

39.4.1 TIMx 控制寄存器 1 (TIMx_CR1)（x = 2 到 5）

TIMx control register 1

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
				rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD[1:0]**：时钟分频 (Clock division)
该位字段指示定时器时钟 (CK_INT) 频率与数字滤波器所使用的采样时钟（ETR、TIx）之间的分频比，
00：$t_{DTS} = t_{CK_INT}$
01：$t_{DTS} = 2 \times t_{CK_INT}$
10：$t_{DTS} = 4 \times t_{CK_INT}$
11：保留

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)
0：TIMx_ARR 寄存器不进行缓冲
1：TIMx_ARR 寄存器进行缓冲

位 6:5 **CMS[1:0]**：中心对齐模式选择 (Center-aligned mode selection)
00：边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。
01：中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。
10：中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。
11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志都会置 1。
注意：只要计数器处于使能状态 (CEN=1)，就不得从边沿对齐模式切换为中心对齐模式。

位 4 **DIR**：方向
0：计数器递增计数
1：计数器递减计数
注意：当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。

位 3 **OPM**：单脉冲模式 (One-pulse mode)
0：计数器在发生更新事件时不会停止计数
1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

位 2 **URS**：更新请求源 (Update request source)

该位由软件置 1 和清零，用以选择 UEV 事件源。

0：使能时，所有以下事件都会生成更新中断或 DMA 请求。此类事件包括：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。

位 1 **UDIS**：更新禁止 (Update disable)

该位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

带缓冲的寄存器被加载为预加载数值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注意：只有事先通过软件将 CEN 位置 1，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将 CEN 位置 1。

在单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。

39.4.2 TIMx 控制寄存器 2 (TIMx_CR2)（x = 2 到 5）

TIMx control register 2

偏移地址：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			CCDS	Res.	Res.	Res.
								rw	rw	rw	rw	rw			

位 15:8 保留，必须保持复位值。

位 7 **TI1S**：TI1 选择 (TI1 selection)

0：TIMx_CH1 引脚连接到 TI1 输入

1：TIMx_CH1、CH2 和 CH3 引脚连接到 TI1 输入（异或组合）。另请参见*第 1939 页的第 38.3.25 节：连接霍尔传感器*

位 6:4 **MMS[2:0]**：主模式选择 (Master mode selection)

这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下：

000：**复位**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。

001：**使能**——计数器使能信号 (CNT_EN) 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑与运算组合而成。

当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。

010：**更新**——选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器。

011：**比较脉冲**——CC1IF 标志置 1 时（即使已为高电平），只要发生捕获或比较匹配，触发输出都会发送一个正脉冲。(TRGO)

100：**比较**——OC1REFC 信号用作触发输出 (TRGO)

101：**比较**——OC2REFC 信号用作触发输出 (TRGO)

110：**比较**——OC3REFC 信号用作触发输出 (TRGO)

111：**比较**——OC4REFC 信号用作触发输出 (TRGO)

注意：必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

位 3 **CCDS**：捕获/比较 DMA 选择 (Capture/compare DMA selection)

0：发生 CCx 事件时发送 CCx DMA 请求

1：发生更新事件时发送 CCx DMA 请求

位 2:0 保留，必须保持复位值。

39.4.3 TIMx 从模式控制寄存器 (TIMx_SMCR)（x = 2 到 5）

TIMx slave mode control register

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS[4:3]		Res.	Res.	Res.	SMS[3]
										rw	rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

位 31:22 保留，必须保持复位值。

位 19:17 保留，必须保持复位值。

位 15 **ETP**：外部触发极性 (External trigger polarity)

该位可选择将 ETR 或 $\overline{\text{ETR}}$ 用于触发操作

0：ETR 未反相，高电平或上升沿有效

1：ETR 反相，低电平或下降沿有效

位 14 **ECE**：外部时钟使能 (External clock enable)

该位可使能外部时钟模式 2。

0：禁止外部时钟模式 2

1：使能外部时钟模式 2。计数器时钟由 ETRF 信号的任意有效边沿提供。

注意：将 ECE 位置 1 与选择外部时钟模式 1 并将 TRGI 连接到 ETRF（SMS=111 且 TS=00111）具有相同效果。

外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF（TS 位不得为 00111）。

如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 ETRF。

位 13:12 **ETPS[1:0]**：外部触发预分频器 (External trigger prescaler)

外部触发信号 ETRP 频率不得超过 CK_INT 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。

00：预分频器关闭

01：2 分频 ETRP 频率

10：4 分频 ETRP 频率

11：8 分频 ETRP 频率

位 11:8 **ETF[3:0]**：外部触发滤波器 (External trigger filter)

该位字段可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样

0001：$f_{SAMPLING}=f_{CK_INT}$，N=2

0010：$f_{SAMPLING}=f_{CK_INT}$，N=4

0011：$f_{SAMPLING}=f_{CK_INT}$，N=8

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 7 **MSM**：主/从模式 (Master/Slave mode)

0：无操作

1：当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于由单个外部事件对多个定时器进行同步的情况。

位 21、20、6、5、4 **TS[4:0]**：触发选择

该位字段可选择将要用于同步计数器的触发输入。

00000：内部触发 0 (ITR0)
00001：内部触发 1 (ITR1)
00010：内部触发 2 (ITR2)
00011：内部触发 3 (ITR3)
00100：TI1 边沿检测器 (TI1F_ED)
00101：滤波后的定时器输入 1 (TI1FP1)
00110：滤波后的定时器输入 2 (TI2FP2)
00111：外部触发输入 (ETRF)
01000：内部触发 4 (ITR4)
01001：内部触发 5 (ITR5)
01010：内部触发 6 (ITR6)
01011：内部触发 7 (ITR7)
01100：内部触发 8 (ITR8)
其他值：保留

有关各定时器 ITRx 含义的详细信息，请参见*第 2036 页的表 355：TIMx 内部触发连接*。

注意：这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。

位 3 保留，必须保持复位值。

位 16、2、1、0 **SMS[3:0]**：从模式选择 (Slave mode selection)

选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。

0000：禁止从模式——如果 CEN = “1”，预分频器时钟直接由内部时钟提供。
0001：编码器模式 1——计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。
0010：编码器模式 2——计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。
0011：编码器模式 3——计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个信号的电平。
0100：复位模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。
0101：门控模式——触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。
0110：触发模式——触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。
0111：外部时钟模式 1——由所选触发信号 (TRGI) 的上升沿提供计数器时钟。
1000：复位 + 触发组合模式——在出现所选触发输入 (TRGI) 上升沿时，
重新初始化计数器，生成一个定时器更新事件并启动计数器。

注意：如果将 TI1F_ED 选作触发输入 (TS=00100)，则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。

注意：必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。

表 355. TIMx 内部触发连接

从 TIM	ITR0	ITR1	ITR2	ITR3	ITR4	ITR5	ITR6	ITR7	ITR8
TIM2	TIM1	TIM8	TIM3	TIM4	-	-	-	-	-
TIM3	TIM1	TIM2	TIM15	TIM4	-	-	-	-	-
TIM4	TIM1	TIM2	TIM3	TIM8	-	-	-	-	-
TIM5	TIM1	TIM8	TIM3	TIM4	-	-	fdcan1_soc	-	OTG_SOF

39.4.4 TIMx DMA/中断使能寄存器 (TIMx_DIER)（x = 2 到 5）

TIMx DMA/Interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

位 15 保留，必须保持复位值。

位 14 **TDE**：触发信号 DMA 请求使能 (Trigger DMA request enable)
0：禁止触发信号 DMA 请求。
1：使能触发信号 DMA 请求。

位 13 保留，必须保持复位值。

位 12 **CC4DE**：捕获/比较 4 DMA 请求使能 (Capture/Compare 4 DMA request enable)
0：禁止 CC4 DMA 请求。
1：使能 CC4 DMA 请求。

位 11 **CC3DE**：捕获/比较 3 DMA 请求使能 (Capture/Compare 3 DMA request enable)
0：禁止 CC3 DMA 请求。
1：使能 CC3 DMA 请求。

位 10 **CC2DE**：捕获/比较 2 DMA 请求使能 (Capture/Compare 2 DMA request enable)
0：禁止 CC2 DMA 请求。
1：使能 CC2 DMA 请求。

位 9 **CC1DE**：捕获/比较 1 DMA 请求使能 (Capture/Compare 1 DMA request enable)
0：禁止 CC1 DMA 请求。
1：使能 CC1 DMA 请求。

位 8 **UDE**：更新 DMA 请求使能 (Update DMA request enable)
0：禁止更新 DMA 请求。
1：使能更新 DMA 请求。

位 7 保留，必须保持复位值。

位 6 **TIE**：触发信号中断使能 (Trigger interrupt enable)
0：禁止触发信号中断。
1：使能触发信号中断。

位 5 保留，必须保持复位值。

位 4 **CC4IE**：捕获/比较 4 中断使能 (Capture/Compare 4 interrupt enable)
0：禁止 CC4 中断。
1：使能 CC4 中断。

位 3 **CC3IE**：捕获/比较 3 中断使能 (Capture/Compare 3 interrupt enable)
0：禁止 CC3 中断。
1：使能 CC3 中断。

位 2 **CC2IE**：捕获/比较 2 中断使能 (Capture/Compare 2 interrupt enable)
0：禁止 CC2 中断。
1：使能 CC2 中断。

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断。
1：使能 CC1 中断。

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断。
1：使能更新中断。

39.4.5 TIMx 状态寄存器 (TIMx_SR)（x = 2 到 5）

TIMx status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位 15:13 保留，必须保持复位值。

位 12 **CC4OF**：捕获/比较 4 过捕获标志 (Capture/Compare 4 overcapture flag)
请参见 CC1OF 说明

位 11 **CC3OF**：捕获/比较 3 过捕获标志 (Capture/Compare 3 overcapture flag)
请参见 CC1OF 说明

位 10 **CC2OF**：捕获/比较 2 过捕获标志 (Capture/compare 2 overcapture flag)
请参见 CC1OF 说明

位 9 **CC1OF**：捕获/比较 1 过捕获标志 (Capture/Compare 1 overcapture flag)
仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。
0：未检测到过捕获。
1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8:7 保留，必须保持复位值。

位 6 **TIF**：触发中断标志 (Trigger interrupt flag)
在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到 TRG 触发事件有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。
0：未发生触发事件。
1：触发中断挂起。

位 5 保留，必须保持复位值。

位 4 **CC4IF**：捕获/比较 4 中断标志 (Capture/Compare 4 interrupt flag)
请参见 CC1IF 说明

位 3 **CC3IF**：捕获/比较 3 中断标志 (Capture/Compare 3 interrupt flag)
请参见 CC1IF 说明

位 2 **CC2IF**：捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag)
请参见 CC1IF 说明

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/compare 1 interrupt flag)

该标志由硬件置 1，但需要通过软件清零（输入捕获或输出比较模式）或通过读取 TIMx_CCR1 寄存器清零（仅限输入捕获模式）。

0：未发生比较匹配/输入捕获

1：发生了比较匹配/输入捕获

如果通道 CC1 配置为输出：当计数器 TIMx_CNT 的值与 TIMx_CCR1 的值匹配时，此标志置 1。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。在中心对齐模式下，可通过 3 种方式将此标志置 1，完整说明请参见 TIMx_CR1 寄存器中的 CMS 位。

如果通道 CC1 配置为输入：当 TIMx_CCR1 寄存器中捕获到计数器值时（如果通过设置 TIMx_CCER 中的 CC1P 和 CC1NP 位定义为边沿有效，则改为在 IC1 上检测到边沿时），该位置 1。

位 0 **UIF**：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

上溢或下溢（对于 TIM2 到 TIM4）以及当 TIMx_CR1 寄存器中的 UDIS = 0 时。

TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（参见同步控制寄存器说明）。

39.4.6 TIMx 事件生成寄存器 (TIMx_EGR)（x = 2 到 5）

TIMx event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
									w		w	w	w	w	w

位 15:7 保留，必须保持复位值。

位 6 **TG**：触发生成 (Trigger generation)

该位由软件置 1 以生成事件，并由硬件自动清零。

0：无操作

1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 5 保留，必须保持复位值。

位 4 **CC4G**：捕获/比较 4 生成 (Capture/compare 4 generation)

请参见 CC1G 说明

位 3 **CC3G**：捕获/比较 3 生成 (Capture/compare 3 generation)

请参见 CC1G 说明

位 2 **CC2G**：捕获/比较 2 生成 (Capture/compare 2 generation)

请参见 CC1G 说明

位 1 **CC1G**：捕获/比较 1 生成 (Capture/compare 1 generation)

该位由软件置 1 以生成事件，并由硬件自动清零。

0：无操作

1：通道 1 上生成捕获/比较事件：

如果通道 CC1 配置为输出：

使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。

如果通道 CC1 配置为输入：

TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)

该位可通过软件置 1，并由硬件自动清零。

0：无操作

1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值 (TIMx_ARR)。

39.4.7 TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1)（x = 2 到 5）

TIMx capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:12 **IC2F[3:0]**：输入捕获 2 滤波器 (Input capture 2 filter)

位 11:10 **IC2PSC[1:0]**：输入捕获 2 预分频器 (Input capture 2 prescaler)

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/compare 2 selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出。

01：CC2 通道配置为输入，IC2 映射到 TI2 上。

10：CC2 通道配置为输入，IC2 映射到 TI1 上。

11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。

位 7:4 **IC1F[3:0]**：输入捕获 1 滤波器 (Input capture 1 filter)

该位字段可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC[1:0]**：输入捕获 1 预分频器 (Input capture 1 prescaler)

该位字段定义 CC1 输入 (IC1) 的预分频比。只要 CC1E=0（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
10：CC1 通道配置为输入，IC1 映射到 TI2 上
11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

39.4.8 TIMx 捕获/比较模式寄存器 1 [复用] (TIMx_CCMR1)（x = 2 到 5）

TIMx capture/compare mode register 1 [alternate]

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 23:17 保留，必须保持复位值。

位 15 **OC2CE**：输出比较 2 清零使能 (Output compare 2 clear enable)

位 24、14:12 **OC2M[3:0]**：输出比较 2 模式 (Output compare 2 mode)
请参见 OC1M 说明——位 6:4

位 11 **OC2PE**：输出比较 2 预装载使能 (Output compare 2 preload enable)

位 10 **OC2FE**：输出比较 2 快速使能 (Output compare 2 fast enable)

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/Compare 2 selection)
该位字段定义通道方向（输入/输出）以及所使用的输入。
00：CC2 通道配置为输出
01：CC2 通道配置为输入，IC2 映射到 TI2 上
10：CC2 通道配置为输入，IC2 映射到 TI1 上
11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。

位 7 **OC1CE**：输出比较 1 清零使能 (Output compare 1 clear enable)
0：OC1Ref 不受 ETRF 输入影响
1：ETRF 输入上检测到高电平时，OC1Ref 立即清零。

位 16，6:4 **OC1M[3:0]**：输出比较 1 模式 (Output compare 1 mode)

这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。当定时器用作软件时基时，可以使用该模式。在定时器操作期间使能冻结模式时，输出保持进入冻结状态前的状态（有效或无效）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便输出有效电平，否则输出无效电平。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便输出无效电平 (OC1REF=0)，否则输出有效电平 (OC1REF=1)。

0111：PWM 模式 2——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便输出无效电平，否则输出有效电平。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便输出有效电平，否则输出无效电平。

1000：可再触发 OPM 模式 1——在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。

1001: 可再触发 OPM 模式 2——在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。

1010: 保留。

1011: 保留。

1100: 组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。

1101: 组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。

1110: 不对称 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

1111：不对称 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

注意：在 PWM 模式下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。

注意：OC1M[3] 位不是连续的，而是在位 16 中。

位 3 **OC1PE**：输出比较 1 预装载使能 (Output compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到影子寄存器中。

位 2 **OC1FE**：输出比较 1 快速使能 (Output compare 1 fast enable)

该位可降低触发事件与定时器输出跳变之间的延迟。单脉冲模式下必须使用（TIMx_CR1 寄存器中的 OPM 位置 1），以便在启动触发后快速发送输出脉冲。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出。

01：CC1 通道配置为输入，IC1 映射到 TI1 上。

10：CC1 通道配置为输入，IC1 映射到 TI2 上。

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

39.4.9 TIMx 捕获/比较模式寄存器 2 (TIMx_CCMR2)（x = 2 到 5）

TIMx capture/compare mode register 2

偏移地址：0x1C

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F[3:0]				IC4PSC[1:0]		CC4S[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:12 **IC4F[3:0]**：输入捕获 4 滤波器 (Input capture 4 filter)

位 11:10 **IC4PSC[1:0]**：输入捕获 4 预分频器 (Input capture 4 prescaler)

位 9:8 **CC4S[1:0]**：捕获/比较 4 选择 (Capture/Compare 4 selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。

00：CC4 通道配置为输出

01：CC4 通道配置为输入，IC4 映射到 TI4 上

10：CC4 通道配置为输入，IC4 映射到 TI3 上

11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC4E = 0），才可向 CC4S 位写入数据。

位 7:4 **IC3F[3:0]**：输入捕获 3 滤波器 (Input capture 3 filter)

位 3:2 **IC3PSC[1:0]**：输入捕获 3 预分频器 (Input capture 3 prescaler)

位 1:0 **CC3S[1:0]**：捕获/比较 3 选择 (Capture/Compare 3 selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。

00：CC3 通道配置为输出

01：CC3 通道配置为输入，IC3 映射到 TI3 上

10：CC3 通道配置为输入，IC3 映射到 TI4 上

11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC3E = 0），才可向 CC3S 位写入数据。

39.4.10 TIMx 捕获/比较模式寄存器 2 [复用] (TIMx_CCMR2)（x = 2 到 5）

TIMx capture/compare mode register 2

偏移地址：0x1C

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M [3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 23:17 保留，必须保持复位值。

位 15 **OC4CE**：输出比较 4 清零使能 (Output compare 4 clear enable)

位 24、14:12 **OC4M[3:0]**：输出比较 4 模式 (Output compare 4 mode)
请参见 OC1M 说明（TIMx_CCMR1 寄存器中的位 6:4）

位 11 **OC4PE**：输出比较 4 预装载使能 (Output compare 4 preload enable)

位 10 **OC4FE**：输出比较 4 快速使能 (Output compare 4 fast enable)

位 9:8 **CC4S[1:0]**：捕获/比较 4 选择 (Capture/Compare 4 selection)
该位字段定义通道方向（输入/输出）以及所使用的输入。
00：CC4 通道配置为输出
01：CC4 通道配置为输入，IC4 映射到 TI4 上
10：CC4 通道配置为输入，IC4 映射到 TI3 上
11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（TIMx_CCER 中的 CC4E = 0），才可向 CC4S 位写入数据。

位 7 **OC3CE**：输出比较 3 清零使能 (Output compare 3 clear enable)

位 16，6:4 **OC3M[3:0]**：输出比较 3 模式 (Output compare 3 mode)
请参见 OC1M 说明（TIMx_CCMR1 寄存器中的位 6:4）

位 3 **OC3PE**：输出比较 3 预装载使能 (Output compare 3 preload enable)

位 2 **OC3FE**：输出比较 3 快速使能 (Output compare 3 fast enable)

位 1:0 **CC3S[1:0]**：捕获/比较 3 选择 (Capture/Compare 3 selection)
该位字段定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（TIMx_CCER 中的 CC3E = 0），才可向 CC3S 位写入数据。

39.4.11 TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 2 到 5）

TIMx capture/compare enable register

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

位 15 **CC4NP**：*捕获/比较 4 输出极性 (Capture/Compare 4 output Polarity)*。
请参见 CC1NP 说明

位 14 保留，必须保持复位值。

位 13 **CC4P**：*捕获/比较 4 输出极性 (Capture/Compare 4 output Polarity)*。
请参见 CC1P 说明

位 12 **CC4E**：*捕获/比较 4 输出使能 (Capture/Compare 4 output enable)*。
请参见 CC1E 说明

位 11 **CC3NP**：*捕获/比较 3 输出极性 (Capture/Compare 3 output polarity)*。
请参见 CC1NP 说明

位 10 保留，必须保持复位值。

位 9 **CC3P**：*捕获/比较 3 输出极性 (Capture/Compare 3 output polarity)*。
请参见 CC1P 说明

位 8 **CC3E**：*捕获/比较 3 输出使能 (Capture/Compare 3 output enable)*。
请参见 CC1E 说明

位 7 **CC2NP**：*捕获/比较 2 输出极性 (Capture/Compare 2 output polarity)*。
请参见 CC1NP 说明

位 6 保留，必须保持复位值。

位 5 **CC2P**：*捕获/比较 2 输出极性 (Capture/Compare 2 output polarity)*。
请参见 CC1P 说明

位 4 **CC2E**：*捕获/比较 2 输出使能 (Capture/Compare 2 output enable)*。
请参见 CC1E 说明

位 3 **CC1NP**：*捕获/比较 1 输出极性 (Capture/Compare 1 output Polarity)*。
CC1 通道配置为输出：在这种情况下，CC1NP 必须保持清零。
CC1 通道配置为输入：该位与 CC1P 配合使用，用以定义 TI1FP1/TI2FP1 的极性。请参见 CC1P 说明。

位 2 保留，必须保持复位值。

位 1 **CC1P**：*捕获/比较 1 输出极性 (Capture/Compare 1 output Polarity)*。

0：OC1 高电平有效（输出模式）/边沿有效性选择（输入模式，见下文）

1：OC1 低电平有效（输出模式）/边沿有效性选择（输入模式，见下文）

CC1 通道配置为输入时，CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。

CC1NP=0，CC1P=0： 非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。

CC1NP=0，CC1P=1： 反相/下降沿触发。电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。

CC1NP=1，CC1P=1： 非反相/上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。

CC1NP=1，CC1P=0：该配置保留，不得使用。

位 0 **CC1E**：*捕获/比较 1 输出使能 (Capture/Compare 1 output enable)*。

0：禁止捕获模式/OC1 无效

1：使能捕获模式/在相应输出引脚上输出 OC1 信号

表 356. 标准 OCx 通道的输出控制位

CCxE 位	OCx 输出状态
0	禁止输出（不由定时器驱动：高阻态）
1	使能输出（tim_ocx = tim_ocxref + 极性）

注： *与标准 OCx 通道相连的外部 IO 引脚的状态取决于 OCx 通道的状态以及 GPIO 控制和复用功能寄存器。*

39.4.12 TIMx 计数器 (TIMx_CNT)（x = 2 到 5）

TIMx counter

该寄存器的位 31 有两种可能的定义，具体取决于 TIMx_CR1 寄存器中 UIFREMAP 的值：

- 本节内容对应于 UIFREMAP = 0 的情况
- 下一节内容对应于 UIFREMAP = 1 的情况

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **CNT[31:16]**：计数器值的最高有效部分（TIM2 和 TIM5）

位 15:0 **CNT[15:0]**：计数器值的最低有效部分 (Least significant part of counter value)

39.4.13 TIMx 计数器 [复用] (TIMx_CNT)（x = 2 到 5）

TIMx counter [alternate]

该寄存器的位 31 有两种可能的定义，具体取决于 TIMx_CR1 寄存器中 UIFREMAP 的值：

- 上一节内容对应于 UIFREMAP = 0 的情况
- 本节内容对应于 UIFREMAP = 1 的情况

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIFCPY	CNT[30:16]														
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **UIFCPY**：UIF 副本 (UIF Copy)
该位是 TIMx_ISR 寄存器中 UIF 位的只读副本

位 30:16 **CNT[30:16]**：计数器值的最高有效部分（TIM2 和 TIM5）

位 15:0 **CNT[15:0]**：计数器值的最低有效部分 (Least significant part of counter value)

39.4.14 TIMx 预分频器 (TIMx_PSC)（x = 2 到 5）

TIMx prescaler

偏移地址：0x28

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)
计数器时钟频率 CK_CNT 等于 f_{CK_PSC} / (PSC[15:0] + 1)。
PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到实际预分频器寄存器的值。

39.4.15 TIMx 自动重载寄存器 (TIMx_ARR)（x = 2 到 5）

TIMx auto-reload register

偏移地址：0x2C

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **ARR[31:16]**：自动重载值的高 16 位（TIM2 和 TIM5）

位 15:0 **ARR[15:0]**：低自动重载值

ARR 为要装载到实际自动重载寄存器的值。

有关 ARR 更新和行为的更多详细信息，请参见*第 1990 页的第 39.3.1 节：时基单元*。

当自动重载值为空时，计数器不工作。

39.4.16 TIMx 捕获/比较寄存器 1 (TIMx_CCR1)（x = 2 到 5）

TIMx capture/compare register 1

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **CCR1[31:16]**：捕获/比较 1 值的高 16 位 (High Capture/Compare 1 value)（TIM2 和 TIM5）

位 15:0 **CCR1[15:0]**：捕获/比较 1 的低 16 位 (Low Capture/Compare 1 value)

如果通道 CC1 配置为输出：

CCR1 是捕获/比较寄存器 1 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 1）。

实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出相应电平的值。

如果通道 CC1 配置为输入：

CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。只能读取 TIMx_CCR1 寄存器，无法对其进行编程。

39.4.17 TIMx 捕获/比较寄存器 2 (TIMx_CCR2)（x = 2 到 5）

TIMx capture/compare register 2

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **CCR2[31:16]**：捕获/比较 2 值的高 16 位 (High Capture/Compare 2 value)（TIM2 和 TIM5）

位 15:0 **CCR2[15:0]**：捕获/比较 2 的低 16 位 (Low Capture/Compare 2 value)

如果通道 CC2 配置为输出：

CCR2 是捕获/比较寄存器 2 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 2）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。

如果通道 CC2 配置为输入：

CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。只能读取 TIMx_CCR2 寄存器，无法对其进行编程。

39.4.18 TIMx 捕获/比较寄存器 3 (TIMx_CCR3)（x = 2 到 5）

TIMx capture/compare register 3

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **CCR3[31:16]**：捕获/比较 3 值的高 16 位 (High Capture/Compare 3 value)（TIM2 和 TIM5）

位 15:0 **CCR3[15:0]**：捕获/比较 4 的低 16 位 (Low Capture/Compare 4 value)

如果通道 CC3 配置为输出：

CCR3 是捕获/比较寄存器 3 的预装载值。

如果没有通过 TIMx_CCMR2 寄存器中的 OC3PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 3）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC3 输出上发出信号的值。

如果通道 CC3 配置为输入：

CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。只能读取 TIMx_CCR3 寄存器，无法对其进行编程。

39.4.19 TIMx 捕获/比较寄存器 4 (TIMx_CCR4)（x = 2 到 5）

TIMx capture/compare register 4

偏移地址：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **CCR4[31:16]**：捕获/比较 4 值的高 16 位 (High Capture/Compare 4 value)（TIM2 和 TIM5）

位 15:0 **CCR4[15:0]**：捕获/比较 4 的低 16 位 (Low Capture/Compare 4 value)

1. 如果 CC4 通道配置为输出（CC4S 位）：
   CCR4 是捕获/比较寄存器 4 的预装载值。
   如果没有通过 TIMx_CCMR2 寄存器中的 OC4PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 4）。
   有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC4 输出上发出信号的值。
2. 如果 CC4 通道配置为输入（TIMx_CCMR4 寄存器中的 CC4S 位）：
   CCR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值。只能读取 TIMx_CCR4 寄存器，无法对其进行编程。

39.4.20 TIMx DMA 控制寄存器 (TIMx_DCR)（x = 2 到 5）

TIMx DMA control register

偏移地址：0x48

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 15:13 保留，必须保持复位值。

位 12:8 **DBL[4:0]**：DMA 分组传送长度 (DMA burst length)

该 5 位向量定义了 DMA 的传送次数（当对 TIMx_DMAR 寄存器进行读或写时，定时器进行一次突发传送）。

00000：1 次传输，
00001：2 次传输，
00010：3 次传输，
...
10001：18 次传输。

位 7:5 保留，必须保持复位值。

位 4:0 **DBA[4:0]**：DMA 基址 (DMA base address)

该 5 位向量定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：
00000：TIMx_CR1
00001：TIMx_CR2
00010：TIMx_SMCR
...

示例：考虑以下传输：DBL = 7 次传输，DBA = TIMx_CR1。这种情况下将向/从自 TIMx_CR1 地址开始的 7 个寄存器传输数据。

39.4.21 TIMx 全传输 DMA 地址 (TIMx_DMAR)（x = 2 到 5）

TIMx DMA address for full transfer

偏移地址：0x4C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **DMAB[15:0]**：DMA 分组传送寄存器 (DMA register for burst accesses)

对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器

(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。

39.4.22 TIM2 复用功能选项寄存器 1 (TIM2_AF1)

TIM2 alternate function option register 1

偏移地址：0x60

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL[3:2]	
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETRSEL[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw														

位 31:18 保留，必须保持复位值。

位 17:14 **ETRSEL[3:0]**：ETR 源选择 (ETR source selection)

这些位选择 ETR 输入源。

0000：ETR 输入连接到 I/O

0001：保留

0010：保留

0011：LSE

0100：SAI1 FS_A

0101：SAI1 FS_B

0110：eth1_ptp_pps_out

0111：eth2_ptp_pps_out

其他值：保留

位 13:0 保留，必须保持复位值。

39.4.23 TIM3 复用功能选项寄存器 1 (TIM3_AF1)

TIM3 alternate function option register 1

偏移地址：0x60

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL[3:2]	
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETRSEL[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw														

位 31:18 保留，必须保持复位值。

位 17:14 **ETRSEL[3:0]**：ETR 源选择 (ETR source selection)

这些位选择 ETR 输入源。
0000：ETR 输入连接到 I/O
0001：保留
0010：保留
0011：保留
0100：保留
0101：保留
0110：eth1_ptp_pps_out
0111：eth2_ptp_pps_out
其他值：保留

位 13:0 保留，必须保持复位值。

39.4.24 TIM4 复用功能选项寄存器 1 (TIM4_AF1)

TIM4 alternate function option register 1

偏移地址：0x60

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL[3:2]	
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETRSEL[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw														

位 31:18 保留，必须保持复位值。

位 17:14 **ETRSEL[3:0]**：ETR 源选择 (ETR source selection)

这些位选择 ETR 输入源。
0000：ETR 输入连接到 I/O
其他值：保留

位 13:0 保留，必须保持复位值。

39.4.25 TIM5 复用功能选项寄存器 1 (TIM5_AF1)

TIM5 alternate function option register 1

偏移地址：0x60

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL[3:2]	
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETRSEL[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw														

位 31:18 保留，必须保持复位值。

位 17:14 **ETRSEL[3:0]**：ETR 源选择 (ETR source selection)
这些位选择 ETR 输入源。
0000：ETR 输入连接到 I/O
0001：SAI2 FS_A 连接到 ETR 输入
0010：SAI2 FS_B 连接到 ETR 输入
0011：OTG_SOF
其他值：保留

位 13:0 保留，必须保持复位值。

39.4.26 TIM2 定时器输入选择寄存器 (TIM2_TISEL)

TIM2 timer input selection register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **TI4SEL[3:0]**：TI4[0] 到 TI4[15] 输入选择 (TI4[0] to TI4[15] input selection)
这些位选择 TI4[0] 到 TI4[15] 输入源。
0000：TIM2_CH4 输入
其他值：保留

位 23:20 保留，必须保持复位值。

位 19:16 **TI3SEL[3:0]**：TI3[0] 到 TI3[15] 输入选择 (TI3[0] to TI3[15] input selection)
这些位选择 TI3[0] 到 TI3[15] 输入源。
0000：TIM2_CH3 输入
其他值：保留

位 15:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：TI2[0] 到 TI2[15] 输入选择 (TI2[0] to TI2[15] input selection)
这些位选择 TI2[0] 到 TI2[15] 输入源。
0000：TIM2_CH2 输入
其他值：保留

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：TI1[0] 到 TI1[15] 输入选择 (TI1[0] to TI1[15] input selection)
这些位选择 TI1[0] 到 TI1[15] 输入源。
0000：TIM2_CH1 输入
其他值：保留

39.4.27 TIM3 定时器输入选择寄存器 (TIM3_TISEL)

TIM3 timer input selection register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **TI4SEL[3:0]**：TI4[0] 到 TI4[15] 输入选择 (TI4[0] to TI4[15] input selection)
这些位选择 TI4[0] 到 TI4[15] 输入源。
0000：TIM3_CH4 输入
其他值：保留

位 23:20 保留，必须保持复位值。

位 19:16 **TI3SEL[3:0]**：TI3[0] 到 TI3[15] 输入选择 (TI3[0] to TI3[15] input selection)
这些位选择 TI3[0] 到 TI3[15] 输入源。
0000：TIM3_CH3 输入
其他值：保留

位 15:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：TI2[0] 到 TI2[15] 输入选择 (TI2[0] to TI2[15] input selection)
这些位选择 TI2[0] 到 TI2[15] 输入源。
0000：TIM3_CH2 输入
其他值：保留

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：TI1[0] 到 TI1[15] 输入选择 (TI1[0] to TI1[15] input selection)
这些位选择 TI1[0] 到 TI1[15] 输入源。
0000：TIM3_CH1 输入
其他值：保留

39.4.28 TIM4 定时器输入选择寄存器 (TIM4_TISEL)

TIM4 timer input selection register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **TI4SEL[3:0]**：TI4[0] 到 TI4[15] 输入选择 (TI4[0] to TI4[15] input selection)
这些位选择 TI4[0] 到 TI4[15] 输入源。
0000：TIM4_CH4 输入
其他值：保留

位 23:20 保留，必须保持复位值。

位 19:16 **TI3SEL[3:0]**：TI3[0] 到 TI3[15] 输入选择 (TI3[0] to TI3[15] input selection)
这些位选择 TI3[0] 到 TI3[15] 输入源。
0000：TIM4_CH3 输入
其他值：保留

位 15:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：TI2[0] 到 TI2[15] 输入选择 (TI2[0] to TI2[15] input selection)
这些位选择 TI2[0] 到 TI2[15] 输入源。
0000：TIM4_CH2 输入
其他值：保留

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：TI1[0] 到 TI1[15] 输入选择 (TI1[0] to TI1[15] input selection)
这些位选择 TI1[0] 到 TI1[15] 输入源。
0000：TIM4_CH1 输入
其他值：保留

39.4.29 TIM5 定时器输入选择寄存器 (TIM5_TISEL)

TIM5 timer input selection register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **TI4SEL[3:0]**：TI4[0] 到 TI4[15] 输入选择 (TI4[0] to TI4[15] input selection)
这些位选择 TI4[0] 到 TI4[15] 输入源。
0000：TIM5_CH4 输入
其他值：保留

位 23:20 保留，必须保持复位值。

位 19:16 **TI3SEL[3:0]**：TI3[0] 到 TI3[15] 输入选择 (TI3[0] to TI3[15] input selection)
这些位选择 TI3[0] 到 TI3[15] 输入源。
0000：TIM5_CH3 输入
其他值：保留

位 15:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：TI2[0] 到 TI2[15] 输入选择 (TI2[0] to TI2[15] input selection)
这些位选择 TI2[0] 到 TI2[15] 输入源。
0000：TIM5_CH2 输入
其他值：保留

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：TI1[0] 到 TI1[15] 输入选择 (TI1[0] to TI1[15] input selection)
这些位选择 TI1[0] 到 TI1[15] 输入源。
0000：TIM5_CH1 输入
0001：fdcan1_tmp
0010：fdcan1_rtp
其他值：保留

39.4.30 TIMx 寄存器映射

TIMx 寄存器按照下表所述方式映射：

表 357. TIM2/TIM3/TIM4/TIM5 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIMx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMA	Res.	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0	0	0	0	0	0	0	0
0x04	**TIMx_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			CCDS	Res.	Res.	Res.
	Reset value																									0	0	0	0	0			
0x08	**TIMx_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS [4:3]		Res.	Res.	Res.	SMS[3]	ETP	ECE	ETPS [1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value											0	0				0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0x0C	**TIMx_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																		0		0	0	0	0	0		0		0	0	0	0	0
0x10	**TIMx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value																				0	0	0	0			0		0	0	0	0	0
0x14	**TIMx_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																										0		0	0	0	0	0
0x18	**TIMx_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	OC2CE	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2 PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**TIMx_CCMR2** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]	O24CE	OC4M [2:0]			OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR2 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]				IC4 PSC [1:0]		CC4S [1:0]		IC3F[3:0]				IC3 PSC [1:0]		CC3S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**TIMx_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
	Reset value																	0		0	0	0		0	0	0		0	0	0		0	0

表 357. TIM2/TIM3/TIM4/TIM5 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x24	**TIMx_CNT**	CNT[31] or UIFCPY	CNT[30:16] (TIM2 and TIM5 only, reserved on the other timers)															CNT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIMx_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIMx_ARR**	ARR[31:16] (TIM2 and TIM5 only, reserved on the other timers)																ARR[15:0]															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	Reserved																																
0x34	**TIMx_CCR1**	CCR1[31:16] (TIM2 and TIM5 only, reserved on the other timers)																CCR1[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIMx_CCR2**	CCR2[31:16] (TIM2 and TIM5 only, reserved on the other timers)																CCR2[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**TIMx_CCR3**	CCR3[31:16] (TIM2 and TIM5 only, reserved on the other timers)																CCR3[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	**TIMx_CCR4**	CCR4[31:16] (TIM2 and TIM5 only, reserved on the other timers)																CCR4[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	Reserved																																
0x48	**TIMx_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIMx_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**TIM2_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL [3:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value															0	0	0	0														
0x60	**TIM3_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL [3:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value															0	0	0	0														

表 357. TIM2/TIM3/TIM4/TIM5 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x60	**TIM4_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value															0	0	0	0														
0x60	**TIM5_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value															0	0	0	0														
0x68	**TIM2_TISEL**	Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]				Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0
0x68	**TIM3_TISEL**	Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]				Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0
0x68	**TIM4_TISEL**	Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]				Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0
0x68	**TIM5_TISEL**	Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]				Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

40 通用定时器 (TIM12/TIM13/TIM14)

40.1 TIM12/TIM13/TIM14 简介

TIM12/TIM13/TIM14 通用定时器包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

它们可用于多种用途，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（输出比较、PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

TIM12/TIM13/TIM14 定时器彼此完全独立，不共享任何资源。如*第 40.3.17 节：定时器同步 (TIM12)*中所述，它们可以一起同步操作。

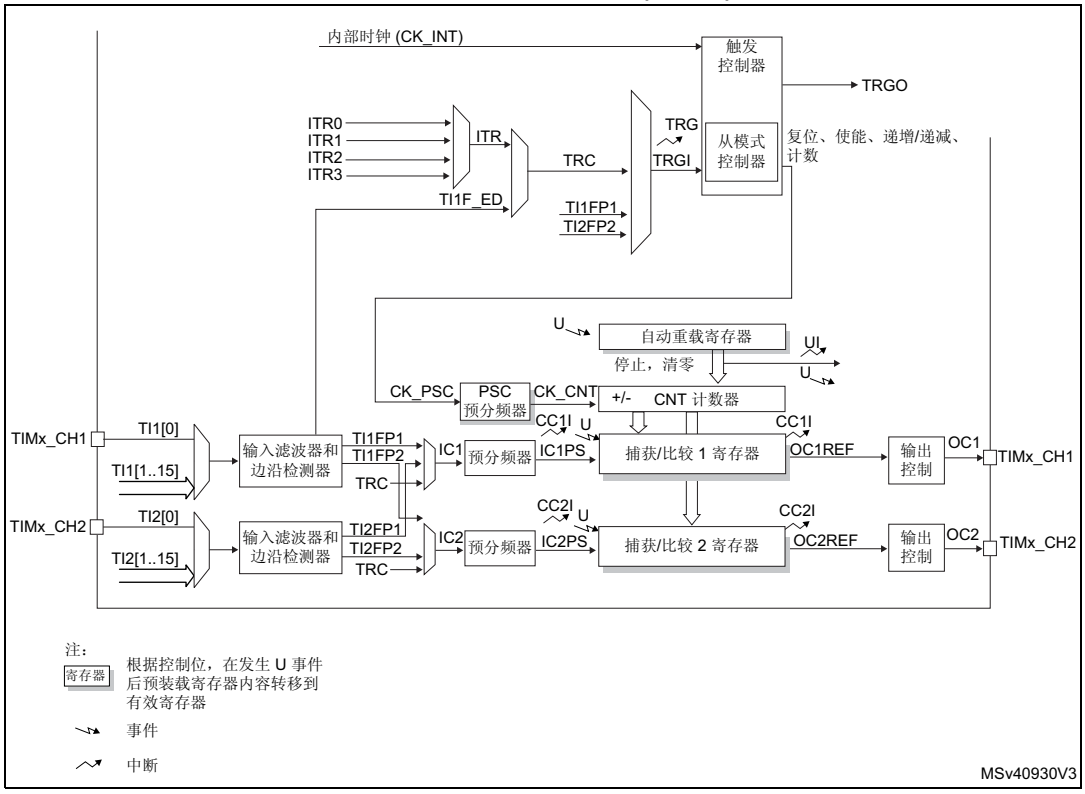
40.2 TIM12/TIM13/TIM14 主要特性

40.2.1 TIM12 主要特性

TIM12 通用定时器具有以下特性：

- 16 位自动重载递增计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65536 之间
- 多达 2 个独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿对齐模式）
  - 单脉冲模式输出
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路
- 发生如下事件时生成中断：
  - 更新：计数器上溢、计数器初始化（通过软件或内部触发）
  - 触发事件（计数器启动、停止、初始化或者由内部触发计数）
  - 输入捕获
  - 输出比较

图 445. 通用定时器框图 (TIM12)

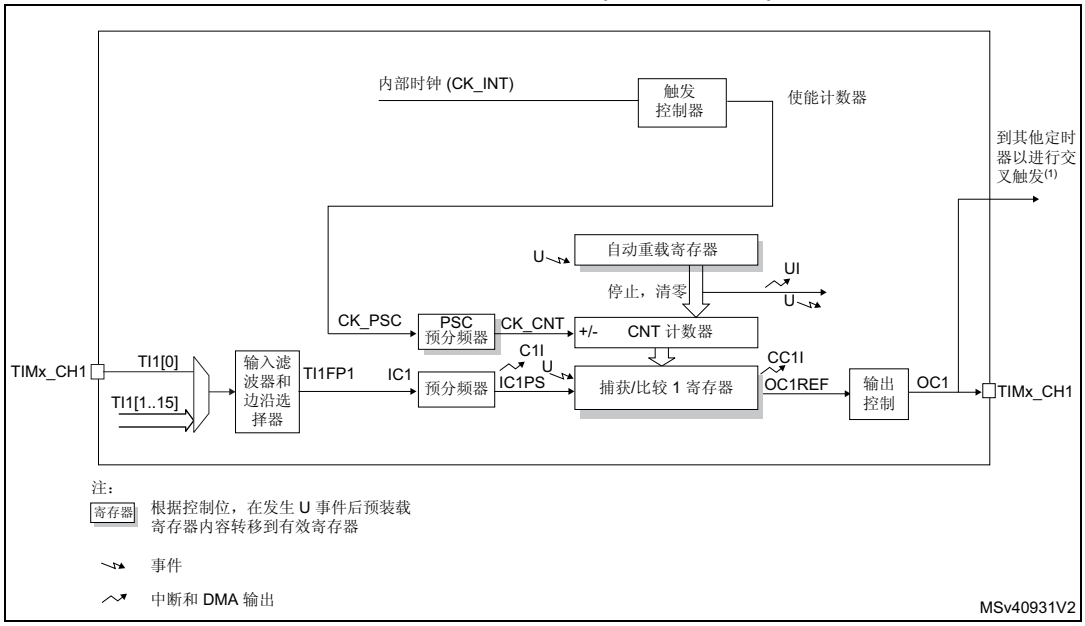


40.2.2 TIM13/TIM14 主要特性

通用定时器 TIM13/TIM14 具有以下特性：

- 16 位自动重载递增计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65536 之间
- 独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿对齐模式）
  - 单脉冲模式输出
- 发生如下事件时生成中断：
  - 更新：计数器上溢、计数器初始化（通过软件）
  - 输入捕获
  - 输出比较

图 446. 通用定时器框图 (TIM13/TIM14)



1. 该信号可用于触发一些从定时器，请参见*第 40.3.18 节：使用定时器输出作为其他定时器的触发 (TIM13/TIM14)*。

40.3 TIM12/TIM13/TIM14 功能描述

40.3.1 时基单元

定时器的主要模块由一个 16 位递增计数器及其相关的自动重载寄存器组成。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以随时直接写入影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将产生更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

注意，计数器将在 TIMx_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。

预分频器说明

预分频器可对计数器时钟进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 TIMx_PSC 寄存器中的 16 位寄存器所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 447 和*图 448* 以一些示例说明在预分频比实时变化时计数器的行为。

图 447. 预分频器分频由 1 变为 2 时的计数器时序图

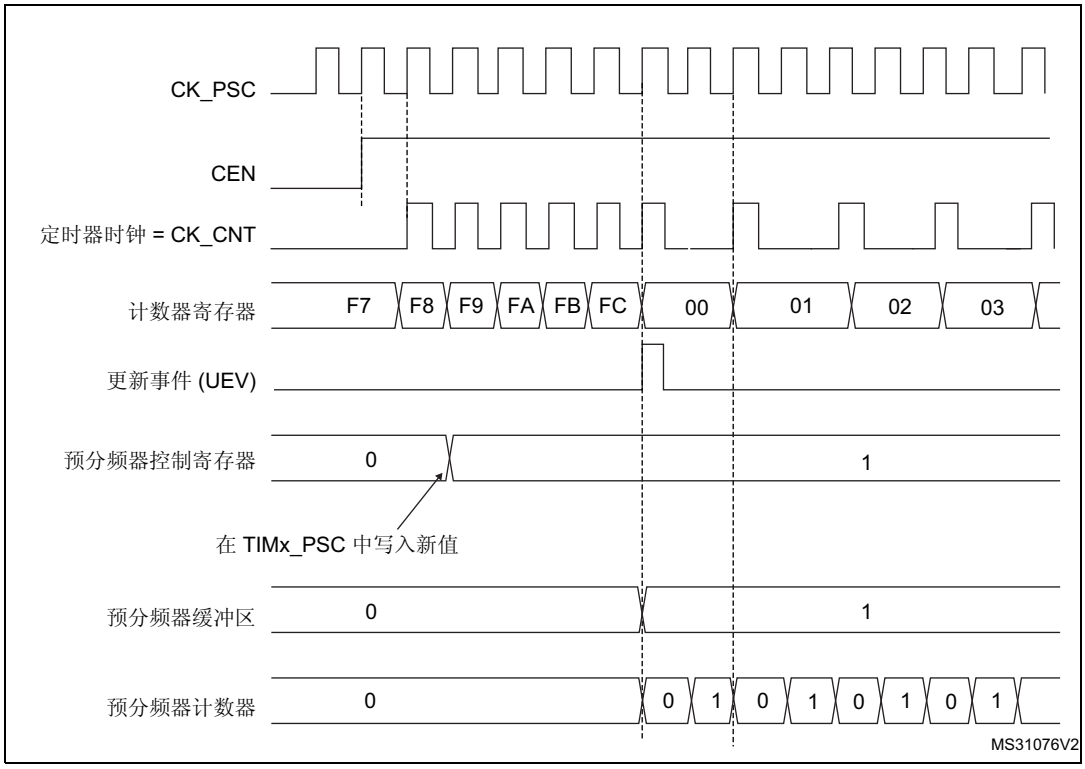
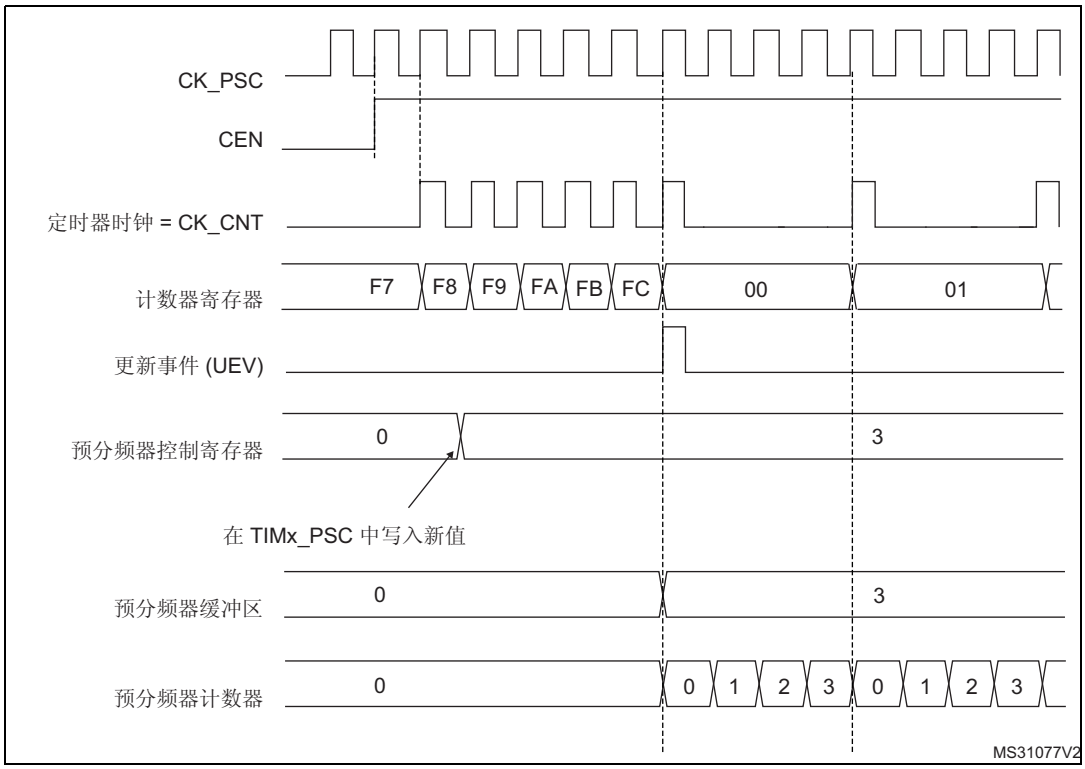


图 448. 预分频器分频由 1 变为 4 时的计数器时序图



40.3.2 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

否则，将在每次计数器上溢时产生更新事件。将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用 TIM12 上的从模式控制器）时，也会生成更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断）。利用这一特性，在定时器的输入捕获操作时可以通过对 UG 位置 1 让计数器复位而不至于触发更新中断，也就避免了在捕获中断里触发更新中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 自动重载影子寄存器将以预装载值 (TIMx_ARR) 进行更新，
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 449. 计数器时序图，1 分频内部时钟

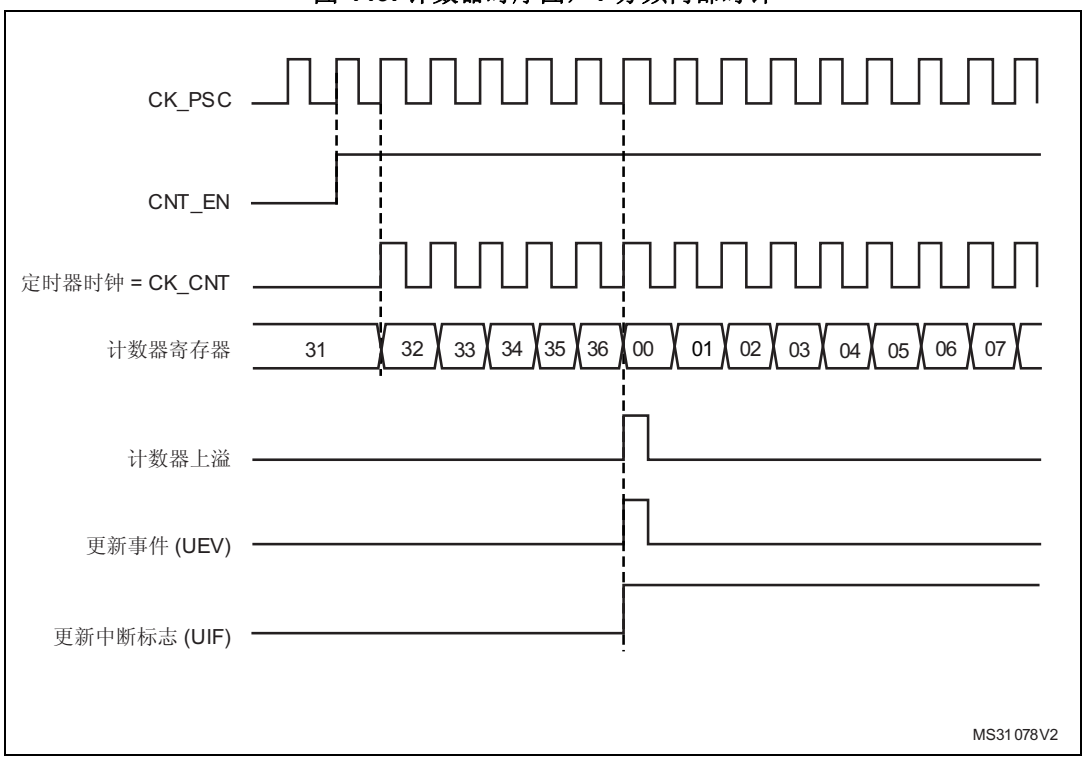


图 450. 计数器时序图，2 分频内部时钟

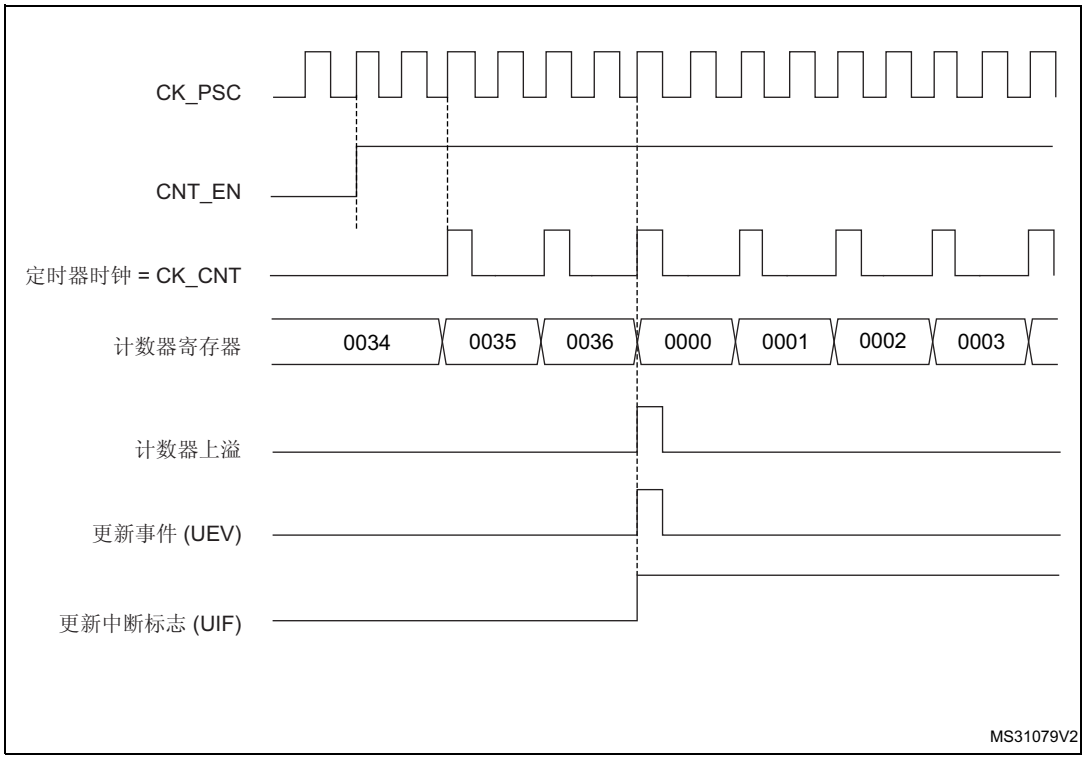


图 451. 计数器时序图，4 分频内部时钟

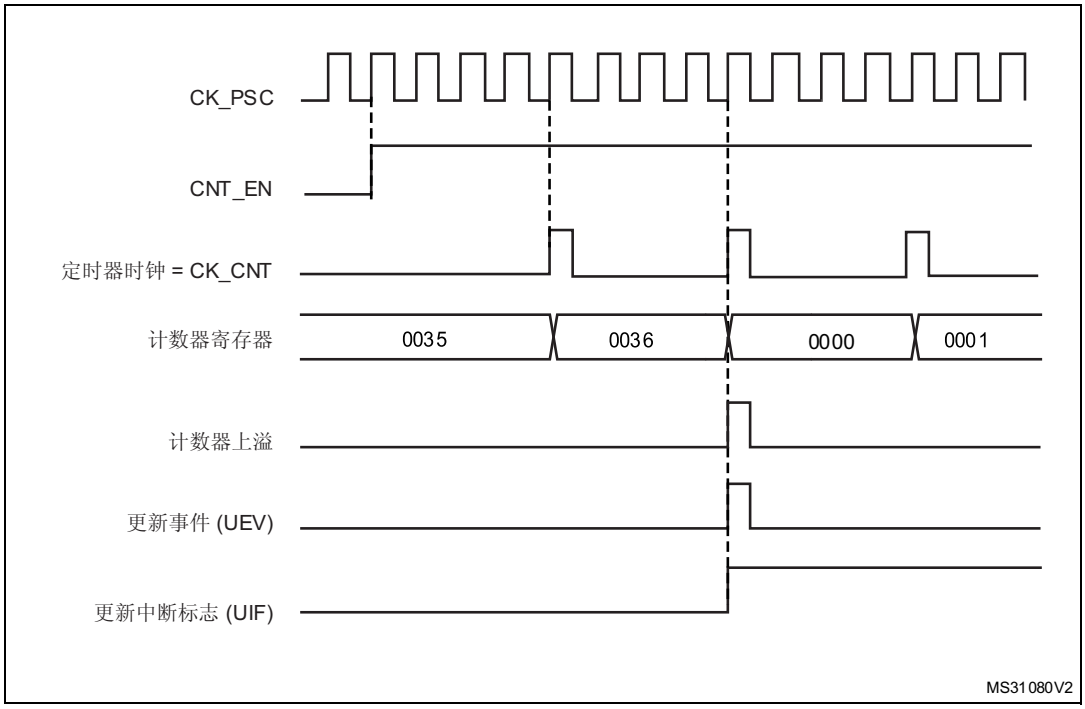


图 452. 计数器时序图，N 分频内部时钟

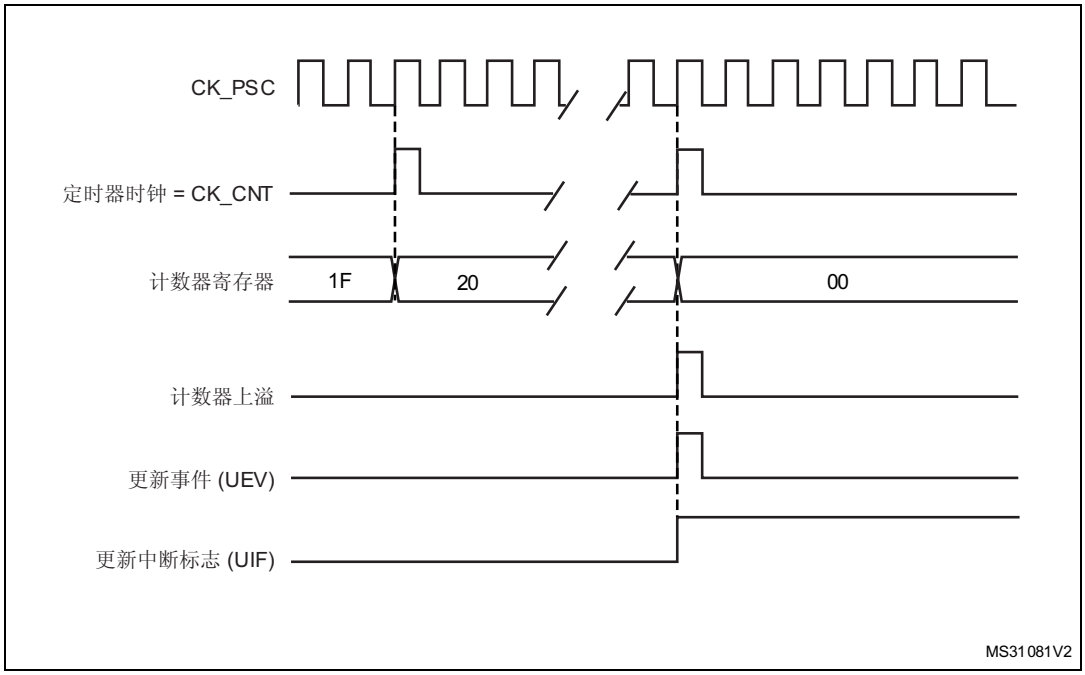


图 453. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

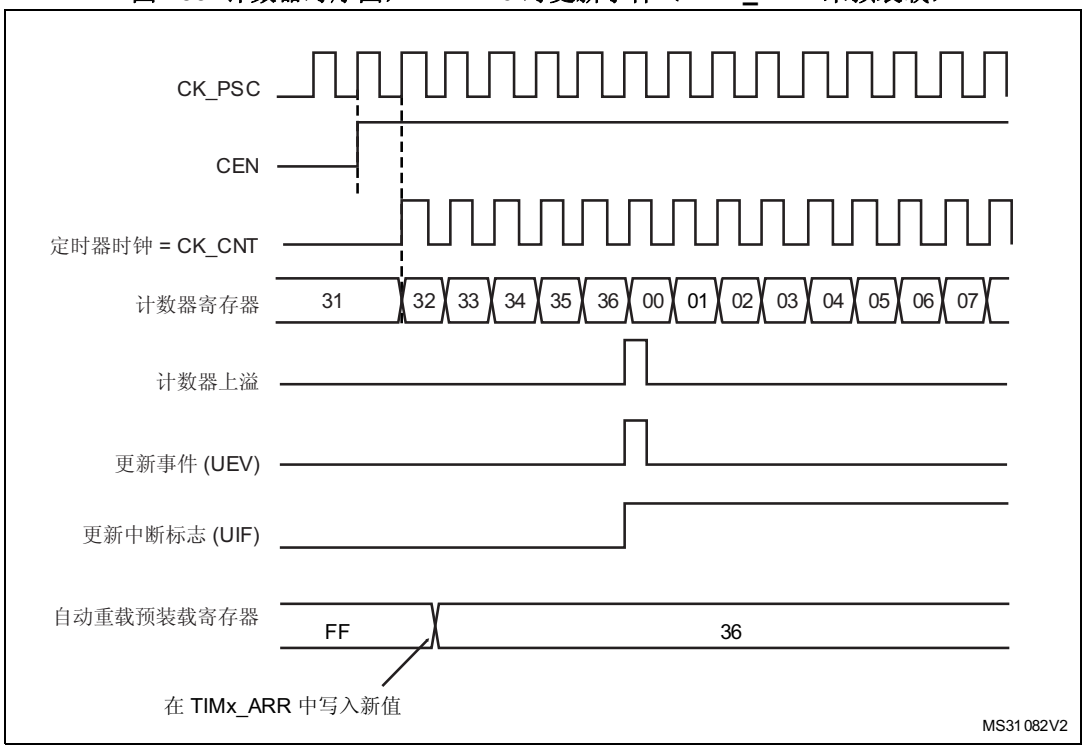
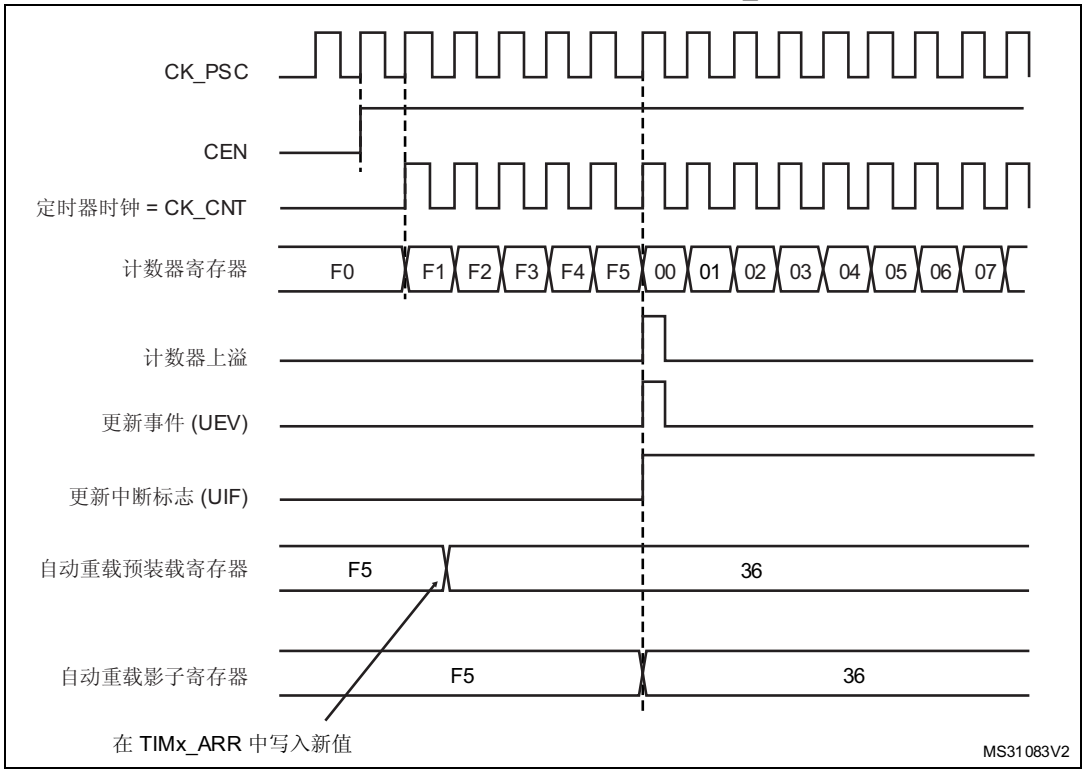


图 454. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载）



40.3.3 时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟 (CK_INT)
- 外部时钟模式 1（针对 TIM12）：外部输入引脚 (TIx)
- 内部触发输入 (ITRx)（针对 TIM12）：连接来自其它定时器的触发输出。例如，其它定时器可配置为 TIM12 的预分频器。更多详细信息，请参见*将一个定时器用作另一个定时器的预分频器*一节。

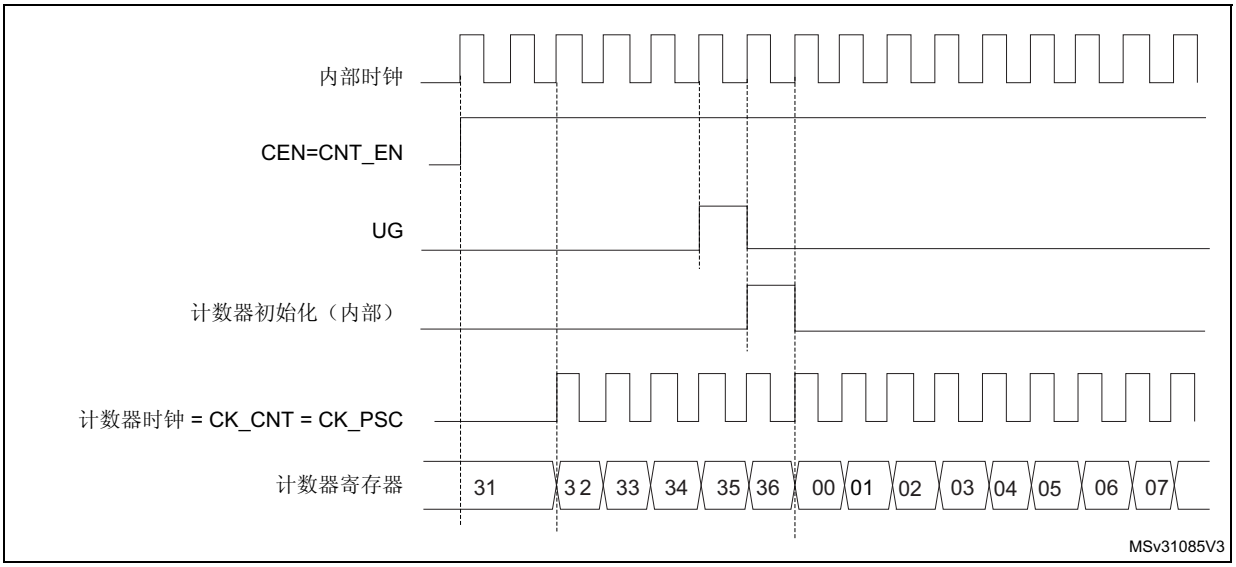
内部时钟源 (CK_INT)

内部时钟源是 TIM13/TIM14 的默认时钟源。

对于 TIM12，在禁止从模式控制器后选择内部时钟源（SMS = “000”）。然后，将 TIMx_CR1 寄存器中的 CEN 位和 TIMx_EGR 寄存器中的 UG 位用作控制位，并且只能通过软件对其进行更改（保持清零状态的 UG 除外）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 455 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

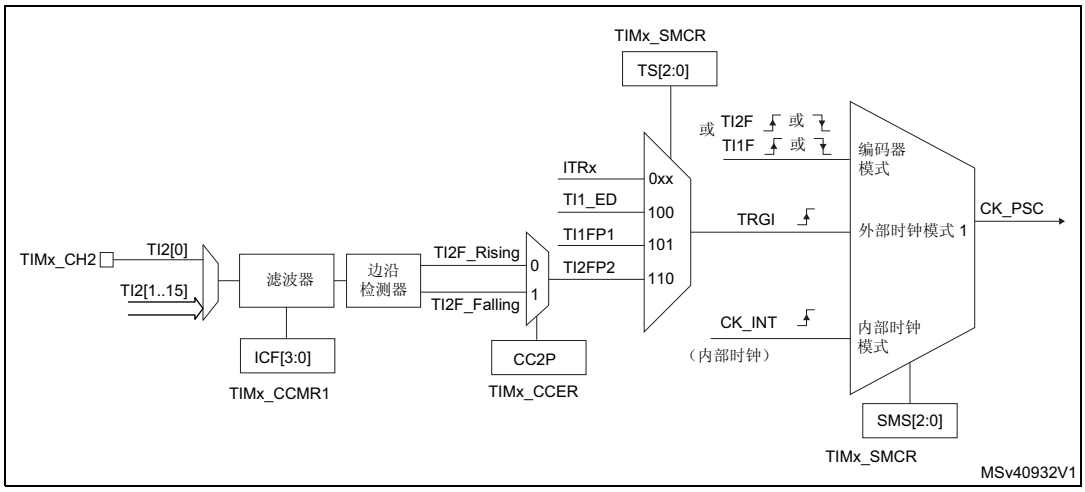
图 455. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1 (TIM12)

当 TIMx_SMCR 寄存器中的 SMS=“111”时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 456. TI2 外部时钟连接示例



例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

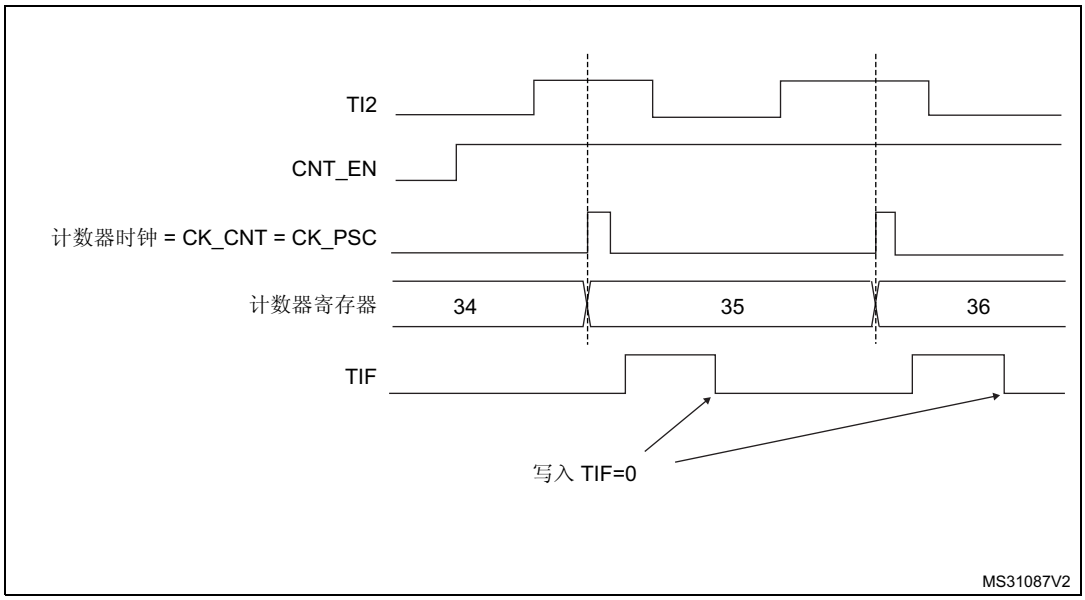
1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2[x] 源（内部或外部）。
2. 通过在 TIMx_CCMR1 寄存器中写入 CC2S = “01”来配置通道 2，使其能够检测 TI2 输入的上升沿。
3. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=“0000”）。
4. 通过在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”来选择上升沿极性。
5. 通过在 TIMx_SMCR 寄存器中写入 SMS=“111”，使定时器在外部时钟模式 1 下工作。
6. 通过在 TIMx_SMCR 寄存器中写入 TS=“110”来选择 TI2 作为触发输入源。
7. 通过在 TIMx_CR1 寄存器中写入 CEN=“1”来使能计数器。

注： *由于捕获预分频器不用于触发操作，因此无需对其进行配置。*

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 457. 外部时钟模式 1 下的控制电路



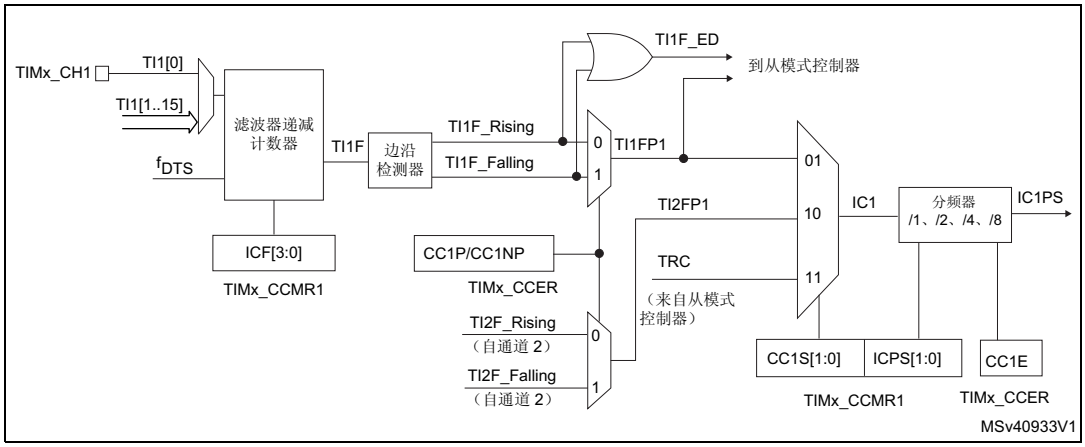
40.3.4 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入单元（数字滤波、多路复用和预分频器）和一个输出单元（比较器和输出控制）构建而成。

图 458 到*图 460* 简要介绍了一个捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 458. 捕获/比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 459. 捕获/比较通道 1 主电路

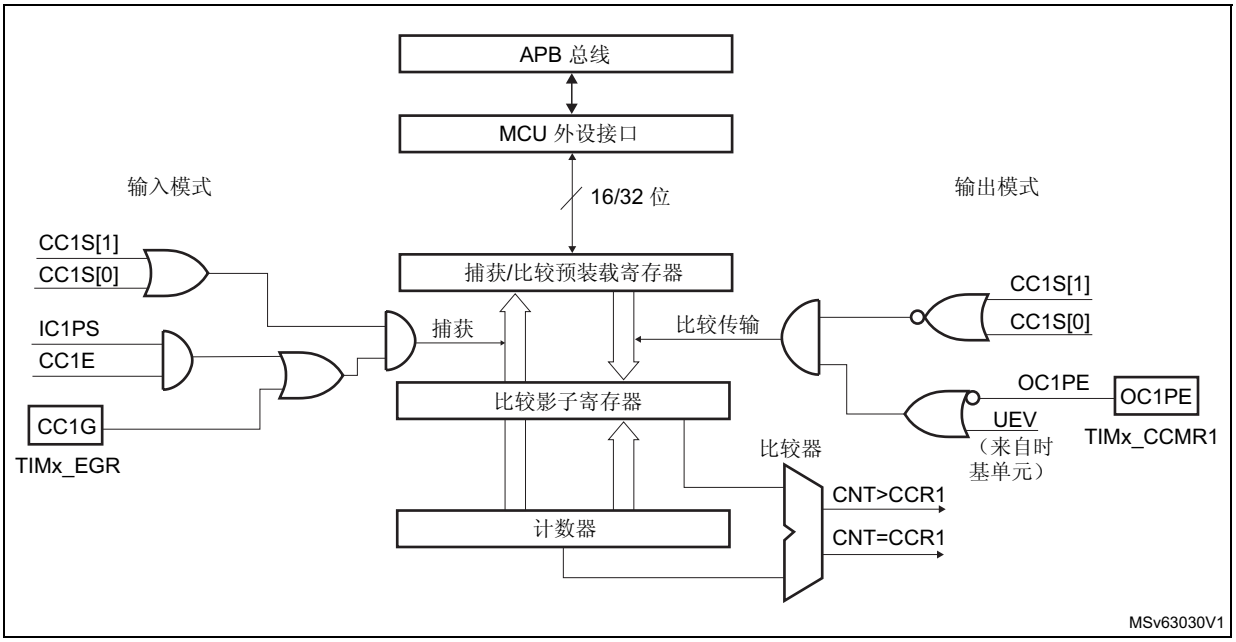
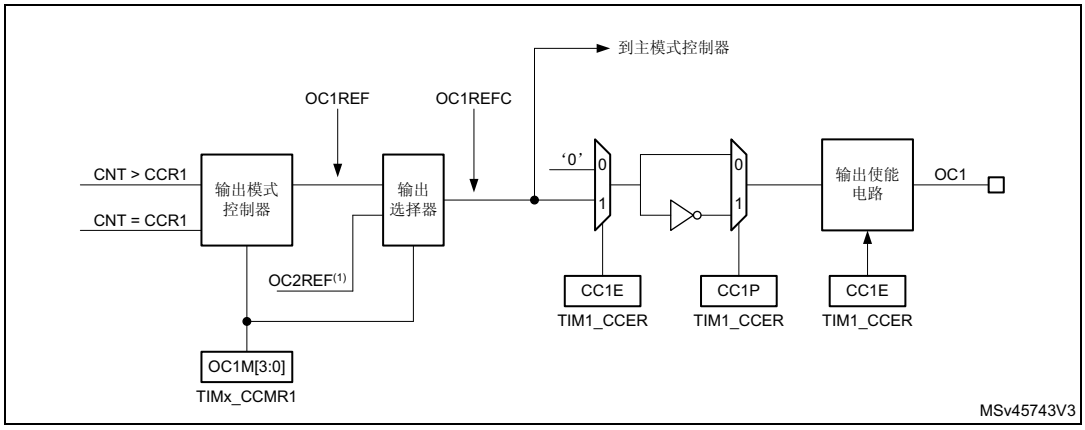


图 460. 捕获/比较通道的输出阶段（通道 1）



1. 只有 TIM12 提供。

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。读写访问的始终是预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

40.3.5 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将过捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件方法向 CCxIF 写入 0 来给 CCxIF 清零，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1[x] 源（内部或外部）。
2. 选择有效输入：TIMx_CCR1 必须关联到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入“01”。只要 CC1S 不等于“00”，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
3. 根据关联到定时器的信号，对相关输入滤波参数进行编程（如果输入为 TIx 输入之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设输入信号发生翻转时，信号需要 5 个内部时钟周期才能稳定，则必须设置滤波时间大于 5 个内部时钟周期。若在 TI1 上连续 8 次检测到新电平采样值（以 f_{DTS} 频率采样）判定沿跳变合法，然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入“0011”。
4. 通过在 TIMx_CCER 寄存器中将 CC1P 位和 CC1NP 位编程为“00”，选择 TI1 上的有效转换边沿（本例中为上升沿）。
5. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。
6. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
7. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。

要处理过捕获，建议在读取捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的过捕获信息。

注： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断请求。*

40.3.6 PWM 输入模式（仅适用于 TIM12）

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

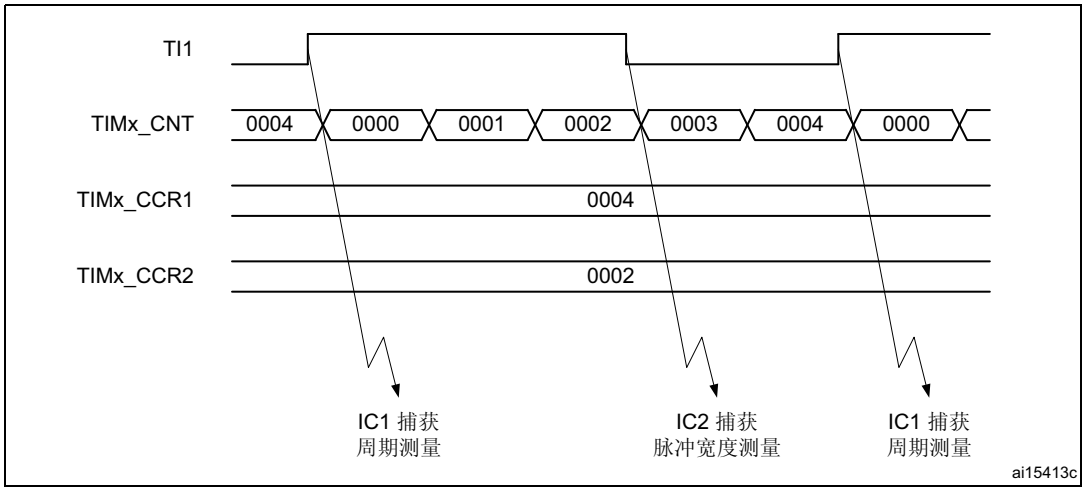
- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1[x] 源（内部或外部）。
2. 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入“01”（选择 TI1）。

3. 选择 TI1FP1 的有效极性（同时用于 TIMx_CCR1 中的捕获和计数器清零）：将 CC1P 位和 CC1NP 位编程为“00”（上升沿有效）。
4. 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 位写入“10”（选择 TI1）。
5. 选择 TI1FP2 的有效极性（用于 TIMx_CCR2 中的捕获）：将 CC2P 位和 CC2NP 位编程为“10”（下降沿有效）。
6. 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入“00101”（选择 TI1FP1）。
7. 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入“100”。
8. 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 461. PWM 输入模式时序



1. PWM 输入模式只能与 TIMx_CH1/TIMx_CH2 信号配合使用，因为只有 TI1FP1 和 TI2FP2 与从模式控制器相连。

40.3.7 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 = ‘00’）下，可直接由软件将每个输出比较信号（OCxREF 和 OCx）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号 (OCxREF/OCx) 强制设置为有效电平，用户只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入“0101”。OCxREF 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如：CCxP=“0”（OCx 高电平有效）=> 将 OCx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入“0100”，可将 OCxREF 信号强制设置为低电平。

当然，即使强制输出模式下，TIMx_CCRx 影子寄存器与计数器之间的比较仍然会执行，而且允许将标志置 1。因此可相应发送中断请求。下面的输出比较模式一节对此进行了介绍。

40.3.8 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

1. 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平（OCxM=“0000”），也可设置为有效电平（OCxM=“0001”）、无效电平（OCxM=“0010”）或进行翻转（OCxM=“0011”）。
2. 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
3. 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。

使用 TIMx_CCMRx 寄存器中的 OCxPE 位，可将 TIMx_CCRx 寄存器配置为带或不带预装功能。

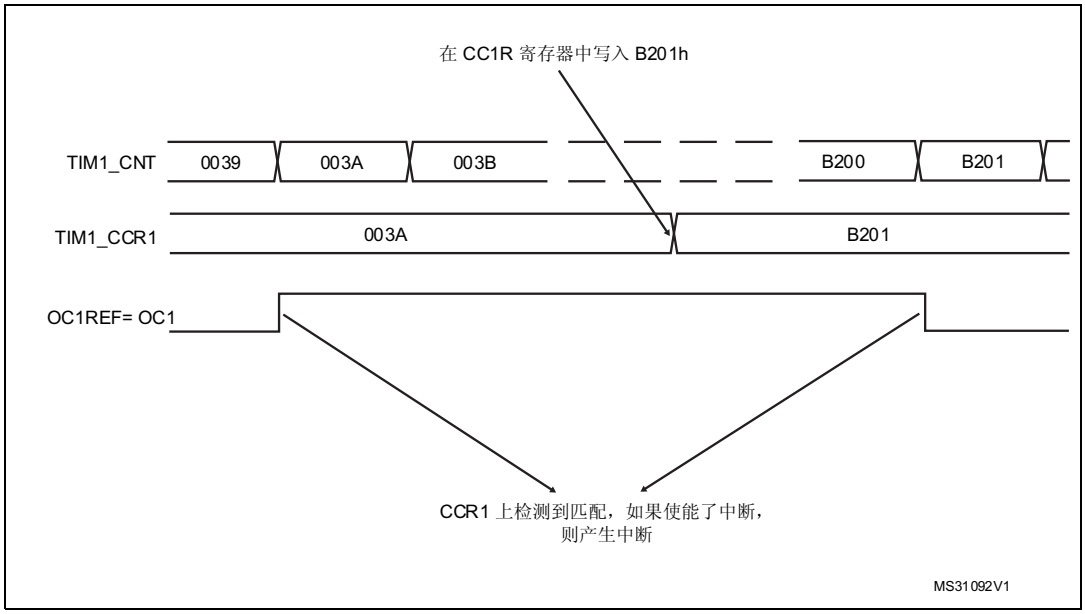
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。时间分辨率可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤：

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，则需将 CCxIE 位置 1。
4. 选择输出模式。例如：
   - 当 CNT 与 CCRx 匹配时，写入 OCxM =“0011”以翻转 OCx 输出引脚。
   - 写入 OCxPE =“0”以禁止预装功能
   - 写入 CCxP =“0”以选择高电平有效极性
   - 写入 CCxE =“1”以使能输出
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装功能（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIMx_CCRx 影子寄存器）。*图 462* 给出了一个示例。

图 462. 输出比较模式，翻转 OC1



40.3.9 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIMx_ARR 寄存器值决定，其占空比则由 TIMx_CCRx 寄存器值决定。

各通道可以独立选择 PWM 模式（每个 OCx 输出对应一个 PWM），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“0110”（PWM 模式 1）或“0111”（PWM 模式 2）。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

由于只有在发生更新事件时预装载寄存器内容才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

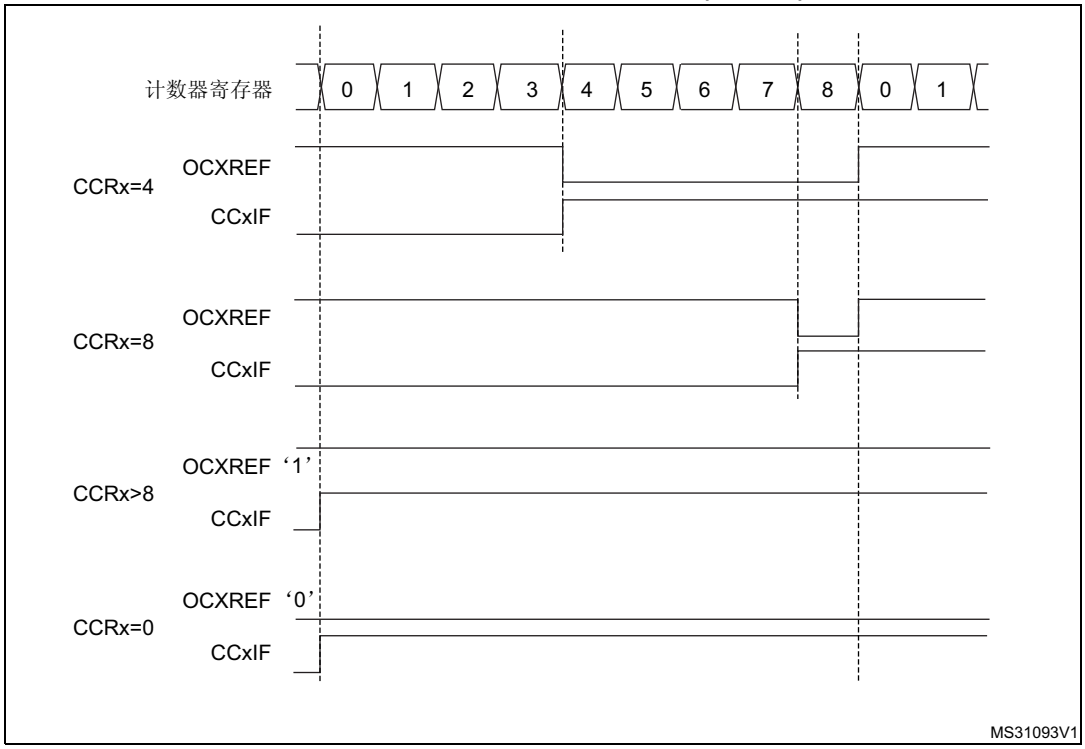
OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。OCx 输出通过将 TIMx_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息，请参见 TIMx_CCERx 寄存器说明。

在 PWM 模式（1 或 2）下，TIMx_CNT 始终与 TIMx_CCRx 进行比较，以确定 TIMx_CNT ≤ TIMx_CCRx 是否成立。

只有计数器采用递增方式计数后，定时器才能够在边沿对齐模式下生成 PWM。

以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxRef 保持为“0”。*图 463* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 463. 边沿对齐模式的 PWM 波形 (ARR=8)



40.3.10 组合 PWM 模式（仅适用于 TIM12）

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的“逻辑或”运算或者“逻辑与”运算组合组成。

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 寄存器控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

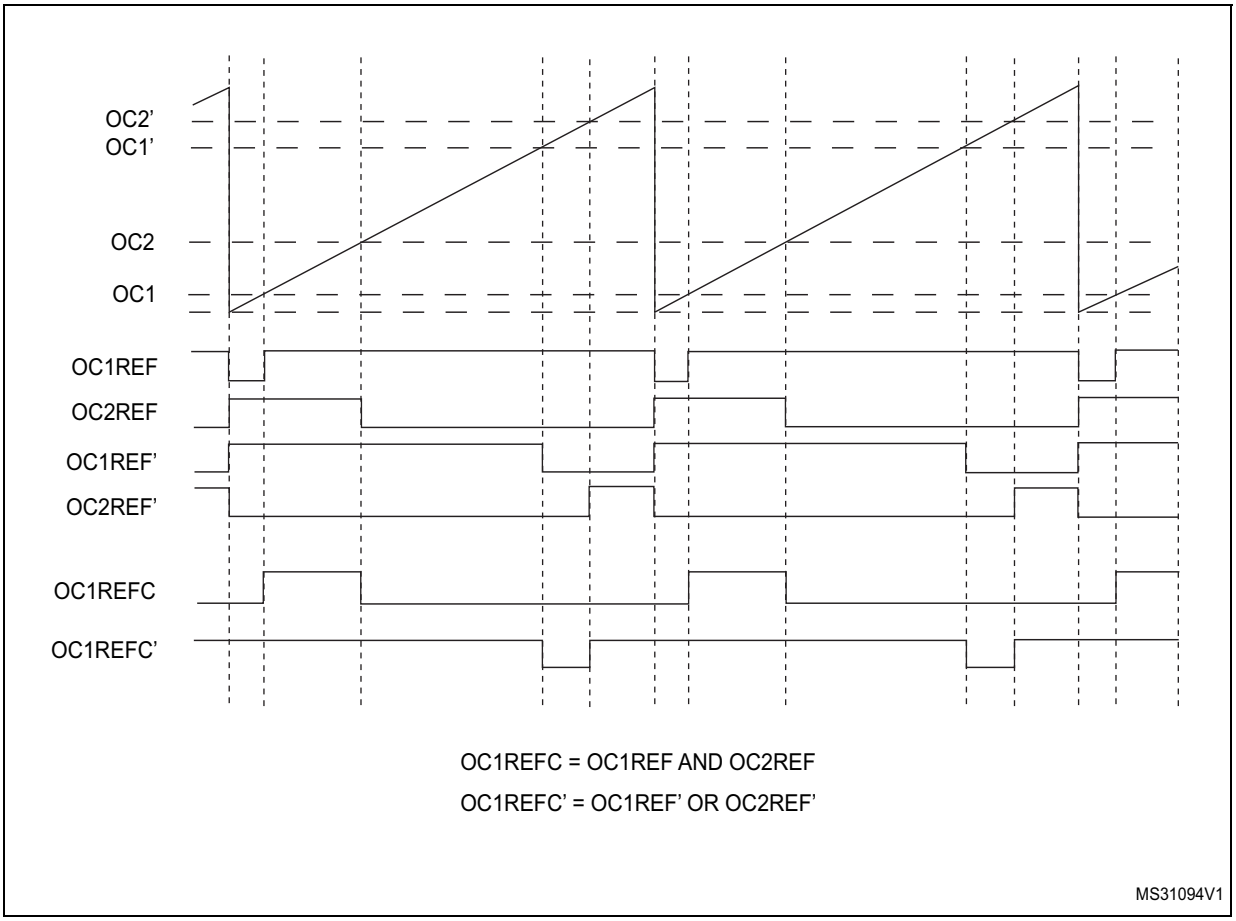
当给定通道用作组合 PWM 通道时，其互补通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注： *出于兼容性原因，OCxM[3:0] 位字段分为两部分，最高有效位与最低有效的 3 位不相邻。*

图 464 显示了组合 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 配置为组合 PWM 模式 2，
- 通道 2 配置为 PWM 模式 1，

图 464. 通道 1 和通道 2 上的组合 PWM 模式



40.3.11 单脉冲模式

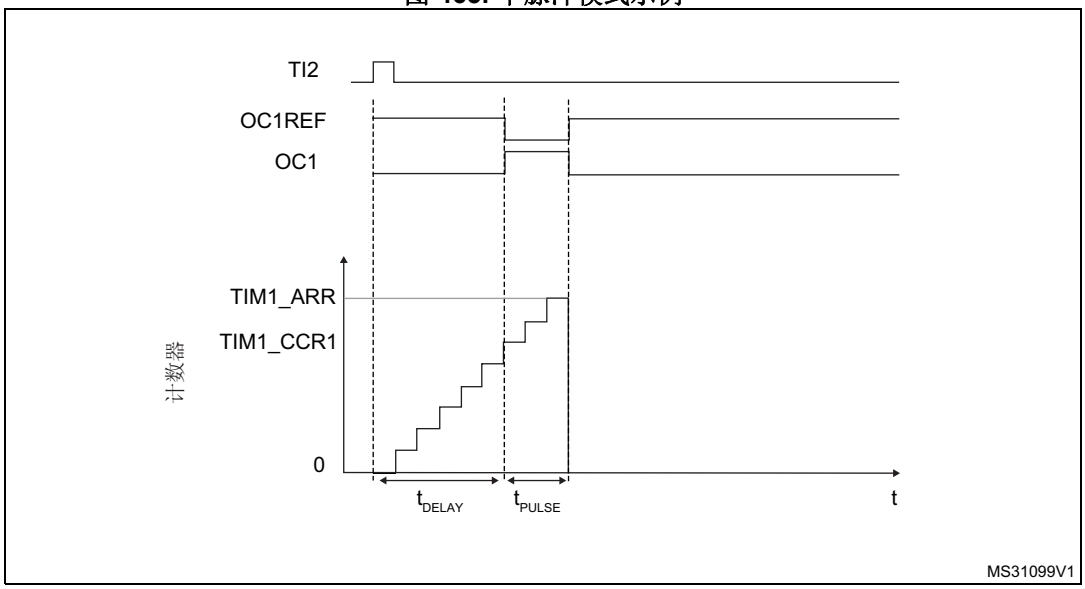
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。通过将 TIMx_CR1 寄存器中的 OPM 位置 1 选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

CNT < CCRx ≤ ARR（特别注意，0 < CCRx）

图 465. 单脉冲模式示例



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2[x] 源（内部或外部）。
2. 在 TIMx_CCMR1 寄存器中写入 CC2S=“01”，以将 TI2FP2 映射到 TI2。
3. 在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
4. 通过在 TIMx_SMCR 寄存器中写入 TS=“00110”，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
5. 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），以使用 TI2FP2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，必须通过在 TIMx_CCMR1 寄存器中写入 OC1M=“0111”，来使能 PWM 模式 2。可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

由于仅需要 1 个脉冲（单脉冲模式），因此必须向 TIMx_CR1 寄存器的 OPM 位写入 1，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIMx_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特例：OCx 快速使能

在单脉冲模式下，TIx 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟（t_{DELAY} 最小值）。

如果要输出延迟时间最短的波形，可以将 TIMx_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 OCxRef（和 OCx）对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

40.3.12 可再触发单脉冲模式（仅限 TIM12）

该模式可以启动计数器来响应激励信号，并且能产生长度可编程的脉冲，但与*第 40.3.11 节：单脉冲模式*所描述的不可再触发单脉冲模式间存在以下差别：

- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长

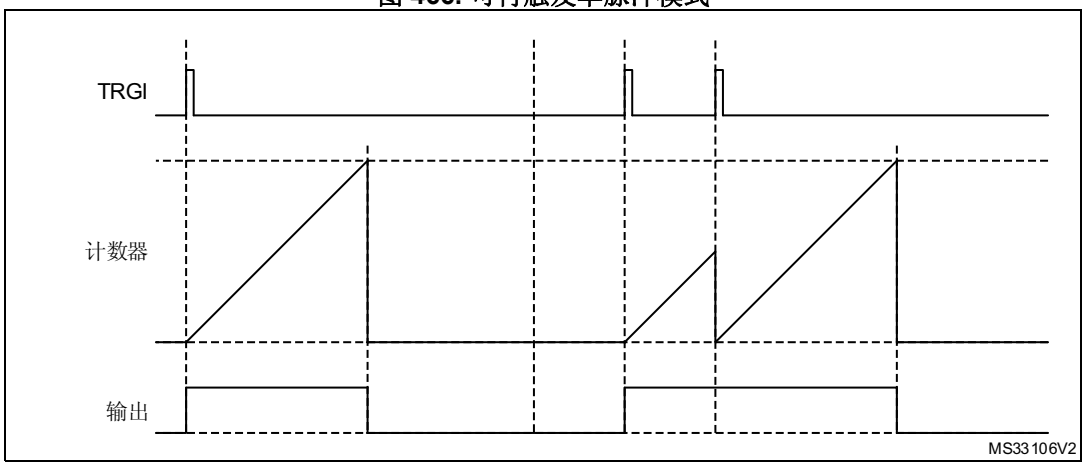
定时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0] = “1000”（组合复位 + 触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0] 位设置为“1000”或“1001”。

定时器配置为递增计数模式时，相应的 CCRx 必须置 0（ARR 寄存器设置脉冲长度）。如果定时器配置为递减计数模式，CCRx 必须高于或等于 ARR。

注： *出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位字段分为两部分，最高有效位与最低有效的 3 位不相邻。*

此模式不能与中心对齐 PWM 模式一起使用。在 TIMx_CR1 中必须设置 CMS[1:0] = 00。

图 466. 可再触发单脉冲模式



40.3.13 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的更新事件情况。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

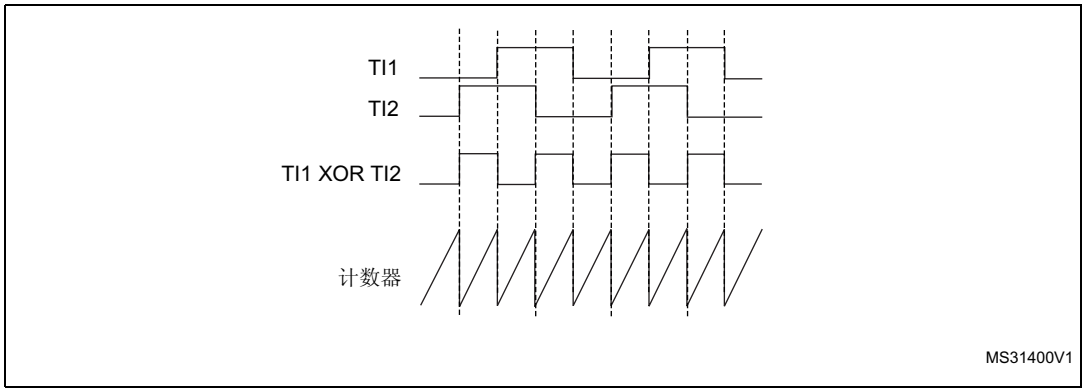
UIF 和 UIFCPY 标志之间的置位没有延迟。

40.3.14 定时器输入异或功能

通过 TIMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 和 TIMx_CH2 这两个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。这样可用于测量两个输入信号上边沿之间的间隔，如下面的*图 467* 所示。

图 467. 测量 2 个信号上边沿之间的时间间隔



40.3.15 TIM12 外部触发同步

TIM12 定时器可在多种模式下与外部触发实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

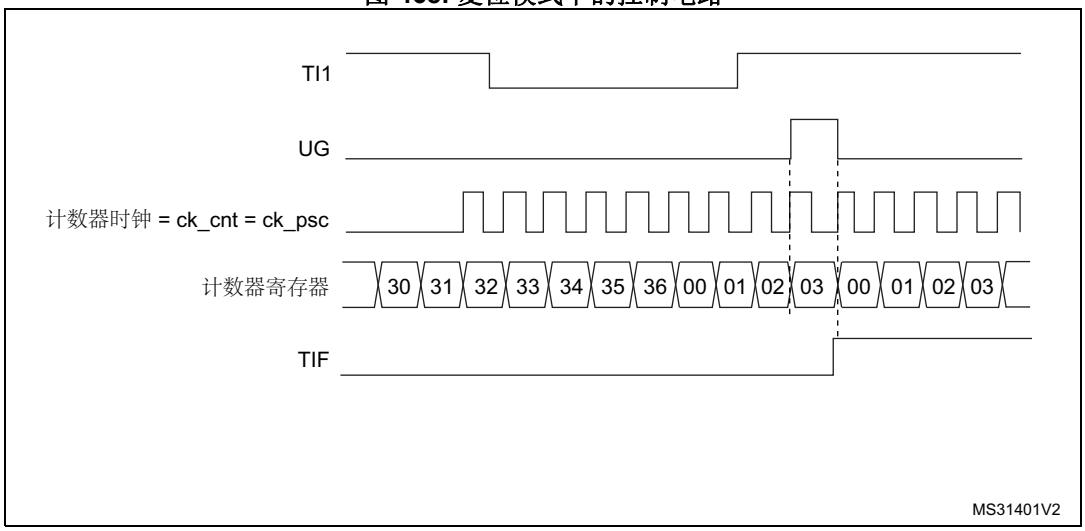
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

1. 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC1F=“0000”）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S = “01”。将 TIMx_CCER 寄存器中的 CC1P 和 CC1NP 编程为“00”，以确定极性（仅检测上升沿）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=“100”，将定时器配置为复位模式。通过在 TIMx_SMCR 寄存器中写入 TS=“00101”来选择 TI1 作为输入源。
3. 通过在 TIMx_CR1 寄存器中写入 CEN=“1”来启动计数器。

计数器开始根据内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断，还可发送中断请求（取决于 TIMx_DIER 寄存器中的 TIE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 468. 复位模式下的控制电路



从模式：门控模式

输入信号的电平可用来使能计数器。

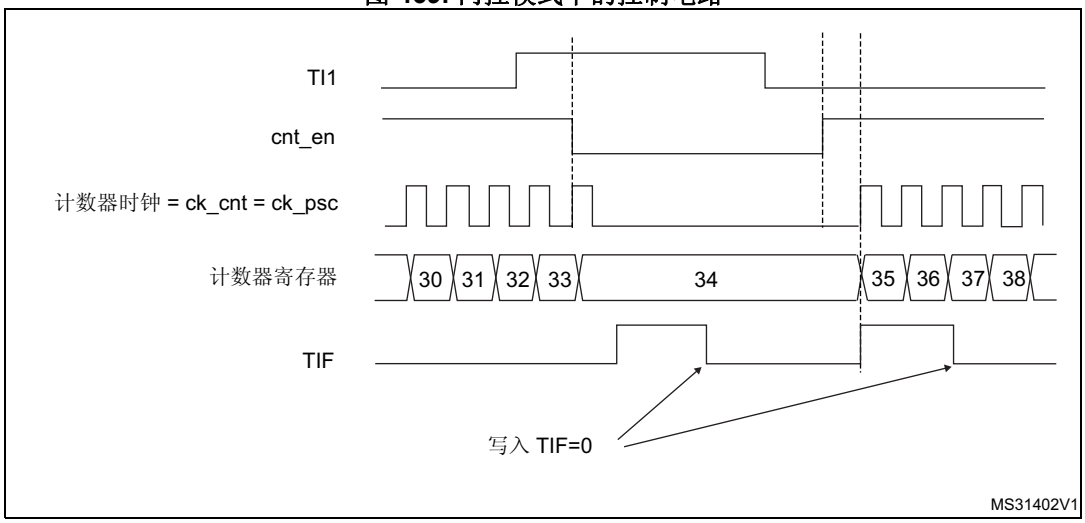
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

1. 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC1F=“0000”）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S = “01”。将 TIMx_CCER 寄存器中的 CC1P 和 CC1NP 分别编程为“1”和“0”，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=“101”，将定时器配置为门控模式。通过在 TIMx_SMCR 寄存器中写入 TS=“00101”来选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=“1”，使能计数器（在门控模式下，如果 CEN=“0”，则不论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 469. 门控模式下的控制电路



从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

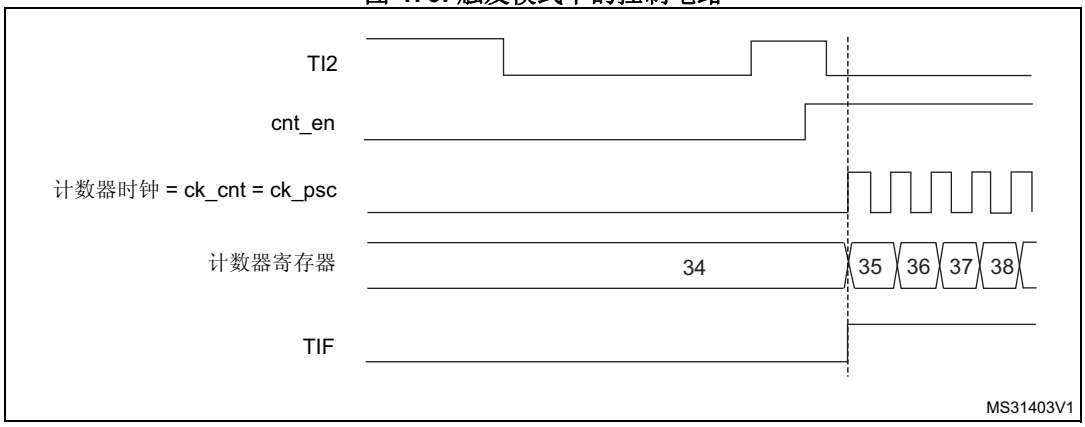
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

1. 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC2F=“0000”）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=“01”。将 TIMx_CCER 寄存器中的 CC2P 和 CC2NP 分别编程为“1”和“0”，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=“110”，将定时器配置为触发模式。通过在 TIMx_SMCR 寄存器中写入 TS=“00110”来选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 470. 触发模式下的控制电路



40.3.16 从模式——组合复位 + 触发模式

在这种情况下，在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个定时器更新事件，并启动计数器。

该模式用于单脉冲模式。

40.3.17 定时器同步 (TIM12)

TIM 定时器从内部链接在一起，以实现定时器同步或级联。有关详细信息，请参见*第 39.3.19 节：定时器同步*。

注： *必须先使能从定时器的时钟，才能从主定时器接收事件；从主定时器接收触发信号时，不得实时更改从定时器的时钟。*

40.3.18 使用定时器输出作为其他定时器的触发 (TIM13/TIM14)

单通道定时器没有主模式。但是，可以使用 OC1 输出信号来触发其他定时器（包括本文档其他部分所述的定时器）。通过查看关于设备上任何 TIMx_SMCR 寄存器的“TIMx 内部触发连接”表，可以确定哪些定时器可以作为目标从定时器。

OC1 信号的脉冲宽度必须至少设为目标定时器时钟周期的 2 倍，以便确保从定时器会检测到触发。

例如，如果目标定时器 CK_INT 时钟频率是源定时器的 1/4，则 OC1 的脉冲宽度必须至少是 8 个源定时器时钟周期。

40.3.19 调试模式

当系统进入调试模式（处理器内核停止）时，TIMx 计数器会根据 DBGMCU 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。更多详细信息，请参见*第 59.10.6 节：微处理器调试单元 (DBGMCU)*

40.4 TIM12 寄存器

有关寄存器说明中使用的缩写，请参见*第 1.2 节*。

外设寄存器的写访问仅支持半字（16 位）或字（32 位）。而读访问可支持字节（8 位）、半字（16 位）或字（32 位）。

40.4.1 TIM12 控制寄存器 1 (TIM12_CR1)

TIM12 control register 1

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIFRE MAP	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
				rw		rw	rw	rw				rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD[1:0]**：时钟分频 (Clock division)
该位字段指示定时器时钟 (CK_INT) 频率与数字滤波器所使用的采样时钟 (TIx) 之间的分频比。
00：$t_{DTS} = t_{CK_INT}$
01：$t_{DTS} = 2 \times t_{CK_INT}$
10：$t_{DTS} = 4 \times t_{CK_INT}$
11：保留

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)
0：TIMx_ARR 寄存器不进行缓冲。
1：TIMx_ARR 寄存器进行缓冲。

位 6:4 保留，必须保持复位值。

位 3 **OPM**：单脉冲模式 (One-pulse mode)
0：计数器在发生更新事件时不会停止计数
1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）。

位 2 **URS**：更新请求源 (Update request source)
该位由软件置 1 和清零，用以选择 UEV 事件源。
0：使能后，所有以下事件都会生成更新事件。此类事件包括：
– 计数器上溢
– 将 UG 位置 1
– 通过从模式控制器生成的更新事件
1：使能后，只有计数器上溢会生成更新中断。

位 1 **UDIS**：更新禁止 (Update disable)
该位由软件置 1 和清零，用以使能/禁止更新事件 (UEV) 生成。
0：使能 UEV。UEV 可通过以下事件之一生成：
– 计数器上溢
– 将 UG 位置 1
带缓冲的寄存器被加载为预加载数值。
1：禁止 UEV。不会生成 UEV，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。如果 UG 位置 1，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)
0：禁止计数器
1：使能计数器
在单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。
注意：只有事先通过软件将 CEN 位置 1，才可以使用外部时钟和门控
模式。而触发模式可通过硬件自动将 CEN 位置 1。

40.4.2 TIM12 控制寄存器 2 (TIM12_CR2)

TIM12 control register 2

偏移地址：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			Res.	Res.	Res.	Res.
								rw	rw	rw	rw				

位 15:8 保留，必须保持复位值。

位 7 **TI1S**：TI1 选择 (TI1 selection)

0：TIM12_CH1 引脚连接到 TI1 输入

1：TIM12_CH1、TIM12_CH2 引脚连接到 TI1 输入（异或组合）

位 6:4 **MMS[2:0]**：主模式选择 (Master mode selection)

这些位可选择主模式下将要发送到从定时器以实现同步 (TRGO) 的信息。这些位的组合如下：

000：复位——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。

001：使能——计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑与运算组合而成。当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。

010：更新——选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器。

011：比较脉冲——CC1IF 标志置 1 时（即使已为高电平），只要发生捕获或比较匹配，触发输出都会发送一个正脉冲。(TRGO)。

100：比较——OC1REFC 信号用作触发输出 (TRGO)

101：比较——OC2REFC 信号用作触发输出 (TRGO)

位 3:0 保留，必须保持复位值。

40.4.3 TIM12 从模式控制寄存器 (TIM12_SMCR)

TIM12 slave mode control register

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS[4:3]		Res.	Res.	Res.	SMS[3]
										rw	rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSM	TS[2:0]			Res.	SMS[2:0]		
								rw	rw	rw	rw		rw	rw	rw

位 31:22 保留，必须保持复位值。

位 19:17 保留，必须保持复位值。

位 15:8 保留，必须保持复位值。

位 7 **MSM：**主/从模式 (Master/Slave mode)

0：无操作

1：当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于要在发生单个外部事件时对多个定时器进行同步的情况。

位 21、20、6、5、4 **TS[4:0]：**触发选择

TS[4:0] 位字段可选择将要用于同步计数器的触发输入。

00000：内部触发 0 (ITR0)

00001：内部触发 1 (ITR1)

00010：内部触发 2 (ITR2)

00011：内部触发 3 (ITR3)

00100：TI1 边沿检测器 (TI1F_ED)

00101：滤波后的定时器输入 1 (TI1FP1)

00110：滤波后的定时器输入 2 (TI2FP2)

其他值：保留

有关各定时器的 ITRx 含义的更多详细信息，请参见*第 2087 页的表 358：TIMx 内部触发连接*。

注意：这些位只能在未使用的情况下（例如，SMS=“000”时）进行更改，以避免转换时出现错误的边沿检测。

位 3 保留，必须保持复位值。

位 16、2、1、0 **SMS[3:0]：**从模式选择 (Slave mode selection)

选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。

0000：禁止从模式——如果 CEN = “1”，预分频器时钟直接由内部时钟提供。

0001：保留

0010：保留

0011：保留

0100：复位模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。

0101：门控模式——触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。

0110：触发模式——触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。

0111：外部时钟模式 1——由所选触发信号 (TRGI) 的上升沿提供计数器时钟。

1000：组合复位 + 触发模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。

其他代码：保留。

注意：如果将 TI1F_ED 选作触发输入（TS=“00100”），则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。

注意：必须先使能从定时器的时钟，才能从主定时器接收事件；从主定时器接收触发信号时，不得实时更改从定时器的时钟。

表 358. TIMx 内部触发连接

从 TIM	ITR0 （TS = “00000”）	ITR1 （TS = “00001”）	ITR2 （TS = “00010”）	ITR3 （TS = “00011”）
TIM12	TIM4	TIM5	TIM13 OC1	TIM14 OC1

40.4.4 中断使能寄存器 (TIM12_DIER)

TIM12 Interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIE	Res.	Res.	Res.	CC2IE	CC1IE	UIE
									rw				rw	rw	rw

位 15:7 保留，必须保持复位值。

位 6 **TIE**：触发信号中断使能 (Trigger interrupt enable)
0：禁止触发信号中断。
1：使能触发信号中断。

位 5:3 保留，必须保持复位值。

位 2 **CC2IE**：捕获/比较 2 中断使能 (Capture/Compare 2 interrupt enable)
0：禁止 CC2 中断。
1：使能 CC2 中断。

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断。
1：使能 CC1 中断。

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断。
1：使能更新中断。

40.4.5 TIM12 状态寄存器 (TIM12_SR)

TIM12 status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	Res.	TIF	Res.	Res.	Res.	CC2IF	CC1IF	UIF
					rc_w0	rc_w0			rc_w0				rc_w0	rc_w0	rc_w0

位 15:11 保留，必须保持复位值。

位 10 **CC2OF**：捕获/比较 2 过捕获标志 (Capture/compare 2 overcapture flag)
请参见 CC1OF 说明

位 9 **CC1OF**：捕获/比较 1 过捕获标志 (Capture/Compare 1 overcapture flag)
仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。
0：未检测到过捕获。
1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8:7 保留，必须保持复位值。

位 6 **TIF**：触发中断标志 (Trigger interrupt flag)

在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到 TRG 触发事件有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。

0：未发生触发事件。

1：触发中断挂起。

位 5:3 保留，必须保持复位值。

位 2 **CC2IF**：捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag)

请参见 CC1IF 说明

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/compare 1 interrupt flag)

该标志由硬件置 1，但需要通过软件清零（输入捕获或输出比较模式）或通过读取 TIMx_CCR1 寄存器清零（仅限输入捕获模式）。

0：未发生比较匹配/输入捕获

1：发生了比较匹配/输入捕获

如果通道 CC1 配置为输出：当计数器 TIMx_CNT 的值与 TIMx_CCR1 的值匹配时，此标志置 1。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。在中心对齐模式下，可通过 3 种方式将此标志置 1，完整说明请参见 TIMx_CR1 寄存器中的 CMS 位。

如果通道 CC1 配置为输入：当 TIMx_CCR1 寄存器中捕获到计数器值时（如果通过设置 TIMx_CCER 中的 CC1P 和 CC1NP 位定义为边沿有效，则改为在 IC1 上检测到边沿时），该位置 1。

位 0 **UIF**：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

– 上溢且当 TIMx_CR1 寄存器中 UDIS = “0”时。

– 当由于 TIMx_CR1 寄存器中 URS = “0”且 UDIS = “0”而通过软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

– 当由于 TIMx_CR1 寄存器中 URS = “0”且 UDIS = “0”而通过触发事件（请参见同步控制寄存器说明）重新初始化 CNT 时。

40.4.6 TIM12 事件生成寄存器 (TIM12_EGR)

TIM12 event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	Res.	Res.	CC2G	CC1G	UG
									w				w	w	w

位 15:7 保留，必须保持复位值。

位 6 **TG**：触发生成 (Trigger generation)

该位由软件置 1 以生成事件，并由硬件自动清零。

0：无操作

1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断

位 5:3 保留，必须保持复位值。

位 2 **CC2G**：捕获/比较 2 生成 (Capture/compare 2 generation)
请参见 CC1G 说明

位 1 **CC1G**：捕获/比较 1 生成 (Capture/compare 1 generation)
该位由软件置 1 以生成事件，并由硬件自动清零。
0：无操作
1：通道 1 上生成捕获/比较事件：
如果通道 CC1 配置为输出：
使能后，CC1IF 标志置 1 并发送相应中断。
如果通道 CC1 配置为输入：
TIMx_CCR1 寄存器中将捕获到计数器的当前值。使能后，CC1IF 标志置 1 并发送相应中断。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)
该位可通过软件置 1，并由硬件自动清零。
0：无操作
1：重新初始化计数器并生成寄存器更新事件。预分频器计数器也将清零，但预分频比不受影响。计数器清零。

40.4.7 TIM12 捕获/比较模式寄存器 1 (TIM12_CCMR1)

TIM12 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。该寄存器的所有其它位在输入模式和输出模式下的功能不同。

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:12 **IC2F[3:0]**：输入捕获 2 滤波器 (Input capture 2 filter)

位 11:10 **IC2PSC[1:0]**：输入捕获 2 预分频器 (Input capture 2 prescaler)

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/compare 2 selection)
该位字段定义通道方向（输入/输出）以及所使用的输入。
00：CC2 通道配置为输出
01：CC2 通道配置为输入，IC2 映射到 TI2 上
10：CC2 通道配置为输入，IC2 映射到 TI1 上
11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。

位 7:4 **IC1F[3:0]**：输入捕获 1 滤波器 (Input capture 1 filter)

该位字段可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样

0001：$f_{SAMPLING}=f_{CK_INT}$，N=2

0010：$f_{SAMPLING}=f_{CK_INT}$，N=4

0011：$f_{SAMPLING}=f_{CK_INT}$，N=8

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC[1:0]**：输入捕获 1 预分频器 (Input capture 1 prescaler)

该位字段用于定义 CC1 输入 (IC1) 的预分频比。

只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获

01：每发生 2 个事件便执行一次捕获

10：每发生 4 个事件便执行一次捕获

11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

10：CC1 通道配置为输入，IC1 映射到 TI2 上

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

40.4.8 TIM12 捕获/比较模式寄存器 1 [复用] (TIM12_CCMR1)

TIM12 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。该寄存器的所有其它位在输入模式和输出模式下的功能不同。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 23:17 保留，必须保持复位值。

位 15 保留，必须保持复位值。

位 24、14:12 **OC2M[3:0]**：输出比较 2 模式 (Output compare 2 mode)
有关位说明，请参见 OC1M[3:0]。

位 11 **OC2PE**：输出比较 2 预装载使能 (Output compare 2 preload enable)

位 10 **OC2FE**：输出比较 2 快速使能 (Output compare 2 fast enable)

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/Compare 2 selection)
该位字段定义通道方向（输入/输出）以及所使用的输入。
00：CC2 通道配置为输出
01：CC2 通道配置为输入，IC2 映射到 TI2 上
10：CC2 通道配置为输入，IC2 映射到 TI1 上
11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。

位 7 保留，必须保持复位值。

位 16，6:4 **OC1M[3:0]**：输出比较 1 模式 (Output compare 1 mode)（请参见 OC1M[3] 的位 16）

这些位定义提供 OC1 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 的有效电平则取决于 CC1P。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。当定时器用作软件时基时，可以使用该模式。在定时器操作期间使能冻结模式时，输出保持进入冻结状态前的状态（有效或无效）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转

0100：强制变为无效电平——OC1REF 强制变为低电平

0101：强制变为有效电平——OC1REF 强制变为高电平

0110：PWM 模式 1——只要 TIMx_CNT < TIMx_CCR1，通道 1 便为有效状态，否则为无效状态

0111：PWM 模式 2——只要 TIMx_CNT < TIMx_CCR1，通道 1 便为无效状态，否则为有效状态

1000：可再触发 OPM 模式 1——通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。

1001：可再触发 OPM 模式 2——通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。

1010：保留。

1011：保留。

1100：组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。

1101：组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。

1110: 保留。

1111：保留。

注意：在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。

注意：OC1M[3] 位不是连续的，而是在位 16 中。

位 3 **OC1PE**：输出比较 1 预装载使能 (Output compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到有效寄存器中

位 2 **OC1FE**：输出比较 1 快速使能 (Output compare 1 fast enable)

该位可降低触发事件与定时器输出跳变之间的延迟。单脉冲模式下必须使用（TIMx_CR1 寄存器中的 OPM 位置 1），以便在启动触发后快速发送输出脉冲。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OC1FE 才会起作用。

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)
该位字段定义通道方向（输入/输出）以及所使用的输入。
00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
10：CC1 通道配置为输入，IC1 映射到 TI2 上
11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

40.4.9 TIM12 捕获/比较使能寄存器 (TIM12_CCER)

TIM12 capture/compare enable register

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
								rw		rw	rw	rw		rw	rw

位 15:8 保留，必须保持复位值。

位 7 **CC2NP**：捕获/比较 2 输出极性 (Capture/Compare 2 output polarity)
请参见 CC1NP 说明

位 6 保留，必须保持复位值。

位 5 **CC2P**：捕获/比较 2 输出极性 (Capture/Compare 2 output polarity)
请参见 CC1P 说明

位 4 **CC2E**：捕获/比较 2 输出使能 (Capture/Compare 2 output enable)
请参见 CC1E 说明

位 3 **CC1NP**：捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output polarity)
CC1 通道配置为输出：CC1NP 必须保持清零
CC1 通道配置为输入：CC1NP 与 CC1P 配合使用可定义 TI1FP1/TI2FP1 的极性（请参见 CC1P 说明）。

位 2 保留，必须保持复位值。

位 1 **CC1P**：捕获/比较 1 输出极性 (Capture/Compare 1 output Polarity)。

0：OC1 高电平有效（输出模式）/边沿有效性选择（输入模式，见下文）

1：OC1 低电平有效（输出模式）/边沿有效性选择（输入模式，见下文）

CC1 通道配置为输入时，CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。

CC1NP=0，CC1P=0：非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。

CC1NP=0，CC1P=1：反相/下降沿触发。电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。

CC1NP=1，CC1P=1：非反相/上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。

CC1NP=1，CC1P=0：该配置保留，不得使用。

位 0 **CC1E**：捕获/比较 1 输出使能 (Capture/Compare 1 output enable)。

0：禁止捕获模式/OC1 无效

1：使能捕获模式/在相应输出引脚上输出 OC1 信号

表 359. 标准 OCx 通道的输出控制位

CCxE 位	OCx 输出状态
0	禁止输出（不由定时器驱动：高阻态）
1	使能输出（tim_ocx = tim_ocxref + 极性）

注：　与标准 OCx 通道相连的外部 I/O 引脚的状态取决于通道 OCx 的状态以及 GPIO 寄存器。

40.4.10 TIM12 计数器 (TIM12_CNT)

TIM12 counter

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIF CPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **UIFCPY**：UIF 副本 (UIF Copy)

该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

40.4.11 TIM12 预分频器 (TIM12_PSC)

TIM12 prescaler

偏移地址：0x28

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)

计数器时钟频率 CK_CNT 等于 f_{CK_PSC} / (PSC[15:0] + 1)。

PSC 存放着每次发生更新事件时要加载到有效预分频器寄存器的值。

（包括当计数器通过 TIMx_EGR 寄存器的 UG 位或在“复位模式”下通过触发控制器清零时）。

40.4.12 TIM12 自动重载寄存器 (TIM12_ARR)

TIM12 auto-reload register

偏移地址：0x2C

复位值：0xFFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)

ARR 为要装载到有效自动重载寄存器的值。

有关 ARR 更新和行为的更多详细信息，请参见*第 2064 页的第 40.3.1 节：时基单元*。

当自动重载值为空时，计数器不工作。

40.4.13 TIM12 捕获/比较寄存器 1 (TIM12_CCR1)

TIM12 capture/compare register 1

偏移地址：0x34

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR1[15:0]**：捕获/比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：

CCR1 是捕获/比较寄存器 1 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 1）。

实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。

如果通道 CC1 配置为输入：

CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。

40.4.14 TIM12 捕获/比较寄存器 2 (TIM12_CCR2)

TIM12 capture/compare register 2

偏移地址：0x38

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR2[15:0]**：捕获/比较 2 值 (Capture/Compare 2 value)

如果通道 CC2 配置为输出：

CCR2 是捕获/比较寄存器 2 的预装载值。

如果没有通过 TIMx_CCMR2 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 2）。

活动捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。

如果通道 CC2 配置为输入：

CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。

40.4.15 TIM12 定时器输入选择寄存器 (TIM12_TISEL)

TIM12 timer input selection register

偏移地址：0x68

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：选择 TI2[0] 到 TI2[15] 输入 (selects TI2[0] to TI2[15] input)

0000：TIM12_CH2 输入

其他：保留

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)

0000：TIM12_CH1 输入

0001：hsi_cal_ck

0010：csi_cal_ck

其他：保留

40.4.16 TIM12 寄存器映射

TIM12 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 360. TIM12 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIMx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMA	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0				0	0	0	0
0x04	**TIM12_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			Res.	Res.	Res.	Res.
	Reset value																									0	0	0	0				
0x08	**TIMx_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS [4:3] Res.		Res.	Res.	Res.	SMS[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value											0	0				0									0	0	0	0		0	0	0
0x0C	**TIMx_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIE	Res.	Res.	Res.	CC2IE	CC1IE	UIE
	Reset value																										0				0	0	0
0x10	**TIMx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	Res.	TIF	Res.	Res.	Res.	CC2IF	CC1IF	UIF
	Reset value																						0	0			0				0	0	0
0x14	**TIMx_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	Res.	Res.	CC2G	CC1G	UG
	Reset value																										0				0	0	0
0x18	**TIMx_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value								0								0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
	TIMx_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2 PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	Reserved	Res.																															
0x20	**TIMx_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
	Reset value																									0		0	0	0		0	0
0x24	**TIMx_CNT**	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIMx_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 360. TIM12 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x2C	**TIMx_ARR**	Reserved																ARR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	Reserved	Reserved																															
0x34	**TIMx_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIMx_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C to 0x64	Reserved	Res.																															
0x68	**TIM12_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value																					0	0	0	0					0	0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

40.5 TIM13/TIM14 寄存器

外设寄存器的写访问仅支持半字（16 位）或字（32 位）。而读访问可支持字节（8 位）、半字（16 位）或字（32 位）。

40.5.1 TIMx 控制寄存器 1 (TIMx_CR1)（x = 13 到 14）

TIMx control register 1

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
				rw		rw	rw	rw				rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD[1:0]**：时钟分频 (Clock division)
该位字段指示定时器时钟 (CK_INT) 频率与数字滤波器所使用的采样时钟 (TIx) 之间的分频比。
00：$t_{DTS} = t_{CK_INT}$
01：$t_{DTS} = 2 \times t_{CK_INT}$
10：$t_{DTS} = 4 \times t_{CK_INT}$
11：保留

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)
0：TIMx_ARR 寄存器不进行缓冲
1：TIMx_ARR 寄存器进行缓冲

位 6:4 保留，必须保持复位值。

位 3 **OPM**：单脉冲模式 (One-pulse mode)
0：计数器在发生更新事件时不会停止计数
1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）。

位 2 **URS**：更新请求源 (Update request source)

该位由软件置 1 和清零，用以选择更新中断 (UEV) 源。

0：使能后，所有以下事件都会生成 UEV：

– 计数器上溢

– 将 UG 位置 1

1：使能后，只有计数器上溢会生成 UEV。

位 1 **UDIS**：更新禁止 (Update disable)

该位由软件置 1 和清零，用以使能/禁止更新中断 (UEV) 事件生成。

0：使能 UEV。UEV 可通过以下事件之一生成：

– 计数器上溢

– 将 UG 位置 1

带缓冲的寄存器被加载为预加载数值。

1：禁止 UEV。不会生成 UEV，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。如果 UG 位置 1，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

*注意：只有事先通过软件将 CEN 位置 1，才可以使用外部时钟和门控*模式。而触发模式可通过硬件自动将 CEN 位置 1。

40.5.2 TIMx 中断使能寄存器 (TIMx_DIER)（x = 13 到 14）

TIMx Interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IE	UIE
														rw	rw

位 15:2 保留，必须保持复位值。

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)

0：禁止 CC1 中断

1：使能 CC1 中断

位 0 **UIE**：更新中断使能 (Update interrupt enable)

0：禁止更新中断

1：使能更新中断

40.5.3 TIMx 状态寄存器 (TIMx_SR)（x = 13 到 14）

TIMx status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IF	UIF
						rc_w0								rc_w0	rc_w0

位 15:10 保留，必须保持复位值。

位 9 **CC1OF**：捕获/比较 1 过捕获标志 (Capture/Compare 1 overcapture flag)

仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。

0：未检测到过捕获。

1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8:2 保留，必须保持复位值。

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/compare 1 interrupt flag)

该标志由硬件置 1，但需要通过软件清零（输入捕获或输出比较模式）或通过读取 TIMx_CCR1 寄存器清零（仅限输入捕获模式）。

0：未发生比较匹配/输入捕获

1：发生了比较匹配/输入捕获

如果通道 CC1 配置为输出：当计数器 TIMx_CNT 的值与 TIMx_CCR1 的值匹配时，此标志置 1。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。在中心对齐模式下，可通过 3 种方式将此标志置 1，完整说明请参见 TIMx_CR1 寄存器中的 CMS 位。

如果通道 CC1 配置为输入：当 TIMx_CCR1 寄存器中捕获到计数器值时（如果通过设置 TIMx_CCER 中的 CC1P 和 CC1NP 位定义为边沿有效，则改为在 IC1 上检测到边沿时），该位置 1。

位 0 **UIF**：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

- 上溢且当 TIMx_CR1 寄存器中 UDIS =“0”时。
- 当由于 TIMx_CR1 寄存器中 URS =“0”且 UDIS =“0”而通过软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

40.5.4 TIMx 事件生成寄存器 (TIMx_EGR)（x = 13 到 14）

TIMx event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1G	UG
														w	w

位 15:2 保留，必须保持复位值。

位 1 **CC1G**：捕获/比较 1 生成 (Capture/compare 1 generation)

该位由软件置 1 以生成事件，并由硬件自动清零。

0：无操作

1：通道 1 上生成捕获/比较事件：

如果通道 CC1 配置为输出：

使能后，CC1IF 标志置 1 并发送相应的中断。

如果通道 CC1 配置为输入：

TIMx_CCR1 寄存器中将捕获到计数器当前值。使能后，CC1IF 标志置 1 并发送相应中断。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)

该位可通过软件置 1，并由硬件自动清零。

0：无操作

1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。计数器清零。

40.5.5 TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1)（x = 13 到 14）

TIMx capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:4 **IC1F[3:0]**：输入捕获 1 滤波器 (Input capture 1 filter)

该位字段可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC[1:0]**：输入捕获 1 预分频器 (Input capture 1 prescaler)

该位字段定义 CC1 输入 (IC1) 的预分频比。
只要 CC1E= “0” （TIMx_CCER 寄存器），预分频器便立即复位。
00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。
00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
10：保留
11：保留

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

40.5.6 TIMx 捕获/比较模式寄存器 1 [复用] (TIMx_CCMR1)（x = 13 到 14）

TIMx capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
									rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 15:7 保留，必须保持复位值。

位 16，6:4 **OC1M[3:0]**：输出比较 1 模式 (Output compare 1 mode)（请参见 OC1M[3] 的位 16）

这些位定义提供 OC1 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 的有效电平则取决于 CC1P 位。

0000：冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。当定时器用作软件时基时，可以使用该模式。在定时器操作期间使能冻结模式时，输出保持进入冻结状态前的状态（有效或无效）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT = TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——只要 TIMx_CNT < TIMx_CCR1，通道 1 便输出有效电平，否则输出无效电平。

0111：PWM 模式 2——只要 TIMx_CNT < TIMx_CCR1，通道 1 便输出无效电平，否则输出有效电平。

其他值：保留

注意：在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。

注意：OC1M[3] 位不是连续的，而是在位 16 中。

位 3 **OC1PE**：输出比较 1 预装载使能 (Output compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到影子寄存器中。

位 2 **OC1FE**：输出比较 1 快速使能 (Output compare 1 fast enable)

该位可降低触发事件与定时器输出跳变之间的延迟。单脉冲模式下必须使用（TIMx_CR1 寄存器中的 OPM 位置 1），以便在启动触发后快速发送输出脉冲。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OC1FE 才会起作用。

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出。

01：CC1 通道配置为输入，IC1 映射到 TI1 上。

10：保留。

11：保留。

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

40.5.7 TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 13 到 14）

TIMx capture/compare enable register

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	Res.	CC1P	CC1E
												rw		rw	rw

位 15:4 保留，必须保持复位值。

位 3 **CC1NP**：捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output polarity)。
CC1 通道配置为输出：CC1NP 必须保持清零。
CC1 通道配置为输入：CC1NP 位与 CC1P 配合使用可定义 TI1FP1 的极性（请参见 CC1P 说明）。

位 2 保留，必须保持复位值。

位 1 **CC1P**：捕获/比较 1 输出极性 (Capture/Compare 1 output Polarity)。
0：OC1 高电平有效（输出模式）/边沿有效性选择（输入模式，见下文）
1：OC1 低电平有效（输出模式）/边沿有效性选择（输入模式，见下文）
CC1 通道配置为输入时，CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。
CC1NP=0，CC1P=0：非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。
CC1NP=0，CC1P=1：反相/下降沿触发。电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。
CC1NP=1，CC1P=1：非反相/上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。
CC1NP=1，CC1P=0：该配置保留，不得使用。

位 0 **CC1E**：捕获/比较 1 输出使能 (Capture/Compare 1 output enable)。
0：禁止捕获模式/OC1 无效
1：使能捕获模式/在相应输出引脚上输出 OC1 信号

表 361. 标准 OCx 通道的输出控制位

CCxE 位	OCx 输出状态
0	禁止输出（不由定时器驱动：高阻态）
1	使能输出（tim_ocx = tim_ocxref + 极性）

注： *与标准 OCx 通道相连的外部 I/O 引脚的状态取决于通道 OCx 的状态以及 GPIO 寄存器。*

40.5.8 TIMx 计数器 (TIMx_CNT)（x = 13 到 14）

TIMx counter

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIF CPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **UIFCPY**：UIF 副本 (UIF Copy)
该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

40.5.9 TIMx 预分频器 (TIMx_PSC)（x = 13 到 14）

TIMx prescaler

偏移地址：0x28

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)
计数器时钟频率 CK_CNT 等于 f_{CK_PSC} / (PSC[15:0] + 1)。
PSC 包含在每次发生更新事件时要装载到有效预分频器寄存器的值。
（包括当计数器通过 TIMx_EGR 寄存器的 UG 位或在“复位模式”下通过触发控制器清零时）。

40.5.10 TIMx 自动重载寄存器 (TIMx_ARR)（x = 13 到 14）

TIMx auto-reload register

偏移地址：0x2C

复位值：0xFFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)
ARR 为要装载到实际自动重载寄存器的值。
有关 ARR 更新和行为的详细信息，请参见*第 2064 页的第 40.3.1 节：时基单元*。
当自动重载值为空时，计数器不工作。

40.5.11 TIMx 捕获/比较寄存器 1 (TIMx_CCR1)（x = 13 到 14）

TIMx capture/compare register 1

偏移地址：0x34

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR1[15:0]**：捕获/比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：
CCR1 是捕获/比较寄存器 1 的预装载值。
如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 1）。
实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出相应电平的值。

如果通道 CC1 配置为输入：
CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。

40.5.12 TIM13 定时器输入选择寄存器 (TIM13_TISEL)

TIM13 timer input selection register

偏移地址：0x68

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
												rw	rw	rw	rw

位 15:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)
0000：TIM13_CH1 输入
其他：保留

40.5.13 TIM14 定时器输入选择寄存器 (TIM14_TISEL)

TIM14 timer input selection register

偏移地址：0x68

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
												rw	rw	rw	rw

位 15:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)

0000：TIM14_CH1 输入

其他：保留

40.5.14 TIM13/TIM14 寄存器映射

TIMx 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 362. TIM13/TIM14 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIMx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMA	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0				0	0	0	0
0x04 to 0x08	Reserved	Res.																															
0x0C	**TIMx_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IE	UIE
	Reset value																															0	0
0x10	**TIMx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1OF		Res.	Res.	Res.	Res.	Res.	Res.	CC1IF	UIF
	Reset value																							0								0	0
0x14	**TIMx_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1G	UG
	Reset value																															0	0
0x18	**TIMx_CCMR1** Output compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value																0										0	0	0	0	0	0	0
	TIMx_CCMR1 Input capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0x1C	Reserved	Res.																															

表 362. TIM13/TIM14 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x20	**TIMx_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	Res.	CC1P	CC1E
	Reset value																													0		0	0
0x24	**TIMx_CNT**	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIMx_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIMx_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	Reserved	Res.																															
0x34	**TIMx_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38 to 0x64	Reserved	Res.																															
0x68	**TIM13_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value																													0	0	0	0
0x68	**TIM14_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

41 通用定时器 (TIM15/TIM16/TIM17)

41.1 TIM15/TIM16/TIM17 简介

TIM15/TIM16/TIM17 定时器包含一个 16 位自动重装载计数器，该计数器由可编程预分频器驱动。

此类定时器用途广泛，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（输出比较、PWM 和带死区插入的互补 PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

TIM15/TIM16/TIM17 定时器彼此完全独立，不共享任何资源。如*第 41.4.23 节：定时器同步 (TIM15)* 中所述，TIM15 可以进行同步。

41.2 TIM15 主要特性

TIM15 包括下列特性：

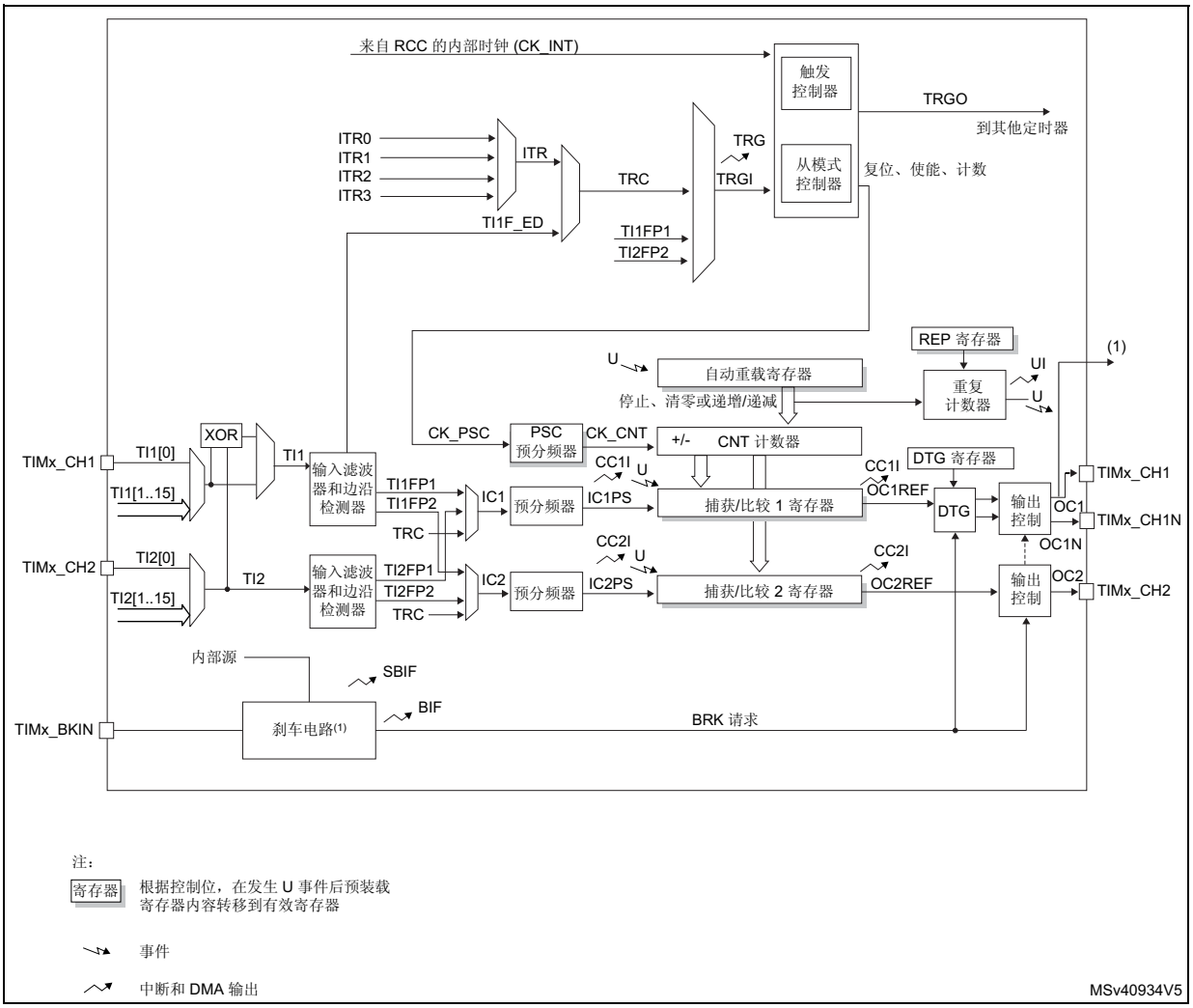
- 16 位自动重载递增计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65535 之间
- 多达 2 个独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿模式）
  - 单脉冲模式输出
- 带可编程死区的互补输出（仅适用于通道 1）
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路
- 重复计数器，用于仅在给定数目的计数周期后更新定时器寄存器。
- 用于将定时器的输出信号置于复位状态或已知状态的刹车输入。
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢、计数器初始化（通过软件或内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - 输入捕获
  - 输出比较
  - 刹车输入（中断请求）

41.3 TIM16/TIM17 主要特性

TIM16/TIM17 定时器包括下列特性：

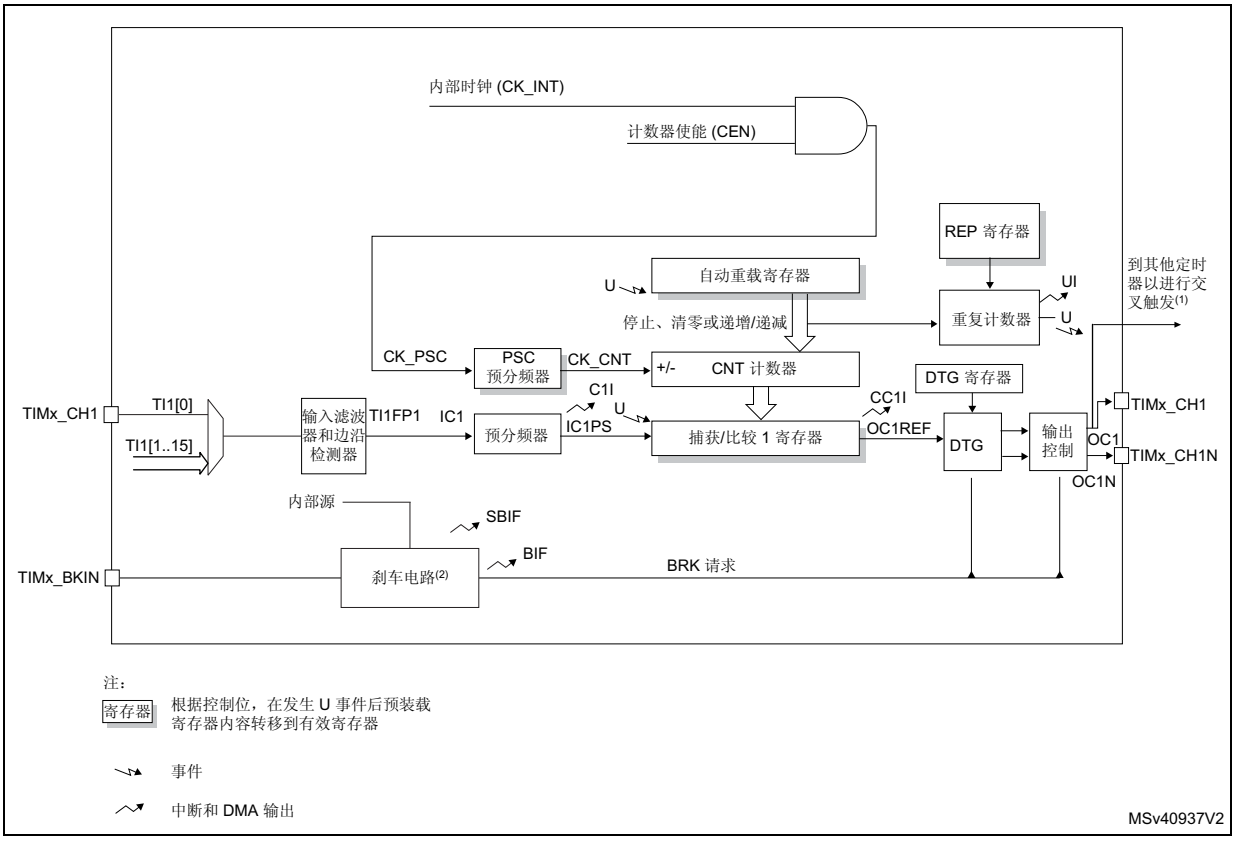
- 16 位自动重装载递增计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65535 之间
- 一个具有以下用途的通道：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿对齐模式）
  - 单脉冲模式输出
- 带可编程死区的互补输出
- 重复计数器，用于仅在给定数目的计数周期后更新定时器寄存器。
- 用于将定时器的输出信号置于复位状态或已知状态的刹车输入。
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢
  - 输入捕获
  - 输出比较
  - 刹车输入

图 471. TIM15 框图



1. 内部刹车事件源是 PVD 输出。

图 472. TIM16/TIM17 框图



1. 该信号可用于触发一些从定时器，请参见*第 41.4.24 节：使用定时器输出作为其他定时器的触发 (TIM16/TIM17)*。
2. 内部刹车事件源是 PVD 输出。

41.4 TIM15/TIM16/TIM17 功能描述

41.4.1 时基单元

可编程高级控制定时器的主要模块是一个 16 位递增计数器及其相关的自动重载寄存器。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)
- 重复计数器寄存器 (TIMx_RCR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以随时直接写入影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重装载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将产生更新事件。该更新事件也可由软件产生。下文将详细介绍每个配置中更新事件的产生方式。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

注意，计数器将在 TIMx_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。

预分频器说明

预分频器可对计数器时钟进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 TIMx_PSC 寄存器中的 16 位寄存器所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 473 和 *图 474* 以一些示例说明在预分频比实时变化时计数器的行为：

图 473. 预分频器分频由 1 变为 2 时的计数器时序图

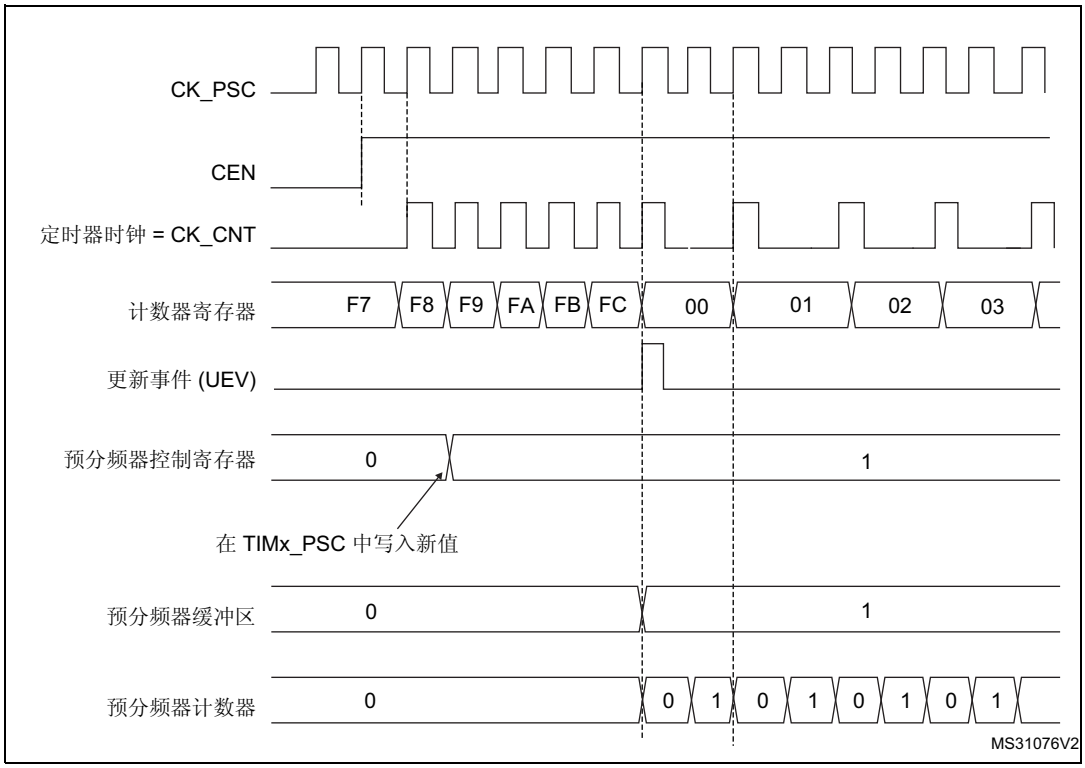
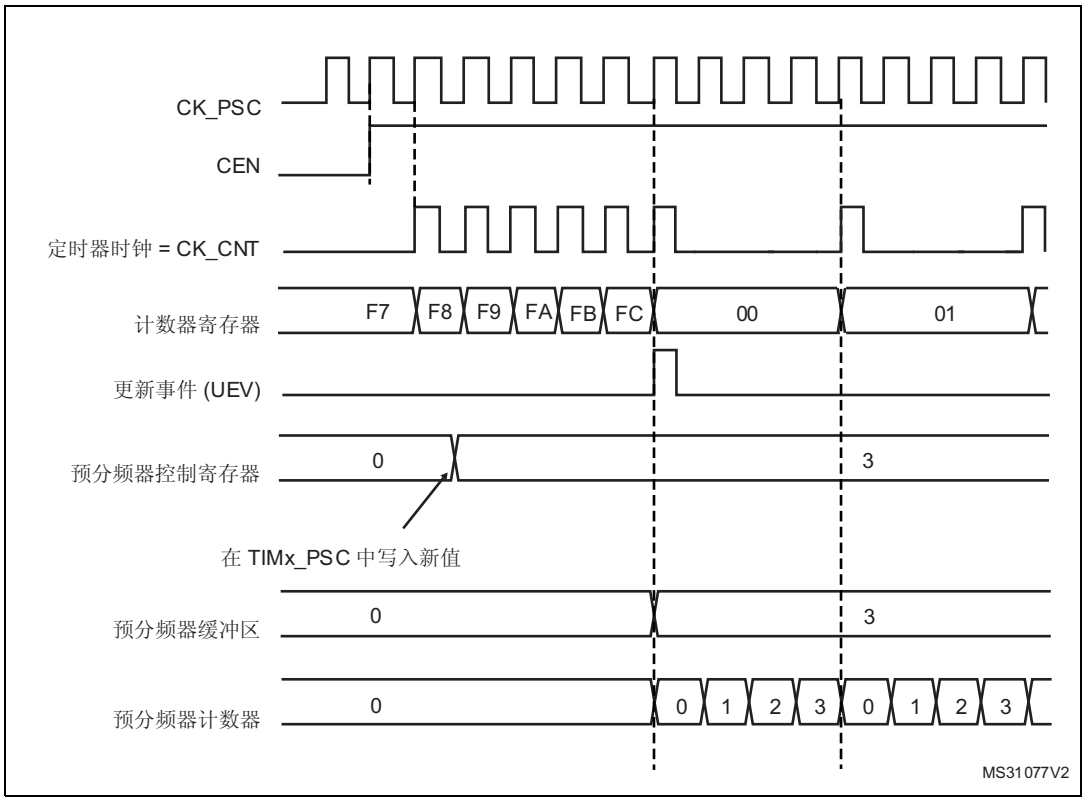


图 474. 预分频器分频由 1 变为 4 时的计数器时序图



41.4.2 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中编程的次数 (TIMx_RCR) 后，将生成更新事件 (UEV)。否则，将在每次计数器上溢时产生更新事件。

将 TIMx_EGR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。利用这一特性，在定时器的输入捕获操作时可以通过对 UG 位置 1 让计数器复位而不至于触发更新中断，也就避免了在捕获中断里触发更新中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容，
- 自动重载影子寄存器将以预装载值 (TIMx_ARR) 进行更新，
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 475. 计数器时序图，1 分频内部时钟

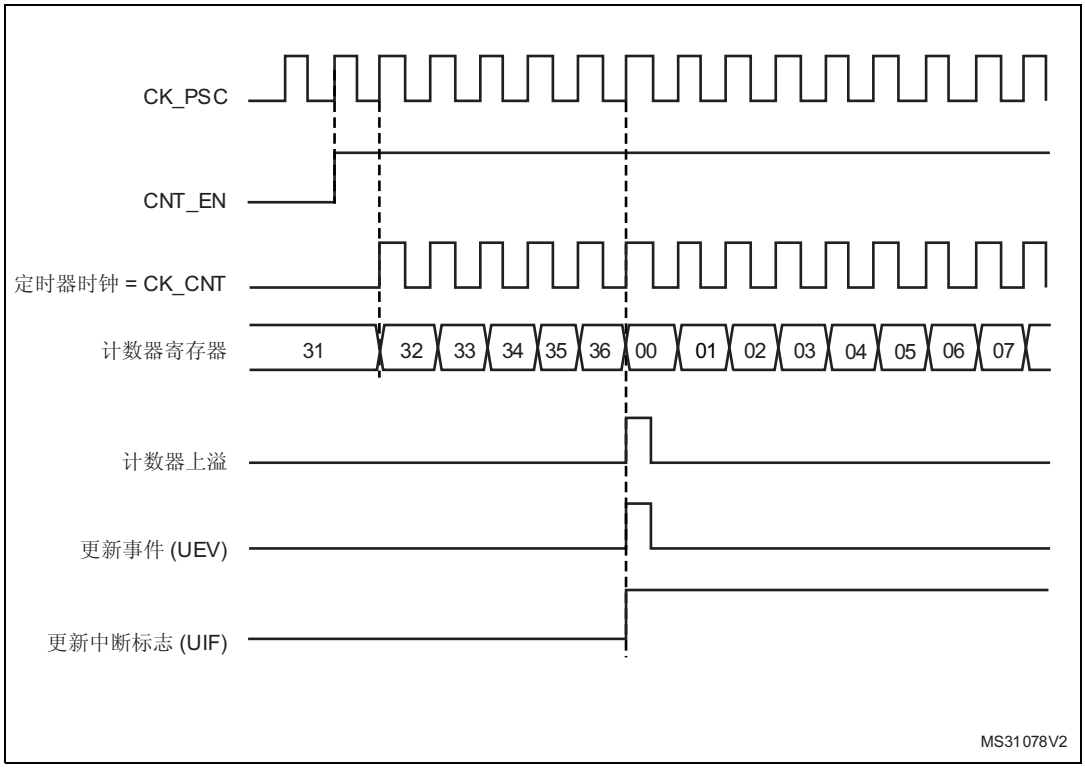


图 476. 计数器时序图，2 分频内部时钟

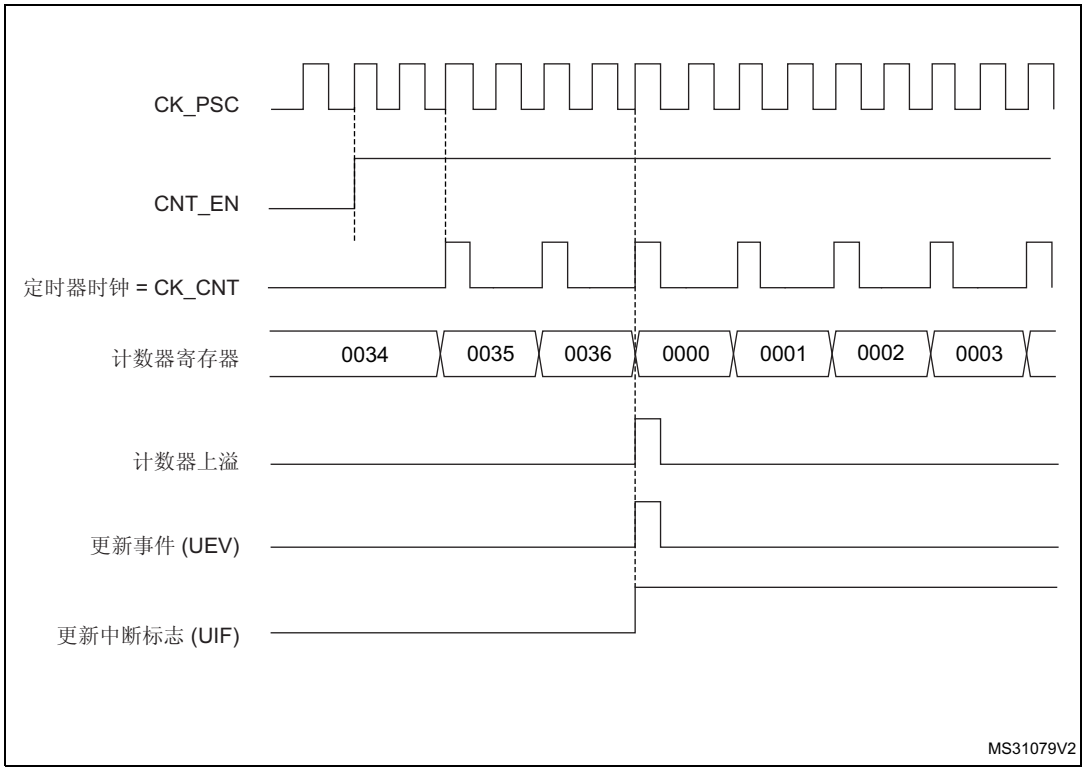


图 477. 计数器时序图，4 分频内部时钟

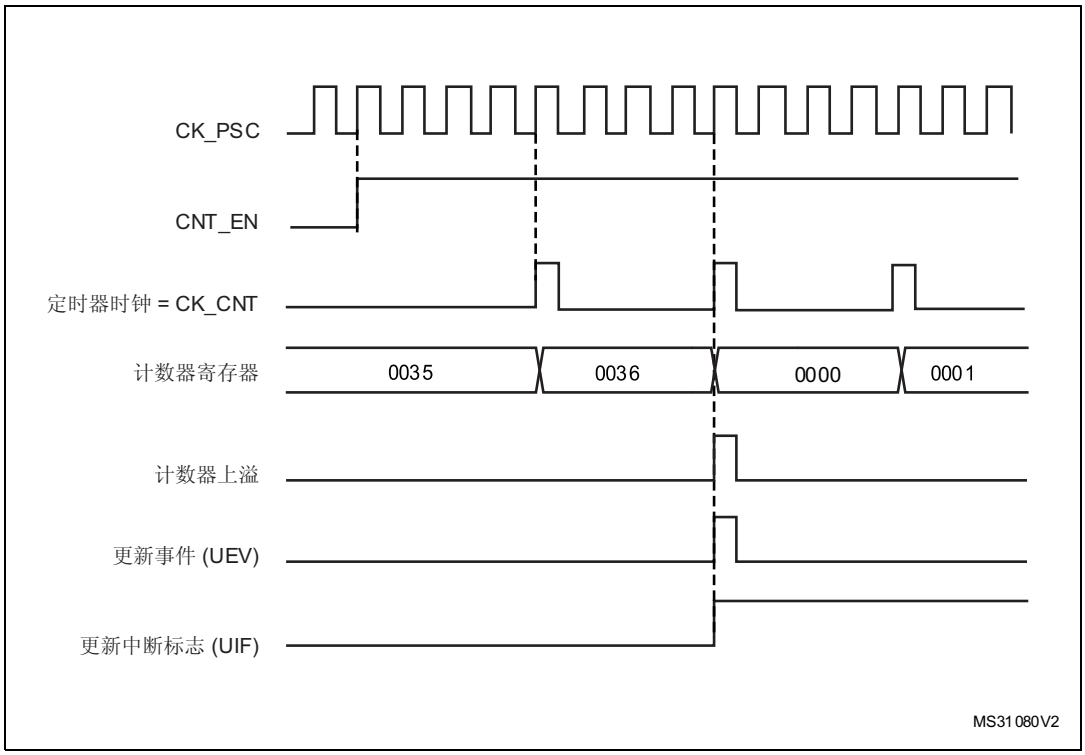


图 478. 计数器时序图，N 分频内部时钟

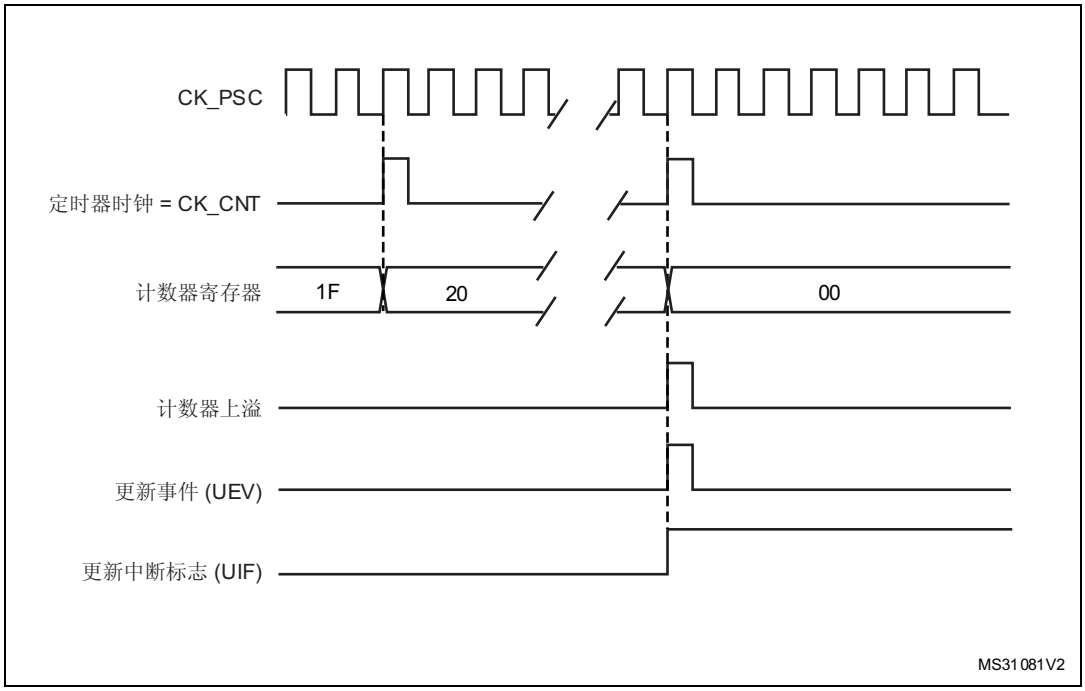


图 479. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

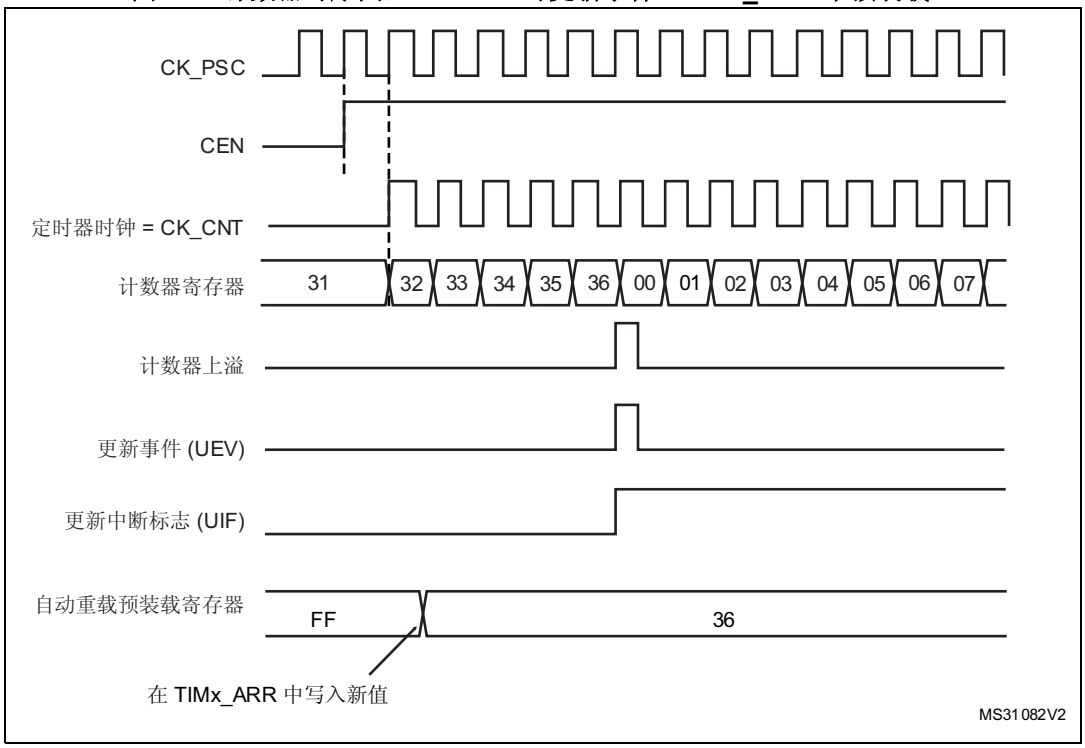
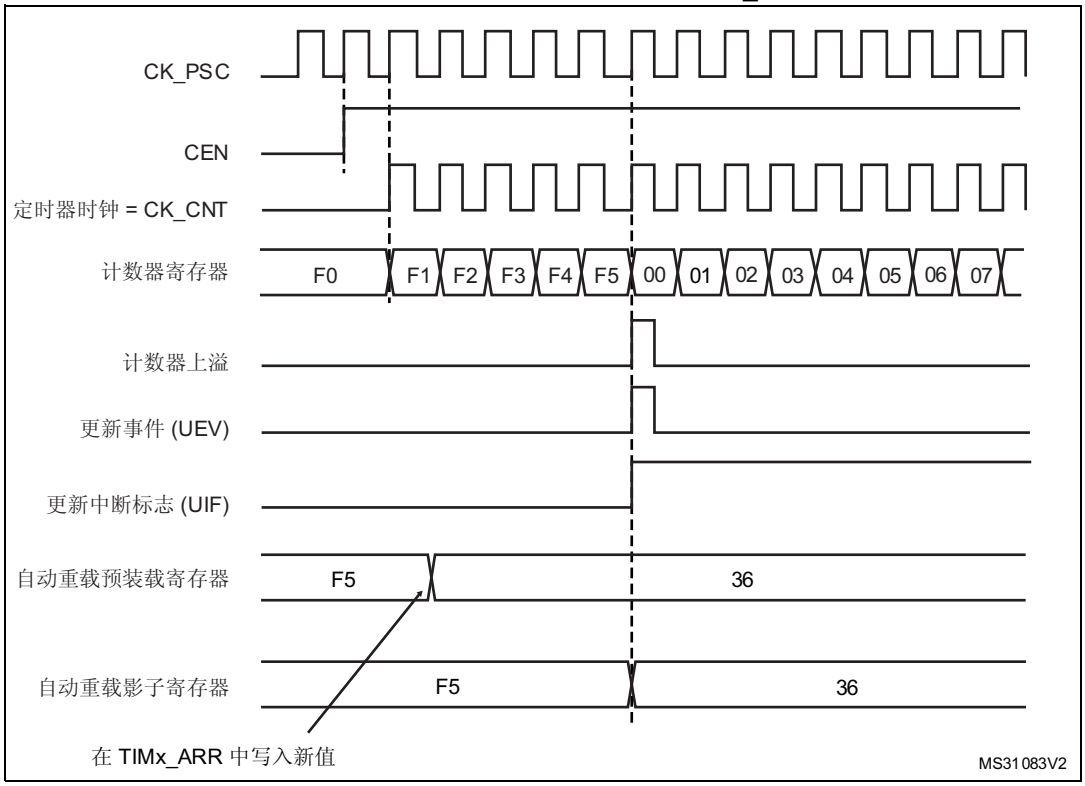


图 480. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载）



41.4.3 重复计数器

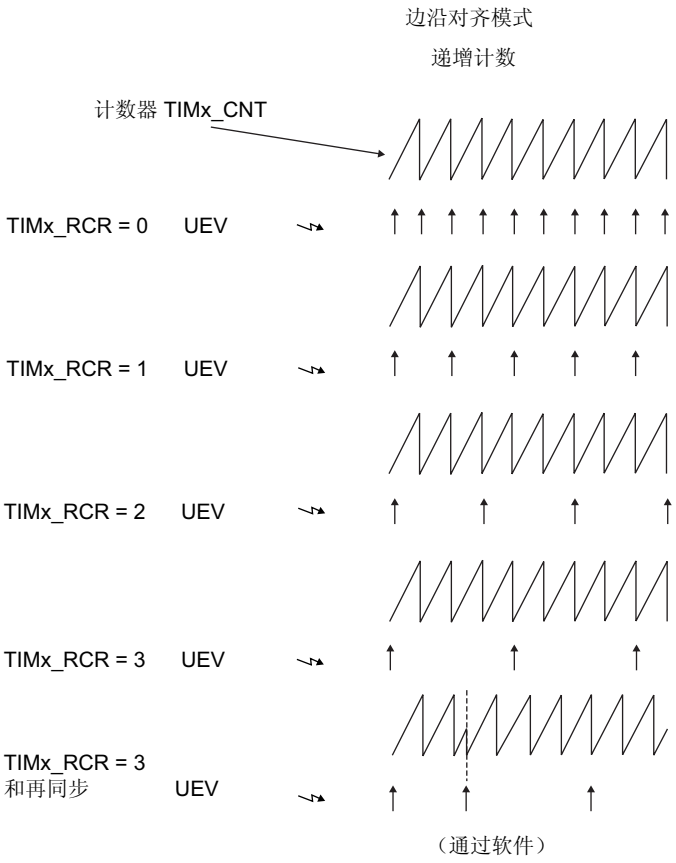
*第 41.4.1 节：时基单元*介绍如何因计数器上溢而生成更新事件 (UEV)。实际上，只有当重复计数器达到零时，才会生成更新事件。这在生成 PWM 信号时很有用。

这意味着，每当发生 N+1 次计数器上溢（其中，N 是 TIMx_RCR 重复计数器寄存器中的值），数据就将从预装载寄存器转移到影子寄存器（TIMx_ARR 自动重载寄存器、TIMx_PSC 预分频器寄存器以及比较模式下的 TIMx_CCRx 捕获/比较寄存器）。

重复计数器在每个计数器上溢时递减。

重复计数器是自动重载类型；其重复率为 TIMx_RCR 寄存器所定义的值（请参见*图 481*）。当更新事件由软件（通过将 TIMx_EGR 寄存器的 UG 位置 1）或硬件（通过从模式控制器）生成时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装载 TIMx_RCR 寄存器的内容。

图 481. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例



41.4.4 时钟选择

计数器时钟可由下列时钟源提供：

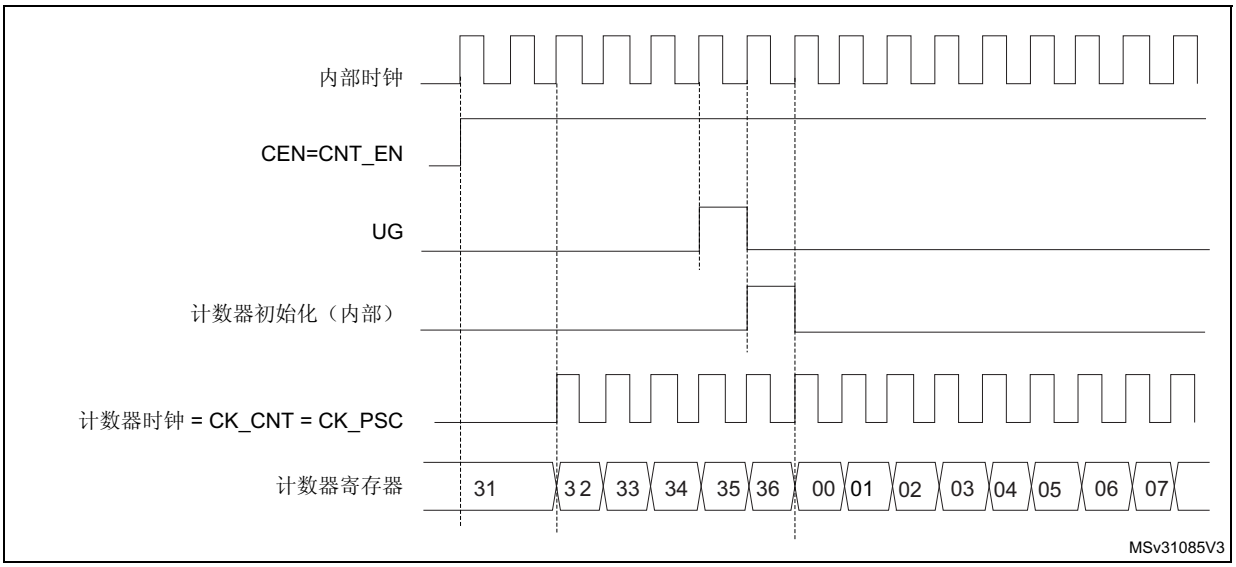
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚
- 内部触发输入 (ITRx)（仅适用于 TIM15）：使用一个定时器作为另一个定时器的预分频器，例如可以将 TIM1 配置为 TIM15 的预分频器。更多详细信息，请参见*第 2026 页的将一个定时器用作另一个定时器的预分频器*。

内部时钟源 (CK_INT)

如果禁止从模式控制器 (SMS=000)，则 CEN 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍保持自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 482 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

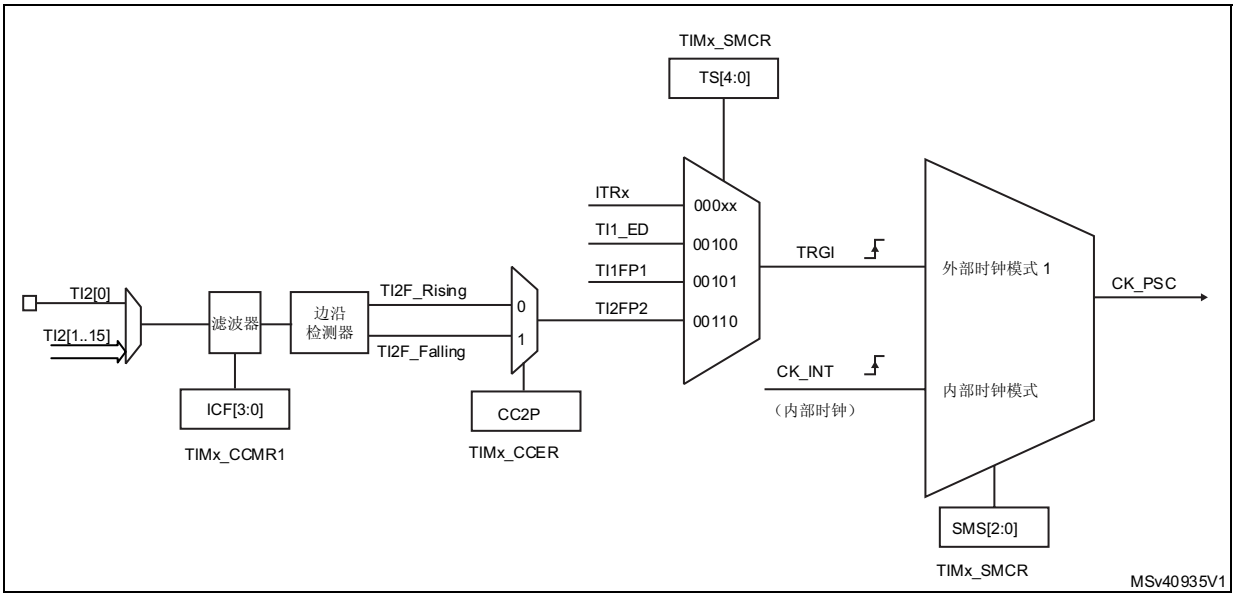
图 482. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 483. TI2 外部时钟连接示例



例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2[x] 源（内部或外部）。
2. 通过在 TIMx_CCMR1 寄存器中写入 CC2S = “01” 来配置通道 2，使其能够检测 TI2 输入的上升沿。

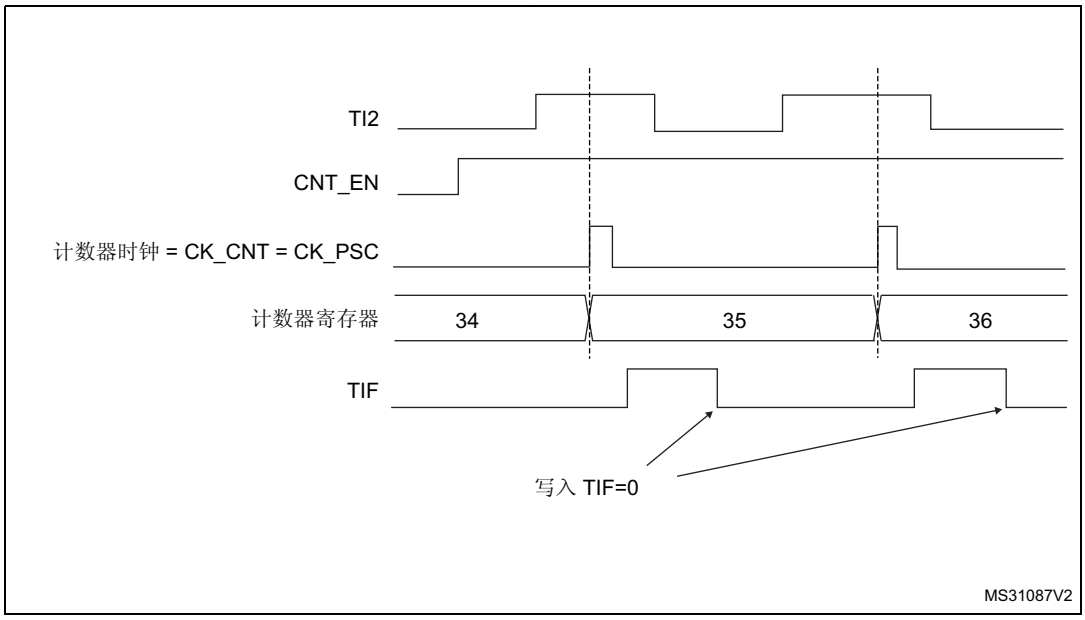
3. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。
4. 通过在 TIMx_CCER 寄存器中写入 CC2P=0 来选择上升沿极性。
5. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
6. 通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为触发输入源。
7. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

注： *由于捕获预分频器不用于触发操作，因此无需对其进行配置。*

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 484. 外部时钟模式 1 下的控制电路



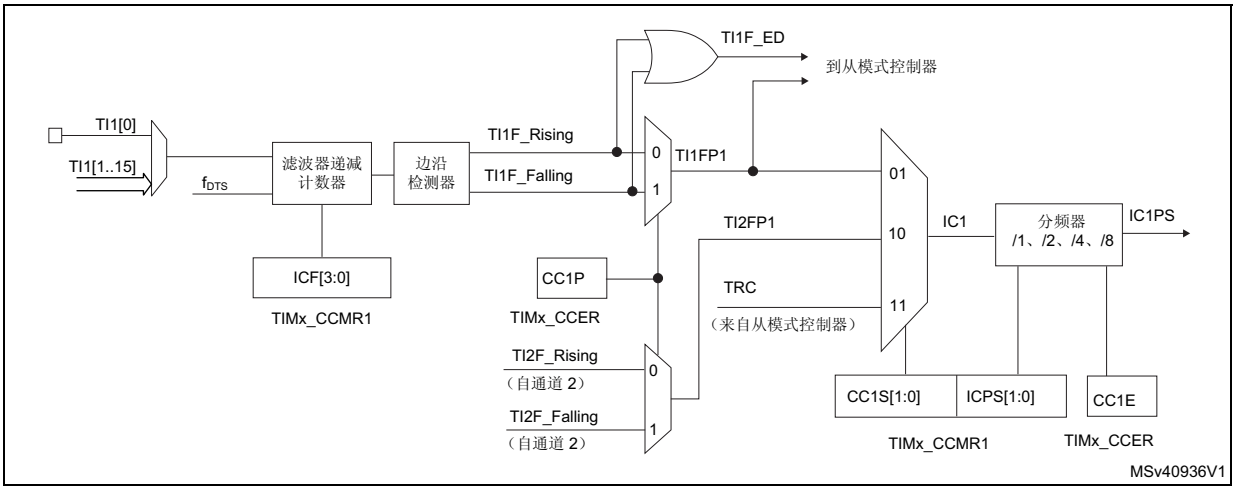
41.4.5 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入单元（数字滤波、多路复用和预分频器）和一个输出单元（比较器和输出控制）构建而成。

图 485 到*图 488* 简要介绍了一个捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 485. 捕获/比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 486. 捕获/比较通道 1 主电路

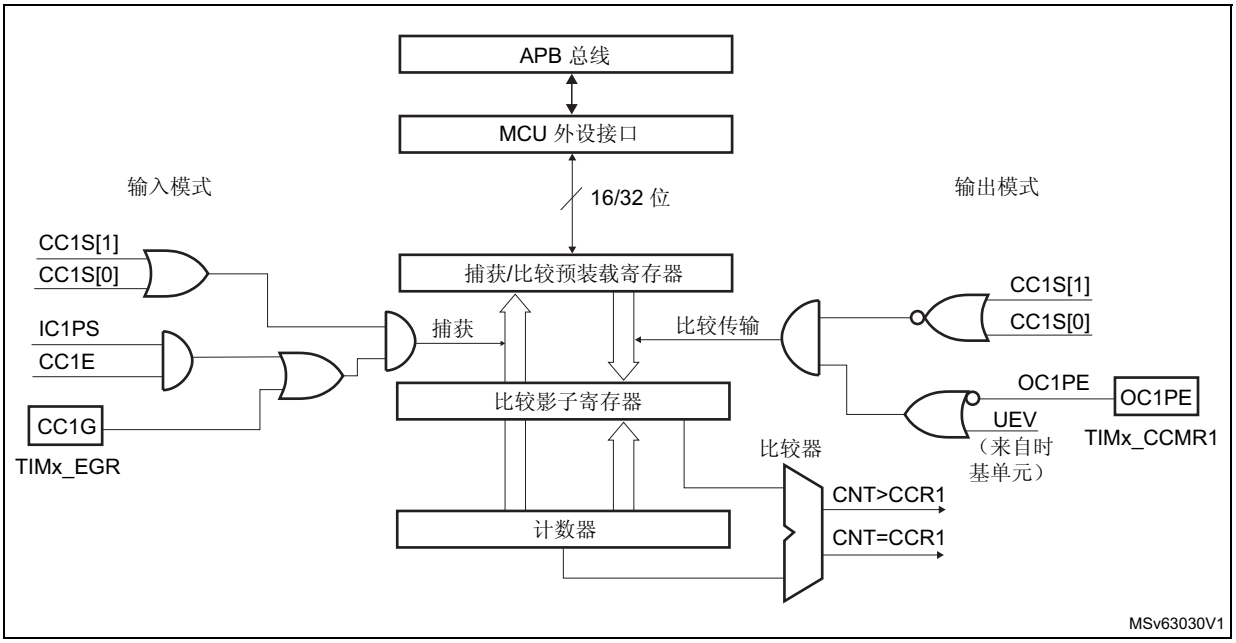


图 487. 捕获/比较通道的输出阶段（通道 1）

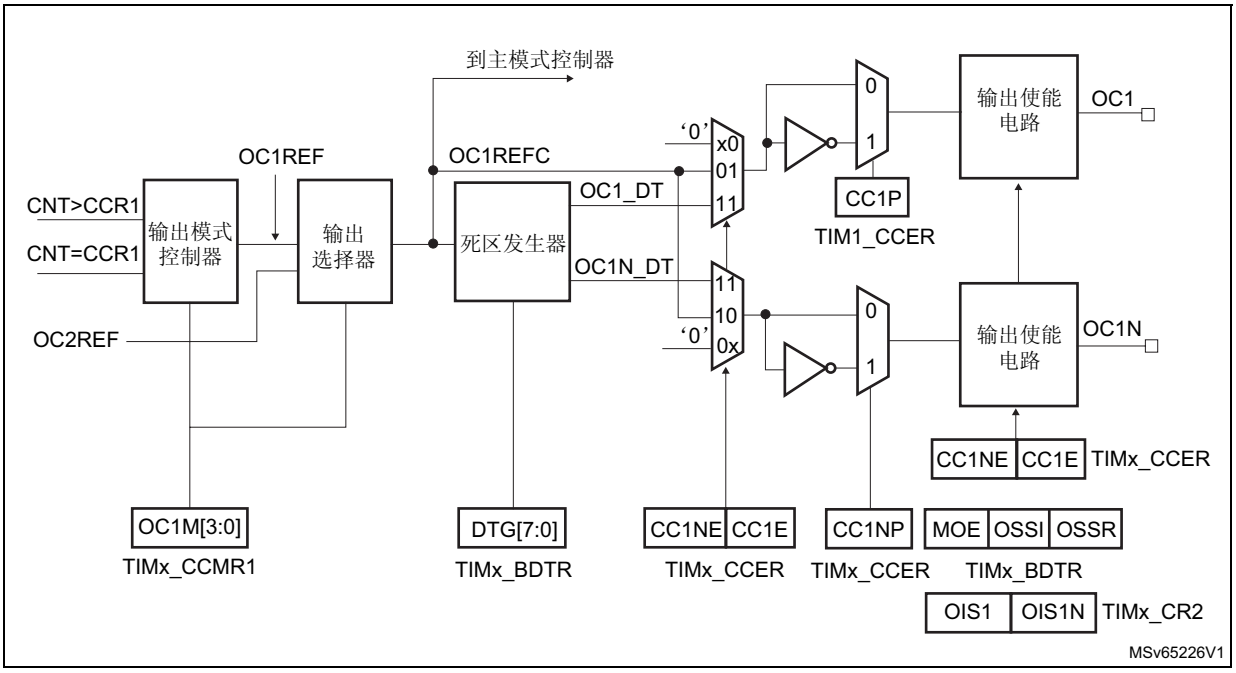
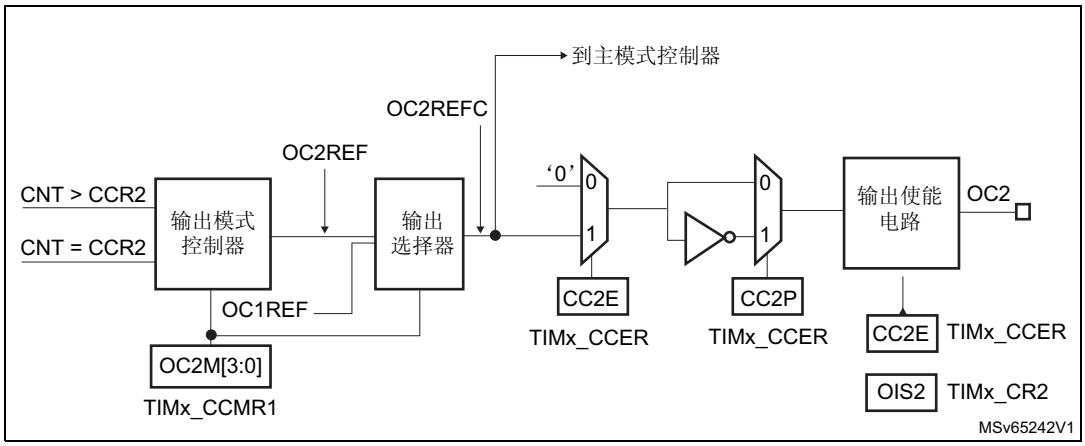


图 488. 捕获/比较通道的输出阶段（TIM15 的通道 2）



捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。读写访问的始终是预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

41.4.6 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将过捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件方法向 CCxIF 写入 0 来给 CCxIF 清零，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择有效输入：TIMx_CCR1 必须关联到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
3. 根据关联到定时器的信号，对相关输入滤波参数进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设输入信号发生翻转时，信号需要 5 个内部时钟周期才能稳定，则必须设置滤波时间大于 5 个内部时钟周期。若在 TI1 上连续 8 次检测到新电平采样值（以 f_{DTS} 频率采样）判定沿跳变合法，则向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
4. 通过向 TIMx_CCER 寄存器中的 CC1P 位写入 0，选择 TI1 上的有效转换边沿（本例中为上升沿）。
5. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。
6. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
7. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理过捕获，建议在读取捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的过捕获信息。

注： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。*

41.4.7 PWM 输入模式（仅适用于 TIM15）

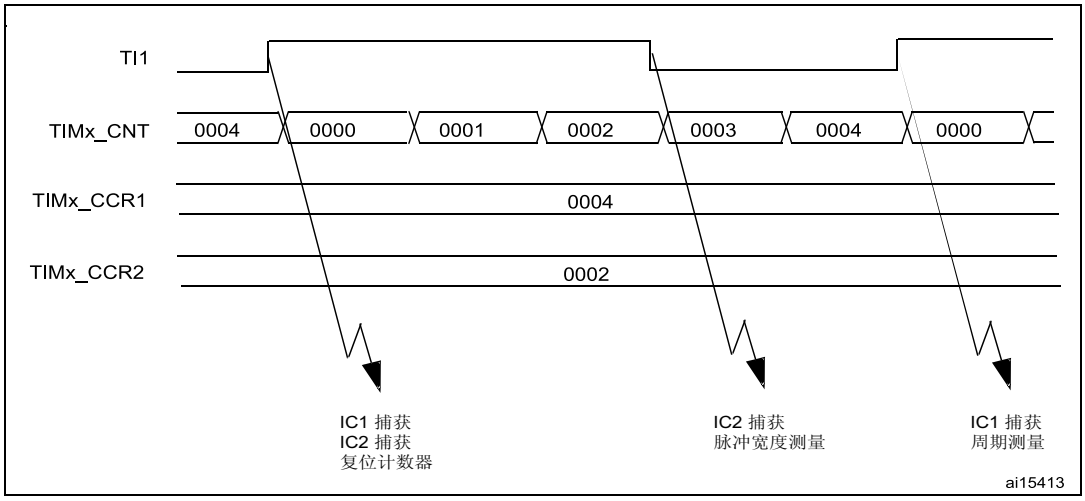
此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1[x] 源（内部或外部）。
2. 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
3. 选择 TI1FP1 的有效极性（用于在 TIMx_CCR1 中捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
4. 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10（选择 TI1）。
5. 选择 TI1FP2 的有效极性（用于在 TIMx_CCR2 中捕获）：向 CC2P 位和 CC2NP 位写入“10”（下降沿有效）。
6. 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入 00101（选择 TI1FP1）。
7. 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入 100。
8. 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 489. PWM 输入模式时序



1. PWM 输入模式只能与 TIMx_CH1/TIMx_CH2 信号配合使用，因为只有 TI1FP1 和 TI2FP2 与从模式控制器相连。

41.4.8 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（OCxREF 和 OCx/OCxN）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号 (OCXREF/OCx) 强制设置为有效电平，只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入 101。OCxREF 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如：CCxP=0（OCx 高电平有效）=> OCx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 100，可将 OCxREF 信号强制设置为低电平。

当然，即使强制输出模式下，TIMx_CCRx 影子寄存器与计数器之间的比较仍然会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。下面的输出比较模式一节对此进行了介绍。

41.4.9 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平 (OCXM=000)，也可设置为有效电平 (OCXM=001)、无效电平 (OCXM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应使能位（TIMx_DIER 寄存器的 CCxDE 位，TIMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

使用 TIMx_CCMRx 寄存器中的 OCxPE 位，可将 TIMx_CCRx 寄存器配置为带或不带预装功能。

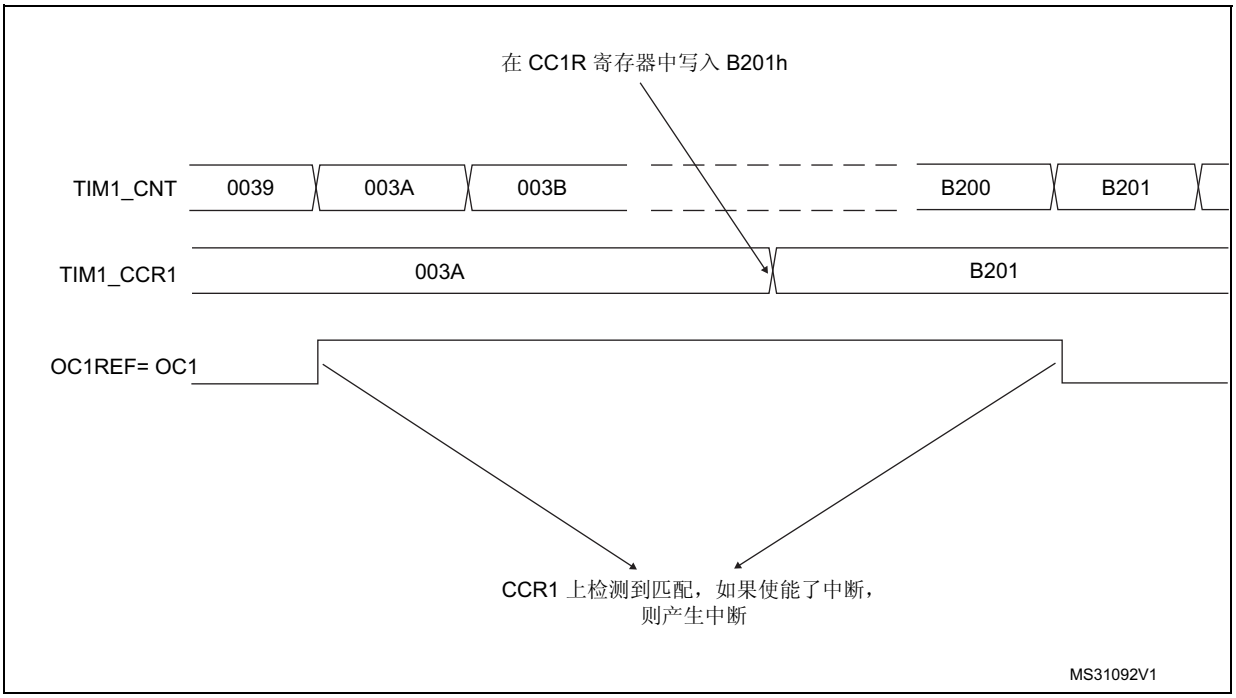
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。时间分辨率可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，则需将 CCxIE 位置 1。
4. 选择输出模式。例如：
   - 当 CNT 与 CCRx 匹配时，写入 OCxM = 011 以翻转 OCx 输出引脚
   - 写入 OCxPE = 0 以禁止预装功能
   - 写入 CCxP = 0 以选择高电平有效极性
   - 写入 CCxE = 1 以使能输出
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装功能（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIMx_CCRx 影子寄存器）。*图 490* 给出了一个示例。

图 490. 输出比较模式，翻转 OC1



41.4.10 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIMx_ARR 寄存器值决定，其占空比则由 TIMx_CCRx 寄存器值决定。

各通道可以独立选择 PWM 模式（每个 OCx 输出对应一个 PWM），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“110”（PWM 模式 1）或“111”（PWM 模式 2）。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

由于只有在发生更新事件时预装载寄存器内容才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

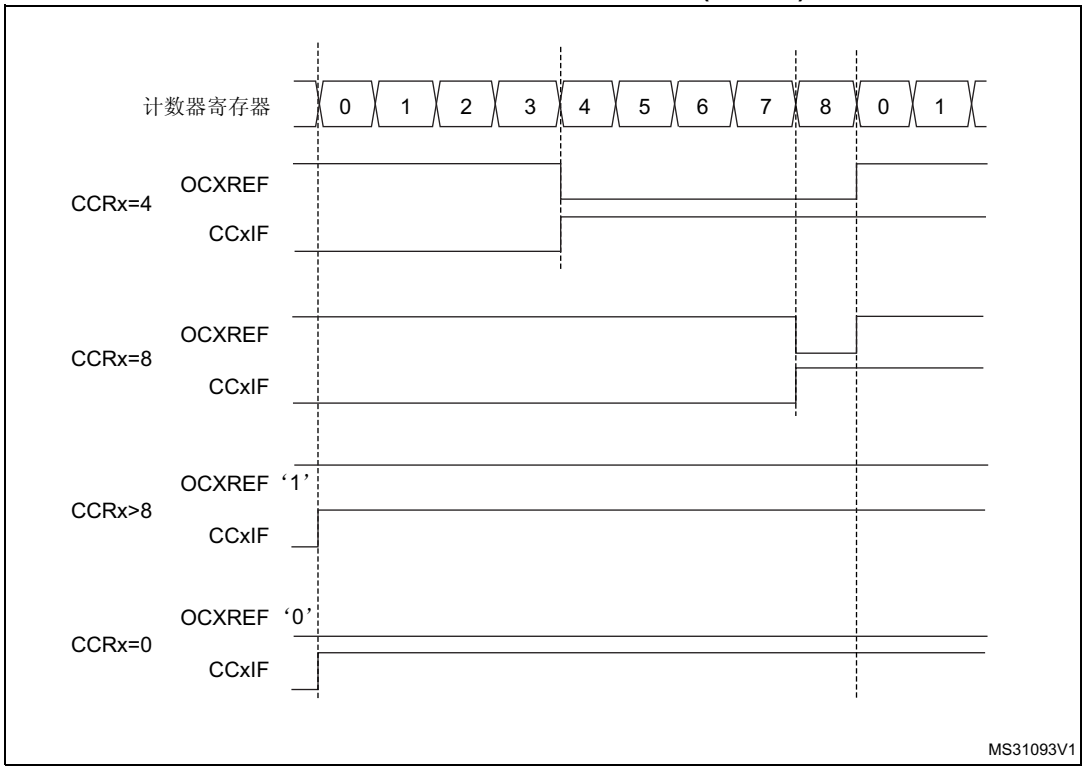
OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。通过 CCxE、CCxNE、MOE、OSSI 和 OSSR 位（TIMx_CCER 和 TIMx_BDTR 寄存器）的组合使能 OCx 输出。有关详细信息，请参见 TIMx_CCER 寄存器说明。

在 PWM 模式（1 或 2）下，TIMx_CNT 总是与 TIMx_CCRx 进行比较，以确定是 TIMx_CCRx ≤ TIMx_CNT 还是 TIMx_CNT ≤ TIMx_CCRx（取决于计数器计数方向）。

TIM15/TIM16/TIM17 只能递增计数。请参见*第 2117 页的递增计数模式*。

以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxRef 保持为“0”。*图 491* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 491. 边沿对齐模式的 PWM 波形 (ARR=8)



41.4.11 组合 PWM 模式（仅适用于 TIM15）

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的“逻辑或”运算或者“逻辑与”运算组合组成。

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 寄存器控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

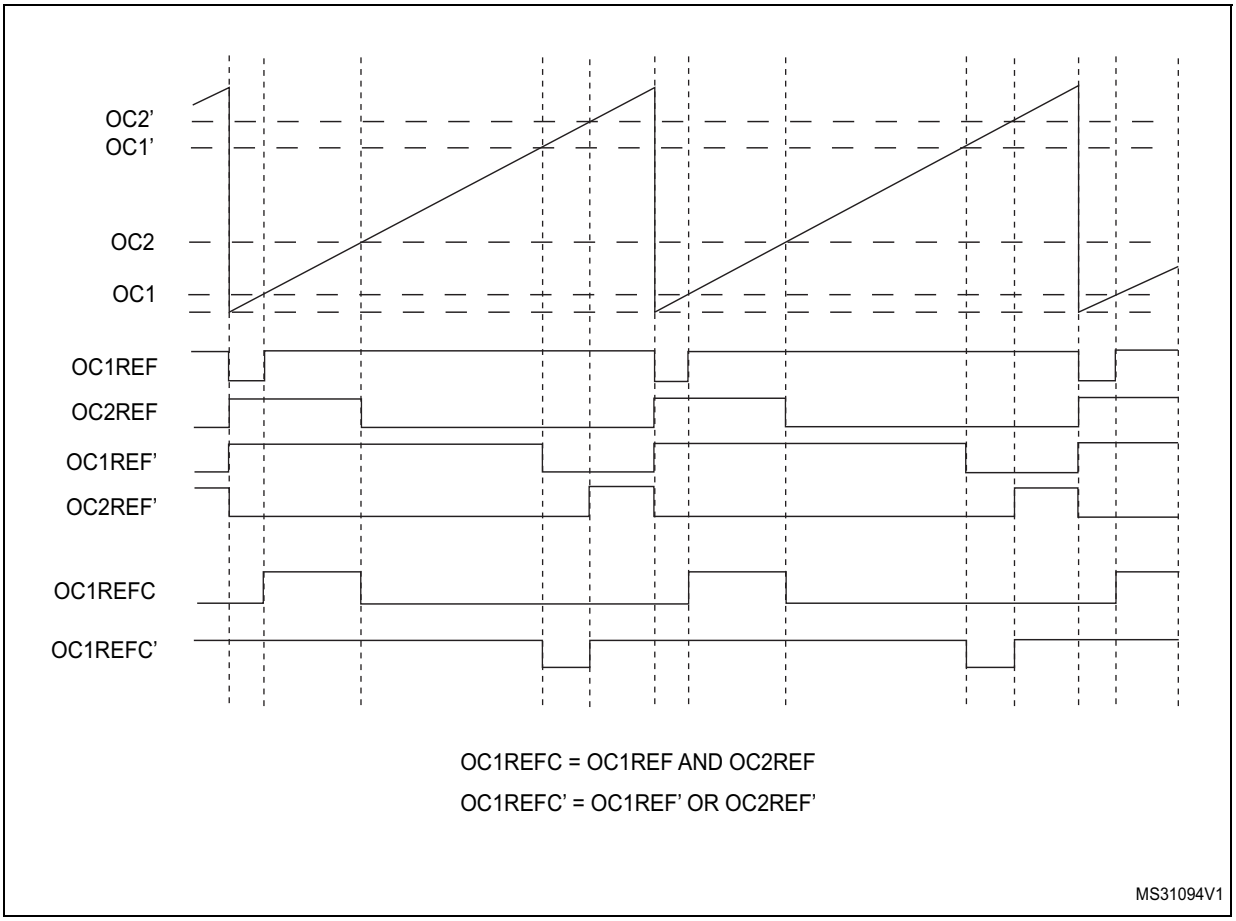
当给定通道用作组合 PWM 通道时，其互补通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注： *出于兼容性原因，OCxM[3:0] 位字段分为两部分，最高有效位与最低有效的 3 位不相邻。*

图 492 显示了不对称 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 配置为组合 PWM 模式 2，
- 通道 2 配置为 PWM 模式 1，

图 492. 通道 1 和通道 2 上的组合 PWM 模式



41.4.12 互补输出和死区插入

TIM15/TIM16/TIM17 通用定时器可以输出一路互补信号，并管理输出的关断和接通。

这段时间通常称为死区，用户必须根据与输出相连接的设备及其特性（电平转换器的固有延迟、开关设备产生的延迟...）来调整死区时间

每路输出可以独立选择输出极性（主输出 OCx 或互补输出 OCxN）。可通过对 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位执行写操作来完成极性选择。

互补信号 OCx 和 OCxN 通过以下多个控制位的组合进行激活：TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位以及 TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。更多详细信息，请参见 *第 2179 节的表 367：具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM16/17)*。应当注意，切换至空闲状态（MOE 位变为 0）的时刻，死区仍然有效。

CCxE 和 CCxNE 位同时置 1 并且 MOE 位置 1（若存在刹车电路）时，将使能死区插入。每个通道有一个 10 位死区发生器。将基于参考波形 OCxREF 生成 2 个输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高电平有效：

- 输出信号 OCx 与参考信号相同，只是其上升沿相对参考上升沿存在延迟。
- 输出信号 OCxN 与参考信号相反，并且其上升沿相对参考下降沿存在延迟。

如果延迟时间大于有效输出（OCx 或 OCxN）的宽度，则不会产生相应的脉冲。

下图所示为死区发生器的输出信号与参考信号 OCxREF 之间的关系。（在这些示例中，假定 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1）

图 493. 带死区插入的互补输出

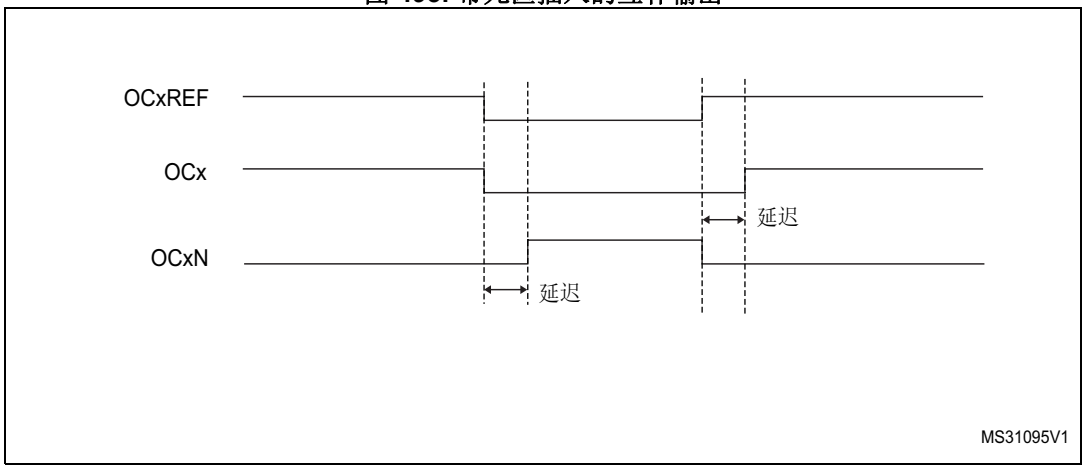


图 494. 延迟时间大于负脉冲宽度的死区波形

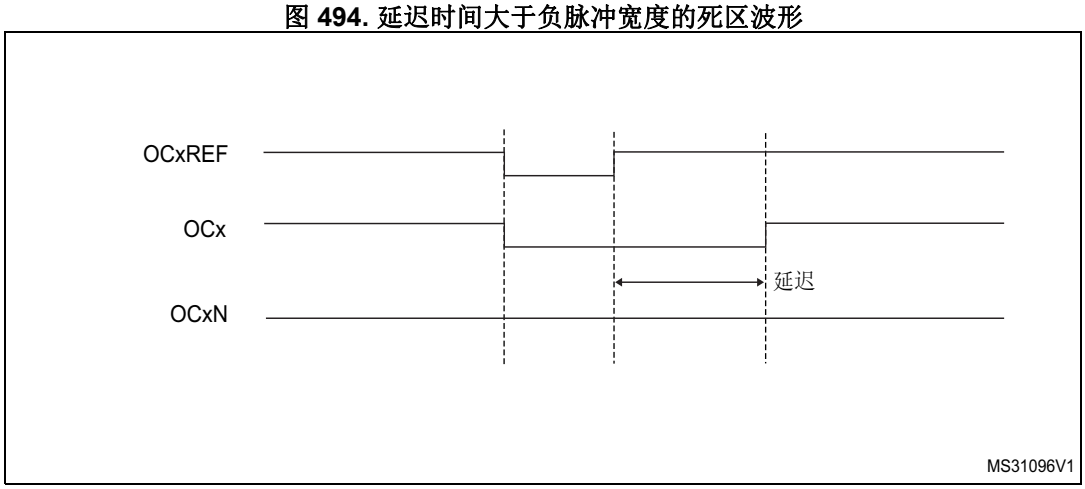
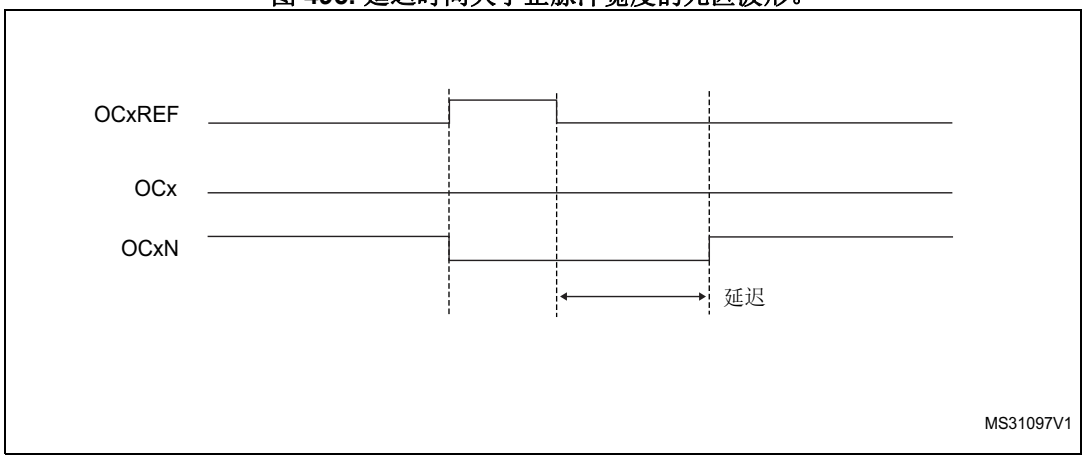


图 495. 延迟时间大于正脉冲宽度的死区波形。



死区延迟对于所有通道均相同，可通过 TIMx_BDTR 寄存器中的 DTG 位进行编程。有关延迟时间计算的信息，请参见*第 2182 页的第 41.6.14 节：TIMx 刹车和死区寄存器 (TIMx_BDTR)（x = 16 到 17）*。

将 OCxREF 重定向到 OCx 或 OCxN

在输出模式（强制输出模式、输出比较模式或 PWM 模式）下，通过配置 TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位，可将 OCxREF 重定向到 OCx 输出或 OCxN 输出。

通过此功能，可以在一个输出上发送特定波形（如 PWM 或静态有效电平），而同时使互补输出保持其无效电平。或者，使两个输出同时保持无效电平，或者两个输出同时处于有效电平，两者互补并且带死区。

注： *如果仅使能 OCxN (CCxE=0, CCxNE=1)，两者不互补，一旦 OCxREF 为高电平，OCxN 即变为有效。例如，如果 CCxNP=0，则 OCxN=OCxRef。另一方面，如果同时使能 OCx 和 OCxN (CCxE=CCxNE=1)，OCx 在 OCxREF 为高电平时变为有效，而 OCxN 则与之互补，在 OCxREF 为低电平时变为有效。*

41.4.13 使用刹车功能

刹车功能的目的是保护由 TIM15/TIM16/TIM17 定时器生成的 PWM 信号所驱动的电源开关。刹车输入通常被连接到功率级和三相逆变器的故障输出。激活时，刹车电路会关闭 PWM 输出，并将其强制为预定义的安全状态。

断路通道收集系统级故障（时钟失效和奇偶校验错误等）和应用故障（来自输入引脚和内置比较器），可以在死区持续时间后将输出强制为预定义的电平（有效或无效）。

刹车期间的输出使能信号和输出电平取决于多个控制位：

- TIMx_BDTR 寄存器中的 MOE 位，允许通过软件使能/禁止输出，在发生刹车事件和刹车 2 事件时复位。
- TIMx_BDTR 寄存器中的 OSSI 位，定义定时器将输出控制在无效状态下，还是将输出控制释放给 GPIO 控制器（通常使其处于高阻态模式）
- TIMx_CR2 寄存器中的 OISx 和 OISxN 位，将输出设置为关断电平（有效或无效）。无论 OISx 和 OISxN 的值为何，均无法在给定时间将 OCx 和 OCxN 输出同时设置为有效电平。更多详细信息，请参见*第 2160 节的表 365：具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM15)*。

退出复位状态后，刹车功能处于禁止状态，MOE 位处于低电平。通过设置 TIMx_BDTR 寄存器中的 BKE 位来使能刹车功能。刹车输入的极性可通过该寄存器中的 BKP 位来选择。BKE 和 BKP 位可同时修改。对 BKE 和 BKP 位执行写操作时，写操作会在 1 个 APB 时钟周期的延迟后生效。因此，执行写操作后，需要等待 1 个 APB 时钟周期，才能准确回读此位。

由于 MOE 下降沿可能是异步信号，因此在实际信号（作用于输出）与同步控制位（位于 TIMx_BDTR 寄存器中）之间插入了再同步电路，从而在异步信号与同步信号之间产生延迟。具体而言，如果在 MOE 处于低电平时将其置 1，则必须首先插入延迟（空指令），才能准确进行读取。这是因为写入的是异步信号，而读取的却是同步信号。

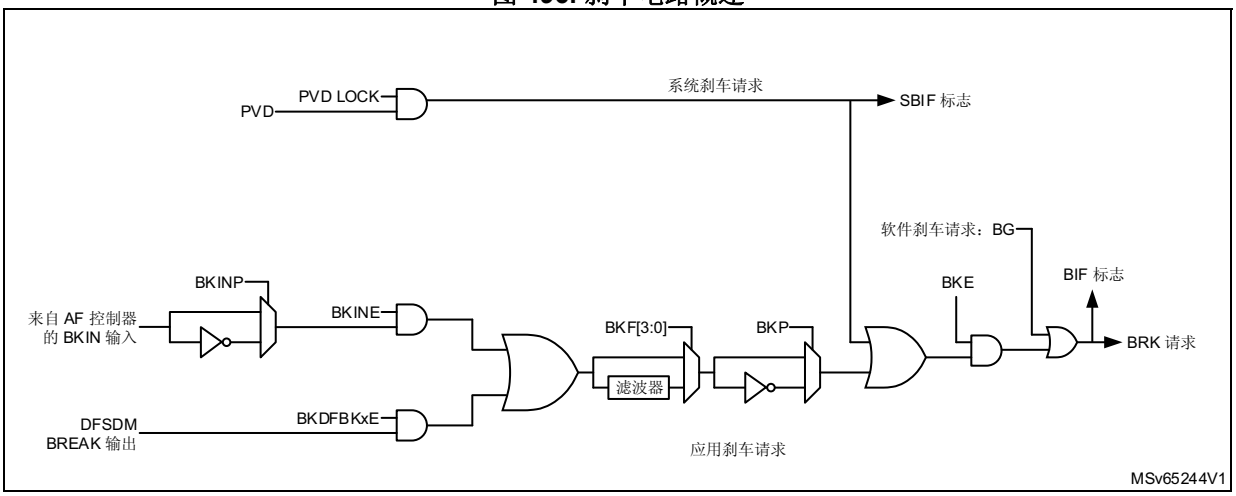
滤波器可编程（TIMx_BDTR 寄存器中的 BKF[3:0] 位），以滤除伪事件。

刹车可以从 TIMx_OR2 寄存器的多个源产生，这些源可以单独使能并使用可编程边沿有效性。

刹车 (BRK) 通道的源为：

- 连接到 BKIN 引脚的外部源（由 GPIO 复用功能寄存器选定），具有极性选择和可选的数字滤波
- 内部源：
  - DFSDM1 外设的模拟看门狗输出
  - PVD 输出

图 496. 刹车电路概述



小心： 只有禁止可编程滤波器时才能保证异步（无时钟）操作。如果使能可编程滤波器，必须使用故障安全时钟模式（例如，使用内部 PLL 和/或 CSS）来保证能够处理刹车事件。

发生刹车（刹车输入上出现所选电平）时：

- MOE 位异步清零，使输出处于无效状态、空闲状态甚至释放控制权给 GPIO 控制器（通过 OSSI 位进行选择）。即使 MCU 振荡器关闭，该功能仍然有效。
- MOE=0 时，将以 TIMx_CR2 寄存器 OISx 位中设定的电平驱动每个输出通道。如果 OSSI=0，定时器将释放输出控制（由 GPIO 控制器接管），否则使能输出保持高电平。
- 使用互补输出时：
  - 输出首先置于复位状态或无效状态（取决于极性）。这是异步操作，因此即使没有为定时器提供时钟，该操作仍有效。
  - 如果定时器时钟仍存在，则将重新激活死区发生器，进而在死区后以 OISx 和 OISxN 位中编程的电平驱动输出。即使在这种情况下，也不能同时将 OCx 和 OCxN 驱动至其有效电平。请注意，MOE 进行再同步，因此死区的持续时间会比通常情况长一些（约 2 个 ck_tim 时钟周期）。
  - 如果 OSSI=0，定时器将释放使能输出（由强制高阻态的 GPIO 接管），否则使能输出将保持高电平或在 CCxE 或 CCxNE 位之一为高电平时立即变为高电平。
- 将刹车状态标志（TIMx_SR 寄存器中的 BIF 位）置 1。如果 TIMx_DIER 寄存器中的 BIE 位置 1，可产生中断。
- 如果 TIMx_BDTR 寄存器中的 AOE 位置 1，则 MOE 位会在发生下一更新事件 (UEV) 时自动再次置 1。这一特性有许多用处，比如，可用于实现调节器的功能。否则，MOE 将始终保持低电平，直到再次向该位写入 1。这种情况下，这一特性可用于确保安全。可以将刹车输入连接到功率驱动器的警报、温度传感器或任何安全元件。

注： *当 AOE 位置 1 时，如果 CPU 将 MOE 复位，则输出处于空闲状态并根据 OSSI 值被强制设为无效电平或高阻态。*
如果 CPU 同时将 MOE 和 AOE 位复位，则输出处于禁止状态并由 TIMx_CR2 寄存器中 OISx 位设定的电平驱动。

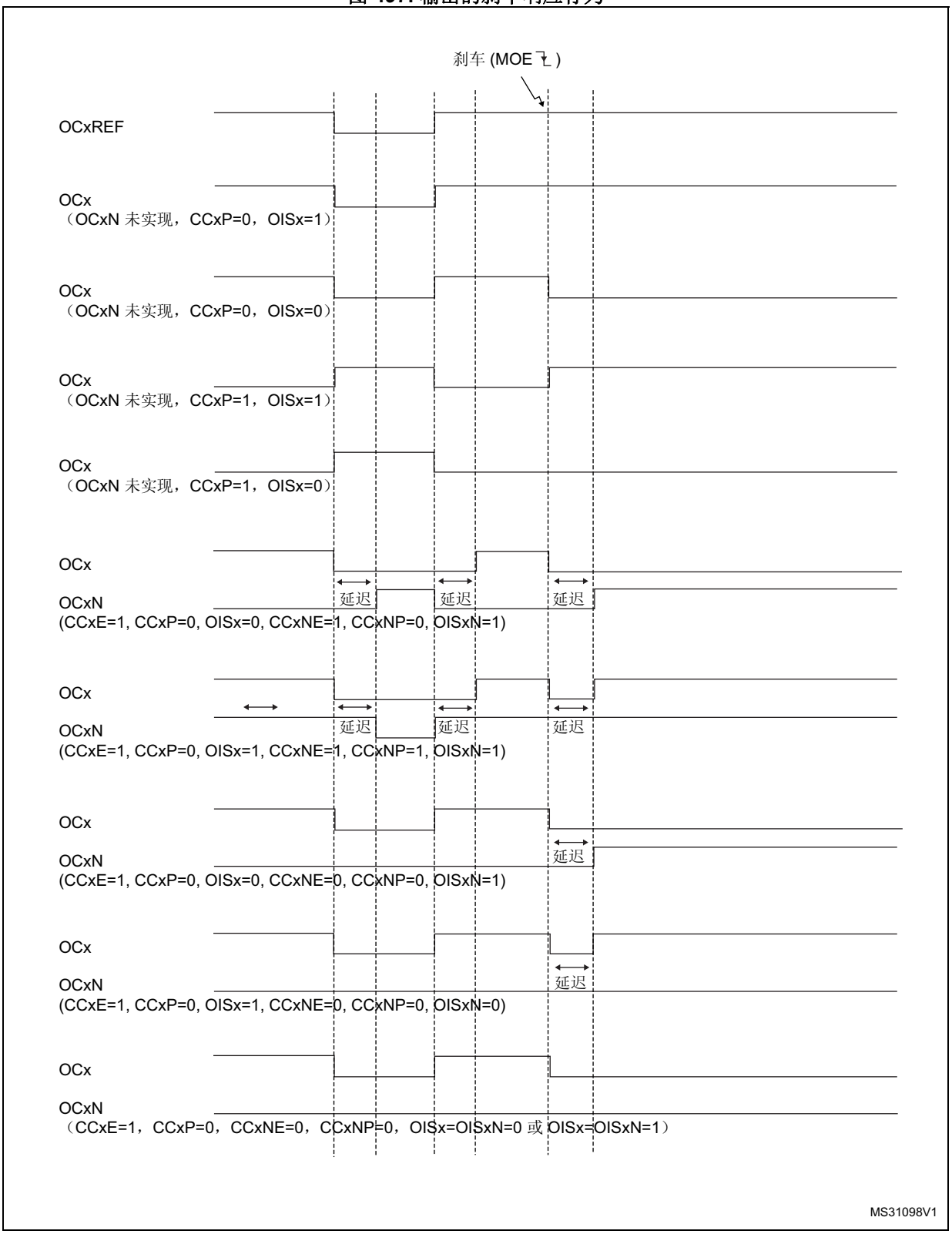
注： *刹车输入为电平有效。因此，当刹车输入为有效电平时，不能将 MOE 位置 1（自动或通过软件）。同时，不能将状态标志 BIF 清零。*

刹车可由 BRK 输入生成，该输入具有可编程极性，其使能位 BKE 位于 TIMx_BDTR 寄存器中。

除刹车输入和输出管理外，刹车电路内部还实施了写保护，用以保护应用的安全。通过该功能，用户可冻结多个参数配置（死区持续时间、OCx/OCxN 极性和禁止时的状态、OCxM 配置、刹车使能和极性）。可以通过 TIMx_BDTR 寄存器中的 LOCK 位，从 3 种保护级别中进行选择。请参见*第 2182 页的第 41.6.14 节：TIMx 刹车和死区寄存器 (TIMx_BDTR)（x = 16 到 17）*。MCU 复位后只能对 LOCK 位执行一次写操作。

图 497 所示为输出对刹车响应行为的示例。

图 497. 输出的刹车响应行为



41.4.14 双向刹车输入

TIM15/TIM16/TIM17 具有双向刹车 I/O，如*图 498* 所示。

它们可以实现以下用途：

- 将板级全局刹车信号用于向外部 MCU 或栅极驱动器发送故障信号，唯一的特定引脚作为输入输出双向状态脚。
- 在必须将多个内部和外部刹车输入合并时，将内部刹车源和多个外部开漏比较器输出“或”连接在一起，触发唯一刹车事件。

使用 TIMxBDTR 寄存器的 BKBID 位将刹车输入配置为双向模式。可以使用 TIMxBDTR 寄存器中的 LOCK 位，将 BKBID 编程位锁定在只读模式（处于锁定级别 1 或更高级别）。

双向模式需要将 I/O 配置为开漏模式且使极性低电平有效（使用 BKINP 和 BKP 位）。任何来自系统（例如 CSS）、片上外设或刹车输入的刹车请求都会强制将刹车输入置为低电平，以通知发生了故障事件。如果未正确设置极性位（高电平有效极性），则出于安全目的禁止双向模式。

软件刹车事件 (BG) 也会导致刹车 I/O 被强制为“0”，从而向外部组件指示定时器已进入刹车状态。但是仅在刹车使能时 (BKE = 1) 有效。当生成软件刹车事件且 BKE = 0 时，输出将被置于安全状态，并且刹车标志置 1，但对刹车 I/O 无影响。

安全解除机制可防止系统最终锁定（刹车输入上的低电平会触发刹车，进而将相同输入强制置为低电平）。

当 BKDSRM 位置 1 时，会释放刹车输出以清除故障信号，从而使系统能够重新启动。

在任何情况下都不能禁止刹车保护电路：

- 刹车输入路径始终有效：即使 BKDSRM 位置 1 且释放开漏控制，刹车事件也仍然有效。这样可以在发生刹车期间防止 PWM 输出重新启动。
- 使能输出（MOE 位置 1）后，BKDSRM 位不能解除刹车保护（请参见*表 363*）

表 363. 刹车保护解除条件

MOE	BKDIR	BKDSRM	刹车保护状态
0	0	X	启动
0	1	0	启动
0	1	1	解除
1	X	X	启动

启动和重新启动刹车电路

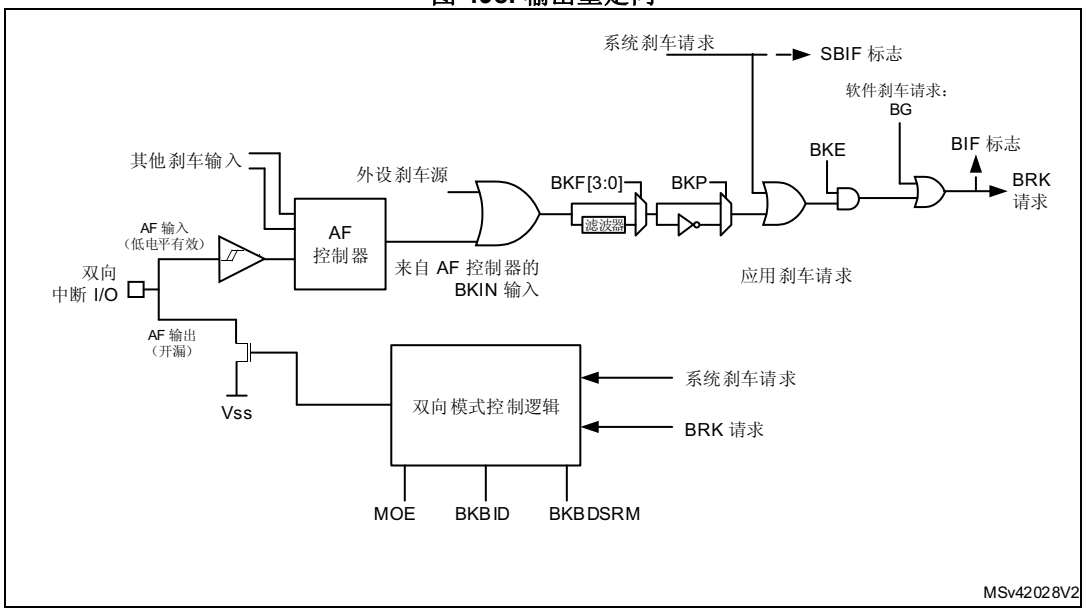
默认情况下（外设复位配置）会启动刹车电路（在输入或双向模式下）。

发生刹车事件后，必须按照以下步骤重新启动保护：

- 必须将 BKDSRM 位置 1，以释放输出控制
- 软件必须等待系统刹车条件消失（如果有），并清零 SBIF 状态标志（或在重新启动前由系统清零）
- 软件必须轮询 BKDSRM 位，直到该位由硬件清零（当应用刹车条件消失时）

此后，刹车电路即启动并激活，可以通过将 MOE 位置 1 来重新使能 PWM 输出。

图 498. 输出重定向



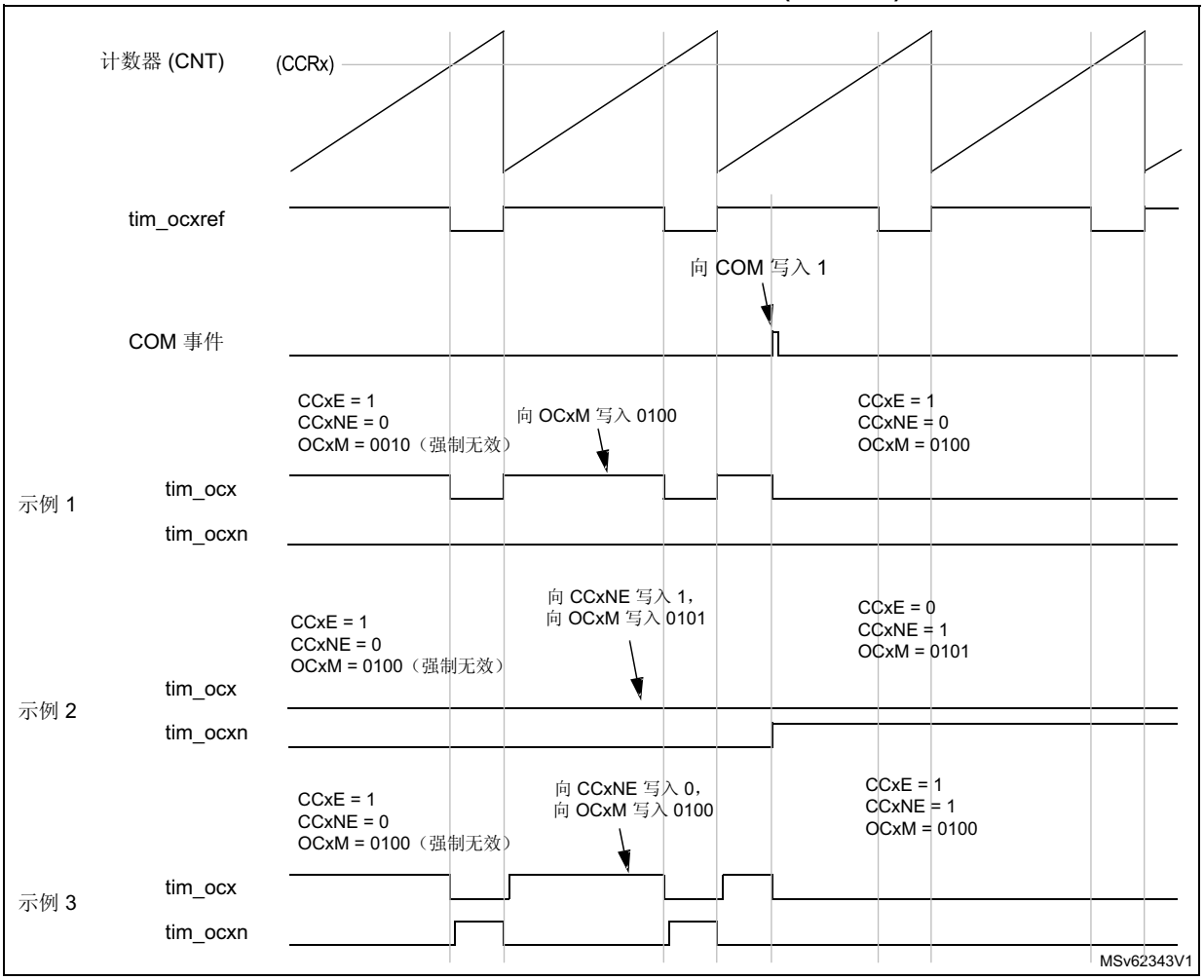
41.4.15 生成 6 步 PWM

当通道使用互补输出时，OCxM、CCxE 和 CCxNE 位上提供预装载位。发生 COM 换向事件时，这些预装载位将传输到影子位。因此，用户可以预先编程下一步骤的配置，并同时更改所有通道的配置。COM 可由软件通过将 TIMx_EGR 寄存器中的 COM 位置 1 而生成，也可以由硬件在 tim_trgi 上升沿生成。

发生 COM 事件时，某个标志位（TIMx_SR 寄存器中的 COMIF 位）将会置 1。这时，如果 TIMx_DIER 寄存器中的 COMIE 位置 1，将产生中断；如果 TIMx_DIER 寄存器中的 COMDE 位置 1，则将产生 DMA 请求。

图 499 以 3 种不同的编程配置为例，显示了发生 COM 事件时 tim_ocx 和 tim_ocxn 输出的行为。

图 499. COM 事件生成 6 步 PWM 的示例 (OSSR=1)



41.4.16 单脉冲模式

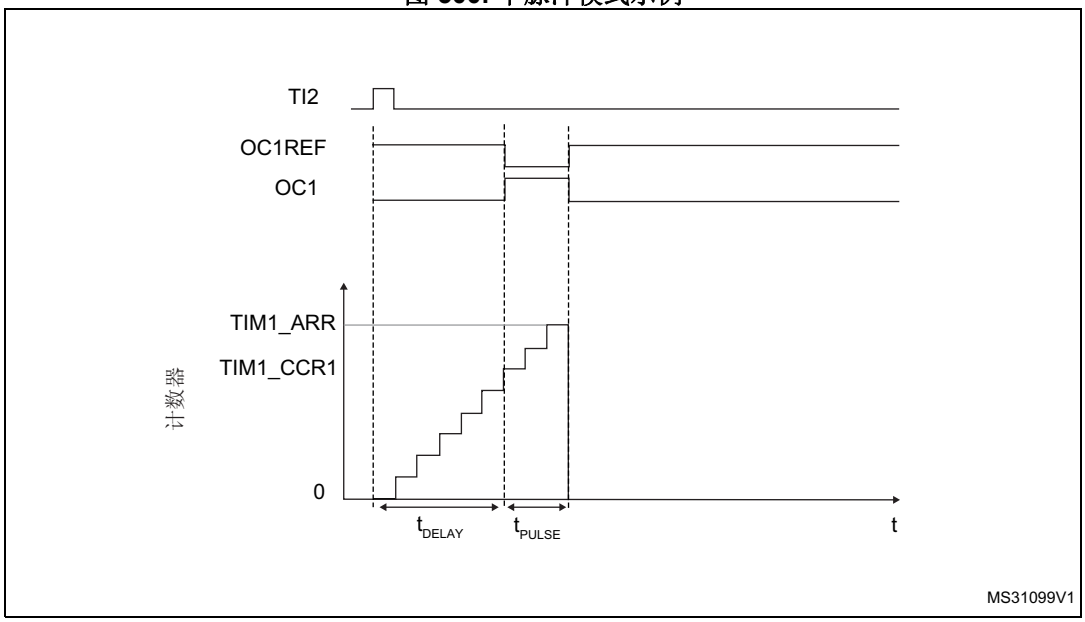
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。通过将 TIMx_CR1 寄存器中的 OPM 位置 1 选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

- CNT < CCRx ≤ ARR（特别注意，0 < CCRx）

图 500. 单脉冲模式示例



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2[x] 源（内部或外部）。
2. 在 TIMx_CCMR1 寄存器中写入 CC2S=“01”，以将 TI2FP2 映射到 TI2。
3. 在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
4. 通过在 TIMx_SMCR 寄存器中写入 TS=“00110”，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
5. 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），以使用 TI2FP2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，必须通过在 TIMx_CCMR1 寄存器中写入 OC1M=111，来使能 PWM 模式 2。可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在

TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

由于仅需要 1 个脉冲，因此必须向 TIMx_CR1 寄存器的 OPM 位写入 1，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。

特例：OCx 快速使能

在单脉冲模式下，TIx 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟（t_{DELAY} 最小值）。

如果要输出延迟时间最短的波形，可以将 TIMx_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 OCxRef（和 OCx）对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

41.4.17 可再触发单脉冲模式（仅限 TIM15）

该模式可以启动计数器来响应激励信号，并且能产生长度可编程的脉冲，但与*第 41.4.16 节*所描述的不可再触发单脉冲模式间存在以下差别：

- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长

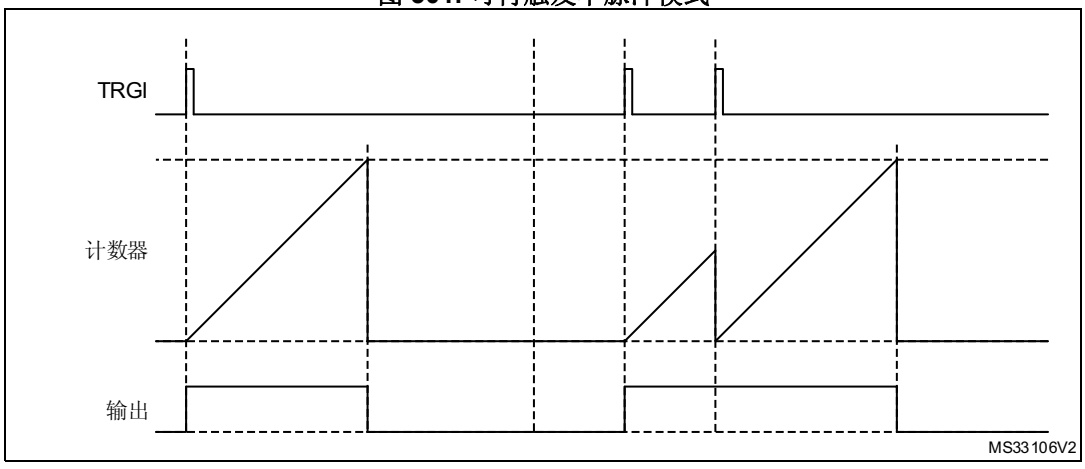
定时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0] =“1000”（组合复位 + 触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0] 位设置为“1000”或“1001”。

定时器配置为递增计数模式时，相应的 CCRx 必须置 0（ARR 寄存器设置脉冲长度）。如果定时器配置为递减计数模式，CCRx 必须高于或等于 ARR。

注： *出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位字段分为两部分，最高有效位与最低有效的 3 位不相邻。*

此模式不能与中心对齐 PWM 模式一起使用。在 TIMx_CR1 中必须设置 CMS[1:0] = 00。

图 501. 可再触发单脉冲模式



41.4.18 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可按原子方式读取计数器值以及由 UIFCPY 标志发出的更新事件情况。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

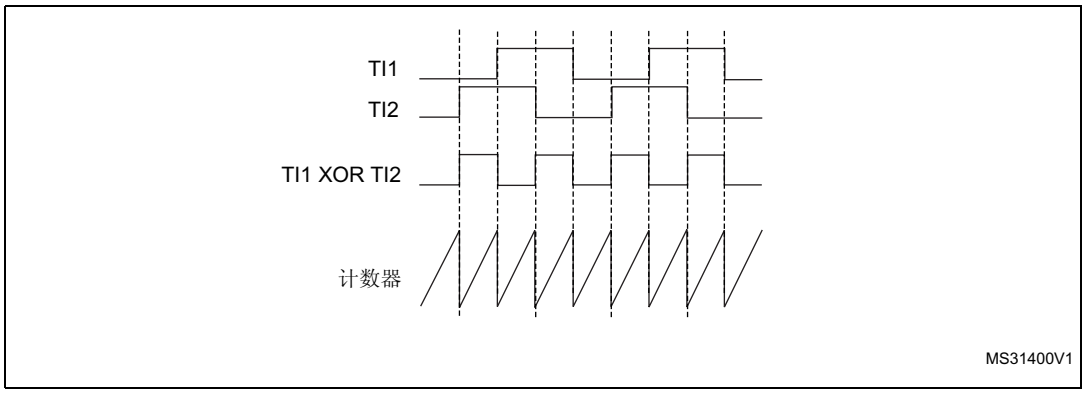
UIF 和 UIFCPY 标志之间的置位没有延迟。

41.4.19 定时器输入异或功能（仅适用于 TIM15）

通过 TIMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 和 TIMx_CH2 这两个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。这样可用于测量两个输入信号上边沿之间的间隔，如下面的*图 502* 所示。

图 502. 测量 2 个信号上边沿之间的时间间隔



41.4.20 外部触发同步（仅适用于 TIM15）

TIM 定时器从内部链接在一起，以实现定时器同步或级联。

TIM15 定时器可在多种模式下与外部触发实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

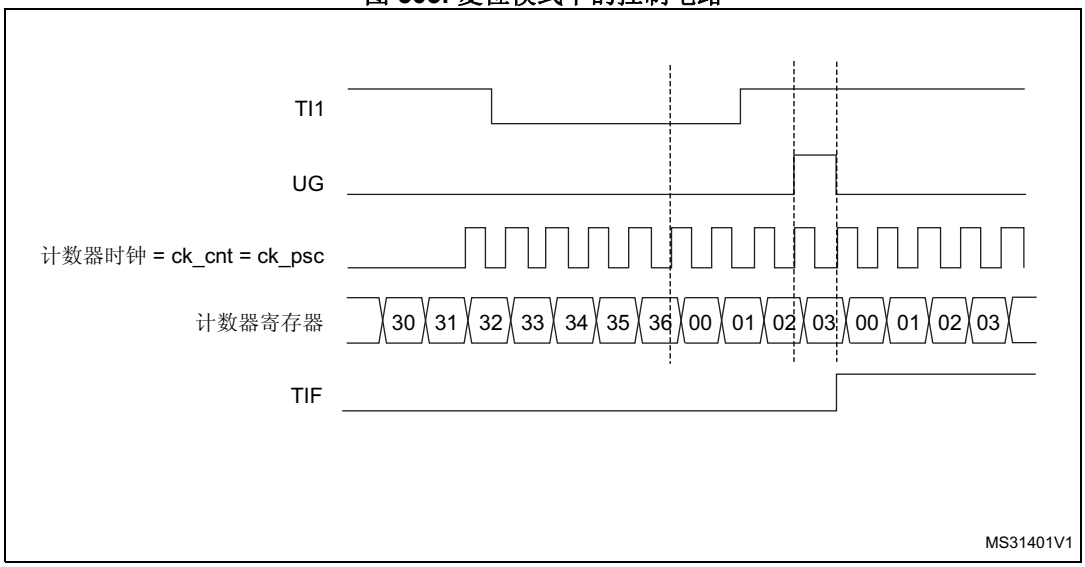
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

1. 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S = 01。在 TIMx_CCER 寄存器中写入 CC1P = “0”和 CC1NP = “0”，验证极性（仅检测上升沿）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。通过在 TIMx_SMCR 寄存器中写入 TS=00101 来选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 503. 复位模式下的控制电路



从模式：门控模式

输入信号的电平可用来使能计数器。

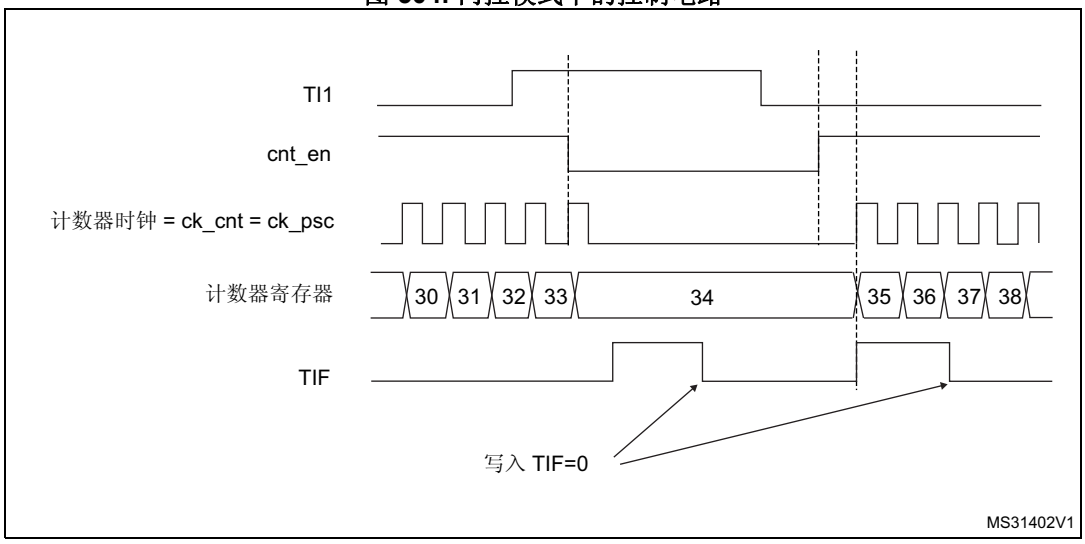
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

1. 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P = 1 和 CC1NP = “0”，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。通过在 TIMx_SMCR 寄存器中写入 TS=00101 来选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 504. 门控模式下的控制电路



从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

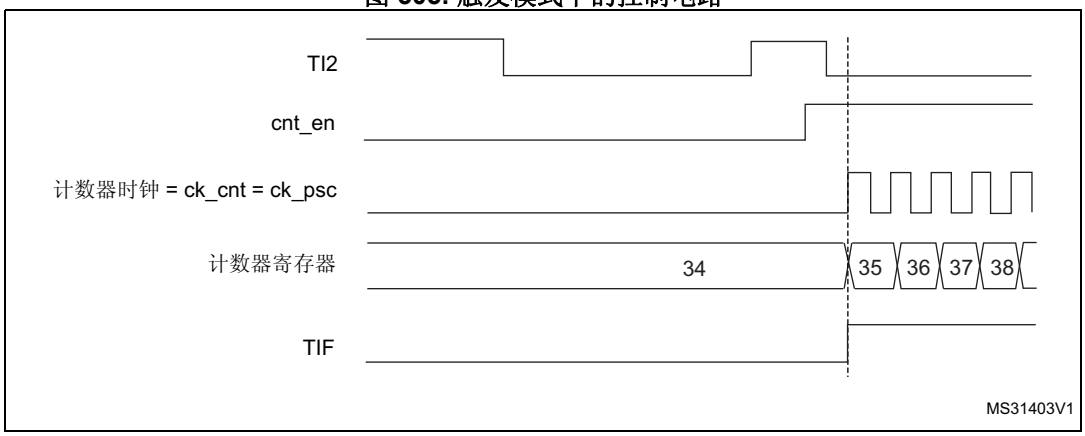
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

1. 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位配置为只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P =“1”和 CC2NP =“0”，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 505. 触发模式下的控制电路



41.4.21 从模式——组合复位 + 触发模式

在这种情况下，在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个定时器更新事件，并启动计数器。

该模式用于单脉冲模式。

41.4.22 DMA 分组传送模式

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对定时器的几个寄存器多次重新编程而无需软件开销，但也可以用于定期地一次性地批量读取定时器的多个寄存器的内容。

DMA 控制器访问目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（分组）。每次写入 TIMx_DMAR 寄存器都会重定向到定时器其中的一个寄存器。

TIMx_DCR 寄存器中的 DBL[4:0] 位设置 DMA 突发传输的长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次分组传送，即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA[4:0] 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：

00000：TIMx_CR1，

00001：TIMx_CR2，

00010：TIMx_SMCR，

例如，定时器 DMA 连续传送功能用于在发生更新事件后将 CCRx 寄存器（x = 2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

1. 将相应的 DMA 通道配置如下：
   - DMA 通道外设地址为 DMAR 寄存器地址。
   - DMA 通道内存地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
   - 要传输的数据量 = 3（参见下文注释）。
   - 禁止循环模式。
2. 通过将 DBA 和 DBL 位字段配置如下来配置 DCR 寄存器：
   DBL = 3 次传输，DBA = 0xE。
3. 使能 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 TIMx
5. 使能 DMA 通道

本例适用于每个 CCRx 寄存器只更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间，data1 传输到 CCR2，data2 传输到 CCR3，data3 传输到 CCR4；在第二个更新 DMA 请求期间，data4 传输到 CCR2，data5 传输到 CCR3，data6 传输到 CCR4。

注： *可以将空值写入保留的寄存器中。*

41.4.23 定时器同步 (TIM15)

TIMx 定时器从内部连接在一起，以实现定时器同步或链接。有关详细信息，请参见*第 39.3.19 节：定时器同步*。

注： *必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。*

41.4.24 使用定时器输出作为其他定时器的触发 (TIM16/TIM17)

单通道定时器没有主模式。但是，可以使用 OC1 输出信号来触发其他定时器（包括本文档其他部分所述的定时器）。通过查看关于设备上任何 TIMx_SMCR 寄存器的“TIMx 内部触发连接”表，可以确定哪些定时器可以作为目标从定时器。

OC1 信号的脉冲宽度必须至少设为目标定时器时钟周期的 2 倍，以便确保从定时器能够检测到触发。

例如，如果目标定时器 CK_INT 时钟频率是源定时器的 1/4，则 OC1 的脉冲宽度必须至少是 8 个源定时器时钟周期。

41.4.25 调试模式

当系统进入调试模式（处理器内核停止）时，TIMx 计数器会根据 DBGMCU 模块中的 TIMx 位选择继续正常工作或者停止工作。更多详细信息，请参见*第 59.10.6 节：微处理器调试单元 (DBGMCU)*。

为了安全起见，当计数器停止（DBGMCU_D2APB2FZ1 中的 TIMx = 1）时，输出被禁止（就像 MOE 位被复位一样）。可以将输出强制变为无效状态（OSSI 位 = 1），或者通过 GPIO 控制器（OSSI 位 = 0）来控制输出，以将其强制为高阻态。

41.5 TIM15 寄存器

有关寄存器说明中使用的缩写，请参见*第 1.2 节*。

外设寄存器可按半字（16 位）或字（32 位）访问。

41.5.1 TIM15 控制寄存器 1 (TIM15_CR1)

TIM15 control register 1

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
				rw		rw	rw	rw				rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD[1:0]**：时钟分频 (Clock division)
该位字段指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器 (TIx) 所使用的死区及采样时钟 (t_{DTS}) 之间的分频比
00：$t_{DTS} = t_{CK_INT}$
01：$t_{DTS} = 2 * t_{CK_INT}$
10：$t_{DTS} = 4 * t_{CK_INT}$
11：保留，不要设置成此值

位 7 **ARPE**：自动重装载预装载使能 (Auto-reload preload enable)
0：TIMx_ARR 寄存器不进行缓冲
1：TIMx_ARR 寄存器进行缓冲

位 6:4 保留，必须保持复位值。

位 3 **OPM**：单脉冲模式 (One-pulse mode)
0：计数器在发生更新事件时不会停止计数
1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

位 2 **URS**：更新请求源 (Update request source)
该位由软件置 1 和清零，用以选择 UEV 事件源。
0：使能后，所有以下事件都会生成更新事件。此类事件包括：
– 计数器上溢/下溢
– 将 UG 位置 1
– 通过从模式控制器生成的更新事件
1：使能后，只有计数器上溢/下溢会生成更新事件。

位 1 **UDIS**：更新禁止 (Update disable)

该位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

带缓冲的寄存器被加载为预加载数值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注意：只有事先通过软件将 CEN 位置 1，才可以使用外部时钟和门控模式。而触发模式可通过硬件自动将 CEN 位置 1。

41.5.2 TIM15 控制寄存器 2 (TIM15_CR2)

TIM15 control register 2

偏移地址：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
					rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位 15:11 保留，必须保持复位值。

位 10 **OIS2**：输出空闲状态 2（OC2 输出）(Output Idle state 2 (OC2 output))

0：当 MOE = 0 时，OC2 = 0

1：当 MOE = 0 时，OC2 = 1

注意：只要编程了 LOCK（TIM15_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，该位即无法修改。

位 9 **OIS1N**：输出空闲状态 1（OC1N 输出）(Output Idle state 1 (OC1N output))

0：当 MOE=0 时，经过死区时间后 OC1N=0

1：当 MOE=0 时，经过死区时间后 OC1N=1

注意：只要编程了 LOCK（TIM15_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，该位即无法修改。

位 8 **OIS1**：输出空闲状态 1（OC1 输出）(Output Idle state 1 (OC1 output))

0：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0

1：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=1

注意：只要编程了 LOCK（TIM15_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，该位即无法修改。

位 7 **TI1S**：TI1 选择 (TI1 selection)

0：TIMx_CH1 引脚连接到 TI1 输入

1：TIMx_CH1、TIMx_CH2 引脚连接到 TI1 输入（异或组合）

位 6:4 **MMS[2:0]**：主模式选择 (Master mode selection)

这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下：

000：**复位**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。

001：**使能**——计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑与运算组合而成。当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。

010：**更新**——选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器。

011：**比较脉冲**——CC1IF 标志置 1 时（即使已为高电平），只要发生捕获或比较匹配，触发输出都会发送一个正脉冲。(TRGO)。

100：**比较**——OC1REFC 信号用作触发输出 (TRGO)

101：**比较**——OC2REFC 信号用作触发输出 (TRGO)

位 3 **CCDS**：捕获/比较 DMA 选择 (Capture/compare DMA selection)

0：发生 CCx 事件时发送 CCx DMA 请求

1：发生更新事件时发送 CCx DMA 请求

位 2 **CCUS**：捕获/比较控制更新选择 (Capture/compare control update selection)

0：如果捕获/比较控制位 (CCPC=1) 进行预装载，仅通过将 COMG 位置 1 来对这些位进行更新。

1：如果捕获/比较控制位 (CCPC=1) 进行预装载，可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新。

注意：该位仅对具有互补输出的通道有效。

位 1 保留，必须保持复位值。

位 0 **CCPC**：捕获/比较预装载控制 (Capture/compare preloaded control)

0：CCxE、CCxNE 和 OCxM 位未进行预装载

1：CCxE、CCxNE 和 OCxM 位进行了预装载，写入这些位后，仅当发生换向事件 (COM)（COMG 位置 1 或在 TRGI 上检测到上升沿，取决于 CCUS 位）时才会对这些位进行更新。

注意：该位仅对具有互补输出的通道有效。

41.5.3 TIM15 从模式控制寄存器 (TIM15_SMCR)

TIM15 slave mode control register

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS[4:3]		Res.	Res.	Res.	SMS[3]
										rw	rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSM	TS[2:0]			Res.	SMS[2:0]		
								rw	rw	rw	rw		rw	rw	rw

位 31:22 保留，必须保持复位值。

位 19:17 保留，必须保持复位值。

位 15:8 保留，必须保持复位值。

位 7 **MSM**：主/从模式 (Master/slave mode)

0：无操作

1：当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于由单个外部事件对多个定时器进行同步的情况。

位 21、20、6、5、4 **TS[4:0]**：触发选择

该位字段可选择将要用于同步计数器的触发输入。

00000：内部触发 0 (ITR0)

00001：内部触发 1 (ITR1)

00010：内部触发 2 (ITR2)

00011：内部触发 3 (ITR3)

00100：TI1 边沿检测器 (TI1F_ED)

00101：滤波后的定时器输入 1 (TI1FP1)

00110：滤波后的定时器输入 2 (TI2FP2)

其他：保留

有关各定时器 ITRx 含义的详细信息，请参见*第 2151 页的表 364：TIMx 内部触发连接*。

注意：这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。

位 3 保留，必须保持复位值。

位 16、2、1、0 **SMS[3:0]**：从模式选择 (Slave mode selection)

选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。

0000： 禁止从模式——如果 CEN =“1”，预分频器时钟直接由内部时钟提供。

0001： 保留

0010： 保留

0011： 保留

0100： 复位模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。

0101： 门控模式——触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。

0110： 触发模式——触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。

0111： 外部时钟模式 1——由所选触发信号 (TRGI) 的上升沿提供计数器时钟。

1000： 组合复位 + 触发模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。

其他代码：保留。

注意：如果将 TI1F_ED 选作触发输入（TS=“00100”），则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。

注意：必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。

表 364. TIMx 内部触发连接

从 TIM	ITR0 (TS = 00000)	ITR1 (TS = 00001)	ITR2 (TS = 00010)	ITR3 (TS = 00011)
TIM15	TIM1	TIM3	TIM16, OC1	TIM17 OC1

41.5.4 TIM15 DMA/中断使能寄存器 (TIM15_DIER)

TIM15 DMA/interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	Res.	Res.	Res.	CC1DE	UDE	BIE	TIE	COMIE	Res.	Res.	CC2IE	CC1IE	UIE
	rw	rw				rw	rw	rw	rw	rw			rw	rw	rw

位 15 保留，必须保持复位值。

位 14 **TDE**：触发信号 DMA 请求使能 (Trigger DMA request enable)

0：禁止触发信号 DMA 请求

1：使能触发信号 DMA 请求

位 13 **COMDE**：COM DMA 请求使能 (COM DMA request enable)

0：禁止 COM DMA 请求

1：使能 COM DMA 请求

位 12:10 保留，必须保持复位值。

位 9 **CC1DE**：捕获/比较 1 DMA 请求使能 (Capture/Compare 1 DMA request enable)
0：禁止 CC1 DMA 请求
1：使能 CC1 DMA 请求

位 8 **UDE**：更新 DMA 请求使能 (Update DMA request enable)
0：禁止更新 DMA 请求
1：使能更新 DMA 请求

位 7 **BIE**：刹车中断使能 (Break interrupt enable)
0：禁止刹车中断
1：使能刹车中断

位 6 **TIE**：触发信号中断使能 (Trigger interrupt enable)
0：禁止触发信号中断
1：使能触发信号中断

位 5 **COMIE**：COM 中断使能 (COM interrupt enable)
0：禁止 COM 中断
1：使能 COM 中断

位 4:3 保留，必须保持复位值。

位 2 **CC2IE**：捕获/比较 2 中断使能 (Capture/Compare 2 interrupt enable)
0：禁止 CC2 中断
1：使能 CC2 中断

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断
1：使能 CC1 中断

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断
1：使能更新中断

41.5.5 TIM15 状态寄存器 (TIM15_SR)

TIM15 status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	Res.	Res.	CC2IF	CC1IF	UIF
					rc_w0	rc_w0		rc_w0	rc_w0	rc_w0			rc_w0	rc_w0	rc_w0

位 15:11 保留，必须保持复位值。

位 10 **CC2OF**：捕获/比较 2 过捕获标志 (Capture/compare 2 overcapture flag)
请参见 CC1OF 说明

位 9 **CC1OF**：捕获/比较 1 过捕获标志 (Capture/Compare 1 overcapture flag)
仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。
0：未检测到过捕获
1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8 保留，必须保持复位值。

位 7 **BIF**：刹车中断标志 (Break interrupt flag)

只要刹车输入变为有效状态，此标志便由硬件置 1。刹车输入无效后可通过软件对其清零。

0：未发生刹车事件

1：在刹车输入上检测到有效电平

位 6 **TIF**：触发中断标志 (Trigger interrupt flag)

该标志在发生 TRG 触发事件时由硬件置 1（在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，在门控模式情况下选择两个边沿）。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。

0：未发生触发事件

1：触发中断挂起

位 5 **COMIF**：COM 中断标志 (COM interrupt flag)

此标志在发生一个 COM 事件时（一旦捕获/比较控制位——CCxE、CCxNE、OCxM——已更新）由硬件置 1。但需要通过软件清零。

0：未发生 COM 事件

1：COM 中断挂起

位 4:3 保留，必须保持复位值。

位 2 **CC2IF**：捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag)

请参见 CC1IF 说明

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/Compare 1 interrupt flag)

该标志由硬件置 1，但需要通过软件清零（输入捕获或输出比较模式）或通过读取 TIMx_CCR1 寄存器清零（仅限输入捕获模式）。

0：未发生比较匹配/输入捕获

1：发生了比较匹配/输入捕获

如果通道 CC1 配置为输出：当计数器 TIMx_CNT 的值与 TIMx_CCR1 的值匹配时，此标志置 1。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。在中心对齐模式下，可通过 3 种方式将此标志置 1，完整说明请参见 TIMx_CR1 寄存器中的 CMS 位。

如果通道 CC1 配置为输入：当 TIMx_CCR1 寄存器中捕获到计数器值时（如果通过设置 TIMx_CCER 中的 CC1P 和 CC1NP 位定义为边沿有效，则改为在 IC1 上检测到边沿时），该位置 1。

位 0 **UIF**：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

– TIMx_CR1 寄存器中的 UDIS = 0，并且重复计数器值上溢时（重复计数器 = 0 时更新）。

– TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

– TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（请参见 *第 41.5.3 节：TIM15 从模式控制寄存器 (TIM15_SMCR)*）。

41.5.6 TIM15 事件产生寄存器 (TIM15_EGR)

TIM15 event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	Res.	Res.	CC2G	CC1G	UG
								w	w	rw			w	w	w

位 15:8 保留，必须保持复位值。

位 7 **BG**：刹车生成 (Break generation)
该位由软件置 1 以生成事件，并由硬件自动清零。
0：无操作
1：生成刹车事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 6 **TG**：触发生成 (Trigger generation)
该位由软件置 1 以生成事件，并由硬件自动清零。
0：无操作
1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 5 **COMG**：捕获/比较控制更新生成 (Capture/Compare control update generation)
该位可通过软件置 1，并由硬件自动清零。
0：无操作
1：CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位
注意：该位仅对具有互补输出的通道有效。

位 4:3 保留，必须保持复位值。

位 2 **CC2G**：捕获/比较 2 生成 (Capture/Compare 2 generation)
请参见 CC1G 说明

位 1 **CC1G**：捕获/比较 1 生成 (Capture/Compare 1 generation)
该位由软件置 1 以生成事件，并由硬件自动清零。
0：无操作
1：通道 1 上生成捕获/比较事件：
如果通道 CC1 配置为输出：
使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。
如果通道 CC1 配置为输入：
TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)
该位可通过软件置 1，并由硬件自动清零。
0：无操作
1：重新初始化计数器并生成一个寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。

41.5.7 TIM15 捕获/比较模式寄存器 1 (TIM15_CCMR1)

TIM15 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:12 **IC2F[3:0]**：输入捕获 2 滤波器 (Input capture 2 filter)

位 11:10 **IC2PSC[1:0]**：输入捕获 2 预分频器 (Input capture 2 prescaler)

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/Compare 2 selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出

01：CC2 通道配置为输入，IC2 映射到 TI2 上

10：CC2 通道配置为输入，IC2 映射到 TI1 上

11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据。

位 7:4 **IC1F[3:0]**：输入捕获 1 滤波器 (Input capture 1 filter)

该位字段可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样

0001：$f_{SAMPLING}=f_{CK_INT}$，N=2

0010：$f_{SAMPLING}=f_{CK_INT}$，N=4

0011：$f_{SAMPLING}=f_{CK_INT}$，N=8

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC[1:0]**：输入捕获 1 预分频器 (Input capture 1 prescaler)

该位字段定义 CC1 输入 (IC1) 的预分频比。只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
10：CC1 通道配置为输入，IC1 映射到 TI2 上
11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

41.5.8 TIM15 捕获/比较模式寄存器 1 [复用] (TIM15_CCMR1)

TIM15 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		Res.	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 23:17 保留，必须保持复位值。

位 15 保留，必须保持复位值。

位 24、14:12 **OC2M[3:0]**：输出比较 2 模式 (Output Compare 2 mode)

位 11 **OC2PE**：输出比较 2 预装载使能 (Output Compare 2 preload enable)

位 10 **OC2FE**：输出比较 2 快速使能 (Output Compare 2 fast enable)

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/Compare 2 selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出。

01：CC2 通道配置为输入，IC2 映射到 TI2 上。

10：CC2 通道配置为输入，IC2 映射到 TI1 上。

11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据。

位 7 保留，必须保持复位值。

位 16，6:4 **OC1M[3:0]**：输出比较 1 模式 (Output Compare 1 mode)

这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——只要 TIMx_CNT < TIMx_CCR1，通道 1 便输出有效电平，否则输出无效电平。

0111：PWM 模式 2——只要 TIMx_CNT < TIMx_CCR1，通道 1 便输出无效电平，否则输出有效电平。

1000：可再触发 OPM 模式 1——在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。

1001: 可再触发 OPM 模式 2——在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。

1010: 保留

1011: 保留

1100: 组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。

1101: 组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。

1110: 保留。

1111：保留。

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。

在 PWM 模式下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。

该位字段将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成 COM 事件时，OC1M 有效位才会从预装载位获取新值。

OC1M[3] 位不是连续的，而是在位 16 中。

位 3 **OC1PE**：输出比较 1 预装载使能 (Output Compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到影子寄存器中。

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。

位 2 **OC1FE**：输出比较 1 快速使能 (Output Compare 1 fast enable)

该位可降低触发事件与定时器输出跳变之间的延迟。单脉冲模式下必须使用（TIMx_CR1 寄存器中的 OPM 位置 1），以便在启动触发后快速发送输出脉冲。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出。

01：CC1 通道配置为输入，IC1 映射到 TI1 上。

10：CC1 通道配置为输入，IC1 映射到 TI2 上。

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

41.5.9 TIM15 捕获/比较使能寄存器 (TIM15_CCER)

TIM15 capture/compare enable register

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
								rw		rw	rw	rw	rw	rw	rw

位 15:8 保留，必须保持复位值。

位 7 **CC2NP**：捕获/比较 2 互补输出极性 (Capture/Compare 2 complementary output polarity)
请参见 CC1NP 说明

位 6 保留，必须保持复位值。

位 5 **CC2P**：捕获/比较 2 输出极性 (Capture/Compare 2 output polarity)
请参见 CC1P 说明

位 4 **CC2E**：捕获/比较 2 输出使能 (Capture/Compare 2 output enable)
请参见 CC1E 说明

位 3 **CC1NP**：捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output polarity)

CC1 通道配置为输出：

0：OC1N 高电平有效

1：OC1N 低电平有效

CC1 通道配置为输入：

该位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1P 说明。

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3 且 CC1S=“00”（通道配置为输出），该位立即变为不可写状态。

该位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NP 有效位才会从预装载位获取新值。

位 2 **CC1NE**：捕获/比较 1 互补输出使能 (Capture/Compare 1 complementary output enable)

0：关闭——OC1N 未激活。OC1N 电平与 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位存在函数关系。

1：开启——基于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值，在相应输出引脚上输出 OC1N 信号。

位 1 **CC1P**：捕获/比较 1 输出极性 (Capture/Compare 1 output polarity)

0：OC1 高电平有效（输出模式）/边沿有效性选择（输入模式，见下文）

1：OC1 低电平有效（输出模式）/边沿有效性选择（输入模式，见下文）

CC1 通道配置为输入时，CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。

CC1NP=0，CC1P=0：非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。

CC1NP=0，CC1P=1：反相/下降沿触发。电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。

CC1NP=1，CC1P=1：非反相/上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。

CC1NP=1，CC1P=0：该配置保留，不得使用。

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3，该位立即变为不可写状态。

该位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。

位 0 **CC1E**：捕获/比较 1 输出使能 (Capture/Compare 1 output enable)

0：禁止捕获模式/OC1 无效（见下文）

1：使能捕获模式/在相应输出引脚上输出 OC1 信号

当 CC1 通道配置为输出时，OC1 电平取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位，与 CC1E 位的状态无关。有关详细信息，请参见*表 365*。

表 365. 具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM15)

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	X	0	0	禁止输出（不由定时器驱动：高阻态） OCx=0 OCxN=0、OCxN_EN=0	
		0	0	1	禁止输出（不由定时器驱动：高阻态） OCx=0	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		0	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	禁止输出（不由定时器驱动：高阻态） OCxN=0
		X	1	1	OCREF + 极性 + 死区	OCREF 互补项（对 OCREF 进行“非”运算）+ 极性 + 死区
		1	0	1	关闭状态（输出为无效状态） OCx=CCxP	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		1	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP、OCx_EN=1	关闭状态（输出为无效状态） OCxN=CCxNP、OCxN_EN=1
0	0	X	X	X	禁止输出（不由定时器驱动：高阻态）	
	1		0	0		
			0	1	关闭状态（输出为无效状态） 异步：OCx=CCxP、OCxN=CCxNP 如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，从而假定 OISx 和 OISxN 在有效状态下与 OCX 和 OCxN 不对应	
			1	0		
			1	1		

1. 如果一个通道的两个输出均未使用（由 GPIO 控制器接管控制），则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

注： *与互补通道 OCx 和 OCxN 相连的外部 I/O 引脚的状态取决于通道 OCx 和 OCxN 的状态以及 GPIO 控制和复用功能寄存器。*

41.5.10 TIM15 计数器 (TIM15_CNT)

TIM15 counter

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIF CPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **UIFCPY**：UIF 副本 (UIF Copy)
该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

41.5.11 TIM15 预分频器 (TIM15_PSC)

TIM15 prescaler

偏移地址：0x28

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)
计数器时钟频率 (CK_CNT) 等于 f_{CK_PSC} / (PSC[15:0] + 1)。
PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到实际预分频器寄存器的值。

41.5.12 TIM15 自动重载寄存器 (TIM15_ARR)

TIM15 auto-reload register

偏移地址：0x2C

复位值：0xFFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)
ARR 为要装载到实际自动重载寄存器的值。
有关 ARR 更新和行为的更多详细信息，请参见*第 2115 页的第 41.4.1 节：时基单元*。
当自动重载值为空时，计数器不工作。

41.5.13 TIM15 重复计数器寄存器 (TIM15_RCR)

TIM15 repetition counter register

偏移地址：0x30

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 15:8 保留，必须保持复位值。

位 7:0 **REP[7:0]**：重复计数器值 (Repetition counter value)

使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即，从预装载寄存器向影子寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。

与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP 值，因此在生成下一重复更新事件之前，向 TIMx_RCR 寄存器写入任何值都无影响。

这意味着在 PWM 模式下，每产生 (REP+1) 个 PWM 脉冲就产生更新事件。

41.5.14 TIM15 捕获/比较寄存器 1 (TIM15_CCR1)

TIM15 capture/compare register 1

偏移地址：0x34

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR1[15:0]**：捕获/比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：

CCR1 是捕获/比较寄存器 1 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 1）。

实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出相应电平的值。

如果通道 CC1 配置为输入：

CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。

41.5.15 TIM15 捕获/比较寄存器 2 (TIM15_CCR2)

TIM15 capture/compare register 2

偏移地址：0x38

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR2[15:0]**：捕获/比较 2 值 (Capture/Compare 2 value)

如果通道 CC2 配置为输出：

CCR2 是捕获/比较寄存器 2 的预装载值。

如果没有通过 TIMx_CCMR2 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 2）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。

如果通道 CC2 配置为输入：

CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。

41.5.16 TIM15 刹车和死区寄存器 (TIM15_BDTR)

TIM15 break and dead-time register

偏移地址：0x44

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	BKBID	Res.	BK DSRM	Res.	Res.	Res.	Res.	Res.	Res.	BKF[3:0]			
			rw		rw							rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注： *由于可以根据 LOCK 配置锁定位 BKBID、BKDSRM、BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0] 位的写操作，因此必须在第一次对 TIMx_BDTR 寄存器执行写访问时对这些位进行配置。*

位 31:29 保留，必须保持复位值。

位 28 **BKBID**：双向刹车 (Break bidirectional)

0：刹车输入 BRK 为输入模式

1：刹车输入 BRK 为双向模式

在双向模式下（BKBID 位置 1），刹车输入配置为输入模式和开漏输出模式。任何激活的刹车事件都将使刹车输入上呈逻辑低电平，以向外部设备指示发生了内部刹车事件。

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 27 保留，必须保持复位值。

位 26 **BKDSRM**：刹车解除 (Break Disarm)

0：启动刹车输入 BRK

1：解除刹车输入 BRK

当没有刹车源被激活时，该位由硬件清零。

必须通过软件将 BKDSRM 位置 1 以释放双向输出控制（开漏输出处于高阻态），然后不断轮询该位，直到其由硬件复位，指示故障条件已消失。

注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 25:20 保留，必须保持复位值。

位 19:16 **BKF[3:0]**：刹车滤波器 (Break filter)

该位字段可定义 BRK 输入信号的采样频率和适用于 BRK 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：

0000：无滤波器，BRK 异步工作

0001：$f_{SAMPLING}=f_{CK_INT}$，N=2

0010：$f_{SAMPLING}=f_{CK_INT}$，N=4

0011：$f_{SAMPLING}=f_{CK_INT}$，N=8

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

位 15 **MOE**：主输出使能 (Main output enable)

只要刹车输入变为有效状态，该位便由硬件异步清零。该位由软件置 1，也可根据 AOE 位状态自动置 1。该位仅对配置为输出的通道有效。

0：OC 和 OCN 输出被禁止或被强制为空闲状态，具体取决于 OSSI 位。

1：如果 OC 和 OCN 输出的相应使能位（TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位）均置 1，则使能 OC 和 OCN 输出。

有关详细信息，请参见 OC/OCN 使能说明（*第 2158 页的第 41.5.9 节：TIM15 捕获/比较使能寄存器 (TIM15_CCER)*）。

位 14 **AOE**：自动输出使能 (Automatic output enable)

0：MOE 只能由软件置 1

1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1（如果刹车输入无效）

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 13 **BKP**：刹车极性 (Break polarity)

0：刹车输入 BRK 为低电平有效

1：刹车输入 BRK 为高电平有效

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。
对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 12 **BKE**：刹车使能 (Break enable)

0：禁止刹车输入（BRK 和 CCS 时钟故障事件）

1：使能刹车输入（BRK 和 CCS 时钟故障事件）

编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 11 **OSSR**：运行模式下的关闭状态选择 (Off-state selection for Run mode)

该位在 MOE = 1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。

有关详细信息，请参见 OC/OCN 使能说明（*第 2158 页的第 41.5.9 节：TIM15 捕获/比较使能寄存器 (TIM15_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（输出控制器呈高阻态，相应输出由 GPIO 接管）

1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平（输出仍由定时器控制）。

注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，该位即无法修改。

位 10 **OSSI**：空闲模式下的关闭状态选择 (Off-state selection for Idle mode)

该位在 MOE=0 时作用于配置为输出的通道。

有关详细信息，请参见 OC/OCN 使能说明（*第 2158 页的第 41.5.9 节：TIM15 捕获/比较使能寄存器 (TIM15_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号=0）

1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便将 OC/OCN 输出首先强制为其空闲电平。然后设置 OC/OCN 使能输出信号=1

注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，该位即无法修改。

位 9:8 **LOCK[1:0]**：锁定配置 (Lock configuration)

这些位用于针对软件错误提供写保护。

00：关闭锁定——不对任何位提供写保护

01：锁定级别 1，此时无法对 TIMx_BDTR 寄存器中的 DTG 位、TIMx_CR2 寄存器中的 OISx 和 OISxN 位以及 TIMx_BDTR 寄存器中的 BKE/BKP/AOE 位执行写操作。

10：锁定级别 2，此时无法对锁定级别 1 中适用的各位、CC 极性位（TIMx_CCER 寄存器中的 CCxP/CCxNP 位，只要通过 CCxS 位将相关通道配置为输出）以及 OSSR 和 OSSI 位执行写操作。

11：锁定级别 3，此时无法对锁定级别 2 中适用的各位、CC 控制位（TIMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。

注意：复位后只能对 LOCK 位执行一次写操作。对 TIMx_BDTR 寄存器执行写操作后其中的内容将冻结，直到下一次复位。

位 7:0 **DTG[7:0]**：配置死区发生器 (Dead-time generator setup)

该位字段定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。

DTG[7:5] = 0xx => DT = DTG[7:0] x t_{dtg}，其中 $t_{dtg} = t_{DTS}$

DTG[7:5] = 10x => DT = (64 + DTG[5:0]) x t_{dtg}，其中 $t_{dtg} = 2 \times t_{DTS}$

DTG[7:5] = 110 => DT = (32 + DTG[4:0]) x t_{dtg}，其中 $t_{dtg} = 8 \times t_{DTS}$

DTG[7:5] = 111 => DT = (32 + DTG[4:0]) x t_{dtg}，其中 $t_{dtg} = 16 \times t_{DTS}$

示例：如果 t_{DTS} = 125 ns (8 MHz)，则可能的死区值为：

0 到 15875 ns（步长为 125 ns），

16 µs 到 31750 ns（步长为 250 ns），

32 µs 到 63 µs（步长为 1 µs），

64 µs 到 126 µs（步长为 2 µs）

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，该位字段即无法修改。

41.5.17 TIM15 DMA 控制寄存器 (TIM15_DCR)

TIM15 DMA control register

偏移地址：0x48

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 15:13 保留，必须保持复位值。

位 12:8 **DBL[4:0]**：DMA 分组传送长度 (DMA burst length)

该 5 位字段定义了 DMA 的传送长度（当对 TIMx_DMAR 地址进行读或写时，定时器进行一次突发传输）。

00000：1 次传输，
00001：2 次传输，
00010：3 次传输，
...
10001：18 次传输。

位 7:5 保留，必须保持复位值。

位 4:0 **DBA[4:0]**：DMA 基址 (DMA base address)

该 5 位字段定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：
00000：TIMx_CR1，
00001：TIMx_CR2，
00010：TIMx_SMCR，
...

41.5.18 TIM15 全传输 DMA 地址 (TIM15_DMAR)

TIM15 DMA address for full transfer

偏移地址：0x4C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **DMAB[15:0]**：DMA 分组传送寄存器 (DMA register for burst accesses)

对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器

(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。

41.5.19 TIM15 复用寄存器 1 (TIM15_AF1)

TIM15 alternate register 1

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	BKINP	BKDF1 BK0E	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINE
						rw	rw								rw

位 31:10 保留，必须保持复位值。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
该位选择 BKIN 复用功能输入有效电平，必须与 BKP 极性位一起编程。
0：BKIN 输入为低电平有效
1：BKIN 输入为高电平有效
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 8 **BKDF1BK0E**：BRK dfsdm1_break[0] 使能 (BRK dfsdm1_break[0] enable)
该位使能定时器 BRK 输入的 dfsdm1_break[0]。dfsdm1_break[0] 输出与其他 BRK 源进行“或”运算。
0：禁止 dfsdm1_break[0] 输入
1：使能 dfsdm1_break[0] 输入
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 7:1 保留，必须保持复位值。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
该位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

41.5.20 TIM15 输入选择寄存器 (TIM15_TISEL)

TIM15 input selection register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：选择 TI2[0] 到 TI2[15] 输入 (selects TI2[0] to TI2[15] input)
0000：TIM15_CH2 输入
0001：TIM2_CH2 输入
0010：TIM3_CH2 输入
0011：TIM4_CH2 输入
其他值：保留

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)
0000：TIM15_CH1 输入
0001：TIM2_CH1 输入
0010：TIM3_CH1 输入
0011：TIM4_CH1 输入
0100：LSE
0101：CSI
0110：MCO2
0111: hsi_cal_ck
1000: csi_cal_ck
其他：保留

41.5.21 TIM15 寄存器映射

TIM15 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 366. TIM15 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIM15_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0				0	0	0	0
0x04	**TIM15_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
	Reset value																						0	0	0	0	0	0	0	0	0		0
0x08	**TIM15_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS [4:3]		Res.	Res.	Res.	SMS[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value											0	0				0									0	0	0	0		0	0	0
0x0C	**TIM15_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	COMDE	Res.	Res.	Res.	CC1DE	UDE	BIE	TIE	COMIE	Res.	Res.	CC2IE	CC1IE	UIE
	Reset value																		0	0				0	0	0	0	0			0	0	0
0x10	**TIM15_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	Res.	Res.	CC2IF	CC1IF	UIF
	Reset value																						0	0		0	0	0			0	0	0

表 366. TIM15 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x14	**TIM15_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	Res.	Res.	CC2G	CC1G	UG
	Reset value																									0	0	0			0	0	0
0x18	**TIM15_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value								0								0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
	TIM15_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2 PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**TIM15_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	Reset value																									0		0	0	0	0	0	0
0x24	**TIM15_CNT**	UIFCPY or Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIM15_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIM15_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	**TIM15_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x34	**TIM15_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIM15_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIM15_BDTR**	Res.	Res.	Res.	BKBID	Res.	BKDSRM	Res.	Res.	Res.	Res.	Res.	Res.	BKF[3:0]				MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DTG[7:0]							
	Reset value				0		0							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 366. TIM15 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x48	**TIM15_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIM15_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**TIM15_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINP	BKDF1BK0E	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINE
	Reset value																							0	0								1
0x68	**TIM15_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value																					0	0	0	0					0	0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

41.6 TIM16/TIM17 寄存器

有关寄存器说明中使用的缩写，请参见*第 1.2 节*。

外设寄存器可按半字（16 位）或字（32 位）访问。

41.6.1 TIMx 控制寄存器 1 (TIMx_CR1)（x = 16 到 17）

TIMx control register 1

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
				rw		rw	rw	rw				rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD[1:0]**：时钟分频 (Clock division)
该位字段指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器所使用的采样时钟 (t_{DTS}) 之间的分频比。
00：$t_{DTS} = t_{CK_INT}$
01：$t_{DTS} = 2 * t_{CK_INT}$
10：$t_{DTS} = 4 * t_{CK_INT}$
11：保留，不要设置成此值

位 7 **ARPE**：自动重装载预装载使能 (Auto-reload preload enable)
0：TIMx_ARR 寄存器不进行缓冲
1：TIMx_ARR 寄存器进行缓冲

位 6:4 保留，必须保持复位值。

位 3 **OPM**：单脉冲模式 (One pulse mode)
0：计数器在发生更新事件时不会停止计数
1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

位 2 **URS**：更新请求源 (Update request source)
该位由软件置 1 和清零，用以选择 UEV 事件源。
0：使能时，所有以下事件都会生成更新中断或 DMA 请求。此类事件包括：
– 计数器上溢/下溢
– 将 UG 位置 1
– 通过从模式控制器生成的更新事件
1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。

位 1 **UDIS**：更新禁止 (Update disable)

该位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

带缓冲的寄存器被加载为预加载数值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注意：只有事先通过软件将 CEN 位置 1，才可以使用外部时钟和门控模式。而触发模式可通过硬件自动将 CEN 位置 1。

41.6.2 TIMx 控制寄存器 2 (TIMx_CR2)（x = 16 到 17）

TIMx control register 2

偏移地址：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1	Res.	Res.	Res.	Res.	CCDS	CCUS	Res.	CCPC
						rw	rw					rw	rw		rw

位 15:10 保留，必须保持复位值。

位 9 **OIS1N**：输出空闲状态 1（OC1N 输出）(Output Idle state 1 (OC1N output))

0：当 MOE=0 时，经过死区时间后 OC1N=0

1：当 MOE=0 时，经过死区时间后 OC1N=1

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，该位即无法修改。

位 8 **OIS1**：输出空闲状态 1（OC1 输出）(Output Idle state 1 (OC1 output))

0：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0

1：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=1

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，该位即无法修改。

位 7:4 保留，必须保持复位值。

位 3 **CCDS**：捕获/比较 DMA 选择 (Capture/compare DMA selection)

0：发生 CCx 事件时发送 CCx DMA 请求

1：发生更新事件时发送 CCx DMA 请求

位 2 **CCUS**：捕获/比较控制更新选择 (Capture/compare control update selection)

0：如果捕获/比较控制位 (CCPC=1) 进行预装载，仅通过将 COMG 位置 1 来对这些位进行更新。

1：如果捕获/比较控制位 (CCPC=1) 进行预装载，可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新。

注意：该位仅对具有互补输出的通道有效。

位 1 保留，必须保持复位值。

位 0 **CCPC**：捕获/比较预装载控制 (Capture/compare preloaded control)
0：CCxE、CCxNE 和 OCxM 位未进行预装载
1：CCxE、CCxNE 和 OCxM 位在写入后被预加载，只有当 COM 位置 1 时才进行更新。
注意：该位仅对具有互补输出的通道有效。

41.6.3 TIMx DMA/中断使能寄存器 (TIMx_DIER)（x = 16 到 17）

TIMx DMA/interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1DE	UDE	BIE	Res.	COMIE	Res.	Res.	Res.	CC1IE	UIE
						rw	rw	rw		rw				rw	rw

位 15:10 保留，必须保持复位值。

位 9 **CC1DE**：捕获/比较 1 DMA 请求使能 (Capture/Compare 1 DMA request enable)
0：禁止 CC1 DMA 请求
1：使能 CC1 DMA 请求

位 8 **UDE**：更新 DMA 请求使能 (Update DMA request enable)
0：禁止更新 DMA 请求
1：使能更新 DMA 请求

位 7 **BIE**：刹车中断使能 (Break interrupt enable)
0：禁止刹车中断
1：使能刹车中断

位 6 保留，必须保持复位值。

位 5 **COMIE**：COM 中断使能 (COM interrupt enable)
0：禁止 COM 中断
1：使能 COM 中断

位 4:2 保留，必须保持复位值。

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断
1：使能 CC1 中断

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断
1：使能更新中断

41.6.4 TIMx 状态寄存器 (TIMx_SR)（x = 16 到 17）

TIMx status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	BIF	Res.	COMIF	Res.	Res.	Res.	CC1IF	UIF
						rc_w0		rc_w0		rc_w0				rc_w0	rc_w0

位 15:10 保留，必须保持复位值。

位 9 **CC1OF**：捕获/比较 1 过捕获标志 (Capture/Compare 1 overcapture flag)

仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。

0：未检测到过捕获

1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8 保留，必须保持复位值。

位 7 **BIF**：刹车中断标志 (Break interrupt flag)

只要刹车输入变为有效状态，此标志便由硬件置 1。刹车输入无效后可通过软件对其清零。

0：未发生刹车事件

1：在刹车输入上检测到有效电平

位 6 保留，必须保持复位值。

位 5 **COMIF**：COM 中断标志 (COM interrupt flag)

此标志在发生一个 COM 事件时（一旦捕获/比较控制位——CCxE、CCxNE、OCxM——已更新）由硬件置 1。但需要通过软件清零。

0：未发生 COM 事件

1：COM 中断挂起

位 4:2 保留，必须保持复位值。

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/Compare 1 interrupt flag)

该标志由硬件置 1，但需要通过软件清零（输入捕获或输出比较模式）或通过读取 TIMx_CCR1 寄存器清零（仅限输入捕获模式）。

0：未发生比较匹配/输入捕获

1：发生了比较匹配/输入捕获

如果通道 CC1 配置为输出：当计数器 TIMx_CNT 的值与 TIMx_CCR1 的值匹配时，此标志置 1。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。在中心对齐模式下，可通过 3 种方式将此标志置 1，完整说明请参见 TIMx_CR1 寄存器中的 CMS 位。

如果通道 CC1 配置为输入：当 TIMx_CCR1 寄存器中捕获到计数器值时（如果通过设置 TIMx_CCER 中的 CC1P 和 CC1NP 位定义为边沿有效，则改为在 IC1 上检测到边沿时），该位置 1。

位 0 **UIF**：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

- TIMx_CR1 寄存器中的 UDIS = 0，并且重复计数器值上溢时（重复计数器 = 0 时更新）。
- TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

41.6.5 TIMx 事件生成寄存器 (TIMx_EGR)（x = 16 到 17）

TIMx event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	Res.	COMG	Res.	Res.	Res.	CC1G	UG
								w		w				w	w

位 15:8 保留，必须保持复位值。

位 7 **BG**：刹车生成 (Break generation)
该位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作。
1：生成刹车事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 6 保留，必须保持复位值。

位 5 **COMG**：捕获/比较控制更新生成 (Capture/Compare control update generation)
该位可通过软件置 1，并由硬件自动清零。
0：无操作
1：CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位
注意：该位仅对具有互补输出的通道有效。

位 4:2 保留，必须保持复位值。

位 1 **CC1G**：捕获/比较 1 生成 (Capture/Compare 1 generation)
该位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作。
1：通道 1 上生成捕获/比较事件：
如果通道 CC1 配置为输出：
使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。
如果通道 CC1 配置为输入：
TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)
该位可通过软件置 1，并由硬件自动清零。
0：不执行任何操作。
1：重新初始化计数器并生成一个寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。

41.6.6 TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1)（x = 16 到 17）

TIMx capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:4 **IC1F[3:0]**：输入捕获 1 滤波器 (Input capture 1 filter)

该位字段可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样

0001：$f_{SAMPLING}=f_{CK_INT}$，N=2

0010：$f_{SAMPLING}=f_{CK_INT}$，N=4

0011：$f_{SAMPLING}=f_{CK_INT}$，N=8

0100：$f_{SAMPLING}=f_{DTS}/2$，N=

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC[1:0]**：输入捕获 1 预分频器 (Input capture 1 prescaler)

该位字段定义 CC1 输入 (IC1) 的预分频比。

只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获。

01：每发生 2 个事件便执行一次捕获

10：每发生 4 个事件便执行一次捕获

11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

其他值：保留

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

41.6.7 TIMx 捕获/比较模式寄存器 1 [复用] (TIMx_CCMR1)（x = 16 到 17）

TIMx capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
									rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 15:7 保留，必须保持复位值。

位 16，6:4 **OC1M[3:0]**：输出比较 1 模式 (Output Compare 1 mode)

这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。当定时器用作软件时基时，可以使用该模式。在定时器操作期间使能冻结模式时，输出保持进入冻结状态前的状态（有效或无效）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——只要 TIMx_CNT < TIMx_CCR1，通道 1 便输出有效电平，否则输出无效电平。

0111：PWM 模式 2——只要 TIMx_CNT < TIMx_CCR1，通道 1 便输出无效电平，否则输出有效电平。

所有其他值：保留

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。

在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。

OC1M[3] 位不是连续的，而是在位 16 中。

位 3 **OC1PE**：输出比较 1 预装载使能 (Output Compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到影子寄存器中。

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。

位 2 **OC1FE**：输出比较 1 快速使能 (Output Compare 1 fast enable)

该位可降低触发事件与定时器输出跳变之间的延迟。单脉冲模式下必须使用（TIMx_CR1 寄存器中的 OPM 位置 1），以便在启动触发后快速发送输出脉冲。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OC1FE 才会起作用。

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)

该位字段定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

其他值：保留

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

41.6.8 TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 16 到 17）

TIMx capture/compare enable register

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E
												rw	rw	rw	rw

位 15:4 保留，必须保持复位值。

位 3 **CC1NP**：捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output polarity)

CC1 通道配置为输出：

0：OC1N 高电平有效

1：OC1N 低电平有效

CC1 通道配置为输入：

该位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1P 说明。

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3 且 CC1S=“00”（通道配置为输出），该位立即变为不可写状态。

该位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NP 有效位才会从预装载位获取新值。

位 2 **CC1NE**：捕获/比较 1 互补输出使能 (Capture/Compare 1 complementary output enable)

0：关闭——OC1N 未激活。OC1N 电平与 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位存在函数关系。

1：开启——基于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值，在相应输出引脚上输出 OC1N 信号。

位 1 **CC1P**：捕获/比较 1 输出极性 (Capture/Compare 1 output polarity)

0：OC1 高电平有效（输出模式）/边沿有效性选择（输入模式，见下文）

1：OC1 低电平有效（输出模式）/边沿有效性选择（输入模式，见下文）

CC1 通道配置为输入时，CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。

CC1NP=0，CC1P=0：非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。

CC1NP=0，CC1P=1：反相/下降沿触发。电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。

CC1NP=1，CC1P=1：非反相/上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。

CC1NP=1，CC1P=0：该配置保留，不得使用。

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3，该位立即变为不可写状态。

该位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。

位 0 **CC1E**：捕获/比较 1 输出使能 (Capture/Compare 1 output enable)

0：禁止捕获模式/OC1 无效（见下文）

1：使能捕获模式/在相应输出引脚上输出 OC1 信号

当 CC1 通道配置为输出时，OC1 电平取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位，与 CC1E 位的状态无关。有关详细信息，请参见*表 367*。

表 367. 具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM16/17)

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	X	0	0	禁止输出（不由定时器驱动：高阻态） OCx=0 OCxN=0、OCxN_EN=0	
		0	0	1	禁止输出（不由定时器驱动：高阻态） OCx=0	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		0	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	禁止输出（不由定时器驱动：高阻态） OCxN=0
		X	1	1	OCREF + 极性 + 死区	OCREF 互补项（对 OCREF 进行“非”运算）+ 极性 + 死区
		1	0	1	关闭状态（输出为无效状态） OCx=CCxP	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		1	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP、OCx_EN=1	关闭状态（输出为无效状态） OCxN=CCxNP、OCxN_EN=1

表 367. 具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM16/17)（续）

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
0	0	X	X	X	禁止输出（不由定时器驱动：高阻态）。	
	1		0	0		
			0	1	关闭状态（输出为无效状态） 异步：OCx=CCxP、OCxN=CCxNP 如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，从而假定 OISx 和 OISxN 在有效状态下与 OCX 和 OCxN 不对应	
			1	0		
			1	1		

1. 如果一个通道的两个输出均未使用（由 GPIO 控制器接管控制），则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

注： *与互补通道 OCx 和 OCxN 相连的外部 I/O 引脚的状态取决于通道 OCx 和 OCxN 的状态以及 GPIO 控制和复用功能寄存器。*

41.6.9 TIMx 计数器 (TIMx_CNT)（x = 16 到 17）

TIMx counter

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIF CPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **UIFCPY**：UIF 副本 (UIF Copy)

该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。如果 TIMx_CR1 中的 UIFREMAP 位复位，则位 31 保留，读为 0。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

41.6.10 TIMx 预分频器 (TIMx_PSC)（x = 16 到 17）

TIMx prescaler

偏移地址：0x28

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)

计数器时钟频率 (CK_CNT) 等于 f_{CK_PSC} / (PSC[15:0] + 1)。

PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到实际预分频器寄存器的值。

41.6.11 TIMx 自动重载寄存器 (TIMx_ARR)（x = 16 到 17）

TIMx auto-reload register

偏移地址：0x2C

复位值：0xFFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)

ARR 为要装载到实际自动重载寄存器的值。

有关 ARR 更新和行为的更多详细信息，请参见*第 2115 页的第 41.4.1 节：时基单元*。

当自动重载值为空时，计数器不工作。

41.6.12 TIMx 重复计数器寄存器 (TIMx_RCR)（x = 16 到 17）

TIMx repetition counter register

偏移地址：0x30

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 15:8 保留，必须保持复位值。

位 7:0 **REP[7:0]**：重复计数器值 (Repetition counter value)

使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即，从预装载寄存器向影子寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。

与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP 值，因此在生成下一重复更新事件之前，向 TIMx_RCR 寄存器写入任何值都无影响。

这意味着在 PWM 模式下，每产生 (REP+1) 个 PWM 脉冲就产生更新事件。

41.6.13 TIMx 捕获/比较寄存器 1 (TIMx_CCR1)（x = 16 到 17）

TIMx capture/compare register 1

偏移地址：0x34

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR1[15:0]**：捕获/比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：

CCR1 是捕获/比较寄存器 1 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 1）。

实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出相应电平的值。

如果通道 CC1 配置为输入：

CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。

41.6.14 TIMx 刹车和死区寄存器 (TIMx_BDTR)（x = 16 到 17）

TIMx break and dead-time register

偏移地址：0x44

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	BKBID	Res.	BK DSRM	Res.	Res.	Res.	Res.	Res.	Res.	BKF[3:0]			
			rw		rw							rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注： *由于可以根据 LOCK 配置锁定位 BKBID、BKDSRM、BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0] 位的写操作，因此必须在第一次对 TIMx_BDTR 寄存器执行写访问时对这些位进行配置。*

位 31:29 保留，必须保持复位值。

位 28 **BKBID**：双向刹车 (Break bidirectional)

0：刹车输入 BRK 为输入模式

1：刹车输入 BRK 为双向模式

在双向模式下（BKBID 位置 1），刹车输入配置为输入模式和开漏输出模式。任何激活的刹车事件都将使刹车输入上呈逻辑低电平，以向外部设备指示发生了内部刹车事件。

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 27 保留，必须保持复位值。

位 26 **BKDSRM**：刹车解除 (Break Disarm)

0：启动刹车输入 BRK

1：解除刹车输入 BRK

当没有刹车源被激活时，该位由硬件清零。

必须通过软件将 BKDSRM 位置 1 以释放双向输出控制（开漏输出处于高阻态），然后不断轮询该位，直到其由硬件复位，指示故障条件已消失。

注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 25:20 保留，必须保持复位值。

位 19:16 **BKF[3:0]**：刹车滤波器 (Break filter)

该位字段可定义 BRK 输入的采样频率和适用于 BRK 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：

0000：无滤波器，BRK 异步工作

0001：$f_{SAMPLING}=f_{CK_INT}$，N=2

0010：$f_{SAMPLING}=f_{CK_INT}$，N=4

0011：$f_{SAMPLING}=f_{CK_INT}$，N=8

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

位 15 **MOE**：主输出使能 (Main output enable)

只要刹车输入变为有效状态，该位便由硬件异步清零。该位由软件置 1，也可根据 AOE 位状态自动置 1。该位仅对配置为输出的通道有效。

0：OC 和 OCN 输出被禁止或被强制为空闲状态，具体取决于 OSSI 位。

1：如果 OC 和 OCN 输出的相应使能位（TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位）均置 1，则使能 OC 和 OCN 输出。

有关详细信息，请参见 OC/OCN 使能说明（*第 2178 页的第 41.6.8 节：TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 16 到 17）*）。

位 14 **AOE**：自动输出使能 (Automatic output enable)

0：MOE 只能由软件置 1

1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1（如果刹车输入无效）

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 13 **BKP**：刹车极性 (Break polarity)

0：刹车输入 BRK 为低电平有效

1：刹车输入 BRK 为高电平有效

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。
对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 12 **BKE**：刹车使能 (Break enable)

0：禁止刹车输入（BRK 和 CCS 时钟故障事件）

1：使能刹车输入（BRK 和 CCS 时钟故障事件）

注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。
对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 11 **OSSR**：运行模式下的关闭状态选择 (Off-state selection for Run mode)

该位在 MOE = 1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。

有关详细信息，请参见 OC/OCN 使能说明（*第 2178 页的第 41.6.8 节：TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 16 到 17）*）。

0：处于无效状态时，禁止 OC/OCN 输出（输出控制器呈高阻态，相应输出由 GPIO 接管）

1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平（输出仍由定时器控制）。

注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，该位即无法修改。

位 10 **OSSI**：空闲模式下的关闭状态选择 (Off-state selection for Idle mode)

该位在 MOE=0 时作用于配置为输出的通道。

有关详细信息，请参见 OC/OCN 使能说明（*第 2178 页的第 41.6.8 节：TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 16 到 17）*）。

0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号 =0）

1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便将 OC/OCN 输出首先强制为其空闲电平。然后设置 OC/OCN 使能输出信号 =1

注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，该位即无法修改。

位 9:8 **LOCK[1:0]**：锁定配置 (Lock configuration)

这些位用于针对软件错误提供写保护。

00：关闭锁定——不对任何位提供写保护

01：锁定级别 1，此时无法对 TIMx_BDTR 寄存器中的 DTG 位、TIMx_CR2 寄存器中的 OISx 和 OISxN 位以及 TIMx_BDTR 寄存器中的 BKE/BKP/AOE 位执行写操作。

10：锁定级别 2，此时无法对锁定级别 1 中适用的各位、CC 极性位（TIMx_CCER 寄存器中的 CCxP/CCxNP 位，只要通过 CCxS 位将相关通道配置为输出）以及 OSSR 和 OSSI 位执行写操作。

11：锁定级别 3，此时无法对锁定级别 2 中适用的各位、CC 控制位（TIMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。

注意：复位后只能对 LOCK 位执行一次写操作。对 TIMx_BDTR 寄存器执行写操作后其中的内容将冻结，直到下一次复位。

位 7:0 **DTG[7:0]**：配置死区发生器 (Dead-time generator setup)

该位字段定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。

DTG[7:5] = 0xx => DT = DTG[7:0] x t_{dtg}，其中 $t_{dtg} = t_{DTS}$

DTG[7:5] = 10x => DT = (64 + DTG[5:0]) x t_{dtg}，其中 $t_{dtg} = 2 \times t_{DTS}$

DTG[7:5] = 110 => DT = (32 + DTG[4:0]) x t_{dtg}，其中 $t_{dtg} = 8 \times t_{DTS}$

DTG[7:5] = 111 => DT = (32 + DTG[4:0]) x t_{dtg}，其中 $t_{dtg} = 16 \times t_{DTS}$

示例：如果 t_{DTS} = 125 ns (8 MHz)，则可能的死区值为：

0 到 15875 ns（步长为 125 ns），

16 µs 到 31750 ns（步长为 250 ns），

32 µs 到 63 µs（步长为 1 µs），

64 µs 到 126 µs（步长为 2 µs）

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，该位字段即无法修改。

41.6.15 TIMx DMA 控制寄存器 (TIMx_DCR)（x = 16 到 17）

TIMx DMA control register

偏移地址：0x48

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 15:13 保留，必须保持复位值。

位 12:8 **DBL[4:0]**：DMA 分组传送长度 (DMA burst length)

该 5 位字段向量定义了 DMA 的传送长度（当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次分组传送），即一个 DMA 要分组传送的次数。可按半字或字节进行传送（请参见下面的示例）。

00000：1 次传输，
00001：2 次传输，
00010：3 次传输，
...
10001：18 次传输。

位 7:5 保留，必须保持复位值。

位 4:0 **DBA[4:0]**：DMA 基址 (DMA base address)

该 5 位字段定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：
00000：TIMx_CR1，
00001：TIMx_CR2，
00010：TIMx_SMCR，
...

示例：考虑以下传输：DBL = 7 次传输，DBA = TIMx_CR1。这种情况下将向/从自 TIMx_CR1 地址开始的 7 个寄存器传输数据。

41.6.16 TIMx 全传输 DMA 地址 (TIMx_DMAR)（x = 16 到 17）

TIMx DMA address for full transfer

偏移地址：0x4C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **DMAB[15:0]**：DMA 分组传送寄存器 (DMA register for burst accesses)

对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器

(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。

41.6.17 TIM16 复用功能寄存器 1 (TIM16_AF1)

TIM16 alternate function register 1

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	BKINP	BKDF1 BK1E	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINE
						rw	rw								rw

位 31:10 保留，必须保持复位值。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)

该位选择 BKIN 复用功能输入有效电平，必须与 BKP 极性位一起编程。

0：BKIN 输入为低电平有效

1：BKIN 输入为高电平有效

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 8 **BKDF1BK1E**：BRK dfsdm1_break[1] 使能 (BRK dfsdm1_break[1] enable)

该位使能定时器 BRK 输入的 dfsdm1_break[1]。dfsdm1_break[1] 输出与其他 BRK 源进行“或”运算。

0：禁止 dfsdm1_break[1] 输入

1：使能 dfsdm1_break[1] 输入

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 7:1 保留，必须保持复位值。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)

该位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。

0：禁止 BKIN 输入

1：使能 BKIN 输入

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

41.6.18 TIM16 输入选择寄存器 (TIM16_TISEL)

TIM16 input selection register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)

0000：TIM16_CH1 输入
0001：LSI
0010：LSE
0011：WKUP_IT
其他：保留

41.6.19 TIM17 复用功能寄存器 1 (TIM17_AF1)

TIM17 alternate function register 1

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	BKINP	BKDF1 BK2E	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINE
						rw	rw								rw

位 31:10 保留，必须保持复位值。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)

该位选择 BKIN 复用功能输入有效电平，必须与 BKP 极性位一起编程。
0：BKIN 输入为低电平有效
1：BKIN 输入为高电平有效

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 8 **BKDF1BK2E**：BRK dfsdm1_break[2] 使能 (BRK dfsdm1_break[2] enable)
该位使能定时器 BRK 输入的 dfsdm1_break[2]。dfsdm1_break[2] 输出与其他 BRK 源进行“或”运算。
0：禁止 dfsdm1_break[2] 输入
1：使能 dfsdm1_break[2] 输入
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 7:1 保留，必须保持复位值。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
该位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

41.6.20 TIM17 输入选择寄存器 (TIM17_TISEL)

TIM17 input selection register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)
0000：TIM17_CH1 输入
0001：SPDIFRX_FRAME_SYNC 连到 TIM17_CH1，以测量所接收 SPDIF 帧的时钟漂移。
0010：HSE_RTC
0011：MCO1
其他：保留

41.6.21 TIM16/TIM17 寄存器映射

TIM16/TIM17 寄存器 可映射为 16 位可寻址寄存器，如下表所述：

表 368. TIM16/TIM17 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIMx_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0				0	0	0	0
0x04	TIMx_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1					CCDS	CCUS	Res.	CCPC
	Reset value																							0	0					0	0		0
0x0C	TIMx_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1DE	UDE	BIE	Res.	COMIE	Res.	Res.	Res.	CC1IE	UIE
	Reset value																							0	0	0		0				0	0
0x10	TIMx_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	BIF	Res.	COMIF	Res.	Res.	Res.	CC1IF	UIF
	Reset value																							0		0		0				0	0
0x14	TIMx_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	Res.	COMG	Res.	Res.	Res.	CC1G	UG
	Reset value																									0		0				0	0
0x18	TIMx_CCMR1 Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value																0										0	0	0	0	0	0	0
	TIMx_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1PSC [1:0]		CC1S [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0x20	TIMx_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E
	Reset value																													0	0	0	0
0x24	TIMx_CNT	UIFCPY or Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	TIMx_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	TIMx_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

表 368. TIM16/TIM17 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x30	**TIMx_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x34	**TIMx_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIMx_BDTR**	Res.	Res.	Res.	BKBID	Res.	BKDSRM	Res.	Res.	Res.	Res.	Res.	Res.	BKF[3:0]				MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DTG[7:0]							
	Reset value				0		0							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**TIMx_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIMx_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**TIM16_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINP	BKDF1BK1E	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINE
	Reset value																							0	0								1
0x60	**TIM17_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINP	BKDF1BK2E	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINE
	Reset value																							0	0								1
0x68	**TIM16_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value																													0	0	0	0
0x68	**TIM17_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

42 基本定时器 (TIM6/TIM7)

42.1 TIM6/TIM7 简介

基本定时器 TIM6 和 TIM7 包含一个 16 位自动重装载计数器，该计数器由可编程预分频器驱动。

基本定时器可用作通用定时器以生成时基。

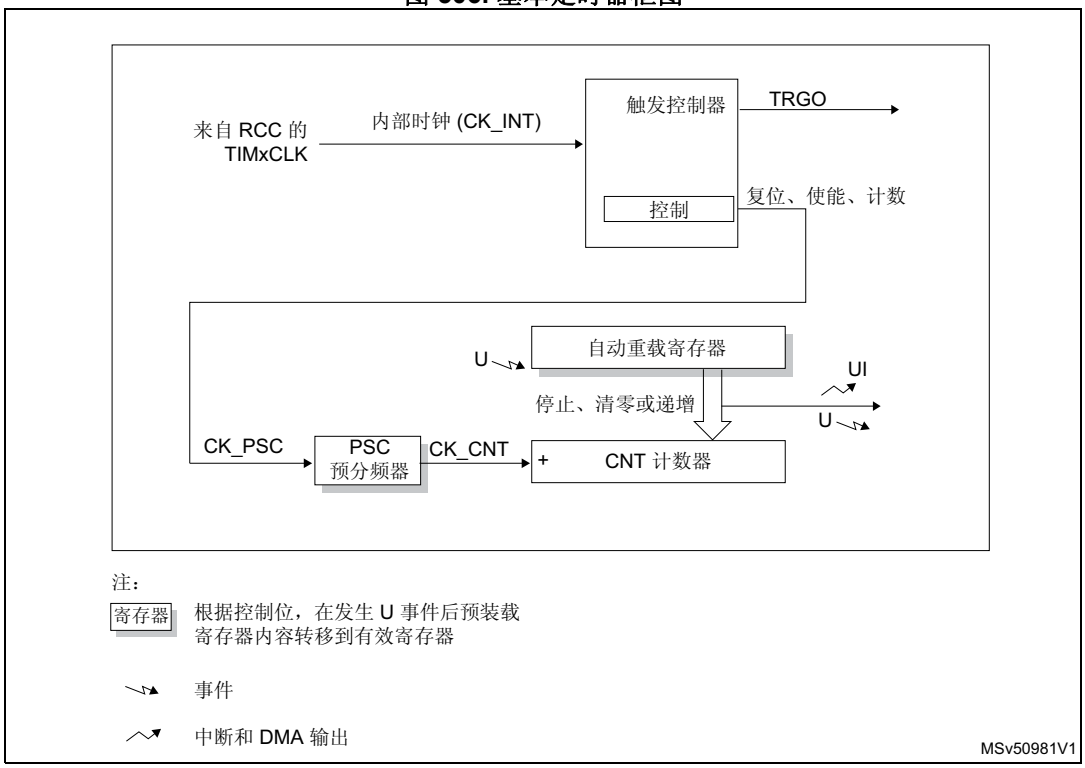
这些定时器彼此完全独立，不共享任何资源。

42.2 TIM6/TIM7 主要特性

基本定时器 (TIM6/TIM7) 的特性包括：

- 16 位自动重装载递增计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65535 之间
- 发生如下更新事件时会生成中断/DMA 请求：计数器上溢

图 506. 基本定时器框图



42.3 TIM6/TIM7 功能描述

42.3.1 时基单元

可编程定时器的主要模块由一个 16 位递增计数器及其相关的自动重载寄存器组成。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。每次尝试对自动重载寄存器执行读写操作时，都会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 UEV 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重装载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将详细介绍各配置的更新事件的产生过程。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器。

请注意，实际的计数器使能信号 CNT_EN 在 CEN 置 1 的一个时钟周期后被置 1。

预分频器说明

预分频器可对计数器时钟进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 TIMx_PSC 寄存器中的 16 位寄存器所控制的 16 位计数器。由于 TIMx_PSC 控制寄存器有缓冲，因此可对预分频器进行实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 507 和*图 508* 以一些示例说明在预分频比实时变化时计数器的行为。

图 507. 预分频器分频由 1 变为 2 时的计数器时序图

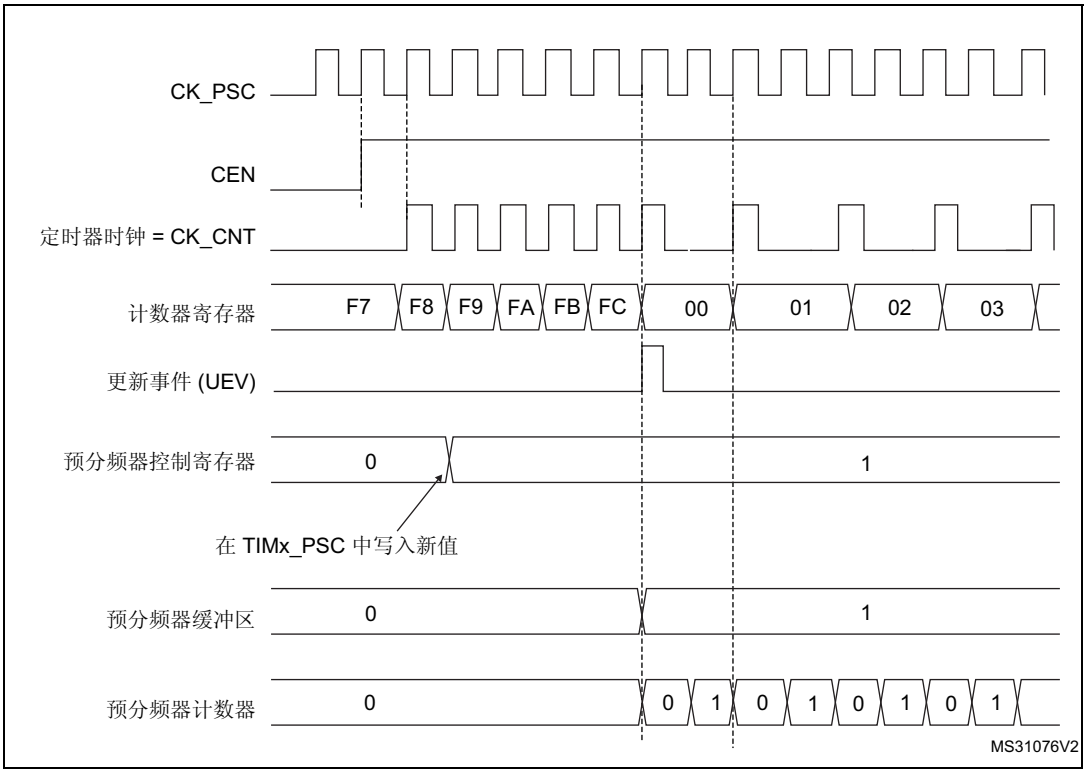
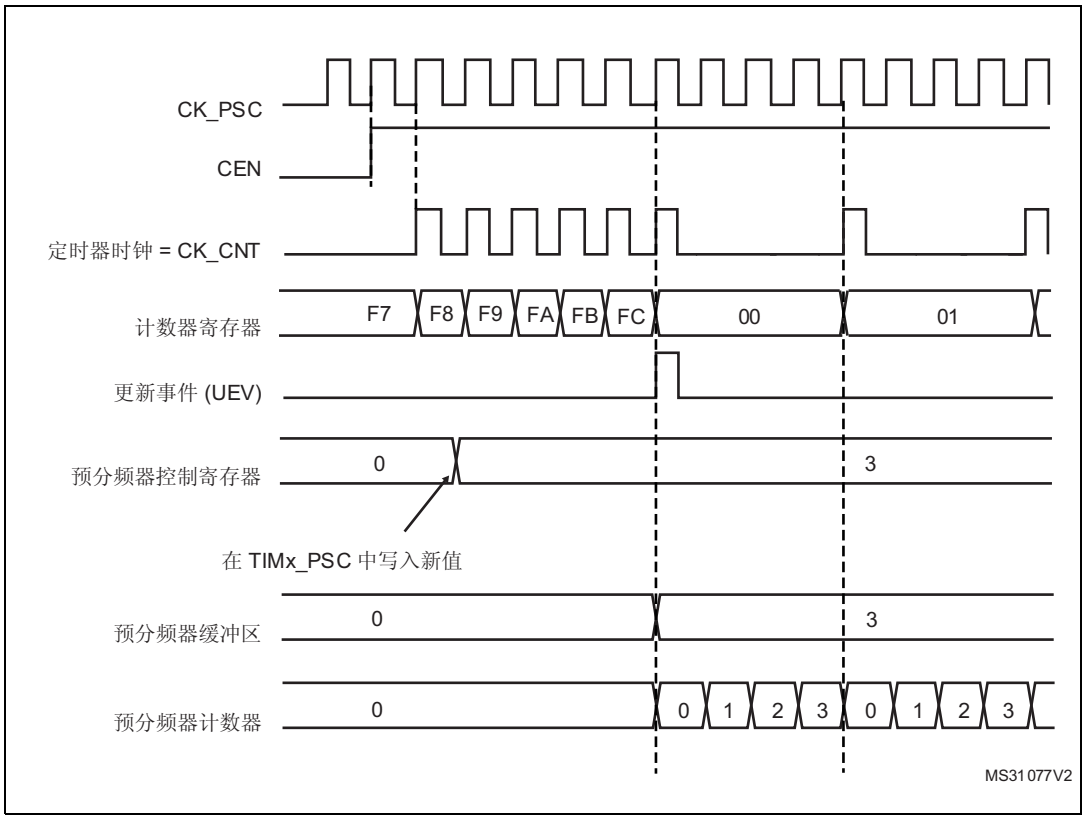


图 508. 预分频器分频由 1 变为 4 时的计数器时序图



42.3.2 计数模式

计数器从 0 计数到自动重载值（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

每次发生计数器上溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以产生更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。这样，直到 UDIS 位清零前便不会生成任何更新事件，但计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）
- 使用预装载值 (TIMx_ARR) 更新自动重载影子寄存器

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 509. 计数器时序图，1 分频内部时钟

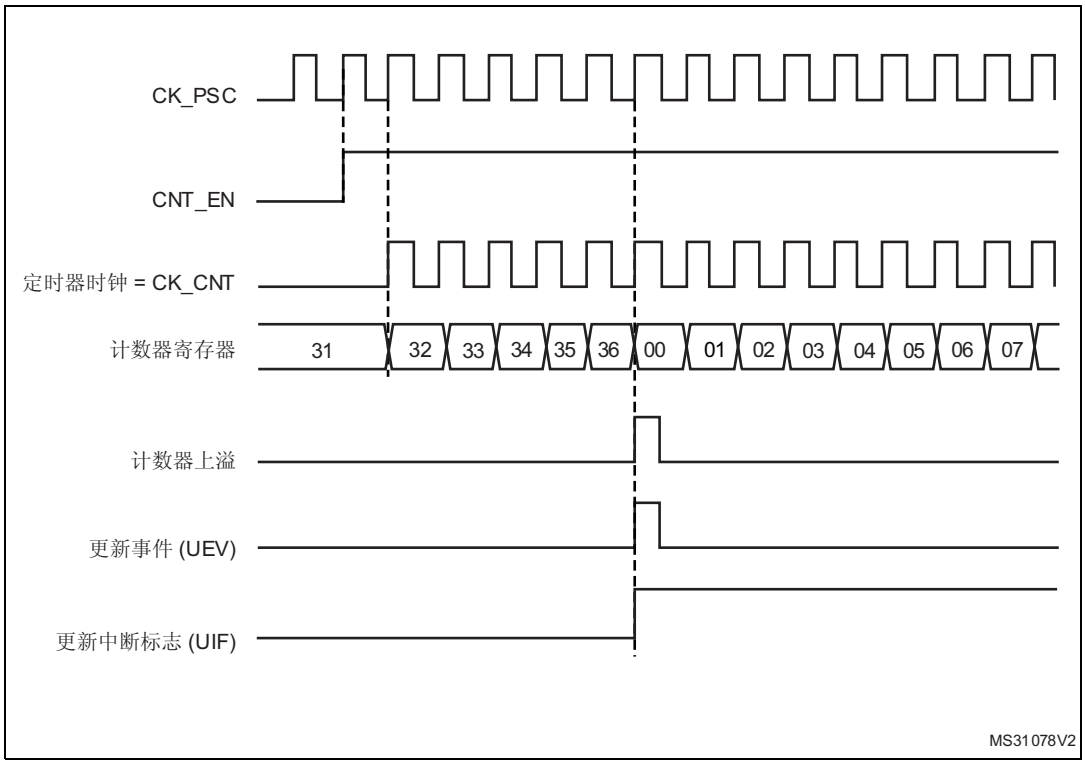


图 510. 计数器时序图，2 分频内部时钟

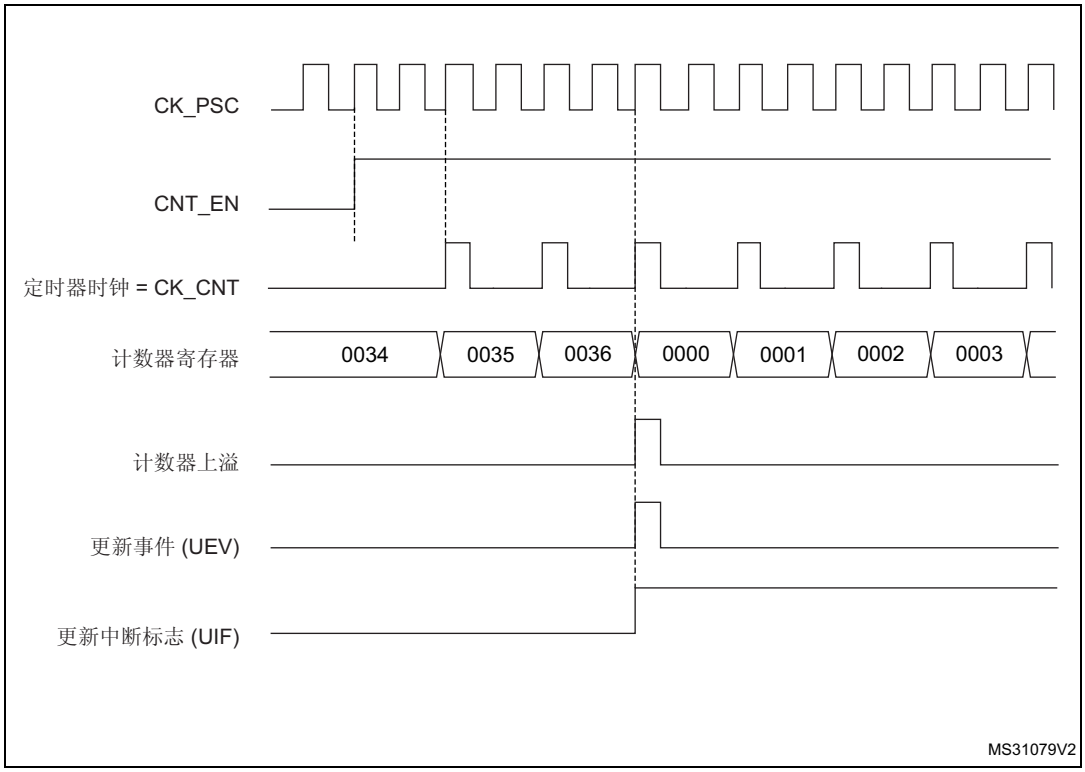


图 511. 计数器时序图，4 分频内部时钟

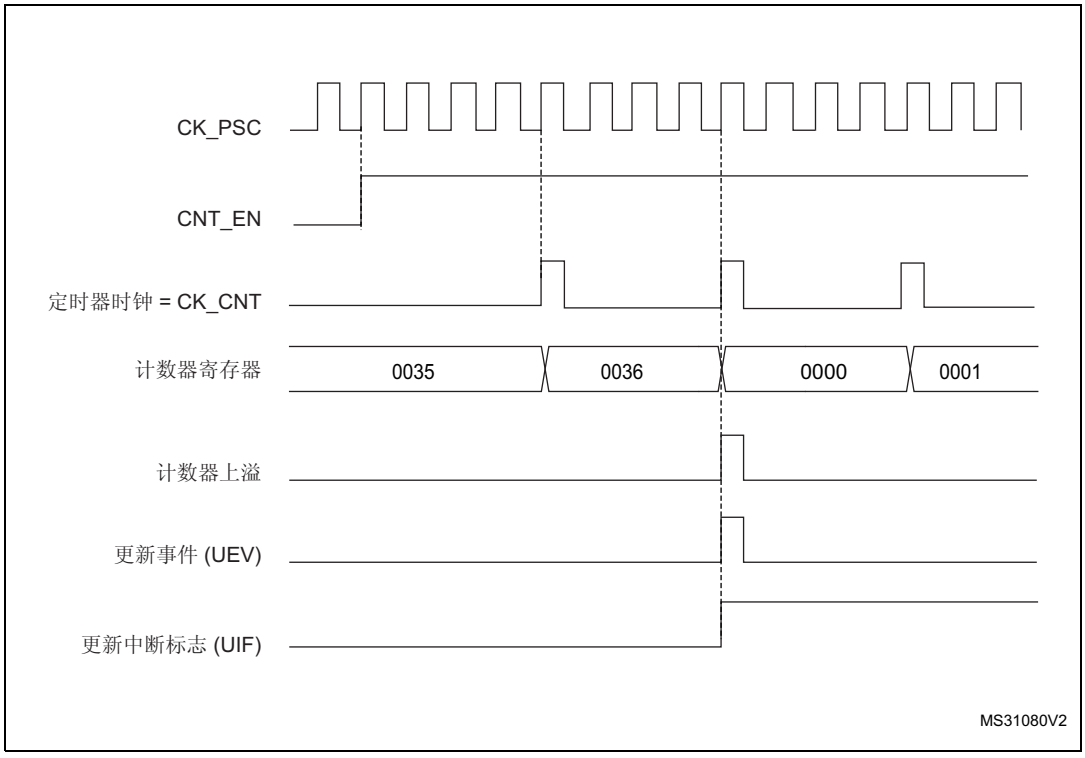


图 512. 计数器时序图，N 分频内部时钟

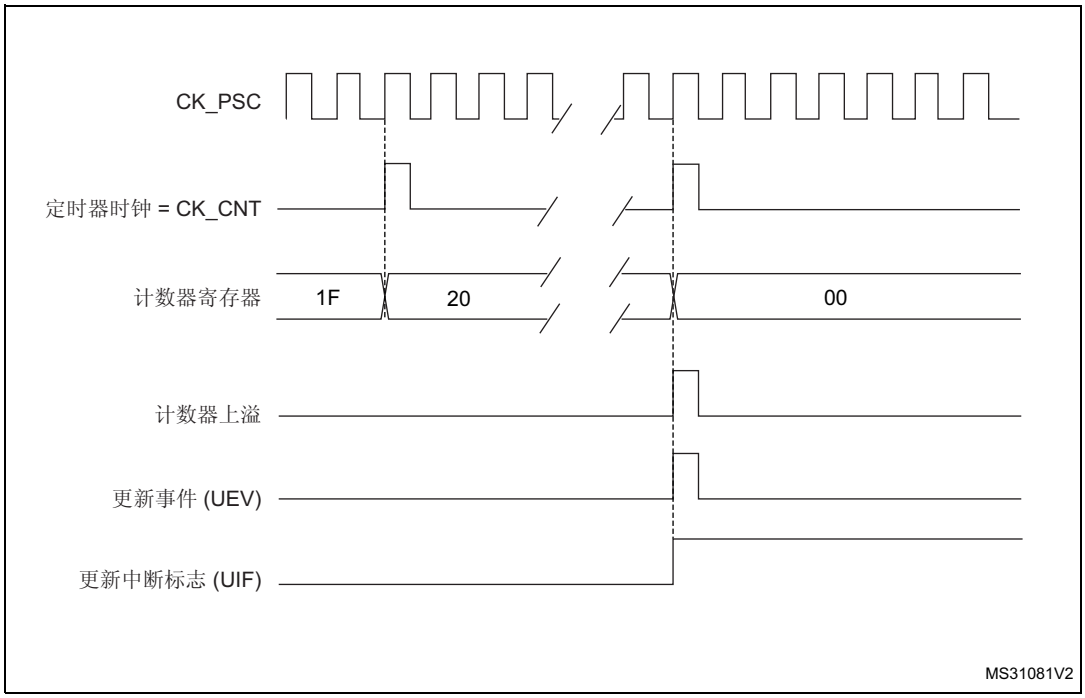


图 513. 计数器时序图，ARPE = 0 时更新事件（TIMx_ARR 未预装载）

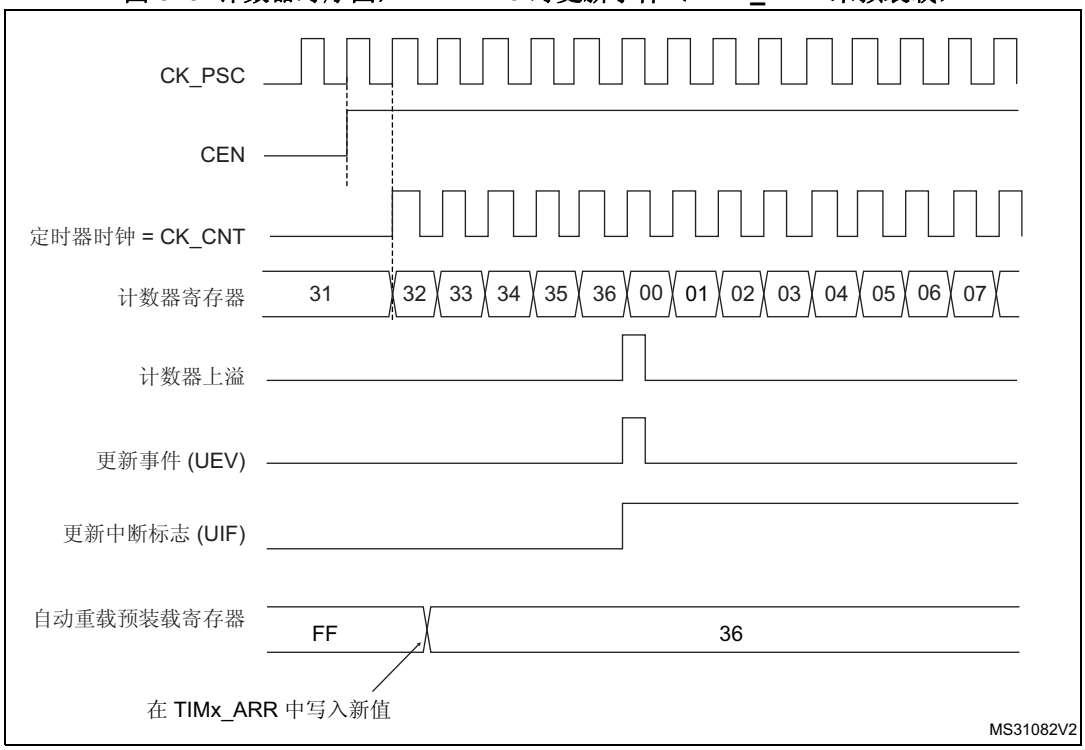
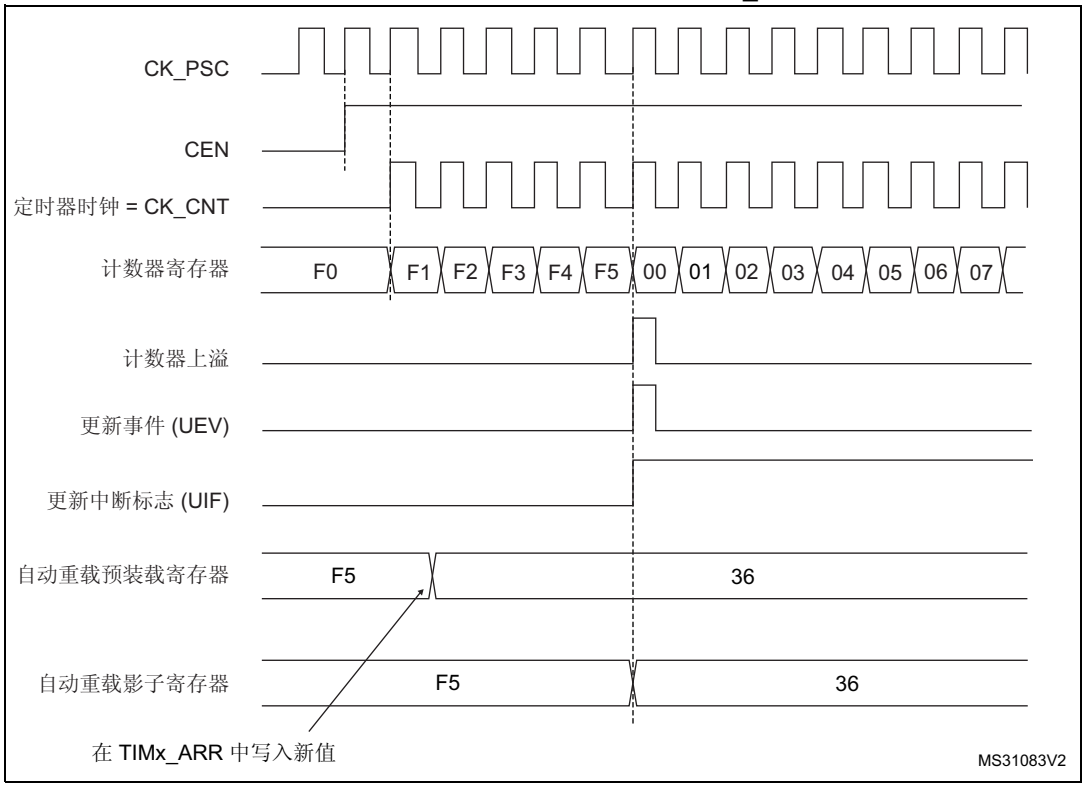


图 514. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载）



42.3.3 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时器计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的更新事件情况。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

UIF 和 UIFCPY 标志之间的置位没有延迟。

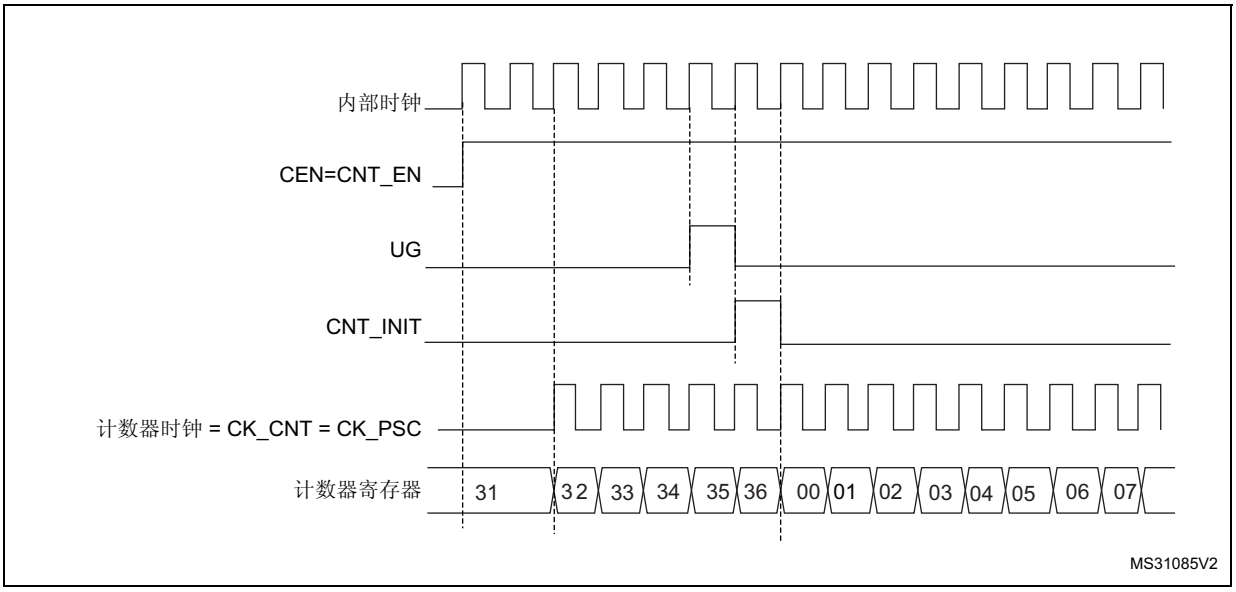
42.3.4 时钟源

计数器时钟由内部时钟 (CK_INT) 源提供。

CEN 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（保持自动清零的 UG 除外）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 515 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

图 515. 正常模式下的控制电路，1 分频内部时钟



42.3.5 调试模式

当系统进入调试模式（处理器内核停止）时，TIMx 计数器会根据 DBGMCU 模块中的 TIMx 位选择继续正常工作或者停止工作。更多详细信息，请参见*第 59.10.6 节：微处理器调试单元 (DBGMCU)*。

42.4 TIM6/TIM7 寄存器

有关寄存器说明中使用的缩写，请参见*第 109 页的第 1.2 节*。

外设寄存器可按半字（16 位）或字（32 位）访问。

42.4.1 TIMx 控制寄存器 1 (TIMx_CR1)（x = 6 到 7）

TIMx control register 1

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIFREMAP	Res.	Res.	Res.	ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
				rw				rw				rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10:8 保留，必须保持复位值。

位 7 **ARPE**：自动重装载预装载使能 (Auto-reload preload enable)

0：TIMx_ARR 寄存器不进行缓冲。

1：TIMx_ARR 寄存器进行缓冲。

位 6:4 保留，必须保持复位值。

位 3 **OPM**：单脉冲模式 (One-pulse mode)

0：计数器在发生更新事件时不会停止计数

1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）。

位 2 **URS**：更新请求源 (Update request source)

该位由软件置 1 和清零，用以选择 UEV 事件源。

0：使能时，所有以下事件都会生成更新中断或 DMA 请求。此类事件包括：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。

位 1 **UDIS**：更新禁止 (Update disable)

该位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

带缓冲的寄存器被加载为预加载数值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR 和 PSC）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注意：只有事先通过软件将 CEN 位置 1，才可以使用门控模式。而触发模式可通过硬件自动将 CEN 位置 1。

在单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。

42.4.2 TIMx 控制寄存器 2 (TIMx_CR2)（x = 6 到 7）

TIMx control register 2

偏移地址：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS[2:0]			Res.	Res.	Res.	Res.
									rw	rw	rw				

位 15:7 保留，必须保持复位值。

位 6:4 **MMS[2:0]**：主模式选择 (Master mode selection)

这些位用于选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下：

000：**复位**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入产生（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。

001：**使能**——计数器使能信号 (CNT_EN) 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。

当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中对 MSM 位的说明）。

010：**更新**——选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器。

注意：必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

位 3:0 保留，必须保持复位值。

42.4.3 TIMx DMA/中断使能寄存器 (TIMx_DIER)（x = 6 到 7）

TIMx DMA/Interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIE
							rw								rw

位 15:9 保留，必须保持复位值。

位 8 **UDE**：更新 DMA 请求使能 (Update DMA request enable)

0：禁止更新 DMA 请求。
1：使能更新 DMA 请求。

位 7:1 保留，必须保持复位值。

位 0 **UIE**：更新中断使能 (Update interrupt enable)

0：禁止更新中断。
1：使能更新中断。

42.4.4 TIMx 状态寄存器 (TIMx_SR)（x = 6 到 7）

TIMx status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIF
															rc_w0

位 15:1 保留，必须保持复位值。

位 0 **UIF**：更新中断标志 (Update interrupt flag)
该位在发生更新事件时通过硬件置 1。但需要通过软件清零。
0：未发生更新。
1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：
– 重复计数器值上溢或下溢并且 TIMx_CR1 寄存器中的 UDIS=0 时。
– 当由于 TIMx_CR1 寄存器中 URS = 0 且 UDIS = 0 而通过软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

42.4.5 TIMx 事件生成寄存器 (TIMx_EGR)（x = 6 到 7）

TIMx event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UG
															w

位 15:1 保留，必须保持复位值。

位 0 **UG**：更新生成 (Update generation)
该位可通过软件置 1，并由硬件自动清零。
0：不执行任何操作。
1：重新初始化定时器并产生更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。

42.4.6 TIMx 计数器 (TIMx_CNT)（x = 6 到 7）

TIMx counter

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIF CPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **UIFCPY**：UIF 副本 (UIF Copy)
该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。如果 TIMx_CR1 中的 UIFREMAP 位复位，则位 31 保留，读为 0。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

42.4.7 TIMx 预分频器 (TIMx_PSC)（x = 6 到 7）

TIMx prescaler

偏移地址：0x28

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)
计数器时钟频率 CK_CNT 等于 f_{CK_PSC} / (PSC[15:0] + 1)。
PSC 存放着每次发生更新事件时要加载到有效预分频器寄存器的值。
（包括当计数器通过 TIMx_EGR 寄存器的 UG 位或在“复位模式”下通过触发控制器清零时）。

42.4.8 TIMx 自动重载寄存器 (TIMx_ARR)（x = 6 到 7）

TIMx auto-reload register

偏移地址：0x2C

复位值：0xFFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **ARR[15:0]**：预分频器值 (Prescaler value)
ARR 为要装载到有效自动重载寄存器的值。
有关 ARR 更新和行为的详细信息，请参见*第 2192 页的第 42.3.1 节：时基单元*。
当自动重载值为空时，计数器不工作。

42.4.9 TIMx 寄存器映射

TIMx 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 369. TIMx 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIMx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	Res.	Res.	ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																					0				0				0	0	0	0
0x04	**TIMx_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS [2:0]			Res.	Res.	Res.	Res.
	Reset value																										0	0	0				
0x08	Reserved																																
0x0C	**TIMx_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIE
	Reset value																								0								0
0x10	**TIMx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIF
	Reset value																																0
0x14	**TIMx_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UG
	Reset value																																0
0x18-0x20	Reserved																																
0x24	**TIMx_CNT**	UIFCPY or Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIMx_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIMx_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

43 低功耗定时器 (LPTIM)

43.1 简介

LPTIM 是一个 16 位定时器，可从降低功耗的最终发展中受益。由于 LPTIM 的时钟源具有多样性，因此 LPTIM 能够在所有电源模式（待机模式除外）下保持运行状态。即使没有内部时钟源，LPTIM 也能运行，鉴于这一点，可将其用作“脉冲计数器”，这种脉冲计数器在某些应用中十分有用。此外，LPTIM 还能将系统从低功耗模式唤醒，因此非常适合实现“超时功能”，而且功耗极低。

LPTIM 引入了一个灵活的时钟方案，该方案能够提供所需的功能和性能，同时还能最大程度地降低功耗。

43.2 LPTIM 主要特性

- 16 位递增计数器
- 3 位预分频器，可采用 8 种分频系数（1、2、4、8、16、32、64 和 128）
- 可选时钟
  - 内部时钟源：可配置的内部时钟源（请参见 RCC 部分）
  - LPTIM 输入的外部时钟源（在没有内置振荡器运行的情况下工作，由脉冲计数器应用使用）
- 16 位 ARR 自动重载寄存器
- 16 位比较寄存器
- 连续/单触发模式
- 可选软件/硬件输入触发
- 可编程数字防抖动干扰滤波器
- 可配置输出：脉冲、PWM
- 可配置 I/O 极性
- 编码器模式

43.3 LPTIM 实现

表 370 介绍了 STM32MP13xx 设备上的 LPTIM 实现。LPTIM1 和 LPTIM2 支持所有特性。LPTIM3、LPTIM4 和 LPTIM5 支持一部分特性。

表 370. STM32MP13xx LPTIM 特性

LPTIM 特性	LPTIM1	LPTIM2	LPTIM3	LPTIM4	LPTIM5
编码器模式	X	X	-	-	-
外部输入时钟	X	X	X	-	-
从 Stop + LP-Stop 或 LPLV-Stop 唤醒（内核时钟仅来自 LSE 或 LSI）	X	X	X	X	X

43.4 LPTIM 功能描述

43.4.1 LPTIM 框图

图 516. 低功耗定时器框图（LPTIM1 和 LPTIM2）

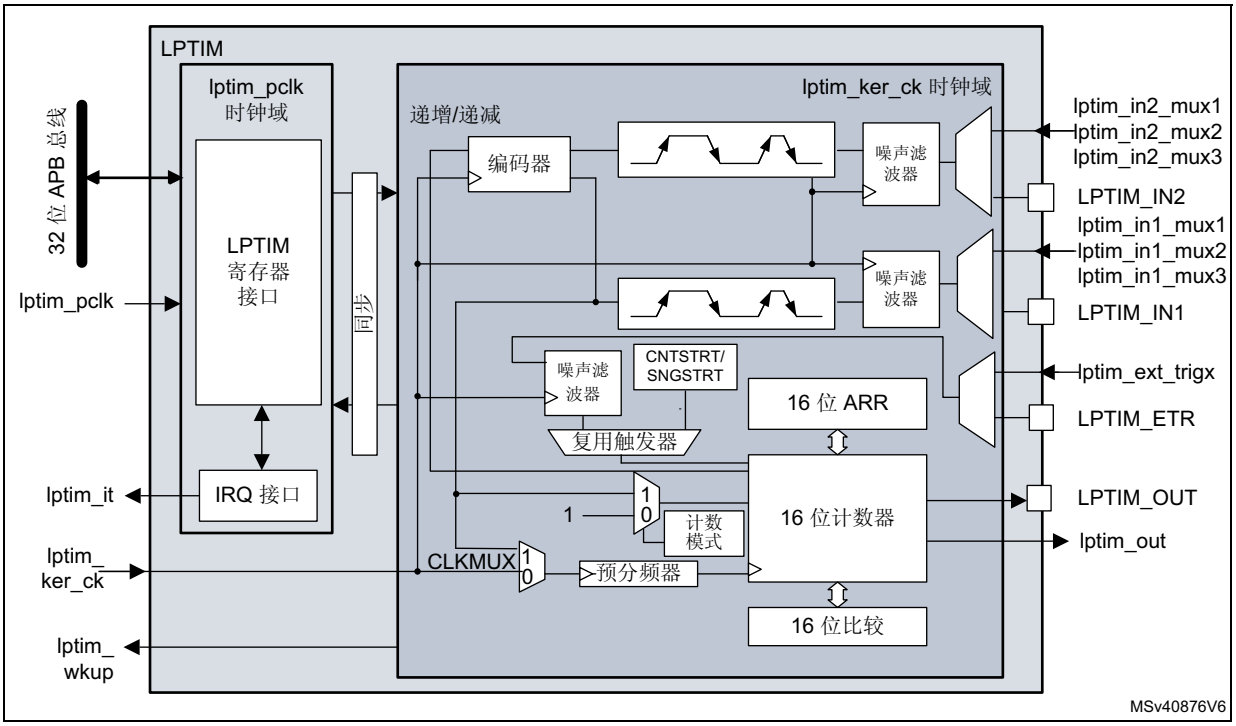


图 517. 低功耗定时器框图 (LPTIM3)

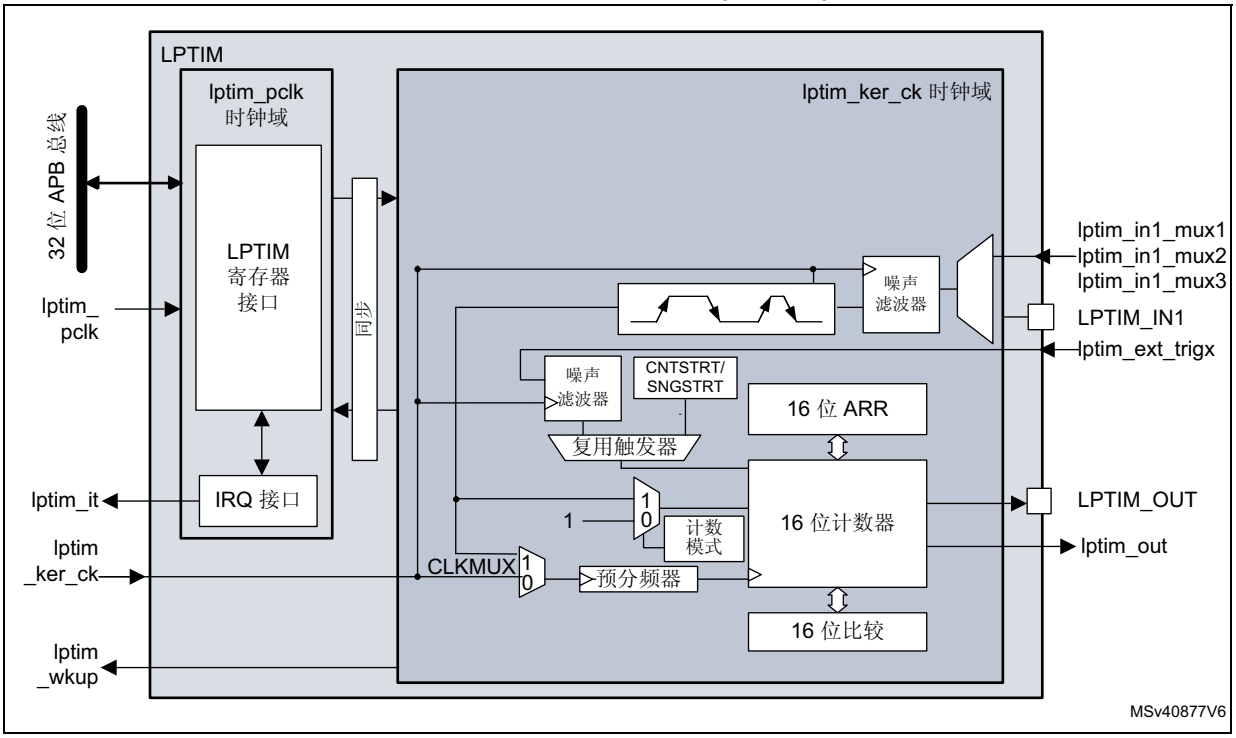
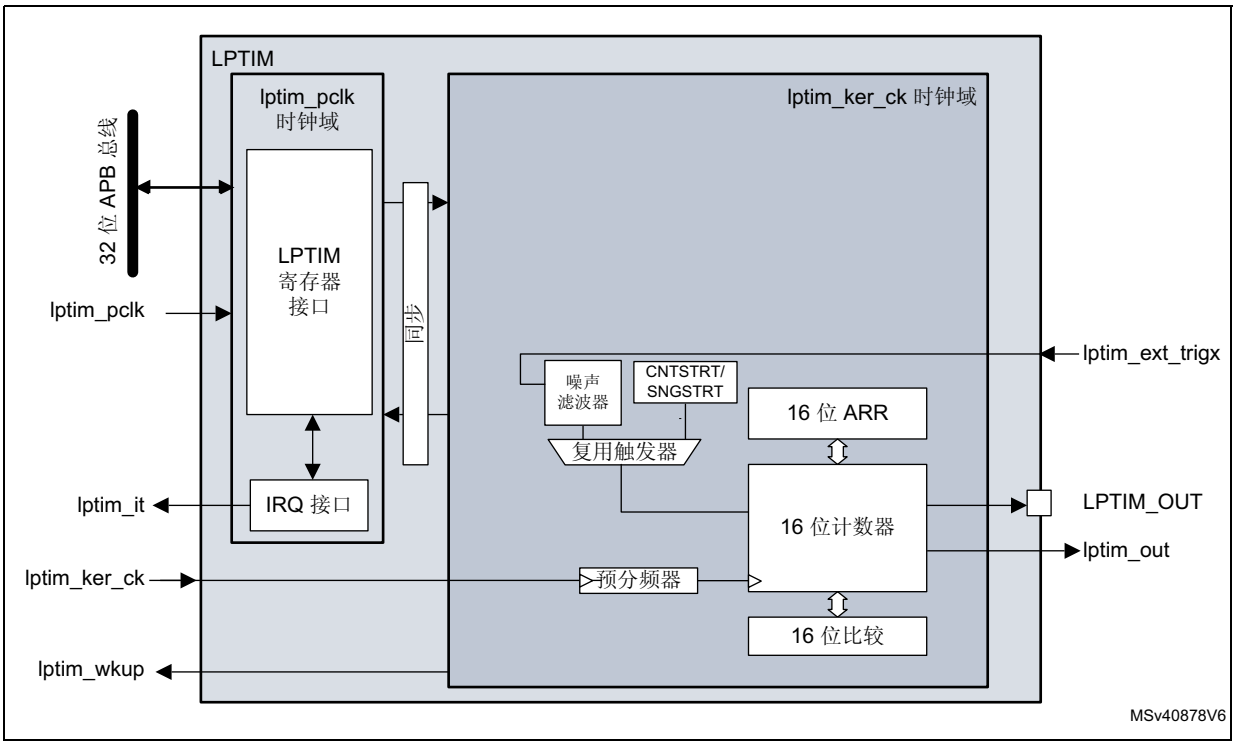


图 518. 低功耗定时器框图（LPTIM4 和 LPTIM5）



43.4.2 LPTIM 引脚和内部信号

以下各表分别列出了 LPTIM 引脚和内部信号。

表 371. LPTIM 输入/输出引脚

名称	信号类型	说明
LPTIM_IN1	数字输入	复用器输入 0 上 GPIO 引脚的 LPTIM 输入 1
LPTIM_IN2	数字输入	复用器输入 0 上 GPIO 引脚的 LPTIM 输入 2
LPTIM_ETR	数字输入	LPTIM 外部触发 GPIO 引脚
LPTIM_OUT	数字输出	LPTIM 输出 GPIO 引脚

表 372. LPTIM 内部信号

名称	信号类型	说明
lptim_pclk	数字输入	LPTIM APB 时钟域
lptim_ker_ck	数字输入	LPTIM 内核时钟
lptim_in1_mux1	数字输入	连接至复用器输入 1 的内部 LPTIM 输入 1
lptim_in1_mux2	数字输入	连接至复用器输入 2 的内部 LPTIM 输入 1
lptim_in1_mux3	数字输入	连接至复用器输入 3 的内部 LPTIM 输入 1
lptim_in2_mux1	数字输入	连接至复用器输入 1 的内部 LPTIM 输入 2
lptim_in2_mux2	数字输入	连接至复用器输入 2 的内部 LPTIM 输入 2
lptim_in2_mux3	数字输入	连接至复用器输入 3 的内部 LPTIM 输入 2
lptim_ext_trigx	数字输入	LPTIM 外部触发输入 x
lptim_out	数字输出	LPTIM 计数器输出
lptim_it	数字输出	LPTIM 全局中断
lptim_wakeup	数字输出	LPTIM 唤醒事件

43.4.3 LPTIM 输入和触发映射

下面详细给出了 LPTIM 外部触发连接和输入连接。

表 373. LPTIM1 外部触发连接

TRIGSEL	外部触发信号
lptim_ext_trig0	用作 LPTIM1_ETR 复用功能的 GPIO 引脚
lptim_ext_trig1	RTC_ALARMA
lptim_ext_trig2	RTC_ALARMB
lptim_ext_trig3	RTC_TAMP1_OUT
lptim_ext_trig4	RTC_TAMP2_OUT
lptim_ext_trig5	RTC_TAMP3_OUT

表 373. LPTIM1 外部触发连接（续）

TRIGSEL	外部触发信号
lptim_ext_trig6	未连接
lptim_ext_trig7	未连接

表 374. LPTIM2 外部触发连接

TRIGSEL	外部触发信号
lptim_ext_trig0	用作 LPTIM2_ETR 复用功能的 GPIO 引脚
lptim_ext_trig1	RTC_ALARMA
lptim_ext_trig2	RTC_ALARMB
lptim_ext_trig3	RTC_TAMP1_OUT
lptim_ext_trig4	RTC_TAMP2_OUT
lptim_ext_trig5	RTC_TAMP3_OUT
lptim_ext_trig6	未连接
lptim_ext_trig7	未连接

表 375. LPTIM3 外部触发连接

TRIGSEL	外部触发信号
lptim_ext_trig0	LPTIM2_OUT
lptim_ext_trig1	hsi_cal_ck
lptim_ext_trig2	LPTIM4_OUT
lptim_ext_trig3	LPTIM5_OUT
lptim_ext_trig4	SAI1_FS_A
lptim_ext_trig5	SAI1_FS_B
lptim_ext_trig6	未连接
lptim_ext_trig7	未连接

表 376. LPTIM4 外部触发连接

TRIGSEL	外部触发信号
lptim_ext_trig0	LPTIM2_OUT
lptim_ext_trig1	LPTIM3_OUT
lptim_ext_trig2	未连接
lptim_ext_trig3	LPTIM5_OUT
lptim_ext_trig4	SAI2_FS_A
lptim_ext_trig5	SAI2_FS_B

表 376. LPTIM4 外部触发连接（续）

TRIGSEL	外部触发信号
lptim_ext_trig6	未连接
lptim_ext_trig7	未连接

表 377. LPTIM5 外部触发连接

TRIGSEL	外部触发信号
lptim_ext_trig0	LPTIM2_OUT
lptim_ext_trig1	LPTIM3_OUT
lptim_ext_trig2	LPTIM4_OUT
lptim_ext_trig3	未连接
lptim_ext_trig4	未连接
lptim_ext_trig5	未连接
lptim_ext_trig6	未连接
lptim_ext_trig7	未连接

表 378. LPTIM1 输入 1 连接

lptim_in1_mux	LPTIM1 输入 1 连接位置
lptim_in1_mux0	用作 LPTIM1_IN1 复用功能的 GPIO 引脚
lptim_in1_mux1	未连接
lptim_in1_mux2	未连接
lptim_in1_mux3	未连接

表 379. LPTIM1 输入 2 连接

lptim_in2_mux	LPTIM1 输入 2 连接位置
lptim_int2_mux0	用作 LPTIM1_IN2 复用功能的 GPIO 引脚
lptim_in2_mux1	未连接
lptim_in2_mux2	未连接
lptim_in2_mux3	未连接

表 380. LPTIM2 输入 1 连接

lptim_in1_mux	LPTIM2 输入 1 连接位置
lptim_in1_mux0	用作 LPTIM2_IN1 复用功能的 GPIO 引脚
lptim_in1_mux1	未连接
lptim_in1_mux2	未连接
lptim_in1_mux3	未连接

表 381. LPTIM2 输入 2 连接

lptim_in2_mux	LPTIM2 输入 2 连接位置
lptim_int2_mux0	用作 LPTIM2_IN2 复用功能的 GPIO 引脚
lptim_in2_mux1	未连接
lptim_in2_mux2	未连接
lptim_in2_mux3	未连接

表 382. LPTIM3 输入 1 连接

lptim_in1_mux	LPTIM3 输入 1 连接位置
lptim_in1_mux0	HSE 时钟
lptim_in1_mux1	未连接
lptim_in1_mux2	未连接
lptim_in1_mux3	未连接

43.4.4 LPTIM 复位和时钟

LPTIM 可通过多个时钟源提供时钟。它可以由内部时钟信号提供时钟，内部时钟信号可通过 RCC 来选择任意可配置的内部时钟源（更多详细信息，请参见 RCC 部分）。此外，LPTIM 还可通过注入到其外部 Input1 上的外部时钟信号提供时钟。当通过外部时钟源提供时钟时，LPTIM 可以在以下两种配置之一中运行：

- 第一种配置是，LPTIM 通过外部信号提供时钟，但同时通过可配置的内部时钟源为 LPTIM 提供内部时钟信号（请参见 RCC 部分）。
- 第二种配置是，LPTIM 仅由外部时钟源通过外部 Input1 提供时钟。此配置可在进入低功耗模式后所有内置振荡器关闭时，用于实现超时功能或脉冲计数器功能。

对 CKSEL 和 COUNTMODE 位进行编程，可控制 LPTIM 使用外部时钟源还是内部时钟源。

当使用外部时钟源时，可使用 CKPOL 位选择外部时钟信号的有效边沿。如果上升沿和下降沿均为有效边沿，则还应提供内部时钟信号（第一种配置）。在这种情况下，内部时钟信号频率应至少为外部时钟信号频率的五倍。

43.4.5 干扰滤波器

外部（映射到 GPIO）或内部（映射到芯片级或其他嵌入式外设）LPTIM 输入由数字滤波器保护，避免任何毛刺和噪声干扰在 LPTIM 内部传播，从而防止产生意外计数或触发。

在激活数字滤波器之前，应先向 LPTIM 提供内部时钟源。这是保证滤波器正常工作的必要条件。

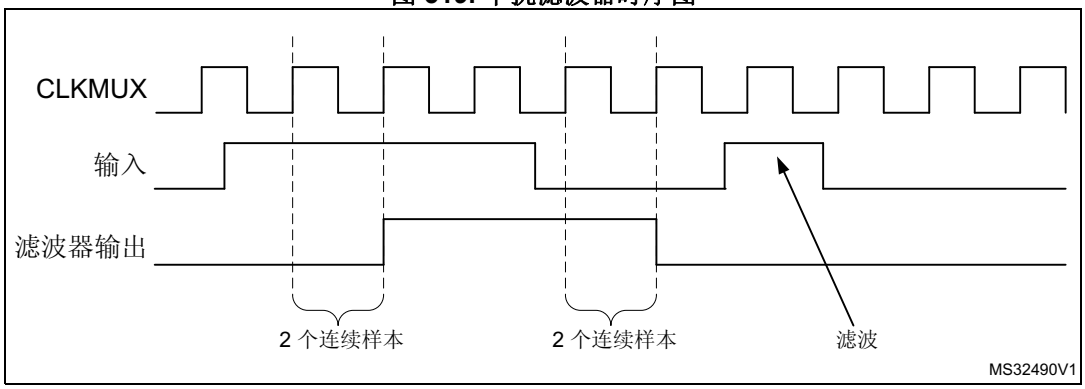
数字滤波器分为两组：

- 第一组数字滤波器保护 LPTIM 外部输入。数字滤波器的敏感性由 CKFLT 位控制。
- 第二组数字滤波器保护 LPTIM 内部触发输入。数字滤波器的敏感性由 TRGFLT 位控制。

注： *数字滤波器的敏感性以组为单位进行控制。无法单独配置同一组内各个数字滤波器的敏感性。*

滤波器的敏感性会影响相同的连续采样的数量，在其中一个 LPTIM 输入上检测到此类连续采样时，才能将某信号电平变化视为有效跳变。*图 519* 给出了编程 2 个连续采样时，干扰滤波器行为的示例。

图 519. 干扰滤波器时序图



注： *不提供内部时钟信号时，必须通过将 CKFLT 和 TRGFLT 位设为 0 来停用数字滤波器。在这种情况下，可使用外部模拟滤波器来防止 LPTIM 外部输入产生干扰。*

43.4.6 预分频器

LPTIM 16 位计数器前面要有一个可配置的 2 次幂预分频器。预分频器的分频比由 PRESC[2:0] 3 位字段进行控制。下表列出了所有可能的分频比：

表 383. 预分频器的分频比

编程	分频系数
000	/1
001	/2
010	/4
011	/8
100	/16
101	/32
110	/64
111	/128

43.4.7 触发多路复用器

LPTIM 计数器可以通过软件启动，或者在检测到 8 个触发输入之一的有效边沿后启动。

TRIGEN[1:0] 用于确定 LPTIM 触发源：

- TRIGEN[1:0] 等于“00”时，LPTIM 计数器会在通过软件将 CNTSTRT 位或 SNGSTRT 位其中之一置 1 后立即启动。TRIGEN[1:0] 的其余三个可能的值用于配置触发输入使用的有效边沿。LPTIM 计数器会在检测到有效边沿后立即启动。
- TRIGEN[1:0] 不等于“00”时，TRIGSEL[2:0] 用于选择使用 8 个触发输入中的哪一个来启动计数器。

外部触发信号视为 LPTIM 的异步信号。因此，检测到触发信号后，由于同步问题，需要延迟两个计数器时钟周期，定时器才能开始运行。

如果在定时器已启动时发生新的触发事件，则此事件将被忽略（除非已使能超时功能）。

注： *必须使能定时器，才能将 SNGSTRT/CNTSTRT 位置 1。当定时器禁止时，对这些位执行的任何写操作都将被硬件丢弃。*

注： *通过软件启动计数器后 (TRIGEN[1:0] = 00)，从 LPTIM_CR 寄存器更新（SNGSTRT 位或 CNTSTRT 位置 1）到计数器真正启动之间会有 3 个内核时钟周期的延迟。*

43.4.8 工作模式

LPTIM 支持以下两种工作模式：

- 连续模式：定时器自由运行，由触发事件启动并且直到被禁止才会停止
- 单触发模式：定时器由触发事件启动，当达到 ARR 值时停止。

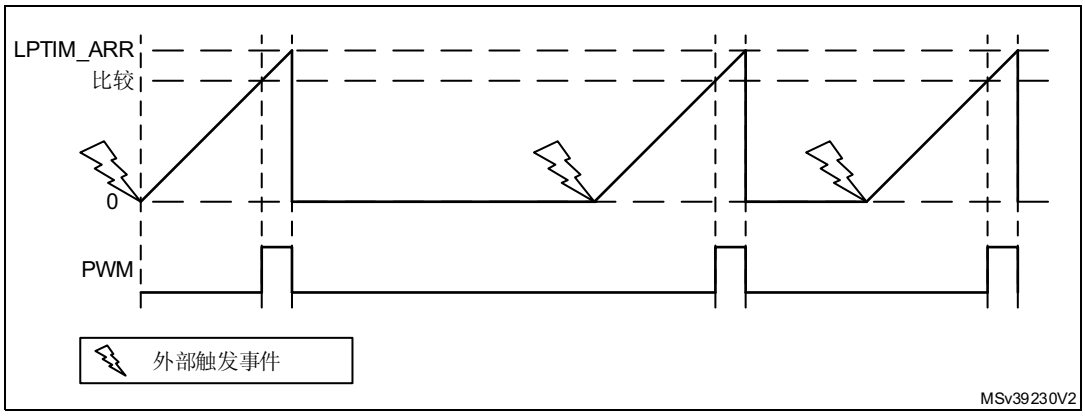
单触发模式

要使能单触发计数，必须将 SNGSTRT 位置 1。

新的触发事件将重新启动定时器。在计数器启动后并在计数器达到 ARR 之前发生的任何触发事件都将被丢弃。

选择外部触发时，在 SNGSTRT 位置 1 后以及计数器寄存器停止后（包含零值）到达的每个外部触发事件都将为计数器启动新的单触发计数周期，如*图 520* 所示。

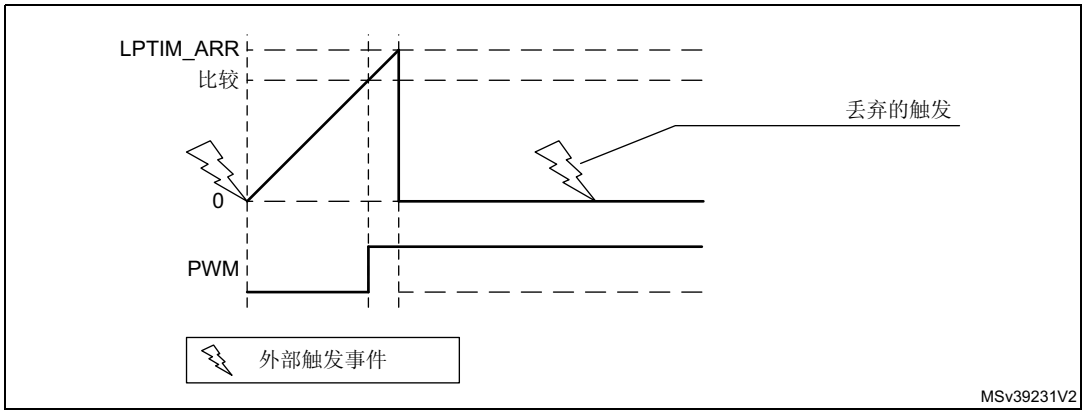
图 520. LPTIM 输出波形，单次计数模式配置



单次设置模式激活：

应注意，当 LPTIM_CFGR 寄存器中的 WAVE 位字段置 1 时，将激活单次设置模式。在这种情况下，计数器仅会在第一个触发事件后启动一次，任何后续触发事件都将被丢弃，如*图 521* 所示。

图 521. LPTIM 输出波形，单次计数模式配置且激活单次设置模式（WAVE 位置 1）



若通过软件启动（TRIGEN[1:0] =“00”），将 SNGSTRT 置 1 会使计数器进行单触发计数。

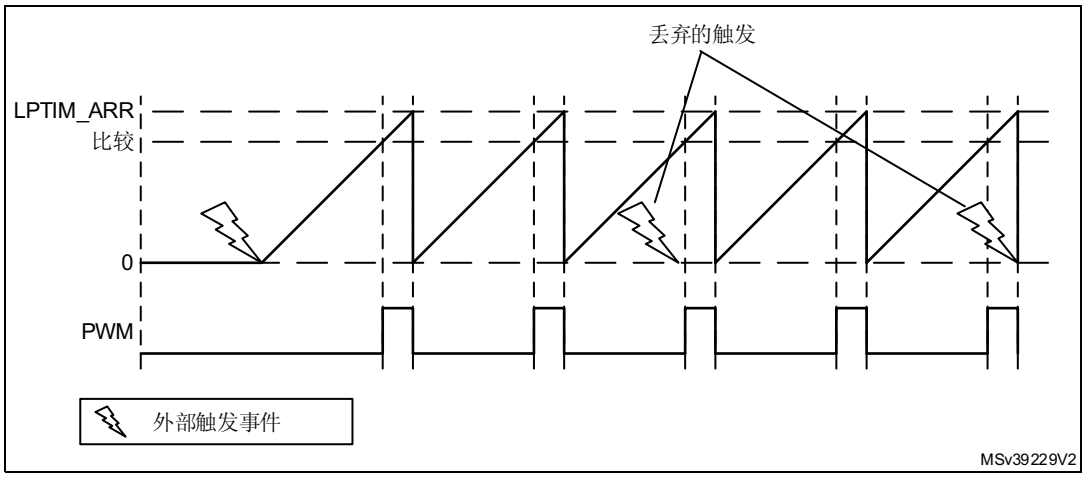
连续模式

要使能连续计数，必须将 CNTSTRT 位置 1。

若选择外部触发，则在 CNTSTRT 置 1 后到达的外部触发事件将启动计数器进行连续计数。任何后续的外部触发事件都将被丢弃，如*图 522* 所示。

若通过软件启动（TRIGEN[1:0] =“00”），将 CNTSTRT 置 1 会使计数器开始连续计数。

图 522. LPTIM 输出波形、连续计数模式配置



SNGSTRT 和 CNTSTRT 位只能在定时器使能时（ENABLE 位置 1）置 1。可以“实时”从单触发模式切换为连续模式。

如果之前选择的是连续模式，则将 SNGSTRT 置 1 会使 LPTIM 切换为单触发模式。计数器（激活时）将在达到 ARR 后立即停止。

如果之前选择的是单触发模式，则将 CNTSTRT 置 1 会使 LPTIM 切换为连续模式。计数器（激活时）将在达到 ARR 后立即重新启动。

43.4.9 超时功能

若在一个选定的触发输入上检测到有效边沿，则可用于复位 LPTIM 计数器。该功能通过 TIMOUT 位进行控制。

第一个触发事件将启动定时器，任何后续的触发事件将复位计数器，且定时器将重新启动。

可实现低功耗超时功能。超时值对应于比较值；如果在预期的时间帧内未发生触发事件，MCU 将由比较匹配事件唤醒。

43.4.10 生成波形

两个 16 位寄存器，LPTIM_ARR（自动重载寄存器）和 LPTIM_CMP（比较寄存器）用于在 LPTIM 输出上生成多个不同的波形

定时器可生成以下波形：

- PWM 模式：只要 LPTIM_CNT 中的计数器值超过 LPTIM_CMP 中的比较值，LPTIM 输出就会置位。只要 LPTIM_ARR 和 LPTIM_CNT 寄存器之间发生匹配，LPTIM 输出就会复位。
- 单脉冲模式：对于第一个脉冲，输出波形与 PWM 模式输出波形类似，随后输出将永久复位
- 单次设置模式：除输出保持最后一个信号电平外（取决于配置的输出极性），输出波形与单脉冲模式输出波形类似。

上述模式要求 LPTIM_ARR 寄存器的值严格大于 LPTIM_CMP 寄存器的值。

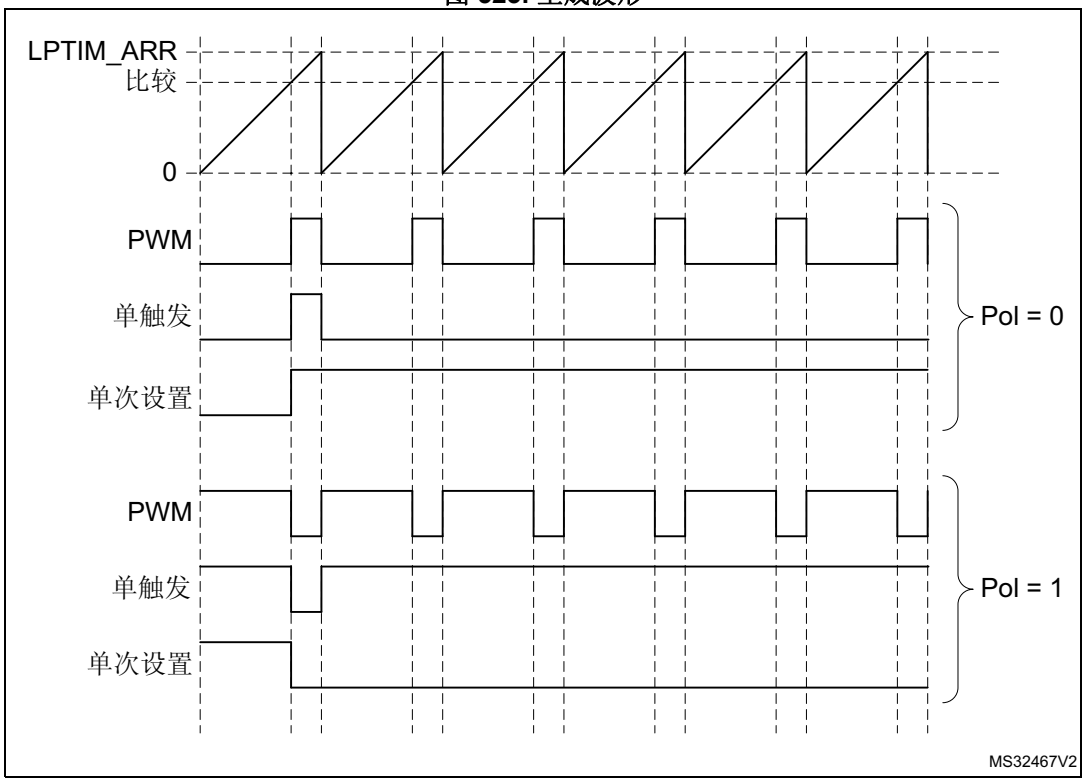
LPTIM 输出波形可通过 WAVE 位配置，具体如下：

- 若将 WAVE 位复位为 0，则会强制 LPTIM 生成 PWM 波形或单脉冲波形，具体取决于将哪个位（CNTSTRT 或 SNGSTRT）置 1。
- 若将 WAVE 位置 1，则会强制 LPTIM 生成单次设置模式波形。

WAVPOL 位控制 LPTIM 输出极性。更改立即生效，因此输出默认值将在极性重新配置后立即更改，甚至会在定时器使能前进行更改。

生成的信号的频率高达 LPTIM 时钟频率 2 分频。*图 523* 给出了可能在 LPTIM 输出上生成的三种波形。此外，此图还显示了通过 WAVPOL 位更改极性所产生的效果。

图 523. 生成波形



43.4.11 寄存器更新

LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器在 APB 总线写操作后会立即更新，若定时器已启动，也可在当前周期结束时更新。

PRELOAD 位控制 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器的更新方式：

- 当 PRELOAD 位复位为 0 时，LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器会在写访问后立即更新。
- 当 PRELOAD 位置 1 时，若定时器已启动，LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器会在当前周期结束时更新。

LPTIM APB 接口和 LPTIM 内核逻辑使用的时钟不同，因此在 APB 写操作后，需要经过一定的延迟，写入值才能用于计数器比较器。在此延迟期间，必须避免向这些寄存器执行其他写操作。

LPTIM_ISR 寄存器中的 ARROK 标志和 CMPOK 标志分别指示 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器的写操作已完成。

向 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器执行写操作后，只有在前一次写操作完成后，才能对同一寄存器执行新的写操作。在 ARROK 标志或 CMPOK 标志置 1 前执行连续的写操作将造成无法预知的结果。

43.4.12 计数器模式

LPTIM 计数器可用于对 LPTIM Input1 上的外部事件进行计数，也可用于对内部时钟周期进行计数。CKSEL 位和 COUNTMODE 位用于控制将使用哪些源更新计数器。

若使用 LPTIM 对 Input1 上的外部事件进行计数，计数器可在上升沿、下降沿或两种边沿进行更新，具体取决于写入 CKPOL[1:0] 位的值。

根据 CKSEL 和 COUNTMODE 值，可选择以下计数模式：

- CKSEL = 0：LPTIM 由内部时钟源提供时钟
  - COUNTMODE = 0

    LPTIM 配置为由内部时钟源提供时钟，LPTIM 计数器配置为在每个内部时钟脉冲后进行更新。
  - COUNTMODE = 1

    LPTIM 外部 Input1 通过提供给 LPTIM 的内部时钟进行采样。

    因此，为了不丢失任何事件，外部 Input1 信号变化的频率在任何时候都不应超过提供给 LPTIM 的内部时钟的频率。此外，不得对提供给 LPTIM 的内部时钟进行预分频 (PRESC[2:0] = 000)。
- CKSEL = 1：LPTIM 由外部时钟源提供时钟

  COUNTMODE 值不相关。

  在这种配置下，LPTIM 无需内部时钟源（已使能干扰滤波器时除外）。注入到 LPTIM 外部 Input1 的信号用作 LPTIM 的系统时钟。该配置适合未使能任何内置振荡器的工作模式。

  对于这种配置，LPTIM 计数器可以在 Input1 时钟信号的上升沿或下降沿进行更新，但不可在上升沿和下降沿均更新。

  由于注入到 LPTIM 外部 Input1 的信号也可用于为 LPTIM 内核逻辑提供时钟，计数器递增计数前存在一些初始延时（使能 LPTIM 后）。更确切地说，LPTIM 外部 Input1 的前五个有效边沿将丢失（使能 LPTIM 后）。

43.4.13 定时器使能

LPTIM_CR 寄存器中的 ENABLE 位用于使能/禁止 LPTIM 内核逻辑。将 ENABLE 位置 1 后，需要延迟两个计数器时钟周期，才能真正使能 LPTIM。

LPTIM_CFGR 和 LPTIM_IER 寄存器必须在禁止 LPTIM 后才能修改。

43.4.14 定时器计数器复位

为了将 LPTIM_CNT 寄存器的内容复位为零，实施了两种复位机制：

- 同步复位机制：同步复位由 LPTIM_CR 寄存器中的 COUNTRST 位控制。在将 COUNTRST 位字段置 1 后，复位信号在 LPTIM 内核时钟域中传播。因此，重要的是要注意，要在经历几个 LPTIM 内核逻辑时钟脉冲之后再考虑复位。这将使 LPTIM 计数器在复位触发和生效之间额外计数几个脉冲。由于 COUNTRST 位位于 APB 时钟域中，并且 LPTIM 计数器位于 LPTIM 内核时钟域中，因此当将 1 写入到 COUNTRST 位时，内核时钟需要 3 个时钟周期的延迟用以同步由 APB 时钟域发出的复位信号。

- 异步复位机制：异步复位由位于 LPTIM_CR 寄存器中的 RSTARE 位控制。当该位置 1 时，对 LPTIM_CNT 寄存器的任何读访问都会将其内容复位为零。应在不提供 LPTIM 内核时钟的时间范围内触发异步复位。例如，当 LPTIM Input1 管脚为外部时钟输入管脚时，只有当足够保证 LPTIM Input1 不会发生反转时，才应用异步复位。
  应注意的是，为了实现可靠的 LPTIM_CNT 寄存器内容读取，必须执行两次连续的读访问并进行比较。当两次读访问的值相等时，可认为读访问可靠。然而，当使能了异步复位时，不可能两次读取 LPTIM_CNT 寄存器。

警告： LPTIM 内没有防止两个复位机制同时使用的机制。所以开发人员应该确保这两个机制是排斥地使用。

43.4.15 编码器模式

此模式用于处理来自正交编码器的信号，此正交编码器用于检测旋转元件的角度位置。编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 LPTIM_ARR 寄存器中编程的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此必须在启动计数器之前配置 LPTIM_ARR。通过两个外部输入信号（Input1 和 Input2）生成时钟信号作为 LPTIM 计数器的时钟。这两个信号间的相位确定计数方向。

仅当 LPTIM 由内部时钟源提供时钟时才可使用编码器模式。Input1 和 Input2 输入上的信号频率不得超过 LPTIM 内部时钟频率 4 分频。必须满足此条件才能确保 LPTIM 正常工作。

方向变化由 LPTIM_ISR 寄存器中的两个递减和递增标志指示。此外，如果通过 DOWNIE 位使能，还可为两种方向变化事件产生中断。

要激活编码器模式，必须将 ENC 位置 1。LPTIM 必须首先配置为连续模式。

当编码器模式激活时，LPTIM 计数器按照增量编码器的速度和方向自动修改。因此，其内容始终代表编码器的位置。计数方向由递增和递减标志指示，对应于编码器转子的旋转方向。

根据使用 CKPOL[1:0] 位配置的边沿敏感性，可得几种不同的计数方案。下表汇总了可能的组合（假设 Input1 和 Input2 不同时切换）。

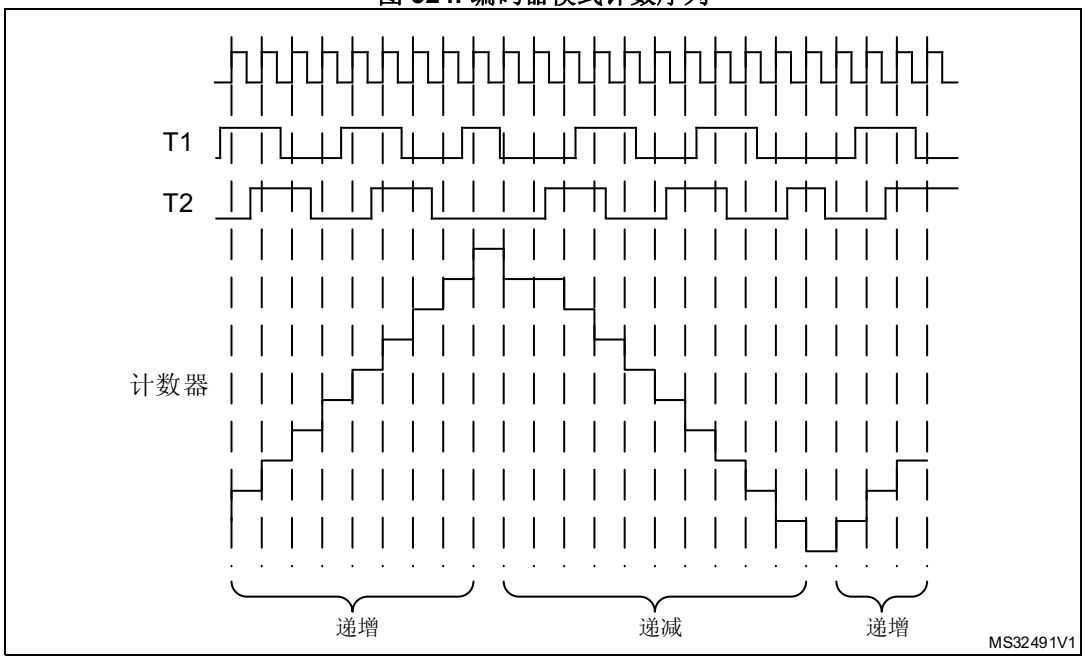
表 384. 编码器计数方案

有效边沿	相反信号的电平（Input1 对应 Input2，Input2 对应 Input1）	Input1 信号		Input2 信号	
		上升	下降	上升	下降
上升沿	高	递减	不计数	递增	不计数
	低	递增	不计数	递减	不计数
下降沿	高	不计数	递增	不计数	递减
	低	不计数	递减	不计数	递增
两种边沿	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

下图所示为编码器模式下配置了两种边沿敏感性的计数序列。

小心： 在此模式下，LPTIM 必须由内部时钟源提供时钟，因此 CKSEL 位必须保持其复位值 0。另外，预分频器分频比必须等于其复位值 1（PRESC[2:0] 位必须为“000”）。

图 524. 编码器模式计数序列



43.4.16 调试模式

当微控制器进入调试模式时（内核停止），LPTIM 计数器会根据 DBG 模块中的 DBG_LPTIM_STOP 配置位选择继续正常工作或者停止工作。

43.5 LPTIM 低功耗模式

表 385. 低功耗模式对 LPTIM 的影响

模式	说明
CSleep（MPU 或 MCU 子系统状态）	无影响。LPTIM 中断可使子系统退出睡眠模式。
Stop + LP-Stop	如果 LPTIM 由 Stop 模式下可用的振荡器提供时钟，则 LPTIM 可以正常工作，中断会使设备退出 Stop 模式。
LPLV-Stop	与 Stop + LP-Stop 模式相同，但只允许将 LSE 或 LSI 用作内核时钟源。
待机或关断	LPTIM 外设掉电，退出待机模式后必须重新初始化。

43.6 LPTIM 中断

若以下事件通过 LPTIM_IER 寄存器使能，则这些事件会生成中断/唤醒事件：

- 比较匹配
- 自动重载匹配（编码器模式下无论哪种方向）
- 外部触发事件
- 自动重载寄存器写操作完成
- 比较寄存器写操作完成
- 方向变化（编码器模式），可编程（递增/递减/同时递增和递减）。

注： *只要 LPTIM_IER 寄存器（中断使能寄存器）中的位在 LPTIM_ISR 寄存器（状态寄存器）中相应标志置 1 后置 1，就不会触发中断*

表 386. 中断事件

中断事件	说明
比较匹配	当计数器寄存器 (LPTIM_CNT) 的内容与比较寄存器 (LPTIM_CMP) 的内容匹配时，生成中断标志。
自动重载匹配	当计数器寄存器 (LPTIM_CNT) 的内容与自动重新加载寄存器 (LPTIM_ARR) 的内容匹配时，生成中断标志。
外部触发事件	当检测到外部触发事件时，会生成中断标志
自动重载寄存器更新成功	当对 LPTIM_ARR 寄存器的写操作完成时，生成中断标志。
比较寄存器更新成功	当对 LPTIM_CMP 寄存器的写操作完成时，生成中断标志。
方向更改	用于编码器模式。嵌入两个中断标志以发出方向更改的信号： – UP 标志发出递增计数方向更改的信号 – DOWN 标志发出递减计数方向更改的信号。

43.7 LPTIM 寄存器

有关寄存器说明中使用的缩写，请参见*第 1.2 节：寄存器相关缩写词列表*。

外设寄存器只能按字（32 位）进行访问。

43.7.1 LPTIM 中断和状态寄存器 (LPTIM_ISR)

LPTIM interrupt and status register

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWN	UP	ARR OK	CMP OK	EXT TRIG	ARRM	CMPM
									r	r	r	r	r	r	r

位 31:7 保留，必须保持复位值。

位 6 **DOWN**：计数方向从递增变为递减 (Counter direction change up to down)

在编码器模式下，由硬件将 DOWN 位置 1 时，会通知应用计数方向由递增变为递减。可以通过向 LPTIM_ICR 寄存器中的 DOWNCF 位写入 1 来将 DOWN 标志清零。

注意：如果 LPTIM 不支持编码器模式功能，则保留该位。请参见第 43.3 节：LPTIM 实现。

位 5 **UP**：计数方向从递减变为递增 (Counter direction change down to up)

在编码器模式下，由硬件将 UP 位置 1 时，会通知应用计数方向由递减变为递增。可以通过向 LPTIM_ICR 寄存器中的 UPCF 位写入 1 来将 UP 标志清零。

注意：如果 LPTIM 不支持编码器模式功能，则保留该位。请参见第 43.3 节：LPTIM 实现。

位 4 **ARROK**：自动重载寄存器更新成功 (Autoreload register update OK)

由硬件将 ARROK 置 1 时，会通知应用 LPTIM_ARR 寄存器的 APB 总线写操作已成功完成。可以通过向 LPTIM_ICR 寄存器中的 ARROKCF 位写入 1 将 ARROK 标志清零。

位 3 **CMPOK**：比较寄存器更新成功

由硬件将 CMPOK 置 1 时，会通知应用 LPTIM_CMP 寄存器的 APB 总线写操作已成功完成。可以通过向 LPTIM_ICR 寄存器中的 CMPOKCF 位写入 1 来将 CMPOK 标志清零。

位 2 **EXTTRIG**：外部触发边沿事件 (External trigger edge event)

由硬件将 EXTTRIG 置 1 时，会通知应用所选的外部触发输入上产生有效边沿。如果由于定时器已启动而忽略触发事件，则不会将此标志置 1。可以通过向 LPTIM_ICR 寄存器中的 EXTTRIGCF 位写入 1 来将 EXTTRIG 标志清零。

位 1 **ARRM**：自动重载匹配 (Autoreload match)

由硬件将 ARRM 置 1 时，会通知应用 LPTIM_CNT 寄存器的值已达到 LPTIM_ARR 寄存器的值。可以通过向 LPTIM_ICR 寄存器中的 ARRMCF 位写入 1 来将 ARRM 标志清零。

位 0 **CMPM**：比较匹配

由硬件将 CMPM 位置 1 时，会通知应用 LPTIM_CNT 寄存器的值已达到 LPTIM_CMP 寄存器的值。可以通过向 LPTIM_ICR 寄存器中的 CMPMCF 位写入 1 来将 CMPM 标志清零。

43.7.2 LPTIM 中断清零寄存器 (LPTIM_ICR)

LPTIM interrupt clear register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNCF	UPCF	ARROKCF	CMPOKCF	EXTTRIGCF	ARRMCF	CMPMCF
									w	w	w	w	w	w	w

位 31:7 保留，必须保持复位值。

位 6 **DOWNCF**：方向变为递减清零标志 (Direction change to down clear flag)
将 1 写入该位时，LPTIM_ISR 寄存器中的 DOWN 标志将清零。
注意：如果 LPTIM 不支持编码器模式功能，则保留该位。请参见第 43.3 节：LPTIM 实现。

位 5 **UPCF**：方向变为递增清零标志 (Direction change to UP clear flag)
将 1 写入该位时，LPTIM_ISR 寄存器中的 UP 标志将清零。
注意：如果 LPTIM 不支持编码器模式功能，则保留该位。请参见第 43.3 节：LPTIM 实现。

位 4 **ARROKCF**：自动重载寄存器更新成功清零标志 (Autoreload register update OK clear flag)
将 1 写入该位时，LPTIM_ISR 寄存器中的 ARROK 标志将清零

位 3 **CMPOKCF**：比较寄存器更新成功清零标志 (Compare register update OK clear flag)
将 1 写入该位时，LPTIM_ISR 寄存器中的 CMPOK 标志将清零

位 2 **EXTTRIGCF**：外部触发有效边沿清零标志 (External trigger valid edge clear flag)
将 1 写入该位时，LPTIM_ISR 寄存器中的 EXTTRIG 标志将清零

位 1 **ARRMCF**：自动重载匹配清零标志 (Autoreload match clear flag)
将 1 写入该位时，LPTIM_ISR 寄存器中的 ARRM 标志将清零

位 0 **CMPMCF**：比较匹配清零标志 (Compare match clear flag)
将 1 写入该位时，LPTIM_ISR 寄存器中的 CMPM 标志将清零。

43.7.3 LPTIM 中断使能寄存器 (LPTIM_IER)

LPTIM interrupt enable register

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNIE	UPIE	ARROKIE	CMPOKIE	EXTTRIGIE	ARRMIE	CMPMIE
									rw	rw	rw	rw	rw	rw	rw

位 31:7 保留，必须保持复位值。

位 6 **DOWNIE**：方向变为递减中断使能 (Direction change to down Interrupt Enable)
0： 禁止 DOWN 中断
1： 使能 DOWN 中断
注意：如果 LPTIM 不支持编码器模式功能，则保留该位。请参见第 43.3 节：LPTIM 实现。

位 5 **UPIE**：方向变为递增中断使能 (Direction change to UP Interrupt Enable)
0： 禁止 UP 中断
1： 使能 UP 中断
注意：如果 LPTIM 不支持编码器模式功能，则保留该位。请参见第 43.3 节：LPTIM 实现。

位 4 **ARROKIE**：自动重载寄存器更新成功中断使能 (Autoreload register update OK Interrupt Enable)
0： 禁止 ARROK 中断
1： 使能 ARROK 中断

位 3 **CMPOKIE**：比较寄存器更新成功中断使能 (Compare register update OK interrupt enable)
0： 禁止 CMPOK 中断
1： 使能 CMPOK 中断

位 2 **EXTTRIGIE**：外部触发有效边沿中断使能 (External trigger valid edge Interrupt Enable)
0： 禁止 EXTTRIG 中断
1： 使能 EXTTRIG 中断

位 1 **ARRMIE**：自动重载匹配中断使能 (Autoreload match Interrupt Enable)
0： 禁止 ARRM 中断
1： 使能 ARRM 中断

位 0 **CMPMIE**：比较匹配中断使能 (Compare match Interrupt Enable)
0： 禁止 CMPM 中断
1： 使能 CMPM 中断

小心： 必须在 LPTIM 已禁止时（ENABLE 位复位为 0）才能修改 LPTIM_IER 寄存器

43.7.4 LPTIM 配置寄存器 (LPTIM_CFGR)

LPTIM configuration register

偏移地址：0x00C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENC	COUNT MODE	PRELOAD	WAVPOL	WAVE	TIMOUT	TRIGEN[1:0]		Res.
							rw	rw	rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIGSEL[2:0]			Res.	PRESC[2:0]			Res.	TRGFLT[1:0]		Res.	CKFLT[1:0]		CKPOL[1:0]		CKSEL
rw	rw	rw		rw	rw	rw		rw	rw		rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29 保留，必须保持复位值。

位 28:25 保留，必须保持复位值。

位 24 **ENC**：编码器模式使能 (Encoder mode enable)

ENC 位控制编码器模式

0： 禁止编码器模式

1： 使能编码器模式

注意：如果 LPTIM 不支持编码器模式功能，则保留该位。请参见第 43.3 节：LPTIM 实现。

位 23 **COUNTMODE**：计数器模式使能 (counter mode enabled)

COUNTMODE 位用于选择 LPTIM 使用哪个时钟源来为计数器提供时钟：

0：计数器在每个内部时钟脉冲后递增

1：计数器在 LPTIM 外部 Input1 上的每个有效时钟脉冲后递增

位 22 **PRELOAD**：寄存器更新模式 (Registers update mode)

PRELOAD 位控制 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器的更新方式

0： 寄存器在每次 APB 总线写访问后更新

1： 寄存器在当前 LPTIM 周期结束时更新

位 21 **WAVPOL**：波形极性 (Waveform shape polarity)

WAVEPOL 位控制输出极性

0： LPTIM 输出反映 LPTIM_CNT 寄存器和 LPTIM_CMP 寄存器之间的比较结果

1： LPTIM 输出反映与 LPTIM_CNT 寄存器和 LPTIM_CMP 寄存器之间的比较结果相反的值

位 20 **WAVE**：波形 (Waveform shape)

WAVE 位控制输出波形

0： 停用单次设置模式

1： 激活单次设置模式

位 19 **TIMOUT**：超时使能 (Timeout enable)

TIMOUT 位控制超时功能

0： 定时器已启动时到达的触发事件将被忽略

1： 定时器已启动时到达的触发事件将复位并重新启动计数器

位 18:17 **TRIGEN[1:0]**：触发使能和极性 (Trigger enable and polarity)

TRIGEN 位控制 LPTIM 计数器是否由外部触发信号启动。如果已选择由外部触发信号启动，触发有效边沿的配置有以下三种：

00：软件触发（由软件启动计数）

01：上升沿为有效边沿

10：下降沿为有效边沿

11：上升沿和下降沿均为有效边沿

位 16 保留，必须保持复位值。

位 15:13 **TRIGSEL[2:0]**：触发选择器 (Trigger selector)

TRIGSEL 位用于选择作为 LPTIM 触发事件的触发源，可用触发源包括以下 8 种：

000：lptim_ext_trig0

001：lptim_ext_trig1

010：lptim_ext_trig2

011：lptim_ext_trig3

100：lptim_ext_trig4

101：lptim_ext_trig5

110：lptim_ext_trig6

111：lptim_ext_trig7

有关详细信息，请参见*第 43.4.3 节：LPTIM 输入和触发映射*。

位 12 保留，必须保持复位值。

位 11:9 **PRESC[2:0]**：时钟预分频器 (Clock prescaler)

PRESC 位配置预分频器的分频系数。分频系数可从以下分频系数中选择：

000：/1
001：/2
010：/4
011：/8
100：/16
101：/32
110：/64
111：/128

位 8 保留，必须保持复位值。

位 7:6 **TRGFLT[1:0]**：触发信号的可配置数字滤波器 (Configurable digital filter for trigger)

TRGFLT 值用于设置连续相同采样的数量，若在内部触发信号电平发生变化时检测到此类连续采样，才会将此电平变化视为有效电平切换。必须存在内部时钟源才能使用此功能

00：任何触发信号有效电平变化均视为有效触发
01：触发信号有效电平变化必须至少稳定 2 个时钟周期，才能将其视为有效触发。
10：触发信号有效电平变化必须至少稳定 4 个时钟周期，才能将其视为有效触发。
11：触发信号有效电平变化必须至少稳定 8 个时钟周期，才能将其视为有效触发。

位 5 保留，必须保持复位值。

位 4:3 **CKFLT[1:0]**：外部时钟的可配置数字滤波器 (Configurable digital filter for external clock)

CKFLT 值用于设置连续相同采样的数量，若在外部时钟信号电平发生变化时检测到此类连续采样，才会将此电平变化视为有效电平切换。必须存在内部时钟源才能使用此功能

00：任何外部时钟信号电平变化均视为有效切换
01：外部时钟信号电平变化必须至少稳定 2 个时钟周期，才能将其视为有效切换。
10：外部时钟信号电平变化必须至少稳定 4 个时钟周期，才能将其视为有效切换。
11：外部时钟信号电平变化必须至少稳定 8 个时钟周期，才能将其视为有效切换。

位 2:1 **CKPOL[1:0]**：时钟极性 (Clock polarity)

如果 LPTIM 由外部时钟源提供时钟：

当 LPTIM 由外部时钟源提供时钟时，CKPOL 位用于配置计数所使用的有效边沿：

00：上升沿为用于计数的有效边沿。
如果将 LPTIM 配置为编码器模式（ENC 位置 1），则编码器子模式 1 有效。
01：下降沿为用于计数的有效边沿
如果将 LPTIM 配置为编码器模式（ENC 位置 1），则编码器子模式 2 有效。
10：上升沿和下降沿均为有效边沿。当外部时钟信号的上升沿和下降沿均视为有效边沿时，LPTIM 还必须由内部时钟源提供时钟，且内部时钟源频率至少等于外部时钟频率的四倍。
如果将 LPTIM 配置为编码器模式（ENC 位置 1），则编码器子模式 3 有效。
11：不允许

有关编码器模式子模式的更多详细信息，请参见*第 43.4.15 节：编码器模式*。

位 0 **CKSEL**：时钟选择器 (Clock selector)

CKSEL 位选择 LPTIM 将使用的时钟源：

0： LPTIM 由内部时钟源（APB 时钟或任意内置振荡器）提供时钟
1： LPTIM 由外部时钟源通过 LPTIM 外部 Input1 提供时钟

小心： 必须在 LPTIM 已禁止时（ENABLE 位复位为 0）才能修改 LPTIM_CFGR 寄存器。

43.7.5 LPTIM 控制寄存器 (LPTIM_CR)

LPTIM control register

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RSTARE	COUNTRST	CNTSTRT	SNGSTRT	ENABLE
											rw	rs	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4 **RSTARE**：读操作后复位使能 (Reset after read enable)

该位由软件置 1 和清零，当 RSTARE 置 1 时，对 LPTIM_CNT 寄存器的任何读取访问都将异步复位 LPTIM_CNT 寄存器的内容。

只有在使能 LPTIM 时才能将该位置 1。

位 3 **COUNTRST**：计数器复位 (Counter reset)

该位通过软件置 1，通过硬件清零。当设置为 1 时，该位将触发 LPTIM_CNT 计数器寄存器的同步复位。由于该复位的同步性质，它仅在 3 个 LPTimer 内核时钟周期（LPTimer 内核时钟可能不同于 APB 时钟）的同步延迟之后发生。

只有在使能 LPTIM 时才能将该位置 1。该位由硬件自动复位。

小心： COUNTRST 在已由硬件清零之前，不得由软件置“1”。因此，软件应在尝试将 COUNTRST 位置“1”之前检查其是否已清零。

位 2 **CNTSTRT**：定时器以连续模式启动 (Timer start in Continuous mode)

该位通过软件置 1，通过硬件清零。

若通过软件启动（TRIGEN[1:0] =“00”），将该位置 1 会使 LPTIM 以连续模式启动。

如果禁止软件启动（TRIGEN[1:0] 不等于“00”），将该位置 1 会使定时器在检测到外部触发信号后立即以连续模式启动。

如果在进行单脉冲模式计数时将该位置 1，则在 LPTIM_ARR 寄存器和 LPTIM_CNT 寄存器下一次匹配时定时器不会停止，LPTIM 计数器将继续以连续模式计数。

只有在使能 LPTIM 时才能将该位置 1。该位由硬件自动复位。

位 1 **SNGSTRT**：LPTIM 以单脉冲模式启动 (LPTIM start in Single mode)

该位通过软件置 1，通过硬件清零。

若通过软件启动（TRIGEN[1:0] =“00”），将该位置 1 会使 LPTIM 以单脉冲模式启动。

如果禁止软件启动（TRIGEN[1:0] 不等于“00”），将该位置 1 会使 LPTIM 在检测到外部触发信号后立即以单脉冲模式启动。

如果在 LPTIM 处于连续计数模式时将该位置 1，LPTIM 将在 LPTIM_ARR 寄存器和 LPTIM_CNT 寄存器下一次匹配时停止。

只有在使能 LPTIM 时才能将该位置 1。该位由硬件自动复位。

位 0 **ENABLE**：LPTIM 使能 (LPTIM enable)

ENABLE 位由软件置 1 和清零。

0：禁止 LPTIM

1：使能 LPTIM

43.7.6 LPTIM 比较寄存器 (LPTIM_CMP)

LPTIM compare register

偏移地址：0x014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **CMP[15:0]**：比较值 (Compare value)
CMP 为 LPTIM 所使用的比较值。

小心： 必须在 LPTIM 已使能时（ENABLE 位置 1）才能修改 LPTIM_CMP 寄存器。

43.7.7 LPTIM 自动重载寄存器 (LPTIM_ARR)

LPTIM autoreload register

偏移地址：0x018

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **ARR[15:0]**：自动重载值 (Auto reload value)
ARR 为 LPTIM 的自动重载值。
此值必须严格大于 CMP[15:0] 的值。

小心： 必须在 LPTIM 已使能时（ENABLE 位置 1）才能修改 LPTIM_ARR 寄存器。

43.7.8 LPTIM 计数器寄存器 (LPTIM_CNT)

LPTIM counter register

偏移地址：0x01C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

当 LPTIM 通过异步时钟运行时，读取 LPTIM_CNT 寄存器会返回不可靠的值。因此在这种情况下，必须连续执行读访问两次，并验证两次返回的值是否相同。

应注意的是，为了实现可靠的 LPTIM_CNT 寄存器读访问，必须执行两次连续的读访问并进行比较。当两次连续读访问的值相等时，可认为读访问可靠。

43.7.9 LPTIM3 选项寄存器 (LPTIM3_OR)

LPTIM3 option register

偏移地址：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OR_0
															rw

位 31:1 保留，必须保持复位值。

位 0 **OR_0**：选项寄存器位 0 (Option register bit 0)

0： LPTIM3 输出连接到 I/O 或其他功能

1： LPTIM3 输出连接到 TAMP

43.7.10 LPTIM 配置寄存器 2 (LPTIM_CFGR2)

LPTIM configuration register 2

偏移地址：0x024

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IN2SEL[1:0]		Res.	Res.	IN1SEL[1:0]	
										rw	rw			rw	rw

位 31:6 保留，必须保持复位值。

位 5:4 **IN2SEL[1:0]**：LPTIM 输入 2 选择 (LPTIM input 2 selection)

IN2SEL 位用于控制 LPTIM 输入 2 复用器，它将 LPTIM 输入 2 连接至其中一个可用输入。

00：lptim_in2_mux0

01：lptim_in2_mux1

10：lptim_in2_mux2

11：lptim_in2_mux3

有关连接详细信息，请参见*第 43.4.3 节：LPTIM 输入和触发映射*。

注意：如果 LPTIM 不支持编码器模式功能，则保留这些位。请参见第 43.3 节：LPTIM 实现。

位 3:2 保留，必须保持复位值。

位 1:0 **IN1SEL[1:0]**：LPTIM 输入 1 选择 (LPTIM input 1 selection)

IN1SEL 位用于控制 LPTIM 输入 1 复用器，它将 LPTIM 输入 1 连接至其中一个可用输入。

00：lptim_in1_mux0

01：lptim_in1_mux1

10：lptim_in1_mux2

11：lptim_in1_mux3

有关连接详细信息，请参见*第 43.4.3 节：LPTIM 输入和触发映射*。

注意：如果 LPTIM 未实现外部输入时钟，则保留这些位。请参见第 43.3 节：LPTIM 实现。

小心： 必须在 LPTIM 已禁止时（ENABLE 位复位为 0）才能修改 LPTIM_CFGR2 寄存器。

43.7.11 LPTIM x 外设硬件配置寄存器 (LPTIMx_HWCFGR)（x = 1 到 5）

LPTIM x peripheral hardware configuration register

偏移地址：0x3F0

复位值：块 10x0001 0804

复位值：块 20x0001 0804

复位值：块 30x0100 0804

复位值：块 40x0000 0804

复位值：块 50x0000 0804

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CFG4[7:0]								Res.	Res.	Res.	Res.	CFG3[3:0]			
r	r	r	r	r	r	r	r					r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG2[7:0]								CFG1[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 **CFG4[7:0]**：外设硬件配置 4 (peripheral hardware configuration 4)
这位字段为只读位字段，读作 CFG4[7:0] = 选项寄存器

位 23:20 保留，必须保持复位值。

位 19:16 **CFG3[3:0]**：外设硬件配置 3 (peripheral hardware configuration 3)
该位字段为只读位字段，读作 CFG3[3:0] = 编码器模式（1：使能，0：禁止）

位 15:8 **CFG2[7:0]**：外设硬件配置 2 (peripheral hardware configuration 2)
该位字段为只读位字段，读作 CFG2[7:0] = 触发器编号

位 7:0 **CFG1[7:0]**：外设硬件配置 1 (peripheral hardware configuration 1)
该位字段为只读位字段，读作 CFG1[7:0] = 输入复用器大小

43.7.12 LPTIM 外设版本标识寄存器 (LPTIM_VERR)

LPTIM peripheral version identification register

偏移地址：0x3F4

复位值：0x0000 0014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[7:4]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[7:4]**：主版本 (Major revision)
该位字段为只读位字段，读作 MAJREV[7:4] = 1。

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)
该位字段为只读位字段，读作 MINREV[3:0] = 4。

43.7.13 LPTIM 外设类型标识寄存器 (LPTIM_PIDR)

LPTIM peripheral type identification register

偏移地址：0x3F8

复位值：0x0012 0011

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P_ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P_ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **P_ID[31:0]**：外设类型标识符 (Peripheral type identifier)
只读寄存器，始终读作 P_ID[31:0] = 0x00120011。

43.7.14 LPTIM 寄存器映射大小标识寄存器 (LPTIM_SIDR)

LPTIM registers map size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
S_ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S_ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **S_ID[31:0]**：寄存器映射大小标识符 (Registers map size identifier)
只读寄存器，始终读作 S_ID[31:0] = 0xA3C5DD01。
该寄存器用于识别分配给 LPTIM 寄存器的存储区域的大小。该存储区域的大小为 1 KB。

43.7.15 LPTIM 寄存器映射

下表对 LPTIM 寄存器进行了汇总。

表 387. LPTIM 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	LPTIM_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWN(1)	UP(1)	ARROK	CMPOK	EXTTRIG	ARRM	CMPM
	Reset value																										0	0	0	0	0	0	0
0x004	LPTIM_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNCF(1)	UPCF(1)	ARROKCF	CMPOKCF	EXTTRIGCF	ARRMCF	CMPMCF
	Reset value																										0	0	0	0	0	0	0
0x008	LPTIM_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNIE(1)	UPIE(1)	ARROKIE	CMPOKIE	EXTTRIGIE	ARRMIE	CMPMIE
	Reset value																										0	0	0	0	0	0	0
0x00C	LPTIM_CFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENC(1)	COUNTMODE	PRELOAD	WAVEPOL	WAVE	TIMOUT	TRIGEN		Res.	TRIGSEL[2:0]			Res.	PRESC			Res.	TRGFLT		Res.	CKFLT		CKPOL		CKSEL
	Reset value								0	0	0	0	0	0	0	0		0	0	0		0	0	0		0	0		0	0	0	0	0
0x010	LPTIM_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RSTARE	COUNTRST	CNTSTRT	SNGSTRT	ENABLE
	Reset value																												0	0	0	0	0
0x014	LPTIM_CMP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMP[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	LPTIM_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
0x01C	LPTIM_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	LPTIM3_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OR_0
	Reset value																																0
0x024	LPTIM_CFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IN2SEL[1:0]		Res.	Res.	IN1SEL[1:0]	
	Reset value																											0	0			0	0
0x3F0	LPTIM_HWCFGR	CFG4[27:20]								Res.	Res.	Res.	Res.	CFG3 [19:16]				CFG2[15:8]								CFG1[7:0]							
	Reset value	x	x	x	x	x	x	x	x					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x3F4	LPTIM_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV [7:4]				MINREV [3:0]			
	Reset value																									0	0	0	1	0	1	0	0

表 387. LPTIM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x3F8	LPTIM_PIDR	P_ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
0x3FC	LPTIM_SIDR	S_ID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

1. 如果 LPTIM 不支持编码器模式功能，则保留该位。请参见*第 43.3 节：LPTIM 实现*。

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

44 系统定时器生成器 (STGEN)

44.1 STGEN 简介

STGEN 可生成一个时间计数值，为设备中的多个处理器和其他模块提供一致的时间视图。

44.2 STGEN 主要特性

- 64 位宽，避免计满返回问题
- 从 0 或一个可编程值开始
- 控制 APB 接口可用来在发生掉电事件时保存和恢复定时器
- 只读 APB 接口可供非安全软件和调试工具读取定时器值
- hltdbg 输入可在全系统调试期间停止定时器值递增

44.3 STGEN 功能描述

Cortex®-A7 通用定时器根据递增系统计数器 (STGEN) 值调度事件及触发中断。

通用定时器可以：

- 生成定时器事件作为中断输出
- 生成事件流
- 支持虚拟化扩展

Cortex®-A7 通用定时器符合以下 Arm 架构参考手册系统计数器要求：

- 宽度：至少 56 位宽
- 频率：以固定频率递增

  在大多数情况下，STGEN 时钟仅来自外部振荡器 (HSE)，但以下情况除外：
  - 在启动阶段，STGEN 由 HSI (64 MHz) 提供时钟，直到设置 HSE。
  - 在低功耗模式（关闭了 HSE 的 Stop 模式、待机模式）期间，STGEN 由软件停止。从这些低功耗模式返回后使能，并由安全软件根据 RTC 调整时间。
- 精度：Arm 架构参考手册未规定所需精度，但建议 24 小时内计数器快慢时间不超出十秒。这一要求由软件来支持，偶尔会根据 RTC 更新 STGEN（递增）。

  使用低频模式不得影响实现的精度。
- 启动：计数器在 POR 时从 0 开始计数。

STGEN 提供系统时间的统一视图。

注： *Arm 建议系统计数器位于始终开启的电源域中，但目前实现中尚未提供这一支持。这样，在进入待机模式前，必须保存并恢复 STGEN，并在退出待机模式时由安全软件恢复。*

在低功耗模式期间，计数器可能会使用较低频率以较大步长递增。在这种情况下，计数器必须支持在高频高精度操作和低频低精度操作之间跳变，并且不影响所需的计数器精度。

软件可以访问 STGENC_CNTFIDO 寄存器，以读取系统计数器的时钟频率，并修改该寄存器的值。

44.3.1 STGEN 框图

下图所列为 STGEN 概述，包括：

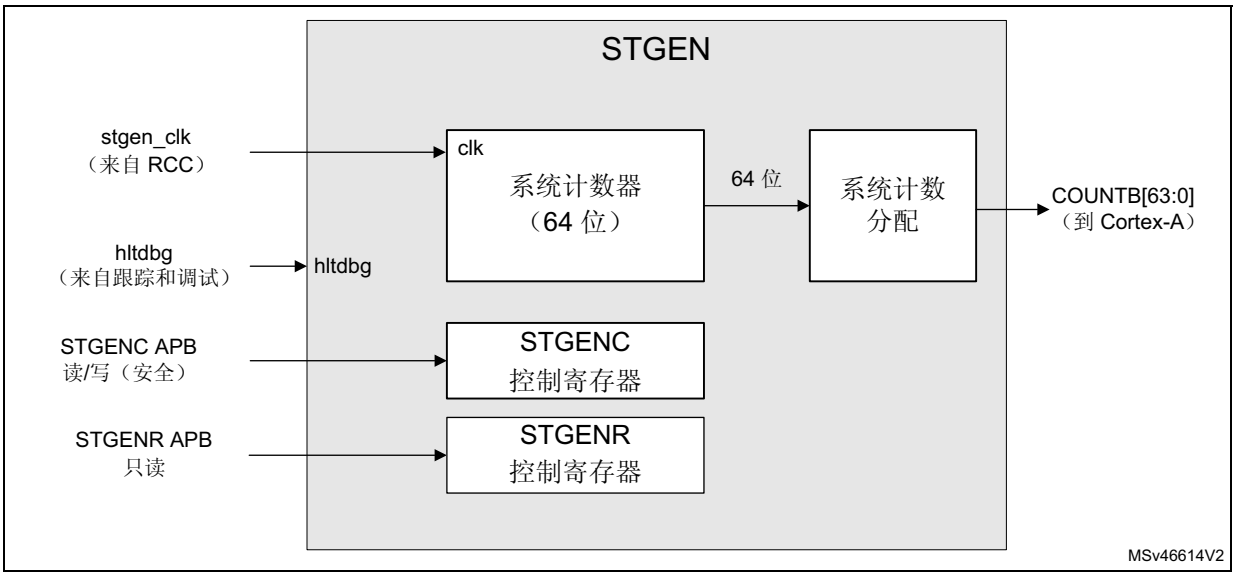
- 64 位系统计数器
- 控制寄存器及两个 APB 端口（对应于 STGENC 和 STGENR 寄存器）
- 系统计数分配逻辑，将系统计数器值格式化为 Cortex®-A7，使用灰色编码和频率插值。

注： *由于 TrustZone 安全隔离，使用两个 APB 端口，如下所述：*

——STGENC 只能通过 Cortex®-A7 安全进行读写访问。

——STGENR 可通过安全或非安全进行只读访问。

图 525. STGEN 框图



注： *hltdbg 是来自 Cortex®-A7 处理器的 DBGACK 输出的 AND 组合，因此当两个 Cortex®-A7 处理器均处于调试状态时，STGEN 会停止。*

44.4 STGEN 寄存器

STGEN 控制寄存器分为两组，可通过两个 APB 接口访问：

- 第一组可通过 STGEN**C** APB 实现读写访问，并且受到 TrustZone 安全保护。
- 第二组可通过 STGEN**R** APB 实现只读访问，与 TrustZone 安全性无关。

44.4.1 STGENC 控制寄存器 (STGENC_CNTCR)

STGENC control register

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HLTDBG	EN
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **HLTDBG**：调试时停止 (Halt on debug)

0：调试时不停止；hltdbg 信号对计数器无影响。

1：调试时停止；当 hltdbg 驱动为高电平时，计数值保持静态。

位 0 **EN**：使能

0：计数器被禁止并且不递增

1：计数器被使能并且递增

44.4.2 STGENC 状态寄存器 (STGENC_CNTSR)

STGENC status register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGH	Res.
														r	

位 31:2 保留，必须保持复位值。

位 1 **DBGH**：调试已停止 (Debug halted)

0：正常工作

1：计数器因系统级调试状态而停止

位 0 保留，必须保持复位值。

44.4.3 STGENC 值低位寄存器 (STGENC_CNTCVL)

STGENC value lower register

偏移地址：0x008

复位值：0x0000 0000

注： *控制接口必须先将 STGENC_CNTCR.EN 位清零，然后才能写入该寄存器。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNTCVL_L_32[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNTCVL_L_32[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CNTCVL_L_32[31:0]**：时间戳计数器当前值的低 32 位 (Current value of the timestamp counter, lower 32 bits)

要更改当前的时间戳值，需要先将新值的低 32 位写入该寄存器，然后再将高 32 位写入 STGENC_CNTCVU。在写入 STGENC_CNTCVU 寄存器之前，时间戳值保持不变。

44.4.4 STGENC 值高位寄存器 (STGENC_CNTCVU)

STGENC value upper register

偏移地址：0x00C

复位值：0x0000 0000

注： *控制接口必须先将 STGENC_CNTCR.EN 位清零，然后才能写入该寄存器。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNTCVU_U_32[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNTCVU_U_32[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CNTCVU_U_32[31:0]**：时间戳计数器当前值的高 32 位 (Current value of the timestamp counter, upper 32 bits)

要更改当前的时间戳值，需先将新值的低 32 位写入 STGENC_CNTCVL，然后再将高 32 位写入该寄存器。当写入该寄存器时，64 位时间戳值会更新为两次写入的值。

44.4.5 STGENC 基频寄存器 (STGENC_CNTFID0)

STGENC base frequency register

偏移地址：0x020

复位值：0x0000 0000

注： *控制接口必须先将 STGEN_CNTCR.EN 位清零，然后才能写入该寄存器。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FREQ[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREQ[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **FREQ[31:0]**：频率（每秒时钟节拍数）

可以指定高达 4 GHz 的频率。该寄存器必须编程为匹配 TSGEN (tsgen_clk) 的时钟输入频率（每秒时钟节拍数）。例如，
对于 50 MHz 的时钟，该寄存器的值必须是 0x02FAF080。

44.4.6 STGENC 外设 ID4 寄存器 (STGENC_PIDR4)

STGENC peripheral ID4 register

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				DES_2[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SIZE[3:0]**：始终为 0b0000

表示该设备仅占用 4 KB 的内存。

位 3:0 **DES_2[3:0]**：设计人员标识的一部分

STGEN_PIDR1.DES_0、STGEN_PIDR2.DES_1 和 STGEN_PIDR4.DES_2 一起标识组件的设计人员。
0b0100 JEDEC 连续代码

44.4.7 STGENC 外设 ID5 寄存器 (STGENC_PIDR5)

STGENC peripheral ID5 register

偏移地址：0xFD4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIDR5[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIDR5[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **PIDR5[31:0]**：外设 ID5 (Peripheral ID5)

44.4.8 STGENC 外设 ID6 寄存器 (STGENC_PIDR6)

STGENC peripheral ID6 register

偏移地址：0xFD8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIDR6[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIDR6[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **PIDR6[31:0]**：外设 ID6 (Peripheral ID6)

44.4.9 STGENC 外设 ID7 寄存器 (STGENC_PIDR7)

STGENC peripheral ID7 register

偏移地址：0xFDC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIDR7[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIDR7[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **PIDR7[31:0]**：外设 ID7 (Peripheral ID7)

44.4.10 STGENC 外设 ID0 寄存器 (STGENC_PIDR0)

STGENC peripheral ID0 register

偏移地址：0xFE0

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PART_0[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PART_0[7:0]**：组件的 12 位产品编号的位 [7:0]
由组件的设计人员指定该编号。

44.4.11 STGENC 外设 ID1 寄存器 (STGENC_PIDR1)

STGENC peripheral ID1 register

偏移地址：0xFE4

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DES_0[3:0]				PART_1[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **DES_0[3:0]**：设计人员标识的一部分
PIDR1.DES_0、PIDR2.DES_1 和 PIDR4.DES_2 一起标识组件的设计人员。

位 3:0 **PART_1[3:0]**：组件的 12 位产品编号的位 [11:8]
由组件的设计人员指定该编号。

44.4.12 STGENC 外设 ID2 寄存器 (STGENC_PIDR2)

STGENC peripheral ID2 register

偏移地址：0xFE8

复位值：0x0000 001B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	DES_1[2:0]		
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：设备版本号 (device revision number)

位 3 **JEDEC**：始终为 1
表示使用了 JEDEC 分配的设计人员 ID。

位 2:0 **DES_1[2:0]**：设计人员标识的一部分
STGEN_PIDR1.DES_0、STGEN_PIDR2.DES_1 和 STGEN_PIDR4.DES_2 一起标识组件的设计人员。

44.4.13 STGENC 外设 ID3 寄存器 (STGENC_PIDR3)

STGENC peripheral ID3 register

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：客户版本
0b0000 表明该组件没有勘误修正。

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
指示客户是否修改了组件的行为。在大多数情况下，该位字段为 0b0000（未修改）。当客户对该组件进行授权修改时，会更改此值。

44.4.14 STGENC 组件 ID0 寄存器 (STGENC_CIDR0)

STGENC component ID0 register

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_0[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PRMBL_0[7:0]**：前导码 0 (Preamble 0)
包含组件标识码的位 [7:0]。

44.4.15 STGENC 组件 ID1 寄存器 (STGENC_CIDR1)

STGENC component ID1 register

偏移地址：0xFF4

复位值：0x0000 00F0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PRMBL_1[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]:**组件的类别 (Class of the component)
例如，组件可以是 ROM 表或通用 CoreSight™ SoC-400 组件。包含组件标识码的位 [15:12]。
0b1111 表示组件是 CoreSight™ SoC-400 组件。

位 3:0 **PRMBL_1[3:0]**：前导码 1 (preamble 1)
包含组件标识码的位 [11:8]。

44.4.16 STGENC 组件 ID2 寄存器 (STGENC_CIDR2)

STGENC component ID2 register

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_2[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PRMBL_2[7:0]**：前导码 2 (Preamble 2)
包含组件标识码的位 [23:16]。

44.4.17 STGENC 组件 ID3 寄存器 (STGENC_CIDR3)

STGENC component ID3 register

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_3[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PRMBL_3[7:0]**：前导码 3 (Preamble 3)
包含组件标识码的位 [31:24]。

44.4.18 STGENR 值低位寄存器 (STGENR_CNTCVL)

STGENR value lower register

偏移地址：0x000

复位值：0x0000 0000

注： *控制接口必须先将 STGEN_CNTCR.EN 位清零，然后才能写入该寄存器。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNTCVL_L_32[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNTCVL_L_32[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CNTCVL_L_32[31:0]**：时间戳计数器的当前值（低 32 位）(Current value of the timestamp counter (the lower 32 bits))

44.4.19 STGENR 值高位寄存器 (STGENR_CNTCVU)

STGENR value upper register

偏移地址：0x004

复位值：0x0000 0000

注： *控制接口必须先将 STGEN_CNTCR.EN 位清零，然后才能写入该寄存器。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNTCVU_U_32[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNTCVU_U_32[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CNTCVU_U_32[31:0]**：时间戳计数器的当前值（高 32 位）(Current value of the timestamp counter (the upper 32 bits))

44.4.20 STGENR 外设 ID4 寄存器 (STGENR_PIDR4)

STGENR peripheral ID4 register

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				DES_2[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SIZE[3:0]**：始终为 0b0000
表示该设备仅占用 4 KB 的内存。

位 3:0 **DES_2[3:0]**：设计人员标识的一部分
STGEN_PIDR1.DES_0、STGEN_PIDR2.DES_1 和 STGEN_PIDR4.DES_2 一起标识组件的设计人员。

44.4.21 STGENR 外设 ID5 寄存器 (STGENR_PIDR5)

STGENR peripheral ID5 register

偏移地址：0xFD4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIDR5[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIDR5[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **PIDR5[31:0]**：外设 ID5 (Peripheral ID5)

44.4.22 STGENR 外设 ID6 寄存器 (STGENR_PIDR6)

STGENR peripheral ID6 register

偏移地址：0xFD8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIDR6[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIDR6[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **PIDR6[31:0]**：外设 ID6 (Peripheral ID6)

44.4.23 STGENR 外设 ID7 寄存器 (STGENR_PIDR7)

STGENR peripheral ID7 register

偏移地址：0xFDC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIDR7[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIDR7[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **PIDR7[31:0]**：外设 ID7 (Peripheral ID7)

44.4.24 STGENR 外设 ID0 寄存器 (STGENR_PIDR0)

STGENR peripheral ID0 register

偏移地址：0xFE0

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PART_0[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PART_0[7:0]**：组件的 12 位产品编号中的位 [7:0]

44.4.25 STGENR 外设 ID1 寄存器 (STGENR_PIDR1)

STGENR peripheral ID1 register

偏移地址：0xFE4

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DES_0[3:0]				PART_1[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **DES_0[3:0]**：设计人员标识的一部分

STGEN_PIDR1.DES_0、STGEN_PIDR2.DES_1 和 STGEN_PIDR4.DES_2 一起标识组件的设计人员。

位 3:0 **PART_1[3:0]**：组件的 12 位产品编号的位 [11:8]

44.4.26 STGENR 外设 ID2 寄存器 (STGENR_PIDR2)

STGENR peripheral ID2 register

偏移地址：0xFE8

复位值：0x0000 001B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	DES_1[2:0]		
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：设备版本号 (Device revision number)

位 3 **JEDEC**：始终为 1

表示使用了 JEDEC 分配的设计人员 ID。

位 2:0 **DES_1[2:0]**：设计人员标识的一部分

STGEN_PIDR1.DES_0、STGEN_PIDR2.DES_1 和 STGEN_PIDR4.DES_2 一起标识组件的设计人员。

44.4.27 STGENR 外设 ID3 寄存器 (STGENR_PIDR3)

STGENR peripheral ID3 register

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：勘误修复标识 (Errata fix identification)
0b0000 表明该组件没有勘误修正。

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
指示客户是否修改了组件的行为。在大多数情况下，该位字段为 0b0000（未修改）。当客户对该组件进行授权修改时，会更改此值。

44.4.28 STGENR 组件 ID0 寄存器 (STGENR_CIDR0)

STGENR component ID0 register

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_0[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PRMBL_0[7:0]**：前导码 0 (Preamble 0)
包含组件标识码的位 [7:0]。

44.4.29 STGENR 组件 ID1 寄存器 (STGENR_CIDR1)

STGENR component ID1 register

偏移地址：0xFF4

复位值：0x0000 00F0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PRMBL_1[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]:** 组件的类别 (Class of the component)
包含组件标识码的位 [15:12]。

位 3:0 **PRMBL_1[3:0]：** 前导码 1 (Preamble 1)
包含组件标识码的位 [11:8]。

44.4.30 STGENR 组件 ID2 寄存器 (STGENR_CIDR2)

STGENR component ID2 register

偏移地址：0xFF8

复位值：0x0000 0050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_2[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PRMBL_2[7:0]：** 前导码 2 (Preamble 2)
包含组件标识码的位 [23:16]。

44.4.31 STGENR 组件 ID3 寄存器 (STGENR_CIDR3)

STGENR component ID3 register

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_3[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PRMBL_3[7:0]**：前导码 3 (Preamble 3)
包含组件标识码的位 [31:24]。

44.4.32 STGEN 寄存器映射

表 388. STGEN 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	STGENC_CNTCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HLTDBG	EN
	Reset value																															0	0
0x004	STGENC_CNTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGH	Res.
	Reset value																															0	
0x008	STGENC_CNTCVL	CNTCVL_L_32[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	STGENC_CNTCVU	CNTCVU_U_32[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0100-0x01C	Reserved	Reserved																															
0x0020	STGENC_CNTFID0	FREQ[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024-0x0FCC	Reserved	Reserved																															
0xFD0	STGENC_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				DES_2[3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4 + 0x4*(x-5) (x=5 to 7)	STGENC_PIDRx	PIDRx[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xFE0	STGENC_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PART_0[7:0]							
	Reset value																									0	0	0	0	0	0	0	1
0xFE4	STGENC_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DES_0[3:0]				PART_1[3:0]			
	Reset value																									1	0	1	1	0	0	0	1

表 388. STGEN 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFE8	STGENC_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	DES_1 [2:0]		
	Reset value																									0	0	0	1	1	0	1	1
0xFEC	STGENC_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	STGENC_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_0[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	STGENC_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PRMBL_1 [3:0]			
	Reset value																									1	1	1	1	0	0	0	0
0xFF8	STGENC_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_2[7:0]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	STGENC_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_3[7:0]							
	Reset value																									1	0	1	1	0	0	0	1

表 389. STGENR 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	STGENR_CNTCVL	CNTCVL_L_32[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	STGENR_CNTCVU	CNTCVU_U_32[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x008-0xFCC	Reserved	Reserved																															
0xFD0	STGENR_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				DES_2[3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4 + 0x4*(x-5) (x=5 to 7)	STGENR_PIDR5x	PIDRx[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xFE0-0xFDC	Reserved	Reserved																															
0xFE0	STGENR_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PART_0[7:0]							
	Reset value																									0	0	0	0	0	0	0	1
0xFE4	STGENR_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DES_0[3:0]				PART_1[3:0]			
	Reset value																									1	0	1	1	0	0	0	1
0xFE8	STGENR_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	DES_1 [2:0]		
	Reset value																									0	0	0	1	1	0	1	1
0xFEC	STGENR_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	STGENR_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_0[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	STGENR_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PRMBL_1 [3:0]			
	Reset value																									1	1	1	1	0	0	0	0
0xFF8	STGENR_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_2[7:0]							
	Reset value																									0	1	0	1	0	0	0	0

表 389. STGENR 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFFC	STGENR_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_3[7:0]							
	Reset value																									1	0	1	1	0	0	0	1

有关寄存器边界地址的信息，请参见*第 2.5 节*。

45 独立看门狗 (IWDG)

45.1 简介

独立看门狗 (IWDG) 外设可检测由于软件或硬件故障而引起的故障，因此具有较高的安全性。

IWDG 的时钟由独立时钟提供，即便在主时钟发生故障时仍然保持工作状态。

此外，看门狗功能在 V_{DD} 电压域中执行，因此即便在低功耗模式下，IWDG 仍然保持正常工作。有关该产品中 IWDG 的功能，请参见*第 45.3 节*。

IWDG 最适合那些需要将看门狗作为一个在主程序之外的进程完全独立运行的应用，它能够在这类应用中非常可靠地检测任何意外行为。

45.2 IWDG 主要特性

- 12 位递减计数器
- 双电压域，可在低功耗模式下工作
- 独立时钟
- 提前唤醒中断生成
- 复位生成
  - 发生超时时
  - 刷新超出预期窗口时

45.3 IWDG 实现

表 390. STM32MP13xx IWDG 特性(1)

IWDG 模式/特性	IWDG1	IWDG2
LSI 用作 IWDG 内核时钟 (iwdg_ker_ck)	X	X
窗口功能	X	X
提前唤醒中断生成	X	X
系统复位生成 (2)	X	X
可在系统 Stop 模式下工作	X	X
可在系统待机模式下工作	X	X
可在系统 Stop 模式下生成中断	X	X
可在系统待机模式下生成中断	-	-
可在 MPU 进入调试模式时被冻结	X (3)	X (4)
用于控制 Stop 模式下活动的选项字节 (5)	X	X
用于控制待机模式下活动的选项字节 (6)	X	X
用于控制硬件模式的选项字节 (7)	X	X

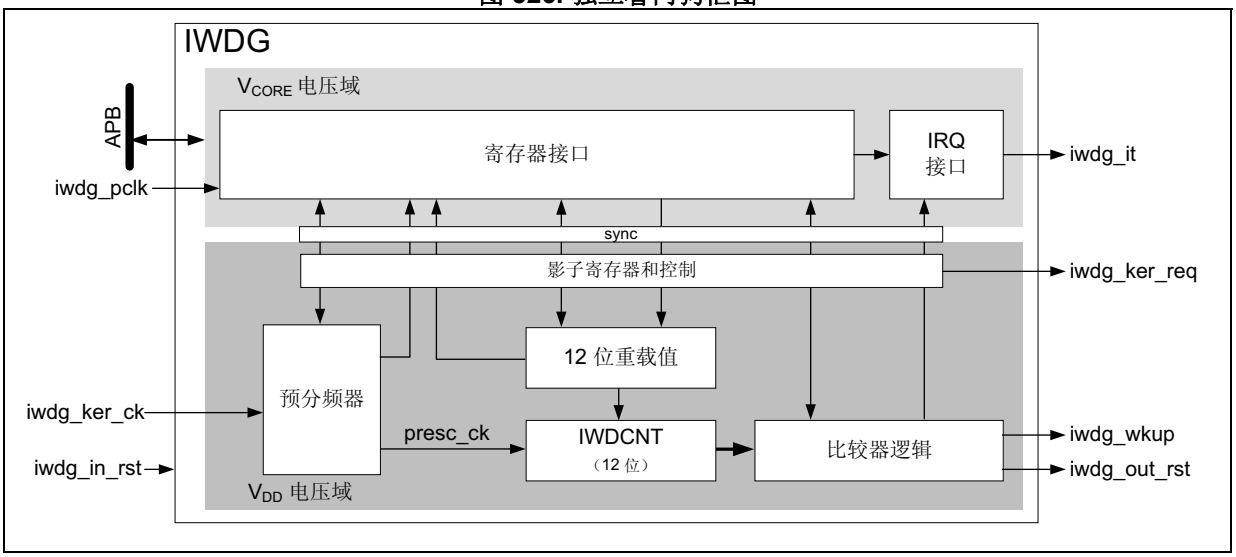
1. “X” = 支持，“-” = 不支持。
2. 有关详细信息，请参见 RCC 部分。
3. 由 DBG 模块中 DBGMCU_APB5FZ1 寄存器的 IWDG1 位控制。
4. 由 DBG 模块中 DBGMCU_APB4FZ1 寄存器的 IWDG2 位控制。
5. IWDG1 和 IWDG2 分别通过选项字节 OTP_IWDG1_FZ_STOP 和 OTP_IWDG2_FZ_STOP 控制。
6. IWDG1 和 IWDG2 分别通过选项字节 OTP_IWDG1_FZ_STANDBY 和 OTP_IWDG2_FZ_STANDBY 控制。
7. IWDG1 和 IWDG2 分别通过选项字节 OTP_IWDG1_HW 和 OTP_IWDG2_HW 控制。

45.4 IWDG 功能描述

45.4.1 IWDG 框图

图 526 给出了独立看门狗模块的功能框图。

图 526. 独立看门狗框图



寄存器和 IRQ 接口位于 V_{CORE} 电压域。看门狗功能本身位于 V_{DD} 电压域，在低功耗模式下仍然保持正常工作。有关 IWDG 功能，请参见*第 45.3 节*。

寄存器接口和 IRQ 接口的时钟主要由 APB 时钟 (iwdg_pclk) 提供，而看门狗功能的时钟则由专用内核时钟 (iwdg_ker_ck) 提供。可通过同步机制在两个电压域之间进行数据交换。请注意，寄存器接口中的大多数寄存器均映射到 V_{DD} 电压域中。

IWDG 递减计数器 (IWDCNT) 的时钟由预分频时钟 (presc_ck) 提供。预分频时钟是由内核时钟 iwdg_ker_ck 根据 PR[3:0] 位字段经预分频器分频后生成的。

45.4.2 IWDG 内部信号

表 391 详细列出了 IWDG 内部信号。

表 391. IWDG 内部输入/输出信号

信号名称	信号类型	说明
iwdg_ker_ck	输入	IWDG 内核时钟
iwdg_ker_req	输入	IWDG 内核时钟请求
iwdg_pclk	输入	IWDG APB 时钟
iwdg_out_rst	输出	IWDG 复位输出
iwdg_in_rst	输入	IWDG 复位输入
iwdg_wkup	输出	IWDG 唤醒事件
iwdg_it	输出	IWDG 提前唤醒中断

45.4.3 软件和硬件看门狗模式

这两种看门狗模式允许应用选择 IWDG 的使能方式：通过软件命令使能（软件看门狗模式）或自动使能（硬件看门狗模式）。软件和硬件看门狗模式对于所有其他功能的工作方式没有影响。

软件看门狗模式为默认工作模式。将值 0x0000 CCCC 写入 *IWDG 密钥寄存器 (IWDG_KR)* 可启动独立看门狗，IWDCNT 从复位值 0xFFF 开始递减计数。

在硬件看门狗模式下，独立看门狗在上电时或每次复位（通过 iwdg_in_rst）时自动启动。IWDCNT 递减计数器从复位值 0xFFF 开始递减计数。硬件看门狗模式功能通过设备选项位使能（有关详细信息，请参见*第 45.3 节*）。

使能 IWDG 时，ONF 标志置 1。

当 IWDCNT 计数到 0x000 时，会生成复位信号（iwdg_out_rst 置为有效）。

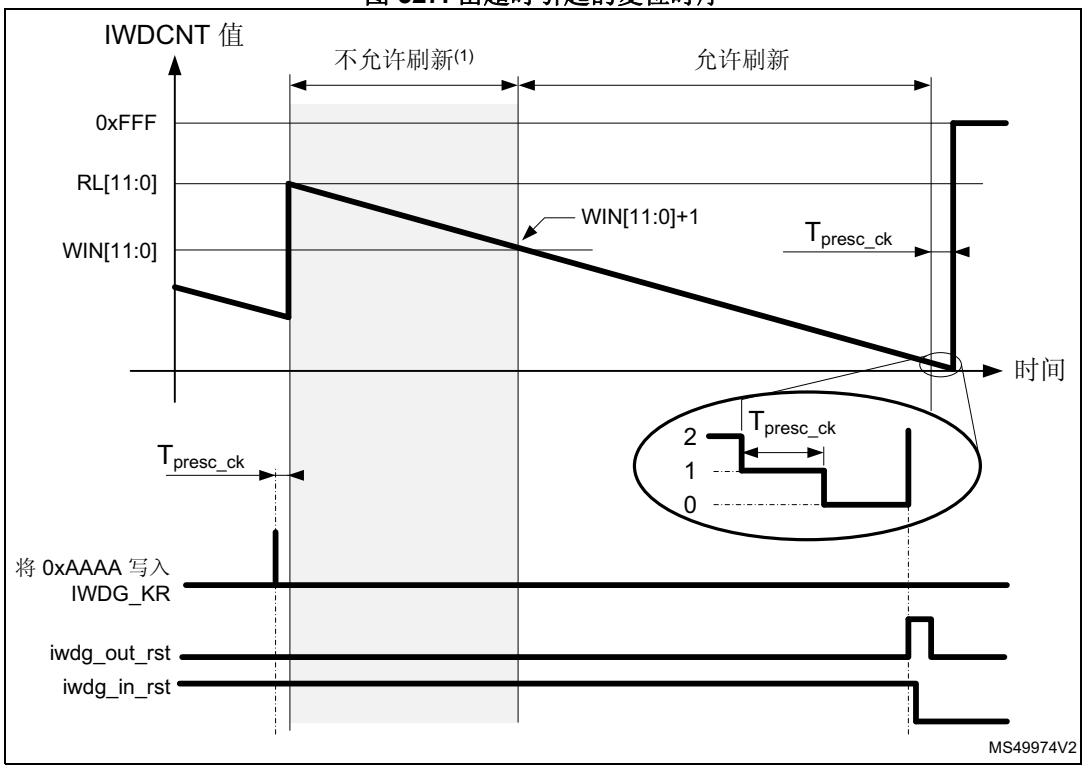
任何时候将键值 0x0000 AAAA 写入 *IWDG 密钥寄存器 (IWDG_KR)*，IWDG_RLR 的值都会被重载到 IWDCNT，从而避免产生看门狗复位。

由于存在再同步延迟，因此必须在 IWDCNT 递减计数器计数到 1 之前刷新 IWDG。

IWDG 一旦启动，便只有在复位（iwdg_in_rst 置为有效）时才能停止。

如*图 527* 所示，在执行刷新命令的一个 presc_ck 周期后，IWDCNT 会重新装载 RL[11:0] 的内容。

图 527. 由超时引起的复位时序



1. 激活窗口选项时。

如果在 IWDCNT 计数到 1 前未刷新 IWDG，IWDG 会产生复位（iwdg_out_rst 置为有效）。随后，RCC 会复位 IWDG（iwdg_in_rst 置为有效）以清除复位源。

45.4.4 窗口选项

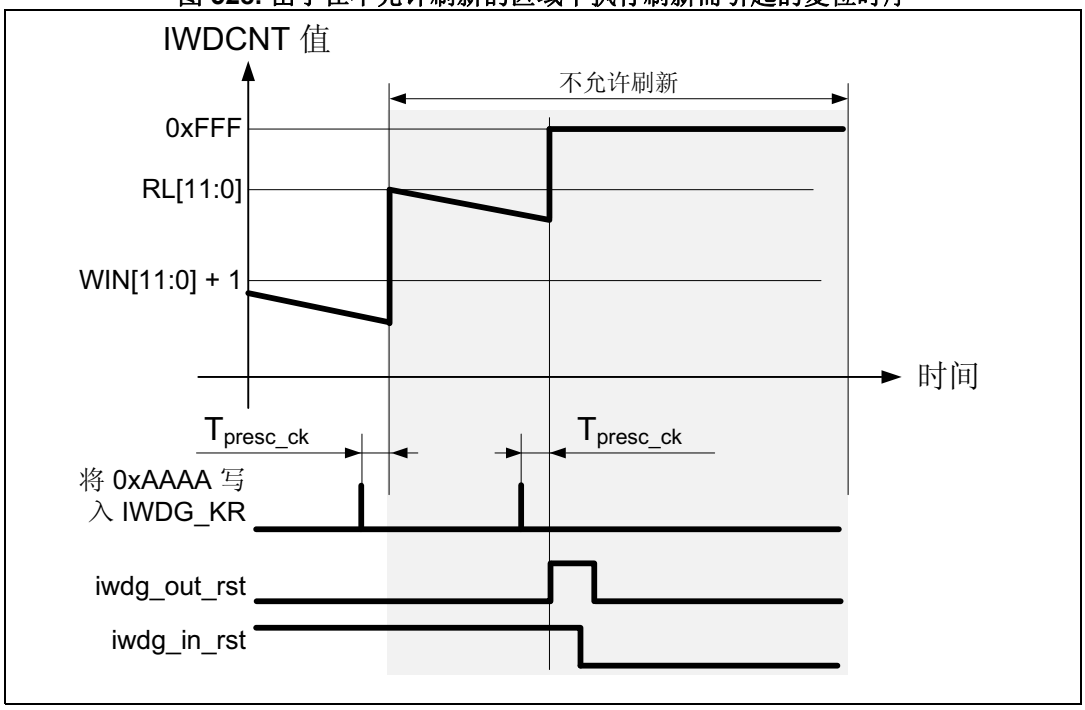
通过在 *IWDG 窗口寄存器 (IWDG_WINR)* 中设置合适的窗口，IWDG 也可以用作窗口看门狗。

当计数器值大于 WIN[11:0] + 1 时，如果执行重载操作，则会产生复位。WIN[11:0] 位于 *IWDG 窗口寄存器 (IWDG_WINR)* 中。如*图 528* 所示，复位信号在意外执行刷新命令的一个 presc_ck 周期后生成。

IWDG 窗口寄存器 (IWDG_WINR) 的默认值为 0x0000 0FFF，因此，如果不更新此默认值，将禁止窗口选项。

只要窗口值发生变化，递减计数器 (IWDCNT) 就会立即重载 RL[11:0] 的值，以便估计下一次执行刷新命令的位置。

图 528. 由于在不允许刷新的区域中执行刷新而引起的复位时序



使能窗口选项时配置 IWDG

1. 通过在 *IWDG 密钥寄存器 (IWDG_KR)* 中写入 0x0000 CCCC 来使能 IWDG。
2. 通过在 *IWDG 密钥寄存器 (IWDG_KR)* 中写入 0x0000 5555 来使能寄存器访问。
3. 通过编程 *IWDG 预分频器寄存器 (IWDG_PR)* 来配置 IWDG 预分频器。
4. 对 *IWDG 重载寄存器 (IWDG_RLR)* 进行写操作。
5. 如果需要，可以通过在 *IWDG 提前唤醒中断寄存器 (IWDG_EWCR)* 中写入适当的值来使能提前唤醒中断并编程提前唤醒比较器。
6. 对 *IWDG 窗口寄存器 (IWDG_WINR)* 进行写操作。这将自动使用 RL[11:0] 的值重载 IWDCNT 递减计数器。
7. 等待寄存器更新 (IWDG_SR = 0x0000 0000)。
8. 将 0x0000 0000 写入 *IWDG 密钥寄存器 (IWDG_KR)* 以对寄存器进行写保护。

注： *如果应用无需在此序列完成后禁止 APB 时钟，则可以跳过步骤 7。*

禁止窗口选项时配置 IWDG

不使用窗口选项时，可按以下步骤配置 IWDG：

1. 通过在 *IWDG 密钥寄存器 (IWDG_KR)* 中写入 0x0000 CCCC 来使能 IWDG。
2. 通过在 *IWDG 密钥寄存器 (IWDG_KR)* 中写入 0x0000 5555 来使能寄存器访问。
3. 通过编程 *IWDG 预分频器寄存器 (IWDG_PR)* 来配置预分频器。
4. 对 *IWDG 重载寄存器 (IWDG_RLR)* 进行写操作。
5. 如果需要，可以通过在 *IWDG 提前唤醒中断寄存器 (IWDG_EWCR)* 中写入适当的值来使能提前唤醒中断并编程提前唤醒比较器。
6. 等待寄存器更新 (IWDG_SR = 0x0000 0000)。
7. 使用 RL[11:0] 的值刷新计数器，然后将 0x0000 AAAA 写入 *IWDG 密钥寄存器 (IWDG_KR)* 以对寄存器进行写保护。

更新窗口比较器

窗口比较器可在 IWDG 运行期间更新。IWDCNT 也会重新装载。可执行以下序列来更新窗口比较器：

1. 通过在 IWDG 密钥寄存器 (IWDG_KR) 中写入 0x0000 5555 来使能寄存器访问。
2. 写入 IWDG 窗口寄存器 (IWDG_WINR)。这将自动使用 RL[11:0] 的值重载 IWDCNT 递减计数器。
3. 等待 WVU = 0
4. 通过向 IWDG_KR 中写入 0x0000 0000 来锁定寄存器

如果应用无需在此序列完成后禁止 APB 时钟，则可以跳过步骤 *3*。

45.4.5 调试

当处理器进入调试模式时（内核停止），IWDCNT 递减计数器会根据产品的调试功能选择继续正常工作或者停止工作。有关该产品功能的详细信息，请参见*第 45.3 节*。

45.4.6 寄存器访问保护

IWDG 预分频器寄存器 (IWDG_PR)、*IWDG 重载寄存器 (IWDG_RLR)*、*IWDG 提前唤醒中断寄存器 (IWDG_EWCR)* 和 *IWDG 窗口寄存器 (IWDG_WINR)* 具有写访问保护。要修改上述寄存器，首先向 *IWDG 密钥寄存器 (IWDG_KR)* 写入 0x0000 5555。而写入其他值则会破坏该序列，从而使寄存器访问保护再次生效。这表示重载操作（即写入 0x0000 AAAA）也会启动写保护功能。

状态寄存器指示预分频值、递减计数器重载值或窗口值是否正在被更新。

45.5 IWDG 低功耗模式

根据选项字节配置，IWDG 可在低功耗模式下继续计数或停止计数。有关详细信息，请参见*第 45.3 节*。

表 392. 低功耗模式对 IWDG 的影响

模式	说明
睡眠	无影响。IWDG 中断可使设备退出该模式。
Stop	IWDG 保持工作状态或不工作，具体取决于选项字节配置。 有关详细信息，请参见*第 45.3 节*。 IWDG 中断可使设备退出 Stop 模式模式。
待机	IWDG 保持工作状态或不工作，具体取决于选项字节配置。 有关详细信息，请参见*第 45.3 节*。 IWDG 中断不会使设备退出待机模式。
关断	IWDG 不工作。

45.6 IWDG 中断

IWDG 可根据递减计数器的值生成提前中断。将 *IWDG 提前唤醒中断寄存器 (IWDG_EWCR)* 寄存器中的 EWIE 位置 1 可使能提前中断。

应用可以使用比较器值 (EWIT[11:0]) 定义必须生成提前中断的位置。

当 IWDCNT 递减计数器计数到 EWIT[11:0] - 1 的值时，iwdg_wkup 会激活，系统可根据需要退出低功耗模式。

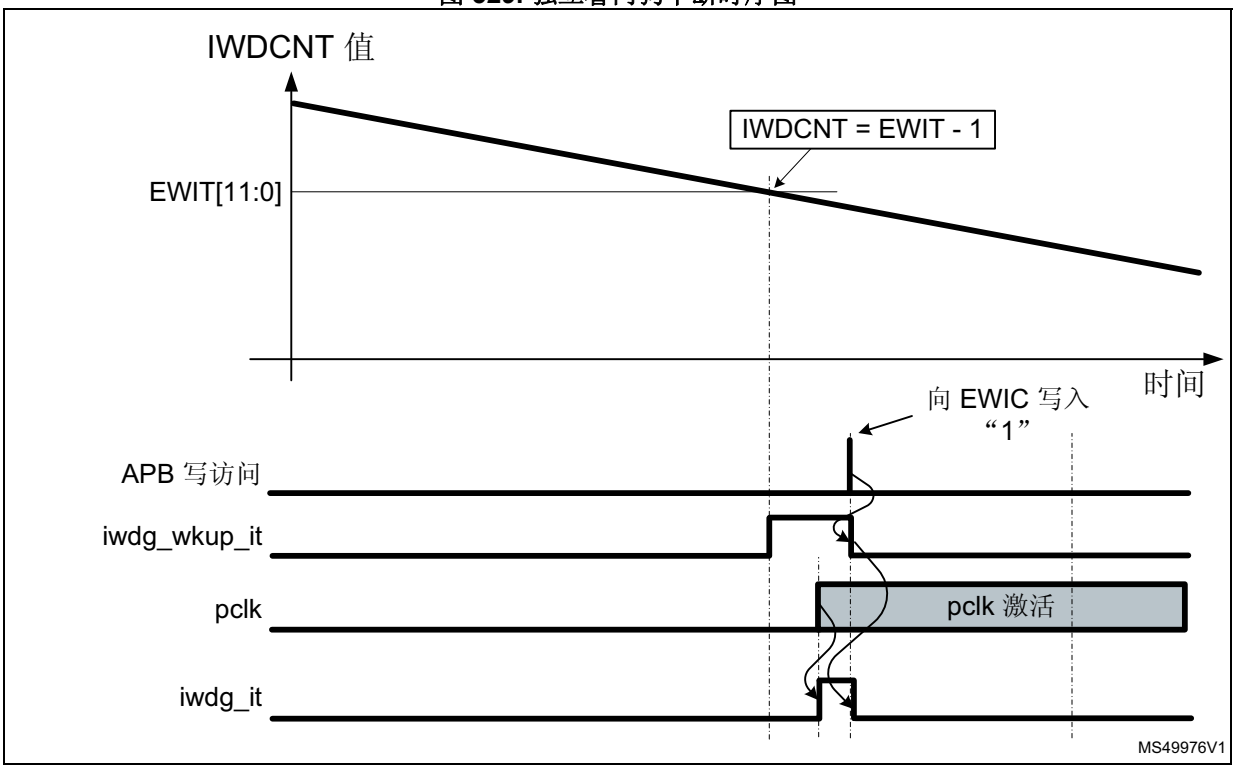
当 APB 时钟可用时，iwdg_it 也会激活。

此外，*IWDG 状态寄存器 (IWDG_SR)* 的 EWIF 标志置 1。

通过向 *IWDG 提前唤醒中断寄存器 (IWDG_EWCR)* 的 EWIC 位写入“1”来确认 EWI 中断。

对 IWDG_EWCR 寄存器的写操作还会触发将递减计数器 (IWDCNT) 刷新为重载值 RL[11:0]。

图 529. 独立看门狗中断时序图



如果在产生看门狗复位之前必须执行特定的安全操作或数据记录，则可使用提前唤醒中断 (EWI)。

更改提前唤醒比较器值

通过执行以下序列，可以随时更改提前唤醒比较器值或使能/禁止中断生成：

1. 通过在 *IWDG 密钥寄存器 (IWDG_KR)* 中写入 0x0000 5555 来使能寄存器访问。
2. 通过在 *IWDG 提前唤醒中断寄存器 (IWDG_EWCR)* 中写入适当的值来使能/禁止提前唤醒中断和/或编程提前唤醒比较器。
3. 等待 EWU =“0”，EWU 位于 *IWDG 状态寄存器 (IWDG_SR)* 中。
4. 将 0x0000 0000 写入 *IWDG 密钥寄存器 (IWDG_KR)* 以对寄存器进行写保护。

如果应用无需在此序列完成后禁止 APB 时钟，则可以跳过步骤 *3*。

表 393 汇总了 IWDG 中断请求。

表 393. IWDG 中断请求

中断事件	事件标志	中断清除方法	中断使能控制位	激活的中断	
				iwdg_it	iwdg_wkup_it
IWDCNT 计数到 EWIT 值	EWIF	向 EWIC 写入“1”	EWIE	Y(1)	Y(2)

1. 当 iwdg_pclk 输入上存在时钟时生成。
2. 当 iwdg_ker_ck 输入上存在时钟时生成。

45.7 IWDG 寄存器

有关寄存器说明中使用的缩写，请参见*第 109 页的第 1.2 节*。

外设寄存器可按半字（16 位）或字（32 位）访问。

寄存器接口中的大多数寄存器均映射到 V_{DD} 电压域中。当 iwdg_in_rst 置为有效时，位于 V_{DD} 电压域中的看门狗逻辑和影子寄存器将会复位。

当应用回读看门狗寄存器时，硬件会将相应影子寄存器的值传输到寄存器接口。

当应用写入看门狗寄存器时，硬件会更新相应的影子寄存器。

45.7.1 IWDG 密钥寄存器 (IWDG_KR)

IWDG key register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:16 保留，必须保持复位值。

位 15:0 **KEY[15:0]**：键值 (Key value)（只能写，读为 0x0000）

这些位可提供多种功能，具体取决于应用写入的值：

- 0xAAAA：将 RL[11:0] 值重载到 IWDCNT 递减计数器（看门狗刷新），并对寄存器进行写保护。必须每隔一段时间便通过软件写入该值，否则当计数器计数到 0 时，看门狗会产生复位。
- 0x5555：使能对寄存器的写访问。
- 0xCCCC：使能看门狗（选择硬件看门狗选项时除外）并对寄存器进行写保护。
- 除 0x5555 以外的其他值：对寄存器进行写保护。

请注意，只有 IWDG_PR、IWDG_RLR、IWDG_EWCR 和 IWDG_WINR 寄存器具有写保护机制。

45.7.2 IWDG 预分频器寄存器 (IWDG_PR)

IWDG prescaler register

偏移地址：0x04

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **PR[3:0]**：预分频系数 (Prescaler divider)

这些位受写访问保护，请参见*第 45.4.6 节*。通过软件设置这些位来选择计数器时钟的预分频系数。若要更改预分频系数，*IWDG 状态寄存器 (IWDG_SR)* 的 PVU 位必须复位。

0000：4 分频
0001：8 分频
0010：16 分频
0011：32 分频
0100：64 分频
0101：128 分频
0110：256 分频
0111：512 分频
其他值：1024 分频

注意：读取该寄存器会返回 V_{DD} 电压域的预分频器值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的/ 有效的。因此，只有在 IWDG 状态寄存器 (IWDG_SR) 中的 PVU 位复位时，从寄存器读取的值才有效。

45.7.3 IWDG 重载寄存器 (IWDG_RLR)

IWDG reload register

偏移地址：0x08

复位值：0x0000 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	RL[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:0 **RL[11:0]**：看门狗计数器重载值 (Watchdog counter reload value)

这些位受写访问保护，请参见*第 45.4.6 节*。这个值由软件设置，每次对 *IWDG 密钥寄存器 (IWDG_KR)* 写入值 0xAAAA 时，这个值就会重载到看门狗计数器中。之后，看门狗计数器便从该装载的值开始递减计数。超时周期由该值和预分频器时钟共同决定。不建议将 RL[11:0] 设置为小于 2 的值。

若要更改重载值，*IWDG 状态寄存器 (IWDG_SR)* 中的 RVU 位必须复位。

注意：读取该寄存器会返回 V_{DD} 电压域的重载值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的/有效的。因此，只有在 IWDG 状态寄存器 (IWDG_SR) 中的 RVU 位复位时，从寄存器读取的值才有效。

45.7.4 IWDG 状态寄存器 (IWDG_SR)

IWDG status register

偏移地址：0x0C

复位值：0x0000 0000 (0xFFFF FEFF)

该寄存器包含各种状态标志。请注意，括号内的掩码值表示 ONF 位的复位值尚未定义。当 IWDG 配置为软件模式时，ONF 位的复位值为 0；当 IWDG 配置为硬件模式时，ONF 位的复位值为 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	EWIF	Res.	Res.	Res.	Res.	Res.	ONF	Res.	Res.	Res.	Res.	EWU	WVU	RVU	PVU
	r						r					r	r	r	r

位 31:15 保留，必须保持复位值。

位 14 **EWIF**：看门狗提前中断标志 (Watchdog early interrupt flag)

该位由硬件置“1”以指示存在挂起的提前中断。该位必须由软件通过向 IWDG_EWCR 的 EWIC 位写入“1”来清零。

位 13:9 保留，必须保持复位值。

位 8 **ONF**：看门狗使能状态位 (Watchdog enable status bit)

IWDG 启动后，该位立即由硬件置“1”。在软件模式下，该位保持为“1”，直到 IWDG 复位为止。在硬件模式下，该位始终置“1”。

0：未激活 IWDG

1：已激活 IWDG，需要应用定期刷新

位 7:4 保留，必须保持复位值。

位 3 **EWU**：看门狗中断比较器值更新 (Watchdog interrupt comparator value update)

该位由硬件置 1 以指示中断比较器值 (EWIT[11:0]) 或 EWIE 正在更新。当在 V_{DD} 电压域下完成更新操作后（最多需要三个个 IWDG 内核时钟 (iwdg_ker_ck) 周期），会通过硬件将该位复位。

EWIT[11:0] 和 EWIE 位字段只有在 EWU 位复位时才可更新。

位 2 **WVU**：看门狗计数器窗口值更新 (Watchdog counter window value update)

该位由硬件置 1 以指示窗口值正在更新。当在 V_{DD} 电压域下完成重载值更新操作后（最多需要三个个 IWDG 内核时钟 (iwdg_ker_ck) 周期），会通过硬件将该位复位。

窗口值只有在 WVU 位复位时才可更新。

该位只有在通用“窗口”= 1 时才生成。

位 1 **RVU**：看门狗计数器重载值更新 (Watchdog counter reload value update)

该位由硬件置 1 以指示重载值正在更新。当在 V_{DD} 电压域下完成重载值更新操作后（最多需要三个个 IWDG 内核时钟 (iwdg_ker_ck) 周期），会通过硬件将该位复位。

重载值只有在 RVU 位复位时才可更新。

位 0 **PVU**：看门狗预分频器值更新 (Watchdog prescaler value update)

该位由硬件置 1 以指示预分频器值正在更新。当在 V_{DD} 电压域下完成预分频器更新操作后（最多需要三个个 IWDG 内核时钟 (iwdg_ker_ck) 周期），会通过硬件将该位复位。

预分频器值只有在 PVU 位复位时才可更新。

注： *如果应用使用多个重载值、预分频器值、提前中断位置值或窗口值，则必须等到 RVU 位复位后才能更改重载值，必须等到 PVU 位复位后才能更改预分频器值，必须等到 WVU 位复位后才能更改窗口值，而且必须等到 EWU 位复位后才能更改提前中断位置值。在更新预分频器和/或重载/窗口/提前中断值之后，无需等到 RVU、PVU、WVU 或 EWU 复位后再继续执行代码（进入低功耗模式时除外）。*

45.7.5 IWDG 窗口寄存器 (IWDG_WINR)

IWDG window register

偏移地址：0x10

复位值：0x0000 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	WIN[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:0 **WIN[11:0]**：看门狗计数器窗口值 (Watchdog counter window value)

这些位受写访问保护，请参见*第 45.4.6 节*。它们包含用于与递减计数器进行比较的窗口值上限。

为防止发生复位，当 IWDCNT 递减计数器的值小于 WIN[11:0] + 1 且大于 1 时必须重载。

若要更改重载值，*IWDG 状态寄存器 (IWDG_SR)* 中的 WVU 位必须复位。

注意：读取该寄存器会返回 V_{DD} 电压域的重载值。如果正在对该寄存器执行写操作，则读取的值可能无效。因此，只有在 IWDG 状态寄存器 (IWDG_SR) 中的 WVU 位复位时，从寄存器读取的值才有效。

45.7.6 IWDG 提前唤醒中断寄存器 (IWDG_EWCR)

IWDG early wake-up interrupt register

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EWIE	EWIC	Res.	Res.	EWIT[11:0]											
rw	w			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **EWIE**：看门狗提前中断使能 (Watchdog early interrupt enable)
由软件置 1 和复位。
0：禁止提前中断接口。
1：使能提前中断接口。
若要更改该位的值，*IWDG 状态寄存器 (IWDG_SR)* 中的 EWU 位必须复位。

位 14 **EWIC**：看门狗提前中断确认 (Watchdog early interrupt acknowledge)
必须由软件向该位写入 1，以确认提前唤醒中断并将 EWIF 标志清零。写入 0 无影响，读取该标志将返回 0。

位 13:12 保留，必须保持复位值。

位 11:0 **EWIT[11:0]**：看门狗计数器窗口值 (Watchdog counter window value)
这些位受写访问保护（请参见*第 45.4.6 节*）。通过软件写入这些位，可以定义在 IWDCNT 递减计数器计数到哪个位置时必须生成提前唤醒中断。当 IWDCNT 小于或等于 EWIT[11:0] - 1 时，生成提前中断。
EWIT[11:0] 必须大于 1。
只有在 EWIE = 1 时才会生成中断。
若要更改重载值，*IWDG 状态寄存器 (IWDG_SR)* 中的 EWU 位必须复位。
注意：读取该寄存器将返回 V_{DD} 电压域的提前唤醒比较器值和中断使能位值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的/有效的。因此，只有在 IWDG 状态寄存器 (IWDG_SR) 中的 EWU 位复位时，从寄存器读取的值才有效。

45.7.7 IWDG 硬件配置寄存器 (IWDG_HWCFGR)

IWDG hardware configuration register

偏移地址：0x3F0

复位值：0x0000 00F1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR_DEFAULT[3:0]				WINDOW[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **PR_DEFAULT[3:0]**：预分频器默认值

该位字段指定预分频器的默认值（请参见 *PR[3:0]：预分频系数 (Prescaler divider)* 的 *IWDG 预分频器寄存器 (IWDG_PR)*

0000：复位后预分频器 (PR[3:0]) 的默认值为 0000’
0001：复位后预分频器 (PR[3:0]) 的默认值为 0001’
0010：复位后预分频器 (PR[3:0]) 的默认值为 0010’
0011：复位后预分频器 (PR[3:0]) 的默认值为 0011’
0100：复位后预分频器 (PR[3:0]) 的默认值为 0100’
0101：复位后预分频器 (PR[3:0]) 的默认值为 0101’
0110：复位后预分频器 (PR[3:0]) 的默认值为 0110’
0111：复位后预分频器 (PR[3:0]) 的默认值为 0111’
1xxx：复位后预分频器 (PR[3:0]) 的默认值为 1000

位 3:0 **WINDOW[3:0]**：支持窗口功能

0000：禁止窗口选项
0001：使能窗口选项
其他值：保留

45.7.8 IWDG 版本寄存器 (IWDG_VERR)

IWDG version register

偏移地址：0x3F4

复位值：0x0000 0031

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：IWDG 主版本 (DMA major revision)

位 3:0 **MINREV[3:0]**：IWDG 次版本 (DMA minor revision)

45.7.9 IWDG 标识寄存器 (IWDG_IDR)

IWDG identification register

偏移地址：0x3F8

复位值：0x0012 0041

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：IWDG 标识符

45.7.10 IWDG 尺寸标识寄存器 (IWDG_SIDR)

IWDG size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：IWDG 大小标识符 (IWDG size identifier)
地址解码范围为 1 KB。

45.7.11 IWDG 寄存器映射

表 394. IWDG 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	IWDG_KR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	IWDG_PR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[3:0]			
	Reset value																													1	1	1	1
0x08	IWDG_RLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RL[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x0C	IWDG_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF	Res.	Res.	Res.	Res.	Res.	ONF	Res.	Res.	Res.	Res.	EWU	WVU	RVU	PVU
	Reset value																		0						x					0	0	0	0
0x10	IWDG_WINR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WIN[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x14	IWDG_EWCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIE	EWIC	Res.	Res.	EWIT[11:0]											
	Reset value																	0	0			0	0	0	0	0	0	0	0	0	0	0	0
0x3F0	**IWDG_HWCFGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR_DEFAULT[3:0]				WINDOW[3:0]			
	Reset value																									1	1	1	1	0	0	0	1
0x3F4	**IWDG_VERR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	0	1	1	0	0	0	1
0x3F8	**IWDG_IDR**	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
0x3FC	**IWDG_SIDR**	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

46 实时时钟 (RTC)

46.1 简介

实时时钟 (RTC) 提供用于管理所有低功耗模式的自动唤醒单元。

实时时钟 (RTC) 是一个独立的 BCD 定时器/计数器。RTC 提供具有可编程闹钟中断功能的日历时钟/日历。

无论设备状态如何（运行模式、低功耗模式或处于复位状态），只要电源电压保持在工作范围内，RTC 便不会停止工作。

RTC 在 V_{BAT} 模式下正常工作。

46.2 RTC 主要特性

RTC 支持以下特性（请参见*图 530：RTC 框图*）：

- 日历具有亚秒、秒、分、小时（12 或 24 格式）、星期几、日、月、年，格式为 BCD（二进码十进数）。
- 可实现 32 位自由运行计数器的二进制模式。
- 自动调整每月是 28、29（闰年）、30 还是 31 天。
- 两个可编程闹钟。
- 可运行时纠正 1 到 32767 个 RTC 时钟脉冲。这可用于与主时钟同步。
- 参考时钟检测：可使用更加精确的第二时钟源（50 或 60 Hz）来提高日历的精确度。
- 数字校准电路具有 0.95 ppm 的分辨率，以补偿晶振的不准确性。
- 时间戳特性可用于保存日历内容。该功能可由时间戳引脚上的事件触发，或由 Tamper 事件触发，也可以通过切换到 V_{BAT} 模式来触发。
- 17 位自动重载唤醒定时器 (WUT)，适用于分辨率和周期可编程的周期性事件。
- TrustZone 支持：
  - RTC 安全特性可配置
  - 闹钟 A、闹钟 B、唤醒定时器和时间戳各自具有独立的安全或非安全配置
- 闹钟 A、闹钟 B、唤醒定时器和时间戳各自具有独立的特权保护

RTC 通过开关供电，当 V_{DD} 电源存在时，该开关选择 V_{DD} 供电，否则选择由 VBAT 引脚供电。

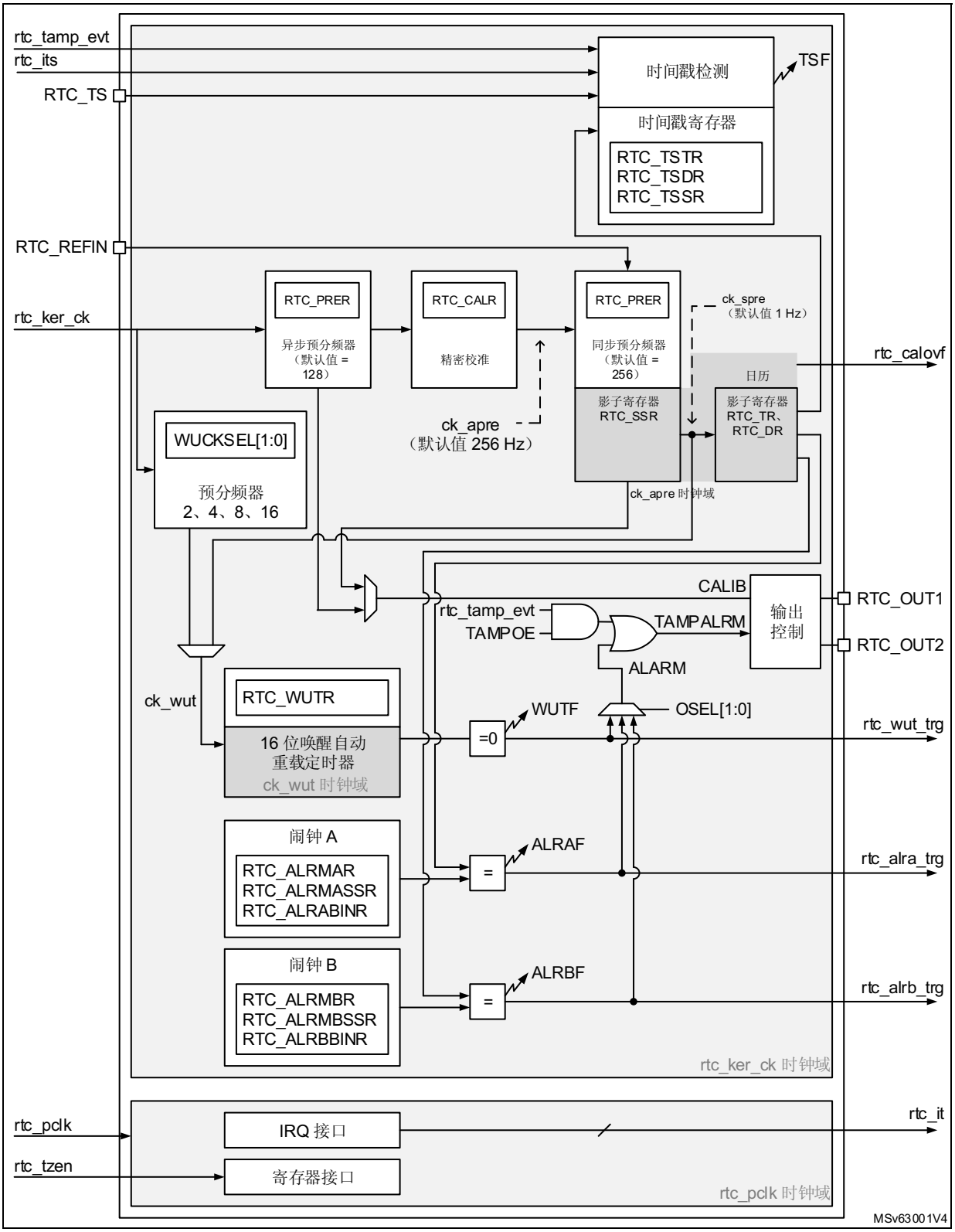
当 RTC 的时钟由 LSE 提供时，RTC 可以在 V_{BAT} 模式和所有低功耗模式下正常工作。

所有 RTC 事件（闹钟、唤醒定时器、时间戳）都可以产生中断，将设备从低功耗模式唤醒。

46.3 RTC 功能描述

46.3.1 RTC 框图

图 530. RTC 框图


rtc_tamp_evt
rtc_its
RTC_TS
时间戳检测
TSF
时间戳寄存器
RTC_TSTR
RTC_TSDR
RTC_TSSR
RTC_REFIN
rtc_ker_ck
RTC_PRER
异步预分频器
（默认值 = 128）
RTC_CALR
精密校准
RTC_PRER
同步预分频器
（默认值 = 256）
ck_spre
（默认值 1 Hz）
rtc_calovf
日历
影子寄存器
RTC_SSR
影子寄存器
RTC_TR、
RTC_DR
WUCKSEL[1:0]
预分频器
2、4、8、16
ck_apre
（默认值 256 Hz）
ck_apre 时钟域
CALIB
rtc_tamp_evt
TAMPOE
TAMPALRM
输出
控制
RTC_OUT1
RTC_OUT2
ALARM
ck_wut
RTC_WUTR
16 位唤醒自动
重载定时器
ck_wut 时钟域
=0
WUTF
OSEL[1:0]
rtc_wut_trg
闹钟 A
RTC_ALRMAR
RTC_ALRMASSR
RTC_ALRABINR
=
ALRAF
rtc_alra_trg
闹钟 B
RTC_ALRMBR
RTC_ALRMBSSR
RTC_ALRBBINR
=
ALRBF
rtc_alrb_trg
rtc_ker_ck 时钟域
rtc_pclk
rtc_tzen
IRQ 接口
寄存器接口
rtc_it
rtc_pclk 时钟域
MSv63001V4

46.3.2 RTC 引脚和内部信号

表 395. RTC 输入/输出引脚

引脚名称	信号类型	说明
RTC_TS	输入	RTC 时间戳输入
RTC_REFIN	输入	RTC 50 Hz 或 60 Hz 参考时钟输入
RTC_OUT1	输出	RTC 输出 1
RTC_OUT2	输出	RTC 输出 2

RTC_OUT1 和 RTC_OUT2 可选择以下两个输出之一：

- CALIB：512 Hz 或 1 Hz 时钟输出（LSE 频率为 32.768 kHz）。可通过将 RTC_CR 寄存器中的 COE 位置 1 来使能此输出。
- TAMPALRM：该输出是 rtc_tamp_evt 与 ALARM 信号的逻辑或运算结果。

ALARM 通过配置 RTC_CR 寄存器中的 OSEL[1:0] 位（可选择闹钟 A、闹钟 B 或唤醒输出）来使能。rtc_tamp_evt 通过将 RTC_CR 寄存器中的 TAMPOE 位置 1（将选择 Tamper 事件输出）来使能。

表 396. RTC 内部输入/输出信号

内部信号名称	信号类型	说明
rtc_ker_ck	输入	RTC 内核时钟，在本文档中也称为 RTCCLK
rtc_pclk	输入	RTC APB 时钟
rtc_its	输入	RTC 内部时间戳事件
rtc_tamp_evt	输入	在 TAMP 外设中检测到的 Tamper 事件（内部或外部）
rtc_tzen	输入	使能 RTC TrustZone
rtc_it	输出	RTC 中断（有关详细信息，请参见*第 46.5 节：RTC 中断*）
rtc_alra_trg	输出	RTC 闹钟 A 事件检测触发
rtc_alrb_trg	输出	RTC 闹钟 B 事件检测触发
rtc_wut_trg	输出	RTC 唤醒定时器事件检测触发
rtc_calovf	输出	RTC 日历上溢：当 RTC 日历达到其最大值时（99 年 12 月 31 日 23:59:59），会生成该信号。随后，日历将被冻结，不会上溢。

RTC 内核时钟通常是 32.768 kHz 的 LSE，但也可以在 RCC 中选择其他时钟源（更多详细信息，请参见 RCC）。当所选时钟不是 LSE 时，一些功能在某些低功耗模式或 V_{BAT} 模式下不可用。更多详细信息，请参见*第 46.4 节：RTC 低功耗模式*。

表 397. RTC 互连

信号名称	源/目标
rtc_its	从电源控制器 (PWR)：主电源丢失/切换到 V_{BAT} 检测输出
rtc_tamp_evt	从 TAMP 外设：tamp_evt
rtc_tzen	连接到 1
rtc_calovf	到 TAMP 外设：tamp_itamp5

触发输出可以用作其他外设的触发信号。

46.3.3 RTC 和 TAMP 控制的 GPIO

电池备份域 (V_{BAT}) 中包含的 GPIO 由使用这些 I/O 提供功能的外设直接控制（无论 GPIO 的配置如何）。

RTC_OUT1、RTC_LSO、RTC_TS、TAMP_IN1 和 TAMP_OUT2 映射到同一引脚 (PC13)。映射在 PC13 上的 RTC_LSCO、RTC 和 TAMP 功能可用于所有低功耗模式和 V_{BAT} 模式。

输出机制遵循*表 398* 中所示的优先级顺序。

表 398. RTC 引脚 PC13 配置(1)

PC13 引脚功能	OSEL[1:0]（ALARM 输出使能）	TAMPOE（TAMPER 输出使能）	COE（CALIB 输出使能）	OUT2EN	TAMPALRM_TYPE	TAMPALRM_PU	TAMP2E=TAMP2AM=1 (ATOSHARE=0) 或 TAMPxE=TAMPxAM=1（ATOSHARE=1 且 ATOSELx=1）	TAMP1E 和 IN1_RMP（TAMP_IN1 输入使能）	TSE（RTC_TS 输入使能）	LSCOEN[1:0]
TAMPALRM 输出推挽	01 或 10 或 11	0	无关	无关	0	0	无关	无关	无关	无关
	00	1								
	01 或 10 或 11	1								

表 398. RTC 引脚 PC13 配置(1)（续）

PC13 引脚功能		OSEL[1:0]（ALARM 输出使能）	TAMPOE（TAMPER 输出使能）	COE（CALIB 输出使能）	OUT2EN	TAMPALRM_TYPE	TAMPALRM_PU	TAMP2E=TAMP2AM=1 (ATOSHARE=0) 或 TAMPxE=TAMPxAM=1（ATOSHARE=1 且 ATOSELx=1）	TAMP1E 和 IN1_RMP（TAMP_IN1 输入使能）	TSE（RTC_TS 输入使能）	LSCOEN[1:0]
TAMPALRM 输出开漏(2)	无上下拉	01 或 10 或 11	0	无关	无关	1	0	无关	无关	无关	无关
		00	1								
		01 或 10 或 11	1								
	内部上拉	01 或 10 或 11	0	无关	无关	1	1	无关	无关	无关	无关
		00	1								
		01 或 10 或 11	1								
CALIB 输出 PP		00	0	1	0	无关	无关	无关	无关	无关	无关
TAMP_OUT2 输出 PP		00	0	0	0	无关	无关	1	无关	无关	无关
TAMP_IN1 输入浮空		00	0	0	无关	无关	无关	0	1	0	无关
		00	0	1	1						
		无关	无关	0							
RTC_TS 和 TAMP_IN1 输入浮空		00	0	0	无关	无关	无关	0	1	1	无关
		00	0	1	1			0			
		无关	无关	0				0			
RTC_TS 输入浮空		00	0	0	无关	无关	无关	0	0	1	无关
		00	0	1	1			0			
		无关	无关	0				0			

表 398. RTC 引脚 PC13 配置(1)（续）

PC13 引脚功能	OSEL[1:0]（ALARM 输出使能）	TAMPOE（TAMPER 输出使能）	COE（CALIB 输出使能）	OUT2EN	TAMPALRM_TYPE	TAMPALRM_PU	TAMP2E=TAMP2AM=1 (ATOSHARE=0) 或 TAMPxE=TAMPxAM=1（ATOSHARE=1 且 ATOSELx=1）	TAMP1E 和 IN1_RMP（TAMP_IN1 输入使能）	TSE（RTC_TS 输入使能）	LSCOEN[1:0]
唤醒引脚	00	0	0	无关	无关	无关	0	0	0	无关
	00	0	1	1			0			
	无关	无关	0				0			
LSCO LSE 时钟输出	00	0	0	无关	无关	无关	0	0	0	01
	00	0	1	1			0			
	无关	无关	0				0			
标准 GPIO	00	0	0	无关	无关	无关	0	0	0	00 或 10 或 11
	00	0	1	1			0			
	无关	无关	0				0			

1. OD：开漏；PP：推挽。
2. 在此配置下，GPIO 必须配置为输入。

RTC_OUT2、TAMP_IN2、TAMP_OUT3 和 RTC_LSCO 映射在同一引脚 (PI1) 上。无论 PI1 的 GPIO 配置如何，PI1 引脚的配置都由 RTC 控制。映射在 PI1 上的 RTC 或 TAMP 功能可用于所有低功耗模式和 VBAT 模式。

输出机制遵循*表 399* 中所示的优先级顺序。

表 399. PI1 配置

PI1 引脚功能		OSEL[1:0]（ALARM 输出使能）	TAMPOE（TAMPER 输出使能）	COE（CALIB 输出使能）	OUT2EN 和 OUT2_RMP	TAMPALRM_TYPE	TAMPALRM_PU	TAMP3E=TAMP3AM=1 (ATOSHARE=0) 或 TAMPxE=TAMPxAM=1（ATOSHARE=1 且 ATOSELx=2）	TAMP2E 和 IN2_RMP	LSCOEN[1:0]
TAMPALRM 输出推挽		01 或 10 或 11	0	无关	1	0	0	无关	无关	无关
		00	1							
		01 或 10 或 11	1							
TAMPALRM 输出开漏	无上下拉	01 或 10 或 11	0	无关	1	1	0	无关	无关	无关
		00	1							
		01 或 10 或 11	1							
	内部上拉	01 或 10 或 11	0	无关	1	1	1	无关	无关	无关
		00	1							
		01 或 10 或 11	1							
CALIB 输出 PP		00	0	1	1	无关	无关	无关	无关	无关
TAMP_OUT3 输出 PP		00	0	0	0	无关	无关	1	无关	无关
TAMP_IN2 输入浮空		00	0	0	0	无关	无关	0	1	无关
唤醒引脚		00	0	0	0	无关	无关	0	0	无关
LSCO LSE 时钟输出		00	0	0	0	无关	无关	0	0	10
标准 GPIO		00	0	0	0	无关	无关	0	0	00、01 或 11

此外，可借助 OUT2EN 位在 PB2 或 PI1 引脚上输出 RTC_OUT2。可以将不同的功能映射到 RTC_OUT1 或 RTC_OUT2 上，具体取决于 OSEL、COE 和 OUT2EN 的配置，如*表 400*所示。

表 400. RTC_OUT 映射

OSEL[1:0] 位 ALARM 输出使能）	COE 位（CALIB 输出使能）	OUT2EN 位	RTC_OUT1 映射到 PC13	RTC_OUT2 映射到 PB2 或 PI1
00	0	0	-	-
00	1		CALIB	-
01 或 10 或 11	无关		TAMPALRM	-
00	0	1	-	-
00	1		-	CALIB
01 或 10 或 11	0		-	TAMPALRM
01 或 10 或 11	1		TAMPALRM	CALIB

46.3.4 RTC 安全保护模式

默认情况下，备份域上电复位后，除了 RTC 安全配置寄存器 (RTC_SECCFGR) 只能在安全模式下写入之外，所有其他 RTC 寄存器都可以在安全模式和非安全模式下读取或写入。RTC 保护配置不受系统复位影响。

当 RTC_SECCFGR 寄存器中的 SEC 位置 1 时：

- 只有在安全模式下才能写入 RTC 寄存器。
- 始终可以在安全模式和非安全模式下读取RTC_SECCFGR、RTC_PRIVCFGR、RTC_MISR、RTC_TR、RTC_DR、RTC_SSR、RTC_PRER 和 RTC_CALR。所有其他 RTC 寄存器只能在安全模式下读取。

当 SEC 位清零时，仍然可以通过将专用控制位 INITSEC、CALSEC、TSSEC、WUTSEC、ALRASEC 或 ALRBSEC 置 1 来保护某些寄存器。如果所有这些位也都置 1，则所有 RTC 寄存器都可以在安全模式和非安全模式下读取或写入。

- 当 INITSEC 置 1 时：
  - RTC_TR、RTC_DR、RTC_PRER 寄存器，RTC_ICSR 寄存器中的 INIT、BIN 和 BCDU，RTC_CR 寄存器中的 FMT 控制位以及 RTC_PRIVCFGR 中的 INITPRIV 只能在安全模式下写入。
  - 这些寄存器和控制位可以在安全模式和非安全模式下读取。
- 当 CALSEC 置 1 时：
  - RTC_SHIFTR 和 RTC_CALR 寄存器，RTC_CR 寄存器中的 ADD1H、SUB1H 和 REFCKON 控制位，以及 RTC_PRIVCFGR 中的 CALPRIV 只能在安全模式下写入。
  - 这些寄存器和控制位可以在安全模式和非安全模式下读取。
- 当 ALRASEC 置 1 时：
  - RTC_ALRMAR、RTC_ALRMASSR 和 RTC_ALRABINR 寄存器，RTC_CR 中的 ALRAE、ALRAFCLR 和 ALRAIE，RTC_SCR 中的 CALRAF，RTC_SR 中的 ALRAF，以及 RTC_SMISR 中的 ALRAMF，都只能在安全模式下读写。
  - RTC_PRIVCFGR 中的 ALRAPRIV 只能在安全模式下写入

- 当 ALRBSEC 置 1 时：
  - RTC_ALRMBR、RTC_ALRMBSSR 和 RTC_ALRBBINR 寄存器，RTC_CR 寄存器中的 ALRBE ALRBFCLR 和 ALRBIE，RTC_SCR 寄存器中的 CALRBF，RTC_SR 寄存器中的 ALRBF 以及 RTC_SMISR 寄存器中的 ALRBMF 只能在安全模式下读取和写入。
  - RTC_PRIVCFGR 中的 ALRBPRIV 只能在安全模式下写入。
- 当 WUTSEC 置 1 时：
  - RTC_WUTR 寄存器，RTC_CR 寄存器中的 WUTE、WUTIE 和 WUCKSEL 控制位，RTC_SCR 寄存器中的 CWUTF，RTC_SR 寄存器中的 WUTF，以及 RTC_SMISR 寄存器中的 WUTMF 只能在特权模式下读取和写入。
  - RTC_PRIVCFGR 中的 WUTPRIV 只能在安全模式下写入
- 当 TSSEC 置 1 时：
  - RTC_TSTR 和 RTC_TSDR 和 RTC_TSSSR 寄存器，RTC_CR 寄存器中的 TAMPTS、ITSE、TSE、TSIE、TSEDGE 控制位，RTC_SCR 寄存器中的 CITSF、CTSOVF 和 CTSF，RTC_SR 寄存器中的 TSF、TSOVF 和 ITSF，以及 RTC_SMISR 寄存器中的 TSMF、TSOVMF 和 ITSMF 只能在安全模式下读取和写入。
  - RTC_PRIVCFGR 中的 TSPRIV 只能在安全模式下写入

对受到安全保护的寄存器进行非安全访问会遭到拒绝：

- 不生成任何总线错误。
- 如果寄存器具有全局保护：通过 TZIC（TrustZone 非法访问控制器）中的标志/中断生成通知。如果寄存器中只有少数几个位受到保护（对于 RTC_CR 等具有混合功能的寄存器），则不会生成通知。
- 写保护时，这些位不会被写入。
- 读保护时，这些位的值读为 0。

只要至少一个功能配置为安全，RCC 中的 RTC 复位和时钟控制也就安全。

46.3.5 RTC 特权保护模式

默认情况下，备份域上电复位后，除了 RTC 特权模式控制寄存器 (RTC_PRIVCFGR) 只能在特权模式下写入之外，所有其他 RTC 寄存器都可以在特权模式和非特权模式下读取或写入。RTC 保护配置不受系统复位影响。

当 RTC_PRIVCFGR 寄存器中的 PRIV 位置 1 时：

- 只能在特权模式下写入 RTC 寄存器。
- 始终可以在特权模式和非特权模式下读取 RTC_SECCFGR、RTC_PRIVCFGR、RTC_TR、RTC_DR、RTC_SSR、RTC_PRER 和 RTC_CALR。所有其他 RTC 寄存器都只能在特权模式下读取。

当 PRIV 位清零时，仍然可以通过将专用控制位 INITPRIV、CALPRIV、TSPRIV、WUTPRIV、ALRAPRV 或 ALRBPRIV 置 1 来保护某些寄存器。如果所有这些位也都清零，则所有 RTC 寄存器都可以在特权模式和非特权模式下读取或写入。

- 当 INITPRIV 置 1 时：
  - RTC_TR、RTC_DR、RTC_PRER 寄存器，RTC_ICSR 中的 INIT、BIN 和 BCDU，RTC_CR 寄存器中的 RTC_ICSR 和 FMT 控制位，以及 RTC_SECCFGR 中的 INITSEC 只能在特权模式下写入。
  - 这些寄存器和控制位可以在特权模式和非特权模式下读取。

- 当 CALPRIV 置 1 时：
  - RTC_SHIFTR 和 RTC_CALR 寄存器，RTC_CR 寄存器中的 ADD1H、SUB1H 和 REFCKON 控制位，以及 RTC_SECCFGR 中的 CALDSEC 只能在特权模式下写入。
  - 这些寄存器和控制位可以在特权模式和非特权模式下读取。
- 当 ALRAPRIV 置 1 时：
  - RTC_ALRMAR、RTC_ALRMASSR 和 RTC_ALRABINR 寄存器，RTC_CR 中的 ALRAE、ALRAFCLR 和 ALRAIE，RTC_SCR 中的 CALRAF，RTC_SR 中的 ALRAF、RTC_MISR 中的 ALRAMF 以及 RTC_SMISR 寄存器只能在特权模式下读取和写入。
  - RTC_SECCFGR 中的 ALRASEC 只能在特权模式下写入。
- 当 ALRBPRIV 置 1 时：
  - RTC_ALRMBR、RTC_ALRMBSSR 和 RTC_ALRBBINR 寄存器，RTC_CR 寄存器中的 ALRBE ALRBFCLR 和 ALRBIE，RTC_SCR 寄存器中的 CALRBF，RTC_SR 寄存器中的 ALRBF 以及 RTC_MISR 寄存器中的 ALRBMF 以及 RTC_SMISR 寄存器只能在特权模式下读取和写入。
  - RTC_SECCFGR 中的 ALRBSEC 只能在特权模式下写入。
- 当 WUTPRIV 置 1 时：
  - RTC_WUTR 寄存器，RTC_CR 寄存器中的 WUTE、WUTIE 和 WUCKSEL 控制位，RTC_SCR 寄存器中的 CWUTF、RTC_SR 寄存器中的 WUTF，RTC_MISR 寄存器中的 WUTMF，以及 RTC_SMISR 寄存器只能在特权模式下读取和写入。
  - RTC_SECCFGR 中的 WUTSEC 只能在特权模式下写入。
- 当 TSPRIV 置 1 时：
  - RTC_TSTR 和 RTC_TSDR 和 RTC_TSSSR 寄存器，RTC_CR 寄存器中的 TAMPTS、ITSE、TSE、TSIE 和 TSEDGE 控制位，RTC_SCR 寄存器中的 CITSF、CTSOVF 和 CTSF 位，RTC_SR 寄存器中的 TSF、TSOVF 和 ITSF，RTC_MISR 寄存器中的 TSMF、TSOVMF 和 ITSMF，以及 RTC_SMISR 寄存器只能在特权模式下读取和写入
  - RTC_SECCFGR 中的 TSSEC 只能在特权模式下写入。

对受特权保护的寄存器的非特权访问将被拒绝：

- 不生成任何总线错误。
- 写保护时，这些位不会被写入。
- 读保护时，这些位的值读为 0。

46.3.6 时钟和预分频器

有关 RTC 时钟 (RTCCLK) 源配置的更多信息，请参见“复位和时钟控制 (RCC)”。

BCD 模式 (BIN=00)

可编程的预分频器阶段可生成 1 Hz 的时钟，用于更新日历。为最大程度地降低功耗，预分频器分为 2 个可编程的预分频器（参见*图 530：RTC 框图*）：

- 一个通过 RTC_PRER 寄存器的 PREDIV_A 位配置的 7 位异步预分频器。
- 一个通过 RTC_PRER 寄存器的 PREDIV_S 位配置的 15 位同步预分频器。

注： *使用两个预分频器时，推荐将异步预分频器配置为较高的值，以最大程度降低功耗。*

要使用频率为 32.768 kHz 的 LSE 获得频率为 1 Hz 的内部时钟 (ck_spre)，需要将异步预分频系数设置为 128，并将同步预分频系数设置为 256。

分频系数的最小值为 1，最大值为 2^{22}。

这对应于约为 4 MHz 的最大输入频率。

f_{ck_apre} 可根据以下公式得出：

$$f_{CK_APRE} = \frac{f_{RTCCLK}}{PREDIV_A + 1}$$

ck_apre 时钟用于为二进制 RTC_SSR 亚秒递减计数器提供时钟。当该计数器计数到 0 时，会使用 PREDIV_S 的内容重载 RTC_SSR。

f_{ck_spre} 可根据以下公式得出：

$$f_{CK_SPRE} = \frac{f_{RTCCLK}}{(PREDIV_S + 1) \times (PREDIV_A + 1)}$$

ck_spre 时钟既可以用于更新日历，也可以用作 16 位唤醒自动重载定时器的时基。为获得较短的超时周期，16 位唤醒自动重载计时器也可以使用由可编程 4 位异步预分频器分频的 RTCCLK 运行（有关详细信息，请参见*第 46.3.10 节：周期性自动唤醒*）。

二进制模式 (BIN=01)

SSR 二进制递减计数器扩展到 32 位长度并自由运行。时间和日期日历 BCD 寄存器不可用。

该递减计数器的时钟由 ck_apre 提供：通过 RTC_PRER 寄存器的 PREDIV_A 位配置的 7 位异步预分频器的输出。

PREDIV_S 值无意义。

混合模式（BIN=10 或 11）

SSR 二进制递减计数器扩展到 32 位长度并自由运行。时间和日期日历 BCD 寄存器也可用。

该递减计数器的时钟由 ck_apre 提供：通过 RTC_PRER 寄存器的 PREDIV_A 位配置的 7 位异步预分频器的输出。位 BCDU[2:0] 用于定义日历何时使用 SSR 最低有效位递增 1 秒。

46.3.7 实时时钟和日历

RTC 日历时间和日期寄存器可通过与 PCLK（APB 时钟）同步的影子寄存器来访问。这些时间和日期寄存器也可以直接访问，无同步时间。

- RTC_SSR 对应于亚秒
- RTC_TR 对应于时间
- RTC_DR 对应于日期

每个 RTCCLK 周期，都将当前日历值复制到影子寄存器，并将 RTC_ICSR 寄存器的 RSF 位置 1（请参见*第 46.6.12 节：RTC 移位控制寄存器 (RTC_SHIFTR)*）。在 Stop 和待机模式下不会执行复制操作。退出这两种模式时，影子寄存器会在最长 4 个 RTCCLK 周期后进行更新。

当应用读取日历寄存器时，它会访问影子寄存器的内容。也可以通过将 RTC_CR 寄存器的 BYPSHAD 控制位置 1 来直接访问日历寄存器。默认情况下，该位被清零，用户访问影子寄存器。

在 BYPSHAD=0 模式下读取 RTC_SSR、RTC_TR 或 RTC_DR 寄存器时，APB 时钟频率 (f_{APB}) 必须至少为 RTC 时钟频率 (f_{RTCCLK}) 的 7 倍。

影子寄存器通过系统复位来复位。

46.3.8 日历超低功耗模式

将 RTC_CALR 寄存器中的 LPCAL 位置 1，可以大幅降低 RTC 功耗。在这种配置下，整个 RTC 仅由 ck_apre 提供时钟，而不是同时由 RTCCLK 和 ck_apre 提供时钟。因此，一些标志的延迟较长，校准窗口也较长（请参见 *RTC 超低功耗模式*一节）。

当异步预分频器的分频系数 (PREDIV_A+1) 不是 2 的幂数时，会忽略 LPCAL 位（假定为 0）。

从 LPCAL=0 切换到 LPCAL=1 或从 LPCAL=1 切换到 LPCAL=0 的操作不会立即生效，需要几个 ck_apre 周期才能完成。

46.3.9 可编程闹钟

RTC 单元提供两个可编程闹钟，即闹钟 A 和闹钟 B。以下说明针对闹钟 A，但同样适用于闹钟 B。

可通过 RTC_CR 寄存器中的 ALRAE 位来使能可编程闹钟功能。

如果日历亚秒、秒、分钟、小时、日期或日分别与闹钟寄存器 RTC_ALRMASSR 和 RTC_ALRMAR 中编程的值相匹配，则 ALRAF 标志会被置为 1。可通过 RTC_ALRMAR 寄存器的 MSKx 位以及 RTC_ALRMASSR 寄存器的 MASKSSx 位单独选择各日历字段。

使用二进制模式时，可以在闹钟二进制寄存器 RTC_ALRABINR 中编程亚秒字段。

可通过 RTC_CR 寄存器中的 ALRAIE 位来使能闹钟中断。

如果闹钟用于为其他外设生成触发事件，则可通过将 RTC_CR 寄存器中的 ALRAFCLR 位置 1 使 ALRAF 由硬件自动清零。在这种配置下，如果只需将 ALRAF 标志清零，则无需任何软件干预。

小心： 如果选择秒字段（RTC_ALRMAR 中的 MSK1 位复位），则 RTC_PRER 寄存器中设置的同步预分频器分频系数必须至少为 3，才能确保闹钟正确地运行。

闹钟 A 和闹钟 B（如果已通过 RTC_CR 寄存器中的位 OSEL[1:0] 使能）可路由到 TAMPALRM 输出。可通过 RTC_CR 寄存器的 POL 位配置 TAMPALRM 输出极性。

46.3.10 周期性自动唤醒

周期性唤醒标志由 16 位可编程自动重载递减计数器生成。唤醒定时器范围可扩展至 17 位。

可通过 RTC_CR 寄存器中的 WUTE 位来使能此唤醒功能。

唤醒定时器时钟输入 ck_wut 可以是：

- 2、4、8 或 16 分频的 RTC 时钟 (RTCCLK)。

  当 RTCCLK 为 LSE (32.768 kHz) 时，可配置的唤醒中断周期介于 122 µs 和 32 s 之间，且分辨率低至 61 µs。
- BCD 模式下的 ck_spre（通常为 1 Hz 内部时钟），或用于在二进制或混合（BCD 二进制）模式下根据 BCDU 的定义更新日历的时钟。

  当 ck_spre 频率为 1 Hz 时，可以实现的唤醒时间为 1s 到 36h 左右，分辨率为 1 秒。这一较大的可编程时间范围分为两部分：
  - WUCKSEL [2:1] = 10 时为 1s 到 18h

– WUCKSEL [2:1] = 11 时约为 18h 到 36h。在后一种情况下，会将 2^{16} 加到 16 位计数器当前值中。当初始化序列完成之后（请参见*第 2281 页的编程唤醒定时器*），定时器开始递减计数。在低功耗模式下使能唤醒功能时，递减计数保持有效。此外，当计数器计数到 0 时，RTC_SR 寄存器的 WUTF 标志会置 1，并且唤醒计数器会使用其重载值（RTC_WUTR 寄存器值）自动重载。

根据 RTC_WUTR 寄存器中的 WUTOCLR，WUTF 标志必须由软件清零 (WUTOCLR = 0x0000)，或者在自动重载递减计数器计数到 WUTOCLR 值 (0x0000 < WUTOCLR ≤ WUT) 时由硬件自动清零。

唤醒标志基于内部信号 rtc_wut 输出，该信号可用于其他外设（请参见*第 46.3.1 节：RTC 框图*部分）。

通过将 RTC_CR 寄存器中的 WUTIE 位置 1 来使能周期性唤醒中断时，它会使设备退出低功耗模式。

如果已通过 RTC_CR 寄存器的位 OSEL[1:0] 使能周期性唤醒标志，则该标志可连接到 TAMPALRM 输出。可通过 RTC_CR 寄存器的 POL 位配置 TAMPALRM 输出极性。

系统复位以及低功耗模式（睡眠、停机和待机）对唤醒定时器没有任何影响。

46.3.11 RTC 初始化和配置

RTC 二进制、BCD 或混合模式

默认情况下，RTC 处于 BCD 模式（RTC_ICSR 寄存器中的 BIN = 00）：RTC_SSR 寄存器包含亚秒字段值 SS[15:0]，由 ck_apre 提供时钟，允许生成 1 Hz 时钟来更新 BCD 格式的日历寄存器（RTC_TR 和 RTC_DR）。

当 RTC 配置为二进制模式时（RTC_ICSR 寄存器中的 BIN = 01）：RTC_SSR 寄存器包含二进制计数器值 SS[31:0]，由 ck_apre 提供时钟。不使用 BCD 格式的日历寄存器（RTC_TR 和 RTC_DR）。

当 RTC 配置为混合模式时（RTC_ICSR 寄存器中的 BIN = 10 或 11）：RTC_SSR 寄存器包含二进制计数器值 SS[31:0]，由 ck_apre 提供时钟。每次 SSR[BCDU+7:0] 计数到 0 时，日历都会更新（增量为 1 秒）。

RTC 寄存器写保护

系统复位后，可通过电源控制外设中的 DBP 位来保护 RTC 寄存器以防止非正常的写访问（请参见 PWR 电源控制部分）。必须将 DBP 位置 1 才能使能 RTC 寄存器的写访问。

备份域复位后，部分 RTC 寄存器受到写保护：RTC_TR、RTC_DR、RTC_PRER、RTC_CALR、RTC_SHIFTR、RTC_ICSR 中的 INIT、BIN 和 BCDU 位以及 RTC_CR 中的 FMT、SUB1H、ADD1H、REFCKON 位。

要解锁 RTC 寄存器的写保护，需要执行以下步骤。

1. 将 0xCA 写入 RTC_WPR 寄存器。
2. 将 0x53 写入 RTC_WPR 寄存器。

写入一个错误的密钥会再次激活写保护。

保护机制不受系统复位影响。

受 INITPRIV 保护的寄存器由 INIT KEY 进行写保护。

受 CALDPRIV 保护的寄存器由 CAL KEY 进行写保护。

如果 RTC_PRIVCFGR 中的 PRIV 或 INITPRIV 置 1，并且/或者 RTC_SECCFGR 中的 SEC 或 INITSEC 置 1：只有在 PRIV、INITPRIV、SEC 和 INITSEC 配置定义的特权和安全模式下完成对 RTC_WPR 寄存器的写访问时，才可以解锁和锁定 INIT KEY。

如果 RTC_PRIVCFGR 中的 PRIV 或 CALPRIV 置 1，并且/或者 RTC_SECCFGR 中的 SEC 或 CALSEC 置 1：只有在 PRIV、CALPRIV、SEC 和 CALSEC 配置定义的特权和安全模式下完成对 RTC_WPR 寄存器的写访问时，才可以解锁和锁定 CAL KEY。

日历初始化和配置

要编程包括时间格式和预分频器配置在内的初始时间和日期日历值，需按照以下顺序操作：

1. 将 RTC_ICSR 寄存器中的 INIT 位置 1 以进入初始化模式。在此模式下，日历计数器将停止工作并且其值可更新。
2. 轮询 RTC_ICSR 寄存器中的 INITF 位。当 INITF 置 1 时进入初始化阶段模式。
   – 如果 LPCAL=0：INITF 在 INIT 位置 1 后的 2 个 RTCCLK 周期左右置 1。
   – 如果 LPCAL=1：INITF 在 INIT 位置 1 后的 2 个 ck_apre 周期内置 1。
3. 要为日历计数器生成 1 Hz 时钟，应编程 RTC_PRER 寄存器中的两个预分频系数以及 RTC_ICSR 寄存器中的 BIN 和 BCDU。
4. 在影子寄存器（RTC_TR 和 RTC_DR）中加载初始时间和日期值，然后通过 RTC_CR 寄存器中的 FMT 位配置时间格式（12 或 24 小时制）。
5. 通过清零 INIT 位退出初始化模式。随后，自动加载实际日历计数器值。
   – 如果 LPCAL=0：在 4 个 RTCCLK 时钟周期后重新开始计数。
   – 如果 LPCAL=1：在最多 (2 RTCCLK + 1 ck_apre) 个周期后重新开始计数。

当初始化序列完成之后，日历开始计数。RTC_SSR 的内容初始化为：

- 在 BCD 模式下 (BIN=00) 初始化为 PREDIV_S
- 在二进制或混合（BCD 二进制）模式下（BIN=01、10 或 11）初始化为 0xFFFF FFFF。

在 BCD 模式下，RTC_SSR 包含同步预分频器计数器的值。这样，便可计算分辨率低至 1 / (PREDIV_S + 1) 秒的 RTC 的准确时间。因此，可通过增大同步预分频器的值 (PREDIV_S[14:0]) 来提高分辨率。将 PREDIV_S 设置为 0x7FFF 时，可得到允许的最大分辨率（30.52 µs，时钟频率为 32768 Hz）。

但是，提高 PREDIV_S 意味着必须降低 PREDIV_A 才能将同步预分频器的输出维持在 1 Hz。这样，异步预分频器的输出频率会增大，RTC 的动态功耗也会相应增加。RTC 的动态功耗针对 PREDIV_A+1 为 2 的幂数的情况进行了优化。

注： *系统复位后，应用可读取 RTC_ICSR 寄存器中的 INITS 标志，以检查日历是否已初始化。如果该标志为 0，表明自系统复位以来，日历还尚未初始化过，其年份位字段一直还保持着备份域复位默认值 (0x00)。*

注： *要在初始化之后读取日历，必须首先用软件检查 RTC_ICSR 寄存器的 RSF 标志是否置 1。*

夏令时

可通过 RTC_CR 寄存器的 SUB1H、ADD1H 和 BKP 位管理夏令时。

利用 SUB1H 或 ADD1H，软件只需单次操作便可在日历中减去或增加一个小时，无需执行整个初始化步骤。

此外，软件还可以使用 BKP 位来记录是否曾经执行过此操作。

编程闹钟

要对可编程的闹钟进行编程或更新，必须执行类似的步骤。以下步骤针对闹钟 A，但同样适用于闹钟 B。

1. 将 RTC_CR 中的 ALRAE 位清零以禁止闹钟 A。
2. 编程闹钟 A 寄存器（RTC_ALRMASSR/RTC_ALRMAR 或 RTC_ALRABINR）。
3. 将 RTC_CR 寄存器中的 ALRAE 位置 1 以再次使能闹钟 A。

注： *由于时钟同步的缘故，RTC_CR 寄存器的每次更改都将需要大约 2 个 RTCCLK 时钟周期来完成。*

编程唤醒定时器

要配置或更改唤醒定时器的自动重载值（RTC_WUTR 中的 WUT[15:0]），需要按照以下顺序操作：

1. 将 RTC_CR 中的 WUTE 清零以禁止唤醒定时器。
2. 轮询 RTC_ICSR 中的 WUTWF，直到该位置 1，以确保可以访问唤醒自动重载定时器和 WUCKSEL[2:0] 位。在日历初始化模式下必须跳过该步骤。
   - 如果 WUCKSEL[2] = 0：WUTWF 在 WUTE 位清零后的 (1 ck_wut + 1 RTCCLK) 个周期左右置 1。
   - 如果 WUCKSEL[2] = 1：WUTWF 在 WUTE 位清零后的 (1 ck_apre + 1 RTCCLK) 个周期内置 1。
3. 编程唤醒自动重载值 WUT[15:0] 和 WUTOCLR[15:0]，并选择唤醒时钟（RTC_CR 中的 WUCKSEL[2:0] 位）。将 RTC_CR 寄存器中的 WUTE 位置 1 以再次使能定时器。唤醒定时器重新开始递减计数。
   - 如果 WUCKSEL[2] = 0：WUTWF 在 WUTE 位置 1 后的 (1 ck_wut + 1 RTCCLK) 个周期左右清零。
   - 如果 WUCKSEL[2] = 1：WUTWF 在 WUTE 位置 1 后的 (1 ck_apre + 1 RTCCLK) 个周期内清零。

46.3.12 读取日历

当 RTC_CR 寄存器中的 BYPSHAD 控制位清零时

要正确读取 RTC 日历寄存器（RTC_SSR、RTC_TR 和 RTC_DR），APB 时钟频率 (f_{PCLK}) 必须等于或大于 RTC 时钟频率 (f_{RTCCLK}) 的七倍。这可以确保同步机制的安全性。

如果 APB 时钟频率低于 RTC 时钟频率的七倍，则软件必须分两次读取日历时间寄存器和日期寄存器。这样，当两次读取的 RTC_TR 结果相同时，才能确保数据正确。否则必须执行第三次读访问。任何情况下，APB 的时钟频率都不能低于 RTC 的时钟频率。

每次将日历寄存器中的值复制到 RTC_SSR、RTC_TR 和 RTC_DR 影子寄存器时，RTC_ICSR 寄存器中的 RSF 位都会置 1。每个 RTCCLK 周期执行一次复制。为确保这 3 个值一致，读取 RTC_SSR 或 RTC_TR 时会锁定高阶日历影子寄存器中的值，直到读取 RTC_DR。为避免软件对日历执行读访问的时间间隔小于 1 个 RTCCLK 周期：第一次读取日历之后必须通过软件将 RSF 清零，并且软件必须等待到 RSF 置 1 之后才可再次读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

从低功耗模式（Stop 模式或待机模式）唤醒之后，必须通过软件将 RSF 清零。之后，软件必须等待至 RSF 再次置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

RSF 位必须在唤醒之后而不是进入低功耗模式之前进行清零。

系统复位之后，软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。实际上，系统复位会将影子寄存器复位为其默认值。

初始化之后（请参见*第 2280 页的日历初始化和配置*），软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

同步之后（请参见*第 46.3.14 节：RTC 同步*），软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

当 RTC_CR 寄存器中的 BYPSHAD 控制位置 1 时（旁路影子寄存器）

读取日历寄存器时会直接从日历计数器获取值，这样便无需等待至 RSF 位置 1。这对于从低功耗模式（Stop 模式或待机模式）退出后的情况特别有用，因为影子寄存器在这些模式下不更新。

当 BYPSHAD 位置 1 时，如果在对寄存器的两次读访问之间出现 RTCCLK 沿，则不同寄存器的结果彼此可能不一致。此外，如果在读操作期间出现 RTCCLK 沿，则可能导致其中一个寄存器的值不正确。软件必须分两次读取所有寄存器，然后将两次结果加以比较来确认数据是否一致和正确。此外，软件也可以只比较两次读取日历寄存器得到的结果的最低位。

注： *当 BYPSHAD = 1 时，读取日历寄存器的指令需要一个额外的 APB 周期才能完成。*

46.3.13 复位 RTC

日历影子寄存器（RTC_SSR、RTC_TR 和 RTC_DR）以及 RTC 状态寄存器 (RTC_ICSR) 的某些位通过所有可用的系统复位源复位为各自的默认值。

而以下寄存器则通过备份域复位来复位为各自的默认值并且不受系统复位的影响：RTC 当前日历寄存器、RTC 控制寄存器 (RTC_CR)、预分频器寄存器 (RTC_PRER)、RTC 校准寄存器 (RTC_CALR)、RTC 移位寄存器 (RTC_SHIFTR)、RTC 时间戳寄存器（RTC_TSSSR、RTC_TSTR 和 RTC_TSDR）、唤醒定时器寄存器 (RTC_WUTR)、闹钟 A 寄存器和闹钟 B 寄存器（RTC_ALRMASSR/RTC_ALRMAR/RTC_ALRABINR 和 RTC_ALRMBSSR/RTC_ALRMBR/RTC_ALRBBINR）。

此外，当由 LSE 提供时钟时，如果复位源并非备份域复位源（有关不受系统复位影响的 RTC 时钟源的详细信息，请参见 RCC），则 RTC 将在系统复位时保持运行状态。发生备份域复位时，RTC 会停止工作，并且所有 RTC 寄存器都会设置为各自的复位值。

46.3.14 RTC 同步

RTC 可与高精度的远程时钟同步。在读取亚秒字段后（RTC_SSR 或 RTC_TSSSR），即可计算远程时钟的时间与 RTC 之间的精准偏差。之后，可使用 RTC_SHIFTR 对 RTC 的时钟进行零点几秒的“移位”，经过调整后可消除此偏差。

可以使用 RTC 移位控制寄存器 (RTC_SHIFTR) 对 RTC 进行微调。可以用大小为 1 个 ck_apre 周期的分辨率对 RTC_SHIFTR 进行写操作来移位（延迟或提前）时钟。

在这种移位操作中，会将 SUBFS[14:0] 值加到同步预分频器计数器 SS[15:0] 中：这会使时钟产生延迟。

在 BCD 或混合模式下，如果同时将 ADD1S 位置 1，则会增加一秒，与此同时减去的时间为零点几秒，因此会使时钟提前。ADD1S 在二进制模式下不起作用。

对 RTC_SHIFTR 寄存器执行写操作以启动移位操作时，硬件会将 SHPF 标志置 1 以指示移位操作挂起。完成移位操作时，硬件会将该位清零。

小心： 在混合模式下（BIN=10 或 11），SUBFS[14:BCDU+8] 必须写入 0。

小心： 在 BCD 模式下初始化移位操作前，用户必须检查确认 SS[15] = 0，以确保不会发生上溢。在混合模式下，用户必须检查确认位 SS[BCDU+8] = 0。

小心： 该同步功能与参考时钟检测功能不兼容：当 REFCKON = 1 时，固件不能对 RTC_SHIFTR 执行写操作。

46.3.15 RTC 参考时钟检测

该功能仅在 BCD 模式 (BIN=00) 下可用。

RTC 日历更新可与参考时钟 RTC_REFIN（通常为市电频率，50 Hz 或 60 Hz）同步。RTC_REFIN 参考时钟的精度应高于 32.768 kHz LSE 时钟。使能 RTC_REFIN 检测时（将 RTC_CR 的 REFCKON 位置 1），日历仍由 LSE 提供时钟，而 RTC_REFIN 用于补偿不准确的日历更新频率 (1 Hz)。

每个 1 Hz 时钟边沿都与最近的 RTC_REFIN 时钟边沿进行比较（如果在给定的时间窗口内发现一个边沿）。在大多数情况下，两个时钟边沿恰好对齐。当 1 Hz 时钟由于 LSE 时钟不精确而发生偏离时，RTC 会稍微偏移 1 Hz 时钟，以便后续的 1 Hz 时钟边沿能够对齐。利用这种机制，可使日历像参考时钟一样精确。

RTC 使用 32.768 kHz 晶振产生的 256 Hz 时钟 (ck_apre) 检测是否存在参考时钟源。大约在日历每次更新时（每 1 秒钟），便会在时间窗口期间执行一次检测。检测到第一个参考时钟边沿时，该窗口等于 7 个 ck_apre 周期。随后的日历更新使用长度为 3 个 ck_apre 周期的较小窗口。

每次在窗口中检测到参考时钟时，都会强制输出 ck_spre 时钟的异步预分频器进行重载。当参考时钟与 1 Hz 时钟对齐时，此操作不起作用，因为预分频器会在同一时刻重载。当时钟不对齐时，重载操作会微调后续的 1 Hz 时钟边沿，使其与参考时钟对齐。

如果参考时钟停止（在 3 个 ck_apre 窗口内未出现参考时钟边沿），日历将仅根据 LSE 时钟进行连续更新。RTC 随后使用 ck_spre 边沿上居中的大检测窗口（7 个 ck_apre 周期）等待参考时钟。

使能 RTC_REFIN 检测后，必须将 PREDIV_A 和 PREDIV_S 设置为各自的默认值：

- PREDIV_A = 0x007F
- PREVID_S = 0x00FF

注： RTC_REFIN 时钟检测在待机模式下不可用。

46.3.16 RTC 平滑数字校准

RTC 频率可采用约 0.954 ppm 的分辨率进行数字校准，校准范围为 -487.1 ppm 到 +488.5 ppm。使用一系列微调（增加和/或减少单独的 ck_cal 脉冲）进行频率校正。

如果 LPCAL=0：ck_cal = RTCCLK

如果 LPCAL=1：ck_cal = ck_apre

这些微调的分布非常均匀，因此 RTC 的校准效果相当好，即使在短时间内持续观察也是如此。

RTC 超低功耗模式

将 RTC 校准寄存器 (RTC_CALR) 中的 LPCAL 位置 1，可以降低 RTC 的功耗。在这种情况下，校准机制应用于 ck_apre 而不是 RTCCLK。最终产生的精度是相同的，但校准的周期约为 2^{20} x PREDIV_A x RTCCLK 个脉冲，而非 2^{20} 个 RTCCLK 脉冲（当 LPCAL=0 时）。

平滑校准机制

平滑校准寄存器 (RTC_CALR) 可指定校准周期内要减少的 ck_cal 时钟周期数：

- 将位 CALM[0] 置 1 时，校准周期内将只减少一个时钟。
- 将 CALM[1] 置 1 时，将减少两个时钟。
- 将 CALM[2] 置 1 时，将减少四个时钟。
- 依此类推，将 CALM[8] 置 1 时，将减少 256 个时钟。

注： *CALM[8:0] (RTC_CALR) 可指定校准周期内要减少的 ck_cal 脉冲数。将位 CALM[0] 置 1 时，校准周期内将只减少 1 个脉冲（当 cal_cnt[19:0] = 0x80000 时）；将 CALM[1] 置 1 时，将减少 2 个周期（当 cal_cnt = 0x40000 和 0xC0000 时）；将 CALM[2] 置 1 时，将减少 4 个周期（当 cal_cnt = 0x20000/0x60000/0xA0000/0xE0000 时）；依此类推，将 CALM[8] 置 1 时，将减少 256 个时钟（当 cal_cnt = 0xXX800 时）。*

使用适当分辨率时，CALM 可使 RTC 频率减少最多 487.1 ppm，而 CALP 可用于使频率增加 488.5 ppm。将 CALP 置 1，可每隔 2^{11} 个 ck_cal 周期有效插入一个额外的 ck_cal 脉冲，这意味着每个校准周期内可增加 512 个时钟。

将 CALM 和 CALP 配合使用时，可在校准周期内增加一个范围为 -511 到 +512 ck_cal 周期的偏差，对应的校准范围为 -487.1 ppm 到 +488.5 ppm，分辨率约为 0.954 ppm。

若输入频率 (F_{RTCCLK}) 已知，可通过以下公式计算有效校准频率 (F_{CAL})：

$$F_{CAL} = F_{RTCCLK} \times [1 + (CALP \times 512 - CALM) / (2^{20} + CALM - CALP \times 512)]$$

小心： PREDIV_A 必须大于或等于 3。

PREDIV_A < 3 条件下的校准

当异步预分频器值（RTC_PRER 寄存器中的 PREDIV_A 位）小于 3 时，不能将 CALP 位置 1。如果 CALP 已置 1 并且 PREDIV_A 位的值小于 3，则会忽略 CALP，即假定 CALP 等于 0 而执行校准。

但是，在 BCD 模式下，要在 PREDIV_A 小于 3 的条件下执行校准，应降低同步预分频器值 (PREDIV_S) 以便每秒内可加速 8 个 ck_cal 时钟周期，这意味着每个校准周期可增加 256 个时钟周期。因此，仅使用 CALM 位，可在每个校准周期内有效增加 255 到 256 个时钟脉冲（对应的校准范围为 243.3 ppm 到 244.1 ppm）。

在标称 RTCCLK 频率 32768 Hz 下，当 PREDIV_A 等于 1 时（分频系数为 2），应将 PREDIV_S 设置为 16379 而不是 16383（少 4）。唯一相关的其他情况是，当 PREDIV_A 等于 0 时，应将 PREDIV_S 设置为 32759 而不是 32767（少 8）。

如果以这种方式减少 PREDIV_S，则采用以下公式计算校准输入时钟的有效频率：

$$F_{CAL} = F_{RTCCLK} \times [1 + (256 - CALM) / (2^{20} + CALM - 256)]$$

在这种情况下，如果 RTCCLK 恰好为 32768.00 Hz，则当 CALM[7:0] 等于 0x100 时（CALM 范围的中值），说明设置正确。

验证 RTC 校准

建议在 LPCAL = 0 的条件下验证 RTC 校准，以获得 32 秒校准周期。

通过测量 RTCCLK 的精确频率，计算正确的 CALM 和 CALP 值以确保 RTC 精度。此外，还为应用提供了一个可选的 1 Hz 输出，用来测量和验证 RTC 精度。

如果在有限的间隔内测量 RTC 的精确频率，则会导致测量期间产生最多 2 个 RTCCLK 时钟周期的测量误差，具体取决于数字校准周期与测量周期的对齐方式。

但是，如果测量周期与校准周期的长度相同，则可以消除此测量误差。在这种情况下，观测到的唯一误差是由数字校准的分辨率导致的误差。

- 默认情况下，校准周期为 32 秒。

  在此模式下，测量整个 32 秒内 1 Hz 输出的精度，可确保测量误差在 0.477 ppm 内（32 秒内为 0.5 个 RTCCLK 周期，受校准分辨率限制）。
- 可将 RTC_CALR 寄存器的 CALW16 位置 1，以强制 16 秒的校准周期。

  此时，可在 16 秒内测量 RTC 精度，产生的最大误差为 0.954 ppm（16 秒内为 0.5 个 RTCCLK 周期）。但是，由于校准分辨率降低，长期的 RTC 精度也会降到 0.954 ppm：将 CALW16 置 1 时，CALM[0] 位将始终保持为 0。
- 可将 RTC_CALR 寄存器的 CALW8 位置 1，以强制 8 秒的校准周期。

  此时，可在 8 秒内测量 RTC 精度，产生的最大误差为 1.907 ppm（8 秒内为 0.5 个 RTCCLK 周期）。长期的 RTC 精度也会降到 1.907 ppm：将 CALW8 置 1 时，CALM[1:0] 位将始终保持为 00。

动态重校准

当 RTC_ICSR/INITF = 0 时，可动态更新校准寄存器 (RTC_CALR)，具体步骤如下：

1. 轮询 RTC_ICSR/RECALPF（重新校准挂起标志）。
2. 如果该标志为 0，则可以根据需要向 RTC_CALR 写入新值。随后 RECALPF 位会被自动置为 1。
3. 新校准设置将在对 RTC_CALR 执行写操作之后的三个 ck_apre 周期内生效。

46.3.17 时间戳功能

将 RTC_CR 寄存器的 TSE 或 ITSE 位置 1 可使能时间戳。

将 TSE 置 1 时：

当在 RTC_TS 引脚上检测到时间戳事件时，日历会保存到时间戳寄存器（RTC_TSSSR、RTC_TSTR 和 RTC_TSDR）中。

将 TAMPTS 置 1 时：

当检测到内部或外部时间戳事件时，日历会保存到时间戳寄存器（RTC_TSSSR、RTC_TSTR 和 RTC_TSDR）中。请参见 *RTC 控制寄存器 (RTC_CR)* 和 *Tamper 事件的时间戳*一节。

将 ITSE 置 1 时：

当检测到内部时间戳事件时，日历会保存到时间戳寄存器（RTC_TSSSR、RTC_TSTR 和 RTC_TSDR）中。内部时间戳事件通过切换到 V_{BAT} 电源生成。

由于内部事件或外部事件而发生时间戳事件时，RTC_SR 寄存器中的时间戳标志位 (TSF) 将置 1。如果是内部事件，RTC_SR 寄存器中的 ITSF 标志也将置 1。

通过将 RTC_CR 寄存器中的 TSIE 位置 1，可在发生时间戳事件时生成中断。

如果在时间戳标志 (TSF) 已置 1 的条件下检测到新的时间戳事件，则时间戳上溢标志 (TSOVF) 将置 1，而时间戳寄存器（RTC_TSTR 和 RTC_TSDR）将保持上一事件的结果。

注： *由于同步过程，TSF 将在来自 RTC_TS 引脚或 rtc_its 内部信号的时间戳事件发生后的 2 个 ck_apre 周期内置 1。TSF 在 Tamper 标志置 1 后的 3 个 ck_apre 周期内置 1。*

TSOVF 在事件发生后的仅 1 个 ck_apre 周期内就会置 1。这意味着，如果两个时间戳事件接连发生，则 TSOVF 可能为“1”而 TSF 仍为“0”。因此，建议只在检测到 TSF 为“1”后再轮询 TSOVF。

小心： 如果在 TSF 位清零后紧接着发生时间戳事件，则 TSF 和 TSOVF 位都将置 1。为防止在时间戳事件发生的同时屏蔽该事件，除非已将 TSF 位读取为 1，否则应用程序不得将 0 写入 TSF 位。

46.3.18 校准时钟输出

将 RTC_CR 寄存器中的 COE 位置 1 时，会在 CALIB 设备输出上提供一个参考时钟。

如果 RTC_CR 寄存器中的 COSEL 位复位且 PREDIV_A = 0x7F，则 CALIB 频率为 $f_{RTCCLK/64}$。这相当于 RTCCLK 频率为 32.768 kHz 时，512 Hz 的校准输出。CALIB 占空比是不规则的：下降沿上存在轻微抖动。因此推荐使用上升沿。

当 COSEL 置 1 且“PREDIV_S+1”为 256 的非零整数倍（即：PREDIV_S[7:0] = 0xFF）时，CALIB 频率为 fRTCCLK/(256 * (PREDIV_A+1))。这相当于 RTCCLK 频率为 32.768 kHz 时，1 Hz 的校准输出，其中预分频器为默认值（PREDIV_A = Ox7F、PREDIV_S = 0xFF）。

注： *COSEL 清零时，CALIB 输出为异步预分频器的第 6 级输出。如果 LPCAL 从 0 变为 1，则在 LPCAL 切换过程中输出可能是不规则的（干扰...）。LPCAL = 1 时，该输出始终可用。LPCAL = 0 且 PREDIV_A < 0x20 时，不存在任何输出。*

COSEL 置 1 时，CALIB 输出为同步预分频器的第 8 级输出。

46.3.19 Tamper 和闹钟输出

RTC_CR 寄存器中的 OSEL[1:0] 控制位用于激活闹钟输出 TAMPALRM，以及选择输出的功能。这些功能可反映 RTC_SR 寄存器中相应标志的内容。

当 RTC_CR 中的 TAMPOE 控制位置 1 时，所有外部和内部 Tamper 标志将进行逻辑或运算并连接到 TAMPALRM 输出。如果 OSEL = 00，则 TAMPALRM 输出仅反映 Tamper 标志。如果 OSEL ≠ 00，则 TAMPALRM 上的信号反映 Tamper 标志以及闹钟 A、闹钟 B 或唤醒标志。

TAMPALRM 输出的极性由 RTC_CR 中的 POL 控制位确定，这样当 POL 置 1 时会输出选定标志位的相反值。

TAMPALRM 输出

使用 RTC_CR 寄存器中的控制位 TAMPALRM_TYPE 可将 TAMPALRM 引脚配置为输出开漏或输出上拉。借助 RTC_CR 中的 TAMPALRM_PU，可以在输出模式下应用内部上拉。

注： *使能* TAMPALRM *输出后，其优先级高于 RTC_OUT1 上的 CALIB。*

如果在 RTC 中将 TAMPALRM 配置为开漏，则必须将 RTC_OUT1 GPIO 配置为输入。

46.4 RTC 低功耗模式

表 401. 低功耗模式对 RTC 的影响

模式	说明
睡眠	无影响 RTC 中断可使设备退出睡眠模式。
Stop	当 RTC 时钟源为 LSE 或 LSI 时，RTC 保持工作状态。RTC 中断可使设备退出 Stop 模式。
待机	当 RTC 时钟源为 LSE 或 LSI 时，RTC 保持工作状态。RTC 中断可使设备退出待机模式。

下表汇总了所有模式下的 RTC 引脚和功能。

表 402. 各个模式下的 RTC 引脚功能

功能	待机模式以外的所有低功耗模式下功能正常	待机模式下功能正常	V_{BAT} 模式下功能正常
RTC_TS	是	是	是
RTC_REFIN	是	否	否
RTC_OUT1	是	是	是
RTC_OUT2	是	是	是

46.5 RTC 中断

中断通道在屏蔽中断状态寄存器中置 1，或在安全屏蔽中断状态寄存器中置 1，具体取决于其安全模式配置。非安全中断输出或安全中断输出也会激活。

表 403. 非安全中断请求

中断缩写	中断事件	事件标志(1)	使能控制位(2)	中断清除方法	退出低功耗模式
RTC_WKUP_ALARM	闹钟 A	ALRAF	ALRAIE 和（ALRASEC=0 且 SEC=0）	向 CALRAF 中写入 1	是(3)
	闹钟 B	ALRBF	ALRBIE 和（ALRBSEC=0 且 SEC=0）	向 CALRBF 中写入 1	是(3)
	唤醒定时器	WUTF	WUTIE 和（WUTSEC=0 且 SEC=0）	向 CWUTF 中写入 1	是(3)
RTC_TS	时间戳	TSF	TSIE 和（TSSEC=0 且 SEC=0）	向 CTSF 中写入 1	是(3)

1. 事件标志位于 RTC_SR 寄存器中。
2. 中断屏蔽标志（由事件标志和使能控制位的逻辑与运算得出）位于 RTC_MISR 寄存器中。
3. 通过振荡器提供时钟时，RTC 可以在低功耗模式下正常工作。

表 404. 安全中断请求

中断缩写	中断事件	事件标志(1)	使能控制位(2)	中断清除方法	退出低功耗模式
RTC_WKUP_ALARM_S	闹钟 A	ALRAF	ALRAIE（ALRASEC=1 或 SEC=1）	向 CALRAF 中写入 1	是(3)
	闹钟 B	ALRBF	ALRBIE（ALRBSEC=1 或 SEC=1）	向 CALRBF 中写入 1	是(3)
	唤醒定时器	WUTF	WUTIE（WUTSEC=1 或 SEC=1）	向 CWUTF 中写入 1	是(3)
RTC_TS_S	时间戳	TSF	TSIE（TSSEC=1 或 SEC=1）	向 CTSF 中写入 1	是(3)

1. 事件标志位于 RTC_SR 寄存器中。
2. 中断屏蔽标志（由事件标志和使能控制位的逻辑与运算得出）位于 RTC_SMISR 寄存器中。
3. 通过振荡器提供时钟时，RTC 可以在低功耗模式下正常工作。

46.6 RTC 寄存器

有关寄存器说明中使用的缩写，请参见参考手册中的*第 1.2 节*。

外设寄存器可按字（32 位）进行访问。

46.6.1 RTC 时间寄存器 (RTC_TR)

RTC time register

RTC_TR 是日历时间影子寄存器。只能在初始化模式下对该寄存器执行写操作。请参见*第 2280 页的日历初始化和配置*和*第 2281 页的读取日历*。

此寄存器受写保护。*第 2279 页的RTC 寄存器写保护*中介绍了写访问的过程。

可以对该寄存器进行写保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

可以对该寄存器进行写保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x00

备份域复位值：0x0000 0000

系统复位值：0x0000 0000（当 BYPSHAD = 0 时，当 BYPSHAD = 1 时不受影响）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]			
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值。

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)

位 15 保留，必须保持复位值。

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)

位 7 保留，必须保持复位值。

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)

46.6.2 RTC 日期寄存器 (RTC_DR)

RTC date register

RTC_DR 是日历日期影子寄存器。只能在初始化模式下对该寄存器执行写操作。请参见*第 2280 页的日历初始化和配置*和*第 2281 页的读取日历*。

此寄存器受写保护。*第 2279 页的RTC 寄存器写保护*中介绍了写访问的过程。

可以对该寄存器进行写保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

可以对该寄存器进行写保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x04

备份域复位值：0x0000 2101

系统复位值：0x0000 2101（当 BYPSHAD = 0 时，当 BYPSHAD = 1 时不受影响）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YT[3:0]				YU[3:0]			
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:20 **YT[3:0]**：年份的十位（BCD 格式）(Year tens in BCD format)

位 19:16 **YU[3:0]**：年份的个位（BCD 格式）(Year units in BCD format)

位 15:13 **WDU[2:0]**：星期几的个位 (Week day units)
000：禁止
001：星期一
...
111：星期日

位 12 **MT**：月份的十位（BCD 格式）(Month tens in BCD format)

位 11:8 **MU[3:0]**：月份的个位（BCD 格式）(Month units in BCD format)

位 7:6 保留，必须保持复位值。

位 5:4 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 3:0 **DU[3:0]**：日期的个位（BCD 格式）(Date units in BCD format)

注： *日历在达到最大值时会冻结，而不会滚动。*

46.6.3 RTC 亚秒寄存器 (RTC_SSR)

RTC subsecond register

偏移地址：0x08

备份域复位值：0x0000 0000

系统复位值：0x0000 0000（当 BYPSHAD = 0 时，当 BYPSHAD = 1 时不受影响）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SS[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SS[31:0]**：同步二进制计数器 (Synchronous binary counter)

SS[31:16]：同步二进制计数器 MSB 值 (Synchronous binary counter MSB values)

选择二进制模式或混合模式时（BIN = 01、10 或 11）：

SS[31:16] 是 SS[31:0] 自由运行递减计数器的高 16 位。

选择 BCD 模式时 (BIN=00)：

SS[31:16] 由硬件强制设为 0x0000。

SS[15:0]：亚秒值/同步二进制计数器 LSB 值 (Subsecond value/synchronous binary counter LSB values)

选择二进制模式时（BIN = 01、10 或 11）：

SS[15:0] 是 SS[31:0] 自由运行递减计数器的低 16 位。

选择 BCD 模式时 (BIN=00)：

SS[15:0] 是同步预分频器计数器的值。此亚秒值可根据以下公式得出：

亚秒值 = (PREDIV_S - SS) / (PREDIV_S + 1)

仅当执行移位操作之后，SS 才能大于 PREDIV_S。在这种情况下，正确的时间/日期比 RTC_TR/RTC_DR 所指示的时间/日期慢一秒钟。

46.6.4 RTC 初始化控制和状态寄存器 (RTC_ICSR)

RTC initialization control and status register

此寄存器受写保护。*第 2279 页的RTC 寄存器写保护*中介绍了写访问的过程。

既可以为该寄存器设置全局保护，也可以为该寄存器的每个位单独设置保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

既可以为该寄存器设置全局保护，也可以为该寄存器的每个位单独设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x0C

备份域复位值：0x0000 0007

系统复位：不受影响（INIT、INITF 和 RSF 位除外，它们在复位时被清零）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RECALPF
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	BCDU[2:0]			BIN[1:0]		INIT	INITF	RSF	INITS	SHPF	WUTWF	Res.	Res.
			rw	rw	rw	rw	rw	rw	r	rc_w0	r	r	r		

位 31:17 保留，必须保持复位值。

位 16 **RECALPF**：重新校准挂起标志 (Recalibration pending Flag)

当软件对 RTC_CALR 寄存器执行写操作时，RECALPF 状态标志将自动置 1，指示 RTC_CALR 寄存器已屏蔽。当采用新的校准设置时，该位恢复为 0。请参见*动态重校准*。

位 15:13 保留，必须保持复位值。

位 12:10 **BCDU[2:0]**：BCD 更新 (BCD update)（BIN = 10 或 11）

在混合模式下，如果同时使用 BCD 日历和二进制扩展计数器（BIN = 10 或 11），则将使用 SSR 最低有效位递增日历秒单元。

0x0：每当 SS[7:0] = 0 时，就会产生 1s 日历增量
0x1：每当 SS[8:0] = 0 时，就会产生 1s 日历增量
0x2：每当 SS[9:0] = 0 时，就会产生 1s 日历增量
0x3：每当 SS[10:0] = 0 时，就会产生 1s 日历增量
0x4：每当 SS[11:0] = 0 时，就会产生 1s 日历增量
0x5：每当 SS[12:0] = 0 时，就会产生 1s 日历增量
0x6：每当 SS[13:0] = 0 时，就会产生 1s 日历增量
0x7：每当 SS[14:0] = 0 时，就会产生 1s 日历增量

位 9:8 **BIN[1:0]**：二进制模式 (Binary mode)

00：自由运行 BCD 日历模式（禁止二进制模式）
01：自由运行二进制模式（禁止 BCD 模式）
10：自由运行 BCD 日历模式和二进制模式
11：自由运行 BCD 日历模式和二进制模式

位 7 **INIT**：初始化模式 (Initialization mode)

0：自由运行模式
1：初始化模式，用于编程时间和日期寄存器（RTC_TR 和 RTC_DR）和预分频器寄存器 (RTC_PRER) 以及 BIN 和 BCDU 位字段。计数器停止计数，当 INIT 被复位后，计数器从新值开始计数。

位 6 **INITF**：初始化标志 (Initialization flag)

当该位置 1 时，RTC 处于初始化状态，此时可更新事件、日期和预分频器寄存器。

0：不允许更新日历寄存器

1：允许更新日历寄存器

位 5 **RSF**：寄存器同步标志 (Registers synchronization flag)

每次将日历寄存器的值复制到影子寄存器（RTC_SSR、RTC_TR 和 RTC_DR）时，都会由硬件将该位置 1。在初始化模式下、移位操作挂起时 (SHPF = 1) 或在旁路影子寄存器模式 (BYPSHAD = 1) 下，该位由硬件清零。该位还可由软件清零。

在初始化模式下，该位可由软件或硬件清零。

0：日历影子寄存器尚未同步

1：日历影子寄存器已同步

位 4 **INITS**：初始化状态标志 (Initialization status flag)

当日历年份字段不为 0 时（备份域复位状态），由硬件将该位置 1。

0：日历尚未初始化

1：日历已经初始化

位 3 **SHPF**：移位操作挂起 (Shift operation pending)

只要通过对 RTC_SHIFTR 寄存器执行写操作来启动移位操作，此标志便由硬件置 1。执行完相应的移位操作后，此标志由硬件清零。对 SHPF 位执行写入操作不起作用。

0：没有移位操作挂起

1：某个移位操作挂起

位 2 **WUTWF**：唤醒定时器写标志 (Wake-up timer write flag)

在 RTC_CR 寄存器中的 WUTE 位置 0 后，当 WUT 的值可更改时，由硬件将该位置 1。

该位在初始化模式下由硬件清零。

0：除初始化模式外，不允许更新唤醒定时器配置

1：允许更新唤醒定时器配置

位 1:0 保留，必须保持复位值。

46.6.5 RTC 预分频器寄存器 (RTC_PRER)

RTC prescaler register

只能在初始化模式下对该寄存器执行写操作。必须通过两次独立的写访问执行初始化。请参见*第 2280 页的日历初始化和配置*。

此寄存器受写保护。*第 2279 页的RTC 寄存器写保护*中介绍了写访问的过程。

可以对该寄存器进行写保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

可以对该寄存器进行写保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x10

备份域复位值：0x007F 00FF

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREDIV_A[6:0]						
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PREDIV_S[14:0]														
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值。

位 22:16 **PREDIV_A[6:0]**：异步预分频系数 (Asynchronous prescaler factor)
下面是异步分频系数的公式：
ck_apre 频率 = RTCCLK 频率/(PREDIV_A+1)

位 15 保留，必须保持复位值。

位 14:0 **PREDIV_S[14:0]**：同步预分频系数 (Synchronous prescaler factor)
下面是同步分频系数的公式：
ck_spre 频率 = ck_apre 频率/(PREDIV_S+1)

46.6.6 RTC 唤醒定时器寄存器 (RTC_WUTR)

RTC wake-up timer register

仅当 RTC_ICSR 中的 WUTWF 置 1 时才可对该寄存器执行写操作。

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x14

备份域复位值：0x0000 FFFF

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WUTOCLR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WUT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **WUTOCLR[15:0]**：唤醒自动重载输出清零值 (Wake-up auto-reload output clear value)

如果 WUTOCLR[15:0] 不等于 0x0000，当自动重载递减计数器计数到 0 时，WUTF 由硬件置 1；当自动重载递减计数器计数到 WUTOCLR[15:0] 时，WUTF 由硬件清零。

如果 WUTOCLR[15:0] = 0x0000，则 WUTF 由软件清零，并在 WUT 递减计数器计数到 0 时由硬件置 1。

位 15:0 **WUT[15:0]**：唤醒自动重载值位 (Wake-up auto-reload value bits)

当使能唤醒定时器时（WUTE 置 1），每 (WUT[15:0] + 1) 个 ck_wut 周期将 WUTF 标志置 1 一次。ck_wut 周期通过 RTC_CR 寄存器的 WUCKSEL[2:0] 位进行选择。

当 WUCKSEL[2] = 1 时，唤醒定时器变为 17 位，WUCKSEL[1] 等效为 WUT[16]，即要重载到定时器的最高有效位。

WUTF 第一次置 1 发生在 WUTE 置 1 后的 WUT 个周期到 (WUT + 2) 个 ck_wut 周期之间。

禁止在 WUCKSEL[2:0] = 011 (RTCCLK/2) 时将 WUT[15:0] 设置为 0x0000。

46.6.7 RTC 控制寄存器 (RTC_CR)

RTC control register

此寄存器受写保护。第 2279 页的RTC 寄存器写保护中介绍了写访问的过程。

既可以为该寄存器设置全局保护，也可以为该寄存器的每个位单独设置保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

既可以为该寄存器设置全局保护，也可以为该寄存器的每个位单独设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x18

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUT2EN	TAMPALRM_TYPE	TAMPALRM_PU	ALRBFCLR	ALRAFCLR	TAMPOE	TAMPTS	ITSE	COE	OSEL[1:0]		POL	COSEL	BKP	SUB1H	ADD1H
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	WUTIE	ALRBIE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	Res.	FMT	BYPSHAD	REFCKON	TSEDGE	WUCKSEL[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31 **OUT2EN**：RTC_OUT2 输出使能 (RTC_OUT2 output enable)

当该位置 1 时，可将 RTC 输出重映射到 RTC_OUT2，具体如下：

OUT2EN = 0：RTC 输出 2 禁止

如果 OSEL ≠ 00 或 TAMPOE = 1：在 RTC_OUT1 上输出 TAMPALRM

如果 OSEL = 00、TAMPOE = 0 且 COE = 1：在 RTC_OUT1 上输出 CALIB

OUT2EN = 1：RTC 输出 2 使能

如果 (OSEL ≠ 00 或 TAMPOE = 1) 且 COE = 0：在 RTC_OUT2 上输出 TAMPALRM

如果 OSEL = 00、TAMPOE = 0 且 COE = 1：在 RTC_OUT2 上输出 CALIB

如果（OSEL ≠ 00 或 TAMPOE = 1）且 COE = 1：在 RTC_OUT2 上输出 CALIB，在 RTC_OUT1 上输出 TAMPALRM。

位 30 **TAMPALRM_TYPE**：TAMPALRM 输出类型 (TAMPALRM output type)

0：TAMPALRM 为推挽输出

1：TAMPALRM 为开漏输出

位 29 **TAMPALRM_PU**：TAMPALRM 上拉使能 (TAMPALRM pull-up enable)

0：不对 TAMPALRM 输出应用上拉

1：对 TAMPALRM 输出应用上拉

位 28 **ALRBFCLR**：闹钟 B 标志自动清零 (Alarm B flag automatic clear)

0：闹钟 B 事件生成触发事件，必须通过软件将 ALRBF 清零来允许下一次闹钟事件。

1：闹钟 B 事件生成触发事件。ALRBF 在 1 个 ck_apre 周期后由硬件自动清零。

位 27 **ALRAFCLR**：闹钟 A 标志自动清零 (Alarm A flag automatic clear)

0：闹钟 A 事件生成触发事件，必须通过软件将 ALRAF 清零来允许下一次闹钟事件。

1：闹钟 A 事件生成触发事件。ALRAF 在 1 个 ck_apre 周期后由硬件自动清零。

位 26 **TAMPOE**：TAMPALRM 上的 Tamper 检测输出使能 (Tamper detection output enable on TAMPALRM)

0：Tamper 标志未发送到 TAMPALRM

1：Tamper 标志发送到 TAMPALRM，并与 OSEL 提供的信号和 POL 提供的极性相结合。

位 25 **TAMPTS**：发生 Tamper 检测事件时激活时间戳 (Activate timestamp on tamper detection event)

0：发生 Tamper 检测事件时不保存 RTC 时间戳

1：发生 Tamper 检测事件时保存 RTC 时间戳

即便 RTC_CR 寄存器中的 TSE = 0，TAMPTS 仍有效。时间戳标志在 Tamper 标志置 1 后的 3 个 ck_apre 周期内置 1。

注意：必须在进入 RTC 初始化模式之前，将 TAMPTS 清零。

位 24 **ITSE**：内部事件时间戳使能 (timestamp on internal event enable)

0：禁止内部事件时间戳

1：使能内部事件时间戳

位 23 **COE**：校准输出使能 (Calibration output enable)
该位使能 CALIB 输出
0：禁止校准输出
1：使能校准输出

位 22:21 **OSEL[1:0]**：输出选择 (Output selection)
这些位用于选择要连接到 TAMPALRM 输出的标志。
00：禁止输出
01：使能闹钟 A 输出
10：使能闹钟 B 输出
11：使能唤醒输出

位 20 **POL**：输出极性 (Output polarity)
该位用于配置 TAMPALRM 输出的极性。
0：当 ALRAF/ALRBF/WUTF 置 1 时（取决于 OSEL[1:0]），或者当 TAMPxF/ITAMPxF 置 1 时（如果 TAMPOE = 1），该引脚为高电平。
1：当 ALRAF/ALRBF/WUTF 置 1 时（取决于 OSEL[1:0]），或者当 TAMPxF/ITAMPxF 置 1 时（如果 TAMPOE = 1），该引脚为低电平。

位 19 **COSEL**：校准输出选择 (Calibration output selection)
当 COE = 1 时，该位可选择 CALIB 上输出的信号。
0：校准输出为 512 Hz
1：校准输出为 1 Hz
在 RTCCLK 为 32.768 kHz 且预分频器为其默认值（PREDIV_A = 127 且 PREDIV_S = 255）的条件下，这些频率有效。请参见*第 46.3.18 节：校准时钟输出*。

位 18 **BKP**：备份 (Backup)
用户可对该位执行写操作以记录是否已对夏令时进行更改。

位 17 **SUB1H**：减少 1 小时（冬季时间更改）(Subtract 1 hour (winter time change))
当该位在初始化模式以外的模式下置 1 时，如果当前小时不是 0，则日历时间将减少 1 小时。该位始终读为 0。
当前小时为 0 时，将该位置 1 没有任何作用。
0：无影响
1：将当前时间减少 1 小时。这可用于冬季时间更改。

位 16 **ADD1H**：增加 1 小时（夏季时间更改）(Add 1 hour (summer time change))
当该位在初始化模式以外的模式下置 1 时，日历时间将增加 1 小时。该位始终读为 0。
0：无影响
1：将当前时间增加 1 小时。这可用于夏季时间更改

位 15 **TSIE**：时间戳中断使能 (Timestamp interrupt enable)
0：禁止时间戳中断
1：时间戳中断使能 (Timestamp interrupt enable)

位 14 **WUTIE**：唤醒定时器中断使能 (Wake-up timer interrupt enable)
0：禁止唤醒定时器中断
1：使能唤醒定时器中断

位 13 **ALRBIE**：闹钟 B 中断使能 (Alarm B interrupt enable)
0：禁止闹钟 B 中断
1：闹钟 B 中断使能 (Alarm B interrupt enable)

位 12 **ALRAIE**：闹钟 A 中断使能 (Alarm A interrupt enable)
0：禁止闹钟 A 中断
1：使能闹钟 A 中断

位 11 **TSE**：时间戳使能 (timestamp enable)
0：禁止时间戳
1：使能时间戳

位 10 **WUTE**：唤醒定时器使能 (Wake-up timer enable)
0：禁止唤醒定时器
1：使能唤醒定时器
注意：唤醒定时器禁止时，需要等到 WUTWF = 1 后才能重新使能。

位 9 **ALRBE**：闹钟 B 使能 (Alarm B enable)
0：禁止闹钟 B
1：使能闹钟 B

位 8 **ALRAE**：闹钟 A 使能 (Alarm A enable)
0：禁止闹钟 A
1：使能闹钟 A

位 7 保留，必须保持复位值。

位 6 **FMT**：小时格式 (Hour format)
0：24 小时 / 天格式
1：AM/PM 小时格式

位 5 **BYPSHAD**：旁路影子寄存器 (Bypass the shadow registers)
0：日历值（从 RTC_SSR、RTC_TR 和 RTC_DR 读取时）取自影子寄存器，该影子寄存器每两个 RTCCLK 周期更新一次。
1：日历值（从 RTC_SSR、RTC_TR 和 RTC_DR 读取时）直接取自日历计数器。
注意：如果 APB 时钟的频率低于 7 倍的 RTCCLK 频率，则必须将 BYPSHAD 置 1。

位 4 **REFCKON**：RTC_REFIN 参考时钟检测使能（50 Hz 或 60 Hz）(RTC_REFIN Reference clock detection enable (50 or 60 Hz))
0：禁止 RTC_REFIN 检测
1：使能 RTC_REFIN 检测
注意：BIN 必须为 0x00 并且 PREDIV_S 必须为 0x00FF。

位 3 **TSEDGE**：时间戳事件有效边沿 (Timestamp event active edge)
0：RTC_TS 输入上升沿生成时间戳事件
1：RTC_TS 输入下降沿生成时间戳事件
TSEDGE 发生更改时，必须复位 TSE 以避免将 TSF 意外置 1。

位 2:0 **WUCKSEL[2:0]**：ck_wut 唤醒时钟选择 (ck_wut wake-up clock selection)
000：选择 RTC/16 时钟
001：选择 RTC/8 时钟
010：选择 RTC/4 时钟
011：选择 RTC/2 时钟
10x：在 BCD 模式下，选择 ck_spre 时钟（通常为 1 Hz）。在二进制模式或混合模式下，则为 BCDU 选择的时钟。
11x：在 BCD 模式下，选择 ck_spre 时钟（通常为 1 Hz）。在二进制模式或混合模式下，则为 BCDU 选择的时钟。此外，将 2^{16} 加到 WUT 计数器值中。

注： *只能在初始化模式下 (RTC_ICSR/INITF = 1) 对该寄存器的位 6 和 4 执行写操作。*

WUT = 唤醒单元计数器值。当 WUCKSEL[2:1 = 11] 时，WUT = (0x0000 到 0xFFFF) + 0x10000（增加的值）。

只能在 RTC_CR WUTE 位 = 0 且 RTC_ICSR WUTWF 位 = 1 时对该寄存器的位 2 到 0 执行写操作。

建议不要在日历小时递增时更改小时，因为这样做可能会屏蔽日历小时的增量。

ADD1H 和 SUB1H 的更改在下一秒生效。

46.6.8 RTC 特权模式控制寄存器 (RTC_PRIVCFGR)

RTC privilege mode control register

该寄存器只能通过 APB 特权访问进行写操作。既可以为该寄存器设置写保护，也可以为该寄存器的每个位单独设置写保护，以防其遭到非安全访问，具体取决于 RTC_SECCFGR 配置（请参见*第 46.3.5 节：RTC 特权保护模式*）。

偏移地址：0x1C

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIV	INIT PRIV	CAL PRIV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS PRIV	WUT PRIV	ALRB PRIV	ALRA PRIV
rw	rw	rw										rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **PRIV**：RTC 特权保护 (RTC privilege protection)

0：所有 RTC 寄存器都可以通过 APB 特权或非特权访问进行写操作（由其他特权保护位保护的寄存器除外）。

1：所有 RTC 寄存器都只能通过 APB 特权访问进行写操作。

位 14 **INITPRIV**：初始化特权保护 (Initialization privilege protection)

0：在 RTC 初始化模式下，日历寄存器和预分频器寄存器可以通过 APB 特权或非特权访问进行写操作。

1：在 RTC 初始化模式下，日历寄存器和预分频器寄存器只能通过 APB 特权访问进行写操作。

位 13 **CALPRIV**：移位寄存器、夏令时存储、校准和参考时钟特权保护 (Shift register, Delight saving, calibration and reference clock privilege protection)

0：移位寄存器、夏令时存储、校准和参考时钟可以通过 APB 特权或非特权访问进行写操作。

1：移位寄存器、夏令时存储、校准和参考时钟只能通过 APB 特权访问进行写操作。

位 12:4 保留，必须保持复位值。

位 3 **TSPRIV**：时间戳特权保护 (Timestamp privilege protection)

0：RTC 时间戳配置和中断清除方法可以通过 APB 特权或非特权访问进行写操作。

1：RTC 时间戳配置和中断清除方法只能通过 APB 特权访问进行写操作。

位 2 **WUTPRIV**：唤醒定时器特权保护 (Wake-up timer privilege protection)

0：RTC 唤醒定时器配置和中断清除方法可以通过 APB 特权或非特权访问进行写操作。

1：RTC 唤醒定时器配置和中断清除方法只能通过 APB 特权访问进行写操作。

位 1 **ALRBPRIV**：闹钟 B 特权保护 (Alarm B privilege protection)

0：RTC 闹钟 B 配置和中断清除方法可以通过 APB 特权或非特权访问进行写操作。

1：RTC 闹钟 B 配置和中断清除方法只能通过 APB 特权访问进行写操作。

位 0 **ALRAPRIV**：闹钟 A 特权保护 (Alarm A privilege protection)

0：RTC 闹钟 A 配置和中断清除方法可以通过 APB 特权或非特权访问进行写操作。

1：RTC 闹钟 A 配置和中断清除方法只能通过 APB 特权访问进行写操作。

注： *有关读保护的详细信息，请参见第 46.3.5 节：RTC 特权保护模式。*

46.6.9 RTC 安全配置寄存器 (RTC_SECCFGR)

RTC secure configuration register

该寄存器只能通过 APB 安全访问进行写操作。

既可以为该寄存器设置全局写保护，也可以为该寄存器的每个位单独设置写保护，以防其遭到非特权访问，具体取决于 RTC_PRIVCFGR 配置（请参见*第 46.3.5 节：RTC 特权保护模式*）。

偏移地址：0x20

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC	INITSEC	CALSEC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSSEC	WUTSEC	ALRBSEC	ALRASEC
rw	rw	rw										rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **SEC**：RTC 全局保护 (RTC global protection)

0：所有 RTC 寄存器都可以通过 APB 安全或非安全访问进行写操作（由其他安全保护位保护的寄存器除外）。

1：所有 RTC 寄存器都只能通过 APB 安全访问进行写操作。

位 14 **INITSEC**：初始化保护 (Initialization protection)

0：在 RTC 初始化模式下，日历寄存器和预分频器寄存器可以通过 APB 安全或非安全访问进行写操作。

1：在 RTC 初始化模式下，日历寄存器和预分频器寄存器只能通过 APB 安全访问进行写操作。

位 13 **CALSEC**：移位寄存器、夏令时存储、校准和参考时钟保护 (Shift register, daylight saving, calibration and reference clock protection)

0：移位寄存器、夏令时存储、校准和参考时钟可以通过 APB 安全或非安全访问进行写操作。

1：移位寄存器、夏令时存储、校准和参考时钟只能通过 APB 安全访问进行写操作。

位 12:4 保留，必须保持复位值。

位 3 **TSSEC**：时间戳保护 (Timestamp protection)

0：RTC 时间戳配置和中断清除方法可以通过 APB 安全或非安全访问进行写操作。

1：RTC 时间戳配置和中断清除方法只能通过 APB 安全访问进行写操作。

位 2 **WUTSEC**：唤醒定时器范围 (Wake-up timer protection)
0：RTC 唤醒定时器配置和中断清除方法可以通过 APB 安全或非安全访问进行写操作。
1：RTC 唤醒定时器配置和中断清除方法只能通过 APB 安全访问进行写操作。

位 1 **ALRBSEC**：闹钟 B 保护 (Alarm B protection)
0：RTC 闹钟 B 配置和中断清除方法可以通过 APB 安全或非安全访问进行写操作。
1：RTC 闹钟 B 配置和中断清除方法只能通过 APB 安全访问进行写操作。

位 0 **ALRASEC**：闹钟 A 保护 (Alarm A protection)
0：RTC 闹钟 A 配置和中断清除方法可以通过 APB 安全或非安全访问进行写操作。
1：RTC 闹钟 A 配置和中断清除方法只能通过 APB 安全访问进行写操作。

注： *有关读保护的详细信息，请参见第 46.3.4 节：RTC 安全保护模式。*

46.6.10 RTC 写保护寄存器 (RTC_WPR)

RTC write protection register

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY[7:0]							
								w	w	w	w	w	w	w	w

位 31:8 保留，必须保持复位值。

位 7:0 **KEY[7:0]**：写保护密钥 (Write protection key)
可通过软件对该字节执行写操作。
读取该字节时，始终返回 0x00。
有关如何解锁 RTC 寄存器写保护的介绍，请参见 *RTC 寄存器写保护*。

46.6.11 RTC 校准寄存器 (RTC_CALR)

RTC calibration register

此寄存器受写保护。*第 2279 页的RTC 寄存器写保护*中介绍了写访问的过程。

可以对该寄存器进行写保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

可以对该寄存器进行写保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x28

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALP	CALW8	CALW 16	LPCAL	Res.	Res.	Res.	CALM[8:0]								
rw	rw	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **CALP**：将 RTC 的频率增加 488.5 ppm (Increase frequency of RTC by 488.5 ppm)

0：不增加 RTCCLK 脉冲。

1：每 2^{11} 个脉冲有效插入一个 RTCCLK 脉冲（将频率增加 488.5 ppm）。

此功能应与 CALM 结合使用，后者在高分辨率下会降低日历的频率。如果输入频率为 32768 Hz，则在 32 秒窗口中增加的 RTCCLK 脉冲数按如下公式计算：(512 × CALP) - CALM。

请参见*第 46.3.16 节：RTC 平滑数字校准*。

位 14 **CALW8**：使用 8 秒校准周期 (Use an 8-second calibration cycle period)

当 CALW8 置 1 时，选择 8 秒校准周期。

注意：当 CALW8 = 1 时，CALM[1:0] 将始终保持为 00。请参见第 46.3.16 节：RTC 平滑数字校准。

位 13 **CALW16**：使用 16 秒校准周期 (Use a 16-second calibration cycle period)

当 CALW16 置 1 时，选择 16 秒校准周期。如果 CALW8 = 1，则该位不可以置 1。

注意：当 CALW16= 1 时，CALM[0] 将始终保持为 0。请参见第 46.3.16 节：RTC 平滑数字校准。

位 12 **LPCAL**：RTC 低功耗模式 (RTC low-power mode)

0：校准窗口为 2^{20} 个 RTCCLK 脉冲，这是一种高功耗模式。只有在需要小于 32 秒的校准窗口时，才可以设置该模式。

1：校准窗口为 2^{20} 个 ck_apre 周期，这是超低功耗模式所需的配置。

位 11:9 保留，必须保持复位值。

位 8:0 **CALM[8:0]**：负校准 (Calibration minus)

在 2^{20} 个 RTCCLK 脉冲内屏蔽 CALM 个脉冲（如果输入频率为 32768 Hz，则为 32 秒）来降低日历的频率。其分辨率为 0.9537 ppm。

要提高日历的频率，则应将此功能与 CALP 结合使用。请参见*第 2283 页的第 46.3.16 节：RTC 平滑数字校准*。

46.6.12 RTC 移位控制寄存器 (RTC_SHIFTR)

RTC shift control register

此寄存器受写保护。*第 2279 页的RTC 寄存器写保护*中介绍了写访问的过程。

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x2C

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SUBFS[14:0]														
	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31 **ADD1S**：增加一秒钟 (Add one second)

0：无影响

1：对时钟/日历增加一秒钟

该位为只写位且始终读为 0。当移位操作挂起（RTC_ICSR 中的 SHPF = 1）时，对该位执行写操作无作用。

此函数应与 SUBFS 配合使用（请参见下文介绍），以便有效地向原子操作机制的时钟添加亚秒值。

位 30:15 保留，必须保持复位值。

位 14:0 **SUBFS[14:0]**：减少亚秒值 (Subtract a fraction of a second)

这些位为只写位且始终读为 0。当移位操作挂起（RTC_ICSR 中的 SHPF = 1）时，对该位执行写操作无作用。

写入 SUBFS 的值将加到同步预分频器计数器中。由于该计数器递减计数，此操作可有效地从时钟减去（延迟）以下时间：

延迟（秒）= SUBFS / (PREDIV_S + 1)

当 ADD1S 函数与 SUBFS 结合使用时，可有效地将亚秒值增加到时钟（提前时钟），使时钟提前以下时间：

提前（秒）= (1 - (SUBFS / (PREDIV_S + 1)))。

在混合 BCD 二进制模式下（BIN=10 或 11），SUBFS[14:BCDU+8] 必须写入 0。

注意：对 SUBFS 执行写操作将使 RSF 清零。软件随后会等待至 RSF = 1 以确定影子寄存器已更新为移位后的时间。

46.6.13 RTC 时间戳时间寄存器 (RTC_TSTR)

RTC timestamp time register

仅当 RTC_SR 中的 TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x30

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]			
									r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	r	r	r	r	r	r	r		r	r	r	r	r	r	r

位 31:23 保留，必须保持复位值。

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)。

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)。

位 15 保留，必须保持复位值。

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)。

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)。

位 7 保留，必须保持复位值。

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)。

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)。

46.6.14 RTC 时间戳日期寄存器 (RTC_TSDR)

RTC timestamp date register

仅当 RTC_SR 中的 TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x34

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
r	r	r	r	r	r	r	r			r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:13 **WDU[2:0]**：星期几的个位 (Week day units)

位 12 **MT**：月份的十位（BCD 格式）(Month tens in BCD format)

位 11:8 **MU[3:0]**：月份的个位（BCD 格式）(Month units in BCD format)

位 7:6 保留，必须保持复位值。

位 5:4 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 3:0 **DU[3:0]**：日期的个位（BCD 格式）(Date units in BCD format)

46.6.15 RTC 时间戳亚秒寄存器 (RTC_TSSSR)

RTC timestamp subsecond register

仅当 RTC_SR 中的 TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x38

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SS[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 SS[31:0]：亚秒值/同步二进制计数器值 (Subsecond value/synchronous binary counter values)
当发生时间戳事件时，SS[31:0] 是同步预分频器计数器的值。

46.6.16 RTC 闹钟 A 寄存器 (RTC_ALRMAR)

RTC alarm A register

仅当 RTC_CR 中的 ALRAE 复位时或在初始化模式下，才可以对该寄存器执行写操作。

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x40

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **MSK4**：闹钟 A 日期掩码 (Alarm A date mask)
0：如果日期/日匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，日期/日无关

位 30 **WDSEL**：星期几选择 (Week day selection)
0：DU[3:0] 代表日期的个位
1：DU[3:0] 代表星期几。DT[1:0] 为无关位。

位 29:28 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 27:24 **DU[3:0]**：日期个位或日（BCD 格式）(Date units or day in BCD format)

位 23 **MSK3**：闹钟 A 小时掩码 (Alarm A hours mask)
0：如果小时匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，小时无关

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)

位 15 **MSK2**：闹钟 A 分钟掩码 (Alarm A minutes mask)
0：如果分钟匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，分钟无关

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)

位 7 **MSK1**：闹钟 A 秒掩码 (Alarm A seconds mask)
0：如果秒匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，秒无关

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)。

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)。

46.6.17 RTC 闹钟 A 亚秒寄存器 (RTC_ALRMASSR)

RTC alarm A subsecond register

仅当 RTC_CR 中的 ALRAE 复位时或在初始化模式下，才可以对该寄存器执行写操作。

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x44

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSCLR	Res.	MASKSS[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw		rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SS[14:0]														
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	rw	rw

位 31 **SSCLR**：闹铃时清零同步计数器（仅限二进制模式）(Clear synchronous counter on alarm (Binary mode only))
0：同步二进制计数器（RTC_SSR 中的 SS[31:0]）处于自由运行状态。
1：同步二进制计数器（RTC_SSR 中的 SS[31:0]）从 0xFFFF FFFF 计数到 RTC_ALRABINR.SS[31:0] 值，并在计数到 RTC_ALRABINR.SS[31:0] 的一个 ck_apre 周期后自动重载为 0xFFFF FFFF。
注意：使用 BCD 模式或混合模式时（BIN = 00、10 或 11），SSCLR 必须保持为 0。

位 30 保留，必须保持复位值。

位 29:24 **MASKSS[5:0]**：屏蔽从该位开始的最高有效位 (Mask the most-significant bits starting at this bit)
0：不对闹钟 A 的亚秒进行比较。当秒单元递增时设置闹钟（假定其余位字段均匹配）。
1：在闹钟 A 比较中，SS[31:1] 为无关位。仅比较 SS[0]。
2：在闹钟 A 比较中，SS[31:2] 为无关位。仅比较 SS[1:0]。
...
31：在闹钟 A 比较中，SS[31] 为无关位。仅比较 SS[30:0]。
从 32 到 63：所有 32 个 SS 位均进行比较，并且必须全部匹配才能激活闹钟。
注意：在 BCD 模式下 (BIN=00)，同步计数器的上溢位（位 31:15）从不进行比较。仅当执行移位操作之后，这些位才不为 0。

位 23:15 保留，必须保持复位值。

位 14:0 **SS[14:0]**：亚秒值 (Subseconds value)

该值与同步预分频器计数器的内容进行比较以确定是否要激活闹钟 A。仅比较位 0 到 MASKSS-1。

该位字段是 RTC_ALRABINR 中 SS[14:0] 的镜像，因此也可以通过 RTC_ALRABINR 对其进行读/写操作。

注意：如果在 TAMP_ATCR1 中的 ATCKSEL[3] = 1 时将 SSCLR 置 1，则 SS[3:0] 必须为 0000。

46.6.18 RTC 闹钟 B 寄存器 (RTC_ALRMBR)

RTC alarm B register

仅当 RTC_CR 中的 ALRBE 复位时或在初始化模式下，才可以对该寄存器执行写操作。

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x48

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WD SEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **MSK4**：闹钟 B 日期掩码 (Alarm B date mask)

0：如果日期和日匹配，则闹钟 B 置 1

1：在闹钟 B 比较中，日期和日无关

位 30 **WDSEL**：星期几选择 (Week day selection)

0：DU[3:0] 代表日期的个位

1：DU[3:0] 代表星期几。DT[1:0] 为无关位。

位 29:28 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 27:24 **DU[3:0]**：日期个位或日（BCD 格式）(Date units or day in BCD format)

位 23 **MSK3**：闹钟 B 小时掩码 (Alarm B hours mask)

0：如果小时匹配，则闹钟 B 置 1

1：在闹钟 B 比较中，小时无关

位 22 **PM**：AM/PM 符号 (AM/PM notation)

0：AM 或 24 小时制

1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)

位 15 **MSK2**：闹钟 B 分钟掩码 (Alarm B minutes mask)
0：如果分钟匹配，则闹钟 B 置 1
1：在闹钟 B 比较中，分钟无关

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)

位 7 **MSK1**：闹钟 B 秒掩码 (Alarm B seconds mask)
0：如果秒匹配，则闹钟 B 置 1
1：在闹钟 B 比较中，秒无关

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)

46.6.19 RTC 闹钟 B 亚秒寄存器 (RTC_ALRMBSSR)

RTC alarm B subsecond register

仅当 RTC_CR 中的 ALRBE 复位时或在初始化模式下，才可以对该寄存器执行写操作。

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x4C

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSCLR	Res.	MASKSS[5:4]		MASKSS[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw		rw	rw	rw	rw	rw	rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SS[14:0]														
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	rw	rw

位 31 **SSCLR**：闹铃时清零同步计数器（仅限二进制模式）(Clear synchronous counter on alarm (Binary mode only))
0：同步二进制计数器（RTC_SSR 中的 SS[31:0]）处于自由运行状态。
1：同步二进制计数器（RTC_SSR 中的 SS[31:0]）从 0xFFFF FFFF 计数到 RTC_ALRBBINR.SS[31:0] 值，并在计数到 RTC_ALRBBINR.SS[31:0] 的一个 ck_apre 周期后自动重载为 0xFFFF FFFF。
注意：使用 BCD 模式或混合模式时（BIN = 00、10 或 11），SSCLR 必须保持为 0。

位 30 保留，必须保持复位值。

位 29:24 **MASKSS[5:0]**：屏蔽从该位开始的最高有效位 (Mask the most-significant bits starting at this bit)
0：不对闹钟 B 的亚秒进行比较。当秒单元递增时设置闹钟（假定其余位字段均匹配）。
1：在闹钟 B 比较中，SS[31:1] 为无关位。仅比较 SS[0]。
2：在闹钟 B 比较中，SS[31:2] 为无关位。仅比较 SS[1:0]。
...
31：在闹钟 B 比较中，SS[31] 为无关位。仅比较 SS[30:0]。
从 32 到 63：所有 32 个 SS 位均进行比较，并且必须全部匹配才能激活闹钟。
注意：在 BCD 模式下 (BIN=00)，同步计数器的上溢位（位 15）从不进行比较。。仅当执行移位操作之后，该位才不为 0。

位 23:15 保留，必须保持复位值。

位 14:0 **SS[14:0]**：亚秒值 (Subseconds value)

该值与同步预分频器计数器的内容进行比较以确定是否要激活闹钟 B。仅比较位 0 到 MASKSS-1。

该位字段是 RTC_ALRBBINR 中 SS[14:0] 的镜像，因此也可以通过 RTC_ALRBBINR 对其进行读/写操作。

注意：如果在 TAMP_ATCR1 中的 ATCKSEL[3] = 1 时将 SSCLR 置 1，则 SS[3:0] 必须为 0000。

46.6.20 RTC 状态寄存器 (RTC_SR)

RTC status register

既可以为该寄存器设置全局保护，也可以为该寄存器的每个位单独设置保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

既可以为该寄存器设置全局保护，也可以为该寄存器的每个位单独设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x50

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSF	TSOVF	TSF	WUTF	ALRBF	ALRAF
										r	r	r	r	r	r

位 31:6 保留，必须保持复位值。

位 5 **ITSF**：内部时间戳标志 (Internal timestamp flag)

发生内部时间戳事件时，由硬件将该标志置 1。

位 4 **TSOVF**：时间戳上溢标志 (Timestamp overflow flag)

当在 TSF 已置 1 的情况下发生时间戳事件时，由硬件将此标志置 1。

建议仅在 TSF 位清零之后再检查并清零 TSOVF 位。否则，如果时间戳事件恰好在清零 TSF 位之前刚刚发生，则上溢事件可能会被漏掉。

位 3 **TSF**：时间戳标志 (Timestamp flag)

发生时间戳事件时，由硬件将该标志置 1。

如果 ITSF 标志已置 1，则必须将 TSF 与 ITSF 一起清零。

注意：如果 TAMPTS = 1，并在 Tamper 事件后的 3 个 ck_apre 周期内读取 Tamper 标志，则 TSF 不会置 1。更多详细信息，请参见 Tamper 事件的时间戳。

位 2 **WUTF**：唤醒定时器标志 (Wake-up timer flag)

当唤醒自动重载计数器计数到 0 时，由硬件将该标志置 1。

如果 WUTOCLR[15:0] 不等于 0x0000，则当唤醒自动重载计数器计数到 WUTOCLR 值时，WUTF 由硬件清零。

如果 WUTOCLR[15:0] 等于 0x0000，则 WUTF 必须由软件清零。

软件必须在 WUTF 再次置 1 的 1.5 个 RTCCLK 周期之前将该标志清零。

位 1 **ALRBF**：闹钟 B 标志 (Alarm B flag)
当时间/日期寄存器（RTC_TR 和 RTC_DR）与闹钟 B 寄存器 (RTC_ALRMBR) 匹配时，由硬件将该标志置 1。

位 0 **ALRAF**：闹钟 A 标志 (Alarm A flag)
当时间/日期寄存器（RTC_TR 和 RTC_DR）与闹钟 A 寄存器 (RTC_ALRMAR) 匹配时，由硬件将该标志置 1。

注： *该寄存器的位在 RTC_SCR 寄存器中相应的清零位置 1 后的几个 APB 时钟周期之后清零。清除标志后，在退出中断程序之前读取该标志，直到读为 0。*

46.6.21 RTC 非安全屏蔽中断状态寄存器 (RTC_MISR)

RTC nonsecure masked interrupt status register

既可以为该寄存器设置全局保护，也可以为该寄存器的每个位单独设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x54

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSMF	TSOVMF	TSMF	WUTMF	ALRBMF	ALRAMF
										r	r	r	r	r	r

位 31:6 保留，必须保持复位值。

位 5 **ITSMF**：内部时间戳非安全屏蔽标志 (Internal timestamp nonsecure masked flag)
当发生内部时间戳事件并产生时间戳非安全中断时，由硬件将该标志置 1。

位 4 **TSOVMF**：时间戳溢出非安全屏蔽标志 (Timestamp overflow nonsecure masked flag)
当在 TSMF 已置 1 的情况下发生时间戳中断时，由硬件将该标志置 1。
建议仅在 TSF 位清零之后再检查并清零 TSOVF 位。否则，如果时间戳事件恰好在清零 TSF 位之前刚刚发生，则上溢事件可能会被漏掉。

位 3 **TSMF**：时间戳非安全屏蔽标志 (Timestamp nonsecure masked flag)
发生时间戳非安全中断时，由硬件将该标志置 1。
如果 ITSF 标志已置 1，则必须将 TSF 与 ITSF 一起清零。

位 2 **WUTMF**：唤醒定时器非安全屏蔽标志 (Wake-up timer nonsecure masked flag)
发生唤醒定时器非安全中断时，由硬件将该标志置 1。
软件必须在 WUTF 再次置 1 的 1.5 个 RTCCLK 周期之前将该标志清零。

位 1 **ALRBMF**：闹钟 B 非安全屏蔽标志 (Alarm B nonsecure masked flag)
发生闹钟 B 非安全中断时，由硬件将该标志置 1。

位 0 **ALRAMF**：闹钟 A 屏蔽标志 (Alarm A masked flag)
发生闹钟 A 非安全中断时，由硬件将该标志置 1。

注： *该寄存器的位在 RTC_SCR 寄存器中相应的清零位置 1 后的几个 APB 时钟周期之后清零。清除标志后，在退出中断程序之前读取该标志，直到读为 0。*

46.6.22 RTC 安全屏蔽中断状态寄存器 (RTC_SMISR)

RTC secure masked interrupt status register

既可以为该寄存器设置全局保护，也可以为该寄存器的每个位单独设置保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

既可以为该寄存器设置全局保护，也可以为该寄存器的每个位单独设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x58

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSMF	TSOVMF	TSMF	WUTMF	ALRBMF	ALRAMF
										r	r	r	r	r	r

位 31:6 保留，必须保持复位值。

位 5 **ITSMF**：内部时间戳中断安全屏蔽标志 (Internal timestamp interrupt secure masked flag)
当发生内部时间戳事件并产生时间戳安全中断时，由硬件将该标志置 1。

位 4 **TSOVMF**：时间戳溢出中断安全屏蔽标志 (Timestamp overflow interrupt secure masked flag)
当在 TSMF 已置 1 的情况下发生时间戳安全中断时，由硬件将该标志置 1。
建议仅在 TSF 位清零之后再检查并清零 TSOVF 位。否则，如果时间戳事件恰好在清零 TSF 位之前刚刚发生，则上溢事件可能会被漏掉。

位 3 **TSMF**：时间戳中断安全屏蔽标志 (Timestamp interrupt secure masked flag)
发生时间戳安全中断时，由硬件将该标志置 1。
如果 ITSF 标志已置 1，则必须将 TSF 与 ITSF 一起清零。

位 2 **WUTMF**：唤醒定时器中断安全屏蔽标志 (Wake-up timer interrupt secure masked flag)
发生唤醒定时器安全中断时，由硬件将该标志置 1。
软件必须在 WUTF 再次置 1 的 1.5 个 RTCCLK 周期之前将该标志清零。

位 1 **ALRBMF**：闹钟 B 中断安全屏蔽标志 (Alarm B interrupt secure masked flag)
发生闹钟 B 安全中断时，由硬件将该标志置 1。

位 0 **ALRAMF**：闹钟 A 中断安全屏蔽标志 (Alarm A interrupt secure masked flag)
发生闹钟 A 安全中断时，由硬件将该标志置 1。

注： *该寄存器的位在 RTC_SCR 寄存器中相应的清零位置 1 后的几个 APB 时钟周期之后清零。清除标志后，在退出中断程序之前读取该标志，直到读为 0。*

46.6.23 RTC 状态清零寄存器 (RTC_SCR)

RTC status clear register

既可以为该寄存器设置全局保护，也可以为该寄存器的每个位单独设置保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

既可以为该寄存器设置全局保护，也可以为该寄存器的每个位单独设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x5C

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CITSF	CTSOVF	CTSF	CWUTF	CALRBF	CALRAF
										w	w	w	w	w	w

位 31:6 保留，必须保持复位值。

位 5 **CITSF**：清零内部时间戳标志 (Clear internal timestamp flag)
向该位写入 1 可清零 RTC_SR 寄存器中的 ITSF 位。

位 4 **CTSOVF**：清零时间戳上溢标志 (Clear timestamp overflow flag)
向该位写入 1 可清零 RTC_SR 寄存器中的 TSOVF 位。
建议仅在 TSF 位清零之后再检查并清零 TSOVF 位。否则，如果时间戳事件恰好在清零 TSF 位之前刚刚发生，则上溢事件可能会被漏掉。

位 3 **CTSF**：清零时间戳标志 (Clear timestamp flag)
向该位写入 1 可清零 RTC_SR 寄存器中的 TSF 位。
如果 ITSF 标志已置 1，则必须通过将 CRSF 和 CITSF 置 1 来将 TSF 与 ITSF 一起清零。

位 2 **CWUTF**：清零唤醒定时器标志 (Clear wake-up timer flag)
向该位写入 1 可清零 RTC_SR 寄存器中的 WUTF 位。

位 1 **CALRBF**：清零闹钟 B 标志 (Clear alarm B flag)
向该位写入 1 可清零 RTC_SR 寄存器中的 ALRBF 位。

位 0 **CALRAF**：清零闹钟 A 标志 (Clear alarm A flag)
向该位写入 1 可清零 RTC_SR 寄存器中的 ALRAF 位。

46.6.24 RTC 选项寄存器 (RTC_OR)

RTC option register

偏移地址：0x60

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSCOEN[1:0]		OUT2_RMP
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:1 **LSCOEN[1:0]**：LSCO LSE 时钟输出使能 (LSCO LSE clock output enable)

00：无影响

01：LSE 在 PC13 上输出

10：LSE 在 PI1 上输出

11：无影响

注意：注意：仅当这些 I/O 上没有使能其他 RTC/TAMP 输出时，LSCO 功能才会在 PC13 或 PI1 上输出。

位 0 **OUT2_RMP**：RTC_OUT2 映射

0：RTC_OUT2 映射在 PB2 上

1：RTC_OUT2 映射在 PI1 上

46.6.25 RTC 闹钟 A 二进制模式寄存器 (RTC_ALRABINR)

RTC alarm A binary mode register

仅当 RTC_CR 中的 ALRAE 复位时或在初始化模式下，才可以对该寄存器执行写操作。

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x70

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SS[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **SS[31:0]**：二进制模式下的同步计数器闹钟值 (Synchronous counter alarm value in Binary mode)

该值与同步计数器的内容进行比较以确定是否要激活闹钟 A。仅比较位 0 到 MASKSS-1。SS[14:0] 是 RTC_ALRMASSRR 中 SS[14:0] 的镜像，因此也可以通过 RTC_ALRMASSR 对其进行读/写操作。

注意：如果在 TAMP_ATCR1 中的 ATCKSEL[3] = 1 时将 SSCLR 置 1，则 SS[3:0] 必须为 0000。

46.6.26 RTC 闹钟 B 二进制模式寄存器 (RTC_ALRBBINR)

RTC alarm B binary mode register

仅当 RTC_CR 中的 ALRBE 复位时或在初始化模式下，才可以对该寄存器执行写操作。

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 46.3.4 节：RTC 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 46.3.5 节：RTC 特权保护模式*。

偏移地址：0x74

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SS[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **SS[31:0]**：二进制模式下的同步计数器闹钟值 (Synchronous counter alarm value in Binary mode)
该值与同步计数器的内容进行比较以确定是否要激活闹钟 B。仅比较位 0 到 MASKSS-1。SS[14:0] 是 RTC_ALRMBSSRR 中 SS[14:0] 的镜像，因此也可以通过 RTC_ALRMBSSR 对其进行读/写操作。

注意：如果在 TAMP_ATCR1 中的 ATCKSEL[3] = 1 时将 SSCLR 置 1，则 SS[3:0] 必须为 0000。

46.6.27 RTC 硬件配置寄存器 2 (RTC_HWCFGR2)

RTC hardware configuration register 2

偏移地址：0x3EC

复位值：0x0000 0100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CID_WIDTH[3:0]				RIF[3:0]				Res.	Res.	Res.	Res.
				r	r	r	r	r	r	r	r				

位 31:12 保留，必须保持复位值。

位 11:8 **CID_WIDTH[3:0]**：CID 长度
0x1：CID 长度为 1 位
0x2：CID 长度为 2 位
0x3：CID 长度为 3 位
0x4：CID 长度为 4 位

位 7:4 **RIF[3:0]**：CID 分区过滤
0x0：禁止 CID 分区过滤
0x1：使能 CID 分区过滤

位 3:0 保留，必须保持复位值。

46.6.28 RTC 硬件配置寄存器 (RTC_HWCFGR1)

RTC hardware configuration register

偏移地址：0x3F0

复位值：0x1103 1111

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BINARY[3:0]				TRUST_ZONE[3:0]				OPTIONREG_OUT[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMESTAMP[3:0]				SMOOTH_CALIB[3:0]				WAKEUP[3:0]				ALARMB[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **BINARY[3:0]**：二进制模式 (Binary mode)
0x0：未实现二进制模式
0x1：已实现二进制模式

位 27:24 **TRUST_ZONE[3:0]**：信任区域
0x0：未实现信任区域
0x1：已实现信任区域

位 23:16 **OPTIONREG_OUT[7:0]**：
0：未实现配置寄存器
1：配置寄存器位 0 已实现
2：配置寄存器位 1:0 已实现
...
32：配置寄存器位 31:0 已实现

位 15:12 **TIMESTAMP[3:0]**：
0x0：未实现时间戳
0x1：已实现时间戳

位 11:8 **SMOOTH_CALIB[3:0]**：
0x0：未实现平滑校准
0x1：已实现平滑校准

位 7:4 **WAKEUP[3:0]**：
0x0：未实现唤醒定时器
0x1：已实现唤醒定时器

位 3:0 **ALARMB[3:0]**：
0x0：未实现闹钟 B
0x1：已实现闹钟 B

46.6.29 RTC 版本寄存器 (RTC_VERR)

RTC version register

偏移地址：0x3F4

复位值：0x0000 003X

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本 (Major revision)

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)

46.6.30 RTC 标识寄存器（RTC_IPIDR）

RTC identification register

偏移地址：0x3F8

复位值：0x0012 0033

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：标识符 (Identifier)

46.6.31 RTC 大小标识寄存器 (RTC_SIDR)

RTC size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识符 (Size identifier)

46.6.32 RTC 寄存器映射

表 405. RTC 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	RTC_TR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT [1:0]		HU[3:0]				Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	Reset value										0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x04	RTC_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YT[3:0]				YU[3:0]				WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT [1:0]		DU[3:0]			
	Reset value									0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1			0	0	0	0	0	1
0x08	RTC_SSR	SS[31:16]																SS[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	RTC_ICSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RECALPF	Res.	Res.	Res.	BCDU [2:0]			BIN [1:0]		INIT	INITF	RSF	INITS	SHPF	WUT WF	Res.	Res.
	Reset value																0				0	0	0	0	0	0	0	0	0	0	1		
0x10	RTC_PRER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREDIV_A[6:0]							PREDIV_S[14:0]															
	Reset value										1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0x14	RTC_WUTR	WUTOCLR[15:0]																WUT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x18	RTC_CR	OUT2EN	TAMPALRM_TYPE	TAMPALRM_PU	ALRBFCLR	ALRAFCLR	TAMPOE	TAMPTS	ITSE	COE	O SEL [1:0]		POL	COSEL	BKP	SUB1H	ADD1H	TSIE	WUTIE	ALRBIE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	Res.	FMT	BYPSHAD	REFCKON	TSEDGE	WUCK SEL[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	RTC_ PRIVCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIV	INITPRIV	CALPRIV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSPRIV	WUTPRIV	ALRBPRIV	ALRAPRIV
	Reset value																	0	0	0										0	0	0	0
0x20	RTC_ SECCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEC	INITSEC	CALSEC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSSEC	WUTSEC	ALRBSEC	ALRASEC
	Reset value																	0	0	0										0	0	0	0
0x24	RTC_WPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x28	RTC_ CALR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALP	CALW8	CALW16	LPCAL	Res.	Res.	Res.	CALM[8:0]								
	Reset value																	0	0	0	0				0	0	0	0	0	0	0	0	0
0x2C	RTC_SHIFTR	ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBFS[14:0]														
	Reset value	0																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 405. RTC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x30	RTC_TSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]				Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	Reset value										0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x34	RTC_TSDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT [1:0]		DU[3:0]			
	Reset value																	0	0	0	0	0	0	0	0			0	0	0	0	0	0
0x38	RTC_TSSSR	SS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	RTC_ALRMAR	MSK4	WDSEL	DT [1:0]		DU[3:0]				MSK3	PM	HT [1:0]		HU[3:0]				MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	RTC_ ALRMASSR	SSCLR	Res.	MASKSS [5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]														
	Reset value	0		0	0	0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	RTC_ALRMBR	MSK4	WDSEL	DT [1:0]		DU[3:0]				MSK3	PM	HT [1:0]		HU[3:0]				MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	RTC_ ALRMBSSR	SSCLR	Res.	MASKSS [5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]														
	Reset value	0		0	0	0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	RTC_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSF	TSOVF	TSF	WUTF	ALRBF	ALRAF
	Reset value																										0	0	0	0	0	0	0
0x54	RTC_MISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSMF	TSOVMF	TSMF	WUTMF	ALRBMF	ALRAMF
	Reset value																										0	0	0	0	0	0	0
0x58	RTC_SMISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSMF	TSOVMF	TSMF	WUTMF	ALRBMF	ALRAMF
	Reset value																										0	0	0	0	0	0	0
0x5C	RTC_SCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CITSF	CTSOVF	CTSF	CWUTF	CALRBF	CALRAF
	Reset value																											0	0	0	0	0	0
0x60	RTC_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSCOEN[1:0]		OUT2_RMP
	Reset value																														0	0	0
0x70	RTC_ ALRABINR	SS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 405. RTC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x74	RTC_ ALRBBINR	SS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3EC	RTC_ HWCFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CID_WIDTH [3:0]				RIF[3:0]				Res.	Res.	Res.	Res.
	Reset value																					0	0	0	1	0	0	0	0				
0x3F0	RTC_ HWCFGR1	BINARY[3:0]				TRUST_ ZONE[3:0]				OPTION_REG_OUT[7:0]								TIMESTAMP [3:0]				SMOOTH_ CALIB[3:0]				WAKEUP [3:0]				ALARMB [3:0]			
	Reset value	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
0x3F4	RTC_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV [3:0]				MINREV [3:0]			
	Reset value																									0	0	1	1	x	x	x	x
0x3F8	RTC_IPIDR	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
0x3FC	RTC_SIDR	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 2.5 节*。

47 Tamper 和备份寄存器 (TAMP)

47.1 简介

防 Tamper 检测电路用于保护敏感数据免受外部攻击。在所有低功耗模式和 V_{BAT} 模式下，32 个 32 位备份寄存器的内容均保持不变。该防 Tamper 检测电路具有 12 个 Tamper 引脚和 13 个内部 Tamper 事件，可对备份寄存器以及设备中的其他机密信息实施保护。外部 Tamper 引脚可配置为边沿检测、带或不带滤波的电平检测或主动 Tamper 模式，该模式通过自动检查 Tamper 引脚是否存在外部开路或短路的情况来提高安全性。

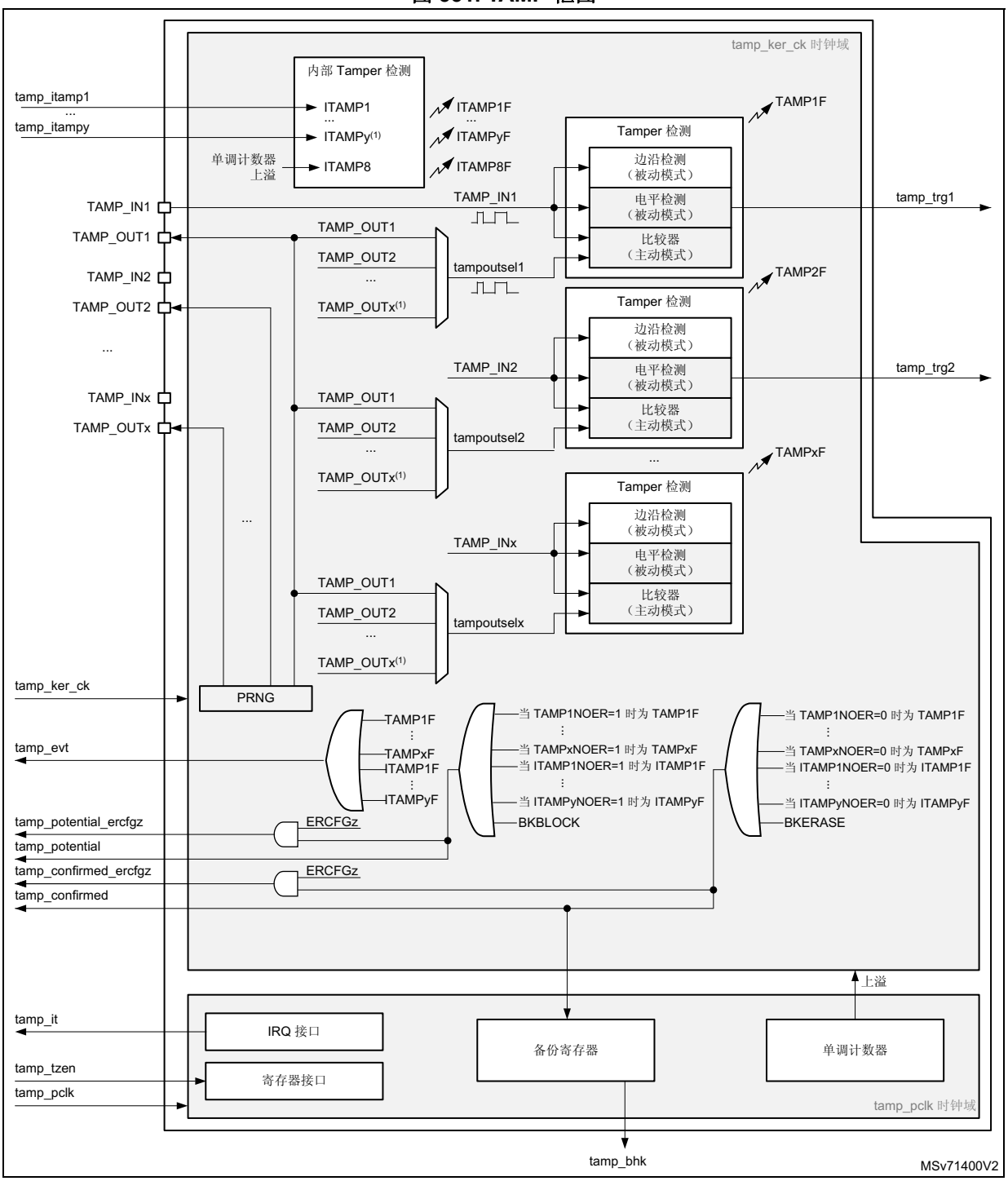
47.2 TAMP 主要特性

- Tamper 检测可以选择擦除备份寄存器、备份 SRAM、SRAM3 和加密外设。受 Tamper 保护的设备资源被称为“设备机密信息”。
- 32 个 32 位备份寄存器：
  - 备份寄存器 (TAMP_BKPxR) 在备份域中实现，它可在 V_{DD} 电源关闭时通过 V_{BAT} 供电。
- 多达 12 个 Tamper 引脚（针对 8 个外部 Tamper 检测事件）：
  - 主动 Tamper 模式：连续比较 Tamper 输出与输入，以防止物理开路/短路攻击。
  - 灵活的主动 Tamper I/O 管理：从 5 个网格（每个输入均与其专用输出相关联）到 8 个网格（多达 8 个 Tamper 输入共用一个输出）
  - 被动 Tamper：超低功耗边沿或电平检测，支持内部上拉硬件管理。
  - 可配置数字滤波器。
- 13 个内部 Tamper 事件，可防止瞬态或环境扰动攻击
- 每个 Tamper 源都可以配置为两种模式：
  - 确认模式：检测到 Tamper 后立即擦除机密信息（包括擦除备份寄存器）
  - 潜在模式：检测到 Tamper 后擦除大多数机密信息（通过软件启动）
- 任何 Tamper 检测都可以生成一个 RTC 时间戳事件。
- TrustZone 支持：
  - Tamper 安全或非安全配置。
  - 备份寄存器配置，分为以下 3 个可配置大小的区域：
    1 个读/写安全区域。
    1 个写安全/读非安全区域。
    1 个读/写非安全区域。
  - 用于安全 AES 的启动硬件密钥，存储在备份寄存器中，可防止读写访问。
- Tamper 配置和备份寄存器特权保护
- 两个单调计数器。

47.3 TAMP 功能描述

47.3.1 TAMP 框图

图 531. TAMP 框图



1. 外部和内部 Tamper 引脚的数量取决于产品。

47.3.2 TAMP 引脚和内部信号

表 406. TAMP 输入/输出引脚

引脚名称	信号类型	说明
TAMP_INx（x = 引脚索引）	输入	Tamper 输入引脚
TAMP_OUTx（x = 引脚索引）	输出	Tamper 输出引脚（仅限主动模式）

表 407. TAMP 内部输入/输出信号

内部信号名称	信号类型	说明
tamp_ker_ck	输入	TAMP 内核时钟，连接到 rtc_ker_ck，在本文档中也称为 RTCCLK
tamp_pclk	输入	TAMP APB 时钟，连接到 rtc_pclk
tamp_itamp[y]（y = 信号索引）	输入	内部 Tamper 事件源
tamp_tzen	输入	使能 TAMP TrustZone
tamp_evt	输出	Tamper 事件检测标志（内部或外部 Tamper），无论采用确认模式配置还是潜在模式配置。
tamp_potential	输出	潜在 Tamper 检测信号，用于设备机密信息(1)保护。 该信号在以下情况下有效： – 在潜在模式下生成 Tamper 事件检测标志（内部或外部 Tamper）。 – 或者通过向 BKBLOCK 写入 1 来完成软件请求
tamp_confirmed	输出	确认 Tamper 检测信号，用于设备机密信息(1)保护。 该信号在以下情况下有效： – 在确认模式下生成 Tamper 事件检测标志（内部或外部 Tamper）。 – 或者通过向 BKERASE 写入 1 来完成软件请求
tamp_potential_ercfgz（z = 信号索引）	输出	仅在 ERCFGz = 1 时生成的潜在 Tamper 检测信号。 该信号在以下情况下有效： – 在潜在模式下生成 Tamper 事件检测标志（内部或外部 Tamper）。 – 或者通过向 BKBLOCK 写入 1 来完成软件请求
tamp_confirmed_ercfgz（z = 信号索引）	输出	仅在 ERCFGz = 1 时生成的确认 Tamper 检测信号。 该信号在以下情况下有效： – 在确认模式下生成 Tamper 事件检测标志（内部或外部 Tamper）。 – 或者通过向 BKERASE 写入 1 来完成软件请求
tamp_it	输出	TAMP 中断（有关详细信息，请参见*第 47.5 节：TAMP 中断*）

表 407. TAMP 内部输入/输出信号（续）

内部信号名称	信号类型	说明
tamp_trg[x] （x = 信号索引）	输出	Tamper 检测触发信号
tamp_bhk	输出	Tamper 启动硬件密钥总线

1. 请参见*表 408：TAMP 互连*。

TAMP 内核时钟通常是 32.768 kHz 的 LSE，但也可以在 RCC 中选择其他时钟源（更多详细信息，请参见 RCC）。一些检测模式在某些低功耗模式或 V_{BAT} 模式下不可用，具体取决于所选时钟（更多详细信息，请参见*第 47.4 节：TAMP 低功耗模式*）。

表 408. TAMP 互连

信号名称	源/目标
tamp_tzen	连接到 1
tamp_evt	rtc_tamp_evt 用于生成时间戳事件
tamp_potential	tamp_potential 信号用于阻止对下列设备机密信息的读写访问： – 备份寄存器 – SRAM3 BSEC 中 RHUK（根硬件唯一密钥）和 BHK（启动硬件密钥）的 SAES 硬件总线被阻断。 tamp_potential 信号用于擦除下列设备机密信息： – SAES、HASH 外设 – PKA SRAM 正在执行擦除操作时，会阻止对设备机密信息的访问。
tamp_confirmed	tamp_confirmed 信号用于擦除下列设备机密信息： – 备份寄存器 – SRAM3 – SAES、HASH 外设 – PKA SRAM 正在执行擦除操作时，会阻止对设备机密信息的访问。 BSEC 中 RHUK（根硬件唯一密钥）的 SAES 硬件总线被阻断。
tamp_potential_ercfg0	当 TAMP_ERCFGR 中的 ERCFG0 位置 1 时，tamp_potential_ercfg0 信号用于阻止对下列设备机密信息的读写访问： – 备份 SRAM
tamp_confirmed_ercfg0	当 TAMP_ERCFGR 中的 ERCFG0 位置 1 时，tamp_confirmed_ercfg0 信号用于擦除下列设备机密信息： – 备份 SRAM(1) 正在执行擦除操作时，会阻止对设备机密信息的访问。
tamp_itamp1	备份域电压阈值监视(2)
tamp_itamp2	温度监视(2)
tamp_itamp3	LSE 监视 (LSECSS)(3)
tamp_itamp4	HSE 监控（CSS + 超频检测）(4)
tamp_itamp5	RTC 日历上溢 (rtc_calovf)

表 408. TAMP 互连（续）

信号名称	源/目标
tamp_itamp6	JTAG/SWD 访问
tamp_itamp7	ADC2 (adc2_awd1) 模拟看门狗监视 1
tamp_itamp8(5)	单调计数器 1 上溢
tamp_itamp9	加密外设故障（SAES、CRYP、PKA 或 TRNG）
tamp_itamp10(6)	单调计数器 2 上溢
tamp_itamp11	Tamper 标志置 1 时产生的 IWDG1 复位（潜在 Tamper 超时）
tamp_itamp12	ADC2 (adc2_awd2) 模拟看门狗监视 2
tamp_itamp13	ADC2 (adc2_awd3) 模拟看门狗监视 3
tamp_bhk	saes_bhk。该总线用于在安全 AES 协处理器中加载启动硬件密钥。

1. 首次对 Backup RAM 进行写访问时其内容将被擦除（如果系统处于 VBAT 模式，则在下一个 POR 之后及首次对 Backup RAM 进行写操作时也会如此。）
2. 必须通过置位 MONEN 来使能该监控功能（位于 *PWR 控制寄存器 2 (PWR_CR2)*）。
3. 必须通过置位 LSECSSEN 来使能该监控功能（位于 RCC_BDCR 寄存器）。
4. HSE 监控是在配置为 HSE 超频检测时，HSECSS 输出与 LPTIM3 输出的或运算。必须通过 *RCC 振荡器时钟使能设置寄存器 (RCC_OCENSETR)* 中的 HSECSSON 位使能 HSECSS。
5. 该信号在 TAMP 外设中生成。
6. 该信号在 TAMP 外设中生成。

47.3.3 RTC 和 TAMP 控制的 GPIO

请参见*第 46.3.3 节：RTC 和 TAMP 控制的 GPIO*。

47.3.4 TAMP 寄存器写保护

系统复位后，可通过电源控制外设中的 DBP 位来保护 TAMP 寄存器（包括备份寄存器），以防止非正常的写访问（请参见 PWR 电源控制部分）。必须将 DBP 位置 1 才能使能 TAMP 寄存器的写访问。

47.3.5 TAMP 安全保护模式

默认情况下，备份域上电复位后，除了 TAMP 安全配置寄存器 (TAMP_SECCFGR) 只能在安全模式下写入之外，所有其他 TAMP 寄存器都可以在安全模式和非安全模式下读取或写入。TAMP 保护配置不受系统复位影响。

- 在以下情况下，TAMP_SECCFGR 寄存器中的 TAMPSEC 位置 1：
  - 只能在安全模式下写入 TAMP 寄存器，但具有自己的保护设置的备份寄存器除外。
  - 始终可以在安全模式和非安全模式下读取 TAMP_SECCFGR、TAMP_PRIVCFGR 和 TAMP_MISR。所有其他 TAMP 寄存器都只能在安全模式下读取，但具有自己的保护设置的备份寄存器和单调计数器除外。
- 当 TAMP_SECCFGR 寄存器中的 CNT1SEC 位置 1 时：TAMP_COUNT1R 只能在安全模式下读取和写入。
- 当 TAMP_SECCFGR 寄存器中的 CNT2SEC 位置 1 时：TAMP_COUNT2R 只能在安全模式下读取和写入。

对受到安全保护的寄存器进行非安全访问会遭到拒绝：

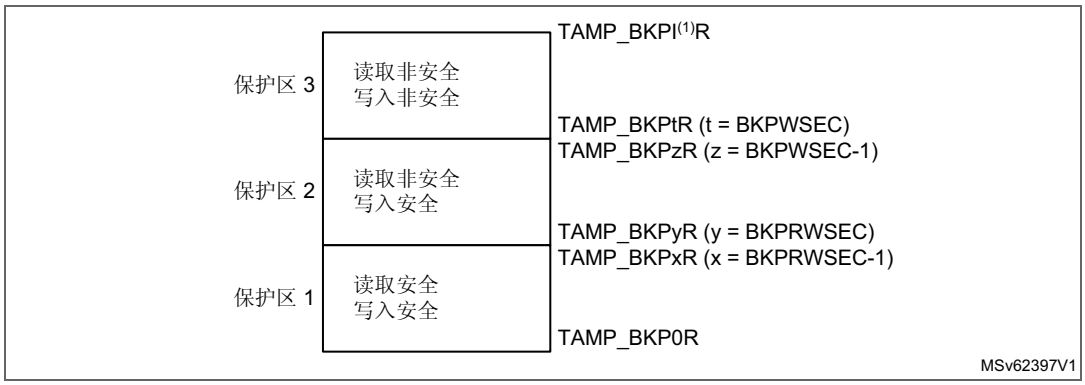
- 不生成任何总线错误。
- 通过 TZIC（TrustZone 非法访问控制器）中的标志/中断生成通知。
- 写保护时，这些位不会被写入。
- 读保护时，这些位的值读为 0。

只要至少一个功能配置为安全，RCC 中的 TAMP 复位和时钟控制也就安全。

47.3.6 备份寄存器保护区

备份寄存器保护通过 BKPRWSEC[7:0] 和 BKPWSEC[7:0] 进行配置（请参见下图）：

图 532. 备份寄存器保护区



1. I = 最后一个备份寄存器索引

如果 TZEN =1，则 TAMP_PRIVCFGR 中的位 BKPWPRIV 和 BKPRWPRIV 只能在安全模式下写入。

47.3.7 TAMP 特权保护模式

默认情况下，备份域上电复位后，除了 TAMP 特权配置寄存器 (TAMP_PRIVCFGR) 只能在特权模式下写入之外，所有其他 TAMP 寄存器都可以在特权模式和非特权模式下读取或写入。TAMP 保护配置不受系统复位影响。

当 TAMP_PRIVCFGR 寄存器中的 TAMPPRIV 位置 1 时：

- 只能在特权模式下写入 TAMP 寄存器，但具有自己的保护设置的备份寄存器和单调计数器除外。
- 当 TAMP_PRIVCFGR 寄存器中的 CNT1PRIV 位置 1 时：TAMP_COUNT1R 只能在特权模式下读取和写入。
- 当 TAMP_PRIVCFGR 寄存器中的 CNT2PRIV 位置 1 时：TAMP_COUNT2R 只能在特权模式下读取和写入。
- 始终可以在特权和非特权模式下读取TAMP_SECCFGR TAMP_PRIVCFGR。所有其他 TAMP 寄存器都只能在特权模式下读取，但具有自己的保护设置的备份寄存器和单调计数器除外。

对于保护区 1，备份寄存器保护通过 BKPRWSEC[7:0] 和 BKPRWPRIV 进行配置；对于保护区 2，备份寄存器保护通过 BKPRWSEC[7:0]、BKPWSEC[7:0] 和 BKPWPRIV 进行配置（请参见*图 532*）。当 BKPRWPRIV 位置 1 时，BHKLOCK 位只能在特权模式下写入。

对受特权保护的寄存器的非特权访问将被拒绝：

- 不生成任何总线错误。
- 写保护时，这些位不会被写入。
- 读保护时，这些位的值读为 0。

47.3.8 启动硬件密钥 (BHK)

前八个备份寄存器 TAMP_BKP0R 到 TAMP_BKP7R 可用于存储安全 AES 的启动硬件密钥。

为此，这些寄存器必须位于保护区 1 中：BKPRWSEC 必须大于或等于 8。

一旦备份寄存器中写入启动硬件密钥，TAMP_SECCFGR 寄存器中的 BHKLOCK 位便必须置 1。一旦 BHKLOCK 置 1，便无法再通过软件访问 8 个备份寄存器：读取这些寄存器时返回 0，写入这些寄存器时被忽略。BHKLOCK 无法由软件清零，它会在发生 Tamper 事件后由硬件清零。此外，也可以使用 BKERASE 命令清零 （在所有情况下，备份寄存器也会一并擦除）。

有关在 SAES 中下载启动硬件密钥的操作步骤的详细信息，请参见安全 AES 协处理器 (SAES) 章节。

47.3.9 Tamper 检测

Tamper 检测的主要目的是保护设备机密信息免受外部攻击。检测的对象是 TAMP_INx（x = 引脚索引）I/O 上的事件，或通过内部监视器检测超范围的设备条件。

Tamper 检测可配置用于以下目的：

- 擦除备份寄存器和存储在 SRAM 或外设中的其他设备机密信息（如*表 408：TAMP 互连*所列）。
- 阻止对备份寄存器和存储在 SRAM 或外设中的其他设备机密信息（如*表 408：TAMP 互连*所列）的读/写访问
- 生成中断，可使设备从低功耗模式唤醒
- 为低功耗定时器生成硬件触发信号，或生成 RTC 时间戳事件

外部 I/O Tamper 检测支持 2 种主要配置：

- 被动模式：监视 TAMP_INx I/O，并基于边沿或电平执行 Tamper 检测。
- 主动模式：连续比较 TAMP_INx（x = 引脚索引）与 TAMP_OUTy（y = 引脚索引），以实现开路/短路检测。

可以使用数字滤波器进行外部 Tamper 检测，以避免发生误检。此外，可以将每个 Tamper 源都配置为潜在模式，以避免硬件在 Tamper 检测时启动机密信息擦除操作。经过软件检查后，可由软件启动机密信息擦除操作。

47.3.10 TAMP 备份寄存器和其他设备机密信息擦除

备份寄存器 (TAMP_BKPxR) 不会通过系统复位来复位，也不会在设备从待机模式唤醒时复位。

当相应的屏蔽位置 1（TAMP_CR2 寄存器中的 TAMPxMSK=1）时，备份寄存器和其他设备机密信息不会重置。

Tamper 检测——确认模式

当 TAMP_CR2 寄存器中的 TAMPxNOER = 0 时，为 TAMPx（外部 Tamper x）选择确认模式。当 TAMP_CR3 寄存器中的 ITAMPxNOER = 0 时，为 ITAMPx（内部 Tamper x）选择确认模式。*表 408：TAMP 互连*中的 tamp_confirmed 和 tamp_confirmed_ercfgx 信号描述了确认模式下 Tamper 检测的作用。

选择该模式后，可以在检测到 Tamper 时自动擦除设备机密信息。

Tamper 检测——潜在模式

当 TAMP_CR2 寄存器中的 TAMPxNOER = 1 时，为 TAMPx（外部 Tamper x）选择潜在模式。当 TAMP_CR3 寄存器中的 ITAMPxNOER = 1 时，为 ITAMPx（内部 Tamper x）选择潜在模式。*表 408：TAMP 互连*中的 tamp_potential 和 tamp_potential_ercfgx 信号描述了潜在模式下 Tamper 检测的作用。

选择该模式后，可以避免在检测到 Tamper 时不可逆地擦除某些设备机密信息。在该模式下，当检测到相应的 Tamper 事件时，一些设备机密信息不会被擦除。此外，在潜在模式下，只要 Tamper 检测标志置 1，就会阻止对这些设备机密信息的读写访问，直到通过将 TAMP_SCR 寄存器中的相应清零标志置 1 的方式将该标志清零。因此，软件可以执行一些检查来区分虚假 Tamper 与真实 Tamper，从而能够在潜在 Tamper 被确认为真实 Tamper 的情况下启动机密信息擦除操作。可通过软件将 TAMP_CR2 寄存器中的 BKERASE 位置 1 来擦除设备机密信息。

潜在 Tamper 切换为确认 Tamper 时的超时

对于一些内部 Tamper，如果在其他 Tamper 标志置 1 时发生独立看门狗复位，则将生成 Tamper 事件（请参见*表 408：TAMP 互连*）。IWDG Tamper 必须在 ITAMPxNOER = 0 的情况下配置。这样，当上一个 Tamper 事件处于潜在模式时，可以通过硬件在超时后强制擦除设备机密信息。如果在未对潜在 Tamper 事件做出任何软件决策之前发生看门狗复位，则相当于将“潜在 Tamper”切换为“确认 Tamper”。

设备资源保护配置

可以对一些设备资源进行配置，将其列入受 Tamper 检测保护的设备机密信息列表。

当 TAMP_ERCFGR 中的 ERCFGz = 0 时，与 ERCFGz 相关联的设备资源不受 TAMP 外设保护：

- 不受 Tamper 检测影响（无论是在确认模式下还是在潜在模式下）
- 不受 BKERASE 软件命令影响
- 不受 BKBLOCK 软件命令影响

当 TAMP_ERCFGR 中的 ERCFGz = 1 时，与 ERCFGz 相关联的设备资源受 TAMP 外设保护：

- 受确认 Tamper 检测和 BKERASE 软件命令的影响，如*表 408：TAMP 互连*中的 tamp_confirmed_ercfgz 信号部分所述
- 受潜在 Tamper 检测和 BKBLOCK 软件命令的影响，如*表 408：TAMP 互连*中的 tamp_potential_ercfgz 信号部分所述

表 409. 设备资源 x Tamper 保护

-	潜在 Tamper 或 BKBLOCK	确认 Tamper 或 BKERASE
RPCFGx = 0	对设备资源 x 无影响	对设备资源 x 无影响
RPCFGx = 1	对设备机密信息 x 进行保护（如 tamp_potential_ercfgx 所述）(1)	对设备机密信息 x 进行保护（如 tamp_confirmed_ercfgx(1) 所述）

1. 请参见*表 408：TAMP 互连*。

通过软件阻止访问设备机密信息

默认情况下，应用程序可以访问设备机密信息，但检测到 Tamper 事件标志的情况除外：只要 Tamper 标志置 1，就不能访问设备机密信息。

可通过软件将 TAMP_CR2 寄存器的 BKBLOCK 位置 1 来阻止访问设备机密信息。只有在 BKBLOCK = 0 且没有任何 Tamper 标志置 1 时，才能访问设备机密信息。

47.3.11 Tamper 检测配置和初始化

可通过将 TAMP_CR 寄存器中相应的 TAMPxE 位置 1 来使能各输入。

每个 TAMP_INx Tamper 检测输入都分别与 TAMP_SR 寄存器中的标志 TAMPxF 相关联。

通过将 TAMP_IER 寄存器中的 TAMPxIE 位置 1，可在发生 Tamper 检测事件时（当 TAMPxF 置 1 时）生成中断。当相应的 TAMPxMSK 置 1 时，不允许将 TAMPxIE 置 1。

出现 Tamper 事件时生成触发输出

低功耗定时器可将 Tamper 事件检测用作触发输入。

将 TAMP_CR 寄存器中的 TAMPxMSK 位清零时，必须通过软件将 TAMPxF 标志清零以便在同一引脚上检测新 Tamper。

将 TAMPxMSK 位置 1 时，TAMPxF 标志会被屏蔽，并在 TAMP_SR 寄存器中保持清零。这种配置允许在 Stop 模式下自动触发低功耗定时器，无需通过唤醒系统来执行 TAMPxF 清零。在这种情况下，备份寄存器不清零。

该功能仅在 Tamper 配置为使用滤波模式的电平检测（TAMPFLT ≠ 00 且未选择主动模式）时可用。请参见*对 Tamper 输入的带滤波电平检测（被动模式）*一节。

Tamper 事件的时间戳

当 RTC_CR 中的 TAMPTS 置 1 时，任何内部或外部 Tamper 事件都会导致时间戳事件的发生。因 Tamper 事件而引发时间戳事件时，如同发生正常时间戳事件一样，RTC_SR 中的 TSF 位或 TSOVF 位会置 1。

注： *TSF 在 TAMPxF 标志置 1 后的 3 个 ck_apre 周期内置 1。如果 RTCCLK 停止，则 TSF 不会置 1（RTCCLK 重启时置 1）。*

注： *如果 TAMPxF 在 TSF 按预期置 1 前被清零，则 TSF 不会置 1。因此，如果 TAMPTS = 1，软件应在清零 Tamper 标志前等待时间戳标志，或者应读取 TAMP 中断程序中的 RTC 计数器值。*

对 Tamper 输入的边沿检测（被动模式）

如果 TAMPFLT 位设置为 00，当相应的 TAMPxTRG 位上出现上升沿或下降沿时，TAMP_INx 引脚将生成 Tamper 检测事件。选择边沿检测时，会禁止 TAMP_INx 输入上的内部上拉电阻。

小心： 当 TAMPFLT = 00 且 TAMPxTRG = 0（上升沿检测）时，如果在使能 Tamper 检测前 Tamper 输入已处于高电平，则可通过硬件来检测 Tamper 事件。

检测到 Tamper 事件并清零后，应当在重新编程备份寄存器 (TAMP_BKPxR) 之前禁止 TAMP_INx，然后再重新使能（TAMPxE 置 1）。这可防止应用在 TAMP_INx 输入值仍指示 Tamper 检测时，对备份寄存器执行写操作。这相当于对 TAMP_INx 输入的电平检测。

注： *当 V_{DD} 电源关闭时，Tamper 检测仍有效。要避免意外复位备份寄存器，应将 TAMPx 映射到的引脚从外部连接到正确的电平。*

对 Tamper 输入的带滤波电平检测（被动模式）

通过将 TAMPFLT 设置为非零值可执行带滤波的电平检测。在 TAMPxTRG 位指定的电平连续出现 2、4 或 8 个（取决于 TAMPFLT 值）采样时生成 Tamper 检测事件。

除非通过将 TAMPPUDIS 置 1 禁止 Tamper 输入，否则在 Tamper 输入的状态被采样前，将通过 I/O 内部上拉电阻对 TAMP_INx 输入预充电。预充电的持续时间由 TAMPPRCH 位确定，允许增大 TAMP_INx 输入上的电容。

可使用 TAMPFREQ 确定用于电平检测的采样频率，以便使 Tamper 检测延迟与上拉电阻功耗之间达到最佳平衡。

注： *有关上拉电阻的电气特性，请参见微控制器数据手册。*

主动 Tamper 检测

当 TAMP_ATCR 中的 TAMPxAM 位置 1 时，Tamper 事件配置为主动模式，该模式基于 TAMP_OUTy 引脚与 TAMP_INx 引脚之间的比较。默认情况下 (ATOSHARE = 0)，在 TAMP_INx 与 TAMP_OUTx (y = x) 之间进行比较。当 ATOSHARE 位置 1 时，同一输出可用于多个 Tamper 输入。借助 ATOSHARE 和 ATOSELx 位，只要选择了与主动 Tamper 输入 TAMP_INx 进行比较的输出 (TAMPxEN = TAMPxAM = 1)，就会在 I/O 上使能 TAMP_OUTy 功能。请参见 TAMP_ATCRx（x = 1 和 2）寄存器中 ATOSHARE 和 ATOSEL 位的说明。

每两个 CK_ATPER 周期 (CK_ATPER = 2^{ATPER} × CK_ATPRE)，TAMP_OUTy 输出引脚就会提供一个由伪随机数发生器 (PRNG) 生成的值。输出该值后，TAMP_OUTy 引脚会在一个 CK_ATPER 周期后输出其相反值。

表 410. 主动 Tamper 输出变化周期

ATCKSEL[3:0]	CK_ATPRE 频率	ATPER[2:0]	Tamper 输出变化 (CK_ATPER) 频率	Tamper 输出变化周期(1) (ms)
0x0	f_{RTCCLK}	0x0	f_{RTCCLK}	0.030
		0x1	$f_{RTCCLK}/2$	0.061
		0x2	$f_{RTCCLK}/4$	0.122
		0x3	$f_{RTCCLK}/8$	0.244
		0x4	$f_{RTCCLK}/16$	0.488
		0x5	$f_{RTCCLK}/32$	0.977
		0x6	$f_{RTCCLK}/64$	1.953
		0x7	$f_{RTCCLK}/128$	3.906
…	…	…	…	…

表 410. 主动 Tamper 输出变化周期（续）

ATCKSEL[3:0]	CK_ATPRE 频率	ATPER[2:0]	Tamper 输出变化 (CK_ATPER) 频率	Tamper 输出变化周期(1) (ms)
0x7	f_{RTCCLK}/128	0x0	f_{RTCCLK}/128	3.906
		0x1	f_{RTCCLK}/256	7.8125
		0x2	f_{RTCCLK}/512	15.625
		0x3	f_{RTCCLK}/1024	31.250
		0x4	f_{RTCCLK}/2048	62.5
		0x5	f_{RTCCLK}/4096	125
		0x6	f_{RTCCLK}/8192	250
		0x7	f_{RTCCLK}/16384	500
0xB	f_{RTCCLK}/2048(2)	0x0	f_{RTCCLK}/2048	62.5
		0x1	f_{RTCCLK}/4096	125
		0x2	f_{RTCCLK}/8192	250
		0x3	f_{RTCCLK}/16384	500
		0x4	f_{RTCCLK}/32768	1000
		0x5	f_{RTCCLK}/65536	2000
		0x6	f_{RTCCLK}/131072	4000
		0x7	f_{RTCCLK}/262144	8000

1. 假设 f_{RTCCLK} = 32768 Hz。
2. 该设置要求 (PREDIV_A+1) = 128 且 (PREDIV_S+1) 是 16 的倍数。

根据选择的 Tamper 输出数量的不同，所选 Tamper 输出使用 PRNG 的频率也有所不同。选择的输出数量取决于 TAMPxAM、TAMPxE、ATOSEL 和 ATOSHARE。

- 当仅选择 1 个输出时：每 16 个 CK_ATPER 周期使用一次 PRNG。
- 当选择 2 个输出时：每 8 个 CK_ATPER 周期使用一次 PRNG。
- 当选择 3 个或 4 个输出时：每 4 个 CK_ATPER 周期使用一次 PRNG。
- 当选择 5 个或更多输出时：每 2 个 CK_ATPER 周期使用一次 PRNG。

PRNG 需要至少 9 个 CK_ATPRE 周期才能输出新值。因此，下表提供了确保正常工作的最小 ATPER 值：

表 411. 最小 ATPER 值

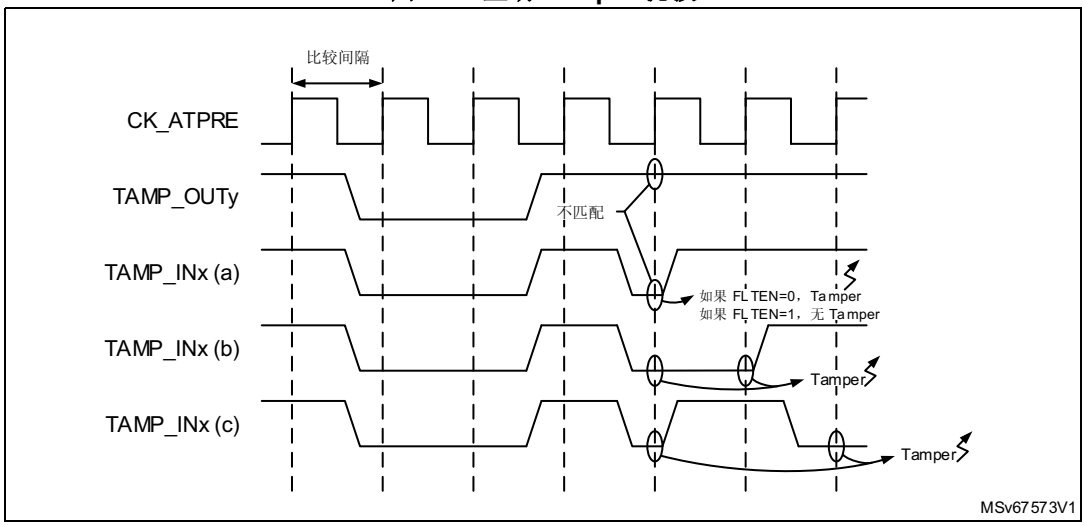
选择的输出数量	ATPER 最小值
1	0
2	1
3 或 4	2
5 或以上	3

TAMP_INx 引脚从外部连接到 TAMP_OUTy 引脚。每个 CK_ATPRE 周期都会将 TAMP_OUTy 输出值与 TAMP_INx 接收值进行比较。如果比较不匹配，TAMP_SR 寄存器中的 TAMPxF 位将会置 1。

例如，可以使用 TAMP_OUT1 与 TAMP_IN1 和 TAMP_IN2 进行比较，方法是在主动模式下配置并使能 TAMP1 和 TAMP2（ATOSHARE = 1、ATOSEL1 = 000 且 ATOSEL2 = 000）。

当 FLTEN = 1 时，主动 Tamper 模式可与输入滤波结合使用。在这种情况下，只有在连续 4 个比较样本中出现 2 次比较错误时，才会检测到 Tamper。

图 533. 主动 Tamper 滤波



如*图 533* 所示，如果 FLTEN = 0，在对 TAMP_INx 输入进行采样时，TAMP_OUTy 输出与相关 TAMP_INx 输入之间的任何不匹配都会生成 Tamper 事件。所有三个示例 (a)、(b) 和 (c) 都显示了这种情况。

如果 FLTEN = 1，则示例 (a) 不会生成 Tamper 事件，因为在连续四次比较中仅检测到一次不匹配。在示例 (b) 中，由于连续两次检测到不匹配，因此生成了 Tamper 事件。在示例 (c) 中，即使在连续采样过程中并未发生不匹配，但由于在连续四次比较中出现了两次不匹配，因此也生成了 Tamper 事件。

将 FLTEN 置 1 可以避免因 TAMP_INx 输入上的干扰、突发或瞬态状态而触发不必要的 Tamper 检测，方法是忽略小于一个 CK_ATPRE 周期的单个脉冲（对 TAMP_ATCR1 寄存器的 ATCKSEL 位字段进行编程）。*表 412* 列出了 ATCKSEL 各种可能设置的最小滤波脉宽（假设 f_{RTCCLK} = 32.768 kHz）。

表 412. 主动 Tamper 滤波脉冲的持续时间

ATCKSEL[3:0]	CK_ATPRE 频率	最小滤波脉宽 (ms)
0x0	f_{RTCCLK}	0.030
0x1	$f_{RTCCLK}/2$	0.061
0x2	$f_{RTCCLK}/4$	0.122
0x3	$f_{RTCCLK}/8$	0.244
0x4	$f_{RTCCLK}/16$	0.488
0x5	$f_{RTCCLK}/32$	0.977
0x6	$f_{RTCCLK}/64$	1.953

表 412. 主动 Tamper 滤波脉冲的持续时间（续）

ATCKSEL[3:0]	CK_ATPRE 频率	最小滤波脉宽 (ms)
0x7	$f_{RTCCLK}/128$	3.906
0xB	$f_{RTCCLK}/2048$	62.500(1)

1. 该设置要求 (PREDIV_A+1) = 128 且 (PREDIV_S+1) 是 16 的倍数。

注： *但是，如果小于一个 CK_ATPRE 周期的多个脉冲在连续四次比较中出现了两次不匹配，可能会引发 Tamper。*

小心： 当 ATCKSEL[3] = 1 时，进入 RTC 初始化模式会停止 CK_ATPRE 和 CK_ATPER 时钟。因此，TAMP_OUTy 引脚停止切换，直到退出 INIT 模式。

请参见 *日历初始化和配置*一节。

如果在 ATCKSEL[3] = 1 时使用 RTC 二进制模式，则另请参见 *RTC 闹钟 A 亚秒寄存器 (RTC_ALRMASSR)*、*RTC 闹钟 B 亚秒寄存器 (RTC_ALRMBSSR)*、*RTC 闹钟 A 二进制模式寄存器 (RTC_ALRABINR)* 和 *RTC 闹钟 B 二进制模式寄存器 (RTC_ALRBBINR)*。

小心： 小心：当 ATCKSEL[3] = 1 时，如果日历上溢，主动 Tamper 检测将不再起作用。当日历上溢时，必须使能内部 Tamper 5 以确保实施 Tamper 保护。

伪随机数发生器必须使用 seed 进行初始化。可通过在 TAMP_ATSEEDR 寄存器中连续写入四个 32 位随机值来完成该初始化操作。对 seed 进行编程会自动将其发送到 PRNG。只要新 seed 由 PRNG 进行传输和转化，TAMP_ATOR 中的 SEEDF 位就会置 1，并且不允许关闭 TAMP APB 时钟。写入第四个 seed 后，转化的持续时间将长达 184 个 APB 时钟周期。因此，写入新 seed 后，用户必须等待至 SEEDF 清零后才能进入低功耗模式。

主动 Tamper 输出只有在写入第一个 seed 并完成转化后才会激活。随后，可以在主动 Tamper 活动期间写入并转化新 seed。

主动 Tamper 初始化

下面给出了系统复位后通过软件初始化主动 Tamper 的流程：
读取 TAMP_ATOR 寄存器中的 INITS。

- 如果 INITS = 0x0（初始化未完成）：
  a) 写入 TAMP_ATCR 以配置主动 Tamper 时钟、滤波器和共用输出（如果有）以及主动模式。
  b) 写入 TAMP_CR1 以使能 Tamper（必须在同一个写访问中使能所需的所有 Tamper）。
  c) 对 TAMP_ATSEEDR 执行四次写操作，以写入 SEED。
  d) 等待至 TAMP_ATOR 中的 SEEDF = 0。备份寄存器随即受到主动 Tamper 保护。
- 如果 INITS = 0x1（初始化已完成）：
  无需初始化。为了增加随机性，应定期提供新的 SEED。提供新 SEED 时，需等待至 SEEDF = 0 之后才能进入低功耗模式，该模式会关闭 TAMP APB 时钟。
- 如果通过软件禁止了 Tamper，之后又重新使能，则在使能 Tamper 后，必须写入 SEED：
  a) 写入 TAMP_CR1 以使能 Tamper（必须在同一个写访问中使能所需的所有 Tamper）。
  b) 对 TAMP_ATSEEDR 执行四次写操作，以写入 SEED。
  c) 等待至 TAMP_ATOR 中的 SEEDF = 0。备份寄存器随即受到主动 Tamper 保护。

47.4 TAMP 低功耗模式

表 413. 低功耗模式对 TAMP 的影响

模式	说明
睡眠	无影响。 TAMP 中断可使设备退出睡眠模式。
停止	对所有功能都没有影响，但使用滤波和主动 Tamper 模式的电平检测除外，这些模式仅在时钟源为 LSE 或 LSI 时保持有效。 Tamper 事件可使设备退出 Stop 模式。
待机	对所有功能都没有影响，但使用滤波和主动 Tamper 模式的电平检测除外，这些模式仅在时钟源为 LSE 或 LSI 时保持有效。 Tamper 事件可使设备退出待机模式。

表 414. 各个模式下的 TAMP 引脚功能

引脚名称	所有低功耗模式下功能正常	V_{BAT} 模式下功能正常
TAMP_IN[8:1]	是	是
TAMP_OUT[5:1]	是	是

表 415. 各模式下的内部 Tamper 功能

内部 Tamper	说明	睡眠模式下功能正常	Stop 模式下功能正常	LPLV-Stop 和 LPLV-Stop2 模式下功能正常	待机模式下功能正常	V_{BAT} 模式下功能正常
tamp_itamp1	备份域电压阈值监视	是	是	是	是	是
tamp_itamp2	温度监视		是	是	是	是
tamp_itamp3	LSE 监视 (CSS)		是	是	是	是
tamp_itamp4	HSE 监控（CSS + 超频检测）		否	否	否	否
tamp_itamp5	RTC 日历上溢 (rtc_calovf)		是	是	是	是
tamp_itamp6	JTAG/SWD 访问		是	是	否	否
tamp_itamp7	ADC2 (adc2_awd1) 模拟看门狗监视 1		否	否	否	否
tamp_itamp8	单调计数器 1 上溢		否	否	否	否
tamp_itamp9	加密外设故障（SAES、CRYP、PKA 或 TRNG）		否	否	否	否
tamp_itamp10	单调计数器 2 上溢		否	否	否	否
tamp_itamp11	Tamper 标志置 1 时产生的 WDG 复位（潜在 Tamper 超时）		是	是	是	是
tamp_itamp12	ADC2 (adc2_awd2) 模拟看门狗监视 2		否	否	否	否
tamp_itamp13	ADC2 (adc2_awd3) 模拟看门狗监视 3		否	否	否	否

47.5 TAMP 中断

中断通道在屏蔽中断状态寄存器中置 1，或在安全屏蔽中断状态寄存器中置 1，具体取决于其安全模式配置 (TAMPSEC)。

表 416. 非安全中断请求

中断缩写	中断事件	事件标志(1)	使能控制位	中断清除方法	退出低功耗模式
TAMP	Tamper x(2)	TAMPxF	TAMPxIE 和 TAMPSEC = 0	向 CTAMPxF 中写入 1	是(3)
	内部 Tamper y(2)	ITAMPyF	ITAMPyIE 和 TAMPSEC = 0	向 CITAMPyF 中写入 1	是(3)

1. 事件标志位于 TAMP_SR 寄存器中。
2. Tamper 事件和内部 Tamper 事件的数量取决于产品。
3. 有关低功耗模式下可用功能的更多详细信息，请参见*表 413：低功耗模式对 TAMP 的影响*。

表 417. 安全中断请求

中断缩写	中断事件	事件标志(1)	使能控制位	中断清除方法	退出低功耗模式
TAMP_S	Tamper x(2)	TAMPxF	TAMPxIE (TAMPSEC = 1)	向 CTAMPxF 中写入 1	是(3)
	内部 Tamper y(2)	ITAMPyF	ITAMPyIE (TAMPSEC = 1)	向 CITAMPyF 中写入 1	是(3)

1. 事件标志位于 TAMP_SR 寄存器中。
2. Tamper 事件和内部 Tamper 事件的数量取决于产品。
3. 有关低功耗模式下可用功能的更多详细信息，请参见*表 413：低功耗模式对 TAMP 的影响*。

47.6 TAMP 寄存器

有关寄存器说明中使用的缩写，请参见参考手册中的*第 1.2 节*。外设寄存器可按字（32 位）进行访问。

47.6.1 TAMP 控制寄存器 1 (TAMP_CR1)

TAMP control register 1

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x00

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	ITAMP13E	ITAMP12E	ITAMP11E	ITAMP10E	ITAMP9E	ITAMP8E	ITAMP7E	ITAMP6E	ITAMP5E	ITAMP4E	ITAMP3E	ITAMP2E	ITAMP1E
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8E	TAMP7E	TAMP6E	TAMP5E	TAMP4E	TAMP3E	TAMP2E	TAMP1E
								rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30 保留，必须保持复位值。

位 29 保留，必须保持复位值。

位 28 **ITAMP13E**：内部 Tamper 13 使能 (Internal tamper 13 enable)

0：禁止内部 Tamper 13。

1：使能内部 Tamper 13。

位 27 **ITAMP12E**：内部 Tamper 12 使能 (Internal tamper 12 enable)
0：禁止内部 Tamper 12。
1：使能内部 Tamper 12。

位 26 **ITAMP11E**：内部 Tamper 11 使能 (Internal tamper 11 enable)
0：禁止内部 Tamper 11。
1：使能内部 Tamper 11。

位 25 **ITAMP10E**：内部 Tamper 10 使能 (Internal tamper 10 enable)
0：禁止内部 Tamper 10。
1：使能内部 Tamper 10。

位 24 **ITAMP9E**：内部 Tamper 9 使能 (Internal tamper 9 enable)
0：禁止内部 Tamper 9。
1：使能内部 Tamper 9。

位 23 **ITAMP8E**：内部 Tamper 8 使能 (Internal tamper 8 enable)
0：禁止内部 Tamper 8。
1：使能内部 Tamper 8。

位 22 **ITAMP7E**：内部 Tamper 7 使能 (Internal tamper 7 enable)
0：禁止内部 Tamper 7。
1：使能内部 Tamper 7。

位 21 **ITAMP6E**：内部 Tamper 6 使能 (Internal tamper 6 enable)
0：禁止内部 Tamper 6。
1：使能内部 Tamper 6。

位 20 **ITAMP5E**：内部 Tamper 5 使能 (Internal tamper 5 enable)
0：禁止内部 Tamper 5。
1：使能内部 Tamper 5。

位 19 **ITAMP4E**：内部 Tamper 4 使能 (Internal tamper 4 enable)
0：禁止内部 Tamper 4。
1：使能内部 Tamper 4。

位 18 **ITAMP3E**：内部 Tamper 3 使能 (Internal tamper 3 enable)
0：禁止内部 Tamper 3。
1：使能内部 Tamper 3。

位 17 **ITAMP2E**：内部 Tamper 2 使能 (Internal tamper 2 enable)
0：禁止内部 Tamper 2。
1：使能内部 Tamper 2。

位 16 **ITAMP1E**：内部 Tamper 1 使能 (Internal tamper 1 enable)
0：禁止内部 Tamper 1。
1：使能内部 Tamper 1。

位 15:8 保留，必须保持复位值。

位 7 **TAMP8E**：对 TAMP_IN8 的 Tamper 检测使能 (Tamper detection on TAMP_IN8 enable)(1)
0：禁止对 TAMP_IN8 的 Tamper 检测。
1：使能对 TAMP_IN8 的 Tamper 检测。

位 6 **TAMP7E**：对 TAMP_IN7 的 Tamper 检测使能 (Tamper detection on TAMP_IN7 enable)(1)
0：禁止对 TAMP_IN7 的 Tamper 检测。
1：使能对 TAMP_IN7 的 Tamper 检测。

位 5 **TAMP6E**：对 TAMP_IN6 的 Tamper 检测使能 (Tamper detection on TAMP_IN6 enable)(1)
0：禁止对 TAMP_IN6 的 Tamper 检测。
1：使能对 TAMP_IN6 的 Tamper 检测。

位 4 **TAMP5E**：对 TAMP_IN5 的 Tamper 检测使能 (Tamper detection on TAMP_IN5 enable)(1)
0：禁止对 TAMP_IN5 的 Tamper 检测。
1：使能对 TAMP_IN5 的 Tamper 检测。

位 3 **TAMP4E**：对 TAMP_IN4 的 Tamper 检测使能 (Tamper detection on TAMP_IN4 enable)(1)
0：禁止对 TAMP_IN4 的 Tamper 检测。
1：使能对 TAMP_IN4 的 Tamper 检测。

位 2 **TAMP3E**：对 TAMP_IN3 的 Tamper 检测使能 (Tamper detection on TAMP_IN3 enable)(1)
0：禁止对 TAMP_IN3 的 Tamper 检测。
1：使能对 TAMP_IN3 的 Tamper 检测。

位 1 **TAMP2E**：对 TAMP_IN2 的 Tamper 检测使能 (Tamper detection on TAMP_IN2 enable)(1)
0：禁止对 TAMP_IN2 的 Tamper 检测。
1：使能对 TAMP_IN2 的 Tamper 检测。

位 0 **TAMP1E**：对 TAMP_IN1 的 Tamper 检测使能 (Tamper detection on TAMP_IN1 enable)(1)
0：禁止对 TAMP_IN1 的 Tamper 检测。
1：使能对 TAMP_IN1 的 Tamper 检测。

1. 必须先配置 Tamper 检测模式（通过 TAMP_FLTCR、TAMP_ATCR1、TAMP_ATCR2 寄存器和 TAMP_CR2 中的 TAMPxTRG 位进行选择），然后才能使能 Tamper 检测。

47.6.2 TAMP 控制寄存器 2 (TAMP_CR2)

TAMP control register 2

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x04

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TAMP8 TRG	TAMP7 TRG	TAMP6 TRG	TAMP5 TRG	TAMP4 TRG	TAMP3 TRG	TAMP2 TRG	TAMP1 TRG	BK ERASE	BK BLOCK	Res.	Res.	Res.	TAMP3 MSK	TAMP2 MSK	TAMP1 MSK
rw	rw	rw	rw	rw	rw	rw	rw	w	rw				rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8 NOER	TAMP7 NOER	TAMP6 NOER	TAMP5 NOER	TAMP4 NOER	TAMP3 NOER	TAMP2 NOER	TAMP1 NOER
								rw	rw	rw	rw	rw	rw	rw	rw

位 31 **TAMP8TRG**：Tamper 8 输入的有效电平（禁止主动模式）(Active level for tamper 8 input (active mode disabled))

0：如果 TAMPFLT ≠ 00，Tamper 8 输入保持低电平会触发 Tamper 检测事件。
如果 TAMPFLT = 00，Tamper 8 输入上升沿会触发 Tamper 检测事件。

1：如果 TAMPFLT ≠ 00，Tamper 8 输入保持高电平会触发 Tamper 检测事件。
如果 TAMPFLT = 00，Tamper 8 输入下降沿会触发 Tamper 检测事件。

位 30 **TAMP7TRG**：Tamper 7 输入的有效电平（禁止主动模式）(Active level for tamper 7 input (active mode disabled))

0：如果 TAMPFLT ≠ 00，Tamper 7 输入保持低电平会触发 Tamper 检测事件。
如果 TAMPFLT = 00，Tamper 7 输入上升沿会触发 Tamper 检测事件。

1：如果 TAMPFLT ≠ 00，Tamper 7 输入保持高电平会触发 Tamper 检测事件。
如果 TAMPFLT = 00，Tamper 7 输入下降沿会触发 Tamper 检测事件。

位 29 **TAMP6TRG**：Tamper 6 输入的有效电平（禁止主动模式）(Active level for tamper 6 input (active mode disabled))

0：如果 TAMPFLT ≠ 00，Tamper 6 输入保持低电平会触发 Tamper 检测事件。
如果 TAMPFLT = 00，Tamper 6 输入上升沿会触发 Tamper 检测事件。

1：如果 TAMPFLT ≠ 00，Tamper 6 输入保持高电平会触发 Tamper 检测事件。
如果 TAMPFLT = 00，Tamper 6 输入下降沿会触发 Tamper 检测事件。

位 28 **TAMP5TRG**：Tamper 5 输入的有效电平（禁止主动模式）(Active level for tamper 5 input (active mode disabled))

0：如果 TAMPFLT ≠ 00，Tamper 5 输入保持低电平会触发 Tamper 检测事件。
如果 TAMPFLT = 00，Tamper 5 输入上升沿会触发 Tamper 检测事件。

1：如果 TAMPFLT ≠ 00，Tamper 5 输入保持高电平会触发 Tamper 检测事件。
如果 TAMPFLT = 00，Tamper 5 输入下降沿会触发 Tamper 检测事件。

位 27 **TAMP4TRG**：Tamper 4 输入的有效电平（禁止主动模式）(Active level for tamper 4 input (active mode disabled))

0：如果 TAMPFLT ≠ 00，Tamper 4 输入保持低电平会触发 Tamper 检测事件。
如果 TAMPFLT = 00，Tamper 4 输入上升沿会触发 Tamper 检测事件。

1：如果 TAMPFLT ≠ 00，Tamper 4 输入保持高电平会触发 Tamper 检测事件。
如果 TAMPFLT = 00，Tamper 4 输入下降沿会触发 Tamper 检测事件。

位 26 **TAMP3TRG**：Tamper 3 输入的有效电平 (Active level for tamper 3 input)

0：如果 TAMPFLT ≠ 00，Tamper 3 输入保持低电平会触发 Tamper 检测事件。
如果 TAMPFLT = 00，Tamper 3 输入上升沿会触发 Tamper 检测事件。

1：如果 TAMPFLT ≠ 00，Tamper 3 输入保持高电平会触发 Tamper 检测事件。
如果 TAMPFLT = 00，Tamper 3 输入下降沿会触发 Tamper 检测事件。

位 25 **TAMP2TRG**：Tamper 2 输入的有效电平 (Active level for tamper 2 input)

0：如果 TAMPFLT ≠ 00，Tamper 2 输入保持低电平会触发 Tamper 检测事件。
如果 TAMPFLT = 00，Tamper 2 输入上升沿会触发 Tamper 检测事件。

1：如果 TAMPFLT ≠ 00，Tamper 2 输入保持高电平会触发 Tamper 检测事件。
如果 TAMPFLT = 00，Tamper 2 输入下降沿会触发 Tamper 检测事件。

位 24 **TAMP1TRG**：Tamper 1 输入的有效电平 (Active level for tamper 1 input)

0：如果 TAMPFLT ≠ 00，Tamper 1 输入保持低电平会触发 Tamper 检测事件。
如果 TAMPFLT = 00，Tamper 1 输入上升沿会触发 Tamper 检测事件。

1：如果 TAMPFLT ≠ 00，Tamper 1 输入保持高电平会触发 Tamper 检测事件。
如果 TAMPFLT = 00，Tamper 1 输入下降沿会触发 Tamper 检测事件。

位 23 **BKERASE**：备份寄存器和设备机密信息(1)擦除 (Backup registers and device secrets erase)

向该位写入“1”会重置备份寄存器和设备机密信息(1)。写入 0 无影响。该位始终读为 0。

位 22 **BKBLOCK**：阻止备份寄存器和设备机密信息(1)访问 (Backup registers and device secrets access blocked)

0：如果没有任何 Tamper 标志置 1，则可以对备份寄存器和设备机密信息(1)进行访问

1：不可以对备份寄存器和设备机密信息(1)进行访问

位 21:19 保留，必须保持复位值。

位 18 **TAMP3MSK**：Tamper 3 屏蔽 (Tamper 3 mask)

0：Tamper 3 事件生成触发事件，必须通过软件将 TAMP3F 清零来允许下一次 Tamper 事件检测。

1：Tamper 3 事件生成触发事件。屏蔽 TAMP3F 并由硬件在内部将其清零。潜在模式备份寄存器和设备机密信息(1)。

TAMP3MSK 置 1 时，不得使能 Tamper 3 中断。

位 17 **TAMP2MSK**：Tamper 2 屏蔽 (Tamper 2 mask)

0：Tamper 2 事件生成触发事件，必须通过软件将 TAMP2F 清零来允许下一次 Tamper 事件检测。

1：Tamper 2 事件生成触发事件。屏蔽 TAMP2F 并由硬件在内部将其清零。潜在模式备份寄存器和设备机密信息(1)。

TAMP2MSK 置 1 时，不得使能 Tamper 2 中断。

位 16 **TAMP1MSK**：Tamper 1 屏蔽 (Tamper 1 mask)

0：Tamper 1 事件生成触发事件，必须通过软件将 TAMP1F 清零来允许下一次 Tamper 事件检测。

1：Tamper 1 事件生成触发事件。屏蔽 TAMP1F 并由硬件在内部将其清零。潜在模式备份寄存器和设备机密信息(1)。

TAMP1MSK 置 1 时，不得使能 Tamper 1 中断。

位 15:8 保留，必须保持复位值。

位 7 **TAMP8NOER**：Tamper 8 潜在模式 (Tamper 8 potential mode)

0：在确认模式(1)下执行 Tamper 8 事件检测。

1：在潜在模式(2)下执行 Tamper 8 事件检测。

位 6 **TAMP7NOER**：Tamper 7 潜在模式 (Tamper 7 potential mode)

0：在确认模式(1)下执行 Tamper 7 事件检测。

1：在潜在模式(2)下执行 Tamper 7 事件检测。

位 5 **TAMP6NOER**：Tamper 6 潜在模式 (Tamper 6 potential mode)

0：在确认模式(1)下执行 Tamper 6 事件检测。

1：在潜在模式(2)下执行 Tamper 6 事件检测。

位 4 **TAMP5NOER**：Tamper 5 潜在模式 (Tamper 5 potential mode)

0：在确认模式(1)下执行 Tamper 5 事件检测。

1：在潜在模式(2)下执行 Tamper 5 事件检测。

位 3 **TAMP4NOER**：Tamper 4 潜在模式 (Tamper 4 potential mode)

0：在确认模式(1)下执行 Tamper 4 事件检测。

1：在潜在模式(2)下执行 Tamper 4 事件检测。

位 2 **TAMP3NOER**：Tamper 3 潜在模式 (Tamper 3 potential mode)

0：在确认模式(1)下执行 Tamper 3 事件检测。

1：在潜在模式(2)下执行 Tamper 3 事件检测。

位 1 **TAMP2NOER**：Tamper 2 潜在模式 (Tamper 2 potential mode)

0：在确认模式(1)下执行 Tamper 2 事件检测。

1：在潜在模式(2)下执行 Tamper 2 事件检测。

位 0 **TAMP1NOER**：Tamper 1 潜在模式 (Tamper 1 potential mode)
0：在确认模式(1)下执行 Tamper 1 事件检测。
1：在潜在模式(2)下执行 Tamper 1 事件检测。

1. *表 408：TAMP 互连*中的 tamp_confirmed 和 tamp_confirmed_ercfgx 信号描述了确认模式下 Tamper 检测的作用。
2. *表 408：TAMP 互连*中的 tamp_potential 和 tamp_potential_ercfgx 信号描述了潜在模式下 Tamper 检测的作用。

47.6.3 TAMP 控制寄存器 3 (TAMP_CR3)

TAMP control register 3

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x08

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	ITAMP13NOER	ITAMP12NOER	ITAMP11NOER	ITAMP10NOER	ITAMP9NOER	ITAMP8NOER	ITAMP7NOER	ITAMP6NOER	ITAMP5NOER	ITAMP4NOER	ITAMP3NOER	ITAMP2NOER	ITAMP1NOER
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 保留，必须保持复位值。

位 14 保留，必须保持复位值。

位 13 保留，必须保持复位值。

位 12 **ITAMP13NOER**：内部 Tamper 13 潜在模式 (Internal tamper 13 potential mode)
0：在确认模式(1)下执行内部 Tamper 13 事件检测。
1：在潜在模式(2)下执行内部 Tamper 13 事件检测。

位 11 **ITAMP12NOER**：内部 Tamper 12 潜在模式 (Internal tamper 12 potential mode)
0：在确认模式(1)下执行内部 Tamper 12 事件检测。
1：在潜在模式(2)下执行内部 Tamper 12 事件检测。

位 10 **ITAMP11NOER**：内部 Tamper 11 潜在模式 (Internal tamper 11 potential mode)
0：在确认模式(1)下执行内部 Tamper 11 事件检测。
1：在潜在模式(2)下执行内部 Tamper 11 事件检测。

位 9 **ITAMP10NOER**：内部 Tamper 10 未擦除 (Internal Tamper 10 潜在模式)
0：在确认模式(1)下执行内部 Tamper 10 事件检测。
1：在潜在模式(2)下执行内部 Tamper 10 事件检测。

位 8 **ITAMP9NOER**：内部 Tamper 9 潜在模式 (Internal tamper 9 potential mode)
0：在确认模式(1)下执行内部 Tamper 9 事件检测。
1：在潜在模式(2)下执行内部 Tamper 9 事件检测。

位 7 **ITAMP8NOER**：内部 Tamper 8 潜在模式 (Internal tamper 8 potential mode)
0：在确认模式(1)下执行内部 Tamper 8 事件检测。
1：在潜在模式(2)下执行内部 Tamper 8 事件检测。

位 6 **ITAMP7NOER**：内部 Tamper 7 潜在模式 (Internal tamper 7 potential mode)
0：在确认模式(1)下执行内部 Tamper 7 事件检测。
1：在潜在模式(2)下执行内部 Tamper 7 事件检测。

位 5 **ITAMP6NOER**：内部 Tamper 6 潜在模式 (Internal tamper 6 potential mode)
0：在确认模式(1)下执行内部 Tamper 6 事件检测。
1：在潜在模式(2)下执行内部 Tamper 6 事件检测。

位 4 **ITAMP5NOER**：内部 Tamper 5 潜在模式 (Internal tamper 5 potential mode)
0：在确认模式(1)下执行内部 Tamper 5 事件检测。
1：在潜在模式(2)下执行内部 Tamper 5 事件检测。

位 3 **ITAMP4NOER**：内部 Tamper 4 潜在模式 (Internal tamper 4 potential mode)
0：在确认模式(1)下执行内部 Tamper 4 事件检测。
1：在潜在模式(2)下执行内部 Tamper 4 事件检测。

位 2 **ITAMP3NOER**：内部 Tamper 3 潜在模式 (Internal tamper 3 potential mode)
0：在确认模式(1)下执行内部 Tamper 3 事件检测。
1：在潜在模式(2)下执行内部 Tamper 3 事件检测。

位 1 **ITAMP2NOER**：内部 Tamper 2 潜在模式 (Internal tamper 2 potential mode)
0：在确认模式(1)下执行内部 Tamper 2 事件检测。
1：在潜在模式(2)下执行内部 Tamper 2 事件检测。

位 0 **ITAMP1NOER**：内部 Tamper 1 潜在模式 (Internal tamper 1 potential mode)
0：在确认模式(1)下执行内部 Tamper 1 事件检测。
1：在潜在模式(2)下执行内部 Tamper 1 事件检测。

1. *表 408：TAMP 互连*中的 tamp_confirmed 和 tamp_confirmed_ercfgx 信号描述了确认模式下内部 Tamper 检测的作用。
2. *表 408：TAMP 互连*中的 tamp_potential 和 tamp_potential_ercfgx 信号描述了潜在模式下内部 Tamper 检测的作用。

47.6.4 TAMP 滤波器控制寄存器 (TAMP_FLTCR)

TAMP filter control register

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x0C

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP PUDIS	TAMPPRCH [1:0]		TAMPFLT [1:0]		TAMPFREQ [2:0]		
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7 **TAMPPUDIS**：TAMP_INx 上拉禁止 (TAMP_INx pull-up disable)

该位决定在每次采样之前是否对每个 TAMPx 引脚都进行预充电。

0：采样之前对 TAMP_INx 引脚进行预充电（使能内部上拉）

1：禁止对 TAMP_INx 引脚进行预充电

位 6:5 **TAMPPRCH[1:0]**：TAMP_INx 预充电持续时间 (TAMP_INx precharge duration)

这些位决定了在每次采样之前激活上拉的持续时间。TAMPPRCH 对每个 TAMP_INx 输入都有效。

0x0：1 个 RTCCLK 周期

0x1：2 个 RTCCLK 周期

0x2：4 个 RTCCLK 周期

0x3：8 个 RTCCLK 周期

位 4:3 **TAMPFLT[1:0]**：TAMP_INx 滤波器计数 (TAMP_INx filter count)

这些位决定了为激活 Tamper 事件所需的指定电平 (TAMP*TRG) 上的连续采样次数。TAMPFLT 对每个 TAMP_INx 输入都有效。

0x0：在 TAMP_INx 输入转变为有效电平的边沿激活 Tamper 事件（TAMP_INx 输入上无内部上拉）。

0x1：在有效电平上连续执行 2 次采样后激活 Tamper 事件。

0x2：在有效电平上连续执行 4 次采样后激活 Tamper 事件。

0x3：在有效电平上连续执行 8 次采样后激活 Tamper 事件。

位 2:0 **TAMPFREQ[2:0]**：Tamper 采样频率 (Tamper sampling frequency)

决定对每个 TAMP_INx 输入进行采样时的频率。

0x0：RTCCLK / 32768（RTCCLK = 32768 Hz 时为 1 Hz）

0x1：RTCCLK / 16384（RTCCLK = 32768 Hz 时为 2 Hz）

0x2：RTCCLK / 8192（RTCCLK = 32768 Hz 时为 4 Hz）

0x3：RTCCLK / 4096（RTCCLK = 32768 Hz 时为 8 Hz）

0x4：RTCCLK / 2048（RTCCLK = 32768 Hz 时为 16 Hz）

0x5：RTCCLK / 1024（RTCCLK = 32768 Hz 时为 32 Hz）

0x6：RTCCLK / 512（RTCCLK = 32768 Hz 时为 64 Hz）

0x7：RTCCLK / 256（RTCCLK = 32768 Hz 时为 128 Hz）

注： *该寄存器仅与被动模式下的 Tamper 输入有关。*

47.6.5 TAMP 主动 Tamper 控制寄存器 1 (TAMP_ATCR1)

TAMP active tamper control register 1

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x10

备份域复位值：0x0007 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLTEN	ATO SHARE	Res.	Res.	Res.	ATPER[2:0]			Res.	Res.	Res.	Res.	ATCKSEL[3:0]			
rw	rw				rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATOSEL4[1:0]		ATOSEL3[1:0]		ATOSEL2[1:0]		ATOSEL1[1:0]		TAMP8 AM	TAMP7 AM	TAMP6 AM	TAMP5 AM	TAMP4 AM	TAMP3 AM	TAMP2 AM	TAMP1 AM
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **FLTEN**：主动 Tamper 滤波器使能 (Active tamper filter enable)

0：禁止主动 Tamper 滤波

1：使能主动 Tamper 滤波：当在连续 4 次采样中出现 2 次比较不匹配时，会检测到 Tamper 事件。

位 30 **ATOSHARE**：主动 Tamper 输出共用 (Active tamper output sharing)

0：每个主动 Tamper 输入 TAMP_INi 都与其专用输出 TAMP_OUTi 进行比较

1：每个主动 Tamper 输入 TAMP_INi 都与 ATOSELi 位定义的 TAMPOUTSELi 进行比较。

位 29:27 保留，必须保持复位值。

位 26:24 **ATPER[2:0]**：主动 Tamper 输出变化周期 (Active tamper output change period)

Tamper 输出每 CK_ATPER = (2^{ATPER} x CK_ATPRE) 个周期变化一次。请参见*表 411：最小 ATPER 值*。

位 23:20 保留，必须保持复位值。

位 19:16 **ATCKSEL[3:0]**：主动 Tamper RTC 异步预分频器时钟选择 (Active tamper RTC asynchronous prescaler clock selection)

这些位用于选择 RTC 异步预分频器级输出。所选时钟为 CK_ATPRE。

0000： 选择 RTCCLK

0001： 选择 RTCCLK/2

0010： 选择 RTCCLK/4

0011： 选择 RTCCLK/8

0100： 选择 RTCCLK/16

0101： 选择 RTCCLK/32

0110： 选择 RTCCLK/64

0111： 选择 RTCCLK/128

1011： 当 (PREDIV_A+1) = 128 且 (PREDIV_S+1) 是 16 的倍数时，选择 RTCCLK/2048。

其他值： 保留

注意：这些位只能在所有主动 Tamper 均已禁止时写入。禁止所有主动 Tamper 后，写保护最多可保持 1.5 个 CK_ATPRE 周期。

位 15:14 **ATOSEL4[1:0]**：主动 Tamper 共用输出 4 选择 (Active tamper shared output 4 selection)

00：TAMPOUTSEL4 = TAMP_OUT1

01：TAMPOUTSEL4 = TAMP_OUT2

10：TAMPOUTSEL4 = TAMP_OUT3

11：TAMPOUTSEL4 = TAMP_OUT4

如果封装引脚排列中未提供 TAMP_OUTx 输出，则输出选择值保留，不得使用。

位 13:12 **ATOSEL3[1:0]**：主动 Tamper 共用输出 3 选择 (Active tamper shared output 3 selection)
00：TAMPOUTSEL3 = TAMP_OUT1
01：TAMPOUTSEL3 = TAMP_OUT2
10：TAMPOUTSEL3 = TAMP_OUT3
11：TAMPOUTSEL3 = TAMP_OUT4
如果封装引脚排列中未提供 TAMP_OUTx 输出，则输出选择值保留，不得使用。

位 11:10 **ATOSEL2[1:0]**：主动 Tamper 共用输出 2 选择 (Active tamper shared output 2 selection)
00：TAMPOUTSEL2 = TAMP_OUT1
01：TAMPOUTSEL2 = TAMP_OUT2
10：TAMPOUTSEL2 = TAMP_OUT3
11：TAMPOUTSEL2 = TAMP_OUT4
如果封装引脚排列中未提供 TAMP_OUTx 输出，则输出选择值保留，不得使用。

位 9:8 **ATOSEL1[1:0]**：主动 Tamper 共用输出 1 选择 (Active tamper shared output 1 selection)
00：TAMPOUTSEL1 = TAMP_OUT1
01：TAMPOUTSEL1 = TAMP_OUT2
10：TAMPOUTSEL1 = TAMP_OUT3
11：TAMPOUTSEL1 = TAMP_OUT4
如果封装引脚排列中未提供 TAMP_OUTx 输出，则输出选择值保留，不得使用。

位 7 **TAMP8AM**：Tamper 8 主动模式 (Tamper 8 active mode)
0：Tamper 8 检测使用被动模式。
1：Tamper 8 检测使用主动模式。

位 6 **TAMP7AM**：Tamper 7 主动模式 (Tamper 7 active mode)
0：Tamper 7 检测使用被动模式。
1：Tamper 7 检测使用主动模式。

位 5 **TAMP6AM**：Tamper 6 主动模式 (Tamper 6 active mode)
0：Tamper 6 检测使用被动模式。
1：Tamper 6 检测使用主动模式。

位 4 **TAMP5AM**：Tamper 5 主动模式 (Tamper 5 active mode)
0：Tamper 5 检测使用被动模式。
1：Tamper 5 检测使用主动模式。

位 3 **TAMP4AM**：Tamper 4 主动模式 (Tamper 4 active mode)
0：Tamper 4 检测使用被动模式。
1：Tamper 4 检测使用主动模式。

位 2 **TAMP3AM**：Tamper 3 主动模式 (Tamper 3 active mode)
0：Tamper 3 检测使用被动模式。
1：Tamper 3 检测使用主动模式。

位 1 **TAMP2AM**：Tamper 2 主动模式 (Tamper 2 active mode)
0：Tamper 2 检测使用被动模式。
1：Tamper 2 检测使用主动模式。

位 0 **TAMP1AM**：Tamper 1 主动模式 (Tamper 1 active mode)
0：Tamper 1 检测使用被动模式。
1：Tamper 1 检测使用主动模式。

注： *当 TAMPxAM 位置 1 时，不允许更改该寄存器中的主动 Tamper 配置，除非 TAMP_CR1 寄存器中所有相应的 TAMPxE 位都清零。*

配置为主动模式的所有 Tamper 必须同时使能（通过在同一个 TAMP_CR1 写访问中将所有相关的 TAMPxE 都置 1）。

配置为主动模式的所有 Tamper 必须同时禁止（通过在同一个 TAMP_CR1 写访问中将所有相关的 TAMPxE 都清零）。

禁止主动 Tamper 后，必须等待至少 1 个 CK_ATPRE 周期才能将其重新使能。

47.6.6 TAMP 主动 Tamper 种子寄存器 (TAMP_ATSEEDR)

TAMP active tamper seed register

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x14

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEED[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEED[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **SEED[31:0]**：伪随机数发生器 seed 值 (Pseudo-random generator seed value)
必须向该寄存器连续写入四个 32 位值，才能为 PRNG 提供 128 位 seed。写入该寄存器会自动将 seed 值发送到 PRNG。

47.6.7 TAMP 主动 Tamper 输出寄存器 (TAMP_ATOR)

TAMP active tamper output register

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x18

备份域复位值：0x0000 0000

系统复位：不受影响（SEEDF 除外，它在复位时被清零）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INITS	SEEDF	Res.	Res.	Res.	Res.	Res.	Res.	PRNG[7:0]							
r	r							r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15 **INITS**：主动 Tamper 初始化状态 (Active tamper initialization status)

当 PRNG 接收到第一个 128 位 seed 时，该标志由硬件置 1，这意味着主动 Tamper 监控功能可用。当禁止主动 Tamper 时，该标志被清零。

位 14 **SEEDF**：Seed 运行标志 (Seed running flag)

将新 seed 写入 TAMP_ATSEEDR 时，该标志由硬件置 1。当 PRNG 接收到该新 seed 时，该标志由硬件清零，也可以通过系统复位清零。只要 SEEDF 置 1，就不得关闭 TAMP APB 时钟。

位 13:8 保留，必须保持复位值。

位 7:0 **PRNG[7:0]**：伪随机数发生器值 (Pseudo-random generator value)

该位字段提供 PRNG 输出的值。由于同步延迟可能导致不一致，因此必须至少读取 PRNG 两次。如果两次读取的值相等，则表示数值正确。

仅当 APB 处于安全模式时，才能读取该位字段。

47.6.8 TAMP 主动 Tamper 控制寄存器 2 (TAMP_ATCR2)

TAMP active tamper control register 2

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x1C

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ATOSEL8[2:0]			ATOSEL7[2:0]			ATOSEL6[2:0]			ATOSEL5[2:0]			ATOSEL4[2:0]			ATOSEL3[2]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATOSEL3[1:0]		ATOSEL2[2:0]			ATOSEL1[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw								

位 31:29 **ATOSEL8[2:0]**：主动 Tamper 共用输出 8 选择 (Active tamper shared output 8 selection)

000：TAMPOUTSEL8 = TAMP_OUT1
001：TAMPOUTSEL8 = TAMP_OUT2
010：TAMPOUTSEL8 = TAMP_OUT3
011：TAMPOUTSEL8 = TAMP_OUT4
100：TAMPOUTSEL8 = TAMP_OUT5
101：TAMPOUTSEL8 = TAMP_OUT6
110：TAMPOUTSEL8 = TAMP_OUT7
111：TAMPOUTSEL8 = TAMP_OUT8
如果封装引脚排列中未提供 TAMP_OUTx 输出，则输出选择值保留，不得使用。

位 28:26 **ATOSEL7[2:0]**：主动 Tamper 共用输出 7 选择 (Active tamper shared output 7 selection)

000：TAMPOUTSEL7 = TAMP_OUT1
001：TAMPOUTSEL7 = TAMP_OUT2
010：TAMPOUTSEL7 = TAMP_OUT3
011：TAMPOUTSEL7 = TAMP_OUT4
100：TAMPOUTSEL7 = TAMP_OUT5
101：TAMPOUTSEL7 = TAMP_OUT6
110：TAMPOUTSEL7 = TAMP_OUT7
111：TAMPOUTSEL7 = TAMP_OUT8
如果封装引脚排列中未提供 TAMP_OUTx 输出，则输出选择值保留，不得使用。

位 25:23 **ATOSEL6[2:0]**：主动 Tamper 共用输出 6 选择 (Active tamper shared output 6 selection)

000：TAMPOUTSEL6 = TAMP_OUT1
001：TAMPOUTSEL6 = TAMP_OUT2
010：TAMPOUTSEL6 = TAMP_OUT3
011：TAMPOUTSEL6 = TAMP_OUT4
100：TAMPOUTSEL6 = TAMP_OUT5
101：TAMPOUTSEL6 = TAMP_OUT6
110：TAMPOUTSEL6 = TAMP_OUT7
111：TAMPOUTSEL6 = TAMP_OUT8
如果封装引脚排列中未提供 TAMP_OUTx 输出，则输出选择值保留，不得使用。

位 22:20 **ATOSEL5[2:0]**：主动 Tamper 共用输出 5 选择 (Active tamper shared output 5 selection)

000：TAMPOUTSEL5 = TAMP_OUT1
001：TAMPOUTSEL5 = TAMP_OUT2
010：TAMPOUTSEL5 = TAMP_OUT3
011：TAMPOUTSEL5 = TAMP_OUT4
100：TAMPOUTSEL5 = TAMP_OUT5
101：TAMPOUTSEL5 = TAMP_OUT6
110：TAMPOUTSEL5 = TAMP_OUT7
111：TAMPOUTSEL5 = TAMP_OUT8
如果封装引脚排列中未提供 TAMP_OUTx 输出，则输出选择值保留，不得使用。

位 19:17 **ATOSEL4[2:0]**：主动 Tamper 共用输出 4 选择 (Active tamper shared output 4 selection)

000：TAMPOUTSEL4 = TAMP_OUT1
001：TAMPOUTSEL4 = TAMP_OUT2
010：TAMPOUTSEL4 = TAMP_OUT3
011：TAMPOUTSEL4 = TAMP_OUT4
100：TAMPOUTSEL4 = TAMP_OUT5
101：TAMPOUTSEL4 = TAMP_OUT6
110：TAMPOUTSEL4 = TAMP_OUT7
111：TAMPOUTSEL4 = TAMP_OUT8

如果封装引脚排列中未提供 TAMP_OUTx 输出，则输出选择值保留，不得使用。

位 18:17 是 TAMP_ATCR1 中 ATOSEL2[1:0] 的镜像，因此也可以通过 TAMP_ATCR1 对其进行读/写操作。

位 16:14 **ATOSEL3[2:0]**：主动 Tamper 共用输出 3 选择 (Active tamper shared output 3 selection)

000：TAMPOUTSEL3 = TAMP_OUT1
001：TAMPOUTSEL3 = TAMP_OUT2
010：TAMPOUTSEL3 = TAMP_OUT3
011：TAMPOUTSEL3 = TAMP_OUT4
100：TAMPOUTSEL3 = TAMP_OUT5
101：TAMPOUTSEL3 = TAMP_OUT6
110：TAMPOUTSEL3 = TAMP_OUT7
111：TAMPOUTSEL3 = TAMP_OUT8

如果封装引脚排列中未提供 TAMP_OUTx 输出，则输出选择值保留，不得使用。

位 15:14 是 TAMP_ATCR1 中 ATOSEL3[1:0] 的镜像，因此也可以通过 TAMP_ATCR1 对其进行读/写操作。

位 13:11 **ATOSEL2[2:0]**：主动 Tamper 共用输出 2 选择 (Active tamper shared output 2 selection)

000：TAMPOUTSEL2 = TAMP_OUT1
001：TAMPOUTSEL2 = TAMP_OUT2
010：TAMPOUTSEL2 = TAMP_OUT3
011：TAMPOUTSEL2 = TAMP_OUT4
100：TAMPOUTSEL2 = TAMP_OUT5
101：TAMPOUTSEL2 = TAMP_OUT6
110：TAMPOUTSEL2 = TAMP_OUT7
111：TAMPOUTSEL2 = TAMP_OUT8

如果封装引脚排列中未提供 TAMP_OUTx 输出，则输出选择值保留，不得使用。

位 12:11 是 TAMP_ATCR1 中 ATOSEL2[1:0] 的镜像，因此也可以通过 TAMP_ATCR1 对其进行读/写操作。

位 10:8 **ATOSEL1[2:0]**：主动 Tamper 共用输出 1 选择 (Active tamper shared output 1 selection)

000：TAMPOUTSEL1 = TAMP_OUT1
001：TAMPOUTSEL1 = TAMP_OUT2
010：TAMPOUTSEL1 = TAMP_OUT3
011：TAMPOUTSEL1 = TAMP_OUT4
100：TAMPOUTSEL1 = TAMP_OUT5
101：TAMPOUTSEL1 = TAMP_OUT6
110：TAMPOUTSEL1 = TAMP_OUT7
111：TAMPOUTSEL1 = TAMP_OUT8

如果封装引脚排列中未提供 TAMP_OUTx 输出，则输出选择值保留，不得使用。

位 9:8 是 TAMP_ATCR1 中 ATOSEL1[1:0] 的镜像，因此也可以通过 TAMP_ATCR1 对其进行读/写操作。

位 7:0 保留，必须保持复位值。

注： *当 TAMPxAM 位置 1 时，不允许更改该寄存器中的主动 Tamper 配置，除非 TAMP_CR1 寄存器中所有相应的 TAMPxE 位都清零。*

配置为主动模式的所有 Tamper 必须同时使能（通过在同一个 TAMP_CR1 写访问中将所有相关的 TAMPxE 都置 1）。

配置为主动模式的所有 Tamper 必须同时禁止（通过在同一个 TAMP_CR1 写访问中将所有相关的 TAMPxE 都清零）。

禁止主动 Tamper 后，必须等待至少 1 个 CK_ATPRE 周期才能将其重新使能。

47.6.9 TAMP 安全配置寄存器（TAMP_SECCFGR）

TAMP secure configuration register

当 TZEN = 1 时，该寄存器只能通过安全 APB 访问进行写操作。当 TZEN = 0 时，BKPRWSEC[7:0]、BKPWSEC[7:0] 和 BHKLOCK 可以通过非安全 APB 访问进行写操作，TAMPSEC、CNT1SEC 和 CNT2SEC 不可写入。

既可以为该寄存器设置全局写保护，也可以为该寄存器的每个位单独设置写保护，以防其遭到非特权访问，具体取决于 TAMP_PRIVCFGR 配置（请参见*第 47.3.7 节：TAMP 特权保护模式*）。

偏移地址：0x20

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TAMPSEC	BHKLOCK	Res.	Res.	Res.	Res.	Res.	Res.	BKPWSEC[7:0]							
rw	rs							rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT1SEC	CNT2SEC	Res.	Res.	Res.	Res.	Res.	Res.	BKPRWSEC[7:0]							
rw	rw							rw	rw	rw	rw	rw	rw	rw	rw

位 31 **TAMPSEC**：Tamper 保护（不包括单调计数器和备份寄存器）(Tamper protection (excluding monotonic counters and backup registers))

0：Tamper 配置和中断可以通过 APB 安全或非安全访问进行写操作。

1：Tamper 配置和中断只能通过 APB 安全访问进行写操作。

注意：有关读保护的详细信息，请参见第 47.3.5 节：TAMP 安全保护模式。

位 30 **BHKLOCK**：启动硬件密钥锁定 (Boot hardware key lock)

该位可读，并且只能通过软件写为 1。Tamper 检测事件发生后，该位由硬件清零（连同备份寄存器一起清零）。

0：备份寄存器（TAMP_BKP0R 到 TAMP_BKP7R）可根据其所属的保护区进行访问。

1：不可对备份寄存器（TAMP_BKP0R 到 TAMP_BKP7R）进行读写访问（读为 0，忽略写操作）。

位 29:24 保留，必须保持复位值。

位 23:16 **BKPWSEC[7:0]**：备份寄存器写保护偏移量 (Backup registers write protection offset)
BKPWSEC 值必须介于 0 到 32 之间。

保护区 2 针对备份寄存器 TAMP_BKPyR (y = BKPRWSEC) 到 TAMP_BKPzR（z = BKPWSEC-1，其中 BKPWSEC > BKPRWSEC）定义：
– 如果 TZEN=1，则只能通过安全访问写入这些备份寄存器。
– 可以通过安全访问或非安全访问读取这些备份寄存器。
如果 BKPWSEC = 0 或 BKPWSEC ≤ BKPRWSEC：无保护区 2。

保护区 3 针对备份寄存器 TAMP_BKPtR（BKPWSEC ≥ BKPRWSEC 时，t = BKPWSEC，否则 t = BKPRWSEC）定义。
– 可以通过安全访问或非安全访问读写这些备份寄存器。
如果 BKPWSEC = 32：无保护区 3。

请参见*图 532：备份寄存器保护区*。

注意：如果TZEN=0：可以通过非安全访问读写保护区 2。
注意：如果 BKPWPRIV 置 1，则只能在特权模式下写入 BKPRWSEC[7:0]。

位 15 **CNT1SEC**：单调计数器 1 安全保护 (Monotonic counter 1 secure protection)
0：单调计数器 1 (TAMP_COUNT1R) 可以通过 APB 安全或非安全访问进行读写操作。
1：单调计数器 1 (TAMP_COUNT1R) 只能通过 APB 安全访问进行读写操作。

位 14 **CNT2SEC**：单调计数器 2 安全保护 (Monotonic counter 2 secure protection)
0：单调计数器 2 (TAMP_COUNT2R) 可以通过 APB 安全或非安全访问进行读写操作。
1：单调计数器 2 (TAMP_COUNT2R) 只能通过 APB 安全访问进行读写操作。

位 13:8 保留，必须保持复位值。

位 7:0 **BKPRWSEC[7:0]**：备份寄存器读/写保护偏移量 (Backup registers read/write protection offset)
BKPRWSEC 值必须介于 0 到 32 之间。

保护区 1 针对备份寄存器 TAMP_BKP0R 到 TAMP_BKPxR（x = BKPRWSEC-1，其中 BKPRWSEC ≥ 1）定义。
– 如果 TZEN=1，则只能通过安全访问读写这些备份寄存器。
如果 BKPRWSEC = 0：无保护区 1。

请参见*图 532：备份寄存器保护区*。

注意：如果TZEN=0：可以通过非安全访问读写保护区 1。
注意：如果 BKPRWPRIV 置 1，则只能在特权模式下写入 BKPRWSEC[7:0]。

47.6.10 TAMP 特权配置寄存器 (TAMP_PRIVCFGR)

TAMP privilege configuration register

该寄存器只能通过 APB 特权访问进行写操作。

当 TZEN = 1 时，既可以为该寄存器设置写保护，也可以为该寄存器的每个位单独设置写保护，以防其遭到非安全访问，具体取决于 TAMP_SECCFGR 的配置（请参见*第 47.3.5 节：TAMP 安全保护模式*）。

偏移地址：0x24

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TAMP PRIV	BKP WPRIV	BKPR WPRIV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT1 PRIV	CNT2 PRIV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw														

位 31 **TAMPPRIV**：Tamper 特权保护（不包括备份寄存器）(Tamper privilege protection (excluding backup registers))

0：Tamper 配置和中断可以通过特权访问或非特权访问进行写操作。

1：Tamper 配置和中断只能通过特权访问进行写操作。

注意：有关读保护的详细信息，请参见第 47.3.7 节：TAMP 特权保护模式。

位 30 **BKPWPRIV**：备份寄存器区域 2 特权保护 (Backup registers zone 2 privilege protection)

0：备份寄存器区域 2 可以通过特权访问或非特权访问进行写操作。

1：备份寄存器区域 2 只能通过特权访问进行写操作。

位 29 **BKPRWPRIV**：备份寄存器区域 1 特权保护 (Backup registers zone 1 privilege protection)

0：备份寄存器区域 1 可以通过特权访问或非特权访问进行读写操作。

1：备份寄存器区域 1 只能通过特权访问进行读写操作。

位 28:16 保留，必须保持复位值。

位 15 **CNT1PRIV**：单调计数器 1 特权保护 (Monotonic counter 1 privilege protection)

0：单调计数器 1 (TAMP_COUNT1R) 可以通过 APB 特权或非特权访问进行读写操作。

1：单调计数器 1 (TAMP_COUNT1R) 只能通过 APB 特权访问进行读写操作。

位 14 **CNT2PRIV**：单调计数器 2 特权保护 (Monotonic counter 2 privilege protection)

0：单调计数器 2 (TAMP_COUNT2R) 可以通过 APB 特权或非特权访问进行读写操作。

1：单调计数器 2 (TAMP_COUNT2R) 只能通过 APB 特权访问进行读写操作。

位 13:0 保留，必须保持复位值。

47.6.11 TAMP 中断使能寄存器 (TAMP_IER)

TAMP interrupt enable register

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x2C

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	ITAMP13IE	ITAMP12IE	ITAMP11IE	ITAMP10IE	ITAMP9IE	ITAMP8IE	ITAMP7IE	ITAMP6IE	ITAMP5IE	ITAMP4IE	ITAMP3IE	ITAMP2IE	ITAMP1IE
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8IE	TAMP7IE	TAMP6IE	TAMP5IE	TAMP4IE	TAMP3IE	TAMP2IE	TAMP1IE
								rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30 保留，必须保持复位值。

位 29 保留，必须保持复位值。

位 28 **ITAMP13IE**：内部 Tamper 13 中断使能 (Internal tamper 13 interrupt enable)

0：禁止内部 Tamper 13 中断。

1：使能内部 Tamper 13 中断。

位 27 **ITAMP12IE**：内部 Tamper 12 中断使能 (Internal tamper 12 interrupt enable)

0：禁止内部 Tamper 12 中断。

1：使能内部 Tamper 12 中断。

位 26 **ITAMP11IE**：内部 Tamper 11 中断使能 (Internal tamper 11 interrupt enable)

0：禁止内部 Tamper 11 中断。

1：使能内部 Tamper 11 中断。

位 25 **ITAMP10IE**：内部 Tamper 10 中断使能 (Internal tamper 10 interrupt enable)

0：禁止内部 Tamper 10 中断。

1：使能内部 Tamper 10 中断。

位 24 **ITAMP9IE**：内部 Tamper 9 中断使能 (Internal tamper 9 interrupt enable)

0：禁止内部 Tamper 9 中断。

1：使能内部 Tamper 9 中断。

位 23 **ITAMP8IE**：内部 Tamper 8 中断使能 (Internal tamper 8 interrupt enable)

0：禁止内部 Tamper 8 中断。

1：使能内部 Tamper 8 中断。

位 22 **ITAMP7IE**：内部 Tamper 7 中断使能 (Internal tamper 7 interrupt enable)

0：禁止内部 Tamper 7 中断。

1：使能内部 Tamper 7 中断。

位 21 **ITAMP6IE**：内部 Tamper 6 中断使能 (Internal tamper 6 interrupt enable)

0：禁止内部 Tamper 6 中断。

1：使能内部 Tamper 6 中断。

位 20 **ITAMP5IE**：内部 Tamper 5 中断使能 (Internal tamper 5 interrupt enable)

0：禁止内部 Tamper 5 中断。

1：使能内部 Tamper 5 中断。

位 19 **ITAMP4IE**：内部 Tamper 4 中断使能 (Internal tamper 4 interrupt enable)

0：禁止内部 Tamper 4 中断。

1：使能内部 Tamper 4 中断。

位 18 **ITAMP3IE**：内部 Tamper 3 中断使能 (Internal tamper 3 interrupt enable)

0：禁止内部 Tamper 3 中断。

1：使能内部 Tamper 3 中断。

位 17 **ITAMP2IE**：内部 Tamper 2 中断使能 (Internal tamper 2 interrupt enable)
0：禁止内部 Tamper 2 中断。
1：使能内部 Tamper 2 中断。

位 16 **ITAMP1IE**：内部 Tamper 1 中断使能 (Internal tamper 1 interrupt enable)
0：禁止内部 Tamper 1 中断。
1：使能内部 Tamper 1 中断。

位 15:8 保留，必须保持复位值。

位 7 **TAMP8IE**：Tamper 8 中断使能 (Tamper 8 interrupt enable)
0：禁止 Tamper 8 中断。
1：使能 Tamper 8 中断。

位 6 **TAMP7IE**：Tamper 7 中断使能 (Tamper 7 interrupt enable)
0：禁止 Tamper 7 中断。
1：使能 Tamper 7 中断。

位 5 **TAMP6IE**：Tamper 6 中断使能 (Tamper 6 interrupt enable)
0：禁止 Tamper 6 中断。
1：使能 Tamper 6 中断。

位 4 **TAMP5IE**：Tamper 5 中断使能 (Tamper 5 interrupt enable)
0：禁止 Tamper 5 中断。
1：使能 Tamper 5 中断。

位 3 **TAMP4IE**：Tamper 4 中断使能 (Tamper 4 interrupt enable)
0：禁止 Tamper 4 中断。
1：使能 Tamper 4 中断。

位 2 **TAMP3IE**：Tamper中断 3 使能 (Tamper 3 interrupt enable)
0：禁止 Tamper 3 中断。
1：使能 Tamper 3 中断。

位 1 **TAMP2IE**：Tamper 2 中断使能 (Tamper 2 interrupt enable)
0：禁止 Tamper 2 中断。
1：使能 Tamper 2 中断。

位 0 **TAMP1IE**：Tamper 1 中断使能 (Tamper 1 interrupt enable)
0：禁止 Tamper 1 中断。
1：使能 Tamper 1 中断。

47.6.12 TAMP 状态寄存器 (TAMP_SR)

TAMP status register

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x30

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	ITAMP13F	ITAMP12F	ITAMP11F	ITAMP10F	ITAMP9F	ITAMP8F	ITAMP7F	ITAMP6F	ITAMP5F	ITAMP4F	ITAMP3F	ITAMP2F	ITAMP1F
			r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8F	TAMP7F	TAMP6F	TAMP5F	TAMP4F	TAMP3F	TAMP2F	TAMP1F
								r	r	r	r	r	r	r	r

位 31 保留，必须保持复位值。

位 30 保留，必须保持复位值。

位 29 保留，必须保持复位值。

位 28 **ITAMP13F**：内部 Tamper 13 标志 (Internal tamper 13 flag)
在内部 Tamper 13 上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 27 **ITAMP12F**：内部 Tamper 12 标志 (Internal tamper 12 flag)
在内部 Tamper 12 上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 26 **ITAMP11F**：内部 Tamper 11 标志 (Internal tamper 11 flag)
在内部 Tamper 11 上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 25 **ITAMP10F**：内部 Tamper 10 标志 (Internal tamper 10 flag)
在内部 Tamper 10 上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 24 **ITAMP9F**：内部 Tamper 9 标志 (Internal tamper 9 flag)
在内部 Tamper 9 上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 23 **ITAMP8F**：内部 Tamper 8 标志 (Internal tamper 8 flag)
在内部 Tamper 8 上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 22 **ITAMP7F**：内部 Tamper 7 标志 (Internal tamper 7 flag)
在内部 Tamper 7 上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 21 **ITAMP6F**：内部 Tamper 6 标志 (Internal tamper 6 flag)
在内部 Tamper 6 上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 20 **ITAMP5F**：内部 Tamper 5 标志 (Internal tamper 5 flag)
在内部 Tamper 5 上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 19 **ITAMP4F**：内部 Tamper 4 标志 (Internal tamper 4 flag)
在内部 Tamper 4 上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 18 **ITAMP3F**：内部 Tamper 3 标志 (Internal tamper 3 flag)
在内部 Tamper 3 上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 17 **ITAMP2F**：内部 Tamper 2 标志 (Internal tamper 2 flag)
在内部 Tamper 2 上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 16 **ITAMP1F**：内部 Tamper 1 标志 (Internal tamper 1 flag)
在内部 Tamper 1 上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 15:8 保留，必须保持复位值。

位 7 **TAMP8F**：TAMP8 检测标志 (TAMP8 detection flag)
在 TAMP8 输入上检测到 Tamper 检测事件时，由硬件将该标志置 1

位 6 **TAMP7F**：TAMP7 检测标志 (TAMP7 detection flag)
在 TAMP7 输入上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 5 **TAMP6F**：TAMP6 检测标志 (TAMP6 detection flag)
在 TAMP6 输入上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 4 **TAMP5F**：TAMP5 检测标志 (TAMP5 detection flag)
在 TAMP5 输入上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 3 **TAMP4F**：TAMP4 检测标志 (TAMP4 detection flag)
在 TAMP4 输入上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 2 **TAMP3F**：TAMP3 检测标志 (TAMP3 detection flag)
在 TAMP3 输入上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 1 **TAMP2F**：TAMP2 检测标志 (TAMP2 detection flag)
在 TAMP2 输入上检测到 Tamper 检测事件时，由硬件将该标志置 1。

位 0 **TAMP1F**：TAMP1 检测标志 (TAMP1 detection flag)
在 TAMP1 输入上检测到 Tamper 检测事件时，由硬件将该标志置 1。

47.6.13 TAMP 非安全屏蔽中断状态寄存器 (TAMP_MISR)

TAMP nonsecure masked interrupt status register

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x34

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	ITAMP13MF	ITAMP12MF	ITAMP11MF	ITAMP10MF	ITAMP9MF	ITAMP8MF	ITAMP7MF	ITAMP6MF	ITAMP5MF	ITAMP4MF	ITAMP3MF	ITAMP2MF	ITAMP1MF
			r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8MF	TAMP7MF	TAMP6MF	TAMP5MF	TAMP4MF	TAMP3MF	TAMP2MF	TAMP1MF
								r	r	r	r	r	r	r	r

位 31 保留，必须保持复位值。

位 30 保留，必须保持复位值。

位 29 保留，必须保持复位值。

位 28 **ITAMP13MF**：内部 Tamper 13 非安全中断屏蔽标志 (internal tamper 13 nonsecure interrupt masked flag)
发生内部 Tamper 13 非安全中断时，由硬件将该标志置 1。

位 27 **ITAMP12MF**：内部 Tamper 12 非安全中断屏蔽标志 (internal tamper 12 nonsecure interrupt masked flag)
发生内部 Tamper 12 非安全中断时，由硬件将该标志置 1。

位 26 **ITAMP11MF**：内部 Tamper 11 非安全中断屏蔽标志 (internal tamper 11 nonsecure interrupt masked flag)
发生内部 Tamper 11 非安全中断时，由硬件将该标志置 1。

位 25 **ITAMP10MF**：内部 Tamper 10 非安全中断屏蔽标志 (internal tamper 10 nonsecure interrupt masked flag)

发生内部 Tamper 10 非安全中断时，由硬件将该标志置 1。

位 24 **ITAMP9MF**：内部 Tamper 9 非安全中断屏蔽标志 (internal tamper 9 nonsecure interrupt masked flag)

发生内部 Tamper 9 非安全中断时，由硬件将该标志置 1。

位 23 **ITAMP8MF**：内部 Tamper 8 非安全中断屏蔽标志 (Internal tamper 8 nonsecure interrupt masked flag)

发生内部 Tamper 8 非安全中断时，由硬件将该标志置 1。

位 22 **ITAMP7MF**：内部 Tamper 7 非安全中断屏蔽标志 (Internal tamper 7 nonsecure interrupt masked flag)

发生内部 Tamper 7 非安全中断时，由硬件将该标志置 1。

位 21 **ITAMP6MF**：内部 Tamper 6 非安全中断屏蔽标志 (Internal tamper 6 nonsecure interrupt masked flag)

发生内部 Tamper 6 非安全中断时，由硬件将该标志置 1。

位 20 **ITAMP5MF**：内部 Tamper 5 非安全中断屏蔽标志 (Internal tamper 5 nonsecure interrupt masked flag)

发生内部 Tamper 5 非安全中断时，由硬件将该标志置 1。

位 19 **ITAMP4MF**：内部 Tamper 4 非安全中断屏蔽标志 (Internal tamper 4 nonsecure interrupt masked flag)

发生内部 Tamper 4 非安全中断时，由硬件将该标志置 1。

位 18 **ITAMP3MF**：内部 Tamper 3 非安全中断屏蔽标志 (Internal tamper 3 nonsecure interrupt masked flag)

发生内部 Tamper 3 非安全中断时，由硬件将该标志置 1。

位 17 **ITAMP2MF**：内部 Tamper 2 非安全中断屏蔽标志 (Internal tamper 2 nonsecure interrupt masked flag)

发生内部 Tamper 2 非安全中断时，由硬件将该标志置 1。

位 16 **ITAMP1MF**：内部 Tamper 1 非安全中断屏蔽标志 (Internal tamper 16 nonsecure interrupt masked flag)

发生内部 Tamper 1 非安全中断时，由硬件将该标志置 1。

位 15:8 保留，必须保持复位值。

位 7 **TAMP8MF**：TAMP8 非安全中断屏蔽标志 (TAMP8 nonsecure interrupt masked flag)

发生 Tamper 8 非安全中断时，由硬件将该标志置 1。

位 6 **TAMP7MF**：TAMP7 非安全中断屏蔽标志 (TAMP7 nonsecure interrupt masked flag)

发生 Tamper 7 非安全中断时，由硬件将该标志置 1。

位 5 **TAMP6MF**：TAMP6 非安全中断屏蔽标志 (TAMP6 nonsecure interrupt masked flag)

发生 Tamper 6 非安全中断时，由硬件将该标志置 1。

位 4 **TAMP5MF**：TAMP5 非安全中断屏蔽标志 (TAMP5 nonsecure interrupt masked flag)

发生 Tamper 5 非安全中断时，由硬件将该标志置 1。

位 3 **TAMP4MF**：TAMP4 非安全中断屏蔽标志 (TAMP4 nonsecure interrupt masked flag)

发生 Tamper 4 非安全中断时，由硬件将该标志置 1。

位 2 **TAMP3MF**：TAMP3 非安全中断屏蔽标志 (TAMP3 nonsecure interrupt masked flag)

发生 Tamper 3 非安全中断时，由硬件将该标志置 1。

位 1 **TAMP2MF**：TAMP2 非安全中断屏蔽标志 (TAMP2 nonsecure interrupt masked flag)
发生 Tamper 2 非安全中断时，由硬件将该标志置 1。

位 0 **TAMP1MF**：TAMP1 非安全中断屏蔽标志 (TAMP1 nonsecure interrupt masked flag)
发生 Tamper 1 非安全中断时，由硬件将该标志置 1。

47.6.14 TAMP 安全屏蔽中断状态寄存器 (TAMP_SMISR)

TAMP secure masked interrupt status register

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x38

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	ITAMP13MF	ITAMP12MF	ITAMP11MF	ITAMP10MF	ITAMP9MF	ITAMP8MF	ITAMP7MF	ITAMP6MF	ITAMP5MF	ITAMP4MF	ITAMP3MF	ITAMP2MF	ITAMP1MF
			r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8MF	TAMP7MF	TAMP6MF	TAMP5MF	TAMP4MF	TAMP3MF	TAMP2MF	TAMP1MF
								r	r	r	r	r	r	r	r

位 31 保留，必须保持复位值。

位 30 保留，必须保持复位值。

位 29 保留，必须保持复位值。

位 28 **ITAMP13MF**：内部 Tamper 13 安全中断屏蔽标志 (internal tamper 13 secure interrupt masked flag)
发生内部 Tamper 13 安全中断时，由硬件将该标志置 1。

位 27 **ITAMP12MF**：内部 Tamper 12 安全中断屏蔽标志 (internal tamper 12 secure interrupt masked flag)
发生内部 Tamper 12 安全中断时，由硬件将该标志置 1。

位 26 **ITAMP11MF**：内部 Tamper 11 安全中断屏蔽标志 (internal tamper 11 secure interrupt masked flag)
发生内部 Tamper 11 安全中断时，由硬件将该标志置 1。

位 25 **ITAMP10MF**：内部 Tamper 10 安全中断屏蔽标志 (internal tamper 10 secure interrupt masked flag)
发生内部 Tamper 10 安全中断时，由硬件将该标志置 1。

位 24 **ITAMP9MF**：内部 Tamper 9 安全中断屏蔽标志 (internal tamper 9 secure interrupt masked flag)
发生内部 Tamper 9 安全中断时，由硬件将该标志置 1。

位 23 **ITAMP8MF**：内部 Tamper 8 安全中断屏蔽标志 (Internal tamper 8 secure interrupt masked flag)
发生内部 Tamper 8 安全中断时，由硬件将该标志置 1。

位 22 **ITAMP7MF**：内部 Tamper 7 安全中断屏蔽标志 (Internal tamper 7 secure interrupt masked flag)
发生内部 Tamper 7 安全中断时，由硬件将该标志置 1。

位 21 **ITAMP6MF**：内部 Tamper 6 安全中断屏蔽标志 (Internal tamper 6 secure interrupt masked flag)
发生内部 Tamper 6 安全中断时，由硬件将该标志置 1。

位 20 **ITAMP5MF**：内部 Tamper 5 安全中断屏蔽标志 (Internal tamper 5 secure interrupt masked flag)
发生内部 Tamper 5 安全中断时，由硬件将该标志置 1。

位 19 **ITAMP4MF**：内部 Tamper 4 安全中断屏蔽标志 (Internal tamper 4 secure interrupt masked flag)
发生内部 Tamper 4 安全中断时，由硬件将该标志置 1。

位 18 **ITAMP3MF**：内部 Tamper 3 安全中断屏蔽标志 (Internal tamper 3 secure interrupt masked flag)
发生内部 Tamper 3 安全中断时，由硬件将该标志置 1。

位 17 **ITAMP2MF**：内部 Tamper 2 安全中断屏蔽标志 (Internal tamper 2 secure interrupt masked flag)
发生内部 Tamper 2 安全中断时，由硬件将该标志置 1。

位 16 **ITAMP1MF**：内部 Tamper 1 安全中断屏蔽标志 (Internal tamper 1 secure interrupt masked flag)
发生内部 Tamper 1 安全中断时，由硬件将该标志置 1。

位 15:8 保留，必须保持复位值。

位 7 **TAMP8MF**：TAMP8 安全中断屏蔽标志 (TAMP8 secure interrupt masked flag)
发生 Tamper 8 安全中断时，由硬件将该标志置 1。

位 6 **TAMP7MF**：TAMP7 安全中断屏蔽标志 (TAMP7 secure interrupt masked flag)
发生 Tamper 7 安全中断时，由硬件将该标志置 1。

位 5 **TAMP6MF**：TAMP6 安全中断屏蔽标志 (TAMP6 secure interrupt masked flag)
发生 Tamper 6 安全中断时，由硬件将该标志置 1。

位 4 **TAMP5MF**：TAMP5 安全中断屏蔽标志 (TAMP5 secure interrupt masked flag)
发生 Tamper 5 安全中断时，由硬件将该标志置 1。

位 3 **TAMP4MF**：TAMP4 安全中断屏蔽标志 (TAMP4 secure interrupt masked flag)
发生 Tamper 4 安全中断时，由硬件将该标志置 1。

位 2 **TAMP3MF**：TAMP3 安全中断屏蔽标志 (TAMP3 secure interrupt masked flag)
发生 Tamper 3 安全中断时，由硬件将该标志置 1。

位 1 **TAMP2MF**：TAMP2 安全中断屏蔽标志 (TAMP2 secure interrupt masked flag)
发生 Tamper 2 安全中断时，由硬件将该标志置 1。

位 0 **TAMP1MF**：TAMP1 安全中断屏蔽标志 (TAMP1 secure interrupt masked flag)
发生 Tamper 1 安全中断时，由硬件将该标志置 1。

47.6.15 TAMP 状态清零寄存器 (TAMP_SCR)

TAMP status clear register

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x3C

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	CITAMP13F	CITAMP12F	CITAMP11F	CITAMP10F	CITAMP9F	CITAMP8F	CITAMP7F	CITAMP6F	CITAMP5F	CITAMP4F	CITAMP3F	CITAMP2F	CITAMP1F
			w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTAMP8F	CTAMP7F	CTAMP6F	CTAMP5F	CTAMP4F	CTAMP3F	CTAMP2F	CTAMP1F
								w	w	w	w	w	w	w	w

位 31 保留，必须保持复位值。

位 30 保留，必须保持复位值。

位 29 保留，必须保持复位值。

位 28 **CITAMP13F**：清零 ITAMP13 检测标志 (Clear ITAMP13 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 ITAMP13F 标志清零。

位 27 **CITAMP12F**：清零 ITAMP12 检测标志 (Clear ITAMP12 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 ITAMP12F 标志清零。

位 26 **CITAMP11F**：清零 ITAMP11 检测标志 (Clear ITAMP11 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 ITAMP11F 标志清零。

位 25 **CITAMP10F**：清零 ITAMP10 检测标志 (Clear ITAMP10 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 ITAMP10F 标志清零。

位 24 **CITAMP9F**：清零 ITAMP9 检测标志 (Clear ITAMP9 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 ITAMP9F 标志清零。

位 23 **CITAMP8F**：清零 ITAMP8 检测标志 (Clear ITAMP8 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 ITAMP8F 标志清零。

位 22 **CITAMP7F**：清零 ITAMP7 检测标志 (Clear ITAMP7 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 ITAMP7F 标志清零。

位 21 **CITAMP6F**：清零 ITAMP6 检测标志 (Clear ITAMP6 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 ITAMP6F 标志清零。

位 20 **CITAMP5F**：清零 ITAMP5 检测标志 (Clear ITAMP5 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 ITAMP5F 标志清零。

位 19 **CITAMP4F**：清零 ITAMP4 检测标志 (Clear ITAMP4 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 ITAMP4F 标志清零。

位 18 **CITAMP3F**：清零 ITAMP3 检测标志 (Clear ITAMP3 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 ITAMP3F 标志清零。

位 17 **CITAMP2F**：清零 ITAMP2 检测标志 (Clear ITAMP2 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 ITAMP2F 标志清零。

位 16 **CITAMP1F**：清零 ITAMP1 检测标志 (Clear ITAMP1 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 ITAMP1F 标志清零。

位 15:8 保留，必须保持复位值。

位 7 **CTAMP8F**：清零 TAMP8 检测标志 (Clear TAMP8 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 TAMP8F 标志清零。

位 6 **CTAMP7F**：清零 TAMP7 检测标志 (Clear TAMP7 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 TAMP7F 标志清零。

位 5 **CTAMP6F**：清零 TAMP6 检测标志 (Clear TAMP6 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 TAMP6F 标志清零。

位 4 **CTAMP5F**：清零 TAMP5 检测标志 (Clear TAMP5 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 TAMP5F 标志清零。

位 3 **CTAMP4F**：清零 TAMP4 检测标志 (Clear TAMP4 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 TAMP4F 标志清零。

位 2 **CTAMP3F**：清零 TAMP3 检测标志 (Clear TAMP3 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 TAMP3F 标志清零。

位 1 **CTAMP2F**：清零 TAMP2 检测标志 (Clear TAMP2 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 TAMP2F 标志清零。

位 0 **CTAMP1F**：清零 TAMP1 检测标志 (Clear TAMP1 detection flag)
向该位写入 1 可将 TAMP_SR 寄存器中的 TAMP1F 标志清零。

47.6.16 TAMP 单调计数器 1 寄存器 (TAMP_COUNT1R)

TAMP monotonic counter 1 register

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x040

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COUNT[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **COUNT[31:0]**：

该寄存器为只读，当对该寄存器进行写访问时，该寄存器递增 1。该寄存器无法计满返回，当计数到最大值时将被冻结。

47.6.17 TAMP 单调计数器 2 寄存器 (TAMP_COUNT2R)

TAMP monotonic counter 2 register

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x044

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COUNT[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **COUNT[31:0]**：单调计数器

该寄存器为只读，当对该寄存器进行写访问时，该寄存器递增 1。该寄存器无法计满返回，当计数到最大值时将被冻结。

47.6.18 TAMP 选项寄存器 (TAMP_OR)

TAMP option register

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x50

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IN4_RMP	IN3_RMP	IN2_RMP	IN1_RMP
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3 **IN4_RMP**：TAMP_IN4 映射
0x0：TAMP_IN4 映射在 PG8 上
0x1：TAMP_IN4 映射在 PI3 上

位 2 **IN3_RMP**：TAMP_IN3 映射
0x0：TAMP_IN3 映射在 PC0 上
0x1：TAMP_IN3 映射在 PI2 上

位 1 **IN2_RMP**：TAMP_IN2 映射
0x0：TAMP_IN2 映射在 PA6 上
0x1：TAMP_IN2 映射在 PI1 上

位 0 **IN1_RMP**：TAMP_IN1 映射
0x0：TAMP_IN1 映射在 PF10 上
0x1：TAMP_IN1 映射在 PC13 上

47.6.19 TAMP 擦除配置寄存器 (TAMP_ERCFGR)

TAMP erase configuration register

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x54

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ER CFG0
															rw

位 31 保留，必须保持复位值。

位 30:8 保留，必须保持复位值。

位 7 保留，必须保持复位值。

位 6 保留，必须保持复位值。

位 5 保留，必须保持复位值。

位 4 保留，必须保持复位值。

位 3 保留，必须保持复位值。

位 2 保留，必须保持复位值。

位 1 保留，必须保持复位值。

位 0 **ERCFG0**：可配置资源 0 保护 (Configurable resource 0 protection)(1)

0： 资源 0 不包含在由 TAMP 外设保护的设备机密信息中

1：资源 0 包含在由 TAMP 外设保护的设备机密信息中

1. 请参见*表 406：TAMP 输入/输出引脚*和*表 408：TAMP 互连*中的 tamp_confirmed_ercfg0 和 tamp_potential_ercfg0 信号。

47.6.20 TAMP 备份 x 寄存器 (TAMP_BKPxR)

TAMP backup x register

可对该寄存器设置保护，以防其遭到非安全访问。请参见*第 47.3.5 节：TAMP 安全保护模式*。

可对该寄存器设置保护，以防其遭到非特权访问。请参见*第 47.3.7 节：TAMP 特权保护模式*。

偏移地址：0x100 + 0x04 * x，（x = 0 到 31）

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BKP[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BKP[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	rw	rw

位 31:0 **BKP[31:0]**：

应用可向/从这些寄存器写入/读取数据。

在默认（擦除）配置中，该寄存器将在发生 Tamper 检测事件时复位。只要至少有一个内部或外部 Tamper 标志置 1，该寄存器就会强制设为复位值。

47.6.21 TAMP 硬件配置寄存器 2 (TAMP_HWCFGR2)

TAMP hardware configuration register 2

偏移地址：0x3EC

复位值：0x0100 0104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUM_DEV_SECRETS[7:0]								CID_WIDTH[3:0]				RIF[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TRUST_ZONE[3:0]				OPTIONREG_OUT[7:0]							
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 **NUM_DEV_SECRETS[7:0]**：

0：TAMP_ERCFGR 未实现

1：TAMP_ERCFGR 通过 1 个位实现

2：TAMP_ERCFGR 通过 2 个位实现

位 23:20 **CID_WIDTH[3:0]**：CID 长度
1：CID 长度为 1 位
2：CID 长度为 2 位
3：CID 长度为 3 位
4：CID 长度为 4 位

位 19:16 **RIF[3:0]**：CID 分区过滤
0x0：禁止 CID 分区过滤
0x1：使能 CID 分区过滤

位 15:12 保留，必须保持复位值。

位 11:8 **TRUST_ZONE[3:0]**：信任区域
0x0：不支持 Trust_zone
0x1：支持 Trust_zone

位 7:0 **OPTIONREG_OUT[7:0]**：
0：未实现配置寄存器
1：配置寄存器位 0 已实现
2：配置寄存器位 1:0 已实现
...
32：配置寄存器位 31:0 已实现

47.6.22 TAMP 硬件配置寄存器 1 (TAMP_HWCFGR1)

TAMP hardware configuration register 1

偏移地址：0x3F0

复位值：0x1FFF 1820

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT_TAMPER[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACTIVE_TAMPER[3:0]				TAMPER[3:0]				BACKUP_REGS[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 **INT_TAMPER[15:0]**：
INT_TAMPER[i] = 0：内部 Tamper i 未实现（i 从 0 到 15）
INT_TAMPER[i] = 1：内部 Tamper i 已实现（i 从 0 到 15）

位 15:12 **ACTIVE_TAMPER[3:0]**：
0x0：不支持主动 Tamper
0x1：支持所有外部 Tamper 事件的主动 Tamper

位 11:8 **TAMPER[3:0]**：
0x0：支持 16 个外部 Tamper 事件（输入引脚）
0x1：支持 1 个外部 Tamper 事件（输入引脚）
0x2：支持 2 个外部 Tamper 事件（输入引脚）
...
0xF：支持 15 个外部 Tamper 事件（输入引脚）

位 7:0 **BACKUP_REGS[7:0]**：
0：无备份寄存器
1：实现了 1 个备份寄存器
2：实现了 2 个备份寄存器
...
128：实现了 128 个备份寄存器

47.6.23 TAMP 版本寄存器 (TAMP_VERR)

TAMP version register

偏移地址：0x3F4

复位值：0x0000 003X

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本 (Major revision)

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)

47.6.24 TAMP 标识寄存器 (TAMP_IPIDR)

TAMP identification register

偏移地址：0x3F8

复位值：0x0012 1033

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：标识符 (Identifier)。

47.6.25 TAMP 大小标识寄存器 (TAMP_SIDR)

TAMP size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识符 (Size identifier)。

47.6.26 TAMP 寄存器映射

表 418. TAMP 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TAMP_CR1	Res.	Res.	Res.	ITAMP13E	ITAMP12E	ITAMP11E	ITAMP10E	ITAMP9E	ITAMP8E	ITAMP7E	ITAMP6E	ITAMP5E	ITAMP4E	ITAMP3E	ITAMP2E	ITAMP1E	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8E	TAMP7E	TAMP6E	TAMP5E	TAMP4E	TAMP3E	TAMP2E	TAMP1E
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x04	TAMP_CR2	TAMP8TRG	TAMP7TRG	TAMP6TRG	TAMP5TRG	TAMP4TRG	TAMP3TRG	TAMP2TRG	TAMP1TRG	BKERASE	BKBLOCK	Res.	Res.	Res.	TAMP3MSK	TAMP2MSK	TAMP1MSK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8NOER	TAMP7NOER	TAMP6NOER	TAMP5NOER	TAMP4NOER	TAMP3NOER	TAMP2NOER	TAMP1NOER
	Reset value	0	0	0	0	0	0	0	0	0	0				0	0	0									0	0	0	0	0	0	0	0
0x08	TAMP_CR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITAMP13NOER	ITAMP12NOER	ITAMP11NOER	ITAMP10NOER	ITAMP9NOER	ITAMP8NOER	ITAMP7NOER	ITAMP6NOER	ITAMP5NOER	ITAMP4NOER	ITAMP3NOER	ITAMP2NOER	ITAMP1NOER
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	TAMP_FLTCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMPPUDIS	.TAMPPRCH[1:0]		.TAMPFLT[1:0]		TAMPFREQ[2:0]		
	Reset value																									0	0	0	0	0	0	0	0
0x10	TAMP_ATCR1	FLTEN	ATOSHARE	Res.	Res.	Res.	AT PER[2:0]			Res.	Res.	Res.	Res.	ATCK SEL[3:0]				ATO SEL4 [1:0]		ATO SEL3 [1:0]		ATO SEL2 [1:0]		ATO SEL1 [1:0]		TAMP8AM	TAMP7AM	TAMP6AM	TAMP5AM	TAMP4AM	TAMP3AM	TAMP2AM	TAMP1AM
	Reset value	0	0				0	0	0					0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	TAMP_ ATSEEDR	SEED[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	TAMP_ATOR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INITS	SEEDF	Res.	Res.	Res.	Res.	Res.	Res.	PRNG[7:0]							
	Reset value																	0	0							0	0	0	0	0	0	0	0
0x1C	TAMP_ATCR2	ATO SEL8 [2:0]			ATO SEL7 [2:0]			ATO SEL6 [2:0]			ATO SEL5 [2:0]			ATO SEL4 [2:0]			ATO SEL3 [2:0]			ATO SEL2 [2:0]			ATO SEL1 [2:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0x20	TAMP_SEC CFGR	TAMPSEC	BHKLOCK	Res.	Res.	Res.	Res.	Res.	Res.	BKPWSEC[7:0]								CNT1SEC	CNT2SEC	Res.	Res.	Res.	Res.	Res.	Res.	BKPRWSEC[7:0]							
	Reset value	0	0							0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0
0x24	TAMP_ PRIVCFGR	TAMPPRIV	BKPWPRIV	BKPRWPRIV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT1PRIV	CNT2SEC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0														0	0														

表 418. TAMP 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x2C	TAMP_IER	Res.	Res.	Res.	ITAMP13IE	ITAMP12IE	ITAMP11IE	ITAMP10IE	ITAMP9IE	ITAMP8IE	ITAMP7IF	ITAMP6IE	ITAMP5IE	ITAMP4IE	ITAMP3IE	ITAMP2IE	.ITAMP1IE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8IE	TAMP7IE	TAMP6IE	TAMP5IE	TAMP4IE	TAMP3IE	TAMP2IE	TAMP1IE
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x30	TAMP_SR	Res.	Res.	Res.	ITAMP13F	ITAMP12F	ITAMP11F	ITAMP10F	ITAMP9F	ITAMP8F	ITAMP7F	ITAMP6F	ITAMP5F	ITAMP4F	ITAMP3F	ITAMP2F	.ITAMP1F	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8F	TAMP7F	TAMP6F	TAMP5F	TAMP4F	TAMP3F	TAMP2F	TAMP1F
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x34	TAMP_MISR	Res.	Res.	Res.	ITAMP13MF	ITAMP12MF	ITAMP11MF	ITAMP10MF	ITAMP9MF	ITAMP8MF	ITAMP7MF	ITAMP6MF	ITAMP5MF	ITAMP4MF	ITAMP3MF	ITAMP2MF	ITAMP1MF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8MF	TAMP7MF	TAMP6MF	TAMP5MF	TAMP4MF	TAMP3MF	TAMP2MF	TAMP1MF
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x38	TAMP_SMISR	Res.	Res.	Res.	ITAMP13F	ITAMP12F	ITAMP11F	ITAMP10F	ITAMP9F	ITAMP8F	ITAMP7F	ITAMP6MF	ITAMP5MF	ITAMP4MF	ITAMP3MF	ITAMP2MF	ITAMP1MF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8MF	TAMP7MF	TAMP6MF	TAMP5MF	TAMP4MF	TAMP3MF	TAMP2MF	TAMP1MF
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x3C	TAMP_SCR	Res.	Res.	Res.	CITAMP13F	CITAMP12F	CITAMP11F	CITAMP10F	CITAMP9F	CITAMP8F	CITAMP7F	CITAMP6F	CITAMP5F	CITAMP4F	CITAMP3F	CITAMP2F	CITAMP1F	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTAMP8F	CTAMP7F	CTAMP6F	CTAMP5F	CTAMP4F	CTAMP3F	CTAMP2F	CTAMP1F
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x40	TAMP_ COUNTR	COUNT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	TAMP_ COUNT2R	COUNT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	TAMP_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IN4_RMP	IN3_RMP	IN2_RMP	IN1_RMP
	Reset value																													0	0	0	0
0x54	TAMP_ ERCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERCFG0
	Reset value																																0
0x100 + 0x04*x, (x= 0 to 31)	TAMP_BKPxR	BKP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3EC	TAMP_ HWCFGR2	NUM_DEV_SECRETS[7:0]								CID_WIDTH [3:0]				RIF[3:0]				Res.	Res.	Res.	Res.	TRUST_ ZONE[3:0]				OPTIONREG_OUT[7:0]							
	Reset value	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0					0	0	0	1	0	0	0	0	0	1	0	0
0x3F0	TAMP_ HWCFGR1	INT_TAMPER[15:0]																ACTIVE_ TAMPER [3:0]				TAMPER [3:0]				BACKUP_REGS [7:0]							
	Reset value	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0
0x3F4	TAMP_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV [3:0]				MINREV [3:0]			
	Reset value																									0	0	1	1	x	x	x	x

表 418. TAMP 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x3F8	TAMP_IPIDR	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	1
0x3FC	TAMP_SIDR	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 2.5 节*。

48 内部集成电路 (I2C) 接口

48.1 简介

I^2C 外设负责处理设备与串行 I^2C（集成电路间）总线之间的接口。它提供多主模式功能，可以控制所有 I^2C 总线特定的序列、协议、仲裁和时序。它支持标准模式 (Sm)、快速模式 (Fm) 和超快速模式 (Fm+)。

I^2C 总线接口还与 SMBus（系统管理总线）和 PMBus®（电源管理总线）兼容。

可使用 DMA 来减轻 CPU 的工作量。

48.2 I2C 主要特性

- 兼容 I^2C 总线规范第 03 版：
  - 从模式和主模式
  - 多主模式功能
  - 标准速度模式（高达 100 kHz）
  - 快速模式（高达 400 kHz）
  - 超快速模式（高达 1 MHz）
  - 7 位和 10 位寻址模式
  - 多个 7 位从地址（2 个从设备地址寄存器，1 个具有可配置的掩码位段）
  - 所有 7 位地址应答模式
  - 广播呼叫
  - 总线上的数据建立和保持时间可软件配置
  - 方便易用的事件管理
  - 可选的时钟延长
  - 软件复位
- 带 DMA 功能的 1 字节缓冲
- 可编程模拟和数字噪声滤波器

还可提供以下特性，具体取决于产品实现（请参见*第 48.3 节*）：

- 兼容 SMBus 规范第 3.0 版：
  - 具有 ACK 控制的硬件 PEC（数据包错误校验）生成和验证
  - 命令和数据应答控制
  - 支持地址解析协议 (ARP)
  - 支持主机和从设备
  - SMBus 报警
  - 超时和空闲条件检测
- 兼容 PMBus 第 1.3 版标准
- 独立时钟：选择独立时钟源可使 I2C 通信速度不受 i2c_pclk 重新编程的影响
- 地址匹配时从 Stop 模式唤醒

48.3 I2C 特性实现

本手册介绍了本手册介绍了 I2C 外设中实现的所有特性。在 STM32MP13xx 设备 I2C1、I2C2、I2C3、I2C4 和 I2C5 中实现了如下表所示的所有特性。

表 419. STM32MP13xx I2C 特性实现

I2C 特性(1)	I2C1	I2C2	I2C3	I2C4	I2C5
7 位寻址模式	X	X	X	X	X
10 位寻址模式	X	X	X	X	X
标准速度模式（高达 100 kbit/s）	X	X	X	X	X
快速模式（高达 400 kb/s）	X	X	X	X	X
超快速模式，20 mA 输出驱动 I/O（高达 1 Mb/s）	X	X	X	X	X
独立时钟	X	X	X	X	X
从 Stop 模式唤醒	X	X	X	X	X
SMBus/PMBus	X	X	X	X	X

1. X = 支持。

48.4 I2C 功能描述

除了接收和发送数据之外，此接口还可以从串行格式转换为并行格式，反之亦然。中断由软件使能或禁止。该接口通过数据引脚 (SDA) 和时钟引脚 (SCL) 连接到 I^2C 总线。它可以连接到标准速度（高达 100 kHz）、快速（高达 400 kHz）或超快速（高达 1 MHz）的 I^2C 总线。

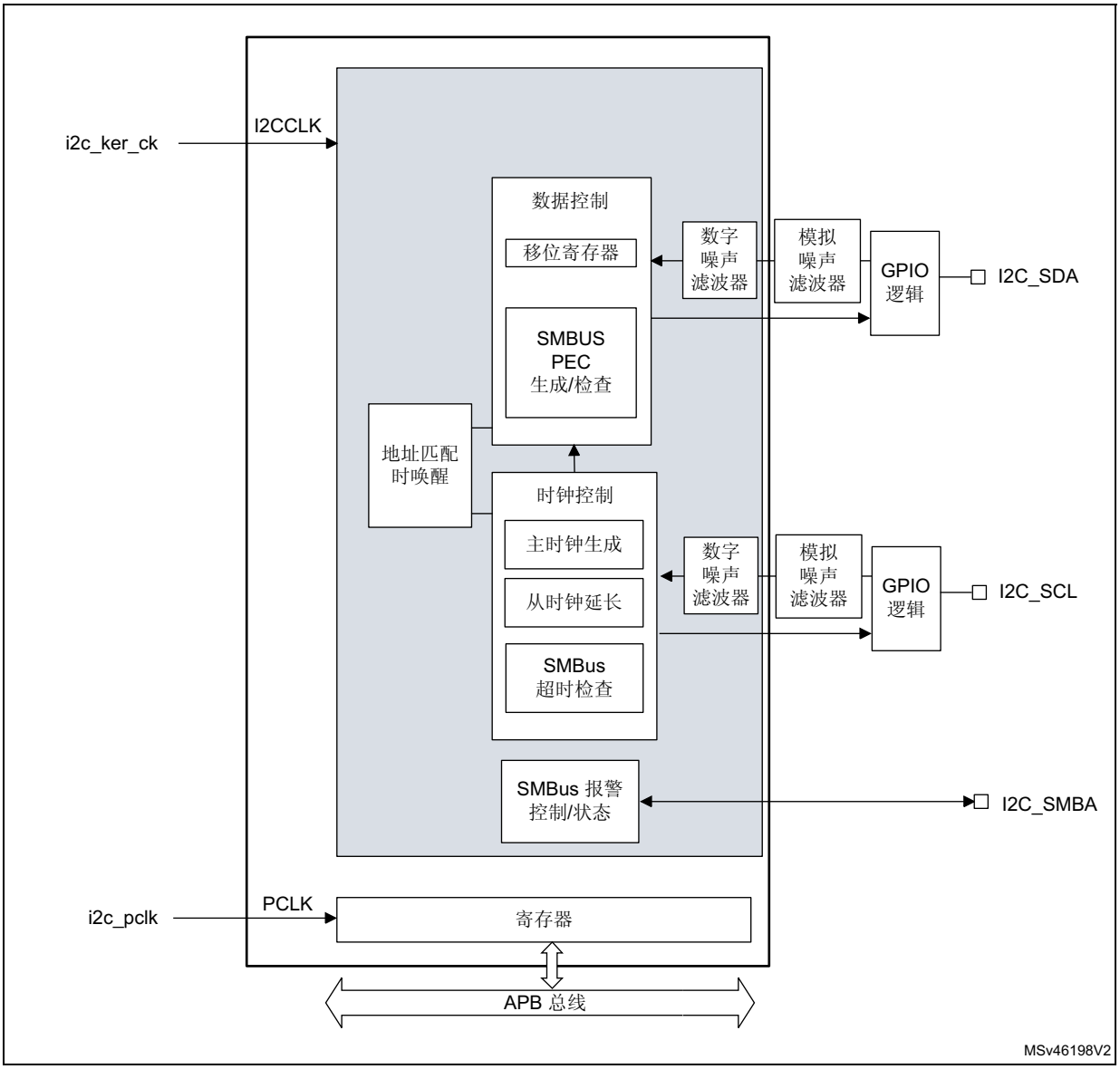
该接口也可通过数据 (SDA) 引脚和时钟 (SCL) 引脚连接到 SMBus。

如果支持 SMBus 功能：还可选用 SMBus 报警引脚 (SMBA)。

48.4.1 I2C 框图

I2C 接口如*图 534* 所示。

图 534. I2C 框图



I2C 的时钟由独立时钟源提供，这使得 I2C 能够独立于 i2c_pclk 频率工作。

对于支持以 20 mA 输出电流驱动超快速模式操作的 I2C I/O，可以通过系统配置控制器 (SYSCFG) 中的控制位使能驱动功能。*请参见第 48.3 节：I2C 特性实现*。

48.4.2 I2C 引脚和内部信号

表 420. I2C 输入/输出引脚

引脚名称	信号类型	说明
I2C_SDA	双向	I2C 数据
I2C_SCL	双向	I2C 时钟
I2C_SMBA	双向	SMBus 报警

表 421. I2C 内部输入/输出信号

内部信号名称	信号类型	说明
i2c_ker_ck	输入	I2C 内核时钟，在本文档中也称为 I2CCLK
i2c_pclk	输入	I2C APB 时钟
i2c_it	输出	I2C 中断，有关中断源的列表，请参见 *表 435*
i2c_rx_dma	输出	I2C 接收数据 DMA 请求 (I2C_RX)
i2c_tx_dma	输出	I2C 发送数据 DMA 请求 (I2C_TX)

48.4.3 I2C 时钟要求

I2C 内核的时钟由 i2c_ker_ck 提供。

i2c_ker_ck 周期 t_{I2CCLK} 必须遵循以下条件：

- $t_{I2CCLK} < (t_{LOW} - t_{filters}) / 4$
- $t_{I2CCLK} < t_{HIGH}$

其中：

t_{LOW}：SCL 低电平时间，t_{HIGH}：SCL 高电平时间

$t_{filters}$：滤波器使能时，该值为模拟滤波器和数字滤波器引入的延时总和。

数字滤波器延时为 DNF x t_{I2CCLK}。

i2c_pclk 时钟周期 t_{PCLK} 必须遵循以下条件：

- $t_{PCLK} < 4 / 3\ t_{SCL}$（$t_{SCL}$：SCL 周期）

小心： 当 I2C 内核的时钟由 i2c_pclk 提供时，该时钟必须遵循 t_{I2CCLK} 的条件。

48.4.4 模式选择

该接口在工作时可选用以下四种模式之一：

- 从发送器
- 从接收器
- 主发送器
- 主接收器

默认情况下，它以从模式工作。接口在生成起始位后会自动由从模式切换为主模式，并在出现仲裁丢失或生成停止位时从主模式切换为从模式，从而实现多主模式功能。

通信流程

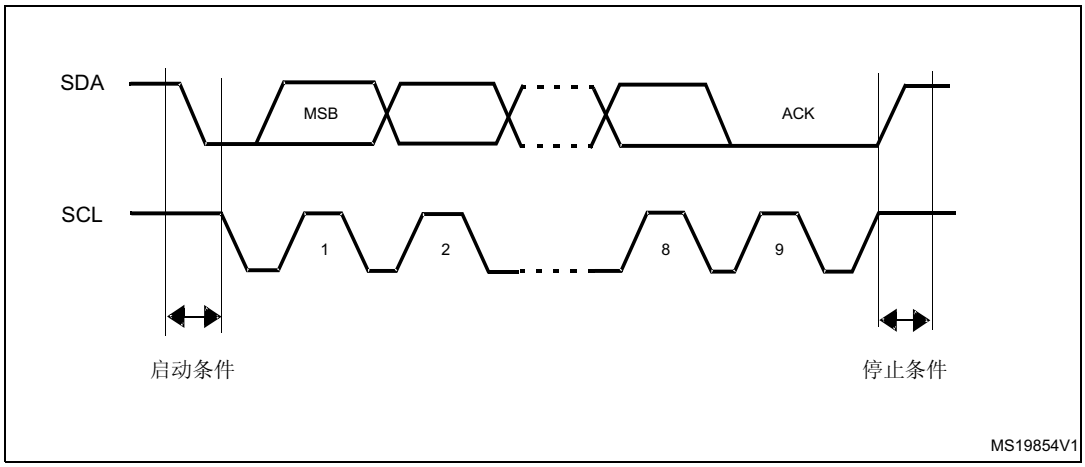
在主模式下，I2C 接口会启动数据传输并生成时钟信号。串行数据传输始终是在出现起始位时开始，在出现停止位时结束。起始位和停止位均在主模式下由软件生成。

在从模式下，该接口能够识别其自身地址（7 或 10 位）以及广播呼叫地址。广播呼叫地址检测可由软件使能或禁止。保留的 SMBus 地址也可由软件使能。

数据和地址均以 8 位字节传输，MSB 在前。起始位后紧随地址字节（7 位地址占据一个字节；10 位地址占据两个字节）。地址始终在主模式下发送。

在字节传输八个时钟周期后是第九个时钟脉冲，在此期间接收器必须向发送器发送一个应答位（请参见*图 535*）。

图 535. I^2C 总线协议



应答位可由软件使能或禁止。I2C 接口地址可通过软件进行选择。

48.4.5 I2C 初始化

使能和禁止外设

必须先在时钟控制器中配置并使能 I2C 外设时钟，之后才能通过将 I2C_CR1 寄存器中的 PE 位置 1 来使能 I2C。

当禁止 I2C (PE = 0) 时，I^2C 将执行软件复位。更多详细信息，请参见*第 48.4.6 节*。

噪声滤波器

通过将 I2C_CR1 寄存器中的 PE 位置 1 来使能 I2C 外设之前，在此之前，用户必须配置噪声滤波器。默认情况下，SDA 和 SCL 输入上开启了模拟噪声滤波器。该滤波器符合 I^2C 规范，此规范要求在快速模式和超快速模式下对脉宽在 50ns 以下的脉冲都要抑制。用户可通过将 ANFOFF 位置 1 来禁止该模拟滤波器，和/或通过配置 I2C_CR1 寄存器中的 DNF[3:0] 位来选择数字滤波器。

使能数字滤波器时，SCL 或 SDA 线的电平只有在电平稳定时间超过 DNF x i2c_ker_ck 个周期后才会发生内部变化，这允许以 1 到 15 个 i2c_ker_ck 周期的可编程长度来抑制尖峰。

表 422. 模拟滤波器与数字滤波器对比

-	模拟滤波器	数字滤波器
抑制尖峰的脉冲宽度	≥ 50 ns	长度可编程为 1 到 15 个 I2C 外设时钟
优点	Stop 模式中仍可用	– 长度可编程：额外的滤波能力与标准要求 – 稳定长度
缺点	随温度、电压和过程变化会发生变化	当使能数字滤波器后，无法在地址匹配时从 Stop 模式唤醒

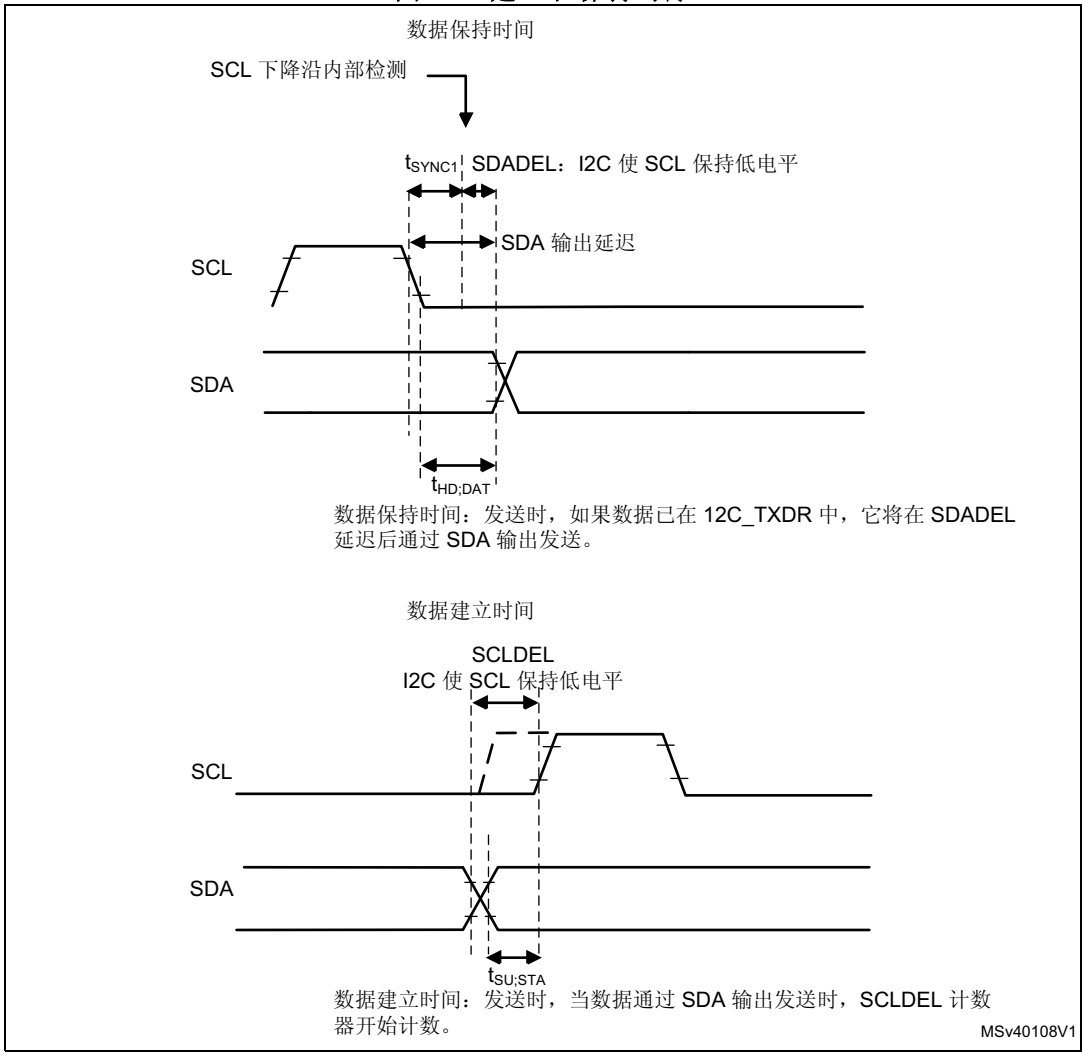
小心： 使能 I2C 时，不能更改滤波器配置。

I2C 时序

必须配置时序，以便保证主模式和从模式下使用正确的数据保持和建立时间。配置方法是编程 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLDEL[3:0] 和 SDADEL[3:0] 位。

STM32CubeMX 工具计算 I2C_TIMINGR 内容并在 I2C 配置窗口中进行显示。

图 536. 建立和保持时序



当内部检测到 SCL 下降沿时，会在发送 SDA 输出之前插入一段延时（t_{SDADEL}，这会影响保持时间 $t_{HD;DAT}$）：t_{SDADEL} = SDADEL x t_{PRESC} + t_{I2CCLK}，其中 t_{PRESC} = (PRESC + 1) x t_{I2CCLK}。

SDA 总输出延时为：

t_{SYNC1} + {[SDADEL x (PRESC + 1) + 1] x t_{I2CCLK}}

t_{SYNC1} 持续时间取决于：

- SCL 下降斜率
- 模拟滤波器（使能时）引入的输入延时：$t_{AF(min)} < t_{AF} < t_{AF(max)}$
- 数字滤波器（使能时）引入的输入延时：t_{DNF} = DNF x t_{I2CCLK}
- SCL 与 i2c_ker_ck 时钟建立同步而产生的延时（2 到 3 个 i2c_ker_ck 周期）

要桥接 SCL 下降沿的未定义区域，用户编程 SDADEL 时必须遵循以下条件：

$\{t_{f\ (max)} + t_{HD;DAT\ (min)} - t_{AF(min)} - [(DNF + 3) \times t_{I2CCLK}]\} / \{(PRESC + 1) \times t_{I2CCLK}\} \leq SDADEL$

$SDADEL \leq \{t_{HD;DAT\ (max)} - t_{AF(max)} - [(DNF + 4) \times t_{I2CCLK}]\} / \{(PRESC + 1) \times t_{I2CCLK}\}$

注： *只有使能模拟滤波器时，公式中才包含 $t_{AF(min)}$ / $t_{AF(max)}$。有关 t_{AF} 值的信息，请参见设备数据手册。*

关于 $t_{HD;DAT}$ 最大值，标准模式下可达 3.45 µs，快速模式可达 0.9 µs，超快速模式下可达 0.45 µs。但必须小于 $t_{VD;DAT}$ 最大值（差值为跳变时间）。只有设备未延长 SCL 信号的低电平周期 (t_{LOW}) 时，才必须满足该最大值条件。如果时钟延长 SCL，数据必须在建立时间内保持有效，之后才能释放时钟。

SDA 上升沿通常为最坏情况。在这种情况下，上述公式变成如下形式：

$SDADEL \leq \{t_{VD;DAT\ (max)} - t_{r\ (max)} - t_{AF\ (max)} - [(DNF + 4) \times t_{I2CCLK}]\} / \{(PRESC + 1) \times t_{I2CCLK}\}$。

注： *NOSTRETCH=0 时会违反该条件，这是因为设备会根据 SCLDEL 值来延长 SCL 低电平时间，以保证建立时间。*

有关 t_f、t_r、$t_{HD;DAT}$ 和 $t_{VD;DAT}$ 标准值的信息，请参见 *表 423*。

- 在 t_{SDADEL} 之后，或者在发送 SDA 输出之后，当从机由于数据尚未写入 I2C_TXDR 寄存器而必须延长时钟时，SCL 线在设置时间内保持在低电平。该建立时间为 t_{SCLDEL} = (SCLDEL + 1) x t_{PRESC}，其中 t_{PRESC} = (PRESC + 1) x t_{I2CCLK}。t_{SCLDEL} 会影响建立时间 $t_{SU;DAT}$。

要桥接 SDA 跳变（上升沿通常为最坏情况）的未定义区域，编程 SCLDEL 时必须遵循以下条件：

$\{[t_{r\ (max)} + t_{SU;DAT\ (min)}] / [(PRESC + 1)] \times t_{I2CCLK}]\} - 1 \leq SCLDEL$

有关 t_r 和 $t_{SU;DAT}$ 标准值的信息，请参见 *表 423*。

将使用的 SDA 和 SCL 跳变时间值就是应用中的值。使用最大值而非标准值会增加 SDADEL 和 SCLDEL 计算的约束条件，但能够确保任意应用的特性。

注： *在发送和接收模式下，对于每个时钟脉冲，检测到 SCL 下降沿后，I2C 主设备或从设备会至少在 [(SDADEL+SCLDEL+1) x (PRESC+1) + 1] x t_{I2CCLK} 期间内延长 SCL 低电平时间。在发送模式下，如果 SDADEL 计数器计数结束后数据还未写入 I2C_TXDR，则 I2C 会继续延长 SCL 低电平时间，直到写入下一个数据。随后，会将新数据 MSB 发送到 SDA 输出，SCLDEL 计数器将开始计数，同时会继续延长 SCL 低电平时间以确保提供充足的数据建立时间。*

如果从模式下 NOSTRETCH = 1，则 SCL 不会延长，因此编程 SDADEL 时必须确保提供充足的建立时间。

表 423. I²C-SMBus 规范数据建立和保持时间

符号	参数	标准模式 (Sm)		快速模式 (Fm)		超快速模式 (Fm+)		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
$t_{HD;DAT}$	数据保持时间	0	-	0	-	0	-	0.3	-	µs
$t_{VD;DAT}$	数据有效时间	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	数据建立时间	250	-	100	-	50	-	250	-	ns
t_r	SDA 和 SCL 信号的上升时间	-	1000	-	300	-	120	-	1000	
t_f	SDA 和 SCL 信号的上升时间	-	300	-	300	-	120	-	300	

此外，在主模式下，必须通过编程 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLH[7:0] 和 SCLL[7:0] 位字段来配置 SCL 时钟的高电平和低电平。

- 当内部检测到 SCL 下降沿时，会在释放 SCL 输出之前插入一段延时。该延时为 $t_{SCLL} = (SCLL + 1) \times t_{PRESC}$，其中 $t_{PRESC} = (PRESC + 1) \times t_{I2CCLK}$。
  t_{SCLL} 会影响 SCL 低电平时间 t_{LOW}。
- 当内部检测到 SCL 上升沿时，会在将 SCL 输出强制为低电平之前插入一段延时。该延时为 $t_{SCLH} = (SCLH + 1) \times t_{PRESC}$，其中 $t_{PRESC} = (PRESC+ 1) \times t_{I2CCLK}$。$t_{SCLH}$ 会影响 SCL 高电平时间 t_{HIGH}。

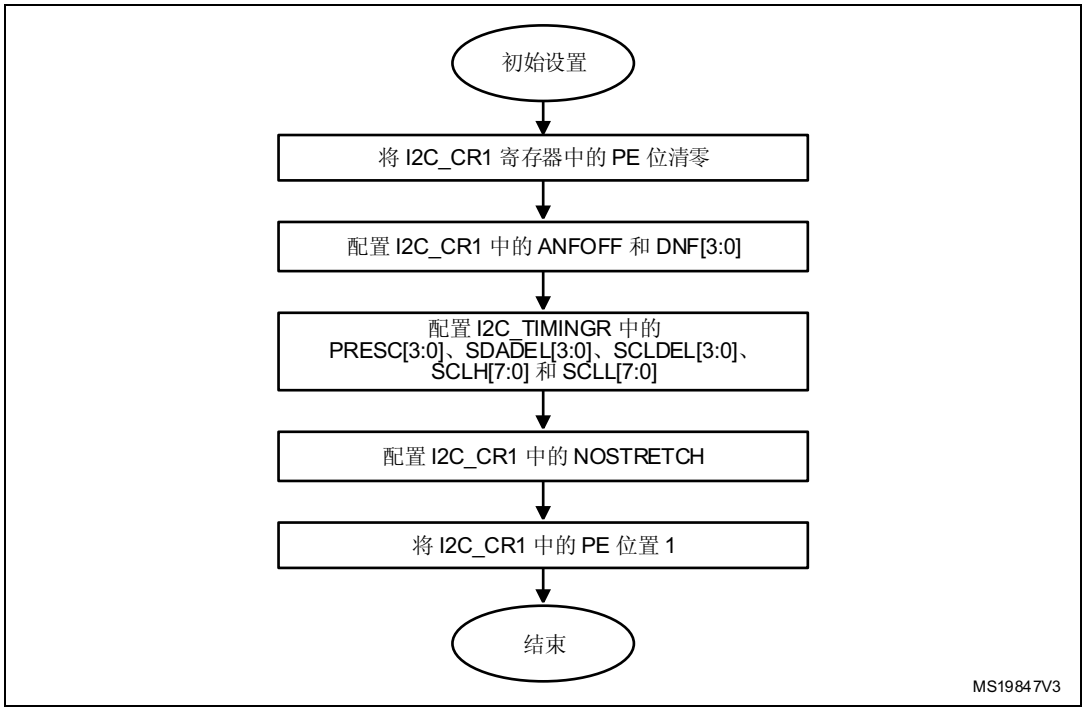
更多详细信息，请参见 *I2C 主模式初始化*。

小心： 使能 I2C 后，不允许更改时序配置。

此外，还必须在使能外设之前配置 I2C 从 NOSTRETCH 模式。更多详细信息，请参见 *I2C 从模式初始化*。

小心： 使能 I2C 后，不允许更改 NOSTRETCH 配置。

图 537. I2C 初始化流程



48.4.6 软件复位

可通过将 I2C_CR1 寄存器中的 PE 位清零来执行软件复位。在这种情况下，I2C 线 SCL 和 SDA 被释放。内部状态机复位，通信控制位和状态位恢复为其复位值。配置寄存器不受影响。

受影响的寄存器位：

1. I2C_CR2 寄存器：START、STOP 和 NACK
2. I2C_ISR 寄存器：BUSY、TXE、TXIS、RXNE、ADDR、NACKF、TCR、TC、STOPF、BERR、ARLO 和 OVR

支持 SMBus 功能时还会影响到以下寄存器位：

1. I2C_CR2 寄存器：PECBYTE
2. I2C_ISR 寄存器：PECERR、TIMEOUT 和 ALERT

必须使 PE 保持低电平持续至少三个 APB 时钟周期，才能成功执行软件复位。使用以下软件序列可确保这一点：

1. 写入 PE = 0
2. 检查 PE = 0
3. 写入 PE = 1

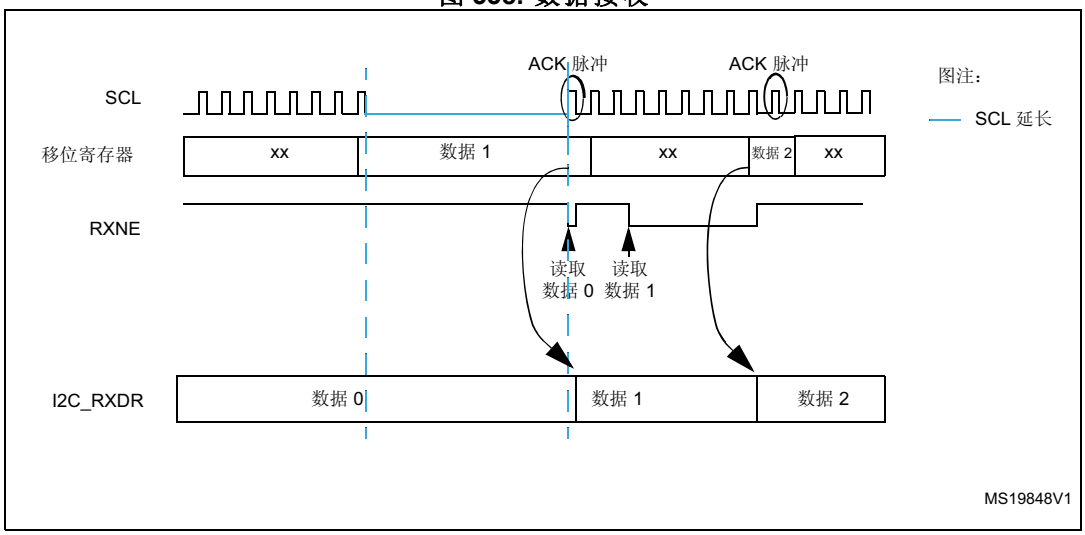
48.4.7 数据传输

数据传输由发送和接收数据寄存器以及移位寄存器来管理。

接收

SDA 输入填充移位寄存器。在第八个 SCL 脉冲后（接收到完整的数据字节时），如果 I2C_RXDR 寄存器为空 (RXNE=0)，则移位寄存器的内容会复制到其中。如果 RXNE = 1（意味着尚未读取上一次接收到的数据字节），则将延长 SCL 线的低电平时间，直到读取了 I2C_RXDR 为止。在第八个和第九个 SCL 脉冲之间（应答脉冲之前）插入一段延长的时间。

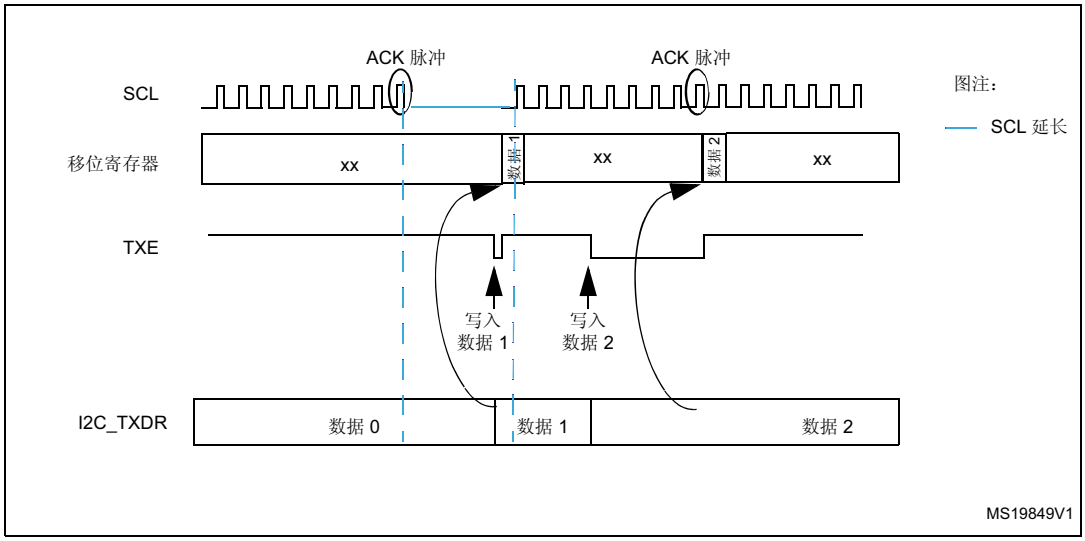
图 538. 数据接收



发送

如果 I2C_TXDR 寄存器不为空 (TXE = 0)，则其内容会在第九个 SCL 脉冲（应答脉冲）后复制到移位寄存器中。然后移位寄存器的内容会移出到 SDA 线上。如果 TXE = 1（意味着 I2C_TXDR 内尚未写入任何数据），则将延长 SCL 线的低电平时间，直到写入了 I2C_TXDR 为止。在第九个 SCL 脉冲后进行延长。

图 539. 数据发送



硬件传输管理

I2C 内置了字节计数器，以便在下列各种模式下管理字节传输和结束通信：

- 主模式下生成 NACK、STOP 和 ReSTART
- 从接收器模式下控制 ACK 是否发出
- SMBus 模式下生成/校验 PEC

字节计数器通常在主模式下使用。在从模式下，字节计数器默认为禁止状态。可以通过软件来使能，方法是将 I2C_CR1 寄存器中的 SBC（从字节控制）位置 1。

待传输的字节数在 I2C_CR2 寄存器的 NBYTES[7:0] 位字段中进行编程。如果待传输的字节数 (NBYTES) 大于 255，或者如果接收器想要控制接收到的数据字节的应答值，就必须通过设置 I2C_CR2 寄存器中的 RELOAD 位来选择重载模式。在该模式下，当 NBYTES 中编程的字节数传输完成时，TCR 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TCR 标志置 1，SCL 便会延长。向 NBYTES 写入一个非零值时，TCR 由软件清零。

在 NBYTES 中设置最后一次传输的字节数前，必须将 RELOAD 位清零。

当主模式下 RELOAD = 0 时，可在以下两种模式下使用计数器：

- **自动结束**（I2C_CR2 寄存器中的 AUTOEND = 1）。在该模式下，一旦完成 NBYTES[7:0] 位字段中所编程字节数的数据传输，主设备便会自动发送停止位。
- **软件结束**（I2C_CR2 寄存器中的 AUTOEND = 0）。在该模式下，一旦完成 NBYTES[7:0] 位字段中所编程字节数的数据传输，TC 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TC 标志置 1，SCL 信号便会延长，需要软件介入操作。当软件将 I2C_CR2 寄存器中的起始位或停止位置 1 时，TC 标志将被清零。当主设备要发送重复起始位时，必须使用该模式。

小心： 当 RELOAD 位置 1 时，AUTOEND 位将不起作用。

表 424. I2C 配置

功能	SBC 位	RELOAD 位	AUTOEND 位
主 Tx/Rx NBYTES + STOP	x	0	1
主 Tx/Rx + NBYTES + RESTART	x	0	0

表 424. I2C 配置（续）

功能	SBC 位	RELOAD 位	AUTOEND 位
从 Tx/Rx，接收的所有字节都要回复应答	0	x	x
具有 ACK 控制的从 Rx	1	1	x

48.4.8 I2C 从模式

I2C 从模式初始化

要在从模式下工作，用户必须至少使能一个从地址。可使用 I2C_OAR1 和 I2C_OAR2 这两个寄存器来编程自身从地址 OA1 和 OA2。

- OA1 既可配置为 7 位寻址模式（默认），也可通过将 I2C_OAR1 寄存器的 OA1MODE 位置 1 配置为 10 位寻址模式。

  通过将 I2C_OAR1 寄存器中的 OA1EN 位置 1 来使能 OA1。
- 如果需要额外的从地址，可配置第二个从地址 OA2。将 I2C_OAR2 寄存器的 OA2MSK[2:0] 位置 1 最多可屏蔽七个 OA2 LSB。因此，当 OA2MSK 配置为 1 到 6 时，将分别只有 OA2[7:2]、OA2[7:3]、OA2[7:4]、OA2[7:5]、OA2[7:6] 或 OA2[7] 与接收到的地址作比较。只要 OA2MSK 不等于 0，OA2 的地址比较器便会排除 I2C 保留地址（0000 XXX 和 1111 XXX），这些地址将不会得到应答。如果 OA2MSK = 7，接收到的所有 7 位地址（保留地址除外）均得到应答。OA2 始终为 7 位地址。

  如果这些保留地址在 I2C_OAR1 或 I2C_OAR2 寄存器中进行了编程并且 OA2MSK = 0，则它们可以在通过特定使能位使能后得到应答。

  通过将 I2C_OAR2 寄存器中的 OA2EN 位置 1 来使能 OA2。
- 通过将 I2C_CR1 寄存器中的 GCEN 位置 1 来使能广播呼叫地址。

当通过 I2C 的其中一个使能地址来寻址到该 I2C 设备时，ADDR 中断状态标志将置 1，并且 ADDRIE 位置 1 时将生成中断。

默认情况下，从设备使用其时钟延长功能（即必要时延长 SCL 信号的低电平时间）来为软件操作的执行提供时机。如果主设备不支持时钟延长，则必须对 I2C 进行如下配置：将 I2C_CR1 寄存器中 NOSTRETCH 位置 1。

接收到 ADDR 中断后，如果使能多个地址，则用户必须读取 I2C_ISR 寄存器中的 ADDCODE[6:0] 位，以确定是哪个地址匹配。还必须检查 DIR 标志，以获悉传输方向。

带时钟延长的从模式 (NOSTRETCH = 0)

在默认模式下，I2C 从设备会在以下情况下延长 SCL 时钟：

- ADDR 标志置 1 时：接收到的地址与其中一个使能的从地址匹配。通过软件将 ADDRCF 位置 1 以清零 ADDR 标志时，将释放该时钟延长。
- 发送时，前一次数据传输已完成但 I2C_TXDR 寄存器中未写入任何新数据，或者 ADDR 标志清零 (TXE = 1) 时未写入第一个数据字节。在 I2C_TXDR 寄存器中写入数据时，将释放该时钟延长。
- 接收时，尚未读取 I2C_RXDR 寄存器但新的数据接收已完成。读取 I2C_RXDR 时，将释放该时钟延长。
- 当从设备字节控制模式和重载模式（SBC = 1 且 RELOAD = 1）下 TCR = 1 时，这意味着最后一个数据字节已完成传输。通过向 NBYTES[7:0] 位字段写入一个非零值以将 TCR 清零时，将释放该时钟延长。
- 检测到 SCL 下降沿后，I2C 会在 $[(SDADEL + SCLDEL + 1) \times (PRESC+ 1) + 1] \times t_{I2CCLK}$ 期间内延长 SCL 低电平时间。

不带时钟延长的从模式 (NOSTRETCH = 1)

当 I2C_CR1 寄存器中的 NOSTRETCH = 1 时，I2C 从设备不会延长 SCL 信号。

- ADDR 标志置 1 时，不会延长 SCL 时钟。
- 发送时，必须在与发送数据对应的第一个 SCL 脉冲出现之前，向 I2C_TXDR 寄存器写入数据。否则，会发生下溢，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。当第一次数据发送开始而 STOPF 位仍置 1（尚未清零）时，OVR 标志也将置 1。因此，如果写入下一次传输要发送的第一个数据后才清零上一次传输的 STOPF 标志，则应提供 OVR 状态，甚至对于待发送的第一个数据也是如此。
- 接收时，必须在下一个数据字节的第九个 SCL 脉冲（ACK 脉冲）出现之前，从 I2C_RXDR 寄存器读取数据。否则，会发生上溢，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

从设备字节控制模式

要在从接收模式下实现字节 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从设备字节控制模式。这样符合 SMBus 标准。

要在从接收模式下实现字节 ACK 控制，必须选择重载模式 (RELOAD = 1)。要控制每个字节，必须在 ADDR 中断子程序中将 NBYTES 初始化为 0x1，并在每接收一个字节后将 NBYTES 重载为 0x1。接收到字节后，TCR 位将置 1，从而延长 SCL 信号的第八个和第九个脉冲之间的低电平时间。用户可以从 I2C_RXDR 寄存器中读取数据，然后通过配置 I2C_CR2 寄存器中的 ACK 位来决定是否应答。通过将 NBYTES 编程为非零值来释放 SCL 延长：发送应答或不应答信号，然后可继续接收下一个字节。

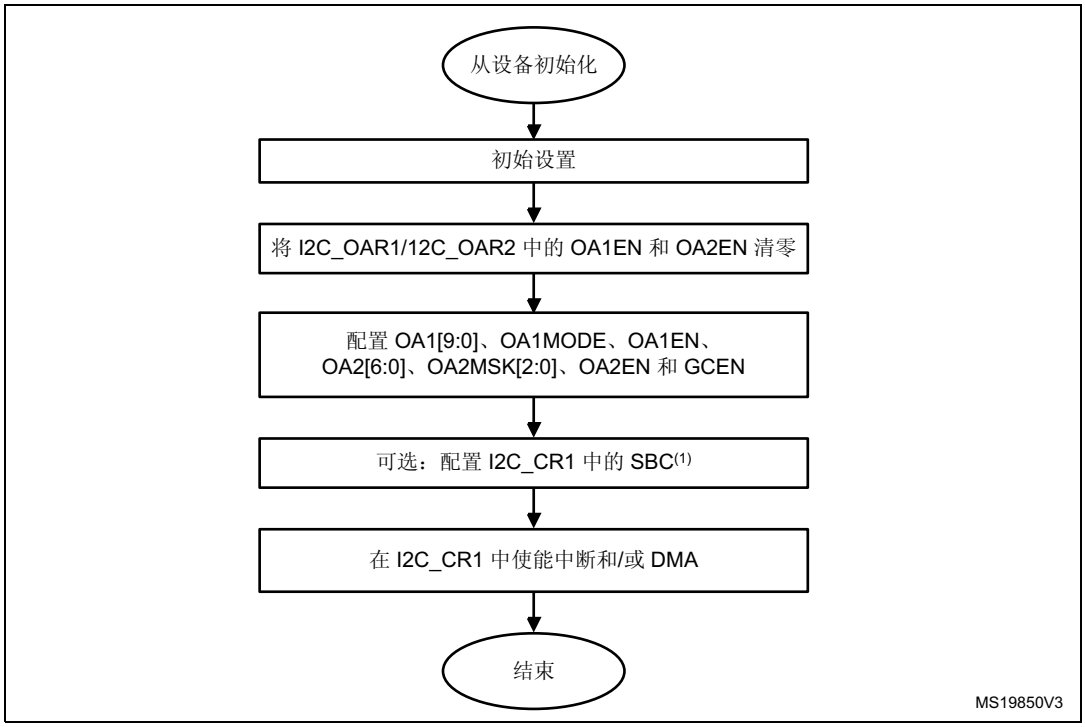
NBYTES 可加载大于 0x1 的值，在这种情况下，接收流在 NBYTES 个数据接收期间是连续的。

注： *SBC 位只能在 I2C 被禁止时、从设备不被寻址时或 ADDR = 1 时配置。*

ADDR = 1 或 TCR = 1 时，可以更改 RELOAD 位的值。

小心： 从设备字节控制模式与 NOSTRETCH 模式不兼容。不允许在 NOSTRETCH = 1 时将 SBC 位置 1。

图 540. 从设备初始化流程



1. 必须将 SBC 置 1 才能支持 SMBus 功能

从发送器

当 I2C_TXDR 寄存器为空时，将生成发送中断状态 (TXIS)。如果 I2C_CR1 寄存器中的 TXIE 位置 1，将产生中断。

I2C_TXDR 寄存器中写入待发送的下一个数据字节时，TXIS 位将被清零。

接收到 NACK 时，I2C_ISR 寄存器中的 NACKF 位将置 1，如果 I2C_CR1 寄存器中的 NACKIE 位置 1，还将生成中断。从机自动释放 SCL 和 SDA 线，以便主机执行停止或重新启动操作。收到 NACK 时，TXIS 位不会置 1。

当接收到 STOP 条件且 I2C_CR1 寄存器中的 STOPIE 位置 1 时，I2C_ISR 寄存器中的 STOPF 标志将置 1 并且会生成中断。在大多数应用中，SBC 位通常编程为 0。在这种情况下，如果接收到从地址 (ADDR=1) 时 TXE = 0，用户可以选择发送 I2C_TXDR 寄存器的内容作为第一个数据字节，也可以选择通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器以编程新的数据字节。

在从设备字节控制模式 (SBC = 1) 下，必须在地址匹配中断子程序 (ADDR = 1) 中向 NBYTES 写入待发送数据的个数。在这种情况下，传输期间 TXIS 事件的数量对应于 NBYTES 中编程的值。

小心： 如果 NOSTRETCH = 1，当 ADDR 标志置 1 时不会延长 SCL 时钟，因此用户无法在 ADDR 子程序中刷新 I2C_TXDR 寄存器的内容，从而编程第一个数据字节。必须在 I2C_TXDR 寄存器中预编程待发送的第一个数据字节：

- 该数据可以是前一个发送消息的最后一个 TXIS 事件中写入的数据。
- 如果该数据字节不是待发送的数据字节，可通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器，从而编程新的数据字节。必须仅在执行完这些操作后再清零 STOPF 位，以确保在地址应答之后第一次数据发送之前执行这些操作。

  如果第一次数据发送开始时 STOPF 仍置 1，则将生成下溢错误（OVR 标志置 1）。

  如果需要 TXIS 事件（发送中断或发送 DMA 请求），用户除了设置 TXE 位之外，还必须设置 TXIS 位来产生该事件。

图 541. I2C 从发送器的传输序列流程 (NOSTRETCH = 0)

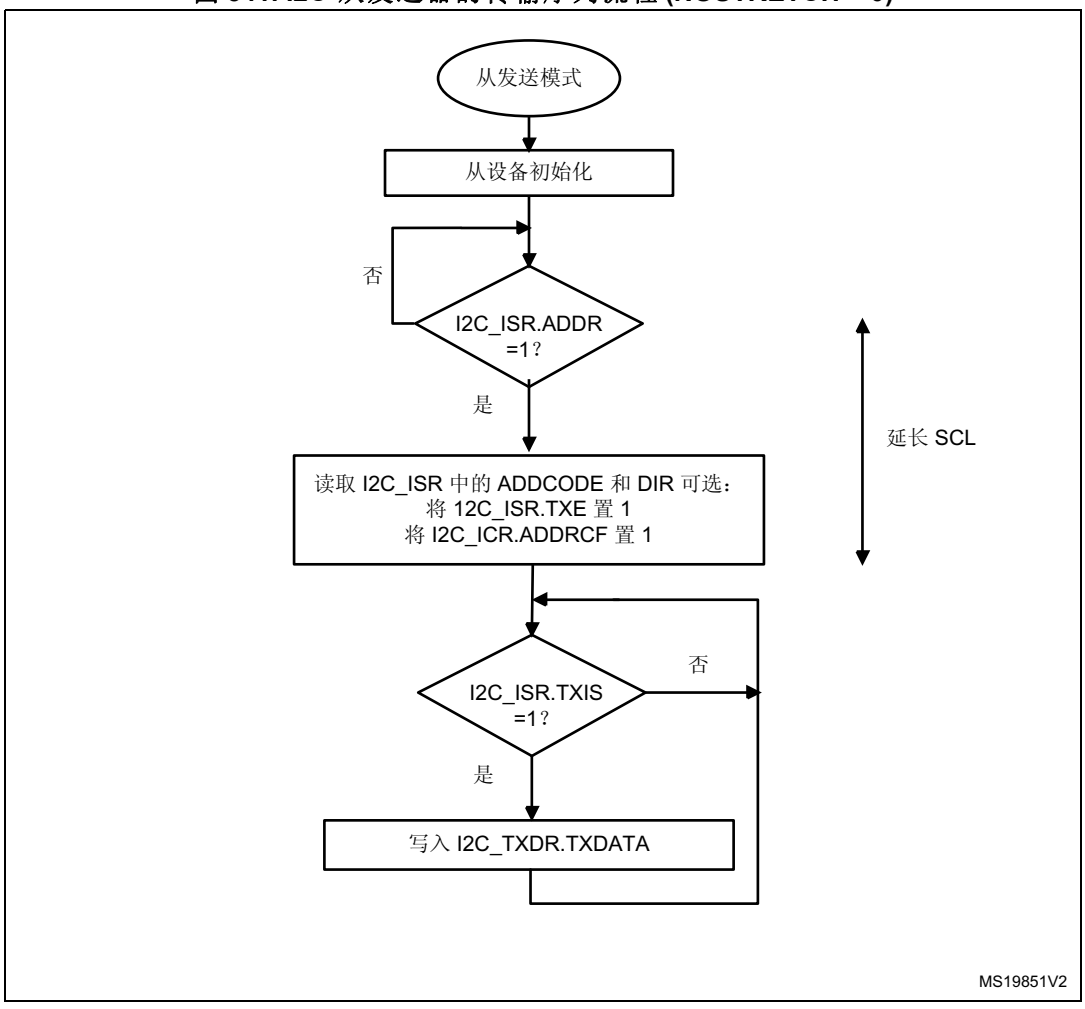


图 542. I2C 从设备发送器的传输序列流程 (NOSTRETCH = 1)

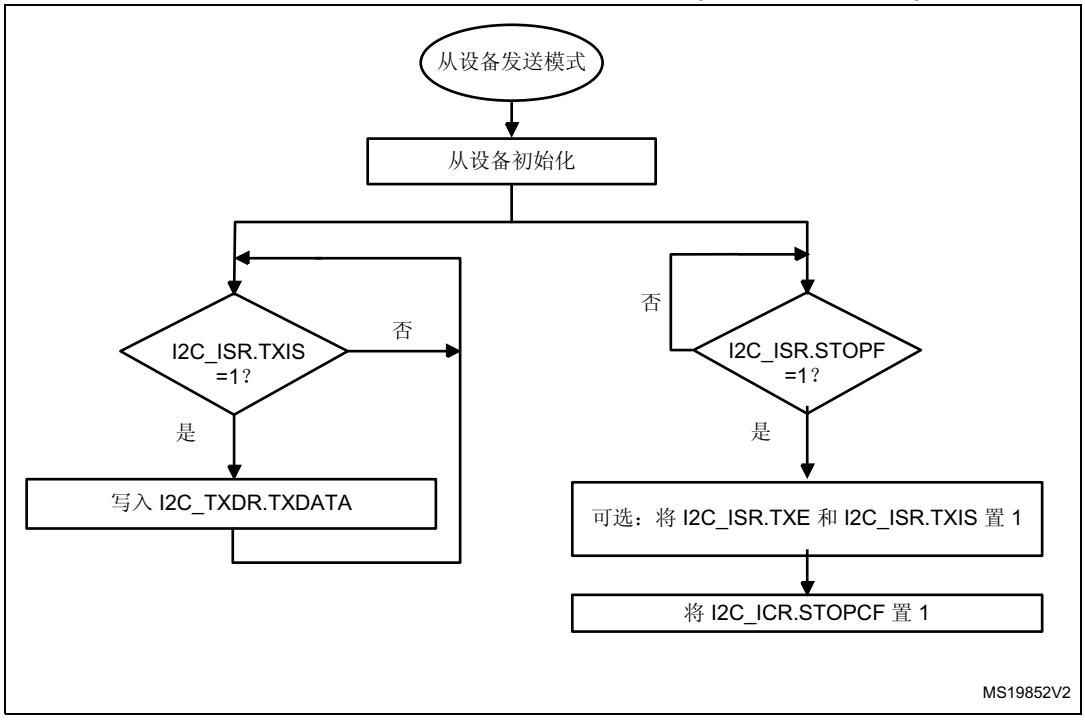
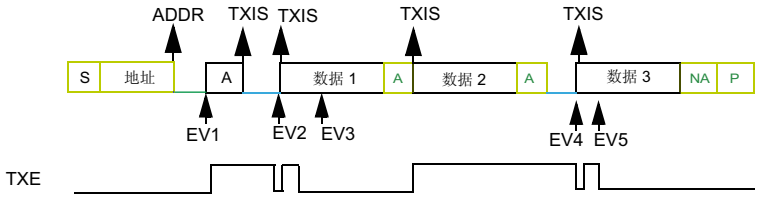


图 543. I2C 从设备发送器的传输总线图（仅限强制性事件）

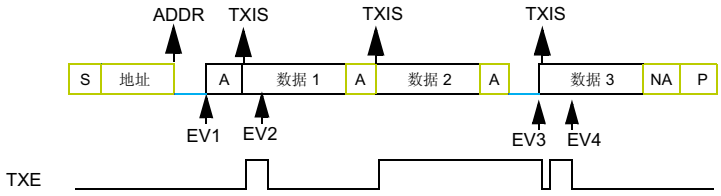
示例：I2C 从设备发送器发送 3 字节数据，第 1 个数据被刷新，NOSTRETCH（非拉伸）= 0：



图注：
发送
接收
SCL 延长

EV1：ADDR ISR：检查 ADDCODE 和 DIR，将 TXE 置 1，将 ADDRCF 置 1
EV2：TXIS ISR：写入数据 1
EV3：TXIS ISR：写入数据 2
EV4：TXIS ISR：写入数据 3
EV5：TXIS ISR：写入数据 4（不发送）

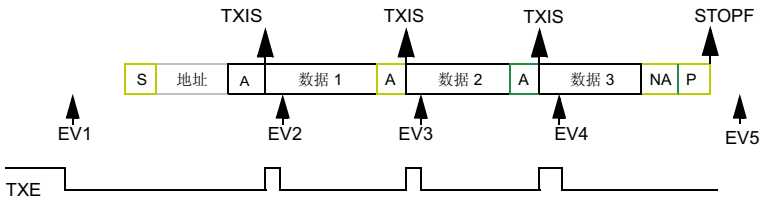
示例：I2C 从设备发送器发送 3 字节，不刷新第 1 个数据，NOSTRETCH=0：



图注：
发送
接收
SCL 延长

EV1：ADDR ISR：检查 ADDCODE 和 DIR，将 ADDRCF 置 1
EV2：TXIS ISR：写入数据 2
EV3：TXIS ISR：写入数据 3
EV4：TXIS ISR：写入数据 4（不发送）

示例：I2C 从设备发送器发送 3 字节，NOSTRETCH=1：



图注：
发送
接收
SCL 延长

Ev1：写入数据 1
EV2：TXIS ISR：写入数据 2
EV3：TXIS ISR：写入数据 3
EV4：TXIS ISR：写入数据 4（不发送）
EV5：STOPF ISR：（可选：将 TXE 和 TXIS 置 1），将 STOPCF 置 1

MS19853V2

从接收器

当 I2C_RXDR 满时，I2C_ISR 中的 RXNE 将置 1，如果 I2C_CR1 中的 RXIE 置 1，还将生成中断。读取 I2C_RXDR 时，将清零 RXNE。

接收到 STOP 条件且 I2C_CR1 寄存器中的 STOPIE 置 1 时，I2C_ISR 中的 STOPF 将置 1 并且会生成中断。

图 544. 从接收器的传输序列流程 (NOSTRETCH = 0)

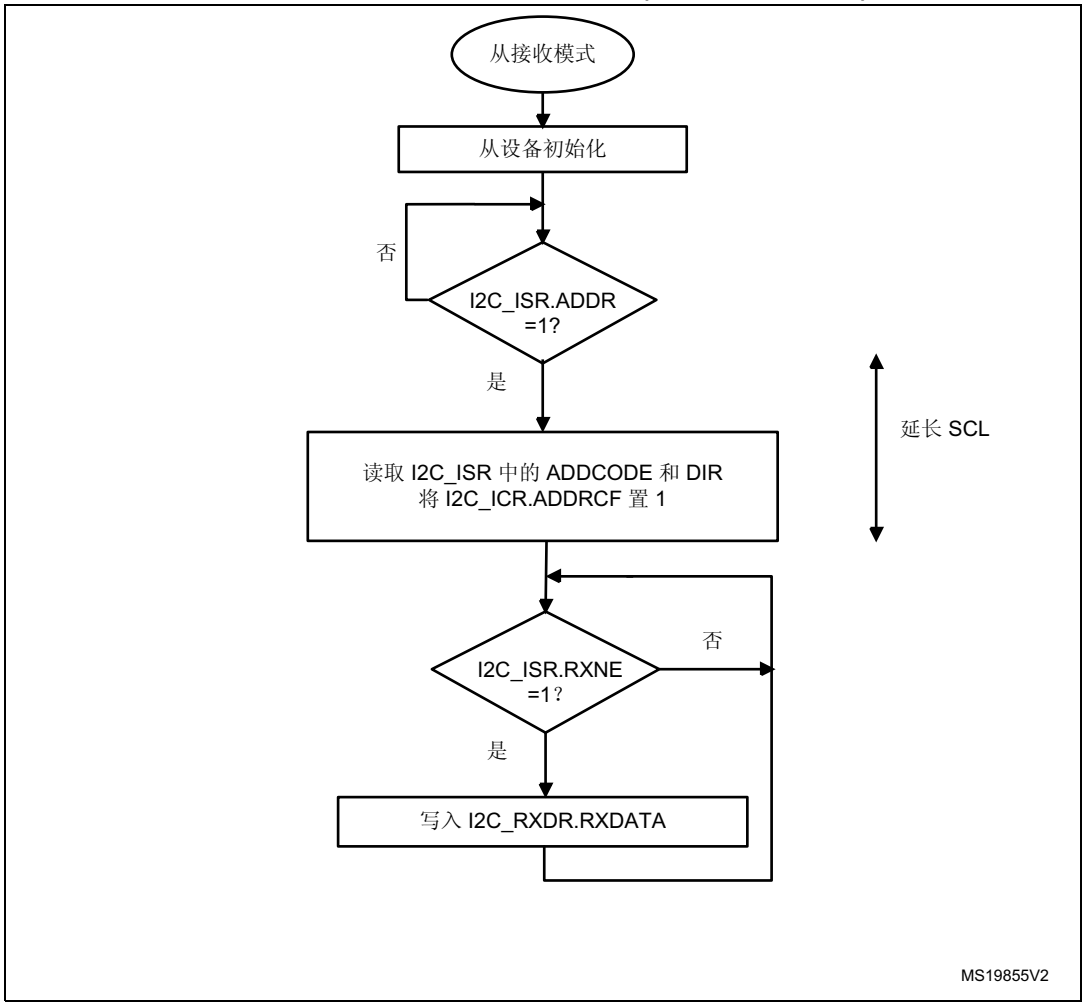


图 545. 从接收器的传输序列流程 (NOSTRETCH = 1)

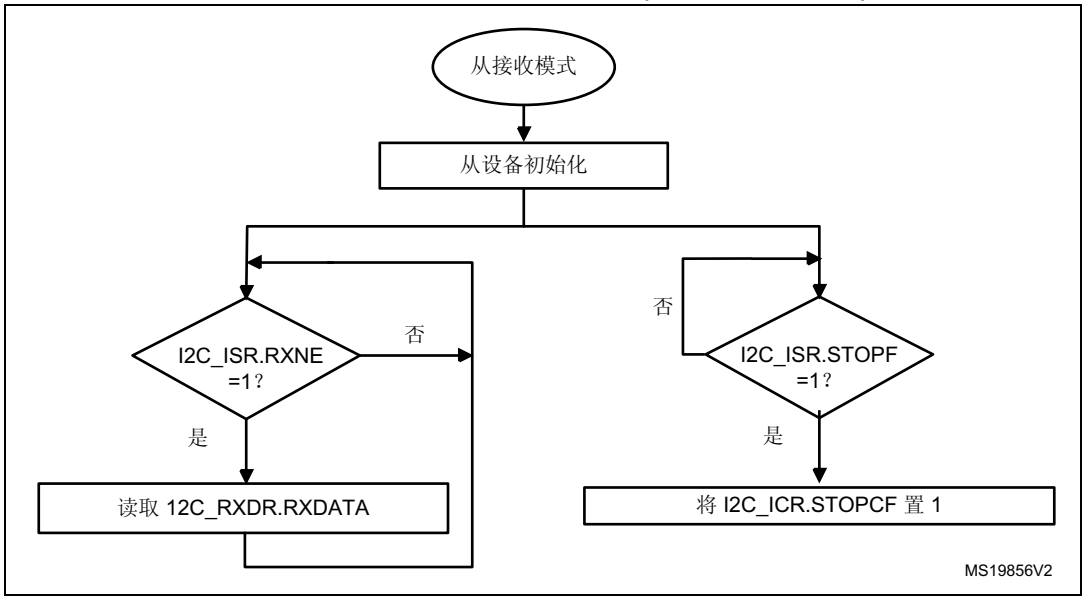
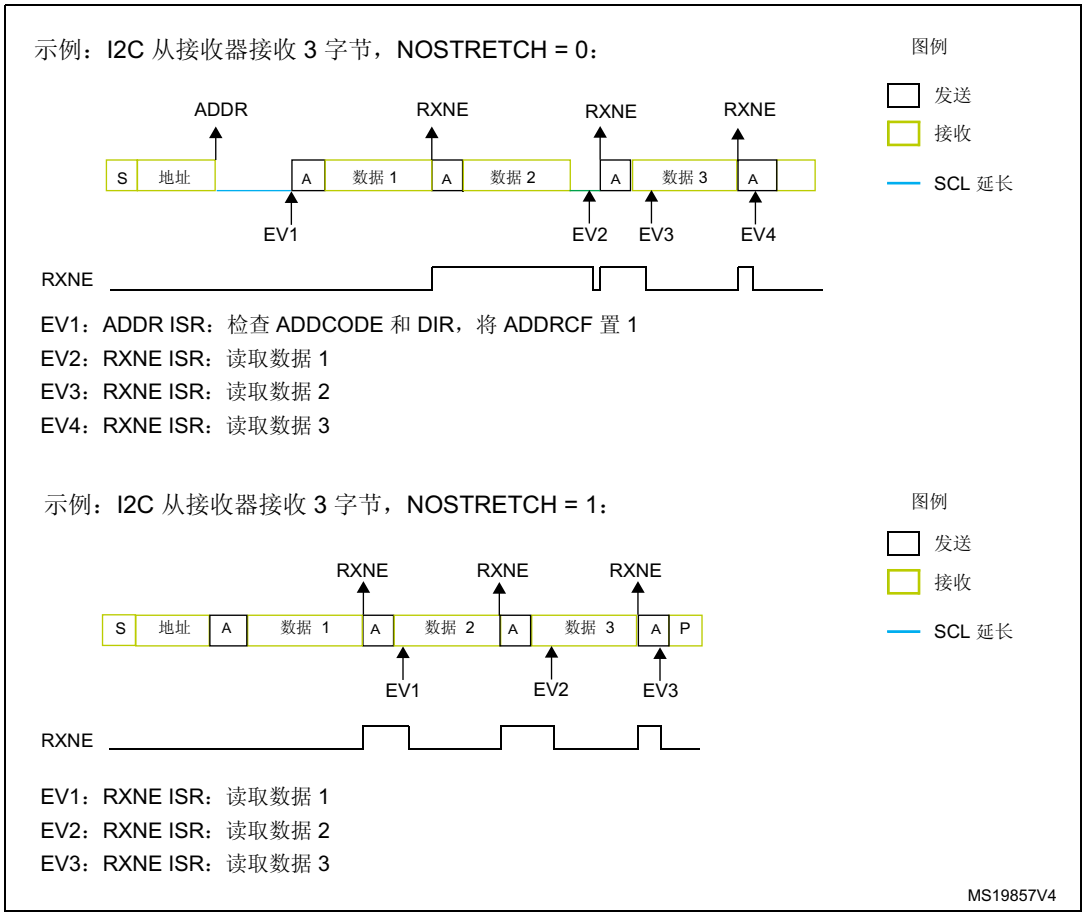


图 546. I2C 从接收器的传输总线图（仅限强制性事件）



48.4.9 I2C 主模式

I2C 主模式初始化

使能外设前，必须通过设置 I2C_TIMINGR 寄存器中的 SCLH 和 SCLL 位来配置 I2C 主时钟。

STM32CubeMX 工具计算 I2C_TIMINGR 内容并在 I2C 配置窗口中进行显示。

为了支持多主环境和从时钟延长，I2C 实现了时钟同步机制。

为了实现时钟同步，需执行以下操作：

- 使用 SCLL 计数器从 SCL 低电平内部检测开始对时钟的低电平进行计数。
- 使用 SCLH 计数器从 SCL 高电平内部检测开始对时钟的高电平进行计数。

I2C 经过 t_{SYNC1} 延时后检测其自身的 SCL 低电平，该延时取决于 SCL 下降沿、SCL 输入噪声滤波器（模拟和数字）以及 SCL 与 I2CxCLK 时钟的同步。一旦 SCLL 计数器达到 I2C_TIMINGR 寄存器的 SCLL[7:0] 位中编程的值，I2C 便会将 SCL 释放为高电平。

I2C 经过 t_{SYNC2} 延时后检测其自身的 SCL 高电平，该延时取决于 SCL 上升沿、SCL 输入噪声滤波器（模拟 + 数字）以及 SCL 与 I2CxCLK 时钟的同步。一旦 SCLH 计数器达到 I2C_TIMINGR 寄存器的 SCLH[7:0] 位中编程的值，I2C 便会使 SCL 变为低电平。

因此，主时钟周期为：

$$t_{SCL} = t_{SYNC1} + t_{SYNC2} + \{[(SCLH+ 1) + (SCLL+ 1)] \times (PRESC+ 1) \times t_{I2CCLK}\}$$

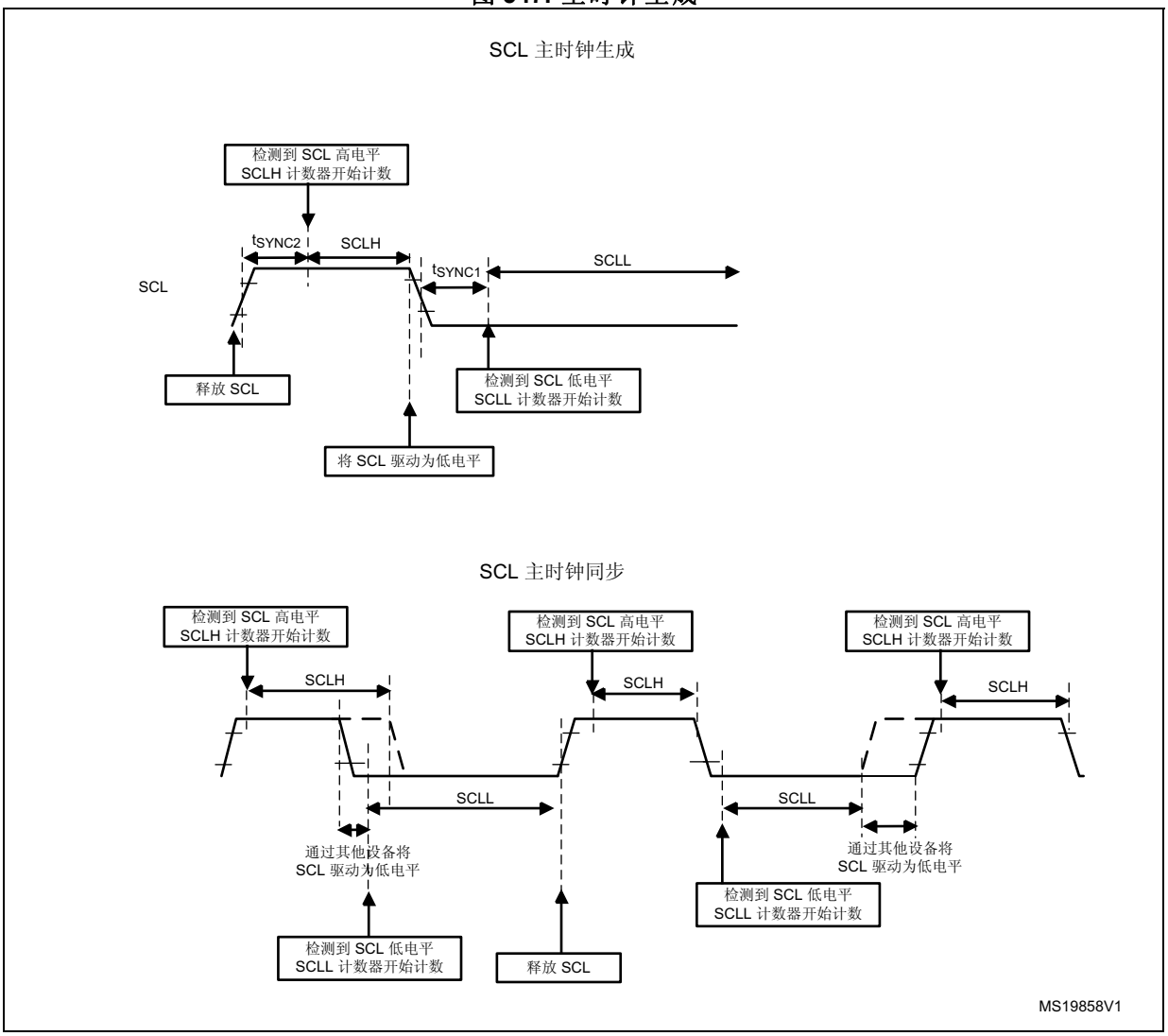
t_{SYNC1} 的持续时间取决于：

- SCL 下降斜率
- 模拟滤波器（使能时）引入的输入延时
- 数字滤波器（使能时）引入的输入延时：DNF x t_{I2CCLK}
- SCL 与 i2c_ker_ck 时钟同步而产生的延时（两到三个 i2c_ker_ck 周期）

t_{SYNC2} 的持续时间取决于：

- SCL 上升斜率
- 模拟滤波器（使能时）引入的输入延时
- 数字滤波器（使能时）引入的输入延时：DNF x t_{I2CCLK}
- SCL 与 i2c_ker_ck 时钟同步而产生的延时（两到三个 i2c_ker_ck 周期）

图 547. 主时钟生成


SCL 主时钟生成
检测到 SCL 高电平
SCLH 计数器开始计数
tSYNC2
SCLH
tSYNC1
SCLL
SCL
释放 SCL
检测到 SCL 低电平
SCLL 计数器开始计数
将 SCL 驱动为低电平
SCL 主时钟同步
检测到 SCL 高电平
SCLH 计数器开始计数
SCLH
SCLL
通过其他设备将
SCL 驱动为低电平
检测到 SCL 低电平
SCLL 计数器开始计数
释放 SCL
MS19858V1

小心： 为了符合 I²C 或 SMBus 规范，主时钟必须遵循下表中给出的时序。

表 425. I²C-SMBus 规范时钟时序

符号	参数	标准模式 (Sm)		快速模式 (Fm)		超快速模式 (Fm+)		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
f_{SCL}	SCL 时钟频率	-	100	-	400	-	1000	-	100	kHz
$t_{HD:STA}$	（重复）起始条件的保持时间	4.0	-	0.6	-	0.26	-	4.0	-	μs
$t_{SU:STA}$	重复起始条件的建立时间	4.7	-	0.6	-	0.26	-	4.7	-	
$t_{SU:STO}$	停止条件的建立时间	4.0	-	0.6	-	0.26	-	4.0	-	
t_{BUF}	停止条件和起始条件之间的总线空闲时间	4.7	-	1.3	-	0.5	-	4.7	-	
t_{LOW}	SCL 时钟的低电平周期	4.7	-	1.3	-	0.5	-	4.7	-	
t_{HIGH}	SCL 时钟的高电平周期	4.0	-	0.6	-	0.26	-	4.0	50	
t_r	SDA 和 SCL 信号的上升时间	-	1000	-	300	-	120	-	1000	ns
t_f	SDA 和 SCL 信号的下降时间	-	300	-	300	-	120	-	300	

注： *SCLL 和 SCLH 也分别用于生成 t_{BUF} / $t_{SU:STA}$ 和 $t_{HD:STA}$ / $t_{SU:STO}$ 时序。*

有关 I2C_TIMINGR 设置与 i2c_ker_ck 频率的示例，请参见*第 48.4.10 节*。

主模式通信初始化（地址阶段）

要发起通信，可在 I2C_CR2 寄存器中为寻址的从设备编程以下参数：

- 寻址模式（7 位或 10 位）：ADD10
- 待发送的从地址：SADD[9:0]
- 传输方向：RD_WRN
- 读取 10 位地址时：HEAD10R 位。必须对 HEAD10R 进行相应配置，以指示传输方向变化时必须发送完整的地址序列，还是只发送地址头。
- 待传输的字节数：NBYTES[7:0]。如果字节数等于或大于 255，则初始化时必须将 NBYTES[7:0] 填充为 0xFF。

然后，用户必须将 I2C_CR2 寄存器中的 START 位置 1。START 位置 1 时，不允许更改上述所有位。

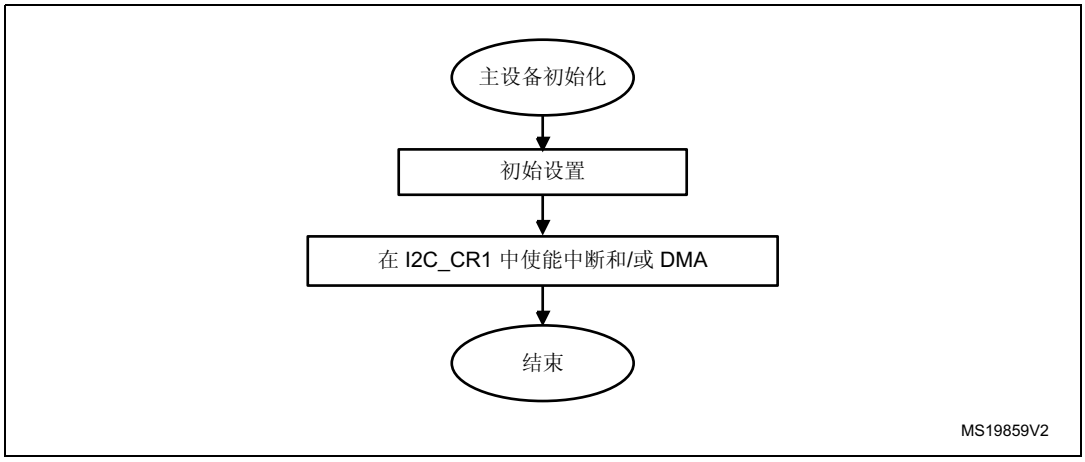
之后，当主设备检测到总线空闲 (BUSY = 0) 时，它会在经过 t_{BUF} 的延时后自动发送起始位，随后发出从设备地址。

仲裁丢失时，主设备将自动切换回从模式，如果从设备地址被选中，还将会自动发送 ACK 应答。

注： *无论接收到的应答值为何，只要已在总线上发送从地址，START 位便会由硬件复位。如果仲裁丢失，START 位也会由硬件复位。*
在 10 位寻址模式下，如果从设备不对从地址的前七个位进行应答，则主设备将自动重新启动从地址发送，直至接收到 ACK。在这种情况下，如果从从设备接收到 NACK，则必须将 ADDRCF 置 1，以停止发送从地址。
如果当 START 位置 1 时，I2C 作为从设备 (ADDR=1) 被寻址，则 I2C 将切换为从模式，START 位将在 ADDRCF 位置 1 时清零。

注： *该步骤同样适用于重复起始位。在这种情况下，BUSY = 1。*

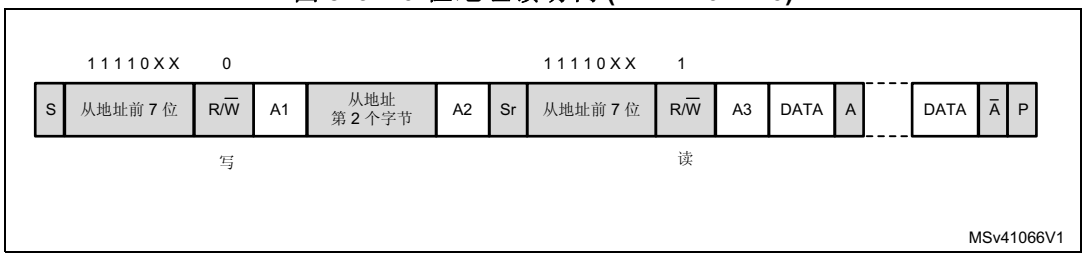
图 548. 主设备初始化流程



主接收器寻址 10 位地址从设备的初始化过程

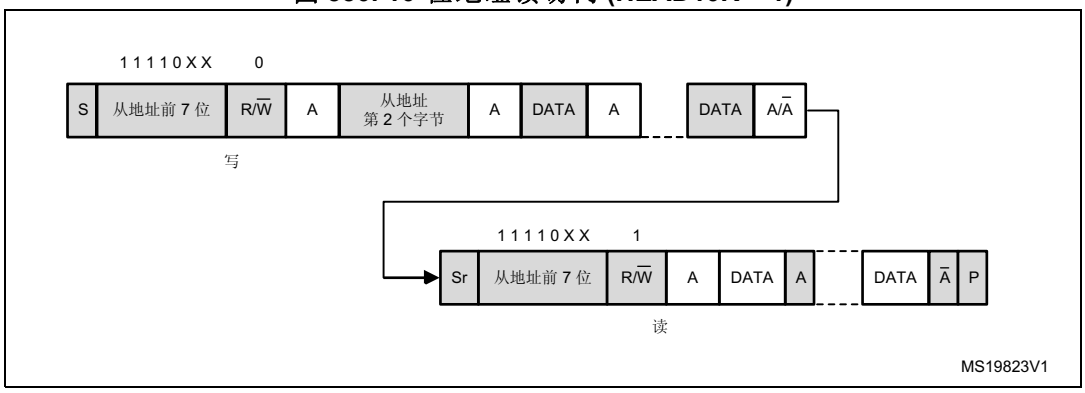
- 如果从地址采用 10 位格式，用户可选择将 I2C_CR2 寄存器中的 HEAD10R 位清零来发送完整的读序列。在这种情况下，主设备会在 START 位置 1 后自动发送以下完整序列：（重复）起始位 + 带写方向的从设备 10 位地址头字节 + 从设备地址第二个字节 +（重复）起始位 + 带读方向的从设备 10 位地址头字节

图 549. 10 位地址读访问 (HEAD10R = 0)



- 如果主设备对 10 位地址从设备进行寻址、向该从设备发送数据、然后再从该从设备读取数据，则必须首先完成主设备发送过程。然后，重复起始位置 1，10 位从地址配置为 HEAD10R = 1。在这种情况下，主设备发送以下序列：重复起始位 + 带读方向的从设备 10 位地址头字节。

图 550. 10 位地址读访问 (HEAD10R = 1)



主发送器

写传输时，在发送完每个字节（即第九个 SCL 脉冲（接收到 ACK 时））后，TXIS 标志将置 1。

如果 I2C_CR1 寄存器中的 TXIE 位置 1，TXIS 事件将生成中断。当 I2C_TXDR 寄存器中写入待发送的下一个数据字节时，该标志将被清零。

传输期间的 TXIS 事件的数量对应于 NBYTES[7:0] 中编程的值。如果待发送的数据字节总数大于 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES 数据传输完成时，TCR 标志将置 1，并且 SCL 线的低电平将被延长，直到 NBYTES[7:0] 被写入非零值。

收到 NACK 时，TXIS 标志不会置 1。

- 当 RELOAD = 0 且 NBYTES 数据传输完成时：
  - 在自动结束模式 (AUTOEND = 1) 下，将自动发送停止位。
  - 在软件结束模式 (AUTOEND = 0) 下，TC 标志将置 1 且 SCL 线的低电平将被延长，以便执行以下软件操作：

    可通过将 I2C_CR2 寄存器中的 START 位置 1 并配置适当的从地址和待传输字节数来请求发送重复起始位。将 START 位置 1 会将 TC 标志清零，并在总线上发送起始位。

    可通过将 I2C_CR2 寄存器中的 STOP 位置 1 来请求停止条件。将 STOP 位置 1 会将 TC 标志清零，并在总线上发送停止条件。
- 如果接收到 NACK：TXIS 标志不会置 1，并且接收到 NACK 后会自动发送停止条件。I2C_ISR 寄存器中的 NACKF 标志置 1，如果 NACKIE 位置 1，还将生成中断。

图 551. I2C 主发送器的传输序列流程（N ≤ 255 字节）

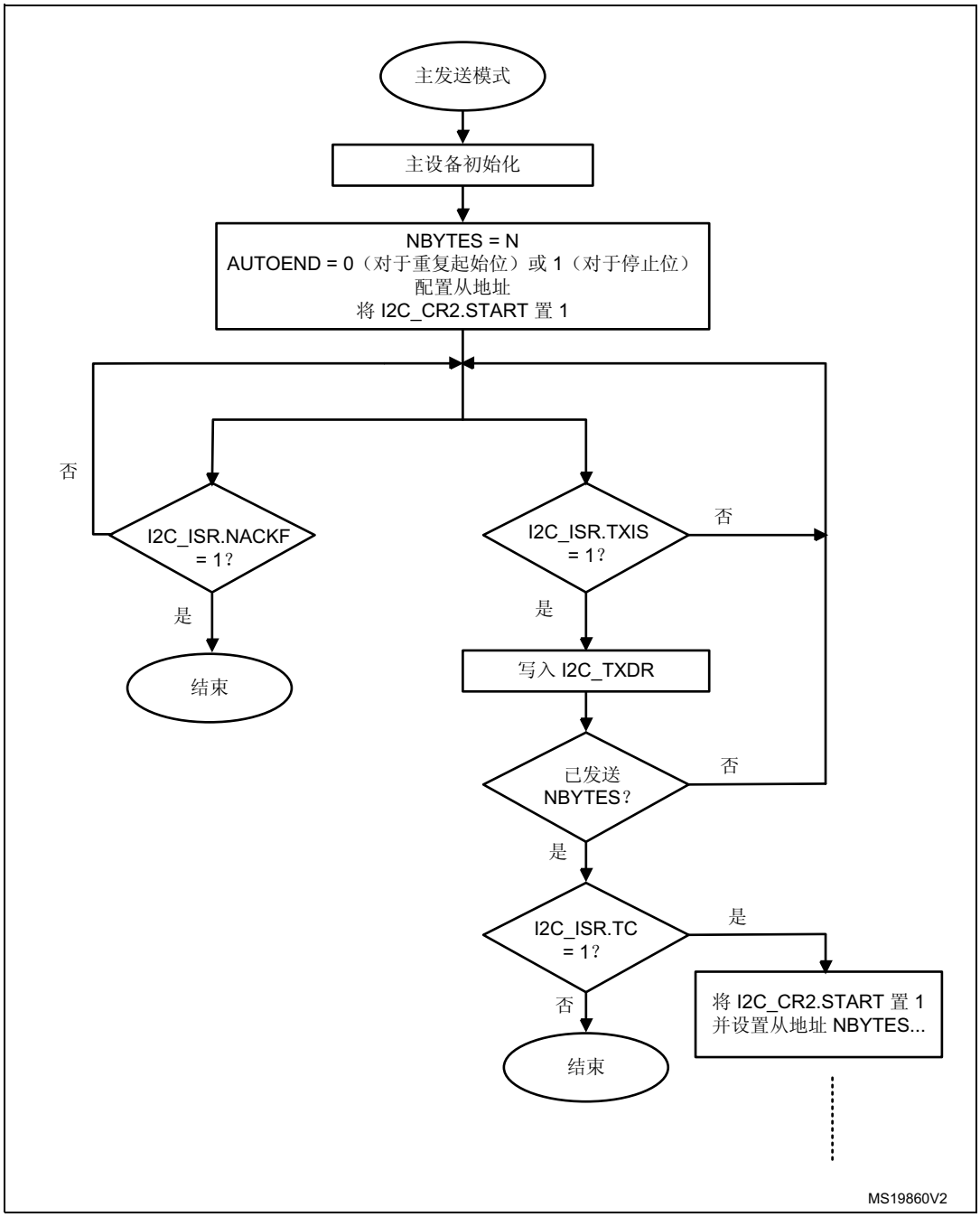


图 552. I2C 主发送器的传输序列流程（N > 255 字节）

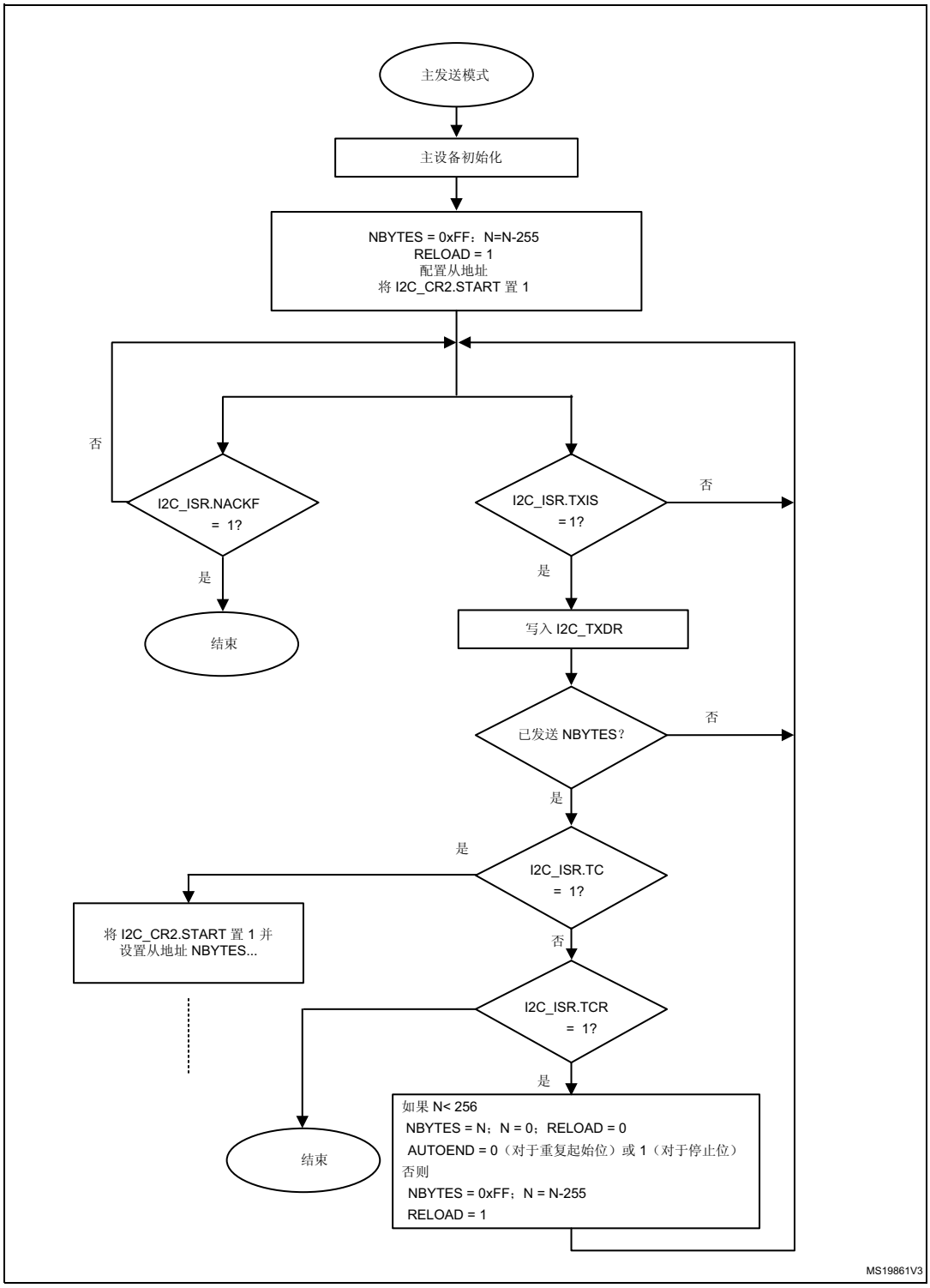
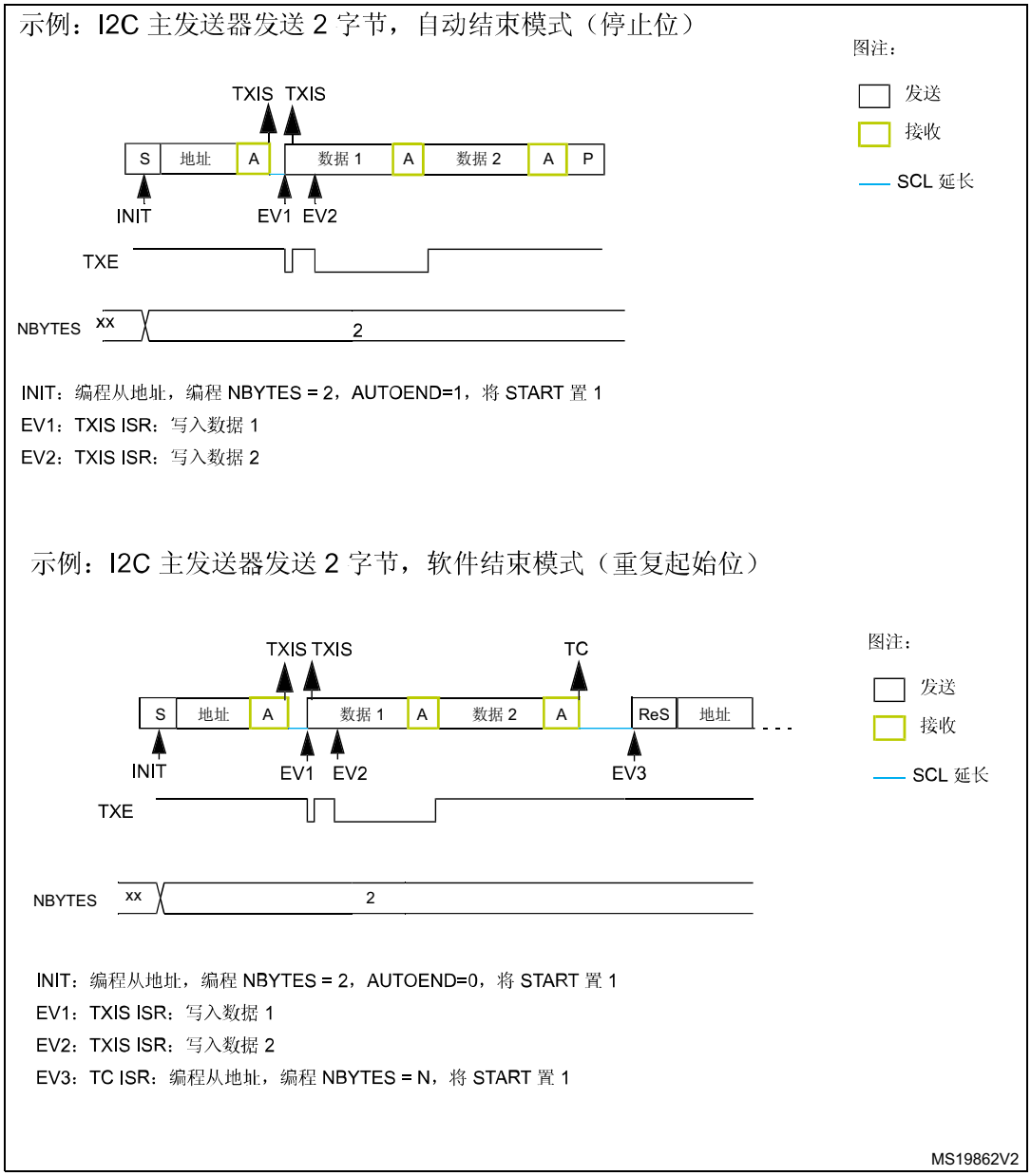


图 553. I2C 主发送器的传输总线图（仅限强制性事件）



主接收器

读传输时，在接收到每个字节（即第八个 SCL 脉冲）后，RXNE 标志将置 1。如果 I2C_CR1 寄存器中的 RXIE 位置 1，RXNE 事件将生成中断。读取 I2C_RXDR 时，将清零该标志。

如果待接收的数据字节总数大于 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES[7:0] 数据传输完成时，TCR 标志将置 1，并且 SCL 线的低电平将被延长，直到 NBYTES[7:0] 被写入非零值。

- 当 RELOAD = 0 且 NBYTES[7:0] 数据传输完成时：
  - 在自动结束模式 (AUTOEND = 1) 下，接收到最后一个字节后，将自动发送 NACK 和停止位。

– 在软件结束模式 (AUTOEND = 0) 下，接收到最后一个字节后，将自动发送 NACK，TC 标志将置 1 且 SCL 线的低电平将被延长，以便执行以下软件操作：

可通过将 I2C_CR2 寄存器中的 START 位置 1 并配置适当的从地址和待传输字节数来请求发送重复起始位。将 START 位置 1 会将 TC 标志清零，并在总线上发送起始位，后跟从地址。

可通过将 I2C_CR2 寄存器中的 STOP 位置 1 来请求停止条件。将 STOP 位置 1 会将 TC 标志清零，并在总线上发送停止条件。

图 554. I2C 主接收器的传输序列流程（N ≤ 255 字节）

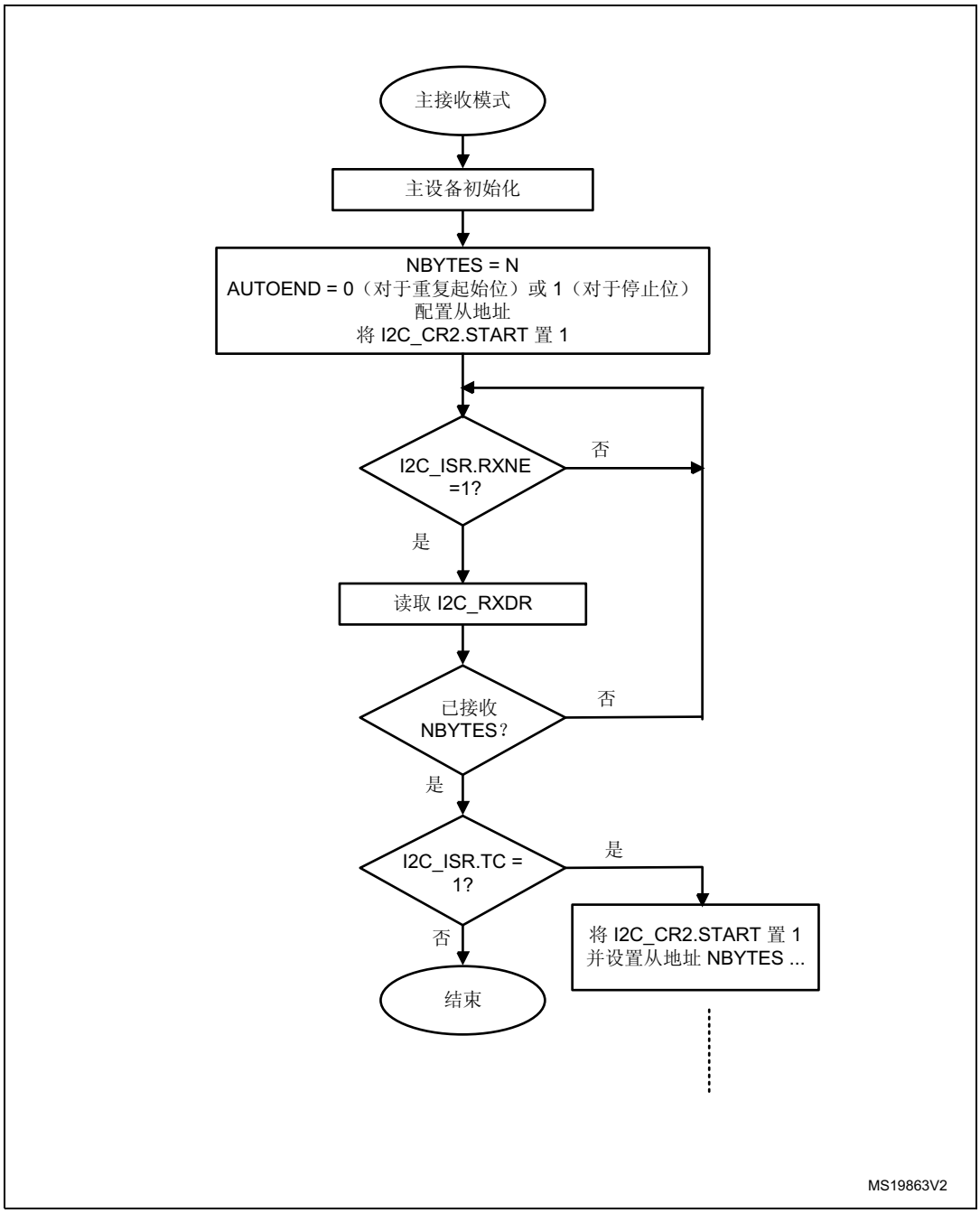


图 555. I2C 主接收器的传输序列流程（N > 255 字节）

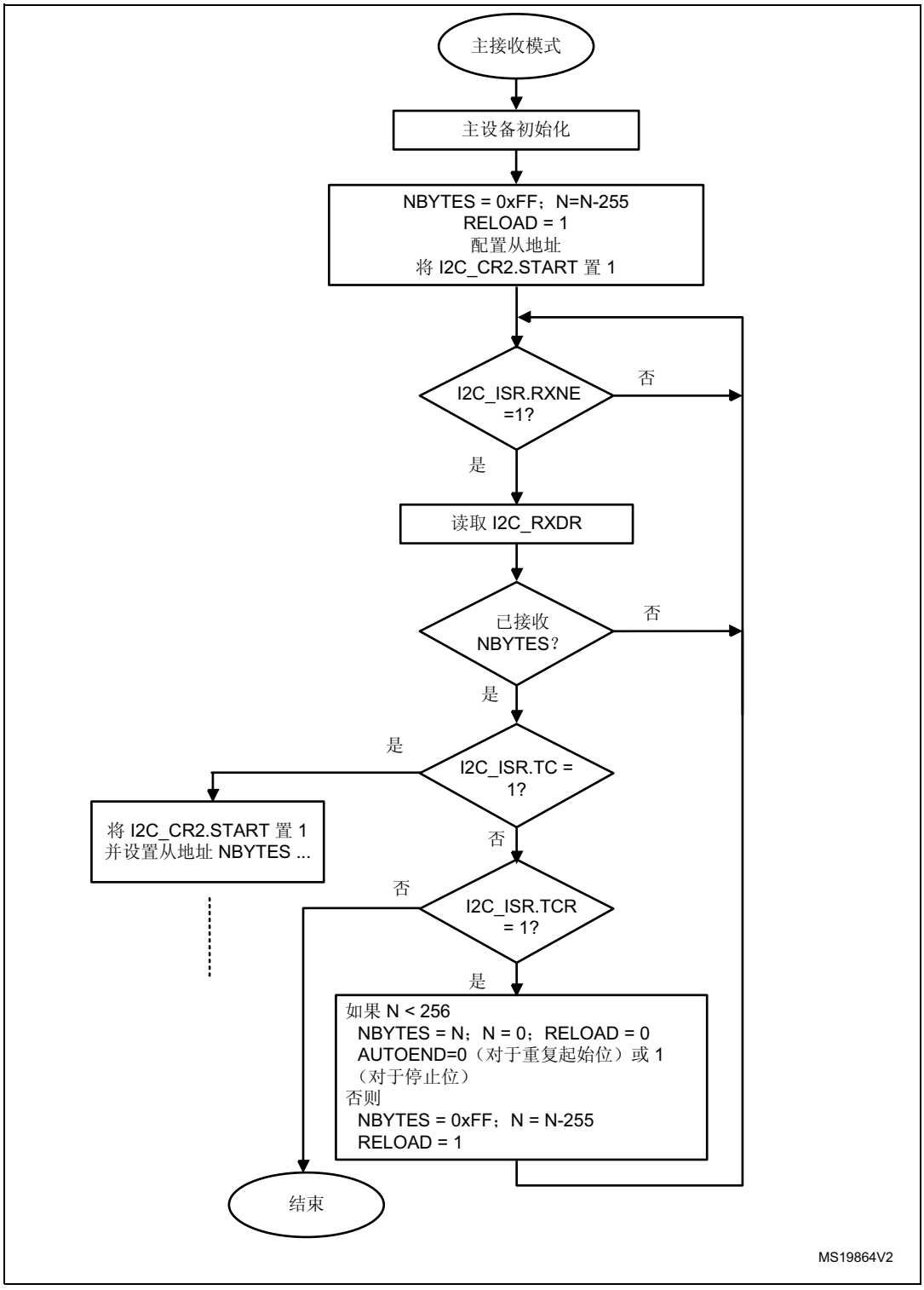
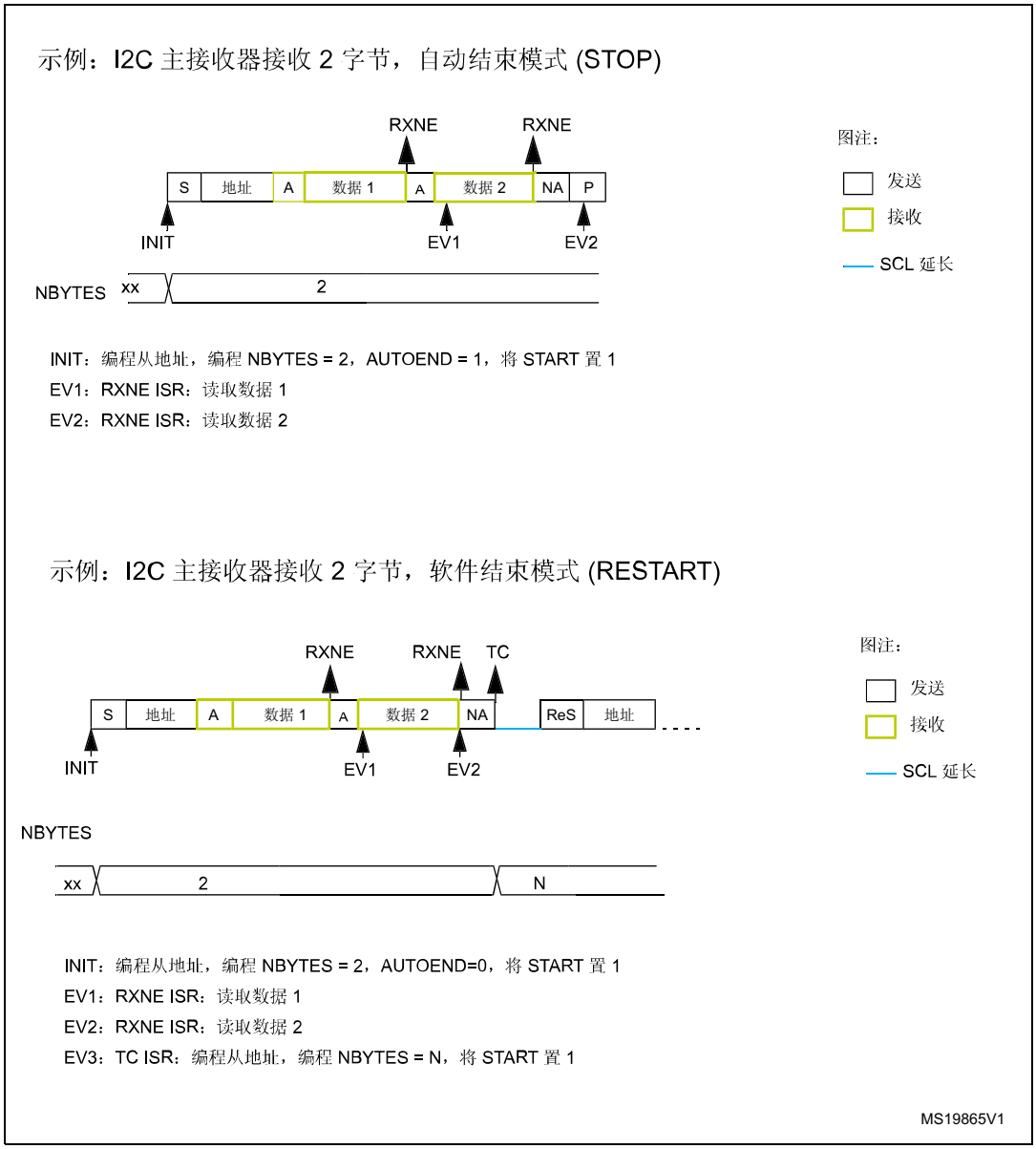


图 556. I2C 主接收器的传输总线图（仅限强制性事件）



48.4.10 I2C_TIMINGR 寄存器配置示例

下文各表提供了相应示例，以介绍如何编程 I2C_TIMINGR 才能获得符合 I^2C 规范的时序。要获取更准确的配置值，应使用 STM32CubeMX 工具（I2C 配置窗口）。

表 426. f_{I2CCLK} = 8 MHz 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	500 kHz
PRESC	0x1	0x1	0x0	0x0
SCLL	0xC7	0x13	0x9	0x6
t_{SCLL}	200 x 250 ns = 50 µs	20 x 250 ns = 5.0 µs	10 x 125 ns = 1250 ns	7 x 125 ns = 875 ns
SCLH	0xC3	0xF	0x3	0x3
t_{SCLH}	196 x 250 ns = 49 µs	16 x 250 ns = 4.0 µs	4 x 125 ns = 500 ns	4 x 125 ns = 500 ns
t_{SCL}(1)	约 100 µs(2)	约 10 µs(2)	约 2.5 µs(3)	约 2.0 µs(4)
SDADEL	0x2	0x2	0x1	0x0
t_{SDADEL}	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	1 x 125 ns = 125 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
t_{SCLDEL}	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	2 x 125 ns = 250 ns

1. 由于 SCL 内部检测存在延时，t_{SCL} 大于 $t_{SCLL} + t_{SCLH}$。为 t_{SCL} 提供的值仅用于举例说明。
2. $t_{SYNC1} + t_{SYNC2}$ 最小值为 4 x t_{I2CCLK} = 500 ns。$t_{SYNC1} + t_{SYNC2}$ = 1000 ns 时的示例。
3. $t_{SYNC1} + t_{SYNC2}$ 最小值为 4 x t_{I2CCLK} = 500 ns。$t_{SYNC1} + t_{SYNC2}$ = 750 ns 时的示例。
4. $t_{SYNC1} + t_{SYNC2}$ 最小值为 4 x t_{I2CCLK} = 500 ns。$t_{SYNC1} + t_{SYNC2}$ = 655 ns 时的示例。

表 427. f_{I2CCLK} = 16 MHz 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	0x3	0x3	0x1	0x0
SCLL	0xC7	0x13	0x9	0x4
t_{SCLL}	200 x 250 ns = 50 µs	20 x 250 ns = 5.0 µs	10 x 125 ns = 1250 ns	5 x 62.5 ns = 312.5 ns
SCLH	0xC3	0xF	0x3	0x2
t_{SCLH}	196 x 250 ns = 49 µs	16 x 250 ns = 4.0 µs	4 x 125 ns = 500 ns	3 x 62.5 ns = 187.5 ns
t_{SCL}(1)	约 100 µs(2)	约 10 µs(2)	约 2.5 µs(3)	约 1.0 µs(4)
SDADEL	0x2	0x2	0x2	0x0
t_{SDADEL}	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	2 x 125 ns = 250 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x2
t_{SCLDEL}	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	3 x 62.5 ns = 187.5 ns

1. 由于 SCL 内部检测存在延时，t_{SCL} 大于 $t_{SCLL} + t_{SCLH}$。为 t_{SCL} 提供的值仅用于举例说明。
2. $t_{SYNC1} + t_{SYNC2}$ 最小值为 4 x t_{I2CCLK} = 250 ns。$t_{SYNC1} + t_{SYNC2}$ = 1000 ns 时的示例。
3. $t_{SYNC1} + t_{SYNC2}$ 最小值为 4 x t_{I2CCLK} = 250 ns。$t_{SYNC1} + t_{SYNC2}$ = 750 ns 时的示例。
4. $t_{SYNC1} + t_{SYNC2}$ 最小值为 4 x t_{I2CCLK} = 250 ns。$t_{SYNC1} + t_{SYNC2}$ = 500 ns 时的示例。

表 428. f_{I2CCLK} = 48 MHz 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	0xB	0xB	0x5	0x5
SCLL	0xC7	0x13	0x9	0x3
t_{SCLL}	200 x 250 ns = 50 µs	20 x 250 ns = 5.0 µs	10 x 125 ns = 1250 ns	4 x 125 ns = 500 ns
SCLH	0xC3	0xF	0x3	0x1
t_{SCLH}	196 x 250 ns = 49 µs	16 x 250 ns = 4.0 µs	4 x 125 ns = 500 ns	2 x 125 ns = 250 ns
t_{SCL}(1)	约 100 µs(2)	约 10 µs(2)	约 2.5 µs(3)	约 875 ns(4)
SDADEL	0x2	0x2	0x3	0x0
t_{SDADEL}	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	3 x 125 ns = 375 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
t_{SCLDEL}	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	2 x 125 ns = 250 ns

1. 由于 SCL 内部检测存在延时，t_{SCL} 大于 t_{SCLL} + t_{SCLH}。为 t_{SCL} 提供的值仅用于举例说明。
2. t_{SYNC1} + t_{SYNC2} 最小值为 4x t_{I2CCLK} = 83.3 ns。t_{SYNC1} + t_{SYNC2} = 1000 ns 时的示例
3. t_{SYNC1} + t_{SYNC2} 最小值为 4x t_{I2CCLK} = 83.3 ns。t_{SYNC1} + t_{SYNC2} = 750 ns 时的示例
4. t_{SYNC1} + t_{SYNC2} 最小值为 4x t_{I2CCLK} = 83.3 ns。t_{SYNC1} + t_{SYNC2} = 250 ns 时的示例

48.4.11 SMBus 特性

仅当支持 SMBus 功能（请参见*第 48.3 节*）时，才涉及本节内容。

简介

系统管理总线 (SMBus) 是一个双线制接口，各设备可通过它在彼此之间或者与系统的其余部分之间进行通信。它以 I^2C 的工作原理为基础。SMBus 可针对系统和电源管理相关的任务提供控制总线。

该外设与 SMBus 规范 (http://smbus.org) 兼容。

系统管理总线规范涉及三类设备。

- 从设备，用于接收或响应命令。
- 主设备，用于发出命令、生成时钟和中止传输。
- 主机，专用的主设备，可提供连接系统 CPU 的主接口。主机必须具有主 - 从设备功能，并且必须支持 SMBus 主机通知协议。系统中只允许存在一个主机。

该外设可配置为主设备或从设备，也可配置为主机。

总线协议

任何给定设备都有十一种可用命令协议。设备既可以在这些协议中任选其一，也可以使用全部协议进行通信。这些协议分别为快速命令、发送字节、接收字节、写入字节、写入字、读取字节、读取字、过程调用、块读取、块写入以及块写入-块读取过程调用。这些协议必须通过用户软件实现。

有关这些协议的详细信息，请参见 SMBus 规范 (http://smbus.org)。

地址解析协议 (ARP)

通过为各个从设备动态分配一个新的唯一地址可解决 SMBus 从地址冲突的问题。为了提供一种机制来针对地址分配隔离各个设备，各设备必须具有唯一的设备标识符 (UDID)。该 128 位数字由软件实现。

该外设支持地址解析协议 (ARP)。通过将 I2C_CR1 寄存器中的 SMBDEN 位置 1 来使能 SMBus 设备默认地址 (0b1100 001)。ARP 命令必须通过用户软件实现。

此外，还将在从模式下执行仲裁以支持 ARP。

有关 SMBus 地址解析协议的详细信息，请参见 SMBus 规范 (http://smbus.org)。

接收的命令和数据应答控制

SMBus 接收器必须能够对接收到的每个命令或数据进行 NACK 应答。要在从模式下实现 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从字节控制模式。更多详细信息，请参见*从设备字节控制模式*。

主机通知协议

该外设通过将 I2C_CR1 寄存器中的 SMBHEN 位置 1 来支持主机通知协议。在这种情况下，主机将应答 SMBus 主机地址 (0b0001 000)。

使用该协议时，该设备用作主设备，而主机用作从设备。

SMBus 报警

设备支持 SMBus ALERT 可选信号。只具备从功能的设备可通过 SMBALERT# 引脚向主机发出信号，指示它想要通信。主机会处理该中断并通过报警响应地址 (0b0001 100) 同时访问所有 SMBALERT# 设备。只有那些将 SMBALERT# 拉到低电平的设备会应答报警响应地址。

如果配置为从设备 (SMBHEN = 0)，则通过将 I2C_CR1 寄存器中的 ALERTEN 位置 1 来将 SMBA 引脚拉为低电平。这同时还会使能报警响应地址。

如果配置为主机 (SMBHEN = 1)，则当 SMBA 引脚上检测到下降沿且 ALERTEN = 1 时，I2C_ISR 寄存器中的 ALERT 标志置 1。如果 I2C_CR1 寄存器中的 ERRIE 位置 1，将生成中断。当 ALERTEN = 0 时，即使外部 SMBA 引脚为低电平，ALERT 线也将被视为高电平。

如果无需 SMBus ALERT 引脚，则当 ALERTEN = 0 时，SMBA 引脚可用作标准 GPIO。

数据包错误校验

SMBus 规范中引入了数据包错误校验机制来提高可靠性和通信稳定性。数据包错误校验的实施方式是在每次消息传输结束时附加数据包错误代码 (PEC)。PEC 的计算方式是对所有消息字节（包括地址和读/写位）使用 CRC-8 多项式 $C(x) = x^8 + x^2 + x + 1$。

外设内置了硬件 PEC 计算器，可在接收到的字节与硬件计算的 PEC 不匹配时自动发送拒绝应答信号。

超时

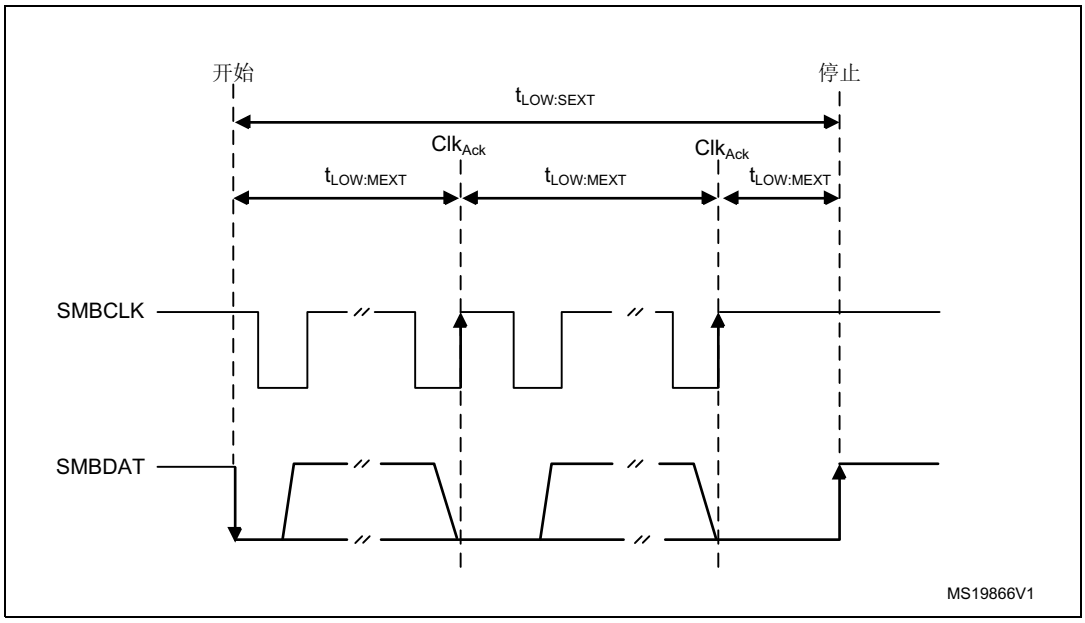
该外设内置了硬件定时器，以符合系统管理总线 (SMBus) 规范中定义的三个超时要求。

表 429. SMBus 超时规范

符号	参数	限值		单位
		最小值	最大值	
$t_{TIMEOUT}$	检测时钟低电平超时	25	35	ms
$t_{LOW:SEXT}$(1)	累积时钟低电平延长时间（从设备）	-	25	
$t_{LOW:MEXT}$(2)	累积时钟低电平延长时间（主设备）	-	10	

1. $t_{LOW:SEXT}$ 是一段累积时间，即给定从设备在一条消息的最初起始到停止期间时钟信号可延长的时间。其他从设备或主设备也可能延长时钟，进而导致时钟低电平总延长时间超过 $t_{LOW:SEXT}$。因此，测量该参数时该全速主设备只寻址一个从设备。
2. $t_{LOW:MEXT}$ 是一段累积时间，即主设备在消息的每个字节（定义为 START 到 ACK、ACK 到 ACK 或 ACK 到 STOP）内时钟信号可延长的时间。从设备或其他主设备也可能延长时钟，进而导致时钟低电平总时间超过 $t_{LOW:MEXT}$（针对给定字节）。因此，测量该参数时该全速主设备只寻址一个从设备。

图 557. $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的超时间隔



总线空闲检测

如果主设备检测到时钟和数据信号的高电平时间 $t_{IDLE} > t_{HIGH,MAX}$，则认为总线空闲（请参见 *I2C 时序*）。

该时序参数已考虑如下情况：主设备已动态添加至总线，但可能尚未检测到 SMBCLK 或 SMBDAT 线上的状态转换。在这种情况下，主设备必须等待足够长的时间，以确定当前未进行传输。外设支持硬件总线空闲检测。

48.4.12 SMBus 初始化

仅当支持 SMBus 功能（请参见*第 48.3 节*）时，才涉及本节内容。

除了 I2C 初始化之外，还必须进行一些其他的特定初始化，以便执行 SMBus 通信。

接收的命令和数据应答控制（从模式）

SMBus 接收器必须能够对接收到的每个命令或数据进行 NACK 应答。要在从模式下实现 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从字节控制模式。更多详细信息，请参见*从设备字节控制模式*。

特定地址（从模式）

必要时必须使能特定的 SMBus 地址。更多详细信息，请参见*总线空闲检测*。

- 通过将 I2C_CR1 寄存器中的 SMBDEN 位置 1 来使能 SMBus 设备默认地址 (0b1100 001)。
- 通过将 I2C_CR1 寄存器中的 SMBHEN 位置 1 来使能 SMBus 主机地址 (0b0001 000)。
- 通过将 I2C_CR1 寄存器中的 ALERTEN 位置 1 来使能报警响应地址 (0b0001100)。

数据包错误校验

通过将 I2C_CR1 寄存器中的 PECEN 位置 1 来使能 PEC 的计算。然后，借助硬件字节计数器（I2C_CR2 寄存器中的 NBYTES[7:0]）来管理 PEC 传输。使能 I2C 之前，必须配置 PECEN 位。

PEC 传输由硬件字节计数器来管理，因此在从模式下连接 SMBus 时必须将 SBC 位置 1。当 PECBYTE 位置 1 且 RELOAD 位清零时，传输完 NBYTES - 1 字节的数据后会传输 PEC。如果 RELOAD 置 1，PECBYTE 将不起作用。

小心： 使能 I2C 时，不允许更改 PECEN 配置。

表 430. 带 PEC 的 SMBus 配置

模式	SBC 位	RELOAD 位	AUTOEND 位	PECBYTE 位
主 Tx/Rx NBYTES + PEC+ STOP	x	0	1	1
主 Tx/Rx NBYTES + PEC + ReSTART	x	0	0	1
从 Tx/Rx + PEC	1	0	x	1

超时检测

将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 和 TEXTEN 位置 1 来使能超时检测。定时器必须按如下方式编程：即在 SMBus 规范规定的时间最大值之前检测出超时情况。

- $t_{TIMEOUT}$ 检查

  要使能 $t_{TIMEOUT}$ 检查，必须将 12 位 TIMEOUTA[11:0] 位编程为定时器重载值，以检查 $t_{TIMEOUT}$ 参数。必须将 TIDLE 位配置为 0，以检测 SCL 低电平超时。

  然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。

  如果 SCL 的低电平持续时间超过 (TIMEOUTA + 1) x 2048 x t_{I2CCLK}，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。

  请参见 *表 431*。

小心： TIMEOUTEN 位置 1 时，不允许更改 TIMEOUTA[11:0] 位和 TIDLE 位的配置。

- $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 检查

  必须根据外设配置为主设备还是从设备来配置 12 位 TIMEOUTB 定时器，以便为从设备校验 $t_{LOW:SEXT}$，为主设备校验 $t_{LOW:MEXT}$。由于标准只规定了最大值，因此用户可以为这两个参数选择相同的值。然后，通过将 I2C_TIMEOUTR 寄存器中的 TEXTEN 位置 1 来使能定时器。

  如果 SMBus 外设延长 SCL 的累积时间超过 (TIMEOUTB + 1) x 2048 x t_{I2CCLK}，并且达到*总线空闲检测*一节给出的超时间隔，则 I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。

  请参见 *表 432*

小心： TEXTEN 位置 1 时，不允许更改 TIMEOUTB 配置。

总线空闲检测

要使能 t_{IDLE} 检查，必须将 12 位 TIMEOUTA[11:0] 位字段编程为定时器重载值，以获取 t_{IDLE} 参数。必须将 TIDLE 位配置为“1”，以检测 SCL 和 SDA 高电平超时。然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。

如果 SCL 和 SDA 线的高电平持续时间超过 (TIMEOUTA + 1) x 4 x t_{I2CCLK}，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。

请参见 *表 433*。

小心： TIMEOUTEN 置 1 时，不允许更改 TIMEOUTA 和 TIDLE 配置。

48.4.13 SMBus：I2C_TIMEOUTR 寄存器配置示例

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 48.3 节*。

- 将 $t_{TIMEOUT}$ 的最大持续时间配置为 25 ms：

表 431. TIMEOUTA 设置示例（最大 $t_{TIMEOUT}$ = 25 ms）

f_{I2CCLK}	TIMEOUTA[11:0] 位	TIDLE 位	TIMEOUTEN 位	$t_{TIMEOUT}$
8 MHz	0x61	0	1	98 x 2048 x 125 ns = 25 ms
16 MHz	0xC3	0	1	196 x 2048 x 62.5 ns = 25 ms
48 MHz	0x249	0	1	586 x 2048 x 20.08 ns = 25 ms

- 将 $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的最大持续时间配置为 8 ms：

表 432. TIMEOUTB 设置示例

f_{I2CCLK}	TIMEOUTB[11:0] 位	TEXTEN 位	$t_{LOW:EXT}$
8 MHz	0x1F	1	32 x 2048 x 125 ns = 8 ms
16 MHz	0x3F	1	64 x 2048 x 62.5 ns = 8 ms
48 MHz	0xBB	1	188 x 2048 x 20.08 ns = 8 ms

- 将 t_{IDLE} 的最大持续时间配置为 50 µs

表 433. TIMEOUTA 设置示例（最大 t_{IDLE} = 50 µs）

f_{I2CCLK}	TIMEOUTA[11:0] 位	TIDLE 位	TIMEOUTEN 位	t_{TIDLE}
8 MHz	0x63	1	1	100 x 4 x 125 ns = 50 µs
16 MHz	0xC7	1	1	200 x 4 x 62.5 ns = 50 µs
48 MHz	0x257	1	1	600 x 4 x 20.08 ns = 50 µs

48.4.14 SMBus 从模式

仅当支持 SMBus 功能（请参见*第 48.3 节*）时，才涉及本节内容。

除了 I2C 从模式传输管理（请参见*第 48.4.8 节*）之外，还提供了额外的软件流程来支持 SMBus。

SMBus 从发送器

在 SMBus 模式下作为从发送器时，必须将 SBC 编程为 1，以使能在完成已编程数据字节数的传输后进行 PEC 传输。当 PECBYTE 位置 1 时，NBYTES[7:0] 中编程的字节数包含 PEC 传发送。在这种情况下，总 TXIS 中断数为 NBYTES - 1，如果主设备在完成 NBYTES - 1 字节的数据传输后请求传输额外的字节，则将自动发送 I2C_PECR 寄存器的内容。

小心： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 558. SMBus 从发送器的传输序列流程（N 字节 + PEC）

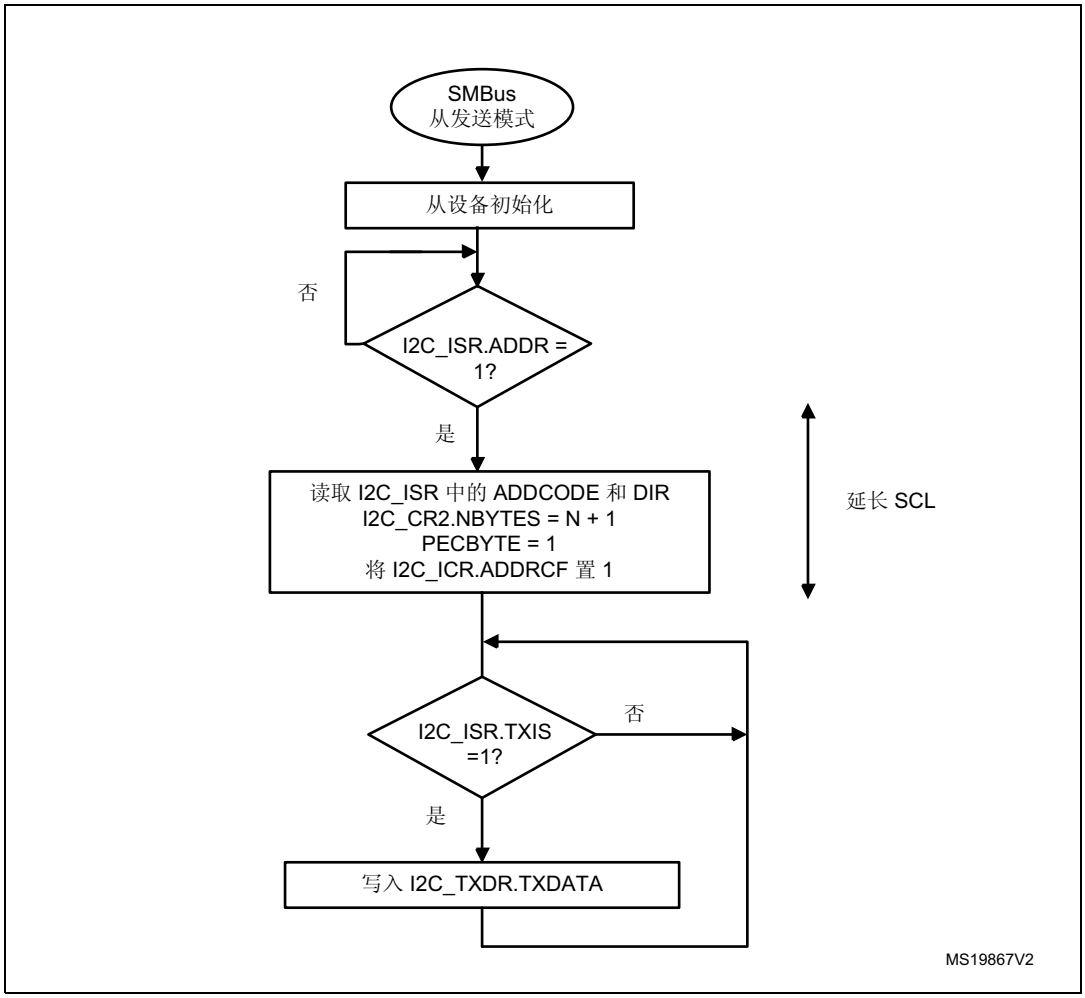
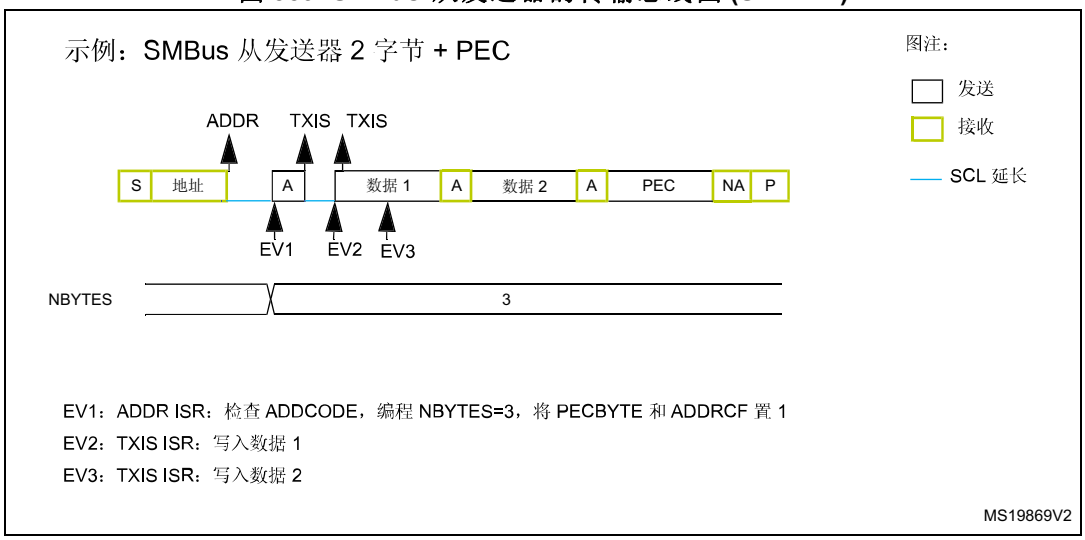


图 559. SMBus 从发送器的传输总线图 (SBC = 1)



SMBus 从接收器

在 SMBus 模式下使用 I2C 时，必须将 SBC 编程为 1，以便在完成已编程数据字节数的传输后进行 PEC 校验。要对每个字节进行 ACK 控制，必须选择重载模式 (RELOAD = 1)。更多详细信息，请参见*从设备字节控制模式*。

要校验 PEC 字节，必须将 RELOAD 位清零并将 PECBYTE 位置 1。在这种情况下，当接收到 NBYTES - 1 字节的数据后，接收的下一个字节将与内部 I2C_PECR 寄存器的内容作比较。如果比较不匹配，则将自动生成 NACK 信号；如果比较匹配，则将自动生成 ACK 信号，而与 ACK 位的值无关。PEC 字节一经接收，便会像任何其它数据一样复制到 I2C_RXDR 寄存器中，并且 RXNE 标志将置 1。

当 PEC 不匹配时，PECERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

如果无需 ACK 软件控制，用户可编程 PECBYTE = 1，在同一写操作下，将 NBYTES 编程为连续接收的字节数。接收到 NBYTES - 1 字节的数据后，会将接收的下一个字节视为 PEC 进行校验。

小心： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 560. SMBus 从接收器的传输序列流程（N 字节 + PEC）

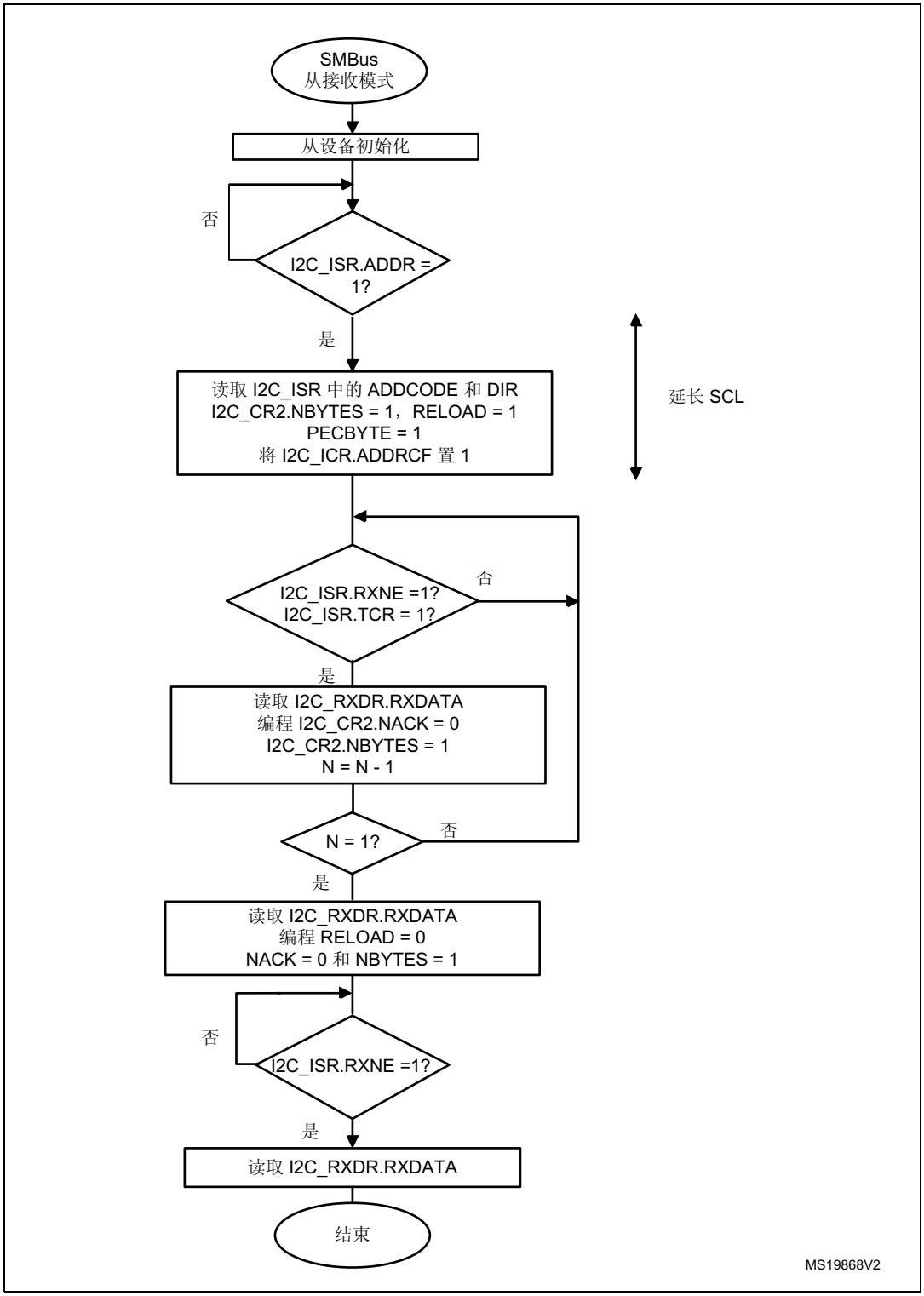
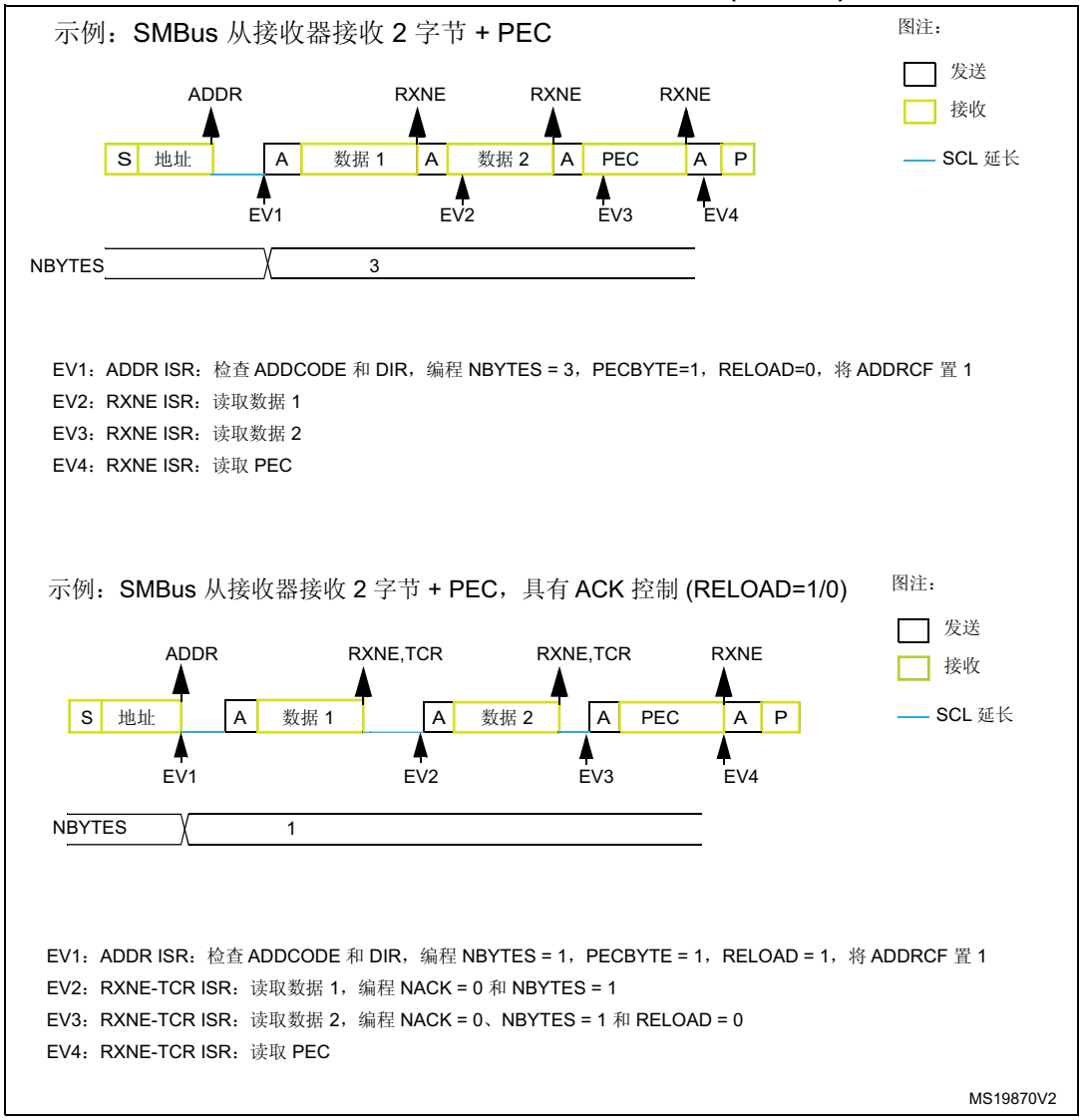


图 561. SMBus 从接收器的总线传输图 (SBC = 1)



仅当支持 SMBus 功能（请参见*第 48.3 节*）时，才涉及本节内容。

除了 I2C 主模式传输管理（请参见*第 48.4.9 节*）之外，还提供了额外的软件流程来支持 SMBus。

SMBus 主发送器

当 SMBus 主设备想要发送 PEC 时，必须在 START 位置 1 前，将 PECBYTE 位置 1 并在 NBYTES[7:0] 位字段中设置字节数。在这种情况下，总 TXIS 中断数为 NBYTES - 1。因此，如果 PECBYTE 位在 NBYTES = 0x1 时置 1，则将自动发送 I2C_PECR 寄存器的内容。

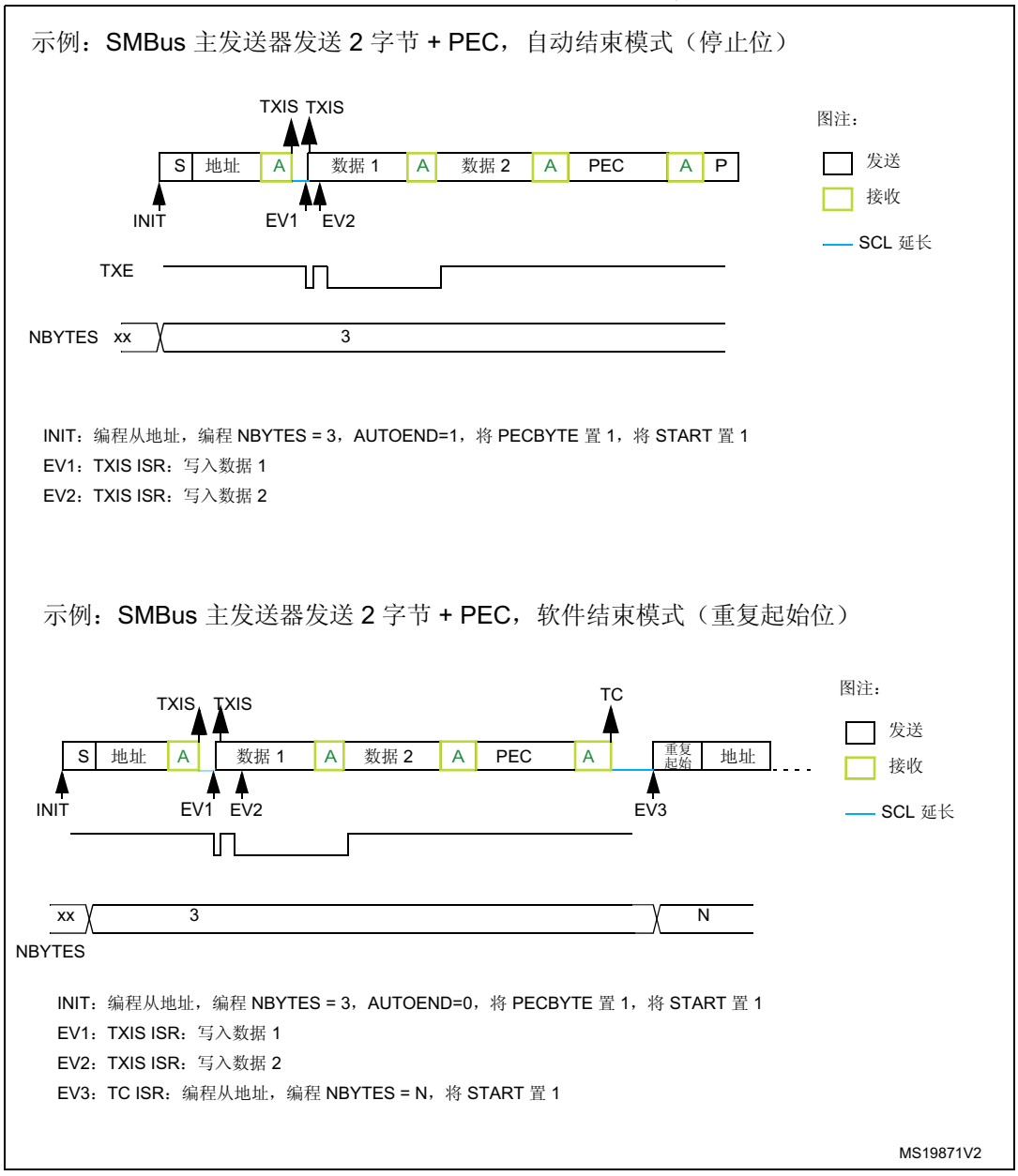
如果 SMBus 主设备想要在 PEC 后发送停止位，则必须选择自动结束模式 (AUTOEND = 1)。在这种情况下，传输 PEC 后将自动发送停止位。

如果 SMBus 主设备想要在 PEC 后发送重复起始位，则必须选择软件模式 (AUTOEND = 0)。在这种情况下，发送 NBYTES - 1 字节的数据后，将发送 I2C_PECR 寄存器的内容，

TC 标志将在传输完 PEC 之后置 1，SCL 线的低电平时间将延长。必须在 TC 中断子程序中设置重复起始位。

小心： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 562. SMBus 主发送器的总线传输图



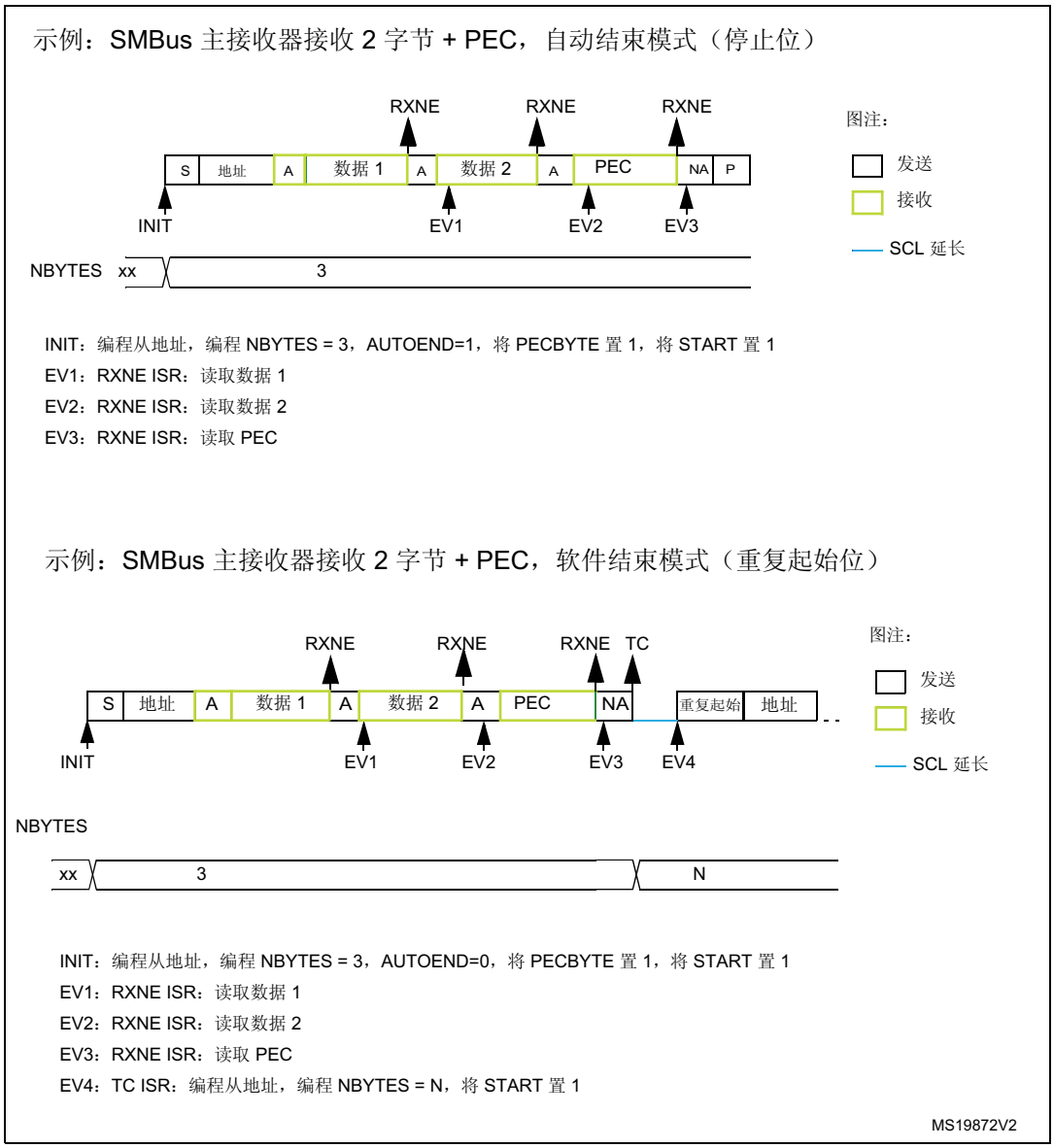
SMBus 主接收器

当 SMBus 主设备想要接收 PEC，并在传输结束后接收 STOP 时，可选择自动结束模式 (AUTOEND=1)。将 START 位置 1 之前，必须将 PECBYTE 位置 1 并设置从地址。在这种情况下，当接收到 NBYTES - 1 字节的数据后，将自动使用 I2C_PECR 寄存器的内容对接收的下一个字节进行校验。PEC 字节（其后跟有停止位）将得到 NACK 响应。

当 SMBus 主接收器想要接收 PEC 字节，并且在传输结束后接收重复起始位时，必须选择软件模式 (AUTOEND = 0)。将 START 位置 1 之前，必须将 PECBYTE 位置 1 并设置从地址。在这种情况下，当接收到 NBYTES - 1 字节的数据后，将自动使用 I2C_PECR 寄存器的内容对接收的下一个字节进行校验。接收到 PEC 字节后，TC 标志将置 1，SCL 线的低电平时间将延长。可以在 TC 中断子程序中设置重复起始位。

小心： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 563. SMBus 主接收器的总线传输图



48.4.15 地址匹配时从 Stop 模式唤醒

仅当支持从 Stop 模式唤醒功能（请参见*第 48.3 节*）时，才涉及本节内容。

被寻址时，I2C 能够将 MCU 从 Stop 模式唤醒（APB 时钟关断）。支持所有寻址模式。

将 I2C_CR1 寄存器中的 WUPEN 位置 1，可以使能从 Stop 模式唤醒功能。对于 I2CCLK，必须选择 HSI 或 CSI 振荡器作为时钟源，以便从 Stop 模式唤醒。

在 Stop 模式期间，HSI 或 CSI 关闭。当检测到 START 时，I2C 接口将 HSI 或 CSI 接通，并延长 SCL 使其处于低电平直到唤醒 HSI 或 CSI。

HSI 或 CSI 随后用来接收地址。

地址匹配的情况下，MCU 唤醒时间内，I2C 延长 SCL 的低电平持续时间。通过软件清零 ADDR 标志后，此延长操作被释放，传输正常进行。

如果地址不匹配，HSI 或 CSI 再次关断，MCU 不被唤醒。

注： *如果 I2C 时钟是系统时钟，或者 WUPEN = 0，则接收到 START 后，*HSI 或 CSI *不会接通。*

只有 ADDR 中断能够唤醒 MCU。因此，当 I2C 以主设备身份或在 ADDR 标志置 1 后以被寻址从设备身份执行传输时，不要进入 Stop 模式。通过在 ADDR 中断程序中清零 SLEEPDEEP 位，然后仅在 STOPF 标志置 1 后再将其置 1，可对此进行管理。

小心： 数字滤波器与从 Stop 模式唤醒功能不兼容。如果 DNF 位不等于 0，则将 WUPEN 位置 1 将不起任何作用。

小心： 只有当 I2C 时钟源为 HSI 或 CSI 振荡器时，该功能才可用。

小心： 必须使能时钟延长 (NOSTRETCH = 0) 才能确保从 Stop 模式唤醒功能正常工作。

小心： 如果禁止从 Stop 模式唤醒 (WUPEN = 0)，则在进入 Stop 模式前必须禁止 I2C 外设 (PE = 0)。

48.4.16 错误条件

以下错误是可能导致通信失败的条件。

总线错误 (BERR)

当检测到起始位或停止位但不位于第九个 SCL 时钟脉冲之后时，会检测到总线错误。当 SDA 边沿出现且 SCL 为高电平时，会检测到起始或停止位。

只有当 I2C 在传输过程中用作主设备或被寻址为从设备时（即未处于从模式下的地址阶段），才会将总线错误标志置 1。

在从模式下检测到错位的起始位或重复起始位时，I2C 会像接收到正确的起始位一样进入地址识别状态。

检测到总线错误时，I2C_ISR 寄存器中的 BERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

仲裁丢失 (ARLO)

当 SDA 线上发送高电平但在 SCL 上升沿却采样到低电平时，会检测到仲裁丢失。

- 在主模式下，将在地址阶段、数据阶段和数据应答阶段检测到仲裁丢失。在这种情况下，SDA 线和 SCL 线被释放，起始控制位由硬件清零，主设备自动切换为从模式。
- 在从模式下，将在数据阶段和数据应答阶段检测到仲裁丢失。在这种情况下，传输停止，SCL 和 SDA 线被释放。

检测到仲裁丢失时，I2C_ISR 寄存器中的 ARLO 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

上溢/下溢错误 (OVR)

当满足 NOSTRETCH = 1 和以下条件时，将在从模式下检测到上溢或下溢错误：

- 在接收过程中接收到一个新的字节，但 RXDR 寄存器的值还未被读取。接收的新字节丢失，自动发送 NACK 来响应新字节。
- 在发送过程中：
  - 当 STOPF = 1 且必须发送第一个数据字节时。TXE = 0 时发送 I2C_TXDR 寄存器的内容，否则发送 0xFF。
  - 必须发送一个新字节但尚未向 I2C_TXDR 寄存器写入数据时，将发送 0xFF。

检测到上溢或下溢错误时，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

数据包错误校验错误 (PECERR)

仅当支持 SMBus 功能（请参见*第 48.3 节*）时，才涉及本节内容。

当接收到的 PEC 字节与 I2C_PECR 寄存器的内容不匹配时，将检测到 PEC 错误。接收到错误的 PEC 后，将自动发送 NACK。

检测到 PEC 错误时，I2C_ISR 寄存器中的 PECERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

超时错误 (TIMEOUT)

仅当支持 SMBus 功能（请参见*第 48.3 节*）时，才涉及本节内容。

满足以下任何条件均会出现超时错误：

- TIDLE = 0 且 SCL 的低电平持续时间达到 TIMEOUTA[11:0] 位中定义的时间：这用于检测 SMBus 超时。
- TIDLE = 1 且 SDA 和 SCL 的高电平持续时间达到 TIMEOUTA[11:0] 位中定义的时间：这用于检测总线空闲情况。
- 主设备的累积时钟低电平延长时间达到了 TIMEOUTB[11:0] 位中定义的时间（SMBus $t_{LOW:MEXT}$ 参数）。
- 从设备的累积时钟低电平延长时间达到了 TIMEOUTB[11:0] 位中定义的时间（SMBus $t_{LOW:SEXT}$ 参数）。

当在主模式下检测到超时时，将自动发送停止位。

当在从模式下检测到超时时，将自动释放 SDA 和 SCL 线。

检测到超时错误时，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

报警 (ALERT)

仅当支持 SMBus 功能（请参见*第 48.3 节*）时，才涉及本节内容。

当 I2C 接口配置为主机 (SMBHEN = 1)、使能了报警引脚检测 (ALERTEN = 1) 并且在 SMBA 引脚上检测到下降沿时，ALERT 标志将置 1。如果 I2C_CR1 寄存器中的 ERRIE 位置 1，将生成中断。

48.4.17 DMA 请求

使用 DMA 进行发送

将 I2C_CR1 寄存器中的 TXDMAEN 位置 1 可以使能 DMA（直接内存访问）进行发送。当 TXIS 位置 1 时，数据将从由 DMA 外设配置的 SRAM 区（请参见*第 16 章：直接内存访问控制器 (DMA)*）装载进 I2C_TXDR 寄存器。

只有数据字节采用 DMA 进行传输。

- 在主模式下：初始化、从地址、方向、字节数和起始位均由软件编程（发送的从地址无法通过 DMA 传输）。当所有数据均通过 DMA 传输时，必须在起始位置 1 之前初始化 DMA。传输结束由 NBYTES 计数器来管理。请参见*主发送器*。
- 在从模式下：
  - 当 NOSTRETCH = 0 时，如果所有数据均通过 DMA 传输，则必须在地址匹配事件之前（或清零 ADDR 之前在 ADDR 中断子程序中）初始化 DMA。
  - 当 NOSTRETCH = 1 时，必须在地址匹配事件之前初始化 DMA。
- 支持 SMBus 时：PEC 传输由 NBYTES 计数器管理。请参见 *SMBus 从发送器*和 *SMBus 主发送器*。

注：如果使用 DMA 进行发送，则无需使能 TXIE 位。

使用 DMA 进行接收

将 I2C_CR1 寄存器中的 RXDMAEN 位置 1 可以使能 DMA（直接内存访问）进行接收。当 RXNE 位置 1 时，数据将从 I2C_RXDR 寄存器装载进由 DMA 外设配置的 SRAM 区（请参见*第 16 章：直接内存访问控制器 (DMA)*）。只有数据（包括 PEC）是通过 DMA 进行传输的。

- 在主模式下：初始化、从地址、方向、字节数和起始位均由软件编程。当所有数据均通过 DMA 传输时，必须在起始位置 1 之前初始化 DMA。传输结束由 NBYTES 计数器来管理。
- 在从模式下，当 NOSTRETCH = 0 时，如果所有数据均通过 DMA 传输，则必须在地址匹配事件之前（或清零 ADDR 标志之前在 ADDR 中断子程序中）初始化 DMA。
- 如果支持 SMBus（请参见*第 48.3 节*）：PEC 传输由 NBYTES 计数器管理。请参见 *SMBus 从接收器*和 *SMBus 主接收器*。

注：如果使用 DMA 进行接收，则无需使能 RXIE 位。

48.4.18 调试模式

当微控制器进入调试模式时（内核停止），SMBus 超时定时器会根据 DBG 模块中的 DBG_I2Cx_ 配置位选择继续正常工作或者停止工作。

48.5 I2C 低功耗模式

表 434. 低功耗模式对 I2C 的影响

模式	说明
Sleep	无影响。I2C 中断可使设备退出睡眠模式。
Stop(1)	I2C 模块的寄存器内容仍被保持。 – WUPEN = 1 并且 I2C 的时钟由内部振荡器 (HSI 或 CSI) 提供：地址识别功能正常。I2C 地址匹配条件会导致设备退出 Stop 模式。 – WUPEN = 0：必须在进入 Stop 模式之前禁止 I2C。
Standby	I2C 外设掉电，退出待机模式后必须重新初始化。

1. 有关每个实例支持的 Stop 模式的信息，请参见*第 48.3 节*。如果不支持从特定 Stop 模式唤醒，则必须在进入此 Stop 模式之前禁止实例。

48.6 I2C 中断

下表列出了 I2C 中断请求。

表 435. I2C 中断请求

中断缩写	中断事件	事件标志	使能控制位	中断清除方法	退出睡眠模式	退出 Stop 模式	退出 Standby 模式
I2C_EV	接收缓冲区非空	RXNE	RXIE	读取 I2C_RXDR 寄存器	是	否	否
	发送缓冲区中断状态	TXIS	TXIE	写入 I2C_TXDR 寄存器			
	停止位检测中断标志	STOPF	STOPIE	写入 STOPCF = 1			
	传输完成等待重载	TCR	TCIE	写入 I2C_CR2 (NBYTES[7:0] ≠ 0)			
	传输完成	TC		写入 START = 1 或 STOP = 1			
	地址匹配	ADDR	ADDRIE	写入 ADDRCF = 1		是(1)	
	接收到 NACK 应答	NACKF	NACKIE	写入 NACKCF = 1		否	
I2C_ER	总线错误	BERR	ERRIE	写入 BERRCF = 1	是	否	否
	仲裁丢失	ARLO		写入 ARLOCF = 1			
	上溢/下溢	OVR		写入 OVRCF = 1			
I2C_ER	PEC 错误	PECERR	ERRIE	写入 PECERRCF = 1	是	否	否
	超时/t_{LOW} 错误	TIMEOUT		写 TIMEOUTCF = 1			
	SMBus 报警	ALERT		写入 ALERTCF = 1			

1. 仅当 I2C 实例支持从 Stop 模式唤醒功能时，ADDR 匹配事件才能将设备从 Stop 模式唤醒。请参见*第 48.3 节*。

48.7 I2C 寄存器

有关寄存器说明中使用的缩写，请参见*第 109 页的第 1.2 节*。

寄存器按字（32 位）进行访问。

48.7.1 I2C 控制寄存器 1 (I2C_CR1)

I2C control register 1

偏移地址：0x00

复位值：0x0000 0000

访问：无等待周期，正在执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x i2c_pclk + 6 x i2c_ker_ck。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC EN	ALERT EN	SMBD EN	SMBH EN	GC EN	WUP EN	否 STRETCH	SBC
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDMA EN	TXDMA EN	Res.	ANF 关闭	DNF[3:0]				ERRIE	TCIE	STOP IE	NACK IE	ADDR IE	RXIE	TXIE	PE
rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23 **PECEN**：PEC 使能 (PEC enable)
0：禁止 PEC 计算
1：使能 PEC 计算
注意：如果不支持 SMBus 功能，该位保留并由硬件强制清零。请参见第 48.3 节。

位 22 **ALERTEN**：SMBus 报警使能 (SMBus alert enable)
0：在主机模式 (SMBHEN = 1) 下，不支持 SMBus 报警引脚 (SMBA)。在设备模式 (SMBHEN = 0) 下，会释放 SMBA 引脚并禁止报警响应地址头（0001100x 后跟 NACK）。
1：在主机模式 (SMBHEN = 1) 下，支持 SMBus 报警引脚。在设备模式 (SMBHEN = 0) 下，会将 SMBA 引脚驱动为低电平并使能报警响应地址头（0001100x 后跟 ACK）。
注意：当 ALERTEN = 0 时，SMBA 引脚可用作标准 GPIO。
如果不支持 SMBus 功能，该位保留并由硬件强制清零。请参见第 48.3 节。

位 21 **SMBDEN**：SMBus 设备默认地址使能 (SMBus device default address enable)
0：禁止设备默认地址。不对地址 0b1100001x 应答。
1：使能设备默认地址。对地址 0b1100001x 应答。
注意：如果不支持 SMBus 功能，该位保留并由硬件强制清零。请参见第 48.3 节。

位 20 **SMBHEN**：SMBus 主机地址使能 (SMBus host address enable)
0：禁止主机地址。不对地址 0b0001000x 应答。
1：使能主机地址。对地址 0b0001000x 应答。
注意：如果不支持 SMBus 功能，该位保留并由硬件强制清零。请参见第 48.3 节。

位 19 **GCEN**：广播呼叫使能 (General call enable)
0：禁止广播呼叫。不对地址 0b00000000 应答。
1：使能广播呼叫。对地址 0b00000000 应答。

位 18 **WUPEN**：从 Stop 模式唤醒使能 (Wake-up from Stop mode enable)
0：禁止从 Stop 模式唤醒。
1：使能从 Stop 模式唤醒。
注意：如果不支持从 Stop 模式唤醒功能，该位保留并由硬件强制清零。请参见第 48.3 节。
注意：只有当 DNF = 0000 时，WUPEN 才能置 1

位 17 **NOSTRETCH**：时钟延长禁止 (Clock stretching disable)
该位用于在从模式下禁止时钟延长。它在主模式下必须保持清零。
0：使能时钟延长
1：禁止时钟延长
注意：该位只能在 I2C 禁止时 (PE = 0) 编程。

位 16 **SBC**：从设备字节控制 (Slave byte control)
该位用于在从设备模式下使能硬件字节控制。
0：禁止从设备字节控制
1：使能从设备字节控制

位 15 **RXDMAEN**：DMA 接收请求使能 (DMA reception requests enable)
0：禁止 DMA 接收请求
1：使能 DMA 接收请求

位 14 **TXDMAEN**：DMA 发送请求使能 (DMA transmission requests enable)
0：禁止 DMA 发送请求
1：使能 DMA 发送请求

位 13 保留，必须保持复位值。

位 12 **ANFOFF**：模拟噪声滤波器关闭 (Analog noise filter OFF)
0：使能模拟噪声滤波器
1：禁止模拟噪声滤波器
注意：该位只能在 I2C 禁止时 (PE = 0) 编程。

位 11:8 **DNF[3:0]**：数字噪声滤波器 (Digital noise filter)
这些位用于配置 SDA 和 SCL 输入端的数字噪声滤波器。数字滤波器可滤除脉宽 DNF[3:0] * t_{I2CCLK} 以下的尖峰
0000：禁止数字滤波器
0001：使能数字滤波器，可滤除的噪声尖峰脉宽可达 1 t_{I2CCLK}
...
1111：使能数字滤波器，可滤除的噪声尖峰脉宽可达 15 t_{I2CCLK}
注意：如果模拟滤波器已使能，数字滤波将叠加在模拟滤波之上。该滤波器只能在 I2C 禁止时 (PE = 0) 编程。

位 7 **ERRIE**：错误中断使能 (Error interrupt enable)
0：禁止错误检测中断
1：使能错误检测中断
注意：以下任一错误均会生成中断：
仲裁丢失 (ARLO)
总线错误检测 (BERR)
上溢/下溢 (OVR)
超时检测 (TIMEOUT)
PEC 错误检测 (PECERR)
报警引脚事件检测 (ALERT)

位 6 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)
0：禁止传输完成中断
1：使能传输完成中断
注意：以下任一事件均会生成中断：
传输完成 (TC)
传输完成等待重载 (TCR)

位 5 **STOPIE**：停止位检测中断使能 (Stop detection interrupt enable)
0：禁止停止位检测 (STOPF) 中断
1：使能停止位检测 (STOPF) 中断

位 4 **NACKIE**：接收到拒绝应答中断使能 (Not acknowledge received interrupt enable)
0：禁止接收到拒绝应答 (NACKF) 中断
1：使能接收到拒绝应答 (NACKF) 中断

位 3 **ADDRIE**：地址匹配中断使能（仅从模式）(Address match interrupt enable (slave only))
0：禁止地址匹配 (ADDR) 中断
1：使能地址匹配 (ADDR) 中断

位 2 **RXIE**：RX 中断使能 (RX interrupt enable)
0：禁止接收 (RXNE) 中断
1：使能接收 (RXNE) 中断

位 1 **TXIE**：TX 中断使能 (TX interrupt enable)
0：禁止发送 (TXIS) 中断
1：使能发送 (TXIS) 中断

位 0 **PE**：外设使能 (Peripheral enable)
0：禁止外设
1：使能外设
注意：当 PE = 0 时，将释放 I2C SCL 线和 SDA 线。内部状态机和状态位均恢复为复位值。清零时，PE 必须保持低电平状态至少三个 APB 时钟周期。

48.7.2 I2C 控制寄存器 2 (I2C_CR2)

I2C control register 2

偏移地址：0x04

复位值：0x0000 0000

访问：无等待周期，正在执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x i2c_pclk + 6 x i2c_ker_ck。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	PEC BYTE	AUTO END	RE LOAD	NBYTES[7:0]							
					rs	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACK	STOP	START	HEAD 10R	ADD10	RD_ WRN	SADD[9:0]									
rs	rs	rs	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:27 保留，必须保持复位值。

位 26 **PECBYTE**：数据包错误校验字节 (Packet error checking byte)

该位由软件置 1，并可在 PEC 传输完成时、接收到停止位或匹配地址时、或者 PE=0 时由硬件清零。

0：不传输 PEC

1：请求 PEC 发送/接收。

注意：向该位写入 0 不起作用。

当 RELOAD 置 1 时，以及在从模式下当 SBC = 0 时，该位不起作用。

注意：如果不支持 SMBus 功能，该位保留并由硬件强制清零。请参见第 48.3 节。

位 25 **AUTOEND**：自动结束模式（主模式）(Automatic end mode (master mode))

该位由软件置 1 和清零，

0：软件结束模式：当 NBYTES 数据传输完成时，TC 标志将置 1，SCL 的低电平时间将延长直到相应软件操作结束。

1：自动结束模式：当 NBYTES 数据传输完成时，将自动发送停止位。

注意：在从模式下，或当 RELOAD 位置 1 时，该位不起作用。

位 24 **RELOAD**：NBYTES 重载模式 (NBYTES reload mode)

该位由软件置 1 和清零，

0：传输 NBYTES 数据（后跟停止位或重复起始位）之后即完成传输。

1：传输 NBYTES 数据之后未完成传输（将重载 NBYTES）。当 NBYTES 数据传输完成时，TCR 标志将置 1，SCL 的低电平时间将延长直到相应软件操作结束。

位 23:16 **NBYTES[7:0]**：字节数 (Number of bytes)

在此设置待发送/接收的字节数。在从模式下，当 SBC = 0 时，该字段为无关位。

注意：START 位置 1 时，不允许更改这些位。

位 15 **NACK**：NACK 生成（从模式）(NACK generation (slave mode))

该位由软件置 1，并可在发送 NACK 时、接收到停止位或匹配地址时、或者 PE = 0 时由硬件清零。

0：在当前接收的字节后发送 ACK。

1：在当前接收的字节后发送 NACK。

注意：向该位写入 0 不起作用。

该位仅在从模式下使用：在主接收模式中，无论 NACK 位的值是多少，都会在停止或重新启动条件之前的最后一个字节之后自动生成非应答 (NACK) 信号。

当从接收器 NOSTRETCH 模式下发生上溢时，无论 NACK 位的值为何，都将自动生成 NACK。

使能硬件 PEC 校验时 (PECBYTE = 1)，PEC 应答值与 NACK 值无关。

位 14 **STOP**：停止位生成（主模式）(Stop generation (master mode))

该位由软件置 1，并可在检测到停止位时或 PE = 0 时由硬件清零。

在主模式下：

0：不生成停止位

1：在当前字节传输完成后生成停止位

注意：向该位写入 0 不起作用。

位 13 **START**：生成起始位 (Start generation)

该位由软件置 1，并可在发送起始位（后跟地址序列）之后、发生仲裁丢失时、出现超时错误时、或者 PE = 0 时由硬件清零。它也可由软件清零，方法是向 I2C_ICR 寄存器中的 ADDRCF 位写入 1。

0：不生成起始位

1：生成重复起始/起始位：

如果 I2C 已处于主模式下且 AUTOEND = 0，则将该位置 1 会在 NBYTES 传输结束后且 RELOAD = 0 的情况下生成重复起始位。

否则，将该位置 1 会在总线释放后立即生成起始位。

注意：向该位写入 0 不起作用。

即使总线繁忙或 I2C 处于从模式，也可将 START 位置 1。

当 RELOAD 置 1 时，该位不起作用。

位 12 **HEAD10R**：读方向传输时，只发送 10 位地址的前 7 位地址头字节（主接收器模式）(10-bit address header only read direction (master receiver mode))

0：主设备发送完整的 10 位从地址读序列：起始位 + 带写方向的 2 字节 10 位地址 + 重复起始位 + 带读方向的 10 位地址的前 7 位。

1：主设备只发送 10 位地址的前 7 位，后面是读方向。

注意：START 位置 1 时，不允许更改此位。

位 11 **ADD10**：10 位寻址模式（主模式）(10-bit addressing mode (master mode))

0：主设备工作在 7 位寻址模式下

1：主设备工作在 10 位寻址模式下

注意：START 位置 1 时，不允许更改此位。

位 10 **RD_WRN**：传输方向（主模式）(Transfer direction (master mode))

0：主设备请求写传输

1：主设备请求读传输

注意：START 位置 1 时，不允许更改此位。

位 9:0 **SADD[9:0]**：从地址（主模式）(Slave address (master mode))

在 7 位寻址模式 (ADD10 = 0) 下：

SADD[7:1] 必须写入待发送的 7 位从地址。SADD[9]、SADD[8] 和 SADD[0] 位为无关位。

在 10 位寻址模式 (ADD10 = 1) 下：

SADD[9:0] 必须写入待发送的 10 位从地址。

注意：START 位置 1 时，不允许更改这些位。

48.7.3 I2C 自身地址 1 寄存器 (I2C_OAR1)

I2C own address 1 register

偏移地址：0x08

复位值：0x0000 0000

访问：无等待周期，正在执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x i2c_pclk + 6 x i2c_ker_ck。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA1EN	Res.	Res.	Res.	Res.	OA1 MODE	OA1[9:0]									
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **OA1EN**：设备自身地址 1 使能 (Own address 1 enable)
0：禁止设备自身地址 1。不对接收的从地址 OA1 应答。
1：使能设备自身地址 1。对接收的从地址 OA1 应答。

位 14:11 保留，必须保持复位值。

位 10 **OA1MODE**：设备自身地址 1 10 位模式 (Own address 1 10-bit mode)
0：设备自身地址 1 为 7 位地址。
1：设备自身地址 1 为 10 位地址。
注意：仅可在 OA1EN = 0 时写入该位。

位 9:0 **OA1[9:0]**：接口自身从地址 (Interface own slave address)
7 位寻址模式：OA1[7:1] 包含 7 位自身从地址。OA1[9]、OA1[8] 和 OA1[0] 位为无关位。
10 位寻址模式：OA1[9:0] 包含 10 位自身从地址。
注意：仅可在 OA1EN = 0 时写入这些位。

48.7.4 I2C 自身地址 2 寄存器 (I2C_OAR2)

I2C own address 2 register

偏移地址：0x0C

复位值：0x0000 0000

访问：无等待周期，正在执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x i2c_pclk + 6 x i2c_ker_ck。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA2EN	Res.	Res.	Res.	Res.	OA2MSK[2:0]			OA2[7:1]							Res.
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位 31:16 保留，必须保持复位值。

位 15 **OA2EN**：设备自身地址 2 使能 (Own address 2 enable)
0：禁止设备自身地址 2。不对接收的从地址 OA2 应答。
1：使能设备自身地址 2。对接收的从地址 OA2 应答。

位 14:11 保留，必须保持复位值。

位 10:8 **OA2MSK[2:0]**：设备自身地址 2 屏蔽位 (Own address 2 masks)

000：无屏蔽
001：OA2[1] 被屏蔽，为无关位。仅比较 OA2[7:2]。
010：OA2[2:1] 被屏蔽，为无关位。仅比较 OA2[7:3]。
011：OA2[3:1] 被屏蔽，为无关位。仅比较 OA2[7:4]。
100：OA2[4:1] 被屏蔽，为无关位。仅比较 OA2[7:5]。
101：OA2[5:1] 被屏蔽，为无关位。仅比较 OA2[7:6]。
110：OA2[6:1] 被屏蔽，为无关位。仅比较 OA2[7]。
111：OA2[7:1] 被屏蔽，为无关位。不进行比较，对接收到的全部 7 位地址（保留位除外）应答。

注意：仅可在 OA2EN = 0 时写入这些位。

只要 OA2MSK ≠ 0，即使比较匹配，也不会对保留的 I2C 地址（0b0000xxx 和 0b1111xxx）进行应答。

位 7:1 **OA2[7:1]**：接口地址 (Interface address)

7 位寻址模式：7 位地址

注意：仅可在 OA2EN = 0 时写入这些位。

位 0 保留，必须保持复位值。

48.7.5 I2C 时序寄存器 (I2C_TIMINGR)

I2C timing register

偏移地址：0x10

复位值：0x0000 0000

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESC[3:0]				Res.	Res.	Res.	Res.	SCLDEL[3:0]				SDADEL[3:0]			
rw	rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH[7:0]								SCLL[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 **PRESC[3:0]**：时序预分频因子 (Timing prescaler)

该位字段用于对 i2c_ker_ck 进行预分频，以生成用于数据建立和保持计数器（请参见 *I2C 时序*）以及 SCL 高电平和低电平计数器（请参见 *I2C 主模式初始化*）的时钟周期 t_{PRESC}。

$t_{PRESC} = (PRESC + 1) \times t_{I2CCLK}$

位 27:24 保留，必须保持复位值。

位 23:20 **SCLDEL[3:0]**：数据建立时间

该位字段用于在 SDA 边沿和 SCL 上升沿之间生成延时 $t_{SCLDEL} = (SCLDEL + 1) \times t_{PRESC}$。在主模式和从模式下，如果 NOSTRETCH = 0，则 SCL 线的低电平时间将在 t_{SCLDEL} 期间延长。

注意：t_{SCLDEL} 用于生成 $t_{SU:DAT}$ 时序。

位 19:16 **SDADEL[3:0]**：数据保持时间 (Data hold time)

该位字段用于在 SCL 下降沿和 SDA 边沿之间生成延时 t_{SDADEL}。在主模式和从模式下，如果 NOSTRETCH = 0，则 SCL 线的低电平时间将在 t_{SDADEL} 期间延长。

t_{SDADEL}= SDADEL x t_{PRESC}

注意：SDADEL 用于生成 $t_{HD:DAT}$ 时序。

位 15:8 **SCLH[7:0]**：SCL 高电平周期（主模式）(SCL high period (master mode))

在主模式下，该位字段用于生成 SCL 高电平周期。

t_{SCLH} = (SCLH + 1) x t_{PRESC}

*注意：SCLH 还用于生成 $t_{SU:STO}$ 和 t*HD:STA *时序。*

位 7:0 **SCLL[7:0]**：SCL 低电平周期（主模式）(SCL low period (master mode))

在主模式下，该位字段用于生成 SCL 低电平周期。

t_{SCLL} = (SCLL + 1) x t_{PRESC}

注意：SCLL 还用于生成 t_{BUF} 和 $t_{SU:STA}$ 时序。

注：　该寄存器必须在 I2C 禁止时 (PE = 0) 进行配置。

注：　STM32CubeMX 工具计算 I2C_TIMINGR 内容并在 I2C 配置窗口中进行显示。

48.7.6 I2C 超时寄存器 (I2C_TIMEOUTR)

I2C timeout register

偏移地址：0x14

复位值：0x0000 0000

访问：无等待周期，正在执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x i2c_pclk + 6 x i2c_ker_ck。

如果不支持 SMBus 功能，该寄存器保留，寄存器位由硬件强制清零。请参见*第 48.3 节*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEXTEN	Res.	Res.	Res.	TIMEOUTB[11:0]											
rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMOUTEN	Res.	Res.	TIDLE	TIMEOUTA[11:0]											
rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **TEXTEN**：时钟信号延长超时使能 (Extended clock timeout enable)

0：禁止时钟信号延长超时检测

1：使能时钟信号延长超时检测当 I2C 接口执行 SCL 延长的累积时间超过 $t_{LOW:EXT}$ 时，将检测到超时错误 (TIMEOUT = 1)。

位 30:28 保留，必须保持复位值。

位 27:16 **TIMEOUTB[11:0]**：总线超时 B (Bus timeout B)

该位字段用于配置累积时钟延长超时：

– 主模式：检测主设备的累积时钟低电平延长时间 ($t_{LOW:MEXT}$)

– 从模式：检测从设备的累积时钟低电平延长时间 ($t_{LOW:SEXT}$)

$t_{LOW:EXT}$ = (TIMEOUTB + TIDLE = 01) x 2048 x t_{I2CCLK}

注意：仅可在 TEXTEN = 0 时写入这些位。

位 15 **TIMOUTEN**：时钟超时使能 (Clock timeout enable)

0：禁止 SCL 超时检测

1：使能 SCL 超时检测：当 SCL 的低电平时间超过 $t_{TIMEOUT}$ (TIDLE = 0)，或 SCL 的高电平时间超过 t_{IDLE} (TIDLE = 1) 时，将检测到超时错误 (TIMEOUT = 1)。

位 14:13 保留，必须保持复位值。

位 12 **TIDLE**：空闲时钟超时检测 (Idle clock timeout detection)

0：TIMEOUTA 用于检测 SCL 低电平超时

1：TIMEOUTA 用于检测 SCL 和 SDA 高电平超时（总线空闲条件）

注意：仅可在 TIMOUTEN = 0 时写入该位。

位 11:0 **TIMEOUTA[11:0]**：总线超时 A (Bus timeout A)

该位字段用于配置：

SCL 低电平超时条件 $t_{TIMEOUT}$（当 TIDLE = 0 时）

$t_{TIMEOUT}$= (TIMEOUTA + 1) x 2048 x t_{I2CCLK}

总线空闲条件，即 SCL 和 SDA 高电平（当 TIDLE = 1 时）

t_{IDLE}= (TIMEOUTA + 1) x 4 x t_{I2CCLK}

注意：仅可在 TIMOUTEN = 0 时写入这些位。

48.7.7 I2C 中断和状态寄存器 (I2C_ISR)

I2C interrupt and status register

偏移地址：0x18

复位值：0x0000 0001

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR
								r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	Res.	ALERT	TIME OUT	PEC ERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
r		r	r	r	r	r	r	r	r	r	r	r	r	rs	rs

位 31:24 保留，必须保持复位值。

位 23:17 **ADDCODE[6:0]**：地址匹配代码（从模式）(Address match code (Slave mode))

发生地址匹配事件时 (ADDR = 1)，这些位更新为接收到的地址。在 10 位地址的情况下，ADDCODE 提供 10 位地址的头字节，后跟地址的两个 MSB。

位 16 **DIR**：传输方向（从模式）(Transfer direction (slave mode))

该标志在发生地址匹配事件时 (ADDR = 1) 更新。

0：写传输，从设备进入接收器模式。

1：读传输，从设备进入发送器模式。

位 15 **BUSY**：总线忙碌 (Bus busy)

该标志用于指示总线上正在进行通信。当检测到 START 条件时由硬件置 1，当检测到 STOP 条件或当 PE = 0 时由硬件清零。

位 14 保留，必须保持复位值。

位 13 **ALERT**：SMBus 报警

当 SMBHEN = 1（SMBus 主机配置）、ALERTEN = 1 且在 SMBA 引脚上检测到 SMBALERT 事件（下降沿）时，该标志由硬件置 1。该位由软件清零，方法是将 ALERTCF 位置 1。

注意：当 PE = 0 时，该位由硬件清零。

注意：如果不支持 SMBus 功能，该位保留并由硬件强制清零。请参见第 48.3 节。

位 12 **TIMEOUT**：超时或 t_{LOW} 检测标志 (Timeout or t_{LOW} detection flag)

发生超时或延长时钟超时时，该标志由硬件置 1。该位由软件清零，方法是将 TIMEOUTCF 位置 1。

注意：当 PE = 0 时，该位由硬件清零。

注意：如果不支持 SMBus 功能，该位保留并由硬件强制清零。请参见第 48.3 节。

位 11 **PECERR**：接收期间 PEC 错误 (PEC error in reception)

当接收到的 PEC 与 PEC 寄存器的内容不匹配时，该标志由硬件置 1。接收到错误的 PEC 后，将自动发送 NACK。该标志由软件清零，方法是将 PECCF 位置 1。

注意：当 PE = 0 时，该位由硬件清零。

注意：如果不支持 SMBus 功能，该位保留并由硬件强制清零。请参见第 48.3 节。

位 10 **OVR**：上溢/下溢（从模式）(Overrun/underrun (slave mode))

在从模式下且 NOSTRETCH = 1 时，如果发生上溢/下溢错误，该标志由硬件置 1。该标志由软件清零，方法是将 OVRCF 位置 1。

注意：当 PE = 0 时，该位由硬件清零。

位 9 **ARLO**：仲裁丢失 (Arbitration lost)

发生仲裁丢失时，该标志由硬件置 1。该标志由软件清零，方法是将 ARLOCF 位置 1。

注意：当 PE = 0 时，该位由硬件清零。

位 8 **BERR**：总线错误

当检测到错位的起始位或停止位，而外设也参与传输时，该标志由硬件置 1。在从模式下的地址阶段，该标志不会置 1。该位由软件清零，方法是将 BERRCF 位置 1。

注意：当 PE = 0 时，该位由硬件清零。

位 7 **TCR**：传输完成等待重载

当 RELOAD = 1 且 NBYTES 数据传输完成时，该标志由硬件置 1。当 NBYTES 写入一个非零值时，该标志由软件清零。

注意：当 PE = 0 时，该位由硬件清零。

该标志单独用于主模式，SBC 位置 1 时单独用于从模式。

位 6 **TC**：传输完成（主模式）(Transfer complete (master mode))

当 RELOAD = 0、AUTOEND = 0 且 NBYTES 数据传输完成时，该标志由硬件置 1。当 START 位或 STOP 位置 1 时，该标志由软件清零。

注意：当 PE = 0 时，该位由硬件清零。

位 5 **STOPF**：停止位检测标志 (Stop detection flag)

当在总线上检测到停止位，且外设也参与本次传输时，该标志由硬件置 1：

- 外设作为主设备，该位置 1 的前提是外设已经发出停止条件。
- 外设作为从设备，该位置 1 的前提条件是此次传输的寻址对象就是该外设。

该标志由软件清零，方法是将 STOPCF 位置 1。

注意：当 PE = 0 时，该位由硬件清零。

位 4 **NACKF**：接收到拒绝应答标志 (Not acknowledge received flag)

发送完字节后接收到 NACK 时，该标志由硬件置 1。该标志由软件清零，方法是将 NACKCF 位置 1。

注意：当 PE = 0 时，该位由硬件清零。

位 3 **ADDR**：地址匹配（从模式）(Address matched (slave mode))
接收到的地址与使能的从设备地址之一匹配时，该位由硬件置 1。该位由软件清零，方法是将 *ADDRCF 位置 1*。
注意：当 PE = 0 时，该位由硬件清零。

位 2 **RXNE**：接收数据寄存器不为空（接收器）(Receive data register not empty (receivers))
当接收到的数据已复制到 I2C_RXDR 寄存器且准备就绪可供读取时，该位由硬件置 1。读取 I2C_RXDR 时，将清零该位。
注意：当 PE = 0 时，该位由硬件清零。

位 1 **TXIS**：发送中断状态（发送器）(Transmit interrupt status (transmitters))
当 I2C_TXDR 寄存器为空时，该位由硬件置 1，待发送的数据必须写入 I2C_TXDR 寄存器。下一个待发送的数据写入 I2C_TXDR 寄存器时，该位被清零。
该位只能在 NOSTRETCH = 1 时由软件写入 1，以生成 TXIS 事件（TXIE = 1 时为中断，TXDMAEN = 1 时为 DMA 请求）。
注意：当 PE = 0 时，该位由硬件清零。

位 0 **TXE**：发送数据寄存器为空（发送器）(Transmit data register empty (transmitters))
当 I2C_TXDR 寄存器为空时，该位由硬件置 1。下一个待发送的数据写入 I2C_TXDR 寄存器时，该位被清零。
该位可由软件写入 1，以刷新发送数据寄存器 I2C_TXDR。
注意：当 PE = 0 时，该位由硬件置 1。

48.7.8 I2C 中断清零寄存器 (I2C_ICR)

I2C interrupt clear register

偏移地址：0x1C

复位值：0x0000 0000

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ALERT CF	TIMOUT CF	PEC CF	OVR CF	ARLO CF	BERR CF	Res.	Res.	STOP CF	NACK CF	ADDR CF	Res.	Res.	Res.
		w	w	w	w	w	w			w	w	w			

位 31:14 保留，必须保持复位值。

位 13 **ALERTCF**：报警标志清零 (Alert flag clear)
注意：将 1 写入该位时，I2C_ISR 寄存器中的 ALERT 标志将清零。
注意：如果不支持 SMBus 功能，该位保留并由硬件强制清零。请参见第 48.3 节。

位 12 **TIMOUTCF**：超时检测标志清零 (Timeout detection flag clear)
将 1 写入该位时，I2C_ISR 寄存器中的 TIMEOUT 标志将清零。
注意：如果不支持 SMBus 功能，该位保留并由硬件强制清零。请参见第 48.3 节。

位 11 **PECCF**：PEC 错误标志清零 (PEC error flag clear)
将 1 写入该位时，I2C_ISR 寄存器中的 PECERR 标志将清零。
注意：如果不支持 SMBus 功能，该位保留并由硬件强制清零。请参见第 48.3 节。

位 10 **OVRCF**：上溢/下溢标志清零 (Overrun/underrun flag clear)
将 1 写入该位时，I2C_ISR 寄存器中的 OVR 标志将清零。

位 9 **ARLOCF**：仲裁丢失标志清零 (Arbitration lost flag clear)
将 1 写入该位时，I2C_ISR 寄存器中的 ARLO 标志将清零。

位 8 **BERRCF**：总线错误标志清零 (Bus error flag clear)
将 1 写入该位时，I2C_ISR 寄存器中的 BERRF 标志将清零。

位 7:6 保留，必须保持复位值。

位 5 **STOPCF**：停止位检测标志清零 (STOP detection flag clear)
将 1 写入该位时，I2C_ISR 寄存器中的 STOPF 标志将清零。

位 4 **NACKCF**：拒绝应答标志清零 (Not acknowledge flag clear)
向该位写入 1 时，I2C_ISR 寄存器中的 NACKF 标志将清零。

位 3 **ADDRCF**：地址匹配标志清零 (Address Matched flag clear)
将 1 写入该位时，I2C_ISR 寄存器中的 ADDR 标志将清零。将 1 写入该位时，I2C_CR2 寄存器中的 START 位也将清零。

位 2:0 保留，必须保持复位值。

48.7.9 I2C PEC 寄存器 (I2C_PECR)

I2C PEC register

偏移地址：0x20

复位值：0x0000 0000

访问：无等待周期

如果不支持 SMBus 功能，该寄存器保留，寄存器位由硬件强制清零。请参见*第 48.3 节*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PEC[7:0]**：数据包错误校验寄存器 (Packet error checking register)
当 PECEN=1 时，该位字段包含内部 PEC。
当 PE = 0 时，PEC 由硬件清零。

48.7.10 I2C 接收数据寄存器 (I2C_RXDR)

I2C receive data register

偏移地址：0x24

复位值：0x0000 0000

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **RXDATA[7:0]**：8 位接收数据 (8-bit receive data)
从 I^2C 总线接收的数据字节。

48.7.11 I2C 发送数据寄存器 (I2C_TXDR)

I2C transmit data register

偏移地址：0x28

复位值：0x0000 0000

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXDATA[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **TXDATA[7:0]**：8 位发送数据 (8-bit transmit data)
待发送到 I^2C 总线的数据字节
注意：仅可在 TXE = 1 时写入这些位。

48.7.12 I2C 硬件配置寄存器 (I2C_HWCFGR)

I2C hardware configuration register

偏移地址：0x3F0

复位值：0x0000 0111

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				WKP[3:0]				ASYN[3:0]				SMBUS[3:0]			
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12 保留，必须保持复位值。

位 11:8 **WKP[3:0]**：

0：未从 Stop 模式唤醒

1：已从 Stop 模式唤醒

位 7:4 **ASYN[3:0]**：

0：未使用独立内核时钟

1：已使用独立内核时钟

位 3:0 **SMBUS[3:0]**：

0：未使能 SMBus 模式

1：已使能 SMBus 模式

48.7.13 I2C 版本寄存器 (I2C_VERR)

I2C version register

偏移地址：0x3F4

复位值：0x0000 0014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本 (Major revision)

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)

48.7.14 I2C 标识寄存器 (I2C_IPIDR)

I2C identification register

偏移地址：0x3F8

复位值：0x0013 0012

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：标识符 (Identifier)。

48.7.15 I2C 大小标识寄存器 (I2C_SIDR)

I2C size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识符 (Size identifier)。

48.7.16 I2C 寄存器映射

下表提供了 I2C 寄存器映射和复位值。

表 436. I2C 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	I2C_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	WUPEN	NOSTRETCH	SBC	RXDMAEN	TXDMAEN	Res.	ANFOFF	DNF[3:0]				ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE
	Reset value									0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	I2C_CR2	Res.	Res.	Res.	Res.	Res.	PECBYTE	AUTOEND	RELOAD	NBYTES[7:0]								NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD[9:0]									
	Reset value						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	I2C_OAR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA1EN	Res.	Res.	Res.	Res.	OA1MODE	OA1[9:0]									
	Reset value																	0					0	0	0	0	0	0	0	0	0	0	0
0x0C	I2C_OAR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA2EN	Res.	Res.	Res.	Res.	OA2MSK [2:0]	OA2[7:1]									Res.
	Reset value																	0					0	0	0	0	0	0	0	0	0	0	
0x10	I2C_TIMINGR	PRESC[3:0]				Res.	Res.	Res.	Res.	SCLDEL [3:0]				SDADEL [3:0]				SCLH[7:0]								SCLL[7:0]							
	Reset value	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	I2C_TIMEOUTR	TEXTEN	Res.	Res.	Res.	TIMEOUTB[11:0]												TIMOUTEN	Res.	Res.	TIDLE	TIMEOUTA[11:0]											
	Reset value	0				0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	I2C_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR	BUSY	Res.	ALERT	TIMEOUT	PECERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
	Reset value									0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x1C	I2C_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALERTCF	TIMOUTCF	PECCF	OVRCF	ARLOCF	BERRCF	Res.	Res.	STOPCF	NACKCF	ADDRCF	Res.	Res.	Res.
	Reset value																			0	0	0	0	0	0			0	0	0			
0x20	I2C_PECR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x24	I2C_RXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x28	I2C_TXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXDATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

表 436. I2C 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x03F0	I2C_HWCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.				WKP[3:0]				ASYN[3:0]				SMBUS[3:0]			
	Reset value																					0	0	0	1	0	0	0	1	0	0	0	1
0x03F4	I2C_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	0	0	1	0	1	0	0
0x03F8	I2C_IPIDR	ID[31:16]																ID[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
0x03FC	I2C_SIDR	SID[31:16]																SID[15:0]															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

49 通用同步/异步收发器 (USART/UART)

本节介绍通用同步/异步收发器 (USART)。

49.1 USART 简介

USART 能够灵活地与外部设备进行全双工数据交换，满足外部设备对工业标准 NRZ 异步串行数据格式的要求。USART 通过小数波特率发生器实现了多种波特率。

USART 不仅支持同步单向通信和半双工单线通信，以及 LIN（局域互连网络）、智能卡协议、IrDA（红外线数据协会）SIR ENDEC 规范和调制解调器操作 (CTS/RTS)，还支持多处理器通信。

通过配置多个缓冲区使用 DMA（直接内存访问）可实现高速数据通信。

49.2 USART 主要特性

- 全双工异步通信
- NRZ 标准格式（标记/空格）
- 可配置为 16 倍过采样或 8 倍过采样，从而在速度容差与时钟容差之间取得最佳平衡
- 波特率发生器系统
- 两个用于收发数据的内部 FIFO
  每个 FIFO 均可由软件使能/禁止，并且均带有一个状态标志。
- 通用可编程收发波特率
- 双时钟域，带有独立于 PCLK 的外设专用内核时钟
- 自动波特率检测
- 数据字长度可编程（7 位、8 位或 9 位）
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 停止位可配置（支持 1 个或 2 个停止位）
- 用于同步通信的同步主/从模式和时钟输出/输入
- SPI 从模式发送下溢错误标志
- 单线半双工通信
- 使用 DMA 实现连续通信
- 使用中心化 DMA 在预留的 SRAM 缓冲区中收/发字节。
- 发送器和接收器有单独的使能位
- 发送和接收的单独信号极性控制
- Tx/Rx 引脚可配置互换
- 调制解调器和 RS-485 收发器的硬件流控制
- 通信控制/错误检测标志

- 奇偶校验控制：
  - 发送奇偶校验位
  - 检查接收的数据字节的奇偶性
- 具有标志的中断源
- 多处理器通信：从静默模式唤醒（通过空闲线检测或地址标记检测）
- 从 Stop 模式唤醒

49.3 USART 扩展特性

- LIN 主模式同步间隔发送功能和 LIN 从模式间隔检测功能
  - 对 USART 进行 LIN 硬件配置时可生成 13 位间隔和检测 10/11 位间隔
- IrDA SIR 编解码器在正常模式下支持 3/16 位持续时间
- 智能卡模式
  - 对于 ISO/IEC 7816-3 标准中定义的智能卡，支持 T = 0 和 T = 1 异步协议
  - 智能卡工作模式下，支持 0.5 和 1.5 个停止位
- 支持 Modbus 通信
  - 超时功能
  - CR/LF 字符识别

49.4 USART 实现

下表介绍了 USART 实现。其中还包含 LPUART 作为参照。

表 437. STM32MP13xx 特性

USART 实例	STM32MP13xC
USART1	完全
USART2	完全
USART3	完全
USART6	完全
UART4	BASIC
UART5	BASIC
UART7	BASIC
UART8	BASIC

表 438. USART 特性

USART 模式/特性(1)	全面的功能集	基本的功能集
调制解调器的硬件流控制	X	X
使用 DMA 进行连续通信	X	X
多处理器通信	X	X
同步模式（主/从）	X	-
智能卡模式	X	-
单线半双工通信	X	X
IrDA SIR ENDEC 模块	X	X
LIN 模式	X	X
双时钟域	X	X
接收器超时中断	X	X
Modbus 通信	X	X
自动波特率检测	X	X
驱动器使能	X	X
USART 数据长度	7 位、8 位和 9 位	
Tx/Rx FIFO	X	X
Tx/Rx FIFO 大小	16	
从低功耗模式唤醒	X(2)	X(2)

1. X = 支持。
2. 支持从 Stop、LP-Stop、LPLV-Stop 和 LPLV-Stop2 模式唤醒。

49.5 USART 功能描述

49.5.1 USART 框图

图 564. USART 框图

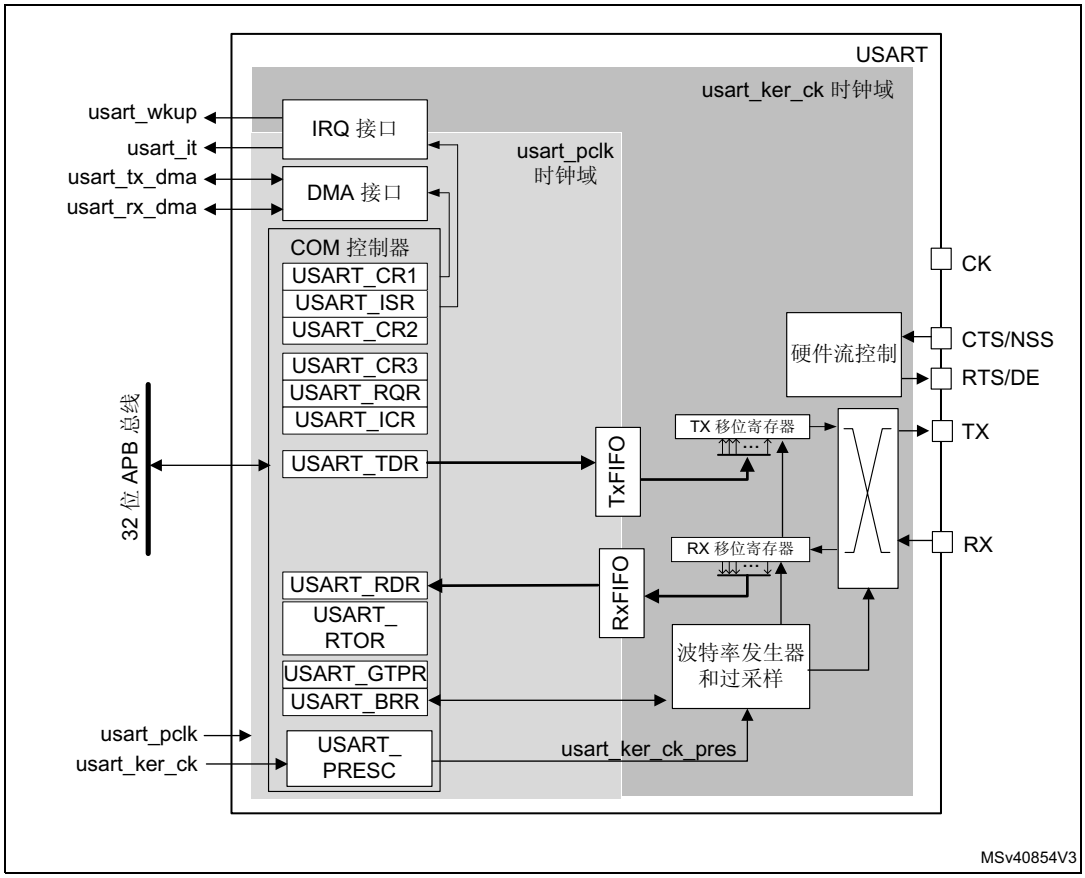


*图 564 中的简化框图*显示的是两个完全独立的时钟域：

- **usart_pclk** 时钟域

  usart_pclk 时钟信号为外设总线接口提供时钟。需要访问 USART 寄存器时，该信号必须有效。

- **usart_ker_ck** 内核时钟域。

  usart_ker_ck 是 USART 时钟源。它独立于 **usart_pclk**，由 RCC 提供。因此，即使 **usart_ker_ck** 时钟停止，也可以对 USART 寄存器执行读/写操作。

  禁用双时钟域功能时，**usart_ker_ck** 时钟与 **usart_pclk** 时钟相同。

usart_pclk 和 **usart_ker_ck** 之间无任何约束：**usart_ker_ck** 可快于/可慢于 **usart_pclk**。唯一的限制是软件以足够快的速度管理通信的能力。

USART 工作在 SPI 从设备模式下时，会使用源自外部 CK 信号（由外部主 SPI 设备提供）的串行接口时钟来处理数据流。**usart_ker_ck** 的时钟频率必须至少 3 倍于 CK 输入上的时钟。

49.5.2 USART 信号

USART 双向通信

USART 双向通信需要至少两个引脚：接收数据输入引脚 (RX) 和发送数据输出引脚 (TX)：

- **RX**（接收数据输入引脚）

  RX 为串行数据输入引脚。采用过采样技术进行数据恢复，即区分有效输入数据和噪声。

- **TX**（发送数据输出引脚）

  如果关闭发送器，该输出引脚模式由其 I/O 端口配置决定。如果使能了发送器但没有需要发送的数据，则 TX 引脚处于高电平。在单线和智能卡模式下，该 I/O 用于发送和接收数据。

RS232 硬件流控制模式

在 RS232 硬件流控制模式下需要以下引脚：

- **CTS**（清除以发送）

  如果驱动为高电平，则该信号用于在当前传输结束时阻止数据发送。

- **RTS**（请求以发送）

  如果为低电平，则该信号用于指示 USART 已准备好接收数据。

RS485 硬件控制模式

在 RS485 硬件控制模式下需要以下引脚：

- **DE**（驱动器使能）

  该信号用于激活外部收发器的发送模式。

注： *DE 和 RTS 共用同一个引脚。*

同步主/从模式和智能卡模式

在同步主/从模式和智能卡模式下需要以下引脚：

- CK

  该引脚在同步主模式和智能卡模式下用作时钟输出。

  它在同步从模式下用作时钟输入。

  在同步主模式下，该引脚用于输出发送器数据时钟，以便按照 SPI 主设备模式进行同步发送（起始位和结束位上无时钟脉冲，可通过软件向最后一个数据位发送时钟脉冲）。同时，RX 引脚可同步接收数据。该机制可用于控制带移位寄存器的外设（如 LCD 驱动器）。时钟相位和极性可通过软件编程。

  在智能卡模式下，CK 输出向智能卡提供时钟。

- NSS

  该引脚在同步从模式下用作从设备选择输入。

注： *NSS 和 CTS 共用同一个引脚。*

49.5.3 USART 字符说明

可通过对 USART_CR1 寄存器中的 M 位（M0：位 12，M1：位 28）进行编程来将字长设置为 7 位、8 位或 9 位（请参见*图 565*）：

- 7 位字符长度：M[1:0] = “10”
- 8 位字符长度：M[1:0] = “00”
- 9 位字符长度：M[1:0] = “01”

注： *在 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。*

在默认情况下，信号（TX 或 RX）在起始位工作期间处于低电平状态。在停止位工作期间处于高电平状态。

通过极性配置控制，可以单独针对每个信号对这些值取反。

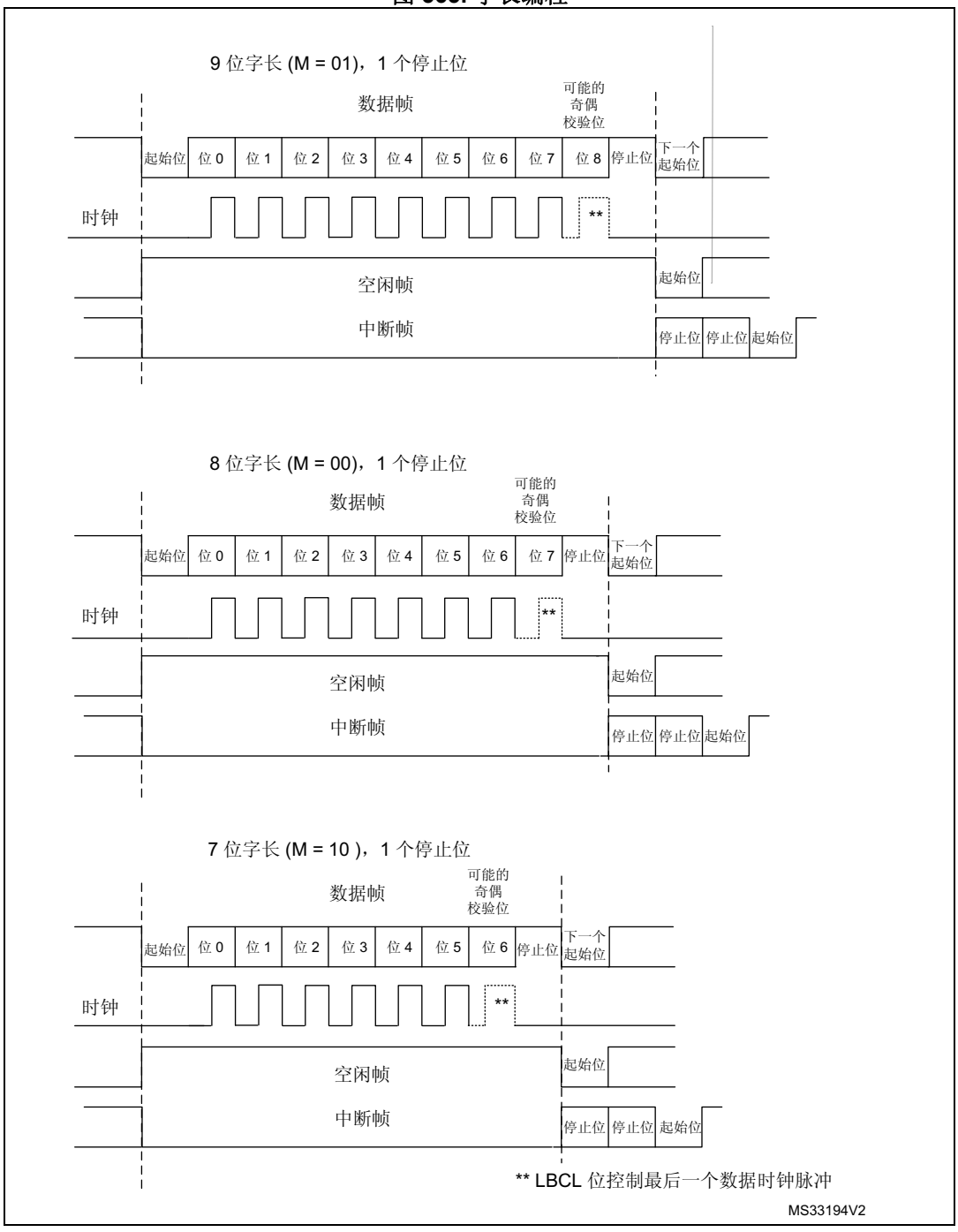
空闲字符可理解为整个帧周期内电平均为“1”（停止位的电平也是“1”）。

刹车字符可理解为在一个帧周期内接收到的电平均为“0”。发送器在中断帧的末尾插入 2 个停止位。

发送和接收操作由通用波特率发生器驱动。当发送器和接收器的使能位置 1 时，将分别生成发送时钟和接收时钟。

下面给出了各个块的详细说明。

图 565. 字长编程


9 位字长 (M = 01)，1 个停止位
数据帧
可能的
奇偶
校验位
起始位
位 0
位 1
位 2
位 3
位 4
位 5
位 6
位 7
位 8
停止位
下一个
起始位
时钟
**
空闲帧
起始位
中断帧
停止位
停止位
起始位
8 位字长 (M = 00)，1 个停止位
数据帧
可能的
奇偶
校验位
起始位
位 0
位 1
位 2
位 3
位 4
位 5
位 6
位 7
停止位
下一个
起始位
时钟
**
空闲帧
起始位
中断帧
停止位
停止位
起始位
7 位字长 (M = 10)，1 个停止位
数据帧
可能的
奇偶
校验位
起始位
位 0
位 1
位 2
位 3
位 4
位 5
位 6
停止位
下一个
起始位
时钟
**
空闲帧
起始位
中断帧
停止位
停止位
起始位
** LBCL 位控制最后一个数据时钟脉冲
MS33194V2

49.5.4 USART FIFO 和阈值

USART 可工作在 FIFO 模式下。

USART 具有一个发送 FIFO (TXFIFO) 和一个接收 FIFO (RXFIFO)。可通过将 USART_CR1 寄存器中的 FIFOEN（位 29）置 1 使能 FIFO 模式。仅 UART、SPI 和智能卡模式下支持该模式。

最大数据字长度为 9 位，因此 TXFIFO 为 9 位宽。不过，RXFIFO 的默认宽度为 12 位。这是因为接收器不仅在 FIFO 中存储数据，而且还存储与每个字符相关的错误标志（奇偶校验错误、噪声错误和帧错误标志）。

注： *接收的数据与相应的标志一起存储在 RXFIFO 中，但读取 RDR 时仅读取数据。*

状态标志位于 USART_ISR 寄存器中。

可以配置触发 Tx 和 RX 中断的 TXFIFO 和 RXFIFO 阈值。这些阈值通过 USART_CR3 控制寄存器中的 RXFTCFG 和 TXFTCFG 位字段进行编程。

在这种情况下：

- 当 RXFIFO 中接收的数据量达到 RXFTCFG 位字段中编程的阈值时，USART_ISR 寄存器中的 RXFT 标志置 1 并生成相应中断（如果使能）。

  这意味着 RXFIFO 被填充，直到 RXFIFO 中的数据量等于编程的阈值。

  已接收到 RXFTCFG 数据：USART_RDR 中有 1 个数据，RXFIFO 中有 (RXFTCFG - 1) 个数据。例如，如果将 RXFTCFG 编程为“101”，则在接收到对应于 FIFO 大小的数据量（RXFIFO 中有（FIFO 大小 - 1）个数据，USART_RDR 中有 1 个数据）时，RXFT 标志将置 1。因此，下一个接收到的数据不会将上溢标志置 1。
- 当 TXFIFO 中空单元数量达到 TXFTCFG 位字段中编程的阈值时，USART_ISR 寄存器中的 TXFT 标志置 1 并生成相应中断（如果使能）。

  这意味着 TXFIFO 被清空，直到 TXFIFO 中的空存储单元量等于编程的阈值。

49.5.5 USART 发送器

发送器可发送 7 位、8 位或 9 位的数据字，具体取决于 M 位的状态。要激活发送器功能，必须将发送使能位 (TE) 置 1。发送移位寄存器中的数据在 TX 引脚输出，相应的时钟脉冲在 CK 引脚输出。

字符发送

USART 发送期间，首先通过 TX 引脚移出数据的最低有效位（默认配置）。在该模式下，USART_TDR 寄存器的缓冲区 (TDR) 位于内部总线和发送移位寄存器之间。

使能 FIFO 模式时，写入到发送数据寄存器 (USART_TDR) 中的数据会在 TXFIFO 中排队。

每个字符前面都有一个起始位，其对应于一个位周期的逻辑低电平。字符由可配置数量的停止位终止。

停止位的数量可配置为 0.5、1、1.5 或 2。

注： *向 USART_TDR 中写入要发送的数据前，TE 位必须先置 1。*

数据发送期间不应复位 TE 位。发送期间复位 TE 位会冻结波特率计数器，进而损坏 TX 引脚上的数据。当前发送的数据随即丢失。

使能 TE 位时，将发送空闲帧。

可配置的停止位

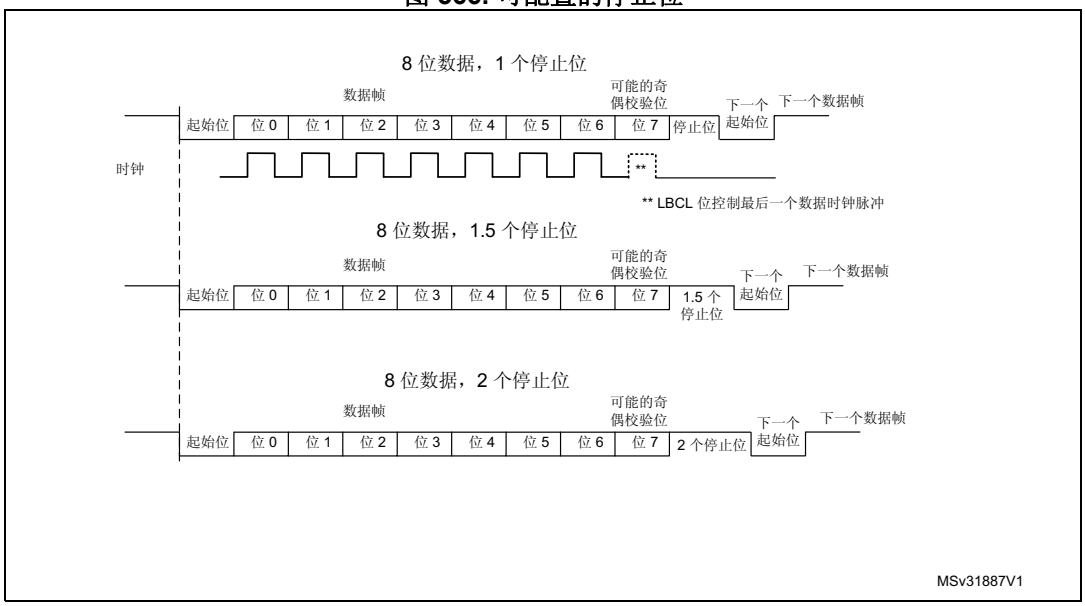
可以在 USART_CR2 的位 13 和位 12 中编程将随各个字符发送的停止位的数量。

- ***1 个停止位***：这是停止位数量的默认值。
- ***2 个停止位***：正常 USART 模式、单线模式和调制解调器模式支持该值。
- ***1.5 个停止位***：用于智能卡模式。

空闲帧发送包括停止位。

刹车发送是 10 个低电平位（M[1:0] = “00”时）或 11 个低电平位（M[1:0] = “01”时）或 9 个低电平位（M[1:0] = “10”时），然后是 2 个停止位（请参见*图 566*）。无法发送长刹车（刹车长度大于 9/10/11 个低电平位）。

图 566. 可配置的停止位



字符发送步骤

要发送字符，需遵循以下步骤：

1. 对 USART_CR1 中的 M 位进行编程以定义字长。
2. 使用 USART_BRR 寄存器选择所需波特率。
3. 对 USART_CR2 中的停止位数量进行编程。
4. 通过向 USART_CR1 寄存器中的 UE 位写入 1 使能 USART。
5. 如果必须进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能 (DMAT)。按照*第 49.5.19 节：使用 USART 和 DMA 进行连续通信*中的说明配置 DMA 寄存器。
6. 将 USART_CR1 中的 TE 位置 1 以便在首次发送时发送一个空闲帧。
7. 在 USART_TDR 寄存器中写入要发送的数据。为每个要在单缓冲区模式下发送的数据重复这一步骤。
   - 禁止 FIFO 模式时，向 USART_TDR 写入数据会将 TXE 标志清零。
   - 使能 FIFO 模式时，向 USART_TDR 写入数据会为 TXFIFO 中增添一个数据。当 TXFNF 标志置 1 时，对 USART_TDR 寄存器的写操作可以执行。该标志会保持置 1，直到 TXFIFO 已满。

8. 将最后一个数据写入 USART_TDR 寄存器后，等待 TC = 1。
   - 禁止 FIFO 模式时，这表示最后一个帧的发送已完成。
   - 使能 FIFO 模式时，这表示 TXFIFO 和移位寄存器均为空。

   当 USART 被禁止或进入暂停模式时，需要执行此检查来避免损坏最后一次发送。

单字节通信

- 禁止 FIFO 模式时

  对发送数据寄存器进行写操作始终会清零 TXE 位。TXE 标志由硬件置 1。它表示：
  - 数据已从 USART_TDR 寄存器移到移位寄存器中且数据发送已开始；
  - USART_TDR 寄存器为空；
  - USART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。

  TXEIE 位置 1 时该标志位会生成中断。

  发送时，写 USART_TDR 寄存器，会将数据存储在 TDR 缓冲区。该数据随后会在当前发送结束时复制到移位寄存器中。

  未发送时，写 USART_TDR 寄存器，会将数据写入移位寄存器，数据发送开始时，TXE 位置 1。
- 使能 FIFO 模式时，TXFNF（TXFIFO 未满）标志由硬件置 1，以指示：
  - TXFIFO 未满；
  - USART_TDR 寄存器为空；
  - USART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。发送时，写 USART_TDR 寄存器，会将数据存储在 TXFIFO。该数据随后会在当前发送结束时从 TXFIFO 复制到移位寄存器中。

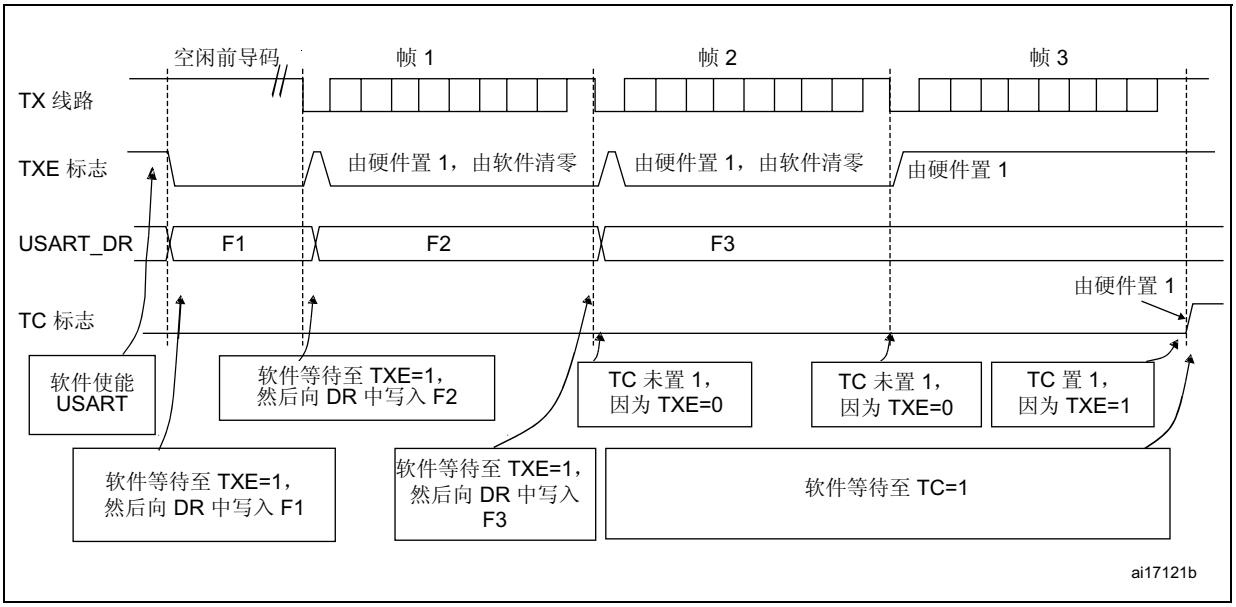
  TXFIFO 未满时，即使在对 USART_TDR 寄存器执行完写操作后，TXFNF 标志也保持为“1”。该标志在 TXFIFO 已满时清零。TXFNFIE 位置 1 时该标志位会生成中断。

  或者，当达到 TXFIFO 阈值时，会生成中断并将数据写入 FIFO。在这种情况下，CPU 可写入由编程的触发阈值定义的数据块。

  如果帧已发送（停止位后）且 TXE 标志（FIFO 模式下的 TXFE）置 1，TC 标志将变为高电平。如果 USART_CR1 寄存器中的 TCIE 位置 1，将生成中断。

向 USART_TDR 寄存器中写入最后一个数据后，必须等待至 TC 置 1，之后才可禁止 USART 或使设备进入低功耗模式（请参见*图 567：发送时的 TC/TXE 行为*）。

图 567. 发送时的 TC/TXE 行为



注： *使能 FIFO 管理时，TXFNF 标志将用于数据发送。*

刹车字符

将 SBKRQ 位置 1 将发送一个刹车字符。中断帧的长度取决于 M 位（请参见*图 565*）。

如果将“1”写入 SBKRQ 位，则当前字符发送完成后，将在 TX 线路上发送一个刹车字符。通过写操作将 SBKF 位置 1 并在刹车字符发送完成时（发送刹车字符后的停止位期间），该位由硬件复位。USART 在中断帧末尾的两位持续时间内插入一个逻辑“1”信号 (STOP)，以确保识别下个帧的起始位。

如果 SBKRQ 位置 1，则在当前发送结束时，会发送一个刹车字符。

使能 FIFO 模式时，即使 TXFIFO 已满，发送刹车字符的优先级也仍高于发送数据的优先级。

空闲字符

将 TE 位置 1 会驱动 USART 在第一个数据帧之前发送一个空闲帧。

49.5.6 USART 接收器

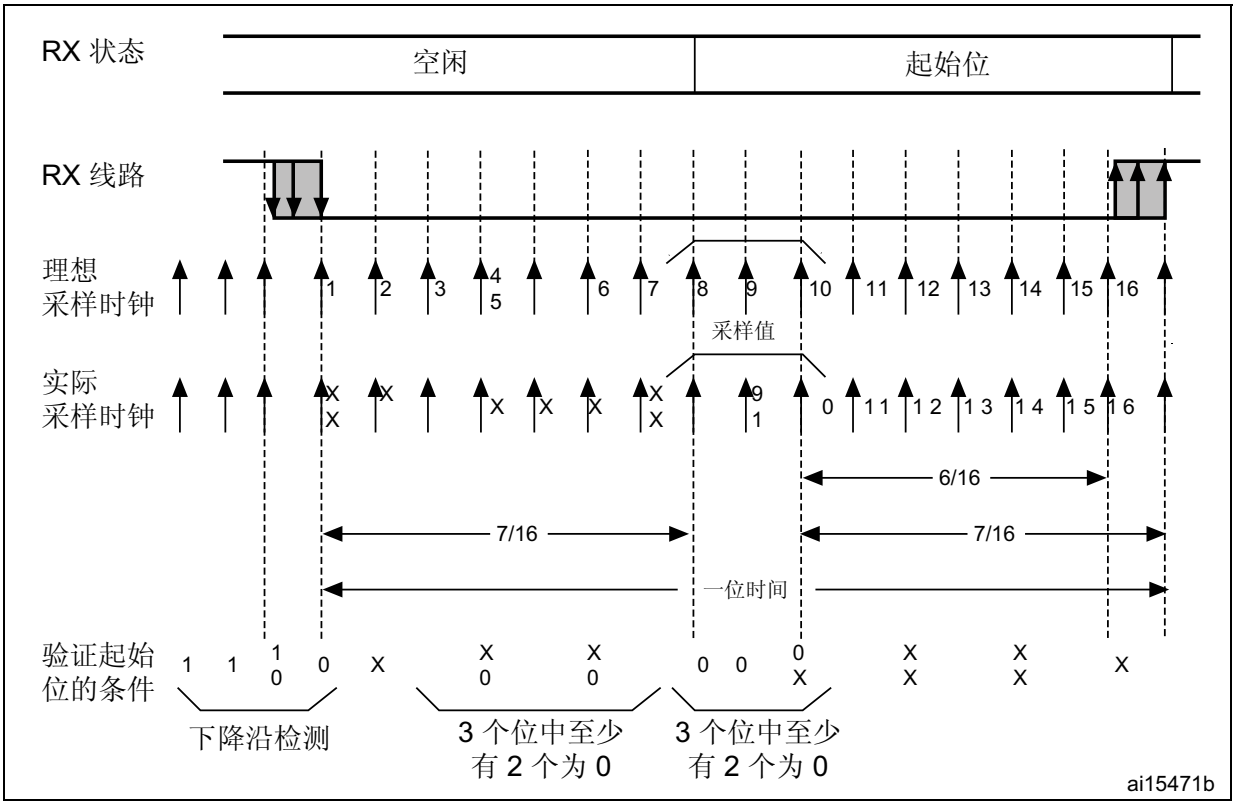
USART 可接收 7 位、8 位或 9 位的数据字，具体取决于 USART_CR1 寄存器中的 M 位。

起始位检测

16 倍或 8 倍过采样时，起始位检测序列相同。

在 USART 中，识别出特定序列的采样时会检测到起始位。该序列为：1 1 1 0 X 0 X 0X 0X 0 X 0X 0。

图 568. 16 倍或 8 倍过采样时的起始位检测



注： *如果序列不完整，起始位检测将中止，接收器将返回空闲状态（无标志位置 1）等待下降沿。*

如果 3 个采样位均为“0”（针对第 3 位、第 5 位和第 7 位进行首次采样时检测到这 3 位均为“0”；针对第 8 位、第 9 位和第 10 位进行第二次采样时仍检测到这 3 位均为“0”），可确认起始位（RXNE 标志置 1 且 RXNEIE = 1 时生成中断；如果使能 FIFO 模式，则 RXFNE 标志置 1 且 RXFNEIE = 1 时生成中断）。

满足以下条件时，可验证起始位但 NE 噪声标志置 1：

a) 对于两次采样，3 个采样位中有 2 位为“0”（针对第 3 位、第 5 位和第 7 位进行采样并且针对第 8 位、第 9 位和第 10 位采样）

或

b) 对于其中一次采样（针对第 3 位、第 5 位和第 7 位进行采样或针对第 8 位、第 9 位和第 10 位进行采样），3 个采样位中有 2 个为“0”。

如果上述条件均不满足，则启动检测中止，接收器返回空闲状态（无标志置 1）。

字符接收

USART 接收期间，首先通过 RX 引脚移出数据的最低有效位（默认配置）。

字符接收步骤

要接收字符，需遵循以下步骤：

1. 对 USART_CR1 中的 M 位进行编程以定义字长。
2. 使用波特率寄存器 USART_BRR 选择所需波特率。
3. 对 USART_CR2 中的停止位数量进行编程。
4. 通过向 USART_CR1 寄存器中的 UE 位写入“1”使能 USART。
5. 如果将进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能 (DMAR)。按照*第 49.5.19 节：使用 USART 和 DMA 进行连续通信*中的说明配置 DMA 寄存器。
6. 将 RE 位 USART_CR1 置 1。这一操作将使能接收器开始搜索起始位。

接收到字符时：

- 如果已禁止 FIFO 模式，则 RXNE 位置 1，这表明移位寄存器的内容已传送到 RDR。也就是说，已接收到并可读取数据（以及其相应的错误标志）。
- 如果已使能 FIFO 模式，则 RXFNE 位置 1，这表示 RXFIFO 非空。读取 USART_RDR 会返回输入到 RXFIFO 中的最早数据。接收到数据时，数据以及相应的错误位将一起被存储在 RXFIFO 中。
- 如果 RXNEIE（使能 FIFO 模式时为 RXFNEIE）位置 1，则会生成中断。
- 如果接收期间已检测到帧错误、噪声错误、奇偶校验错误或上溢错误，错误标志会置 1。
- 在多缓冲区通信模式下：
  - 如果禁止 FIFO 模式，则 RXNE 标志会在每次接收到字节后置 1。该标志在 DMA 读取接收数据寄存器时被清零。
  - 如果使能 FIFO 模式，则 RXFNE 标志在 RXFIFO 非空时置 1。每次收到 DMA 请求后，都会从 RXFIFO 取回数据。当 RXFIFO 非空时（即，当存在要从 RXFIFO 中读取的数据时），会触发 DMA 请求。
- 在单缓冲区模式下：
  - 如果禁止 FIFO 模式，则通过软件对 USART_RDR 寄存器进行读操作来将 RXNE 标志清零。也可以通过将 USART_RQR 寄存器中的 RXFRQ 位编程为“1”来清零 RXNE 标志。RXNE 标志必须在结束接收下一个字符前清零，以避免发生上溢错误。
  - 如果使能 FIFO 模式，则 RXFNE 在 RXFIFO 非空时置 1。每次对 USART_RDR 执行完读操作后，都会从 RXFIFO 中取回数据。RXFIFO 为空时，RXFNE 标志将清零。也可以通过将 USART_RQR 寄存器中的 RXFRQ 位编程为“1”来清零 RXFNE 标志。RXFIFO 已满时，必须在结束接收下一个字符前读取 RXFIFO 中的第一个条目，以避免发生上溢错误。当 RXFNEIE 位置 1 时，RXFNE 标志会生成中断。或者，当达到 RXFIFO 阈值时，会生成中断并从 RXFIFO 中读取数据。在这种情况下，CPU 可读取由编程的阈值定义的数据块。

刹车字符

接收到刹车字符时，USART 将会按照帧错误对其进行处理。

空闲字符

检测到空闲帧时，除了在 IDLEIE 位置 1 时会生成中断外，处理步骤与接收到数据字符的情况相同。

上溢错误

- 禁止 FIFO 模式

  如果在 RXNE 未复位时接收到字符，则会发生上溢错误。

  RXNE 位清零前，数据无法从移位寄存器传送到 RDR 寄存器。每接收到一个字节后，RXNE 标志都将置 1。

  当 RXNE 标志位置 1 时，如果在接收到下一个数据或尚未处理上一个 DMA 请求，则会发生上溢错误。发生上溢错误时：

  – ORE 位置 1。
  – RDR 中的内容不会丢失。可通过读取 USART_RDR 寄存器获得之前的数据。
  – 移位寄存器会被覆盖。之后，上溢期间接收到的任何数据都将丢失。
  – 如果 RXNEIE 或 EIE 位置 1，则会生成中断。

- 使能 FIFO 模式

  移位寄存器准备好传送并且接收 FIFO 已满时，会发生上溢错误。

  在 RXFIFO 中出现一个空闲位置之前，数据无法从移位寄存器传送到 USART_RDR 寄存器。当 RXFIFO 非空时，RXFNE 标志置 1。

  如果 RXFIFO 已满且移位寄存器已准备好传送，则会发生上溢错误。发生上溢错误时：

  – ORE 位置 1。
  – RXFIFO 中的第一个条目不会丢失。通过读取 USART_RDR 寄存器可获得此条目。
  – 移位寄存器会被覆盖。之后，上溢期间接收到的任何数据都将丢失。
  – 如果 RXFNEIE 或 EIE 位置 1，则会生成中断。

通过将 USART_ICR 寄存器中的 ORECF 位置 1 来复位 ORE 位。

注： *ORE 位置 1 时表示至少 1 个数据丢失。*

禁止 FIFO 模式时，有以下两种可能

- *如果 RXNE = 1，则最后一个有效数据存储于接收寄存器 (RDR) 中并且可进行读取，*
- *如果 RXNE = 0，则最后一个有效数据已被读取，因此 RDR 寄存器中没有要读取的数据。接收到新（丢失）数据的同时已读取 RDR 寄存器中的最后一个有效数据时，会发生该情况。*

选择时钟源和合适的过采样方法

时钟源的选择通过时钟控制系统完成（请参见*复位和时钟控制 (RCC)* 部分）。在使能 USART 之前，必须通过 UE 位选择时钟源。

必须遵循以下两个条件选择时钟源：

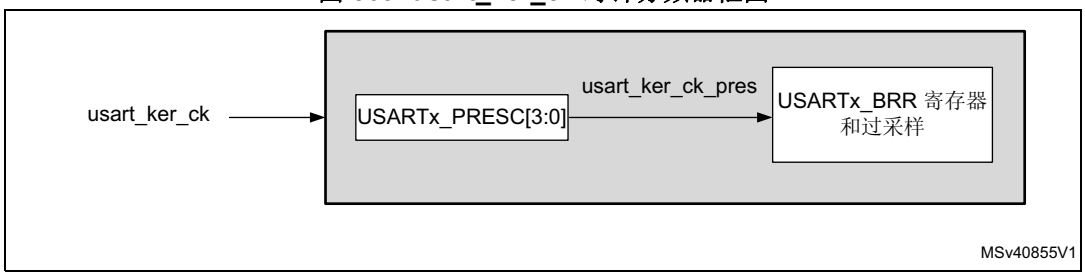
- 可在低功耗模式下使用 USART
- 通信速度。

时钟源频率为 usart_ker_ck。

如果支持双时钟域和从低功耗模式唤醒功能，则 usart_ker_ck 时钟源可在 RCC 中进行配置（请参见*复位和时钟控制 (RCC)* 部分）。否则，usart_ker_ck 时钟与 usart_pclk 相同。

usart_ker_ck 时钟可根据可编程系数（在 USART_PRESC 寄存器中定义）进行分频。

图 569. usart_ker_ck 时钟分频器框图



某些 usart_ker_ck 时钟源允许 USART 在 MCU 处于低功耗模式时接收数据。需要时，USART 可基于所接收的数据和选择的唤醒模式来唤醒 MCU，以通过用软件读取 USART_RDR 寄存器的方式或通过 DMA 的方式传输已接收数据。

对于其他时钟源，系统必须激活才能使能 USART 通信。

时钟源还决定通信速度范围（尤其是最大通信速度）。

接收器采用不同的用户可配置过采样技术（除了同步模式下），可以从噪声中提取有效数据。这可在最大通信速度与抗噪声/时钟误差性能之间实现最佳平衡。

可通过编程 USART_CR1 寄存器中的 OVER8 位来选择采样方法，且采样时钟可以是波特率时钟的 16 倍或 8 倍（请参见*图 570* 和*图 571*）。

根据应用：

- 选择 8 倍过采样 (OVER8 = 1) 以获得更高的速度（高达 usart_ker_ck_pres/8）。这种情况下接收器对时钟偏差的最大容差将会降低（请参见*第 2453 页的第 49.5.8 节：USART 接收器对时钟偏差的容差*）
- 选择 16 倍过采样 (OVER8 = 0) 以增加接收器对时钟偏差的容差。在这种情况下，最大速度被限制为最大 usart_ker_ck_pres/16（其中，usart_ker_ck_pres 为 USART 输入时钟通过预分频器进行分频得到的值）。

可通过编程 USART_CR3 寄存器中的 ONEBIT 位选择用于评估逻辑电平的方法。有两种选择可供使用：

- 在已接收位的中心进行三次采样，从而进行多数表决。这种情况下，如果用于多数表决的 3 次采样结果不相等，NE 位置 1。
- 在已接收位的中心进行单次采样

  根据应用：

  – 在噪声环境下工作时，请选择三次采样的多数表决法 (ONEBIT = 0)；在检测到噪声时请拒绝数据（请参见*图 439*），因为这表示采样过程中产生了干扰。

  – 线路无噪声时请选择单次采样法 (ONEBIT = 1) 以增加接收器对时钟偏差的容差（请参见*第 2453 页的第 49.5.8 节：USART 接收器对时钟偏差的容差*）。这种情况下 NE 位始终不会置 1。

帧中检测到噪声时：

- 在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）的上升沿时 NE 位置 1。
- 无效数据从移位寄存器传送到 USART_RDR 寄存器。
- 单字节通信时无中断产生。然而，在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）生成中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时会发出中断。

通过将 USART_ICR 寄存器中的 NECF 位置 1 复位 NE 位。

注： *SPI 模式不支持噪声错误。*

智能卡、IrDA 和 LIN 模式下不可采用 8 倍过采样。在这些模式下，OVER8 位由硬件强制清零。

图 570. 16 倍过采样时的数据采样

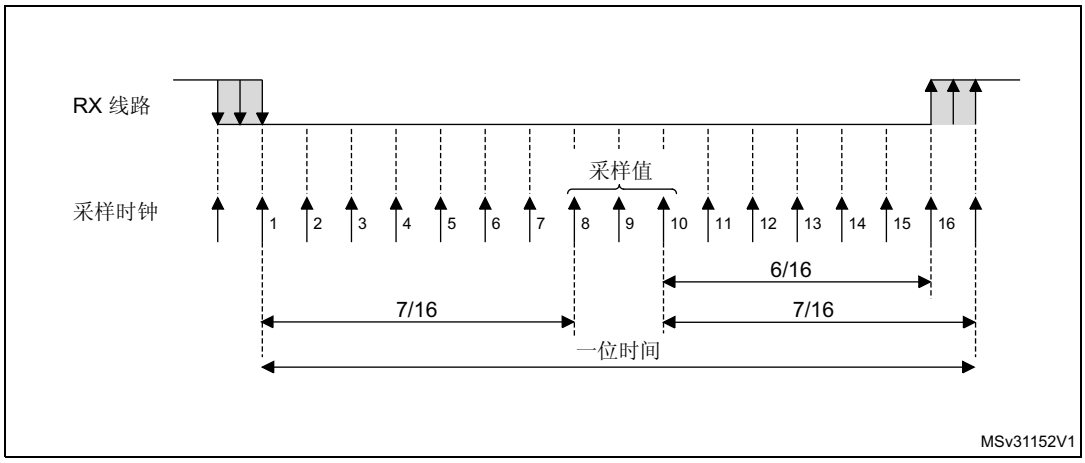


图 571. 8 倍过采样时的数据采样

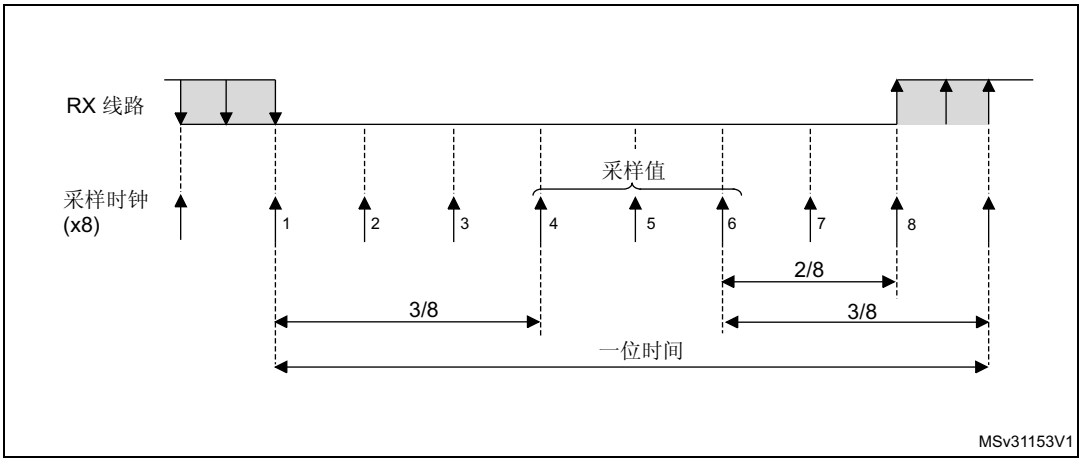


表 439. 通过采样数据进行噪声检测

采样值	NE 状态	接收的位值
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

帧错误

如果接收数据时未在预期时间内识别出停止位，从而出现同步失效或过度的噪声，则会检测到帧错误。

检测到帧错误时：

- FE 位由硬件置 1
- 无效数据从移位寄存器传送到 USART_RDR 寄存器（使能 FIFO 模式时为 RXFIFO）。
- 单字节通信时无中断产生。然而，在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）生成中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时会发出中断。

通过将“1”写入 USART_ICR 寄存器中的 FECF 位来复位 FE 位。

注： *SPI 模式不支持帧错误。*

接收期间可配置的停止位

可通过 USART_CR 的控制位配置要接收的停止位的数量：可以是 1 或 2 个（正常模式下），也可以是 0.5 或 1.5 个（智能卡模式下）。

- **0.5 个停止位（在智能卡模式下接收时）**：不会对 0.5 个停止位进行采样。结果，选择 0.5 个停止位时，无法检测到帧错误和中断帧。
- **1 个停止位**：将在第 8、第 9 和第 10 次采样时对 1 个停止位进行采样。
- 1.5 个停止位（在智能卡模式下）

  在智能卡模式下发送时，设备必须检查数据是否正确发送。因此，必须使能接收器块（USART_CR1 中的 RE = 1）并检查停止位，以测试智能卡是否已检测到奇偶校验错误。

  发生奇偶校验错误时，智能卡会在采样时将数据信号强制为低电平（即 NACK 信号），该信号被标记为帧错误。之后，FE 标志在 1.5 个停止位的末尾由 RXNE 标志（使能 FIFO 模式时为 RXFNE）置 1。在第 16、第 17 和第 18 次采样时对 1.5 个停止位进行采样（停止位采样开始后维持 1 个波特时钟周期）。1.5 个停止位可分为 2 个部分：0.5 个波特时钟周期（未发生任何动作），然后是 1 个正常的停止位周期（一半时间处进行采样）（更多详细信息，请参见*第 2465 页的第 49.5.16 节：USART 接收器超时*）。
- 2 个停止位

  采样 2 个停止位时在第 8、第 9 和第 10 次采样时对第一个停止位进行采样。

  如果在第一个停止位期间检测到帧错误，则帧错误标志会置 1。

  发生帧错误时不检测第 2 个停止位。RXNE 标志（使能 FIFO 模式时为 RXFNE）将在第一个停止位结束时置 1。

49.5.7 USART 波特率生成

接收器和发送器（Rx 和 Tx）的波特率均设置为 USART_BRR 寄存器中编程的值。

公式 1：适用于标准 USART（包括 SPI 模式）的波特率（OVER8 = “0”或“1”）

在 16 倍过采样的情况下，波特率通过以下公式得出：

$$\text{Tx/Rx 波特率} = \frac{\text{usart_ker_ckpres}}{\text{USARTDIV}}$$

在 8 倍过采样的情况下，波特率通过以下公式得出：

$$\text{Tx/Rx 波特率} = \frac{2 \times \text{usart_ker_ckpres}}{\text{USARTDIV}}$$

公式 2：智能卡、LIN 和 IrDA 模式下的波特率 (OVER8 = 0)

波特率可根据以下公式得出：

$$\text{Tx/Rx 波特率} = \frac{\text{usart_ker_ckpres}}{\text{USARTDIV}}$$

USARTDIV 是一个存放在 USART_BRR 寄存器中的无符号定点数。

- 当 OVER8 = 0 时，BRR = USARTDIV。
- 当 OVER8 = 1 时
  - BRR[2:0] = USARTDIV[3:0] 右移 1 位。
  - BRR[3] 必须保持清零。
  - BRR[15:4] = USARTDIV[15:4]

注： *对 USART_BRR 执行写操作后，波特率计数器更新为波特率寄存器中的新值。因此，波特率寄存器的值不应在通信时发生更改。*

16 倍和 8 倍过采样时，USARTDIV 必须大于或等于 16。

如何从 USART_BRR 寄存器中获取 USARTDIV

示例 1

要通过 usart_ker_ck_pres = 8 MHz 获得 9600 波特：

- 16 倍过采样时：
  USARTDIV = 8 000 000/9600
  BRR = USARTDIV = 0d833 = 0x0341
- 8 倍过采样时：
  USARTDIV = 2 * 8 000 000/9600
  USARTDIV = 1666,66 (0d1667 = 0x683)
  BRR[3:0] = 0x3 >> 1 = 0x1
  BRR = 0x681

示例 2

要通过 usart_ker_ck_pres = 48 MHz 获得 921.6 K 波特：

- 16 倍过采样时：
  USARTDIV = 48 000 000/921 600
  BRR = USARTDIV = 0d52 = 0x34
- 8 倍过采样时：
  USARTDIV = 2 * 48 000 000/921 600
  USARTDIV = 104 (0d104 = 0x68)
  BRR[3:0] = USARTDIV[3:0] >> 1 = 0x8 >> 1 = 0x4
  BRR = 0x64

49.5.8 USART 接收器对时钟偏差的容差

仅当总时钟系统偏差小于 USART 接收器的容差时，USART 异步接收器才能正常工作。

影响总偏差的因素包括：

- DTRA：发送器误差引起的偏差（其中还包括发送器本地振荡器的偏差）
- DQUANT：接收器的波特率量化引起的误差
- DREC：接收器本地振荡器的偏差
- DTCL：发送线路引起的偏差（通常是由于收发器所引起，它可能会在低电平到高电平转换时序与高电平到低电平转换时序之间引入不对称）

DTRA + DQUANT + DREC + DTCL + DWU < USART 接收器容差

其中

DWU 是使用从低功耗模式唤醒时因采样点偏差而产生的误差。

M[1:0] = 01 时：

$$DWU = \frac{t_{WUUSART}}{11 \times Tbit}$$

M[1:0] = 00 时：

$$DWU = \frac{t_{WUUSART}}{10 \times Tbit}$$

M[1:0] = 10 时：

$$DWU = \frac{t_{WUUSART}}{9 \times Tbit}$$

$t_{WUUSART}$ 是检测到起始位下降沿与时钟（由外设请求）就绪、达到外设且调压器就绪之间的时间。

USART 接收器在*表 440* 和*表 441* 中指定的最大容许偏差下可正确接收数据，具体取决于以下设置：

- 由 USART_CR1 寄存器中的 M 位定义的 9 位、10 位或 11 位字符长度
- 由 USART_CR1 寄存器中的 OVER8 位定义的 8 倍或 16 倍过采样
- USART_BRR 寄存器的 BRR[3:0] 位等于或不等于 0000
- 使用 1 位或 3 位对数据进行采样，取决于 USART_CR3 寄存器中 ONEBIT 位的值

表 440. BRR [3:0] = 0000 时的 USART 接收器容差

M 位	OVER8 位 = 0		OVER8 位 = 1	
	ONEBIT = 0	ONEBIT = 1	ONEBIT = 0	ONEBIT = 1
00	3.75%	4.375%	2.50%	3.75%
01	3.41%	3.97%	2.27%	3.41%
10	4.16%	4.86%	2.77%	4.16%

表 441. BRR[3:0] 不等于 0000 时的 USART 接收器容差

M 位	OVER8 位 = 0		OVER8 位 = 1	
	ONEBIT = 0	ONEBIT = 1	ONEBIT = 0	ONEBIT = 1
00	3.33%	3.88%	2%	3%
01	3.03%	3.53%	1.82%	2.73%
10	3.7%	4.31%	2.22%	3.33%

注： *当接收的帧恰好包含 10 个（M 位 = 00）、11 个（M 位 = 01）或 9 个（M 位 = 10）位时间的空闲帧时，表 440 和表 441 中指定的数据可能与特例中的数据略微不同。*

49.5.9 USART 自动波特率检测

USART 可根据接收一个字符检测并自动设置 USART_BRR 寄存器的值。自动波特率检测在以下两种情况下非常有用：

- 事先不知道系统的通信速度。
- 系统正在使用精确度相对较低的时钟源且该机制允许在不测量时钟偏差的情况下获得正确的波特率。

时钟源频率必须与预期通信速度兼容。

- 16 倍过采样时，波特率范围为 usart_ker_ck_pres/65535 到 usart_ker_ck_pres/16。
- 8 倍过采样时，波特率范围为 usart_ker_ck_pres/65535 到 usart_ker_ck_pres/8。

在激活自动波特率检测之前，必须通过 USART_CR2 寄存器中的 ABRMOD[1:0] 位字段选择自动波特率检测模式。根据不同的字符模式，存在四种检测模式。在这些自动波特率模式下，波特率在同步接收数据期间被多次测量，每次测量的结果都与前一次进行比较。

这些模式如下：

- **模式 0**：以“1”位开头的任意字符。

  这种情况下，USART 会测量起始位的持续时间（下降沿到上升沿）。
- **模式 1**：以 10xx 位模式开头的任何字符。

  这种情况下，USART 会测量起始位和第一个数据位的持续时间。测量在下降沿到下降沿期间完成，可在信号斜率较小时确保较高的精度。
- **模式 2**：0x7F 字符帧（可以是 LSB 在前模式下的 0x7F 字符，也可以是 MSB 在前模式下的 0xFE 字符）。

  这种情况下，先在起始位结束时更新波特率 (BR)，然后在位 6 结束时更新波特率（根据从下降沿到下降沿执行的测量：BR6）。以 BR 对位 0 到位 6 进行采样，而以 BR6 对字符的其他位进行采样。
- **模式 3**：0x55 字符帧。

  这种情况下，先在起始位结束时更新波特率 (BR)，然后在位 0 结束时更新波特率（根据从下降沿到下降沿执行的测量：BR0），最后在位 6 结束时更新波特率 (BR6)。以 BR 对位 0 进行采样，以 BR0 对位 1 到位 6 进行采样，以 BR6 对字符的其他位进行采样。同时，对 RX 线路的各个中间跳变执行其他检查。如果 RX 上的转换与接收器（基于根据位 0 计算的波特率的接收器）未充分同步，则生成错误。

激活自动波特率检测之前，必须先通过向 USART_BRR 寄存器写入非零的波特率值来初始化该寄存器。

通过将 USART_CR2 寄存器中的 ABREN 位置 1 来激活自动波特率检测。之后 USART 会等待 RX 线路上的第一个字符。通过将 USART_ISR 寄存器中的 ABRF 标志置 1 来指示自动波特率操作完成。如果线路有噪声，则无法保证正确的波特率检测。这种情况下，BRR 值可能会损坏，ABRE 错误标志位会置 1。如果通信速度不在自动波特率检测范围（位持续时间不在 16 个和 65536 个时钟周期（16 倍过采样时）之间，也不在 8 个和 65536 个时钟周期（8 倍过采样时）之间）内，也会出现这种情况。

稍后，可通过复位 ABRF 标志（通过写入“0”）重新启动自动波特率检测。

禁止 FIFO 管理且发生自动波特率错误时，ABRE 标志通过 RXNE 和 FE 位置 1。

使能 FIFO 管理且发生自动波特率错误时，ABRE 标志通过 RXFNE 和 FE 位置 1。

如果使能 FIFO 模式，则应使用第一个 RXFIFO 位置上的数据进行自动波特率检测。因此，在启动自动波特率检测之前，请通过检查 USART_ISR 寄存器的 RXFNE 标志来确保 RXFIFO 为空。

注： *如果在自动波特率操作期间禁止 USART (UE = 0)，则可能损坏 BRR 值。*

49.5.10 USART 多处理器通信

可以执行 USART 多处理器通信（多个 USART 连接在一个网络中）。例如，其中一个 USART 可以是主 USART，其 TX 输出与其他 USART 的 RX 输入相连；而其他 USART 为从 USART，其各自的 TX 输出在逻辑上通过与运算连在一起，并与主 USART 的 RX 输入相连。

在多处理器配置中，理想情况下通常只有预期的消息接收方主动接收完整的消息内容，从而减少由所有未被寻址的接收器造成的冗余 USART 服务开销。

可通过静默功能将未被寻址的设备置于静默模式下。为了使用静默模式功能，必须将 USART_CR1 寄存器中的 MME 位置 1。

注： *使能 FIFO 管理且 MME 已置 1 时，不得清零 MME 位再快速将其置 1（在两个 usart_ker_ck 周期内），否则静默模式可能保持有效。*

使能静默模式时：

- 不得将接收状态位置 1。
- 禁止任何接收中断。
- USART_ISR 寄存器中的 RWU 位置“1”。在某些情况下，RWU 可以由硬件或软件通过 USART_RQR 寄存器中的 MMRQ 位自动控制。

根据 USART_CR1 寄存器中 WAKE 位的设置，USART 可使用以下两种方法进入或退出静默模式：

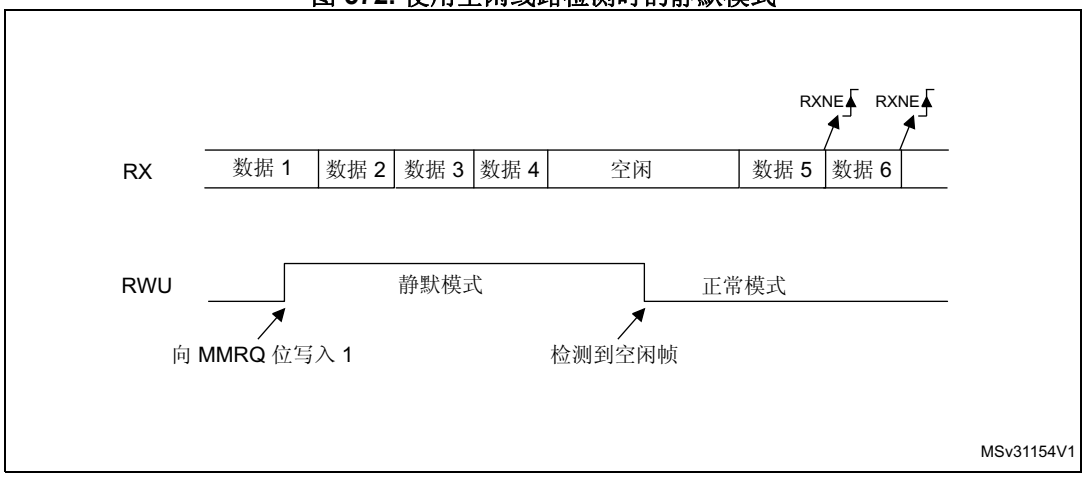
- 如果 WAKE 位被复位，则进行空闲线路检测，
- 如果 WAKE 位置 1，则进行地址标记检测。

空闲线路检测 (WAKE = 0)

向 MMRQ 位写入“1”且 RWU 位自动置 1 时，USART 进入静默模式。

检测到空闲帧时，USART 将唤醒。此时 RWU 位会由硬件清零，但 USART_ISR 寄存器中的 IDLE 位不会置 1。*图 572* 中给出了使用空闲线路检测时静默模式行为的示例。

图 572. 使用空闲线路检测时的静默模式



注： *如果在 IDLE 字符已经过去时将 MMRQ 位置 1，则不会进入静默模式（RWU 未置 1）。*

如果在线路处于空闲状态时激活 USART，在一个 IDLE 帧持续时间后（不只在接收一个字符帧后）会检测到空闲状态。

4 位/7 位地址标记检测 (WAKE = 1)

在此模式下，如果字节的 MSB 为 1，则将这些字节识别为地址，否则将其识别为数据。在地址字节中，目标接收器的地址位于 4 个或 7 个 LSB 中。7 位或 4 位地址检测通过 ADDM7 位来选择。接收器会将此 4 位/7 位字与其地址进行比较，该接收器的地址在 USART_CR2 寄存器的 ADD 位中进行设置。

注： *在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0] 和 ADD[7:0]）。*

当接收到与其编程地址不匹配的地址字符时，USART 会进入静默模式。此时，RWU 位将由硬件置 1。USART 进入静默模式后，RXNE 标志不会针对此地址字节置 1，也不会发出中断或 DMA 请求。使能 FIFO 管理时，软件应确保在进入静默模式之前 RXFIFO 中至少有一个空位置。

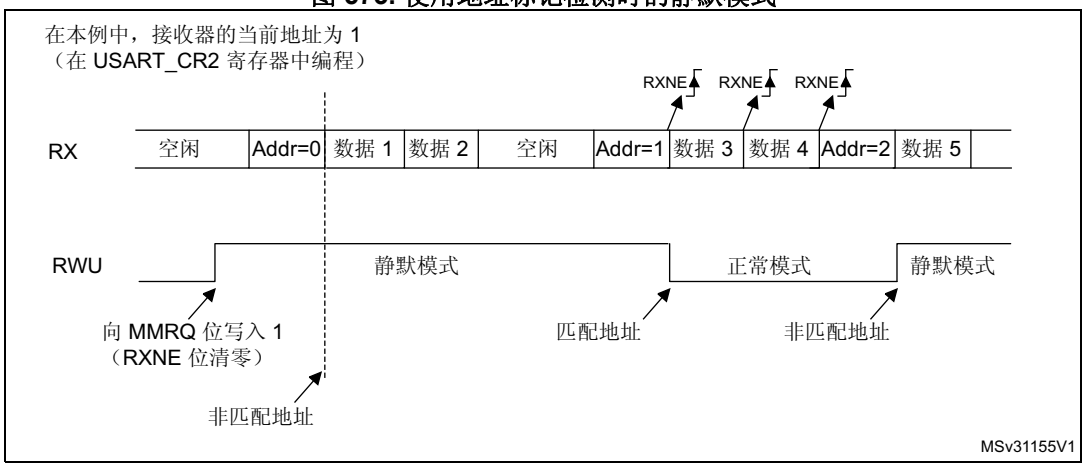
当向 MMRQ 位写入 1 时，USART 也会进入静默模式。这种情况下，RWU 位也自动置 1。

当接收到与编程地址匹配的地址字符时，USART 会退出静默模式。然后 RWU 位被清零，可以开始正常接收后续字节。由于 RWU 位已清零，RXNE/RXFNE 位会针对地址字符置 1。

注： *使能 FIFO 管理时，如果在接收器对数据的最后一位进行采样时 MMRQ 置 1，则可在有效进入静默模式之前接收该数据*

图 573 中给出了使用地址标记检测时静默模式行为的示例。

图 573. 使用地址标记检测时的静默模式



49.5.11 USART Modbus 通信

USART 为 Modbus/RTU 和 Modbus/ASCII 协议的实现提供基本支持。Modbus/RTU 是一个半双工块传输协议。该协议的控制部分（地址识别、块完整性控制和命令解析）必须用软件实现。

USART 为块结束检测提供基本支持，无需软件开销或其他资源。

Modbus/RTU

在此模式下，一个块的结束通过超过 2 个字符时间的“静默”（空闲线路）来识别。此功能通过可编程的超时功能实现。

超时功能和中断必须分别通过 USART_CR2 寄存器中的 RTOEN 位和 USART_CR1 寄存器中的 RTOIE 位激活。与 2 个字符时间（例如 22 个位时间）的超时相对应的值必须在 RTO 寄存器中编程。如果在此期间接收线路空闲，则在接收到最后一个停止位后，将生成中断，同时通知软件当前块接收已完成。

Modbus/ASCII

在此模式下，块结束通过特定 (CR/LF) 字符序列识别。USART 通过字符匹配功能管理此机制。

通过在 ADD[7:0] 位字段中编程 LF ASCII 码以及激活字符匹配中断 (CMIE = 1)，软件可在接收到 LF 时获得通知并检查 DMA 缓存区中的 CR/LF。

49.5.12 USART 极性控制

将 USART_CR1 寄存器中的 PCE 位置 1，可以使能奇偶校验控制（发送时生成奇偶校验位，接收时进行奇偶校验检查）。根据 M 位定义的帧长度，*表 442* 中列出了可能的 USART 帧格式。

表 442. USART 帧格式

M 位	PCE 位	USART 帧(1)
00	0	| SB | 8 位数据 | STB |
00	1	| SB | 7 位数据 | PB | STB |
01	0	| SB | 9 位数据 | STB |
01	1	| SB | 8 位数据 PB | STB |
10	0	| SB | 7 位数据 | STB |
10	1	| SB | 6 位数据 | PB | STB |

1. 图注：SB：起始位，STB：停止位，PB：奇偶校验位。在数据寄存器中，PB 始终位于 MSB 位置（第 8 位或第 7 位，具体取决于 M 位的值）。

偶校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为偶数（帧由 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据 = 00110101 且 4 个位置 1，则在选择偶校验（USART_CR1 寄存器中的 PS 位 = 0）时，奇偶校验位等于 0。

奇校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为奇数（帧由 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据 = 00110101 且 4 个位置 1，则在选择奇校验（USART_CR1 寄存器中的 PS 位 = 1）时，奇偶校验位等于 1。

接收时进行奇偶校验检查

如果奇偶校验检查失败，则 USART_ISR 寄存器中的 PE 标志置 1；如果 USART_CR1 寄存器中 PEIE 位置 1，则会生成中断。通过软件将 1 写入 USART_ICR 寄存器中的 PECF 位来清零 PE 标志。

发送时的奇偶校验生成

如果 USART_CR1 中的 PCE 位置 1，则在数据寄存器中所写入数据的 MSB 位会进行传送，但是会由奇偶校验位进行更改（如果选择偶校验（PS = 0），则“1”的数量为偶数；如果选择奇校验 (PS=1)，则“1”的数量为奇数）。

49.5.13 USART LIN（局域互连网络）模式

仅在支持 LIN 模式时才与本节相关。请参见*第 2436 页的第 49.4 节：USART 实现*。

通过将 USART_CR2 寄存器中的 LINEN 位置 1 来选择 LIN 模式。在 LIN 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 CLKEN 位，
- USART_CR3 寄存器中的 STOP[1:0]、SCEN、HDSEL 和 IREN 位。

LIN 发送

与正常的 USART 发送相比，在 LIN 主设备中发送时必须采用*第 49.5.4 节*中介绍的步骤，同时还具有以下区别：

- M 位清零以配置 8 位字长度。
- LINEN 位置 1 以进入 LIN 模式。此时，将 SBKRQ 位置 1 会发送 13 个“0”位作为刹车字符。然后会发送值为“1”的两个位以进行下一启动检测。

LIN 接收

使能 LIN 模式后，将激活刹车检测电路。该检测完全独立于正常的 USART 接收器。在空闲状态或某个帧期间，只要发生刹车即可检测出来。

接收器（USART_CR1 寄存器中 RE = 1）使能后，电路便开始监测 RX 输入上的启始信号。检测起始位的方法与搜索刹车字符或数据的方法相同。检测到起始位后，电路会对接下来的位进行采样，方法与数据采样相同（第 8、第 9 和第 10 次采样）。如果 10 个（USART_CR2 寄存器中 LBDL = 0 时）或 11 个（USART_CR2 寄存器中 LBDL = 1 时）连续位均检测为“0”，且其后跟随分隔符，则 USART_ISR 寄存器中的 LBDF 标志将会置 1。如果 LBDIE 位 = 1，则会生成中断。在验证中断前，会对分隔符进行检查，因为它表示 RX 线路已恢复到高电平。

如果在第 10 或第 11 次采样前采样到“1”信号，则刹车检测电路会取消当前检测，并重新搜索起始位。

如果禁止 LIN 模式 (LINEN = 0)，接收器会作为正常的 USART 继续工作，不会再进行中断检测。

如果使能 LIN 模式 (LINEN = 1)，只要发生帧错误（例如，在“0”处检测到停止位，这种情况可能出现在任何中断帧中），接收器即会停止，直到中断检测电路接收到“1”（中断字不完整时）或接收到分隔符（检测到中断时）为止。

*第 2460 页的图 574：LIN 模式下的刹车检测（11 位刹车长度——LBDL 位置 1）*中显示了中断检测器状态机和中断标志的行为。

*第 2461 页的图 575：LIN 模式下的刹车检测与帧错误检测*中列出了中断帧的示例。

图 574. LIN 模式下的刹车检测（11 位刹车长度——LBDL 位置 1）

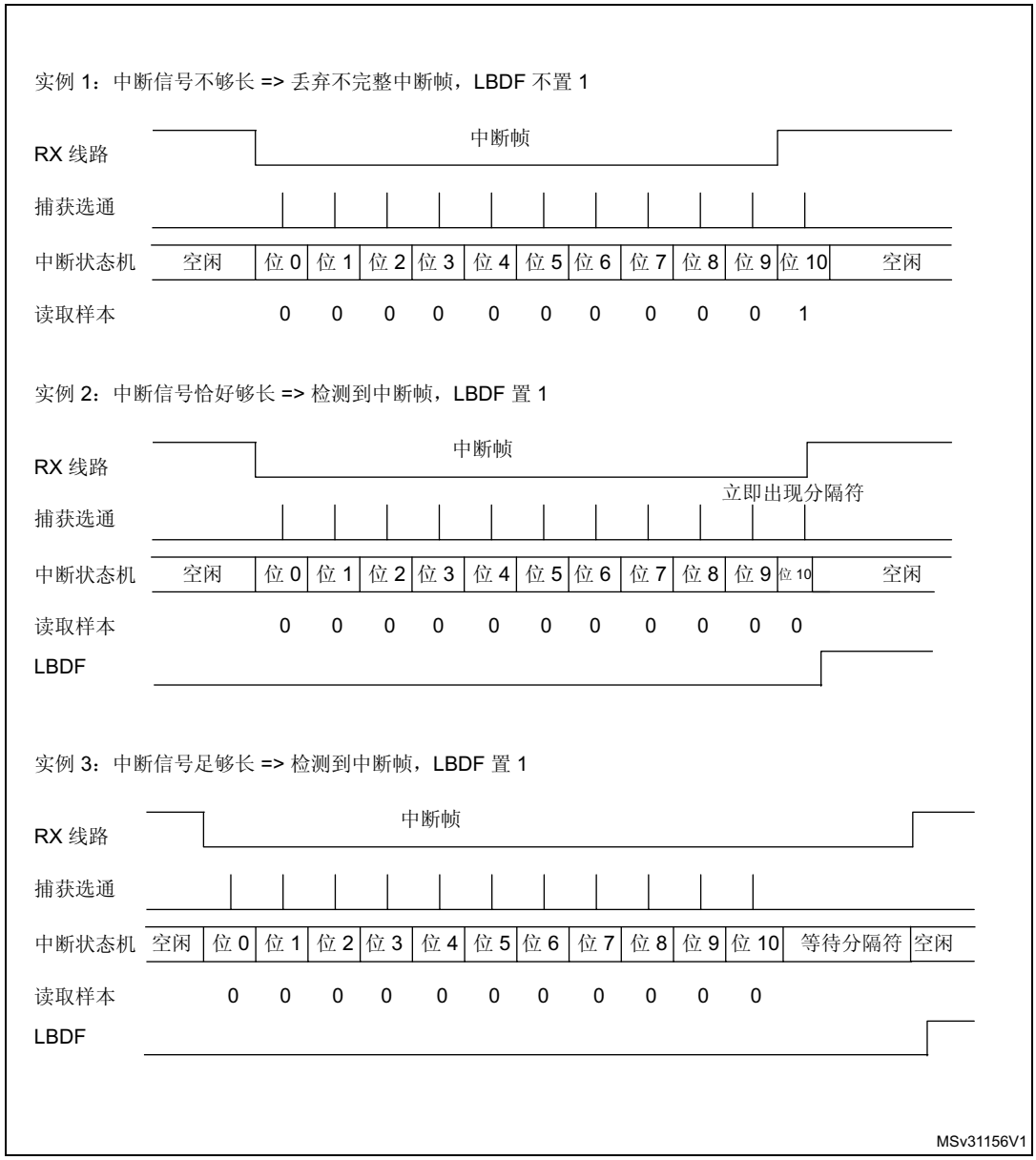
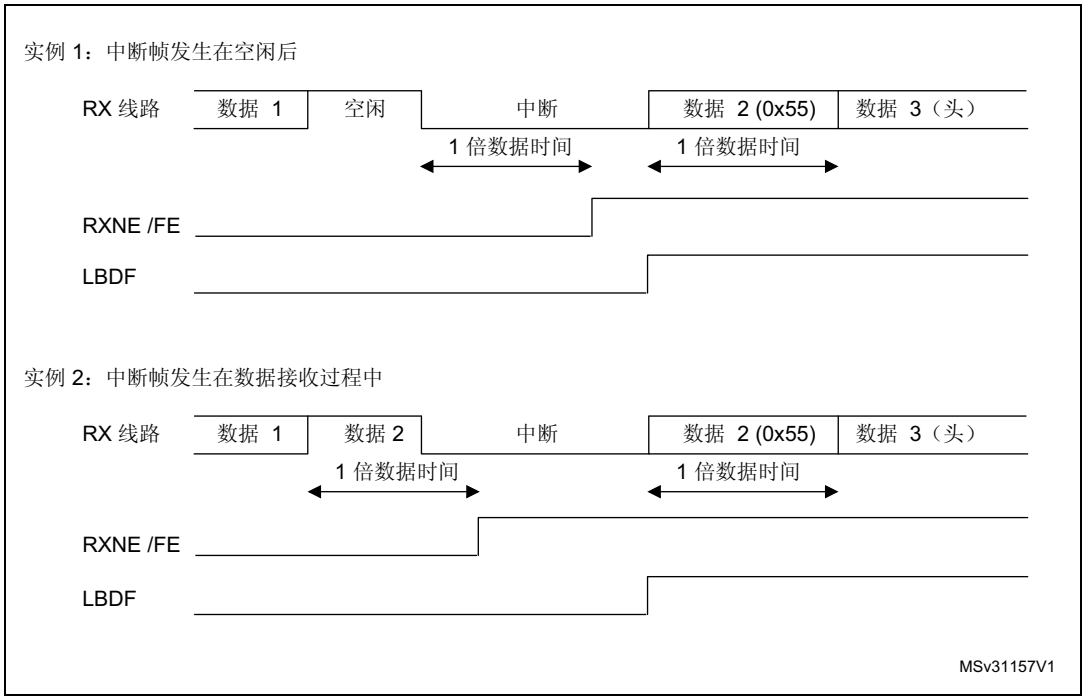


图 575. LIN 模式下的刹车检测与帧错误检测



49.5.14 USART 同步模式

主模式

通过将 USART_CR2 寄存器中的 CLKEN 位编程为“1”来选择同步主模式。在同步模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 位，
- USART_CR3 寄存器中的 SCEN、HDSEL 和 IREN 位。

在此模式下，USART 可用于在主模式下控制双向同步串行通信。CK 引脚是 USART 发送器时钟的输出。在起始位或停止位期间，不会向 CK 引脚发送时钟脉冲。在最后一个有效数据位（地址标记）期间，是否生成时钟脉冲取决于 USART_CR2 寄存器中 LBCL 位的状态。USART_CR2 寄存器中的 CPOL 位用于选择时钟极性；USART_CR2 寄存器中的 CPHA 位用于选择外部时钟的相位（请参见*图 576*、*图 577* 和*图 578*）。

在空闲状态、前导码模式和发送刹车期间，外部 CK 时钟处于未激活状态。

在同步主模式下，USART 发送器的工作方式与异步模式下完全相同。但是，由于 CK 与 TX 同步（根据 CPOL 和 CPHA），因此 TX 上的数据是同步的。

在同步主模式下，USART 接收器的工作方式与异步模式下不同。如果 RE 置 1，则数据在 CK 上采样（上升或下降沿，具体取决于 CPOL 和 CPHA），而不会进行任何过采样。此时必须确保给定建立时间和保持时间（取决于波特率：1/16 位时间）符合要求。

注： *在主模式下，CK 引脚可与 TX 引脚结合使用。因此，仅当使能发送器 (TE = 1) 且正在发送数据时（USART_TDR 数据寄存器已写入），才会提供时钟。这意味着，没有发送数据的情况下无法接收同步数据。*

图 576. USART 同步主发送示例

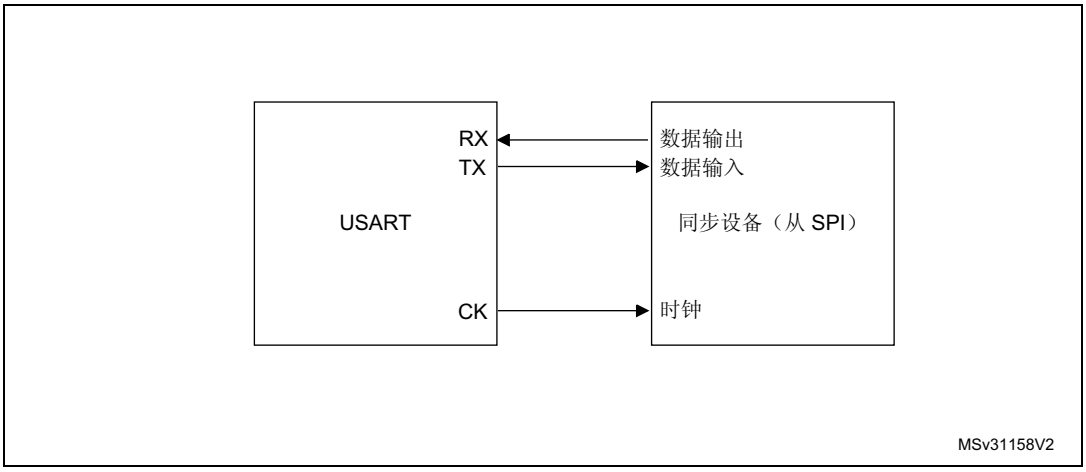


图 577. USART 在同步主模式下的数据时钟时序图（M 位 = 00）

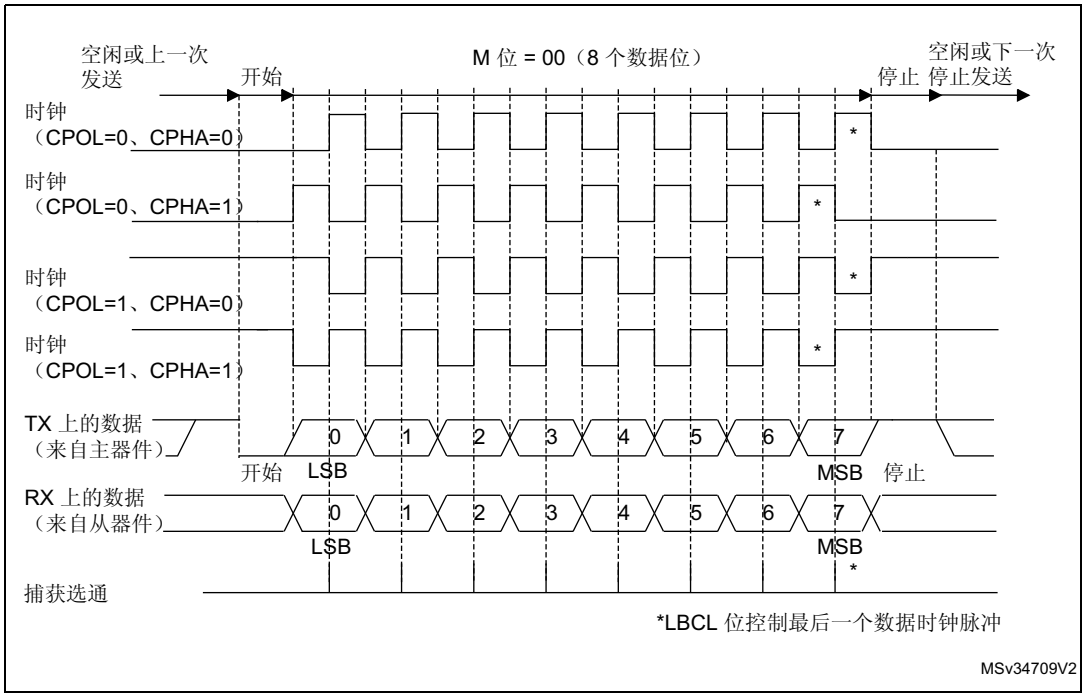
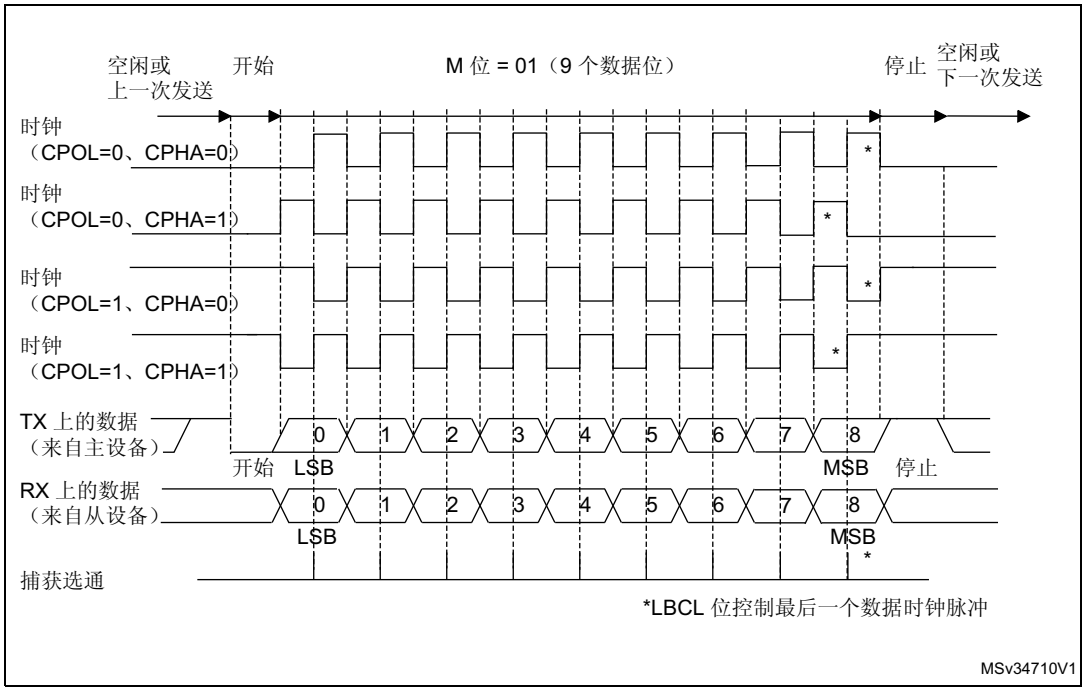


图 578. USART 在同步主模式下的数据时钟时序图（M 位 = 01）



从模式

通过将 USART_CR2 寄存器中的 SLVEN 位编程为“1”来选择同步从模式。在同步从模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 和 CLKEN 位，
- USART_CR3 寄存器中的 SCEN、HDSEL 和 IREN 位。

在此模式下，USART 可用于在从模式下控制双向同步串行通信。在从模式下，CK 引脚是 USART 的输入。

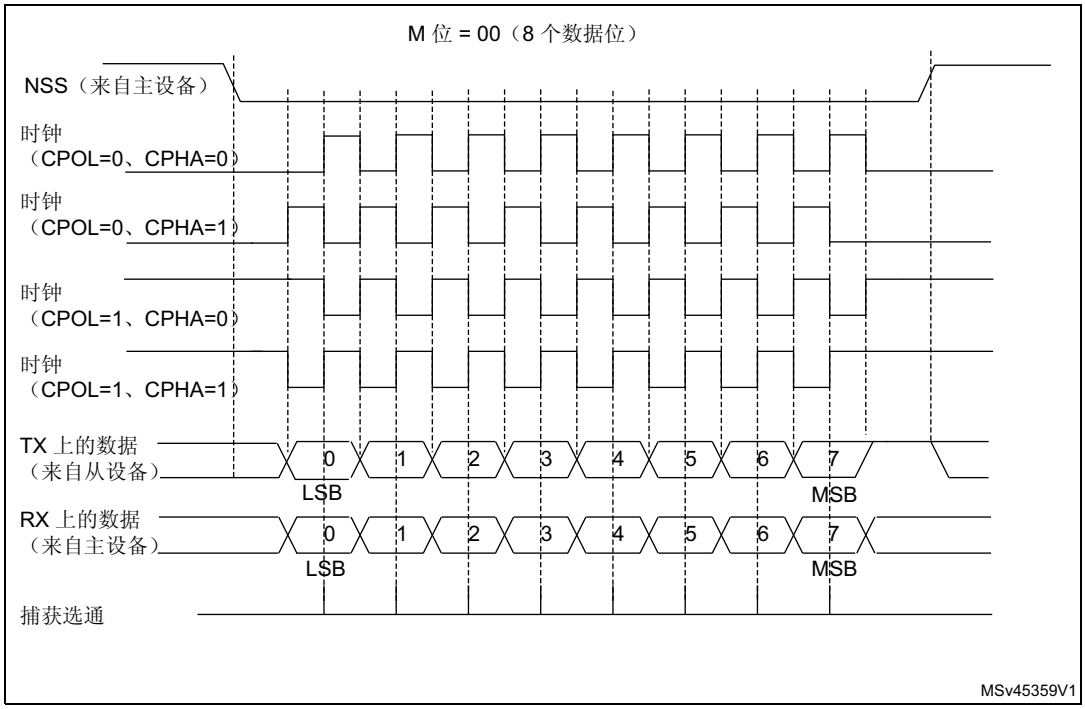
注： *当外设用于 SPI 从设备模式时，外设时钟源 (*usart_ker_ck_pres*) 的频率必须是 CK 输入频率的 3 倍以上。*

USART_CR2 寄存器中的 CPOL 位和 CPHA 位分别用于选择时钟极性和外部时钟的相位（请参见*图 579*）。

从发送模式支持下溢错误标志。如果在软件尚未将任何值加载到 USART_TDR 之前出现第一个数据发送时钟脉冲，则此标志将置 1。

从设备支持硬件和软件 NSS 管理。

图 579. USART 在同步从模式下的数据时钟时序图（M 位 = 00）



从设备选择 (NSS) 引脚管理

可通过 USART_CR2 寄存器中的 DIS_NSS 位设置硬件或软件从设备选择管理：

- 软件 NSS 管理 (DIS_NSS = 1)

  始终选择 SPI 从设备，忽略 NSS 输入引脚。

  外部 NSS 引脚空闲，可供其他应用使用。
- 硬件 NSS 管理 (DIS_NSS = 0)

  SPI 从设备选择取决于 NSS 输入引脚。NSS 为低电平时选择从设备，NSS 为高电平时取消选择从设备。

注： *在 USART 未使能时 (UE = 0)，必须选择 LBCL（仅用于 SPI 主设备模式）、CPOL 和 CPHA 位以确保时钟脉冲正常工作。*

在 SPI 从设备模式下，必须在启动主设备通信之前（或时钟稳定时在相邻帧之间）使能 USART。否则，如果 USART 从设备在主设备处于某个帧中间时使能，它会与主设备失去同步。在通信时钟的第一个边沿到来之前或者正在进行的通信结束之前，从设备的数据寄存器就需要准备就绪，否则 SPI 从设备会发送零。

SPI 从设备下溢错误

发生下溢错误时，USART_ISR 寄存器中的 UDR 标志会置 1，SPI 从设备继续发送最后一个数据，直到下溢错误标志由软件清零。

下溢标志在帧开始时置 1。如果 USART_CR3 寄存器中的 EIE 位置 1，则会触发下溢错误中断。

通过将 USART_ICR 寄存器中的位 UDRCF 置 1 来清零下溢错误标志。

发生下溢错误时，仍然可以对 TDR 寄存器进行写操作。清除下溢错误将允许发送新数据。

如果发生下溢错误且没有新数据被写入 TDR 中，则 TC 标志在帧结束时置 1。

注： *如果向 USART_TDR 写入数据的时间点过于接近第一个 CK 发送边沿，则可能会发生下溢错误。为避免发生此下溢错误，应在第一个 CK 边沿的 3 个* usart_ker_ck *周期之前对 USART_TDR 进行写操作。*

49.5.15 USART 单线半双工通信

通过将 USART_CR3 寄存器中的 HDSEL 位置 1 来选择单线半双工模式。在此模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 和 CLKEN 位，
- USART_CR3 寄存器中的 SCEN 和 IREN 位。

USART 可以配置为遵循单线半双工协议，其中 TX 和 RX 线路从内部相连接。使用 USART_CR3 寄存器中的控制位 HDSEL 可在半双工通信和全双工通信间进行选择。

向 HDSEL 位写入“1”后：

- TX 和 RX 线路从内部相连接。
- 不再使用 RX 引脚。
- 无数据传输时，TX 引脚始终处于释放状态。因此，它在空闲状态或接收过程中用作标准 I/O。这意味着，必须对 I/O 进行配置，以便将 TX 配置为复用功能开漏并外接上拉电阻。

除此之外，通信协议与正常 USART 模式下的通信协议相似。此线路上的任何冲突都必须由软件管理（例如，使用中央仲裁器）。尤其要注意，发送过程永远不会被硬件阻塞，只要数据是在 TE 位置 1 的情况下写入，发送就会持续进行。

49.5.16 USART 接收器超时

接收器超时功能可通过将 USART_CR2 控制寄存器中的 RTOEN 位置 1 来使能。

超时间隔通过 USART_RTOR 寄存器中的 RTO 位字段进行编程。

接收器超时计数器遵循以下规则开始计数：

- STOP =“00”或 STOP =“11”时从停止位结束时开始计数。
- STOP =“10”时从第二个停止位结束时开始计数。
- STOP =“01”时从停止位开始时开始计数。

经过超时间隔后，USART_ISR 寄存器中的 RTOF 标志置 1。如果 USART_CR1 寄存器中的 RTOIE 位置 1，则会产生超时中断。

49.5.17 USART 智能卡模式

仅在支持智能卡模式时才涉及本节内容。请参见*第 2436 页的第 49.4 节：USART 实现*。

通过将 USART_CR3 寄存器中的 SCEN 位置 1 选择智能卡模式。在智能卡模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 位，
- USART_CR3 寄存器中的 HDSEL 和 IREN 位。

此外，还可将 CLKEN 位置 1，以便为智能卡提供时钟。

智能卡接口支持符合 ISO 7816-3 标准的异步智能卡协议。同时支持 T = 0（字符模式）和 T = 1（块模式）。

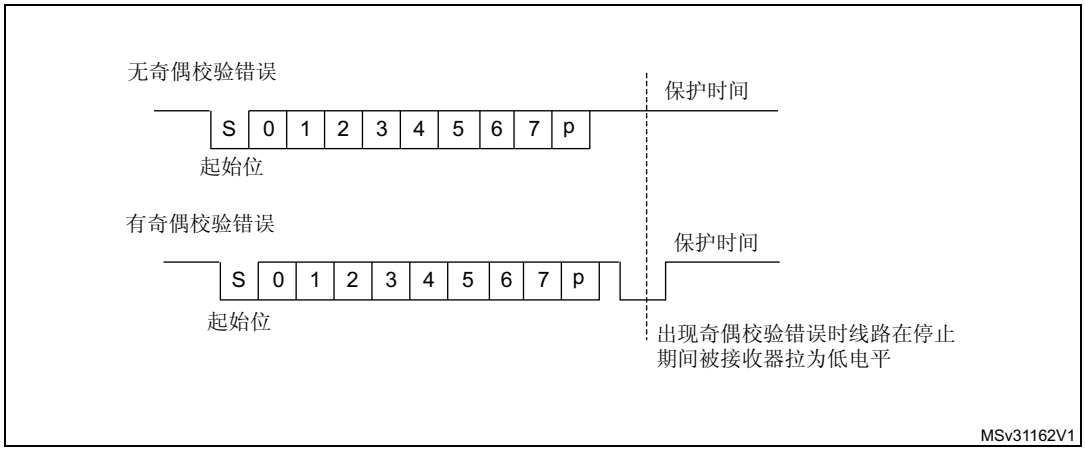
USART 应如下所示进行配置：

- 8 个位加奇偶校验：当 USART_CR1 寄存器中的 M = 1 且 PCE = 1 时
- 发送和接收数据时使用 1.5 个停止位：USART_CR2 寄存器中的 STOP =“11”。接收时也可以选择 0.5 个停止位。

在 T = 0（字符）模式下，奇偶校验错误在保护时间周期内的每个字符结束时指示。

图 580 显示了有奇偶校验错误和无奇偶校验错误时数据线上情况的示例。

图 580. ISO 7816-3 异步协议



连接到智能卡时，USART 的 TX 输出会驱动一条双向线（它也由智能卡驱动）。必须将 TX 引脚配置为开漏引脚。

智能卡模式采用单线半双工通信协议。

- 从发送移位寄存器发送数据会经过至少 1/2 个时钟周期的延迟。正常工作时，已满的发送移位寄存器会在下一个波特时钟边沿开始移位。在智能卡模式下，此发送过程还会进一步经过 1/2 波特时钟周期的延迟。
- 发送时，如果智能卡检测到奇偶校验错误，它会通过将线路驱动为低电平 (NACK) 告知 USART 此状态。此 NACK 信号（将发送线拉低 1 个时钟周期）会导致发送器端（配置为 1.5 个停止位）出现帧错误。USART 可根据协议自动重新发送数据。重试次数在 SCARCNT 位字段中编程。如果经过编程次数的重试后，USART 继续收到 NACK，USART 会停止发送并将该错误以帧错误形式发出。TXE 位（使能 FIFO 模式时为 TXFNF 位）可使用 USART_RQR 寄存器中的 TXFRQ 位置 1。

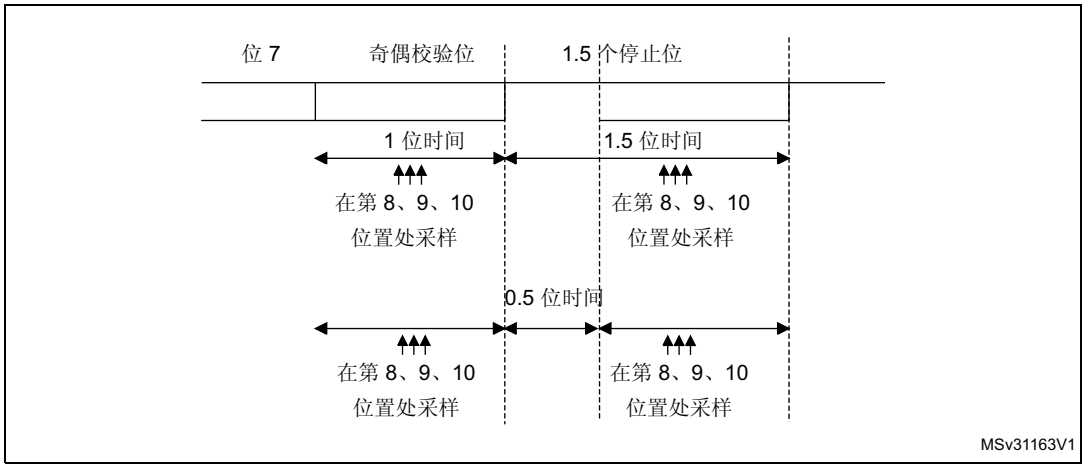
- 发送时智能卡自动重试：在 USART 检测 NACK 与重复字符的起始位之间插入 2.5 个波特率周期的延迟。接收最后一个重复字符后，立即将 TC 位置 1（无保护时间）。如果软件要重复此操作，必须确保标准要求的最短 2 个波特率周期。
- 如果在接收一个使用 1.5 个停止位编程的帧期间检测到奇偶校验错误，则在完成接收帧后，发送线会被拉低一个时钟周期。这是为了向智能卡指出发送到 USART 的数据尚未正确接收。如果 NACK 控制位置 1，接收器会向奇偶校验错误发送“NACK”信号；否则不会发送 NACK 信号（将在 T = 1 模式下会使用）。如果接收到的字符错误，则不会激活 RXNE（使能 FIFO 模式时为 RXFNE）/接收 DMA 请求。根据协议规范，智能卡必须重新发送相同的字符。如果经过 SCARCNT 位字段中指定的最大重试次数后接收到的字符仍然错误，USART 会停止发送 NACK 信号，并将错误以奇偶校验错误的形式发出。
- 接收时智能卡自动重试：如果 USART 向智能卡发送 NACK 信号，但智能卡不重复字符，则 BUSY 标志将保持置 1。
- 发送时，USART 会在两个连续字符之间插入保护时间（按照保护时间寄存器中编程的值）。由于保护时间在前一个字符的停止位后测量，因此必须将 GT[7:0] 寄存器编程为所需 CGT（字符保护时间，如 7816-3 规范所定义）减去 12（一个字符的持续时间）。
- 通过对保护时间寄存器进行编程，可以延迟 TC 标志的置位。正常工作时，当发送移位寄存器为空且没有新的发送请求出现时，会对 TC 标志进行置位。在智能卡模式下，空的发送移位寄存器会触发保护时间计数器，使其递增计数至保护时间寄存器中的值。在此期间，TC 标志被强制为低电平。当保护时间计数器达到设置值时，TC 置位为高电平。TCBGT 标志可用于检测数据传输是否结束，而无需等待保护时间结束。该标志在帧发送结束之后且未从智能卡接收到 NACK 时置 1。
- TC 标志的释放不受智能卡模式的影响。
- 如果在发送端检测到帧错误（由来自接收器的 NACK 信号引起），则发送端的接收块不会将 NACK 作为起始位进行检测。根据 ISO 协议，接收到的 NACK 信号的持续时间可以是 1 或 2 个时钟周期。
- 在接收端，如果检测到奇偶校验错误并发送了 NACK 信号，则接收端不会将 NACK 作为起始位进行检测。

注： *刹车字符在智能卡模式下无效。带有帧错误的 0x00 数据被视为数据，而非刹车。*

当翻转 TE 位时，不会发送空闲帧。空闲帧（在其他配置中进行了定义）在 ISO 协议中未进行定义。

图 581 介绍了 USART 如何对 NACK 信号采样。在本例中，USART 正在发送数据并配置了 1.5 个停止位。USART 的接收部分已被使能，以检查数据的完整性和 NACK 信号。

图 581. 使用 1.5 个停止位检测奇偶校验错误



USART 可以通过 CK 输出为智能卡提供时钟。在智能卡模式下，CK 仅通过一个 5 位预分频器由内部外设输入时钟提供。分频比在 USART_GTPR 寄存器中进行配置。CK 频率可在 usart_ker_ck_pres/2 到 usart_ker_ck_pres/62 之间进行编程，其中 usart_ker_ck_pres 为经过编程的预分频器分频的外设输入时钟。

块模式 (T = 1)

在 T = 1（块）模式下，通过将 USART_CR3 寄存器中的 NACK 位清零可停用奇偶校验错误发送。

在块模式下请求从智能卡执行读操作时，软件必须将 RTOR 寄存器编程为 BWT（块等待时间）- 11。如果在经过此时间段后未从智能卡接收到应答，将生成超时中断。如果在该时间段超时之前接收到第一个字符，则通过 RXNE/RXFNE 中断发出信号指示。

注： *即使在使用 DMA 模式下的 USART 从块模式下的智能卡读取时，也必须使能 RXNE/RXFNE 中断。同时，只有在接收到第一个字节后才可使能 DMA。*

接收到第一个字符（RXNE/RXFNE 中断）后，为允许自动检查两个连续字符间的最长等待时间，必须将 RTO 寄存器编程为 CWT（字符等待时间 - 值 11）。此时间以波特率时间单位表示。前一个字符结束后，如果智能卡未在小于 CWT 的时间段内发送新字符，USART 将通过 RTOF 标志和中断（当 RTOIE 位置 1 时）向软件指示此情况。

注： *按照智能卡协议定义，BWT/CWT 的值应从最后一个字符开始（起始位）时定义。必须将 RTO 寄存器分别编程为 BWT -11 或 CWT -11，并考虑最后一个字符本身的长度。*

块长度计数器用于对 USART 接收到的所有字符进行计数。当 USART 进行发送时，此计数器复位。块长度由智能卡在块的第三个字节（起始字段）中发送。必须将此值编程到 USART_RTOR 寄存器中的 BLEN 位字段。使用 DMA 模式时，在块开始之前，必须将该寄存器位字段编程为最小值 (0x0)。对于该值，在接收到第四个字符后生成中断。软件必须读取 LEN 字段（第三个字节），其值必须从接收缓存区中读取。

在中断驱动接收模式下，块长度可以由软件或者通过编程 BLEN 值来检查。但是在块开始前，可以编程 BLEN 的最大值 (0xFF)。接收到第三个字符后，将编程实际值。

如果块使用 LRC（纵向冗余校验，1 个结尾字节），则 BLEN = LEN。如果块使用 CRC 机制（2 个结尾字节），则必须编程 BLEN = LEN+1。块总长度（包括起始字段、结尾字段和信息字段）等于 BLEN+4。块结束的信号通过 EOBF 标志和中断（EOBIE 位置 1 时）发送给软件。

如果块长度出现错误，则通过 RTO 中断（字符等待时间上溢）发送块结束信号。

注： *错误检查代码 (LRC/CRC) 必须通过软件计算/验证。*

正向约定和反向约定

智能卡协议定义了两种约定：正向约定和反向约定。

正向约定定义为：LSB 在前，逻辑位值 1 对应于线路的 H 状态，奇偶校验为偶校验。要使用此约定，必须编程以下控制位：MSBFIRST = 0，DATAINV = 0（默认值）。

反向约定定义为：MSB 在前，逻辑位值 1 对应于信号线路的 L 状态，奇偶校验为偶校验。要使用此约定，必须编程以下控制位：MSBFIRST = 1，DATAINV = 1。

注： *将逻辑数据值取反（0 = H，1 = L）时，奇偶校验位将同样取反。*

为识别智能卡约定，智能卡会将初始字符 TS 作为 ATR（复位应答）的第一个字符发送。TS 支持两种格式：LHHL LLL LLH 和 LHHL HHH LLH。

- (H) LHHL LLL LLH 建立反向约定：状态 L 编码为值 1，时刻 2 传送最高有效位（MSB 在前）。按反向约定解码时，传送的字节等于“3F”。
- (H) LHHL HHH LLH 建立正向约定：状态 H 编码为值 1，时刻 2 传送最低有效位（LSB 在前）。按正向约定解码时，传送的字节等于“3B”。

在 2 到 10 的九个时刻中，如果有偶数个位设置为 1，则字符的奇偶校验正确。

由于 USART 不了解智能卡使用哪种约定，因此 USART 需要能够识别任意一种模式并相应操作。模式识别不在硬件中完成，而是通过软件序列完成。此外，假设以正向约定配置 USART（默认）而智能卡以反向约定 (TS = LHHL LLL LLH) 应答，则 USART 接收到的字节将为 03，奇偶校验将为奇校验。

因此，有两种方法可用于识别 TS 模式：

方法 1

将 USART 编程为标准智能卡模式/正向约定。这种情况下，TS 模式接收会向智能卡生成奇偶校验错误中断和错误信号。

- 奇偶校验错误中断通知软件智能卡未以正向约定正确应答。之后，软件重新将 USART 编程为反向约定
- 为响应错误信号，智能卡会重试同一 TS 字符，此时重新编程后的 USART 将正确接收该字符。

或者，为应答奇偶校验错误中断，软件可决定重新编程 USART，同时向智能卡生成新的复位命令，然后再次等待 TS。

方法 2

将 USART 编程为 9 位/无奇偶校验模式，无位反向。在此模式下，按如下方式接收两种 TS 模式中的任一种：

(H) LHHL LLL LLH = 0x103：选择反向约定

(H) LHHL HHH LLH = 0x13B：选择正向约定

软件根据这两种模式检查接收到的字符，如果其中任意一种匹配，则相应编程 USART 以接收下一个字符。

如果两种都未被识别，则可能复位智能卡以重新开始协商。

49.5.18 USART IrDA SIR ENDEC 模块

仅在支持 IrDA 模式时才涉及本节内容。请参见*第 2436 页的第 49.4 节：USART 实现*。

通过将 USART_CR3 寄存器中的 IREN 位置 1 来选择 IrDA 模式。在 IrDA 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN、STOP 和 CLKEN 位，
- USART_CR3 寄存器中的 SCEN 和 HDSEL 位。

IrDA SIR 物理层规定使用反相归零 (RZI) 调制方案，它以红外光脉冲表示逻辑 0（参见*图 582*）。

SIR 发送编码器用于调制 USART 发出的非归零 (NRZ) 位流。输出脉冲流会发送到外部输出驱动器和红外线 LED。USART 支持的 SIR ENDEC 比特率最高为 115.2 kbaud。在正常模式下，所发送的脉冲宽度规定为一个位周期的 3/16。

SIR 接收解码器用于解调由红外探测器发出的归零位流，并将接收到的 NRZ 串行位流输出到 USART。在空闲状态下，解码器输入通常为高电平（标记状态）。发送编码器输出的极性与解码器输入相反。当解码器输入为低电平时，会检测到起始位。

- IrDA 是一个半双工通信协议。如果发送器忙（USART 正在向 IrDA 编码器发送数据时），则 IrDA 解码器会忽略 IrDA 接收线上的所有数据；如果接收器忙（USART 正在接收来自 USART 的解码数据时），则 IrDA 不会对 USART 发送到 IrDA 的 TX 上的数据进行编码。在接收数据时，应避免同时进行发送，因为这样做可能会破坏要发送的数据。
- “0”作为高电平脉冲发送，而“1”作为“0”发送。在正常模式下，脉冲宽度规定为所选位周期的 3/16（参见*图 583*）。
- SIR 解码器用于将兼容 IrDA 的接收信号转换为 USART 的位流。
- SIR 接收逻辑将高电平状态视为逻辑“1”，将低电平脉冲视为逻辑“0”。
- 发送编码器输出的极性与解码器输入相反。SIR 输出在空闲时处于低电平状态。
- IrDA 规范要求脉冲容忍值要大于 1.41 µs。可接受的脉冲宽度可通过寄存器设置。接收器端的干扰检测逻辑会滤除宽度小于 2 个 PSC 周期的脉冲（PSC 是在 USART_GTPR 中编程的预分频器值）。宽度小于 1 个 PSC 周期的脉冲都将被拒绝，但宽度大于 1 个而小于 2 个周期的脉冲可能被接受也可能被拒绝，而宽度大于 2 个周期的脉冲将被接受作为有效脉冲。当 PSC = 0 时，IrDA 编码器/解码器不工作。
- 接收器能够与低功耗发送器进行通信。
- 在 IrDA 模式下，USART_CR2 寄存器中的停止位必须配置为“1 个停止位”。

IrDA 低功耗模式

- 发送器

  在低功耗模式下，脉冲宽度不再保持为位周期的 3/16。此时的脉冲宽度为低功耗波特率的 3 倍，最小可为 1.42 MHz。通常此值是 1.8432 MHz (1.42 MHz < PSC < 2.12 MHz)。低功耗模式下的可编程分频器会对系统时钟进行分频，以达到此值。
- 接收器

  在低功耗模式下接收与在正常模式下接收类似。为进行干扰检测，USART 应丢弃持续时间短于 1/PSC 的脉冲。只有当持续时间大于 2 个 IrDA 低功耗波特时钟周期（USART_GTPR 寄存器中的 PSC 值）时，才是有效低电平。

注： *宽度小于两个但大于一个 PSC 周期的脉冲可能被接受，也可能被拒绝。*

接收器的建立时间应由软件进行管理。IrDA 物理层规范规定发送和接收之间至少要经过 10 ms 的延迟（IrDA 是一个半双工协议）。

图 582. IrDA SIR ENDEC 框图

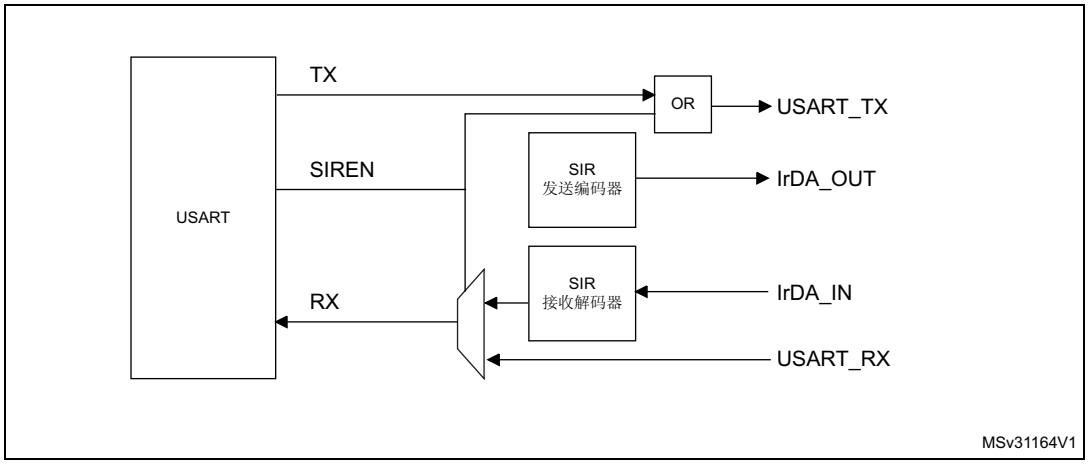
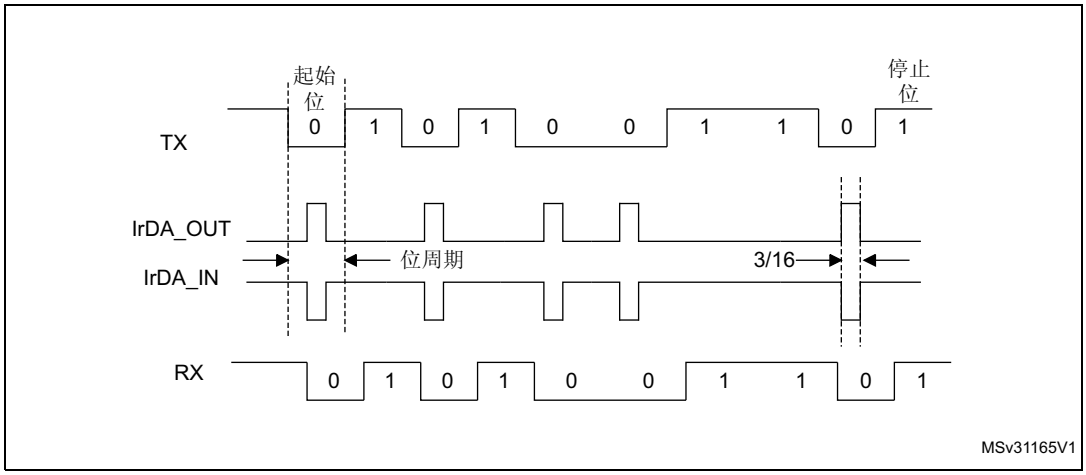


图 583. IrDA 数据调制 (3/16)——正常模式



49.5.19 使用 USART 和 DMA 进行连续通信

USART 能够使用 DMA 进行连续通信。接收缓冲区和发送缓冲区的 DMA 请求是独立生成的。

注： *要确定是否支持 DMA 模式，请参见第 2436 页的第 49.4 节：USART 实现。如果不支持 DMA 模式，请按照第 49.5.6 节中的说明使用 USART。可以将 USART_ISR 寄存器中的 TXE/RXNE 标志清零，从而在禁止 FIFO 时实现连续通信。*

使用 DMA 进行发送

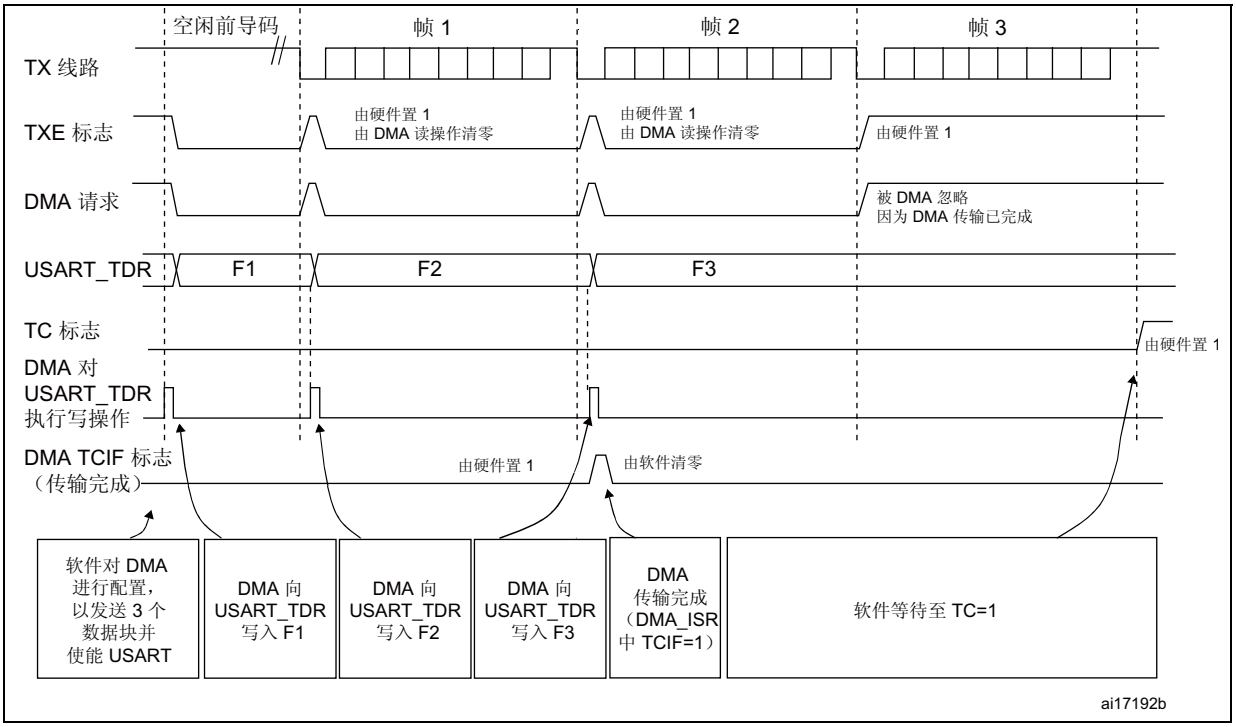
将 USART_CR3 寄存器中的 DMAT 位置 1 可以使能 DMA 模式进行发送。当 TXE 标志（使能 FIFO 模式时为 TXFNF 标志）置 1 时，可使用 DMA 外设（请参见相应的*直接内存访问控制器*部分）将数据从配置的 SRAM 区域加载到 USART_TDR 寄存器。要映射一个 DMA 通道以进行 USART 发送，请按以下步骤操作（x 表示通道编号）：

1. 在 DMA 控制寄存器中写入 USART_TDR 寄存器地址，将其配置为传输的目标地址。每次发生 TXE（使能 FIFO 模式时为 TXFNF）事件后，数据都从内存移动到此地址。
2. 在 DMA 控制寄存器中写入内存地址，将其配置为传输的源地址。每次发生 TXE（使能 FIFO 模式时为 TXFNF）事件后，数据都从该存储区域加载到 USART_TDR 寄存器中。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生 DMA 中断。
6. 通过将 USART_ICR 寄存器中的 TCCF 位置 1，将 USART_ISR 寄存器中的 TC 标志清零。
7. 在 DMA 寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

在发送模式下，DMA 对所有要发送的数据执行了写操作（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 TC 标志进行监视，以确保 USART 通信已完成。在禁止 USART 之前或系统进入低功耗模式之前（禁止外设时钟时）必须执行此步骤，以避免损坏最后一次发送。软件必须等待直到 TC = 1。TC 标志在所有数据发送期间都保持清零状态，然后在最后一帧发送结束时由硬件置 1。

图 584. 使用 DMA 进行发送



注： *使能 FIFO 管理时，DMA 请求由发送 FIFO 未满（即 TXFNF = 1）触发。*

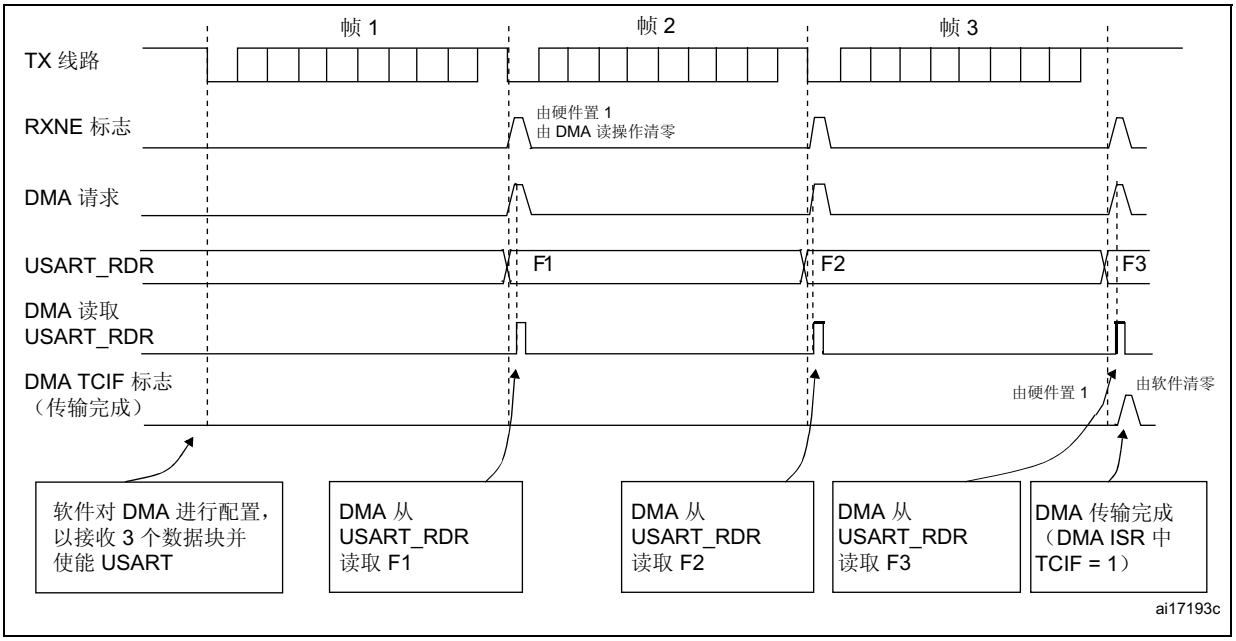
使用 DMA 进行接收

将 USART_CR3 寄存器中的 DMAR 位置 1 可以使能 DMA 模式进行接收。接收数据字节时，可使用 DMA 外设（请参见相应的*直接内存访问控制器*部分）将数据从 USART_RDR 寄存器加载到配置的 SRAM 区域中。要映射一个 DMA 通道以进行 USART 接收，请按以下步骤操作：

1. 在 DMA 控制寄存器中写入 USART_RDR 寄存器地址，将其配置为传输的源地址。每次发生 RXNE（使能 FIFO 模式时为 RXFNE）事件后，数据都从该地址移动到内存。
2. 在 DMA 控制寄存器中写入内存地址，将其配置为传输的目标地址。每次发生 RXNE（使能 FIFO 模式时为 RXFNE）事件后，数据都从 USART_RDR 加载到此存储区域。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 控制寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生中断。
6. 在 DMA 控制寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

图 585. 使用 DMA 进行接收



注： *使能 FIFO 管理时，DMA 请求由接收 FIFO 非空（即 RXFNE = 1）触发。*

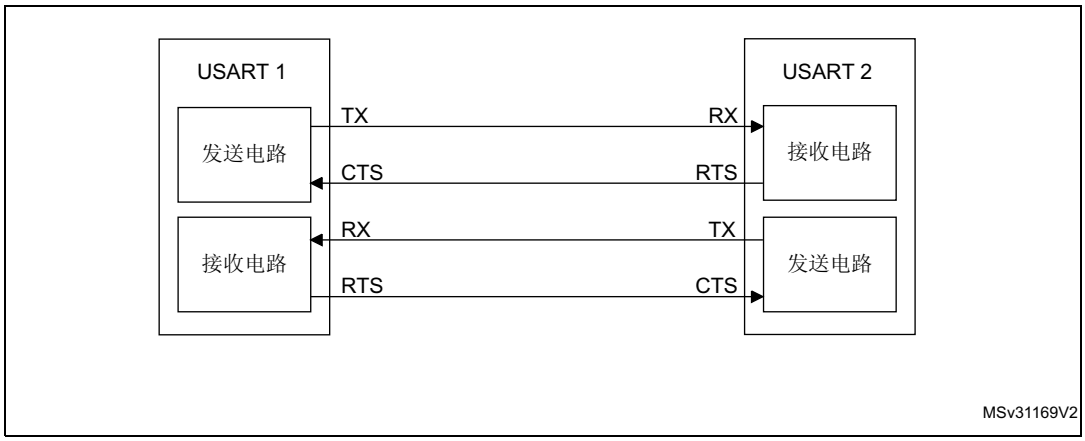
多缓冲区通信中的错误标志和中断生成

在多缓冲区通信模式下，如果事务中发生任何错误，都会在当前字节后将相应错误标志置 1。如果中断使能标志置 1，则会产生中断。在单字节接收过程中，与 RXNE（使能 FIFO 模式时为 RXFNE）一同置位的帧错误、上溢错误和噪声标志具有单独的错误标志中断使能位（USART_CR3 寄存器中的 EIE 位）；如果该位置 1，在发生其中任何一个错误时，都会在当前字节后使能中断。

49.5.20 RS232 硬件流控制和 RS485 驱动器使能

使用 CTS 输入和 RTS 输出可以控制 2 个设备间的串行数据流。*图 586* 显示了在这种模式下如何连接 2 个设备：

图 586. 2 个 USART 间的硬件流控制

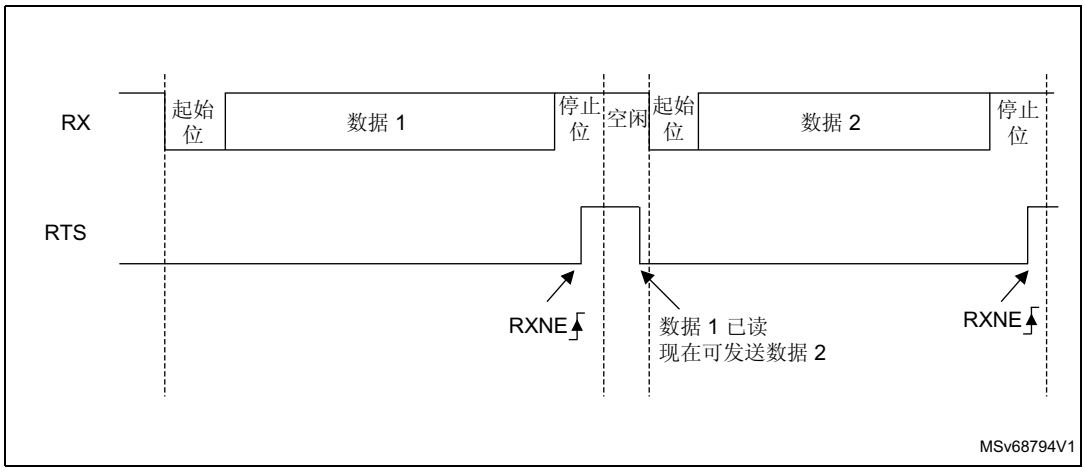


向 USART_CR3 寄存器中的 RTSE 位和 CTSE 位写入“1”，可分别使能 RS232 RTS 和 CTS 流控制。

RS232 RTS 流控制

如果使能 RTS 流控制 (RTSE = 1)，只要 USART 接收器准备好接收新数据，便会将 RTS 变为无效（连接到低电平）。当接收寄存器已满时，会将 RTS 变为有效，表明发送过程会在当前帧结束后停止。*图 587* 显示了在使能 RTS 流控制的情况下进行通信的示例。

图 587. RS232 RTS 流控制



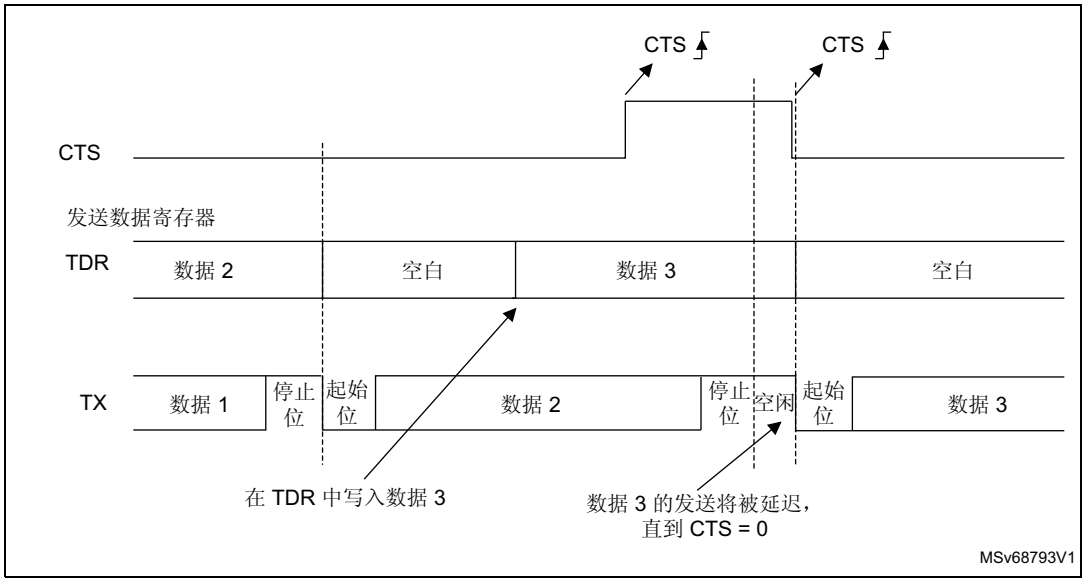
注： *如果使能 FIFO 模式，则仅当 RXFIFO 已满时，RTS 才会变为有效。*

RS232 CTS 流控制

如果使能 CTS 流控制 (CTSE = 1)，则发送器会在发送下一帧前检查 CTS。如果 CTS 无效（连接到低电平），则会发送下一数据（假设数据已准备好发送，即 TXE/TXFE = 0）；否则不会进行发送。如果在发送过程中 CTS 变为有效，则当前发送完成之后，发送器停止。

当 CTSE = 1 时，只要 CTS 发生变化，CTSIF 状态位便会由硬件自动置 1。这指示接收器是否已准备好进行通信。如果 USART_CR3 寄存器中的 CTSIE 位置 1，则会产生中断。*图 588* 显示了在使能 CTS 流控制的情况下进行通信的示例。

图 588. RS232 CTS 流控制



注： *为正常运行，必须在当前字符结束前至少 3 个 USART 时钟源周期内将 CTS 拉低。此外还应注意，当脉冲短于 2 个 PCLK 周期时无法将 CTSCF 标志置 1。*

RS485 驱动器使能

驱动器使能功能可通过将 USART_CR3 控制寄存器中的 DEM 位置 1 使能。这样用户便可通过 DE（驱动器使能）信号激活外部收发器控制。生效时间为激活 DE 信号与 START 位开始间的时间。可以通过 USART_CR1 控制寄存器中的 DEAT [4:0] 位字段编程生效时间。禁止时间为发送的消息中最后一个停止位结束与取消激活 DE 信号间的时间。可以通过 USART_CR1 控制寄存器中的 DEDT [4:0] 位字段编程禁止时间。DE 信号的极性可使用 USART_CR3 控制寄存器中的 DEP 位配置。

在 USART 中，DEAT 和 DEDT 以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

49.5.21 USART 低功耗管理

USART 具有高级低功耗模式功能，即使禁止 usart_pclk 时钟，也能正常传输数据。

UESM 位置 1 时，USART 能够将 MCU 从低功耗模式唤醒。

当 usart_pclk 被关闭时，如果某些特定操作需要激活 **usart_pclk** 时钟，则 USART 会提供唤醒中断 (**usart_wkup**)：

- 禁止 FIFO 模式时

  必须激活 usart_pclk 时钟以清空 USART 数据寄存器。

  在这种情况下，usart_wkup 中断源为 RXNE 置“1”。RXNEIE 位必须在进入低功耗模式之前置 1。

- 使能 FIFO 模式时

  必须激活 usart_pclk 时钟以：

  – 填充 TXFIFO

  – 或清空 RXFIFO

  在这种情况下，usart_wkup 中断源可以为：

  – RXFIFO 非空。此时，RXFNEIE 位必须在进入低功耗模式之前置 1。

  – RxFIFO 已满。此时，RXFFIE 位必须在进入低功耗模式之前置 1，接收到的数据量对应于 RXFIFO 大小，并且 RXFF 标志未置 1。

  – TXFIFO 为空。此时，TXFEIE 位必须在进入低功耗模式之前置 1。

  这允许在低功耗模式下发送/接收 TXFIFO/RXFIFO 中的数据。

  为了避免发生上溢/下溢错误以及在低功耗模式下发送/接收数据，usart_wkup 中断源可以是以下事件之一：

  – 达到 TXFIFO 阈值。此时，TXFTIE 位必须在进入低功耗模式之前置 1。

  – 达到 RXFIFO 阈值。此时，RXFTIE 位必须在进入低功耗模式之前置 1。

  例如，如果唤醒时间短于在线路上接收单个字节所需的时间，则应用可将阈值设为最大 RXFIFO 大小。

  使用 RXFIFO 已满、TXFIFO 为空、RXFIFO 非空和 RXFIFO/TXFIFO 阈值中断将 MCU 从低功耗模式唤醒时，可在低功耗模式下尽可能多地进行 USART 传输，这样有助于优化功耗。

或者，也可通过 WUS 位字段选择特定 **usart_wkup** 中断。

检测到唤醒事件后，WUF 标志会由硬件置 1 并在 WUFIE 位置 1 时生成一个 **usart_wkup** 中断。

注： *在进入低功耗模式之前，请确保未进行任何 USART 传输。检查 BUSY 标志不能确保在进行数据接收时绝不会进入低功耗模式。*

WUF 标志在检测到唤醒事件时置 1，而与 MCU 处于低功耗模式还是工作模式无关。

如果在初始化和使能接收器后立即进入低功耗模式，则必须检查 REACK 位以确保 USART 确实已使能。

当 DMA 用于接收时，它必须在进入低功耗模式前禁止，并在退出低功耗模式后重新使能。

如果使能 FIFO，则只有在使能静默模式的情况下才能在地址匹配时从低功耗模式唤醒。

使用静默模式和低功耗模式

如果 USART 在进入低功耗模式前处于静默模式：

- 不得使用空闲检测来从静默模式唤醒，因为空闲检测无法在低功耗模式下工作。
- 如果使用地址匹配来从静默模式唤醒，则低功耗模式唤醒源也必须是地址匹配。如果 RXNE 标志在进入低功耗模式时置 1，则接口将在地址匹配时和从低功耗模式唤醒时保持静默模式。

注： *使能 FIFO 管理时，静默模式可与从低功耗模式唤醒搭配使用，而且没有任何限制（即，上述关于静默和低功耗模式的两点仅在 FIFO 管理禁止时有效）。*

USART 内核时钟 (usart_ker_ck) 在低功耗模式下关闭时从低功耗模式唤醒

在低功耗模式下，如果在 USART 接收线上检测到下降沿时 usart_ker_ck 时钟处于关闭状态，则 USART 接口会借助 usart_ker_ck_req 信号请求开启 usart_ker_ck 时钟。usart_ker_ck 随后用来接收帧。

如果唤醒事件通过验证，则 MCU 将从低功耗模式唤醒，并会正常进行数据接收。

如果唤醒事件未通过验证，则 usart_ker_ck 会再次关闭，MCU 不会被唤醒并保持在低功耗模式下，且内核时钟请求被释放。

以下示例显示了将唤醒事件编程为“地址匹配检测”并禁止 FIFO 管理的情况。

图 589 所示为唤醒事件通过验证时的 USART 行为。

图 589. 唤醒事件通过验证（唤醒事件 = 地址匹配，禁止 FIFO）

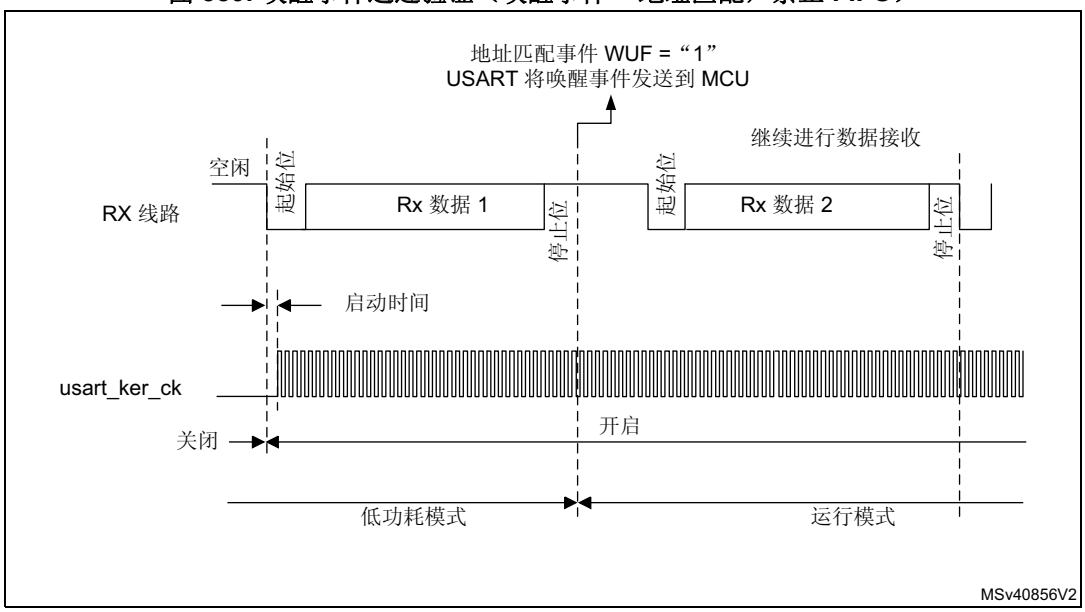
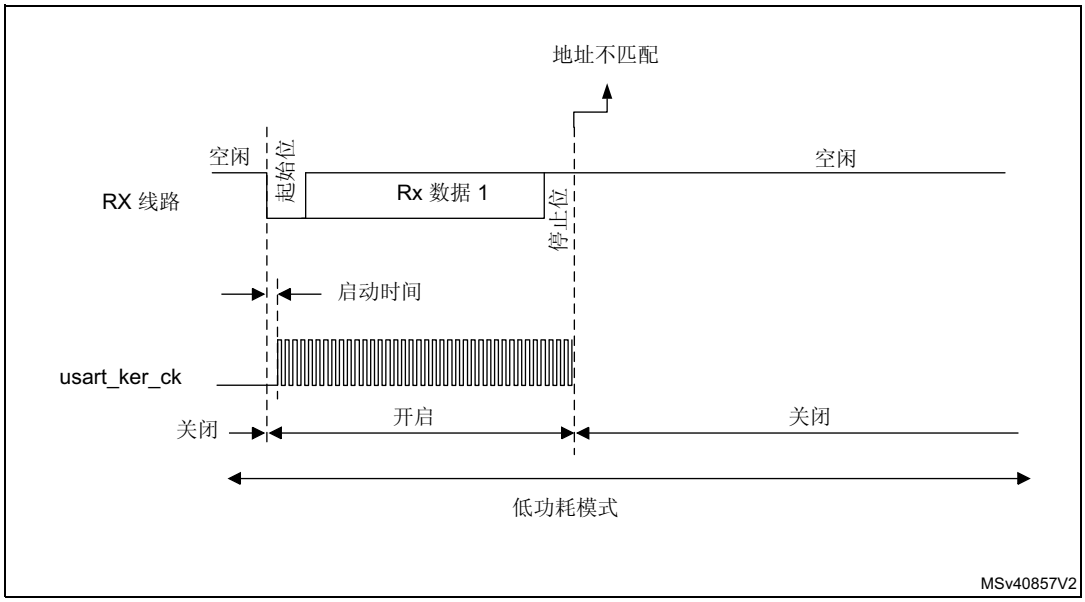


图 590 所示为唤醒事件未通过验证时的 USART 行为。

图 590. 唤醒事件未通过验证（唤醒事件 = 地址匹配，禁止 FIFO）



注： *将地址匹配或任一接收的帧用作唤醒事件时，如上两图适用。如果唤醒事件为起始位检测，则 USART 会在起始位结束时向 MCU 发送唤醒事件。*

确定允许从低功耗模式正确唤醒设备的最大 USART 波特率

允许从低功耗模式正确唤醒设备的最大波特率取决于唤醒时间参数（请参见设备数据手册）和 USART 接收器容差（请参见*第 49.5.8 节：USART 接收器对时钟偏差的容差*）。

举例来说：OVER8 = 0，M 位 = “01”，ONEBIT = 0，BRR [3:0] = 0000。

在这些条件下，根据*表 440：BRR [3:0] = 0000 时的 USART 接收器容差*，USART 接收器的容差为 3.41%。

DTRA + DQUANT + DREC + DTCL + DWU < USART 接收器的容差

$D_{WUmax} = t_{WUUSART} / (11 \times T_{bit\ Min})$

$T_{bit\ Min} = t_{WUUSART} / (11 \times D_{WUmax})$

其中 $t_{WUUSART}$ 为从低功耗模式唤醒的时间。

考虑一种理想情况：参数 DTRA、DQUANT、DREC 和 DTCL 为 0%，则 DWU 的最大值为 3.41%。实际上，我们至少需要考虑 usart_ker_ck 不精确的情况。

例如，如果将 HSI 用作 usart_ker_ck，HSI 的不精确度为 1%，则可以得到：

$t_{WUUSART}$ = 3 µs（仅提供数值作为参考；有关正确数值，请参见设备数据手册）。

D_{WUmax} = 3.41% - 1% = 2.41%

$T_{bit\ min}$ = 3 µs/ (11 x 2.41%) = 11.32 µs。

因此，允许从低功耗模式正确唤醒的最大波特率为：1/11.32 µs = 88.36 Kbaud。

49.6 处于低功耗模式的 USART

表 443. 低功耗模式对 USART 的影响

模式	说明
Sleep	无影响。USART 中断可使设备退出睡眠模式。
Stop(1)	USART 寄存器的内容会被保持。 当 USART 的时钟由 Stop 模式下可用的振荡器提供时，USART 能够将微控制器从 Stop 模式唤醒。
Standby	USART 外设掉电，退出待机模式后必须重新初始化。

1. 请参见*第 49.4 节：USART 实现*以了解给定外设实例是否支持从 Stop 模式唤醒。如果某个实例在给定的 Stop 模式下不起作用，则必须在进入此 Stop 模式之前将其禁止。

49.7 USART 中断

有关所有 USART 中断请求的详细说明，请参见*表 444*。

表 444. USART 中断请求

中断向量	中断事件	事件标志	使能控制位	中断清除方法	退出睡眠模式	退出 Stop (1) 模式	退出待机模式
USART 或 UART	发送数据寄存器为空	TXE	TXEIE	写入 TDR	是	否	否
	发送 FIFO 未满	TXFNF	TXFNFIE	TXFIFO 已满		否	
	发送 FIFO 为空	TXFE	TXFEIE	写入 TDR 或向 TXFRQ 中写入 1		是	
	达到发送 FIFO 阈值	TXFT	TXFTIE	写入 TDR		是	
	CTS 中断	CTSIF	CTSIE	向 CTSCF 中写入 1		否	
	发送完成	TC	TCIE	写入 TDR 或向 TCCF 中写入 1		否	
	保护时间前发送完成	TCBGT	TCBGTIE	写入 TDR 或向 TCBGT 中写入 1		否	

表 444. USART 中断请求（续）

中断向量	中断事件	事件标志	使能控制位	中断清除方法	退出睡眠模式	退出 Stop (1) 模式	退出待机模式
USART 或 UART	接收数据寄存器不为空（已准备好读取数据）	RXNE	RXNEIE	读取 RDR 或向 RXFRQ 中写入 1	是	是	否
	接收 FIFO 非空	RXFNE	RXFNEIE	读取 RDR 直到 RXFIFO 为空或向 RXFRQ 中写入 1		是	
	接收 FIFO 已满	RXFF(2)	RXFFIE	读取 RDR		是	
	达到接收 FIFO 阈值	RXFT	RXFTIE	读取 RDR		是	
	检测到上溢错误	ORE	RXNEIE/ RXFNEIE	向 ORECF 中写入 1		否	
	检测到空闲线路	IDLE	IDLEIE	向 IDLECF 中写入 1		否	
	奇偶校验错误	PE	PEIE	向 PECF 中写入 1		否	
	LIN 刹车	LBDF	LBDIE	向 LBDCF 中写入 1		否	
	多缓冲区通信中的噪声错误	NE	EIE	向 NFCF 中写入 1		否	
	多缓冲区通信中的上溢错误	ORE(3)		向 ORECF 中写入 1		否	
	多缓冲区通信中的帧错误	FE		向 FECF 中写入 1		否	
	字符匹配	CMF	CMIE	向 CMCF 中写入 1		否	
	接收器超时	RTOF	RTOFIE	向 RTOCCF 中写入 1		否	
	块结束	EOBF	EOBIE	向 EOBCF 中写入 1		否	
	从低功耗模式唤醒	WUF	WUFIE	向 WUC 中写入 1		是	
	SPI 从设备下溢错误	UDR	EIE	向 UDRCF 中写入 1		否	

1. 仅当外设实例支持从 Stop 模式唤醒功能时，USART 才能将设备从 Stop 模式唤醒。关于所支持的 Stop 模式的列表，请参见 *第 49.4 节：USART 实现*。
2. 如果 USART 接收 n+1 个数据（n 表示 RXFIFO 大小，RXFIFO 中有 n 个数据，USART_RDR 中有 1 个数据），则 RXFF 标志置 1。在 Stop 模式下，未向 USART_RDR 提供时钟。因此，不会对该寄存器进行写操作，一旦有 n 个数据被接收并写入到 RXFIFO，RXFF 中断便会生效（RXFF 标志未置 1）。
3. 当 OVRDIS = 0 时。

49.8 USART 寄存器

有关寄存器说明中使用的缩写，请参见*第 109 页的第 1.2 节*。

外设寄存器必须按字（32 位）进行访问。

49.8.1 USART 控制寄存器 1 (USART_CR1)

USART control register 1

偏移地址：0x00

复位值：0x0000 0000

同一寄存器可用于使能 FIFO 模式（本节）和禁止 FIFO 模式（下一节）的情况。

使能 FIFO 模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXFFIE	TXFEIE	FIFOEN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXFNFIE	TCIE	RXFNEIE	IDLEIE	TE	RE	UESM	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **RXFFIE**：RXFIFO 变满时中断使能 (RXFIFO full interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：当 USART_ISR 寄存器中的 RXFF = 1 时，生成 USART 中断

位 30 **TXFEIE**：TXFIFO 为空时中断使能 (TXFIFO empty interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：当 USART_ISR 寄存器中的 TXFE = 1 时，生成 USART 中断

位 29 **FIFOEN**：FIFO 模式使能 (FIFO mode enable)

该位由软件置 1 和清零，

0：禁止 FIFO 模式。

1：使能 FIFO 模式。

只有在禁止 USART (UE = 0) 时才能写入该位字段。

注意：FIFO 模式只能在标准 UART 通信、SPI 主/从设备模式和智能卡模式下使用，不得在 IrDA 和 LIN 模式下使能。

位 28 **M1**：字长 (Word length)

该位必须与位 12 (M0) 搭配使用来确定字长度。该位由软件置 1 或清零。

M[1:0] = “00”：1 个起始位，8 个数据位，n 个停止位

M[1:0] = “01”：1 个起始位，9 个数据位，n 个停止位

M[1:0] = “10”：1 个起始位，7 个数据位，n 个停止位

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：在 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。

位 27 **EOBIE**：块结束中断使能 (End-of-block interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：USART_ISR 寄存器中的 EOBF 标志置 1 时生成 USART 中断

注意：如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 26 **RTOIE**：接收器超时中断使能 (Receiver timeout interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：USART_ISR 寄存器中的 RTOF 位置 1 时生成 USART 中断。

注意：如果 USART 不支持接收器超时功能，该位保留且必须保持复位值。第 2436 页的第 49.4 节：USART 实现.

位 25:21 **DEAT[4:0]**：驱动器使能有效时间 (Driver enable assertion time)

该 5 位值用于定义激活 DE（启动器使能）信号与起始位开始间的时间。此时间以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

只有在禁止 USART (UE = 0) 时才能写入该位字段。

注意：如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 20:16 **DEDT[4:0]**：驱动器使能无效时间 (Driver enable deassertion time)

该 5 位值用于定义发送的消息中最后一个停止位结束与取消激活 DE（驱动器使能）信号间的时间。此时间以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

如果在 DEDT 时间内对 USART_TDR 寄存器执行写操作，则新数据仅在经过 DEDT 和 DEAT 时间后才会发送。

只有在禁止 USART (UE = 0) 时才能写入该位字段。

注意：如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 15 **OVER8**：过采样模式 (Oversampling mode)

0：16 倍过采样

1：8 倍过采样

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：在 LIN、IrDA 和智能卡模式下，该位必须保持清零。

位 14 **CMIE**：字符匹配中断使能 (Character match interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：USART_ISR 寄存器中的 CMF 位置 1 时生成 USART 中断。

位 13 **MME**：静默模式使能 (Mute mode enable)

该位可使能 USART 静默模式功能。该位置 1 时，USART 会按照 WAKE 位的定义在活动模式与静默模式之间切换。该位由软件置 1 和清零。

0：接收器永久处于活动模式

1：接收器可在静默模式和活动模式之间切换。

位 12 **M0**：字长 (Word length)

该位与位 28 (M1) 搭配使用来确定字长度。它由软件置 1 或清零（请参见位 28 (M1) 的说明）。

只有在禁止 USART (UE = 0) 时才能写入该位。

位 11 **WAKE**：接收器唤醒方法 (Receiver wake-up method)

该位用于确定 USART 静默模式的唤醒方法。该位由软件置 1 或清零。

0：空闲线路

1：地址标记

只有在禁止 USART (UE = 0) 时才能写入该位字段。

位 10 **PCE**：奇偶校验控制使能 (Parity control enable)

该位选择硬件奇偶校验控制（生成和检测）。使能奇偶校验控制时，计算出的奇偶校验位被插入到 MSB 位置（如果 M = 1，则为第 9 位；如果 M = 0，则为第 8 位），并对接收到的数据检查奇偶校验位。该位由软件置 1 和清零，一旦该位置 1，PCE 在当前字节的后面处于活动状态（在接收和发送时）。

0：禁止奇偶校验控制

1：使能奇偶校验控制

只有在禁止 USART (UE = 0) 时才能写入该位字段。

位 9 **PS**：奇偶校验选择 (Parity selection)

该位用于在使能奇偶校验生成/检测（PCE 位置 1）时选择奇校验或偶校验。该位由软件置 1 和清零。在当前字节的后面选择奇偶校验。

0：偶校验

1：奇校验

只有在禁止 USART (UE = 0) 时才能写入该位字段。

位 8 **PEIE**：PE 中断使能 (PE interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：当 USART_ISR 寄存器中的 PE = 1 时，生成 USART 中断

位 7 **TXFNFIE**：TXFIFO 未满中断使能 (TXFIFO not-full interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：当 USART_ISR 寄存器中的 TXFNF = 1 时，生成 USART 中断

位 6 **TCIE**：发送完成中断使能 (Transmission complete interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：当 USART_ISR 寄存器中的 TC = 1 时，生成 USART 中断

位 5 **RXFNEIE**：RXFIFO 非空中断使能 (RXFIFO not empty interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：当 USART_ISR 寄存器中的 ORE = 1 或 RXFNE = 1 时，生成 USART 中断

位 4 **IDLEIE**：IDLE 中断使能 (IDLE interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：当 USART_ISR 寄存器中的 IDLE = 1 时，生成 USART 中断

位 3 **TE**：发送器使能 (Transmitter enable)

该位使能发送器。该位由软件置 1 和清零。

0：禁止发送器

1：使能发送器

注意：除了在智能卡模式下以外，传送期间 TE 位上的低电平脉冲（“0”后紧跟的是“1”）会在当前字的后面发送一个前导码（空闲线路）。为生成空闲字符，TE 不能立即写入“1”。为确保所需的持续时间，软件可轮询 USART_ISR 寄存器中的 TEACK 位。

在智能卡模式下，当 TE 置 1 时，在发送开始前存在 1 位的时间延迟。

位 2 **RE**：接收器使能 (Receiver enable)

该位使能接收器。该位由软件置 1 和清零。

0：禁止接收器

1：使能接收器并开始搜索起始位

位 1 **UESM**：低功耗模式下的 USART 使能 (USART enable in low-power mode)

当该位清零时，USART 无法将 MCU 从低功耗模式唤醒。

当该位置 1 时，USART 可将 MCU 从低功耗模式唤醒。

该位由软件置 1 和清零，

0：USART 无法将 MCU 从低功耗模式唤醒。

1：USART 能够将 MCU 从低功耗模式唤醒。

注意：建议在进入低功耗模式前将 UESM 位置 1，并在退出低功耗模式时将其清零。

如果 USART 不支持从 Stop 模式唤醒功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 0 **UE**：USART 使能 (USART enable)

该位清零后，USART 预分频器和输出将立即停止，并丢弃所有当前操作。USART 配置会保留，而所有的 USART_ISR 状态标志均会复位。该位由软件置 1 和清零，

0：禁止 USART 预分频器和输出，低功耗模式

1：使能 USART

注意：为进入低功耗模式而不在线路上生成错误，之前必须复位 TE 位，并且软件必须等待 USART_ISR 中的 TC 位置 1 后才能复位 UE 位。

UE = 0 时也会复位 DMA 请求，因此必须在复位 UE 位前禁止 DMA 通道。

在智能卡模式下 (SCEN = 1)，无论 UE 位值为何，当 CLKEN = 1 时，CK 始终可用。

49.8.2 USART 控制寄存器 1 [复用] (USART_CR1)

USART control register 1

偏移地址：0x00

复位值：0x0000 0000

同一寄存器可用于使能 FIFO 模式（上一节）和禁止 FIFO 模式（本节）的情况。

禁止 FIFO 模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	FIFO EN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]				
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29 **FIFOEN**：FIFO 模式使能 (FIFO mode enable)

该位由软件置 1 和清零，

0：禁止 FIFO 模式。

1：使能 FIFO 模式。

只有在禁止 USART (UE = 0) 时才能写入该位字段。

注意：FIFO 模式只能在标准 UART 通信、SPI 主/从设备模式和智能卡模式下使用，不得在 IrDA 和 LIN 模式下使能。

位 28 **M1**：字长 (Word length)

该位必须与位 12 (M0) 搭配使用来确定字长度。该位由软件置 1 或清零。

M[1:0] = “00”：1 个起始位，8 个数据位，n 个停止位

M[1:0] = “01”：1 个起始位，9 个数据位，n 个停止位

M[1:0] = “10”：1 个起始位，7 个数据位，n 个停止位

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：在 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。

位 27 **EOBIE**：块结束中断使能 (End of Block interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：USART_ISR 寄存器中的 EOBF 标志置 1 时生成 USART 中断

注意：如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 26 **RTOIE**：接收器超时中断使能 (Receiver timeout interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：USART_ISR 寄存器中的 RTOF 位置 1 时生成 USART 中断。

注意：如果 USART 不支持接收器超时功能，该位保留且必须保持复位值。第 2436 页的第 49.4 节：USART 实现.

位 25:21 **DEAT[4:0]**：驱动器使能有效时间 (Driver enable assertion time)

该 5 位值用于定义激活 DE（启动器使能）信号与起始位开始间的时间。此时间以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

只有在禁止 USART (UE = 0) 时才能写入该位字段。

注意：如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 20:16 **DEDT[4:0]**：驱动器使能无效时间 (Driver enable deassertion time)

该 5 位值用于定义发送的消息中最后一个停止位结束与取消激活 DE（驱动器使能）信号间的时间。此时间以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

如果在 DEDT 时间内对 USART_TDR 寄存器执行写操作，则新数据仅在经过 DEDT 和 DEAT 时间后才会发送。

只有在禁止 USART (UE = 0) 时才能写入该位字段。

注意：如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 15 **OVER8**：过采样模式 (Oversampling mode)

0：16 倍过采样

1：8 倍过采样

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：在 LIN、IrDA 和智能卡模式下，该位必须保持清零。

位 14 **CMIE**：字符匹配中断使能 (Character match interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：USART_ISR 寄存器中的 CMF 位置 1 时生成 USART 中断。

位 13 **MME**：静默模式使能 (Mute mode enable)

该位可使能 USART 静默模式功能。该位置 1 时，USART 会按照 WAKE 位的定义在活动模式与静默模式之间切换。该位由软件置 1 和清零。

0：接收器永久处于活动模式

1：接收器可在静默模式和活动模式之间切换。

位 12 **M0**：字长 (Word length)

该位与位 28 (M1) 搭配使用来确定字长度。它由软件置 1 或清零（请参见位 28 (M1) 的说明）。

只有在禁止 USART (UE = 0) 时才能写入该位。

位 11 **WAKE**：接收器唤醒方法 (Receiver wake-up method)

该位用于确定 USART 静默模式的唤醒方法。该位由软件置 1 或清零。

0：空闲线路

1：地址标记

只有在禁止 USART (UE = 0) 时才能写入该位字段。

位 10 **PCE**：奇偶校验控制使能 (Parity control enable)

该位选择硬件奇偶校验控制（生成和检测）。使能奇偶校验控制时，计算出的奇偶校验位被插入到 MSB 位置（如果 M = 1，则为第 9 位；如果 M = 0，则为第 8 位），并对接收到的数据检查奇偶校验位。该位由软件置 1 和清零，一旦该位置 1，PCE 在当前字节的后面处于活动状态（在接收和发送时）。

0：禁止奇偶校验控制

1：使能奇偶校验控制

只有在禁止 USART (UE = 0) 时才能写入该位字段。

位 9 **PS**：奇偶校验选择 (Parity selection)

该位用于在使能奇偶校验生成/检测（PCE 位置 1）时选择奇校验或偶校验。该位由软件置 1 和清零。在当前字节的后面选择奇偶校验。

0：偶校验

1：奇校验

只有在禁止 USART (UE = 0) 时才能写入该位字段。

位 8 **PEIE**：PE 中断使能 (PE interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：当 USART_ISR 寄存器中的 PE = 1 时，生成 USART 中断

位 7 **TXEIE**：发送数据寄存器为空

该位由软件置 1 和清零，

0：禁止中断

1：当 USART_ISR 寄存器中的 TXE = 1 时，生成 USART 中断

位 6 **TCIE**：发送完成中断使能 (Transmission complete interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：当 USART_ISR 寄存器中的 TC = 1 时，生成 USART 中断

位 5 **RXNEIE**：接收数据寄存器非空 (Receive data register not empty)

该位由软件置 1 和清零，

0：禁止中断

1：当 USART_ISR 寄存器中的 ORE = 1 或 RXNE = 1 时，生成 USART 中断

位 4 **IDLEIE**：IDLE 中断使能 (IDLE interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：当 USART_ISR 寄存器中的 IDLE = 1 时，生成 USART 中断

位 3 **TE**：发送器使能 (Transmitter enable)

该位使能发送器。该位由软件置 1 和清零。

0：禁止发送器

1：使能发送器

注意：除了在智能卡模式下以外，传送期间 TE 位上的低电平脉冲（“0”后紧跟的是“1”）会在当前字的后面发送一个前导码（空闲线路）。为生成空闲字符，TE 不能立即写入“1”。为确保所需的持续时间，软件可轮询 USART_ISR 寄存器中的 TEACK 位。

在智能卡模式下，当 TE 置 1 时，在发送开始前存在 1 位的时间延迟。

位 2 **RE**：接收器使能 (Receiver enable)

该位使能接收器。该位由软件置 1 和清零。

0：禁止接收器

1：使能接收器并开始搜索起始位

位 1 **UESM**：低功耗模式下的 USART 使能 (USART enable in low-power mode)

当该位清零时，USART 无法将 MCU 从低功耗模式唤醒。

当该位置 1 时，USART 可将 MCU 从低功耗模式唤醒。

该位由软件置 1 和清零，

0：USART 无法将 MCU 从低功耗模式唤醒。

1：USART 能够将 MCU 从低功耗模式唤醒。

注意：建议在进入低功耗模式前将 UESM 位置 1，并在退出低功耗模式时将其清零。

如果 USART 不支持从 Stop 模式唤醒功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 0 **UE**：USART 使能 (USART enable)

该位清零后，USART 预分频器和输出将立即停止，并丢弃所有当前操作。USART 配置会保留，而所有的 USART_ISR 状态标志均会复位。该位由软件置 1 和清零，

0：禁止 USART 预分频器和输出，低功耗模式

1：使能 USART

注意：为进入低功耗模式而不在线路上生成错误，之前必须复位 TE 位，并且软件必须等待 USART_ISR 中的 TC 位置 1 后才能复位 UE 位。

UE = 0 时也会复位 DMA 请求，因此必须在复位 UE 位前禁止 DMA 通道。

在智能卡模式下 (SCEN = 1)，无论 UE 位值为何，当 CLKEN = 1 时，CK 引脚始终可用。

49.8.3 USART 控制寄存器 2 (USART_CR2)

USART control register 2

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD[7:0]								RTOEN	ABRMOD[1:0]		ABREN	MSBFIRST	DATAINV	TXINV	RXINV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	DIS_NSS	Res.	Res.	SLVEN
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw			rw

位 31:24 **ADD[7:0]**：USART 节点的地址 (Address of the USART node)

这些位用于指定静默模式下 USART 节点的地址或低功耗模式/运行模式下要识别的字符代码。

- 在静默模式下：它们用于在多处理器通信时通过 4 位/7 位地址标记检测从静默模式唤醒。发送器发送字符的 MSB 应为 1。在 4 位地址标记检测中，只使用 ADD[3:0] 位。
- 在低功耗模式下：它们用于在字符匹配时从低功耗模式唤醒。
  如果将 WUS[1:0] 编程为 0b00（WUF 在地址匹配时激活），并且已通过将 WUFIE 位置 1 使能 WUF 中断，则当收到的字符与通过 ADD[6:0] 或 ADD[3:0] 位字段（取决于 ADDM7 位）编程的字符匹配时会从低功耗模式唤醒。发送器发送字符的 MSB 应为 1。
- 在静默模式无效时的运行模式下（例如，ModBus 协议中的块结束检测）：接收到的整个字符（8 位）将与 ADD[7:0] 值进行比较，如果匹配，CMF 标志将置 1。如果 CMIE 位置 1，则会生成中断。

仅在禁止接收 (RE = 0) 或禁止 USART (UE = 0) 时才能写入这些位。

位 23 **RTOEN**：接收器超时使能 (Receiver timeout enable)

该位由软件置 1 和清零，

0：禁止接收器超时功能。

1：使能接收器超时功能。

使能此功能后，如果 RX 线路在 RTOR（接收器超时寄存器）中编程的持续时间内处于空闲状态（无接收），则 USART_ISR 寄存器中的 RTOF 标志置 1。

注意：如果 USART 不支持接收器超时功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 22:21 **ABRMOD[1:0]**：自动波特率模式 (Auto baud rate mode)

这些位由软件置 1 和清零，

00：通过测量起始位检测波特率。

01：下降沿到下降沿的测量（接收到的帧必须以一个等于 1 的位开头，即帧 = 10xxxxxx）

10：0x7F 帧检测。

11：0x55 帧检测。

仅在 ABREN = 0 时或禁止 USART (UE = 0) 时才能写入该位字段。

注意：如果 DATAINV = 1 且/或 MSBFIRST = 1，这些模式必须与在线路上时相同，例如 MSBFIRST 的 0xAA。

如果 USART 不支持自动波特率功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 20 **ABREN**：自动波特率检测使能 (Auto baud rate enable)

该位由软件置 1 和清零，

0：禁止自动波特率检测。

1：使能自动波特率检测。

注意：如果 USART 不支持自动波特率功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 19 **MSBFIRST**：最高有效位在前 (Most significant bit first)

该位由软件置 1 和清零，

0：发送/接收数据时位 0 在前，后跟起始位。

1：发送/接收数据时 MSB（位 7/8）在前，后跟起始位。

只有在禁止 USART (UE = 0) 时才能写入该位字段。

位 18 **DATAINV**：二进制数据反向 (Binary data inversion)

该位由软件置 1 和清零，

0：按正/正向逻辑发送/接收数据寄存器中的逻辑数据。（1 = H，0 = L）。

1：按负/反向逻辑发送/接收数据寄存器中的逻辑数据。（1 = L，0 = H）。奇偶校验位也取反。

只有在禁止 USART (UE = 0) 时才能写入该位字段。

位 17 **TXINV**：TX 引脚有效电平反向 (TX pin active level inversion)

该位由软件置 1 和清零，

0：TX 引脚信号使用标准逻辑电平（V_{DD} = 1/空闲，Gnd = 0/标记）工作。

1：TX 引脚信号值取反（V_{DD} = 0/标记，Gnd = 1/空闲）。

允许在 TX 线路上使用外部反相器。

只有在禁止 USART (UE = 0) 时才能写入该位字段。

位 16 **RXINV**：RX 引脚有效电平反向 (RX pin active level inversion)

该位由软件置 1 和清零，

0：RX 引脚信号使用标准逻辑电平（V_{DD} = 1/空闲，Gnd = 0/标记）工作。

1：RX 引脚信号值取反（V_{DD} = 0/标记，Gnd = 1/空闲）。

允许在 RX 线路上使用外部反相器。

只有在禁止 USART (UE = 0) 时才能写入该位字段。

位 15 **SWAP**：交换 TX/RX 引脚 (Swap TX/RX pins)

该位由软件置 1 和清零，

0：按标准引脚排列定义使用 TX/RX 引脚

1：交换 TX 和 RX 引脚功能。允许在与另一个 UART 的交叉连接时工作。

只有在禁止 USART (UE = 0) 时才能写入该位字段。

位 14 **LINEN**：LIN 模式使能 (LIN mode enable)

该位由软件置 1 和清零，

0：禁止 LIN 模式

1：使能 LIN 模式

LIN 模式可以使用 USART_CR1 寄存器中的 SBKRQ 位发送 LIN 同步刹车（13 个低位），并可检测 LIN 同步刹车。

只有在禁止 USART (UE = 0) 时才能写入该位字段。

注意：如果 USART 不支持 LIN 模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 13:12 **STOP[1:0]**：停止位

这些位用于编程停止位。

00：1 个停止位

01：0.5 个停止位

10：2 个停止位

11：1.5 个停止位

只有在禁止 USART (UE = 0) 时才能写入该位字段。

位 11 **CLKEN**：时钟使能 (Clock enable)

该位允许用户使能 CK 引脚。

0：禁止 CK 引脚

1：使能 CK 引脚

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果既不支持同步模式，也不支持智能卡模式，则该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

在智能卡模式下，为了向智能卡正确提供 CK 时钟，必须按以下步骤操作：

UE = 0

SCEN = 1

GTPR 配置

CLKEN= 1

UE = 1

位 10 **CPOL**：时钟极性 (Clock polarity)

该位允许用户在同步模式下选择 CK 引脚上时钟输出的极性。它与 CPHA 位结合使用可获得所需的时钟/数据关系

0：空闲时 CK 引脚为低电平

1：空闲时 CK 引脚为高电平

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持同步模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 9 **CPHA**：时钟相位 (Clock phase)

该位用于在同步模式下选择 CK 引脚上时钟输出的相位。它与 CPOL 位结合使用可获得所需的时钟/数据关系（请参见*图 570* 和*图 571*）

0：从第一个时钟边沿开始采样数据

1：从第二个时钟边沿开始采样数据

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持同步模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 8 **LBCL**：最后一个位时钟脉冲 (Last bit clock pulse)

该位用于在同步模式下选择与发送的最后一个数据位 (MSB) 关联的时钟脉冲是否必须在 CK 引脚上输出。

0：最后一个数据位的时钟脉冲不在 CK 引脚上输出

1：最后一个数据位的时钟脉冲在 CK 引脚上输出

小心： 最后一位为发送的第 7 个、第 8 个或第 9 个数据位，具体取决于 USART_CR1 寄存器中 M 位所选择的 7 位、8 位或 9 位格式。

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持同步模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 7 保留，必须保持复位值。

位 6 **LBDIE**：LIN 刹车检测中断使能 (LIN break detection interrupt enable)

刹车中断屏蔽（使用刹车分隔符进行刹车检测）。

0：禁止中断

1：当 USART_ISR 寄存器中 LBDF = 1 时，生成中断

注意：如果不支持 LIN 模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 5 **LBDL**：LIN 刹车检测长度 (LIN break detection length)

该位用于选择 11 位刹车检测或 10 位刹车检测。

0：10 位刹车检测

1：11 位刹车检测

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持 LIN 模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 4 **ADDM7**：7 位地址检测/4 位地址检测 (7-bit address detection/4-bit address detection)

该位用于选择 4 位地址检测或 7 位地址检测。

0：4 位地址检测

1：7 位地址检测（在 8 位数据模式下）

只有在禁止 USART (UE = 0) 时才能写入该位

注意：在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0] 和 ADD[7:0]）。

位 3 **DIS_NSS**：NSS 引脚使能 (NSS pin enable)

当 DIS_NSS 位置 1 时，忽略 NSS 引脚输入。

0：SPI 从设备选择取决于 NSS 输入引脚。

1：始终选择 SPI 从设备，忽略 NSS 输入引脚。

注意：不支持 SPI 从设备模式时，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 2:1 保留，必须保持复位值。

位 0 **SLVEN**：同步从模式使能 (Synchronous Slave mode enable)

SLVEN 位置 1 时，使能同步从模式。

0：禁止从模式。

1：使能从模式。

注意：不支持 SPI 从设备模式时，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

注：使能发送器时不应对 CPOL、CPHA 和 LBCL 位进行写操作。

49.8.4 USART 控制寄存器 3 (USART_CR3)

USART control register 3

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXFTCFG[2:0]			RXF TIE	RXFTCFG[2:0]			TCBG TIE	TXFTIE	WUFIE	WUS[1:0]		SCARCNT[2:0]			Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVR DIS	ONE BIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HD SEL	IRLP	IREN	EIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 **TXFTCFG[2:0]**：TXFIFO 阈值配置 (TXFIFO threshold configuration)

000：TXFIFO 达到其深度的 1/8
001：TXFIFO 达到其深度的 1/4
010：TXFIFO 达到其深度的 1/2
011：TXFIFO 达到其深度的 3/4
100：TXFIFO 达到其深度的 7/8
101：TXFIFO 变空
其余组合：保留

位 28 **RXFTIE**：RXFIFO 阈值中断使能 (FIFO threshold interrupt enable)

该位由软件置 1 和清零，
0：禁止中断
1：当接收 FIFO 达到 RXFTCFG 中编程的阈值时，生成 USART 中断。

位 27:25 **RXFTCFG[2:0]**：接收 FIFO 阈值配置 (Receive FIFO threshold configuration)

000：接收 FIFO 达到其深度的 1/8
001：接收 FIFO 达到其深度的 1/4
010：接收 FIFO 达到其深度的 1/2
011：接收 FIFO 达到其深度的 3/4
100：接收 FIFO 达到其深度的 7/8
101：接收 FIFO 已满
其余组合：保留

位 24 **TCBGTIE**：保护时间前发送完成中断使能 (Transmission complete before guard time, interrupt enable)

该位由软件置 1 和清零，
0：禁止中断
1：当 USART_ISR 寄存器中的 TCBGT=1 时，生成 USART 中断

注意：如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 23 **TXFTIE**：TXFIFO 阈值中断使能 (TXFIFO threshold interrupt enable)

该位由软件置 1 和清零，
0：禁止中断
1：当 TXFIFO 达到 TXFTCFG 中编程的阈值时，生成 USART 中断。

位 22 **WUFIE**：从低功耗模式唤醒中断使能 (Wake-up from low-power mode interrupt enable)

该位由软件置 1 和清零，

0：禁止中断

1：当 USART_ISR 寄存器中的 WUF = 1 时，生成 USART 中断

注意：WUFIE 必须在进入低功耗模式前置 1。

如果 USART 不支持从 Stop 模式唤醒功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 21:20 **WUS[1:0]**：从低功耗模式唤醒中断标志选择 (Wake-up from low-power mode interrupt flag selection)

该位字段用于指定激活 WUF（从低功耗模式唤醒标志）的事件。

00：WUF 在地址匹配时激活（按 ADD[7:0] 和 ADDM7 所定义）

01：保留。

10：WUF 在起始位检测时激活

11：WUF 在 RXNE/RXFNE 时激活。

只有在禁止 USART (UE = 0) 时才能写入该位字段。

如果 USART 不支持从 Stop 模式唤醒功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 19:17 **SCARCNT[2:0]**：智能卡自动重试计数 (Smartcard auto-retry count)

该位字段用于指定智能卡模式下发送和接收的重试次数。

在发送模式下，该位字段用于指定生成发送错误（FE 位置 1）前自动重新发送的重试次数。

在接收模式下，该位字段用于指定生成接收错误（RXNE/RXFNE 位和 PE 位置 1）前错误接收尝试的次数。

只有在禁止 USART (UE = 0) 时才能编程该位字段。

使能 USART (UE = 1) 时，该位字段只能写入 0x0，以停止重新发送。

0x0：禁止重新发送——发送模式下不会自动重新发送。

0x1 到 0x7：自动重新发送的尝试次数（发出错误信号前）

注意：如果不支持智能卡模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 16 保留，必须保持复位值。

位 15 **DEP**：驱动器使能极性选择 (Driver enable polarity selection)

0：DE 信号高电平有效。

1：DE 信号低电平有效。

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 14 **DEM**：驱动器使能模式 (Driver enable mode)

该位用于通过 DE 信号激活外部收发器控制。

0：禁止 DE 功能。

1：使能 DE 功能。DE 信号在 RTS 引脚上输出。

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持驱动器使能功能，该位保留且必须保持复位值。第 2436 页的第 49.4 节：USART 实现.

位 13 **DDRE**：接收出错时的 DMA 禁止 (DMA Disable on reception error)

0：接收出错时不禁止 DMA。相应的错误标志置 1，但 RXNE 保持为 0 以防止上溢。因此，将不使能 DMA 请求，从而不会传送错误数据（无 DMA 请求），但会传送接收到的下一个正确数据（用于智能卡模式）。

1：接收出错后禁止 DMA。相应的错误标志以及 RXNE 均置 1。屏蔽 DMA 请求，直到错误标志清零。这意味着软件必须首先禁止 DMA 请求 (DMAR = 0) 或者将 RXNE（使能 FIFO 模式时为 RXFNE）清零，然后再将错误标志清零。

只有在禁止 USART (UE=0) 时才能写入该位。

注意：接收错误包括：奇偶校验错误、帧错误或噪声错误。

位 12 **OVRDIS**：上溢禁止 (Overrun disable)

该位用于禁止接收上溢检测。

0：接收新数据前未读取已接收的数据时，上溢错误标志 ORE 置 1。

1：禁止上溢功能。如果在 RXNE 标志仍置 1 时接收到新数据，则 ORE 标志不会置 1，且新接收的数据会覆盖 USART_RDR 寄存器之前的内容。使能 FIFO 模式时，RXFIFO 将被旁路，数据将直接写入 USART_RDR 寄存器中。即使在使能 FIFO 管理时，也将使用 RXNE 标志。

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：此控制位用于检查通信流而不会读取数据

位 11 **ONEBIT**：一个采样位方法使能 (One sample bit method enable)

该位允许用户选择采样方法。选择一个采样位方法后，将禁止噪声检测标志 (NE)。

0：三个采样位方法

1：一个采样位方法

只有在禁止 USART (UE = 0) 时才能写入该位。

位 10 **CTSIE**：CTS 中断使能 (CTS interrupt enable)

0：禁止中断

1：当 USART_ISR 寄存器中 CTSIF = 1 时，生成中断

注意：如果不支持硬件流控制功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 9 **CTSE**：CTS 使能 (CTS enable)

0：禁止 CTS 硬件流控制

1：使能 CTS 模式，仅当 CTS 输入无效（连接到 0）时才发送数据。如果在发送数据时使 CTS 输入有效，会在停止之前完成发送。如果使 CTS 有效时数据已写入数据寄存器，则将延迟发送，直到 CTS 无效。

只有在禁止 USART (UE = 0) 时才能写入该位

注意：如果不支持硬件流控制功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 8 **RTSE**：RTS 使能 (RTS enable)

0：禁止 RTS 硬件流控制

1：使能 RTS 输出，仅当接收缓冲区中有空间时才会请求数据。发送完当前字符后应停止发送数据。可以接收数据时使 RTS 输出无效（拉至 0）。

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持硬件流控制功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 7 **DMAT**：DMA 使能发送器 (DMA enable transmitter)

该位由软件置 1/复位。

1：针对发送使能 DMA 模式

0：针对发送禁止 DMA 模式

位 6 **DMAR**：DMA 使能接收器 (DMA enable receiver)

该位由软件置 1/复位。

1：针对接收使能 DMA 模式

0：针对接收禁止 DMA 模式

位 5 **SCEN**：智能卡模式使能 (Smartcard mode enable)

该位用于使能智能卡模式。

0：禁止智能卡模式

1：使能智能卡模式

只有在禁止 USART (UE = 0) 时才能写入该位字段。

注意：如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 4 **NACK**：智能卡 NACK 使能 (Smartcard NACK enable)

0：出现奇偶校验错误时禁止 NACK 发送

1：出现奇偶校验错误时使能 NACK 发送

只有在禁止 USART (UE = 0) 时才能写入该位字段。

注意：如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 3 **HDSEL**：半双工选择 (Half-duplex selection)

选择单线半双工模式

0：未选择半双工模式

1：选择半双工模式

只有在禁止 USART (UE = 0) 时才能写入该位。

位 2 **IRLP**：IrDA 低功耗 (IrDA low-power)

该位用于选择正常模式和低功耗 IrDA 模式

0：正常模式

1：低功耗模式

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持 IrDA 模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 1 **IREN**：IrDA 模式使能 (IrDA mode enable)

该位由软件置 1 和清零，

0：禁止 IrDA

1：使能 IrDA

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持 IrDA 模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 0 **EIE**：错误中断使能 (Error interrupt enable)

如果出现帧错误、上溢错误、噪声标志或 SPI 从设备下溢错误（USART_ISR 寄存器中的 FE = 1、ORE = 1、NE = 1 或 UDR = 1），则需要使用错误中断使能位来使能中断生成。

0：禁止中断

1：USART_ISR 寄存器中的 FE = 1、ORE = 1、NE = 1 或 UDR = 1（在 SPI 从设备模式下）时，生成中断。

49.8.5 USART 波特率寄存器 (USART_BRR)

USART baud rate register

只有在禁止 USART (UE = 0) 时才能写入此寄存器。在自动波特率检测模式下，该位由硬件自动更新。

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **BRR[15:0]**：USART 波特率 (USART baud rate)

BRR[15:4]

BRR[15:4] = USARTDIV[15:4]

BRR[3:0]

当 OVER8 = 0 时，BRR[3:0] = USARTDIV[3:0]。

当 OVER8 = 1 时：

BRR[2:0] = USARTDIV[3:0] 右移 1 位。

BRR[3] 必须保持清零。

49.8.6 USART 保护时间和预分频器寄存器 (USART_GTPR)

USART guard time and prescaler register

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GT[7:0]								PSC[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:8 **GT[7:0]**：保护时间值 (Guard time value)

该位字段用于编程保护时间值（以波特时钟周期数为单位）。

该位用于智能卡模式。经过此保护时间后，发送完成标志置 1。

只有在禁止 USART (UE = 0) 时才能写入该位字段。

注意：如果不支持智能卡模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 7:0 **PSC[7:0]**：预分频器值 (Prescaler value)

在 IrDA 低功耗和正常的 IrDA 模式下：

PSC[7:0] = IrDA 正常和低功耗波特率

PSC[7:0] 用于编程预分频器，进行 USART 源时钟分频以获得低功耗频率：使用寄存器中给出的值（8 个有效位）对源时钟进行分频：

在智能卡模式下：

PSC[4:0] = 预分频器值

PSC[4:0] 用于编程预分频器，进行 USART 源时钟分频以提供智能卡时钟。将寄存器中给出的值（5 个有效位）乘以 2 得出源时钟频率的分频系数：

00000：保留 - 不编程此值
00001：源时钟 1 分频（IrDA 模式）/2 分频（智能卡模式）
00010：源时钟 2 分频（IrDA 模式）/4 分频（智能卡模式）
00011：源时钟 3 分频（IrDA 模式）/6 分频（智能卡模式）
...
11111: 源时钟 31 分频（IrDA 模式）/62 分频（智能卡模式）
0010 0000：源时钟 32 分频（IrDA 模式）
...
1111 1111：源时钟 255 分频（IrDA 模式）

只有在禁止 USART (UE = 0) 时才能写入该位字段。

注意：如果使用智能卡模式，则位 [7:5] 必须保持清零。
不支持智能卡和 IrDA 模式时，该位字段保留并由硬件强制清零。请参见第 2436 页的第 49.4 节：USART 实现。

49.8.7 USART 接收器超时寄存器 (USART_RTOR)

USART receiver timeout register

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BLEN[7:0]								RTO[23:16]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTO[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **BLEN[7:0]**：块长度

该位字段用于提供智能卡 T = 1 接收下的块长度。其值等于信息字符数 + 结尾字段的长度 (1-LEC/2-CRC) - 1。

例如：

BLEN = 0：0 个信息字符 + LEC

BLEN = 1：0 个信息字符 + CRC

BLEN = 255：254 个信息字符 + CRC（总共 256 个字符）

在智能卡模式下，块长度计数器在 TXE = 0（使能 FIFO 模式时为 TXFE = 0）时复位。

该位字段也可用于其他模式。在这种情况下，块长度计数器在 RE = 0（禁止接收器）和/或 EOBCF 位写入 1 时复位。

注意：块接收开始后可编程此值（使用起始字段中 LEN 字符中的数据）。每个接收到的块只能对此值编程一次。

位 23:0 **RTO[23:0]**：接收器超时值 (Receiver timeout value)

该位字段根据 RX 线上没有活动的位数分配接收器超时值。

在标准模式下，如果在接收到最后一个字符后，在 RTO 值对应的时间内未检测到新的起始位，则 RTOF 标志置 1。

在智能卡模式下，此值用于实施 CWT 和 BWT。有关更多详细信息，请参见智能卡章节。在标准模式下，从接收到的最后一个字符的起始位开始执行 CWT/BWT 测量。

注意：每个接收到的字符只能对此值编程一次。

注：可以实时写入 RTOR。如果新值小于或等于计数器的值，RTOF 标志置 1。

如果不支持接收器超时功能，此寄存器保留并由硬件强制为“0x00000000”。请参见第 2436 页的第 49.4 节：USART 实现。

49.8.8 USART 请求寄存器 (USART_RQR)

USART request register

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ
											w	w	w	w	w

位 31:5 保留，必须保持复位值。

位 4 **TXFRQ**：发送数据刷新请求 (Transmit data flush request)

禁止 FIFO 模式时，向该位写入“1”会将 TXE 标志置 1。这可丢弃发送数据。由于错误 (NACK) 而未发送数据且 USART_ISR 寄存器中的 FE 标志有效时，只能在智能卡模式下使用该位。如果 USART 不支持智能卡模式，该位保留且必须保持复位值。

使能 FIFO 时，TXFRQ 位置 1 以清空整个 FIFO。这会将 TXFE 标志（发送 FIFO 为空，USART_ISR 寄存器中的位 23）置 1。在 UART 模式和智能卡模式下都支持清空发送 FIFO。

注意：在 FIFO 模式下，TXFNF 标志在清空请求期间复位，直到 TxFIFO 为空，以确保数据寄存器中没有写入数据。

位 3 **RXFRQ**：接收数据刷新请求 (Receive data flush request)

向该位写入 1 将清空整个接收 FIFO（即，将 RXFNE 位清零）。

这可以丢弃接收的数据而不对其执行读取操作，并避免发生上溢情况。

位 2 **MMRQ**：静默模式请求 (Mute mode request)

向该位写入 1 可将 USART 置于静默模式，并将 RWU 标志复位。

位 1 **SBKRQ**：发送刹车请求 (Send break request)

向该位写入 1 可将 SBKF 标志置 1 并在发送设备可用后立即请求在线路上发送 BREAK。

注意：如果应用需要在之前已插入的所有数据（包括尚未发送的数据）后发送刹车字符，软件应等到 TXE 标志使能后将 SBKRQ 位置 1。

位 0 **ABRRQ**：自动波特率请求 (Auto baud rate request)

向该位写入 1 可复位 USART_ISR 中的 ABRF 和 ABRE 标志，并请求对下一个接收到的数据帧进行自动波特率测量。

注意：如果 USART 不支持自动波特率功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

49.8.9 USART 中断和状态寄存器 (USART_ISR)

USART interrupt and status register

偏移地址：0x1C

复位值：0x0X80 00C0

使能 FIFO/智能卡模式时，X = 2

使能 FIFO 模式且禁止智能卡模式时，X = 0

同一寄存器可用于使能 FIFO 模式（本节）和禁止 FIFO 模式（下一节）的情况。

使能 FIFO 模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXFT	RXFT	TCBGT	RXFF	TXFE	RE ACK	TE ACK	WUF	RWU	SBKF	CMF	BUSY
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXFNF	TC	RXFNE	IDLE	ORE	NE	FE	PE
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27 **TXFT**：TXFIFO 阈值标志 (FIFO threshold flag)

当 TXFIFO 达到在 USART_CR3 寄存器的 TXFTCFG 中编程的阈值时（即 TXFIFO 包含 TXFTCFG 个空位置），该位由硬件置 1。如果 USART_CR3 寄存器中的 TXFTIE 位 = 1（位 31），则会生成中断。

0：TXFIFO 未达到编程的阈值。

1：TXFIFO 已达到编程的阈值。

位 26 **RXFT**：RXFIFO 阈值标志 (RXFIFO threshold flag)

达到在 USART_CR3 寄存器的 RXFTCFG 中编程的阈值时，该位由硬件置 1。这意味着，接收 FIFO 中有 (RXFTCFG - 1) 个数据，USART_RDR 寄存器中有一个数据。如果 USART_CR3 寄存器中的 RXFTIE 位 = 1（位 27），则会生成中断。

0：接收 FIFO 未达到编程的阈值。

1：接收 FIFO 已达到编程的阈值。

注意：当 RXFTCFG 阈值配置为“101”时，如果存在 16 个数据（即 RXFIFO 中有 15 个数据，USART_RDR 中有 1 个数据），则 RXFT 标志将置 1。因此，接收到的第 17 个数据不会导致上溢错误。接收到第 18 个数据后会发生上溢错误。

位 25 **TCBGT**：保护时间前发送完成标志 (Transmission complete before guard time flag)

当写入到 USART_TDR 中的最后一个数据已正确从移位寄存器中发出时，该位置 1。

如果包含数据的帧完成发送，并且智能卡未发回任何 NACK，则该位在智能卡模式下由硬件置 1。如果 USART_CR3 寄存器中的 TCBGTIE = 1，则会生成中断。

该位由软件清零，方法是向 USART_ICR 寄存器中的 TCBGTCF 写入 1 或向 USART_TDR 寄存器执行写操作。

0：发送未完成或发送未成功完成（即，从智能卡接收到 NACK）

1：发送成功完成（在保护时间结束之前完成，智能卡未发送 NACK）。

注意：如果 USART 不支持智能卡模式，该位保留且保持复位值。如果 USART 支持智能卡模式并使能智能卡模式，则 TCBGT 复位值为“1”。请参见第 2436 页的第 49.4 节：USART 实现。

位 24 **RXFF**：RxFIFO 已满 (RXFIFO full)

当接收到的数据量对应于 RXFIFO 大小 + 1（RXFIFO 已满 + USART_RDR 寄存器中的 1 个数据）时，该位由硬件置 1。

如果 USART_CR1 寄存器中 RXFFIE 位 = 1，则会生成中断。

0：RXFIFO 未满。

1：RXFIFO 已满。

位 23 **TXFE**：TXFIFO 为空 (TXFIFO empty)

当 TXFIFO 为空时，该位由硬件置 1。当 TXFIFO 包含至少一个数据时，该标志被清零。也可以通过向 USART_RQR 寄存器中的位 TXFRQ（位 4）写入 1 将 TXFE 标志置 1。

如果 USART_CR1 寄存器中的 TXFEIE 位 = 1（位 30），则会生成中断。

0：TXFIFO 非空。

1：TXFIFO 为空。

位 22 **REACK**：接收使能确认标志 (Receive enable acknowledge flag)

USART 采用接收使能值时，通过硬件将该位置 1/复位。

该位可用于验证 USART 是否准备好在进入低功耗模式前接收数据。

注意：如果 USART 不支持从 Stop 模式唤醒功能，该位保留且保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 21 **TEACK**：发送使能确认标志 (Transmit enable acknowledge flag)

USART 采用发送使能值时，通过硬件将该位置 1/复位。

通过写入 TE = 0 生成空闲帧请求，然后在 USART_CR1 寄存器中写入 TE = 1 以遵循 TE = 0 最短周期时，可使用该位。

位 20 **WUF**：从低功耗模式唤醒标志 (Wake-up from low-power mode flag)

当检测到唤醒事件时，该位由硬件置 1。事件通过 WUS 位字段定义。该位由软件清零，方法是向 USART_ICR 寄存器中的 WUCF 写入 1。

如果 USART_CR3 寄存器中 WUFIE = 1，则会生成中断。

注意：当 UESM 清零时，WUF 标志也清零。

如果 USART 不支持从 Stop 模式唤醒功能，该位保留且保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 19 **RWU**：接收器从静默模式唤醒 (Receiver wake-up from Mute mode)

该位指示 USART 是否处于静默模式。当识别出唤醒/静默序列时，该位由硬件清零/置 1。静默模式控制序列（地址或 IDLE）通过 USART_CR1 寄存器中的 WAKE 位进行选择。

当选择 IDLE 模式下唤醒时，该位只能通过用软件向 USART_RQR 寄存器中的 MMRQ 位写 1 的方式置 1。

0：接收器处于活动模式

1：接收器处于静默模式

注意：如果 USART 不支持从 Stop 模式唤醒功能，该位保留且保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 18 **SBKF**：发送刹车标志 (Send break flag)

该位指示已请求发送刹车字符。通过将 1 写入 USART_CR3 寄存器中的 SBKRQ 位，该位由软件置 1。该位在刹车发送的停止位期间由硬件自动复位。

0：刹车字符已发送

1：通过将 USART_RQR 寄存器中的 SBKRQ 位置 1 请求刹车字符

位 17 **CMF**：字符匹配标志 (Character match flag)

接收到由 ADD[7:0] 定义的字符后由硬件将该位置 1。通过向 USART_ICR 寄存器中的 CMCF 写入 1，该位由软件清零。

如果 USART_CR1 寄存器中 CMIE = 1，则会生成中断。

0：未检测到字符匹配

1：检测到字符匹配

位 16 **BUSY**：忙标志 (Busy flag)

该位由硬件置 1 和复位，当 RX 线路上正在进行通信（成功检测到起始位）时有效。在接收结束（成功或失败）时复位。

0：USART 处于空闲状态（无接收）

1：正在接收

位 15 **ABRF**：自动波特率标志 (Auto baud rate flag)

已设置自动波特率（RXFNE 也置 1，并在 RXFNEIE = 1 时生成中断），或者自动波特率操作未成功完成时，该位由硬件置 1 (ABRE = 1)（此时，ABRE、RXFNE 和 FE 也置 1）

为请求新的自动波特率检测，通过向 USART_RQR 寄存器中的 ABRRQ 写入 1，该位由软件清零。

注意：如果 USART 不支持自动波特率功能，该位保留且保持复位值。

位 14 **ABRE**：自动波特率错误 (Auto baud rate error)

如果波特率测量失败（波特率超出范围或字符比较失败），该位由硬件置 1。

通过将 1 写入 USART_RQR 寄存器中的 ABRRQ 位，该位由软件清零。

注意：如果 USART 不支持自动波特率功能，该位保留且保持复位值。

位 13 **UDR**：SPI 从设备下溢错误标志 (SPI slave underrun error flag)

在从发送模式下，如果在软件尚未将任何值加载到 USART_TDR 之前出现第一个数据发送时钟脉冲，此标志将置 1。通过将 USART_ICR 寄存器中的 UDRCF 位置 1 来复位此标志。

0：无下溢错误

1：存在下溢错误

注意：如果 USART 不支持 SPI 从设备模式，该位保留且保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 12 **EOBF**：块结束标志 (End of block flag)

接收到完整块后，该位由硬件置 1（例如 T = 1 智能卡模式）。接收到的字节数（从块的起始处，包括起始字段）等于或大于 BLEN + 4 时执行检测。

如果 USART_CR1 寄存器中 EOBIE = 1，则会生成中断。

通过向 USART_ICR 寄存器中的 EOBCF 写入 1，该位由软件清零。

0：未达到块结束

1：已达到块结束（字符数）

注意：如果不支持智能卡模式，该位保留且保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 11 **RTOF**：接收器超时

已经过在 RTOR 寄存器中编程的超时值后，如果无任何通信，该位由硬件置 1。通过向 USART_ICR 寄存器中的 RTOCF 写入 1，该位由软件清零。

如果 USART_CR2 寄存器中 RTOIE = 1，则会生成中断。

在智能卡模式下，该超时对应于 CWT 或 BWT 时间。

0：未达到超时值

1：已达到超时值，未接收到任何数据

注意：如果 RTOR 寄存器中编程的时间值将 2 个字符隔开，则 RTOF 不置 1。如果此时间大于该值 + 2 个采样时间（2/16 或 2/8，具体取决于过采样方法），则 RTOF 标志置 1。

即使 RE = 0，计数器仍会计数，但 RTOF 仅在 RE = 1 时置 1。如果 RE 置 1 时已经超时，则 RTOF 会置 1。

如果 USART 不支持接收器超时功能，该位保留且保持复位值。

位 10 **CTS**：CTS 标志 (CTS flag)

该位由硬件置 1/复位。该位是对 CTS 输入引脚的状态取反。

0：CTS 线置 1

1：CTS 线复位

注意：如果不支持硬件流控制功能，该位保留且保持复位值。

位 9 **CTSIF**：CTS 中断标志 (CTS interrupt flag)

如果 CTSE 位置 1，当 CTS 输入切换时，该位由硬件置 1。通过将 1 写入 USART_ICR 寄存器中的 CTSCF 位，该位由软件清零。

如果 USART_CR3 寄存器中 CTSIE = 1，则会生成中断。

0：CTS 状态线上未发生变化

1：CTS 状态线上发生变化

注意：如果不支持硬件流控制功能，该位保留且保持复位值。

位 8 **LBDF**：LIN 刹车检测标志 (LIN break detection flag)

检测到 LIN 刹车时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 LBDCF 写入 1，该位由软件清零。

如果 USART_CR2 寄存器中 LBDIE = 1，则会生成中断。

0：未检测到 LIN 刹车

1：检测到 LIN 刹车

注意：如果 USART 不支持 LIN 模式，该位保留且保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 7 **TXFNF**：TXFIFO 未满 (TXFIFO not full)

TXFNF 会在 TXFIFO 未满时由硬件置 1，表示可向 USART_TDR 中写入数据。每次对 USART_TDR 进行写操作都会将数据置于 TXFIFO 中。该标志会保持置 1，直到 TXFIFO 已满。当 TXFIFO 已满时，该标志清零，表示不能向 USART_TDR 中写入数据。

如果 USART_CR1 寄存器中 TXFNFIE 位 = 1，则会生成中断。

0：发送 FIFO 已满

1：发送 FIFO 未满

注意：在清空请求期间，TXFNF 保持复位，直到 TXFIFO 为空。发送清空请求（通过将 TXFRQ 位置 1）后，应先检查 TXFNF 标志，然后再写入 TXFIFO（TXFNF 和 TXFE 会同时置 1）。

单缓冲区发送期间使用该位。

位 6 **TC**：发送完成 (Transmission complete)

该位指示写入到 USART_TDR 中的最后一个数据已从移位寄存器中发出。

如果已完成对包含数据的帧的发送并且 TXFE 置 1，则该位由硬件置 1。

如果 USART_CR1 寄存器中 TCIE = 1，则会生成中断。

TC 位由软件清零，方法是向 USART_ICR 寄存器中的 TCCF 写入 1 或向 USART_TDR 寄存器执行写操作。

0：发送未完成

1：发送已完成

注意：如果 TE 位复位且无任何发送正在进行，TC 位会立即置 1。

位 5 **RXFNE**：RXFIFO 非空 (RXFIFO not empty)

RXFIFO 非空时，RXFNE 位由硬件置 1，这表示可以从 USART_RDR 寄存器读取数据。每次对 USART_RDR 进行读操作都会在 RXFIFO 中释放一个位置。

RXFNE 在 RXFIFO 为空时清零。也可以通过向 USART_RQR 寄存器中的 RXFRQ 位写入 1 将 RXFNE 标志位清零。

如果 USART_CR1 寄存器中 RXFNEIE = 1，则会生成中断。

0：未接收到数据

1：已准备好读取接收到的数据

位 4 **IDLE**：检测到空闲线路

检测到空闲线路时，该位由硬件置 1。如果 USART_CR1 寄存器中 IDLEIE = 1，则会生成中断。通过向 USART_ICR 寄存器中的 IDLECF 写入 1，该位由软件清零。

0：未检测到空闲线路

1：检测到空闲线路

注意：直到 RXFNE 位已置 1 时（即，当出现新的空闲线路时）IDLE 位才会被再次置 1。

使能静默模式 (MME = 1) 后，如果 USART 未静默 (RWU = 0)，则 IDLE 置 1，无论是否通过 WAKE 位选择了静默模式。如果 RWU = 1，IDLE 不置 1。

位 3 **ORE**：上溢错误 (Overrun error)

在 RXFF = 1 的情况下，当移位寄存器中当前正在接收的数据

准备好传输到 USART_RDR 寄存器时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 ORECF 写入 1，该位由软件清零。

如果 USART_CR1 寄存器中 RXFNEIE = 1 或 USART_CR3 寄存器中 EIE = 1，则会生成中断。

0：无上溢错误

1：检测到上溢错误

注意：当该位置 1 时，USART_RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。EIE 位置 1 后，如果在多缓冲区通信中 ORE 标志置 1，则会生成中断。

USART_CR3 寄存器中的 OVRDIS 位置 1 时，该位将被永久强制清零（无上溢检测）。

位 2 **NE**：噪声检测标志 (Noise detection flag)

当在接收的帧上检测到噪声时，该位由硬件置 1。该位由软件清零，方法是向 USART_ICR 寄存器中的 NECF 位写入 1。

0：未检测到噪声

1：检测到噪声

注意：该位不会生成中断，因为该位出现的时间与本身生成中断的 RXFNE 位出现的时间相同。EIE 位置 1 后，如果在多缓冲区通信中 NE 标志置 1，则会生成中断。

当线路无噪声时，可以通过将 ONEBIT 位编程为 1 提高 USART 对偏差的容差来禁止 NE 标志（请参见第 2453 页的第 49.5.8 节：USART 接收器对时钟偏差的容差）。

此错误与 USART_RDR 中的字符相关联。

位 1 **FE**：帧错误

当检测到去同步化、过度的噪声或刹车字符时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 FECF 写入 1，该位由软件清零。

在智能卡模式下发送数据时，如果在达到最大发送尝试次数后仍未成功（智能卡向数据帧发送 NACK 信号），则该位置 1。

如果 USART_CR3 寄存器中 EIE = 1，则会生成中断。

0：未检测到帧错误

1：检测到帧错误或刹车字符

注意：此错误与 USART_RDR 中的字符相关联。

位 0 **PE**：奇偶校验错误

当在接收器模式下发生奇偶校验错误时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 PECF 写入 1，该位由软件清零。

如果 USART_CR1 寄存器中的 PEIE = 1，则会生成中断。

0：无奇偶校验错误

1：奇偶校验错误

注意：此错误与 USART_RDR 中的字符相关联。

49.8.10 USART 中断和状态寄存器 [复用] (USART_ISR)

USART interrupt and status register

偏移地址：0x1C

复位值：0x0000 00C0

同一寄存器可用于使能 FIFO 模式（上一节）和禁止 FIFO 模式（本节）的情况。

禁止 FIFO 模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	TCBGT	Res.	Res.	RE ACK	TE ACK	WUF	RWU	SBKF	CMF	BUSY
						r			r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:26 保留，必须保持复位值。

位 25 **TCBGT**：保护时间前发送完成标志 (Transmission complete before guard time flag)

当写入到 USART_TDR 中的最后一个数据已正确从移位寄存器中发出时，该位置 1。

如果包含数据的帧完成发送，并且智能卡未发回任何 NACK，则该位在智能卡模式下由硬件置 1。如果 USART_CR3 寄存器中的 TCBGTIE = 1，则会生成中断。

该位由软件清零，方法是向 USART_ICR 寄存器中的 TCBGTCF 写入 1 或向 USART_TDR 寄存器执行写操作。

0：发送未完成或发送未成功完成（即，从智能卡接收到 NACK）

1：发送成功完成（在保护时间结束之前完成，智能卡未发送 NACK）。

注意：如果 USART 不支持智能卡模式，该位保留且保持复位值。如果 USART 支持智能卡模式并使能智能卡模式，则 TCBGT 复位值为“1”。请参见第 2436 页的第 49.4 节：USART 实现。

位 24:23 保留，必须保持复位值。

位 22 **REACK**：接收使能确认标志 (Receive enable acknowledge flag)

USART 采用接收使能值时，通过硬件将该位置 1/复位。

该位可用于验证 USART 是否准备好在进入低功耗模式前接收数据。

注意：如果 USART 不支持从 Stop 模式唤醒功能，该位保留且保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 21 **TEACK**：发送使能确认标志 (Transmit enable acknowledge flag)

USART 采用发送使能值时，通过硬件将该位置 1/复位。

通过写入 TE = 0 生成空闲帧请求，然后在 USART_CR1 寄存器中写入 TE = 1 以遵循 TE = 0 最短周期时，可使用该位。

位 20 **WUF**：从低功耗模式唤醒标志 (Wake-up from low-power mode flag)

当检测到唤醒事件时，该位由硬件置 1。事件通过 WUS 位字段定义。该位由软件清零，方法是向 USART_ICR 寄存器中的 WUCF 写入 1。

如果 USART_CR3 寄存器中 WUFIE = 1，则会生成中断。

注意：当 UESM 清零时，WUF 标志也清零。

如果 USART 不支持从 Stop 模式唤醒功能，该位保留且保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 19 **RWU**：*接收器从静默模式唤醒 (Receiver wake-up from Mute mode)*

该位指示 USART 是否处于静默模式。当识别出唤醒/静默序列时，该位由硬件清零/置 1。静默模式控制序列（地址或 IDLE）通过 USART_CR1 寄存器中的 WAKE 位进行选择。

当选择 IDLE 模式下唤醒时，该位只能通过用软件向 USART_RQR 寄存器中的 MMRQ 位写 1 的方式置 1。

0：接收器处于活动模式

1：接收器处于静默模式

注意：如果 USART 不支持从 Stop 模式唤醒功能，该位保留且保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 18 **SBKF**：发送刹车标志 (Send break flag)

该位指示已请求发送刹车字符。通过将 1 写入 USART_CR3 寄存器中的 SBKRQ 位，该位由软件置 1。该位在刹车发送的停止位期间由硬件自动复位。

0：刹车字符已发送

1：通过将 USART_RQR 寄存器中的 SBKRQ 位置 1 请求刹车字符

位 17 **CMF**：字符匹配标志 (Character match flag)

接收到由 ADD[7:0] 定义的字符后由硬件将该位置 1。通过向 USART_ICR 寄存器中的 CMCF 写入 1，该位由软件清零。

如果 USART_CR1 寄存器中 CMIE = 1，则会生成中断。

0：未检测到字符匹配

1：检测到字符匹配

位 16 **BUSY**：忙标志 (Busy flag)

该位由硬件置 1 和复位，当 RX 线路上正在进行通信（成功检测到起始位）时有效。在接收结束（成功或失败）时复位。

0：USART 处于空闲状态（无接收）

1：正在接收

位 15 **ABRF**：自动波特率标志 (Auto baud rate flag)

已设置自动波特率（RXNE 也置 1，在 RXNEIE = 1 时生成中断），或者自动波特率操作未成功完成时，该位由硬件置 1 (ABRE = 1)（此时，ABRE、RXNE 和 FE 也置 1）

为请求新的自动波特率检测，通过向 USART_RQR 寄存器中的 ABRRQ 写入 1，该位由软件清零。

注意：如果 USART 不支持自动波特率功能，该位保留且保持复位值。

位 14 **ABRE**：自动波特率错误 (Auto baud rate error)

如果波特率测量失败（波特率超出范围或字符比较失败），该位由硬件置 1。

通过将 1 写入 USART_RQR 寄存器中的 ABRRQ 位，该位由软件清零。

注意：如果 USART 不支持自动波特率功能，该位保留且保持复位值。

位 13 **UDR**：SPI 从设备下溢错误标志 (SPI slave underrun error flag)

在从发送模式下，如果在软件尚未将任何值加载到 USART_TDR 之前出现第一个数据发送时钟脉冲，此标志将置 1。通过将 USART_ICR 寄存器中的 UDRCF 位置 1 来复位此标志。

0：无下溢错误

1：存在下溢错误

注意：如果 USART 不支持 SPI 从设备模式，该位保留且保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 12 **EOBF**：块结束标志 (End of block flag)

接收到完整块后，该位由硬件置 1（例如 T = 1 智能卡模式）。接收到的字节数（从块的起始处，包括起始字段）等于或大于 BLEN + 4 时执行检测。

如果 USART_CR1 寄存器中 EOBIE = 1，则会生成中断。

通过向 USART_ICR 寄存器中的 EOBCF 写入 1，该位由软件清零。

0：未达到块结束

1：已达到块结束（字符数）

注意：如果不支持智能卡模式，该位保留且保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 11 **RTOF**：接收器超时

已经过在 RTOR 寄存器中编程的超时值后，如果无任何通信，该位由硬件置 1。通过向 USART_ICR 寄存器中的 RTOCF 写入 1，该位由软件清零。

如果 USART_CR2 寄存器中 RTOIE = 1，则会生成中断。

在智能卡模式下，该超时对应于 CWT 或 BWT 时间。

0：未达到超时值

1：已达到超时值，未接收到任何数据

注意：如果 RTOR 寄存器中编程的时间值将 2 个字符隔开，则 RTOF 不置 1。如果此时间大于该值 + 2 个采样时间（2/16 或 2/8，具体取决于过采样方法），则 RTOF 标志置 1。

即使 RE = 0，计数器仍会计数，但 RTOF 仅在 RE = 1 时置 1。如果 RE 置 1 时已经超时，则 RTOF 会置 1。

如果 USART 不支持接收器超时功能，该位保留且保持复位值。

位 10 **CTS**：CTS 标志 (CTS flag)

该位由硬件置 1/复位。该位是对 CTS 输入引脚的状态取反。

0：CTS 线置 1

1：CTS 线复位

注意：如果不支持硬件流控制功能，该位保留且保持复位值。

位 9 **CTSIF**：CTS 中断标志 (CTS interrupt flag)

如果 CTSE 位置 1，当 CTS 输入切换时，该位由硬件置 1。通过将 1 写入 USART_ICR 寄存器中的 CTSCF 位，该位由软件清零。

如果 USART_CR3 寄存器中 CTSIE = 1，则会生成中断。

0：CTS 状态线上未发生变化

1：CTS 状态线上发生变化

注意：如果不支持硬件流控制功能，该位保留且保持复位值。

位 8 **LBDF**：LIN 刹车检测标志 (LIN break detection flag)

检测到 LIN 刹车时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 LBDCF 写入 1，该位由软件清零。

如果 USART_CR2 寄存器中 LBDIE = 1，则会生成中断。

0：未检测到 LIN 刹车

1：检测到 LIN 刹车

注意：如果 USART 不支持 LIN 模式，该位保留且保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 7 **TXE**：发送数据寄存器为空

当 USART_TDR 寄存器的内容已传输到移位寄存器时，TXE 由硬件置 1。通过对 USART_TDR 寄存器执行写操作将该位清零。为丢弃数据（仅在智能卡 T = 0 模式下出现发送故障时），也可以通过向 USART_RQR 寄存器中的 TXFRQ 写入 1 来将 TXE 标志置 1。

如果 USART_CR1 寄存器中 TXEIE 位 = 1，则会生成中断。

0：数据寄存器已满

1：数据寄存器未满

位 6 **TC**：发送完成 (Transmission complete)

该位指示写入到 USART_TDR 中的最后一个数据已从移位寄存器中发出。

如果已完成对包含数据的帧的发送并且 TXE 置 1，则该位由硬件置 1。

如果 USART_CR1 寄存器中 TCIE = 1，则会生成中断。

TC 位由软件清零，方法是向 USART_ICR 寄存器中的 TCCF 写入 1 或向 USART_TDR 寄存器执行写操作。

0：发送未完成

1：发送已完成

注意：如果 TE 位复位且无任何发送正在进行，TC 位会立即置 1。

位 5 **RXNE**：读取数据寄存器不为空 (Read data register not empty)

当 USART_RDR 移位寄存器的内容已传输到 USART_RDR 寄存器时，RXNE 位由硬件置 1。通过对 USART_RDR 寄存器执行读取操作将该位清零。也可以通过将 USART_RQR 寄存器中的 RXFRQ 位置 1 将 RXNE 标志位清零。

如果 USART_CR1 寄存器中 RXNEIE = 1，则会生成中断。

0：未接收到数据

1：已准备好读取接收到的数据

位 4 **IDLE**：检测到空闲线路

检测到空闲线路时，该位由硬件置 1。如果 USART_CR1 寄存器中 IDLEIE = 1，则会生成中断。通过向 USART_ICR 寄存器中的 IDLECF 写入 1，该位由软件清零。

0：未检测到空闲线路

1：检测到空闲线路

注意：直到 RXNE 位已置 1 时（即，当出现新的空闲线路时）IDLE 位才会被再次置 1。

使能静默模式 (MME = 1) 后，如果 USART 未静默 (RWU = 0)，则 IDLE 置 1，无论是否通过 WAKE 位选择了静默模式。如果 RWU = 1，IDLE 不置 1。

位 3 **ORE**：上溢错误 (Overrun error)

在 RXFF = 1 的情况下，当移位寄存器中当前正在接收的数据准备好传输到 USART_RDR 寄存器时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 ORECF 写入 1，该位由软件清零。

如果 LPUART_CR1 寄存器中的 RXNEIE = 1 或 EIE = 1，或者 USART_CR3 寄存器中的 EIE = 1，则会生成中断。

0：无上溢错误

1：检测到上溢错误

注意：当该位置 1 时，USART_RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。EIE 位置 1 后，如果在多缓冲区通信中 ORE 标志置 1，则会生成中断。

USART_CR3 寄存器中的 OVRDIS 位置 1 时，该位将被永久强制清零（无上溢检测）。

位 2 **NE**：噪声检测标志 (Noise detection flag)

当在接收的帧上检测到噪声时，该位由硬件置 1。该位由软件清零，方法是向 USART_ICR 寄存器中的 NECF 位写入 1。

0：未检测到噪声

1：检测到噪声

注意：该位不会生成中断，因为该位出现的时间与本身生成中断的 RXNE 位出现的时间相同。EIE 位置 1 后，如果在多缓冲区通信中 NE 标志置 1，则会生成中断。

当线路无噪声时，可以通过将 ONEBIT 位编程为 1 提高 USART 对偏差的容差来禁止 NE 标志（请参见第 2453 页的第 49.5.8 节：USART 接收器对时钟偏差的容差）。

位 1 **FE**：帧错误

当检测到去同步化、过度的噪声或刹车字符时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 FECF 写入 1，该位由软件清零。

在智能卡模式下发送数据时，如果在达到最大发送尝试次数后仍未成功（智能卡向数据帧发送 NACK 信号），则该位置 1。

如果 USART_CR3 寄存器中 EIE = 1，则会生成中断。

0：未检测到帧错误

1：检测到帧错误或刹车字符

位 0 **PE**：奇偶校验错误

当在接收器模式下发生奇偶校验错误时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 PECF 写入 1，该位由软件清零。

如果 USART_CR1 寄存器中的 PEIE = 1，则会生成中断。

0：无奇偶校验错误

1：奇偶校验错误

49.8.11 USART 中断标志清零寄存器 (USART_ICR)

USART interrupt flag clear register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.
											w			w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	UDRCF	EOBCF	RTOCF	Res.	CTSCF	LBDCF	TCBGTCF	TCCF	TXFECF	IDLECF	ORECF	NECF	FECF	PECF
		w	w	w		w	w	w	w	w	w	w	w	w	w

位 31:21 保留，必须保持复位值。

位 20 **WUCF**：从低功耗模式唤醒清零标志 (Wake-up from low-power mode clear flag)
将 1 写入该位时，USART_ISR 寄存器中 WUF 标志将清零。
注意：如果 USART 不支持从 Stop 模式唤醒功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 19:18 保留，必须保持复位值。

位 17 **CMCF**：字符匹配清零标志 (Character match clear flag)
将 1 写入该位时，USART_ISR 寄存器中 CMF 标志将清零。

位 16:14 保留，必须保持复位值。

位 13 **UDRCF**：SPI 从设备下溢清零标志 (SPI slave underrun clear flag)
将 1 写入该位时，USART_ISR 寄存器中 UDRF 标志将清零。
注意：如果 USART 不支持 SPI 从设备模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现

位 12 **EOBCF**：块结束清零标志 (End of block clear flag)
将 1 写入该位时，USART_ISR 寄存器中 EOBF 标志将清零。
注意：如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 11 **RTOCF**：接收器超时清零标志 (Receiver timeout clear flag)
将 1 写入该位时，USART_ISR 寄存器中 RTOF 标志将清零。
注意：如果 USART 不支持接收器超时功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 10 保留，必须保持复位值。

位 9 **CTSCF**：CTS 清零标志 (CTS clear flag)
将 1 写入该位时，USART_ISR 寄存器中 CTSIF 标志将清零。
注意：如果不支持硬件流控制功能，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 8 **LBDCF**：LIN 刹车检测清零标志 (LIN break detection clear flag)
将 1 写入该位时，USART_ISR 寄存器中 LBDF 标志将清零。
注意：如果不支持 LIN 模式，该位保留且必须保持复位值。请参见第 2436 页的第 49.4 节：USART 实现。

位 7 **TCBGTCF**：保护时间前发送完成清零标志 (Transmission complete before Guard time clear flag)
将 1 写入该位时，USART_ISR 寄存器中 TCBGT 标志将清零。

位 6 **TCCF**：发送完成清零标志 (Transmission complete clear flag)
将 1 写入该位时，USART_ISR 寄存器中 TC 标志将清零。

位 5 **TXFECF**：TXFIFO 为空清零标志 (TXFIFO empty clear flag)
将 1 写入该位时，USART_ISR 寄存器中 TXFE 标志将清零。

位 4 **IDLECF**：检测到空闲线路清零标志 (Idle line detected clear flag)
将 1 写入该位时，USART_ISR 寄存器中 IDLE 标志将清零。

位 3 **ORECF**：上溢错误清零标志 (Overrun error clear flag)
将 1 写入该位时，USART_ISR 寄存器中 ORE 标志将清零。

位 2 **NECF**：检测到噪声清零标志 (Noise detected clear flag)
将 1 写入该位时，USART_ISR 寄存器中 NE 标志将清零。

位 1 **FECF**：帧错误清零标志 (Framing error clear flag)
将 1 写入该位时，USART_ISR 寄存器中 FE 标志将清零。

位 0 **PECF**：奇偶校验错误清零标志 (Parity error clear flag)
将 1 写入该位时，USART_ISR 寄存器中 PE 标志将清零。

49.8.12 USART 接收数据寄存器 (USART_RDR)

USART receive data register

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
							r	r	r	r	r	r	r	r	r

位 31:9 保留，必须保持复位值。

位 8:0 **RDR[8:0]**：接收数据值 (Receive data value)
包含接收到的数据字符。
RDR 寄存器在输入移位寄存器和内部总线之间提供了并行接口（请参见*图 564*）。
在使能奇偶校验位的情况下进行接收时，从 MSB 位中读取的值为接收到的奇偶校验位。

49.8.13 USART 发送数据寄存器 (USART_TDR)

USART transmit data register

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:9 保留，必须保持复位值。

位 8:0 **TDR[8:0]**：发送数据值 (Transmit data value)
包含要发送的数据字符。
USART_TDR 寄存器在内部总线和输出移位寄存器之间提供了并行接口（请参见*图 564*）。
在使能奇偶校验位的情况下（USART_CR1 寄存器中的 PCE 位被置 1）进行发送时，由于 MSB 的写入值（位 7 或位 8，具体取决于数据长度）会被奇偶校验位所取代，因此该值不起任何作用。
注意：只能在 TXE/TXFNF = 1 时写入此寄存器。

49.8.14 USART 预分频器寄存器 (USART_PRESC)

USART prescaler register

只有在禁止 USART (UE = 0) 时才能写入此寄存器。

偏移地址：0x2C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **PRESCALER[3:0]**：时钟预分频器 (Clock prescaler)

USART 输入时钟可通过预分频系数进行分频：

0000：输入时钟未分频
0001：输入时钟 2 分频
0010：输入时钟 4 分频
0011：输入时钟 6 分频
0100：输入时钟 8 分频
0101：输入时钟 10 分频
0110：输入时钟 12 分频
0111：输入时钟 16 分频
1000：输入时钟 32 分频
1001：输入时钟 64 分频
1010：输入时钟 128 分频
1011：输入时钟 256 分频
其余组合：保留

注意：为 PRESCALER 编程不允许的值时，编程的预分频值为 1011，即输入时钟除以 256。

49.8.15 USART 硬件配置寄存器 2 (USART_HWCFGR2)

USART Hardware Configuration register 2

偏移地址：0x3EC

复位值：0x0000 0014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFG2[3:0]				CFG1[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **CFG2[3:0]**：USART 硬件配置 2 (USART hardware configuration 2)
CFG2[3:0] = SPI_SLAVE

位 3:0 **CFG1[3:0]**：USART 硬件配置 1 (USART hardware configuration 1)
CFG1[3:0] = LOG_FIFO_DEPTH_RX
RXFIFO 大小 = $2^{LOG_FIFO_DEPTH_RX}$

49.8.16 USART 硬件配置寄存器 1 (USART_HWCFGR1)

USART Hardware Configuration register 1

偏移地址：0x3F0

复位值：0x4011 X111

X = 1 表示 USARTx，X = 0 表示 UARTx

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CFG8[3:0]				CFG7[3:0]				CFG6[3:0]				CFG5[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG4[3:0]				CFG3[3:0]				CFG2[3:0]				CFG1[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **CFG8[3:0]**：USART 硬件配置 8 (USART hardware configuration 8)
CFG8[3:0] = LOG_FIFO_DEPTH_TX
TXFIFO 大小 = $2^{LOG_FIFO_DEPTH_TX}$

位 27:24 **CFG7[3:0]**：USART 硬件配置 7 (USART hardware configuration 7)
CFG7[3:0] = BAUDGEN_TYPE

位 23:20 **CFG6[3:0]**：USART 硬件配置 6 (USART hardware configuration 6)
CFG6[3:0] = DUAL_clock

位 19:16 **CFG5[3:0]**：USART 硬件配置 5 (USART hardware configuration 5)
CFG5[3:0] = ABR_RTO_SCT1

位 15:12 **CFG4[3:0]**：USART 硬件配置 4 (USART hardware configuration 4)
CFG4[3:0] = SMARTCARD_SEL

位 11:8 **CFG3[3:0]**：USART 硬件配置 3 (USART hardware configuration 3)
CFG3[3:0] = IRDA_SEL

位 7:4 **CFG2[3:0]**：USART 硬件配置 2 (USART hardware configuration 2)
CFG2[3:0] = LIN_MASTER

位 3:0 **CFG1[3:0]**：USART 硬件配置 1 (USART hardware configuration 1)
CFG1[3:0] = SCLK_EXTRACT

49.8.17 USART 版本寄存器 (USART_VERR)

USART version register

偏移地址：0x3F4

复位值：0x0000 0024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本 (Major revision)
这些位返回 USART 主版本。

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)
这些位返回 USART 次版本。

49.8.18 USART 标识寄存器 (USART_IPIDR)

USART Identification register

偏移地址：0x3F8

复位值：0x0013 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：外设标识符 (Peripheral identifier)
这些位返回 USART 标识符 ID[31:0]：
ID[31:0] = 0x00130003

49.8.19 USART 大小标识寄存器 (USART_SIDR)

USART Size Identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识 (Size identification)

这些位返回分配给 USART 寄存器的存储区域的大小。

该存储区域的大小为 1 KB。

SID[31:0] = 0xA3C5 DD01

49.8.20 USART 寄存器映射

下表提供了 USART 寄存器映射和复位值。

表 445. USART 寄存器映射和复位值

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	USART_CR1 FIFO enabled	RXFFIE	TXFEIE	FIFOEN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]					OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXFNFIE	TCIE	RXFNEIE	IDLEIE	TE	RE	UESM	UE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00	USART_CR1 FIFO disabled	Res.	Res.	FIFOEN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]					OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	USART_CR2	ADD[7:0]								RTOEN	ABRMOD[1:0]		ABREN	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	LINEN	STOP [1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	DIS_NSS	Res.	Res.	SLVEN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0			0
0x08	USART_CR3	TXFTCFG[2:0]			RXFTIE	RXFTCFG[2:0]			TCBGTIE	TXFTIE	WUFIE	WUS [1:0]	SCAR CNT2:0]				Res.	DEP	DEM	DDRE	OVRDIS	ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	USART_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	USART_GTPR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	GT[7:0]								PSC[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	USART_RTOR	BLEN[7:0]								RTO[23:0]																							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	USART_RQR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ
	Reset value																												0	0	0	0	0
0x1C	USART_ISR FIFO mode enabled	Res.	Res.	Res.	Res.	TXFT	RXFT	TCBGT	RXFF	TXFE	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXFNF	TC	RXFNE	IDLE	ORE	NE	FE	PE
	Reset value					X	X	X	X	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
0x1C	USART_ISR FIFO mode disabled	Res.	Res.	Res.	Res.	Res.	Res.	TCBGT	Res.	Res.	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
	Reset value							0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
0x20	USART_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.	Res.	Res.	UDRCF	EOBCF	RTOCF	Res.	CTSCF	LBDCF	TCBGTCF	TCCF	TXFECF	IDLECF	ORECF	NECF	FECF	PECF
	Reset value												0			0				0	0	0		0	0	0	0	0	0	0	0	0	0
0x24	USART_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0

表 445. USART 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x28	USART_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x2C	USART_ PRESC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[3:0]			
	Reset value																													0	0	0	0
0x3EC	USART_ HWCFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFG2[3:0]				CFG1[3:0]			
	Reset value																									0	0	0	1	0	0	1	0
0x3F0	USART_ HWCFGR1	CFG8[3:0]				CFG7[3:0]				CFG6[3:0]				CFG5[3:0]				CFG4[3:0]				CFG3[3:0]				CFG2[3:0]				CFG1[3:0]			
	Reset value (USARTx)	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
	Reset value (UARTx)	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
0x3F4	USART_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV [3:0]				MINREV [3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
0x3F8	USART_IPIDR	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0x3FC	USART_SIDR	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 2.5 节：存储结构*。

50 串行外设接口 (SPI)

50.1 简介

串行外设接口 (SPI) 可使用特定同步协议与外部设备进行通信。(SPI) 接口支持与外部设备进行半双工、全双工和单工同步串行通信。该接口可配置为主模式或从模式，并且能够在多从或多主配置中运行。在配置为主模式时，它为外部从设备提供通信时钟 (SCK)。片选信号也可由主设备提供，从设备视情况选择接受。默认情况下使用 Motorola 数据格式，但也支持某些其他特定模式。

50.2 SPI 主要特性

- 基于三条线的全双工同步传输
- 基于双线的半双工同步传输，其中一条可作为双向数据线
- 基于双线的单工同步传输，其中一条可作为单向数据线
- 数据大小可选择 4 位到 32 位[a]
- 多主或多从模式功能
- 双时钟域，为外设内核提供独立于 APB 总线时钟的单独时钟
- 8 个主模式波特率预分频器，可达内核频率的一半
- 保护配置和设置
- 对于主模式和从模式都可以通过硬件或者软件进行 SS 管理
- 数据之间的最小延时，以及 SS 与数据流之间的最小延时均可调
- 可配置的 SS 信号极性和时序，MISO x MOSI 交换功能
- 可编程的时钟极性和相位
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 可以对传输的数据量进行编程，以控制 SS 和 CRC
- 可触发中断的专用发送和接收标志
- Stop 模式（未向外设提供时钟）下从设备发送和/或接收功能，以及唤醒功能
- SPI Motorola 和 TI 格式支持
- 硬件 CRC 功能可通过以下方式在事务的结尾验证通信的完整性：
  - 在发送模式下添加 CRC 校验值
  - 在接收模式下自动进行 CRC 错误校验
- 在数据溢出、CRC 错误、数据欠载、模式故障以及因操作模式而异的帧错误等情况下，具备错误检测及中断功能
- 两个嵌入式 8 位宽接收和发送 FIFO（FIFO 大小取决于实例）
- 可配置的 FIFO 阈值（数据封装）
- 可由系统 DMA 控制器处理数据流
- 从模式下，下溢条件可配置（支持级联循环缓冲区）

a. 24 位和 32 位数据宽度并非始终可用。请参见*第 50.3 节：SPI 实现*。

50.3 SPI 实现

表 446. STM32MP13xxSPI 特性

SPI 模式/特性	SPI2S1	SPI2S2	SPI2S3	SPI2S4	SPI5
接收和发送 FIFO 大小 [x 8 位]	16	16	16	16	8
最大可配置数据和 CRC 大小 [位]	32	32	32	32	16
I2S 功能	是	是	是	是	否

50.4 SPI 功能描述

50.4.1 SPI 框图

SPI 支持在 MCU 与外部设备之间进行同步串行通信。应用软件可通过轮询状态标志或使用专用 SPI 中断对通信进行管理。SPI 的主要组件及其交互方式如以下框图所示（*图 591*）。

图 591. SPI2S 框图

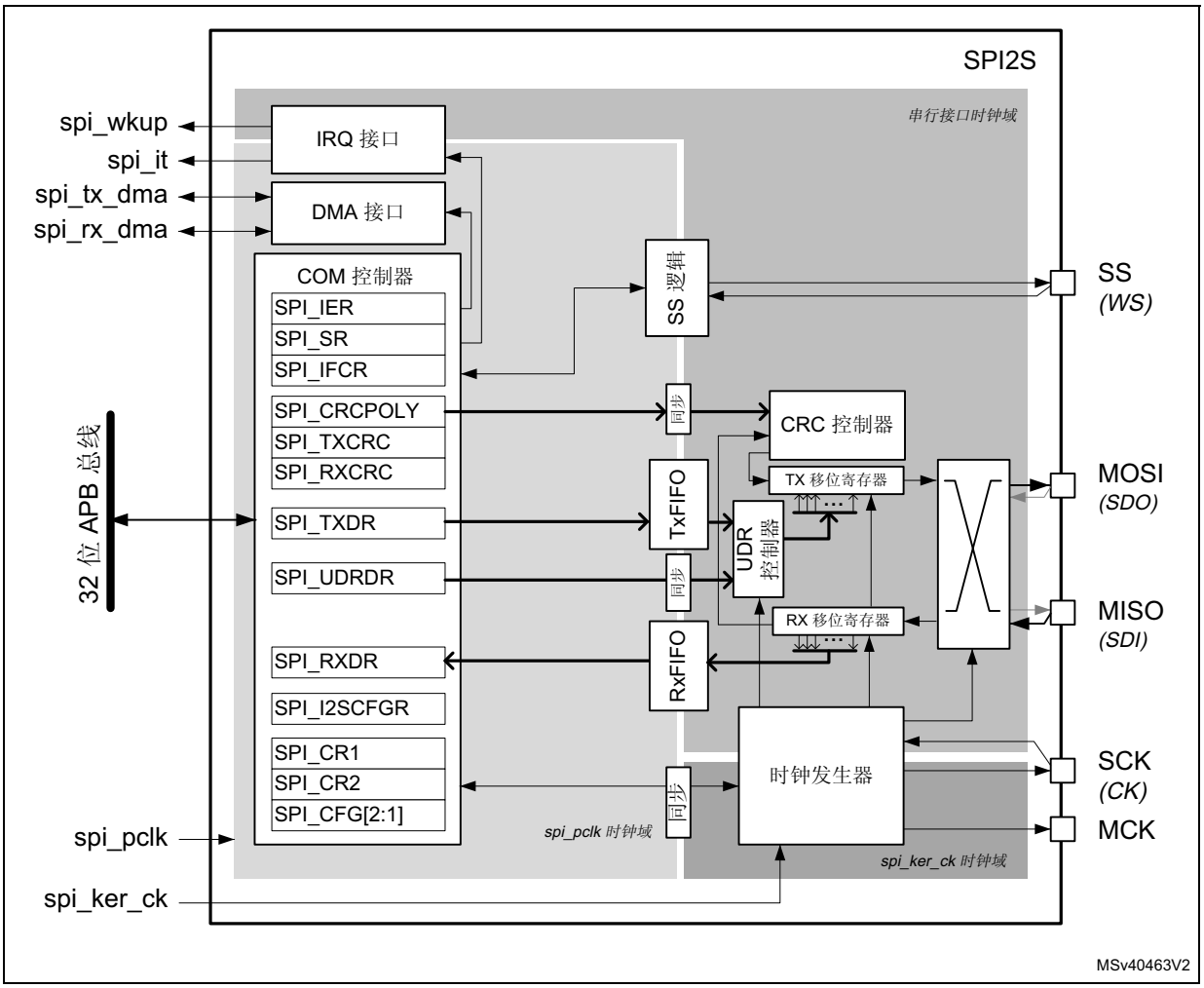


图 591 的简图显示了三个完全独立的时钟域：

- **spi_pclk** 时钟域
- **spi_ker_ck** 内核时钟域
- 串行接口时钟域

这些域之间的所有控制和状态信号均严格同步。这些时钟信号之间的频率比没有特定限制。用户必须考虑使用与数据流速度兼容的比率，以避免数据下溢或上溢。

spi_pclk 时钟信号给外设总线（APB 总线）供时钟。需要访问 SPI 寄存器时，该时钟必须有效。

SPI 主接口需要来自 RCC 的 **spi_ker_ck** 内核时钟在通信期间处于有效状态，以便通过时钟发生器分频器提供串行接口 SCK 时钟。

SPI 工作在从模式下进行数据传输时，由 SPI 主设备提供时钟 SCK 信号。在这种情况下，即使 **spi_pclk** 和 **spi_ker_ck** 时钟信号无效，SPI 从设备也能接收和发送数据。

因此，工作在从模式下的 SPI，如果需要设置寄存器的标志位，那么需要提供外设总线时钟（例如：出现上溢或下溢时，有关详细信息，请参见*第 50.5.2 节：SPI 错误标志*）。总线进入空闲状态时，不能进行上述过程。在特定情况下，从设备甚至需要时钟发生器工作（请参见*第 50.5.1 节：TI 模式*）。

50.4.2 SPI 信号

四个 I/O 引脚专用于与外部设备进行 SPI 通信。

- **MISO**：主输入 / 从输出数据。通常情况下，此引脚用于在从模式下发送数据和在主模式下接收数据。
- **MOSI**：主输出 / 从输入数据。通常情况下，此引脚用于在主模式下发送数据和在从模式下接收数据。
- **SCK**：SPI 主设备的串行时钟输出引脚以及 SPI 从设备的串行时钟输入引脚。
- **SS**：从设备选择引脚。根据 SPI 和 SS 设置，该引脚可用于：
  - 选择单个从设备以进行通信
  - 同步数据帧或
  - 检测多个主设备之间是否存在冲突

  有关详细信息，请参见*第 50.4.7 节：从设备选择 (SS) 引脚管理*。

SPI 总线支持一个主设备与一个或多个从设备之间进行通信。该总线至少由两条线构成：一条用于时钟信号，另一条用于同步数据传输。其他信号可以根据 SPI 节点间的数据交换及其从设备选择信号管理进行添加。MOSI 和 MISO 引脚之间的功能可以在任意 SPI 模式下反相（请参见 SPI_CFG2 寄存器的 IOSWP 位）。

如果在该实例中实现了 I2S 模式，则可在该模式下共用所有这些引脚。请参见*第 50.9.2 节：与 SPI 功能共享的引脚*。

50.4.3 SPI 通信一般情况

SPI 支持 MCU 基于目标设备和应用要求使用不同的配置进行通信。这些配置使用 2 条或 3 条线（通过软件 SS 管理），也可以使用 3 条或 4 条线（通过硬件 SS 管理）。通信始终由主设备发起和控制。主设备基于 SCK 线提供时钟信号，并在由硬件进行管理时通过 SS 线选择或同步从设备以实现通信。主从设备之间通过 MOSI 和/或 MISO 线实现数据流动。在以下拓扑图中，数据流方向用黑色箭头突出显示。

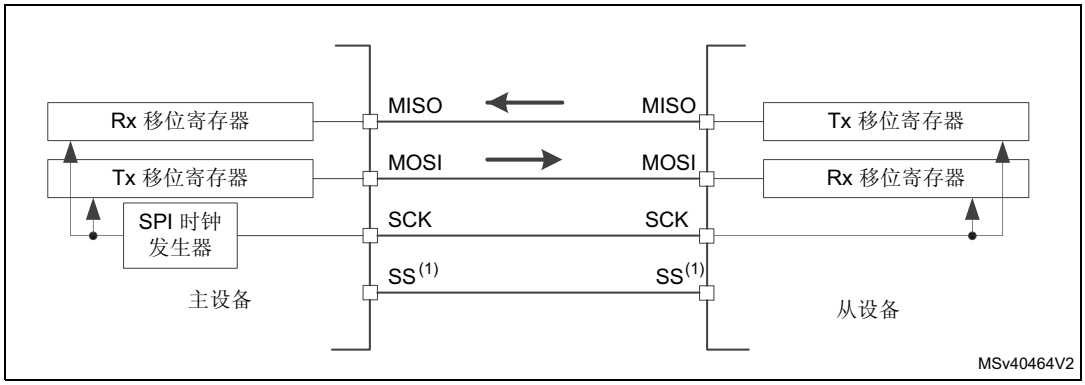
50.4.4 一个主设备和一个从设备之间的通信

通信流可使用以下 3 种模式之一：全双工（3 线）、半双工（2 线）或单工（2 线）。SS 信号在单一主从设备配置下是可选信号，并且一般不使用。不过，SS 信号在此配置下可有助于同步数据流，并在默认情况下用于某些特定 SPI 模式（例如，TI 模式）。

全双工通信

默认情况下，SPI 配置为全双工通信（SPI_CFG2 寄存器中的位 COMM[1:0] = 00）。在这种配置下，主设备和从设备的移位寄存器通过 MOSI 和 MISO 引脚之间的两条单向线连接。在 SPI 通信过程中，数据随主设备提供的 SCK 时钟边沿同步移位。主设备通过 MOSI 线将待发送的数据发送给从设备，同时通过 MISO 线从从设备接收数据。当数据帧传输完成时（所有位均移出），主设备和从设备之间即完成信息交换。

图 592. 全双工单个主设备/单个从设备应用

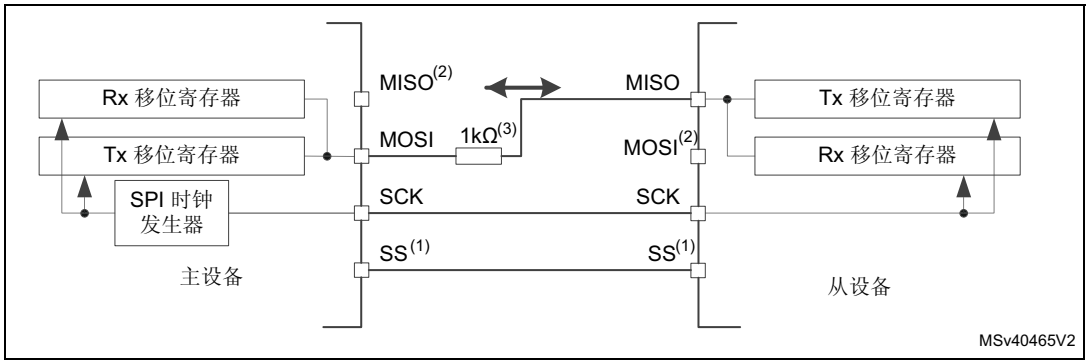


1. 要使 SPI 接口工作，并非一定要应用 SS 引脚互连（有关详细信息，请参见*第 50.4.7 节：从设备选择 (SS) 引脚管理*）。

半双工通信

通过设置 SPI_CFG2 寄存器中的 COMM[1:0] = 11，SPI 可采用半双工模式进行通信。在这种配置下，使用一条交叉连接线将主设备和从设备的移位寄存器连接起来。在此通信过程中，数据随 SCK 时钟边沿在移位寄存器之间进行移位，传输方向由主设备和从设备通过各自 SPI_CR1 寄存器中的 HDDIR 位进行选择。请注意，更改通信方向时，必须禁止 SPI。在该配置下，主设备的 MISO 引脚和从设备的 MOSI 引脚空闲，可在其他应用中用作 GPIO。

图 593. 半双工单个主设备/单个从设备应用



1. 要使 SPI 接口工作，并非一定要应用 SS 引脚互连（有关详细信息，请参见*第 50.4.7 节：从设备选择 (SS) 引脚管理*）。
2. 在该配置下，主设备的 MISO 引脚和从设备的 MOSI 引脚可用作 GPIO
3. 当以双向模式工作的两个节点间的通信方向不是同步变化时，会出现临界情况，新发送器访问共用数据线，而前一个发送器仍保持线路上的相反值（值取决于 SPI 配置和通信数据）。两个节点会出现冲突，在共用线上短暂提供相反的输出电平，直到下一个节点也相应地改变其方向设置。建议此模式下在 MISO 和 MOSI 引脚之间插入串行电阻以在这种情况下保护输出并限制电流在二者之间流过。

单工通信

通过 SPI_CFG2 寄存器中的 COMM[1:0] 位字段将 SPI 设置为只发送模式或只接收模式，可使 SPI 以单工模式进行通信。在这种配置下，仅使用一条线在主设备和从设备的移位寄存器之间进行传输。其余 MISO 或 MOSI 引脚对不用于通信，可用作标准 GPIO。

- 只发送模式：COMM[1:0] = 01

  只要 TxFIFO 中存在可用数据，并且主设备传输正在进行，只发送模式下的主设备就会生成时钟。

  如果仅发送模式下的从设备接收到 SCK 引脚上的时钟并且 SS 引脚（或软件管理的内部信号）有效，则该从设备会发送数据（请参见*第 50.4.7 节：从设备选择 (SS) 引脚管理*）。

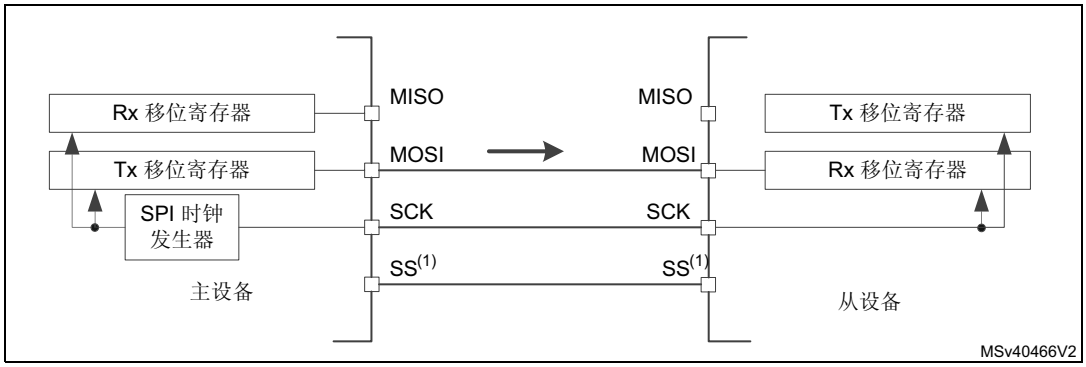
- **只接收模式：COMM[1:0] = 10**

  在主模式下，MOSI 输出被禁止，并可用作 GPIO。只要 SPI 使能并且 SPI_CR1 寄存器中的 CSTART 位置 1，就会持续生成时钟信号。时钟既可以由软件通过将 SPI_CR1 寄存器的 CSUSP 位置 1 进行明确请求来停止，也可以在 RxFIFO 已满时（SPI_CR1 的 MASRX 位置 1 的情况下）自动停止。

  在从设备配置下，MISO 输出被禁止，该引脚可用作 GPIO。当从设备选择信号有效时，从设备继续从 MOSI 引脚接收数据（请参见*第 50.4.7 节：从设备选择 (SS) 引脚管理*）。

注： *无论在何种主模式和从模式下，通过更改 SPI_CFG2 寄存器中的 IOSWP 位值，可将专用于发送的数据引脚替换为专用于接收的数据引脚，反之亦然。（该位只能在 SPI 禁止时进行修改）。*
任何单工通信均可以通过半双工通信的一种变型来替换，该变型中设置的数据传输方向不变（在 HDDIR 位保持不变的同时，双向模式处于使能状态）。

图 594. 单工单个主设备/单个从设备应用（主设备为仅发送模式/从设备为仅接收模式）

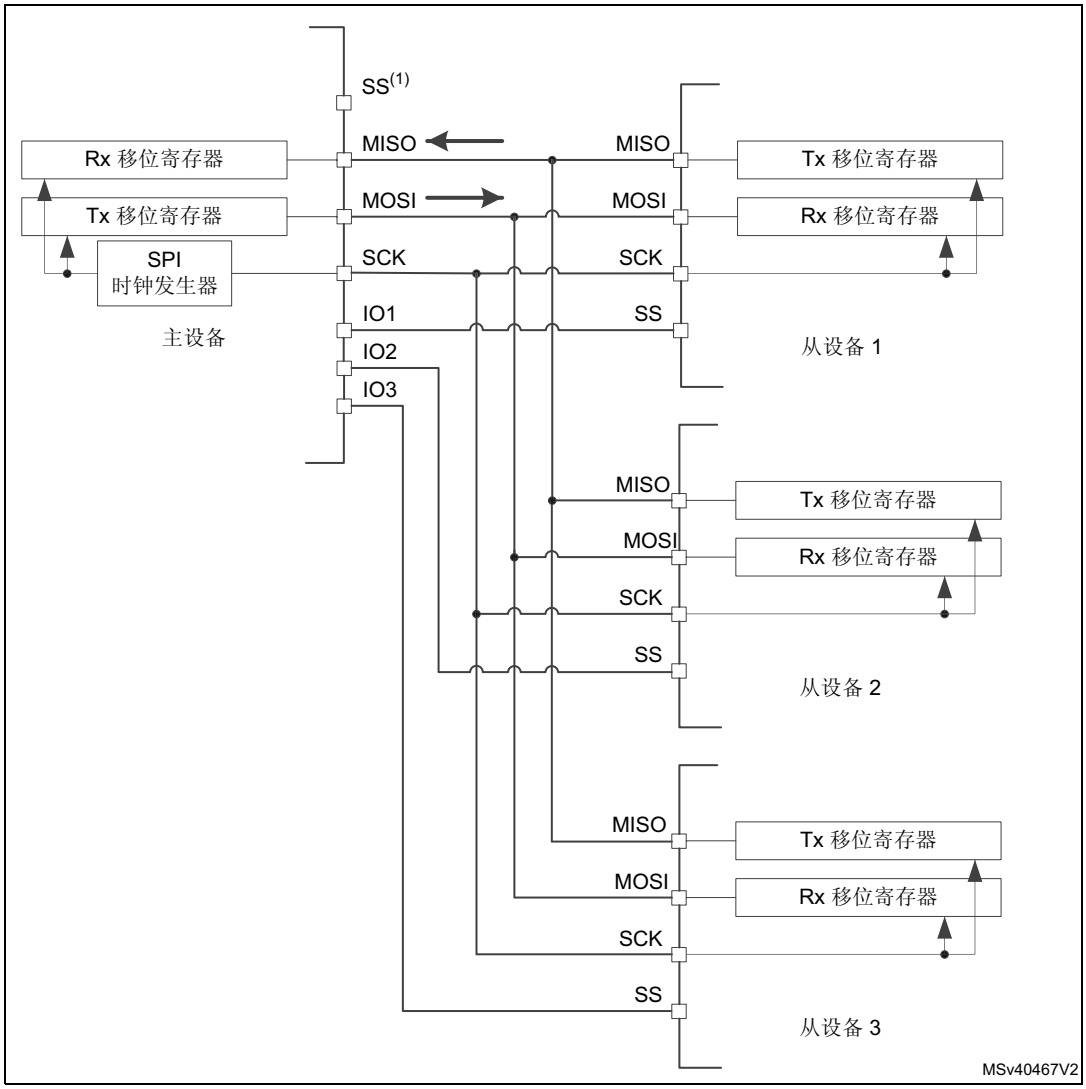


1. 要使 SPI 接口工作，并非一定要应用 SS 引脚互连（有关详细信息，请参见*第 50.4.7 节：从设备选择 (SS) 引脚管理*）。
2. 在这种配置下，两个 MISO 引脚均可用作 GPIO。

50.4.5 标准多从设备通信

在具有两个或多个独立从设备的配置下，主设备使用星型拓扑和专用的 GPIO 引脚来分别管理每个从设备的片选线（请参见*图 595.*）。主设备必须通过拉低与从设备 SS 输入相连的 GPIO 的电平来单独选择一个从设备（在同一时刻，只有一个从设备可以控制共用 MISO 线上的数据）。完成上述操作后，随即会在主设备和所选从设备之间建立通信。这种拓扑除了具有简便性优势外，还支持对各个从设备应用特定 SPI 配置，因为所有通信会话均在一个主设备-从设备对内单独进行。此外，如果无需从从设备中读取任何信息，主设备可以将相同的信息发送到多个从设备。

图 595. 使用星形拓扑的主设备和三个独立从设备

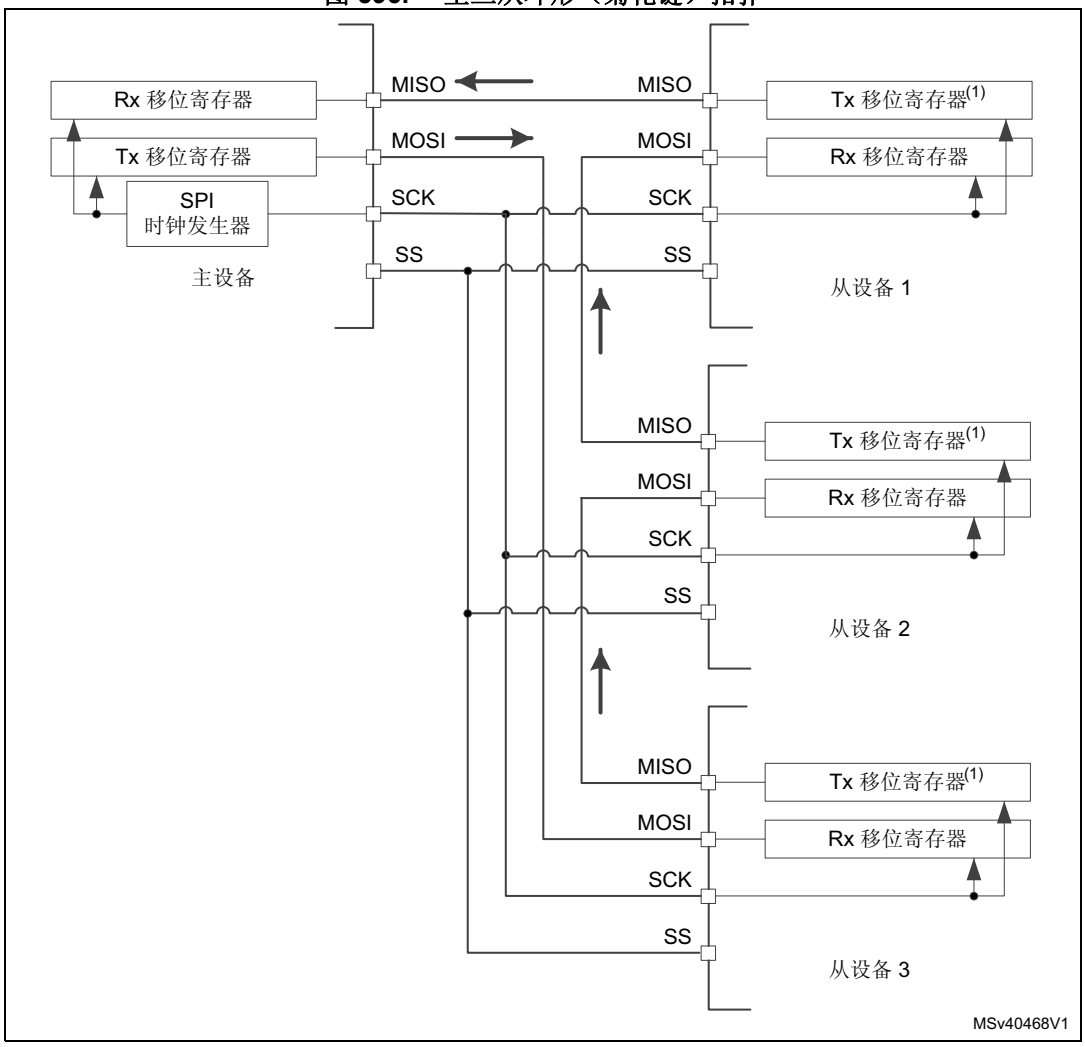


1. 主设备单个 SS 引脚硬件输出功能无法支持此拓扑（由软件控制下的一组 GPIO 代替），用户应避免在引脚处进行 SPI AF 设置（有关详细信息，请参见*第 50.4.7 节：从设备选择 (SS) 引脚管理*）。
2. 如果应用程序无法确保主设备在一定时间提供不超过一个 SS 有效信号，则最好将 MISO 引脚配置为开漏配置，并在 MISO 线上外接上拉电阻，以防止线上互连的从设备输出之间发生任何冲突。另外，可应用推挽配置而无需为从设备额外增加电阻。（请参见*第 913 页的第 11.3.7 节：I/O 复用功能输入/输出*。）

当采用环形拓扑时，主设备可以与所有从设备进行 SPI 通信（请参见*图 596*）。所有从设备类似于串行菊花链上的简单移位寄存器，通过共用的从设备选择信号和时钟进行控制。所有信息在环上同时移位，然后返回到主设备。会话的长度是固定的，主设备与从设备所处理的数据帧数相同。在菊花链中处理第一个数据帧时，主设备只通过第一个从节点输入向菊花链中最后一个从节点发送信息，主设备接收的第一条信息来自最后一个节点输出。相应地，用来结束会话的最后处理的数据发送给第一个从节点，同时其最先传出的数据在会话期间通过菊花链上所有其他从设备正好到达主设备输入。采用该拓扑时，菊花链中所有节点的数据格式配置和时钟设置都必须相同。由于接收和发送移位寄存器在内部是彼此分开的，因此当在接收器和发射器之间通过硬件交互信息时，必须使用 TxFIFO 从设备的下溢功能来完成。在这种情况下，发送下溢功能被配置为重复最后接收到的数据帧的模式 (UDRCFG[1:0] = 01)。在会话开始之前，每个从设备把单个数据写入 TxFIFO，然后发起会话传输（通常用到从设备状态信息）。在这种情况下，处理完该第一个数据帧后会实际发生下溢（数据处理结束时

必须在对从设备设置下溢检测：UDRDET[1:0] = 01）。为了能够立即清除内部下溢条件，并再次通过 TxFIFO 内容重新启动会话，用户必须先禁止再使能会话之间的 SPI，并通过新的单个数据模式填充 TxFIFO。

图 596. 一主三从环形（菊花链）拓扑



1. 在该配置下，在从设备上使用了下溢特性，一旦从设备的 TxFIFO 变空，从设备可以将先前接收的数据发送到 Rx 移位寄存器中。

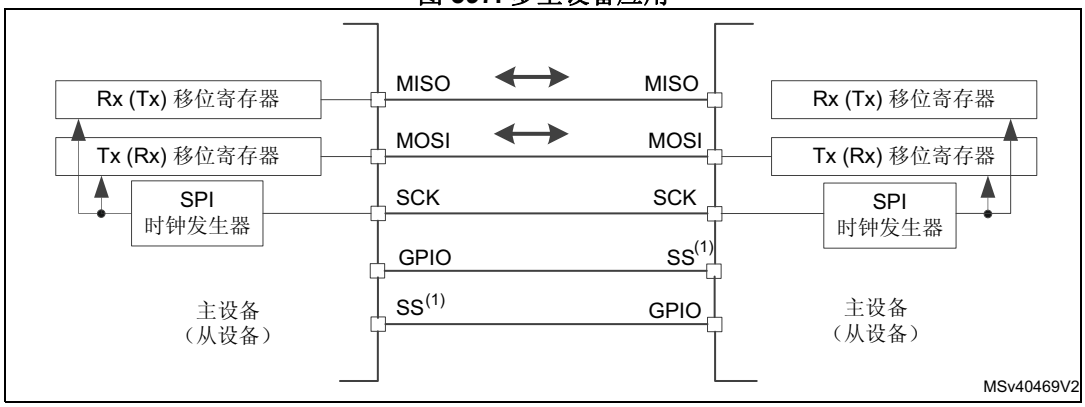
50.4.6 多主设备通信

如果 SPI 总线未用于多主功能，用户可使用内置功能来检测尝试同时控制总线的两个节点间是否存在潜在冲突。对于该检测，SS 引脚配置为硬件输入模式。由于此时只有一个结点可将其输出施加到公用数据线上，因此无法连接超过两个以此模式工作的 SPI 节点。

当节点无效时，默认情况下均保持从模式。一旦一个节点要接管对总线的控制，它会将自身切换到主模式，然后通过专用 GPIO 引脚向其它节点的从设备选择输入施加有效电平。会话完成后，有效的从设备选择信号将被释放，控制总线的节点会短暂切换回被动从模式，等待下一个会话开始。

如果两个节点同时发出各自的控制请求，则会出现总线冲突（请参见模式故障 MODF 事件）。随后，用户可应用某个简单的仲裁过程（例如，在两个节点上施加不同的预定义超时来推迟下一个尝试）。

图 597. 多主设备应用



1. 在两个节点上，SS 引脚都配置为硬件输入模式。当被动节点配置为从设备时，其有效电平将使能 MISO 线输出。

50.4.7 从设备选择 (SS) 引脚管理

在从模式下，SS 用作标准的“片选”输入，使从设备与主设备进行通信。在主模式下，SS 可用作输出或输入。用作输入时，可防止多主模式总线冲突；用作输出时，可驱动单个从设备的从设备选择信号。SS 信号可在内部进行管理（SS 输入信号的软件管理），或在 SS 输入和输出均与 SS 引脚相关联时在外部进行管理（硬件 SS 管理）。用户可以通过 SSIOP 位设置来配置（SS 引脚上）输入/输出外部信号的有效电平。如果 SS 电平等于 SSIOP，则将其视为处于活动状态。

可以使用 SPI_CFG2 寄存器中的 SSM 位设置硬件或软件从设备选择管理：

- **软件 SS 管理 (SSM = 1)**：在这种配置下，由 SPI_CR1 寄存器中的 SSI 位的值内部驱动从设备选择信息。外部 SS 引脚空闲，可供其他应用使用（用作 GPIO 或其他复用功能）。
- **硬件 SS 管理 (SSM = 0)**：在这种情况下，可行的配置有两种：所用配置取决于 SS 输出配置（SPI_CFG2 寄存器中的 SSOE 位）。
  - **SS 输出使能 (SSOE = 1)**：仅在将 MCU 设置为主设备时才使用该配置。SS 引脚由硬件管理。该功能与 CSTART 和 EOT 控制相关。因此，主设备必须应用适当的 TSIZE>0 设置来正确控制 SS 输出。即使 SPI AF 未应用于 SS 引脚（则可用作标准 GPIO），仍要保持 SSOE = 1 的设置，以确保默认 SS 输入电平，并防止在主 SS 内部逻辑（仅适用于多主拓扑）的输入处进行任何模式故障评估。

  a) 如果 SSOM = 0 且 SP = 000，则在主设备传输开始 (CSTART = 1) 时，SS 信号会被驱动至有效电平，并始终保持有效，直到其 EOT 标志置 1 或传输进入挂起状态。

  b) 当 SP = 001 时，根据 TI 模式定义生成脉冲。

  c) 当 SSOM = 1、SP = 000 且 MIDI>1 时，SS 在各个数据帧之间的脉冲无效，并在 MIDI 值减 1 得到的值所定义的 SPI 时钟周期（1 到 14）内保持无效。

  – **SS 输出禁止（SSM = 0，SSOE = 0）：**

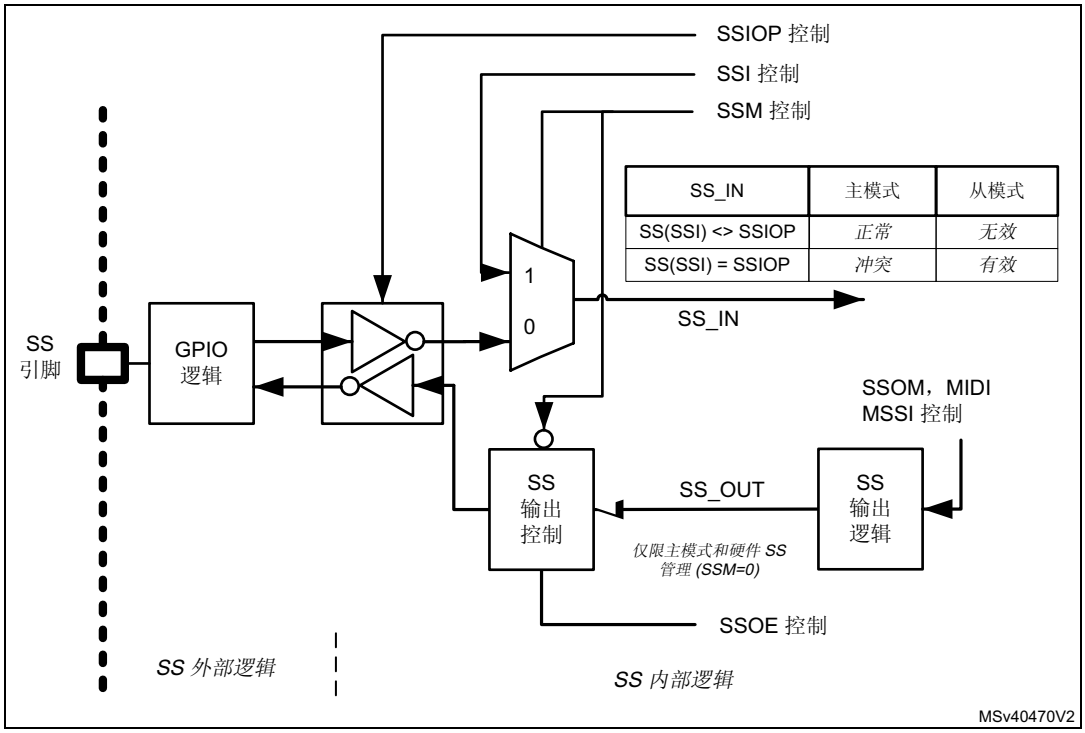
  a) 如果微控制器在总线上用作主设备，则此配置可实现多主模式功能。如果在该模式下将 SS 引脚拉至有效电平，SPI 将进入主模式故障状态，SPI 设备将在从模式 (MASTER = 0) 下自动进行重新配置。

  b) 在从模式下，SS 引脚用作标准的“片选”输入，当 SS 线为有效电平时将选择此从设备。

注：在模式故障条件下自动切换到从模式可避免数据和时钟线上发生潜在冲突。由于 SPE 在这种情况下自动复位，接收和发送 FIFO 均会刷新，并且当前数据会丢失。

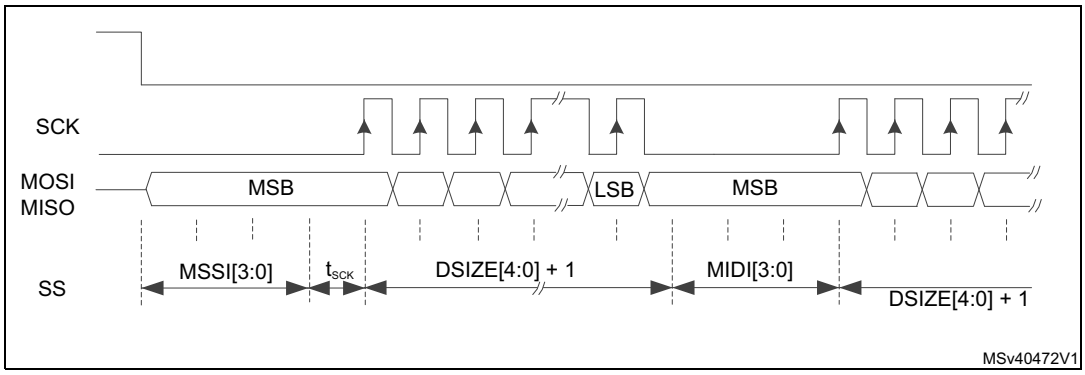
注：SPI 从设备在硬件 SS 管理模式下使能时，即使在发现 SS 处于有效电平的情况下，也会忽略所有通信，直到从设备检测到 SS 信号的开始（从无效电平变为有效电平的事务），以将从设备与主设备同步。因此，外部 SS 引脚固定时不能使用硬件管理模式。SS 软件管理不提供此类保护功能。总线上无通信时，必须更改 SSI 位，在传输之间 SCK 信号处于空闲状态电平（仅限这种情况）。

图 598. SS 控制逻辑原理图



当采用硬件输出片选 (SS) 控制 (SSM = 0，SSOE = 1) 时，通过配置 MIDI [3:0] 和 MSSI [3:0] 位字段，用户可以控制数据帧之间 SS 信号的时序，并在每次传输开始时插入额外延迟（以分隔 SS 和时钟的起始）。当从设备需要减缓数据传输速率，以便获得足够空间来正确处理数据时，这一功能会很有用（请参见*图 599：数据流时序控制（SSOE = 1，SSOM = 0，SSM = 0）*）

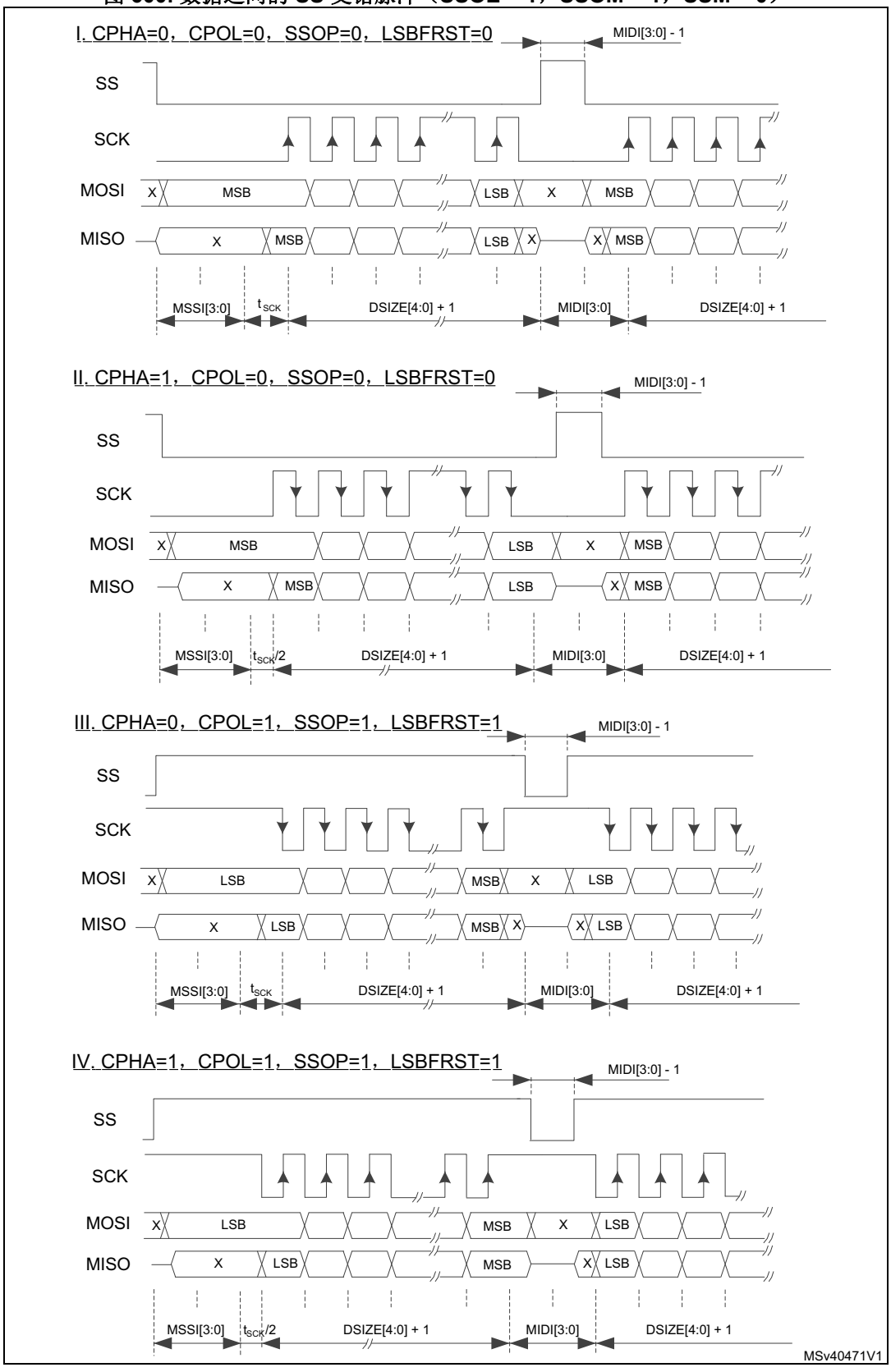
图 599. 数据流时序控制（SSOE = 1，SSOM = 0，SSM = 0）



1. MSSI[3:0] = 0011，MIDI[3:0] = 0011（MIDI[3;0] =0 时，SCK 流连续）。
2. CPHA = 0，CPOL = 0，SSOP = 0，LSBFRST = 0。

此外，位 SSOM =1 设置会调用特定模式，在该模式下，如果有足够的空间可用则会在数据帧之间加入交错脉冲（必须将 MIDI[3:0] 设置为大于一个 SPI 周期的值）。*图 600：数据之间的 SS 交错脉冲（SSOE = 1，SSOM = 1，SSM = 0）*给出了一些配置示例。

图 600. 数据之间的 SS 交错脉冲（SSOE = 1，SSOM = 1，SSM = 0）



1. MSSI[3:0] = 0010，MIDI[3:0] = 0010。
2. MIDI[3:0] > 1 时数据之间的 SS 交错。

50.4.8 通信格式

SPI 通信过程中，将同时执行接收和发送操作。串行时钟 (SCK) 对数据线上的信息的移位和采样进行同步。通信格式取决于时钟相位、时钟极性和数据帧格式。为了能够在彼此间进行通信，主设备和从设备必须遵循相同的通信格式并且必须正确同步。

时钟相位和极性控制

通过 SPI_CFG2 寄存器中的 CPOL 和 CPHA 位，可以用软件选择四种可能的时序关系。CPOL（时钟极性）位控制不传输任何数据时时钟的空闲状态值。该位对主设备和从设备都有作用。如果复位 CPOL，SCK 引脚在空闲状态处于低电平。如果将 CPOL 置 1，SCK 引脚在空闲状态处于高电平。

如果将 CPHA 位置 1，则会在 SCK 引脚的第二个边沿捕获传输的第一个数据位（如果复位 CPOL 位，则为下降沿；如果将 CPOL 位置 1，则为上升沿）。即，在每次出现该时钟边沿时锁存数据。如果将 CPHA 位复位，则会在 SCK 引脚的第一个边沿捕获传输的第一个数据位（如果将 CPOL 位置 1，则为下降沿；如果将 CPOL 位复位，则为上升沿）。即，在每次出现该时钟边沿时锁存数据。

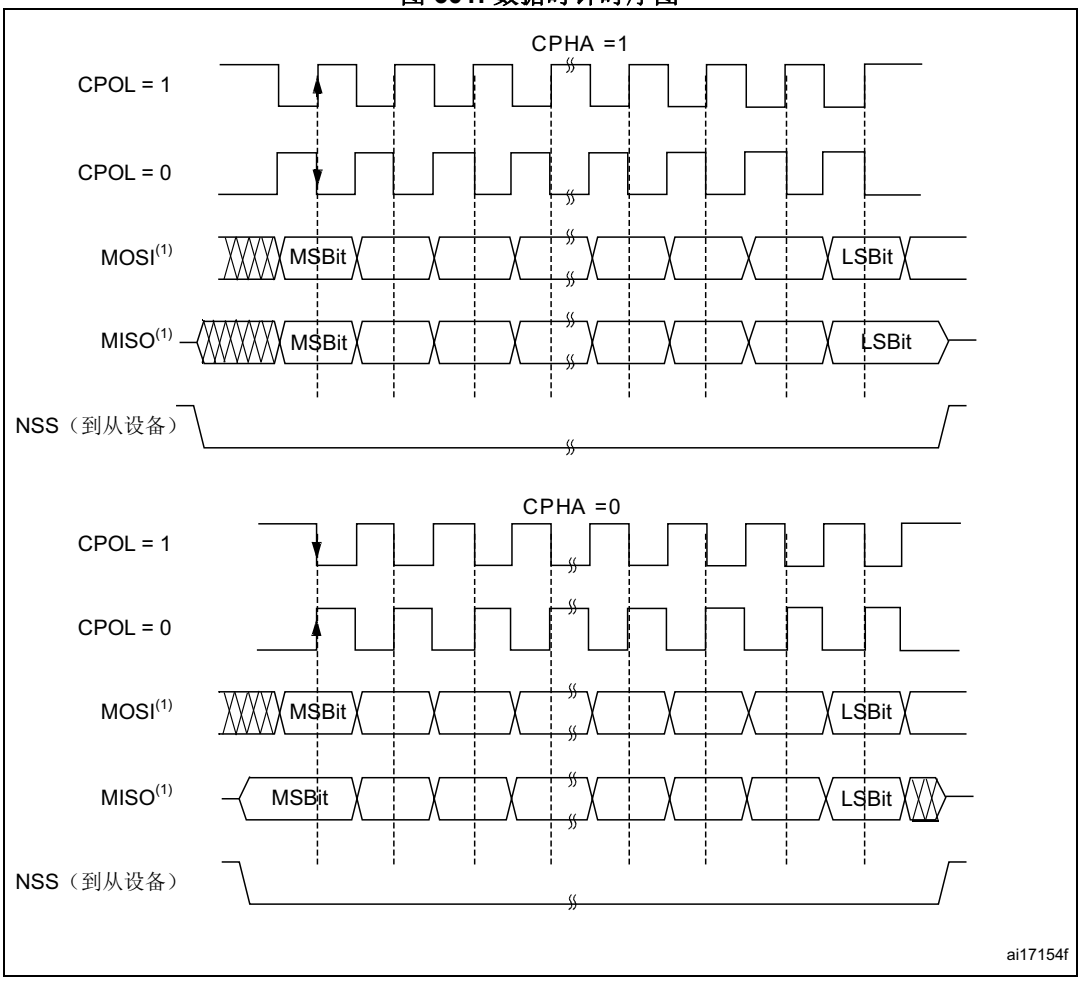
CPOL（时钟极性）和 CPHA（时钟相位）位的组合用于选择数据捕获时钟边沿（*图 601：数据时钟时序图*中的虚线）。

图 601 给出了在 CPHA 和 CPOL 位的四种组合下的 SPI 全双工传输。

注： *在切换 CPOL/CPHA 位之前，必须通过复位 SPE 位来禁止 SPI。*

SCK 的空闲状态必须与 SPI_CFG2 寄存器中选择的极性相对应（如果 CPOL = 1，则上拉 SCK 引脚；如果 CPOL = 0，则下拉 SCK 引脚）。

图 601. 数据时钟时序图



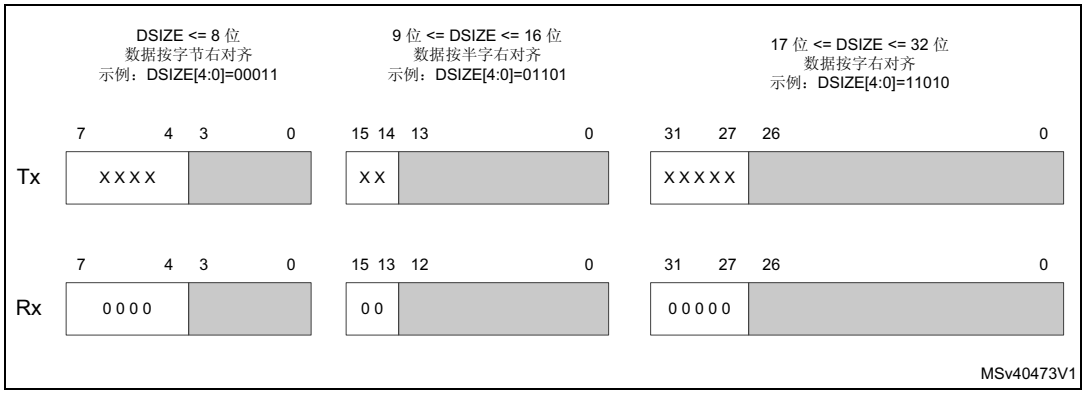
1. 数据位的顺序取决于 LSBFRST 位的设置。

数据帧格式

SPI 移位寄存器可设置为以 MSB 在前或 LSB 在前的方式移出数据，具体取决于 SPI_CFG2 寄存器的 LSBFRST 位的值。数据帧的长度通过 DSIZE[4:0] 位进行选择。数据帧的长度可设置为 4 位到 32 位，且此设置对于发送和接收均适用。访问 SPI_TXDR/SPI_RXDR 寄存器时，数据帧始终按字节（数据不超过一个字节时）、半字或字进行右对齐（请参见*图 602*）。

如果访问量是配置的数据大小的倍数，则会自动进行数据封装。在通信过程中，只会为数据帧内的位提供时钟并传输这些位。

图 602. 数据大小不等于 8 位、16 位或 32 位时的数据对齐



注： *数据长度至少为 4 位。如果所选的数据长度不足 4 位，则会将数据帧长度强制为 4 位。*

50.4.9 SPI 配置

主设备和从设备的配置步骤几乎相同。对于具体的模式设置，请遵从相应章节的内容。若要对标准通信进行初始化，在使能 SPI 之前，需要执行以下步骤：

1. 对相应的 GPIO 寄存器执行写操作：在 MOSI、MISO、SCK 和 SS 引脚（如果适用）上配置 GPIO 复用功能。
2. 对 SPI_CFG1 和 SPI_CFG2 寄存器进行写操作，以便为所有非保留位以及相应的位字段设置合适的值，但以下内容除外：
   a) 仅在主模式下才需要并考虑 SSOM、SSOE、MBR[2:0]、MIDI[3:0] 和 MSSI[3:0]。
   b) 仅在从模式下才需要并考虑 UDRDET[1:0] 和 UDRCFG[1:0]。仅当从设备配置为 TI 模式时才考虑 MBR[2:0] 设置。
   c) CRCEN 置 1 时需要配置 CRCSIZE[4:0]。
   d) TI 模式下不需要 CPOL、CPHA、LSBFRST、SSOM、SOE、SSIOP、MSSI、MIDI 和 SSM。
   e) SPI_CFG2 寄存器的 AFCNTR 位置 1 后，无论外设使能如何，所有 SPI 输出都会开始传播到相关的 GPIO 引脚，因此，之后对 SPI_CFG1 和 SPI_CFG2 寄存器的配置进行的任何更改都会影响这些引脚的信号电平。
   f) SPI_I2SCFGR 寄存器的 I2SMOD 位必须保持清零，以防偶然 I2S 配置的意外影响。
3. 对 SPI_CR2 寄存器进行写操作以选择传输的长度，如果值未知，则必须将 TSIZE 编程为零。
4. 对 SPI_CRCPOLY 进行写操作，并对 SPI_CR1 寄存器的 TCRCINI、RCRCINI 和 CRC33_17 位进行写操作，以根据需要配置 CRC 多项式和 CRC 计算。
5. 如果使用 DMA 数据流，请在 DMA 寄存器中配置 SPI Tx 和 Rx 专用的 DMA 数据流（请参见“使用 DMA 进行通信”一章）。
6. 如果需要，可配置 SPI_CR1 寄存器中的 SSI、HDDIR 和 MASRX。
7. 如果需要配置保护，请对 SPI_CFG1 寄存器的 IOLOCK 位进行编程（安全起见）。

50.4.10 使能 SPI 的步骤

建议在主设备发送时钟前配置和使能 SPI 从设备。但是，假设 SS 信号由从设备硬件管理，或者在采用 SS 信号软件管理时，从机软件将其保持为无效状态（参见*第 50.4.7 节：从设备选择 (SS) 引脚管理*），那么即使在总线上数据传输正在进行时完成配置和使能操作，也不会产生影响。从发送器的数据寄存器必须包含待发送的数据，这样主发送器才能开始提供时钟。在由 SS 选择 SPI 从设备之前，SCK 信号必须稳定到所选极性所对应的空闲状态电平，否则后续通信事务可能会不同步。

SPI 从设备在硬件 SS 管理模式下使能时，即使在发现 SS 处于有效电平的情况下，也会忽略所有通信，直到从设备检测到 SS 信号的开始（从无效电平变为有效电平的事务），以将从设备与主设备同步。因此，外部 SS 引脚固定时不能使用硬件管理模式。SS 软件管理不提供此类保护功能。在这种情况下，当总线上无通信时，SSI 位必须更改，在传输之间 SCK 信号处于空闲状态电平。

当 SPI 处于使能状态，CSTART 位置 1 且 TxFIFO 非空时，或者对 TxFIFO 执行下一次写操作时，全双工模式（或任何只发送模式）下的主设备开始通信。

在任何主设备仅接收模式下，在 SPI 使能并且 CSTART 位置 1 后，主设备开始通信，时钟开始运行。

有关处理 DMA 的信息，请参见*第 50.4.14 节：使用 DMA（直接内存寻址）进行通信*。

50.4.11 SPI 数据发送和接收过程

数据通信格式的设置遵循的基本原则是，必须在一个会话（事务）内传输一定数量的大小灵活的数据，同时，数据处理可以有效地累积到 SPI 数据寄存器的单次访问中（数据封装），如果以一致的大数据包收集数据，则甚至可以将其分组为一系列数据处理服务。数据处理服务基于 FIFO 数据包占用事件。因此，必须根据专用数据包标志来专门提供完整的数据包。

为了更好地理解本节后面的复杂内容，用户应首先了解以下各项的配置影响和含义：

数据大小 (DSIZE) – 定义数据帧（设置单个数据帧的位数）。

FIFO 阈值 (FTHLV) – 定义数据包，设置单个数据包的数据帧数量，以及数据包占用事件的发生情况，以通过软件或 DMA 处理 SPI 数据寄存器。

数据访问 – 发生数据包事件时，在应用程序与 SPI FIFO 之间传输数据时，处理 SPI 数据寄存器内容的一种方法。其取决于数据包大小配置。可以选择性地通过单次访问寄存器（通过数据封装）或通过此类访问序列（当提供更大的数据包时）来有效地进行多数据处理。

FIFO 大小 – 容纳可用数据的容量或空间。数据在此空间内的压缩和排列方式取决于数据大小和内部硬件效率。FTHLV 设置必须基于 FIFO 容量，以至少存储两个数据包。

事务大小 (TSIZE) – 定义总体上可能由多个数据包服务覆盖的一个事务会话中所涉及的数据帧总数。无需将此数值与数据包大小对齐（如果 TSIZE 编程正确，则支持处理最后一个未对齐的数据包）。

通过 RxFIFO 和 TxFIFO 进行数据处理

所有 SPI 数据转换均会通过按字节组织的嵌入式 FIFO（N x 8 位）。FIFO 的大小 (N) 与产品和外设实例相关。这使得 SPI 能够以连续流工作，并能防止在数据帧长度较短或中断/DMA 延时过长时发生上溢。每个方向都有其自身的 FIFO，分别被称为 TxFIFO 和 RxFIFO。

FIFO 内容的处理基于完全由专用 FIFO 数据包占用标志（TXP、RXP 或 DXP）引发的数据包提供事件。这些标志的出现与否取决于数据交换模式（双工、单工）、数据帧大小（帧中的位数）以及数据在数据包中的组织排列方式。当以定义 FIFO 阈值的方式将数据编排到数据包中时，数据包事件的频率会显著降低。这样，就可以通过单次 SPI 数据寄存器访问或访问序列，基于单个 FIFO 占用数据包事件，有效地处理按数据包分组的多个数据帧，从而消耗更少的系统性能。用户可以通过转换数据寄存器地址来控制访问类型，以强制应用具体的 CPU 指令来读取或写入寄存器。访问可以是 8 位、16 位或 32 位，但必须至少访问单数据帧。务必使数据包大小 (FTHLVL) 和数据大小 (DSIZE) 的设置与所应用的数据寄存器访问（无论是否应用单次访问或访问序列）始终保持平衡，以便在发生相关事件时应用和完成单个数据包服务。原则上，无论是否应用 DMA、中断或轮询，FIFO 占用标志的出现和清除能力都是通用的。

对 SPI_RXDR 寄存器执行读访问时，会返回尚未读取的 RxFIFO 中存储的最早的值。对 SPI_TXDR 执行写访问时，会在发送队列结束时将写入的数据存储到 TxFIFO 中。

SPI_RXDR 寄存器的读访问必须通过 RXP 事件进行管理。接收处于激活状态时，如果接收 FIFO 中至少有一个完整数据包（通过 SPI_CFG1 寄存器的 FTHLV[3:0] 位定义为接收器阈值）可用，则该标志由硬件置 1。通过软件或 DMA 读取 SPI_RXDR 时，只要 RxFIFO 中的可用数据少于完整的单个数据包，就会清零 RXP。

RXPIE 位置 1 时 RXP 会触发中断。

RXP 标志置 1 时，应用程序会执行适当数量的 SPI 数据寄存器读取操作，以下载一个数据包的内容。下载完整数据包后，应用软件或 DMA 会检查 RXP 值，以查看接收 FIFO 中是否存在待决的其他数据包，如果存在，则逐个数据包对其进行下载，直到 RXP 读为 0。RxFIFO 最多可存储 N 个数据帧（帧大小 =< 8 位时）、N/2 个数据帧（8 位 < 帧大小 =< 16 位时）、N/3 个数据帧（16 位 < 帧大小 =< 24 位时）或 N/4 个数据帧（数据帧大小 > 24 位时），其中 N 表示 FIFO 的大小（以字节计）。

接收结束时，可能会发生 RxFIFO 中仍然有某些数据可用，而不会达到 FTHLV 级别的情况，因此 RXP 不会置 1。在这种情况下，FIFO 中的剩余 RX 数据帧数将由 SPI_SR 寄存器中的 RXWNE 和 RXPLVL 位字段指示。传输大小和数据包大小未对齐时，如果在传输中接收到的最后数据数目不能完全达到配置的数据包大小，则会发生上述情况。不过，应用软件仍可从 RxFIFO 中进行用于先前完整数据包的标准次数的读取操作，而不会产生不利影响：只会从 RxFIFO 中弹出一致数据（完整的数据帧），而冗余读取次数（或任何不完整的数据）将为读数 0。通过这样的方式，应用软件可以使用相同的方式处理传输中的所有数据，并且可减轻负担来预知传输中的最后一个数据的接收，可减轻对要从 RxFIFO 中弹出的数据的适当读取次数进行计算的负担。

同样地，待发送数据帧的写访问通过 TXP 事件进行管理。发送处于激活状态时，如果应用程序有足够的空间来将至少一个完整的数据包（通过 SPI_CFG1 寄存器的 FTHLV[3:0] 位进行定义）推入到发送 FIFO 中，则该标志由硬件置 1。TxFIFO 由软件和/或 DMA 进行填充后，TXP 会被清零，当前可用于下一完整数据包的空间将丢失。逐帧存储新数据包时，如果从 TxFIFO 中释放数据，则这会导致 TXP 信号的振荡。TXTF 置 1 或禁止 SPI 时，如果没有足够的空间来存储至少一个数据帧（未遵循 TXP 事件），则对 TxFIFO 进行的任何写操作都将被忽略。

TXP 会在 TXPIE 位置 1 时触发中断，或在 TXDMAEN 置 1 时触发 DMA 请求。TXPIE 屏蔽在 TXTF 标志置 1 时由硬件清零。

TXP 标志置 1 时，应用软件会执行适当数量的 SPI 数据寄存器写入操作，以上传一整个数据包的内容。上传完新的完整数据包后，应用软件或 DMA 会检查 TXP 值，以查看是否可将其他数据包推入 TxFIFO 中，如果可以，则逐个数据包对其进行上传，直到在数据包加载结束时 TXP 读为 0。

传输大小和数据包大小未对齐时，传输中的最后一个数据量可短于配置的数据包大小。不过，应用仍可进行用于先前数据包的标准次数的数据寄存器写入操作，而不会产生不利影响：只会将一致数据推入 TxFIFO 中，而冗余写入次数将被忽略。通过这样的方式，应用软件可以使用相同的方式处理传输中的所有数据，并且可减轻负担来预知传输中的最后一个数据的发送，可减轻对要推入 TxFIFO 中的最后一个数据的适当写入次数进行计算的负担。仅在最后一个数据情况下，如果 TxFIFO 中有足够的空间来存储剩余数据进而完成当前传输，TXP 事件将由 SPI 置为有效。

TXP 和 RXP 事件均可通过中断进行轮询或处理。在全双工模式下，DXP 位可作为通用 TXP 和 RXP 事件进行监视。

将 DXP 标志置 1 时，应用软件会向 SPI 数据寄存器执行适当数量的写入操作，来上传一个用于发送的完整数据包的内容，然后从 SPI 数据寄存器进行相同数量的读取操作来下载一个数据包的内容。上传并下载一个数据包后，应用软件或 DMA 会检查 DXP 值，以查看是否可按顺序推入和弹出其他数据包，如果可以，则逐个数据包对其进行上传/下载，直到 DXP 读为 0。

DXPIE 位置 1 时 DXP 会触发中断。DXPIE 屏蔽在 TXTF 标志置 1 时由硬件清零。

DXP 对于全双工通信十分有用，可优化数据上传/下载性能，并减少支持 SPI 传输所需的中断数或 DMA 序列，从而最大限度地减少对 CPU 带宽和系统电源的需求，特别是在 SPI 工作在 Stop 模式时，尤为如此。

只中继 DXP 中断时，如果 TXP 和 RXP 事件由通用程序处理，则用户必须考虑这种简化的缺点，因为在这种情况下 TXP 服务会被故意延迟。这是因为 TXP 事件通常发生在接收 RXP 事件之前，以允许 TXP 处理最后帧的上一事务，将 TxFIFO 完全清空，否则主设备无法提供连续的 SCK 时钟流，从设备甚至会出现下溢。可能的解决方法是，在会话开始之前预先填充几个数据包，并在事务结束时通过 EOT 专门处理在 TXTF 事件后接收到的所有数据（因为 TXTF 会在事务结束时抑制 DXP 中断）。如果使能 CRC 计算，则在事务结束时，若仅依靠 EOT，用户必须计算出额外的空间，以容纳 RxFIFO 中的 CRC 帧。

另一种管理数据交换的方式是使用 DMA（请参见*使用 DMA（直接内存寻址）进行通信*）。

如果在 RXFIFO 已满时收到下一个数据，将发生上溢事件（请参见*第 50.5.2 节：SPI 错误标志*中的 OVR 标志说明）。上溢事件可通过中断来轮询或处理。

这可能会发生在从模式下或主模式下（全双工或仅接收模式，MASRX = 0）。在主设备仅接收模式下，MASRX = 1 时，如果 RxFIFO 已满，则生成的时钟将自动停止，因此可防止发生上溢。

禁止 SPI (SPE = 0) 时，RxFIFO 和 TxFIFO 内容保持清空状态。

事务处理

可通过一个通信事务传送一些数据帧从而完成一条消息。用户可通过存储在 TSIZE 和 TSER 字段中的值，处理一条消息中的多个数据。原则上，消息的传输在通过将 CSTART 位置 1 来使能 SPI 时开始，并在处理完所需数据总量后结束。如适用，事务的结束可控制 CRC 和硬件 SS 管理。强烈建议在下一个事务开始之前禁止然后重新使能 SPI（尽管设置未更改），以正确重启内部状态机。如果在 CSTART 置 1 时 TSIZE 保持为零，则会初始化无限事务（不控制传输大小）。在无限事务期间，仅支持与 FIFO 阈值相符的已处理数据量。如果数据量（或其分组数据包数量）不可预测，则用户必须将 FIFO 阈值设置（数据包大小）保持为单个数据 (FTHLV = 0)，以确保每个数据帧引发其自己的数据包事件并由应用程序或 DMA 进行处理。

凭借设置可清零 CSTART 位的 CSUSP 位，可随时挂起事务。在此类软件挂起后必须始终禁止 SPI，然后在下一个事务开始前重新使能。

在主模式下，用户可增加当前会话中的数据量。处理完 TSIZE 中的数据量时，如果 TSER 包含非零值，则会将 TSER 的内容复制到 TSIZE 中，并自动清零 TSER 值。通信事务随即会增加，增加的量为重新加载到 TSIZE 中的值所对应的数据量。在这种情况下，由于通信事务继续进行，不会出现 EOT 事件。完成重新加载操作后，如果 TSERFIE 置 1，则 TSERF 标志置 1 并会发生中断。用户可在 TSER 由硬件清零之后但尚未进行下次重新加载时，将下一个非零值写入 TSER 中，因此在重复此过程时可处理无数个数据。

应用任一数据扩展时，数据扩展都始终以对齐的数据包开始。因此，建议将要扩展的数据量始终与数据包大小保持对齐，否则，在应用扩展之前的最后一个数据包必须作为不完整数据包进行处理（请参见数据封装一章）。如果数据总数未对齐，则用户必须在最后一个扩展周期内将其余未对齐的数据量写到 TSER 中，然后在 EOT 事件处理程序中以标准方式处理数据的最后一个不完整数据包。

例如，如果用户想要在基于 8 位数据大小的配置应用数据量扩展时传输 23 个字节，则使用被设为 4 个数据的数据包和 32 位至 FIFO 的访问，这与下一序列是否正确无关

- TSIZE = 16 TSER = 7；
- TSIZE = 12 TSER = 8；最后一个扩展 TSER = 3；

因为在最后一次（第 6 次）访问 FIFO 时，会忽略最后一个未对齐的 MSB 字节。

如果对诸如以下要扩展的数据应用未对齐的序列：

- TSIZE = 15 TSER = 8 或
- TSIZE = 8 TSER = 7；最后一个扩展 TSER = 8；

则 MSB 字节在第 4 次访问 FIFO 时被忽略，而其他访问会始终处理 FIFO 中的 4 个数据。

使能发送后，序列即开始，只要主设备的 TxFIFO 中存在数据便一直继续。时钟信号由主设备永久性提供，直至 TxFIFO 变为空，之后时钟信号停止，等待其他数据。

在仅接收模式、半双工（COMM[1:0] = 11，HDDIR = 0）或单工 (COMM[1:0] = 10) 模式下，主设备在使能 SPI 时启动序列并通过将 CSTART 位置 1 来启动传输。时钟信号由主设备提供，且直至主设备禁止/挂起 SPI 或仅接收模式才会停止。在此之前，主设备会永久性接收数据帧。可通过软件控制以及通过向 SPI_CR1 寄存器的 CSUSP 位写入 1 将接收挂起，或者可在 MASRX = 1 且 RxFIFO 已满时自动将接收挂起。已完成在 SPI_CR2 寄存器的 TSIZE 和 TSER 字段中编程的帧数时，接收也会自动停止。

为禁止主设备仅接收模式，必须先挂起 SPI。SPI 被挂起时，会在更改配置前完成当前帧。

小心： 如果在主设备上，当接收正在进行且未做任何暂停操作时，将 SPI 使能位 (SPE) 写为 0，那么时钟将停止，当前帧不会完成，接收先入先出缓冲区 (RxFIFO) 也会被清空。

虽然主设备能够在连续模式下（SCK 信号持续输出）完成所有数据传输，但它必须随时考虑从设备处理数据流及其内容的能力。必要时，主设备必须降低通信速度，提供较慢的时钟或带有足够延时的单独帧或数据会话（通过将 MIDI[3:0] 位置 1），或者，提供初始延时（通过将 MSSI[1:0] 置 1），这会延迟传输的开始，从而为从设备提供充分的空间来准备数据。请注意，来自从设备的数据始终由主设备传输和处理，即使从设备无法及时正确地准备数据也是如此。从设备最好使用 DMA，尤其是数据帧较短，按字节访问 FIFO 且 SPI 总线速率较高时。

为了对来自从设备发送器节点的 SPI 通信流添加软件控制功能，可使用写入到 SPI_UDRDR（SPI 下溢数据寄存器）中的特定值。在从设备侧，当 TxFIFO 变为空时，该值将作为下一个数据自动送出，并且在主设备接收器侧可通过软件进行解析（丢弃或解析为类似 XOFF 的命令，以便通过软件挂起主设备接收器）。

在多从星型拓扑中，一次只能使能一个从设备用于输出数据。选定与主设备通信的从设备需要在与主设备开始通信之前检测其 SS 输入是否变为有效电平。在单个从设备系统中，无需通过 SS 来控制从设备，但此时提供此脉冲通常会更好，以在每个数据序列开始时同步从设备。SS 可通过软件和硬件进行管理（请参见*第 50.4.7 节：从设备选择 (SS) 引脚管理*）。

50.4.12 禁止 SPI 的步骤

当禁止 SPI 时，必须按照本段中介绍的禁止步骤进行操作。

在主模式下，在外设时钟停止，系统进入低功耗模式之前执行该操作十分重要。否则，在这种情况下，会损坏正在进行的传输。

在从模式下，SPI 通信可在 **spi_pclk** 和 **spi_ker_ck** 时钟停止时继续不间断进行，直至达到结束通信或数据服务请求条件。通常，可通过将系统置于 Stop 模式来停止 **spi_pclk**。更多相关信息，请参见 RCC 部分。

当处于全双工或仅发送模式下的主设备停止提供待发送的数据时，可结束任何事务。在这种情况下，时钟在最后一个数据传输后停止。可轮询 TXC 标志（或通过 EOTIE = 1 使能中断），以等待发送最后一个数据帧。

为停止外设，当主设备处于仅接收模式下时，必须首先通过将 CSUSP 置 1 挂起 SPI 通信。

当 SPI 被挂起时，已接收但未读取的数据始终存储在 RxFIFO 中。

禁止 SPI 时，RxFIFO 会被清空。为了防止丢失未读取的数据，用户必须通过读取所有剩余的数据（如 SPI_SR 寄存器的 RXP、RXWNE 和 RXPLVL 字段所示），来确保在禁止 SPI 时 RxFIFO 为空。

标准禁止步骤通过轮询 EOT 和/或 TXC 状态来检查发送会话是否（完全）结束。还可以在必须识别正在处理的传输是否结束的特定情况下完成这种检查，例如：

- 当主设备通过与 SPI 无关的 GPIO 处理 SS 信号（例如在多从机星形拓扑的情况下），且必须为从设备提供正确的 SS 脉冲结束时，或者
- 最后一个数据帧或 CRC 帧传输仍在外设总线中处理时，来自 DMA 或 FIFO 的传输数据流完成。

当 TSIZE>0 时，EOT 和 TXC 信号相等，因此无论采用何种 SPI 通信模式，EOT 轮询均可靠，以检查总线活动的结束。当 TSIZE = 0 时，用户必须根据所应用的 SPI 模式和数据流终止方式，来检查 TXC、SUSP 或 FIFO 占用标志。

主模式下的正确禁止步骤如下（使用仅接收模式时除外）：

1. 等待 TXC = 1 和/或 EOT = 1（不再有待发送的数据和发送最后一个数据帧）。使用 CRC 时，会在块中的最后一个数据被处理后自动发送。在这种情况下，TXC/EOT 会在 CRC 帧完成时置 1。当发送挂起时，软件必须等到 CSTART 位清零。
2. 读取所有 RxFIFO 数据（直到 RXWNE = 0 和 RXPLVL = 00）
3. 禁止 SPI (SPE = 0)。

主设备仅接收模式的正确禁止步骤如下：

1. 等待 EOT 或通过挂起 SPI (CSUSP = 1) 中断接收流
2. 如果接收流已挂起，等待至 SUSP = 1（最后一个数据帧已处理完）。
3. 读取所有 RxFIFO 数据（直到 RXWNE = 0 和 RXPLVL = 00）
4. 禁止 SPI (SPE = 0)。

在从模式下，禁止 SPI 时，任何正在进行的数据都将丢失。

50.4.13 数据封装

从用户的角度来看，数据封装方式有两种，这两种方式可彼此覆盖：

- 向 TxFIFO 中写入数据或从 RxFIFO 中读取数据时的访问类型。
  如果数据大小成倍地小于对 SPI_TXDR 或 SPI_RXDR 寄存器执行的访问大小，则可通过单次访问有效地推入或提取多个数据。
- 在单一软件服务期间要处理的数据量
  将数据分组，并专门针对整个数据包的内容累计进行 FIFO（先进先出队列）服务，而不是逐帧单独处理数据，这样更为便捷。之后，所有 FIFO 占用事件均与该阈值相关，而所需服务通过具有中断和/或唤醒功能的适当标志来反映。

若数据帧大小不足一个字节（小于或等于 8 位），则在 SPI_RXDR/SPI_TXDR 寄存器上执行任何 16 位或 32 位读写访问时都将自动使用数据封装。在这种情况下将并行处理多数据帧类型。最初，SPI 以所访问字的 LSB 中存储的模式工作，然后以 MSB 中存储的另一种模式来工作。*图 603* 给出了数据封装模式序列处理的示例。此时 DSIZE[3:0] 被配置为 4 位，在对发送器的 SPI_TXDR 寄存器进行单次 16 位或 32 位访问后，会向 TxFIFO 中写入两个或四个数据帧。

当数据帧大小在 9 位到 16 位之间时，若进行 32 位访问，会自动采用数据打包方式。将首先使用最低有效半字。（与 LSBFRST 值无关）

如果将 RxFIFO 阈值设为 1 个帧（且以帧为单位读取数据，即解封形式），则该序列可在接收器中生成两个或四个 RXP 事件，或者，如果 SPI_CFG1 寄存器中的 FTHLV[3:0] 字段被配置为以封装模式读取的多个帧（16 位或 32 位读访问），则可生成一个 RXP 事件。

数据根据*图 602：数据大小不等于 8 位、16 位或 32 位时的数据对齐*进行对齐。有效位仅在总线上执行。未使用的位在发送器侧无关紧要，而在接收器侧则用零填充。

将短数据帧（<8 位或 < 16 位）与大型数据访问模式（16 位或 32 位）搭配使用时，FTHLV 值必须被配置为帧/数据访问数的倍数（即，如果将 32 位访问用于最高 8 位的帧则为 4 的倍数，或者，如果将 16 位访问用于最高为 8 位的帧或者将 32 位访问用于最高 16 位的帧，则为 2 的倍数）。

RxFIFO 阈值设置必须始终高于后续读访问大小，否则将读取额外的伪数据。

不允许小于配置的数据大小的 FIFO 数据访问。始终须确保至少访问一个完整的数据帧。

如果在 FIFO 中存在不完整的数据包（即，少于 4x8 位帧或一个单独的 16 位帧可用），则会出现特定问题。

解决该问题的方法有两种：

A. 不使用 TSIZE 位字段

在发送器侧，只需将任意奇序列的最后一个数据帧以 8 位/16 位访问的方式写入 SPI_TXDR 即可。

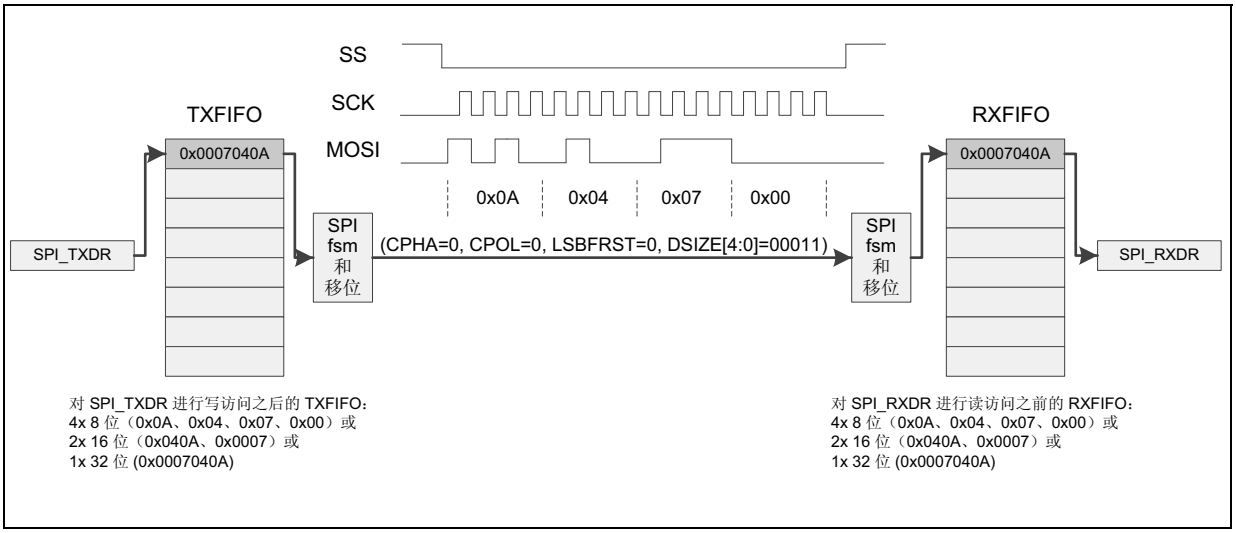
在接收器侧，可通过任意访问读取剩余数据。任何额外的数据读取都会用零填充。轮询 RXWNE 和 RXPLVL 可用于检测 RX 数据何时在 RxFIFO 中可用。（可以在系统级利用超时来检测轮询）

B. 使用 TSIZE 位字段

在发送器侧，事务在遇到 EOT 事件时由主设备停止。

在接收过程中，RXP 标志不会在 EOT 置 1 时置 1。要接收的数据数 (TSIZE) 不是数据包大小的倍数时，剩余数据的数目通过 SPI_SR 寄存器的 RXWNE 和 RXPLVL 位字段指示。可通过任意访问读取剩余数据。任何额外的读取都会用零填充。

图 603. 将数据封装到 FIFO 中以进行发送和接收



1. DSIZE[3:0] 配置为 4 位，数据右对齐，有效位仅在总线上执行，其顺序取决于 LSBFRST，如果置 1，所有数据帧的顺序都将反向。

50.4.14 使用 DMA（直接内存寻址）进行通信

为了以最大速度工作并且为了促进避免上溢所需的数据寄存器读/写过程，SPI 提供了 DMA 功能，该功能采用了简单的请求/应答协议。

将 SPI_CFG1 寄存器的使能位 TXDMAEN 或 RXDMAEN 置 1 时，将请求 DMA 访问。必须向发送缓冲区和接收缓冲区发出单独的请求，以完成所定义数据包的服务。

- 在发送过程中，每次 TXP 置 1 都会触发一系列 DMA 请求。然后，DMA 将对 SPI_TXDR 寄存器执行一系列写操作。
- 在接收过程中，每次 RXP 置 1 都会触发一系列 DMA 请求。然后，DMA 将对 SPI_RXDR 寄存器执行一系列读操作。如果 EOT 在事务结束时置 1，且最后一个数据包不完整，则会根据 RXWNE 和 RXPLVL[1:0] 设置自动激活 DMA 请求来读取剩余数据。

如果将 SPI 编程为仅接收模式，则 UDR 将永不置 1。

将 SPI 编程为发送模式时，如果发送数据未准备好，则 TXP 和 UDR 最终可在从设备侧置 1。在这种情况下，将根据 UDR 管理选择在 TX 线上发送一些数据。

在单工模式下使用 SPI 时，用户必须仅使能足够的 DMA 通道，同时禁止未使用的互补通道。

如果将 SPI 编程为仅发送模式，则 RXP 和 OVR 将永不置 1。

如果将 SPI 编程为全双工模式，若接收数据未读取，则 RXP 和 OVR 最终会置 1。

在发送模式下，DMA 或用户写入所有要发送的数据（SPI_SR 寄存器中的 TXTF 标志置 1）后，可以对 EOT（或 TXC（TISEZE=0 时））标志进行监视，以确保 SPI 通信已完成。在关闭 SPI 之前或在主模式下关掉 **spi_pclk** 之前必须执行此步骤，以避免损坏最后一次发送。软件必须首先等待至 EOT = 1 和/或 TXC = 1。

通过 DMA 开始通信时，为防止 DMA 通道管理引发错误事件，必须按顺序执行以下步骤：

1. 如果使用 DMA Rx，通过 SPI_CFG1 寄存器中的 RXDMAEN 位来使能 DMA 接收缓冲区。
2. 如果使用 DMA，则通过 DMA 寄存器来使能 Tx 和 Rx 的 DMA 请求。
3. 如果使用 DMA Tx，通过 SPI_CFG1 寄存器中的 TXDMAEN 位来使能 DMA 发送缓冲区。
4. 通过将 SPE 位置 1 使能 SPI。

要关闭通信，必须按顺序执行以下步骤：

1. 如果使用 DMA，则通过 DMA 寄存器来禁止 Tx 和 Rx 的 DMA 请求。
2. 通过后续 SPI 禁止步骤来禁止 SPI。
3. 如果使用 DMA Tx 和/或 DMA Rx，通过将 SPI_CFG1 寄存器中的 TXDMAEN 和 RXDMAEN 位清零来禁止 DMA 发送缓冲区和接收缓冲区。

使用 DMA 时的数据封装

如果由 DMA 管理传输（SPI_CFG1 寄存器中的 TXDMAEN 和 RXDMAEN 位置 1），则将根据为 SPI TX 和 SPI RX 的 DMA 通道配置的 PSIZE 值自动使能/禁止封装模式。

如果 DMA 通道的 PSIZE 值等于 16 位或 32 位，且 SPI 数据大小小于或等于 8 位，则将使能封装模式。类似地，如果 DMA 通道的 PSIZE 值等于 32 位，且 SPI 数据大小小于或等于 16 位，则将使能封装模式。然后，DMA 将自动管理对 SPI_TXDR 寄存器的写操作。

无论使用何种数据封装模式，也无论要传输的数据量是否为 DMA 数据大小（16 位或 32 位）的倍数，如果帧大小较小，则 DMA 会根据 TSIZE 位字段设置自动完成传输。

此外，最后几个数据帧还可由软件以单一/解封模式写入。

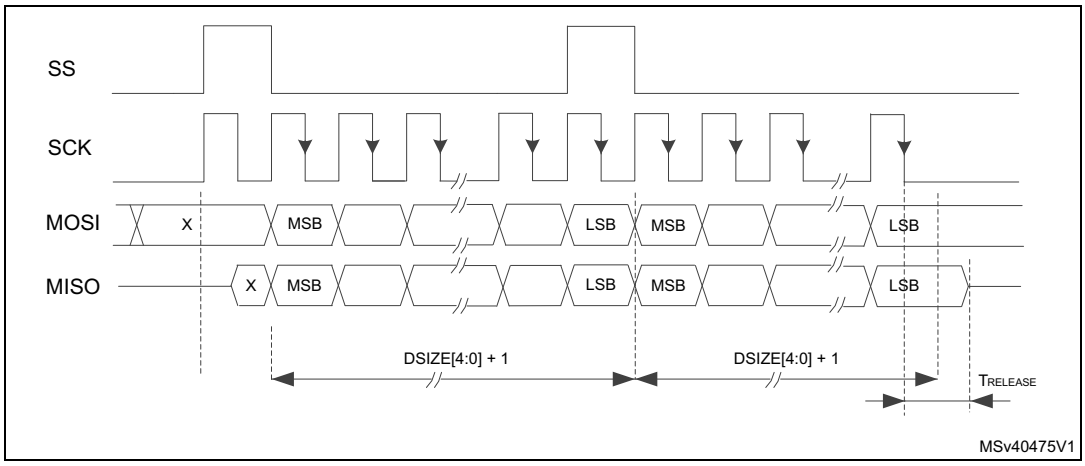
要配置 DMA，禁止小于配置的数据大小的 DMA 数据访问。始终须确保至少访问一个完整的数据帧。

50.5 SPI 特定模式和控制

50.5.1 TI 模式

通过对 SPI_CFG2 寄存器的 SP[2:0] 位字段进行特定设置，可将 SPI 配置为符合 TI 协议。SCK 和 SS 信号极性、相位和流以及位顺序均固定，因此在 SPI 处于 TI 模式配置下时，无需 CPOL、CPHA、LSBFRST、SSOM、SSOE、SSIOP 和 SSM 的设置。SS 信号通过 LSB 数据位的脉冲同步协议，如*图 604：TI 模式传输*所示。

图 604. TI 模式传输



在从模式下，时钟发生器用于定义当前事务完成时，MISO 引脚的从设备输出变为高阻态的时间。主波特率设置（SPI_CFG1 中的 MBR[2:0]）适用，并且可以使用任意波特率，可以非常灵活地确定此时刻。MISO 信号变为 HiZ 的延迟 (TRELEASE) 也取决于内部重新同步，这需要另外 2-4 个时钟信号周期来馈送到发生器。具体公式如下：

$$\frac{Tbaud}{2} + 2 \times Tspi_ker_ck \leq Trelease \leq \frac{Tbaud}{2} + 4 \times Tspi_ker_ck$$

如果从设备在进行数据事务时检测到错位的 SS 脉冲，TIFRE 标志将置 1。

50.5.2 SPI 错误标志

如果以下其中一个错误标志置 1 且已通过将相应的中断使能位置 1 使能了中断，则将生成 SPI 中断。

上溢标志 (OVR)

当主设备或从设备接收了数据但 RxFIFO 没有足够的空间来存储接收的数据时，将出现上溢的情况。如果软件或 DMA 没有足够的时间来读取之前接收到的数据（存储在 RxFIFO 中），则可能会发生这种情况。

当出现上溢的情况时，OVR 标志置 1，新接收的值不会覆盖 RxFIFO 中之前的值。新接收的值将被丢弃，之后发送的所有数据都将丢失。OVRIE 位置 1 时 OVR 标志会触发中断。向 SPI_IFCR 的 OVRC 位写入 1 可以将 OVR 位清零。为了防止接下来发生溢出事件，必须在通过软件读取清空 RxFIFO 后进行清零。建议尽可能释放 RxFIFO 空间，这意味着根据 RXP

标志的指示读出所有可用数据包。在主模式下，用户可通过自动通信挂起（MASRX 位）来防止出现 RxFIFO 上溢。

下溢标志 (UDR)

在从设备发送模式下，如果在 UDRDET 位指定的时刻，从设备 TxFIFO 中没有数据可用于发送，则会由硬件在内部捕获下溢情况。UDR 标志设置随后通过硬件传播到状态寄存器（请参见下面的注）。UDRIE 位置 1 时 UDR 会触发中断。

捕获到下溢情况后，下一个为发送提供的数据将取决于 UDRCFG 位。从设备可提供最后存储到其 TxFIFO 中的数据、之前从主设备接收到的数据，或者由用户存储到 UDRDR 寄存器中的常数模式。第二种配置可用于环形拓扑结构（请参见*图 596*）。软件将 UDR 标志清零后会重新使能标准发送，此清零操作会由硬件传播到 SPI 逻辑。用户必须向 TxFIFO 中写入一些数据，然后再清零 UDR 标志，以防捕获到下一个下溢情况。

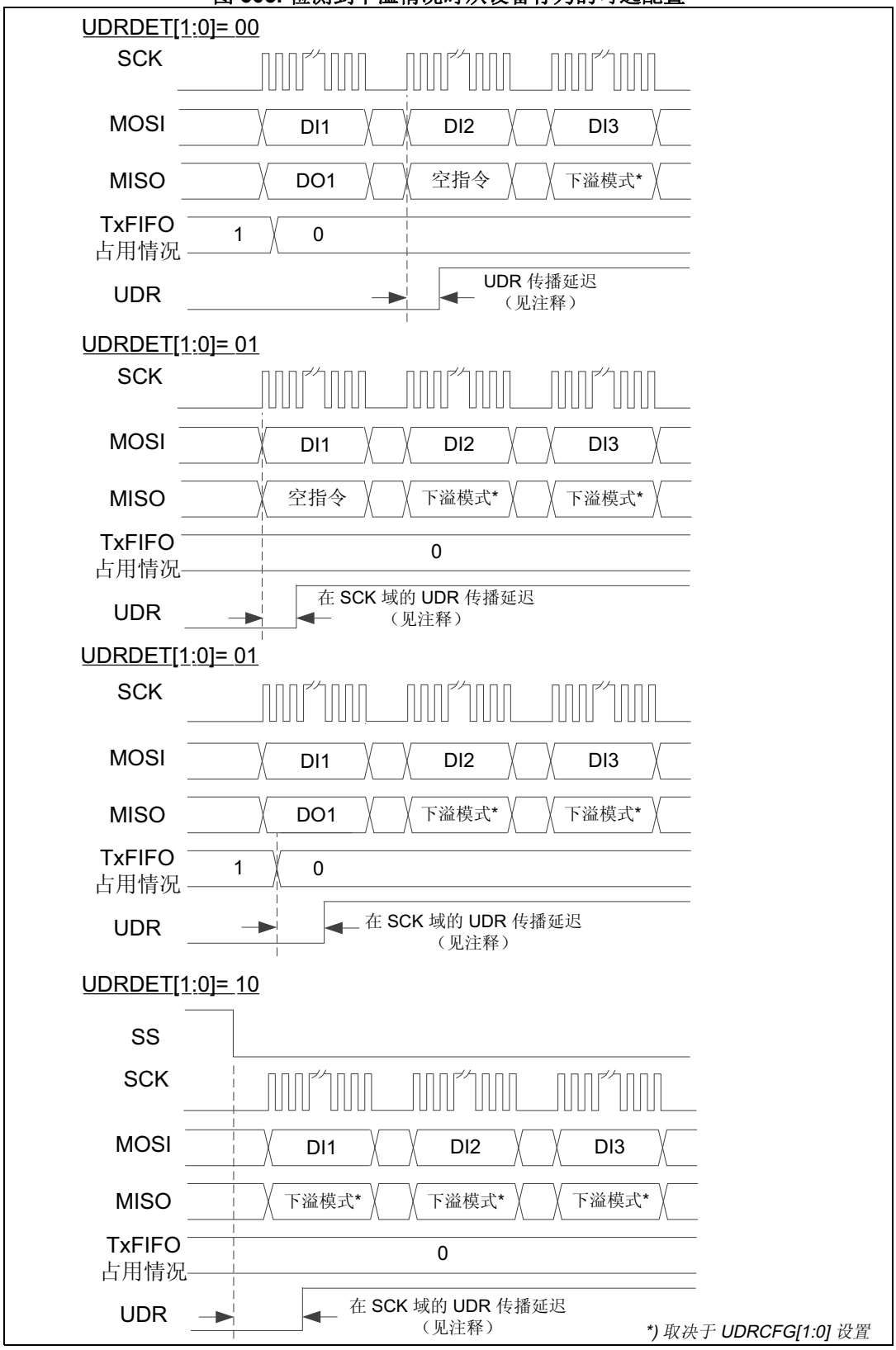
当应用配置 UDRDET[1:0]=00 时，只要 TxFIFO 为空时主设备开始传送新数据帧，就会对下溢状况进行评估。然后，始终会在最后一个有效数据与 UDRCFG[1:0] 寄存器定义的适当下溢模式之间插入一个附加的空（意外）数据（请参见）。当应用任何其他 UDRDET[1:0] 配置时，如果检查下溢条件时从设备的 TxFIFO 不为空，则不会发生这种情况（请参见*图 605：检测到下溢情况时从设备行为的可选配置*）。

从设备处理的数据不可预测，尤其是在 TxFIFO 为空，且尚未捕获或刚清除完下溢条件时，开始或继续事务的情况下，更为如此。通常，UDRDET[1:0] = 00 或刚刚使能 SPI 时，或者在刚刚开始已规定大小的事务时，会出现这种情况。在这种情况下，前几个位可能会损坏，此外，在数据事务开始之前过于紧密地向空 TxFIFO 中写入第一个数据（到 TxFIFO 的数据传播会占用几个 APB 时钟周期）时，也会出现这种情况。如果用户不能确保适时将数据写入空 TxFIFO 中，必须避免 UDRDET[1:0] = 00 设置。

要正确处理下溢控制功能，用户必须避免以下严重干扰：

- 主设备开始提供时钟时对空 TxFIFO 进行的任何填充操作（特别是当 UDRDET[1:0] = 00 时）
- 当 TxFIFO 为空时对 UDR 标志进行的任何清零操作
- 任何 UDRDET[1:0]=00 与 UDRCFG[1:0]=10 的设置（以避免未定义的空数据重复）
- 当帧与帧之间 SS 信号不翻转的情况下，必须在每帧检测数据下溢，此时设置 UDRDET[1:0] = 10
- 在 SS 由软件管理时设置 UDRDET[1:0] = 10

图 605. 检测到下溢情况时从设备行为的可选配置



注： *要对 UDR 事件进行硬件传播，总线上需要有额外流量。传播始终需要事件发生后的几个额外的 SPI 时钟周期（硬件捕获下溢并由软件清零）。如果在数据帧事务接近结束时通过软件将 UDR 标志清零，或者 SCK 线在帧之间处于空闲状态，则在再次处理来自 TxFIFO 的有效数据之前，会先由从设备在内部发送下一个额外下溢模式。用户可以通过在会话之间禁止/使能 SPI 来防止出现这种情况，以重新启动下溢逻辑，从而通过有效数据启动下一个会话。*

模式故障 (MODF)

当主设备的内部 SS 信号（SS 硬件模式下为 SS 引脚，SS 软件模式下为 SSI 位）被拉低时，将发生模式故障。这将以以下方式自动影响 SPI 接口：

- 如果 MODFIE 位置 1，MODF 位将置 1，并触发中断请求。
- SPE 位强制为零，同时 MODF 位置 1。这将阻止所有外设输出，并禁止 SPI 接口。
- MASTER 位清零，从而强制设备进入从模式。

向 SPI_IFCR 的 MODFC 位写入 1 可以将 MODF 清零。

为避免在包含多个 MCU 的系统中发生多从冲突，在重新使能 SPI 之前，必须将 SPE 位置 1，以将 SS 引脚拉至无效电平。

为安全起见，硬件不允许在 MODF 位置 1 时将 SPE 位置 1。在从设备中，MODF 位不可置 1，但由前一次多主模式冲突引起时除外。

在多主系统中，当主设备接管总线时，正确的软件程序必须如下所示：

- 在 SSOE = 0 时切换到主模式
  （如果另一个主设备占用总线，可能会发生冲突。在这种情况下会引发 MODF，从而阻止任何下一个节点切换到主模式）
- 将专用于另一个主设备 SS 控制的 GPIO 引脚置于有效电平
- 执行数据事务
- 将专用于另一个主设备 SS 控制的 GPIO 引脚置于非有效电平
- 切换回从模式

CRC 错误 (CRCE)

当 SPI_CFG1 寄存器中的 CRCEN 位置 1 时，此标志用于验证接收数据的有效性。在接收了最后一个数据后（由 TSIZE 定义），如果移位寄存器中接收的值与接收器 SPI_RXCRC 的值不匹配，则 SPI_SR 寄存器中的 CRCE 标志将置 1。RCEIE 位置 1 时 CRCE 标志会触发中断。向 SPI_IFCR 的 CRCEC 位写入 1 可以将 CRCE 位清零。

TI 模式帧格式错误 (TIFRE)

如果 SPI 在从模式下工作，并配置为符合 TI 模式协议，则在通信进行期间出现 SS 脉冲时，将检测到 TI 模式帧格式错误。出现此错误时，SPI_SR 寄存器中的 TIFRE 标志将置 1。发生错误时不会禁止 SPI，但会忽略 SS 脉冲，并且 SPI 会等待下一个 SS 脉冲，然后再开始新的传输。由于错误检测可能导致丢失几个数据字节，因此数据可能会损坏。

向 SPI_IFCR 的 TIFREC 位写入 1 可以将 TIFRE 标志清零。如果 TIFREIE 位置 1，则检测到 SS 错误时将生成中断。由于无法再保证数据一致性，因此应由软件重新启动主从设备之间的通信。

50.5.3 CRC 计算

为检查发送数据和接收数据的可靠性，使用两个独立的 33 位或 17 位 CRC 计算器。对于最大数据大小为 32 位的情况，SPI 可提供 5 到 33 位的 CRC 多项式长度（*CRC 计算涉及的位数*）；对于最大数据大小被限制为 16 位的外设实例，则可提供 5 到 17 位的多项式长度。

多项式的长度由存储在 SPI_CRCPOLY 寄存器中的值的最高有效位定义，它必须大于 SPI_CFG1 寄存器的 DSIZE[4:0] 位字段所定义的数据帧大小（位数）。要获得完整大小的多项式，多项式长度必须大于外设实例的最大数据大小，而且必须将 SPI_CR1 寄存器的 CRC33_17 位置 1 以选择多项式字符串的最高有效位。例如，要选择标准的 CRC16-CCITT (XMODEM) 多项式 x^16 + x^12 + x^5 + 1，对于 32 位实例，向 SPI_CRCPOLY 寄存器写入 0x11021，而要获得 16 位实例的完整大小，则要写入 0x1021，同时将 CRC33_17 位置 1。

SPI_CFG1 中的 CRCSIZE 位字段随后会定义 CRC 计算寄存器中被处理并与 CRC 帧进行比较的最高有效位数。其定义与数据帧长度无关，但必须等于数据帧大小或是数据帧大小的整数倍，且不能超过实例的最大数据大小。

为了充分获得 CRC 计算功能的好处，多项式长度设置必须与 CRC 模式大小相对应，否则会对计算中未使用的位进行处理，并且如果其大小设置为大于多项式长度，则 CRC 模式末尾应全部为零。

CRC 原理

在使能 SPI (SPE = 1) 前，通过将 SPI_CFG1 寄存器中的 CRCEN 位置 1 来使能 CRC 计算。然后使用由 CRCPOLY 寄存器和 CRC33_17 位定义的 CRC 多项式计算 CRC 值。如果使能了 SPI，则可以更改 CRC 多项式，但只有在总线上无通信时才可进行更改。

按照由 SPI_CR1 寄存器的 CPHA 位和 CPOL 位定义的采样时钟边沿，逐位进行 CRC 计算。会在 SPI_CR2 寄存器专门定义的数据块末尾，自动检查计算得到的 CRC 值。

如果在接收到的数据内部计算得出的 CRC 与从设备发送器接收到的 CRC 之间检测到不匹配，则 CRCERR 标志将置 1，以指示存在数据损坏错误。CRC 的正确处理步骤取决于 SPI 配置和所选的传输管理。

CRC 传输管理

通信开始后将一直持续到必须发送完或接收完 SPI_DR 寄存器中的最后一个数据帧。

传输的长度必须由 TSIZE 和 TSER 定义。处理了所需数量的数据后，将发送 TXCRC，并将线路上接收到的数据与 RXCRC 值进行比较。

如果使能了 CRC，则不能将 TSIZE 设置为值 0xFFFF。例如，发送 65535 个具有 CRC 的数据的正确方式是设置：

- TSIZE = 0xFFFE 和 TSER = 1（数据包配置为保存一个数据时）
- TSIZE = 0xFFFC 和 TSER = 3（数据包保存 4 个数据时——以确保 TSIZE 值与使用了扩展位的数据包大小对齐）。

在发送过程中，CRC 计算在 CRC 事务期间被冻结，TXCRC 将以帧的形式进行发送，帧的长度等于 CRCSIZE 位字段的值。

在接收过程中，处理完所需的数据量后，RXCRC 也会被冻结。随后以帧的形式接收要与 RXCRC 寄存器内容进行比较的信息，帧长等于 CRCSIZE 值。

完成 CRC 帧后，将执行自动校验，将接收的 CRC 值与 SPI_RXCRC 寄存器中计算的值进行比较。软件必须校验 SPI_SR 寄存器中的 CRCERR 标志，以确定数据传输是否损坏。软件通过向 CRCERRC 写入 1 将 CRCERR 标志清零。

用户无需考虑刷新冗余 CRC 信息，该过程将自动完成。

复位 SPI_TXCRC 和 SPI_RXCRC 值

在 CRC 阶段后对新数据进行采样时，SPI_TXCRC 和 SPI_RXCRC 值将自动初始化。借此，可使用 DMA 循环模式来不间断地传输数据（中间 CRC 校验阶段会涉及一些数据块）。接收器和发送器的初始化模式可配置为零或全 1，具体取决于对 SPI_CR1 寄存器的位 TCRCINI 和 RCRCINI 进行的设置。

禁止 SPI 时，CRC 值复位。

50.6 低功耗模式管理

SPI 具有先进的低功耗模式功能，即使禁止 **spi_pclk** 时钟，也仍能在 FIFO 和串行接口之间正常传输数据。

在主模式下，需要 **spi_ker_ck** 内核时钟来提供串行接口的时序。

在从模式下，还可在 FIFO 和串行接口之间进行数据传输时删除 **spi_ker_ck** 时钟。在该模式下，时钟由外部 SPI 设备提供。

对 **spi_pclk** 时钟进行门控（如果 SPI 处于从模式下，则还对 **spi_ker_ck** 时钟进行门控）时，如果诸如以下某个特定操作需要激活 **spi_pclk** 时钟，则 SPI 会提供唤醒事件信号 (**spi_wkup**)：

- 填充 TxFIFO
- 清空 RxFIFO
- 发出其他信号：传输结束、错误...

spi_ker_ck 和 **spi_pclk** 时钟的生成由 RCC 块根据寄存器设置和处理器模式进行控制。更多详细信息，请参见 RCC 部分。

Spi_pclk 时钟请求保持挂起状态，直到已使能中断的相关标志保持置 1 状态时为止。因此，在对低功耗敏感的系统中务必尽快处理这些挂起的请求并清零其标志，并且应用必须在将 SPI 切换到低功耗模式（即移除 **spi_pclk**）之前确认所有挂起的中断事件。

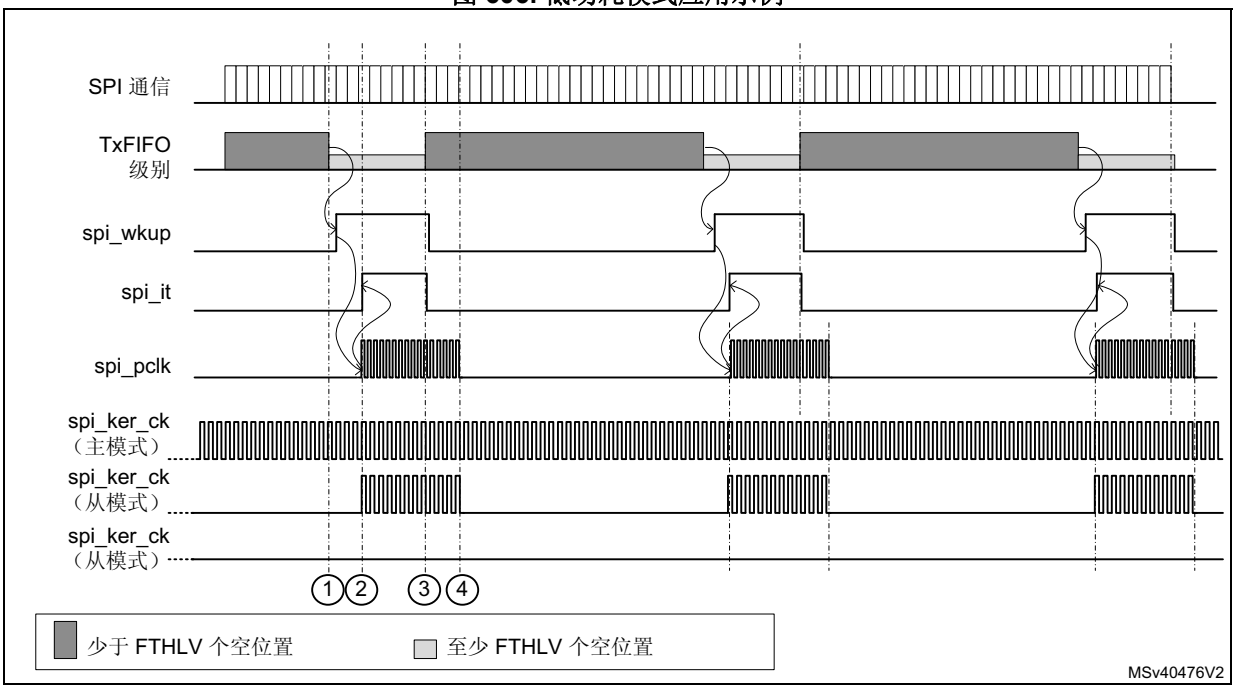
图 606 所示为 SPI2S 工作在低功耗模式时的时钟处理示例。给出的示例适用于发送模式。

在主模式下，生成时序需要 **spi_ker_ck** 时钟。

图 606 所示为从模式支持的两种 **spi_ker_ck** 内核时钟处理情形：

- 在大多数从模式下，均可禁止 **spi_ker_ck** 内核时钟
- 对于某些产品，可能要根据系统状态激活 **spi_ker_ck** 内核时钟。

图 606. 低功耗模式应用示例



该图清楚地显示，需要将数据从内存传输到 SPI2S TxFIFO 时，必须向 SPI2S 提供 **spi_pclk**。下面对最重要步骤进行了说明：

- **第 1 步：**
  TxFIFO 级别低于编程的阈值，此事件 (TXP) 会激活 **spi_wkup** 信号。该信号通常用于从低功耗模式唤醒系统，从而激活总线时钟 (**spi_pclk**)。
- **第 2 步：**
  spi_pclk 激活时，**spi_it** 也会激活，产品已准备好通过 DMA 或软件来填充 TxFIFO。另请注意，对于某些产品，系统唤醒还会自动使能 **spi_ker_ck** 内核时钟。
- **第 3 步**
  TxFIFO 中的空位置数小于 FTHLV 时，将禁用 **spi_wkup** 和 **spi_it** 信号，但 TxFIFO 填充过程可能仍会继续。请注意，**spi_wkup** 下降沿与串行接口时钟域对齐，**spi_it** 的下降沿与 **spi_pclk** 时钟域对齐。
- **第 4 步：**
  TxFIFO 填充已完成；软件可将系统切换回低功耗模式，直至出现下一个 **spi_wkup**。

50.7 SPI 唤醒和中断

表 447 概述了能够生成中断事件 (**spi_it**) 的 SPI 事件。其中的某些事件还具有低功耗模式唤醒功能 (**spi_wkup**)。

大多数事件均可在使用特定中断使能控制位时单独使能和禁止。

与事件相关的标志通过特定方法清零。有关事件标志的更多详细信息，请参见 SPI 寄存器的说明。一个未清除且已使能中断的请求会生成一个 SPI 外设时钟 (spi_pclk) 请求，从而增加整体功耗。

表 447. SPI 唤醒和中断请求

中断事件	事件标志(1)	使能控制位	事件清除方法	中断/唤醒是否已激活	
				spi_it	spi_wkup
TxFIFO 准备就绪，可装载数据（存在可供一个数据包使用的空间——FIFO 阈值）	TXP	TXPIE	TxFIFO 包含的空位置数少于 FTHLV 个时，TXP 由硬件清零	是	是
在 RxFIFO 中接收到数据（一个数据包可用——FIFO 阈值）	RXP	RXPIE	RxFIFO 包含的采样数少于 FTHLV 个时，RXP 由硬件清零		是
TXP 和 RXP 均有效	DXP	DXPIE	TXP 或 RXP 被清零时		是
发送传输已填充	TXTF	TXTFIE	向 TXTFC 写入 1		否
下溢	UDR	UDRIE	向 UDRC 写入 1		是
上溢	OVR	OVRIE	向 OVRC 写入 1		是
CRC 错误	CRCE	CRCEIE	向 CRCEC 写入 1		是
TI 帧格式错误	TIFRE	TIFREIE	向 TIFREC 写入 1		否
模式故障	MODF	MODFIE	向 MODFC 写入 1		否
传输结束（完整传输序列完成——基于 TSIZE 值）	EOT	EOTIE	向 EOTC 写入 1		是
主模式处于挂起状态	SUSP		向 SUSPC 写入 1		是
TxFIFO 发送完成（TxFIFO 为空）	TXC(2)		总线上开始进行发送活动时，TXC 由硬件清零		否
TSER 值传输到 TSIZE（可将新值装载到 TSER）	TSERF	TSERFIE	向 TSERFC 写入 1		否

1. 有关事件标志的更多详细信息，请参见 SPI2S 寄存器说明。
2. TXC 标志的行为取决于 TSIZE 设置。当 TSIZE>0 时，该标志完全遵循 EOT 标志，包括由 EOTC 清零。

50.8 I2S 主要特性

- 全双工通信
- 半双工通信（仅作为发送器或接收器）
- 主模式或从模式操作
- 8 位可编程线性预分频器
- 数据长度可以是 16 位、24 位或 32 位[a]
- 在主模式下通道长度可以是 16 位或 32 位，在从模式下则可为任一值
- 可编程时钟极性
- 错误标志指示，以提高可靠性：下溢、上溢和帧错误
- 嵌入式 Rx 和 TxFIFO
- 支持的 I^2S 协议：
  - I^2S Philips 标准
  - MSB 对齐标准（左对齐）
  - LSB 对齐标准（右对齐）
  - PCM 标准（支持短帧同步和长帧同步）
- 数据排序可编程（LSb 在前或 MSb 在前）
- 用于发送和接收的 DMA 功能
- 可输出主时钟以驱动外部音频元件：
  - 对于所有 I2S 模式，$F_{MCK} = 256 \times F_{WS}$
  - 对于所有 PCM 模式，$F_{MCK} = 128 \times F_{WS}$

注： *F_{MCK} 是主时钟频率，F_{WS} 是音频采样频率。*

a. 24 位和 32 位数据宽度并非始终可用。请参见*第 50.3 节：SPI 实现*。

50.9 I2S 功能描述

50.9.1 I2S 一般描述

图 591 所示的框图同样适用于 I2S 模式。

I2SMOD 位置 1 时，SPI/I2S 块可工作在 I2S/PCM 模式下。专用寄存器 (SPI_I2SCFGR) 可用于配置专用 I2S 参数，包括时钟发生器和串行链路接口。

将 SPI/I2S 置于主模式下时，I2S/PCM 功能使用时钟发生器生成通信时钟。此时钟发生器也是主时钟输出 (MCK) 的源。

诸如 RxFIFO、TxFIFO、DMA 等资源和部分中断信号与 SPI 功能共享。低功耗模式功能也可用于 I2S 模式，请参见*第 50.6 节：低功耗模式管理*和*第 50.10 节：I2S 唤醒和中断*。

50.9.2 与 SPI 功能共享的引脚

I2S 与 SPI 共享以下四个共用引脚：

- SDO：串行数据输出（映射到 MOSI 引脚上），用于在主模式下发送音频采样以及在从模式下接收音频采样。请参见*第 2557 页的串行数据线交换一节*。
- SDI：串行数据输入（映射到 MISO 引脚上），用于在主模式下接收音频采样以及在从模式下发送音频采样。请参见*第 2557 页的串行数据线交换一节*。
- WS：字选择（映射到 SS 引脚上），表示帧同步。它在主模式下被配置为输出，在从模式下被配置为输入。
- CK：串行时钟（映射到 SCK 引脚上），表示串行位时钟。它在主模式下被配置为输出，在从模式下被配置为输入。

当某些外部音频设备需要使用主时钟输出时，可以使用其他引脚：

- MCK：主时钟（单独映射），适用于 I2S 配置为主模式的情况。

50.9.3 I2S/PCM 模式下可用的位字段

选择 I2S/PCM 模式（I2SMOD = “1”）时，一些位字段不再可用，必须将其强制设为特定值，以便保证 I2S/PCM 功能正常工作。*表 448* 列出了 I2S/PCM 模式下可用的位和位字段，并指示了其中哪些必须强制设为特定值。

表 448. PCM/I2S 模式下可用的位字段

寄存器名称	PCM/I2S 模式下可用的位字段	关于其他位字段的限制
SPI/I2S 控制寄存器 1 (SPI_CR1)	IOLOCK、CSUSP、CSTART、SPE	其他位字段设为其各自的复位值
SPI 控制寄存器 2 (SPI_CR2)	-	设为复位值
SPI 配置寄存器 1 (SPI_CFG1)	TXDMAEN、RXDMAEN、FTHLV	其他位字段设为其各自的复位值
SPI 配置寄存器 2 (SPI_CFG2)	AFCNTR、LSBFRST、IOSWP	其他位字段设为其各自的复位值
SPI/I2S 中断使能寄存器 (SPI_IER)	TIFREIE、OVRIE、UDRIE、TXPIE、RXPIE	
SPI/I2S 状态寄存器 (SPI_SR)	SUSP、TIFRE、OVR、UDR、TXP、RXP	其他标志不相关
SPI/I2S 中断/状态标志清零寄存器 (SPI_IFCR)	SUSPC、TIFREC、OVRC、UDRC	其他位字段设为其各自的复位值
SPI/I2S 发送数据寄存器 (SPI_TXDR)	整个寄存器	-
SPI/I2S 接收数据寄存器 (SPI_RXDR)	整个寄存器	-
SPI 多项式寄存器 (SPI_CRCPOLY)	-	设为复位值
SPI 发送器 CRC 寄存器 (SPI_TXCRC)	-	
SPI 接收器 CRC 寄存器 (SPI_RXCRC)	-	
SPI 下溢数据寄存器 (SPI_UDRDR)	-	
SPI/I2S 配置寄存器 (SPI_I2SCFGR)	整个寄存器	-

50.9.4 从模式和主模式

SPI/I2S 块都支持采用 I2S 或 PCM 协议的主模式和从模式。

在主模式下，CK、WS 和 MCK 信号均被设置为输出。

在从模式下，CK 和 WS 信号被设置为输入。从模式下无法使用信号 MCK。

为了改善从模式下 SPI/I2S 块的稳健性，外设根据 WS 信号在每次接收和发送时再同步。这意味着：

- 使用 I2S Philips 标准时，在每个 WS 沿跳变之后，每个数据的移入或移出都会被触发一个时钟周期。
- 使用 I2S MSB 对齐标准时，如果检测到 WS 沿跳变，则会触发各个数据的移入或移出。
- 在 PCM 标准中，每个数据的移入或移出操作，均在 WS 信号的有效沿之后一个位时钟触发。

注： *这种再同步机制不适用于 I2S LSB 对齐标准。*

注： *另请注意，将 SPI/I2S 配置为从模式时，无需提供内核时钟。*

50.9.5 支持的音频协议

I2S/PCM 接口支持四种音频标准，可使用 SPI_I2SCFGR 寄存器中的 I2SSTD[1:0] 和 PCMSYNC 位对其进行配置。

使用 I2S 协议时，在左右两个通道上时分复用音频数据。WS 信号用于指示哪个通道应被视为左通道，以及哪个通道应被视为右通道。

在 I2S 主模式下，支持以下四种帧格式：

- 将 16 位数据封装在 16 位通道中
- 将 16 位数据封装在 32 位通道中
- 将 24 位数据封装在 32 位通道中(a)
- 将 32 位数据封装在 32 位通道中(a)

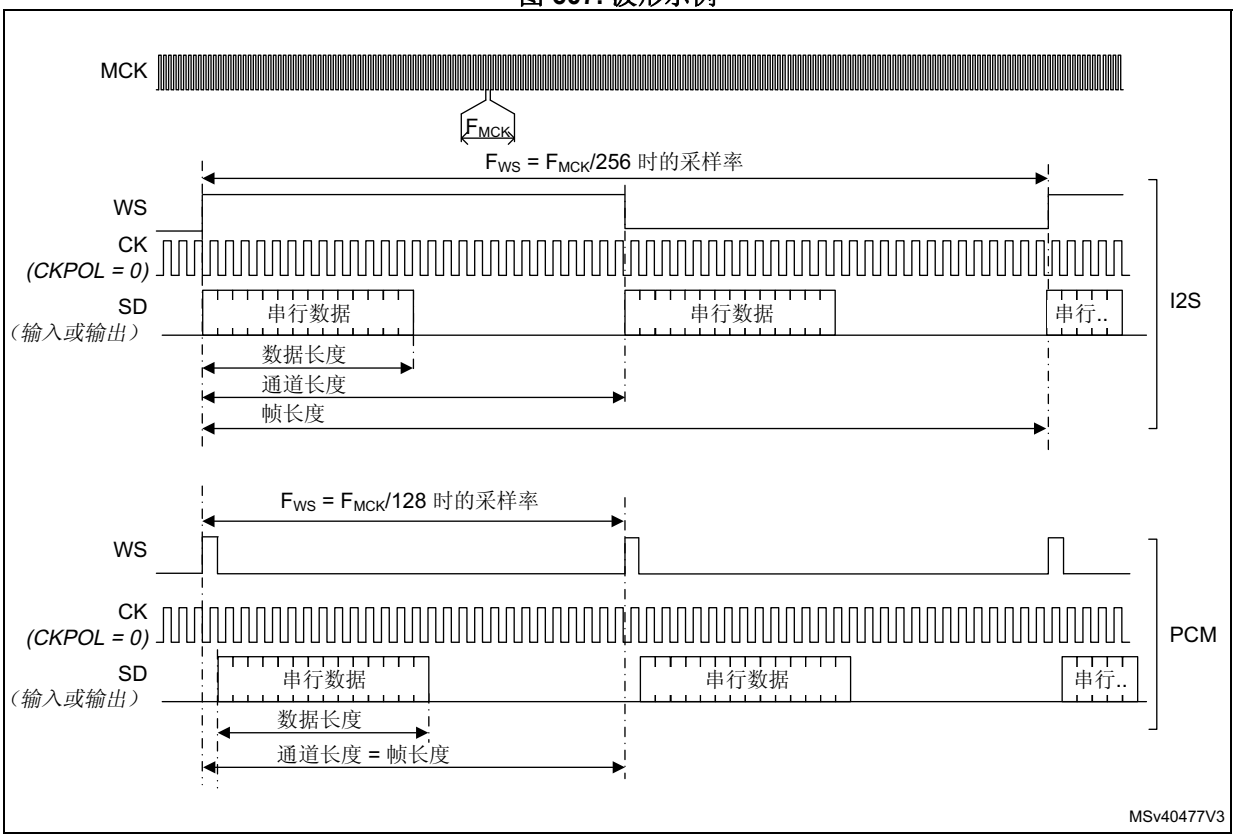
在 PCM 主模式下，支持以下三种帧格式：

- 将 16 位数据封装在 16 位通道中
- 将 16 位数据封装在 32 位通道中
- 将 24 位数据封装在 32 位通道中(a)

a. 24 位和 32 位数据宽度并非始终可用。请参见*第 50.3 节：SPI 实现*。

下图显示了本节中使用的主要定义：数据长度、通道长度和帧长度。

图 607. 波形示例



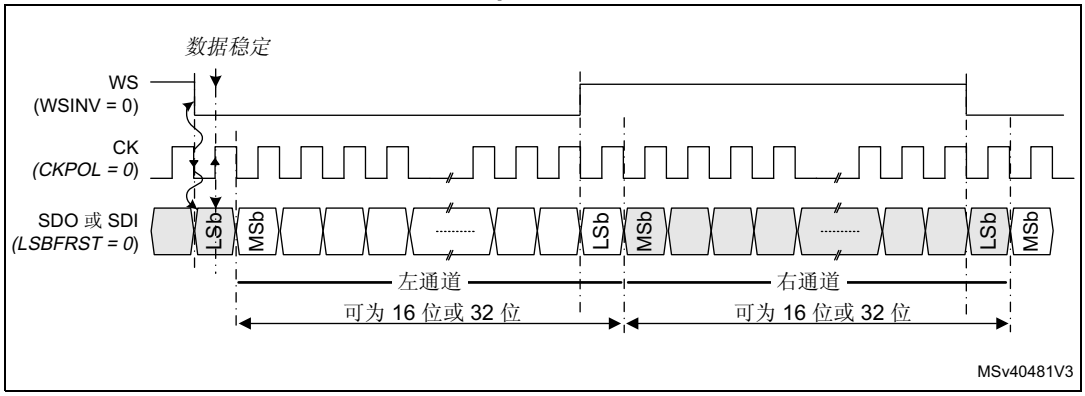
1. 通道长度 (CHLEN) 必须始终大于或等于数据长度 (DATLEN)。

I²S Philips 标准

要选择 I2S Philips 标准，将 I2SSTD 设为 0b00 即可。主模式和从模式均支持该标准。

对于该标准，在第一个位（对于 I2S Philips 标准，为 MSb）可用之前 WS 信号将翻转一个 CK 时钟周期。WS 的下降沿跳变表示下一个传输的数据将用于左通道，上升沿跳变则表示下一个传输的数据用于右通道。

图 608. 主模式 I2S Philips 协议波形（16/32 位全精度）



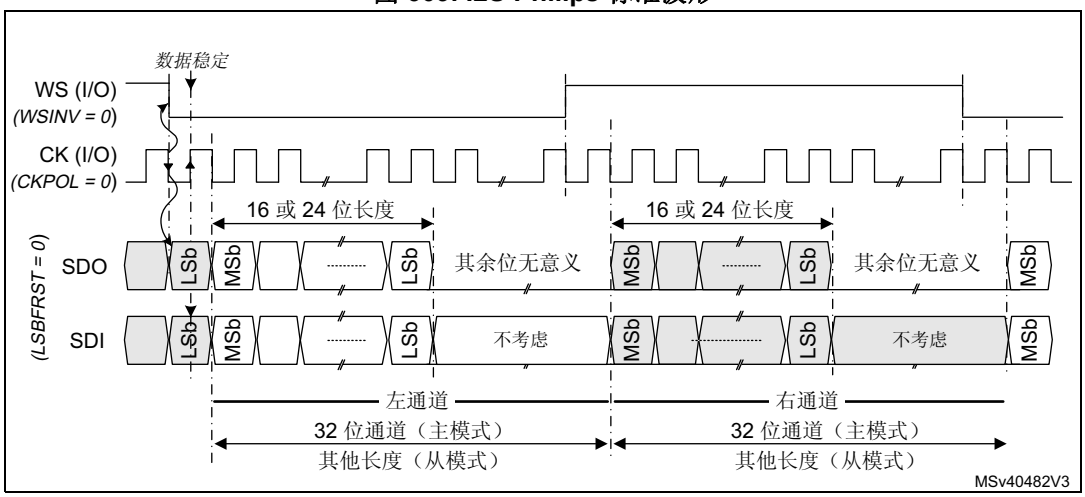
1. 24 位和 32 位数据宽度并不总是受支持（DATLEN = 01 或 10）。有关支持的数据大小，请参见*第 50.3 节：SPI 实现*。

为与 I2S Philips 协议匹配，要将 CKPOL 设为 0。有关 WS 信号处理的信息，请参见*CK 采样边沿的选择*。

图 608 所示为通道长度等于数据长度时 SPI/I2S 所生成的波形示例。更确切地说，这适用于 CHLEN = 0 且 DATLEN = 0b00 或 CHLEN = 1 且 DATLEN = 0b10 的情况。

有关 WS 信号处理的信息，请参见 *WS 反转控制*。

图 609. I2S Philips 标准波形



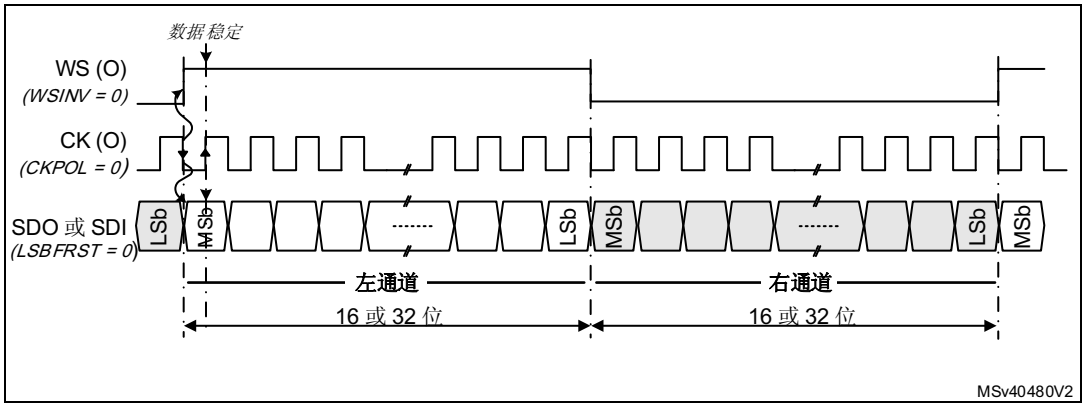
1. 24 位和 32 位数据宽度并不总是受支持（DATLEN = 01 或 10）。有关支持的数据大小，请参见*第 50.3 节：SPI 实现*。

如果通道长度大于数据长度，则将 SPI/I2S 配置为发送模式时，其余位被强制清零。这同时适用于主模式和从模式。

MSB 对齐标准

对于该标准，提供第一个数据位时，WS 信号翻转。如果 WS 为高电平，则传输的数据表示左通道；如果 WS 为低电平则表示右通道。

图 610. 主模式 MSB 对齐的 16 位或 32 位全精度长度

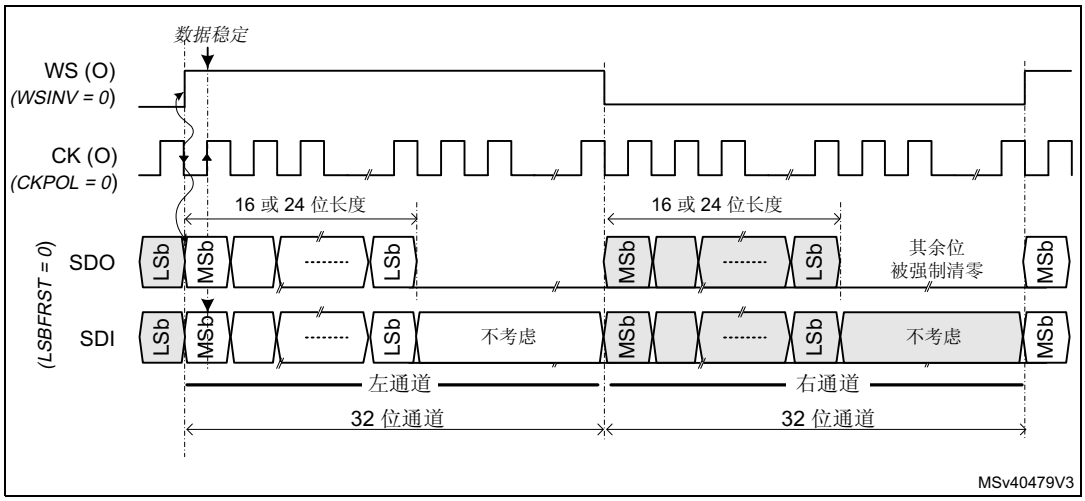


1. 24 位和 32 位数据宽度并不总是被支持（DATLEN = 01 或 10）。有关支持的数据大小，请参见*第 50.3 节：SPI 实现*。

为与 I2S MSB 对齐协议匹配，要将 CKPOL 设为 0。有关 WS 信号处理的信息，请参见 *CK 采样边沿的选择*。

有关 WS 信号处理的信息，请参见 *WS 反转控制*。

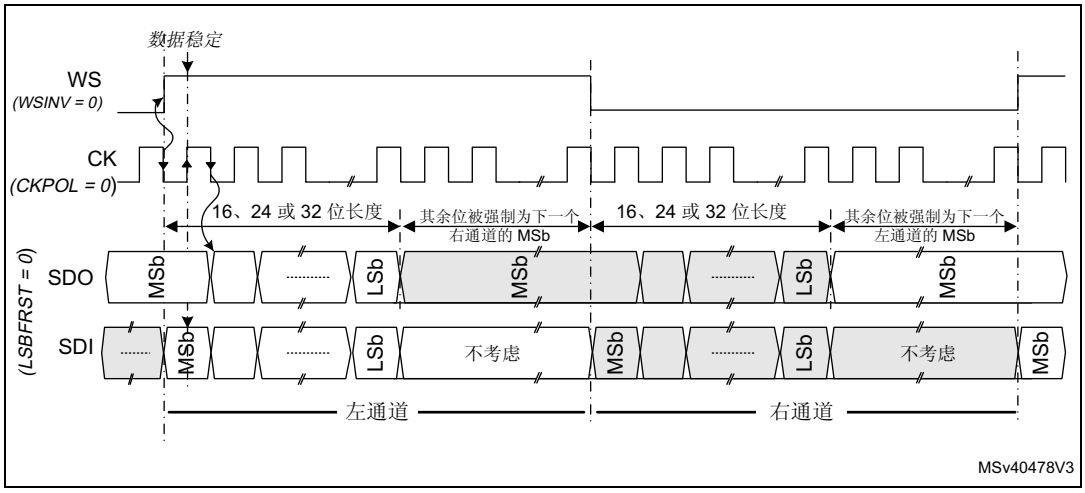
图 611. 主模式 MSB 对齐的 16 位或 24 位数据长度



1. 24 位和 32 位数据宽度并不总是被支持（DATLEN = 01 或 10）。有关支持的数据大小，请参见*第 50.3 节：SPI 实现*。

如果通道长度大于数据长度，则将 SPI/I2S 配置为主设备发送模式时，其余位被强制清零。在从设备发送模式下，其余位被强制为要生成的下一个数据的第一位的值，以避免出现时序问题（请参见*图 612*）。

图 612. 从模式 MSB 对齐



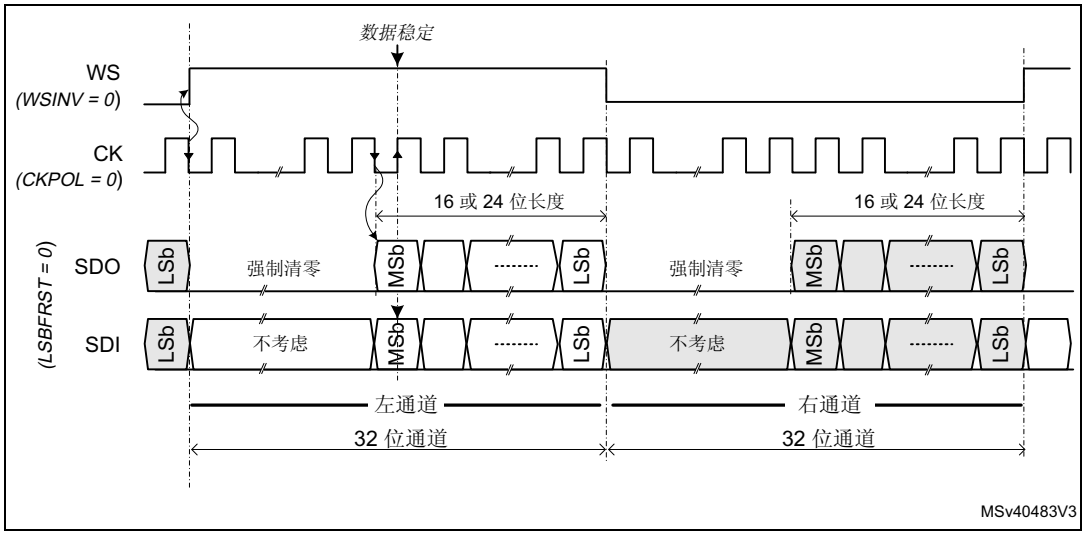
1. 24 位和 32 位数据宽度并不总是被支持（DATLEN = 01 或 10）。有关支持的数据大小，请参见*第 50.3 节：SPI 实现*。

LSB 对齐标准

该标准与主模式下的 MSB 对齐标准类似（对于 16 位和 32 位全精度帧格式，没有任何不同）。LSB 对齐的 16 位或 32 位全精度格式的波形与 MSB 对齐模式下的波形类似（请参见*图 610*），因为通道和数据的长度相同。

注： *在 LSB 对齐格式下，主模式和从模式仅支持 16 位和 32 位通道长度。这是因为，如果发送器和接收器侧不清楚通道长度，则无法正确传输数据。*

图 613. LSB 对齐的 16 位或 24 位数据长度



1. 24 位和 32 位数据宽度并不总是被支持（DATLEN = 01 或 10）。有关支持的数据大小，请参见*第 50.3 节：SPI 实现*。

为与 I2S LSB 对齐协议匹配，要将 CKPOL 设为 0。有关 WS 信号处理的信息，请参见 *CK 采样边沿的选择*。

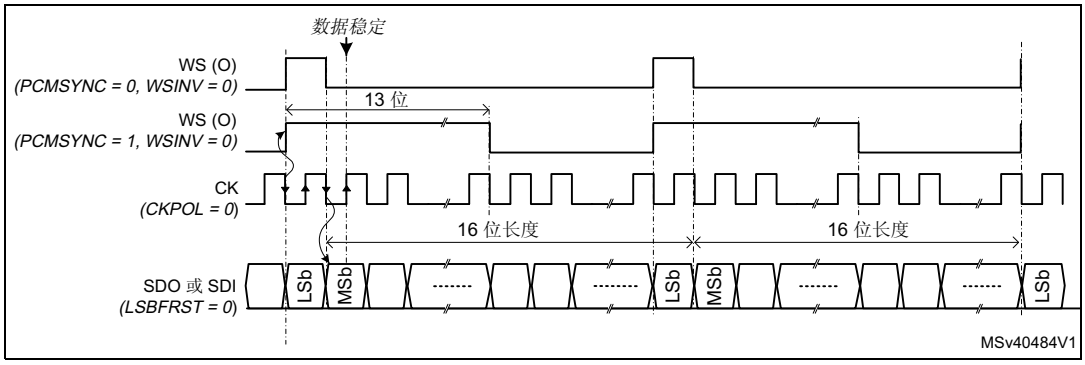
有关 WS 信号处理的信息，请参见 *WS 反转控制*。

PCM 标准

对于 PCM 标准，无需使用通道信息。可使用两种 PCM 模式（短帧和长帧），并且可使用 SPI_I2SCFGR 寄存器中的 PCMSYNC 位来配置。

注： *PCM 长帧和短帧之间的区别仅仅是帧同步的宽度：对于两种协议，帧的有效边沿在第一个位的一个 CK 时钟周期之前被生成（在从模式下则为预期）。*

图 614. 帧长度等于数据长度时的主模式 PCM



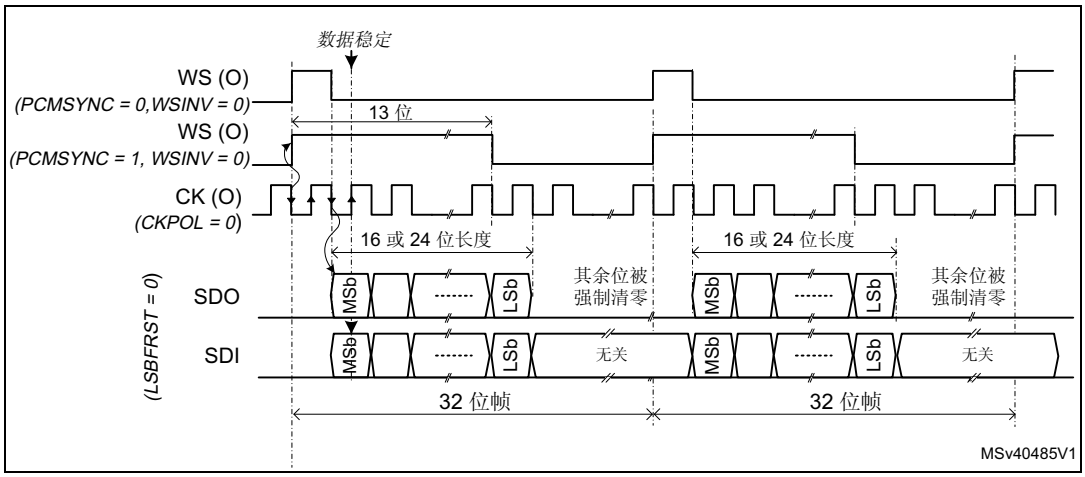
对于长帧同步，在主模式下会将 WS 信号持续 13 个周期。

将通道长度设置为 32 位时，可以使用 16 位或 24 位数据大小。

对于短帧同步，WS 同步信号的持续时间仅为一个周期。

有关 WS 信号处理的信息，请参见 *WS 反转控制*。

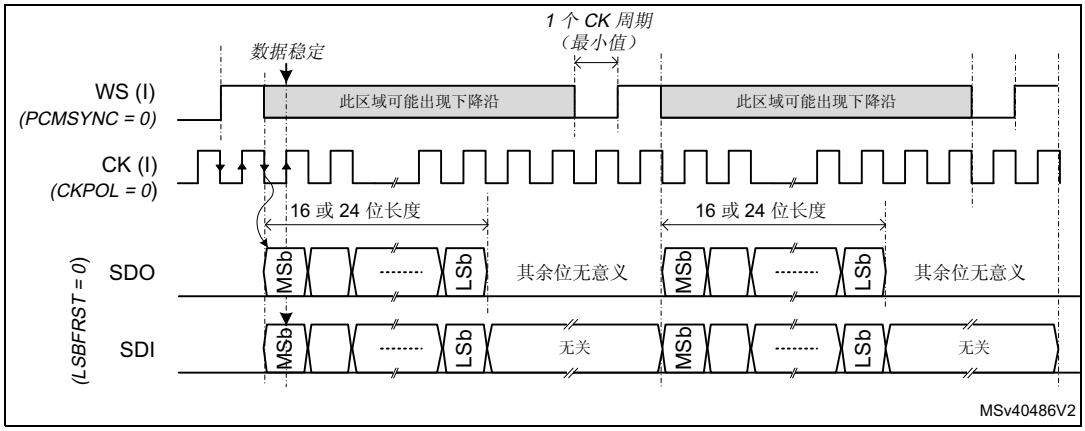
图 615. 主模式 PCM 标准波形（16 位或 24 位数据长度）



1. 24 位和 32 位数据宽度并不总是被支持（DATLEN = 01 或 10）。有关支持的数据大小，请参见*第 50.3 节：SPI 实现*。

如果将 PCM 协议用于从模式，则帧长度可以不是 16 位或 32 位。如*图 616* 所示，在从模式下，由于 WS 的上升沿检测到帧开始，因此可以接受 WS 的各种脉冲宽度。唯一的限制是 WS 必须在至少一个 CK 周期内恢复其非有效状态。

图 616. 从模式 PCM 波形



1. 24 位和 32 位数据宽度并不总是被支持（DATLEN = 01 或 10）。有关支持的数据大小，请参见*第 50.3 节：SPI 实现*。

为与 PCM 协议匹配，要将 CKPOL 设为 0。有关 WS 信号处理的信息，请参见 *CK 采样边沿的选择*。

50.9.6 更灵活的串行接口

从模式下的帧长度可变

在从模式下，只要通道长度大于数据长度，便可接受非 16 位或 32 位的通道长度。这适用于除 I2S LSB 对齐协议外的所有协议。

数据排序

对于所有数据格式和通信标准，可通过 *SPI 配置寄存器 2 (SPI_CFG2)* 的 LSBFRST 位来选择数据排序（MSb 在前或 LSb 在前）。

CK 采样边沿的选择

SPI/I2S 配置寄存器 (SPI_I2SCFGR) 的位 CKPOL 允许用户为从模式和主模式选择 CK 的采样边沿极性，这适用于所有协议。

- CKPOL = 0 时，在 CK 的下降沿更改串行数据 SDO 和 WS（主模式下），在上升沿读取串行数据 SDI 和 WS（从模式下）。
- CKPOL = 1 时，在 CK 的上升沿更改串行数据 SDO 和 WS（主模式下），在下降沿读取串行数据 SDI 和 WS（从模式下）。

WS 反转控制

可通过将 WSINV 置 1，为主模式和从模式反转默认 WS 信号极性，这适用于所有协议。默认情况下，WS 极性如下：

- 使用 I2S Philips 标准时，WS 对于左通道为低电平，对于右通道为高电平
- 在 MSB/LSB 对齐模式下，WS 对于左通道为高电平，对于右通道为低电平
- 在 PCM 模式下，帧起始通过 WS 的上升沿表示。

WSINV 置 1 时，WS 极性将反转，因此：

- 使用 I2S Philips 标准时，WS 对于左通道为高电平，对于右通道为低电平
- 在 MSB/LSB 对齐模式下，WS 对于左通道为低电平，对于右通道为高电平
- 在 PCM 模式下，帧起始通过 WS 的下降沿表示。

WSINV 位于 *SPI/I2S 配置寄存器 (SPI_I2SCFGR)* 中。

IO 的控制

SPI/I2S 块借助 *SPI 配置寄存器 2 (SPI_CFG2)* 的 AFCNTR 位，允许在使能 SPI/I2S 之前将 WS 和 CK 信号稳定在无效状态。

可通过使用以下序列编程 CKPOL 和 WSINV 来完成上述过程：

假设 AFCNTR 最初被设为 0

– 将 I2SMOD 置 1（以通知硬件通过 CKPOL 和 WSINV 控制 CK 和 WS 极性）。
– 将位 CKPOL 和 WSINV 设为所需值。
– 将 AFCNTR 置 1。
随即会根据 CKPOL 和 WSINV 值设置 CK 和 WS I/O 的无效电平，即使尚未使能 SPI/I2S，亦如此。
– 然后执行 I2S/PCM 的激活序列

表 449 所示为将 AFCNTR 位置 1 时以及在使能 SPI/I2S 块之前的 WS 和 CK 信号电平（即，无效电平）。请注意，WS 电平还取决于所选协议。

表 449. AFCNTR = 1 时在使能 SPI/I2S 之前的 WS 和 CK 电平

WSINV	I2SSTD		使能 SPI/I2S 之前的 WS 电平
0	I2S 标准 (00)	→	高
	其他	→	低
1	I2S 标准 (00)	→	低
	其他	→	高

CKPOL		使能 SPI/I2S 之前的 CK 电平
0	→	低
1	→	高

注： *为避免 WS 和 CK 上发生电气冲突，当 SPI2S 处于从模式时，AFCNTR 位不得置 1。*

串行数据线交换

SPI/I2S 能通过 *SPI 配置寄存器 2 (SPI_CFG2)* 的 IOSWP 位来交换 SDI 和 SDO 线的功能。*表 450* 给出了该功能的详细信息。

表 450. 串行数据线交换

配置	IOSWP	SDI 方向	SDO 方向
主/从 RX	0	IN	-
	1	-	IN
主/从 TX	0	-	OUT
	1	OUT	-

表 450. 串行数据线交换（续）

配置	IOSWP	SDI 方向	SDO 方向
主/从全双工	0	IN	OUT
	1	OUT	IN

简单起见，*I2S 功能描述*部分所示的波形都是基于 IOSWP = 0。

50.9.7 启动序列

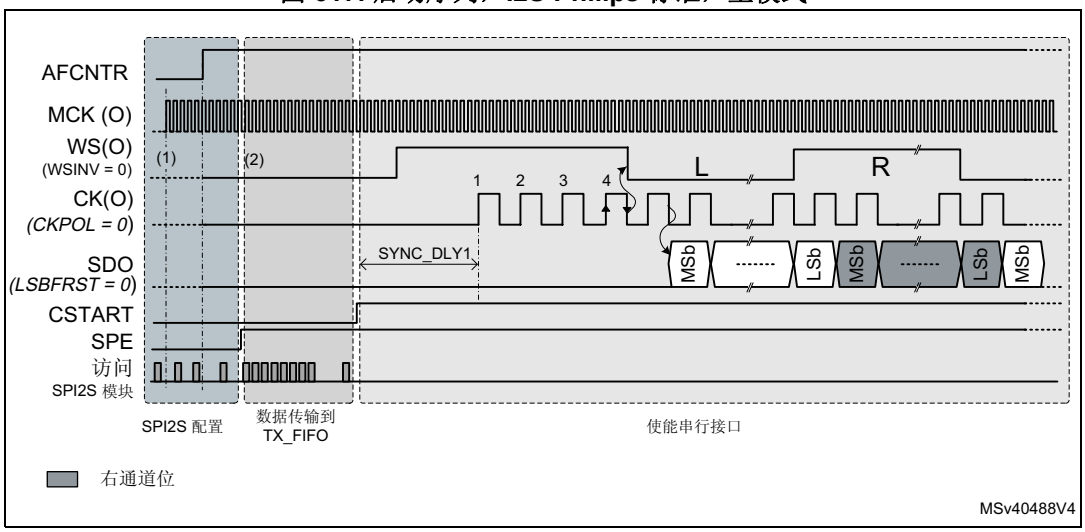
将 SPE 位设为 0 时，不允许用户对 SPI_RXDR 和 SPI_TXDR 寄存器进行读操作和写操作，但允许访问其他寄存器。

如果应用要使用 SPI/I2S 块，则用户必须按以下步骤进行操作：

1. 确保将 SPE 设为 0，否则向 SPE 写入 0。
2. 根据所需配置去编程所有配置和控制寄存器。有关详细的编程示例，请参见 *第 50.9.16 节*。
3. 通过将 SPI_IFCR 寄存器的 SUSPC、TIFREC、OVRC 和 UDRC 位置 1 来清零所有状态标志。请注意，如果未将标志 SUSP 清零（通过 SUSPC 位），则 CSTART 控制位无效。
4. 将 SPE 位置 1，以激活 SPI/I2S 块。该位置 1 时，串行接口仍然被禁止，但 DMA 和中断服务处于工作状态，因而可将数据传输到 TxFIFO。
5. 将位 CSTART 置 1，以激活串行接口。

如*图 617* 所示，对于符合 I2S Philips 标准的主 TX，一旦 CSTART 位置 1 且 TxFIFO 非空，就会在一个重新同步延迟 (SYNC_DLY2) 后开始生成 WS 和 CK 信号。请注意，在 WS 下降沿之前的 4 个上升沿，位时钟 CK 处于激活状态，以确保外部从设备可正确检测 WS 沿跳变。其他标准的相应行为与此类似。

图 617. 启动序列，I2S Philips 标准，主模式



1. 在该图中，在将 SPE 位置 1 之前使能了 MCK。更多信息，请参见 *MCK 生成*。
2. 请注意，如果将 AFCNTR 位置 1，则在配置阶段，WS 和 CK 信号的电平将由 SPI/I2S 块进行控制

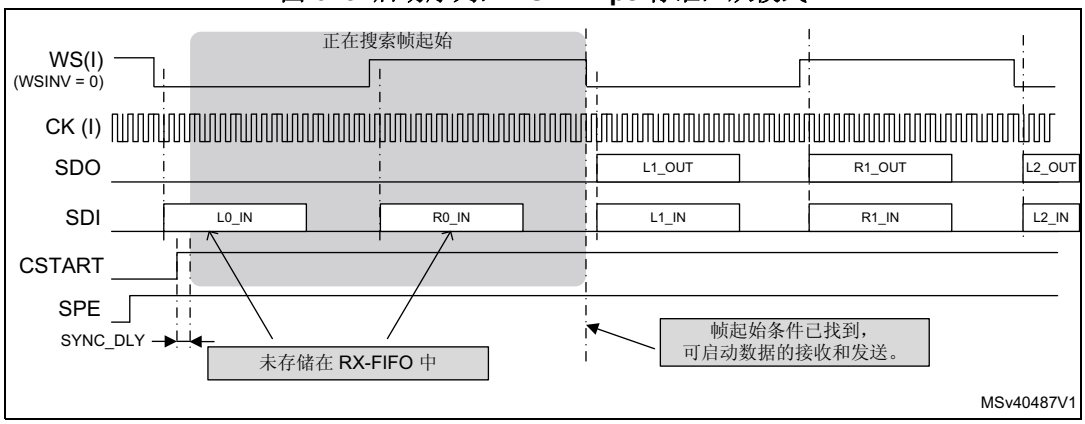
注： *由于时钟域再同步，因此在约 3 个 CK 时钟周期 (SYNC_DLY1) 后硬件会考虑 CSTART 位。*

在从模式下，如果将 CSTART 位置 1，则在满足帧起始条件时将开始数据传输：

- 对于 I2S Philips 标准，帧起始条件为 WS 信号下降沿。将在延迟一个时钟周期后开始发送/接收。
  如果 WSINV = 1，则帧起始条件为上升沿。
- 对于其他协议，帧起始条件为 WS 信号上升沿。对于 MSB 对齐协议，在 WS 上升沿开始发送/接收。对于 PCM 协议，将在延迟一个时钟周期后开始发送/接收。
  如果 WSINV = 1，则帧起始条件为下降沿。

图 618 所示为采用 I2S Philips 标准的从模式的启动序列示例。

图 618. 启动序列，I2S Philips 标准，从模式



注： *由于时钟域再同步，因此在 2 个 CK 时钟周期 (SYNC_DLY) 后硬件会考虑 CSTART 位。*

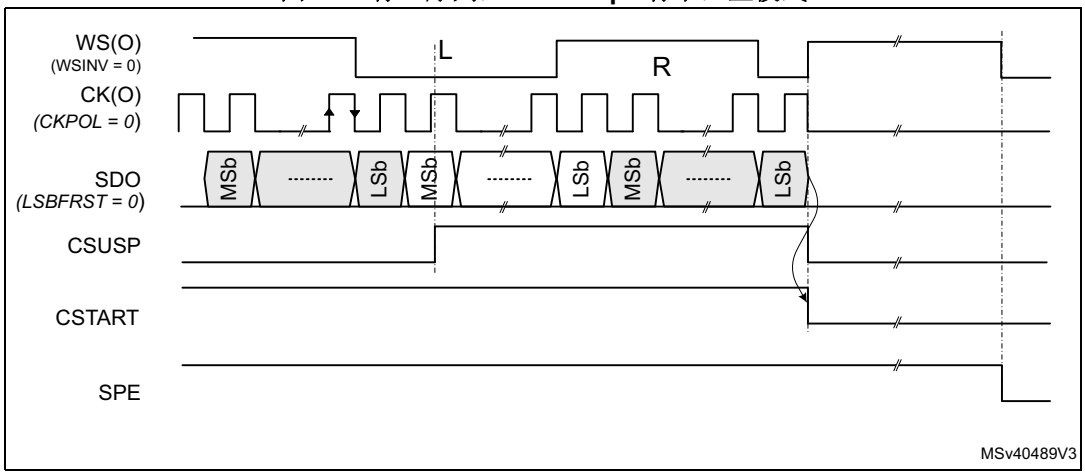
50.9.8 停止序列

应用可通过将 SPE 位置 0 来停止 I2S/PCM 传输。在这种情况下，通信将立即停止，而不会等到当前帧结束。

在主模式下，还可以在当前帧结束时停止 I2S/PCM 传输。为此，用户必须将 CSUSP 位置 1，并轮询 CSTART 位，直到其值为 0。当前立体声（如果选择了 I2S 模式）或单声道采样完全移入或移出时，CSTART 位将变为 0。之后可将 SPE 位设为 0。

图 619 所示为主模式下的停止序列示例。CSUSP 位置 1，在传送左采样期间，传输继续进行，直到传输右采样的最后一位。随后 CSTART 和 CSUSP 返回至 0，CK 和 WS 信号返回到无效状态，用户可以将 SPE 设为 0。

图 619. 停止序列，I2S Philips 标准，主模式



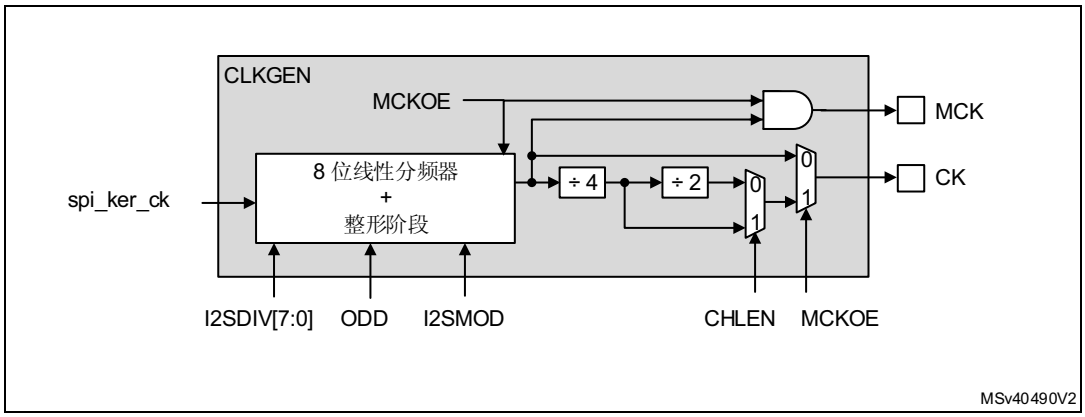
注： *在从模式下，停止序列仅由 SPE 位控制。*

50.9.9 时钟发生器

将 I2S 或 PCM 配置为主模式时，用户需要对时钟发生器进行编程，以便基于所需频率产生帧同步 (WS)、位时钟 (CK) 和主时钟 (MCK)。

如果将 I2S 或 PCM 用于从模式下，则无需配置时钟发生器。

图 620. I^2S 时钟发生器架构



MCK、CK 和 WS 上生成的频率主要取决于 I2SDIV、ODD、CHLEN 和 MCKOE。MCKOE 位指示是否需要生成主时钟。主时钟的频率是帧同步频率的 128 或 256 倍。如果要为外部音频编解码器提供参考时钟，则通常需要该主时钟。

注： *在主模式下，总线时钟频率 (F_{pclk}) 和位时钟频率 (F_{CK}) 之比无特定限制。总线时钟频率必须足够高，这样才能支持数据吞吐量。*

在 I2S 模式下，生成主时钟 (MCKOE = 1) 时，可通过以下公式得出帧同步的频率：

$$F_{WS} = \frac{F_{i2s_clk}}{256 \times \{(2 \times I2SDIV) + ODD\}}$$

在 PCM 模式下通过以下公式得出频率：

$$F_{WS} = \frac{F_{i2s_clk}}{128 \times \{(2 \times I2SDIV) + ODD\}}$$

此外，可通过下列公式得出 MCK 频率 (F_{MCK})：

$$F_{MCK} = \frac{F_{i2s_clk}}{\{(2 \times I2SDIV) + ODD\}}$$

在 I2S 模式下，禁止主时钟 (MCKOE = 0) 时，可通过以下公式得出帧同步的频率：

$$F_{WS} = \frac{F_{i2s_clk}}{32 \times (CHLEN + 1) \times \{(2 \times I2SDIV) + ODD\}}$$

在 PCM 模式下通过以下公式得出频率：

$$F_{WS} = \frac{F_{i2s_clk}}{16 \times (CHLEN + 1) \times \{(2 \times I2SDIV) + ODD\}}$$

其中，F_{WS} 表示帧同步的频率，F_{i2s_clk} 表示提供给 SPI/I2S 块的内核时钟的频率。

注： *CHLEN 和 ODD 既可以为 0 也可以为 1。*
ODD = 0 时，I2SDIV 可以为 0 到 255 之间的任一值；但当 ODD = 1 时，不允许值 I2SDIV = 1。
I2SDIV = 0 时，{(2 x I2SDIV) + ODD} 强制为 1。

注： *{(2 x I2SDIV) + ODD} 为奇数时，MCK 或 CK 信号的占空比不会为 50%。必须谨慎使用奇数比：因为它会影响建立和保持时间的裕量。例如，如果 {(2 x I2SDIV) + ODD} = 5，则占空比可为 40%。*

表 451 所示为适用于 I2S 模式的时钟发生器编程示例。

MCK 生成

无论 SPE 位如何，都可生成主时钟 MCK。MCK 的生成由以下位控制：

- I2SMOD 必须等于 1，
- I2SCFG 必须选择主模式，
- MCKOE 必须设置为 1。

表 451. 适合常见 I2S 频率的 CLKGEN 编程示例

i2s_clk (MHz)	通道长度（位）	I2SDIV	ODD	MCK	采样率：F_{WS} (kHz)
12.288	16	12	0	否	16
12.288	32	6	0		16
12.288	16	6	0		32
12.288	32	3	0		32
49.152	16	16	0		48
49.152	32	8	0		48
49.152	16	8	0		96
49.152	32	4	0		96
49.152	16	4	0		192
49.152	32	2	0		192
4.096	16 或 32	0	-	是	16
24.576	16 或 32	3	0		
49.152	16 或 32	3	0		32
12.288	16 或 32	0	-		48
49.152	16 或 32	2	0		
61.44	16 或 32	2	1		
98.304	16 或 32	2	0		96
196.608	16 或 32	2	0		192

50.9.10 内部 FIFO

I2S 接口可将专用 FIFO 用于 RX 和 TX 路径。可通过 SPI_TXDR 寄存器将要发送的采样写入 TxFIFO。通过 SPI_RXDR 寄存器读取 RxFIFO。

数据对齐和排序

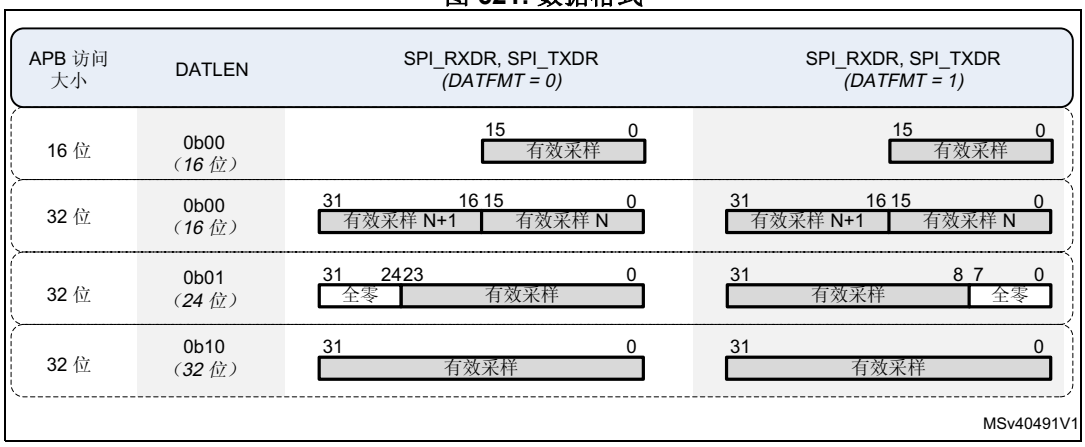
可通过 DATFMT 位选择对齐到 SPI_RXDR 和 SPI_TXDR 寄存器的数据。

另请注意，SPI_RXDR 或 SPI_TXDR 中数据的格式还取决于通过 APB 总线对这些寄存器进行访问的方式。

图 621 显示了 APB 访问大小、DATFMT 和 DATLEN 之间的允许设置。

注： *APB 访问大小为 32 位且 DATLEN = 0 时应格外注意。对于读操作，RxFIFO 必须至少包含两个数据，否则读取的数据将无效。同样，对于写操作，TxFIFO 必须至少有两个空位置，否则数据会丢失。*

图 621. 数据格式



1. 在 I2S 模式下，采样 N 表示左采样，采样 N+1 表示右采样。

1. 24 位和 32 位数据宽度并不总是受支持（DATLEN = 01 或 10）。有关支持的数据大小，请参见*第 50.3 节：SPI 实现*。

可根据可编程的 FIFO 阈值生成中断或 DMA 请求。FIFO 阈值对 RX 通用，TxFIFO 可通过 FTHVL 进行调整。

在 I2S 模式下，左音频采样和右音频采样在 FIFO 中交错。这表示，对于发送操作，用户必须开始先用左采样然后再用右采样来填充 TxFIFO，依此类推。对于接收模式，从 RxFIFO 中读取的第一个数据用于表示左通道，下一个数据用于表示右通道，依此类推。

请注意，SPE 位置 0 时，FIFO 的读指针和写指针将复位。

更多详细信息，请参见*第 50.9.11 节*和*第 50.9.15 节*。

FIFO 大小优化

字节是 FIFO 的基本元素。这样可以优化 FIFO 位置数。例如，数据大小固定为 24 位时，每个音频采样都会占用 3 个基本 FIFO 元素。

举例来说，具有 16 个基本元素的 FIFO 的深度可为：

- 8 个采样，适合 DATLEN = 0 的情况（16 位），
- 5 个采样，适合 DATLEN = 1 的情况（24 位）[a]，
- 4 个采样，适合 DATLEN = 2 的情况（32 位）[a]。

50.9.11 FIFO 状态标志

应用程序可通过两个状态标志来全面监视 I2S 接口的状态。两个标志均可生成一个中断请求。如果使能 RXPIE 位则会生成接收中断，如果使能 TXPIE 位则会生成发送中断。这些位均位于 SPI_IER 寄存器中。

达到了 TxFIFO 阈值 (TXP)

该标志置 1 时，表示 TxFIFO 至少包含 FTHLV 个空位置。这样，要发送的 FTHLV 个新数据就可以写入 SPI_TXDR 中。当空位置数量少于 FTHLV 个时，TXP 标志将复位。请注意，如果禁止 I2S（SPE 位复位），则 TXP = 1。

达到了 RxFIFO 阈值 (RXP)

该标志置 1 时，表示 RxFIFO 中至少有 FTHLV 个有效数据，因此用户可以通过 SPI_RXDR 读取这些数据。RxFIFO 包含的数据数量少于 FTHLV 个时，该标志会复位。

有关 I2S 模式下中断功能的更多信息，请参见*第 50.10 节*。

50.9.12 下溢情况的处理

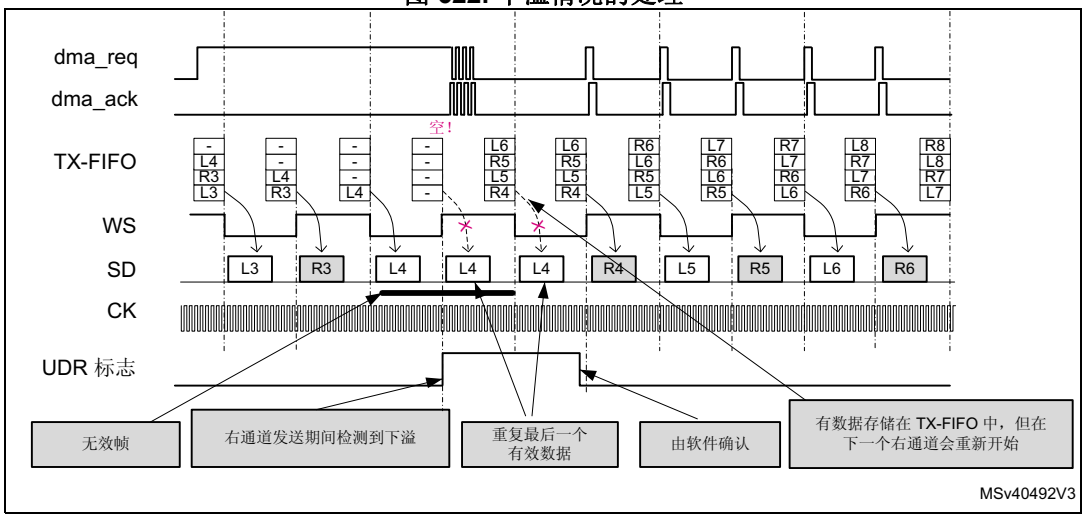
在发送模式下，需要将新数据载入移位寄存器中，而 TxFIFO 已空时，UDR 标志置 1。在这种情况下，至少会丢失一个数据。

在 I2S 模式下，存在一种硬件机制，可防止出现偏离情况（左右通道互换）。如下图所示，发生下溢时，只要不满足重新启动条件，外设就会重复左右通道上的最后一个有效数据。重新开始发送：

- TxFIFO 中存在足够的数据时，
- UDR 标志由软件清零时，
- 满足重新开始条件时：
  - 如果在需要发送右通道数据时发生下溢，则当需要发送右通道时重新开始发送，或者
  - 如果在需要发送左通道数据时发生下溢，则在需要发送左通道时重新开始发送。

a. 24 位和 32 位数据宽度并不总是受支持（DATLEN = 01 或 10）。有关支持的数据大小，请参见*第 50.3 节：SPI 实现*。

图 622. 下溢情况的处理



如果 SPI_IER 寄存器的 UDRIE 位置 1，则 UDR 标志可触发中断。通过向 SPI_IFCR 寄存器的 UDRC 位写入 1 来将 UDR 位清零。

当模块配置为 PCM 模式时，如果 TxFIFO 中有足够的数据，并且 UDR 标志由软件清零，发送将在下一帧开始时重新开始。

注： *主模式或从模式下都可能发生下溢情况。在主模式下，发生下溢时，不会对 WS、CK 和 MCK 信号进行门控。*

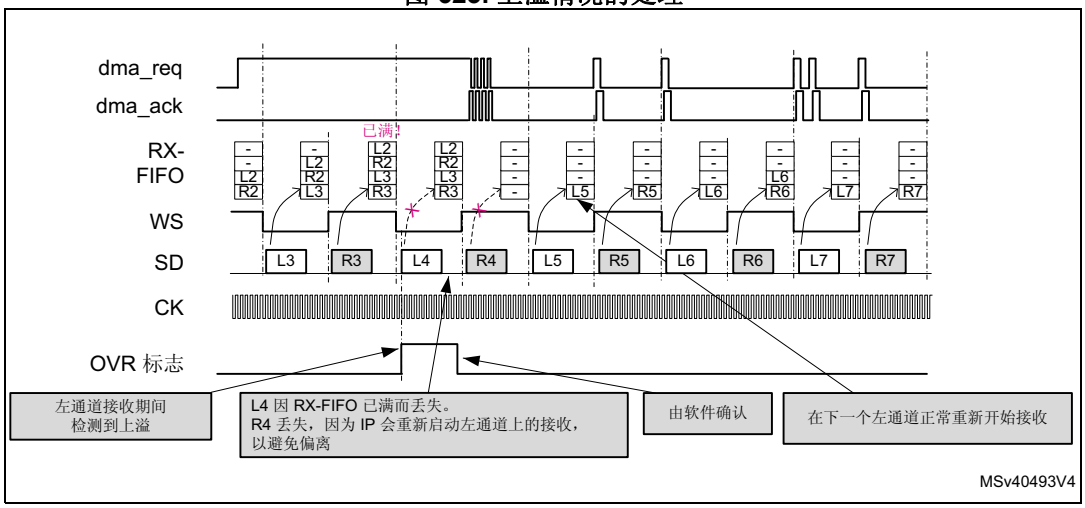
出于再同步的原因，在经过至少 2 个 CK 时钟周期后，硬件将考虑 UDR 标志的变化。

50.9.13 上溢情况的处理

需要将接收到的数据写入 RxFIFO 中，而 RxFIFO 已满时，OVR 标志置 1。因此，某些传入的数据会丢失。

在 I2S 模式下，存在一种硬件机制，可防止出现偏离情况（左右通道互换）。如下图所示，发生上溢时，只要不满足重新启动条件，外设就会停止向 RxFIFO 中写入数据。RxFIFO 中有足够的空间，且 OVR 标志清零时，如果在接收右通道数据时发生上溢，则块会通过向 RxFIFO 中写入下一个右通道开始；或者，如果在接收左通道数据时发生上溢，则块会通过写入下一个左通道开始。

图 623. 上溢情况的处理



如果 SPI_IER 寄存器中的 OVRIE 位置 1，可产生中断。通过向 SPI_IFCR 寄存器的 OVRC 位写入 1 来将 OVR 位清零。

如果该模块配置为 PCM 模式，则发生上溢时，只要不满足重新启动的条件，外设就会停止向 RxFIFO 中写入数据。RxFIFO 中有足够的空间，且 OVR 标志清零时，下一个接收的数据将写入 RxFIFO 中。

注： *主模式或从模式下都可能发生上溢情况。在主模式下，发生上溢时，不会对 WS、CK 和 MCK 信号进行门控。*

50.9.14 帧错误检测

配置为从模式时，SPI/I2S 块会检测以下两种帧错误：

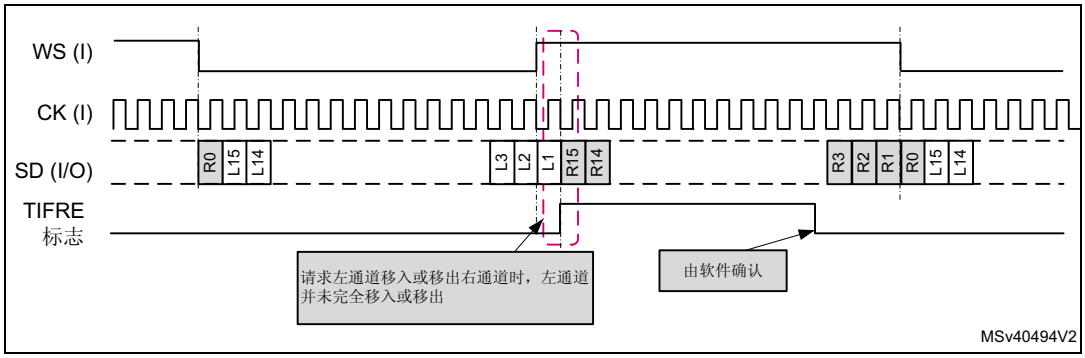
- 在先前数据的移入或移出未完成时接收到帧同步（早期帧错误）。通过 FIXCH = 0 选择此模式。
- 在意外位置发生帧同步。通过 FIXCH = 1 选择此模式。

在从模式下，如果由外部主设备提供的帧长度不是 32 位或 64 位，则用户必须将 FIXCH 设为 0。由于 SPI/I2S 会将每次传输均与 WS 同步，因此不存在偏离风险，但是在噪声环境下，如果 CK 信号出现毛刺，则采样可能会受到影响，而应用不会了解这一情况。

如果外部主设备提供的帧长度等于 32 位或 64 位，则用户可以将 FIXCH 置 1 并相应调整 CHLEN。由于 SPI/I2S 会将每次传输均与 WS 同步，因此仍然不存在偏离风险，并且如果每个通道边界之间的位时钟量不同于 CHLEN，则帧错误标志 (TIFRE) 将置 1。

图 624 所示为帧错误检测示例。SPI/I2S 块处于从模式下，左通道的位时钟周期数不足以用于移入或移出数据。该图显示的是，正在进行的传输被中断，下一次传输已启动，以保持与 WS 信号对齐。

图 624. 帧错误检测，FIXCH = 0

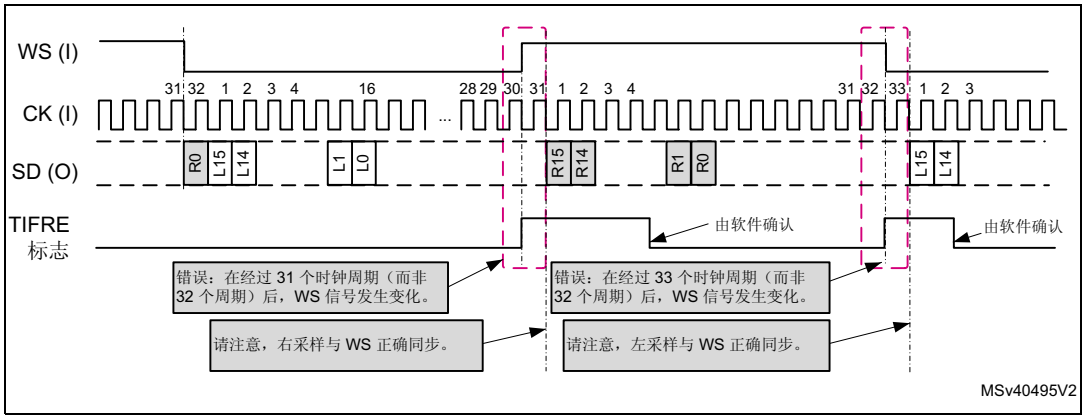


如果 TIFREIE 位置 1，则可生成中断。通过向 SPI_IFCR 寄存器的 TIFREC 位写入 1 来将帧错误标志 (TIFRE) 清零。

可以通过将位 FIXCH 置 1 来扩大帧错误标志的覆盖范围。该位置 1 时，SPI/I2S 可预计在从模式下的固定通道长度。这意味着根据 CHLEN，预期通道长度可以是 16 位或 32 位。如*图 625* 所示，在该模式下，SPI/I2S 模块能够检测 WS 信号是否在预期时间（过早或过晚）发生变化。

注： *图 624* 和*图 625* 所示为适用于从设备发送模式的机制，但该机制还适用于从设备接收和从全双工模式。

图 625. 帧错误检测，FIXCH = 1



通常会因噪声环境干扰了 WS 或 CK 信号的良好接收而进行帧错误检测。

注： *如果在同一帧内出现上溢和早期帧，则 SPI/I2S 无法正常恢复。在这种情况下，用户必须禁止并重新使能 SPI/I2S。*

50.9.15 DMA 接口

I2S/PCM 模式与 SPI 功能共用相同的 DMA 请求线。TX 和 RX 路径有单独的 DMA 通道。每个 DMA 通道均可通过 SPI_CFG1 寄存器的 RXDMAEN 和 TXDMAEN 位使能。

在接收模式下，DMA 接口的工作如下：

1. 硬件评估 RxFIFO 级别。
2. 如果 RxFIFO 至少包含 FTHLV 个采样，则会生成 FTHLV DMA 请求。
   – FTHLV DMA 请求完成时，硬件返回到第 1 步，如此循环往复。
3. 如果 RxFIFO 包含的采样数少于 FTHLV 个，则不会生成 DMA 请求，硬件返回到第 1 步，如此循环往复。

在发送模式下，DMA 接口的工作如下：

1. 硬件评估 TxFIFO 级别。
2. 如果 TxFIFO 至少包含 FTHLV 个空位置，则会生成 FTHLV DMA 请求。
   – FTHLV DMA 请求完成时，硬件返回到第 1 步，如此循环往复。
3. 如果 TxFIFO 包含的空位置数少于 FTHLV 个，则不会生成 DMA 请求，硬件返回到第 1 步，如此循环往复。

50.9.16 编程示例

主模式 I2S Philips 标准，发送

该示例显示如何使用主时钟以 48 kHz 采样率对接口进行编程，以便支持主设备发送模式下的 Philips I2S 标准。假定 SPI/I2S 正在从电路的时钟控制器接收 61.44 MHz 的内核时钟 (i2s_clk)。

启动步骤

1. 使能总线接口时钟（pclk 或 hclk），必要时释放复位信号，以便能够对 SPI/I2S 块进行编程。
2. 确保 SPI/I2S 块正确接收内核频率（在本示例中为 61.44 MHz）。
3. 确保 SPE 设为 0。
4. 对时钟发生器进行编程，以提供 MCK 时钟，以及获得恰好为 48 kHz 的帧同步率。因此，I2SDIV = 2，ODD = 1，且 MCKOE = 1。
5. 编程串行接口协议：CKPOL = 0，WSINV = 0，LSBFRST = 0，CHLEN = 1（每通道 32 位），DATLEN = 1（24 位），I2SSTD = 0（Philips 标准），I2SCFG = 2（主设备发送模式），I2SMOD = 1（针对 I2S/PCM 模式）。在进行后续步骤之前必须更新寄存器 SPI_I2SCFGR。
6. 通过将 FTHLV 设置为所需值来调整 FIFO 阈值。例如，如果需要 2 个音频采样的阈值，则 FTHLV = 1。
7. 清除所有状态标志寄存器。
8. 使能会生成中断的标志，例如 UDRIE。注意，TIFRE 在主模式下没有意义。
9. 如果数据传输使用 DMA，则要：
   a) 编程 DMA 外设
   b) 使用有效的音频采样初始化内存缓冲区
   c) 使能 DMA 通道
10. 如果通过中断进行数据传输，则用户必须通过将 TXPIE 位置 1 来使能中断。

11. 将 SPE 置 1，该位置 1 后，可能会发生以下操作：
    – 如果使能中断生成，SPI/I2S 将生成中断请求，允许中断处理程序填充 TxFIFO。
    – 如果使能 DMA 传输 (TXDMAEN = 1)，SPI/I2S 将生成 DMA 请求来填充 TxFIFO
12. 最后，在使能串行接口之前，用户必须确保 TxFIFO 非空。这对于避免在使能接口时出现下溢条件至关重要。然后可通过将 CSTART 位置 1 来使能 SPI/I2S 块。CSTART 位位于 SPI_CR1 寄存器中。

在主模式下的停止步骤

1. 将 CSUSP 位置 1，以停止正在进行的传输
2. 检查 CSTART 位的值，直至其变为 0
3. 停止 DMA 外设、总线时钟...
4. 将 SPE 位置 0 以禁止 SPI/I2S 块

从模式 I2S Philips 标准，接收

该示例显示如何以 48 kHz 采样率对接口进行编程，以便支持从设备接收器模式下的 I2S Philips 标准协议。请注意，在从模式下，SPI/I2S 块不能控制接收采样的采样率。在本示例中，我们假定外部主设备具有通道长度为 24 位的 I2S 帧结构。因此，FIXCH 置 1 时，我们无法使用为帧错误检测提供的功能。

步骤

1. 通过 RCC 模块使能总线接口和内核时钟，根据需要将复位信号置为有效并加以释放。
2. 对 AFMUX 进行编程以选择所需的 I/O。在当前示例中为 CK、WS、SDI。
3. 编程串行接口协议：CKPOL = 0，WSINV = 0，LSBFRST = 0，FIXCH = 0（因为通道长度不为 16 位和 32 位），DATLExN = 0（16 位），I2SSTD = 0（Philips 协议），I2SCFG = 1（从设备接收模式），I2SMOD = 1（针对 I2S 模式）。
4. 通过将 FTHLV 设置为所需值来调整 FIFO 阈值。例如，如果需要 2 个音频采样的阈值，则 FTHLV = 1。
5. 清除所有状态标志寄存器。
6. 使能会生成中断的标志，例如 UDRIE 和 TIFRE。
7. 如果数据传输使用 DMA，则要：
    – 编程 DMA 外设：一个 RX 通道
    – 使能 DMA 通道
    – 在 SPI/I2S 块中，通过将 RXDMAEN 位置 1 来使能 DMA。
8. 如果通过中断进行数据传输，则用户必须通过将 RXPIE 位置 1 来使能中断。
9. 将 SPE 置 1。
10. 最后，用户可将位 CSTART 置 1 来使能串行接口。SPI/I2S 在下次出现外部主设备发送的左通道数据时，开始将数据存储到 RxFIFO 中。

在从模式下的停止步骤

1. 将 SPE 位置 0 以禁止 SPI/I2S 块
2. 停止 DMA 外设、总线时钟...

50.10 I2S 唤醒和中断

在 PCM/I2S 模式下，可根据*表 452* 中描述的事件生成中断 (**spi_it**) 或唤醒事件信号 (**spi_wkup**)。

中断事件可分别进行使能和禁止。

表 452. I2S 中断请求

中断事件	事件标志	使能控制位	事件清除方法	中断/唤醒是否已激活	
				spi_it	spi_wkup
达到 TxFIFO 阈值	TXP	TXPIE	TxFIFO 包含的空位置数少于 FTHLV 个时，TXP 标志清零	是	是
达到 RxFIFO 阈值	RXP	RXPIE	RxFIFO 包含的采样数少于 FTHLV 个时，RXP 标志清零		
上溢错误	OVR	OVRIE	通过向 OVRC 写入 1 来清零 OVR		
下溢错误	UDR	UDRIE	通过向 UDRC 写入 1 来清零 UDR		
帧错误	TIFRE	TIFREIE	通过向 TIFREC 写入 1 来清零 TIFRE		否

50.11 SPI/I2S 寄存器

50.11.1 SPI/I2S 控制寄存器 1 (SPI_CR1)

SPI/I2S control register 1

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IO LOCK
															rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCRC INI	RCRC INI	CRC33 _17	SSI	HDDIR	CSUSP	C START	MAS RX	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPE
rw	rw	rw	rw	rw	w	rs	rw								rw

位 31:17 保留，必须保持复位值。

位 16 **IOLOCK**：锁定相关 I/O 的 AF 配置 (Locking the AF configuration of associated I/Os)

该位由软件置位，每当 SPE 位从 1 变为 0 时，由硬件清零。

0：AF 配置未锁定

1：AF 配置已锁定

该位置 1 时，不能再修改 SPI_CFG2 寄存器的内容。只有禁止了 SPI 时才可以将该位置 1，否则它会受到写保护。当 MODF 位置 1 时，该位清零，无法置 1。

位 15 **TCRCINI**：发送器 CRC 计算初始化模式控制 (CRC calculation initialization pattern control for transmitter)

0：应用全零模式

1：应用全 1 模式

位 14 **RCRCINI**：接收器 CRC 计算初始化模式控制 (CRC calculation initialization pattern control for receiver)

0：应用全零模式

1：应用全 1 模式

位 13 **CRC33_17**：32 位 CRC 多项式配置 (32-bit CRC polynomial configuration)

0：不使用完整大小（33 位或 17 位）CRC 多项式

1：使用完整大小（33 位或 17 位）CRC 多项式

位 12 **SSI**：内部 SS 信号输入电平 (Internal SS signal input level)

仅当 SSM 位置 1 时，该位才有效。此位的值将作用到外设 SS 输入上，并忽略 SS 引脚的 I/O 值。

位 11 **HDDIR**：半双工模式下的 Rx/Tx 方向 (Rx/Tx direction at Half-duplex mode)

在半双工配置下，HDDIR 位确立了数据传输的 Rx/Tx 方向。在全双工或任何单工配置下，忽略该位。

0：SPI 为接收器

1：SPI 为发送器

位 10 **CSUSP**：主挂起请求 (Master suspend request)

该位读为零。

在主模式下，该位由软件置 1 时，CSTART 位将在当前帧结束时复位，通信将被挂起。用户必须检查 SUSP 标志以检查帧事务的结束。

在禁止 SPI 或进入低功耗模式之前，主模式通信必须挂起（通过该位或将 TXDR 保持为空来实现）。该位也可用于 SPI 或 I2S 模式。

软件挂起后，必须清零 SUSP 标志并在下一个事务开始之前禁止然后重新使能 SPI。

位 9 **CSTART**：主传输开始 (Master transfer start)

该位由软件置 1，用于开始进行 SPI 或 I2S/PCM 通信。在 SPI 模式下，传输结束 (EOT) 标志置 1 或接受事务挂起请求时，该位由硬件清零。在 I2S/PCM 模式下，它也由硬件清零，如停止序列部分所述。

0：主传输处于空闲状态

1：主传输正在进行，或通过自动挂起被临时挂起

在 SPI 模式下，该位仅在主模式下使用。如果使能了发送，则只有当发送 FIFO 中存在数据时，才能启动或继续通信。

位 8 **MASRX**：接收模式下主设备自动挂起 (master automatic SUSP in Receive mode)

该位由软件置 1 和清零，用于控制主接收器模式下的连续 SPI 传输以及自动管理，以避免出现上溢情况。

0：无论是否出现上溢情况，SPI 流/时钟生成都会连续。（数据会丢失）

1：在达到上溢情况之前，SPI 流在 RxFIFO 已满状态下暂时挂起。SPI 通信挂起时，SUSP 标志会置 1。

SPI 通信由硬件自动挂起时，可能会因内部同步延迟而导致同步输出下一帧的几个位。

因此，数据大小低于 8 位时，自动挂起并不十分可靠。在这种情况下，可通过结合在 MIDI 参数保持非零值时所应用的数据帧之间插入的延迟，来实现安全挂起；数据大小和交错 SPI 周期之和必须始终至少产生长度为 8 个 SPI 时钟周期的间隔。软件清零 SUSP 位后，通信将恢复并继续进行后续位传输，不受任何限制。在 SUSP 位清零之前，用户必须根据 RXP 标志指示读出 RxFIFO 中所有可用的数据包，尽可能地释放 RxFIFO 空间，以防止任何后续挂起。

位 7:1 保留，必须保持复位值。

位 0 **SPE**：串行外设使能 (Serial peripheral enable)

该位由软件置 1 和清零。

0：禁止串行外设。

1：使能串行外设。

当 SPE=1 时，将使能 SPI 数据传输，配置寄存器 SPI_CFG1、SPI_CFG2、CRCPOLY 和 UDRDR 以及 SPI_CR1 中的 IOLOCK 位受写保护。只有在 SPE = 0 时，才能更改。

当 SPE=0 时，任何 SPI 操作都会停止并被禁止，所有使能了中断的未清除请求保持挂起状态，并会传播 spi_plck 时钟请求，SS 输出在主设备上停用，内部状态机复位，所有 FIFO 的内容都将被清空，CRC 计算初始化，接收数据寄存器读为零。

MODF 错误标志有效时，SPE 清零且不能置 1。

50.11.2 SPI 控制寄存器 2 (SPI_CR2)

SPI control register 2

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSER[15:0]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIZE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **TSER[15:0]**：处理完存储在 TSIZE 中的先前数据量时就将数据传输扩展量重新载入到 TSIZE (number of data transfer extension to be reload into TSIZE just when a previous number of data stored at TSIZE is transacted)

该寄存器只能在其内容被清空时由软件置 1。在完成 TSIZE 重载后，它由硬件清零。必须在 CTSIZE 计数器达到零之前，提前编程 TSER 值，否则将不会考虑重载，且通信将以正常 EOT 事件终止。

位 15:0 **TSIZE[15:0]**：当前传输的数据量 (Number of data at current transfer)

当这些位由软件更改时，必须禁止 SPI。该位字段也可以由硬件更新，以便通过 TSER 值重新加载（如果适用）。

CSTART 置 1 且 TSIZE 存储零值时，将初始化无限事务。如果使能了 CRC，则不能将 TSIZE 设置为值 0xFFFF。

50.11.3 SPI 配置寄存器 1 (SPI_CFG1)

SPI configuration register 1

偏移地址：0x08

复位值：0x0007 0007

当 SPI 使能时，该寄存器的内容受到写保护，但 TXDMAEN 位和 RXDMAEN 位除外。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	MBR[2:0]			Res.	Res.	Res.	Res.	Res.	CRC EN	Res.	CRCSIZE[4:0]				
	rw	rw	rw						rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX DMA EN	RX DMA EN	Res.	UDRDET[1:0]		UDRCFG[1:0]		FTHLV[3:0]				DSIZE[4:0]				
rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:28 **MBR [2:0]**：主波特率 (master baud rate)
000：SPI 主时钟/2
001：SPI 主时钟/4
010：SPI 主时钟/8
011：SPI 主时钟/16
100：SPI 主时钟/32
101：SPI 主时钟/64
110：SPI 主时钟/128
111：SPI 主时钟/256
注意：在 TI 模式下工作的从设备也考虑 MBR 设置（请参见第 50.5.1 节：TI 模式）。

位 27:23 保留，必须保持复位值。

位 22 **CRCEN**：硬件 CRC 计算使能 (hardware CRC computation enable)
0：禁止 CRC 计算
1：使能 CRC 计算

位 21 保留，必须保持复位值。

位 20:16 **CRCSIZE [4:0]**：要处理和比较的 CRC 帧的长度 (length of CRC frame to be transacted and compared)
处理或比较 CRC 结果时，多项式计算会考虑最高有效位。多项式的长度不受该设置的影响。
00000：保留
00001：保留
00010：保留
00011：4 位
00100：5 位
00101：6 位
00110：7 位
00111：8 位
.....
11101：30 位
11110：31 位
11111: 32 位
该值必须设置与数据大小 (DSIZE[4:0]) 相等或者是数据大小的倍数。其最大大小不能超出实例的 DSIZE 最大值。
注意：如果通过 CRCEN=0 禁止 CRC 计算，则 CRCSIZE 位字段必须保持默认设置。
注意：对于数据大小被限制为 16 位的外设实例，CRCSIZE 位字段的最高有效位将保留。

位 15 **TXDMAEN**：Tx DMA 数据流使能 (Tx DMA stream enable)
0：禁止 Tx DMA
1：使能 Tx DMA

位 14 **RXDMAEN**：Rx DMA 数据流使能 (Rx DMA stream enable)
0：禁止 Rx-DMA
1：使能 Rx-DMA

位 13 保留，必须保持复位值。

位 12:11 **UDRDET [1:0]**：从发送器下溢条件检测 (detection of underrun condition at slave transmitter)
00：数据帧开始时检测到下溢（无第 1 位保护）
01：在最后一个数据帧结束时检测到下溢
10：在有效 SS 信号开始时检测到下溢
11：保留
有关更多详细信息，请参见*图 605：检测到下溢情况时从设备行为的可选配置*。

位 10:9 **UDRCFG [1:0]**：下溢条件时从发送器行为 (behavior of slave transmitter at underrun condition)

00：从设备发送一个常数，该常数由用户在 SPI_UDRDR 寄存器中定义

01：从设备重复上一次从主设备接收到的数据帧

10：从设备重复其上一次发送的数据帧

11：保留

当从设备配置为仅发送模式 (COMM[1:0] = 01) 时，设置为 UDRCFG[1:0]=01 时从设备将重复全零模式。

有关更多详细信息，请参见*图 605：检测到下溢情况时从设备行为的可选配置*。

位 8:5 **FTHLV[3:0]**：FIFO 阈值级别 (FIFO threshold level)

定义单个数据包的数据帧数。数据包的大小不得超过 FIFO 空间的 1/2。

0000：1 个数据

0001：2 个数据

0010：3 个数据

0011：4 个数据

0100：5 个数据

0101：6 个数据

0110：7 个数据

0111：8 个数据

1000：9 个数据

1001：10 个数据

1010：11 个数据

1011：12 个数据

1100：13 个数据

1101：14 个数据

1110：15 个数据

1111：16 个数据

如果配置的数据包大小与数据寄存器访问并行位数对齐，则 SPI 接口的效率更高：

– 如果将 SPI 数据寄存器作为 16 位寄存器进行访问，且 DSIZE ≤ 8 位，则最好选择 FTHLV = 2、4、6。

– 如果将 SPI 数据寄存器作为 32 位寄存器进行访问，且 DSIZE > 8 位,则最好选择 FTHLV = 2、4、6 等，而如果 DSIZE ≤ 8 位，则最好选择 FTHLV = 4、8、12 等。

位 4:0 **DSIZE [4:0]**：单个 SPI 数据帧的位数 (number of bits in at single SPI data frame)

00000：未使用

00001：未使用

00010：未使用

00011：4 位

00100：5 位

00101：6 位

00110：7 位

00111：8 位

.....

11101：30 位

11110：31 位

11111: 32 位

注意：对于数据大小被限制为 16 位的外设实例，DSIZE 位字段的最高有效位将保留。

50.11.4 SPI 配置寄存器 2 (SPI_CFG2)

SPI configuration register 2

偏移地址：0x0C

复位值：0x0000 0000

SPI 使能或 SPI_CR1 寄存器中的 IOLOCK 位置 1 时，该寄存器的内容受到写保护。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AF CNTR	SSOM	SSOE	SSIOP	Res.	SSM	CPOL	CPHA	LSB FRST	MAS TER	SP[2:0]			COMM[1:0]		Res.
rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IOSWP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MIDI[3:0]				MSSI[3:0]			
rw								rw	rw	rw	rw	rw	rw	rw	rw

位 31 **AFCNTR**：复用功能 GPIO 控制 **(alternate function GPIOs control)**

仅当 SPE = 0 时，才考虑该位。

0：外设在被禁止时不对 GPIO 进行控制

1：外设始终控制所有相关 GPIO

出于特定配置原因（例如 CRC 复位、CPHA 或 HDDIR 更改）而必须暂时禁止 SPI 主设备时，将该位置 1 可将配置为复用功能模式的相关输出强制为当前 SPI 配置所对应的状态，以防止出现毛刺。在从模式下不得使用该位，因为任何从设备发送器一旦禁止 SPI，就不得强制其 MISO 输出。

注意：该位也可用于 PCM 和 I2S 模式。

位 30 **SSOM**：主模式 SS 输出管理 (SS output management in master mode)

SSOE 使能时，在主模式下考虑该位。它允许在两个连续数据传输之间配置 SS 输出。

0：SS 保持为有效电平，直到数据传输完成，随即会通过 EOT 标志变为无效状态

1：当 MIDI[3:0] > 1 时，SPI 数据帧与 SS 非有效脉冲交错

位 29 **SSOE**：SS 输出使能 (SS output enable)

该位仅在主模式下使用

0：禁止 SS 输出，SPI 可在多主配置下工作

1：使能 SS 输出。SPI 不能在多主模式环境下工作。在传输完成后或者 SPI 被禁止后，它会根据 SSOM、MIDI、MSSI、SSIOP 位设置将 SS 引脚强制为无效电平

位 28 **SSIOP**：SS 输入/输出极性 (SS input/output polarity)

0：SS 信号低电平有效

1：SS 信号高电平有效

位 27 保留，必须保持复位值。

位 26 **SSM**：SS 信号输入的软件管理 (software management of SS signal input)

0：SS 输入值由 SS PAD 决定

1：SS 输入值由 SSI 位决定

当主设备使用硬件 SS 输出时（SSM = 0 且 SSOE = 1），SS 信号输入在内部强制为非有效状态，以防止出现主模式故障错误。

位 25 **CPOL**：时钟极性 (clock polarity)

0：空闲时 SCK 信号为低电平

1：空闲时 SCK 信号为高电平

位 24 **CPHA**：时钟相位 (clock phase)
0：从第一个时钟边沿开始采样数据
1：从第二个时钟边沿开始采样数据

位 23 **LSBFRST**：数据帧格式 (data frame format)
0：先发送 MSB
1：先发送 LSB
注意：该位也可用于 PCM 和 I2S 模式。

位 22 **MASTER**：SPI 主模式 (SPI master)
0：SPI 从模式
1：SPI 主模式

位 21:19 **SP[2:0]**：串行协议 (Serial protocol)
000：SPI Motorola
001：SPI TI
其他值：保留，不得使用

位 18:17 **COMM[1:0]**：SPI 通信模式 (SPI communication mode)
00：全双工
01：单工发送器
10：单工接收器
11：半双工

位 16 保留，必须保持复位值。

位 15 **IOSWP**：交换 MISO 和 MOSI 引脚的功能 (swap functionality of MISO and MOSI pins)
0：不交换
1：交换 MOSI 和 MISO 引脚
该位置 1 时，MISO 和 MOSI 引脚复用功能互换。
原 MISO 引脚变为 MOSI 引脚，原 MOSI 引脚变为 MISO 引脚。
注意：该位也可用于 PCM 和 I2S 模式。

位 14:8 保留，必须保持复位值。

位 7:4 **MIDI [3:0]**：主模式数据间空闲 (master Inter-Data Idleness)
指定主模式下在两个连续数据帧之间插入的最小时间延迟（以 SPI 时钟周期表示）。
0000：无延迟
0001：1 个时钟周期延迟
...
1111：15 个时钟周期延迟
注意：TI 模式不支持该功能。

位 3:0 **MSSI [3:0]**：主模式 SS 空闲 (master SS idleness)
指定使能 SSOE 时在主模式下额外插入到 SS 有效边沿和第一个会话数据开始之间的额外延迟（用 SPI 时钟周期数表示）。
0000：无额外延迟
0001：增加了 1 个时钟周期的延迟
...
1111：增加了 15 个时钟周期的延迟
注意：TI 模式不支持该功能。

50.11.5 SPI/I2S 中断使能寄存器 (SPI_IER)

SPI/I2S interrupt enable register

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	TSERFIE	MODFIE	TIFREIE	CRCEIE	OVRIE	UDRIE	TXTFIE	EOTIE	DXPIE	TXPIE	RXPIE
					rw	rw	rw	rw	rw	rw	rw	rw	rs	rs	rw

位 31:11 保留，必须保持复位值。

位 10 **TSERFIE**：附加事务数重载中断使能 (additional number of transactions reload interrupt enable)
0：禁止 TSERF 中断
1：使能 TSERF 中断

位 9 **MODFIE**：模式故障中断使能 (mode fault interrupt enable)
0：禁止 MODF 中断
1：使能 MODF 中断

位 8 **TIFREIE**：使能 TIFRE 中断
0：禁止 TIFRE 中断
1：使能 TIFRE 中断

位 7 **CRCEIE**：CRC 错误中断使能 (CRC error interrupt enable)
0：禁止 CRC 中断
1：使能 CRC 中断

位 6 **OVRIE**：使能 OVR 中断
0：禁止 OVR 中断
1：使能 OVR 中断

位 5 **UDRIE**：UDR 中断使能 (UDR interrupt enable)
0：禁止 UDR 中断
1：使能 UDR 中断

位 4 **TXTFIE**：TXTFIE 中断使能 (TXTFIE interrupt enable)
0：禁止 TXTF 中断
1：使能 TXTF 中断

位 3 **EOTIE**：EOT、SUSP 和 TXC 中断使能 (EOT, SUSP and TXC interrupt enable)
0：禁止 EOT/SUSP/TXC 中断
1：使能 EOT/SUSP/TXC 中断

位 2 **DXPIE**：使能 DXP 中断 (DXP interrupt enabled)
DXPIE 由软件置 1，并由 TXTF 标志置 1 事件清零。
0：禁止 DXP 中断
1：使能 DXP 中断 (DXP interrupt enabled)

位 1 **TXPIE**：TXP 中断使能 (TXP Interrupt enable)
TXPIE 由软件置 1，并由 TXTF 标志置 1 事件清零。
0：禁止 TXP 中断
1：使能 TXP 中断

位 0 **RXPIE**：使能 RXP 中断 (RXP Interrupt Enable)
0：禁止 RXP 中断
1：使能 RXP 中断

50.11.6 SPI/I2S 状态寄存器 (SPI_SR)

SPI/I2S status register

偏移地址：0x14

复位值：0x0000 1002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTSIZE[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXWNE	RXPLVL[1:0]		TXC	SUSP	TSERF	MODF	TIFRE	CRCE	OVR	UDR	TXTF	EOT	DXP	TXP	RXP
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 **CTSIZE[15:0]**：当前 TSIZE 会话中剩余的数据帧数 (number of data frames remaining in current TSIZE session)
当总线上有数据传输时候，该值并不十分可靠，在 LP 模式下也如此。

位 15 **RXWNE**：RxFIFO 字非空 (RxFIFO word not empty)
0：数据占用的 RxFIFO 空间少于 4 个字节
1：至少有四个字节的 RxFIFO 空间被数据占用
注意：该位值不依赖于 DSIZE 设置，并随残留数据所占用的 RxFIFO 的相关 RXPLVL[1:0] 信息一起保留。

位 14:13 **RXPLVL[1:0]**：RxFIFO 封装级别
当 RXWNE = 0 且数据大小设置为 16 位时，该值指定 RxFIFO 中保留的剩余数据帧数。
00：RxFIFO 中没有下一个可用的帧
01：1 个帧可用
10：2 个帧可用*
11：3 个帧可用*
(*) 当数据大小设置为 8 位时的可选计数。
当帧大小大于 16 位时，这些位读为 00。因此，如果数据大小设置为 17 位到 24 位，则 RWNE 和 RXPLVL 位均无法检测到在 FIFO 接收到的单个数据帧。这样，用户就必须应用其他方法，如 TSIZE>0 或 FTHLV=0。

位 12 **TXC**：TxFIFO 发送完成 (TxFIFO transmission complete)
该标志的行为取决于 TSIZE 设置。
如果 TSIZE = 0，则 TXC 专门由硬件更改，只要 TxFIFO 变为空就会置 1，总线上无任何活动。
如果 TSIZE <> 0，则没有特定理由监视 TXC，因为它只是复制 EOT 标志值（包括其软件清零）。EOTIE 置 1 时，TXC 会生成中断。
0：当前数据事务仍在进行中、TxFIFO 中存在数据或者正在进行最后一个帧发送。
1：最后一个 TxFIFO 帧发送已完成

位 11 **SUSP**：暂停状态 (suspension status)

在主模式下，当发出 CSUSP 请求时，硬件会设置 SUSP 位。这可能发生在 CSUSP 请求发出后当前帧一完成时，或者在主自动暂停接收模式（SPI_CR1 寄存器中的 MASRX 位被置位）下，接收 FIFO 满的情况下。

当 EOTIE 置 1 时，SUSP 会生成中断。

在通过向 SPI 中断标志清除寄存器 (SPI_IFCR) 中的 SUSPC 位写入 1 来禁用 SPI 之前，必须先清除该位。

0：SPI 未挂起（主模式有效或其他模式）。

1：主模式挂起（当前帧已完成）。

位 10 **TSERF**：已重载要处理的额外 SPI 数据量 (additional number of SPI data to be transacted was reload)

通过向 SPI_IFCR 的 TSERFC 位写入 1 或通过对 TSER[15:0] (SPI_CR2) 寄存器进行写操作来将该位清零

0：不接收

1：接收了额外数据量，当前事务继续进行

位 9 **MODF**：模式故障 (mode fault)

0：无模式故障

1：检测到模式故障

向 SPI_IFCR 的 MODFC 位写入 1 可将该位清零。当模式故障标志位 (MODF) 置位时，SPI 控制寄存器 1 (SPI_CR1) 中的 SPI 使能位 (SPE) 和 I/O 锁定位 (IOLOCK) 会被复位，且对这两位的设置操作将被禁止。

位 8 **TIFRE**：TI 帧格式错误 (TI frame format error)

0：无 TI 帧错误

1：检测到 TI 帧错误

向 SPI_IFCR 的 TIFREC 位写入 1 可将该位清零

位 7 **CRCE**：CRC 错误 (CRC error)

0：无 CRC 错误

1：检测到 CRC 错误

向 SPI_IFCR 的 CRCEC 位写入 1 可将该位清零

位 6 **OVR**：上溢 (Overrun)

0：无上溢

1：检测到上溢

向 SPI_IFCR 的 OVRC 位写入 1 可将该位清零

位 5 **UDR**：下溢 (underrun)

0：无下溢

1：检测到下溢

向 SPI_IFCR 的 UDRC 位写入 1 可将该位清零

注意：在 SPI 模式下，UDR 标志仅适用于从模式。在 I2S/PCM 模式下，该标志（如果可用）适用于主模式和从模式。

位 4 **TXTF**：发送传输已填充 (transmission transfer filled)

0：TxFIFO 上传正在进行或者未启动

1：TxFIFO 上传已完成

某次传输中的所有数据包均由应用软件或 DMA 提交以用于发送时，即，TSIZE 数据量已推入到 TxFIFO 中时，TXTF 将由硬件置 1。

该位可由软件通过向 SPI_IFCR 的 TXTFC 位写入 1 来清零

TXTFIE 位置 1 时 TXTF 标志会触发中断。

TXTF 设置可清除 TXIE 和 DXPIE 屏蔽，因此可在计算禁止 TXP 和 DXP 中断的时间时减轻应用软件的负担。

位 3 **EOT**：传输结束 (end of transfer)

一旦完整传输完成，即当 SPI 上发送和/或接收的数据量为 TSIZE 时，EOT 即会由硬件置 1。EOT 可由软件通过向 SPI_IFCR 的 EOTC 位写入 1 来清零

EOTIE 位置 1 时 EOT 标志会触发中断。

如果在 TXTF 标志置 1 以及 DXPIE 清零之前始终使用 DXP 标志，则可使用 EOT 一次性下载 RxFIFO 中包含的最后数据包。

0：传输正在进行或未启动

1：传输完成

在主模式下，EOT 事件会终止数据事务，并且可选择性处理 SS 输出。应用 CRC 时，EOT 事件将延伸至 CRC 帧事务。

位 2 **DXP**：双工数据包 (duplex packet)

0：TxFIFO 已满且/或 RxFIFO 为空

1：TxFIFO 具有用于写操作的空间，且 RxFIFO 至少包含一个用于读操作的数据包

TXP 和 RXP 标志均置 1 时（无论 SPI 为何模式），DXP 标志将置 1。

位 1 **TXP**：可用的 Tx 数据包空间 (Tx-packet space available)

0：TxFIFO 中没有足够的空间来放置下一个数据包

1：TxFIFO 具有足够的可用位置来容纳 1 个数据包

TXP 标志由硬件进行更改。如果使能 SPI，它将监视 TxFIFO 中当前可用的总体空间。在 TxFIFO 中存储完整的数据包后，必须对其进行检查。

位 0 **RXP**：接收数据包可用 (Rx-packet available)

0：RxFIFO 为空或接收到的数据包不完整

1：RxFIFO 至少包含 1 个数据包

RXP 标志由硬件进行更改。如果使能 SPI，它将监视 RxFIFO 中当前可用的整体数据量。从 RxFIFO 中完整读取一个数据包后，必须对其进行检查。

50.11.7 SPI/I2S 中断/状态标志清零寄存器 (SPI_IFCR)

SPI/I2S interrupt/status flags clear register

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SUSPC	TSERFC	MODFC	TIFREC	CRCEC	OVRC	UDRC	TXTFC	EOTC	Res.	Res.	Res.
				w	w	w	w	w	w	w	w	w			

位 31:12 保留，必须保持复位值。

位 11 **SUSPC**：挂起标志清零 (SUSPend flag clear)

向该位写入 1 可将 SPI_SR 寄存器的 SUSP 标志清零

位 10 **TSERFC**：TSERFC 标志清零 (TSERFC flag clear)

向该位写入 1 可将 SPI_SR 寄存器中的 TSERF 标志清零

注意：*TSERF 还可通过对 TSER[15:0] (SPI_CR2) 寄存器进行写操作来复位*

位 9 **MODFC**：模式故障标志清零 (mode fault flag clear)

向该位写入 1 可将 SPI_SR 寄存器的 MODF 标志清零

位 8 **TIFREC**：TI 帧格式错误标志清零 (TI frame format error flag clear)
向该位写入 1 可将 SPI_SR 寄存器中的 TIFRE 标志清零

位 7 **CRCEC**：CRC 错误标志清零 (CRC error flag clear)
向该位写入 1 可将 SPI_SR 寄存器的 CRCE 标志清零

位 6 **OVRC**：上溢标志清零 (overrun flag clear)
向该位写入 1 可将 SPI_SR 寄存器中的 OVR 标志清零

位 5 **UDRC**：下溢标志清零 (underrun flag clear)
向该位写入 1 可将 SPI_SR 寄存器中的 UDR 标志清零

位 4 **TXTFC**：发送传输填充标志清零 (transmission Transfer Filled flag clear)
向该位写入 1 可将 SPI_SR 寄存器中的 TXTF 标志清零

位 3 **EOTC**：传输结束标志清零 (end of transfer flag clear)
向该位写入 1 可将 SPI_SR 寄存器中的 EOT 标志清零

位 2:0 保留，必须保持复位值。

50.11.8 SPI/I2S 发送数据寄存器 (SPI_TXDR)

SPI/I2S transmit data register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXDR[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXDR[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **TXDR[31:0]**：发送数据寄存器 (transmit data register)
该寄存器用作连接 TxFIFO 的接口。对其进行写入操作可访问 TxFIFO。

注意：数据始终右对齐。写入寄存器时将忽略未使用位，读取寄存器时会将未使用位读为 0。

注意：DR 可按字节进行访问（8 位访问）：在这种情况下，通过单次访问只能写入一个数据字节。

按半字进行访问（16 位访问）：在这种情况下，通过单次访问可写入 2 个数据字节或 1 个半字数据。

逐字进行访问（32 位访问）：在这种情况下，通过单次访问可写入 4 个数据字节或者 2 个半字数据或字数据。

禁止对该寄存器进行小于所配置数据大小的写访问。

50.11.9 SPI/I2S 接收数据寄存器 (SPI_RXDR)

SPI/I2S receive data register

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXDR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RXDR[31:0]**：接收数据寄存器 (receive data register)

该寄存器用作连接 RxFIFO 的接口。读取该寄存器即使对 RxFIFO 进行访问。

注意：数据始终右对齐。读取该寄存器时，未使用的位将读为 0。对该寄存器进行的写操作将被忽略。

注意：DR 可按字节进行访问（8 位访问）：在这种情况下，通过单次访问只能读取一个数据字节

按半字进行访问（16 位访问）：在这种情况下，通过单次访问可读取 2 个数据字节或 1 个半字数据

逐字进行访问（32 位访问）：在这种情况下，通过单次访问可读取 4 个数据字节或者 2 个半字数据或字数据。

禁止对该寄存器进行小于所配置数据大小的读访问。

50.11.10 SPI 多项式寄存器 (SPI_CRCPOLY)

SPI polynomial register

偏移地址：0x40

复位值：0x0000 0107

使能 SPI 时，该寄存器的内容受到写保护。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CRCPOLY[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCPOLY[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CRCPOLY[31:0]**：CRC 多项式寄存器 (CRC polynomial register)

此寄存器包含用于 CRC 计算的多项式。

默认 9 位多项式设置 0x107 对应于 DSIZE 的默认 8 位设置。它与多项式字符串（字符串的最高有效位始终被隐藏）长度固定的一些其他 ST 产品使用的设置 0x07 兼容。

多项式的长度由存储在该寄存器中的值的最高有效位给出。必须将其设置为大于 DSIZE 的值。此外，将 DSIZE 配置为最大 32 位或 16 位大小并使能 CRC 时，还必须通过 CRCPOLY 寄存器将 CRC33_17 位置 1（以使多项式长度大于数据大小）。

对于数据大小被限制为 16 位的外设实例，位 31-16 保留。对这些地址执行 32 位访问时，不受限制。保留的位 31-16 始终读为零，并忽略所有写操作。

50.11.11 SPI 发送器 CRC 寄存器 (SPI_TXCRC)

SPI transmitter CRC register

偏移地址：0x44

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXCRC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TXCRC[31:0]**：发送器的 CRC 寄存器 (CRC register for transmitter)

使能 CRC 计算后，TXCRC[31:0] 位将包含后续发送字节在计算后所得到的 CRC 值。向 SPI_CR1 的 CRCEN 位写入 1 或完整处理数据块时，将初始化 CRC 计算。CRC 通过 SPI_CRCPOLY 寄存器中编程的多项式连续计算。

计算时考虑的位数取决于 SPI_CRCPOLY 寄存器和 SPI_CFG1 寄存器的 CRCSIZE 位设置。

注意：进行通信时读取该寄存器会返回一个错误值。

不适用于 I^2S 模式。

对于数据大小被限制为 16 位的外设实例，位 31-16 保留。对这些地址执行 32 位访问时，不受限制。保留的位 31-16 始终读为零，并忽略所有写操作。

注意：当软件读取该寄存器的内容时，CRCSIZE 位字段的配置不予考虑。在这种情况下，不会对未使用的位应用屏蔽。

50.11.12 SPI 接收器 CRC 寄存器 (SPI_RXCRC)

SPI receiver CRC register

偏移地址：0x48

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXCRC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RXCRC[31:0]**：接收器的 CRC 寄存器 (CRC register for receiver)

使能 CRC 计算后，RXCRC[31:0] 位将包含后续接收字节在计算后所得到的 CRC 值。向 SPI_CR1 的 CRCEN 位写入 1 或完整处理数据块时，将初始化 CRC 计算。CRC 通过 SPI_CRCPOLY 寄存器中编程的多项式连续计算。

计算时考虑的位数取决于 SPI_CRCPOLY 寄存器和 SPI_CFG1 寄存器的 CRCSIZE 位设置。

注意：进行通信时读取该寄存器会返回一个错误值。

不适用于 I^2S 模式。

对于数据大小被限制为 16 位的外设实例，位 31-16 保留。对这些地址执行 32 位访问时，不受限制。保留的位 31-16 始终读为零，并忽略所有写操作。

注意：当软件读取该寄存器的内容时，CRCSIZE 位字段的配置不予考虑。在这种情况下，不会对未使用的位应用屏蔽。

50.11.13 SPI 下溢数据寄存器 (SPI_UDRDR)

SPI underrun data register

偏移地址：0x4C

复位值：0x0000 0000

使能 SPI 时，该寄存器的内容受到写保护。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UDRDR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UDRDR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **UDRDR[31:0]**：从模式下溢时的数据 (data at slave underrun condition)

仅在从模式下和下溢条件下考虑该寄存器。所考虑的位数取决于 SPI_CFG1 寄存器的 DSIZE 位设置。下溢条件的处理取决于 SPI_CFG1 寄存器的 UDRDET 和 UDRCFG 位是否置 1。

对于数据大小被限制为 16 位的外设实例，位 31-16 保留。对这些地址执行 32 位访问时，不受限制。保留的位 31-16 始终读为零，并忽略所有写操作。

50.11.14 SPI/I2S 配置寄存器 (SPI_I2SCFGR)

SPI/I2S configuration register

偏移地址：0x50

复位值：0x0000 0000

注：当 I2S（集成电路内置音频总线）禁用时 (SPE = 0)，必须对该寄存器中所有有文档说明的位进行配置。除了 I2SMOD 位在 SPI 模式下需保持为 0，这些位在 SPI 模式下均不使用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	MCK OE	ODD	I2SDIV[7:0]							
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DAT FMT	WSINV	FIXCH	CKPOL	CHLEN	DATLEN[1:0]		PCM SYNC	Res.	I2SSTD[1:0]		I2SCFG[2:0]			I2S MOD
	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25 **MCKOE**：主时钟输出使能 (master clock output enable)

0：禁止主时钟输出

1：使能主时钟输出

位 24 **ODD**：预分频器的奇数因子 (odd factor for the prescaler)

0：实际分频值为 = I2SDIV *2

1：实际分频值为 = (I2SDIV * 2) + 1

有关详细信息，请参见*第 50.9.9 节：时钟发生器*。

位 23:16 **I2SDIV[7:0]**：I^2S 线性预分频器

当 ODD 也等于 1 时，I2SDIV 可以取除了值 1 之外的任何值。

有关详细信息，请参见*第 50.9.9 节：时钟发生器*。

位 15 保留，必须保持复位值。

位 14 **DATFMT**：数据格式 (data format)

0：SPI_RXDR 或 SPI_TXDR 内的数据右对齐

1：SPI_RXDR 或 SPI_TXDR 内的数据左对齐。

位 13 **WSINV**：字选择反转 (Word select inversion)

该位用于反转 WS 信号的默认极性。

0：在 I2S Philips 标准中，左通道传输在 WS 下降沿后一个 CK 周期开始，右通道传输在 WS 上升沿后一个 CK 周期开始。

在 MSB 或 LSB 对齐模式下，WS 为高电平时传输左通道，WS 为低电平时传输右通道。

在 PCM 模式下，数据传输在 WS 上升沿之后的一个 CK 周期开始。

1：在 I2S Philips 标准中，左通道传输在 WS 上升沿后一个 CK 周期开始，右通道传输在 WS 下降沿后一个 CK 周期开始。

在 MSB 或 LSB 对齐模式下，WS 为低电平时传输左通道，WS 为高电平时传输右通道。

在 PCM 模式下，数据传输在 WS 上升沿之后的一个 CK 周期开始。

位 12 **FIXCH**：从模式下的固定通道长度 (fixed channel length in slave)

0：从模式下的通道长度不为 16 位或 32 位（不取决于 CHLEN）

1：从模式下的通道长度应为 16 位或 32 位（取决于 CHLEN）

位 11 **CKPOL**：串行音频时钟极性 (serial audio clock polarity)

0：由 SPI/I2S 生成的信号（即，SDO 和 WS）在 CK 的下降沿发生更改，由 SPI/I2S 接收的信号（即，SDI 和 WS）在 CK 的上升沿进行读取。

1：由 SPI/I2S 生成的信号（即，SDO 和 WS）在 CK 的上升沿发生更改，由 SPI/I2S 接收的信号（即，SDI 和 WS）在 CK 的下降沿进行读取。

位 10 **CHLEN**：通道长度（每个音频通道的位数）(channel length (number of bits per audio channel))

0：16 位

1：32 位

所选通道长度必须始终大于或等于数据长度 (DATLEN)。

位 9:8 **DATLEN[1:0]**：要传输的数据长度 (data length to be transferred)

00：16 位数据长度
01：24 位数据长度
10：32 位数据长度
11：不允许
所选数据长度不得大于通道长度 (CHLEN)。

注意：并非始终支持 24 位和 32 位数据宽度（DATLEN = 01 或 10），请参见第 50.3 节：SPI 实现确认支持的数据大小。

位 7 **PCMSYNC**：PCM 帧同步 (PCM frame synchronization)

0：短帧同步
1：长帧同步

位 6 保留，必须保持复位值。

位 5:4 **I2SSTD[1:0]**：I²S 标准选择

00：I²S Philips 标准
01：MSB 对齐标准（左对齐）
10：LSB 对齐标准（右对齐）
11：PCM 标准
有关 I²S 标准的详细信息，请参见*第 50.9.5 节：支持的音频协议*

位 3:1 **I2SCFG[2:0]**：I2S 配置模式 (I2S configuration mode)

000：从模式——发送
001：从模式——接收
010：主模式——发送
011：主模式——接收
100：从模式——全双工
101：主模式——全双工
其他，未使用

位 0 **I2SMOD**：I2S 模式选择 (I2S mode selection)

0：选择 SPI 模式
1：选择 I2S/PCM 模式

50.11.15 SPI/I2S 硬件配置寄存器 (SPI_SPI_I2S_HWCFGR)

SPI/I2S hardware configuration register

偏移地址：0x03F0

复位值：0x000X XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSCFG[3:0]			
												r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2SCFG[3:0]				CRCCFG[3:0]				RXFCFG[3:0]				TXFCFG[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:20 保留，必须保持复位值。

位 19:16 **DSCFG[3:0]**：SPI 数据大小配置 (SPI data size configuration)
0000：数据大小可配置为 4 位到 16 位
0001：数据大小可配置为 4 位到 32 位
所有其他组合均保留。

位 15:12 **I2SCFG[3:0]**：I2S 配置 (I2S configuration)
0000：I2S 支持未实现
0001：I2S 支持已实现
所有其他组合均保留。

位 11:8 **CRCCFG[3:0]**：SPI 的 CRC 配置 (CRC configuration for SPI)
0000：CRC 支持未实现
0001：CRC 支持已实现
所有其他组合均保留。

位 7:4 **RXFCFG[3:0]**：RxFIFO 大小 (RxFIFO size)
0002：FIFO 大小为 4 字节
0003：FIFO 大小为 8 字节
0004：FIFO 大小为 16 字节
0005：FIFO 大小为 32 字节
所有其他组合均保留。

位 3:0 **TXFCFG[3:0]**：TxFIFO 大小 (TxFIFO size)
0002：FIFO 大小为 4 字节
0003：FIFO 大小为 8 字节
0004：FIFO 大小为 16 字节
0005：FIFO 大小为 32 字节
所有其他组合均保留。

50.11.16 SPI/I2S 版本寄存器 (SPI_VERR)

SPI/I2S version register

偏移地址：0x03F4

复位值：0x0000 0011

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：IP 的主版本 (major revision of the IP)。

位 3:0 **MINREV[3:0]**：IP 的次版本 (minor revision of the IP)。

50.11.17 SPI/I2S 标识寄存器 (SPI_IPIDR)

SPI/I2S identification register

偏移地址：0x03F8

复位值：0x0013 0022

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：IP 标识 (IP identification)
用于标识 IP 的 MCD 标识码。

50.11.18 SPI/I2S 大小标识寄存器 (SPI_SIDR)

SPI/I2S size identification register

偏移地址：0x03FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识 (size identification)
位[31:8]：固定代码 0xA3C5DD0
位[7:0]：地址解码边界范围
0x01：1 KB
0x02：2 KB
0x04：4 KB
0x08：8 KB
所有其他组合均保留。

50.12 SPI 寄存器映射和复位值

表 453. SPI 寄存器映射和复位值

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**SPI_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOLOCK	TCRCINI	RCRCINI	CRC33_17	SSI	HDDIR	CSUSP	CSTART	MASRX	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPE
	Reset value																0	0	0	0	0	0	0	0	0								0
0x04	**SPI_CR2**	TSER[15:0]																TSIZE[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	**SPI_CFG1**	Res.	MBR[2:0]			Res.	Res.	Res.	Res.	Res.	CRCEN	Res.	CRCSIZE[4:0]					TXDMAEN	RXDMAEN	Res.	UDRDET[1:0]		UDRCFG[1:0]		FTHLV[3:0]				DSIZE[4:0]				
	Reset value		0	0	0						0		1	1	1	1	1	0	0		0	0	0	0	0	0	0	0	0	0	1	1	1
0x0C	**SPI_CFG2**	AFCNTR	SSOM	SSOE	SSIOP	Res.	SSM	CPOL	CPHA	LSBFRST	MASTER	SP[2:0]			COMM[1:0]		Res.	IOSWP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MIDI[3:0]				MSSI[3:0]			
	Reset value	0	0	0	0		0	0	0	0	0	0	0	0	0	0		0								0	0	0	0	0	0	0	0
0x10	**SPI_IER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSERFIE	MODFIE	TIFREIE	CRCEIE	OVRIE	UDRIE	TXTFIE	EOTIE	DXPIE	TXPIE	RXPIE
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x14	**SPI_SR**	CTSIZE[15:0]																RXWNE	RXPLVL[1:0]		TXC	SUSP	TSERF	MODF	TIFRE	CRCE	OVR	UDR	TXTF	EOT	DPXP	TXP	RXP
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
0x18	**SPI_IFCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUSPC	TSERFC	MODFC	TIFREC	CRCEC	OVRC	UDRC	TXTFC	EOTC	Res.	Res.	Res.
	Reset value																					0	0	0	0	0	0	0	0	0			
0x20	**SPI_TXDR**	TXDR[31:16]																TXDR[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24 - 0x2C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x30	**SPI_RXDR**	RXDR[31:16]																RXDR[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34 - 0x3C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x40	**SPI_CRCPOLY**	CRCPOLY[31:16](1)																CRCPOLY[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	
0x44	**SPI_TXCRC**	TXCRC[31:16](1)																TXCRC[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**SPI_RXCRC**	RXCRC[31:16](1)																RXCRC[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	**SPI_UDRDR**	UDRDR[31:16](1)																UDRDR[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 453. SPI 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x50	**SPI_I2SCFGR**	Res.	Res.	Res.	Res.	Res.	Res.	MCKOE	ODD	I2SDIV[7:0]								Res.	DATFMT	WSINV	FIXCH	CKPOL	CHLEN	DATLEN[1:0]		PCMSYNC	Res.	I2SSTD[1:0]		I2SCFG[2:0]			I2SMOD
	Reset value							0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0		0	0	0	0	0	0
0x03F0	**SPI_I2S_ HWCFGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSCFG[3:0]				I2SCFG[3:0]				CRCCFG [3:0]				RXFCFG [3:0]				TXFCFG [3:0]			
	Reset value													0	0	0	x	0	0	0	x	0	0	0	x	0	x	x	x	0	x	x	x
0x03F4	**SPI_VERR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV [3:0]				MINREV [3:0]			
	Reset value																									0	0	0	1	0	0	0	1
0x03F8	**SPI_IPIDR**	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
0x03FC	**SPI_SIDR**	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

1. 对于数据大小被限制为 16 位的外设实例，位 31-16 保留。对这些地址执行 32 位访问时，不受限制。保留的位 31-16 始终读为零，并忽略所有写操作。*请参见寄存器边界地址表。*

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

51 串行音频接口 (SAI)

51.1 简介

SAI 接口（串行音频接口）灵活性高、配置多样，可支持多种音频协议。该接口适用于许多立体声或单声道应用。例如，它可配置为支持 I2S 标准、LSB 或 MSB 对齐、PCM/DSP、TDM 和 AC'97 等协议。将音频模块配置为发送器时，SAI 接口可提供 SPDIF 输出。

SAI 通过两个完全独立的音频子模块来实现这种灵活性和可重配置性。每个模块都有自己的时钟发生器和 I/O 线控制器。

SAI 可以配置为主模式或配置为从模式。音频子模块既可作为接收器，又可作为发送器；既可与另一模块同步，又可以不同步。

SAI 可与其他 SAI 相连接来同步运行。

51.2 SAI 主要特性

- 具有两个独立的音频子模块，子模块既可作为接收器，也可作为发送器，并带有自身的 FIFO。
- 每个音频子模块集成多达 8 个字的 FIFO。
- 两个音频子模块间可以是同步或异步模式。
- 多个 SAI 之间可实现同步。
- 两个音频子模块的主/从配置相互独立。
- 当两个音频子模块都配置为主模式时，每个子模块的时钟发生器产生独立的音频采样频率。
- 数据大小可配置：8 位、10 位、16 位、20 位、24 位或 32 位。
- 音频协议：I2S、LSB 或 MSB 对齐、PCM/DSP、TDM、AC'97
- PDM 接口，支持多达 4 对麦克风
- 如果需要，可提供 SPDIF 输出。
- 高达 16 个大小可配置的 Slot。
- 每帧的位数可配置。
- 帧同步有效电平可配置（偏移、位长、电平）。
- 可配置 Slot 中第一个有效位的位置。
- 支持 LSB 或 MSB 数据传输。
- 支持静音模式。
- 具有立体声/单声道音频帧功能。
- 通信时钟选通边沿可配置 (SCK)。
- 错误标志对应相应中断（分别使能时）。
  - 上溢和下溢检测，
  - 从模式下的帧同步信号提前检测，
  - 从模式下的帧同步信号滞后检测，
  - 接收时编解码器未就绪（AC'97 模式）。

- 支持如下中断源（使能时）：
  - 错误，
  - FIFO 请求。
- DMA 接口有 2 个通道。

51.3 SAI 实现

表 454. STM32MP13xx SAI 特性(1)

SAI 特性	SAI1	SAI2
I2S、LSB 或 MSB 对齐、PCM/DSP、TDM、AC'97	X	X
FIFO 大小	8 字	8 字
SPDIF	X	X
PDM	X (2)	X(2)

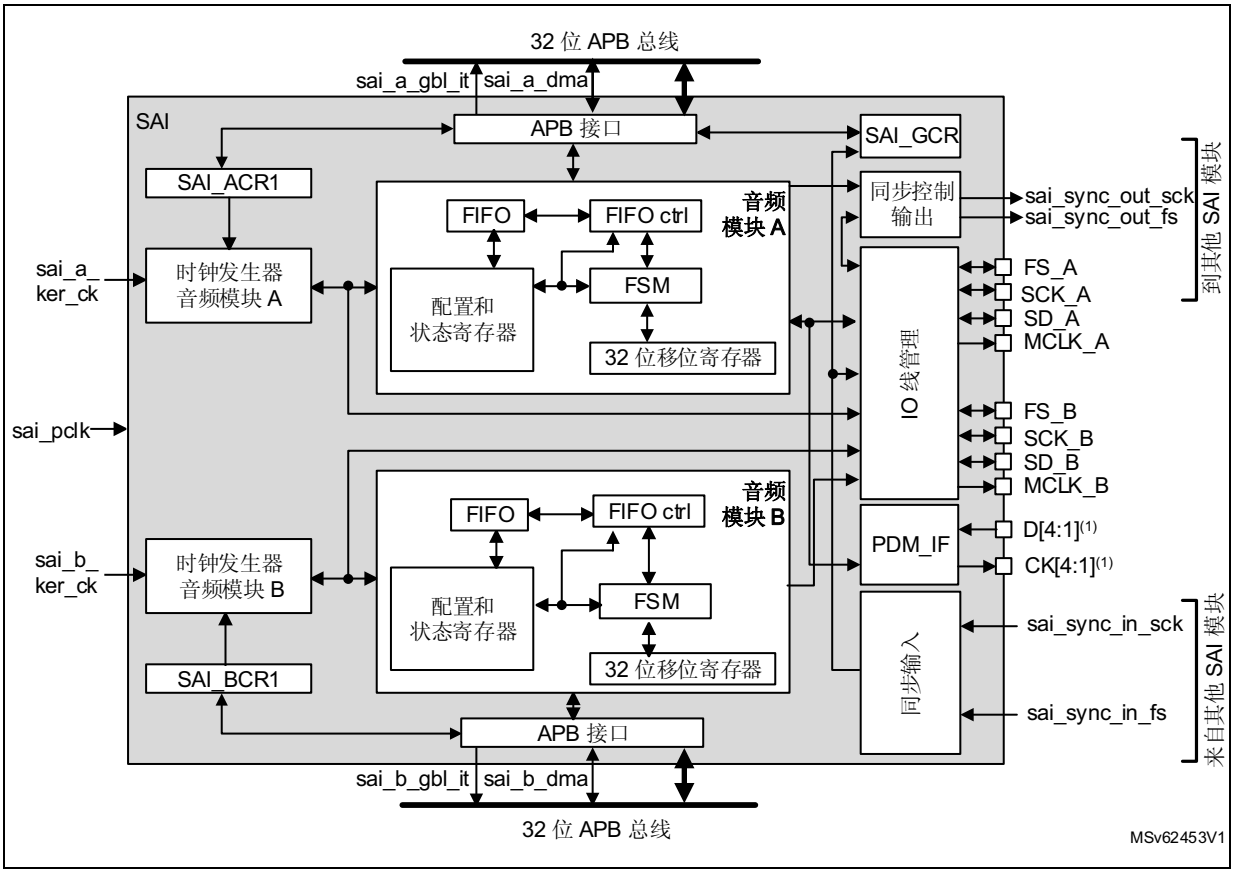
1. “X” = 支持，“-” = 不支持。
2. 只有信号 D[4:1] 和 CK[2:1] 可用。

51.4 SAI 功能描述

51.4.1 SAI 框图

图 626 给出了 SAI 框图，*表 455* 和*表 456* 列出了 SAI 的内部信号和外部信号。

图 626. SAI 功能框图



1. 这些信号可能并非在所有 SAI 实例中均提供。有关详细信息，请参见*第 51.3 节：SAI 实现*。

SAI 主要由两个各自带有时钟发生器的音频子模块组成。每个音频模块集成一个 32 位移位寄存器，该寄存器由模块自身的功能状态机控制。数据存储和读取都是通过专用的 FIFO 来完成。FIFO 可通过 CPU 访问，也可通过 DMA 访问以减轻 CPU 的通信负担。每个音频模块是独立的。这两个音频子模块可彼此同步。

I/O 线控制器管理 SAI 中指定音频模块的一组 4 个专用引脚（SD、SCK、FS、MCLK）。如果将两个子模块声明为同步模块，则其中一些引脚可以共用，从而可留出一些引脚用作通用 I/O。MCLK 引脚是否可用作输出引脚取决于实际应用和解码要求以及音频模块是否配置为主模式。

如果将一个 SAI 配置为与其他 SAI 同步运行，可以释放更多 I/O（引脚 SD_x 除外）。

可配置功能状态机来处理多种音频协议。一些寄存器用于设置所需协议（音频帧波形发生器）。

音频子模块在主模式或从模式下均可用作发送器或接收器。主模式意味着从 SAI 生成 SCK_x 位时钟和帧同步信号，而从模式则意味着 SCK_x 位时钟和帧同步信号来自另一外部或内部主设备。在特殊情况下，FS 信号方向与主模式或从模式定义不直接相关。在 AC'97 协议中，即使 SAI（链接控制器）设置为需要接收 SCK 时钟，FS 信号也会是 SAI 输出（从模式下也是如此）。

注： *为方便阅读此部分，符号 SAI_x 指 SAI_A 或 SAI_B，其中“x”代表 SAI A 子模块或 SAI B 子模块。*

51.4.2 SAI 引脚和内部信号

表 455. SAI 内部输入/输出信号

内部信号名称	信号类型	说明
sai_a_gbl_it/ sai_b_gbl_it	输出	音频模块 A 和 B 全局中断。
sai_a_dma、 sai_b_dma	输入/输出	音频模块 A 和 B DMA 确认与请求。
sai_sync_out_sck、 sai_sync_out_fs	输出	与其他 SAI 模块交换的内部时钟和帧同步输出信号。
sai_sync_in_sck、 sai_sync_in_fs	输入	与其他 SAI 模块交换的内部时钟和帧同步输入信号。
sai_a_ker_ck/ sai_b_ker_ck	输入	音频模块 A/B 内核时钟。
sai_pclk	输入	APB 时钟。

表 456. SAI 输入/输出引脚

名称	信号类型	备注
SAI_SCK_A/B	输入/输出	音频模块 A/B 位时钟。
SAI_MCLK_A/B	输出	音频模块 A/B 主时钟。
SAI_SD_A/B	输入/输出	用于模块 A/B 的数据线。
SAI_FS_A/B	输入/输出	用于音频模块 A/B 的帧同步线。
SAI_CK[4:1]	输出	PDM 比特流时钟(1)。
SAI_D[4:1]	输入	PDM 比特流数据(1)。

1. 这些信号可能并非在所有 SAI 实例中均提供。有关详细信息，请参见*第 51.3 节：SAI 实现*。

51.4.3 SAI 的主要模式

SAI 的每个音频子模块均可通过所选音频模块的 SAI_xCR1 寄存器中的 MODE 位配置为主模式或从模式。

主模式

在主模式下，SAI 会向连接的外部设备提供时钟信号：

- 位时钟和帧同步分别在引脚 SCK_x 和 FS_x 上输出。
- 如果需要，SAI 也可以在 MCLK_x 引脚上生成主时钟。

SCK_x、FS_x 和 MCLK_x 均配置为输出。

从模式

SAI 从外部设备接收时钟信号。

- 如果将 SAI 子模块配置为异步模式，则 SCK_x 引脚和 FS_x 引脚将被配置为输入。
- 如果将 SAI 子模块配置为与另一个 SAI 接口或第二个音频子模块同步运行，则会释放相应的 SCK_x 引脚和 FS_x 引脚以用作通用 I/O。

在从模式下，不使用 MCLK_x 引脚，可将其分配给其他功能。

建议在使能主设备前先使能从设备。

配置和使能 SAI 模式

可通过相应音频模块的 SAI_xCR1 寄存器中的 MODE 位将每个音频子模块独立定义为发送器或接收器。为此，SAI_SD_x 引脚分别被配置为输出或输入。

可使用两个不同的 MCLK 和 SCK 时钟频率配置同一 SAI 中的两个主音频模块。在这种情况下，必须将这两个音频模块配置为异步模式。

SAI 中的每个音频模块均通过 SAI_xCR1 寄存器中的 SAIEN 位使能。在从模式下，该位一经激活，发送器或接收器便会对时钟线、数据线和同步线上的活动敏感。

在主 TX 模式下，即使 FIFO 中没有数据，使能音频模块也会立即为外部从模块产生位时钟，但 FS 信号的产生受 FIFO 中是否存在数据的控制。FIFO 接收到要发送的第一个数据后，此数据将输出到外部从模块。如果 FIFO 中没有要发送的数据，则随后将在音频帧中传送值 0，并会产生一个下溢标志。

在从模式下，音频帧起始于音频模块使能且检测到 SOF 时。

在从 TX 模式下，使能音频模块后的第一个帧上不可能出现下溢事件，因为此时的强制操作顺序如下：

1. 通过软件或 DMA 写入 SAI_xDR。
2. 直到等到 FIFO 阈值 (FLH) 标志不为 0b000（FIFO 空）。
3. 使能音频模块为从发送器模式。

51.4.4 SAI 同步模式

同步模式存在两个同步级别，即音频子模块级别和 SAI 级别。

内部同步

将音频子模块配置为与同一 SAI 中的第二个音频子模块同步运行。在这种情况下，将共用位时钟和帧同步信号，以减少通信时占用外部引脚的数量。配置为同步模式的音频模块将释放其 SCK_x、FS_x 和 MCLK_x 引脚以用作 GPIO，而配置为异步模式的音频模块将使用其 FS_x、SCK_x 和 MCLK_x I/O 引脚（如果该音频模块被配置为主模块）。

通常情况下，处于同步模式的音频模块可用于将串行音频接口 (SAI) 配置为全双工模式。两个音频模块中的一个可配置为主模块，另一个为从模块；也可将两个均配置为从模块；一个模块为异步模块（SAI_xCR1 中的相应位 SYNCEN[1:0] 设置为 00），另一个为同步模块（SAI_xCR1 中的相应位 SYNCEN[1:0] 设置为 01）。

注： *由于存在内部重新同步阶段，因此 PCLK APB 频率必须大于时钟频率位率的两倍。*

外部同步

音频子模块也可以配置为与其他 SAI 同步运行，具体方法如下：

1. 配置为其他 SAI 的同步源的 SAI 必须定义其音频子模块中的哪个子模块应向其他 SAI 提供 FS 和 SCK 信号。此操作通过编程 SYNCOUT[1:0] 位完成。
2. 接收同步信号的 SAI 必须通过 SYNCIN[1:0] 位字段设置合适的值来选择哪个 SAI 提供同步。之后，对于这两个 SAI 音频子模块的每个子模块，用户都必须通过 SYNCEN 位指定其是否与其他 SAI 同步运行。

注： *SYNCIN[1:0] 位和 SYNCOUT[1:0] 位位于 SAI_GCR 寄存器中，SYNCEN 位位于 SAI_xCR1 寄存器中。*

如果指定 SAI 中的两个音频子模块都需要与另一个 SAI 同步，则可以选择以下配置之一：

- 通过 SYNCEN[1:0] 位将每个音频子模块均配置为与另一个 SAI 模块同步。
- 通过 SYNCEN[1:0] 位将一个音频子模块配置为与另一个 SAI 模块同步。随后，通过 SYNCEN[1:0] 位将其他音频子模块配置为与第二个 SAI 音频子模块同步。

下表显示了如何根据所使用的 SAI 模块选择合适的同步信号。例如，SAI2 可通过将 SAI2 SYNCIN 设置为 0 来选择通过 SAI1 进行同步。如果 SAI1 要选择通过 SAI2 进行同步，则必须将 SAI1 SYNCIN 置 1。标注为“保留”的位不得使用。

表 457. 外部同步选择

模块实例	SYNCIN= 3	SYNCIN= 2	SYNCIN= 1	SYNCIN= 0
SAI1	保留	保留	SAI2	保留
SAI2	保留	保留	保留	SAI1

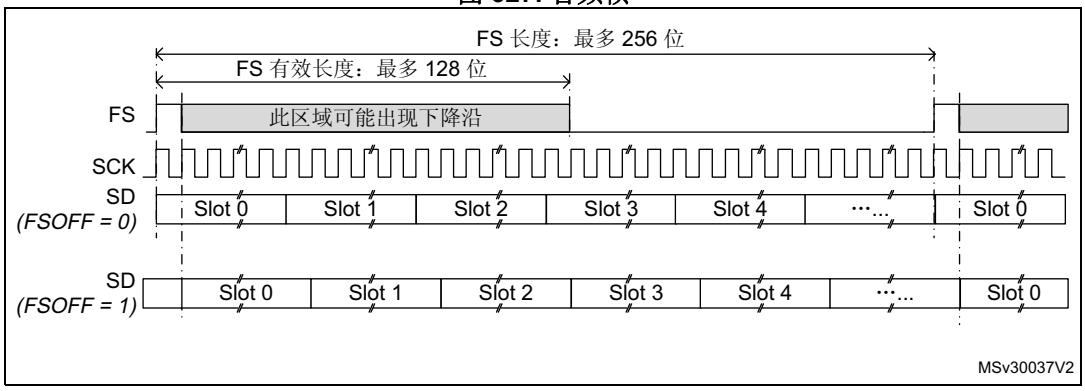
51.4.5 音频数据大小

通过配置 SAI_xCR1 寄存器中的 DS[2:0] 位，配置音频帧的数据大小。数据大小可以是 8 位、10 位、16 位、20 位、24 位或 32 位。在传输期间，将首先发送数据的 MSB 或 LSB，具体取决于 SAI_xCR1 寄存器中的 LSBFIRST 位的配置。

51.4.6 帧同步

FS 信号用作音频帧中的帧同步信号 (SOF)。此信号的波形完全可配置，以在帧同步时，支持各种具有特殊规格的音频协议。这一可重新配置性通过寄存器 SAI_xFRCR 来实现。*图 627* 说明了这种灵活性。

图 627. 音频帧



在 AC'97 模式或 SPDIF 模式下（SAI_xCR1 寄存器中的位 PRTCFG[1:0] = 10 或 PRTCFG[1:0] = 01），帧同步信号的波形被强制配置为支持 AC'97 协议。SAI_xFRCR 寄存器值被忽略。

每个音频模块相互独立，因此均需要特定的配置。

帧长度

- 主模式

  将 SAI_xFRCR 寄存器中的 FRL[7:0] 位字段置 1，可将音频帧的长度配置为最长 256 个位时钟周期。

  如果帧长度大于为该帧声明的 Slot 数，则要发送的剩余位用 0 填充，或者 SD 线释放为高阻态，具体取决于 SAI_xCR2 寄存器中的位 TRIS 的状态（见 *FS 信号的作用*）。在接收模式下，剩余位被忽略。

  如果将位 NODIV 清零，(FRL+1) 必须等于 2 的几次幂（从 8 到 256）以确保音频帧的每个位时钟周期都包含整数个 MCLK 脉冲。

  如果将位 NODIV 置 1，则 (FRL+1) 字段可采用从 8 到 256 的任意值。请参见*第 51.4.8 节：SAI 时钟发生器*。

- 从模式

  音频帧的长度主要用于指定由外部主模块向从模块发送的每个音频帧的位时钟周期数。它主要用于从主模块中检测音频帧传输期间出现的提前或滞后帧同步信号。这种情况下会生成错误。有关详细信息，请参见*第 51.4.14 节：错误标志*。

  在从模式下，SAI_xFRCR 寄存器中的 FRL[7:0] 位的配置不受任何限制。

帧中的位数等于 FRL[7:0] + 1。

音频帧中要传输的最小位数为 8。

帧同步极性

SAI_xFRCR 寄存器中的 FSPOL 位用于设置 FS 引脚的有效极性，通过该极性来启动帧。该帧起始信号为边沿触发。

在从模式下，音频模块等待一个有效帧来启动发送或接收。起始帧与此信号同步。只有通信期间未检测到起始帧并且将其视同预期的起始帧时，帧同步信号才有效（请参见*第 51.4.14 节：错误标志*）。

在主模式下，每次音频帧完成时均会发送帧同步信号，直至 SAI_xCR1 寄存器中的 SAIEN 位清零。如果前一个音频帧结束时 FIFO 中不存在数据，则按照*第 51.4.14 节：错误标志*所述管理下溢条件，但音频通信流不会中断。

帧同步有效电平长度

SAI_xFRCR 寄存器的 FSALL[6:0] 位用于配置帧同步信号的有效电平的长度。该长度可设置为 1 到 128 个位时钟周期。

例如，有效长度在 I2S、LSB 或 MSB 对齐模式下为帧长的一半，在 PCM/DSP 或 TDM 模式下为 1 位。

帧同步偏移

基于应用中支持的音频协议（例如 I2S 标准协议和 MSB 对齐协议），可以在发送音频帧的最后一位或第一位时将帧同步信号置为有效。通过 SAI_xFRCR 寄存器中的 FSOFF 位选择两个配置之一。

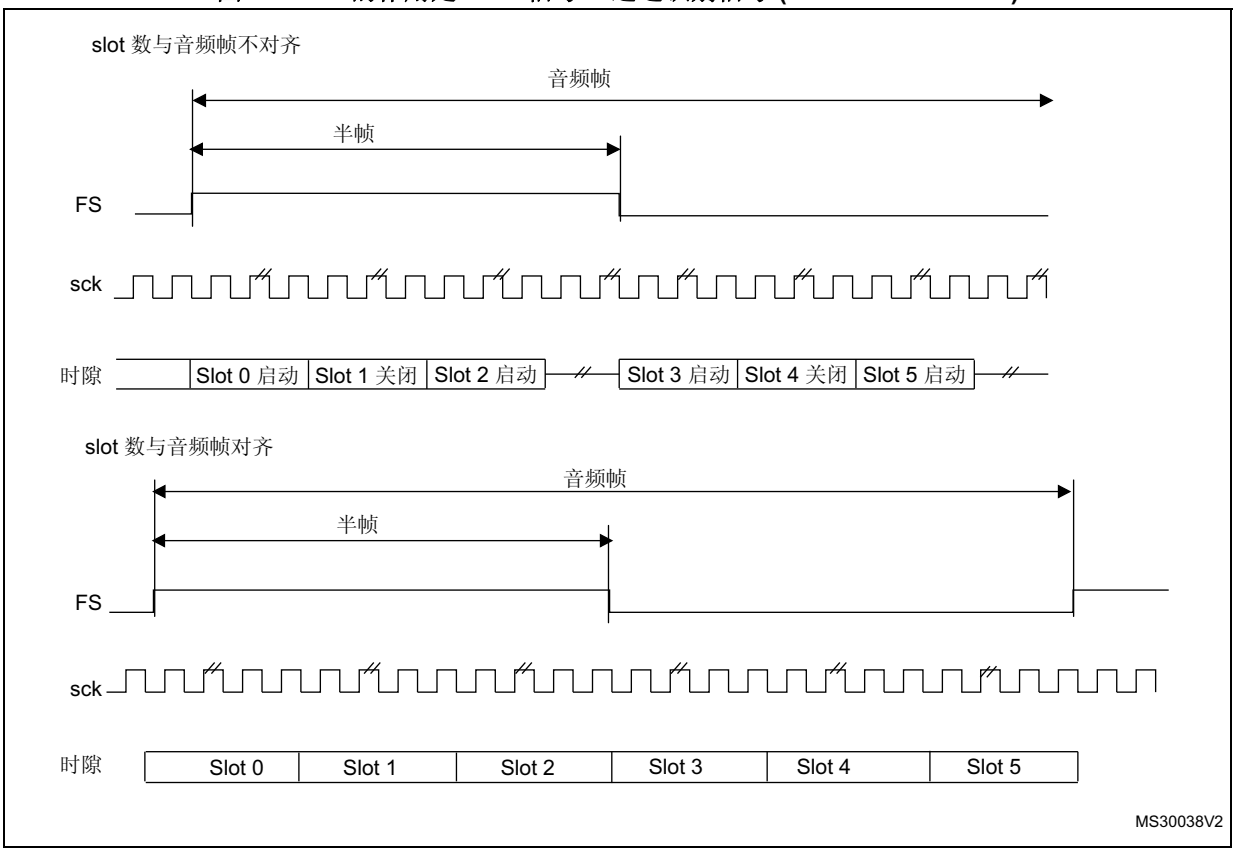
FS 信号的作用

FS 信号可以有不同的含义，具体取决于 FS 的功能。SAI_xFRCR 寄存器中的 FSDEF 位用于选择 FS 信号的含义：

- 0：SOF 信号，例如 PCM/DSP、TDM、AC'97 和音频协议，
- 1：音频帧内的 SOF 信号和通道识别信号，例如 I2S、MSB 或 LSB 对齐协议。

当 FS 信号被视为帧内的 SOF 信号和通道识别信号时，声明的 Slot 数必须是一半用于左通道，一半用于右通道。如果半个音频帧上的位时钟周期数大于某个通道的专用 Slot 数，若 TRIS = 0，则 SAI_xCR2 寄存器中的剩余位时钟周期将发送 0。否则若 TRIS = 1，SD 线将释放为高阻态。接收时，直到通道发生变化，才会考虑剩余位时钟。

图 628. FS 的作用是 SOF 信号 + 通道识别信号 (FSDEF = TRIS = 1)

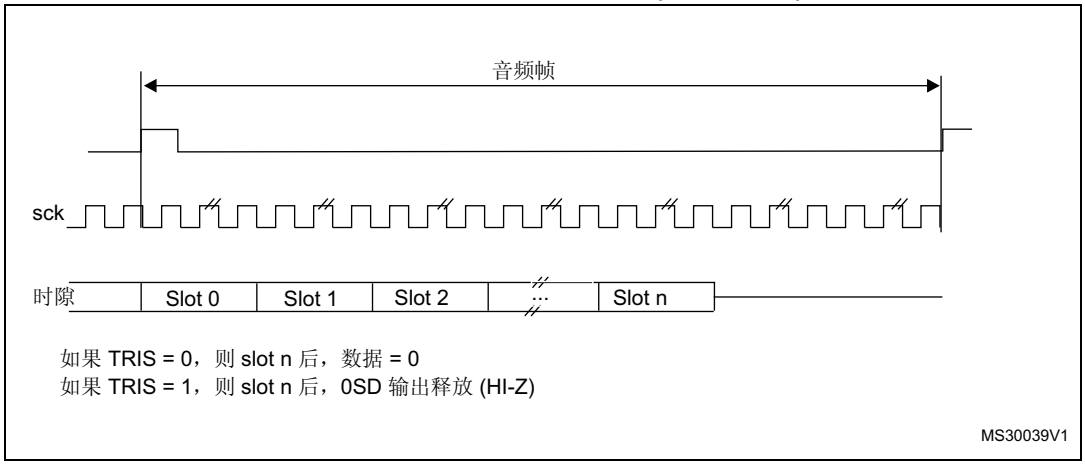


1. 帧长度必须为偶数。

如果 SAI_xFRCR 中的 FSDEF 位保持清零状态，则 FS 信号等效于 SOF 信号，如果 SAI_xSLOTR 中的 NBSLOT[3:0] 位定义的 Slot 数乘以 SAI_xSLOTR 中的 SLOTSZ[1:0] 位配置的 Slot 位数所得的结果小于帧大小（SAI_xFRCR 寄存器中的 FRL[7:0] 位），则：

- 如果 SAI_xCR2 寄存器中的 TRIS = 0，则最后一个 Slot 后的剩余位强制为 0，直至发送器的帧结束，
- 如果 TRIS = 1，则传输这些剩余位时，数据线释放为高阻态。在接收模式下，这些位被丢弃。

图 629. FS 的作用是 SOF 信号 (FSDEF = 0)



在发送器模式下将音频模块配置为获取 SD 线上的 SPDIF 输出时，将不使用 FS 信号。相应的 FS I/O 被释放，留作他用。

51.4.7 Slot 配置

Slot 是音频帧中的基本元素。音频帧中的 Slot 数等于 NBSLOT[3:0] + 1。

每个音频帧的最大 Slot 数固定为 16。

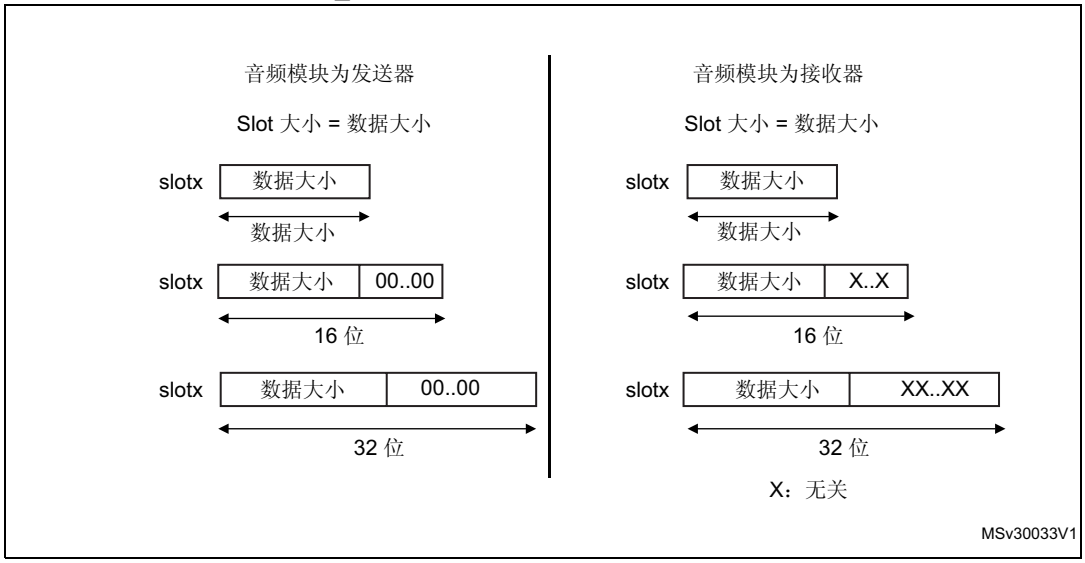
对于 AC'97 协议或 SPDIF 协议（位 PRTCFG[1:0] = 10 或 PRTCFG[1:0] = 01 时），Slot 数自动按照协议规范设置，NBSLOT[3:0] 的值被忽略。

通过设置 SAI_xSLOTR 寄存器的 SLOTEN[15:0] 位，可将各个 Slot 定义为有效 Slot 或无效 Slot。

传输一个无效 Slot 时，SD 数据线将强制为 0 或释放为高阻态，具体取决于 TRIS 位在发送器模式下的配置（请参见*无效 Slot 上的输出数据线管理*）。在接收器模式下，从该 Slot 结束后接收的数据被忽略。结果，不会有 FIFO 访问，也不会有与此无效 Slot 状态相关的 FIFO 读/写请求。

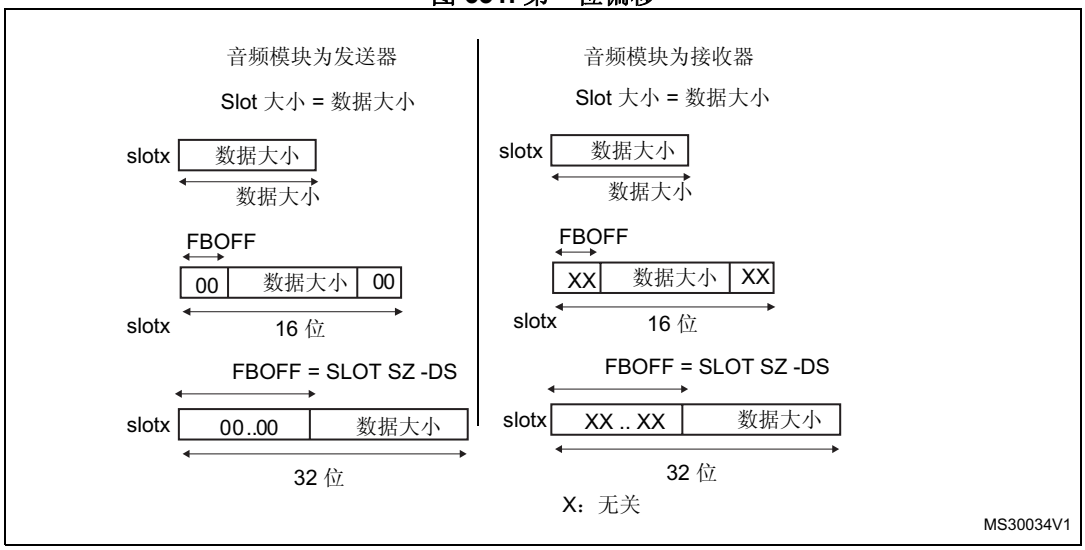
Slot 大小也可配置，如*图 630.* 所示。通过设置 SAI_xSLOTR 寄存器中的 SLOTSZ[1:0] 位字段来选择 Slot 的大小。该大小将应用于音频帧中的所有 Slot。

图 630. SAI_xSLOTR 中的 FBOFF = 0 时的 Slot 大小配置



可以选择 Slot 内要传输的第一个数据位的位置，此偏移通过 SAI_xSLOTR 寄存器中的 FBOFF[4:0] 位配置。在发送器模式下，将从 Slot 开始时注入 0 值，直至到达此偏移位置。接收时，偏移阶段中的位被忽略。此特性适用于 LSB 对齐协议（如果偏移等于 Slot 大小减去数据大小）。

图 631. 第一位偏移



要避免出现错误 SAI 行为，必须遵循以下条件：

FBOFF ≤ (SLOTSZ - DS)，

DS ≤ SLOTSZ，

NBSLOT x SLOTSZ ≤ FRL（帧长度）

SAI_xFRCR 寄存器中的 FSDEF 位置 1 时，Slot 数必须为偶数。

在 AC'97 和 SPDIF 协议（位 PRTCFG[1:0] = 10 或 PRTCFG[1:0] = 01）中，Slot 大小按照 *第 51.4.11 节：AC'97 链路控制器*中的定义自动设置。

51.4.8 SAI 时钟发生器

每个音频模块都具有自己的时钟发生器。时钟发生器通过 sai_x_ker_ck 构建主时钟 (MCLK_x) 和位时钟 (SCK_x) 信号。sai_x_ker_ck 时钟由产品的时钟控制器 (RCC) 提供。

主时钟 (MCLK_x) 的生成

当音频模块定义为主模式或从模式时，时钟发生器提供主时钟 (MCLK_x)。即使相应模块的 SAIEN 位置 0，只要 MCKEN 位置 1，就会生成主时钟。如果 MCLK_x 时钟用作外部音频设备的系统时钟，则此功能非常有用，因为它允许在激活音频流之前生成 MCLK_x。

要在传输音频采样之前在 MCLK_x 输出上生成主时钟，用户应用必须遵循以下顺序：

1. 检查 SAIEN 是否为 0。
2. 将 MCKDIV[5:0] 分频器编程为所需的值。
3. 将 MCKEN 位置 1。
4. 稍后，应用程序可以配置 SAI 的其他部分，并将 SAIEN 位置 1 以开始传输音频采样。

为避免对 MCLK_x 输出上生成的时钟造成干扰，建议不要执行以下操作：

- 当 MCKEN = 1 时更改 MCKDIV
- 当 SAIEN = 1 时将 MCKEN 置 0

当通过 MCKEN 位来打开或者关闭 MCLK_x 时 (SAIEN = 0)，SAI 此外设能保证 MCLK_x 输出没有杂波。

表 458 列出了 MCLK_x 激活条件。

表 458. MCLK_x 激活条件

MCLKEN	NODIV	模块 x 的 SAIEN	MCLK_x
0	X	0	禁止
1			使能
0	1	1	禁止
1			使能
X	0		使能

注： *当 MCLKEN 置 1 时，MCLK_x 也可以在 AC'97 模式下生成。*

位时钟 (SCK_x) 的生成

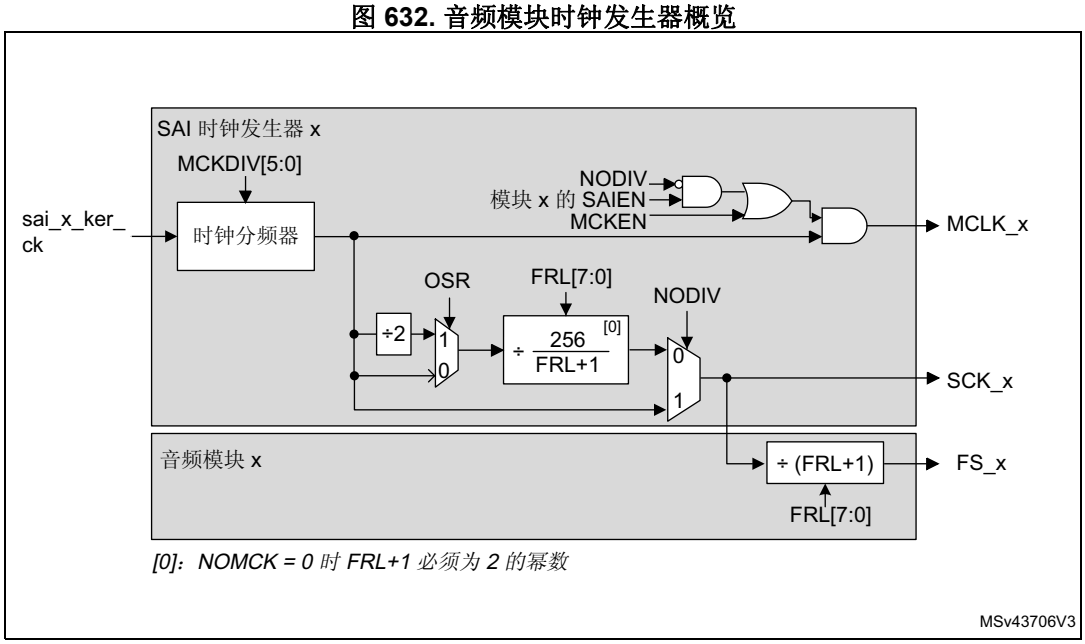
当音频模块定义为主模式时，时钟发生器提供位时钟 (SCK_x)。帧同步 (FS_x) 还可源自时钟发生器提供的信号。

在从模式下，将忽略 NODIV 和 OSR 位字段的值，并且不生成 SCK_x 时钟。

SCK_x 的位时钟选通边沿可以通过 CKSTR 位字段进行配置，该边沿在主模式和从模式下均可正常工作。

图 632 给出了音频模块时钟发生器的架构。

图 632. 音频模块时钟发生器概览



必须使用 NODIV 位强制主时钟 (MCLK_x) 频率和帧同步 (FS_x) 频率之间的比值为 256 或 512。

- 如果 NODIV 置 0，帧同步与主时钟之间的频率比为固定值 512 或 256（根据 OSR 值），但帧长度必须为 2 的幂数。下面进行了详细说明。
- 如果 NODIV 置 1，则应用可以通过 MCKDIV 调整位时钟 (SCK_x) 的频率。此外，只要帧长度大于或等于 8（即 FRL[7:0] > 6），帧长度值就没有限制。帧同步频率取决于 MCKDIV 和帧长度 (FRL[7:0])。在这种情况下，MCLK_x 的频率等于 SCK_x。

NODIV、MCKEN、SAIEN、OVR、CKSTR 和 MCKDIV[5:0] 位属于 SAI_xCR1 寄存器，而 FRL[7:0] 属于 SAI_xFRCR。

NODIV = 0 时的时钟发生器编程

此时，MCLK_x 频率为：

- 如果 OSR = 0，$F_{MCLK_x} = 256 \times F_{FS_x}$
- 如果 OSR = 1，$F_{MCLK_x} = 512 \times F_{FS_x}$

MCKDIV 不为 0 时，MCLK_x 频率由如下公式计算得出：

$$F_{MCLK_x} = \frac{F_{sai_x_ker_ck}}{MCKDIV}$$

帧同步频率由如下公式计算得出：

$$F_{FS_x} = \frac{F_{sai_x_ker_ck}}{MCKDIV \times (OSR + 1) \times 256}$$

位时钟频率 (SCK_x) 由如下公式计算得出：

$$F_{SCK_x} = \frac{F_{sai_x_ker_ck} \times (FRL + 1)}{MCKDIV \times (OSR + 1) \times 256}$$

注： *当 NODIV 等于 0 时，(FRL+1) 必须为 2 的幂数。此外，(FRL+1) 必须介于 8 到 256 之间。(FRL +1) 代表音频帧中的位时钟数。*

MCKDIV 分频比为奇数时，MCLK 的占空比不为 50%。如果 MCKDIV 为奇数、OSR 等于 0 或 (FRL+1) = 2^8，位时钟信号 (SCK_x) 的占空比也可以不是 50%。

建议将 MCKDIV 编程为偶数值或较大的值（大于 10）。

请注意，MCKDIV = 0 时得到的结果与 MCKDIV = 1 时得到的结果相同。

NODIV = 1 时的时钟发生器编程

MCKDIV 不为 0 时，位时钟 (SCK_x) 的频率由如下公式计算得出：

$$F_{SCK_x} = F_{MCLK_x} = \frac{F_{sai_x_ker_ck}}{MCKDIV}$$

帧同步 (FS_x) 的频率由如下公式计算得出：

$$F_{FS_x} = \frac{F_{sai_x_ker_ck}}{(FRL + 1) \times MCKDIV}$$

注： *NODIV 置 1 时，(FRL+1) 可采用 8 到 256 之间的任意值。*

请注意，MCKDIV = 0 时得到的结果与 MCKDIV = 1 时得到的结果相同。

时钟发生器编程示例

表 459 列出了针对 48 kHz、96 kHz 和 192 kHz 频率的编程示例。

表 459. 时钟发生器编程示例

输入 sai_x_ker_ck 时钟频率	MCLK	F_{MCLK}/F_{FS}	FRL (1)	OSR	NODIV	MCKEN	MCKDIV[5:0]	音频采样频率 (F_{FS})
98.304 MHz	Y	512	2^N-1	1	0	1	0 或 1	192 kHz
		512	2^N-1	1	0	1	2	96 kHz
		512	2^N-1	1	0	1	4	48 kHz
		256	2^N-1	0	0	1	2	192 kHz
		256	2^N-1	0	0	1	4	96 kHz
		256	2^N-1	0	0	1	8	48 kHz
	N	-	63	-	1	0	8	192 kHz
		-	63	-	1	0	16	96 kHz
		-	63	-	1	0	32	48 kHz

1. N 表示介于 3 到 8 之间的整数值。

51.4.9 内部 FIFO

SAI 中的每个音频模块都有自己的 FIFO。根据模块是被定义为发送器还是接收器，可相应地读取或写入其 FIFO。因此只存在一个 FIFO 请求与 SAI_xSR 寄存器中的 FREQ 位相关。

如果 SAI_xIM 寄存器中的 FREQIE 位使能，将产生中断。这取决于：

- FIFO 阈值设置（SAI_xCR2 中的 FLVL 位）
- 通信方向（发送器或接收器）。请参见*在发送器模式下产生中断*和*在接收模式下产生中断*。

在发送器模式下产生中断

根据发送器模式下的 FIFO 配置产生中断：

- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 空时（FTH[2:0] 置为 0b000），如果 SAI_xDR 寄存器中没有数据（SAI_xSR 中的 FLVL[2:0] 位小于 001b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当 FIFO 不再为空时（SAI_xSR 中的 FLVL[2:0] 位不是 0b000），即 FIFO 中存储一个或多个数据时，此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之一满时（FTH[2:0] 置为 001b），如果不到四分之一的 FIFO 包含数据（SAI_xSR 中的 FLVL[2:0] 位小于 0b010），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当至少四分之一的 FIFO 包含数据时（SAI_xSR 中的 FLVL[2:0] 位大于等于 0b010），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 半满时（FTH[2:0] 置为 0b010），如果不到一半的 FIFO 包含数据（SAI_xSR 中的 FLVL[2:0] 位小于 011b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当至少一半的 FIFO 包含数据时（SAI_xSR 中的 FLVL[2:0] 位大于或等于 011b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。

- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之三满时（FTH[2:0] 置为 011b），如果不到四分之三的 FIFO 包含数据（SAI_xSR 中的 FLVL[2:0] 位小于 0b100），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当至少四分之三的 FIFO 包含数据时（SAI_xSR 中的 FLVL[2:0] 位大于或等于 0b100），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 满时（FTH[2:0] 置为 0b100），如果 FIFO 不满（SAI_xSR 中的 FLVL[2:0] 位小于 101b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当 FIFO 已满时（SAI_xSR 中的 FLVL[2:0] 位等于值 101b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。

在接收模式下产生中断

根据接收模式下的 FIFO 配置产生中断：

- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 空时（FTH[2:0] 置为 0b000），如果 SAI_xDR 寄存器中至少有一个数据（SAI_xSR 中的 FLVL[2:0] 位大于或等于 001b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当 FIFO 变为空时（SAI_xSR 中的 FLVL[2:0] 位等于 0b000），即 FIFO 中未存储数据时，此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之一满时（FTH[2:0] 置为 001b），如果至少有四分之一的 FIFO 数据单元可用（SAI_xSR 中的 FLVL[2:0] 位大于或等于 0b010），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当不到四分之一的 FIFO 数据单元可用时（SAI_xSR 中的 FLVL[2:0] 位小于 0b010），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 半满时（FTH[2:0] 置为值 0b010），如果至少有一半的 FIFO 数据单元可用（SAI_xSR 中的 FLVL[2:0] 位大于或等于 011b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当不到一半的 FIFO 数据单元可用时（SAI_xSR 中的 FLVL[2:0] 位小于 011b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之三满时（FTH[2:0] 置为值 011b），如果至少有四分之三的 FIFO 数据单元可用（SAI_xSR 中的 FLVL[2:0] 位大于或等于 0b100），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当不到四分之三的 FIFO 数据单元可用时（SAI_xSR 中的 FLVL[2:0] 位小于 0b100），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 满时（FTH[2:0] 置为 0b100），如果 FIFO 已满（SAI_xSR 中的 FLVL[2:0] 位等于 101b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当 FIFO 不满时（SAI_xSR 中的 FLVL[2:0] 位小于 101b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。

与中断的产生类似，如果 SAI_xCR1 寄存器中的 DMAEN 位置 1，则 SAI 可使用 DMA。FREQ 位的有效机制与上述 FREQIE 的中断产生机制相同。

每个 FIFO 均是一个 8 字 FIFO。无论访问大小为何，每次对 FIFO 进行读写时，均针对 1 个字的 FIFO 单元进行操作。每个 FIFO 字包含一个音频 Slot。每次访问 SAI_xDR 寄存器后，FIFO 指针递增一个字。

数据必须以右对齐方式写入 SAI_xDR。

接收到的数据以右对齐方式存储到 SAI_xDR。

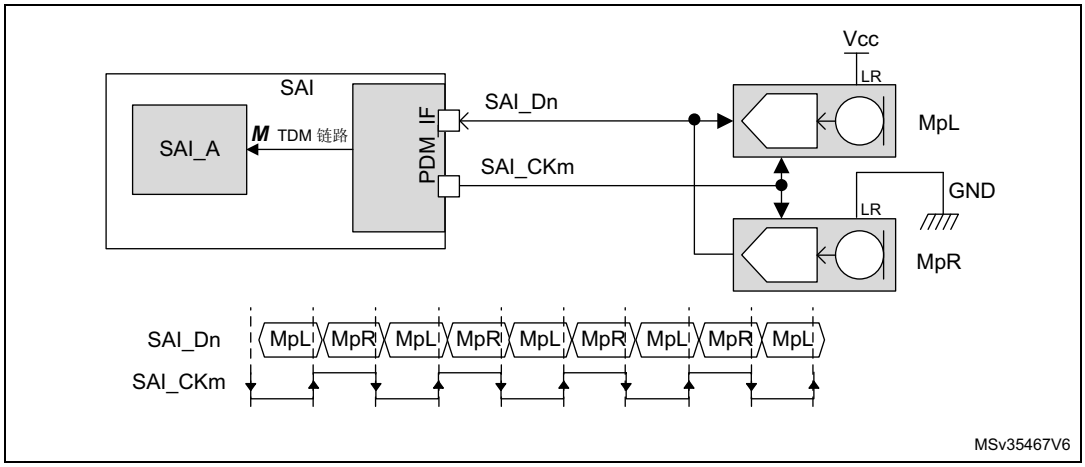
将 SAI_xCR2 寄存器中的 FFLUSH 位置 1 禁止 SAI 后，可重新初始化 FIFO 指针。如果 FFLUSH 在 SAI 使能情况下置 1，则 FIFO 中的数据自动清除。

51.4.10 PDM 接口

PDM（脉冲密度调制）接口旨在支持数字麦克风，最多可并联 4 对数字麦克风。在不同的产品实现中，支持的麦克风数量可能不到 4 对（请参见*第 51.3 节：SAI 实现*）。

图 633 所示为一对数字麦克风通过 PDM 接口实现的典型连接。两个麦克风共用同一比特流时钟和数据线。借助配置引脚 (LR)，其中一个麦克风可在 SAI_CK[m] 上升沿提供有效数据，而另一个麦克风则在 SAI_CK[m] 下降沿提供有效数据（m 表示时钟线的数量）。

图 633. PDM 典型连接和时序



1. **n** 表示数据线的数量，**p** 表示麦克风对的数量。

PDM 功能旨在与配置为 TDM 主模式的 SAI_A 子模块配合使用。它不能与 SAI_B 子模块搭配使用。PDM 接口使用由 SAI_A 的 TDM 接口提供的时序信号，并对其进行调整来生成时钟 (SAI_CK[m])。

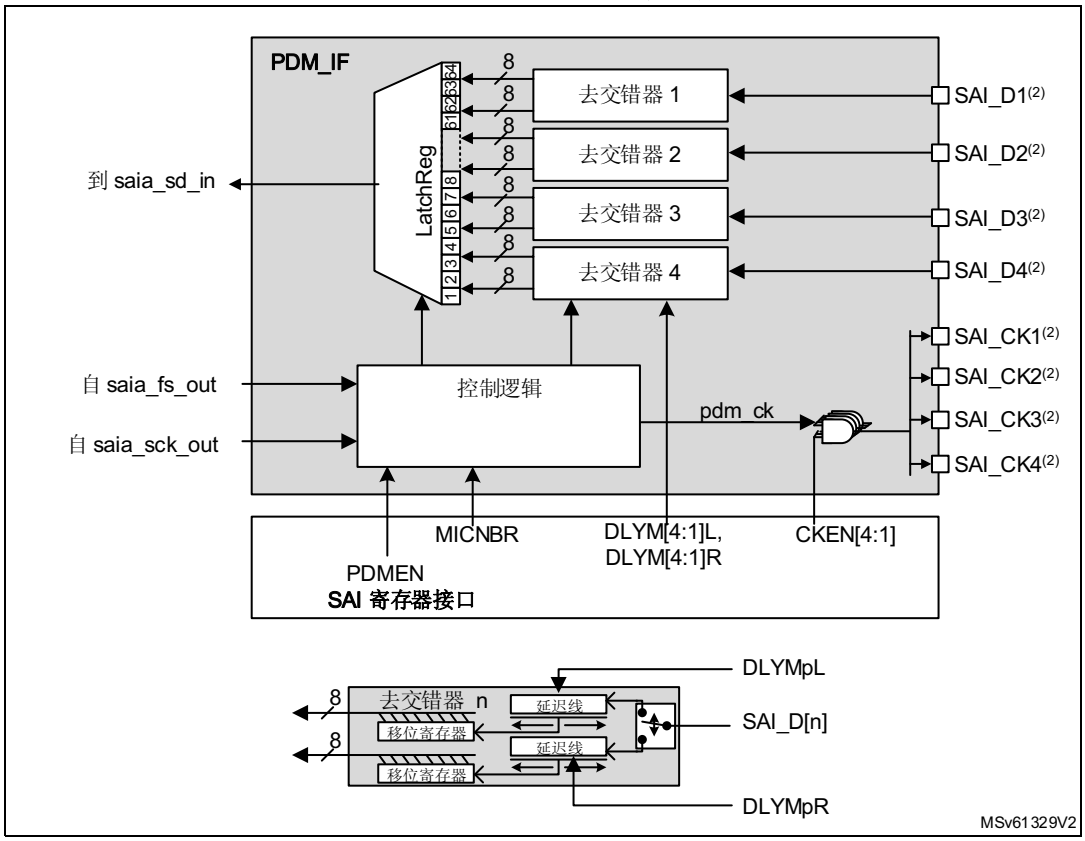
PDM 中的数据处理序列如下：

1. PDM 接口通过从 SAI_A 的 TDM 接口接收的位时钟构建比特流时钟。
2. 接收自麦克风的比特流数据 (SAI_D[n]) 进行去交错，然后经过 7 位延迟线，以根据比特流时钟的精度对每个麦克风的延迟进行微调。
3. 移位寄存器将每个串行比特流转换为字节。
4. 最后通过 TDM 接口的串行数据线将得到的字节移出到 SAI_A。

下面的*图 634* 给出了 PDM 接口框图，其中包括去交错的详细视图。

注： *PDM 接口未嵌入抽取滤波器，因此无法根据比特流构建 PCM 音频采样，需要借助应用软件来执行这一操作。*

图 634. 详细的 PDM 接口模块框图



1. **n** 表示数据线的数量，**p** 表示麦克风对的数量。
2. 这些信号可能并非在所有 SAI 实例中均提供。有关详细信息，请参见*第 51.3 节：SAI 实现*。

可通过 SAI_PDMCR 寄存器中的 PDMEN 位使能 PDM 接口。不过，必须在使能 SAI_A 模块之前使能 PDM 接口。

为减少内存占用量，用户可根据应用需求选择麦克风的数量。这可以通过 MICNBR[1:0] 位来实现。用户可以选择 2、4、6 或 8 个麦克风。例如，如果应用需要使用 3 个麦克风，则用户必须选择 4 个麦克风。

使能 PDM 接口

要使能 PDM 接口，需遵循以下序列：

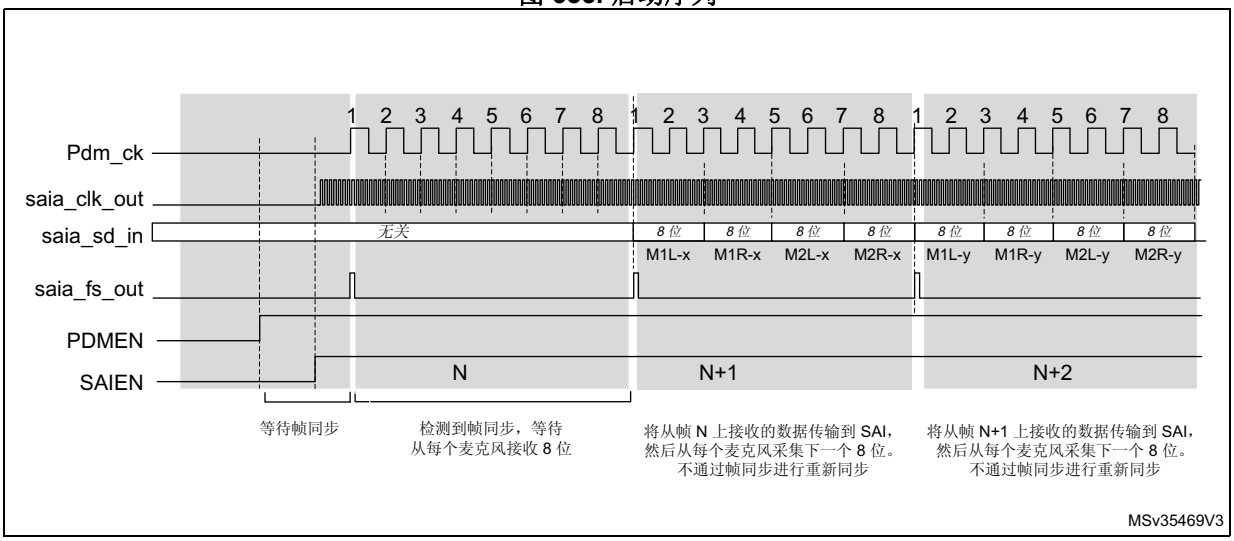
1. 将 SAI_A 配置为 TDM 主模式（请参见*表 460*）。
2. 按以下步骤配置 PDM 接口：
   a) 通过 MICNBR 定义数字麦克风的数量。
   b) 通过将 CKEN 的相应位置 1 来使能应用所需的比特流时钟。
3. 通过 PDMEN 位使能 PDM 接口。
4. 使能 SAI_A。

注： *使能 PDM 接口和 SAI_A 后，在 SAI_ADR 上接收到的前 2 个 TDMA 帧将是无效帧，必须将其丢弃。*

启动序列

图 635 显示了启动序列：PDM 接口使能后，会等待帧同步事件，然后再开始采集麦克风采样。经过 8 个 SAI_CK 时钟周期之后，来自每个麦克风的数据字节将变为可用，并会通过 TDM 接口传输到 SAI。

图 635. 启动序列



SAI_ADR 数据格式

在 SAI_ADR 寄存器中，将根据以下参数对来自麦克风的数据进行排列：

- 麦克风的数量
- 所选 Slot 宽度
- 左对齐第一位

Slot 宽度定义了 SAI_ADR 寄存器中每个字的有效位数。

如果选择 32 位的 Slot 宽度，则 SAI_ADR 中的每个数据包含 32 个有用位。这样可减少存储到内存中的字数。不过，软件必须相应地执行某些操作来对每个麦克风的数据进行去交错。

另一方面，如果将 Slot 宽度设为 8 位，则 SAI_ADR 中的每个数据包含 8 个有用位。这样会增加存储到内存中的字数。不过，这样有助于避免额外的处理操作，因为每个字包含一个麦克风的信息。

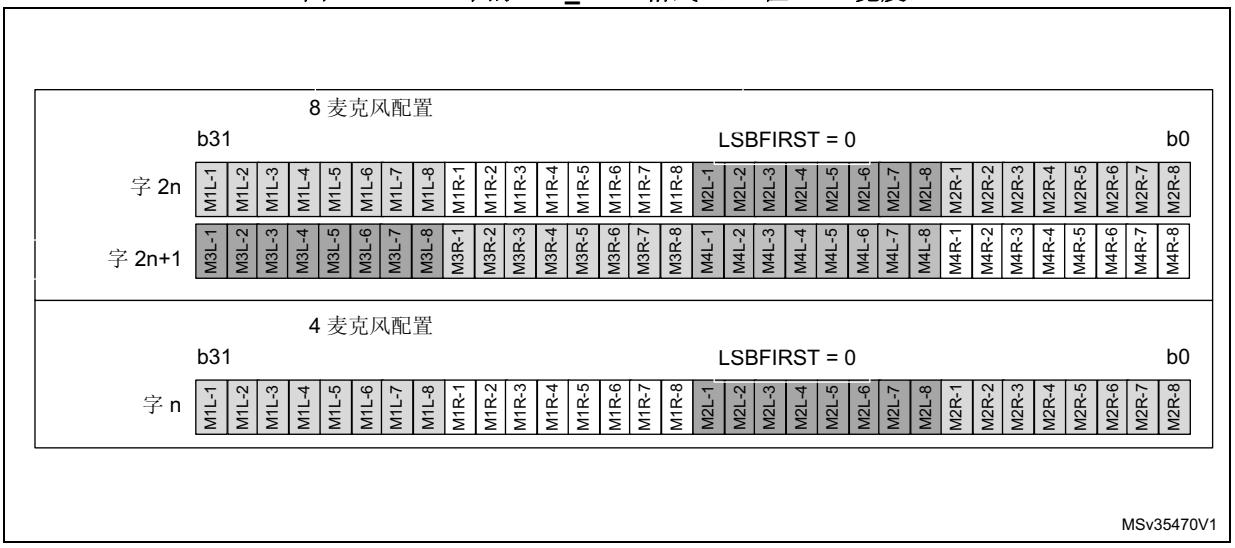
SAI_ADR 数据格式示例

- **32 位 Slot 宽度**（DS = 0b111 且 SLOTSZ = 0）。请参见*图 636*。

  对于 8 麦克风配置，连续对 SAI_ADR 寄存器进行两次字读操作可获取所有麦克风的数据字节。

  对于 4 麦克风配置，每次对 SAI_ADR 寄存器进行字读操作均可获取所有麦克风的数据字节。

图 636. TDM 中的 SAI_ADR 格式（32 位 Slot 宽度）

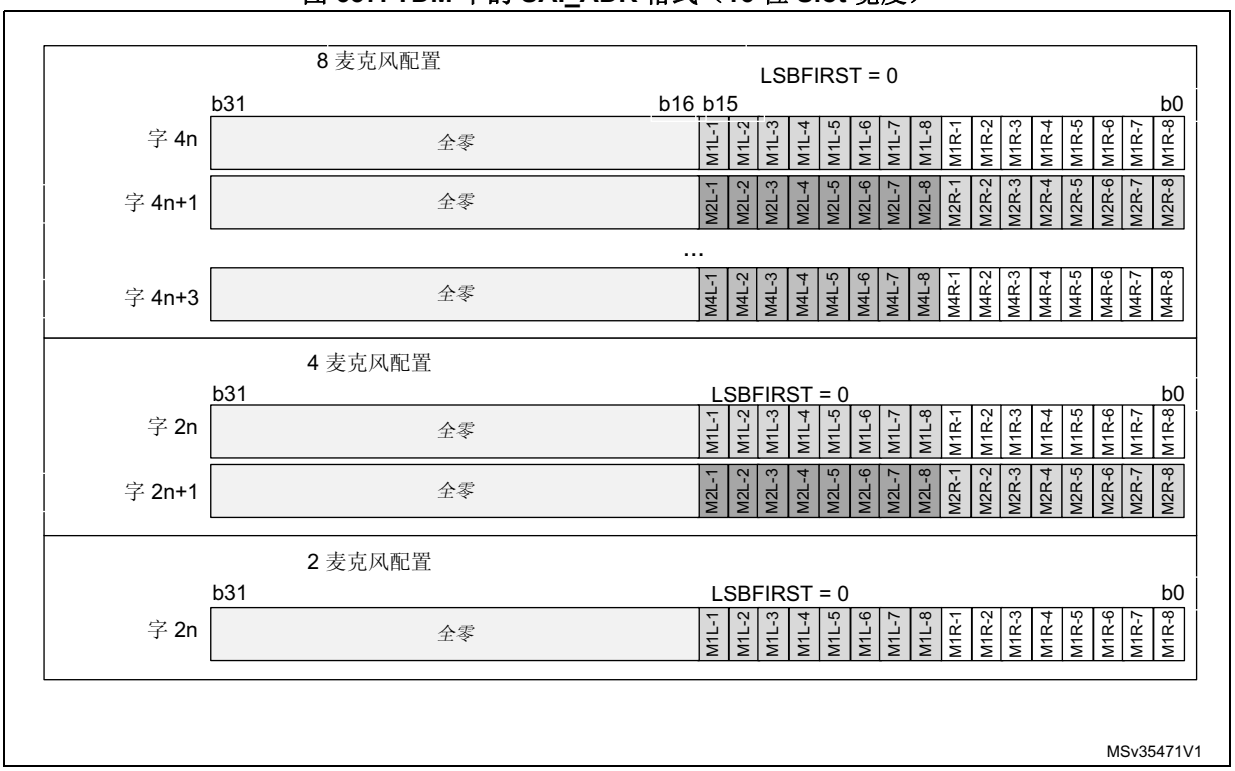


- **16 位 Slot 宽度**（DS = 0b100 且 SLOTSZ = 0）。请参见*图 637*。

  对于 8 麦克风配置，连续对 SAI_ADR 寄存器进行四次字读操作可获取所有麦克风的数据字节。请注意，SAI_ADR 的 16 位数据为右对齐。

  对于 4 麦克风配置或 2 麦克风配置，SAI 行为类似于 8 麦克风配置。获取 4 个麦克风的数据字节需要多达 2 个 16 位字；而获取 2 个麦克风的数据字节需要 1 个 16 位字。

图 637. TDM 中的 SAI_ADR 格式（16 位 Slot 宽度）

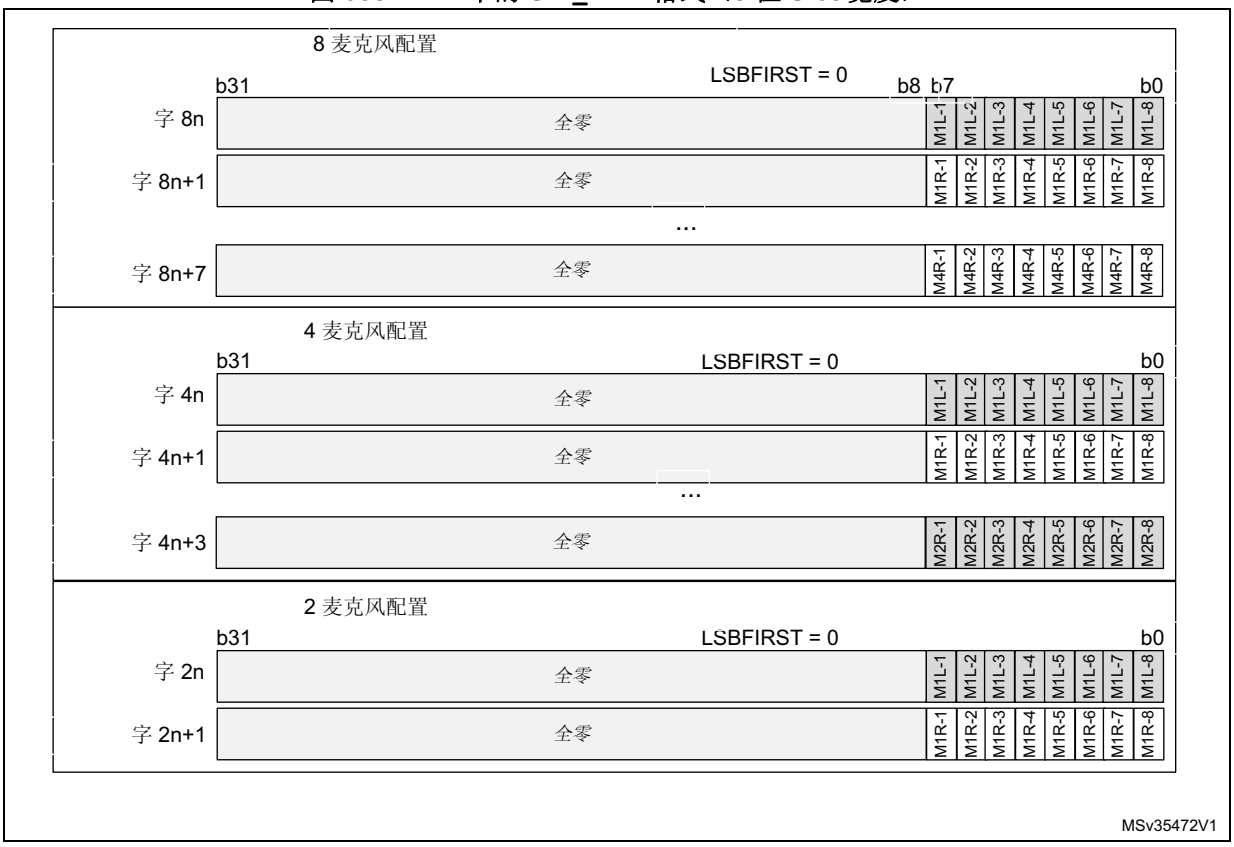


- **使用 8 位 Slot 宽度**（DS = 0b010 且 SLOTSZ = 0）。请参见*图 638*。

  对于 8 麦克风配置，连续对 SAI_ADR 寄存器进行 8 次字读操作可获取所有麦克风的数据字节。请注意，SAI_ADR 的 8 位数据为右对齐。

  对于 4 麦克风配置或 2 麦克风配置，SAI 行为类似于 8 麦克风配置。获取 4 个麦克风的数据字节需要多达 4 个 8 位字；而获取 2 个麦克风的数据字节需要 2 个 8 位字。

图 638. TDM 中的 SAI_ADR 格式（8 位 Slot 宽度）



PDM 接口的 TDM 配置

SAI_A TDM 接口从内部连接到 PDM 接口，以获取麦克风采样。用户应用程序必须按照*表 460* 所示配置 PDM 接口，以确保与 PDM 接口的连接良好。

表 460. TDM 设置

位字段	值	备注
MODE	0b01	必须为主接收模式
PRTCFG	0b00	TDM 的自由协议
DS	X	根据所需数据格式，通过帧长度和 Slot 数（FRL 和 NBSLOT）进行调整。请参见*表 461*。
LSBFIRST	X	可根据所需数据格式使用该参数
CKSTR	0	信号在 SCK_A 位时钟的上升沿发生跳变，在位时钟的下降沿达到稳定状态。
MONO	0	立体声模式

表 460. TDM 设置（续）

位字段	值	备注
FRL	X	根据麦克风数量 (MICNBR) 进行调整。请参见*表 461*。
FSALL	0	脉宽为一个位时钟周期
FSDEF	0	FS 信号为帧起始信号
FSPOL	1	FS 为高电平有效
FSOFF	0	在 Slot 0 的第一位上使能 FS
FBOFF	0	Slot 上无偏移
SLOTSZ	0	Slot 大小 = 数据大小
NBSLOT	X	根据所需数据格式，通过 Slot 大小和帧长度（FRL 和 DS）进行调整。请参见*表 461*。
SLOTEN	X	根据 NBSLOT 进行调整
NODIV	1	无需生成主时钟 MCLK
MCKDIV	X	取决于提供给 sai_a_ker_ck 输入的频率。 必须对该参数进行调整，以生成合适的比特流时钟频率。请参见*表 461*。

调整比特流时钟频率

为正确编程 SAI TDM 接口，用户应用程序必须考虑*表 460* 中给出的设置，并遵循以下顺序：

1. 使用以下公式，根据 PDM 比特流时钟所需的频率调整位时钟频率 (F_{SCK_A})：

$$F_{SCK_A} = F_{PDM_CK} \times (MICNBR + 1) \times 2$$

MICNBR 可以是 0、1、2 或 3（0 = 2 个麦克风，请参见*第 51.6.18 节*）

2. 使用以下公式设置帧长度 (FRL)

$$FRL = (16 \times (MICNBR + 1)) - 1$$

3. 将 Slot 大小 (DS) 配置为 (FRL+1) 的倍数。

表 461. TDM 帧配置示例(1)(2)

麦克风采样率	麦克风数量	所需 SAI_CKn 频率 (3)	位时钟 (SCK_A) 频率	帧同步 (FS_A) 频率	FRL	DS	NBSLOT	备注
48 kHz	最多 8 个	3.072 MHz	24.576 MHz	384 kHz	63	0b111	1	每个帧 2 个 32 位 Slot
		3.072 MHz	24.576 MHz	384 kHz	63	0b100	3	每个帧 4 个 16 位 Slot
		3.072 MHz	24.576 MHz	384 kHz	63	0b010	7	每个帧 8 个 8 位 Slot
	最多 6 个	3.072 MHz	18.432 MHz	384 kHz	47	0b110	1	每个帧 2 个 24 位 Slot
		3.072 MHz	18.432 MHz	384 kHz	47	0b100	2	每个帧 3 个 16 位 Slot
		3.072 MHz	18.432 MHz	384 kHz	47	0b010	5	每个帧 6 个 8 位 Slot
	最多 4 个	3.072 MHz	12.288 MHz	384 kHz	31	0b111	0	每个帧 1 个 32 位 Slot
		3.072 MHz	12.288 MHz	384 kHz	31	0b100	1	每个帧 2 个 16 位 Slot
		3.072 MHz	12.288 MHz	384 kHz	31	0b010	3	每个帧 4 个 8 位 Slot
	最多 2 个	3.072 MHz	6.144 MHz	384 kHz	15	0b100	0	每个帧 1 个 16 位 Slot
		3.072 MHz	6.144 MHz	384 kHz	15	0b010	1	每个帧 2 个 8 位 Slot
16 kHz	最多 8 个	1.024 MHz	8.192 MHz	128 kHz	63	0b111	1	每个帧 2 个 32 位 Slot
		1.024 MHz	8.192 MHz	128 kHz	63	0b100	3	每个帧 4 个 16 位 Slot
		1.024 MHz	8.192 MHz	128 kHz	63	0b010	7	每个帧 8 个 8 位 Slot
	最多 6 个	1.024 MHz	6.144 MHz	128 kHz	47	0b110	1	每个帧 2 个 24 位 Slot
		1.024 MHz	6.144 MHz	128 kHz	47	0b010	5	每个帧 6 个 8 位 Slot
	最多 4 个	1.024 MHz	4.096 MHz	128 kHz	31	0b111	0	每个帧 1 个 32 位 Slot
		1.024 MHz	4.096 MHz	128 kHz	31	0b100	1	每个帧 2 个 16 位 Slot
		1.024 MHz	4.096 MHz	128 kHz	31	0b010	3	每个帧 4 个 8 位 Slot
	最多 2 个	1.024 MHz	2.048 MHz	128 kHz	15	0b100	0	每个帧 1 个 16 位 Slot
		1.024 MHz	2.048 MHz	128 kHz	15	0b010	1	每个帧 2 个 8 位 Slot

1. 有关 TDM 配置的更多信息，请参见*表 460：TDM 设置*。提供给 SAI 的 sai_a_ker_ck 时钟频率必须是 SCK_A 频率的倍数，并且必须相应地编程 MCKDIV。
2. 上述 sai_a_ker_ck 频率仅作为示例给出。请参见*复位和时钟控制器 (RCC) 部分以确定设备是否可以生成这些频率*。
3. 上表给出了抽取率为 64 时所允许的设置。

调整延迟线

使能 PDM 接口时，应用程序可通过 SAI_PDMDLY 寄存器实时调整每个麦克风输入的延迟单元数。

新的延迟值在两个 TDM 帧后生效。

51.4.11 AC’97 链路控制器

SAI 可用作 AC’97 链路控制器。在此协议中：

- Slot 数和 Slot 大小固定。
- 帧同步信号定义完善并且波形固定。

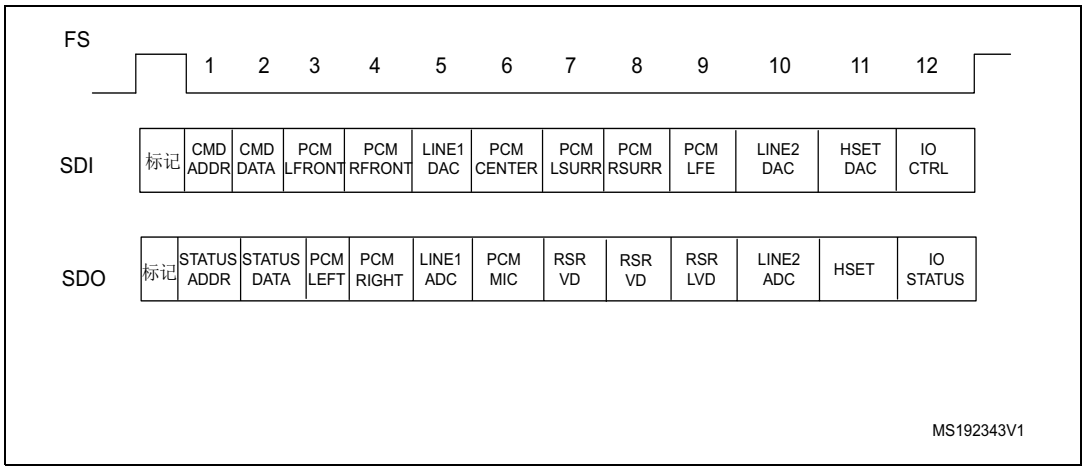
要选择此协议，可将 SAI_xCR1 寄存器中的 PRTCFG[1:0] 位置为 10。选择 AC’97 模式时，只能使用 16 位或 20 位的数据大小，否则将无法保证 SAI 运行正常。

- 因此，NBSLOT[3:0] 和 SLOTSZ[1:0] 位将被忽略。
- Slot 数固定为 13。第一个 Slot 为 16 位宽，所有其他 Slot 均为 20 位宽（数据 Slot）。
- SAI_xSLOTR 寄存器中的 FBOFF[4:0] 位被忽略。
- SAI_xFRCR 寄存器被忽略。
- 未使用 MCLK。

无论采用主配置还是从配置，由于 AC’97 链路控制器驱动 FS 信号，异步模块发出的 FS 信号将自动配置为输出。

图 639 给出了 AC’97 音频帧的结构。

图 639. AC’97 音频帧



注： *在 AC’97 协议中，TAG 的位 2 将保留（始终为 0），因此无论 SAI FIFO 中写入何值，TAG 的位 2 均强制为 0。*

有关 TAG 表示的详细信息，请参见 AC’97 协议标准。

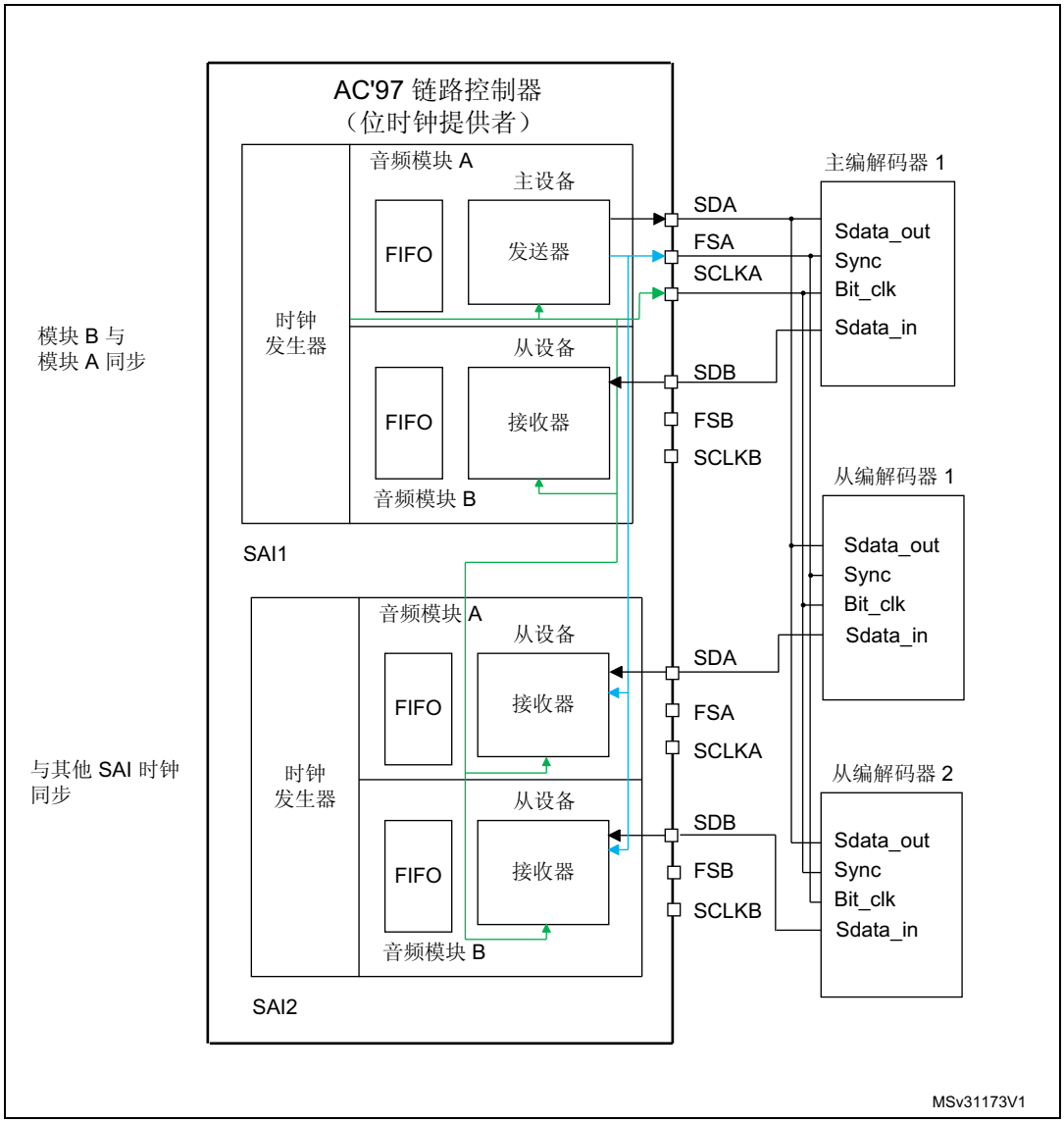
可将一个 SAI 用于 AC’97 点对点通信。

使用两个 SAI（针对具有两个嵌入式 SAI 的设备）可以控制三个外部 AC’97 解码器，如*图 640* 所示。

在 SAI1 中，必须将音频模块 A 声明为异步主发送器，而将音频模块 B 定义为从接收器，并在内部与音频模块 A 同步。

在从接收器模式下，同时针对音频模块 A 和音频模块 B 将 SAI2 配置为与外部 SAI1 同步。

图 640. 至少具有 2 个嵌入式 SAI 的设备的典型 AC’97 配置示例（三个外部 AC’97 解码器）



在接收器模式下，用作 AC’97 链路控制器的 SAI 无需 FIFO 请求，因此当 Slot 0 中的编码解码器就绪位解码为低电平时，FIFO 中不存储任何数据。如果 SAI_xIM 寄存器中的 CNRDYIE 位使能，则 SAI_xSR 寄存器中的标志 CNRDY 将置 1 并会产生一个中断。此标志专用于 AC’97 协议。

AC’97 模式下的时钟发生器编程

在 AC’97 模式下，帧长度固定为 256 位，其频率必须设置为 48 kHz。*第 51.4.8 节：SAI 时钟发生器*中给出的规则应在 FRL = 255 时使用，以生成适当的帧速率 (F_{FS_x})。

51.4.12 SPDIF 输出

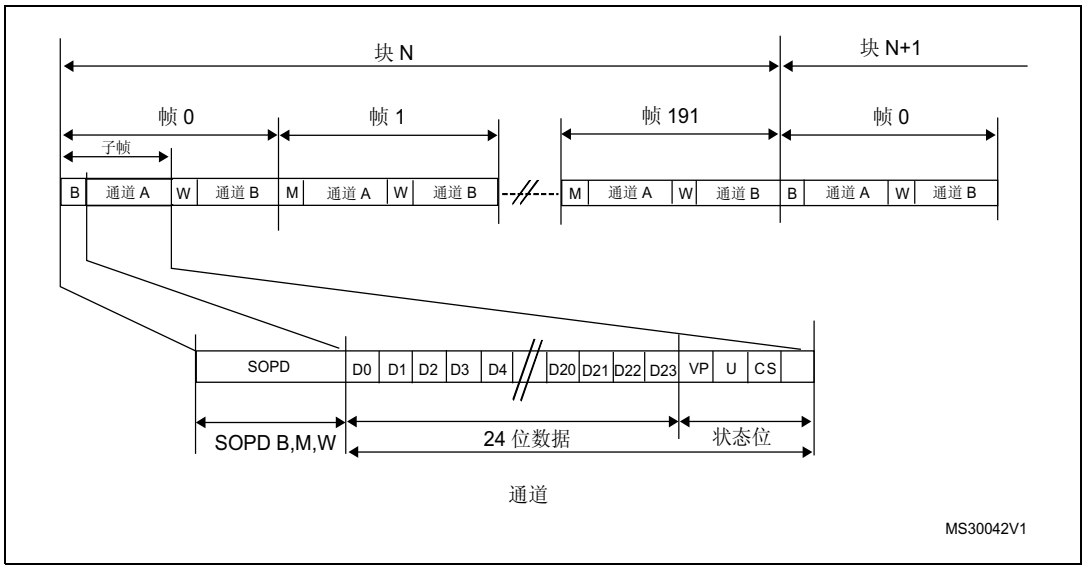
SPDIF 接口仅在发送器模式下可用。它支持 IEC60958 音频标准。

要选择 SPDIF 模式，需将 SAI_xCR1 寄存器中的 PRTCFG[1:0] 位置为 01。

对于 SPDIF 协议：

- 仅使能 SD 数据线。
- 释放 FS、SCK、MCLK I/O 引脚。
- 为使能 SAI 的时钟发生器并管理 SD 线上的数据速率，将 MODE[1]位强制设置为 0 以选择主模式。
- 数据大小强制设置为 24 位。忽略在 SAI_xCR1 寄存器的 DS[2:0] 位中设置的值。
- 必须配置时钟发生器以定义符号率，已知位时钟必须为符号率的两倍。数据采用曼彻斯特协议进行编码。
- 忽略 SAI_xFRCR 和 SAI_xSLOTR 寄存器。在内部配置 SAI 使其符合 SPDIF 协议要求，如*图 641* 所示。

图 641. SPDIF 格式



SPDIF 块包含 192 个帧。每个帧由两个 32 位的子帧构成，通常一个子帧用于左通道，一个用于右通道。每个子帧由一个 SOPD 模式（4 位）构成，用于指定该子帧是否为块的开始（并用于识别通道 A），或者是否标识块中某处的通道 A，或者是否指代通道 B（请参见*表 462*）。接下来的 28 位是通道信息，由 24 个数据位和 4 个状态位组成。

表 462. SOPD 模式

SOPD	前导码编码		说明
	最后一位是 0	最后一位是 1	
B	11101000	00010111	块开始处的通道 A 数据
W	11100100	00011011	块中某处的通道 B 数据
M	11100010	00011101	通道 A 数据

必须按如下方式填充 SAI_xDR 中存储的数据：

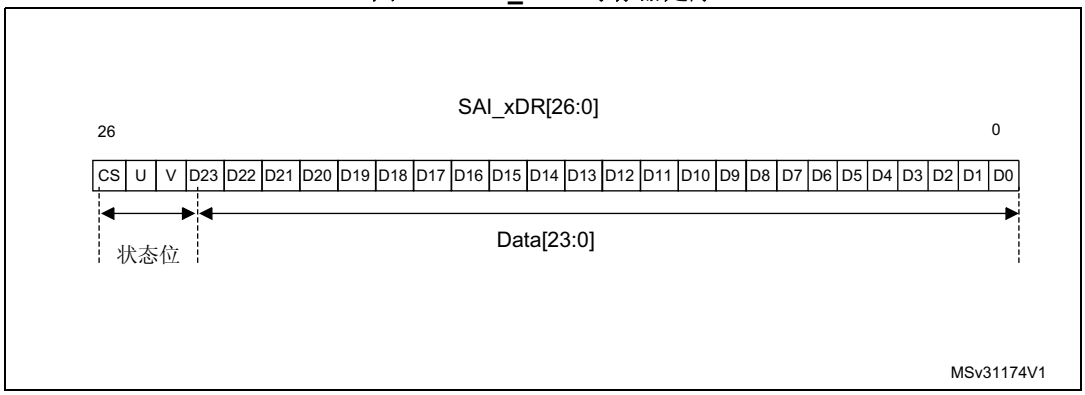
- SAI_xDR[26:24] 包含通道状态位、用户位和有效性位。
- SAI_xDR[23:0] 包含所考虑通道的 24 位数据。

如果数据大小为 20 位，必须将数据映射到 SAI_xDR[23:4] 上。

如果数据大小为 16 位，必须将数据映射到 SAI_xDR[23:8] 上。

SAI_xDR[23] 始终代表 MSB。

图 642. SAI_xDR 寄存器定序



注： *执行传输时，LSB 始终优先。*

SAI 首先在块中发送每个子帧的适当前导码。随后在 SD 线上发送 SAI_xDR（以曼彻斯特协议进行编码）。SAI 通过传输按*表 463* 所述计算的奇偶校验位来结束子帧。

表 463. 奇偶校验位计算

SAI_xDR[26:0]	传输奇偶校验位 P 值
奇数个 0	0
奇数个 1	1

在 SPDIF 模式下，下溢是在 SAI_xSR 寄存器唯一个用到的错误标志。因为 SAI 只能在发送器模式下工作。因此，要从通过下溢中断或下溢状态位检测到的下溢错误中恢复，必须按顺序执行以下步骤：

1. 如果已使用 DMA，则禁止 DMA 流（通过 DMA 外设）。
2. 禁止 SAI 并通过轮询 SAI_xCR1 寄存器中的 SAIEN 位确认已从物理上禁止外设。
3. 将 SAI_xCLRFR 寄存器中的 COVRUNDR 标志清零。
4. 通过将 SAI_xCR2 寄存器中的 FFLUSH 位置 1 来刷新 FIFO。
   软件需要指向与新块开始（前导码 B 的数据）相对应的后续数据的地址。如果已使用 DMA，必须相应地更新 DMA 源起始地址指针。
5. 如果用于管理数据的 DMA 根据新的源起始地址来传输数据，则再次使能 DMA 流（DMA 外设）。
6. 通过将 SAI_xCR1 寄存器中的 SAIEN 位置 1 再次使能 SAI。

SPDIF 发生器模式下的时钟发生器编程

对于 SPDIF 发生器，SAI 应提供等于符号率两倍的位时钟。下表给出了符号率相对于音频采样率的通常示例。

表 464. 音频采样频率与符号率

音频采样频率 (F_S)	符号率
44.1 kHz	2.8224 MHz
48 kHz	3.072 MHz
96 kHz	6.144 MHz
192 kHz	12.288 MHz

通常，音频采样频率 (F_S) 与位时钟速率 (F_{SCK_X}) 之间的关系由以下公式给出：

$$F_S = \frac{F_{SCK_x}}{128}$$

按如下公式计算位时钟速率：

$$F_{SCK_x} = \frac{F_{sai_x_ker_ck}}{MCKDIV}$$

注： *仅当在 SAI_ACR1 寄存器中将 NODIV 置 1 时，上述公式才有效。*

51.4.13 特性

根据所选的音频协议，SAI 接口内嵌了一些特定的实用功能。这些功能可通过 SAI_xCR2 寄存器的特定位来访问。

静音模式

当音频子模块用作发送器或接收器时，可使用静音模式。

发送模式下的音频子模块

在发送器模式下，可随时选择静音模式。静音模式对于全部音频帧均有效。在帧传输过程中将 SAI_xCR2 寄存器的 MUTE 置 1，将使能静音模式。

该静音模式位仅在帧结束时选通。如果帧结束时置 1，静音模式将在新的音频帧开始时激活，并持续整个帧长度，直至下次帧结束；然后将选通该位，以确定下一帧是否仍为静音帧。

如果通过 SAI_xSLOTR 寄存器的 NBSLOT[3:0] 位设置的 Slot 数小于或等于 2，可指定静音模式下发送的值是否为 0 或该值是否为每个 Slot 的最后一个值。通过 SAI_xCR2 寄存器中的 MUTEVAL 位进行选择。

如果在 SAI_xSLOTR 寄存器的 NBSLOT[3:0] 位中设置的 Slot 数大于 2，由于在各 Slot 的每位上都发送值 0，因此 SAI_xCR2 中的 MUTEVAL 位没有意义。

在静音模式下，FIFO 指针仍递增，这意味着将丢弃 FIFO 中请求在静音模式下传输的数据。

接收模式下的音频子模块

在接收模式下，对于给定数量的连续音频帧（SAI_xCR2 寄存器中的 MUTECNT[5:0] 位），如果在音频帧所有声明的有效 Slot 接收到 0，则可检测到从外部发送器发来的静音模式。

检测到相应数量的静音帧时，SAI_xSR 寄存器中的 MUTEDET 标志置 1 并会在 SAI_xCR2 中的 MUTEDETIE 位置 1 情况下产生中断。

在音频子模块禁止时或在有效 Slot 内至少接收到音频帧中的一个数据时，静音帧计数器清零。计数器达到 MUTECNT[5:0] 位中指定的值时，仅产生一次中断。随后中断事件在计数器清零时重新初始化。

注： *静音模式不可用于 SPDIF 音频模块。*

单声道/立体声模式

在发送器模式下，如果 Slot 数等于 2（SAI_xSLOTR 中的 NBSLOT[3:0] = 0001），则可寻址单声道模式，而无需在内存中进行任何数据预处理。在这种情况下，由于发送时 Slot 0 的数据被复制到数据 Slot 1 中，FIFO 的访问时间将减少一半。

要使能单声道模式，

1. 将 SAI_xCR1 寄存器中的 MONO 位置 1。
2. 将 SAI_xSLOTR 中的 NBSLOT 置 1，并将 SLOTEN 设置为 3。

在接收模式下，MONO 位可置 1 并且仅在 Slot 数等于 2（与发送器模式下相同）时才有意义。当该位置 1 时，只有 Slot 0 的数据存储到 FIFO 中。Slot 1 的数据由于被认为与前一个 Slot 的数据相同而被丢弃。如果接收模式下的数据流量是左声道数据和右声道数据明显不同的真立体声音频流，则 MONO 位没有意义。由软件完成从输出立体声文件到等效单声道文件的转换。

压扩模式

移动通信应用可能需要通过数据压扩算法处理待发送或待接收的数据。

应用软件可根据 SAI_xCR2 寄存器中的 COMP[1:0] 位（仅当选择自由协议模式时使用），选择在 SD 串行输出线（压缩）发送数据前是否处理数据，以及是否在 SD 串行输入线（扩展）接收数据后扩展数据，如*图 643* 所示。所支持的两个压扩模式是 μ-Law 和 A-Law，二者是 CCITT G.711 推荐标准的一部分。

美国和日本采用的压扩标准是 μ-Law，该标准允许 14 位动态范围（SAI_xCR2 寄存器中的 COMP[1:0] = 10）。

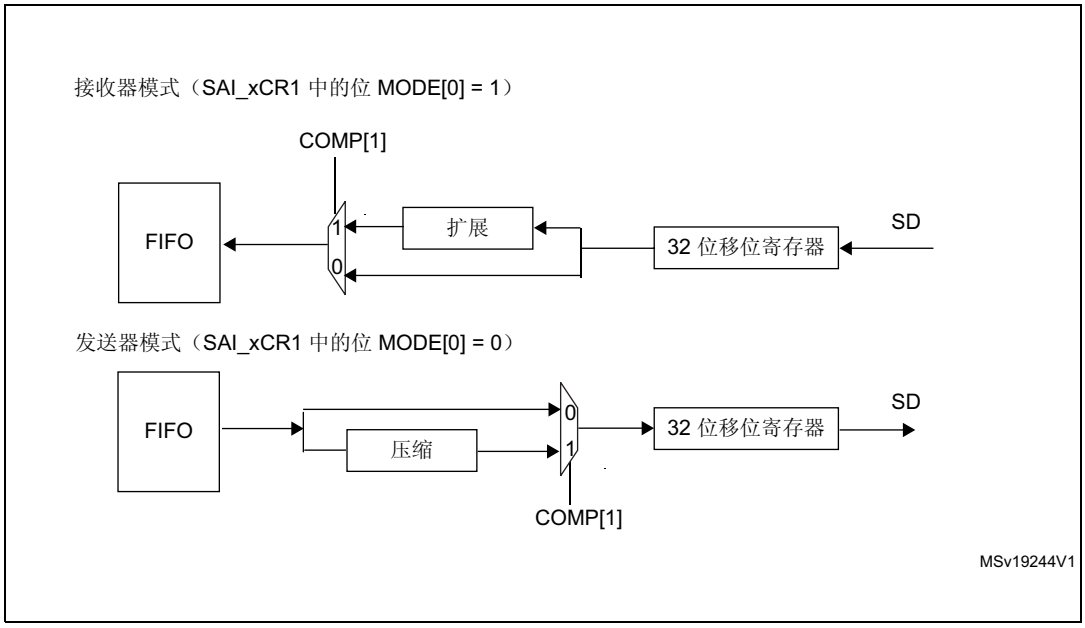
欧洲压扩标准是 A-Law，该标准支持 13 位动态范围（SAI_xCR2 寄存器中的 COMP[1:0] = 11）。

可根据 1 的补码或 2 的补码表示来计算 μ-Law 或 A-Law 压扩标准，具体取决于 SAI_xCR2 寄存器中的 CPL 位设置。

在 μ-Law 和 A-Law 标准中，数据将编码为采用 MSB 对齐的 8 位。压扩数据始终为 8 位宽。因此，当 SAI 音频模块使能（SAI_xCR1 寄存器中的位 SAIEN = 1）并且通过 COMP[1:0] 位选择这两个压扩模式之一时，SAI_xCR1 寄存器中的 DS[2:0] 位强制为 010。

如果无需压扩处理，则 COMP[1:0] 位必须保持清零。

图 643. SAI 的音频模块中的数据压扩硬件



1. 选择 AC'97 或 SPDIF 时不适用。

通过 SAI_xCR2 自动选择扩展模式和压缩模式：

- 如果 SAI 音频模块配置为发送器，且 SAI_xCR2 寄存器中的 COMP[1] 位置 1，则采用压缩模式。
- 如果 SAI 音频模块声明为接收器，则采用扩展算法。

无效 Slot 上的输出数据线管理

在发送器模式下，当在数据线上发送无效 Slot 时，可选择 SD 线输出的行为（通过 TRIS 位实现）。

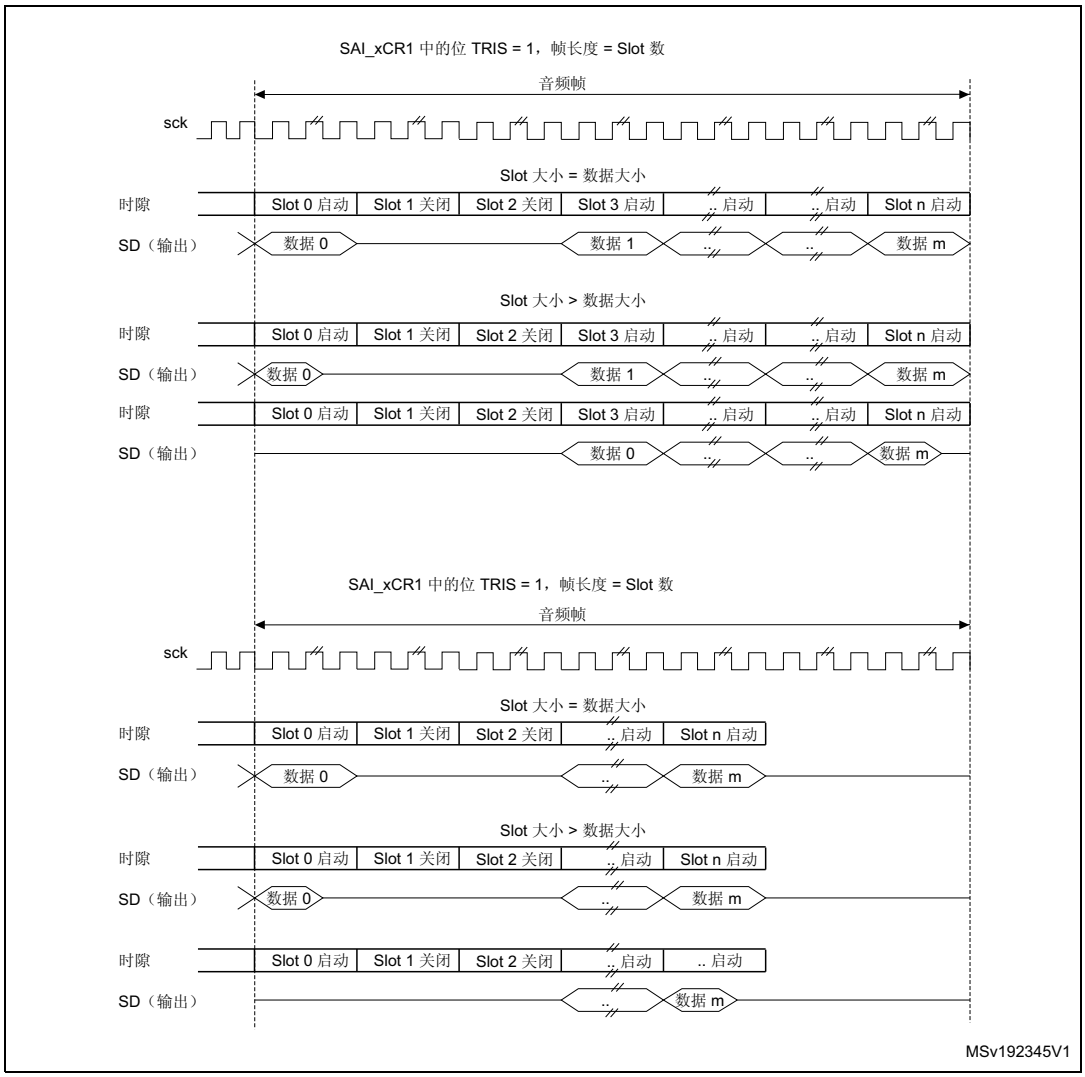
- 发送无效 Slot 时，SAI 将 SD 输出线强制为 0
- 该输出线在最有一个数据位传输结束时释放为高阻态，为另一个连接此节点的发送器释放该数据线。

切记不要让两个发送器同时驱动同一个 SD 输出引脚，否则可能导致短路。为确保存在发送间隙，如果数据低于 32 位，可通过在 SAI_xSLOTR 寄存器中设置 SLOTSZ[1:0] = 10 将数据扩展到 32 位。随后，如果下一 Slot 声明为无效，则 SD 输出引脚在有效 Slot 的 LSB 结束时（将数据扩展到 32 位的填 0 阶段）置为三态。

此外，如果 Slot 数乘以 Slot 大小所得结果小于帧长度，则在通过填 0 来补充音频帧结束时，SD 输出线置为三态。

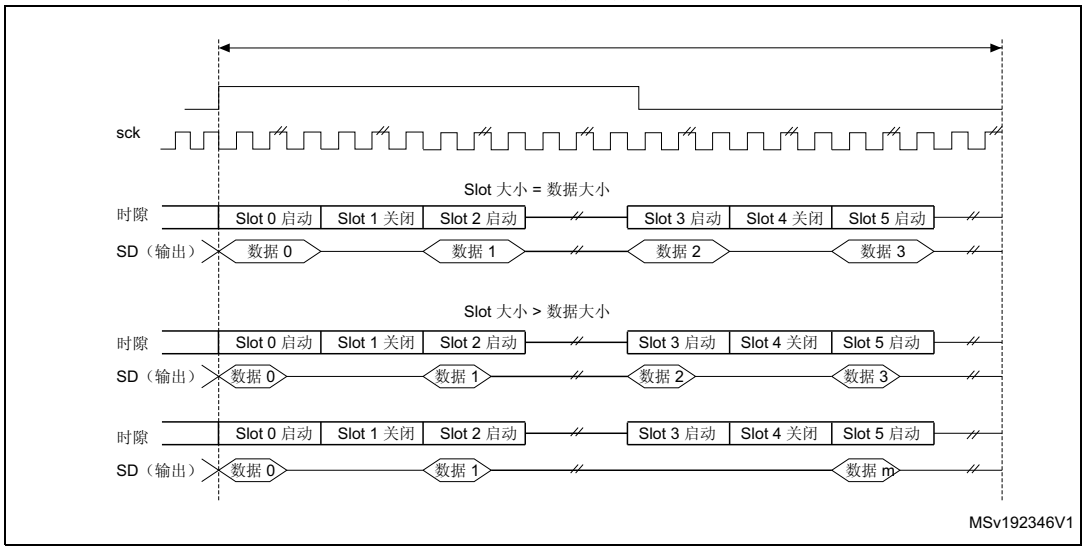
图 644 说明了这些行为。

图 644. 发送无效 Slot 时 SD 输出线上的三态策略



当所选音频协议使用 FS 信号作为 SOF 信号或通道识别信号（SAI_xFRCR 寄存器中的位 FSDEF = 1）时，将按照*图 645* 管理三态模式（其中，SAI_xCR1 寄存器中的位 TRIS = 1，FSDEF = 1，半帧长大于 Slot 数/2 且 NBSLOT = 6）。

图 645. 采用 I2S 等协议时输出数据线上的三态策略



如果 SAI_xCR2 寄存器中的 TRIS 位清零，*图 644* 和*图 645* 上的 SD 输出线上的所有高阻态将替换为使用值 0 驱动。

51.4.14 错误标志

SAI 使用以下错误标志：

- FIFO 上溢/下溢
- 帧同步提前检测
- 帧同步滞后检测
- 编解码器未就绪（仅限 AC'97）
- 主模式时钟配置错误。

FIFO 上溢/下溢 (OVRUDR)

FIFO 上溢/下溢位是 SAI_xSR 寄存器中的 OVRUDR 位。

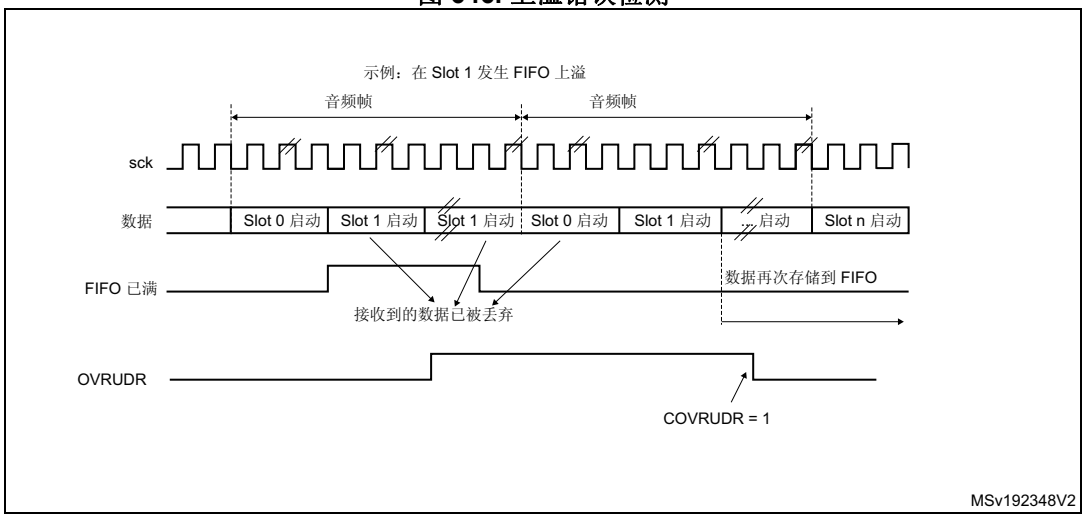
由于音频模块既可作为接收器，又可作为发送器，并且指定 SAI 中的每个音频模块都具有自己的 SAI_xSR 寄存器，因此上溢或下溢错误共用同一位。

上溢

若音频模块配置为接收器，则在 FIFO 已满且无法再存储接收数据的情况下又收到音频帧数据时，将出现上溢情况。这种情况下，接收数据将丢失，SAI_xSR 寄存器中的 OVRUDR 标志置 1；如果 SAI_xIM 寄存器中的 OVRUDRIE 位置 1，还将生成中断。内部将存储发生上溢时的 Slot 编号。FIFO 无法再存储更多数据，直至释放出空间存储新数据为止。在 FIFO 释放了至少一个数据的空间时，SAI 音频模块接收器从检测到上溢后内部记录的 Slot 编号开始接收来自新音频帧的新数据，这样可避免目标内存中出现数据 Slot 不对齐的情况（请参见*图 646*）。

SAI_xCLRFR 寄存器中的 COVRUDR 位置 1 时将清零 OVRUDR 标志。

图 646. 上溢错误检测



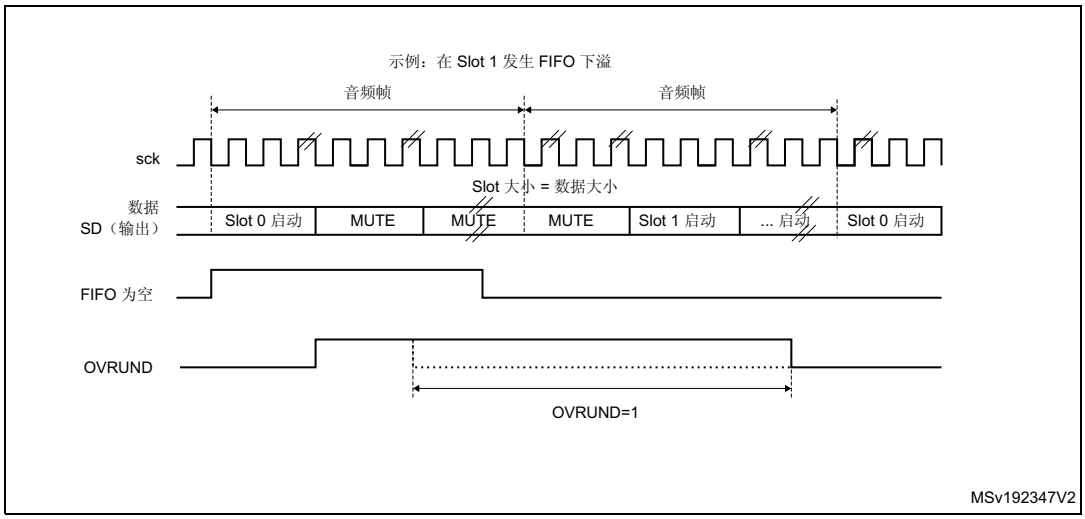
下溢

当 SAI 中的音频模块用作发送器时，如果需要发送数据时 FIFO 为空，则可能出现下溢。如果检测到下溢，则将存储发生事件的 Slot 编号并发送 MUTE 值 (00)，直到 FIFO 准备好发送与检测到下溢的 Slot 对应的数据（请参见*图 647*）。这样可避免内存指针与音频帧中的 Slot 之间发生同步失效。

下溢事件会使 SAI_xSR 寄存器中的 OVRUDR 标志置 1，如果 SAI_xIM 寄存器中的 OVRUDRIE 位置 1，还将生成中断。要清零该标志，可将 SAI_xCLRFR 寄存器中的 COVRUDR 位置 1。

将音频子模块配置为主模式或从模式时，会发生下溢事件。

图 647. FIFO 下溢事件



帧同步提前检测 (AFSDET)

AFSDET 标志仅在从模式下使用。主模式下不会使能该标志。由于帧长度、帧极性和帧偏移已定义且已知，该标志用于指示是否比预期更早检测到帧同步 (FS) 信号。

出现帧检测提前时，SAI_xSR 寄存器中的 AFSDET 标志将置 1。

对 FS 提前不敏感的当前音频帧不受该检测影响。也就是说 FS 信号的“寄生”事件将被标记但不干扰当前音频帧。

如果 SAI_xIM 寄存器中的 AFSDETIE 位置 1，将生成中断。要清零 AFSDET 标志，必须将 SAI_xCLRFR 寄存器中的 CAFSDET 位置 1。

为了在出现帧检测提前错误后重新与主模块同步，需要执行以下四个步骤：

1. 通过复位 SAI_xCR1 寄存器中的 SAIEN 位禁止 SAI 模块。为确保禁止 SAI，读回 SAIEN 位并检查其是否已设置为 0。
2. 通过 SAI_xCR2 寄存器的 FFLUS 位刷新 FIFO。
3. 重新使能 SAI 外设（SAIEN 位置 1）。
4. SAI 模块等待 FS 使能以重新开始与主模块同步。

注： *AC’97 模式下不使能 AFSDET 标志，原因是 SAI 音频模块作为链路控制器，即使在声明为从模块时也会生成 FS 信号。由于未使用 FS 信号，因此它在 SPDIF 模式下无意义。*

帧同步滞后检测

只有当 SAI 音频模块用作从模块时，SAI_xSR 寄存器中的 LFSDET 标志才可置 1。SAI_xFRCR 寄存器中，帧长度、帧极性和帧偏移配置均已知。

如果外部主模块未在预定时间发送 FS 信号生成过晚），LFSDET 标志将置 1，如果 SAI_xIM 寄存器中的 LFSDETIE 位置 1，还将生成中断。

SAI_xCLRFR 寄存器中的 CLFSDET 位置 1 时将清零 LFSDET 标志。

在检测到相应错误时帧同步滞后检测标志置 1，SAI 需要重新与主模块同步（请参见 *帧同步提前检测 (AFSDET)* 中所述的顺序）。

在噪声环境中，音频模块的状态机可能会错误检测到对 SCK 时钟的干扰，并将 SAI 数据移位到错误的帧位置。SAI 会检测到此事件，并将其报告为帧同步滞后检测错误。

如果外部主模块不是在连续模式下管理音频数据帧发送，则不会对帧造成破坏，大多数应用都不会出现这种情况。在这种情况下，LFSDET 标志置 1。

注： *AC’97 模式下不使能 LFSDET 标志，原因是 SAI 音频模块作为链路控制器，即使在声明为从模块时也会生成 FS 信号。由于协议未使用 FS 信号，因此该位在 SPDIF 模式下无意义。*

编解码器未就绪 (CNRDY AC’97)

仅当 SAI 音频模块配置为在 AC’97 模式下工作时（SAI_xCR1 寄存器中的 PRTCFG[1:0] = 10），SAI_xSR 寄存器中的 CNRDY 标志才有意义。如果 SAI_xIM 寄存器中的 CNRDYIE 位置 1，则在 CNRDY 标志置 1 时将生成中断。

在接收 AC’97 音频帧的 TAG 0 (slot0) 期间，当编解码器未准备好进行通信时，将使能 CNRDY。这种情况下，在 TAG 0 指示编解码器就绪之前，数据都不会自动存储到 FIFO，原因是编解码器未就绪。编解码器就绪后将捕获 SAI_xSLOTR 寄存器中定义的所有有效 Slot。

要清零 CNRDY 标志，必须将 SAI_xCLRFR 寄存器中的 CCNRDY 位置 1。

主模式时钟配置错误 (NODIV = 0)

当音频模块在主模式下工作 (MODE[1] = 0) 且 NODIV 位等于 0 时，如果满足以下条件，则只要使能 SAI，WCKCFG 标志便会置 1：

- (FRL+1) 不是 2 的几次幂，并且
- (FRL+1) 不在 8 和 256 之间。

MODE 位、NODIV 位和 SAIEN 位属于 SAI_xCR1 寄存器，FRL 位属于 SAI_xFRCR 寄存器。

如果 WCKCFGIE 位置 1，则当 SAI_xSR 寄存器中的 WCKCFG 标志置 1 时将生成中断。要清零该标志，可将 SAI_xCLRFR 寄存器中的 CWCKCFG 位置 1。

当 WCKCFG 位置 1 时，音频模块会自动禁止，因此将对 SAIEN 位执行硬件清零。

51.4.15 禁止 SAI

可随时通过清零 SAI_xCR1 寄存器中的 SAIEN 位禁止 SAI 音频模块。所有已开始的帧将在 SAI 停止工作前自动完成。SAIEN 位将保持高电平，直到当前音频帧传输结束时 SAI 完全关闭。

如果 SAI 中有与另一个音频模块同步工作的音频模块，则必须先禁止以主模式工作的音频模块。

51.4.16 SAI DMA 接口

为减轻 CPU 负担和优化总线带宽，每个 SAI 音频模块都具有独立的 DMA 接口以便对 SAI_xDR 寄存器进行读/写操作（访问内部 FIFO）。每个音频子模块都有一个支持基本 DMA 请求/应答协议的 DMA 通道。

要为 DMA 传输配置音频子模块，可将 SAI_xCR1 寄存器中的 DMAEN 位置 1。DMA 请求直接由 FIFO 控制器管理，具体取决于 FIFO 阈值（更多详细信息，请参见*第 51.4.9 节：内部 FIFO*）。DMA 传输方向与 SAI 音频子模块配置相关：

- 如果音频模块用作发送器，则音频模块的 FIFO 控制器将输出 DMA 请求以向 FIFO 加载 SAI_xDR 寄存器中写入的数据。
- 如果音频模块用作接收器，则 DMA 请求与来自 SAI_xDR 寄存器的读取操作相关。

按照下面的顺序将 SAI 接口配置为 DMA 模式：

1. 配置 SAI 和 FIFO 阈值以指定何时启动 DMA 请求。
2. 配置 SAI DMA 通道。
3. 使能 DMA。
4. 使能 SAI 接口。

注： *配置 SAI 模块前，必须禁止 SAI DMA 通道。*

51.5 SAI 中断

SAI 支持 7 个中断源，如*表 465* 所示。

表 465. SAI 中断源

中断缩写	中断源	中断组	音频模块模式	中断使能	中断清零
SAI	FREQ	FREQ	主或从 接收器或发送器	SAI_xIM 寄存器中的 FREQIE	取决于： – FIFO 阈值设置（SAI_xCR2 中的 FLVL 位） – 通信方向（发送器或接收器） 更多详细信息，请参见*第 51.4.9 节：内部 FIFO*
	OVRUDR	ERROR	主或从 接收器或发送器	SAI_xIM 寄存器中的 OVRUDRIE	SAI_xCLRFR 寄存器中的 COVRUDR = 1
	AFSDET	ERROR	从 （不适用于 AC’97 模式和 SPDIF 模式）	SAI_xIM 寄存器中的 AFSDETIE	SAI_xCLRFR 寄存器中的 CAFSDET = 1
	LFSDET	ERROR	从 （不适用于 AC’97 模式和 SPDIF 模式）	SAI_xIM 寄存器中的 LFSDETIE	SAI_xCLRFR 寄存器中的 CLFSDET = 1
	CNRDY	ERROR	从 （仅限 AC’97 模式）	SAI_xIM 寄存器中的 CNRDYIE	SAI_xCLRFR 寄存器中的 CCNRDY = 1
	MUTEDET	MUTE	主或从 仅限接收器模式	SAI_xIM 寄存器中的 MUTEDETIE	SAI_xCLRFR 寄存器中的 CMUTEDET = 1
	WCKCFG	ERROR	主模式且 SAI_xCR1 寄存器中的 NODIV = 0	SAI_xIM 寄存器中的 WCKCFGIE	SAI_xCLRFR 寄存器中的 CWCKCFG = 1

按照以下顺序使能中断：

1. 禁止 SAI 中断。
2. 配置 SAI。
3. 配置 SAI 中断源。
4. 使能 SAI。

51.6 SAI 寄存器

外设寄存器必须按字（32 位）进行访问。

51.6.1 全局配置寄存器 (SAI_GCR)

SAI global configuration register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYNCOUT[1:0]		Res.	Res.	SYNCIN[1:0]	
										rw	rw			rw	rw

位 31:6 保留，必须保持复位值。

位 5:4 **SYNCOUT[1:0]**：同步输出 (Synchronization outputs)

这些位由软件置 1 和清零，

00：无同步输出信号。将音频模块配置为 SPDIF 模式时，必须将 SYNCOUT[1:0] 配置为“无同步输出信号”

01：模块 A 用于其他 SAI 的进一步同步

10：模块 B 用于其他 SAI 的进一步同步

11：保留。必须在音频模块（A 和 B）禁止的情况下设置这些位。

位 3:2 保留，必须保持复位值。

位 1:0 **SYNCIN[1:0]**：同步输入 (Synchronization inputs)

这些位由软件置 1 和清零，

有关如何编程该位字段的信息，请参见 *表 457：外部同步选择*。

必须在音频模块（A 和 B）禁止的情况下配置这些位。

如果将两个音频模块之一定义为与外部 SAI 在同步模式下工作（SAI_ACR1 或 SAI_BCR1 寄存器中的 SYNCEN[1:0] = 10），这些位便有意义。

51.6.2 SAI 配置寄存器 1 (SAI_ACR1)

SAI configuration register 1

偏移地址：0x004

复位值：0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	MCKEN	OSR	MCKDIV[5:0]						NODIV	Res.	DMAEN	SAIEN
				rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	OUTDRIV	MONO	SYNCEN[1:0]		CKSTR	LSBFIRST	DS[2:0]			Res.	PRTCFG[1:0]		MODE[1:0]	
		rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27 **MCKEN**：主时钟生成使能 (Master clock generation enable)
0：不生成主时钟
1：独立于 SAIEN 位生成主时钟

位 26 **OSR**：主时钟的过采样率 (Oversampling ratio for master clock)
只有在 NODIV 位置 0 时，该位才有意义。
0：主时钟频率 = F_{FS} x 256
1：主时钟频率 = F_{FS} x 512

位 25:20 **MCKDIV[5:0]**：主时钟分频器 (Master clock divider)
这些位由软件置 1 和清零，
000000: 内核时钟输入 (sai_x_ker_ck) 1 分频。
否则，主时钟频率将根据*第 51.4.8 节：SAI 时钟发生器*中的公式计算。
音频模块为从模块时，这些位没有意义。
必须在音频模块禁止的情况下配置这些位。

位 19 **NODIV**：无分频器
该位由软件置 1 和清零，
0：主时钟发生器和帧同步之间的比值固定为 256 或 512
1：主时钟发生器和帧同步之间的比值取决于 FRL[7:0]

位 18 保留，必须保持复位值。

位 17 **DMAEN**：DMA 使能
该位由软件置 1 和清零，
0：禁止 DMA
1：使能 DMA
注意：在接收器模式下，必须在 DMAEN 位置 1 前配置 MODE[1:0] 位，以避免 DMA 请求，原因是复位后音频模块将默认以发送模式工作。

位 16 **SAIEN**：音频模块使能 (Audio block enable)

该位由软件置 1。

要关闭音频模块，必须由应用软件将该位编程为 0 并轮询该位，直到该位读回 0，这表示该模块已完全禁止。将该位置 1 前，先检查该位是否已置 0，否则不会执行此使能命令。

该位可用于控制 SAI 音频模块的状态。如果在音频帧传输期间禁止该位，则正在进行的传输将完成并且单元在该音频帧传输结束时完全禁止。

0：禁止 SAI 音频模块

1：使能 SAI 音频模块

注意：当 SAI 模块（A 或 B）配置为主模式时，SAI 模块输入中必须有时钟，然后才能将 SAIEN 位置 1。

位 15:14 保留，必须保持复位值。

位 13 **OUTDRIV**：输出驱动 (Output drive)

该位由软件置 1 和清零，

0：当 SAIEN 置 1 时驱动音频模块输出

1：在该位置 1 后立即驱动音频模块输出。

注意：该位必须在音频模块配置后的使能前置 1。

位 12 **MONO**：单声道模式 (Mono mode)

该位由软件置 1 和清零，仅当 Slot 数为 2 时，该位才有意义。如果选择了单声道模式，则当音频模块用作发送器时，Slot 0 的数据将复制到 Slot 1 上。在接收模式下，将丢弃 Slot 1 并仅存储从 Slot 0 接收的数据。更多详细信息，请参见*单声道/立体声模式*一节。

0：立体声模式

1：单声道模式

位 11:10 **SYNCEN[1:0]**：同步使能 (Synchronization enable)

这些位由软件置 1 和清零，必须在音频子模块禁止的情况下配置这些位。

00：音频子模块处于异步模式。

01：音频子模块与另一个内部音频子模块同步。这种情况下，必须将该音频子模块配置为从模式。

10：音频子模块与外部 SAI 的嵌入式外设同步。这种情况下，必须将该音频子模块配置为从模式。

11：保留

注意：使能 SPDIF 模式后，必须将音频子模块配置为异步。

位 9 **CKSTR**：时钟选通边沿 (Clock strobing edge)

该位由软件置 1 和清零，必须在音频模块禁止的情况下配置该位。SPDIF 音频协议下该位没有意义。

0：在 SCK 上升沿更改 SAI 生成的信号，而在 SCK 下降沿对 SAI 接收的信号进行采样。

1：在 SCK 下降沿更改 SAI 生成的信号，而在 SCK 上升沿对 SAI 接收的信号进行采样。

位 8 **LSBFIRST**：最低有效位优先 (Least significant bit first)

该位由软件置 1 和清零，必须在音频模块禁止的情况下配置该位。AC’97 音频协议下该位没有意义，原因是传输 AC’97 数据时，数据的 MSB 位优先。SPDIF 音频协议下该位没有意义，原因是传输 SPDIF 数据时，数据的 LSB 位优先。

0：优先传输数据的 MSB

1：优先传输数据的 LSB

位 7:5 **DS[2:0]**：数据大小 (Data size)

这些位由软件置 1 和清零，选择 SPDIF 协议时（位 PRTCFG[1:0]），将忽略这些位，原因是在这种情况下帧和数据大小是固定的。通过 COMP[1:0] 位选择压扩模式时，将忽略 DS[1:0]，原因是算法已将数据大小固定为 8 位。

必须在音频模块禁止的情况下配置这些位。

000：保留

001：保留

010：8 位

011：10 位

100：16 位

101：20 位

110：24 位

111：32 位

位 4 保留，必须保持复位值。

位 3:2 **PRTCFG[1:0]**：协议配置 (Protocol configuration)

这些位由软件置 1 和清零，必须在音频模块禁止的情况下配置这些位。

00：自由协议。通过设置大部分配置寄存器位以及帧配置寄存器，自由协议允许用户使用音频模块这一强大的配置功能来处理特定的音频协议（如 I2S、LSB/MSB 对齐、TDM、PCM/DSP...）。

01：SPDIF 协议

10：AC'97 协议

11：保留

位 1:0 **MODE[1:0]**：SAIx 音频模块模式 (SAIx audio block mode)

这些位由软件置 1 和清零，必须在 SAIx 音频模块禁止的情况下配置这些位。

00：主发送器

01：主接收器

10：从发送器

11：从接收器

注意：如果将音频模块配置为 SPDIF 模式，则将强制设置主发送器模式 (MODE[1:0] = 00)。

51.6.3 SAI 配置寄存器 1 (SAI_BCR1)

SAI configuration register 1

偏移地址：0x024

复位值：0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	MCKEN	OSR	MCKDIV[5:0]						NODIV	Res.	DMAEN	SAIEN
				rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	OUTDRIV	MONO	SYNCEN[1:0]		CKSTR	LSBFIRST	DS[2:0]			Res.	PRTCFG[1:0]		MODE[1:0]	
		rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27 **MCKEN**：主时钟生成使能 (Master clock generation enable)

0：不生成主时钟

1：独立于 SAIEN 位生成主时钟

位 26 **OSR**：主时钟的过采样率 (Oversampling ratio for master clock)
只有在 NODIV 位置 0 时，该位才有意义。
0：主时钟频率 = F_{FS} x 256
1：主时钟频率 = F_{FS} x 512

位 25:20 **MCKDIV[5:0]**：主时钟分频器 (Master clock divider)
这些位由软件置 1 和清零，
000000: 内核时钟输入 (sai_x_ker_ck) 1 分频。
否则，主时钟频率将根据*第 51.4.8 节：SAI 时钟发生器*中的公式计算。
音频模块为从模块时，这些位没有意义。
必须在音频模块禁止的情况下配置这些位。

位 19 **NODIV**：无分频器
该位由软件置 1 和清零，
0：主时钟发生器和帧同步之间的比值固定为 256 或 512
1：主时钟发生器和帧同步之间的比值取决于 FRL[7:0]

位 18 保留，必须保持复位值。

位 17 **DMAEN**：DMA 使能
该位由软件置 1 和清零，
0：禁止 DMA
1：使能 DMA
注意：在接收器模式下，必须在 DMAEN 位置 1 前配置 MODE[1:0] 位，以避免 DMA 请求，原因是复位后音频模块将默认以发送模式工作。

位 16 **SAIEN**：音频模块使能 (Audio block enable)
该位由软件置 1。
要关闭音频模块，必须由应用软件将该位编程为 0 并轮询该位，直到该位读回 0，这表示该模块已完全禁止。将该位置 1 前，先检查该位是否已置 0，否则不会执行此使能命令。
该位可用于控制 SAI 音频模块的状态。如果在音频帧传输期间禁止该位，则正在进行的传输将完成并且单元在该音频帧传输结束时完全禁止。
0：禁止 SAI 音频模块
1：使能 SAI 音频模块
注意：当 SAI 模块（A 或 B）配置为主模式时，SAI 模块输入中必须有时钟，然后才能将 SAIEN 位置 1。

位 15:14 保留，必须保持复位值。

位 13 **OUTDRIV**：输出驱动 (Output drive)
该位由软件置 1 和清零，
0：当 SAIEN 置 1 时驱动音频模块输出
1：在该位置 1 后立即驱动音频模块输出。
注意：该位必须在音频模块配置后的使能前置 1。

位 12 **MONO**：单声道模式 (Mono mode)
该位由软件置 1 和清零，仅当 Slot 数为 2 时，该位才有意义。如果选择了单声道模式，则当音频模块用作发送器时，Slot 0 的数据将复制到 Slot 1 上。在接收模式下，将丢弃 Slot 1 并仅存储从 Slot 0 接收的数据。更多详细信息，请参见*单声道/立体声模式*一节。
0：立体声模式
1：单声道模式

位 11:10 **SYNCEN[1:0]**：同步使能 (Synchronization enable)

这些位由软件置 1 和清零，必须在音频子模块禁止的情况下配置这些位。

00：音频子模块处于异步模式。

01：音频子模块与另一个内部音频子模块同步。这种情况下，必须将该音频子模块配置为从模式。

10：音频子模块与外部 SAI 的嵌入式外设同步。这种情况下，必须将该音频子模块配置为从模式。

11：保留

注意：使能 SPDIF 模式后，必须将音频子模块配置为异步。

位 9 **CKSTR**：时钟选通边沿 (Clock strobing edge)

该位由软件置 1 和清零，必须在音频模块禁止的情况下配置该位。SPDIF 音频协议下该位没有意义。

0：在 SCK 上升沿更改 SAI 生成的信号，而在 SCK 下降沿对 SAI 接收的信号进行采样。

1：在 SCK 下降沿更改 SAI 生成的信号，而在 SCK 上升沿对 SAI 接收的信号进行采样。

位 8 **LSBFIRST**：最低有效位优先 (Least significant bit first)

该位由软件置 1 和清零，必须在音频模块禁止的情况下配置该位。AC’97 音频协议下该位没有意义，原因是传输 AC’97 数据时，数据的 MSB 位优先。SPDIF 音频协议下该位没有意义，原因是传输 SPDIF 数据时，数据的 LSB 位优先。

0：优先传输数据的 MSB

1：优先传输数据的 LSB

位 7:5 **DS[2:0]**：数据大小 (Data size)

这些位由软件置 1 和清零，选择 SPDIF 协议时（位 PRTCFG[1:0]），将忽略这些位，原因是在这种情况下帧和数据大小是固定的。通过 COMP[1:0] 位选择压扩模式时，将忽略 DS[1:0]，原因是算法已将数据大小固定为 8 位。

必须在音频模块禁止的情况下配置这些位。

000：保留

001：保留

010：8 位

011：10 位

100：16 位

101：20 位

110：24 位

111：32 位

位 4 保留，必须保持复位值。

位 3:2 **PRTCFG[1:0]**：协议配置 (Protocol configuration)

这些位由软件置 1 和清零，必须在音频模块禁止的情况下配置这些位。

00：自由协议。通过设置大部分配置寄存器位以及帧配置寄存器，自由协议允许用户使用音频模块这一强大的配置功能来处理特定的音频协议（如 I2S、LSB/MSB 对齐、TDM、PCM/DSP...）。

01：SPDIF 协议

10：AC’97 协议

11：保留

位 1:0 **MODE[1:0]**：SAIx 音频模块模式 (SAIx audio block mode)

这些位由软件置 1 和清零，必须在 SAIx 音频模块禁止的情况下配置这些位。

00：主发送器

01：主接收器

10：从发送器

11：从接收器

注意：如果将音频模块配置为 SPDIF 模式，则将强制设置主发送器模式 (MODE[1:0] = 00)。在主发送器模式下，音频模块将立即开始生成 FS 和时钟。

51.6.4 SAI 配置寄存器 2 (SAI_ACR2)

SAI configuration register 2

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP[1:0]		CPL	MUTECNT[5:0]						MUTE VAL	MUTE	TRIS	F FLUSH	FTH[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:14 **COMP[1:0]**：压扩模式 (Companding mode)

这些位由软件置 1 和清零，μ-Law 和 A-Law 算法是 CCITT G.711 建议的一部分，要使用何种补码类型取决于 *CPL 位*。

数据扩展还是数据压缩由 MODE[0] 位的状态确定。

如果将音频模块配置为发送器，则应用数据压缩。

如果将音频模块配置为接收器，则自动应用数据扩展。

更多详细信息，请参见*压扩模式*一节。

00：不支持压扩算法

01：保留。

10：μ-Law 算法

11：A-Law 算法

注意：仅当选择了自由协议模式时才能使用压扩模式。

位 13 **CPL**：补码位 (Complement bit)。

该位由软件置 1 和清零，

该位定义用于压扩模式的补码类型。

0：1 的补码表示。

1：2 的补码表示。

注意：仅当压扩模式为 μ-Law 算法或 A-Law 算法时该位才有效。

位 12:7 **MUTECNT[5:0]**：静音计数器 (Mute counter)。

这些位由软件置 1 和清零，这些位仅用于接收模式。

这些位中所设置的值将与接收模式下检测到的连续静音帧数量进行比较。当静音帧数量与该值相等时，MUTEDET 标志置 1，并且在 MUTEDETIE 位置 1 的情况下，还将生成中断。

更多详细信息，请参见*静音模式*一节。

位 6 **MUTEVAL**：静音值 (Mute value)。

该位由软件置 1 和清零，必须在使能音频模块 (SAIEN) 前写入。仅当音频模块用作发送器并且 Slot 数小于或等于 2，并且 MUTE 位已置 1 时，该位才有意义。

如果声明了 2 个以上的 Slot，则无论 MUTEVAL 位的值为何，静音模式下发送的位值都将等于 0。

如果 Slot 数小于或等于 2 且 MUTEVAL = 1，则为每个 Slot 发送的 MUTE 值将是上一帧期间发送的值。

更多详细信息，请参见*静音模式*一节。

0：静音模式期间发送位值 0。

1：静音模式期间发送上一个值。

注意：该位对 SPDIF 音频模块无意义，不得使用。

位 5 **MUTE**：静音 (Mute)。

该位由软件置 1 和清零，仅当音频模块用作发送器时该位才有意义。Slot 数小于或等于 2 时，MUTE 值与 MUTEVAL 值相关；Slot 数大于 2 时，MUTE 值等于 0。

更多详细信息，请参见*静音模式*一节。

0：禁止静音模式。

1：使能静默模式

注意：该位对 SPDIF 音频模块无意义，不得使用。

位 4 **TRIS**：数据线的三态管理 (Tristate management on data line)。

该位由软件置 1 和清零，仅当将音频模块配置为发送器时该位才有意义。当音频模块配置为 SPDIF 模式时不使用该位。必须在 SAI 禁止时配置该位。

更多详细信息，请参见*无效 Slot 上的输出数据线管理*一节。

0：Slot 无效时，SD 输出线仍由 SAI 驱动。

1：SD 输出线将在上一个有效 Slot（下一个 Slot 无效）的最后一个数据位传输结束时释放（高阻态）。

位 3 **FFLUSH**：执行 FIFO 刷新 (FIFO flush)。

该位由软件置 1。始终读为 0。必须在 SAI 禁止时配置该位。

0：禁止 FIFO 刷新。

1：执行 FIFO 刷新。将该位编程为 1 可触发 FIFO 刷新。所有的内部 FIFO 指针（读和写）将清零。这种情况下，仍存留在 FIFO 中的数据丢失（发送或接收数据不会继续丢失）。刷新 SAI 前，必须禁止 DMA 数据流 / 中断。

位 2:0 **FTH[2:0]**：FIFO 阈值 (FIFO threshold)。

该位由软件置 1 和清零，

000：FIFO 空

001：¼ FIFO

010：½ FIFO

011：¾ FIFO

100：FIFO 满

101：保留

110：保留

111：保留

51.6.5 SAI 配置寄存器 2 (SAI_BCR2)

SAI configuration register 2

偏移地址：0x028

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP[1:0]		CPL	MUTECNT[5:0]						MUTE VAL	MUTE	TRIS	F FLUSH	FTH[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:14 **COMP[1:0]**：压扩模式 (Companding mode)

这些位由软件置 1 和清零，μ-Law 和 A-Law 算法是 CCITT G.711 建议的一部分，要使用何种补码类型取决于 *CPL 位*。

数据扩展还是数据压缩由 MODE[0] 位的状态确定。

如果将音频模块配置为发送器，则应用数据压缩。

如果将音频模块配置为接收器，则自动应用数据扩展。

更多详细信息，请参见*压扩模式*一节。

00：不支持压扩算法

01：保留。

10：μ-Law 算法

11：A-Law 算法

注意：仅当选择了自由协议模式时才能使用压扩模式。

位 13 **CPL**：补码位 (Complement bit)。

该位由软件置 1 和清零，

该位定义用于压扩模式的补码类型。

0：1 的补码表示。

1：2 的补码表示。

注意：仅当压扩模式为 μ-Law 算法或 A-Law 算法时该位才有效。

位 12:7 **MUTECNT[5:0]**：静音计数器 (Mute counter)。

这些位由软件置 1 和清零，这些位仅用于接收模式。

这些位中所设置的值将与接收模式下检测到的连续静音帧数量进行比较。当静音帧数量与该值相等时，MUTEDET 标志置 1，并且在 MUTEDETIE 位置 1 的情况下，还将生成中断。

更多详细信息，请参见*静音模式*一节。

位 6 **MUTEVAL**：静音值 (Mute value)。

该位由软件置 1 和清零，必须在使能音频模块 (SAIEN) 前写入。仅当音频模块用作发送器并且 Slot 数小于或等于 2，并且 MUTE 位已置 1 时，该位才有意义。

如果声明了 2 个以上的 Slot，则无论 MUTEVAL 位的值为何，静音模式下发送的位值都将等于 0。

如果 Slot 数小于或等于 2 且 MUTEVAL = 1，则为每个 Slot 发送的 MUTE 值将是上一帧期间发送的值。

更多详细信息，请参见*静音模式*一节。

0：静音模式期间发送位值 0。

1：静音模式期间发送上一个值。

注意：该位对 SPDIF 音频模块无意义，不得使用。

位 5 **MUTE**：静音 (Mute)。

该位由软件置 1 和清零，仅当音频模块用作发送器时该位才有意义。Slot 数小于或等于 2 时，MUTE 值与 MUTEVAL 值相关；Slot 数大于 2 时，MUTE 值等于 0。

更多详细信息，请参见*静音模式*一节。

0：禁止静音模式。

1：使能静音模式

注意：该位对 SPDIF 音频模块无意义，不得使用。

位 4 **TRIS**：数据线的三态管理 (Tristate management on data line)。

该位由软件置 1 和清零，仅当将音频模块配置为发送器时该位才有意义。当音频模块配置为 SPDIF 模式时不使用该位。必须在 SAI 禁止时配置该位。

更多详细信息，请参见*无效 Slot 上的输出数据线管理*一节。

0：Slot 无效时，SD 输出线仍由 SAI 驱动。

1：SD 输出线将在上一个有效 Slot（下一个 Slot 无效）的最后一个数据位传输结束时释放（高阻态）。

位 3 **FFLUSH**：FIFO 刷新 (FIFO flush)。

该位由软件置 1。始终读为 0。必须在 SAI 禁止时配置该位。

0：禁止 FIFO 刷新。

1：FIFO 刷新 (FIFO flush)。将该位编程为 1 可触发 FIFO 刷新。所有的内部 FIFO 指针（读和写）将清零。这种情况下，仍存留在 FIFO 中的数据丢失（发送或接收数据不会继续丢失）。刷新 SAI 前，必须禁止 DMA 数据流/中断。

位 2:0 **FTH[2:0]**：FIFO 阈值 (FIFO threshold)。

该位由软件置 1 和清零，

000：FIFO 空

001：¼ FIFO

010：½ FIFO

011：¾ FIFO

100：FIFO 满

101：保留

110：保留

111：保留

51.6.6 SAI 帧配置寄存器 (SAI_AFRCR)

SAI frame configuration register

偏移地址：0x00C

复位值：0x0000 0007

注： *该寄存器对于 AC'97 和 SPDIF 音频协议无意义。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSOFF	FSPOL	FSDEF
													rw	rw	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	FSALL[6:0]							FRL[7:0]							
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:19 保留，必须保持复位值。

位 18 **FSOFF**：帧同步偏移 (Frame synchronization offset)。

该位由软件置 1 和清零，该位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。必须在音频模块禁止的情况下配置该位。

0：在 Slot 0 的第一位上使能 FS。

1：在 Slot 0 第一位的前一位上使能 FS。

位 17 **FSPOL**：帧同步极性 (Frame synchronization polarity)。

该位由软件置 1 和清零，该位用于配置 FS 信号上的 SOF 电平。该位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。

必须在音频模块禁止的情况下配置该位。

0：FS 为低电平有效（下降沿）

1：FS 为高电平有效（上升沿）

位 16 **FSDEF**：帧同步定义 (Frame synchronization definition)。
该位由软件置 1 和清零，
0：FS 信号为起始帧信号
1：FS 信号为 SOF 信号 + 通道识别信号
该位置 1 时，SAI_xSLOTR 寄存器中定义的 Slot 数必须为偶数。这意味着有半数 Slot 用于左通道，其他 Slot 用于右通道（例如，对于 I2S 或 MSB/LSB 对齐等协议，该位必须置 1）。
该位对 AC’97 或 SPDIF 音频模块配置无意义，从而也不使用。必须在音频模块禁止的情况下配置该位。

位 15 保留，必须保持复位值。

位 14:8 **FSALL[6:0]**：帧同步有效电平长度 (Frame synchronization active level length)
这些位由软件置 1 和清零，这些位用于指定音频帧中 FS 信号的有效电平长度，以位时钟数 (SCK) + 1 (FSALL[6:0] + 1) 表示这些位对 AC’97 或 SPDIF 音频模块配置无意义，从而也不使用。
必须在音频模块禁止的情况下配置这些位。

位 7:0 **FRL[7:0]**：帧长度 (Frame length)。
这些位由软件置 1 和清零，这些位用于定义以 SCK 时钟周期数表示的音频帧长度：帧中的位数等于 FRL[7:0] + 1。
音频帧中发送的位数必须大于或等于 8，否则音频模块将出现操作异常。数据大小为 8 位且在 SAI_xSLOTR 寄存器的 NBSLOT[4:0] 中只定义了一个 Slot 0 (NBSLOT[3:0] = 0000) 时便属于这种情况。
在主模式下，如果使用主时钟（MCLK_x 引脚上提供），则帧长度必须为 8 到 256 之间的一个等于 2 的几次幂的数。不使用主时钟 (NODIV = 1) 时，建议将帧长度编程为 8 到 256 之间的值。
这些位对 AC’97 或 SPDIF 音频模块配置无意义，从而也不使用。必须在音频模块禁止的情况下配置这些位。

51.6.7 帧配置寄存器 (SAI_BFRCR)

SAI frame configuration register

偏移地址：0x02C

复位值：0x0000 0007

注： *该寄存器对于 AC’97 和 SPDIF 音频协议无意义*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSOFF	FSPOL	FSDEF
													rw	rw	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	FSALL[6:0]							FRL[7:0]							
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:19 保留，必须保持复位值。

位 18 **FSOFF**：帧同步偏移 (Frame synchronization offset)。
该位由软件置 1 和清零，该位对 AC’97 或 SPDIF 音频模块配置无意义，从而也不使用。必须在音频模块禁止的情况下配置该位。
0：在 Slot 0 的第一位上使能 FS。
1：在 Slot 0 第一位的前一位上使能 FS。

位 17 **FSPOL**：帧同步极性 (Frame synchronization polarity)。

该位由软件置 1 和清零，该位用于配置 FS 信号上的 SOF 电平。该位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。

必须在音频模块禁止的情况下配置该位。

0：FS 为低电平有效（下降沿）

1：FS 为高电平有效（上升沿）

位 16 **FSDEF**：帧同步定义 (Frame synchronization definition)。

该位由软件置 1 和清零，

0：FS 信号为起始帧信号

1：FS 信号为 SOF 信号 + 通道识别信号

该位置 1 时，SAI_xSLOTR 寄存器中定义的 Slot 数必须为偶数。这意味着有半数 Slot 用于左通道，其他 Slot 用于右通道（例如，对于 I2S 或 MSB/LSB 对齐等协议，该位必须置 1）。

该位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。必须在音频模块禁止的情况下配置该位。

位 15 保留，必须保持复位值。

位 14:8 **FSALL[6:0]**：帧同步有效电平长度 (Frame synchronization active level length)

这些位由软件置 1 和清零，这些位用于指定音频帧中 FS 信号的有效电平长度，以位时钟数 (SCK) + 1 (FSALL[6:0] + 1) 表示这些位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。

必须在音频模块禁止的情况下配置这些位。

位 7:0 **FRL[7:0]**：帧长度 (Frame length)。

这些位由软件置 1 和清零，这些位用于定义以 SCK 时钟周期数表示的音频帧长度：帧中的位数等于 FRL[7:0] + 1。

音频帧中发送的位数必须大于或等于 8，否则音频模块将出现操作异常。数据大小为 8 位且在 SAI_xSLOTR 寄存器的 NBSLOT[4:0] 中只定义了一个 Slot 0 (NBSLOT[3:0] = 0000) 时便属于这种情况。

在主模式下，如果使用主时钟（MCLK_x 引脚上提供），则帧长度必须为 8 到 256 之间的一个等于 2 的几次幂的数。不使用主时钟 (NODIV = 1) 时，建议将帧长度编程为 8 到 256 之间的值。

这些位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。

51.6.8 SAI Slot 寄存器 (SAI_ASLOTR)

SAI slot register

偏移地址：0x010

复位值：0x0000 0000

注：　该寄存器对于 AC'97 和 SPDIF 音频协议无意义。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOTEN[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	NBSLOT[3:0]				SLOTSZ[1:0]		Res.	FBOFF[4:0]				
				rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:16 **SLOTEN[15:0]**：Slot 使能 (Slot enable)。
这些位由软件置 1 和清零，
每个 SLOTEN 位都对应于从 0 到 15 的一个 Slot 位置（最多 16 个 Slot）。
0：无效 Slot。
1：有效 Slot。
必须在音频模块禁止的情况下使能 Slot。
在 AC'97 或 SPDIF 模式下会忽略这些位。

位 15:12 保留，必须保持复位值。

位 11:8 **NBSLOT[3:0]**：音频帧中的 Slot 数 (Number of slots in an audio frame)。
这些位由软件置 1 和清零，
该位字段中设置的值表示音频帧中的 Slot 数 + 1（包括无效 Slot 数）。Slot 数最大值为 16。
SAI_xFRCR 寄存器中的 FSDEF 位置 1 时，Slot 数必须为偶数。
必须在音频模块禁止的情况下配置 Slot 数。
在 AC'97 或 SPDIF 模式下会忽略这些位。

位 7:6 **SLOTSZ[1:0]**：Slot 大小 (Slot size)
该位由软件置 1 和清零。
Slot 大小必须大于或等于数据大小。如果不满足该条件，SAI 的行为将不确定。
有关如何驱动 SD 线的信息，请参见*无效 Slot 上的输出数据线管理*。
必须在音频模块禁止的情况下配置这些位。
在 AC'97 或 SPDIF 模式下会忽略这些位。
00：Slot 大小与数据大小（在 SAI_xCR1 寄存器的 DS[3:0] 位中指定）相当。
01：16 位
10：32 位
11：保留

位 5 保留，必须保持复位值。

位 4:0 **FBOFF[4:0]**：第一位偏移
这些位由软件置 1 和清零，
该位字段中设置的值定义 Slot 中第一个数据传输位的位置。它表示一个偏移值。在发送模式下，此数据位字段以外的位将强制清零。在接收模式下，将丢弃额外接收的位。
必须在音频模块禁止的情况下配置这些位。
在 AC'97 或 SPDIF 模式下会忽略这些位。

51.6.9 SAI Slot 寄存器 (SAI_BSLOTR)

SAI slot register

偏移地址：0x030

复位值：0x0000 0000

注： *该寄存器对于 AC'97 和 SPDIF 音频协议无意义。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOTEN[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	NBSLOT[3:0]				SLOTSZ[1:0]		Res.	FBOFF[4:0]				
				rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:16 **SLOTEN[15:0]**：Slot 使能 (Slot enable)。
这些位由软件置 1 和清零，
每个 SLOTEN 位都对应于从 0 到 15 的一个 Slot 位置（最多 16 个 Slot）。
0：无效 Slot。
1：有效 Slot。
必须在音频模块禁止的情况下使能 Slot。
在 AC'97 或 SPDIF 模式下会忽略这些位。

位 15:12 保留，必须保持复位值。

位 11:8 **NBSLOT[3:0]**：音频帧中的 Slot 数 (Number of slots in an audio frame)。
这些位由软件置 1 和清零，
该位字段中设置的值表示音频帧中的 Slot 数 + 1（包括无效 Slot 数）。Slot 数最大值为 16。
SAI_xFRCR 寄存器中的 FSDEF 位置 1 时，Slot 数必须为偶数。
必须在音频模块禁止的情况下配置 Slot 数。
在 AC'97 或 SPDIF 模式下会忽略这些位。

位 7:6 **SLOTSZ[1:0]**：Slot 大小 (Slot size)
该位由软件置 1 和清零。
Slot 大小必须大于或等于数据大小。如果不满足该条件，SAI 的行为将不确定。
有关如何驱动 SD 线的信息，请参见*无效 Slot 上的输出数据线管理*。
必须在音频模块禁止的情况下配置这些位。
在 AC'97 或 SPDIF 模式下会忽略这些位。
00：Slot 大小与数据大小（在 SAI_xCR1 寄存器的 DS[3:0] 位中指定）相当。
01：16 位
10：32 位
11：保留

位 5 保留，必须保持复位值。

位 4:0 **FBOFF[4:0]**：第一位偏移
这些位由软件置 1 和清零，
该位字段中设置的值定义 Slot 中第一个数据传输位的位置。它表示一个偏移值。在发送模式下，此数据位字段以外的位将强制清零。在接收模式下，将丢弃额外接收的位。
必须在音频模块禁止的情况下配置这些位。
在 AC'97 或 SPDIF 模式下会忽略这些位。

51.6.10 SAI 中断屏蔽寄存器 (SAI_AIM)

SAI interrupt mask register

偏移地址：0x014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDETIE	AFSDETIE	CNRDYIE	FREQIE	WCKCFGIE	MUTEDETIE	OVRUDRIE
									rw	rw	rw	rw	rw	rw	rw

位 31:7 保留，必须保持复位值。

位 6 **LFSDETIE**：帧同步滞后检测中断使能 (Late frame synchronization detection interrupt enable)。
该位由软件置 1 和清零，
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_xSR 寄存器中的 LFSDET 位置 1，则生成中断。
该位对于 AC'97、SPDIF 模式无意义；若音频模块为主模块，该位也无意义。

位 5 **AFSDETIE**：帧同步提前检测中断使能 (Anticipated frame synchronization detection interrupt enable)。
该位由软件置 1 和清零，
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_xSR 寄存器中的 AFSDET 位置 1，则生成中断。
该位对于 AC'97、SPDIF 模式无意义；若音频模块为主模块，该位也无意义。

位 4 **CNRDYIE**：编解码器未就绪中断使能 (Codec not ready interrupt enable) (AC'97)。
该位由软件置 1 和清零，
0：禁止中断
1：使能中断
若使能该中断，音频模块将在 AC'97 帧的 Slot 0 (tag0) 中检测连接到该线路的编解码器是否就绪。如果未就绪，SAI_xSR 寄存器中的 CNRDY 标志置 1，并生成中断。
仅当通过 PRTCFG[1:0] 位选择了 AC'97 模式且音频模块用作接收器时，该位才有意义。

位 3 **FREQIE**：FIFO 请求中断使能 (FIFO request interrupt enable)。
该位由软件置 1 和清零，
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_xSR 寄存器中的 FREQ 位置 1，则生成中断。
在接收器模式下，必须在 FREQIE 位置 1 前配置 MODE 位，以避免寄生中断，原因是复位后音频模块将默认用作发送器。

位 2 **WCKCFGIE**：时钟配置错误中断使能 (Wrong clock configuration interrupt enable)。
该位由软件置 1 和清零，
0：禁止中断
1：使能中断
仅在将音频模块配置为主模式 (MODE[1] = 0) 且 NODIV = 0 时才执行该位。
该位在 SAI_xSR 寄存器中的 WCKCFG 标志置 1 时生成中断。
*注意：该位仅用于*自由协议模式，*其他模式下没有意义。*

位 1 **MUTEDETIE**：静音检测中断使能 (Mute detection interrupt enable)。
该位由软件置 1 和清零，
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_xSR 寄存器中的 MUTEDET 位置 1，则生成中断。
仅当音频模块配置为以发送模式工作时该位才有意义。

位 0 **OVRUDRIE**：上溢/下溢中断使能 (Overrun/underrun interrupt enable)。
该位由软件置 1 和清零，
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_xSR 寄存器中的 OVRUDR 位置 1，则生成中断。

51.6.11 SAI 中断屏蔽寄存器 (SAI_BIM)

SAI interrupt mask register

偏移地址：0x034

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDETIE	AFSDETIE	CNRDYIE	FREQIE	WCKCFGIE	MUTEDETIE	OVRUDRIE
									rw	rw	rw	rw	rw	rw	rw

位 31:7 保留，必须保持复位值。

位 6 **LFSDETIE**：帧同步滞后检测中断使能 (Late frame synchronization detection interrupt enable)。
该位由软件置 1 和清零，
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_xSR 寄存器中的 LFSDET 位置 1，则生成中断。
该位对于 AC’97、SPDIF 模式无意义；若音频模块为主模块，该位也无意义。

位 5 **AFSDETIE**：帧同步提前检测中断使能 (Anticipated frame synchronization detection interrupt enable)。
该位由软件置 1 和清零，
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_xSR 寄存器中的 AFSDET 位置 1，则生成中断。
该位对于 AC’97、SPDIF 模式无意义；若音频模块为主模块，该位也无意义。

位 4 **CNRDYIE**：编解码器未就绪中断使能 (Codec not ready interrupt enable) (AC’97)。
该位由软件置 1 和清零，
0：禁止中断
1：使能中断
若使能该中断，音频模块将在 AC’97 帧的 Slot 0 (tag0) 中检测连接到该线路的编解码器是否就绪。如果未就绪，SAI_xSR 寄存器中的 CNRDY 标志置 1，并生成中断。
仅当通过 PRTCFG[1:0] 位选择了 AC’97 模式且音频模块用作接收器时，该位才有意义。

位 3 **FREQIE**：FIFO 请求中断使能 (FIFO request interrupt enable)。
该位由软件置 1 和清零，
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_xSR 寄存器中的 FREQ 位置 1，则生成中断。
在接收器模式下，必须在 FREQIE 位置 1 前配置 MODE 位，以避免寄生中断，原因是复位后音频模块将默认用作发送器。

位 2 **WCKCFGIE**：时钟配置错误中断使能 (Wrong clock configuration interrupt enable)。
该位由软件置 1 和清零，
0：禁止中断
1：使能中断
仅在将音频模块配置为主模式 (MODE[1] = 0) 且 NODIV = 0 时才执行该位。
该位在 SAI_xSR 寄存器中的 WCKCFG 标志置 1 时生成中断。
注意：该位仅用于自由协议模式，其他模式下没有意义。

位 1 **MUTEDETIE**：静音检测中断使能 (Mute detection interrupt enable)。
该位由软件置 1 和清零，
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_xSR 寄存器中的 MUTEDET 位置 1，则生成中断。
仅当音频模块配置为以发送模式工作时该位才有意义。

位 0 **OVRUDRIE**：上溢/下溢中断使能 (Overrun/underrun interrupt enable)。
该位由软件置 1 和清零，
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_xSR 寄存器中的 OVRUDR 位置 1，则生成中断。

51.6.12 SAI 状态寄存器 (SAI_ASR)

SAI status register

偏移地址：0x018

复位值：0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLVL[2:0]		
													r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	AFSDET	CNRDY	FREQ	WCKCFG	MUTEDET	OVRUDR
									r	r	r	r	r	r	r

位 31:19 保留，必须保持复位值。

位 18:16 **FLVL[2:0]**：FIFO 阈值 (FIFO level threshold)。
该位为只读。FIFO 阈值标志只通过硬件管理，其设置取决于 SAI 模块的配置（发送器或接收器模式）。
000：FIFO 空（发送器和接收器模式）
001：FIFO ≤ ¼ 但不为空（发送器模式），FIFO < ¼ 但不为空（接收器模式）
010：¼ < FIFO ≤ ½（发送器模式），¼ ≤ FIFO < ½（接收器模式）
011：½ < FIFO ≤ ¾（发送器模式），½ ≤ FIFO < ¾（接收器模式）
100：¾ < FIFO 但未满（发送器模式），¾ ≤ FIFO 但未满（接收器模式）
101：FIFO 满（发送器和接收器模式）
其他值：保留

位 15:7 保留，必须保持复位值。

位 6 **LFSDET**：帧同步滞后检测 (Late frame synchronization detection)。
该位为只读。
0：无错误。
1：帧同步信号未在正确的时刻出现。
仅当音频模块配置为以从模式工作时，此标志才能置 1。
不适用于 AC’97 或 SPDIF 模式。
它可在 SAI_xIM 寄存器中的 LFSDETIE 位置 1 时生成中断。
在软件将 SAI_xCLRFR 寄存器中的 CLFSDET 位置 1 时清零该标志

位 5 **AFSDET**：帧同步提前检测 (Anticipated frame synchronization detection)。
该位为只读。
0：无错误。
1：提前检测到帧同步信号。
仅当音频模块配置为以从模式工作时，此标志才能置 1。
不适用于 AC'97 或 SPDIF 模式。
它可在 SAI_xIM 寄存器中的 AFSDETIE 位置 1 时生成中断。
在软件将 SAI_xCLRFR 寄存器中的 CAFSDET 位置 1 时清零该标志。

位 4 **CNRDY**：编解码器未就绪 (Codec not ready)。
该位为只读。
0：外部 AC'97 编解码器已就绪
1：外部 AC'97 编解码器未就绪
仅当在 SAI_xCR1 寄存器中选择了 AC'97 音频模块并且音频模块配置为接收器模式时，才使用该位。
它可在 SAI_xIM 寄存器中的 CNRDYIE 位置 1 时生成中断。
在软件将 SAI_xCLRFR 寄存器中的 CCNRDY 位置 1 时清零该标志。

位 3 **FREQ**：FIFO 请求 (FIFO request)。
该位为只读。
0：无 FIFO 请求。
1：FIFO 请求读取或写入 SAI_xDR。
请求内容取决于音频模块的配置：
– 如果模块配置为发送模式，则 FIFO 请求与向 SAI_xDR 中写入相关。
– 如果音频模块配置为接收模式，则 FIFO 请求与从 SAI_xDR 中读取相关。
此标志可在 SAI_xIM 寄存器中的 FREQIE 位置 1 时生成中断。

位 2 **WCKCFG**：时钟配置错误标志 (Wrong clock configuration flag)。
该位为只读。
0：时钟配置正确
1：时钟配置不符合*第 51.4.6 节：帧同步*中定义的帧长度规范（SAI_xFRCR 寄存器中 FRL[7:0] 位的配置）
该位仅在音频模块工作在主模式 (MODE[1] = 0) 下且 NODIV = 0 时使用。
它可在 SAI_xIM 寄存器中的 WCKCFGIE 位置 1 时生成中断。
在软件将 SAI_xCLRFR 寄存器中的 CWCKCFG 位置 1 时清零该标志。

位 1 **MUTEDET**：静音检测 (Mute detection)。
该位为只读。
0：SD 输入线上未检测到 MUTE 值
1：在 SD 输入线上检测到指定数量的连续音频帧中的 MUTE 值（0 值）
如果在指定音频帧的每个 Slot 或在一定数量（在 SAI_xCR2 寄存器中的 MUTECNT 位中设置）的连续音频帧中接收到连续的 0 值，则该标志置 1。
它可在 SAI_xIM 寄存器中的 MUTEDETIE 位置 1 时生成中断。
在软件将 SAI_xCLRFR 寄存器中的 CMUTEDET 位置 1 时清零该标志。

位 0 **OVRUDR**：上溢/下溢 (Overrun / underrun)。
该位为只读。
0：无上溢/下溢错误。
1：检测到上溢/下溢错误。
仅当音频模块分别被配置为接收器和发送器时才会发生上溢和下溢情况。
它可在 SAI_xIM 寄存器中的 OVRUDRIE 位置 1 时生成中断。
在软件将 SAI_xCLRFR 寄存器中的 COVRUDR 位置 1 时清零该标志。

51.6.13 SAI 状态寄存器 (SAI_BSR)

SAI status register

偏移地址：0x038

复位值：0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLVL[2:0]		
													r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	AFSDET	CNRDY	FREQ	WCKCFG	MUTEDET	OVRUDR
									r	r	r	r	r	r	r

位 31:19 保留，必须保持复位值。

位 18:16 **FLVL[2:0]**：FIFO 阈值 (FIFO level threshold)。

该位为只读。FIFO 阈值标志只通过硬件管理，其设置取决于 SAI 模块的配置（发送器或接收器模式）。

000：FIFO 空（发送器和接收器模式）

001：FIFO ≤ ¼ 但不为空（发送器模式），FIFO < ¼ 但不为空（接收器模式）

010：¼ < FIFO ≤ ½（发送器模式），¼ ≤ FIFO < ½（接收器模式）

011：½ < FIFO ≤ ¾（发送器模式），½ ≤ FIFO < ¾（接收器模式）

100：¾ < FIFO 但未满（发送器模式），¾ ≤ FIFO 但未满（接收器模式）

101：FIFO 满（发送器和接收器模式）

其他值：保留

位 15:7 保留，必须保持复位值。

位 6 **LFSDET**：帧同步滞后检测 (Late frame synchronization detection)。

该位为只读。

0：无错误。

1：帧同步信号未在正确的时刻出现。

仅当音频模块配置为以从模式工作时，此标志才能置 1。

不适用于 AC'97 或 SPDIF 模式。

它可在 SAI_xIM 寄存器中的 LFSDETIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CLFSDET 位置 1 时清零该标志

位 5 **AFSDET**：帧同步提前检测 (Anticipated frame synchronization detection)。

该位为只读。

0：无错误。

1：提前检测到帧同步信号。

仅当音频模块配置为以从模式工作时，此标志才能置 1。

不适用于 AC'97 或 SPDIF 模式。

它可在 SAI_xIM 寄存器中的 AFSDETIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CAFSDET 位置 1 时清零该标志。

位 4 **CNRDY**：编解码器未就绪 (Codec not ready)。

该位为只读。

0：外部 AC'97 编解码器已就绪

1：外部 AC'97 编解码器未就绪

仅当在 SAI_xCR1 寄存器中选择了 AC'97 音频模块并且音频模块配置为接收器模式时，才使用该位。

它可在 SAI_xIM 寄存器中的 CNRDYIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CCNRDY 位置 1 时清零该标志。

位 3 **FREQ**：FIFO 请求 (FIFO request)。

该位为只读。

0：无 FIFO 请求。

1：FIFO 请求读取或写入 SAI_xDR。

请求内容取决于音频模块的配置：

– 如果模块配置为发送模式，则 FIFO 请求与向 SAI_xDR 中写入相关。

– 如果音频模块配置为接收模式，则 FIFO 请求与从 SAI_xDR 中读取相关。

此标志可在 SAI_xIM 寄存器中的 FREQIE 位置 1 时生成中断。

位 2 **WCKCFG**：时钟配置错误标志 (Wrong clock configuration flag)。

该位为只读。

0：时钟配置正确

1：时钟配置不符合*第 51.4.6 节：帧同步*中定义的帧长度规范（SAI_xFRCR 寄存器中 FRL[7:0] 位的配置）

该位仅在音频模块工作在主模式 (MODE[1] = 0) 下且 NODIV = 0 时使用。

它可在 SAI_xIM 寄存器中的 WCKCFGIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CWCKCFG 位置 1 时清零该标志。

位 1 **MUTEDET**：静音检测 (Mute detection)。

该位为只读。

0：SD 输入线上未检测到 MUTE 值

1：在 SD 输入线上检测到指定数量的连续音频帧中的 MUTE 值（0 值）

如果在指定音频帧的每个 Slot 或在一定数量（在 SAI_xCR2 寄存器中的 MUTECNT 位中设置）的连续音频帧中接收到连续的 0 值，则该标志置 1。

它可在 SAI_xIM 寄存器中的 MUTEDETIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CMUTEDET 位置 1 时清零该标志。

位 0 **OVRUDR**：上溢/下溢 (Overrun / underrun)。

该位为只读。

0：无上溢/下溢错误。

1：检测到上溢/下溢错误。

仅当音频模块分别被配置为接收器和发送器时才会发生上溢和下溢情况。

它可在 SAI_xIM 寄存器中的 OVRUDRIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 COVRUDR 位置 1 时清零该标志。

51.6.14 SAI 清零标志寄存器 (SAI_ACLRFR)

SAI clear flag register

偏移地址：0x01C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLFSDET	CAFSDET	CCNRDY	Res.	CWCKCFG	CMUTEDET	COVRUDR
									w	w	w		w	w	w

位 31:7 保留，必须保持复位值。

位 6 **CLFSDET**：清零帧同步滞后检测标志 (Clear late frame synchronization detection flag)。
该位为只写位。
将该位编程为 1 可清零 SAI_xSR 寄存器中的 LFSDET 标志。
该位不适用于 AC'97 或 SPDIF 模式
读取该位将始终返回值 0。

位 5 **CAFSDET**：清零帧同步提前检测标志 (Clear anticipated frame synchronization detection flag)。
该位为只写位。
将该位编程为 1 可清零 SAI_xSR 寄存器中的 AFSDET 标志。
不适用于 AC'97 或 SPDIF 模式。
读取该位将始终返回值 0。

位 4 **CCNRDY**：清零编解码器未就绪标志 (Clear codec not ready flag)。
该位为只写位。
将该位编程为 1 可清零 SAI_xSR 寄存器中的 CNRDY 标志。
仅当在 SAI_xCR1 寄存器中选择了 AC'97 音频协议时，才使用该位。
读取该位将始终返回值 0。

位 3 保留，必须保持复位值。

位 2 **CWCKCFG**：清零时钟配置错误标志 (Clear wrong clock configuration flag)。
该位为只写位。
将该位编程为 1 可清零 SAI_xSR 寄存器中的 WCKCFG 标志。
仅当音频模块设置为主模块时 (MODE[1] = 0) 且 SAI_xCR1 寄存器中的 NODIV = 0 时，才使用该位。
读取该位将始终返回值 0。

位 1 **CMUTEDET**：静音检测标志 (Mute detection flag)。
该位为只写位。
将该位编程为 1 可清零 SAI_xSR 寄存器中的 MUTEDET 标志。
读取该位将始终返回值 0。

位 0 **COVRUDR**：清零上溢/下溢标志 (Clear overrun/underrun)。
该位为只写位。
将该位编程为 1 可清零 SAI_xSR 寄存器中的 OVRUDR 标志。
读取该位将始终返回值 0。

51.6.15 SAI 清零标志寄存器 (SAI_BCLRFR)

SAI clear flag register

偏移地址：0x03C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLFSDET	CAFSDET	CCNRDY	Res.	CWCKCFG	CMUTEDET	COVRUDR
									w	w	w		w	w	w

位 31:7 保留，必须保持复位值。

位 6 **CLFSDET**：清零帧同步滞后检测标志 (Clear late frame synchronization detection flag)。
该位为只写位。
将该位编程为 1 可清零 SAI_xSR 寄存器中的 LFSDET 标志。
该位不适用于 AC'97 或 SPDIF 模式
读取该位将始终返回值 0。

位 5 **CAFSDET**：清零帧同步提前检测标志 (Clear anticipated frame synchronization detection flag)。
该位为只写位。
将该位编程为 1 可清零 SAI_xSR 寄存器中的 AFSDET 标志。
不适用于 AC'97 或 SPDIF 模式。
读取该位将始终返回值 0。

位 4 **CCNRDY**：清零编解码器未就绪标志 (Clear codec not ready flag)。
该位为只写位。
将该位编程为 1 可清零 SAI_xSR 寄存器中的 CNRDY 标志。
仅当在 SAI_xCR1 寄存器中选择了 AC'97 音频协议时，才使用该位。
读取该位将始终返回值 0。

位 3 保留，必须保持复位值。

位 2 **CWCKCFG**：清零时钟配置错误标志 (Clear wrong clock configuration flag)。
该位为只写位。
将该位编程为 1 可清零 SAI_xSR 寄存器中的 WCKCFG 标志。
仅当音频模块设置为主模块时 (MODE[1] = 0) 且 SAI_xCR1 寄存器中的 NODIV = 0 时，才使用该位。
读取该位将始终返回值 0。

位 1 **CMUTEDET**：静音检测标志 (Mute detection flag)。
该位为只写位。
将该位编程为 1 可清零 SAI_xSR 寄存器中的 MUTEDET 标志。
读取该位将始终返回值 0。

位 0 **COVRUDR**：清零上溢/下溢标志 (Clear overrun/underrun)。
该位为只写位。
将该位编程为 1 可清零 SAI_xSR 寄存器中的 OVRUDR 标志。
读取该位将始终返回值 0。

51.6.16 SAI 数据寄存器 (SAI_ADR)

SAI data register

偏移地址：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DATA[31:0]**：数据 (Data)
若 FIFO 未满，写入该寄存器可向 FIFO 加载数据。
若 FIFO 非空，读取该寄存器可清空 FIFO。

51.6.17 SAI 数据寄存器 (SAI_BDR)

SAI data register

偏移地址：0x040

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DATA[31:0]**：数据 (Data)
若 FIFO 未满，写入该寄存器可向 FIFO 加载数据。
若 FIFO 非空，读取该寄存器可清空 FIFO。

51.6.18 SAI PDM 控制寄存器 (SAI_PDMCR)

SAI PDM control register

偏移地址：0x0044

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CKEN4	CKEN3	CKEN2	CKEN1	Res.	Res.	MICNBR[1:0]		Res.	Res.	Res.	PDMEN
				rw	rw	rw	rw			rw	rw				rw

位 31:16 保留，必须保持复位值。

位 15:12 保留，必须保持复位值。

位 11 **CKEN4**：4 号比特流时钟使能 (Clock enable of bitstream clock number 4)
该位由软件置 1 和清零，
0：禁止 SAI_CK4 时钟
1：使能 SAI_CK4 时钟
注意：建议不要在 PDMEN = 1 时配置该位。
SAI_CK4 可能并非在所有 SAI 实例中均提供。有关详细信息，请参见第 51.3 节：SAI 实现。

位 10 **CKEN3**：3 号比特流时钟使能 (Clock enable of bitstream clock number 3)
该位由软件置 1 和清零，
0：禁止 SAI_CK3 时钟
1：使能 SAI_CK3 时钟
注意：建议不要在 PDMEN = 1 时配置该位。
SAI_CK3 可能并非在所有 SAI 实例中均提供。有关详细信息，请参见第 51.3 节：SAI 实现。

位 9 **CKEN2**：2 号比特流时钟使能 (Clock enable of bitstream clock number 2)
该位由软件置 1 和清零，
0：禁止 SAI_CK2 时钟
1：使能 SAI_CK2 时钟
注意：建议不要在 PDMEN = 1 时配置该位。
SAI_CK2 可能并非在所有 SAI 实例中均提供。有关详细信息，请参见第 51.3 节：SAI 实现。

位 8 **CKEN1**：1 号比特流时钟使能 (Clock enable of bitstream clock number 1)
该位由软件置 1 和清零，
0：禁止 SAI_CK1 时钟
1：使能 SAI_CK1 时钟
注意：建议不要在 PDMEN = 1 时配置该位。
SAI_CK1 可能并非在所有 SAI 实例中均提供。有关详细信息，请参见第 51.3 节：SAI 实现。

位 7:6 保留，必须保持复位值。

位 5:4 **MICNBR[1:0]**：麦克风数量 (Number of microphones)
该位由软件置 1 和清零，
00：配置有 2 个麦克风
01：配置有 4 个麦克风
10：配置有 6 个麦克风
11：配置有 8 个麦克风
*注意：建议不要在 PDMEN = 1 时配置该位字段。**
完整的数据线集可能并非在所有 SAI 实例中均提供。有关详细信息，请参见第 51.3 节：SAI 实现。

位 3:1 保留，必须保持复位值。

位 0 **PDMEN**：PDM 使能 (PDM enable)
该位由软件置 1 和清零，该位可用于控制 PDM 接口模块的状态。
在使能 PDM 接口之前，确保 SAI 已工作在 TDM 主模式下。
0：禁止 PDM 接口
1：使能 PDM 接口

51.6.19 SAI PDM 延迟寄存器 (SAI_PDMDLY)

SAI PDM delay register

偏移地址：0x0048

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DLYM4R[2:0]			Res.	DLYM4L[2:0]			Res.	DLYM3R[2:0]			Res.	DLYM3L[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DLYM2R[2:0]			Res.	DLYM2L[2:0]			Res.	DLYM1R[2:0]			Res.	DLYM1L[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

位 31 保留，必须保持复位值。

位 30:28 **DLYM4R[2:0]**：**第 4 对**的第二个麦克风的延迟线 (Delay line for second microphone of **pair 4**)

该位由软件置 1 和清零，

000：无延迟

001：延迟 1 个 T_{SAI_CK} 周期

010：延迟 2 个 T_{SAI_CK} 周期

...

111：延迟 7 个 T_{SAI_CK} 周期

该位字段可实时更改。

注意：只有在 D4 线可用时才能使用该位字段。请参见第 51.3 节：SAI 实现以确认是否可用。

位 27 保留，必须保持复位值。

位 26:24 **DLYM4L[2:0]**：第 4 对的第一个麦克风的延迟线 (Delay line for first microphone of pair 4)

该位由软件置 1 和清零，

000：无延迟

001：延迟 1 个 T_{SAI_CK} 周期

010：延迟 2 个 T_{SAI_CK} 周期

...

111：延迟 7 个 T_{SAI_CK} 周期

该位字段可实时更改。

注意：只有在 D4 线可用时才能使用该位字段。请参见第 51.3 节：SAI 实现以确认是否可用。

位 23 保留，必须保持复位值。

位 22:20 **DLYM3R[2:0]**：第 3 对的第二个麦克风的延迟线 (Delay line for second microphone of pair 3)

该位由软件置 1 和清零，

000：无延迟

001：延迟 1 个 T_{SAI_CK} 周期

010：延迟 2 个 T_{SAI_CK} 周期

...

111：延迟 7 个 T_{SAI_CK} 周期

该位字段可实时更改。

注意：只有在 D3 线可用时才能使用该位字段。请参见第 51.3 节：SAI 实现以确认是否可用。

位 19 保留，必须保持复位值。

位 18:16 **DLYM3L[2:0]**：第 3 对的第一个麦克风的延迟线 (Delay line for first microphone of pair 3)

该位由软件置 1 和清零，

000：无延迟

001：延迟 1 个 T_{SAI_CK} 周期

010：延迟 2 个 T_{SAI_CK} 周期

...

111：延迟 7 个 T_{SAI_CK} 周期

该位字段可实时更改。

注意：只有在 D3 线可用时才能使用该位字段。请参见第 51.3 节：SAI 实现以确认是否可用。

位 15 保留，必须保持复位值。

位 14:12 **DLYM2R[2:0]**：第 2 对的第二个麦克风的延迟线 (Delay line for second microphone of pair 2)

该位由软件置 1 和清零，

000：无延迟

001：延迟 1 个 T_{SAI_CK} 周期

010：延迟 2 个 T_{SAI_CK} 周期

...

111：延迟 7 个 T_{SAI_CK} 周期

该位字段可实时更改。

注意：只有在 D2 线可用时才能使用该位字段。请参见第 51.3 节：SAI 实现以确认是否可用。

位 11 保留，必须保持复位值。

位 10:8 **DLYM2L[2:0]**：第 2 对的第一个麦克风的延迟线 (Delay line for first microphone of pair 2)

该位由软件置 1 和清零，

000：无延迟

001：延迟 1 个 T_{SAI_CK} 周期

010：延迟 2 个 T_{SAI_CK} 周期

...

111：延迟 7 个 T_{SAI_CK} 周期

该位字段可实时更改。

注意：只有在 D2 线可用时才能使用该位字段。请参见第 51.3 节：SAI 实现以确认是否可用。

位 7 保留，必须保持复位值。

位 6:4 **DLYM1R[2:0]**：第 1 对的第二个麦克风的延迟线调整 (Delay line adjust for second microphone of pair 1)

该位由软件置 1 和清零，

000：无延迟

001：延迟 1 个 T_{SAI_CK} 周期

010：延迟 2 个 T_{SAI_CK} 周期

...

111：延迟 7 个 T_{SAI_CK} 周期

该位字段可实时更改。

注意：只有在 D1 线可用时才能使用该位字段。请参见第 51.3 节：SAI 实现以确认是否可用。

位 3 保留，必须保持复位值。

位 2:0 **DLYM1L[2:0]**：第 1 对的第一个麦克风的延迟线调整 (Delay line adjust for first microphone of pair 1)

该位由软件置 1 和清零，

000：无延迟

001：延迟 1 个 T_{SAI_CK} 周期

010：延迟 2 个 T_{SAI_CK} 周期

...

111：延迟 7 个 T_{SAI_CK} 周期

该位字段可实时更改。

注意：只有在 D1 线可用时才能使用该位字段。请参见第 51.3 节：SAI 实现以确认是否可用。

51.6.20 SAI 硬件配置寄存器 (SAI_HWCFGR)

SAI hardware configuration register

偏移地址：0x03F0

复位值：0x0000 0108

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPTION_REGOUT[7:4]			
												r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTION_REGOUT[3:0]				SPDIF_PDM[3:0]				FIFO_SIZE[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:20 保留，必须保持复位值。

位 19:12 **OPTION_REGOUT[7:0]**：SAI_IOR 支持状态寄存器

0：未实现 SAI_IOR 寄存器。

1：已实现 SAI_IOR 寄存器，仅位置 0 有 1 个控制位。

2：已实现 SAI_IOR 寄存器，位置 0 和 1 有 2 个控制位。

...

16：已实现 SAI_IOR 寄存器，位置 0 到 15 有 16 个控制位。

位 11:8 **SPDIF_PDM[3:0]**：SPDIF_OUT 和 PDM 接口支持状态

0：SPDIF-OUT 和 PDM 接口不可用。

1：SPDIF-OUT 和 PDM 接口可用。

位 7:0 **FIFO_SIZE[7:0]**：SAIA 和 SAIB 的 FIFO 大小

4：FIFO 深度为 4 个字。

8：FIFO 深度为 8 个字。

12：FIFO 深度为 12 个字。

16：FIFO 深度为 16 个字。

20：FIFO 深度为 20 个字。

24：FIFO 深度为 24 个字。

28：FIFO 深度为 28 个字。

32：FIFO 深度为 32 个字。

其他值：无效值

51.6.21 SAI 版本寄存器 (SAI_VERR)

SAI version register

偏移地址：0x03F4

复位值：0x0000 0021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：SAI 主版本 (SAI major revision)

位 3:0 **MINREV[3:0]**：SAI 次版本 (SAI minor revision)

51.6.22 SAI 标识寄存器 (SAI_IPIDR)

SAI identification register

偏移地址：0x03F8

复位值：0x0013 0031

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：SAI 标识符 (SAI identifier)

51.6.23 SAI 存储空间标识寄存器 (SAI_SIDR)

SAI size identification register

偏移地址：0x03FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：存储空间大小标识 (Size identification)

这些位返回分配给 SAI 寄存器的存储区域的大小。

该存储区域的大小为 1 KB。

51.6.24 SAI 寄存器映射

表 466. SAI 寄存器映射和复位值

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	**SAI_GCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYNCOUT[1:0]		Res.	Res.	SYNCIN[1:0]	
	Reset value																											0	0			0	0
0x0004 or 0x0024	**SAI_xCR1**	Res.	Res.	Res.	Res.	MCKEN	OSR	MCKDIV[5:0]						NODIV	Res.	DMAEN	SAIEN	Res.	Res.	OUTDRIV	MONO	SYNCEN[1:0]		CKSTR	LSBFIRST	DS[2:0]		Res.	PRTCFG[1:0]		MODE[1:0]	
	Reset value					0	0	0	0	0	0	0	0	0		0	0			0	0	0	0	0	0	0	1	0		0	0	0	0
0x0008 or 0x0028	**SAI_xCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP[1:0]		CPL	MUTECN[5:0]							MUTE VAL	MUTE	TRIS	FFLUS	FTH[2:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x000C or 0x002C	**SAI_xFRCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSOFF	FSPOL	FSDEF	Res.	FSALL[6:0]							FRL[7:0]							
	Reset value														0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0x0010 or 0x0030	**SAI_xSLOTR**	SLOTEN[15:0]																Res.	Res.	Res.	Res.	NBSLOT[3:0]				SLOTSZ[1:0]		Res.	FBOFF[4:0]				
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0		0	0	0	0	0

表 466. SAI 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0014 or 0x0034	**SAI_xIM**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDETIE	AFSDETIE	CNRDYIE	FREQIE	WCKCFGIE	MUTEDETIE	OVRUDRIE
	Reset value																										0	0	0	0	0	0	0
0x0018 or 0x0038	**SAI_xSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLVL[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	AFSDET	CNRDY	FREQ	WCKCFG	MUTEDET	OVRUDR
	Reset value														0	0	0										0	0	0	0	1	0	0
0x001C or 0x003C	**SAI_xCLRFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLFSDET	CAFSDET	CCNRDY	Res.	CWCKCFG	CMUTEDET	COVRUDR
	Reset value																										0	0	0		0	0	0
0x0020 or 0x0040	**SAI_xDR**	DATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	**SAI_PDMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKEN4	CKEN3	CKEN2	CKEN1	Res.	Res.	MICNBR[1:0]		Res.	Res.	Res.	PDMEN
	Reset value																					0	0	0	0			0	0				0
0x0048	**SAI_PDMDLY**	Res.	DLYM4R[2:0]			Res.	DLYM4L[2:0]			Res.	DLYM3R[2:0]			Res.	DLYM3L[2:0]			Res.	DLYM2R[2:0]			Res.	DLYM2L[2:0]			Res.	DLYM1R[2:0]			Res.	DLYM1L[2:0]		
	Reset value		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0
0x004C to 0x03EC	Reserved																																
0x03F0	**SAI_HWCFGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPTION_REGOUT[7:0]								SPDIF_ PDM[3:0]				FIFO_SIZE[7:0]							
	Reset value													0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
0x03F4	**SAI_VERR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV [3:0]				MINREV [3:0]			
	Reset value	0x0000 0021																															
0x03F8	**SAI_IPIDR**	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1
0x03FC	**SAI_SIDR**	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 2.5 节*。

52 SPDIF 接收器接口 (SPDIFRX)

52.1 SPDIFRX 接口简介

SPDIFRX 接口处理 S/PDIF 音频协议。

52.2 SPDIFRX 主要特性

- 提供多达 4 路输入
- 自动符号率检测
- 最大符号率：12.288 MHz
- 支持 8 kHz 到 192 kHz[a] 的立体声
- 支持 IEC-60958 和 IEC-61937 音频标准，消费类应用
- 奇偶校验位管理
- 使用 DMA 通信进行音频采样
- 对于控制和用户信息采用 DMA 通信
- 中断功能

52.3 SPDIFRX 功能描述

SPDIFRX 外设用于接收符合 IEC-60958 和 EC-61937 标准的 S/PDIF 流。这些标准支持高采样率的简单立体声以及压缩的多通道环绕声，例如由 Dolby 或 DTS 定义的音频。

接收器提供所有必要的功能来检测符号率并解码传入的数据。这使用户可以使用专用通路和通道信息来简化接口的处理操作。*图 648* 给出了简化框图。

SPDIFRX_DC 模块负责解码从 SPDIFRX_IN[3:0] 输入接收的 S/PDIF 数据流。该模块重新采样传入的信号，解码曼彻斯特数据流，并识别帧、子帧和块元素。它传送到 REG_IF 部分、解码数据和相关的状态标志。

该外设可通过 APB 总线完全控制并且能够处理两个 DMA 通道：

- 一个专用于传输音频采样的 DMA 通道
- 一个专用于传输 IEC60958 通道状态和用户信息的 DMA 通道

此外，还提供了中断服务，既可用作 DMA 的复用功能，也可用来指示外设的错误或关键状态。

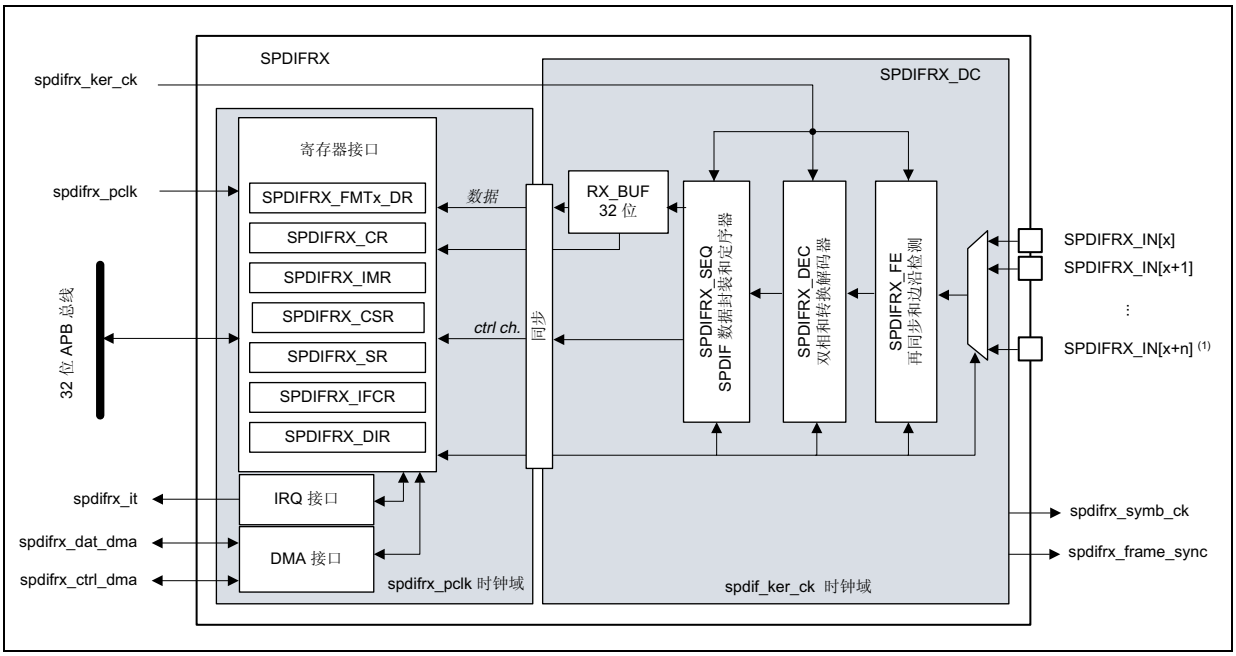
SPDIFRX 还提供名为 spdifrx_frame_sync 的信号，该信号会在每次检测到子帧的前导码时翻转。因此占空比为 50%，频率等于帧速率。

该信号可与定时器事件相连，以计算频率偏移。

a. 检查 RCC 功能以确认可以支持的采样率。

此外，SPDIFRX 还提供了一个以符号率翻转的信号，名为 **spdifrx_symb_ck**。

图 648. SPDIFRX 框图



1. “n”固定为 3，“x”设置为 0。

52.3.1 SPDIFRX 引脚和内部信号

表 467 列出了 SPDIFRX 内部输入/输出信号，*表 468* 列出了 SPDIFRX 引脚（复用功能）。

表 467. SPDIFRX 内部输入/输出信号

信号名称	信号类型	说明
spdifrx_ker_ck	数字输入	SPDIFRX 内核时钟
spdifrx_pclk	数字输入	SPDIFRX 寄存器接口时钟
spdifrx_it	数字输出	SPDIFRX 全局中断
spdifrx_dat_dma	数字输入/输出	SPDIFRX DMA 数据传输请求（和应答）
spdifrx_ctrl_dma	数字输入/输出	SPDIFRX DMA 通道状态和用户信息传输请求（和应答）
spdifrx_frame_sync	数字输出	SPDIFRX 帧速率同步信号
spdifrx_symb_ck	数字输出	SPDIFRX 通道符号时钟

表 468. SPDIFRX 引脚

信号名称	信号类型	说明
SPDIFRX_IN0	数字输入	S/PDIF 信号输入 0
SPDIFRX_IN1	数字输入	S/PDIF 信号输入 1
SPDIFRX_IN2	数字输入	S/PDIF 信号输入 2
SPDIFRX_IN3	数字输入	S/PDIF 信号输入 3

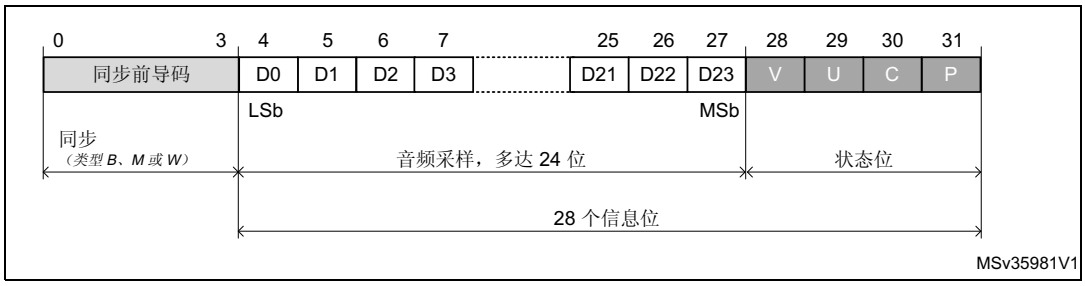
52.3.2 S/PDIF 协议 (IEC-60958)

S/PDIF 块

一个 S/PDIF 帧由两个子帧组成（见*图 649*）。每个子帧包含 32 位（或时隙）：

- 位 0 到位 3 包含同步前导码之一
- 位 4 到位 27 包含以线性 2 的补码表示的音频采样字。最高有效位 (MSB) 为位 27。使用 20 位编码范围时，位 8 到位 27 包含音频采样字，其中位 8 为 LSB。
- 位 28（有效性位“V”）表示数据是否有效（例如转换为模拟数据）
- 位 29（用户数据位“U”）包含用户数据信息，如光盘的音轨编号。
- 位 30（通道状态位“C”）包含通道状态信息，如采样率和复制保护。
- 位 31（奇偶检验位“P”）包含奇偶校验位，这样位 4 到位 31（含）将包含偶数个“1”和偶数个“0”（偶校验）。

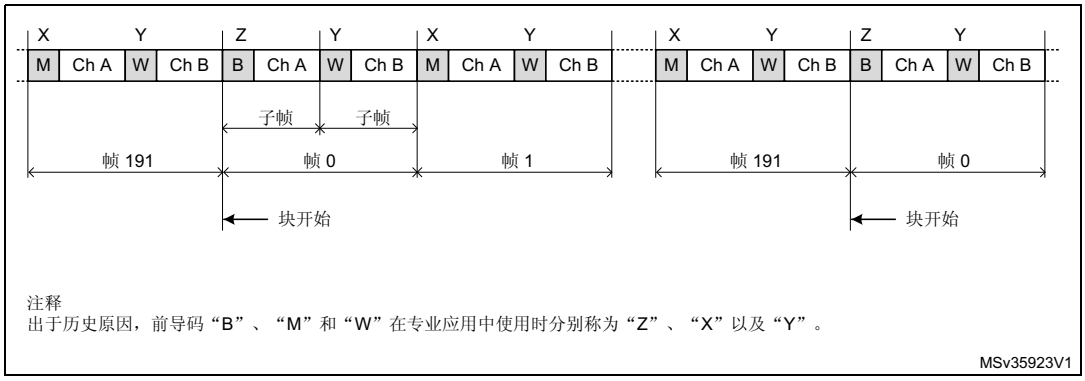
图 649. S/PDIF 子帧格式



对于线性编码音频应用，第一个子帧（立体声操作中的左声道或“A”通道以及单声道操作中的主通道）通常以前导码“M”开始。但是，前导码每 192 帧切换为前导码“B”一次，以识别用于组织通道状态和用户信息的块结构的开始。第二个子帧（立体声操作中的右声道或“B”通道以及单声道操作中的辅助通道）始终以前导码“W”开始。

一个 S/PDIF 块包含 192 对 32 位子帧。

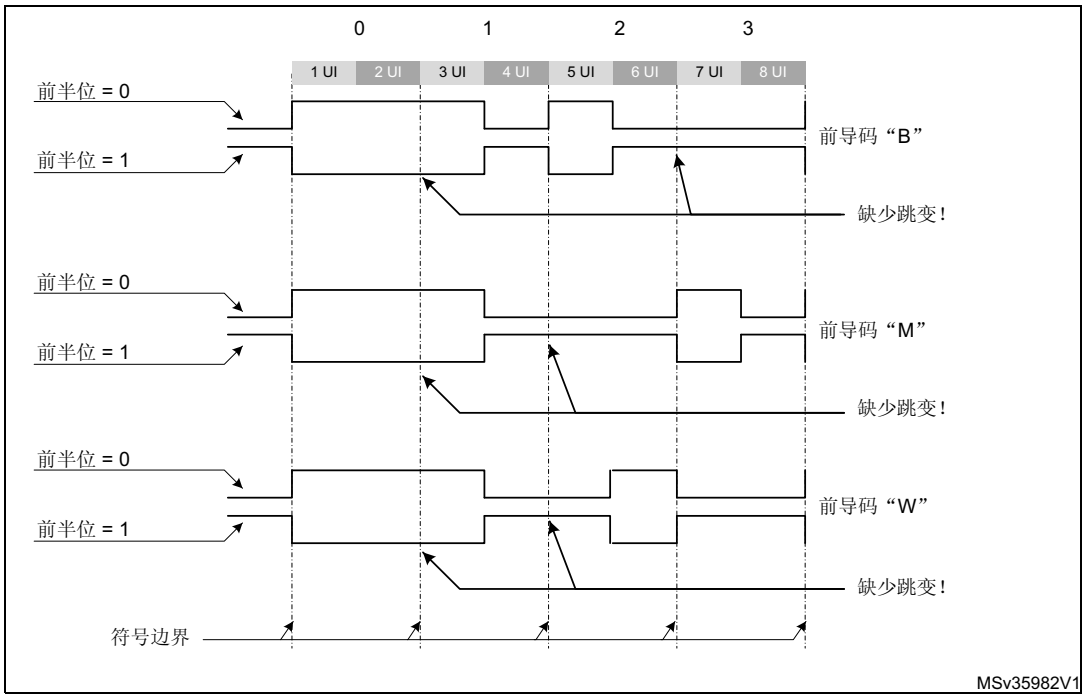
图 650. S/PDIF 块格式



同步前导码

前导码模式反转或者不符合前半位值。使能第一个帧的第一个“B”前导码的传输前，此前半位值为线路的电平。对于其它前导码，此前半位值为之前子帧的奇偶校验位的第二个半位。*图 651* 中介绍了前导码模式 B、M 和 W。

图 651. S/PDIF 前导码



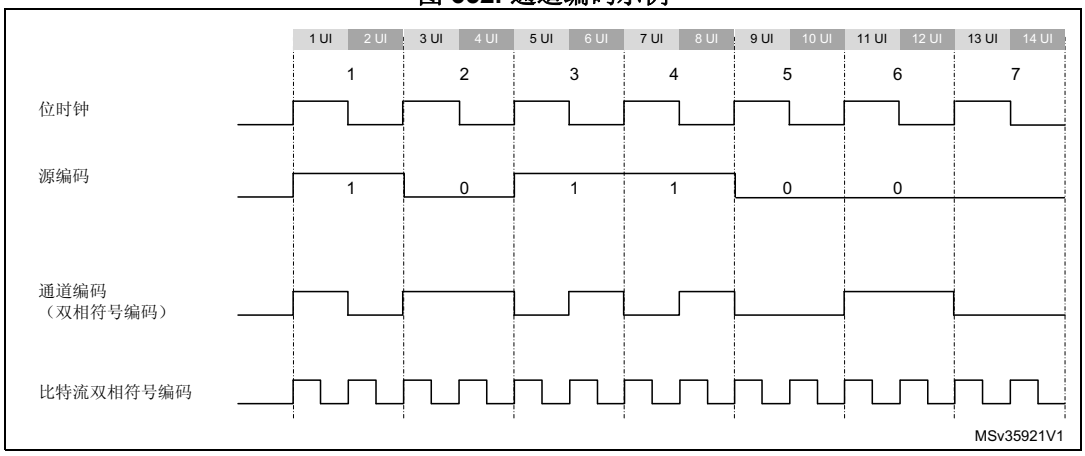
信息位编码

为最大程度减小传输线路上的直流分量值，并加快时钟从数据流中恢复的速度，位 4 到位 31 采用双相符号编码。

要传输的各个位通过由两个连续二进制状态构成的符号表示。符号的第一个状态始终不同于前一个符号的第二个状态。如果要传输的位为逻辑 0，则符号的第二个状态与第一个状态相同。但如果该位为逻辑 1，则两个状态不同。在 IEC-60958 规范中，这些状态称为“UI”（单位间隔）。

首先传输 24 个数据位中的 LSB。

图 652. 通道编码示例



52.3.3 SPDIFRX 解码器 (SPDIFRX_DC)

主要原则

SPDIFRX 用来解码 S/PDIF 数据流的技术以测量两个连续边沿之间时间间隔为基础。可在 S/PDIF 数据流中找到三类时间间隔：

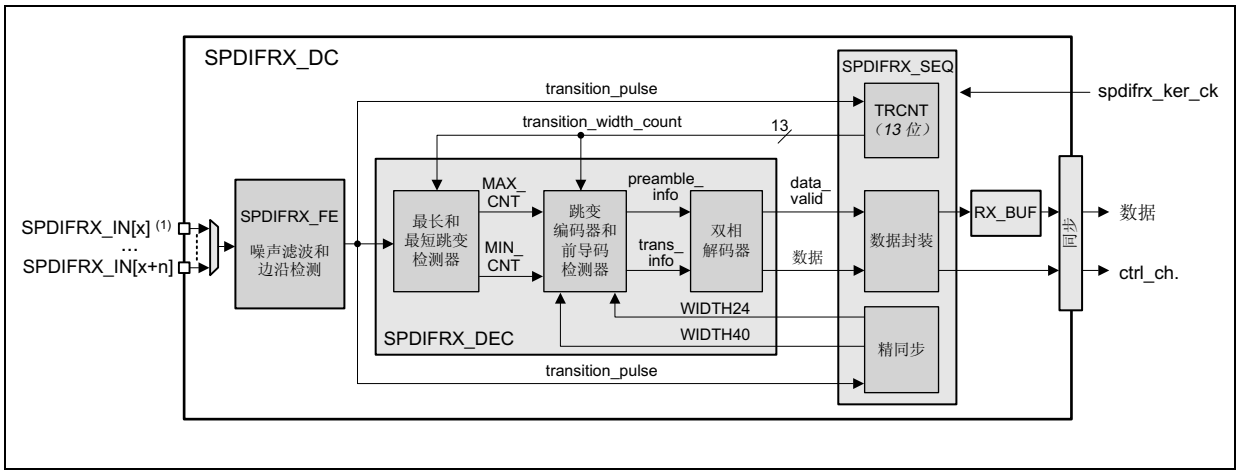
- 长时间间隔，持续 3 个 UI，记为 TL。仅在前导码期间出现。
- 中等时间间隔，持续 2 个 UI，记为 TM。它既出现在一些前导码中，也出现在信息字段中。
- 短时间间隔，持续 1 个 UI，记为 TS。它既出现在一些前导码中，也出现在信息字段中。

SPDIFRX_DC 模块负责解码接收到的 S/PDIF 数据流。它负责以下功能：

- 对传入信号进行重新采样和滤波
- 估算时间间隔
- 估算符号率和同步性
- 解码串行数据并验证完整性
- 检测块和子帧前导码
- 连续跟踪符号率

图 653 详细说明了 SPDIFRX 解码器。

图 653. SPDIFRX 解码器



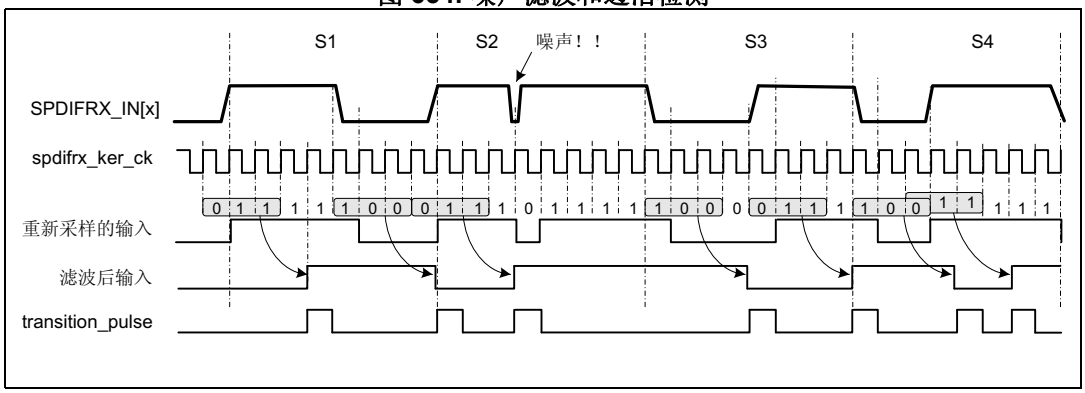
1. “n”固定为 3，“x”设置为 0。

噪声滤波和上升/下降沿检测

使用 spdifrx_ker_ck 时钟（采集时钟）重新采样所选的 SPDIFRX_IN 上接收到的 S/PDIF 信号。采用简单滤波来去除杂波。通过检测边沿转换的阶段执行此过程。边沿转换的检测步骤如下：

- 当采样到序列 0 后跟两个 1 时，检测到上升沿。
- 当采样到序列 1 后跟两个 0 时，检测到下降沿。
- 在上升沿后，预期为下降沿序列。
- 在下降沿后，预期为上升沿序列。

图 654. 噪声滤波和边沿检测



最长和最短转换检测器

最长和最短转换检测器模块可检测两次转换间的最大 (MAX_CNT) 和最小 (MIN_CNT) 持续时间。TRCNT 计数器用于测量时间间隔的持续时间。它由 spdifrx_ker_ck 时钟信号驱动。对于每个转换脉冲，都会存储计数器值并复位计数器以重新开始计数。

最大持续时间通常出现在前导码周期内。最大持续时间以 MAX_CNT 发送。最小持续时间以 MIN_CNT 发送。

当转换定时器过期时，最长和最短转换的搜索将停止。转换定时器类似于每转换 70 次传入信号就触发一次的看门狗定时器。请注意，计数 70 次转换可确保延迟略长于一个子帧。

请注意，当 TRCNT 因两个脉冲间的时间间隔过长而发生上溢时，SPDIFRX 将停止并且 SPDIFRX_SR 寄存器的标志 TERR 将置 1。

转换编码器和前导码检测器

转换编码器和前导码检测器接收 MAX_CNT 和 MIN_CNT。它还从 TRCNT 计数器接收当前转换宽度（参见*图 653*）。该模块通过将当前转换宽度与两个不同阈值（TH_{HI} 和 TH_{LO}）进行比较来编码当前转换宽度。

- 如果当前转换宽度小于 (TH_{LO} - 1)，则接收到的数据为数据位“1”的一半，并编码为 TS。
- 如果当前转换宽度大于 (TH_{LO} - 1) 且小于 TH_{HI}，则接收到的数据为数据位“0”，并编码为 TM。
- 如果当前转换宽度大于 TH_{HI}，则接收到的数据为前导码的长脉冲，并编码为 TL。
- 否则将生成错误代码（FERR 标志置 1）。

使用两种不同方法来详细阐述阈值 TH_{HI} 和 TH_{LO}。

如果外设正在执行其初始同步（“粗同步”），则阈值的计算方法如下：

- TH_{LO} = MAX_CNT / 2。
- TH_{HI} = MIN_CNT + MAX_CNT / 2。

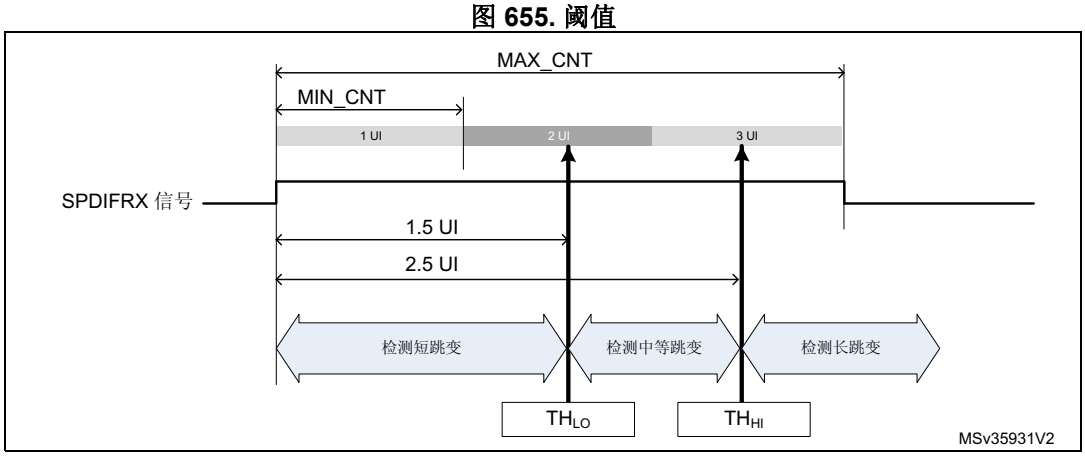
“粗同步”完成后，SPDIFRX 将使用更精确的参考来详细阐述阈值。SPDIFRX 将测量 24 个信号 (WIDTH24) 的长度来定义 TH_{LO}，测量 40 个信号 (WIDTH40) 的长度来定义 TH_{HI}。TH_{HI} 和 TH_{LO} 的计算方法如下：

- TH_{LO} = (WIDTH24) / 32
- TH_{HI} = (WIDTH40) / 32

第二个同步阶段称为“精同步”。其他相关信息，请参见*图 657*。

如后面的图所示，理想情况下 TH_{LO} 等于 1.5 个 UI，TH_{HI} 等于 2.5 个 UI。

图 655. 阈值



前导码检测器检查特定序列的四个连续转换来确定它们是否构成前导码的一部分。假设 TRANS0、TRANS1、TRANS2 和 TRANS3 表示四个连续的转换，编码方式如上面所述。*表 469* 给出了这四个转换的值以构成前导码。缺少该模式表示这些转换构成子帧中数据的一部分并且双相解码器将对它们进行解码。

表 469. 前导码的转换序列

前导码类型	双相数据模式	TRANS3	TRANS2	TRANS1	TRANS0
前导码 B	11101000	TL	TS	TS	TL
前导码 M	11100010	TL	TL	TS	TS
前导码 W	11100100	TL	TM	TS	TM

双相解码器

双相解码器使用由**转换编码器和前导码检测器**提供的转换信息来解码输入的双相符号数据流。首先，它等待前导码检测信息。在前导码检测后，它将解码以下转换信息：

- 如果传入的转换信息为 TM，则解码为“0”。
- 两个连续的 TS 解码为“1”。
- 任何其它转换序列将生成错误信号（FERR 置 1）。

通过这种方式解码 28 个数据位后，此模块将查找紧跟的前导码数据。如果新前导码与预期不同，则该模块将生成一个错误信号（FERR 置 1）。有关错误标志的更多信息，请参见*第 52.3.9 节：接收错误*。

数据封装

该模块负责解码 IEC-60958 帧和块。它还处理向 RX_BUF 或 SPDIFRX_CSR 寄存器写入数据。

52.3.4 SPDIFRX 对时钟偏差的容差

SPDIFRX 对时钟偏差的容差取决于一个位时隙中的采样时钟周期数。spdifrx_ker_ck 速度越快，接收越稳定。spdifrx_ker_ck 频率和信号率间的比率必须至少为 11。

有两种现象（至少）会降低接收质量：

- 反映两个连续转换之间长度差异的周期间抖动。
- 反映周期间抖动积累效果的长期抖动。它可被看作低频符号调制。

52.3.5 SPDIFRX 同步

将 SPDIFRXEN 设置为 01 或 11 时，同步阶段开始。*图 656* 给出了同步过程。

如果 SPDIFRX_CR 寄存器的 WFA 位置 1，则在开始同步过程前，外设必须首先检测所选 SPDIFRX_IN 线路上的活动。通过检测所选 SPDIFRX_IN 上的四个跳变来执行活动检测。外设保持在该状态，直至检测不到跳变。因为仅当所选 SPDIFRX_IN 输入上出现活动时，SPDIFRX 才切换为粗同步模式，因此该功能在避免同步错误时特别有用。其他相关信息，请参见*第 52.4 节：编程步骤*。
用户仍然可以通过将 SPDIFRXEN 设置为 0 来使 SPDIFRX 进入 STATE_IDLE。如果 WFA 设置为 0，外设将直接启动粗同步而不检查活动。

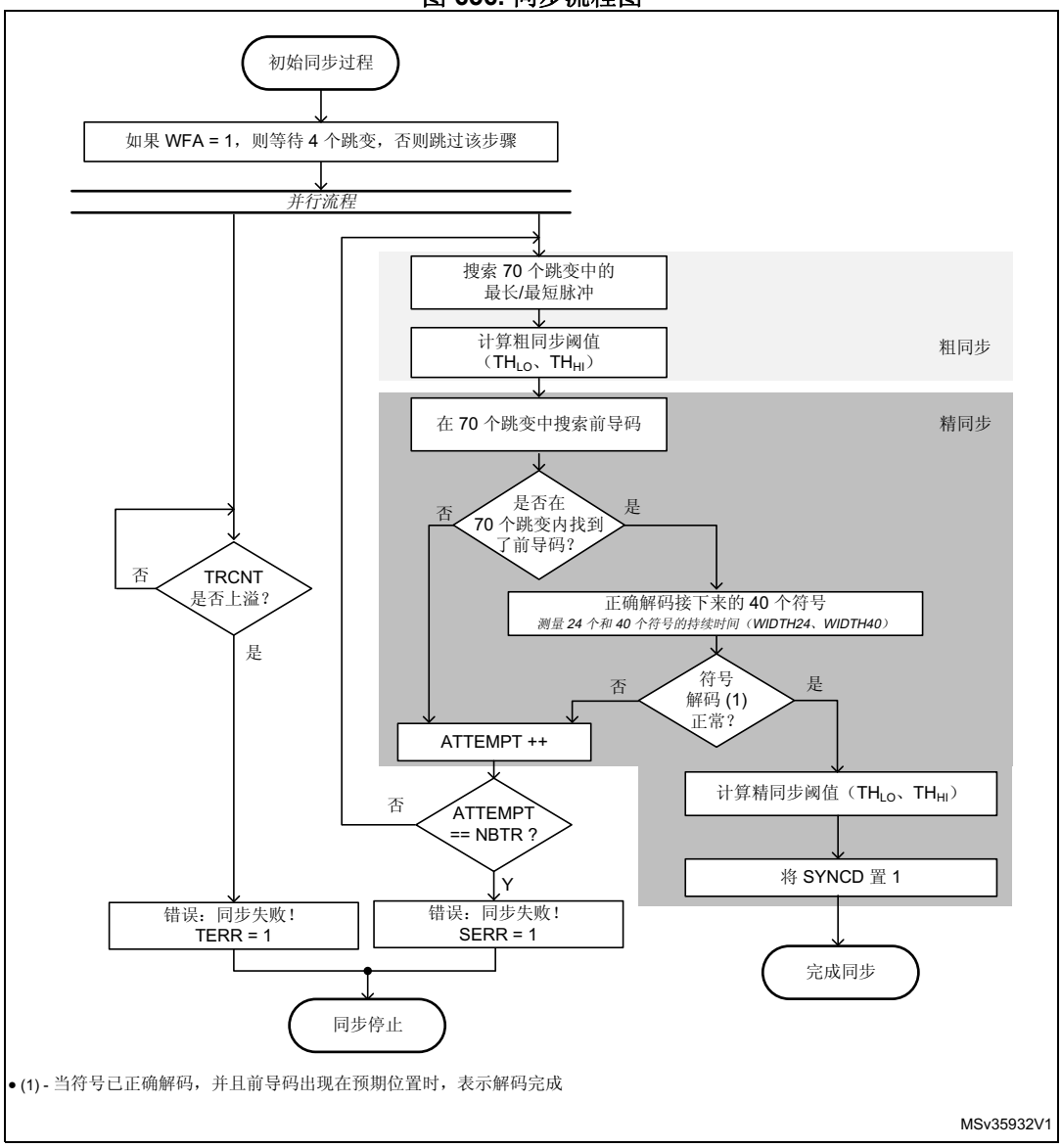
下一步将初步估算阈值（粗同步），以便执行精细同步（精同步）。由于 SPDIFRX 线路上的干扰，第一次可能无法正确执行该过程。为此，用户可在将 SERR 错误标志置 1 前编程允许的重试次数 (NBTR)。
当 SPDIFRX 能够正确测量 24 个和 40 个连续符号的持续时间并完成精同步时，阈值将更新并且标志 SYNCD 置 1。其他相关信息，请参见*转换编码器和前导码检测器*一节。

可以检测到两类错误：

- TRCNT 的上溢，这通常意味着输入线路上没有有效的 S/PDIF 数据流。该上溢由 TERR 标志指示。
- 重试次数达到了设定的值。这意味着 S/PDIF 信号上存在较强的抖动。该错误由 SERR 标志指示。

完成第一次精同步后，如果检测到下一个“B”前导码，将开始接收通道状态 (C) 和用户数据 (U)（请参见*图 660*）。随后，用户可通过 SPDIFRX_CSR 寄存器读取 IEC-60958 C 和 U 位。根据这些信息，用户便可选择适当的 DRFMT 和 RXSTEO 设置。例如，如果用户检测到当前音频流传输编码数据，则可以在开始接收数据之前将 RXSTEO 设为 0，将 DRFMT 设为 10。请注意，当 SPDIFRXEN = 11 时，不能修改 DRFMT 和 RXSTEO。如果 SPDIFRXEN 已经设置为 11，则忽略对这些位字段的写操作，尽管可使用导致 SPDIFRXEN 变为 11 的相同写指令来修改这些位字段。
然后在开始保存音频采样前，SPDIFRX 将等待 SPDIFRXEN = 11 和“B”前导码。

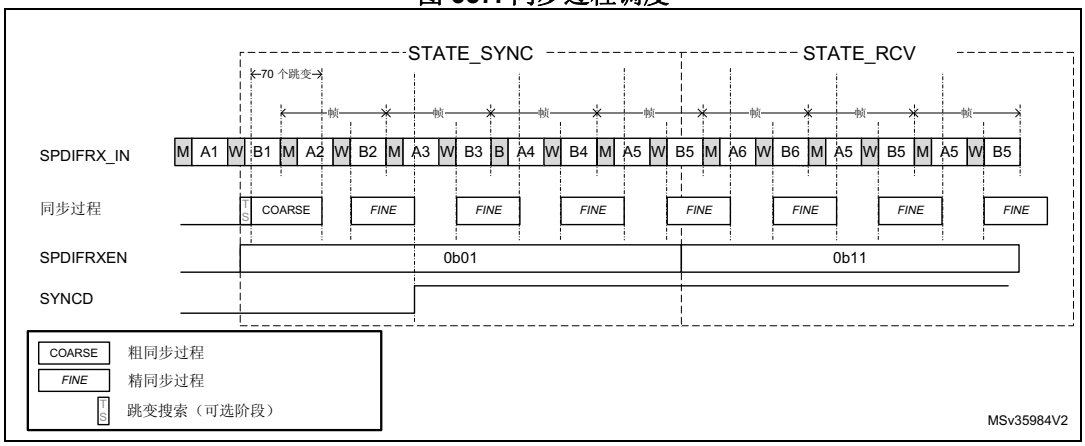
图 656. 同步流程图



有关 TRCNT 上溢的更多信息，请参见*帧结构和同步错误*。

如*图 657* 中所示，每一帧都将重新触发精同步过程来更新阈值，以便连续跟踪 S/PDIF 同步。

图 657. 同步过程调度



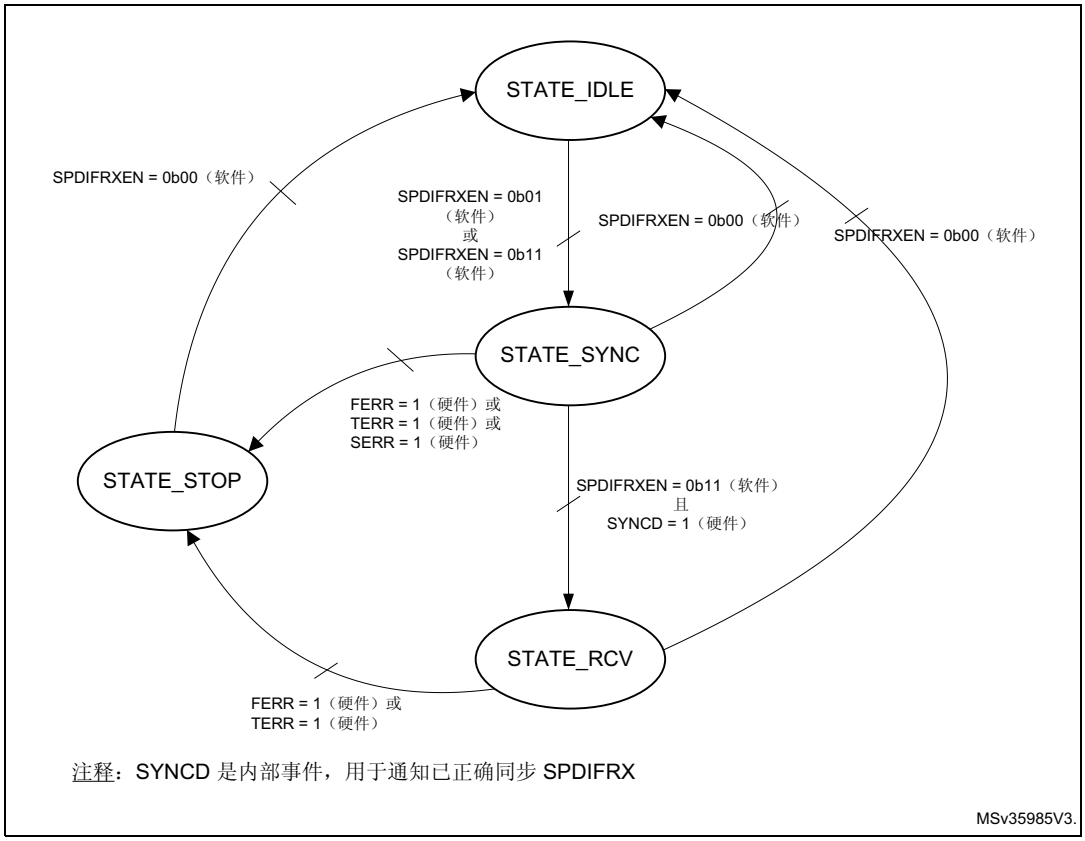
52.3.6 SPDIFRX 处理

软件可以通过 SPDIFRXEN 字段控制 SPDIFRX 的状态。SPDIFRX 可进入以下状态之一：

- STATE_IDLE：
  禁止外设，spdifrx_ker_ck 域复位。spdifrx_pclk 域功能正常。
- STATE_SYNC：
  外设与数据流同步，阈值定期更新，可通过 DMA 中断读取用户和通道状态。音频样本不提供给接收缓冲区。
- STATE_RCV：
  外设与数据流同步，阈值定期更新，可通过中断或 DMA 通道读取用户、通道状态和音频采样。当 SPDIFRXEN 变为 11 时，SPDIFRX 将在开始保存音频采样前一直等待“B”前导码。
- STOP_STATE：
  外设不再同步，用户、通道状态和音频采样的接收将停止。预期软件将重启 SPDIFRX。

图 658 给出了 SPDIFRX 可能的状态以及如何从一个状态转换为其它状态。受软件控制的位后跟说明“(SW)”，受 SPDIFRX 控制的位后跟说明“(HW)”。

图 658. SPDIFRX 状态



当 SPDIFRX 处于 STATE_IDLE 时：

- 通过将 SPDIFRXEN 设置为 01 或 11，软件可转换为 STATE_SYNC。

当 SPDIFRX 处于 STATE_SYNC 时：

- 如果同步失败或者接收的数据未正确解码且无法在不进行再同步的情况下恢复（FERR 或 SERR 或 TERR = 1），则 SPDIFRX 将进入 STATE_STOP 状态并且等待软件应答。
- 当同步阶段完成时，如果 SPDIFRXEN = 01，外设将保持该状态。
- 软件可随时将 SPDIFRXEN 设置为 0，随后 SPDIFRX 立刻返回至 STATE_IDLE。如果 DMA 传输正在进行，则将正确完成。
- 如果 SPDIFRXEN = 11 且 SYNCD = 1，则 SPDIFRX 进入 STATE_RCV

当 SPDIFRX 处于 STATE_RCV 时：

- 如果接收的数据未正确解码且无法在不进行再同步的情况下恢复（FERR 或 SERR 或 TERR = 1），则 SPDIFRX 将进入 STATE_STOP 状态并且等待软件应答。
- 软件可随时将 SPDIFRXEN 设置为 0，随后 SPDIFRX 立刻返回至 STATE_IDLE。如果 DMA 传输正在进行，则将正确完成。

当 SPDIFRX 处于 STATE_STOP 时：

- SPDIFRX 停止接收和同步，并等待软件将 SPDIFRXEN 位设置为 0，以清零错误标志。

当 SPDIFRXEN 设置为 0 时，SPDIFRX 被禁止，这意味着所有状态机被复位，并且 RX_BUF 被刷新。还要注意，标志 FERR、SERR 和 TERR 也会复位。

52.3.7 数据接收管理

SPDIFRX 为音频采样接收提供了双缓冲区。32 位缓冲区位于 spdifrx_ker_ck 时钟域 (RX_BUF) 和 SPDIFRX_FMTx_DR 寄存器中。如果 SPDIFRX_FMTx_DR 为空，则包含在 RX_BUF 中的有效数据将立刻传输至 SPDIFRX_FMTx_DR。

当满足以下两个条件时，包含在 RX_BUF 中的有效数据将传输至 SPDIFRX_FMTx_DR：

- 检测到奇偶校验位 (P) 和下一个前导码之间的转换（这表示字已经完全接收到）。
- SPDIFRX_FMTx_DR 为空。

双字缓冲区为延迟限制提供了更高的灵活性。

允许的最大延迟为 T_{SAMPLE} - $2T_{PCLK}$ - $2T_{spdifrx_ker_ck}$

其中，T_{SAMPLE} 是接收的立体声音频采样的音频采样率，T_{PCLK} 是 spdifrx_pclk 时钟的周期，$T_{spdifrx_ker_ck}$ 是 spdifrx_ker_ck 时钟的周期。

SPDIFRX 可以使用 DMA（spdifrx_dat_dma 和 spdifrx_ctrl_dma）或中断将音频采样传输到内存中。建议选择 DMA，有关更多信息，请参见*第 52.3.12 节：DMA 接口*。

SPDIFRX 提供多种处理接收数据的方法。用户可以分别处理控制信息流和音频采样流，或将二者一起处理。

对于各子帧，数据接收寄存器 SPDIFRX_FMTx_DR 包含 24 个数据位和可选的 V、U、C、PE 状态位以及 PT（请参见*混合数据和控制流*）。

请注意，PE 位表示奇偶校验错误位，该位将在解码的子帧中检测到奇偶校验错误时置 1。PT 字段包含前导码类型（B、M 或 W）。

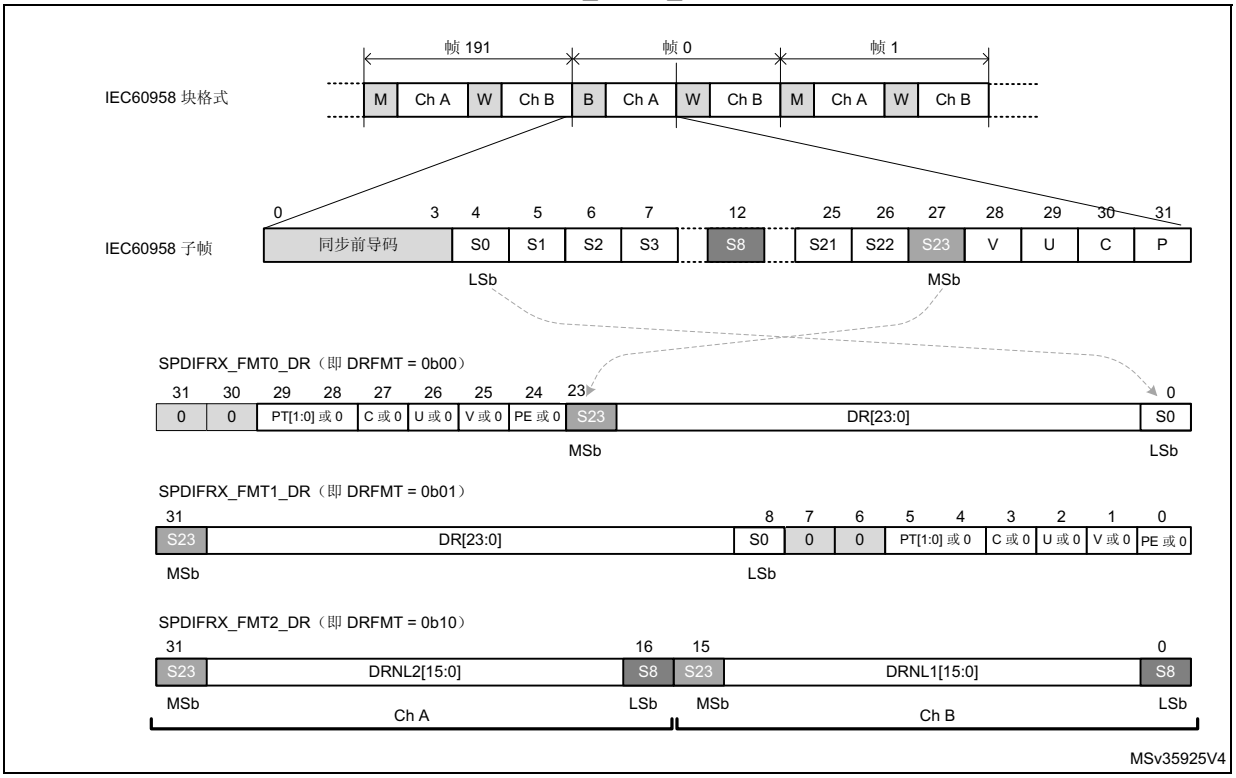
V、U 和 C 是从 S/PDIF 接口接收的值的直接副本。

DRFMT 位允许选择*图 659* 中所示的 3 种音频格式。

本文介绍了 3 个数据寄存器：SPDIFRX_FMTx[2:0]（x = 2 到 0），但实际上只有一个物理数据寄存器，有 3 种可能的格式：

- 当 DRFMT = 0 时，数据寄存器的格式为 SPDIFRX_FMT0_DR 所描述的格式
- 当 DRFMT = 1 时，数据寄存器的格式为 SPDIFRX_FMT1_DR 所描述的格式
- 当 DRFMT = 2 时，数据寄存器的格式为 SPDIFRX_FMT2_DR 所描述的格式

图 659. SPDIFRX_FMTx_DR 寄存器格式



将 DRFMT 设置为 00 或 01，可以使数据在 SPDIFRX_FMTx_DR 寄存器中左对齐或右对齐。可以根据软件所需的处理方式启用状态信息或将其强制设置为零。

通过 DRFMT = 10 得出的格式在非线性模式下相关，因为每个子帧仅使用 16 位。使用此格式，两个连续子帧的数据存储到 SPDIFRX_FMTx_DR 中，内存占用量将除以二。请注意，当 RXSTEO = 1 时，不存在偏离风险（即，来自通道 A 的数据始终存储在 SPDIFRX_FMTx_DR[31:16] 中）。如果 RXSTEO = 0，则在上溢情况下存在偏离风险。此时，SPDIFRX_FMTx_DR[31:16] 始终包含最早的值，SPDIFRX_FMTx_DR[15:0] 则包含较新的值（请参见*图 661*）。

在此格式中，状态信息不能与数据混合，但用户仍可通过 SPDIFRX_CSR 寄存器获取这些数据，并使用专用的 DMA 通道或中断将它们传输到内存中（请参见*第 52.3.8 节：专用控制流*）

混合数据和控制流

用户可以选择使用此模式，以完全灵活地处理控制流。用户可以选择应保存到数据寄存器 (SPDIFRX_FMTx_DR) 中的位字段。

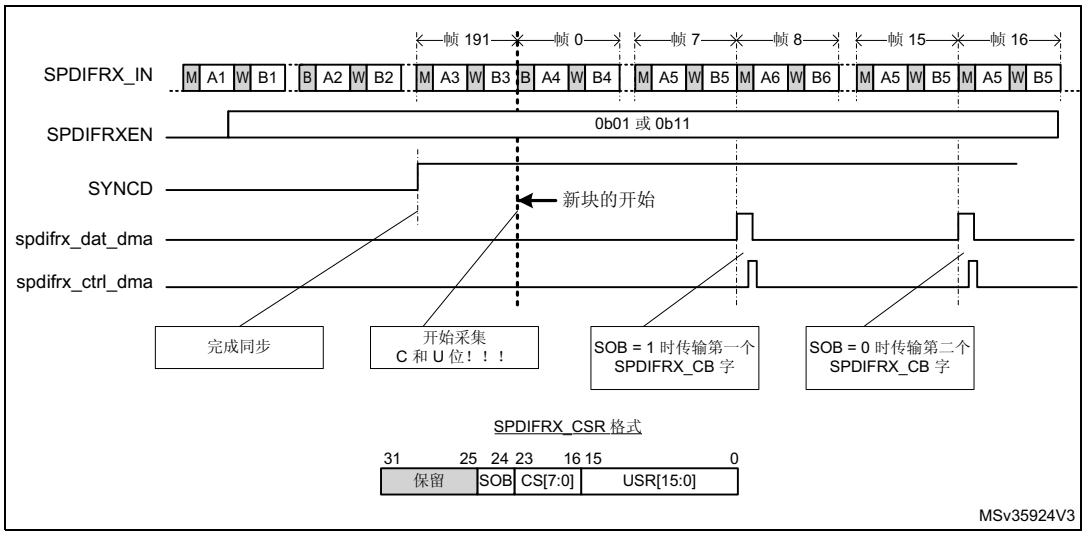
- 当位 PMSK = 1 时，将屏蔽奇偶校验错误信息（设置为 0），否则将其复制到 SPDIFRX_FMTx_DR 中。
- 当位 VMSK = 1 时，将屏蔽有效性信息（设置为 0），否则将其复制到 SPDIFRX_FMTx_DR 中。
- 当位 CUMSK = 1 时，将屏蔽通道状态和使用的数据信息（设置为 0），否则将其复制到 SPDIFRX_FMTx_DR 中。
- 当位 PTMSK = 1 时，将屏蔽前导码类型信息（设置为 0），否则将其复制到 SPDIFRX_FMTx_DR 中。

52.3.8 专用控制流

SPDIFRX 可以通过专用 DMA 通道捕获用户数据和通道状态信息。该功能允许 SPDIFRX 连续采集通道状态和用户信息。采集将在 IEC 60958 块开始时启动。有两个位字段可用于控制此路径：CBDMAEN 和 SPDIFRXEN。当 SPDIFRXEN 设为 01 或 0x11 后，同步阶段完成后开始采集。当接收到 8 个通道状态和 16 个用户数据位时，会将它们封装并存储到 SPDIFRX_CSR 寄存器中。如果 CBDMAEN 位置 1，则触发 DMA 请求（请参见*图 660*）。

如果 CS[0] 对应新块的第一位，则 SOB 位置 1。请参见*第 52.5.8 节：SPDIFRX 通道状态寄存器 (SPDIFRX_CSR)*。可使用 CHSEL 位选择用户想要从通道 A 还是通道 B 选择通道状态信息 (C)。

图 660. 通道/用户数据格式



注： *检测到第一个块开始（B 前导码）后，SPDIFRX 将每 8 个帧检测一次前导码类型。*

注： *SPDIFRX_FMTx_DR 寄存器中的上溢错误不影响该路径。*

52.3.9 接收错误

帧结构和同步错误

发生下列情况之一时，SPDIFRX 将检测到错误：

- 出现以下情况时，FERR 位置 1：
  - 对于 28 个信息位中的每个位，如果一个符号转换序列不正确：例如短脉冲未按对分组。
  - 如果前导码出现在异常位置，或未接收到预期的前导码。
- 当同步失败时，SERR 位置 1，原因是重试次数超过了设定的值。
- 当用于估算两次传输的时间间隔的计数器 (TRCNT) 产生上溢时，TERR 位置 1。
  在 8192 个 spdifrx_ker_ck 时钟周期内未检测到转换时，将发生上溢。这表示最大时间间隔为 11.6 帧。

当这些标志之一变为 1 时，将忽略所选 SPDIFRX_IN 上的通信，并将在 SPDIFRX_CR 寄存器的 IFEIE 位置 1 时生成中断。

发生这些错误之一时的正常步骤为：

- 将 SPDIFRXEN 设置为 0 以清零错误标志
- 将 SPDIFRXEN 设置为 01 或 11 以重新启动 SPDIFRX

其他相关信息，请参见*图 658*。

奇偶校验错误

对于各子帧，28 个信息位中预期包含偶数个 0 和 1。否则，SPDIFRX_SR 寄存器的 PERR 位将被置位，并且如果 SPDIFRX_CR 寄存器中的奇偶校验中断使能 PERRIE 位置 1，则将生成中断。不会暂停输入数据的接收，并且即使中断仍处于挂起状态，SPDIFRX 也会继续将数据传输到 SPDIFRX_FMTx_DR 中。

通过 PERRCF 位清零 PERR 标志来确认中断。

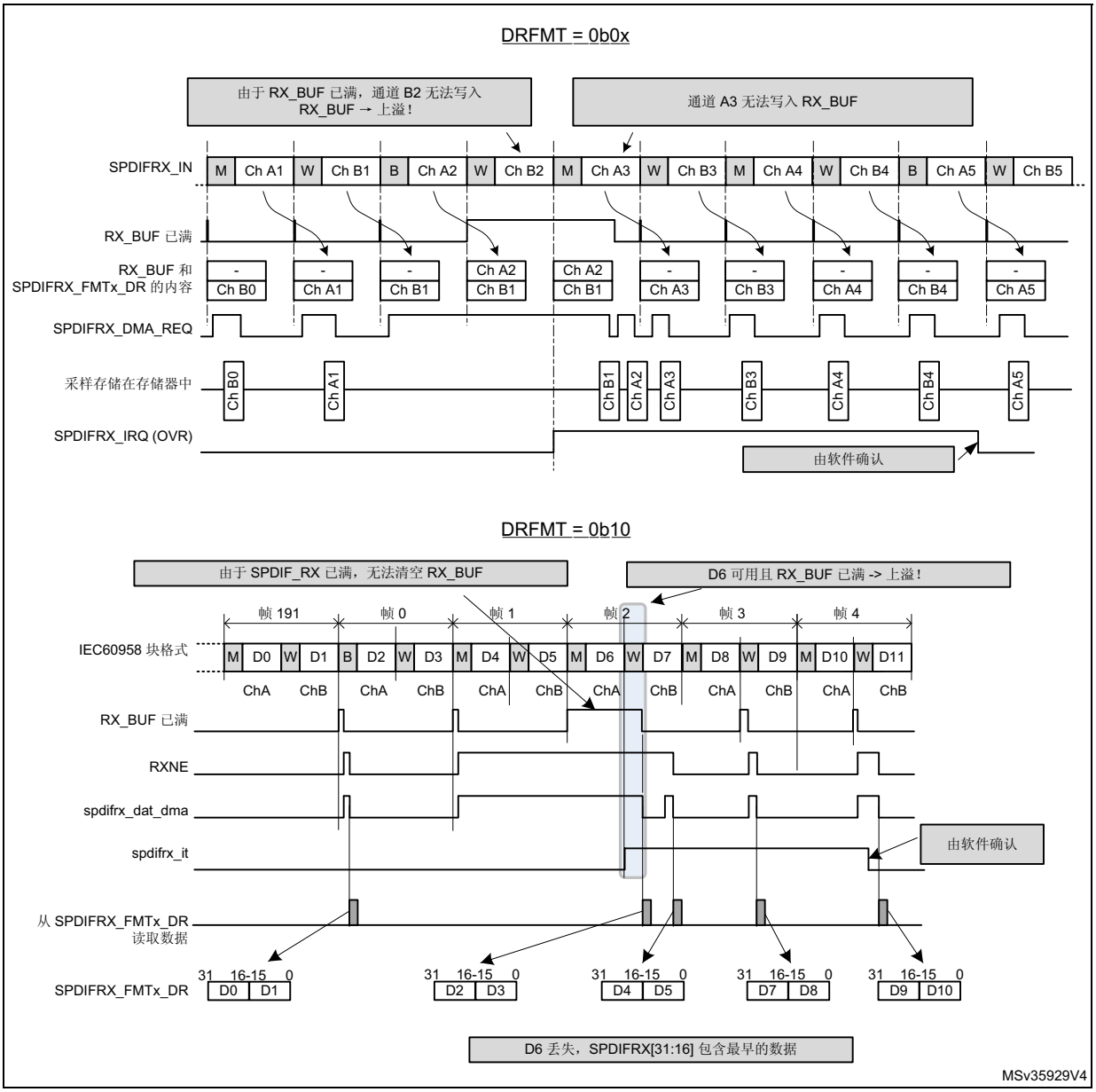
如果软件要确保 PDIFRX_FMTx_DR 寄存器中读取的数据与 PERR 位的值保持一致，必须将 PMSK 位设置为 0。

上溢错误 (Overrun error)

如果 SPDIFRX_FMTx_DR 和 RX_BUF 均已满，而 SPDIFRX_DC 又需要向 RX_BUF 中写入新采样，则将丢弃这一新采样，并触发上溢条件。SPDIFRX_SR 寄存器中的上溢错误标志 OVR 置 1，并且如果 SPDIFRX_CR 寄存器的 OVRIE 位置 1，则将生成中断。

如果 RXSTEO 位设置为 0，则只要 RX_BUF 为空，SPDIFRX 便会存储接下来传入的数据，即使 OVR 标志仍然处于挂起状态。主要目的是尽可能地减少丢失的采样数。请注意，无论 DRFMT 的值为何，这种行为都是类似的。请参见*图 661*。

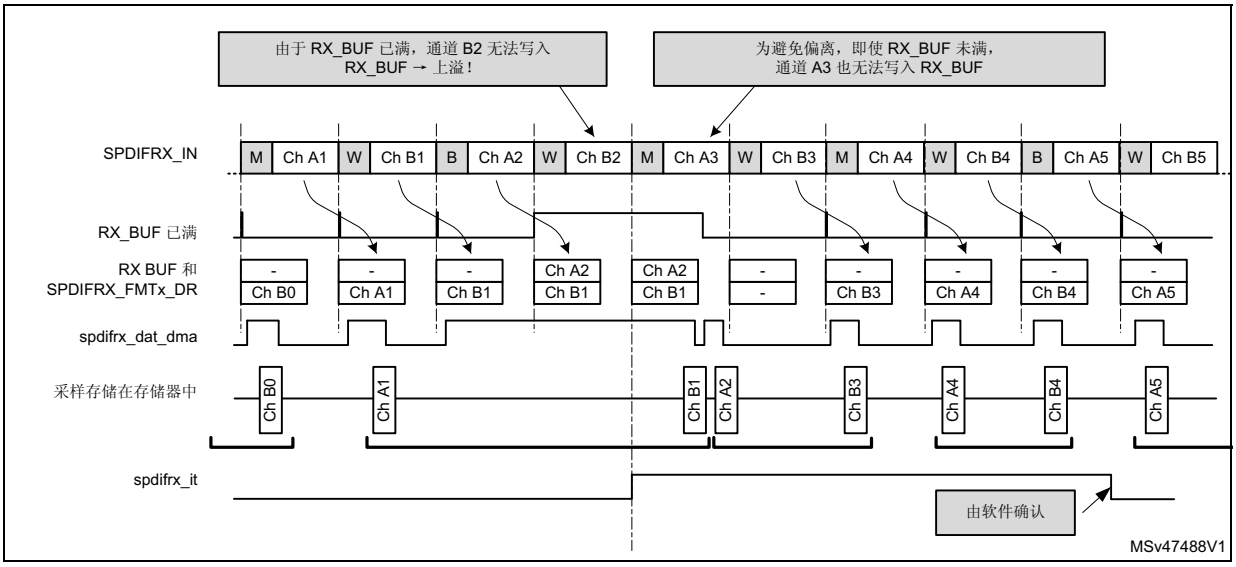
图 661. RXSTEO = 0 时的 S/PDIF 上溢错误



如果 RXSTEO 位置 1，则意味着传输的是立体声数据，此时 SPDIFRX 必须避免左右通道之间的偏离。因此，即使在 RX_BUF 内仍有空间，外设也必须丢弃一秒的采样以避免发生偏离。随后，传入的采样可正常写入到 RX_BUF 中，即使 OVR 标志仍处于挂起状态。请参见 *图 662*。

在 OVRCF 位置 1 时，OVR 标志由软件清零。

图 662. RXSTEO = 1 时的 S/PDIF 上溢错误



52.3.10 时钟策略

SPDIFRX 块需要两个不同的时钟：

- APB 时钟 (spdifrx_pclk)，用于寄存器接口，
- spdifrx_ker_ck，主要由 SPDIFRX_DC 部分使用。这些时钟不应锁相，因此需要重新同步这些时钟域中的所有信号（*图 648* 中的 SYNC 块）。

为正确解码传入的 S/PDIF 数据流，SPDIFRX_DC 必须以至少为最大符号率的 11 倍或者音频采样率的 704 倍的时钟重新采样接收的数据。例如，如果用户预期接收到最高 12.288 MHz 的符号率，则采样率至少为 135.2 MHz。SPDIFRX_DC 使用的时钟为 spdifrx_ker_ck。

spdifrx_pclk 的频率必须至少等于符号率。

表 470. spdifrx_ker_ck 的最小频率与音频采样率(1)

符号率	spdifrx_ker_ck 的最小频率	备注
3.072 MHz	33.8 MHz	用于 48 kHz 数据流
6.144 MHz	67.6 MHz	用于 96 kHz 数据流
12.288 MHz	135.2 MHz	用于 192 kHz 数据流

1. 检查 RCC 功能以确认可以支持的采样率。

52.3.11 符号时钟生成

SPDIFRX 模块在 **spdifrx_symb_ck** 信号上提供了符号时钟，不但可用作其他音频设备（例如，SAI 或 SPI/I2S）的参考内核时钟，而且可用于 SPDIFRX 转 I2S 桥接功能。

符号时钟使用 WIDTH24、WIDTH40 和符号边界的值构建。

- 在接收子帧同步前导码期间，符号时钟的下降沿和上升沿通过 WIDTH24 和 WIDTH40 的值形成。请注意，如果 SPDIFRX 为 STATE_STOP 或 STATE_IDLE，WIDTH24 和 WIDTH40 也用于生成符号时钟。有关详细信息，请参见 *表 471*。
- 在接收子帧有效负载期间，SPDIFRX 使用符号边界来生成上升沿，WIDTH24 和 WIDTH40 的值则用于生成下降沿。

在接收子帧有效负载期间，符号时钟的占空比接近 50%。不过，SPDIFRX 从使用 WIDTH24 和 WIDTH40 生成的符号时钟转换到由符号时钟边界生成的时钟时，占空比会发生改变，反之亦然。

符号时钟会产生大幅抖动的主要原因是：

- 使用 spdifrx_ker_ck 时钟重新采样 S/PDIF 信号
- 符号时钟生成模式发生转换

因此，如果将符号时钟用作 A/D 或 D/A 转换器的参考时钟，则应用程序必须考虑质量下降问题。

该符号时钟的生成由 CKSEN 位控制。如果 CKSEN = “1”，则在 SPDIFRX 成功完成首次精同步 (SYNCD = 1) 时，以及从所选 SPDIFRX 输入接收正确的数据时，会生成符号时钟。

当 SPDIFRX 进入 STATE_STOP 或 STATE_IDLE 时，如果位 CKSBKPEN = “0”，则会对符号时钟进行门控。如果 CKSBKPEN = “1”，则在 SPDIFRX 正确同步（即，WIDTH24 和 WIDTH40 的值有效）时，仍会生成备份符号时钟。*表 471* 详细介绍了有关控制符号时钟生成的条件。

表 471. spdifrx_symb_ck 的生成条件

SPDIFRX 状态和条件	CKSEN	CKSBKPEN	spdifrx_symb_ck state
任意状态	0	X	禁止
– SPDIFRX 处于 STATE_SYNC，成功完成精同步（SYNCD = “1”） – SPDIFRX 处于 STATE_RCV，通过所选 SPDIFRX 输入接收到有效数据。	1	0	使能
– SPDIFRX 处于 STATE_IDLE – SPDIFRX 处于 STATE_STOP – SPDIFRX 未完成精同步（正在进行） – SPDIFRX 处于 STATE_RCV，但在所选 SPDIFRX 输入上未检测到数据（转换）。		0	禁止
– SPDIFRX 处于 STATE_IDLE，但存在 WIDTH40 和 WIDTH24 的有效值 – SPDIFRX 处于 STATE_SYNC，成功完成精同步（SYNCD = “1”） – SPDIFRX 处于 STATE_SYNC，正在进行的精同步尚未完成，但 WIDTH40 和 WIDTH24 包含来自上一次同步的有效值 – SPDIFRX 处于 STATE_RCV，通过所选 SPDIFRX 输入接收到有效数据 – SPDIFRX 处于 STATE_STOP，但存在 WIDTH40 和 WIDTH24 的有效值。	1	1	使能
– SPDIFRX 处于 IDLE，WIDTH40 和 WIDTH24 的值无效 – SPDIFRX 处于 STOP，WIDTH40 和 WIDTH24 的值无效（SERR = “1”） – SPDIFRX 处于 STATE_SYNC，WIDTH40 和 WIDTH24 的值无效，且正在进行的精同步尚未完成 – SPDIFRX 处于 STATE_RCV，在所选 SPDIFRX 输入上未检测到转换			禁止

请注意，当标志 SERR 置 “1” 时，无法生成符号时钟和备份时钟，因为不存在同步。

请注意，如果 CKSEN 和 CKSBKPEN 均置 “1”，则在 SPDIFRX 从 STATE_SYNC 或 STATE_RCV 切换到 STATE_STOP 或 STATE_IDLE 时，符号时钟会丢失某些转换。

CKSEN 和 CKSBKPEN 位在 *SPDIFRX 控制寄存器 (SPDIFRX_CR)* 中。

52.3.12 DMA 接口

SPDIFRX 能够使用 DMA 执行通信。

注： *有关 DMA 控制器可用性的信息，请参见产品规范。*

SPDIFRX 提供两个独立的 DMA 通道：

- 一个专门用于数据传输的 DMA 通道
- 一个专用于传输通道状态和用户信息的 DMA 通道

可通过将 SPDIFRX_CR 寄存器中的 RXDMAEN 位置 1 来启用数据的 DMA 模式。在这种情况下，一旦 SPDIFRX_FMTx_DR 非空，SPDIFRX 接口就会向 DMA 发送传输请求。DMA 通过 SPDIFRX_FMTx_DR 寄存器读取接收到的数据，无需 CPU 干预。

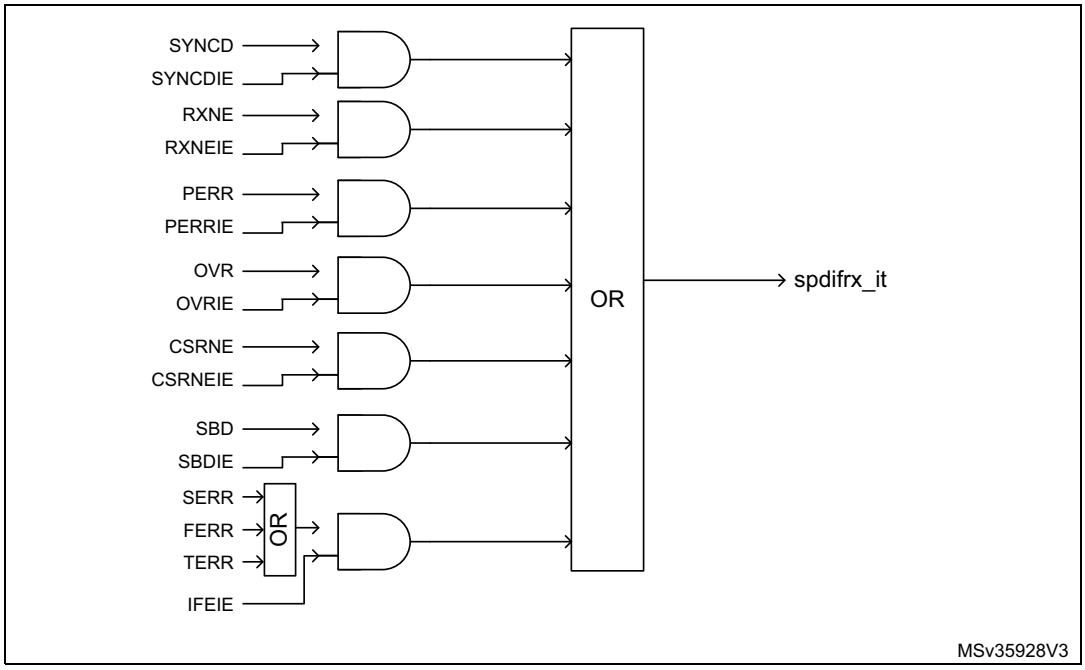
有关控制数据的 DMA 的用法，请参见*第 52.3.8 节：专用控制流*。

52.3.13 中断生成

下列各项共享一条中断线：

- 数据流的接收事件 (RXNE)
- 控制流的接收事件 (CSRNE)
- 数据损坏检测 (PERR)
- 传输流中断 (OVR)
- 帧结构和同步错误（SERR、TERR 和 FERR）
- 新块中段的开始 (SBD)
- 同步完成 (SYNCD)

图 663. SPDIFRX 接口中断映射图



清零中断源

- 读取 SPDIFRX_FMTx_DR 寄存器时，将清零 RXNE。
- 读取 SPDIFRX_CSR 寄存器时，将清零 CSRNE。
- 当 SPDIFRXEN 被设置为 0 时，将清零 FERR。
- 当 SPDIFRXEN 被设置为 0 时，将清零 SERR。
- 当 SPDIFRXEN 被设置为 0 时，将清零 TERR。
- 其它中断源通过 SPDIFRX_IFCR 寄存器清零

注： *仅当 SPDIFRX 与输入数据流同步时 (SYNCD = 1)，才会发生 SBD 事件。*
当包含 B 前导码的子帧因上溢而丢失时，将无法保证 SBD 标志行为。

52.3.14 寄存器保护

SPDIFRX 块内嵌某种硬件保护功能以避免错误使用控制寄存器。下表根据 SPDIFRX 的状态说明了位字段属性。

表 472. 位字段属性与 SPDIFRX 状态

寄存器	位字段	SPDIFRXEN		
		00 (STATE_IDLE)	01 (STATE_SYNC)	11 (STATE_RCV)
SPDIFRX_CR	INSEL	rw	r	r
	WFA	rw	r	r
	NBTR	rw	r	r
	CHSEL	rw	r	r
	CBDMAEN	rw	rw	rw
	PTMSK	rw	rw	rw
	CUMSK	rw	rw	rw
	VMSK	rw	rw	rw
	PMSK	rw	rw	rw
	DRFMT	rw	rw	r
	RXSTEO	rw	rw	r
	RXDMAEN	rw	rw	rw
SPDIFRX_IMR	所有位字段	rw	rw	rw

表中清楚地显示，当 SPDIFRX 处于 STATE_IDLE 状态时，必须对 INSEL 等位字段进行编程。在 SPDIFRX 的其他状态下，硬件会阻止向该位字段写入。

注： *即使硬件允许“实时”写入 CBDMAEN 和 RXDMAEN，也不推荐在 SPDIFRX 接收数据时使能 DMA。*

请注意，各屏蔽位（例如 PMSK、VMSK）可以在任意 SPDIFRX 状态下进行“实时”更改，但任何更改都不影响已经保存到 SPDIFRX_FMTx_DR 中的数据。

52.4 编程步骤

下列示例将说明 SPDIFRX 块的完整激活顺序。数据路径以及通道状态和用户信息均使用专用 DMA 通道。激活顺序随即分为下面几个步骤：

- 等待所选 SPDIFRX_IN 输入上的有效数据
- 与 S/PDIF 数据流同步
- 读取通道状态和用户数据以建立完整的音频路径
- 开始采集数据

一种检查有效数据是否可以进入 SPDIFRX_IN 线路的简单方法是通过将 WFA 位置 1 将 SPDIFRX 切换为 STATE_SYNC。后面将重点对检测进行说明。此外，也可以按如下方式实现此功能：

- 软件必须不时检查（例如，每隔 100 ms）SPDIFRX 是否可找到同步。可以通过检查 TERR 位是否置 1 来执行此操作。当该位置 1 时，表示没有发现活动。
- 将 SPDIFRX_IN 输入连接到外部中断事件块以便检测 SPDIFRX_IN 线路的转换。当检测到活动时，可将 SPDIFRXEN 设置为 01 或 11。

  对于这两种实现，WFA 位设置为 0。

52.4.1 初始化阶段

- 初始化功能具有如下形式：
- 为音频采样和 IEC60958 通道状态和用户信息配置 DMA 传输（DMA 通道选择和活动、优先级、要传输的数据数量、循环/非循环模式和 DMA 中断）
- 配置目标地址：
  - 将 SPDIFRX_CSR 寄存器的地址配置为 IEC60958 通道状态和用户信息的源地址
  - 将 SPDIFRX_FMTx_DR 寄存器的地址配置为音频采样的源地址
  - 使能 spdifrx_ker_ck 的生成。要定义最小时钟频率与支持的音频采样率，请参见 *表 470*。
    请注意，无法提前获知已接收数据流的音频采样率。这意味着用户必须至少选择的 spdifrx_ker_ck 频率为应处理的最大音频采样率的 704 倍：例如，如果应用能够处理采样率最大为 96 kHz 的数据流，则 $F_{spdifrx_ker_ck}$ 至少应为 704 x 96 kHz = 67.6 MHz
- 使能错误和事件信号的中断（IFEIE = SYNCDIE = OVRIE，PERRIE = 1，其它位设置为 0）。注意，SYNCDIE 可设置为 0。
- 配置 SPDIFRX_CR 寄存器：
  - INSEL 必须选择所需输入
  - NBTR = 2，WFA = 1（允许重试 16 次，等待活动，直至进入同步阶段）
  - PTMSK = CUMSK = 1（前导码，C 和 U 位不与数据混合）
  - VMSK = PMSK = 0（奇偶校验错误和有效性位与数据混合）
  - CHSEL = 0（从子帧 A 读取通道状态）
  - DRFMT = 01（数据左对齐）
  - RXSTEO = 1（预期立体声模式为线性）
  - CBDMAEN = RXDMAEN = 1（使能 DMA 通道）
  - SPDIFRXEN = 01（将 SPDIFRX 切换为 STATE_SYNC）
- CPU 可以进入 WFI 模式

随后 CPU 将收到来自 DMA 或 SPDIFRX 的中断。

52.4.2 处理来自 SPDIFRX 的中断

当收到来自 SPDIFRX 的中断时，软件必须通过读取 SPDIFRX_SR 寄存器来确认中断源。

- 如果 SYNCD 置 1，则意味着同步已正常完成。此示例中无需执行任何操作，因为 DMA 已进行编程。软件仅需要等待 DMA 中断以读取通道状态信息。
  必须通过将 SPDIFRX_IFCR 寄存器的 SYNCDCF 位置 1 来清零 SYNCD 标志。
- 如果 TERR 或 SERR 或 FERR 设置为 1，则软件必须将 SPDIFRXEN 设置为 0，并从初始化阶段重新启动。
  - TERR 表示在同步阶段期间或之后发生超时。
  - SERR 表示因为达到允许的最大重试次数而同步失败。
  - FERR 表示同步失败后读取数据（例如异常前导码、错误数据解码）。
- 如果 PERR 置 1，则意味着已检测到奇偶校验错误，从而表明接收到的音频采样、通道状态或用户数据位中的某一位损坏。这里采取的操作取决于应用：可以直接丢弃当前通道状态信息，原因是其不可靠。无需从初始化阶段重新启动，因为同步不会丢失。
  必须通过将 SPDIFRX_IFCR 寄存器的 PERRCF 位置 1 来清零 PERR 标志。

52.4.3 处理来自 DMA 的中断

如果中断来自通道状态所用的 DMA 通道 (SPDIFRX_CSR)：

如果未发生错误（即 PERR），则 CPU 可以开始解码通道信息。例如，通道状态的位 1 通知用户当前数据流是否为线性。该信息对于建立正常处理链十分重要。同样，通道状态的位 24 到位 27 给出了传入数据流的采样频率。

借助该信息，用户可在启动数据接收之前配置 RXSTEO 位和 DRFMT 位字段。例如，如果当前数据流为非线性 PCM，则 RXSTEO 将被设置为 0，而 DRFMT 将被设置为 10。随后用户可通过将 SPDIFRXEN 设置为 11 来使能数据接收。

SOB 置 1 表示开始一个新数据块。该信息有助于软件识别通道状态的位 0。请注意，若每次向内存传输 24 个值 DMA 都产生一个中断，则第一个字始终对应新数据块的开始位置。

如果中断来自音频采样所用的 DMA 通道 (SPDIFRX_FMTx_DR)：

这里执行的过程取决于数据类型（线性或非线性）以及所选的数据格式。

例如，在线性模式中，如果 PE 或 V 位置 1，则将在本地执行一个特殊过程以避免输出上出现杂波。在非线性模式下，这些位并不重要，因此数据帧具有各自的校验和。

52.5 SPDIFRX 接口寄存器

52.5.1 SPDIFRX 控制寄存器 (SPDIFRX_CR)

SPDIFRX control register

该寄存器仅允许 32 位访问。

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKSBKPEN	CKSEN	Res.	INSEL[2:0]		
										rw	rw		rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	WFA	NBTR[1:0]		CHSEL	CBDMAEN	PTMSK	CUMSK	VMSK	PMSK	DRFMT[1:0]		RXSTEO	RXDMAEN	SPDIFRXEN[1:0]	
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21 **CKSBKPEN**：备份符号时钟使能 (Backup symbol clock enable)
该位由软件置 1/复位。
1：CKSEN = 1 时，SPDIFRX 会生成备份符号时钟。
0：SPDIFRX 不生成备份符号时钟。

位 20 **CKSEN**：符号时钟使能 (Symbol clock enable)
该位由软件置 1/复位。
1：SPDIFRX 生成符号时钟。
0：SPDIFRX 不生成符号时钟。

位 19 保留，必须保持复位值。

位 18:16 **INSEL[2:0]**：SPDIFRX 输入选择 (SPDIFRX input selection)(1)
000：选择 SPDIFRX_IN0
001：选择 SPDIFRX_IN1
010：选择 SPDIFRX_IN2
011：选择 SPDIFRX_IN3
其他值：保留

位 15 保留，必须保持复位值。

位 14 **WFA**：等待活动 (Wait for activity)(1)
该位由软件置 1/复位。
1：执行同步前，SPDIFRX 等待 SPDIFRX_IN 线路上的活动（4 次跳变）
0：执行同步前，SPDIFRX 不等待 SPDIFRX_IN 线路上的活动

位 13:12 **NBTR[1:0]**：同步阶段期间允许的最大重试次数 (Maximum allowed re-tries during synchronization phase)(1)

00：不允许重试（仅尝试一次）
01：允许重试 3 次
10：允许重试 15 次
11：允许重试 63 次

位 11 **CHSEL**：通道选择 (Channel selection)(1)

该位由软件置 1/复位。
1：控制流从通道 B 获取通道状态
0：控制流从通道 A 获取通道状态

位 10 **CBDMAEN**：控制流的控制缓冲区 DMA 使能 (Control buffer DMA enable for control flow)(1)

该位由软件置 1/复位。
1：使能 DMA 模式以接收通道状态和使用的数据信息。
0：禁止 DMA 模式接收通道状态和使用的数据信息。
注意：当此位置 1 时，每当 CSRNE 标志置 1 时，即产生 DMA 请求。

位 9 **PTMSK**：前导码类型的屏蔽位 (Mask of preamble type bits)(1)

该位由软件置 1/复位。
1：不将前导码类型位复制到 SPDIFRX_FMTx_DR 中，而是写入 0
0：将前导码类型位复制到 SPDIFRX_FMTx_DR 中

位 8 **CUMSK**：通道状态和用户的屏蔽位 (Mask of channel status and user bits)(1)

该位由软件置 1/复位。
1：不将通道状态和用户位复制到 SPDIFRX_FMTx_DR 中，而是写入 0。
0：将通道状态和用户位复制到 SPDIFRX_FMTx_DR 中。

位 7 **VMSK**：有效性的屏蔽位 (Mask of validity bit)(1)

该位由软件置 1/复位。
1：不将有效性位复制到 SPDIFRX_FMTx_DR 中，而是写入 0。
0：将有效性位复制到 SPDIFRX_FMTx_DR 中。

位 6 **PMSK**：屏蔽奇偶校验错误位 (Mask parity error bit)(1)

该位由软件置 1/复位。
1：不将奇偶校验错误位复制到 SPDIFRX_FMTx_DR 中，写入 0 来替代。
0：将奇偶校验错误位复制到 SPDIFRX_FMTx_DR 中。

位 5:4 **DRFMT[1:0]**：RX 数据格式 (RX data format)(1)

该位由软件置 1/复位。
11：保留
10：通过将两个 16 位采样置于一个 32 位字中来封装数据采样。
01：数据采样采用左对齐 (MSB)
00：数据采样采用右对齐 (LSB)

位 3 **RXSTEO**：立体声模式 (Stereo mode)(1)

该位由软件置 1/复位。
1：外设处于立体声模式。
0：外设处于单声道模式。
注意：该位用于在发生上溢时处理偏离。

位 2 **RXDMAEN**：数据流的接收器 DMA 使能 (Receiver DMA enable for data flow)(1)

该位由软件置 1/复位。
1：针对接收使能 DMA 模式。
0：针对接收禁止 DMA 模式。
注意：当此位置 1 时，每当 RXNE 标志置 1 时，即产生 DMA 请求。

位 1:0 **SPDIFRXEN[1:0]**：外设模块使能 (Peripheral block enable)(1)

该位字段由软件修改。

更改外设阶段，仅以下三种状态有效：STATE_IDLE、STATE_SYNC 和 STATE_RCV。

00：禁止 SPDIFRX (STATE_IDLE)。

01：仅使能 SPDIFRX 同步。

10：保留

11：使能 SPDIF 接收器。

注意：无法从 STATE_RCV 转换为 STATE_SYNC，用户必须首先进入 STATE_IDLE。
可以从 STATE_IDLE 转换为 STATE_RCV：在这种情况下，外设从 STATE_IDLE 转换为 STATE_SYNC，并且一旦执行完同步便进入 STATE_RCV。

1. 有关位字段属性的更多信息，请参见*第 52.3.14 节：寄存器保护*。

52.5.2 SPDIFRX 中断屏蔽寄存器 (SPDIFRX_IMR)

SPDIFRX interrupt mask register

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IFEIE	SYNCDIE	SBLKIE	OVRIE	PERRIE	CSRNEIE	RXNEIE
									rw	rw	rw	rw	rw	rw	rw

位 31:7 保留，必须保持复位值。

位 6 **IFEIE**：串行接口错误中断使能 (Serial interface error interrupt enable)

该位由软件置 1 和清零，

0：禁止中断。

1：当 SPDIFRX_SR 寄存器中 SERR = 1、TERR = 1 或 FERR = 1 时，生成 SPDIFRX 接口中断。

位 5 **SYNCDIE**：同步完成 (Synchronization done)

该位由软件置 1 和清零，

0：禁止中断。

1：当 SPDIFRX_SR 寄存器中 SYNCD = 1 时，生成 SPDIFRX 接口中断。

位 4 **SBLKIE**：同步块检测中断使能 (Synchronization block detected interrupt enable)

该位由软件置 1 和清零，

0：禁止中断。

1：当 SPDIFRX_SR 寄存器中 SBD = 1 时，生成 SPDIFRX 接口中断。

位 3 **OVRIE**：上溢错误中断使能（Overrun error interrupt enable）

该位由软件置 1 和清零，

0：禁止中断。

1：当 SPDIFRX_SR 寄存器中 OVR = 1 时，生成 SPDIFRX 接口中断。

位 2 **PERRIE**：奇偶校验错误中断使能（Parity error interrupt enable）

该位由软件置 1 和清零，

0：禁止中断。

1：当 SPDIFRX_SR 寄存器中 PERR = 1 时，生成 SPDIFRX 接口中断。

位 1 **CSRNEIE**：控制缓冲区就绪中断使能 (Control buffer ready interrupt enable)
该位由软件置 1 和清零，
0：禁止中断。
1：当 SPDIFRX_SR 寄存器中 CSRNE = 1 时，生成 SPDIFRX 接口中断。

位 0 **RXNEIE**：RXNE 中断使能 (RXNE interrupt enable)
该位由软件置 1 和清零，
0：禁止中断。
1：当 SPDIFRX_SR 寄存器中 RXNE = 1 时，生成 SPDIFRX 接口中断。

52.5.3 SPDIFRX 状态寄存器 (SPDIFRX_SR)

SPDIFRX status register

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	WIDTH5[14:0]														
	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TERR	SERR	FERR	SYNCD	SBD	OVR	PERR	CSRNE	RXNE
							r	r	r	r	r	r	r	r	r

位 31 保留，必须保持复位值。

位 30:16 **WIDTH5[14:0]**：使用 spdifrx_ker_ck 计数 5 个符号的持续时间 (Duration of 5 symbols counted with spdifrx_ker_ck)
该值表示 5 个连续符号的时间内包含的 spdifrx_ker_ck 时钟周期数。该值可用于估算 S/PDIF 符号率。其精度受 spdifrx_ker_ck 频率的限制。
例如，如果 spdifrx_ker_ck 固定为 84 MHz，则 WIDTH5 = 147d。S/PDIF 数据流的采样率估算值为：
$Fs = 5 \times F_{spdifrx_ker_ck} / (WIDTH5 \times 64) \sim 44.6$ kHz，因此最接近的标准采样率为 44.1 kHz。
请注意，当 SYNCD 变为高电平时，WIDTH5 将由硬件更新，随后每帧都更新。

位 15:9 保留，必须保持复位值。

位 8 **TERR**：超时错误 (Time-out error)
当计数器 TRCNT 值达到最大值时，该位由硬件置 1。它表示两次转换之间的时间间隔过长。这通常意味着 SPDIFRX_IN 输入上没有有效信号。
该标志通过将 SPDIFRXEN 写 0 来清零。
如果 SPDIFRX_IMR 寄存器中 IFEIE=1，则会生成中断。
0：未检测到序列错误。
1：检测到序列错误。

位 7 **SERR**：同步错误 (Synchronization error)
当同步出于 NBTR 重试次数方面的原因而失败时，该位由硬件置 1。
该标志通过将 SPDIFRXEN 写 0 来清零。
如果 SPDIFRX_IMR 寄存器中 IFEIE = 1，则会生成中断。
0：未检测到同步错误。
1：检测到同步错误。

位 6 **FERR**：帧错误 (Framing error)

当在接收数据期间发生错误时，该位由硬件置 1：例如前导码未出现在预期位置，短转换未按对分组。

仅当同步完成时 (SYNCD = 1)，该位才能由硬件置 1。

该标志通过将 SPDIFRXEN 写 0 来清零。

如果 SPDIFRX_IMR 寄存器中 IFEIE=1，则会生成中断。

0：未检测到曼彻斯特编码违例

1：检测到曼彻斯特编码违例

位 5 **SYNCD**：同步完成 (Synchronization done)

当初始同步阶段正确完成时，该位由硬件置 1。

向 SPDIFRX_IFCR 寄存器中的相应位写入 1 即可将该标志清零。

如果 SPDIFRX_IMR 寄存器中 SYNCDIE = 1，则会生成中断。

0：同步挂起。

1：同步完成。

位 4 **SBD**：检测到同步块 (Synchronization block detected)

当检测到“B”前导码时，该位由硬件置 1。

向 SPDIFRX_IFCR 寄存器中的相应位写入 1 即可将该标志清零。

如果 SPDIFRX_IMR 寄存器中 SBLKIE = 1，则会生成中断。

0：未检测到“B”前导码。

1：检测到“B”前导码。

位 3 **OVR**：上溢错误 (Overrun error)

在 RXNE = 1 且 SPDIFRX_FMTx_DR 和 RX_BUF 均已满的情况下，当 SPDIFRX_FMTx_DR 寄存器中的接收数据准备好传输时，该位由硬件置 1。

向 SPDIFRX_IFCR 寄存器中的相应位写入 1 即可将该标志清零。

如果 SPDIFRX_IMR 寄存器中 OVRIE=1，则会生成中断。

0：无上溢错误

1：检测到上溢错误。

注意：当该位置 1 时，SPDIFRX_FMTx_DR 寄存器的内容不会丢失，但最后接收的数据会丢失。

位 2 **PERR**：奇偶校验错误 (Parity error)

当所接收子帧的数据和状态位中包含奇数个 0 和 1 时，该位由硬件置 1。

向 SPDIFRX_IFCR 寄存器中的相应位写入 1 即可将该标志清零。

如果 SPDIFRX_IMR 寄存器中 PIE = 1，则会生成中断。

0：无奇偶校验错误

1：奇偶校验错误

位 1 **CSRNE**：控制缓冲寄存器非空 (Control buffer register not empty)

当有效的控制信息就绪时，该位由硬件置 1。

当读取 SPDIFRX_CSR 寄存器时，该标志清零。

如果 SPDIFRX_IMR 寄存器中 CBRDYIE = 1，则会生成中断。

0：SPDIFRX_CSR 寄存器中无控制字可用

1：SPDIFRX_CSR 寄存器中有控制字可用

位 0 **RXNE**：读取数据寄存器不为空 (Read data register not empty)

当有效数据进入 SPDIFRX_FMTx_DR 寄存器时，该位由硬件置 1。

可通过读取 SPDIFRX_FMTx_DR 寄存器清零该标志。

如果 SPDIFRX_IMR 寄存器中 RXNEIE=1，则会生成中断。

0：未接收到数据。

1：已准备好读取接收到的数据

52.5.4 SPDIFRX 中断标志清零寄存器 (SPDIFRX_IFCR)

SPDIFRX interrupt flag clear register

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYNCDCF	SBDCF	OVRCF	PERRCF	Res.	Res.
										w	w	w	w		

位 31:6 保留，必须保持复位值。

位 5 **SYNCDCF**：清零同步完成标志 (clears the synchronization done flag)
向该位写入 1 可清零 SPDIFRX_SR 寄存器中的 SYNCD 标志。
读取该位将始终返回值 0。

位 4 **SBDCF**：清零同步块检测标志 (clears the synchronization block detected flag)
向该位写入 1 可清零 SPDIFRX_SR 寄存器中的 SBD 标志。
读取该位将始终返回值 0。

位 3 **OVRCF**：清零上溢错误标志 (clears the overrun error flag)
向该位写入 1 可清零 SPDIFRX_SR 寄存器中的 OVR 标志。
读取该位将始终返回值 0。

位 2 **PERRCF**：清零奇偶校验错误标志 (clears the parity error flag)
向该位写入 1 可清零 SPDIFRX_SR 寄存器中的 PERR 标志。
读取该位将始终返回值 0。

位 1:0 保留，必须保持复位值。

52.5.5 SPDIFRX 数据输入寄存器 (SPDIFRX_FMT0_DR)

SPDIFRX data input register

偏移地址：0x10

复位值：0x0000 0000

根据 DRFMT 的值，该寄存器可采用三种不同格式。以下是 DRFMT = 00 时的格式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	PT[1:0]		C	U	V	PE	DR[23:16]							
		r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:30 保留，必须保持复位值。

位 29:28 **Pt[1:0]：前导码类型**：前导码类型 (preamble type)
这些位表示接收到的前导码。
00：未使用
01：接收到前导码 B
10：接收到前导码 M
11：接收到前导码 W
请注意，如果 PTMSK = 1，该位字段将强制被设置为零

位 27 **C**：通道状态位 (channel status bit)
如果 CUMSK = 0，则包含接收到的通道状态位，否则被强制设置为 0

位 26 **U**：用户位 (user bit)
如果 CUMSK = 0，则包含接收到的用户位，否则被强制设置为 0

位 25 **V**：有效性位 (validity bit)
如果 VMSK = 0，则包含接收到的有效性位，否则被强制设置为 0

位 24 **PE**：奇偶校验错误位 (parity error bit)
如果 PMSK = 0，则包含 PERR 位的副本，否则被强制设置为 0

位 23:0 **DR[23:0]**：数据值 (data value)
包含 24 个接收到的数据位，与 D[23] 对齐

52.5.6 SPDIFRX 数据输入寄存器 (SPDIFRX_FMT1_DR)

SPDIFRX data input register

偏移地址：0x10

复位值：0x0000 0000

根据 DRFMT 的值，该寄存器可采用三种不同格式。以下是 DRFMT = 01 时的格式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DR[23:8]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[7:0]								Res.	Res.	PT[1:0]		C	U	V	PE
r	r	r	r	r	r	r	r			r	r	r	r	r	r

位 31:8 **DR[23:0]**：数据值 (data value)
包含 24 个接收到的数据位，与 D[23] 对齐

位 7:6 保留，必须保持复位值。

位 5:4 **Pt[1:0]：前导码类型**：前导码类型 (preamble type)
这些位表示接收到的前导码。
00：未使用
01：接收到前导码 B
10：接收到前导码 M
11：接收到前导码 W
请注意，如果 PTMSK = 1，该位字段将强制被设置为零

位 3 **C**：通道状态位 (channel Status bit)
如果 CUMSK = 0，则包含接收到的通道状态位，否则被强制设置为 0

位 2 **U**：用户位 (user bit)
如果 CUMSK = 0，则包含接收到的用户位，否则被强制设置为 0

位 1 **V**：有效性位 (validity bit)
如果 VMSK = 0，则包含接收到的有效性位，否则被强制设置为 0

位 0 **PE**：奇偶校验错误位 (parity error bit)
如果 PMSK = 0，则包含 PERR 位的副本，否则被强制设置为 0

52.5.7 SPDIFRX 数据输入寄存器 (SPDIFRX_FMT2_DR)

SPDIFRX data input register

偏移地址：0x10

复位值：0x0000 0000

根据 DRFMT 的值，该寄存器可采用 3 种不同格式。

DRFMT = 10 时的数据格式专用于非线性模式，因为仅使用了 16 位（S/PDIF 子帧中的位 23 到位 8）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRNL2[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRNL1[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 **DRNL2[15:0]**：数据值 (data value)
该位字段包含通道 A

位 15:0 **DRNL1[15:0]**：数据值 (data value)
该位字段包含通道 B

52.5.8 SPDIFRX 通道状态寄存器 (SPDIFRX_CSR)

SPDIFRX channel status register

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOB	CS[7:0]							
							r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:25 保留，必须保持复位值。

位 24 **SOB**：块开始 (start of block)
该位表示位 CS[0] 是否对应于新块的第一位
0：CS[0] 不是新块的第一位
1：CS[0] 是新块的第一位

位 23:16 **CS[7:0]**：通道 A 状态信息 (channel A status information)
位 CS[0] 是最早的值

位 15:0 **USR[15:0]**：用户数据信息 (user data information)
位 USR[0] 是最早的值，来自通道 A，USR[1] 来自通道 B。
因此，n 为偶数时，USR[n] 位来自通道 A，否则来自通道 B。

52.5.9 SPDIFRX 调试信息寄存器 (SPDIFRX_DIR)

SPDIFRX debug information register

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	TLO[12:0]												
			r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	THI[12:0]												
			r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:29 保留，必须保持复位值。

位 28:16 **TLO[12:0]**：阈值下限 (threshold LOW) (TLO = 1.5 x UI / $T_{spdifrx_ker_ck}$)
该位字段包含当前阈值下限的估算值。该值可用于估算接收数据流的采样率。TLO 的精度受限于 spdifrx_ker_ck 的周期。采样率按如下公式估算：
采样率 = [2 x TLO x $T_{spdifrx_ker_ck}$ +/- $T_{spdifrx_ker_ck}$] x 2/3
请注意，当 SYNCD 变为高电平时，TLO 将由硬件更新，随后每帧都更新。

位 15:13 保留，必须保持复位值。

位 12:0 **THI[12:0]**：阈值上限 (threshold HIGH) (THI = 2.5 x UI / $T_{spdifrx_ker_ck}$)
该位字段包含当前阈值上限的估算值。该值可用于估算接收数据流的采样率。THI 的精度受限于 spdifrx_ker_ck 的周期。采样率按如下公式估算：
采样率 = [2 x THI x $T_{spdifrx_ker_ck}$ +/- $T_{spdifrx_ker_ck}$] x 2/5
请注意，当 SYNCD 变为高电平时，THI 将由硬件更新，随后每帧都更新。

52.5.10 SPDIFRX 版本寄存器 (SPDIFRX_VERR)

SPDIFRX version register

偏移地址：0x3F4

复位值：0x0000 0012

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：SPDIFRX 主版本 (SPDIFRX major revision)
主版本为 1。

位 3:0 **MINREV[3:0]**：SPDIFRX 次版本 (SPDIFRX minor revision)
次版本为 2。

52.5.11 SPDIFRX 标识寄存器 (SPDIFRX_IPIDR)

SPDIFRX identification register

偏移地址：0x3F8

复位值：0x0013 0041

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：SPDIFRX 标识符 (SPDIFRX identifier)

52.5.12 SPDIFRX 大小标识寄存器 (SPDIFRX_SIDR)

SPDIFRX size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识 (Size identification)

这些位返回分配给 SPDIFRX 寄存器的存储区域的大小。

该存储区域的大小为 1 KB。

52.5.13 SPDIFRX 接口寄存器映射

表 473. SPDIFRX 接口寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SPDIFRX_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKSBKPEN	CKSEN	Res.	INSEL[2:0]			Res.	WFA	NBTR[1:0]		CHSEL	CBDMAEN	PTMSK	CUMSK	VMSK	PMSK	DRFMT[1:0]		RXSTEO	RXDMAEN	SPDIFRXEN[1:0]	
	Reset value											0	0		0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	SPDIFRX_IMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IFEIE	SYNCDIE	SBLKIE	OVRIE	PERRIE	CSRNEIE	RXNEIE
	Reset value																										0	0	0	0	0	0	0
0x08	SPDIFRX_SR	Res.	WIDTH5[14:0]															Res.	Res.	Res.	Res.	Res.	Res.	Res.	TERR	SERR	FERR	SYNCD	SBD	OVR	PERR	CSRNE	RXNE
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								0	0	0	0	0	0	0	0	0
0x0C	SPDIFRX_IFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYNCDCF	SBDCF	OVRCF	PERRCF	Res.	Res.
	Reset value																											0	0	0	0		
0x10	SPDIFRX_FMT0_DR	Res.	Res.	PT[1:0]		C	U	V	PE	DR[23:0]																							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	SPDIFRX_FMT1_DR	DR[23:0]																								Res.		PT[1:0]		C	U	V	PE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 473. SPDIFRX 接口寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x10	SPDIFRX_FMT2_DR	DRNL2[15:0]																DRNL1[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	SPDIFRX_CSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOB	CS[7:0]								USR[15:0]															
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	SPDIFRX_DIR	Res.	Res.	Res.	TLO[12:0]													Res.	Res.	Res.	THI[12:0]												
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0
0x3F4	SPDIFRX_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV [3:0]				MINREV [3:0]			
	Reset value																									0	0	0	1	0	0	1	0
0x3F8	SPDIFRX_IPIDR	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
0x3FC	SPDIFRX_SIDR	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 2.5 节*。

53 具有灵活数据速率的控制器局域网络 (FDCAN)

53.1 简介

控制器局域网络 (CAN) 子系统（请参见*图 664*）由两个 CAN 模块、一个共享消息 RAM 和一个时钟校准单元组成。关于这些组成部分的基地址，请参见产品存储映射结构。

FDCAN 模块符合 ISO 11898-1: 2015（CAN 协议规范第 2.0 版 A、B 部分）和 CAN FD 协议规范第 1.0 版。

此外，第一个 CAN 模块 FDCAN1 支持 ISO 11898-4 中规定的时间触发 CAN (TTCAN)，包括事件同步时间触发通信、全局系统时间和时钟漂移补偿。FDCAN1 还额外包含专供时间触发功能使用的寄存器。CAN FD 可与事件触发和时间触发 CAN 通信一起使用。

一个 10 KB 的消息 RAM 实现了过滤器、接收 FIFO、接收缓冲区、发送事件 FIFO、发送缓冲器（TTCAN 触发）功能。这些内存是 FDCAN 模块之间共享的。

通用时钟校准单元是可选的。通过评估 FDCAN1 接收到的 CAN 消息，该单元可基于 HSI 内部 RC 振荡器和 PLL 为各个 FDCAN 生成经过校准的时钟。

表 474 和*表 475* 分别对 CAN 子系统 I/O 信号和引脚进行了详细介绍。

表 474. CAN 子系统 I/O 信号

名称	类型	说明
fdcan_ker_ck	数字输入	CAN 子系统内核时钟输入
fdcan_pclk		CAN 子系统 APB 接口时钟输入
fdcan_cal_it	数字输出	FDCAN 校准中断
fdcan1_intr0_it	数字输出	FDCAN1 中断 0
fdcan1_intr1_it		FDCAN1 中断 1
fdcan2_intr0_it	数字输出	FDCAN2 中断 0
fdcan2_intr1_it		FDCAN2 中断 1
fdcan1_swt[0:3]	数字输入	秒表触发输入
fdcan1_evt[0:3]		事件触发输入
fdcan1_ts[0:15]		外部时间戳向量
fdcan1_soc	数字输出	周期开始脉冲
fdcan1_rtp		寄存器时间标记脉冲
fdcan1_tmp		触发时间标记脉冲

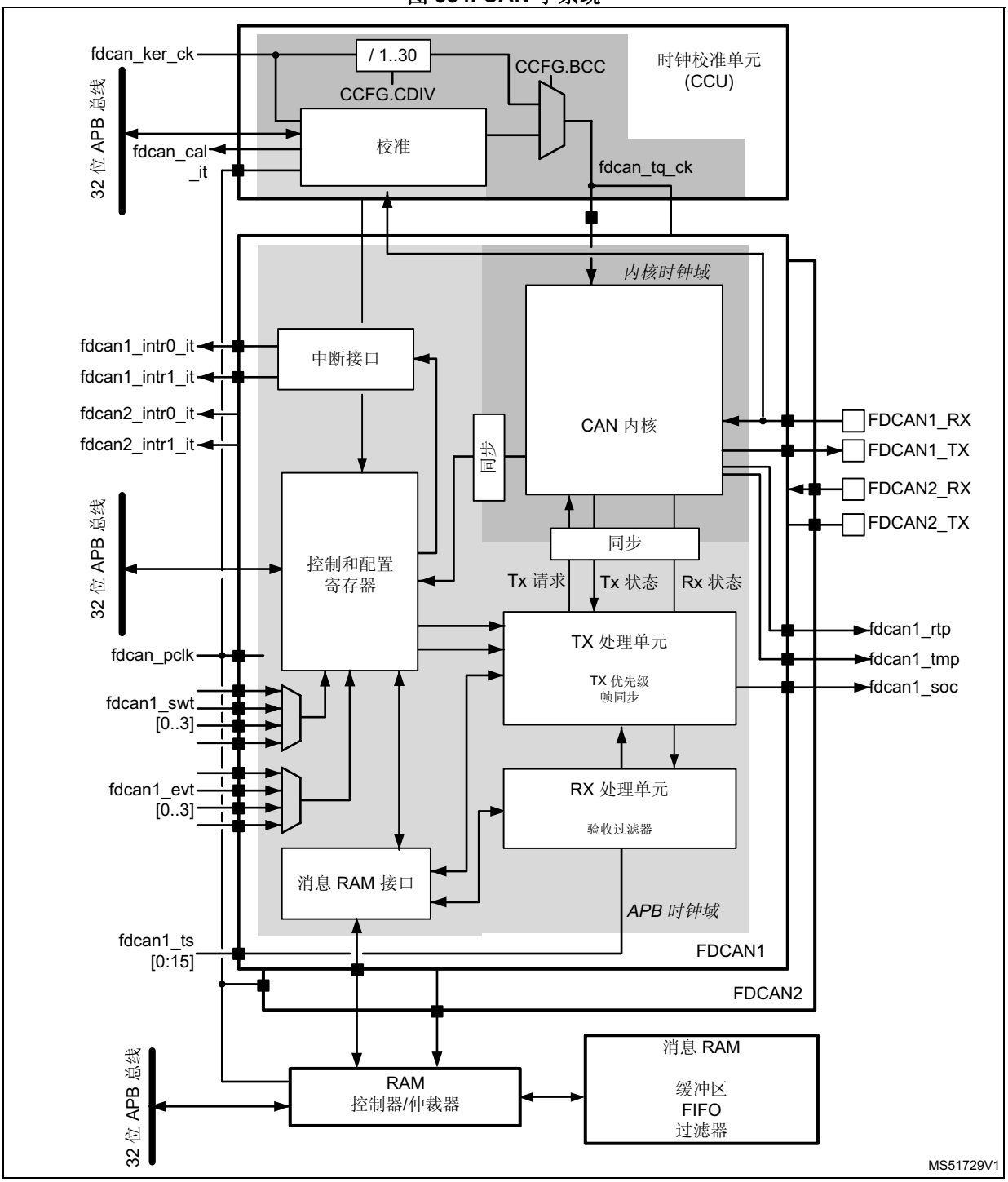
表 475. CAN 子系统 I/O 引脚

名称	类型	说明
FDCAN1_RX	数字输入	FDCAN1 接收引脚
FDCAN1_TX	数字输出	FDCAN1 发送引脚
FDCAN2_RX	数字输入	FDCAN2 接收引脚
FDCAN2_TX	数字输出	FDCAN2 发送引脚

表 476. CAN 触发器

类型	ID	源	目标
内部	SWT0	TIM2_TRGO	-
	SWT1	TIM3_TRGO	-
	SWT2	ETH1_PPS	-
	SWT3	ETH2_PPS	-
	EVT0	TIM2_TRGO	-
	EVT1	TIM3_TRGO	-
	EVT2	ETH1_PPS	-
	EVT3	ETH2_PPS	-
外部	fdcan_tt_soc	-	TIM5_ITR6
	fdcan_tt_tmp	-	TIM5_TI1_1、ETH1_PTP3、ETH2_PTP3
	fdcan_tt_rtp	-	TIM5_TI1_2

图 664. CAN 子系统


fdcan_ker_ck
/ 1..30
CCFG.CDIV
CCFG.BCC
时钟校准单元
(CCU)
32 位 APB 总线
fdcan_cal
_it
校准
fdcan_tq_ck
内核时钟域
fdcan1_intr0_it
fdcan1_intr1_it
fdcan2_intr0_it
fdcan2_intr1_it
中断接口
CAN 内核
同步
FDCAN1_RX
FDCAN1_TX
FDCAN2_RX
FDCAN2_TX
32 位 APB 总线
控制和配置
寄存器
同步
Tx 请求
Tx 状态
Rx 状态
fdcan_pclk
TX 处理单元
TX 优先级
帧同步
fdcan1_rtp
fdcan1_tmp
fdcan1_soc
fdcan1_swt
[0..3]
fdcan1_evt
[0..3]
RX 处理单元
验收过滤器
消息 RAM 接口
APB 时钟域
fdcan1_ts
[0:15]
FDCAN1
FDCAN2
32 位 APB 总线
RAM
控制器/仲裁器
消息 RAM
缓冲区
FIFO
过滤器
MS51729V1

53.2 FDCAN 主要特性

- 符合 CAN 协议第 2.0 版 A、B 部分和 ISO 11898-1:2015, -4
- CAN FD 最大支持 64 Byte 数据负载
- 硬件完全支持 TTCAN 协议级别 1 和级别 2（仅限 FDCAN1）
- 支持事件同步时间触发通信（仅限 FDCAN1）
- CAN 错误记录
- 支持 AUTOSAR 和 J1939
- 改进了接收过滤
- 两个可配置接收 FIFO
- 接收到高优先级消息时单独发出信号指示
- 多达 64 个专用接收缓冲区
- 多达 32 个专用发送缓冲区
- 可配置发送 FIFO/队列
- 可配置发送事件 FIFO
- 多个 FDCAN 模块共用同一个消息 RAM
- 可编程环回测试模式
- 可屏蔽模块中断
- 两个时钟域：APB 总线接口和 CAN 内核时钟
- 支持掉电

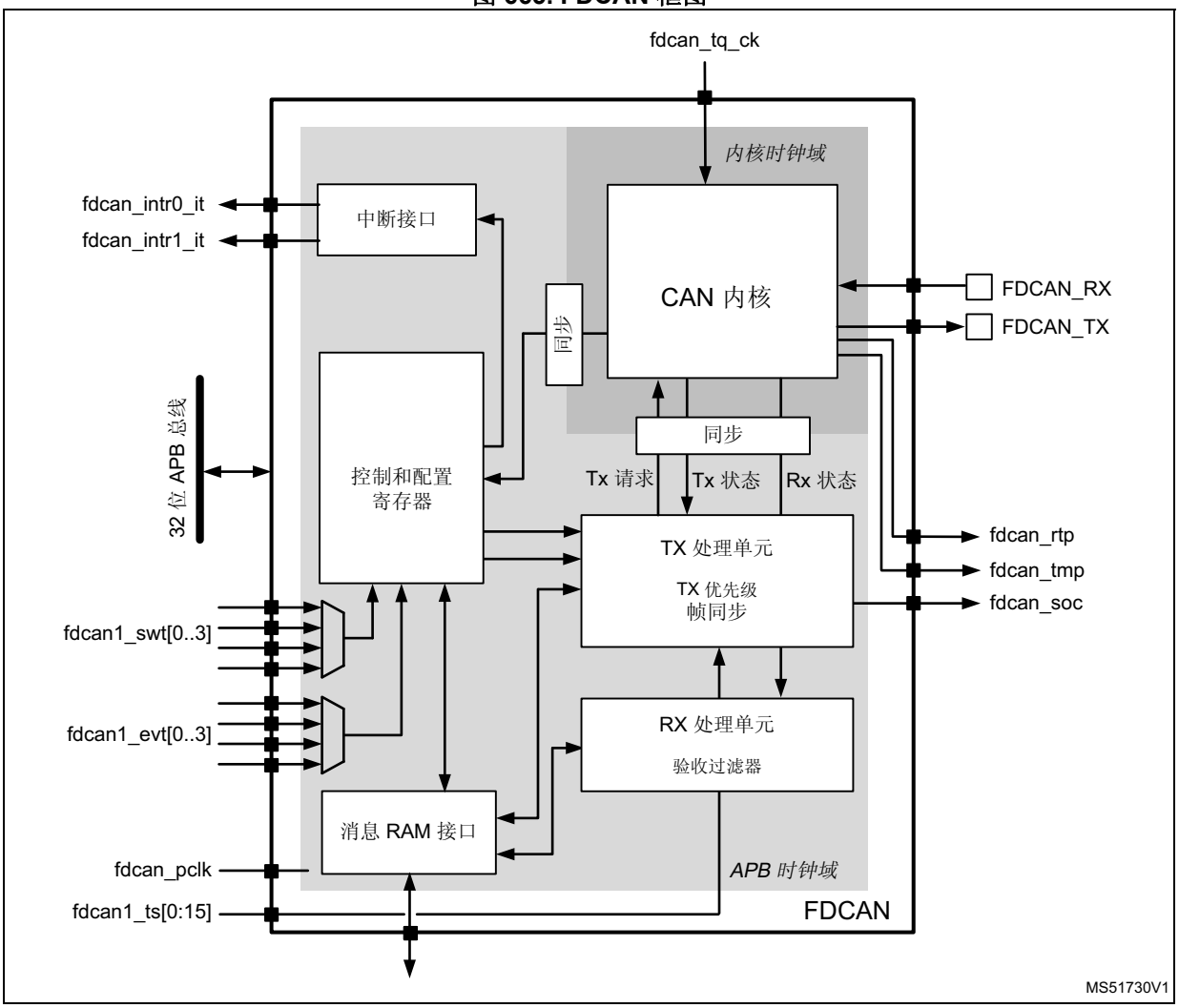
53.3 FDCAN 实现

表 477. 主要特性

模块	FDCAN1	FDCAN2
TTCAN	X	-

53.4 FDCAN 功能描述

图 665. FDCAN 框图



双中断线

FDCAN 外设提供 fdcan_intr0_it 和 fdcan_intr1_it 两条中断线。通过对 FDCAN_ILE 寄存器中的 EINT0 和 EINT1 位进行编程，可分别使能或禁止这两条中断线。

CAN 内核

CAN 内核包含协议控制器和接收/发送移位寄存器。它可处理所有 ISO 11898-1:2015 协议功能，并支持 11 位和 29 位标识符。

同步

该模块将来自 APB 时钟域的信号同步到 CAN 内核时钟域，以及将来自 CAN 时钟域的信号同步到 APB 内核时钟域。

发送处理单元

该处理单元控制从消息 RAM 到 CAN 内核的消息传输。最多可配置 32 个发送缓冲区进行发送。发送缓冲区可用作专用发送缓冲区、发送 FIFO（发送队列的组成部分）或二者的组合。发送事件 FIFO 会将发送时间戳与相应的消息 ID 存储在一起，另外还支持取消发送。

在 FDCAN1 上，发送处理单元还会实现帧同步实体 (FSE)，按照 ISO11898-4 的规定控制时间触发通信。它会将自身与 CAN 总线上的参考消息进行同步，控制周期时间和全局时间，并根据预先定义的消息调度（系统矩阵）处理发送操作。它还处理系统矩阵中与消息 RAM 内的消息相关联的时间标记。秒表触发、事件触发和时间标记中断是同步接口。

接收处理单元

该处理单元控制着从 CAN 内核接收的消息到外部消息 RAM 的传输。接收处理单元支持两个接收 FIFO，每一个的大小均可配置，并且最多有 64 个专用接收缓冲区用于存储所有已通过验收过滤的消息。专用接收缓冲区仅用于存储具有特定标识符的消息，与接收 FIFO 有所不同。每条消息均与其接收时间戳存储在一起。对于 11 位 ID，最多可定义 128 个过滤器；对于 29 位 ID，最多可定义 64 个过滤器。

APB 接口

将 FDCAN 连接至 APB 总线。

消息 RAM 接口

通过 RAM 控制器/仲裁器将 FDCAN 访问连接到外部 10 KB 消息 RAM。

53.4.1 工作模式

软件初始化

要进行软件初始化，首先应通过软件或硬件复位、或者进入 Bus_Off 状态将 FDCAN_CCCR 寄存器中的 INIT 位置 1。当 FDCAN_CCCR 寄存器中的 INIT 位置 1 时，从 CAN 总线传入和传出的消息都将停止，并且 CAN 总线输出 FDCAN_TX 的状态为隐性（高）。错误管理逻辑 (EML) 的计数器保持不变。将 FDCAN_CCCR 中的 INIT 位置 1 不会更改任何配置寄存器。将 FDCAN_CCCR 中的 INIT 位清零会结束软件初始化。随后，位流处理单元 (BSP) 会等待出现 11 个连续隐性位 (Bus_Idle) 序列，以此将其自身与 CAN 总线上的数据传输进行同步，然后才能参与总线活动并开始消息传输。

仅当 FDCAN_CCCR 寄存器中的 INIT 位以及 FDCAN_CCCR 寄存器中的 CCE 位均置 1 时，才能访问 FDCAN 配置寄存器。

FDCAN_CCCR 寄存器中的 CCE 位只能在 FDCAN_CCCR 中的 INIT 位置 1 时才能置 1/清零。FDCAN_CCCR 寄存器中的 CCE 位会在 FDCAN_CCCR 中的 INIT 位清零时自动清零。

FDCAN_CCCR 寄存器中的 CCE 位置 1 后，以下寄存器会复位：

- FDCAN_HPMS——高优先级消息状态
- FDCAN_ RXF0S——接收 FIFO 0 状态
- FDCAN_RXF1S——接收 FIFO 1 状态
- FDCAN_TXFQS——发送 FIFO/队列状态
- FDCAN_TXBRP——发送缓冲区请求挂起
- FDCAN_TXBTO——发送缓冲区发送已发生
- FDCAN_TXBCF——发送缓冲区取消完成
- FDCAN_TXEFS——发送事件 FIFO 状态
- FDCAN_TTOST——TT（时间触发）工作状态（仅限 FDCAN1）
- FDCAN_TTLGT——TT 本地事件和全局时间，仅会复位全局时间 TTLGT.GT（仅限 FDCAN1）
- FDCAN_TTCTC——TT 周期时间和计数（仅限 FDCAN1）
- FDCAN_TTCSM——TT 周期同步标记（仅限 FDCAN1）

FDCAN_CCCR 中的 CCE 位置 1 后，FDCAN_TOCV 寄存器中的超时计数器值 TOC 位会预设为由 FDCAN_TOCC 寄存器中的 TOP 位配置的值。

此外，FDCAN_CCCR 中的 CCE 位置 1 时，发送处理单元和接收处理单元中的状态机会保持空闲状态。

仅当 FDCAN_CCCR 寄存器中的 CCE 位清零时，才能写入以下寄存器：

- FDCAN_TXBAR——发送缓冲区添加请求
- FDCAN_TXBCR——发送缓冲区取消请求

仅当 FDCAN_CCCR 寄存器中的 INIT 位和 CCE 位均已置 1 时，才能通过软件将 FDCAN_CCCR 中的 TEST 位和 MON 位置 1。这两个位可随时复位。仅当 FDCAN_CCCR 中的 INIT 位和 CCE 位均已置 1 时，才能将 FDCAN_CCCR 中的 DAR 位置 1/清零。

正常工作

FDCAN1 在硬件复位后的默认工作模式是无时间触发的事件驱动 CAN 通信 (FDCAN_TTOCF.OM = 00)。要更改 TT 工作模式，需要先将 FDCAN_CCCR 寄存器中的 INIT 位和 CCE 位置 1。

FDCAN 完成初始化且 FDCAN_CCCR 寄存器中的 INIT 位清零后，FDCAN 会将其自身与 CAN 同步并准备好进行通信。

通过接收过滤后，包含消息 ID 和 DLC 的内容会存储到专用的接收缓冲区中，或者存储到接收 FIFO 0 或接收 FIFO 1 中。

发送消息时，可初始化或更新专用发送缓冲区，发送 FIFO，或发送队列。不支持在接收到远程帧后自动发送。

CAN FD 操作

CAN FD 协议有两种变体。第一种是长帧模式 (LFM)，在该模式下，CAN 帧的数据字段长度可以超过八个字节。第二种是快速帧模式 (FFM)，在该模式下，CAN 帧的控制字段、数据字段和 CRC 字段的发送比特率要高于帧开始和结束的发送比特率。快速帧模式可与长帧模式结合使用。

带 11 位标识符的 CAN 帧中的原保留位和带 29 位标识符的 CAN 帧中的第一保留位被解码为 FDF。FDF 隐性表示 CAN FD 帧，而 FDF 显性则表示经典 CAN 帧。在 CAN FD 帧中，FDF 之后的两个位 res 和 BRS 决定是否切换这一 CAN FD 帧中的比特率。CAN FD 比特率切换通过 res 显性和 BRS 隐性指示。res 隐性的编码保留，供协议扩展使用。如果 FDCAN 接收到的帧为 FDF 隐性和 res 隐性，则会通过将 FDCAN_PSR.PXE 位置 1 的方式发出协议异常事件。如果使能协议异常处理 (FDCAN_CCCR.PXHD = 0)，下一采样点的工作状态会从接收状态 (FDCAN_PSR.ACT = 10) 变为整合状态 (FDCAN_PSR.ACT = 00)。如果禁止协议异常处理 (FDCAN_CCCR.PXHD = 1)，FDCAN 会将隐性 res 位当作格式错误处理，并以错误帧进行响应。

CAN FD 操作通过编程 FDCAN_CCCR.FDOE 来使能。如果 FDCAN_CCCR.FDOE = 1，则会使能 CAN FD 帧的发送和接收。始终可进行经典 CAN 帧的发送和接收。发送 FD CAN 还是经典 CAN 帧可通过相应发送缓冲区中的 FDF 位配置。如果 FDCAN_CCCR.FDOE = 0，接收到的帧会被当作经典 CAN 帧，从而会在接收到 CAN FD 帧时发送错误帧。如果 CAN FD 操作已禁止，即使发送缓冲区元素的 FDF 位置 1，也不会发送 CAN FD 帧。仅当 FDCAN_CCCR.INIT 和 FDCAN_CCCR.CCE 均置 1 时，才能更改 FDCAN_CCCR.FDOE 和 FDCAN_CCCR.BRSE。

如果 FDCAN_CCCR.FDOE = 0，则会忽略 FDF 和 BRS 位的设置，并会以经典 CAN 格式发送帧。如果 FDCAN_CCCR.FDOE = 1 且 FDCAN_CCCR.BRSE = 0，则仅会评估发送缓冲区元素的 FDF 位。如果 FDCAN_CCCR.FDOE = 1 且 FDCAN_CCCR.BRSE = 1，则会使能具有比特率切换功能的 CAN FD 帧发送。所有 FDF 和 BRS 位置 1 的发送缓冲区均会以具有比特率切换功能的 CAN FD 格式进行发送。

建议仅当满足以下条件下在 CAN 操作期间切换模式：

- CAN FD 数据阶段的故障率显著高于 CAN FD 仲裁阶段的故障率。在这种情况下，应禁止 CAN FD 比特率切换功能来传输。
- 系统启动期间，所有节点在被验证为能够以 CAN FD 格式进行通信之前，都将发送经典 CAN 消息。如果通过验证，所有节点会切换为 CAN FD 操作。
- CAN 局部网络中的唤醒消息必须以经典 CAN 格式进行发送。
- 并非所有节点均能进行 CAN FD 操作时采用行结束编程。非 CAN FD 节点在编程完成之前会一直保持静默模式。随后，所有节点会切换回经典 CAN 通信。

在 FDCAN 格式中，DLC 的编码与经典 CAN 格式不同。DLC 代码 0 到 8 的编码与经典 CAN 相同，代码 9 到 15（在经典 CAN 中，所有代码的数据字段均为 8 个字节）是按照*表 478* 进行编码的。

表 478. FDCAN 中的 DLC 编码

DLC	9	10	11	12	13	14	15
数据字节数	12	16	20	24	32	48	64

在 CAN FD 快速帧中，如果该位为隐性，则帧内位时序将在 BRS（比特率切换）位之后切换。在 BRS 位之前，在 CAN FD 仲裁阶段，使用的是名义 CAN 位时序，由位时序和预分频器寄存器 FDCAN_NBTP 定义。在接下来的 CAN FD 数据阶段，使用的是快速位时序，由快

速位时序和预分频器寄存器 FDCAN_DBTP 定义。位时长会在出现 CRC 分隔符或检测到错误时从快速时长切换回来，以先发生的事件为准。

CAN FD 数据阶段的最大可配置比特率取决于 FDCAN 内核时钟频率。举例来说，如果 FDCAN 内核时钟频率为 20 MHz，最短可配置位时间为四个时间片 (tq)，则数据阶段的比特率为 5 Mb/s。

在 CAN FD 长帧和 CAN FD 快速帧这两种数据帧格式中，位 ESI（错误状态指示符）的值是由发送开始时的发送器错误状态决定的。如果发送器为错误被动状态，则 ESI 会隐性传送，否则会显性传送。在 CAN FD 远程帧中，ESI 位始终显性传送，与发送器错误状态无关。CAN FD 远程帧的数据长度代码以 0 的形式发送。

当 FDCAN 发送缓冲区配置为 DLC > 8 的 CAN FD 发送时，前 8 个字节会按照发送缓冲区中的配置发送，而数据字段的剩余部分会用 0xCC 填充。当 FDCAN 接收到 DLC > 8 的 FDCAN 帧时，该帧的前 8 个字节会存储到匹配的接收缓冲区或接收 FIFO 中。剩余字节会被丢弃。

收发器延迟补偿

在 CAN FD 发送的数据阶段，只有一个节点会进行发送，所有其他节点都是接收器。总线的长度没有影响。通过引脚 FDCAN_TX 进行发送时，协议控制器会通过引脚 FDCAN_RX 接收到其本地 CAN 收发器发送的数据。接收数据的延迟为 CAN 收发器环路延迟。如果该延迟大于 TSEG1（采样点之前的时间段），则会检测到位错误。如果不进行收发器延迟补偿，CAN FD 帧数据阶段的比特率会受到收发器环路延迟的限制。

FDCAN 会实施延迟补偿机制来补偿 CAN 收发器环路延迟，从而可在 CAN FD 数据阶段以较高的比特率进行发送，而不受特定 CAN 收发器延迟的影响。

为了检查发送节点的数据阶段是否存在位错误，会将延迟发送数据与在第二采样点 (SSP) 接收到的数据进行比较。如果检测到位错误，发送器将在下一常规采样点对这一位错误作出响应。在仲裁阶段，始终会禁止延迟补偿。

发送器延迟补偿会使能数据位时间短于发送器延迟的配置，新版 ISO11898-1 中进行了详细介绍。发送器延迟补偿是通过将 FDCAN_DBTP.TDC 位置 1 使能的。

接收到的位会与在 SSP 发送的位进行比较。SSP 位置定义为从 FDCAN 发送输出引脚 FDCAN_TX 通过收发器到接收输入引脚 FDCAN_RX 测得的延迟与由 FDCAN_TDCR.TDCO 配置的发送器延迟补偿偏移之和。发送器延迟补偿偏移用于调整已接收位中 SSP 的位置（例如数据阶段的半个位时间）。第二采样点的位置会向下舍入到下一个最小时间片（mtq，即一个 fdcan_tq_ck 时钟周期）的整数。

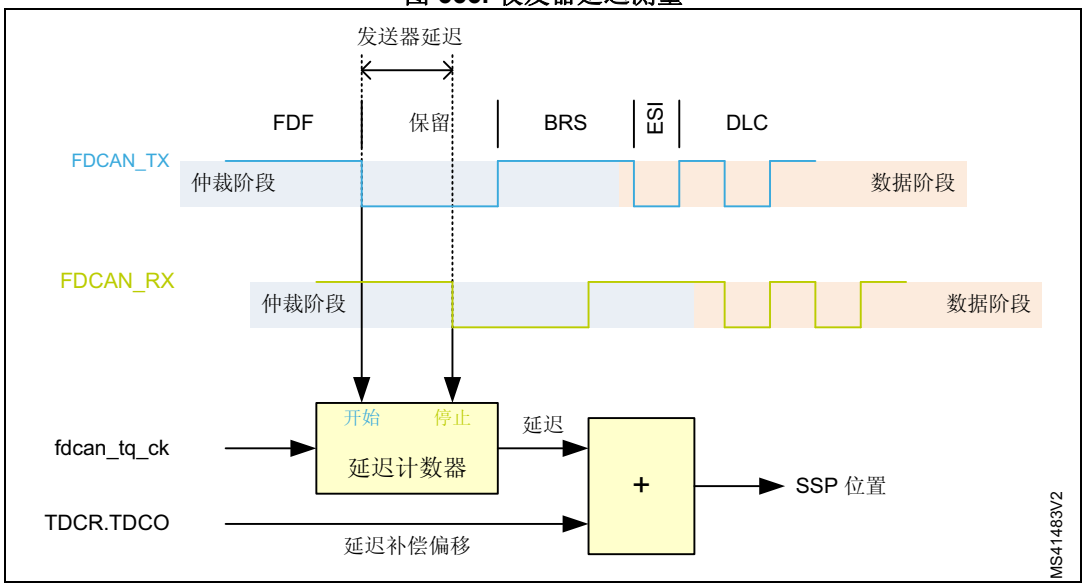
FDCAN_PSR.TDCV 显示实际的发送延迟补偿值，当 FDCAN_CCCR.INIT 置 1 时清零，如果 FDCAN_DBTP.TDC 置 1，则每次发送 FD 帧时都会更新。

对于在 FDCAN 中实施的发送器延迟补偿，必须考虑以下边界条件：

- 测得的 FDCANx_TX 到 FDCANx_RX 的延迟与配置的发送器延迟补偿偏移 FDCAN_TDCR.TDCO 之和必须小于数据阶段的 6 个位时间。
- 测得的 FDCANx_TX 到 FDCANx_RX 的延迟与配置的发送器延迟补偿偏移 FDCAN_TDCR.TDCO 之和必须小于或等于 127 mtq。如果二者之和超过 127 mtq，则为发送器延迟补偿使用最大值 (127 mtq)。
- 数据阶段在 CRC 分隔符的采样点结束，此时会停止检查在 SSP 接收到的位。

如果通过编程 FDCAN_DBTP.TDC = 1 使能发送器延迟补偿，则会在每个 FDF 位到 res 位的下降沿处发送的 CAN FD 帧中开始测量。在发送器的接收输入引脚 FDCAN_TX 上观察到该边沿时，会停止测量。该测量的分辨率为一个 mtq。

图 666. 收发器延迟测量



为了避免接收到的 FDF 位中的显性噪声信号导致在接收到的 res 位下降沿之前结束延迟补偿测量（造成 SSP 位置过早），可通过编程 FDCAN_TDCR.TDCF 来使能发送器延迟补偿过滤器窗口，从而定义最小 SSP 位置值。进行发送器延迟测量时，会忽略可能导致 SSP 位置更加提前的 FDCANx_RX 上的显性边沿。当 SSP 位置至少为 FDCAN_TDCR.TDCF 设定值且 FDCAN_RX 为低电平时，会停止测量。

受限工作模式

在受限工作模式下，节点能够接收数据帧和远程帧，并能对有效帧进行确认，但不会发送数据帧、远程帧、有效错误帧或过载帧。如果存在错误条件或过载条件，则不会发送显性位，而会等待出现总线空闲条件，以便使自身与 CAN 通信重新同步。错误记录 (FDCAN_ECR.CEL) 激活时，错误计数器（FDCAN_ECR.REC 和 FDCAN_ECR.TEC）会冻结。软件可通过将 FDCAN_CCCR.ASM 位置 1 的方式将 FDCAN 设置为进入受限工作模式。仅当 FDCAN_CCCR.CCE 和 FDCAN_CCCR.INIT 均设为 1 时，才能通过软件将该位置 1。该位可随时通过软件清零。

当发送处理单元无法及时从消息 RAM 中读取数据时，会自动进入受限工作模式。要退出受限工作模式，软件必须复位 FDCAN_CCCR.ASM。

受限工作模式可用于根据不同 CAN 比特率对自身进行调整的应用中。在这种情况下，应用会测试不同比特率，并在收到有效帧后退出受限工作模式。

FDCAN_CCCR.ASM 还受时钟校准单元控制。时钟校准过程使能后，会进入受限工作模式，FDCAN_CCR.ASM 位会置 1。校准完成后，FDCAN_CCR.ASM 位会清零。

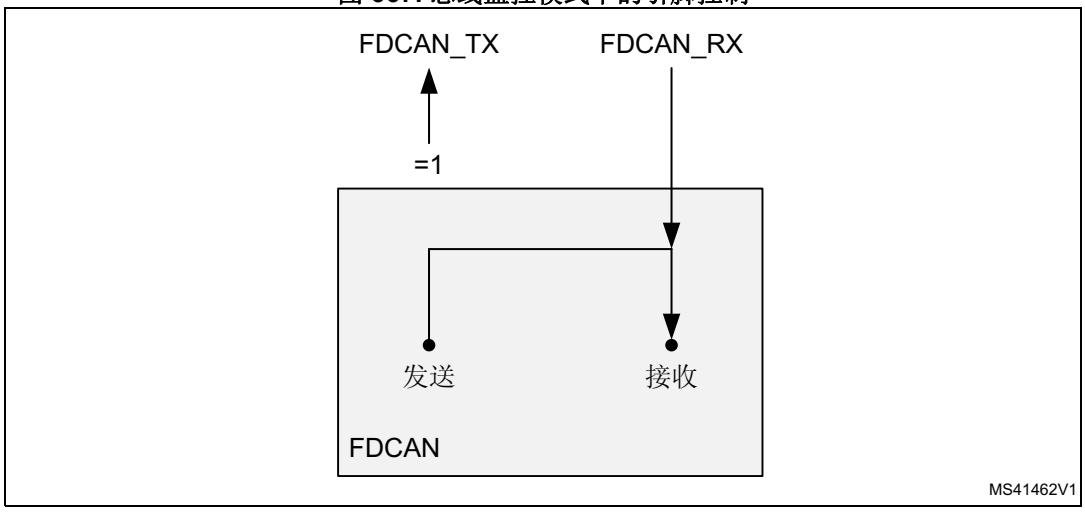
注： *受限工作模式不得与环回模式（内部或外部）结合使用。*

总线监控模式

将 FDCAN_CCCR.MON 位置 1 或进入错误级别 S3 (FDCAN_TTOST.EL = 11) 时，FDCAN 会被置于总线监控模式。在总线监控模式下（更多详细信息，请参见 ISO11898-1，10.12 总线监控），FDCAN 能够接收有效数据帧和有效远程帧，但不能启动发送过程。在该模式下，FDCAN 仅会在 CAN 总线上发送隐性位，如果 FDCAN 必须发送一个显性位（ACK 位、过载标志、活动错误标志），该位将在内部被改道发送，以便 FDCAN 可以监视该显性位，但 CAN 总线可以保持隐性状态。在总线监控模式下，寄存器 FDCAN_TXBRP 会保持复位状态。

总线监控模式可用于分析 CAN 总线上的流量，同时又不会因发送显性位对其造成影响。*图 667* 显示了总线监控模式下 FDCAN_TX 和 FDCAN_RX 信号与 FDCAN 的连接。

图 667. 总线监控模式下的引脚控制



禁止自动重发送 (DAR) 模式

根据 CAN 规范（请参见 ISO11898-1，6.3.3 恢复管理），FDCAN 提供的方法可自动重发仲裁失败的帧或在发送期间受到错误干扰的帧。默认使能自动重发送。

为支持 ISO 11898-1: 2015 第 9.2 章所描述的时间触发通信，可以通过 FDCAN_CCCR.DAR 禁止自动重发送。

禁止自动重发送 (DAR) 模式下的帧发送

在 DAR 模式下，所有 CAN 总线启动的发送都会自动取消。发送缓冲区发送请求挂起位 FDCAN_TXBRP.TRPx 会在成功发送后复位，如果取消时发送尚未开始、发送已因仲裁失败而中止，或者帧发送期间出错，该位也会复位。

- 成功发送：
  - 相应的发送缓冲区已发送位 FDCAN_TXBTO.TOx 置 1
  - 相应的发送缓冲区已取消位 FDCAN_TXBCF.CFx 不置 1
- 即使已取消发送也会成功进行发送：
  - 相应的发送缓冲区已发送位 FDCAN_TXBTO.TOx 置 1
  - 相应的发送缓冲区已取消位 FDCAN_TXBCF.CFx 置 1
- 仲裁丢失或帧发送受到干扰：
  - 相应的发送缓冲区已发送位 FDCAN_TXBTO.TOx 不置 1
  - 相应的发送缓冲区已取消位 FDCAN_TXBCF.CFx 置 1

对于帧发送成功的情况，如果使能了发送事件存储，则会记录一个发送事件，写入的事件类型 ET = 10（即使已取消也会进行发送）。

掉电（休眠模式）

FDCAN 可通过寄存器 FDCAN_CCCR.CSR 设置进入掉电模式，该模式由时钟停止请求输入控制。只要时钟停止请求生效，FDCAN_CCCR.CSR 位就读为 1。

所有挂起发送请求完成后，FDCAN 会一直等待至检测到总线空闲状态。检测到总线空闲状态之后，FDCAN 会将 FDCAN_CCCR.INIT 置 1，以免继续进行 CAN 传输。此时，FDCAN 会通过将 FDCAN_CCCR.CSA 置 1 确认其准备好进入掉电状态。在该模式下，时钟关闭之前，可继续访问寄存器。对 FDCAN_CCCR.INIT 的写访问不起作用。此时，可以关闭模块时钟输入。

退出掉电模式之前，应用必须在复位 CC 控制寄存器标志 FDCAN_CCCR.CSR 之前开启模块时钟。FDCAN 将通过复位 FDCAN_CCCR.CSA 确认时钟开启。之后，应用可通过复位 FDCAN_CCCR.INIT 位重新开始 CAN 通信。

测试模式

要使能对 FDCAN 测试寄存器的写访问（请参见*第 53.5.4 节*），必须将 FDCAN_CCCR.TEST 位置 1，从而能够配置测试模式和功能。

可通过编程 FDCAN_TEST.TX 将四种输出功能用于 CAN 发送引脚 FDCAN_TX。除了默认功能（串行数据输出）之外，还可驱动 CAN 采样点信号来监控 FDCAN 位时长，并可驱动常量显性或隐性值。引脚 FDCAN_RX 的实际值可从 FDCAN_TEST.RX 读取。这两种功能均可用于检查 CAN 总线物理层。

由于 CAN 内核时钟与 APB 时钟域之间的同步机制，在 FDCAN_TX 输出引脚上显示新配置之前，对 FDCAN_TEST.TX 的写操作之间可能有几个 APB 时钟周期的延迟。这一点还适用于通过 FDCAN_TEST.RX 读取 FDCAN_RX 输入引脚的情况。

注： *测试模式仅限用于生产测试和自检。对 FDCAN_TX 引脚进行软件控制会干扰所有 CAN 协议功能。不建议为应用使用测试模式。*

外部环回模式

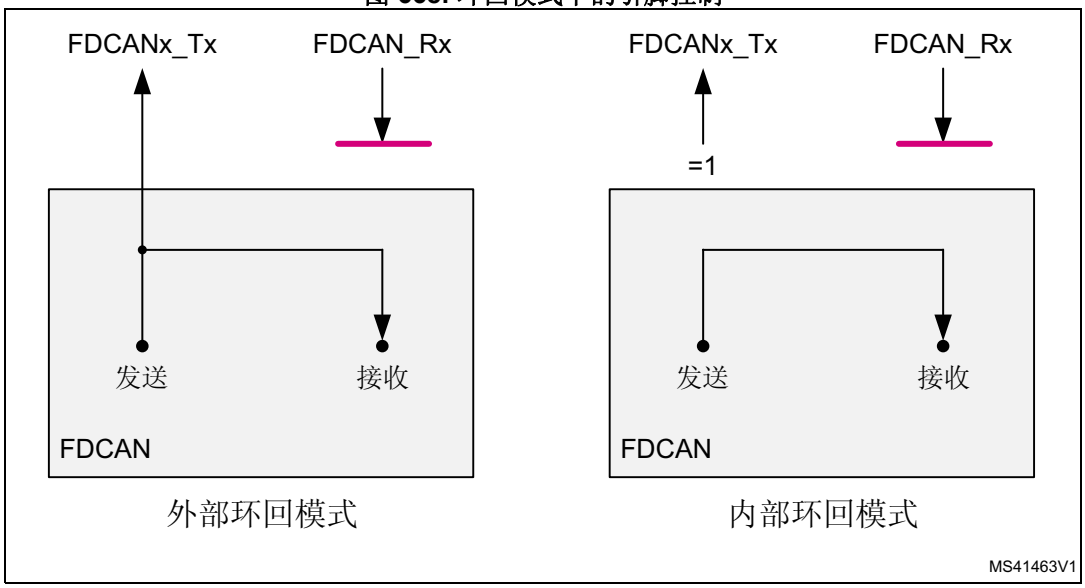
在外部环回模式下，可通过将 FDCAN_TEST.LBCK 编程为 1 来将 FDCAN 置 1。在环回模式下，FDCAN 将其自身发送的消息作为接收的消息来处理，并将消息存储（如果这些消息通过了验收过滤）在接收 FIFO 中。*图 668*（左侧）显示了外部环回模式下发送和接收信号 FDCAN_TX 和 FDCAN_RX 与 FDCAN 的连接。

该模式为硬件自检提供。为了不受外部仿真的影响，FDCAN 在环回模式下将忽略确认错误（即在数据/远程帧的确认时隙采样到的隐性位）。在此模式下，FDCAN 将执行从发送输出到其接收输入的内部反馈。FDCAN 将忽略 FDCAN_RX 输入引脚的实际值。从 FDCAN_TX 发送引脚可以监视发送的消息。

内部环回模式

将 FDCAN_TEST.LBCK 和 FDCAN_CCCR.MON 位编程为 1 会进入内部环回模式。该模式可用于“热自检”，也就是说，FDCAN 可以进行检测，同时又不会影响与 FDCAN_TX 和 FDCAN_RX 引脚相连接的运行中的 CAN 系统。在此模式下，FDCAN_RX 引脚与 FDCAN 断开连接，FDCAN_TX 引脚则保持隐性。*图 668*（右侧）显示了内部环回模式下 FDCAN_TX 和 FDCAN_RX 引脚与 FDCAN 的连接。

图 668. 环回模式下的引脚控制



应用看门狗（仅限 FDCAN1）

通过读取寄存器 FDCAN_TTOST 来处理应用看门狗。如果未及时处理应用看门狗，位 FDCAN_TTOST.AWE 会置 1，所有 TTCAN 通信都会停止，FDCAN1 会设为进入总线监控模式。

将应用看门狗限值 FDCAN_TTOCF.AWL 编程为 0x00 可禁止 TT 应用看门狗。不得在 TTCAN 应用程序中禁止 TT 应用看门狗。

时间戳生成

FDCAN 提供一个 16 位回卷计数器来生成时间戳。预分频器 FDCAN_TSCC.TCP 可配置为以 CAN 位时间的倍数 (1…16) 为计数器提供时钟。计数器值可通过 FDCAN_TSCV.TCV 读取。对寄存器 FDCAN_TSCV.TCV 进行写访问会将计数器复位为 0。时间戳计数器回卷时，中断标志 FDCAN_IR.TSW 会置 1。

开始接收/发送帧时，会捕获计数器值，并将该值存储在接收缓冲区/接收 FIFO (RXTS[15:0]) 或发送事件 FIFO (TXTS[15:0]) 元素的时间戳部分。

通过编程 FDCAN_TSCC.TSS 位，可以使用 16 位时间戳。

超时计数器

FDCAN 提供 16 位超时计数器来指示接收 FIFO 0、接收 FIFO 1 和发送事件 FIFO 的超时条件。它作为递减计数器工作，并且与时间戳计数器一样，使用由 FDCAN_TSCC.TCP 控制的预分频器。超时计数器通过寄存器 FDCAN_TOCC 配置。实际计数器值可从 FDCAN_TOCV.TOC 读取。仅当 FDCAN_CCCR.INIT = 0 时，才会启动超时计数器。当 FDCAN_CCCR.INIT = 1 时，超时计数器停止计数，例如 FDCAN 进入 Bus_Off 状态。

工作模式是通过 FDCAN_TOCC.TOS 选择的。在连续模式下工作时，计数器会在 FDCAN_CCCR.INIT 复位时开始计数。对 FDCAN_TOCV 进行写操作会将计数器预设为由 FDCAN_TOCC.TOP 配置的值并继续递减计数。

超时计数器由其中一个 FIFO 控制时，空 FIFO 会将计数器预设为由 FDCAN_TOCC.TOP 配置的值。存储了第一个 FIFO 元素后，会开始递减计数。写入 FDCAN_TOCV 无效。

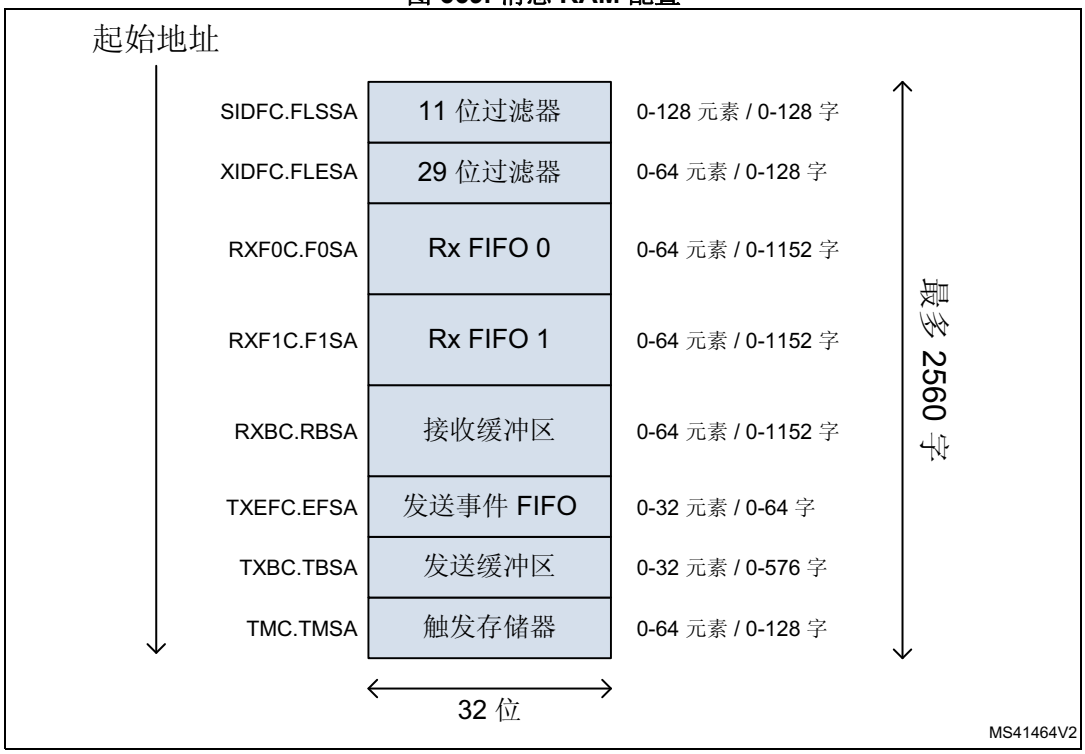
计数器达到 0 时，中断标志 FDCAN_IR.TOO 置 1。在连续模式下，计数器会在达到 FDCAN_TOCC.TOP 后立即重启。

注： *超时计数器的时钟信号来自 CAN 内核采样点信号。因此，由于 CAN 内核同步/重新同步机制的原因，超时计数器递减的时间点可能有所不同。如果使用 FDCAN 中的波特率切换功能，则超时计数器在仲裁字段和数据字段中使用不同的时钟。*

53.4.2 消息 RAM

消息 RAM 的宽度为 32 位。FDCAN 模块可配置为在消息 RAM 中分配多达 2560 个字。无需配置列在*图 669* 中的每个部分，各部分的顺序也无限制。

图 669. 消息 RAM 配置



当 FDCAN 对消息 RAM 进行寻址时，会寻址 32 位字，而不是单独字节。配置的起始地址为 32 位字地址，也就是说，仅会评估位 15 到位 2，两个最低有效位会被忽略。

注： *FDCAN 不会检查消息 RAM 的配置是否存在错误。因此在配置不同部分的起始地址以及各部分的元素数量时必须多加留意，以免造成数据篡改或丢失。*

接收处理

接收处理单元控制着验收过滤、将收到的信息传输到接收缓冲区或其中一个接收 FIFO（共两个）以及接收 FIFO 放入和取出的索引。

验收过滤器

FDCAN 能够配置两组验收过滤器，一组供标准标识符使用，另一组供扩展标识符使用。这两组过滤器可分配给接收缓冲区、接收 FIFO 0 或接收 FIFO 1。进行验收过滤时，会从第 0 个元素开始执行每个过滤器列表，直至过滤出第一个匹配元素。过滤出第一个匹配元素后，验收过滤会停止。不会为该消息评估以下过滤器元素。

主要特性有：

- 每个过滤器元素可配置为
  - 范围过滤器（对于 11 位过滤器为 0 到 128 个元素，对于 29 位过滤器为 0 到 64 个元素）
  - 用于一个或两个专用 ID 的过滤器
  - 典型位屏蔽过滤器
- 每个过滤器元素均可配置为实现验收过滤或拒绝过滤
- 每个过滤器元素均可单独使能/禁止
- 会按顺序检查过滤器，过滤出第一个匹配的过滤器元素后会停止执行

相关配置寄存器位：

- 全局过滤器配置 (FDCAN_GFC)
- 标准 ID 过滤器配置 (FDCAN_SIDFC)
- 扩展 ID 过滤器配置 (FDCAN_XIDF)
- 扩展 ID 和掩码 (FDCAN_XIDAM)

根据过滤器元素的配置 (SFEC/EFEC)，发生匹配时会触发下列操作之一：

- 将接收到的帧存储在 FIFO 0 或 FIFO 1 中
- 将接收到的帧存储在接收缓冲区中
- 将接收到的帧存储在接收缓冲区中，并在过滤器事件引脚上生成脉冲
- 拒绝接收到的帧
- 将高优先级消息中断标志 FDCAN_IR.HPM 置 1
- 将高优先级消息中断标志 FDCAN_IR.HPM 置 1，并将接收到的帧存储在 FIFO 0 或 FIFO 1 中
- 将高优先级消息中断标志 FDCAN_IR.HPM 置 1，并将接收到的帧存储在 FIFO 0 或 FIFO 1 中

接收到完整的标识符后，开始验收过滤。验收过滤完成后，如果已找到匹配的接收缓冲区或接收 FIFO，消息处理单元便会开始将接收到的消息数据以 32 位的形式写入匹配的接收缓冲区或接收 FIFO 中。如果 CAN 协议控制器已检测到错误条件（例如 CRC 错误），则会丢弃此消息，丢弃后的影响如下：

- **接收缓冲区**
  匹配接收缓冲区的新数据标志不会置 1，但接收缓冲区（部分）会被接收到的数据覆盖。有关错误类型，请参见 FDCAN_PSR.LEC 和 FDCAN_PSR.DLEC。
- **接收 FIFO**
  匹配接收 FIFO 的放入索引不会更新，但相关接收 FIFO 元素（部分）会被接收到的数据覆盖。有关错误类型，请参见 FDCAN_PSR.LEC 和 FDCAN_PSR.DLEC。如果匹配的接收 FIFO 在覆盖模式下工作，必须考虑*接收 FIFO 覆盖模式*中介绍的边界条件。

注： *当接收的消息写入两个接收 FIFO 之一、或写入接收缓冲区中时，未修改的已接收标识符会独立于所用过滤器进行存储。验收过滤过程的结果很大程度上取决于所配置的过滤器元素的顺序。*

范围过滤器

该过滤器会将所有接收到的帧与由 SF1ID/SF2ID 和 EF1ID/EF2ID 定义的范围内的消息 ID 进行匹配检查。

范围过滤与扩展帧一起使用时，有两种可能的情况：

- EFT = 00：应用范围过滤器之前，会将已接收帧的消息 ID 与扩展 ID 与掩码 (FDCAN_XIDAM) 进行与运算
- EFT = 11：不会为范围过滤使用扩展 ID 与掩码 (FDCAN_XIDAM)

专用 ID 过滤器

可将过滤器元素配置为过滤一个或两个特定的消息 ID。要过滤一个特定的消息 ID，过滤器元素必须配置为 SF1ID = SF2ID 和 EF1ID = EF2ID。

典型位屏蔽过滤器

典型位屏蔽过滤用于通过屏蔽已接收消息 ID 的单个位来过滤消息 ID 组。进行典型位屏蔽过滤时，SF1ID/EF1ID 用作消息 ID 过滤器，SF2ID/EF2ID 则用作过滤器屏蔽位。

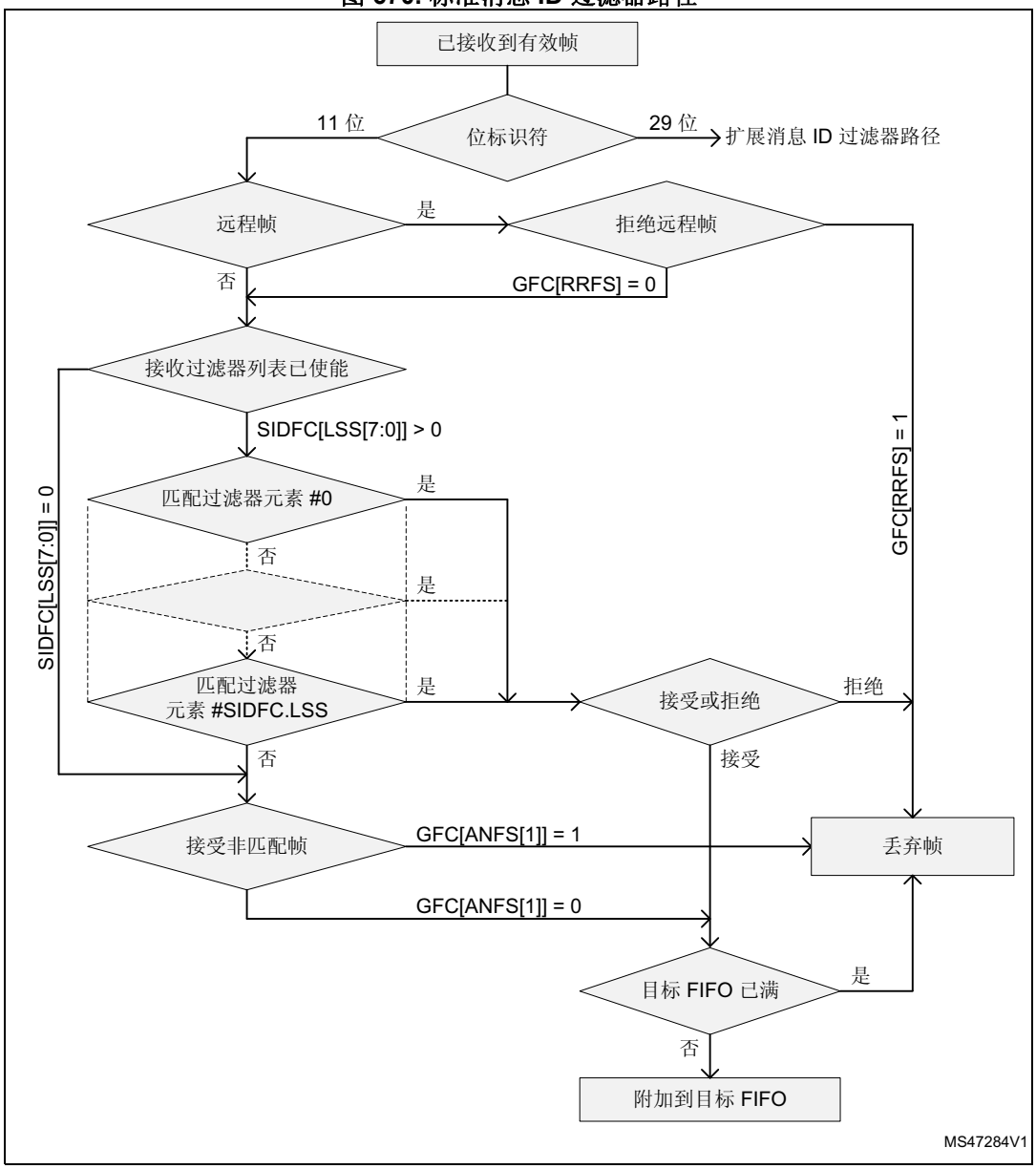
过滤器屏蔽的 0 位将屏蔽掉已配置 ID 过滤器的相应位位置，例如，已接收消息 ID 在该位位置的值与验收过滤无关。只有相应屏蔽位为 1 的已接收消息 ID 的位才与验收过滤有关。

如果所有屏蔽位均为 1，则仅当接收到的消息 ID 和消息 ID 过滤器完全相同时，才会出现匹配。如果所有屏蔽位均为 0，则所有消息 ID 都匹配。

标准消息 ID 过滤

图 670 介绍了标准消息 ID（11 位标识符）的过滤流程。*第 53.4.21 节*中介绍了标准消息 ID 过滤器元素。

图 670. 标准消息 ID 过滤器路径

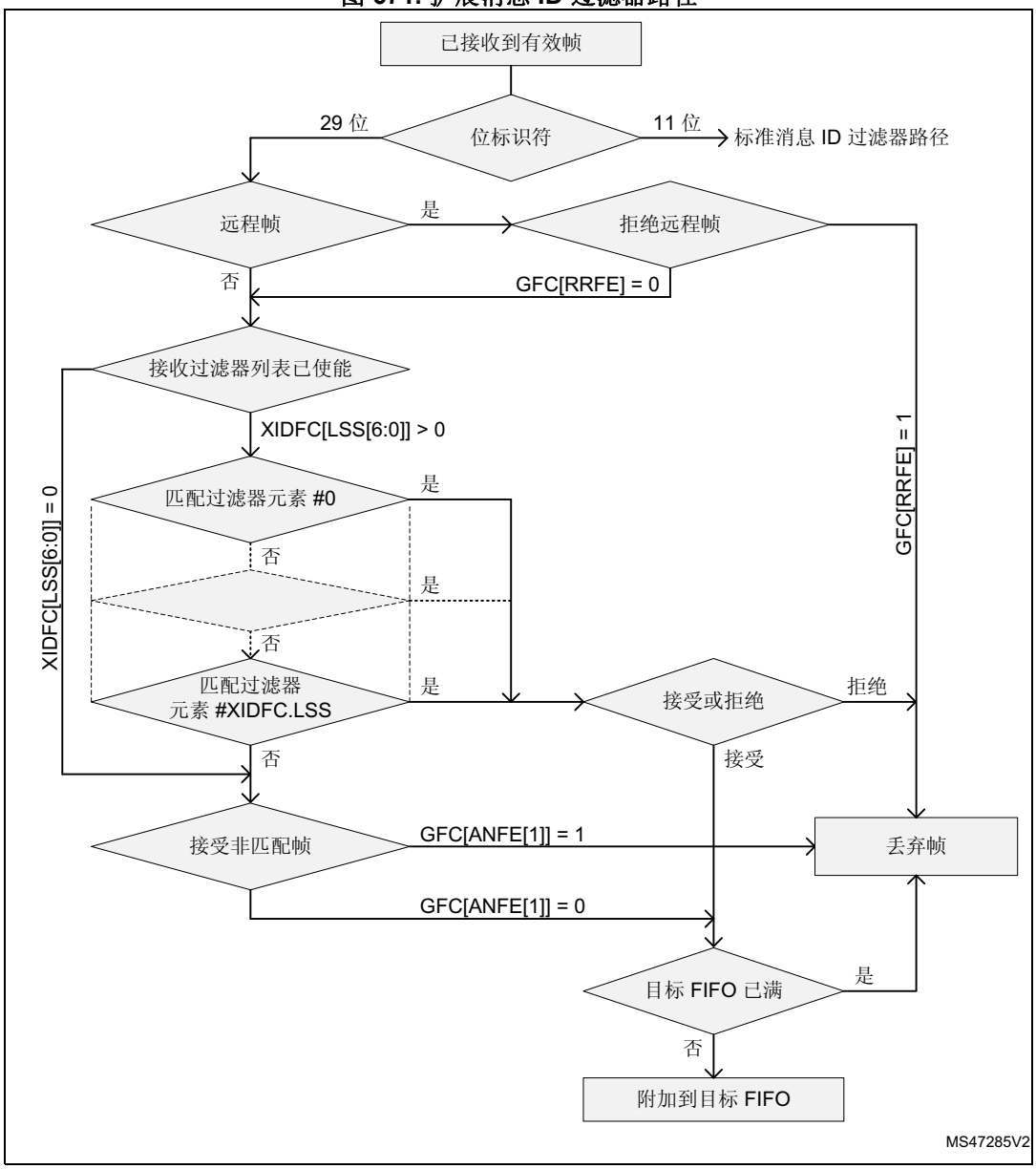


已接收帧的远程发送请求位 (RTR) 和标识符扩展位 (IDE) 会在全局过滤器配置 (FDCAN_GFC) 和标准 ID 过滤器配置 (FDCAN_SIDFC) 消息 ID 的控制下，与已配置过滤器元素列表进行比较。

扩展消息 ID 过滤

图 671 介绍了扩展消息 ID（29 位标识符）的过滤流程。*第 53.4.22 节*中介绍了扩展消息 ID 过滤器元素。

图 671. 扩展消息 ID 过滤器路径



已接收帧的远程发送请求位 (RTR) 和标识符扩展位 (IDE) 会在全局过滤器配置 FDCAN_GFC 和扩展 ID 过滤器配置 FDCAN_XIDFC 消息 ID 的控制下，与已配置过滤器元素列表进行比较。

扩展 ID 与掩码 (FDCAN_XIDAM) 会在执行过滤器列表之前与接收到的标识符进行与运算。

接收 FIFO

接收 FIFO 0 和接收 FIFO 1 最多可分别保存 64 个元素。两个接收 FIFO 的配置是通过寄存器 FDCAN_RXF0C 和 FDCAN_RXF1C 完成的。

通过验收过滤的已接收消息会传输到由匹配过滤器元素配置的接收 FIFO 中。有关可用于接收 FIFO 0 和接收 FIFO 1 的过滤器机制说明，请参见*验收过滤器*。*第 53.4.18 节*中介绍了接收缓冲区和 FIFO 元素。

当 FDCAN_IR.RFnF 指示接收 FIFO 已满时，需要至少读出一条信息，并且接收 FIFO 的取出索引增加，才能继续写入消息到接收 FIFO。如果在相应的接收 FIFO 已满时收到消息，此消息会被丢弃，中断标志 FDCAN_IR.RFnL 会置 1。

为了避免接收 FIFO 溢出，可使用接收 FIFO 阈值。当接收 FIFO 填充级别达到由 FDCAN_RXFnC.FnWM 配置的接收 FIFO 阈值时，中断标志 FDCAN_IR.RFnW 会置 1。

从接收 FIFO 中读取消息时，需要将接收 FIFO 的取出索引 RXFnS[FnGI] 加上 FIFO 元素的大小，然后加到对应的接收 FIFO 的起始地址 RXFnC[FnSA] 上。

接收 FIFO 阻止模式

接收 FIFO 阻止模式是通过 RXFnC.FnOM = 0 配置的。该模式是接收 FIFO 的默认工作模式。

达到接收 FIFO 已满条件时 (RXFnS.FnPI = RXFnS.FnGI)，需要至少读出一条信息，并且接收 FIFO 的取出索引增加，才能继续写入消息到接收 FIFO。接收 FIFO 已满条件是通过 RXFnS.FnF = 1 指示的。此外，中断标志 FDCAN_IR.RFnF 也会置 1。

如果在相应的接收 FIFO 已满时收到消息，此消息会被丢弃，并会通过 RXFnS.RFnL = 1 指示消息丢失条件。此外，中断标志 FDCAN_IR.RFnL 也会置 1。

接收 FIFO 覆盖模式

接收 FIFO 覆盖模式是通过 RXFnC.FnOM = 1 配置的。

当 RXFnS.FnF = 1 指示接收 FIFO 已满条件时 (RXFnS.FnPI = RXFnS.FnGI)，FIFO 接收的下一条消息覆盖最原始的 FIFO 消息。放入索引和取出索引都会加 1。

如果接收 FIFO 在覆盖模式下工作，并且指示了接收 FIFO 已满条件，则应至少从获取索引 + 1 处开始读取接收 FIFO 元素。这是因为 CPU 从消息 RAM 读取消息（取出索引）时可能会将接收到的消息写入消息 RAM（放入索引）。在这种情况下，从相应的接收 FIFO 元素中读取的数据可能不一致。从接收 FIFO 读取数据时向获取索引中添加偏移可避免此问题。偏移取决于 CPU 访问 Rx FIFO 的速度。*图 673* 显示了读取接收 FIFO 时与取出索引的偏移为 2。在这种情况下，存储在元素 1 和元素 2 中的两条消息会丢失。

从接收 FIFO 读取数据后，读取的最后一个元素的编号必须写入接收 FIFO 确认索引 RXFnA.FnA 中。这样会使获取索引增加到该元素编号。如果放入索引未增加到该接收 FIFO 元素，接收 FIFO 已满条件会复位（RXFnS.FnF = 0）。

专用接收缓冲区

FDCAN 支持多达 64 个专用接收缓冲区。专用接收缓冲区部分的起始地址是通过 FDCAN_RXBC.RBSA 配置的。

对于每个接收缓冲区，必须配置 SFEC/EFEC = 111 和 SFID2/EFID2[10:9] = 00 的标准或扩展消息 ID 过滤器元素（请参见*第 53.4.21 节*和*第 53.4.22 节*）。

接收到的消息被过滤器元素接受后，会存储在由过滤器元素引用的消息 RAM 中的接收缓冲区中。存储格式与接收 FIFO 元素的格式相同。此外，中断寄存器中的标志 FDCAN_IR.DRX（专用接收缓冲区中存储的消息）也会置 1。

表 479. 接收缓冲区过滤器配置示例

过滤器元素	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID 消息 1	00	00 0000
1	ID 消息 2	00	00 0001
2	ID 消息 3	00	00 0010

匹配接收消息的最后一个字写入消息 RAM 中后，寄存器 NDAT1,2 中相应的新数据标志会置 1。只要新数据标志保持置 1 状态，相应的接收缓冲区就会锁定，以免被接收到的匹配帧更新。新数据标志必须由用户向相应的位位置写入“1”进行复位。

接收缓冲区新数据标志置 1 时，引用该特定接收缓冲区的消息 ID 过滤器元素将不匹配，因此会继续进行验收过滤。以下消息 ID 过滤器元素可能导致接收到的消息存储到另一接收缓冲区或接收 FIFO 中，或者导致消息被拒绝，具体取决于过滤器配置。

接收缓冲区处理

- 复位中断标志 FDCAN_IR.DRX
- 读取新数据寄存器
- 从消息 RAM 中读取消息
- 复位已处理消息的新数据标志

过滤调试消息

要过滤调试消息，需为三个调试消息分别配置一个标准/扩展消息 ID 过滤器元素。要使能过滤器元素过滤调试消息，必须将 SFEC/EFEC 编程为 111。在这种情况下，SFID1/SFID2 和 EFID1/EFID2 的含义不同。SFID2/EFID2[10:9] 控制着调试消息处理状态机，而 SFID2/EFID2[5:0] 控制着已接收调试消息的存储位置。

存储调试消息时，相应的新数据标志和 FDCAN_IR.DRX 均不会置 1。可通过 FDCAN_RXF1S.DMS 监视调试消息的接收情况。

表 480. 调试消息过滤器配置示例

过滤器元素	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID 调试消息 A	01	11 1101
1	ID 调试消息 B	10	11 1110
2	ID 调试消息 C	11	11 1111

发送处理

发送处理单元处理专用发送缓冲区、发送 FIFO 和发送队列的发送请求。它控制着发送消息到 CAN 内核的传输、放入和取出索引以及发送事件 FIFO，最多可为消息发送设置 32 个发送缓冲区（请参见 *专用发送缓冲区*）。根据元素大小 (FDCAN_RXESC) 的配置，使用两个到十六个 32 位字 (Rn = 3 ...17) 存储 CAN 消息数据字段。

表 481. 帧发送的可能配置

FDCAN_CCCR		发送缓冲区元素		帧发送
BRSE	FDOE	FDF	BRS	
忽略	0	忽略	忽略	经典 CAN
0	1	0	忽略	经典 CAN
0	1	1	忽略	不进行比特率切换的 FD
1	1	0	忽略	经典 CAN
1	1	1	0	不进行比特率切换的 FD
1	1	1	1	进行比特率切换的 FD

注： *AUTOSAR 至少需要三个发送队列缓冲区，并支持发送取消。*

当发送缓冲区请求挂起寄存器 FDCAN_TXBRP 更新时，或者发送已启动时，发送处理单元会启动发送扫描来检查优先级最高的挂起发送请求（消息 ID 最小的发送缓冲区）。

发送暂停

该功能用于 CAN 消息标识符（永久性地）被指定为特定值、并且无法轻易更改的 CAN 系统中。这些消息标识符的 CAN 仲裁优先级可能高于其他已定义的消息，而在特定的应用中，其相对仲裁优先级应反转。这样可能会导致如下情况：一个 ECU 发送的 CAN 消息突发造成 CAN 仲裁优先级较低的另一个 ECU CAN 消息被延迟。

举例来说，如果 CAN ECU-1 使能了此功能，并且其应用软件请求发送四条消息，成功发送第一条消息后，将等待总线空闲的两个 CAN 位时间，然后才允许开始发送下一条请求的消息。如果其他 ECU 有挂起的消息，则会在空闲时间开始发送这些消息，不需要与 ECU-1 的下一条消息进行仲裁。接收到消息后，ECU-1 可在接收到的消息释放 CAN 总线后立即开始进行下一次发送。

此功能是通过 FDCAN_CCCR 寄存器中的 TXP 位控制的。如果该位置 1，则每次成功发送消息后，FDCAN 都暂停两个 CAN 位时间，然后再开始进行下一次发送。这样一来，即使网络中其他节点的优先级标识符较低，也能发送消息。默认禁止此功能 (FDCAN_CCCR.TXP = 0)。

此功能可使来自单个节点的突发发送变得松散一些，避免出现应用程序错误地请求过多传输的“胡乱指示”情况。

专用发送缓冲区

专用发送缓冲区用于在 CPU 完全控制下进行消息发送。每个专用发送缓冲区都配置了一个特定消息 ID。如果多个发送缓冲区配置为使用同一消息 ID，则会先发送缓冲区编号最小的发送缓冲区。

如果数据部分已更新，则会通过添加 FDCAN_TXBAR.ARn 位请求发送。请求的消息在内部会与发送 FIFO 或发送队列中的消息进行仲裁，在外部会与 CAN 总线上的消息进行仲裁，并会根据其消息 ID 发送出去。

专用的发送缓冲区会在消息 RAM 中分配四个 32 位字。因此，消息 RAM 中专用发送缓冲区起始地址的计算方法是：将发送缓冲区索引 (0 … 31) 乘以四，再与发送缓冲区起始地址 FDCAN_TXBC.TBSA 相加。

表 482. 发送缓冲区/FIFO - 队列元素大小

FDCAN_TXESC.TBDS[2:0]	数据字段（字节）	元素大小（RAM 字）
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

发送 FIFO

发送 FIFO 操作是通过将 FDCAN_TXBC.TFQM 编程为 0 配置的。发送 FIFO 中存储的消息是先从取出索引 FDCAN_TXFQS.TFGI 引用的消息开始发送的。每次发送后，取出索引会循环递增，直至发送 FIFO 为空。发送 FIFO 可按消息写入发送 FIFO 的顺序发送来自不同发送缓冲区但消息 ID 相同的消息。FDCAN 会计算取出索引和放入索引之差作为发送 FIFO 空闲级别 FDCAN_TXFQS.TFFL，用于指示可用（空闲）的发送 FIFO 元素数。

新的消息必须写入以放入索引 FDCAN_TXFQS.TFQPI 引用的发送缓冲区开始的发送 FIFO 中。添加请求会将放入索引增加到下一空闲发送 FIFO 元素。放入索引达到取出索引后，会指示发送 FIFO 已满 (FDCAN_TXFQS.TFQF = 1)。在这种情况下，下一条消息已发送且取出索引已递增之前，不应继续向发送 FIFO 写入消息。

如果有一条消息添加到发送 FIFO，则会通过向与发送 FIFO 放入索引引用的发送缓冲区相关的 FDCAN_TXBAR 位写入 1 来请求发送消息。

如果有多条 (n) 消息添加到发送 FIFO，则会写入以放入索引开始的 n 个连续发送缓冲区中。随后会通过 FDCAN_TXBAR 请求发送。放入索引随后会循环递增 n。请求的发送缓冲区数不应超过发送 FIFO 空闲级别指示的空闲发送缓冲区数。

如果取出索引引用的发送缓冲区的发送请求取消，取出索引会增加到下一个具有挂起发送请求的发送缓冲区，并会重新计算发送 FIFO 空闲级别。如果取消对其他任何发送缓冲区的发送，取出索引和 FIFO 空闲级别保持不变。

发送 FIFO 元素会在消息 RAM 中分配四个 32 位字。因此，下一可用（空闲）发送 FIFO 缓冲区的起始地址是通过将放入索引 FDCAN_TXFQS.TFQPI (0 … 31) 乘以四再与发送缓冲区起始地址 FDCAN_TXBC.TBSA 相加计算得出的。

发送队列

发送队列操作是通过将 FDCAN_TXBC.TFQM 编程为 1 配置的。发送队列中存储的消息是先从消息 ID 最小（优先级最高）的消息开始发送的。如果多个队列缓冲区配置为使用同一消息 ID，则会先发送缓冲区编号最小的队列缓冲区。

新消息必须写入放入索引 FDCAN_TXFQS.TFQPI 引用的发送缓冲区中。添加请求会将放入索引循环增加到下一空闲发送缓冲区。如果发送队列已满 (FDCAN_TXFQS.TFQF = 1)，则放入索引无效，并且至少在有一个请求的消息已发出或挂起的发送请求已取消之前，不应继续向发送队列写入消息。

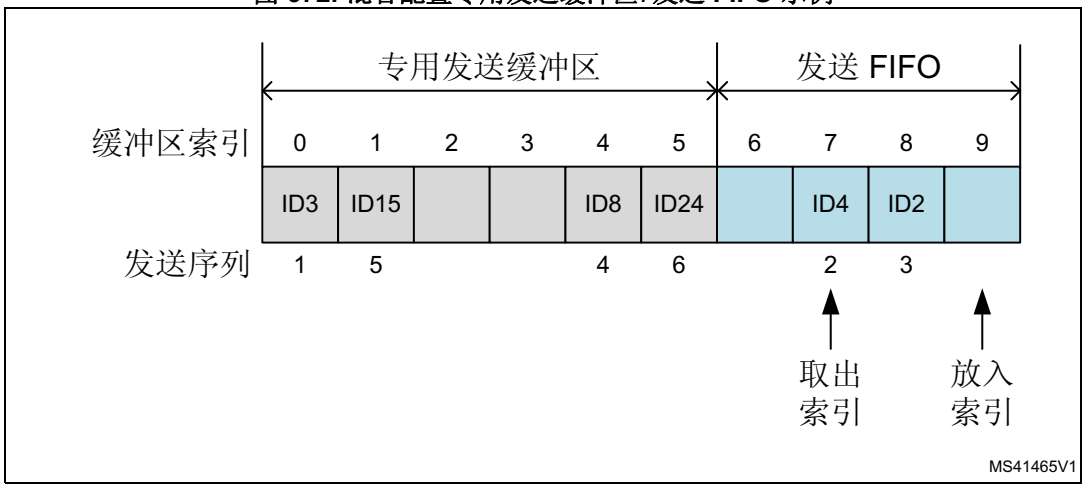
应用可使用寄存器 FDCAN_TXBRP 来代替放入索引，并可将消息放入任何没有挂起传输请求的发送缓冲区中。

发送队列缓冲区会在消息 RAM 中分配四个 32 位字。因此，下一可用（空闲）发送队列缓冲区的起始地址是通过将发送队列放入索引 FDCAN_TXFQS.TFQPI (0 ... 31) 乘以四再与发送缓冲区起始地址 FDCAN_TXBC.TBSA 相加计算得出的。

混合专用发送缓冲区 / 发送 FIFO

在这种情况下，消息 RAM 中的发送缓冲区会被划分为一组专用发送缓冲区和一个发送 FIFO。专用发送缓冲区的数量是通过 FDCAN_TXBC.NDTB 配置的。分配给发送 FIFO 的发送缓冲区数量是通过 FDCAN_TXBC.TFQS 配置的。如果 FDCAN_TXBC.TFQS 编程为 0，则仅会使用专用发送缓冲区。

图 672. 混合配置专用发送缓冲区 / 发送 FIFO 示例



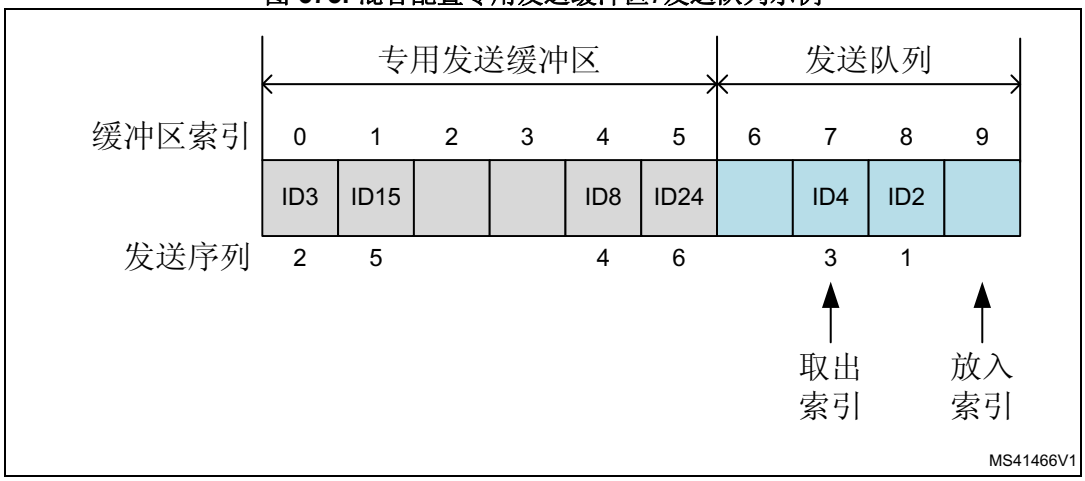
发送优先次序：

- 扫描专用发送缓冲区和挂起时间最久的发送 FIFO 缓冲区（FDCAN_TXFS.TFGI 引用的缓冲区）
- 消息 ID 最小的缓冲区优先级最高，下次将发送该缓冲区的数据

混合专用发送缓冲区 / 发送队列

在这种情况下，消息 RAM 中的发送缓冲区会被划分为一组专用发送缓冲区和一个发送队列。专用发送缓冲区的数量是通过 FDCAN_TXBC.NDTB 配置的。发送队列缓冲区的数量是通过 FDCAN_TXBC.TFQS 配置的。如果 FDCAN_TXBC.TFQS 编程为 0，则仅会使用专用发送缓冲区。

图 673. 混合配置专用发送缓冲区/发送队列示例



发送优先级设置：

- 扫描所有激活了发送请求的发送缓冲区
- 消息 ID 最小的发送缓冲区优先级最高，下次将发送该缓冲区的数据

发送取消

FDCAN 支持发送取消功能。要取消专用发送缓冲区或发送队列缓冲区请求的发送，用户必须向寄存器 FDCAN_TXBCR 中相应的位位置（= 发送缓冲区的数量）写入 1。发送取消不适用于发送 FIFO 操作。

成功取消后，寄存器 FDCAN_TXBCF 的相应位会设为 1。

如果已经在从发送缓冲区进行发送时请求发送取消，则只要发送正在进行中，相应的 FDCAN_TXBRP 位就会保持置 1 状态。如果发送成功，相应的 FDCAN_TXBTO 和 FDCAN_TXBCF 位会置 1。如果发送不成功，则不会重复发送，只会将相应的 FDCAN_TXBCF 位置 1。

注： *如果恰好在挂起的发送开始之前取消此次发送，即使该节点中有另一消息挂起，取消后也会有一个短暂的时间窗，在此期间不会开始任何发送。这样一来，另一节点便可以发送优先级可能比该节点中第二条消息低的消息。*

发送事件处理

为了支持发送事件处理，FDCAN 实现了发送事件 FIFO。FDCAN 在 CAN 总线上发送消息后，消息 ID 和时间戳会存储在发送事件 FIFO 元素中。为了将发送事件关联到发送事件 FIFO 元素，已发送的发送缓冲区中的消息标志会被复制到发送事件 FIFO 元素中。

发送事件 FIFO 最多可配置为 32 个元素。*发送 FIFO* 中介绍了发送事件 FIFO 元素。根据元素大小 (FDCAN_TXESC) 的配置，会使用 2 到 16 个 32 位字 (Tn = 3 ..17) 来存储 CAN 消息数据字段。

发送事件 FIFO 的用途是将处理发送状态信息与处理发送消息分开，也就是让发送缓冲区仅保存要发送的消息，而将发送状态单独存储在发送事件 FIFO 中。这样做有很大的优势，尤其是在处理动态管理的发送队列时，发送缓冲区可在发送成功后立即用于新消息。覆盖发送缓冲区之前，不需要保存该发送缓冲区的发送状态信息。

当 FDCAN_IR.TEFF 指示发送事件 FIFO 已满条件时，需要至少读取一个元素，并且发送事件 FIFO 的取出索引增加，才能继续写入元素到发送事件 FIFO。如果在发送事件 FIFO 已满时发生发送事件，此事件会被丢弃，中断标志 FDCAN_IR.TEFL 会置 1。

为了避免发送事件 FIFO 溢出，可使用发送事件 FIFO 阈值。当发送事件 FIFO 填充级别达到由 FDCAN_TXEFC.EFWM 配置的发送事件 FIFO 阈值时，中断标志 FDCAN_IR.TEFW 会置 1。

从发送事件 FIFO 读取数据时，必须将发送事件 FIFO 获取索引 FDCAN_TXEFS.EFGI 乘以 2 再与发送事件 FIFO 起始地址 FDCAN_TXEFC.EFSA 相加。

53.4.3 FIFO 确认处理

接收 FIFO 0、接收 FIFO 1 和发送事件 FIFO 是通过向相应的 FIFO 确认索引进行写操作控制的，请参见 *FDCAN 接收 FIFO 0 确认寄存器 (FDCAN_RXF0A)*、*FDCAN 接收 FIFO 1 确认寄存器 (FDCAN_RXF1A)* 和 *FDCAN 发送事件 FIFO 配置寄存器 (FDCAN_TXEFC)*。对 FIFO 确认索引进行写操作会将 FIFO 获取索引设置为 FIFO 确认索引加 1，进而会更新 FIFO 填充级别。有两种用例：

- 如果只从 FIFO 读取了一个元素（获取索引指向的元素），则该取出索引值会写入 FIFO 确认索引中。
- 如果已从 FIFO 中读取一系列元素，则仅在该读取序列结束时写入一次 FIFO 确认索引（数值：读取的最后一个元素的索引）即可更新 FIFO 获取索引。

由于 CPU 可自由访问 FDCAN 消息 RAM，以任意顺序（不考虑取出索引）读取 FIFO 元素时需要特别留意。这一点在从两个接收 FIFO 之一读取高优先级消息时会很有用。在这种情况下，不应写入 FIFO 确认索引，否则会将取出索引设为错误的位置，还会改变 FIFO 填充级别。在这种情况下，一些时间较久的 FIFO 元素会丢失。

注： *应用必须确保 FIFO 确认索引中已写入有效值。FDCAN 不会检查数值是否正确。*

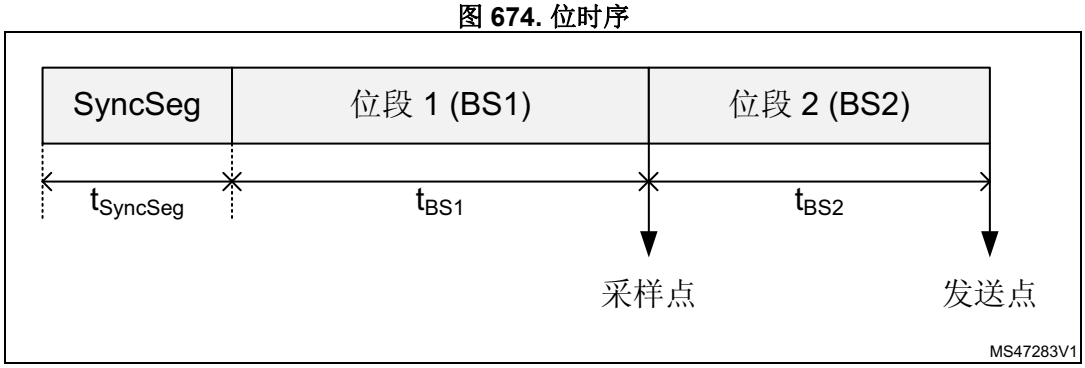
53.4.4 位时序

位时序逻辑将监视串行总线，执行采样并调整采样点，在调整采样点时，需要在起始位边沿进行同步，并在后续的边沿进行再同步。

如*图 674* 中所示，通过将位时间划分为以下三段，即可解释其工作过程：

- 同步段（SYNC_SEG）：位变化应该在此时间段内发生，具体为一个时间片的固定长度 (1 x tq)。
- 位段 1 (BS1)：定义采样点的位置。它包括 CAN 标准的 PROP_SEG 和 PHASE_SEG1，其持续长度可以在 1 到 16 个时间片之间调整，但也可以自动加长，以补偿不同网络节点的频率差异所导致的正相位漂移。
- 位段 2 (BS2)：定义发送点的位置。它代表 CAN 标准的 PHASE_SEG2，其持续长度可以在 1 到 8 个时间片之间调整，但也可以自动缩短，以补偿负相位漂移。

图 674. 位时序



波特率是位时间的倒数（波特率 = 1 / 位时间），而位时间又是三个分量的总和。如*图 674*中所示，位时间 = $t_{SyncSeg}$ + t_{BS1} + t_{BS2}，其中：

- 对于标称位时间
  - tq = (FDCAN_NBTP.NBRP[8:0] + 1) * $t_{fdcan_tq_ck}$
  - $t_{SyncSeg}$ = 1 tq
  - t_{BS1} = tq * (FDCAN_NBTP.NTSEG1[7:0] + 1)
  - t_{BS2} = tq * (FDCAN_NBTP.NTSEG2[6:0] + 1)
- 对于数据位时间
  - tq = (FDCAN_DBTP.DBRP[4:0] + 1) * $t_{fdcan_tq_ck}$
  - $t_{SyncSeg}$ = 1 tq
  - t_{BS1} = tq * (FDCAN_DBTP.DTSEG1[4:0] + 1)
  - t_{BS2} = tq * (FDCAN_DBTP.DTSEG2[3:0] + 1)

（再）同步跳转宽度 (SJW) 定义位段加长或缩短的上限。它可以在 1 到 4 个时间片之间调整。

有效边沿是指一个位时间内总线电平从显性到隐性的第一次转换（前提是控制器本身不发送隐性位）。

如果在 BS1 而不是 SYNC_SEG 中检测到有效边沿，则 BS1 会延长最多 SJW，以便延迟采样点。

相反地，如果在 BS2 而不是 SYNC_SEG 中检测到有效边沿，则 BS2 会缩短最多 SJW，以便提前发送点。

为了避免编程错误，位时序寄存器只能在设备处于待机模式时进行配置。寄存器 FDCAN_DBTP 和 FDCAN_NBTP（分别专用于数据位时序和标称位时序）只能在 FDCAN_CCCR.CCE 和 FDCAN_CCCR.INIT 置 1 时进行访问。

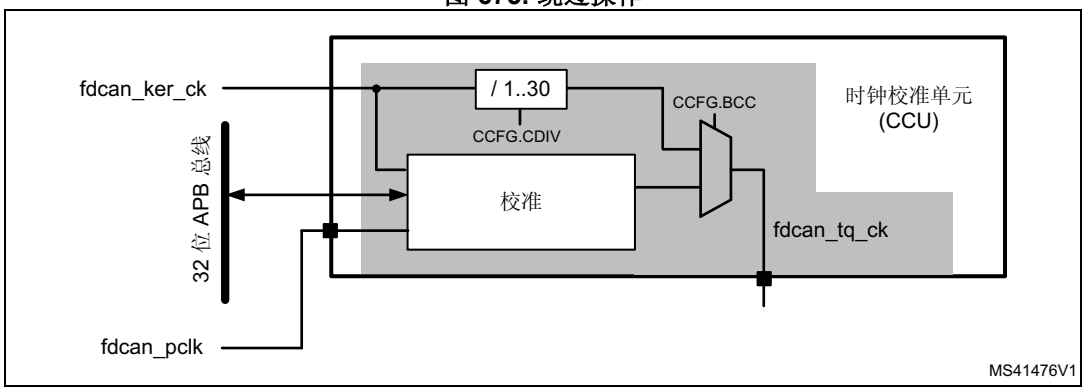
注： *有关 CAN 位时序和再同步机制的详细说明，请参见 ISO 11898-1 标准。*

53.4.5 CAN 时钟校准

设备复位后，时钟校准单元 (CCU) 不会向 FDCAN 提供有效的时钟信号。必须通过 FDCAN_CCFG 寄存器对 CCU 进行初始化。仅当 FDCAN1 的 FDCAN_CCCR.CCE 和 FDCAN_CCCR.INIT 位均置 1 时，才能向 FDCAN_CCFG 寄存器写入数据。因此，需要先完成 CCU 和 FDCAN1 初始化，所有 FDCAN 模块才能够工作。

当 FDCAN_CCFG.BCC = 1 时，会绕过时钟校准（请参见*图 675*）。

图 675. 绕过操作



工作条件

CAN 时钟校准单元的工作条件如下：

- CAN 内核时钟频率 fdcan_ker_ck 至少为 80 MHz
- FDCAN 比特率：
  - 标称比特率：高达 1 Mb/s
  - 数据比特率：介于标称比特率与 8 Mb/s 之间

FDCAN 时钟校准单元会生成介于 0.5 Mhz 和 25 MHz 范围内的经校准时间片时钟 fdcan_tq_ck。

注： *FDCAN 要求 CAN 时间片时钟始终小于或等于 APB 时钟 (fdcan_tq_ck < fdcan_pclk)。绕过 CAN 时钟校准单元时 (FDCAN_CCFG.BCC = 1)，必须考虑这一点。*

校准精度

Precision_Calibrated 状态下的校准精度取决于下列因素。

- CAN 内核时钟输入 fdcan_ker_ck 的动态时钟容差
- 测量误差。对于用于校准测量的每个位序列，存在最大一个 fdcan_pclk 周期的误差。位时间测量使用的位数为 32 位或 64 位，具体取决于 FDCAN_CCFG.CFL 的配置。
- 校准机制中可容忍的误差

选择校准消息之间的距离时，必须满足 FDCAN1 模块的时钟容差要求。

注： *动态时钟容差是因温度或工作电压变化等原因造成的两条校准消息之间的时钟频率变化。*

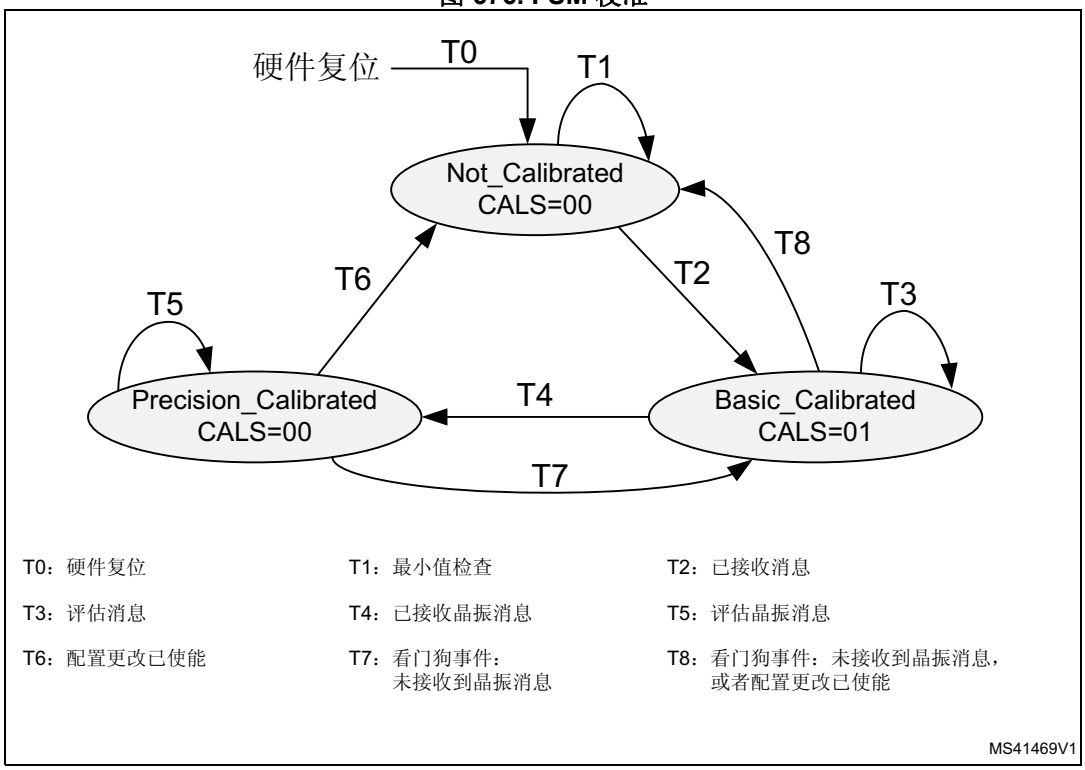
功能描述

通过 CAN 消息执行时间片时钟 fdcan_tq_ck 校准时，应调整用于通过时钟 fdcan_ker_ck 生成 CAN 协议时间片 tq 的时钟分频器。

1. 第一步：基本校准
   测量从隐性到显性的两个连续下降沿之间的最短距离，该时间将假定为两个 CAN 位时间，以 PLL 时钟周期计数。每次新测量发现两个边沿间的距离变短时，都会更新时钟分频器 (FDCAN_CCFG.CDIV)。当 CAN 协议控制器检测到有效 CAN 消息时，基本校准完成。

2. 第二步：精确校准
校准状态机通过计算 fdcan_ker_ck 周期数来测量 CAN 帧中较长的位序列长度。该位序列的长度可通过 FDCAN_CCFG.CLF 配置为 32 或 64 位。对于 32/64 位的校准字段长度，需要至少 2/6 字节数据字段的校准消息。精确校准基于通过测量更长的位序列计算出的新时钟分频器值。

图 676. FSM 校准



如果 FDCAN_CCU_IE.CSCE 中断使能，校准状态的变化会将中断标志 FDCAN_CCU_IR.CSC 置 1。该标志会保持置 1 状态，直到在 FDCAN_CCU_IR.CSC 中写入 1 来清零。

在精确校准完成之前，FDCAN 会在受限模式下工作（不发送帧、不发送错误或过载标志、不计算错误数量）。如果 fdcan_ker_clk 校准是由软件通过寄存器 FDCAN_CCU_CSTAT 评估校准状态来完成的，则 FDCAN 必须设为受限工作模式 (FDCAN_CCCR.ASM = 1)，直到 CAN 单元校准进入 Precision_Calibrated 状态（参见*应用*）。

只能对使用晶振控制的稳定时钟的节点所发送的有效 CAN 帧执行精确校准。校准帧是通过 FDCAN1 验收过滤检测的，必须在 FDCAN1 中配置过滤器元素和接收缓冲区，以识别和存储校准消息。接收到校准消息后，必须复位接收缓冲区的新数据标志，从而可指示下一条校准消息。

如果网络中只有一个使用晶振时钟的 CAN 发送器，则该节点必须在通过数据字段或标识符中至少一个 1010 序列启动后发送其第一条消息。这样可确保非晶振节点能够进入 Basic_Calibrated 状态，随后确认晶振节点消息。

必须在由校准看门狗监视的预定义最大时间间隔内重复执行精确校准。

注： *当 CAN 时钟校准单元从 Precision_Calibrated 状态切换回 Basic_Calibrated 状态时，会禁止校准正常信号，FDCAN1 会完成当前进行的发送过程，然后进入受限工作模式（不发送帧、不发送错误或过载标志、不计算错误数量）。*

配置

CAN 时钟校准单元是通过寄存器 FDCAN_CCFG 配置的，也就在 FDCAN1 将 FDCAN_CCCR.CCE 和 FDCAN_CCCR.INIT 位置 1 时。

对于基本校准，将会测量引脚 FDCAN1_RX 上两个连续下降沿之间的最小振荡器周期数。时钟周期数取决于对输入 fdcan_ker_ck 应用的时钟频率。如果测得的时钟周期数小于由 FDCAN_CCFG.OCPM 配置的最小值（例如由于 FDCAN1_RX 上存在噪声信号），该值会被丢弃，测量会继续进行。

建议将 FDCAN_CCFG.OCPM 配置为略小于两个 CAN 位时间：

FDCAN_CCFG.OCPM < ((2 个 CAN 位时间) / fdcan_ker_ck 周期) / 32

用于精确校准的位字段的长度可通过 FDCAN_CCFG.CFL 配置为 32 或 64 位。用于精确校准的位数对校准精度和两个校准消息之间的最大距离有影响。

通过 FDCAN_CCFG.TQBT 配置的每个位时间的时间片数与测得的振荡器时钟周期 FDCAN_CCU_CSTAT.OCPC 数共同定义每个位时间的振荡器时钟数。

如果通过配置 FDCAN_CCFG.BCC = 1 绕过时钟校准，则需要通过 FDCAN_CCFG.CDIV 配置内部时钟分频器，以满足 fdcan_tq_ck < fdcan_pclk 的条件。

注： *当 CAN 时钟校准有效时 (FDCAN_CCFG.BCC = 0)，必须将 FDCAN 模块的波特率预分频器配置为无效。*

状态指示

可以通过读取寄存器 FDCAN_CCU_CSTAT 来监视 CAN 单元时钟校准状态。当处于 Precision_Calibrated 状态时，振荡器时钟周期计数器 FDCAN_CCU_CSTAT.OCPC 会指示校准字段中信号振荡器时钟周期数，而 FDCAN_CCU_CSTAT.TQC 则指示校准字段中的时间片数。

校准状态由 FDCAN_CCU_CSTAT.CALS 监视。校准状态的变化会将中断标志 FDCAN_CCU_IR.CSC 置 1。如果 FDCAN_CCU_IE.CSCE 中断使能，该标志会保持置 1 状态，直到在 FDCAN_CCU_IR.CSC 中写入 1 来清零。

校准看门狗事件也会将中断标志 FDCAN_CCU_IR.CWE 置 1。如果由 FDCAN_CCU_IE.CWEE（设为高电平）使能，它将保持有效状态，直至由 FDCAN_CCU_IE.CWE 复位。

53.4.6 应用

绕过时钟校准

CCU 内部时钟分频器配置为一分频 (FDCAN_CCFG.CDIV = 0x0000)。在该工作模式下，输入时钟 fdcan_ker_ck 会直接发送到时钟输出 fdcan_tq_ck。在这种情况下，fdcan_tq_ck 与连接到 CCU 的 FDCAN1 和 FDCAN2 的配置和状态无关。FDCAN_ker_ck 高于 80 MHz 时，可以进行 CAN FD 操作。

软件校准

CAN 时钟校准单元还支持通过软件调整片上振荡器的方式对 fdcan_ker_ck 进行校准。要计算微调值，用户必须从 FDCAN_CCU_CSTAT 读取 CCU 状态。来自 fdcan_ker_ck 的时钟被发送到输出 fdcan_tq_ck (FDCAN_CCFG.BCC = 1)。

输入时钟 fdcan_ker_ck 必须至少为 80 MHz。必须通过 FDCAN_CCFG.CDIV 配置 CCU 的时钟分频器，以便将 fdcan_tq_ck 调整到有效范围内。所有其他配置参数必须通过 FDCAN_CCFG 设置。为使 tFDCAN1 和 FDCAN2 正常工作，APB 时钟 fdcan_pclk 需要等于或大于时间片时钟 (fdcan_tq_ck)。不能进行 CAN FD 操作。

启动时，FDCAN 模块必须在 FDCAN_CCCR.INIT 复位之前均配置为受限工作模式 (FDCAN_CCCR.ASM = 1)。必须调整输入时钟 fdcan_ker_ck，直到 CAN 时钟校准单元达到 Precision_Calibrated 状态。现在，软件可以复位 FDCAN_CCCR.ASM，CANFD1 和 CANFD2 可以开始正常工作。

工作期间，软件必须定期检查 CAN 时钟校准单元是否仍处于 Precision_Calibrated 状态。如果 CAN 时钟校准单元已因 fdcan_ker_ck 偏移而退出 Precision_Calibrated 状态，则必须将 FDCAN_CCCR.INIT、FDCAN_CCCR.CCE 和 FDCAN_CCCR.ASM 编程为 1，从而使 FDCAN 进入受限工作模式。成功调整 fdcan_ker_ck 后（CAN 时钟校准单元处于 Precision_Calibrated 状态），FDCAN 模块可恢复正常工作。

注： *微调精度必须满足已配置比特率的 CAN 时钟容差要求。*

时钟校准有效

将 FDCAN_CCFG.BCC 复位为 0 可进入此工作模式。在此工作模式下，fdcan_ker_ck 由 CCU 控制。

CCU 输出信号 fdcan_tq_ck 的生成取决于 FDCAN1 的状态。输入时钟 fdcan_ker_ck 必须高于 80 MHz。需要配置 CCU 和 FDCAN1。不能进行 CAN FD 操作。

如果 FDCAN1 变为 Bus_Off，或其 INIT 位由用户命令置 1 (FDCAN_CCCR.INIT = 1)，则 CCU 进入 Not_Calibrated 状态。CANFD1 和 CANFD2 进入受限工作模式。

注： *FDCAN_CCFG.BCC 的复位值配置为 0 时，此模式为复位后的默认工作模式。*

53.4.7 TTCAN 操作（仅限 FDCAN1）

参考消息

参考消息是具有特定 CAN 标识符的数据帧。该消息会被除时间主控节点（参考消息发送方）以外的所有节点接收和接受。

对于 1 级参考消息，数据长度必须至少为 1；对于 0 级和 2 级参考消息，数据长度必须至少为 4；否则消息不会作为参考消息被接受。参考消息可通过其他数据最多扩展为总共八个 CAN 数据字节。标识符中除三个 LSB 以外的所有位会将消息定性为参考消息。最后三位指定最多八个潜在时间主控节点的优先级。保留的位会以逻辑 0 的形式发送，接收器会忽略这些位。参考消息是通过寄存器 FDCAN_TTRMC 配置的。

时间主控节点会发送参考消息。如果参考消息受到错误的干扰，则会立即重发。重发时，已发送的 Master_Ref_Mark 会更新。参考消息会定期发送，但允许停止定期发送（Next_is_Gap 位）并在下一基本周期开始时通过当前时间主控节点或其他潜在时间主控节点之一发起事件同步发送。

发送参考消息的节点是当前时间主控节点。时间主控节点可以发送其他消息。如果当前时间主控节点发生故障，优先级最高的潜在时间主控节点会代替它执行操作。如果某个节点既不是时间主控节点，也不是潜在时间主控节点，则该节点为时间接收节点。

1 级

1 级操作是通过 FDCAN_TTOCF.OM = 01 and FDCAN_TTOCF.GEN 配置的。1 级操作不提供外部时钟同步。与参考消息相关的信息存储在第一个数据字节中，如*表 483* 所示。Cycle_Count 为可选位。

表 483. 1 级参考消息的第一个字节

位	0	1	2	3	4	5	6	7
第一个字节	Next_is_Gap	保留	Cycle_Count[5;0]					

2 级

2 级操作是通过 FDCAN_TTOCF.OM = 10 和 FDCAN_TTOCF.GEN 配置的。与参考消息相关的信息存储在前四个数据字节中，如*表 484* 所示。Cycle_Count 和 FDCAN_NTU_Res 的低四位为可选位。TTCAN 不会对接收到的参考消息中的 NTU_Res[3:0] 进行评估，始终会将这些位作为 0 发送。

表 484. 2 级参考消息的前四个字节

位	0	1	2	3	4	5	6	7
第一个字节	Next_is_Gap	保留	Cycle_Count[5;0]					
第二个字节	NTU_Res[6:4]			NTU_Res[3:0]				Disc_Bit
第三个字节	Master_Ref_Mark[7:0]							
第四个字节	Master_Ref_Mark[15:8]							

0 级

0 级操作是通过 FDCAN_TTOCF.OM = 11 配置的。0 级参考消息中不能实现外部事件同步时间触发操作。与参考消息相关的信息存储在前四个数据字节中，如下表所示。在 0 级参考消息中，Next_is_Gap 始终为 0。Cycle_Count 和 NTU_Res 的低四位为可选位。TTCAN 不会对接收到的参考消息中的 NTU_Res[3:0] 进行评估，始终会将这些位作为 0 发送。

表 485. 0 级参考消息的前四个字节

位	0	1	2	3	4	5	6	7
第一个字节	Next_is_Gap	保留	Cycle_Count[5;0]					
第二个字节	NTU_Res[6:4]			NTU_Res[3:0]				Disc_Bit
第三个字节	Master_Ref_Mark[7:0]							
第四个字节	Master_Ref_Mark[15:8]							

53.4.8 TTCAN 配置

TTCAN 定时

网络时间单位 (NTU) 是所有时间的测量单位。NTU 在整个网络内是恒定不变的，被网络系统设计人员定义为优先项。在 TTCAN 1 级参考消息中，NTU 是标称 CAN 位时间。在 TTCAN 0 级和 2 级参考消息中，NTU 是零点几个物理秒。

NTU 是本地时间的时基。每经过一个 NTU，本地时间的整数部分（16 位值）会递增 1 次。周期时间和全局时间均源自本地时间。本地时间、周期时间和全局时间的小数部分（3 位值）不可读。

在 TTCAN 0 级和 2 级参考消息中，NTU 的长度由时间单位比 TUR 定义。TUR 通常是一个非整数，由以下公式计算得出 TUR = FDCAN_TURNA.NAV / FDCAN_TURCF.DC。NTU 长度由以下公式计算得出 NTU = CAN 时钟周期 x TUR。

TUR 分子配置 NC 为 18 位数，FDCAN_TURCF[NCL[15:0]] 可编程在 0x0000 到 0xFFFF 范围内。FDCAN_TURCF[NCH[17:16]] 通过硬编码的方式写入 0b01。如果将 0xnnnn 写入 FDCAN_TURCF[NCL[15:0]]，FDCAN_TURNA.NAV 的起始值为 0x10000 + 0x0nnnn = 0x1nnnn。TUR 分母配置 FDCAN_TURCF.DC 为 14 位数。FDCAN_TURCF.DC 可编程在 0x0001 到 0x3FFF 范围内（0x0000 为非法值）。

在 1 级参考消息中，NC 必须等于或高于 4 x FDCAN_TURCF.DC。在 0 级和 2 级参考消息中，要使 NTU 内部小数部分达到 3 位分辨率，NC 必须高于或等于 8 x FDCAN_TURCF.DC。

硬件复位会将 FDCAN_TURCF.DC 预设为 0x1000，将 FDCAN_TURCF.NCL 预设为 0x10000，因此 NTU 会包含 16 个 CAN 时钟周期。在 FDCAN_CCCR.INIT 复位或 FDCAN_TURCF.ELT 置 1 之前，本地时间和应用看门狗都不会启动。配置 NTU 之前，FDCAN_TURCF.ELT 不会置 1。将 FDCAN_TURCF.ELT 置 1 还会锁定对寄存器 FDCAN_TURCF 的写访问。

如果 FDCAN_TURCF.ELT 置 1，则启动时 FDCAN_TURNA.NAV 会从 NC (= FDCAN_TURCF.NCL + 0x10000) 更新。TTCAN 1 级参考消息中没有偏移补偿。FDCAN_TURNA.NAV 在工作期间不会改变，始终等于 NC。

在 TTCAN 0 级和 2 级参考消息中，FDCAN_TURNA.NAV 会在两种情况下发生改变。作为时间被控节点或备份时间主控节点工作且 FDCAN_TTOCF.ECC 位置 1 时，只要 TTCAN 处于同步状态 In_Schedule 或 In_Gap，FDCAN_TURNA.NAV 就会自动更新为通过监视的全局时间速度计算出的值。当它失去同步时，会返回到 NC。当作为实际的时间主控节点工作，并且 FDCAN_TTOCF.EECS 置 1 时，用户可以更新 FDCAN_TURCF.NCL。当用户将 FDCAN_TTOCN.ECS 置 1 时，FDCAN_TURNA.NAV 将从下一参考消息的 NC 新值进行更新。状态标志 FDCAN_TTOST.WECS 会在 FDCAN_TTOCN.ECS 置 1 时置 1，并会在 FDCAN_TURNA.NAV 更新时清零。当 FDCAN_TTOST.WECS 置 1 时，FDCAN_TURCF.NCL 被写锁定。

在 TTCAN 0 级和 2 级参考消息中，时钟校准过程会在 NC ± 2(FDCAN_TTOCF.LDSDL + 5) 的同步偏差限值范围 SDL 内调整 FDCAN_TURNA.NAV。FDCAN_TURCF.NCL 应编程为最大的适用值，以便在计算 FDCAN_TURNA.NAV 时达到最佳精度。

同步偏差 SD 是 NC 与 FDCAN_TURNA.NAV 之差 (SD = | NC - FDCAN_TURNA.NAV |)。该值受同步偏差限值 SDL 的限制，SDL 是通过其双对数 FDCAN_TTOCF.LDSDL (SDL = 2 (FDCAN_TTOCF.LDSDL + 5)) 配置的，不应超过 CAN 位定时配置提供的时钟容差。每个新的基本周期都会计算 SD。如果计算出的 TURNA[NAV] 与 NC 的偏差大于 SDL，或者参考消息中的 Disc_Bit 置 1，则偏移补偿会暂停，FDCAN_TTIR.GTE 会置 1，FDCAN_TTOSC.QCS 会复位，或者在 Disc_Bit = 1 的情况下，FDCAN_TTIR.GTD 会置 1。

表 486. TUR 配置示例

TUR	8	10	24	50	510	125000	32.5	100/12	529/17
NC	0x1FFF8	0x1FFFE	0x1FFF8	0x1FFEA	0x1FFFE	0x1E848	0x1FFE0	0x19000	0x10880
FDCAN _TURCF.DC	0x3FFF	0x3333	0x1555	0x0A3D	0x0101	0x0001	0x0FC0	0x3000	0x0880

FDCAN_TTOCN.ECS 通过下一条参考消息调度 NC 进行激活。FDCAN_TTOCN.SGT 通过下一条参考消息调度 FDCAN_TTGTP.TP 进行激活。当 FDCAN 为实际时间主控节点时，设置 FDCAN_TTOCN.ECS 和 FDCAN_TTOCN.SGT 时需要将 FDCAN_TTOCF.EECS 置 1（使能外部时钟同步）。

TTCAN 模块提供应用看门狗来验证应用程序的功能。用户必须定期处理该看门狗，否则所有 CAN 总线活动都会停止。应用看门狗限值 FDCAN_TTOCF.AWL 指定了两次处理看门狗的时间间隔（以 NTU 数为单位）。NTU 数最大值为 256。通过读取寄存器 FDCAN_TTOST 来处理应用看门狗。FDCAN_TTOST.AWE 指示是否及时处理了看门狗。如果应用无法处理应用看门狗，中断标志 FDCAN_TTIR.AW 将置 1。进行软件开发时，可通过将 FDCAN_TTOCF.AWL 编程为 0x00 禁止应用看门狗，请参见*第 53.6.3 节*。

接口信号定时

导致输出 FDCAN 触发时间标记中断脉冲 fdcan1_tmp 的计时事件，如果发生多次实例，则为 fdcan_tmp；如果只有一个实例，则为 fdcan_tmp。FDCAN 寄存器时间标记中断脉冲 fdcan1_rtp，如果发生多次实例，则为 fdcan_rtp；如果只有一个实例，则为 fdcan_rtp，它们是在 CAN 时钟域中生成的。

需考虑在 APB 时钟域中出现这一事件之前（FDCAN_TTIR.TTMI 置 1 或 FDCAN_TTIR.RTMI 置 1）使用一个时钟域交叉延迟。举例来说，信号可连接至另一 FDCAN 节点的定时输入 (fdcan_swt/fdcan_evt)，以便自动同步两个 TTCAN 网络。

参考消息完成（发送或接收）时，输出 FDCAN 周期开始脉冲 cycle_fdcan1_soc（多个实例时，只有一个实例时为 cycle_fdcan_soc）会激活。输出在 APB 时钟域中进行控制。

53.4.9 消息调度

FDCAN_TTOCF.TM 控制着 TTCAN 作为潜在时间主控节点还是时间被控节点工作。如果是潜在时间主控节点，参考消息标识符 FDCAN_TTRMC.RID 的三个 LSB 定义主控节点优先级，0 代表最高优先级，7 代表最低优先级。网络中不能存在两个使用相同主控优先级的节点。FDCAN_TTRMC.RID 用于识别参考消息。FDCAN_TTRMC.RMPS 与时间被控节点无关。

初始参考触发偏移 FDCAN_TTOCF.IRTO 是 7 位值，定义了在预期发送参考消息、但总线仍处于空闲状态时，备份时间主控节点等待多久（以 NTU 数为单位）后才会开始发送参考消息。FDCAN_TTOCF.IRTO 的建议值为主控节点优先级乘以相应的系数，具体系数值取决于网络中潜在时间主控节点之间的预期时钟偏移。在当前时间主控节点发生故障的情况下，各备份时间主控节点启动参考消息的顺序应与其主控节点优先级相对应，即使时钟偏移达到最大值时也不例外。

FDCAN_TTOCF.OM 决定了节点在 TTCAN 0 级、1 级还是 2 级中工作。在一个网络中，所有潜在时间主控节点都必须在同一级别工作。时间被控节点可工作在 2 级网络中的 1 级，但反过来不可以。最后的设置步骤是通过 FDCAN_TTOCF.OM 配置 TTCAN 工作模式。如果 FDCAN_TTOCF.OM = 00（事件驱动 CAN 通信），FDCAN 会按照 ISO 11898-1: 2015 的规定工作，没有时间触发器。如果 FDCAN_TTOCF.OM = 01（1 级），FDCAN 会按照 ISO 11898-4 的规定工作，但不能将基本周期同步到外部事件，参考消息中的 Next_is_Gap 位会

被忽略。如果 FDCAN_TTOCF.OM = 10（2 级），TTCAN 会按照 ISO 11898-4 的规定工作，包括基本周期的事件同步启动。如果 FDCAN_TTOCF.OM = 11（0 级），FDCAN 会作为事件驱动 CAN 工作，但会保留 2 级中已校准的全局时基。

FDCAN_TTOCF.EECS 可实现外部时钟同步，从而允许当前时间主控节点的应用程序在时间触发操作期间更新 TUR 配置，根据外部参考值调整时钟速度和（仅限 0 级和 2 级中）全局时钟相位。

TT 矩阵限值寄存器中的 FDCAN_TTMLM.ENTT 指定了系统矩阵中预期的 Tx_Trigger 数。该值为独占、单独仲裁和合并仲裁窗口对应的 Tx_Trigger 之和，不包括 Tx_Ref_Trigger。需要注意的是，该值通常不是 Tx_Trigger 内存元素数；必须考虑系统矩阵中的基本周期数和触发重复系数。

如果 FDCAN_TTMLM.ENTT 的配置不准确，则会导致发送计数下溢（FDCAN_TTIR.TXU = 1 且 FDCAN_TTOST.EL = 01，严重程度 1）或者发送计数上溢（FDCAN_TTIR.TXO = 1 且 FDCAN_TTOST.EL = 10，严重程度 2）。

注： *如果节点观察到的第一条参考消息没有 Cycle_Count 0，则该节点可能会使用其发送计数完成第一个矩阵周期，从而会造成发送计数下溢。只要节点处于同步状态，其 Tx_Trigger 就不会触发发送。*

FDCAN_TTMLM.CCM 指定系统矩阵中最后一个基本周期的编号。基本周期计数从 0 开始。如果系统矩阵包含 8 个基本周期，FDCAN_TTMLM.CCM 将为 7。FDCAN_TTMLM.CCM 会被时间被控节点忽略，参考消息的接收器会将接收到的周期计数视为实际基本周期的有效周期计数。

FDCAN_TTMLM.TXEW 指定发送使能窗口的长度（以 NTU 数为单位）。发送使能窗口是可开始发送的时间窗口开始处的时间段。如果由于与之前的时间窗口消息稍有重叠等原因而无法在发送使能窗口内启动消息发送，则无法在该时间窗口内启动发送。选择 FDCAN_TTMLM.TXEW 时必须考虑网络同步质量以及时间窗口长度与消息长度之间的关系。

触发内存

触发内存是 TTCAN 连接到的外部消息 RAM 的一部分（请参见*第 53.5.26 节*），最多存储 64 个触发元素。触发内存元素包括时间标记 TM、周期代码 CC、触发类型 TYPE、过滤器类型 FTYPE、消息编号 MNR、消息状态计数 MSC、时间标记事件内部 TMIN、时间标记事件外部 TMEX（请参见*第 53.4.23 节*）。

时间标记定义了触发在哪一周期时间生效。触发内存中的触发必须按其时间标记进行排序。时间标记最小的触发元素会写入第一个触发内存字中。对于类型为 Tx_Ref_Trigger、Tx_Ref_Trigger_Gap、Watch_Trigger、Watch_Trigger_Gap 和 End_of_List 的触发，会忽略消息编号和周期代码。

如果周期时间达到实际触发的时间标记，FSE 会切换到下一触发，并开始从触发内存中读取以下触发。如果是发送触发，一旦 FSE 切换到其触发，发送处理单元就会开始从消息 RAM 中读取消息。RAM 访问速度定义了发送触发与它之前的触发之间的最短时间步，发送处理单元必须能够在达到发送触发时间标记之前做好发送准备。RAM 访问速度还限制了两个匹配触发之间的非匹配（对于其周期代码）触发数，必须在达到其时间标记之前读取下一匹配触发。如果参考消息长度为 n 个 NTU，则时间标记小于 n 的触发将永不会激活，并会被视为配置错误处理。

周期时间的起始点是参考消息帧起始位的采样点。当周期时间达到 Tx_Ref_Trigger 时间标记时，会请求下一条参考消息。FDCAN 会在下一采样点对发送请求作出响应。会在帧起始位捕获新的 Sync_Mark，但周期时间会增加，直到参考消息成功发送（或接收），Sync_Mark 被当作新的 Ref_Mark。此时，周期时间会重新开始。因此，周期时间值绝不会小于 n（初始化时例外），n 表示参考消息的长度（以 NTU 数为单位）。

一个基本周期的长度：Tx_Ref_Trigger 时间标记 + 1 NTU + 1 CAN 位时间。

CAN FD 网络中所有节点的触发列表都是不同的。每个节点只知道自己的发送消息对应的 Tx_Trigger、自己处理的接收消息对应的 Rx_Trigger 以及与参考消息相关的触发。

触发类型

Tx_Ref_Trigger (TYPE = 0000) 和 Tx_Ref_Trigger_Gap (TYPE = 0001) 会触发通过时间主控节点发送参考消息。如果时间被控节点在其触发内存中遇到 Tx_Ref_Trigger(_Gap)，则会检测到配置错误（FDCAN_TTOST.EL = 11，严重程度 3）。Tx_Ref_Trigger_Gap 仅用于外部事件同步时间触发工作模式。在该模式下，当 FDCAN 同步状态为 In_Gap (FDCAN_TTOST.SYS = 10) 时，会忽略 Tx_Ref_Trigger。

Tx_Trigger_Single (TYPE = 0010)、Tx_Trigger_Continuous (TYPE = 0011)、Tx_Trigger_Arbitration (TYPE = 0100) 和 Tx_Trigger_Merged (TYPE = 0101) 会触发启动发送过程。这些触发类型定义了时间窗口起点。

如果消息缓冲区发送请求挂起位置 1，Tx_Trigger_Single 会在独占时间窗口中启动单次发送。成功发送后，发送请求挂起位会复位。

如果消息缓冲区发送请求挂起位置 1，Tx_Trigger_Continuous 会在独占时间窗口中启动发送。成功发送后，发送请求挂起位保持置 1 状态，并且消息缓冲区会在下一匹配的时间窗口中再次发送。

Tx_Trigger_Arbitration 启动仲裁时间窗口，Tx_Trigger_Merged 启动合并仲裁时间窗口。合并仲裁时间窗口的最后一个 Tx_Trigger 的类型必须为 Tx_Trigger_Arbitration。如果类型为 Tx_Trigger_Merged 的触发之后不是类型为 Tx_Trigger_Merged 或 Tx_Trigger_Arbitration 的 Tx_Trigger，则会检测到配置错误（FDCAN_TTOST.EL = 11，严重程度 3）。可为同一发送消息缓冲区定义多个 Tx_Trigger，某些基本周期中可能会忽略它们，具体取决于周期代码。计算预计 Tx_Trigger 数 (FDCAN_TTMLM.ENTT) 时，必须考虑周期代码。

Watch_Trigger (TYPE = 0110) 和 Watch_Trigger_Gap (TYPE = 0111) 会检查缺失的参考消息。时间主控节点和时间被控节点都会使用它们。Watch_Trigger_Gap 仅用于外部事件同步时间触发工作模式。在该模式下，当 FDCAN 同步状态为 In_Gap (FDCAN_TTOST.SYS = 10) 时，会忽略 Watch_Trigger。

Rx_Trigger (TYPE = 1000) 用于检查独占时间窗口中周期性消息的接收情况。在达到 In_Schedule 或 In_Gap 状态之前，Rx_Trigger 不会激活。Rx_Trigger 的时间标记必须放置在消息发送结束之后，与时间窗口边界无关。某些基本周期中可能会忽略 Rx_Trigger，具体取决于周期代码。在 Rx_Trigger 的时间标记处，会检查在该时间标记之前以及周期开始之后或上一 Rx_Trigger 之后接收的最后一条消息是否与 MNR 引用的验收过滤器元素相匹配。接受的消息会根据验收过滤结果存储在两个接收 FIFO 之一，与 Rx_Trigger 无关。Rx_Trigger 引用的验收过滤器元素应放置在过滤器列表开始处，以确保过滤会在达到 Rx_Trigger 时间标记之前完成。

Time_Base_Trigger (TYPE = 1001) 用于根据 TMIN 和 TMEX 的配置生成内部/外部事件。

End_of_List (TYPE = 1010 … 1111) 属于非法触发类型，如果在触发内存中出现 Watch_Trigger 和 Watch_Trigger_Gap 之前出现了 End_of_List 触发，则会检测到配置错误（FDCAN_TTOST.EL = 11，严重程度 3）。

节点触发列表的限制

同一周期时间和周期计数不能同时激活两个触发，但在不同基本周期中激活的触发（周期代码不同）可共用同一时间标记。

Rx_Trigger 和 Time_Base_Trigger 不能置于 Tx_Trigger_Single/Continuous/Arbitration 的发送使能窗口中，但可置于 Tx_Trigger_Merged 之后。

如果触发置于 Watch_Trigger 之后（或者在 FDCAN_TTOST.SYS = 10 时置于 Watch_Trigger_Gap 之后），触发将永远不会激活。如果参考消息按时发送，监视触发本身将不会激活。

所有未使用的触发内存字（在 Watch_Trigger 之后；FDCAN_TTOST.SYS = 10 时则在 Watch_Trigger_Gap 之后）必须设为触发类型 End_of_List。

潜在时间主控节点的典型触发列表将以 Tx_Trigger 和 Rx_Trigger 数目开头，后接 Tx_Ref_Trigger 和 Watch_Trigger。对于进行外部事件同步时间触发通信的网络，后接 Tx_Ref_Trigger_Gap 和 the Watch_Trigger_Gap。时间被控节点的触发列表与之相同，但不包含 Tx_Ref_Trigger 和 Tx_Ref_Trigger_Gap。

每个基本周期开始时，也就是每次接收或发送参考消息时，都会从第一个触发内存元素开始对触发列表进行处理。FSE 会查找第一个周期代码与当前周期计数匹配的触发。FSE 会等待周期时间达到触发时间标记并激活触发。随后，FSE 会在列表中查找下一个周期代码与当前周期计数匹配的触发。

对于 Tx_Ref_Trigger 和 Tx_Ref_Trigger_Gap 附近的时间，需要做特殊考虑。在竞争主控的时间主控节点中，Tx_Ref_Trigger 的有效时间标记可能会递减，以便成为第一个启动参考消息的节点。在备份时间主控节点中，Tx_Ref_Trigger 或 Tx_Ref_Trigger_Gap 的有效时间标记是其配置的时间标记与参考触发偏移 FDCAN_TTOCF.IRTO 之和。如果达到 2 级错误 (FDCAN_TTOST.EL = 10)，有效时间标记则为其时间标记与 0x127 之和。此范围内不应放入其他触发元素，否则时间标记的出现顺序可能会混乱并标记为配置错误。只要参考消息及时到达，Tx_Ref_Trigger 之后到达的触发元素就可能永远也不会生效。

以下参数之间具有相关性：

- APB 时钟频率
- 触发 RAM 访问的速度和等待时间
- 验收过滤器列表的长度
- 触发元素数量
- 触发元素周期代码过滤的复杂程度
- 触发元素时间标记之间的偏移

触发处理示例

下例介绍了如何通过节点系统矩阵生成触发列表。假设节点 A 是第一个时间主控节点，并已知道*表 487* 所示的系统矩阵的部分。

表 487. 系统矩阵，节点 A

周期计数	时间标记						
	1	2	3	4	5	6	7
0	Tx7	-	-	-	-	TxRef	错误
1	Rx3	-	Tx2、Tx4		-	TxRef	错误
2	-	-	-	-	-	TxRef	错误
3	Tx7	-	Rx5	-	-	TxRef	错误
4	Tx7	-	-	Rx6	-	TxRef	错误

周期计数从 0 开始，计数到 0、1、3、7、15、31、63 为止（系统矩阵中对应的基本周期编号分别为 1、2、4、8、16、32、64）。最大周期计数由 FDCAN_TTMLM.CCM 配置。周期代码 CC 由重复系数（= 最高有效位 1 的值）以及系统矩阵中的第一个基本周期编号（= 最高有效位 1 之后的位字段）组成。

例如，如果周期代码为 0b0010011（重复系数 = 16，第一个基本周期 = 3），并且最大周期计数 FDCAN_TTMLM.CCM = 0x3F，则会在周期计数达到 3、19、35、51 时出现匹配。

触发元素由时间标记 TM、周期代码 CC、触发类型 TYPE 和消息编号 MNR 组成。进行发送时，MNR 会引用发送缓冲区编号 (0 ...31)。进行接收时，MNR 会引用验收过滤期间匹配的过滤器元素的编号 (0 ...127)。根据过滤器类型 FTYPE 的配置，将引用 11 位或 29 位消息 ID 过滤器列表。

此外，还可以配置触发元素，以生成时间标记事件内部 TMIN 和时间标记事件外部 TMEX。消息状态计数 MSC 保持计数器值 (0 ...7)，用于在触发元素时间标记激活时立即调度独占时间窗口中周期性消息的错误。

表 488. 触发列表，节点 A

触发	时间标记 TM[15:0]	周期代码 CC [6:0]	触发类型 TYPE [3:0]	消息编号 MNR [6:0]
0	Mark1	0b0000100	Tx_Trigger_Single	7
1	标记 1	0b1000000	Rx_Trigger	3
2	标记 1	0b1000011	Tx_Trigger_Single	7
3	标记 3	0b1000001	Tx_Trigger_Merged	2
4	标记 3	0b1000011	Rx_Trigger	5
5	标记 4	0b1000001	Tx_Trigger_Arbitration	4
6	标记 4	0b1000100	Rx_Trigger	6
7	标记 6	N/A	Tx_Ref_Trigger	0（参考）
8	标记 7	N/A	Watch_Trigger	N/A
9	N/A	N/A	End_of_List	N/A

Tx_Trigger_Single、Tx_Trigger_Continuous、Tx_Trigger_Merged、Tx_Trigger_Arbitration、Rx_Trigger 和 Time_Base_Trigger 仅对指定的周期代码有效。对于所有其他触发类型，会忽略周期代码。

FSE 启动基本周期时，会从 0 开始扫描触发列表，直至达到时间标记大于周期时间且其周期代码 CC 与实际周期计数相匹配的触发，或者遇到了类型为 Tx_Ref_Trigger、Tx_Ref_Trigger_Gap、Watch_Trigger 或 Watch_Trigger_Gap 的触发。

当周期时间达到时间标记 TM 时，会启动由触发类型 TYPE 和消息编号 MNR 定义的操作。达到 End_of_List 时，配置中会出错。

在标记 6 处，会发送参考消息（始终为 TxRef）。发送参考消息后，FSE 会返回到触发列表开头。达到标记 7 处的监视触发时，节点不能发送参考消息；此时会启动错误处理过程。

配置错误检测

下列情况下，会通过 FDCAN_TTOST.EL = 11（严重程度 3）指示配置错误：

- FSE 到达列表中周期代码与当前周期计数匹配、但时间标记小于周期时间的触发。
- 之前激活的触发为 Tx_Trigger_Merged，FSE 到达列表中周期代码与当前周期计数匹配、但既不是 Tx_Trigger_Merged、也不是 Tx_Trigger_Arbitration、Time_Base_Trigger 或 Rx_Trigger 的触发。
- FDCAN_TTOCF.TM=0 的节点（时间被控节点）的 FSE 遇到 Tx_Ref_Trigger 或 Tx_Ref_Trigger_Gap。
- 在具有匹配周期代码的 Tx_Trigger 的发送使能窗口（由 FDCAN_TTMLM.TXEW 定义）中放置的任何时间标记处。
- 时间标记放置在 Tx_Ref_Trigger 的时间标记附近，参考触发偏移 FDCAN_TTOST.RTO 会导致在周期时间中的测量顺序反转。

TTCAN 调度初始化

FDCAN_CCCR.INI 复位时，会开始同步到 TTCAN 消息调度。TTCAN 可严格按照时间触发 (FDCAN_TTOCF.GEN = 0) 或外部事件同步时间触发 (FDCAN_TTOCF.GEN = 1) 操作。FDCAN_TTOST.SYS = 00 时（非同步），所有节点都会在其触发列表开头以周期时间 0 启动，除参考消息外，不会使能任何发送。处于外部事件同步时间触发工作模式下的节点将忽略 Tx_Ref_Trigger 和 Watch_Trigger，并将使用 Tx_Ref_Trigger_Gap 和 Watch_Trigger_Gap 代替，直到第一条参考消息决定是否激活间隙。

时间被控节点

完成配置后，如果时间被控节点在达到 Watch_Trigger 之前未接收到任何消息，则会忽略其 Watch_Trigger 和 Watch_Trigger_Gap。当时间被控节点达到 Init_Watch_Trigger 时，中断标志 FDCAN_TTIR.IWT 置 1，FSE 冻结，周期时间失效，但节点仍可参与 CAN 总线通信（进行确认或发送错误标志）。接收到的第一条参考消息将重新启动 FSE 和周期时间。

注： *Init_Watch_Trigger 不属于触发列表。它作为内部计数器执行操作，会递增计数到 0xFFFF（= 最大周期时间）。*

如果时间被控节点在达到 Watch_Trigger 之前接收到了任何除参考消息以外的消息，则会判定发生了致命错误（FDCAN_TTOST.EL = 11，严重程度 3），并将中断标志 FDCAN_TTIR.WT 置 1，然后切断其 CAN 总线输出并进入总线监控模式（FDCAN_CCCR.MON 设为 1）。在总线监控模式下，仍可以接收消息，但不能发送任何显性位，因此不能提供确认。

注： *要退出致命错误状态，用户必须设置 FDCAN_CCCR.INIT = 1。FDCAN_CCCR.INIT 复位后，节点会重新开始 TTCAN 通信。*

如果同步过程中未出错，第一条参考消息会设置 FDCAN_TTOST.SYS = 01（同步），第二条参考消息会将 FDCAN 同步状态（具体取决于 Next_is_Gap 位）设为 FDCAN_TTOST.SYS = 11 (In_Schedule) 或 FDCAN_TTOST.SYS = 10 (In_Gap)，从而使能所有 Tx_Trigger 和 Rx_Trigger。

潜在时间主控节点

完成配置后，潜在时间主控节点将在达到其 Tx_Ref_Trigger（外部事件同步时间触发操作时为 Tx_Ref_Trigger_Gap）后开始发送参考消息。如果潜在时间主控节点在达到 Watch_Trigger 之前未接收到任何消息或未成功发送参考消息（假设的原因：所有其他节点仍处于复位或配置状态，不能提供确认），则会忽略其 Watch_Trigger 和 Watch_Trigger_Gap。当潜在时间主控节点达到 Init_Watch_Trigger 时，尝试进行的发送会中止，中断标志 FDCAN_TTIR.IWT 置 1，FSE 冻结，周期时间失效，但节点仍可参与 CAN 总线通信（提供确认或发送错误标

志）。复位 FDCAN_TTIR.IWT 后，将在下一次满足 Init_Watch_Trigger 条件或接收另一 CAN 消息时重新使能参考消息发送。接收参考消息不会重新启动 FSE。

如果潜在时间主控节点在接收到除参考消息以外的消息后达到 Watch_Trigger，则会判定发生了致命错误（FDCAN_TTIR.IWT，严重程度 3），并将中断标志 FDCAN_TTIR.WT 置 1，然后切断其 CAN 总线输出并进入总线监控模式（FDCAN_CCCR.MON 设为 1）。在总线监控模式下，仍可以接收消息，但不能发送任何显性位，因此不能提供确认。

如果初始化过程中未检测到错误，第一条参考消息会设置 FDCAN_TTOST.SYS = 01（同步），第二条参考消息会将 FDCAN 同步状态（具体取决于 Next_is_Gap 位）设为 FDCAN_TTOST.SYS = 11 (In_Schedule) 或 FDCAN_TTOST.SYS = 10 (In_Gap)，从而使能所有 Tx_Trigger 和 Rx_Trigger。

如果潜在时间主控节点是上一条参考消息的发送方，则该潜在时间主控节点就是当前时间主控节点 (FDCAN_TTOST.MS = 11)，否则是备份时间主控节点 (FDCAN_TTOST.MS = 10)。

当所有潜在时间主控节点均已完成配置后，网络中时间主控优先级最高的节点将成为当前时间主控节点。

53.4.10 TTCAN 间隙控制

所有与间隙控制相关的功能仅在 FDCAN 工作于外部事件同步时间触发模式下时 (FDCAN_TTOCF.GEN = 1) 才能使用。在该工作模式下，由于会在系统矩阵的基本周期之间插入间隙，因此 FDCAN 消息调度可能被中断。所有连接到 CAN 网络的节点必须针对外部事件同步时间触发操作进行配置。

在间隙期间，所有发送都会停止，CAN 总线保持空闲状态。间隙在下一条参考消息启动新的基本周期时结束。间隙在本身由参考消息启动（例如 Next_is_Gap = 1）的基本周期结束时开始。间隙是通过当前时间主控节点启动的。

当前时间主控节点可通过两种方式启动间隙。当应用程序写入 FDCAN_TTOCN.NIG = 1 时，可在软件控制下启动间隙。Next_is_Gap 位以 1 的形式与下一参考消息一起发送。当应用程序通过写入 FDCAN_TTOCN.GCS = 1 的方式使能事件触发输入引脚 fdcan_evt 时，也可在硬件控制下启动时隙。当参考消息启动且 FDCAN_TTOCN.GCS 置 1 时，事件触发引脚 fdcan_evt 上的高电平会将 Next_is_Gap 置 1。

参考消息完成后，FDCAN_TTOST.WFE 位将立即向时间主控节点以及时间被控节点发布间隙。当前的基本周期将继续，直至达到其最后的时间窗口。最后的时间窗口之后的时间就是间隙时间。

对于实际时间主控节点和潜在时间主控节点，当最后一个基本周期结束并且间隙时间开始时，FDCAN_TTOST.GSI 将置 1。在作为时间被控节点的节点中，位 FDCAN_TTOST.GSI 将保持为 0。

当潜在时间主控节点处于同步状态 In_Gap (FDCAN_TTOST.SYS = 10) 时，可通过四种方法有计划地结束间隙：

1. 在软件控制下写入 FDCAN_TTOCN.FGP = 1。
2. 在硬件控制下 (FDCAN_TTOCN.GCS = 1)，如果事件触发输入引脚 fdcan_evt 上出现从高电平变为低电平的边沿，则 FDCAN_TTOCN.FGP 会置 1，并会重新启动调度。
3. 第三种方法是时间触发重启。当 FDCAN_TTOCN.TMG = 1 时，下一个寄存器时间标记中断 (FDCAN_TTIR.RTMI = 1) 会将 FDCAN_TTOCN.FGP 置 1 并启动参考消息。
4. 最后一种方法是：任何潜在时间主控节点会在达到其 Tx_Ref_Trigger_Gap 后结束间隙，前提是要同步的事件并未及时发生。

以上任何方法都不会导致基本周期因参考消息而中断。

在间隙时间开始后将 FDCAN_TTOCN.FGP 置 1 会立即开始发送参考消息，继而会同步消息调度。如果 FDCAN_TTOCN.FGP 在间隙时间开始之前置 1（基本周期仍在进行中），则将在基本周期结束时（Tx_Ref_Trigger 时）启动下一参考消息，消息调度中将不存在间隙时间。

在严格的时间触发操作中，参考消息中的位 Next_is_Gap = 1 将被忽略，同时事件触发输入引脚 fdcan_evt 以及位 FDCAN_TTOCN.NIG、FDCAN_TTOCN.FGP 和 FDCAN_TTOCN.TMG 也会被忽略。

53.4.11 秒表

秒表功能可捕获由外部事件触发的 FDCAN 内部时间值（本地时间、周期时间或全局时间）。

要使能秒表功能，应用程序必须先通过 FDCAN_TTOCN.SWS 将本地时间、周期时间或全局时间定义为秒表源。如果 FDCAN_TTOCN.SWS 不是 00，并且 TT 中断寄存器标志 FDCAN_TTIR.SWE 为 0，则 FDCAN_TTTOCN.SWS 选择的实际时间值将在秒表触发引脚 fdcan_swt 的下一上升/下降沿（通过 FDCAN_TTOCN.SWP 配置）复制到 FDCAN_TTCPT.SWV。这会将中断标志 FDCAN_TTIR.SWE 置 1。应用程序读取 FDCAN_TTCPT.SWV 后，可通过将 FDCAN_TTIR.SWE 复位为 0 的方式使能下一秒表事件。

53.4.12 本地时间、周期时间、全局时间和外部时钟同步

时间触发 CAN 可分为两个级别：

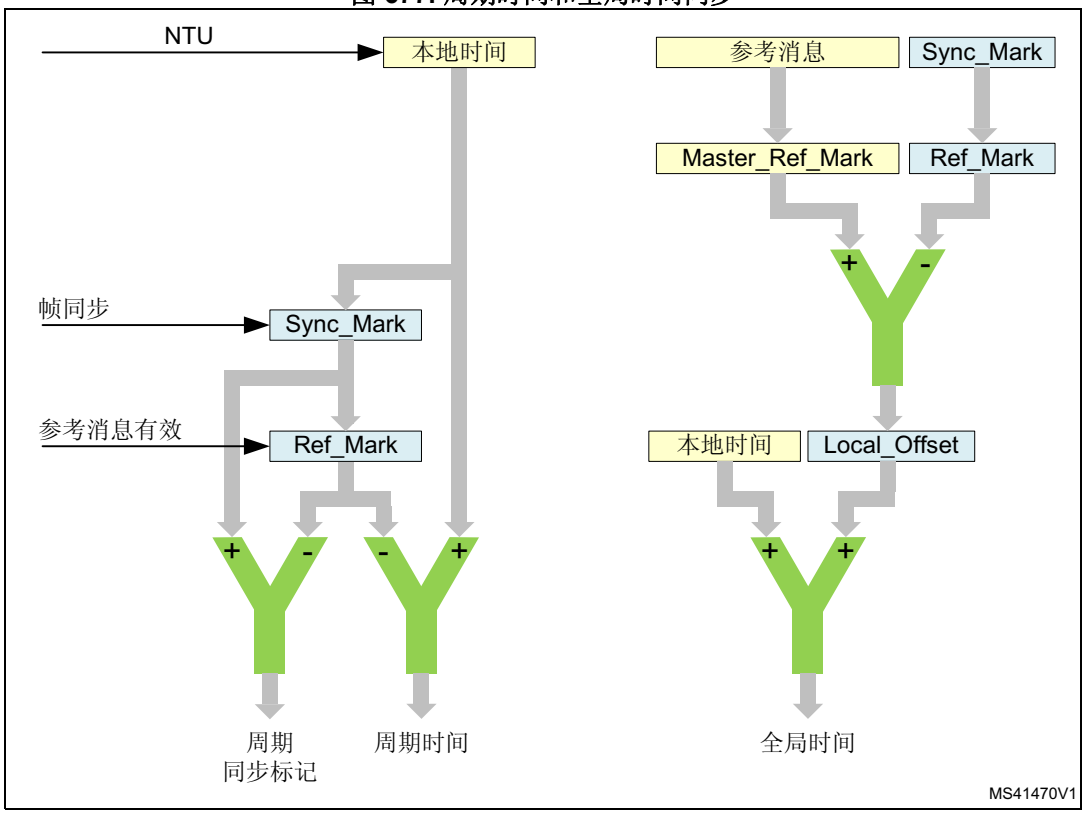
1. 1 级仅使用周期时间提供时间触发操作。
2. 2 级提供增强的同步质量、全局时间和外部时钟同步。在两种级别下，所有定时特性均基于本地时基，即本地时间。

本地时间为 16 位循环计数器，每个 NTU 都会递增。在内部，NTU 由 3 位计数器表示，可被视为本地时间的小数部分（三个二进制数位）。一般情况下，3 位 NTU 计数器每隔一个 NTU 会递增 8 次。如果 NTU 的长度小于 8 个 CAN 时钟周期（也许在 1 级中是这样配置的，或者是 2 级中的时钟校准造成的），则会调整 NTU 分数的长度，调整后的 NTU 计数器每隔一个 NTU 仅会递增 4 次。

图 677 介绍了包括时间主控节点在内的所有 FDCAN 节点以同一方式执行的周期时间和全局时间的同步。如果接收或发送的消息调用对消息本地时间的捕获，则该消息为帧同步事件。这一帧同步事件发生在每个帧起始 (SoF) 位的采样点，并会使本地时间以 Sync_Mark 形式存储。将会捕获 Sync_Marks 和 Ref_Marks，包括 3 位小数部分。

只要成功发送或接收有效的参考消息，内部 Ref_Mark 就会通过 Sync_Mark 进行更新。Ref_Mark 和 Sync_Mark 之差是存储在寄存器 FDCAN_TTCSM 中的周期同步标记（周期同步标记 = Sync_Mark - Ref_Mark）。Ref_Mark 与本地时间实际值之差的 16 个最高有效位为周期时间（周期时间 = 本地时间 - Ref_Mark）。

图 677. 周期时间和全局时间同步



可从 FDCAN_TTCTC.CT 读取的周期时间是节点本地时间与 Ref_Mark 之差，两者均同步到 APB 时钟域并截断为 16 位。

全局时间仅存在于 TTCAN 0 级和 2 级中，在 1 级中无效。全局时间的节点视图为全局时间的本地图像（以本地 NTU 数为单位）。完成配置后，潜在时间主控节点将使用自己的本地时间作为全局时间。时间主控节点通过在参考消息中发送自己的 Ref_Mark 作为 Master_Ref_Mark（字节 3 和 4），以将自己的本地时间建立为全局时间。可从 FDCAN_TTLGT.GT 读取的全局时间是节点本地时间与其本地偏移之和，两者均同步到 APB 时钟域并截断为 16 位。小数部分仅用于时钟同步。

接收参考消息的节点会通过将其本地 Ref_Mark 与接收到的 Master_Ref_Mark 进行比较来计算其本地偏移（请参见*图 678*）。全局时间的节点视图为本地时间加上本地偏移。在从未接收到另一时间主控节点参考消息的潜在时间主控节点中，Local_Offset 为 0。如果某个节点在首次接收到其他参考消息后成为当前时间主控节点，Local_Offset 将冻结为其上一个值。在时间接收节点中，由于存在时钟偏移，在另一节点成为时间主控节点时、或者出现全局时间不连续的情况下（由参考消息中的 Disc_Bit 指示），可能会对 Local_Offset 稍作调整。除全局时间不连续的情况之外，由寄存器 FDCAN_TTLGT 提供给应用程序的全局时间会通过低通滤波进行平滑处理，以得到连续单调值。

图 678. TTCAN 0 级和 2 级偏移补偿

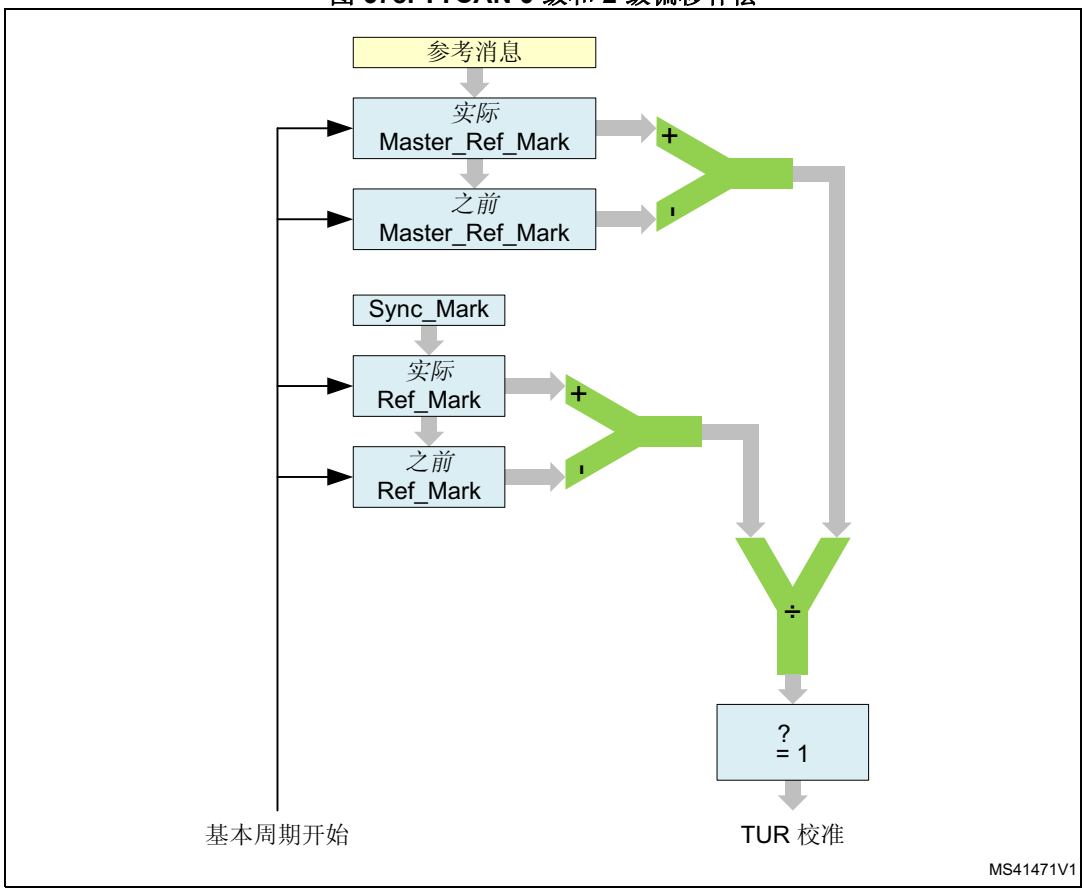


图 678 介绍了 TTCAN 0 级和 2 级中，接收节点每次是怎样通过比较本地时间与全局时间中基本周期的长度来补偿其自身的本地时钟与时间主控节点时钟之间的偏移的。如果两个值之间存在偏差，参考消息中的 Disc_Bit 不会置 1，并且会计算 FDCAN_TURNA.NAV 的新值。如果同步偏差 SD = | NC - FDCAN_TURNA.NAV | ≤ SDL（同步偏差限值），则 FDCAN_TURNA.NAV 的新值会生效，否则会暂停自动偏移补偿。

在 TTCAN 0 级和 2 级中，FDCAN_TTOST.QCS 指示自动偏移补偿处于激活状态还是暂停状态。在 TTCAN 1 级中，FDCAN_TOST.QCS 始终为 1。

当前时间主控节点可将其本地时钟速度和全局时间相位同步到外部时钟源。此操作是通过位 FDCAN_TTOCF.EECS 使能的。

可使用秒表功能（请参见*第 53.4.11 节*）测量本地时钟与外部时钟的时钟速度之差。调整本地时钟速度时，要先将新计算的分子配置下限写入 FDCAN_TURCF.NCL（操作过程中不能更新 FDCAN_TURCF.DC）。向 FDCAN_TTOCN.ECS 写入 1 后，新值会生效。

调整全局时间相位时，要先将相位偏移写入 TT 全局时间预设寄存器 FDCAN_TTGTP。向 FDCAN_TTOCN.SGT 写入 1 后，新值会生效。在调整完全局时间相位后发送的第一条参考消息会将 Disc_Bit 设为 1。

FDCAN_TTOST.QGTP 指示节点全局时间是否与时间主控节点全局时间同相。在 TTCAN 1 级中以及 TTCAN 0、2 级中超过同步偏差限值 (FDCAN_TTOST.QCS = 0) 的情况下，FDCAN_TTOST.QGTP 永久为 0。通过对全局时间进行低通滤波为应用提供连续单调值时，此位暂时为 0。如果上一参考消息包含 Disc_Bit = 1 或者 FDCAN_TTOST.QCS = 0，则不会进行低通滤波。

53.4.13 TTCAN 错误级别

ISO 11898-4 规定了四种错误严重等级：

1. S0——无错
2. S1——警告——仅通知应用，响应特定于应用。
3. S2 错误——通知应用。将会禁止在独占或仲裁时间窗口中的所有发送（也就是说，可能不会启动任何数据帧或远程帧）。潜在时钟主控节点仍会发送参考触发偏移 FDCAN_TTOST.RTO 设为最大值 127 的参考消息。
4. S3——严重错误——通知应用。所有 CAN 总线操作都会停止，也就是说不允许发送显性位，且 FDCAN_CCCR.MON 会置 1。应用更新配置（将 FDCAN_CCCR.CCE 置 1）之前，S3 错误条件都会保持有效。

如果同时检测到多个错误，则会处理优先级最高的错误。检测到错误时，会通过 FDCAN_TTIR.ELC 通知应用。错误级别通过 FDCAN_TTOST.EL 进行监控。

按照 ISO 11898-4 的要求，TTCAN 会指示以下错误条件：

- Config_error (S3)
  - 如果合并仲裁时间窗口未正确关闭或者 Tx_Trigger 的时间标记超过 Tx_Ref_Trigger，则将错误级别 FDCAN_TTOST.EL 设为 11。
- Watch_Trigger_Reached (S3)
  - 如果因参考消息丢失而到达监视触发，则将错误级别 FDCAN_TTOST.EL 设为 11。
- Application_Watchdog (S3)
  - 如果应用无法处理应用看门狗，则将错误级别 FDCAN_TTOST.EL 设为 11。应用看门狗通过 FDCAN_TTOCF.AWL 进行配置。应用看门狗通过读取寄存器 FDCAN_TTOST 进行处理。如果未及时处理看门狗，位 FDCAN_TTOST.AWE 和中断标志 FDCAN_TTIR.AW 会置 1，所有 FDCAN 通信停止，FDCAN 设为进入总线监控模式（FDCAN_CCCR.MON 设为 1）。
- CAN_Bus_Off (S3)
  - 进入 CAN_Bus_Off 状态会将错误级别 FDCAN_TTOST.EL 设为 11。CAN_Bus_Off 状态通过 FDCAN_PSR.BO = 1 和 FDCAN_CCCR.INIT = 1 来指示。
- Scheduling_Error_2 (S2)
  - 如果其中一个 Tx_Trigger 的 MSC 已达到 7，则将错误级别 FDCAN_TTOST.EL 设为 10。此外，中断标志 FDCAN_TTIR.SE2 也会置 1。如果之前的矩阵周期中没有任何 Tx_Trigger 的 MSC 为 7，则会在矩阵周期开始时将错误级别 FDCAN_TTOST.EL 复位为 00。
- Tx_Overflow (S2)
  - 如果发送计数等于或大于预期的 Tx_Trigger 数 FDCAN_TTMLM.ENTT，且发生了 Tx_Trigger 事件，则会将错误级别 FDCAN_TTOST.EL 设为 10。此外，中断标志 FDCAN_TTIR.TXO 也会置 1。如果新矩阵周期开始时发送计数大于 FDCAN_TTMLM.ENTT，则会将错误级别 FDCAN_TTOST.EL 复位为 00。
- Scheduling_Error_1 (S1)
  - 如果一个矩阵周期内（独占时间窗口中的）所有触发内存元素最大 MSC 与最小 MSC 之差大于 2，或者独占 Rx_Trigger 的其中一个 MSC 达到了 7，则会将错误级别 FDCAN_TTOST.EL 设为 01。此外，中断标志 FDCAN_TTIR.SE1 也会置 1。如果一个矩阵周期内以上条件均无效，则会将错误级别 FDCAN_TTOST.EL 复位为 00。

- Tx_Underflow (S1)
  - 如果新矩阵周期开始时发送计数小于预期的 Tx_Trigger 数 FDCAN_TTMLM.ENTT，则会将错误级别 FDCAN_TTOST.EL 设为 01。此外，中断标志 FDCAN_TTIR.TXU 也会置 1。如果新矩阵周期开始时发送计数至少为 FDCAN_TTMLM.ENTT，则会将错误级别 FDCAN_TTOST.EL 复位为 00。

53.4.14 TTCAN 消息处理

参考消息

对于潜在时间主控节点，参考消息的标识符是通过 TFDCAN_TTRMC.RID 配置的。不需要使用专用发送缓冲区来发送参考消息。发送参考消息时，TTCAN 1 级的第一个数据字节（也就是 TTCAN 0 级的前四个数据字节和 TTCAN 2 级的前四个数据字节）将由 FSE 提供。

如果参考消息有效负载选择 FDCAN_TTRMC.RMPS 置 1，其余参考消息负载（1 级：字节 2-8；0、2 级：字节 5-6）将从发送缓冲区 0 获取。在这种情况下，会使用消息缓冲区 0 中的数据长度 DLC 代码。

表 489. 随参考消息一起发送的数据字节数

FDCAN_TTRMC.RMPS	FDCAN_TXBRP.TRP0	0 级	1 级	2 级
0	0	4	1	4
0	1	4	1	4
1	0	4	1	4
1	1	4 + MBO	1 + MBO	4 + MBO

要随参考消息一起发送额外的有效负载，对于 1 级，必须配置 DLC > 1；对于 0 级和 2 级，需要配置 DLC > 4。此外，消息缓冲区 0 的发送请求挂起位 FDCAN_TXBRP.TRP0 必须置 1（请参见*表 489*）。如果参考消息启动时位 FDCAN_TXBRP.TRP0 未置 1，则参考消息仅会与由 FSE 提供的数据字节一起发送。

对参考消息进行验收过滤时，会使用参考标识符 FDCAN_TTRMC.RID。

消息接收

对于事件驱动的 CAN 通信，消息接收是通过同样的方法在接收 FIFO 中完成的（请参见*接收处理单元*）。

消息状态计数 MSC 是相应的触发内存元素的组成部分，配置过程中必须初始化为 0。当 TTCAN 处于同步状态 In_Gap 或 In_Schedule 时，会更新该值。接收到消息 Rx_Trigger 时，会进行更新。此时，会检查在该基本周期中接收到的最新消息在哪一验收过滤器元素发生了匹配。匹配过滤器编号会作为验收过滤器结果存储起来。如果此编号与该触发内存元素中定义的过滤器编号相同，则 MSC 会递减 1。如果验收过滤器结果与为该过滤器元素定义的过滤器编号不同，或者验收过滤器结果清零，则 MSC 会递增 1。每次出现 Rx_Trigger 以及每次周期开始时，上一验收过滤器结果都会清零。

设置 Rx_Trigger 的时间标记值时，应确保目标消息的接收和验收过滤已完成。此时必须考虑 RAM 访问时间以及过滤器列表的顺序。建议将用于 Rx_Trigger 的过滤器放置在过滤器列表开头。不建议为参考消息使用 Rx_Trigger。

消息发送

为了发送时间触发消息，TTCAN 提供 32 个专用发送缓冲区（请参见*发送暂停*）。当 FDCAN 配置为进行时间触发操作时（FDCAN_TTOCF.OM = 01 或 10），发送 FIFO 或发送队列不可用。

触发内存中的每个 Tx_Trigger 都指向包含特定消息的发送缓冲区。如果某一给定的发送缓冲区包含要在基本周期或矩阵周期中发送多次的消息，则该发送缓冲区可能有多个 Tx_Trigger。

应用程序必须定期在同步到周期时间的时间点更新数据。用户负责确保不会发送未完全更新的消息。为了确保这一点，用户必须按以下方式进行操作：

Tx_Trigger_Single/Tx_Trigger_Merged/Tx_Trigger_Arbitration

- 读取 FDCAN_TXBTO 以检查之前的发送是否已完成
- 更新发送缓冲区配置和/或有效负载
- 发出添加请求，将发送缓冲区请求挂起位置 1

Tx_Trigger_Continous

- 发出取消请求，将发送缓冲区请求挂起位复位
- 读取 FDCAN_TXBCF 以检查取消是否已完成
- 更新发送缓冲区配置和/或有效负载
- 发出添加请求，将发送缓冲区请求挂起位置 1

与相应 Tx_Trigger 一起存储的 MSC 会提供关于发送是否成功的信息。

如果发送因 CAN 总线在相应的发送使能窗口内未处于空闲状态而无法开始，或者消息已启动且无法成功完成，则 MSC 会递增 1。如果消息成功发送，或者消息本可以在其发送使能窗口中启动、但因发送被禁止的原因而未启动（错误级别为 S2 的 TTCAN 或用户禁止了这一特殊消息），MSC 会递减 1。

发送缓冲区可进行动态管理，也就是说，多条标识符不同的消息可共用同一发送缓冲区元素。在这种情况下，用户必须通过检查 FDCAN_TXBRP 来确保要重新配置的发送缓冲区元素没有挂起的发送请求。

如果应更新具有挂起的发送请求的发送缓冲区，用户必须先发出取消请求，并通过读取 FDCAN_TXBCF 检查取消是否完成，然后再开始更新。

发送处理单元会将消息从消息 RAM 传输到其中间输出缓冲区，消息会放入刚好在 Tx_Trigger 元素（定义发送窗口起点）之前激活的触发元素处。传输时间期间以及传输时间之后，发送消息可能无法更新，其 FDCAN_TXBRP 位可能不会更改。要控制该传输时间，可在 Tx_Trigger 之前放置附加的触发元素。举例来说，触发元素可能是不需要进行其他任何操作的 Time_Base_Trigger。Tx_Trigger 与上一次触发的时间标记之差必须足够大，以确保即使其他模块对 RAM 的访问负载很高时，发送处理单元也能从消息 RAM 中读取四个字。

在独占时间窗口中发送

当周期时间达到 Tx_Trigger_Single 或 Tx_Trigger_Continuous 的时间标记时，会开始时间触发发送。不会在总线上与来自其他节点的消息进行仲裁。MSC 会根据发送尝试结果进行更新。成功完成通过 Tx_Trigger_Single 启动的发送后，相应的发送缓冲区请求挂起位会复位。成功完成通过 Tx_Trigger_Continuous 启动的发送后，相应的发送缓冲区请求挂起位会保持置 1。如果发送因受到干扰未成功完成，将在下一次（其中一个）Tx_Trigger 激活时重复发送。

在仲裁时间窗口中发送

当周期时间达到 Tx_Trigger_Arbitration 的时间标记时，会开始时间触发发送。多个节点可同时开始发送。在这种情况下，消息必须与来自其他节点的消息进行仲裁。MSC 未更新。如果发送不成功（仲裁失败或受到干扰），将在下一次（其中一个）Tx_Trigger 激活时重复发送。

在合并仲裁时间窗口中发送

合并仲裁时间窗口的用途是使多个节点能够按照 CAN 仲裁提供的顺序即时发送有限数量的帧，而不是通过一个节点进行突发发送。由于节点在该时间窗口中没有独占访问权限，因此可能出现部分请求的发送不成功的情况。

失去仲裁或受到错误干扰的消息可在同一合并仲裁时间窗口中重新发送。如果相应的发送请求挂起标志成功通过取消发送进行了复位，则不会启动重新发送过程。

在单独发送窗口中，发送处理单元会发送由触发元素的消息编号指示的消息。在合并仲裁时间窗口中，最多可处理触发列表中的三个消息编号，并且将按照触发列表定义的顺序尝试对其进行发送。如果在发送第一条消息之前达到第四条消息的时间标记（或者由用户取消），将会忽略第四个请求。

合并仲裁时间窗口中的发送不是通过时间触发的。消息发送可在其时间标记之前开始，如果总线不处于空闲状态，也可以在时间标记之后开始。

由特定节点在合并仲裁时间窗口中发送的消息将按照其 Tx_Trigger 的顺序开始发送，因此，如果 CAN 优先级较低的消息竞争总线流量，可能会阻碍以下优先级较高的消息成功发送。配置触发列表时必须考虑这一点。Time_Base_Trigger 可放置在相邻的 Tx_Trigger 之间，以定义相应发送缓冲区数据需要更新的截止时间。

53.4.15 TTCAN 中断和错误处理

TT 中断寄存器 FDCAN_TTIR 由四部分组成。每个中断均可单独通过 TT 中断使能寄存器 FDCAN_TTIE 中的相应位使能。必须由用户清零，否则将一直保持置位。通过向对应位位置写入 1 将标志清零。

第一部分包括标志 CER、AW、WT 和 IWT。每个标志都指示会停止 CAN 通信的致命错误。除 IWT 之外，这些错误条件需要先重新配置 FDCAN 模块，然后才能重新开始进行通信。

第二部分包括标志 ELC、SE1、SE2、TXO、TXU 和 GTE。每个标志都指示 CAN 通信受到干扰的错误。如果 CAN 通信中断是瞬时故障造成的（例如 CAN 总线上的干扰），则会通过 FDCAN 协议故障处理进行处理，不需要应用程序干预。

第三部分由标志 GTD、GTW、SWE、TTMI 和 RTMI 组成。前两个标志是由全局时间事件（仅限 0 级、2 级）控制的，需要应用程序作出响应。如果通过引脚 fdcan_swt 的上升沿触发秒表事件，则会捕获内部时间值。触发时间标记中断会通知应用已达到特定 Time_Base_Trigger。寄存器时间标记中断指示 FDCAN_TTOCN.TMC 引用的时间（周期时间、本地时间或全局时间）等于时间标记 FDCAN_TTTMK.TM。也可用于结束间隙。

第四部分由标志 SOG、CSM、SMC 和 SBC 组成。这些标志可用于将应用程序同步到通信调度。

53.4.16 0 级

TTCAN 0 级不是 ISO11898-4 的一部分。该工作模式会使处于 TTCAN 2 级的硬件保持已校准的全局时基，根据 ISO11898-1 的规定，该工作模式还可用于事件驱动 CAN。

0 级操作是通过 FDCAN_TTOCF.OM = 11 配置的。在该模式下，FDCAN 在事件驱动 CAN 通信中工作，没有固定的调度，FDCAN_TTOCF.GEN 的配置会被忽略。0 级参考消息中不能实现外部事件同步操作。同步时基通过发送参考消息来保持。

在 0 级参考消息中，触发内存未激活，因此不需要进行配置。当周期时间达到 FDCAN_TTOCF.IRTO 0x200 时，时间标记中断标志 (FDCAN_TTIR.TTMI) 置 1，它会提醒用户为消息缓冲区 0 设置发送请求。当周期时间达到 0xFF00 时，Watch_Trigger 中断标志 (FDCAN_TTIR.WT) 会置 1。选择这些值后，会有足够的裕量进行稳定的时钟校准。不会进行后续的 TT 错误检查。

此外，还可以使用寄存器时间标记中断 (FDCAN_TTIR.RTMI)。

参考消息针对 2 级操作进行配置。已接收的参考消息通过在寄存器 FDCAN_TTRMC 中配置的标识符来识别。要发送参考消息，仅可使用消息缓冲区 0。用户为消息缓冲区 0 设置发送请求后，节点可随时发送参考消息，不存在参考触发偏移。

0 级操作是通过以下寄存器位配置的：

- FDCAN_TTRMC
- FDCAN_TTOCF，但 EVTP、AWL、GEN 位除外
- FDCAN_TTMLM，但 ENTT、TXEW 位除外
- FDCAN_TURCF

0 级操作是通过以下寄存器位控制的：

- FDCAN_TTOCN，但 NIG、TMG、FGP、GCS、TTMIE 位除外
- FDCAN_TTGTP
- FDCAN_TTTMK
- FDCAN_TTIR，不包括 CER、AW、IWT SE2、SE1、TXO、TXU、SOG 位（无功能）
- FDCAN_TTIR 以下位的功能发生了变化
  - 不是由触发内存定义的 TTMI——在周期时间 FDCAN_TTOCF.IRTO 0x200 激活
  - 不是由触发内存定义的 WT - 在周期时间 0xFF00 激活

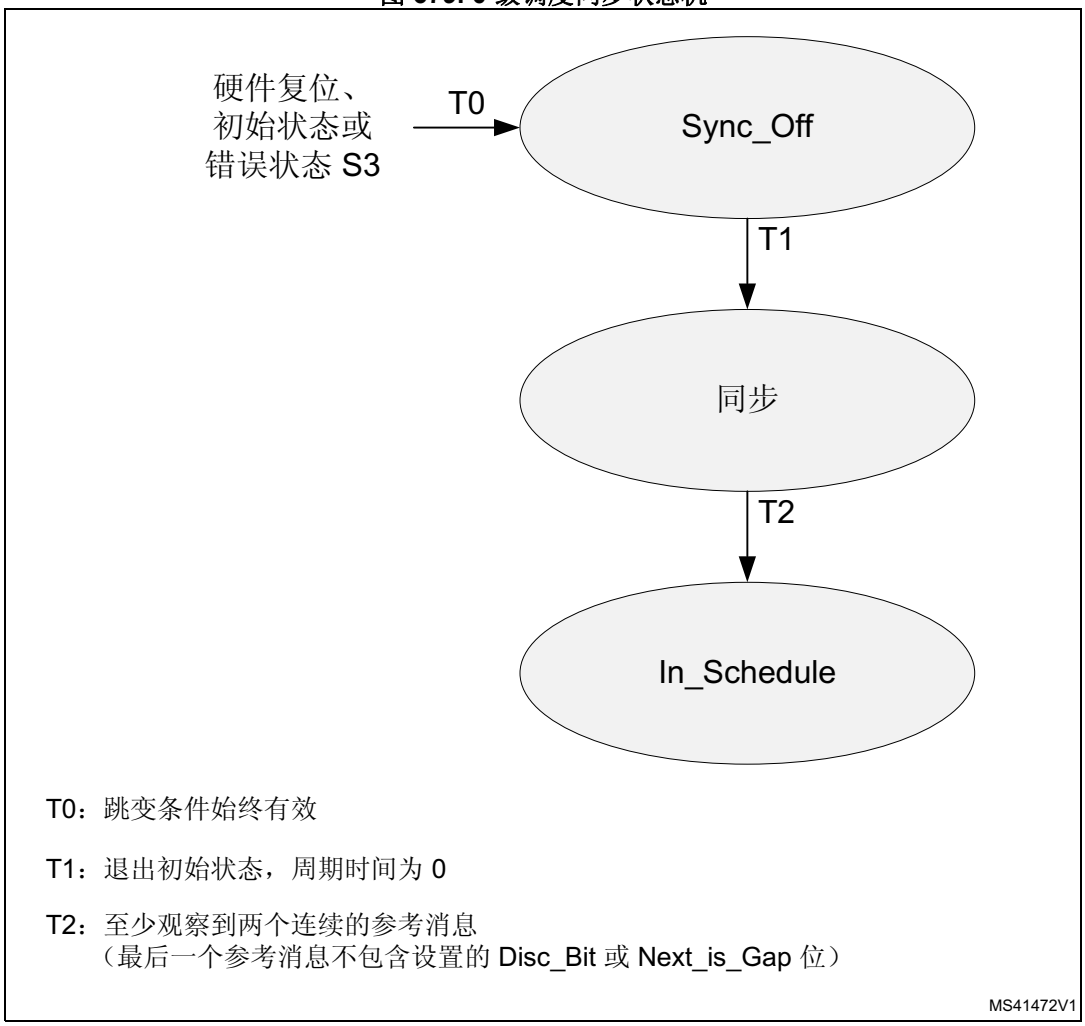
0 级操作是通过以下寄存器位指示的：

- TTOST，不包括 AWE、WFE、GSI、GFI、RTO 位（无功能）

同步

图 679 介绍了 TTCAN 0 级操作中的状态和状态转换。0 级没有 In_Gap 状态。

图 679. 0 级调度同步状态机



错误级别处理

0 级操作期间，仅可能出现以下错误条件：

- Watch_Trigger_Reached (S3)，已达到周期时间 0xFF00
- CAN_Bus_Off (S3)

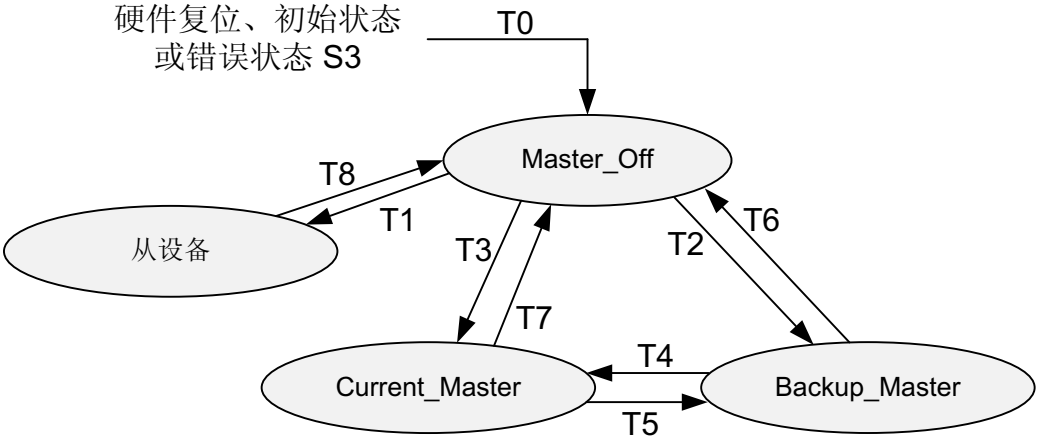
由于不会发生 S1 和 S2 错误，因此错误级别仅可在 S0（无错）和 S3（严重错误）之间切换。在 TTCAN 0 级中，对 S3 错误的处理方式有所不同。达到错误级别 S3 时，FDCAN_TTOST.SYS 和 FDCAN_TTOST.MS 均会复位，中断标志 FDCAN_TTOST.SYS 和 FDCAN_TTOST.MS 会置 1。

进入错误级别 S3 (FDCAN_TTOST.EL = 11) 时，不会进入总线监控模式（与 TTCAN 1 级和 2 级不同）。发送（时间主控节点）或接收（时间被控节点）下一参考消息后，会自动退出 S3 错误级别。

主控/被控节点关系

图 680 介绍了 TTCAN 0 级中主控/被控节点的关系。如果发生 S3 错误，FDCAN 会恢复到状态 Master_Off。

图 680. 0 级主控节点与被控节点的关系



T0：跳变条件始终有效

T1：观察到参考消息（非潜在时间主控节点时）

T2：观察到参考消息（主设备优先级与自身主设备优先级不同），错误状态 ≠ S3

T3：观察到参考消息（主设备优先级与自身主设备优先级相同），错误状态 ≠ S3

T4：观察到参考消息（自身主设备优先级）

T5：观察到参考消息（主设备优先级高于自身主设备优先级）

T6：错误状态 S3

T7：错误状态 S3

T8：错误状态 S3

MS41473V1

53.4.17 与外部时间调度同步

可使用此功能将 FDCAN 调度的相位同步到外部调度（例如另一 TTCAN 网络或 FlexRay 网络的调度）。仅当 FDCAN 为当前时间主控节点时 (FDCAN_TTOST.MS = 11)，才可使用此功能。

外部同步是通过事件触发输入引脚 fdcan_evt 控制的。如果 FDCAN_TTOCN.ESCN 位置 1，则当事件触发引脚 fdcan_evt 上出现上升沿时，FDCAN 会将其实际周期时间与通过 FDCAN_TTGTP.CTP 配置的目标相位值进行比较。

将 FDCAN_TTOCN.ESCN 置 1 之前，用户需要使用 FDCAN 间隙控制等功能（请参见*第 53.4.10 节*）调整两个时间调度的相位。当用户将 FDCAN_TTOCN.ESCN 置 1 时，FDCAN_TTOSTSPL 会置 1。

如果事件触发引脚 fdcan_evt 上升沿处的周期时间与目标相位值 FDCAN_TTGTP.CTP 之差大于 9 个 NTU，锁相位 FDCAN_TTOST.SPL 会复位，中断标志 FDCAN_TTIR.CSM 置 1。当另一节点变为时间主控节点时，FDCAN_TTOST.SPL 也会复位（且 FDCAN_TTIR.CSM 会置 1）。

如果 FDCAN_TTOST.SPL 和 FDCAN_TTOCN.ESCN 均置 1，并且事件触发引脚 fdcan_evt 上升沿处的周期时间与目标相位值 FDCAN_TTGTP.CTP 之差小于或等于 9 个 NTU，锁相位 FDCAN_TTOST.SPL 保持置 1，测得的差值会作为参考触发偏移值，用于调整下一次发送的参考消息的相位。

注： *每次基本周期开始时，都会对事件触发引脚 fdcan_evt 使能上升沿检测。第一个上升沿会触发实际周期时间与 FDCAN_TTGTP.CTP 的比较。下一基本周期开始之前的所有后续边沿都会被忽略。*

53.4.18 FDCAN 接收缓冲区和 FIFO 元素

最多可在消息 RAM 中配置 64 个接收缓冲区和两个接收 FIFO。每个接收 FIFO 部分均可配置为最多存储 64 个已接收消息。接收缓冲区/FIFO 元素的结构如*表 490* 所示，相关说明请参见*表 491*。

表 490. 接收缓冲区和 FIFO 元素

位	31	30	29	28:24	23	22	21	20:16	15:8	7:0
R0	ESI	XTD	RTR	ID[28:0]						
R1	ANMF	FIDX[6:0]			Res.	FDF	BRS	DLC[3:0]	RXTS[15:0]	
R2	DB3[7:0]				DB2[7:0]				DB1[7:0]	DB0[7:0]
R3	DB7[7:0]				DB6[7:0]				DB5[7:0]	DB4[7:0]
⋮	⋮				⋮				⋮	
Rn	DBm[7:0]				DBm-1[7:0]				DBm-2[7:0]	DBm-3[7:0]

可对元素大小进行配置，以通过寄存器 FDCAN_RXESC 存储数据字段多达 64 字节的 CAN FD 消息。

表 491. 接收缓冲区和 FIFO 元素说明

位字段	说明
R0 位 31 ESI	错误状态指示符 (Error state indicator) 0：发送节点为错误主动状态 1：发送节点为错误被动状态
R0 位 30 XTD	扩展标识符 (Extended identifier) 向用户指示接收到的帧具有标准标识符还是扩展标识符。 0：11 位标准标识符 1：29 位扩展标识符
R0 位 29 RTR	远程发送请求 (Remote transmission request) 向用户指示接收到的帧属于数据帧还是远程帧。 0：接收到的帧为数据帧 1：接收到的帧为远程帧
R0 位 28:0 ID[28:0]	标识符 (Identifier) 标准标识符或扩展标识符取决于 XTD 位。标准标识符存储在 ID[28:18] 中。

表 491. 接收缓冲区和 FIFO 元素说明（续）

位字段	说明
R1 位 31 ANMF	接受非匹配帧 (Accepted non-matching frame) 可通过 FDCAN_GFC.ANFS 和 FDCAN_GFC.ANFE 使能接受非匹配帧。 0：接收到的帧与过滤器索引 FIDX 匹配 1：接收到的帧与任何接收过滤器元素都不匹配
R1 位 30:24 FIDX[6:0]	过滤器索引 (Filter index) 0-127 = 匹配接收验收过滤器元素的索引（ANMF = 1 时无效）。 范围为 0 到 FDCAN_SIDFC.LSS.- 1 或 FDCAN_XIDFC.LSE.- 1。
R1 位 21 FDF	FD 格式 (FD Format) 0：标准帧格式 1：FDCAN 帧格式（新 DLC 编码和 CRC）
R1 位 20 BRS	比特率切换 (Bitrate Switch) 0：接收帧时不切换比特率 1：接收帧时切换比特率
R1 位 19:16 DLC[3:0]	数据长度代码 (Data length code) 0-8：经典 CAN + CAN FD：接收到的帧包含 0-8 个数据字节 9-15：经典 CAN：接收到的帧包含 8 个数据字节 9-15：CAN FD：接收到的帧包含 12/16/20/24/32/48/64 个数据字节
R1 位 15:0 RXTS[15:0]	接收时间戳 (Rx timestamp) 开始接收帧时捕获的时间戳计数器值。分辨率取决于时间戳计数器预分频器 FDCAN_TSCC.TCP 的配置。
R2 位 31:24 DB3[7:0]	数据字节 3 (Data Byte 3)
R2 位 23:16 DB2[7:0]	数据字节 2 (Data Byte 2)
R2 位 15:8 DB1[7:0]	数据字节 1 (Data Byte 1)
R2 位 7:0 DB0[7:0]	数据字节 0 (Data Byte 0)
R3 位 31:24 DB7[7:0]	数据字节 7 (Data Byte 7)
R3 位 23:16 DB6[7:0]	数据字节 6 (Data Byte 6)
R3 位 15:8 DB5[7:0]	数据字节 5 (Data Byte 5)
R3 位 7:0 DB4[7:0]	数据字节 4 (Data Byte 4)
⋮	⋮
Rn 位 31:24 DBm[7:0]	数据字节 m (Data Byte m)

表 491. 接收缓冲区和 FIFO 元素说明（续）

位字段	说明
Rn 位 23:16 DBm-1[7:0]	数据字节 m-1 (Data Byte m-1)
Rn 位 15:8 DBm-2[7:0]	数据字节 m-2 (Data Byte m-2)
Rn 位 7:0 DBm-3[7:0]	数据字节 m-3 (Data Byte m-3)

53.4.19 FDCAN 发送缓冲区元素

发送缓冲区部分可配置为保存专用发送缓冲区以及发送 FIFO/发送队列。如果发送缓冲区部分由专用发送缓冲区和发送 FIFO/发送队列共享，则专用发送缓冲区会在发送缓冲区部分的起始处开始，后面是分配给发送 FIFO 或发送队列的缓冲区。发送处理单元会评估发送缓冲区配置 FDCAN_TXBC.TFQS 和 FDCAN_TXBC.NDTB，以区分专用发送缓冲区与发送 FIFO/发送队列。可对元素大小进行配置，以通过寄存器 FDCAN_TXESC 存储数据字段多达 64 字节的 CAN FD 消息。

表 492. 发送缓冲区和 FIFO 元素

位	31			24	23					16	15	8	7	0
T0	ESI	XTD	RTR	ID[28:0]										
T1	MM[7:0]				EFC	Res.	FDF	BPS	DLC[3:0]		保留			
T2	DB3[7:0]				DB2[7:0]						DB1[7:0]		DB0[7:0]	
T3	DB7[7:0]				DB6[7:0]						DB5[7:0]		DB4[7:0]	
⋮	⋮				⋮						⋮			
Tn	DBm[7:0]				DBm-1[7:0]						DBm-2[7:0]		DBm-3[7:0]	

表 493. 发送缓冲区元素说明

位字段	说明
T0 位 31 ESI(1)	错误状态指示符 (Error state indicator) 0：CAN FD 格式的 ESI 位仅取决于错误被动标志 1：CAN FD 格式的 ESI 位会进行隐性发送
T0 位 30 XTD	扩展标识符 (Extended identifier) 0：11 位标准标识符 1：29 位扩展标识符
T0 位 29 RTR(2)	远程发送请求 (Remote transmission request) 0：发送数据帧 1：发送远程帧
T0 位 28:0 ID[28:0]	标识符 (Identifier) 标准标识符或扩展标识符取决于 XTD 位。必须将标准标识符写入 ID[28:18]。

表 493. 发送缓冲区元素说明（续）

位字段	说明
T1 位 31:24 MM[7:0]	消息标记 (Message marker) 在发送缓冲区配置期间由 CPU 写入。复制到发送事件 FIFO 元素中，用于标识发送消息状态。
T1 位 23 EFC	事件 FIFO 控制 (Event FIFO control) 0：不存储发送事件 1：存储发送事件
T1 位 21 FDF	FD 格式 (FD Format) 0：以经典 CAN 格式发送帧 1：以 CAN FD 格式发送帧
T1 位 20 BRS(3)	比特率切换 (Bitrate Switch) 0：发送 CAN FD 帧时不切换比特率 1：发送 CAN FD 帧时切换比特率
T1 位 19:16 DLC[3:0]	数据长度代码 (Data length code) 0-8：经典 CAN + CAN FD：接收到的帧包含 0-8 个数据字节 9-15：经典 CAN：接收到的帧包含 8 个数据字节 9-15：CAN FD：接收到的帧包含 12/16/20/24/32/48/64 个数据字节
T2 位 31:24 DB3[7:0]	数据字节 3 (Data Byte 3)
T2 位 23:16 DB2[7:0]	数据字节 2 (Data Byte 2)
T2 位 15:8 DB1[7:0]	数据字节 1 (Data Byte 1)
T2 位 7:0 DB0[7:0]	数据字节 0 (Data Byte 0)
T3 位 31:24 DB7[7:0]	数据字节 7 (Data Byte 7)
T3 位 23:16 DB6[7:0]	数据字节 6 (Data Byte 6)
T3 位 15:8 DB5[7:0]	数据字节 5 (Data Byte 5)
T3 位 7:0 DB4[7:0]	数据字节 4 (Data Byte 4)
⋮	⋮
Tn 位 31:24 DBm[7:0]	数据字节 m (Data Byte m)
Tn 位 23:16 DBm-1[7:0]	数据字节 m-1 (Data Byte m-1)

表 493. 发送缓冲区元素说明（续）

位字段	说明
Tn 位 15:8 DBm-2[7:0]	数据字节 m-2 (Data Byte m-2)
Tn 位 7:0 DBm-3[7:0]	数据字节 m-3 (Data Byte m-3)

1. 发送缓冲区的 ESI 位会与错误被动标志进行或运算，以确定已发送 FD 帧中 ESI 位的值。按照 CAN FD 协议规范的要求，错误主动节点可选择隐性发送 ESI 位，但错误被动节点始终隐性发送 ESI 位。
2. 当 RTR = 1 时，即使 FDCAN_CCCR.FDOE 使能以 CAN FD 格式进行发送，FDCAN 也会按照 ISO11898-1 的规定发送远程帧。
3. 仅当 CAN FD 操作使能时 (FDCAN_CCCR.FDOE = 1)，才会评估 ESI、FDF 和 BRS 位。仅当额外设置 FDCAN_CCCR.BRSE = 1 时，才会评估 BRS 位。

53.4.20 FDCAN 发送事件 FIFO 元素

每个元素均存储与已发送消息相关的消息。通过读取发送事件 FIFO，用户会按消息的发送顺序获取此类信息。关于发送事件 FIFO 的状态信息，可从寄存器 FDCAN_TXEFS 中获取。

表 494. 发送事件 FIFO 元素

位	31			24	23					16	15	8	7	0
E0	ESI	XTD	RTR	ID[28:0]										
E1	MM[7:0]				ET[1:0]	EDL	BRS	DLC[3:0]	TXTS[15:0]					

表 495. 发送事件 FIFO 元素说明

位字段	说明
E0 位 31 ESI	错误状态指示符 (Error state indicator) – 0：发送节点为错误主动状态 – 1：发送节点为错误被动状态
E0 位 30 XTD	扩展标识符 (Extended Identifier) – 0：11 位标准标识符 – 1：29 位扩展标识符
E0 位 29 RTR	远程发送请求 (Remote transmission request) – 0：发送数据帧 – 1：发送远程帧
E0 位 28:0 ID[28:0]	标识符 (Identifier) 标准标识符或扩展标识符取决于 XTD 位。必须将标准标识符写入 ID[28:18]。
E1 位 31:24 MM[7:0]	消息标记 (Message marker) 从发送缓冲区复制到发送事件 FIFO 元素中，用于标识发送消息状态。

表 495. 发送事件 FIFO 元素说明（续）

位字段	说明
E1 位 23:22 EFC	事件类型 (Event type) – 00：保留 – 01：发送事件 – 10：取消后仍发送（在 DAR 模式下发送时始终设为该值） – 11：保留
E1 位 21 EDL	扩展数据长度 (Extended data length) – 0：标准帧格式 – 1：FDCAN 帧格式（新 DLC 编码和 CRC）
E1 位 20 BRS	比特率切换 (Bitrate Switch) – 0：发送帧时不切换比特率 – 1：发送帧时切换比特率
T1 位 19:16 DLC[3:0]	数据长度代码 (Data length code) – 0-8：发送包含 0-8 个数据字节的帧 – 9-15：发送包含 8 个数据字节的帧
E1 位 15:0 TXTS[15:0]	发送时间戳 (Tx timestamp) 开始发送帧时捕获的时间戳计数器值。分辨率取决于时间戳计数器预分频器 FDCAN_TSCC.TCP 的配置。

53.4.21 FDCAN 标准消息 ID 过滤器元素

最多可为 11 位标准 ID 配置 128 个过滤器元素。访问标准消息 ID 过滤器元素时，其地址为过滤器列表标准起始地址 FDCAN_SIDFC.FLSSA 加上过滤器元素的索引 (0 ... 127)。

表 496. 标准消息 ID 过滤器元素

位	31		24	23	16	15	8	7	0
S0	SFT[1:0]	SFEC[2:0]		SFID1[10:0]		Res.		SFID2[10:0]	

表 497. 标准消息 ID 过滤器元素位字段说明

位字段		说明
位 31:30 SFT[1:0](1)		标准过滤器类型 (Standard filter type) – 00：从 SFID1 到 SFID2 的范围过滤器 – 01：用于 SFID1 或 SFID2 的双 ID 过滤器 – 10：典型过滤器：SFID1 = 过滤器，SFID2 = 掩码 – 11：禁止过滤器元素
位 29:27 SFEC[2:0]		标准过滤器元素配置 (Standard filter element configuration) 所有使能的过滤器元素都用于对标准帧进行验收过滤。当发现第一个匹配的已使能过滤器元素时，或者到达过滤器列表结尾处时，会停止验收过滤。如果 SFEC =“100”、“101”或“110”，则出现匹配时，会将中断标志 FDCAN_IR.HPM 置 1，并会生成中断（若使能）。在这种情况下，寄存器 FDCAN_HPMS 会更新为优先级匹配的状态。 – 000：禁止过滤器元素 – 001：如果过滤器匹配，则存储在接收 FIFO 0 中 – 010：如果过滤器匹配，则存储在接收 FIFO 1 中 – 011：如果过滤器匹配，则拒绝 ID – 100：如果过滤器匹配，则设置优先级 – 101：如果过滤器匹配，则设置优先级并存储在 FIFO 0 中 – 110：如果过滤器匹配，则设置优先级并存储在 FIFO 1 中 – 111：存储在接收缓冲区中或作为调试消息，忽略 FDCAN_SFT[1:0] 的配置
位 26:16 SFID1[10:0]		标准过滤器 ID 1 (Standard filter ID 1) 标准 ID 过滤器元素的第一个 ID。 过滤接收缓冲区或调试消息时，此位字段定义要存储的标准消息的 ID。接收的标识符必须完全匹配，不使用屏蔽机制。
位 15:0	SFID2[15:10]	标准过滤器 ID 2 (Standard filter ID 2) 该位字段具有不同含义，具体视 SFEC 的配置而定： – SFEC = 001 ...110 标准 ID 过滤器元素的第二个 ID。 – SFEC = 111：用于接收缓冲区或调试消息的过滤器
	SFID2[10:9]	确定接收的消息存储在接收缓冲区中，还是作为调试消息序列中的消息 A、B 或 C 进行处理。 – 00：将消息存储在接收缓冲区中 – 01：调试消息 A – 10：调试消息 B – 11：调试消息 C
	SFID2[8:6]	用于控制扩展接口的过滤器事件引脚。如果相应的位位置为 1，则当发生过滤器匹配时，会在相关过滤器事件引脚上产生脉冲，脉冲持续时间为一个 fdcan_pclk 周期。 SFID2[8] 由校准单元使用。
	SFID2[5:0]	定义与用于存储匹配消息的接收缓冲区起始地址 FDCAN_RXBC.RBSA 的偏移。

1. 当 SFT =“11”时，会禁止过滤器元素，验收过滤会继续进行（与 SFEC =“000”时的行为相同）。

注：　　如果配置了保留值，则会将过滤器元素视为已禁止。

53.4.22 FDCAN 扩展消息 ID 过滤器元素

最多可为 29 位扩展 ID 配置 64 个过滤器元素。访问扩展消息 ID 过滤器元素时，其地址为过滤器列表扩展起始地址 FDCAN_XIDFC.FLESA 加上过滤器元素的索引 (0 ... 63) 的二倍。

表 498. 扩展消息 ID 过滤器元素

位	31	30	29	28	0
F0	EFEC[2:0]			EFID1[28:0]	
F1	EFTI[1:0]		保留	EFID2[28:0]	

表 499. 扩展消息 ID 过滤器元素位字段说明

位字段	说明
F0 位 31:29 EFEC[2:0]	扩展过滤器元素配置 (Extended filter element configuration) 所有使能的过滤器元素都用于对扩展帧进行验收过滤。当发现第一个匹配的已使能过滤器元素时，或者到达过滤器列表结尾处时，会停止验收过滤。如果 EFEC = 100、101 或 110，则出现匹配时，会将中断标志 FDCAN_IR.HPM 置 1，并会生成中断（若使能）。在这种情况下，寄存器 FDCAN_HPMS 会更新为优先级匹配的状态。 – 000：禁止过滤器元素 – 001：如果过滤器匹配，则存储在接收 FIFO 0 中 – 010：如果过滤器匹配，则存储在接收 FIFO 1 中 – 011：如果过滤器匹配，则拒绝 ID – 100：如果过滤器匹配，则设置优先级 – 101：如果过滤器匹配，则设置优先级并存储在 FIFO 0 中 – 110：如果过滤器匹配，则设置优先级并存储在 FIFO 1 中 – 111：存储在接收缓冲区中，忽略 EFT[1:0] 的配置
F0 位 28:0 EFID1[28:0]	扩展过滤器 ID 1 (Extended filter ID 1) 扩展 ID 过滤器元素的第一个 ID。 过滤接收缓冲区或调试消息时，此位字段定义要存储的扩展消息的 ID。接收的标识符必须完全匹配，仅使用 FDCAN_XIDAM 屏蔽机制。
F1 位 31:30 EFT[1:0]	扩展过滤器类型 (Extended filter type) – 00：从 EF1ID 到 EF2ID 的范围过滤器 (EF2ID ≥ EF1ID) – 01：用于 EF1ID 或 EF2ID 的双 ID 过滤器 – 10：典型过滤器：EF1ID = 过滤器，EF2ID = 掩码 – 11：从 EF1ID 到 EF2ID 的范围过滤器 (EF2ID ≥ EF1ID)，未应用 FDCAN_XIDAM 掩码

表 499. 扩展消息 ID 过滤器元素位字段说明（续）

位字段		说明
F1 位 28:0	EFID2[10:0]	扩展过滤器 ID 2 (Extended filter ID 2) 该位字段具有不同含义，具体视 EFEC 的配置而定： – SFEC = 001 ...110 扩展 ID 过滤器元素的第二个 ID。 – SFEC = 111：用于接收缓冲区或调试消息的过滤器
	EFID2[10:9]	确定接收的消息存储在接收缓冲区中，还是作为调试消息序列中的消息 A、B 或 C 进行处理。 – 00：将消息存储在接收缓冲区中 – 01：调试消息 A – 10：调试消息 B – 11：调试消息 C
	EFID2[8:6]	用于控制扩展接口的过滤器事件引脚。如果相应的位位置为 1，则当发生过滤器匹配时，会在相关过滤器事件引脚上产生脉冲，脉冲持续时间为一个 fdcan_pclk 周期。 EFID2[8] 接口由校准单元使用。
	EFID2[5:0]	定义与用于存储匹配消息的接收缓冲区起始地址 FDCAN_RXBC.RBSA 的偏移。

53.4.23 FDCAN 触发内存元素

最多可配置 64 个触发内存元素。访问触发内存元素时，其地址为触发内存起始地址 FDCAN_TTTMC.TMSA 加上触发内存元素的索引 (0 … 63)。

表 500. 触发内存元素

位	31 24	23 16	15 8	7 0
T0	TM[15:0]		Res. CC[6:0]	Res. TMIN TMEX TYPE[3:0]
T1	Res.	FTYPE MNR[6:0]	Res.	MSC[2:0]

表 501. 触发内存元素说明

位字段	说明
T0 位 31:16 TM[15:0]	时间标记 触发激活的周期时间。
T0 位 14:8 CC[6:0]	周期代码 触发有效的周期计数。触发类型为 Tx_Ref_Trigger、Tx_Ref_Trigger_Gap、Watch_Trigger、Watch_Trigger_Gap、End_of_List 时会忽略。 – 0b000000x 对所有周期有效 – 周期计数 mod2 = c 时，每 2 个周期，0b000001c 有效 – 周期计数 mod4 = cc 时，每 4 个周期，0b00001cc 有效 – 周期计数 mod8 = ccc 时，每 8 个周期，0b0001ccc 有效 – 周期计数 mod16 = cccc 时，每 16 个周期，0b001cccc 有效 – 周期计数 mod32 = ccccc 时，每 32 个周期，0b01ccccc 有效 – 周期计数 mod64 = cccccc 时，每 64 个周期，0b1cccccc 有效
T0 位 5 TMIN	内部时间标记事件 – 0：无操作 – 1：当触发内存元素激活时，FDCAN_TTIR.TTMI 会置 1

表 501. 触发内存元素说明（续）

位字段	说明
T0 位 4 TMEX	外部时间标记事件 – 0：无操作 – 1：当触发内存元素的时间标记激活且 FDCAN_TTOCN.TTMIE = 1 时，会在输出 fdcan1_tmp（多个实例时，仅一个实例时为 fdcan_tmp）上产生长度为一个周期的脉冲。
T0 位 3:0 TYPE[3:0]	触发类型 – 0000 Tx_Ref_Trigger——不处于间隙中时有效 – 0001 Tx_Ref_Trigger——处于间隙中时有效 – 0010 Tx_Trigger_Single——在独占时间窗口中开始单次发送 – 0011 Tx_Trigger_Continuous——在独占时间窗口中开始连续发送 – 0100 Tx_Trigger_Arbitration——在仲裁时间窗口中开始发送 – 0101 Tx_Trigger_Merged——启动合并仲裁窗口 – 0110 Watch_Trigger——不处于间隙中时有效 – 0111 Watch_Trigger_Gap——处于间隙中时有效 – 1000 Rx_Trigger——检查接收情况 – 1001 Time_Base_Trigger——仅控制 TMIN、TMEX – 1010 … 1111=End_of_List——非法类型，会导致配置错误
T1 位 23FTYPE	过滤器类型 – 0：11 位标准消息 ID – 1：29 位扩展消息 ID
T1 位 22:16 MNR[6:0](1)	消息编号 – 发送：触发对配置的发送缓冲区编号有效。有效值为 0 到 31。 – 接收：触发对标准/扩展消息 ID 过滤器元素编号有效。有效值分别为 0 到 63 和 0 到 127。
T1 位 2:0 MSC[2:0]	消息状态计数 计算独占时间窗口中周期性消息的调度错误数。不包含用于仲裁消息的功能和事件驱动 CAN 通信中的功能 (ISO11898-1)。 – 0-7 = 实际状态

1. 当 FDCAN 处于 INIT 状态时，必须写入触发内存元素。不允许对未处于 INIT 状态的触发内存元素进行写访问。但如果 TMIN 和 TMEX 被定义为 TYPE Tx_Ref_Trigger 触发内存元素的组成部分，则属于例外情况。在这种情况下，TMIN 和 TMEX 会在达到由实际参考触发偏移 (TTOST[RTO]) 修改的时间标记时激活。

53.5 FDCAN 寄存器

53.5.1 FDCAN 核心版本寄存器 (FDCAN_CREL)

FDCAN core release register

偏移地址：0x0000

复位值：0x3214 1218

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REL[3:0]				STEP[3:0]				SUBSTEP[3:0]				YEAR[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MON[7:0]								DAY[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **REL[3:0]**：核心版本 = 3

位 27:24 **STEP[3:0]**：核心版本主阶段号 = 2

位 23:20 **SUBSTEP[3:0]**：核心版本子阶段号 = 1

位 19:16 **YEAR[3:0]**：发布年份 = 4

位 15:8 **MON[7:0]**：发布月份 = 12

位 7:0 **DAY[7:0]**：发布日期 = 18

53.5.2 FDCAN 字节序寄存器 (FDCAN_ENDN)

FDCAN Endian register

偏移地址：0x0004

复位值：0x8765 4321

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETV[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETV[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ETV[31:0]**：字节序测试值 (Endianness test value)

字节序测试值为 0x8765 4321。

53.5.3 FDCAN 数据位定时和预分频器寄存器 (FDCAN_DBTP)

FDCAN data bit timing and prescaler register

偏移地址：0x000C

复位值：0x0000 0A33

该寄存器专用于数据位定时相位，仅当 FDCAN_CCCR.CCE 和 FDCAN_CCCR.INIT 置 1 时才可写入。CAN 时间片的可编程范围为 1 到 32 个 FDCAN 时钟周期。tq = (DBRP + 1) 个 FDCAN 时钟周期。

DTSEG1 为 Prop_Seg 与 Phase_Seg1 之和。DTSEG2 为 Phase_Seg2。
因此，位时间的长度为 (DTSEG1 + DTSEG2 + 3)（编程值）tq 或
(Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2) tq（函数值）。

信息处理时间 (IPT) 为零，这意味着下一位的数据在采样点后的第一个时钟边沿可用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDC	Res.	Res.	DBRP[4:0]				
								rw			rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DTSEG1[4:0]					DTSEG2[3:0]				DSJW[3:0]			
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23 **TDC**：收发器延迟补偿 (Transceiver delay compensation)
0：禁止收发器延迟补偿
1：使能收发器延迟补偿

位 22:21 保留，必须保持复位值。

位 20:16 **DBRP[4:0]**：数据比特率预分频器 (Data bitrate prescaler)
生成位时间片时将振荡器频率除以的值。位时间为该时间片的倍数。比特率预分频器的有效值为 0 到 31。硬件将该值解析为编程值加 1。

位 15:13 保留，必须保持复位值。

位 12:8 **DTSEG1[4:0]**：采样点之前的数据时间段 (Data time segment before sample point)
有效值为 0 到 31。硬件使用的值为编程值加 1，即 t_{BS1} = (DTSEG1 + 1) x tq。

位 7:4 **DTSEG2[3:0]**：采样点之后的数据时间段 (Data time segment after sample point)
有效值为 0 到 15。硬件使用的值为编程值加 1，即 t_{BS2} = (DTSEG2 + 1) x tq。

位 3:0 **DSJW[3:0]**：同步跳转宽度 (Synchronization jump width)
有效值为 0 到 15。硬件使用的值为编程值加 1，即 t_{SJW} = (DSJW + 1) x tq。

注： *如果 FDCAN 时钟为 8 MHz，复位值 0x00000A33 会将 FDCAN 的比特率配置为较快的 500 kb/s。*

注： *数据阶段比特率必须高于或等于标称比特率。*

53.5.4 FDCAN 测试寄存器 (FDCAN_TEST)

FDCAN test register

必须通过将 FDCAN_CCCR.TEST 位置 1 来使能对该寄存器进行写访问。FDCAN_CCCR.TEST 位复位时，所有寄存器恢复为复位值。

接收引脚 FDCANx_TX 的环回模式和软件控制属于硬件测试模式。将 TX 编程为 00 以外的值可能会干扰 CAN 总线上的消息传输。

偏移地址：0x0010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RX	TX[1:0]		LBCK	Res.	Res.	Res.	Res.
								r	rw	rw	rw				

位 31:8 保留，必须保持复位值。

位 7 **RX**：接收引脚 (Receive pin)

监视发送引脚 FDCANx_RX 的实际值

0：CAN 总线为显性 (FDCANx_RX = 0)

1：CAN 总线为隐性 (FDCANx_RX = 1)

位 6:5 **TX[1:0]**：发送引脚控制 (Control of transmit pin)

00：复位值，FDCANx_TX 由 CAN 内核控制，会在 CAN 位时间结束时更新

01：可在引脚 FDCANx_TX 监控采样点

10：引脚 FDCANx_TX 上为显性 (0) 电平

11：引脚 FDCANx_TX 上为隐性 (1) 电平

位 4 **LBCK**：环回模式 (Loop back mode)

0：复位值，禁止环回模式

1：使能环回模式（请参见*测试模式*）

位 3:0 保留，必须保持复位值。

53.5.5 FDCAN RAM 看门狗寄存器 (FDCAN_RWD)

FDCAN RAM watchdog register

RAM 看门狗监视消息 RAM 的 READY 输出。对消息 RAM 进行访问会启动消息 RAM 看门狗计数器，计数器的起始值由 FDCAN_RWD.WDC 位配置。

当消息 RAM 通过激活其 READY 输出指示访问成功完成时，计数器会重新载入 FDCAN_RWD.WDC 位的值。如果计数器递减为 0 之前消息 RAM 没有发出响应，则计数器会停止计数，中断标志 FDCAN_IR.WDI 位置 1。RAM 看门狗计数器由 fdcan_pclk 时钟驱动。

偏移地址：0x0014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDV[7:0]								WDC[7:0]							
r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:8 **WDV[7:0]**：看门狗值 (Watchdog value)
实际消息 RAM 看门狗计数器值。

位 7:0 **WDC[7:0]**：看门狗配置 (Watchdog configuration)
消息 RAM 看门狗计数器的起始值。如果使用复位值 00，计数器会禁止。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

53.5.6 FDCAN CC 控制寄存器 (FDCAN_CCCR)

FDCAN CC control register

偏移地址：0x0018

复位值：0x0000 0001

有关各个位置 1 和复位的详细信息，请参见*软件初始化*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NISO	TXP	EFBI	PXHD	Res.	Res.	BRSE	FDOE	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	r	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **NISO**：非 ISO 操作 (Non ISO operation)
如果该位置 1，FDCAN 会使用 Bosch CAN FD 规范 V1.0 规定的 CAN FD 帧格式。
0：符合 ISO11898-1 规定的 CAN FD 帧格式
1：符合 Bosch CAN FD 规范 V1.0 规定的 CAN FD 帧格式

位 14 **TXP**：
如果该位置 1，成功发送帧后，FDCAN 会先暂停两个 CAN 位时间，然后再开始进行下一次发送。
0：禁止
1：使能

位 13 **EFBI**：总线同步期间的边沿过滤 (Edge filtering during bus integration)
0：禁止边沿过滤
1：需要两个连续显性 tq 才能检测硬同步边沿

位 12 **PXHD**：协议异常处理禁止 (Protocol exception handling disable)
0：使能协议异常处理
1：禁止协议异常处理

位 11:10 保留，必须保持复位值。

位 9 **BRSE**：FDCAN 比特率切换 (FDCAN bitrate switching)

0：禁止发送时进行比特率切换

1：使能发送时进行比特率切换

位 8 **FDOE**：FD 操作使能 (FD operation enable)

0：禁止 FD 操作

1：使能 FD 操作

位 7 **TEST**：测试模式使能 (Test mode enable)

0：正常操作，寄存器 TEST 保存复位值

1：测试模式，使能对寄存器 TEST 的写访问

位 6 **DAR**：禁止自动重发送 (Disable automatic retransmission)

0：使能自动重发送未成功发送的消息

1：禁止自动重发送

位 5 **MON**：总线监控模式

仅当 CCE 和 INIT 均置 1 时，才能通过软件将 MON 位置 1。该位可随时由用户复位。

0：禁止总线监控模式

1：使能总线监控模式

位 4 **CSR**：时钟停止请求 (Clock stop request)

0：未请求时钟停止

1：已请求时钟停止。如果请求时钟停止，则在所有挂起的传输请求均已完成且 CAN 总线达到空闲状态之后，INIT 会先置 1，然后 CSA 会置 1。

位 3 **CSA**：时钟停止确认 (Clock stop acknowledge)

0：未确认时钟停止

1：可通过停止 APB 时钟和内核时钟将 FDCAN 设为掉电状态

位 2 **ASM**：ASM 受限工作模式 (ASM restricted operation mode)

受限工作模式可用于能够自行调整通信比特率的应用。应用会测试不同比特率，并在收到有效帧后退出受限工作模式。在可选的受限工作模式下，节点能够发送和接收数据帧和远程帧，并能对有效帧进行确认，但不会发送有效错误帧或过载帧。如果存在错误条件或过载条件，则不会发送显性位，而会等待出现总线空闲条件，以便使自身与 CAN 通信重新同步。错误计数器不会递增。仅当 CCE 和 INIT 均置 1 时，才能通过软件将 ASM 位置 1。该位可随时通过软件复位。当发送处理单元无法及时从消息 RAM 中读取数据时，该位会自动置 1。

如果 FDCAN 连接到 CAN 时钟校准单元，只要校准未完成，ASM 位就会由硬件置 1。

0：正常 CAN 操作

1：受限工作模式激活

位 1 **CCE**：配置更改使能 (Configuration change enable)

0：CPU 对受保护的配置寄存器没有写访问权限

1：CPU 对受保护的配置寄存器有写访问权限（当 FDCAN_CCCR.INIT = 1 时，如果 INIT 位清零，则 CCE 位会自动清零）

位 0 **INIT**：初始化

0：正常工作

1：开始初始化（当 FDCAN_CCCR.INIT = 1 时，如果 INIT 位清零，则 CCE 位自动清零）

注： *由于两个时钟域之间存在同步机制，可能会经过一定的延迟之后才能读取写入 INIT 的值。因此，程序员必须在修改 INIT 的值之前，通过读取 INIT 来确保之前写入的值已经生效。*

53.5.7 FDCAN 标称位定时和预分频器寄存器 (FDCAN_NBTP)

FDCAN nominal bit timing and prescaler register

偏移地址：0x001C

复位值：0x0600 0A03

该寄存器专用于在仲裁阶段使用的标称数据位定时，仅当 FDCAN_CCCR.CCE 和 FDCAN_CCCR.INIT 置 1 时才可写入。CAN 位时间的可编程范围为 4 到 81 个 tq。CAN 时间片的可编程范围为 1 到 1024 个 FDCAN 内核时钟周期。

tq = (BRP + 1) 个 FDCAN 时钟周期 fdcan_tq_ck

NTSEG1 为 Prop_Seg 与 Phase_Seg1 之和。NTSEG2 为 Phase_Seg2。因此，位时间的长度为（编程值）[NTSEG1 + NTSEG2 + 3] tq 或（函数值）[Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] tq。

信息处理时间 (IPT) 为零，这意味着下一位的数据在采样点后的第一个时钟边沿可用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NSJW[6:0]							NBRP[8:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NTSEG1[7:0]								Res.	NTSEG2[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:25 **NSJW[6:0]**：标称（重新）同步跳转宽度 (Nominal (re)synchronization jump width)
应小于 NTSEG2，有效值为 0 到 127。硬件使用的值为编程值加 1，即 t_{SJW} = (NSJW + 1) x tq。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 24:16 **NBRP[8:0]**：比特率预分频器 (Bitrate prescaler)
生成位时间片时将振荡器频率除以的值。位时间为该时间片的倍数。有效值为 0 到 511。硬件使用的值为编程值加 1。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 15:8 **NTSEG1[7:0]**：采样点之前的标称时间段 (Nominal time segment before sample point)
有效值为 0 到 255。硬件使用的值为编程值加 1 (t_{BS1} = (NTSEG1 + 1) x tq)。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 7 保留，必须保持复位值。

位 6:0 **NTSEG2[6:0]**：采样点之后的标称时间段 (Nominal time segment after sample point)
有效值为 0 到 127。硬件使用的值为编程值加 1 (t_{BS2} = (NTSEG2 + 1) x tq)。

注： *如果 CAN 内核时钟为 8 MHz，复位值 0x00000A33 会将 FDCAN 的比特率配置为 125 kb/s。*

53.5.8 FDCAN 时间戳计数器配置寄存器 (FDCAN_TSCC)

FDCAN timestamp counter configuration register

偏移地址：0x0020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCP[3:0]			
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSS[1:0]	
														rw	rw

位 31:20 保留，必须保持复位值。

位 19:16 **TCP[3:0]**：时间戳计数器预分频器 (Timestamp counter prescaler)

将时间戳和超时计数器时间单位配置为 CAN 位时间的倍数 [1 … 16]。

实际上，硬件会将该值解析为编程值加 1。

在 CAN FD 模式下，由于仲裁阶段与数据阶段之间的 CAN 位时间不同，内部时间戳计数器 TCP 不会提供恒定的时基。因此，CAN FD 需要使用外部计数器生成时间戳 (TSS = 10)。

这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 15:2 保留，必须保持复位值。

位 1:0 **TSS[1:0]**：时间戳选择 (Timestamp select)

00：时间戳计数器值始终为 0x0000

01：时间戳计数器值按 TCP 递增

10：使用 TIM3 提供的外部时间戳计数器值 (tim3_cnt[0:15])

11：与 00 相同。

这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

53.5.9 FDCAN 时间戳计数器值寄存器 (FDCAN_TSCV)

FDCAN timestamp counter value register

偏移地址：0x0024

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSC[15:0]															
rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w

位 31:16 保留，必须保持复位值。

位 15:0 **TSC[15:0]**：时间戳计数器 (Timestamp counter)

内部/外部时间戳计数器值是在（接收和发送）帧开始时捕获的。当 FDCAN_TSCC.TSS = 01 时，时间戳计数器会以 CAN 为时间的倍数 [1 ... 16] 递增，具体取决于 FDCAN_TSCC.TCP 的配置。回卷操作会将中断标志 FDCAN_IR.TSW 置 1。写访问会将计数器复位为 0。当 FDCAN_TSCC.TSS = 10 时，TSC 会反映外部时间戳计数器值。写访问没有影响。

注： *“回卷”是指时间戳计数器值不是由于对 FDCAN_TSCV 进行写访问的原因而由非零值变为 0。*

53.5.10 FDCAN 超时计数器配置寄存器 (FDCAN_TOCC)

FDCAN timeout counter configuration register

偏移地址：0x0028

复位值：0xFFFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TOP[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TOS[1:0]		ETOC
													rw	rw	rw

位 31:16 **TOP[15:0]**：超时时长 (Timeout period)

超时计数器（递减计数器）的起始值。配置超时时长。

这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 15:3 保留，必须保持复位值。

位 2:1 **TOS[1:0]**：超时选择 (Timeout select)

在连续模式下工作时，对 FDCAN_TOCV 进行写访问会将计数器预设为由 FDCAN_TOCC.TOP 配置的值并继续递减计数。超时计数器由其中一个 FIFO 控制时，空 FIFO 会将计数器预设为由 FDCAN_TOCC.TOP 配置的值。存储了第一个 FIFO 元素后，会开始递减计数。

00：连续工作

01：超时由发送事件 FIFO 控制

10：超时由接收 FIFO 0 控制

11：超时由接收 FIFO 1 控制

这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 0 **ETOC**：使能超时计数器

0：禁止超时计数器

1：使能超时计数器

该位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 1 (CCE) 和位 0 (INIT) 置 1 时，才能写入。

有关更多详细信息，请参见*超时计数器*。

53.5.11 FDCAN 超时计数器值寄存器 (FDCAN_TOCV)

FDCAN timeout counter value register

偏移地址：0x002C

复位值：0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOC[15:0]															
rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w

位 31:16 保留，必须保持复位值。

位 15:0 **TOC[15:0]**：超时计数器 (Timeout counter)

超时计数器会以 CAN 位时间的倍数 [1 ... 16] 递减，具体取决于 FDCAN_TSCC.TCP 的配置。当计数器递减至 0 时，中断标志 FDCAN_IR.TOO 会置 1，超时计数器会停止计数。开始和复位/重启条件是通过 FDCAN_TOCC.TOS 配置的。

53.5.12 FDCAN 错误计数器寄存器 (FDCAN_ECR)

FDCAN error counter register

偏移地址：0x0040

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CEL[7:0]							
								rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RP	REC[6:0]							TEC[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 保留，必须保持复位值。

位 23:16 **CEL[7:0]**：CAN 错误记录

每当 CAN 协议错误导致发送错误计数器或接收错误计数器递增时，计数器都会递增。对 CEL 进行读访问时，计数器会复位。计数器值达到 0xFF 时，会停止计数；TEC 或 REC 下一次递增时，中断标志 FDCAN_IR.ELO 会置 1。

位 15 **RP**：接收错误被动 (Receive error passive)

0：接收错误计数器低于错误被动级别 128

1：接收错误计数器已达到错误被动级别 128

位 14:8 **REC[6:0]**：接收错误计数器 (Receive error counter)

接收错误计数器的实际状态，取值范围为 0 到 127。

位 7:0 **TEC[7:0]**：发送错误计数器 (Transmit error counter)

发送错误计数器的实际状态，取值范围为 0 到 255。

如果 FDCAN_CCCR.ASM 置 1，则检测到 CAN 协议错误时，CAN 协议控制器不会使 TEC 和 REC 递增，但 CEL 仍会递增。

53.5.13 FDCAN 协议状态寄存器 (FDCAN_PSR)

FDCAN protocol status register

偏移地址：0x0044

复位值：0x0000 0707

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDCV[6:0]						
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PXE	REDL	RBRS	RESI	DLEC[2:0]			BO	EW	EP	ACT[1:0]		LEC[2:0]		
	rc_r	rc_r	rc_r	rc_r	rs_r	rs_r	rs_r	r	r	r	r	r	rs_r	rs_r	rs_r

位 31:23 保留，必须保持复位值。

位 22:16 **TDCV[6:0]**：发送器延迟补偿值 (Transmitter delay compensation value)
第二采样点的位置，由从 FDCAN_TX 到 FDCAN_RX 之间测得的延迟与 FDCAN_TDCR.TDCO 之和定义。在数据阶段，SSP 位置为已发送位起点与第二采样点之间的最小时间片 (mtq) 数。有效值为 0 到 127 个 mtq。

位 15 保留，必须保持复位值。

位 14 **PXE**：协议异常事件 (Protocol exception event)
0：自上次读访问起未发生协议异常事件
1：已发生协议异常事件

位 13 **REDL**：已接收 FDCAN 消息 (Received FDCAN message)
该位的设置与验收过滤无关。
0：由于该位是通过 CPU 复位的，因此尚未接收到 FDCAN 消息
1：已接收到 EDL 标志置 1 的 FDCAN 格式的消息
访问类型为 RX：进行读访问时复位。

位 12 **RBRS**：上次接收的 FDCAN 消息的 BRS 标志 (BRS flag of last received FDCAN message)
该位与 REDL 一起设置，与验收过滤无关。
0：上次接收的 FDCAN 消息的 BRS 标志未置 1
1：上次接收的 FDCAN 消息的 BRS 标志已置 1
访问类型为 RX：进行读访问时复位。

位 11 **RESI**：上次接收的 FDCAN 消息的 ESI 标志 (ESI flag of last received FDCAN message)
该位与 REDL 一起设置，与验收过滤无关。
0：上次接收的 FDCAN 消息的 ESI 标志未置 1
1：上次接收的 FDCAN 消息的 ESI 标志已置 1
访问类型为 RX：进行读访问时复位。

位 10:8 **DLEC[2:0]**：数据上一次错误代码 (Data last error code)
在 BRS 标志置 1 的 FDCAN 帧数据阶段发生的上一次错误类型。编码与 LEC 相同。当 BRS 标志置 1 的 FDCAN 帧已正确传输（接收或发送）时，该位字段将被清零。
访问类型为 RS：进行读访问时置 1。

位 7 **BO**：Bus_Off 状态 (Bus_Off status)
0：FDCAN 未处于 Bus_Off 状态
1：FDCAN 处于 Bus_Off 状态

位 6 **EW**：警告状态 (Warning status)
0：两个错误计数器的值均小于 Error_Warning 限值 96
1：至少有一个错误计数器已达到 Error_Warning 限值 96

位 5 **EP**：错误被动 (Error passive)
0：FDCAN 处于 Error_Active 状态。此位通常会参与总线通信，并会在检测到错误后发送主动错误标志
1：FDCAN 处于 Error_Passive 状态

位 4:3 **ACT[1:0]**：活动 (Activity)
监控 CAN 模块通信状态。
00：同步中：节点在 CAN 通信同步阶段
01：空闲：节点既不是接收器，也不是发送器
10：接收器：节点作为接收器工作
11：发送器：节点作为发送器工作

位 2:0 **LEC[2:0]**：上一个错误代码 (Last error code)

LEC 指示 CAN 总线上发生的上一个错误的类型。消息已正确传输后（接收或发送），该位字段将被清零。

000：无错误：自消息成功接收或发送而将 LEC 复位后，未发生任何错误。

001：内容错误：在已接收的消息中，有一部分连续出现 5 个以上的相等位，这种情况是不允许发生的。

010：格式错误：已接收帧的固定格式部分的格式不正确。

011：AckError：由 FDCAN 发送的消息未被另一节点确认。

100：Bit1Error：在消息发送过程中（仲裁字段例外），设备希望发送隐性电平（位逻辑值为 1），但受监控的总线值为显性。

101：Bit0Error：在消息发送过程中（或者确认位，或者主动错误标志，或者过载标志），设备希望发送显性电平（数据或标识符位逻辑值 0），但受监控的总线值为隐性。在 Bus_Off 恢复期间，每次监控到 11 个隐性位组成的序列时，此状态都会置 1。这样，CPU 便可监控 Bus_Off 恢复序列的进行（指示总线并未保持显性或持续受到干扰）。

110：CRCError：已接收消息的 CRC 校验和不正确。传入消息的 CRC 与通过已接收数据计算出的 CRC 不匹配。

111：NoChange：如果对协议状态寄存器进行读访问，则会将 LEC 重新初始化为 7。如果 LEC 显示的值为“7”，则自 CPU 上一次对协议状态寄存器进行读访问起，未检测到任何 CAN 总线事件。

访问类型为 RS：进行读访问时置 1。

注： *如果采用 FDCAN 格式的帧已到达 BRS 标志置 1 的数据阶段，下一 CAN 事件（错误或有效帧）将显示在 FLEC 中，而不显示在 LEC 中。FDCAN CRC 序列中的固定内容位中的错误将显示为格式错误，而不是内容错误。*

注： *无法通过将 FDCAN_CCCR.INIT 置 1 或复位来缩短 Bus_Off 恢复时序（请参见 CAN 规范第 2.0 版或 ISO11898-1）。如果设备进入 Bus_Off 状态，则会将其自身的 FDCAN_CCCR.INIT 置 1，从而会停止所有总线活动。一旦 FDCAN_CCCR.INIT 由 CPU 清零，设备随后将等待总线空闲状态出现 129 次（129 × 11 个连续隐性位），然后才会恢复正常操作。Bus_Off 恢复时序结束时，错误管理计数器将复位。在 FDCAN_CCCR.INIT 复位后的等待时间内，每次监控到由 11 个隐性位组成的时序时，Bit0 错误代码都会写入 FDCAN_PSR.LEC 中，从而使 CPU 能够准备好检查 CAN 总线保持显性还是持续受到干扰，并能够监控 Bus_Off 恢复时序。FDCAN_ECR.REC 用于对这些时序进行计数。*

53.5.14 FDCAN 发送延迟补偿寄存器 (FDCAN_TDCR)

FDCAN transmitter delay compensation register

偏移地址：0x0048

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDCO[6:0]							Res.	TDCF[6:0]						
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:15 保留，必须保持复位值。

位 14:8 **TDCO[6:0]**：发送器延迟补偿偏移 (Transmitter delay compensation offset)
定义从 FDCAN_TX 到 FDCAN_RX 测得的延迟与第二采样点之间距离的偏移值。有效值为 0 到 127 个 mtq。
这些位受写保护，也就是说，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入这些位。

位 7 保留，必须保持复位值。

位 6:0 **TDCF[6:0]**：发送器延迟补偿过滤器窗口长度 (Transmitter delay compensation filter window length)
定义 SSP 位置的最小值，要测量发送器延迟，会忽略 FDCAN_RX 上会导致 SSP 位置提前的显性边沿。
这些位受写保护，也就是说，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入这些位。

53.5.15 FDCAN 中断寄存器 (FDCAN_IR)

FDCAN interrupt register

如果检测到其中一个列出的条件（边沿有效），标志会置 1。必须由用户清零，否则将一直保持置位状态。通过向对应位位置写入 1 将标志清零。

写入 0 不会起任何作用。硬复位会将寄存器清零。IE 的配置控制着是否生成中断。ILS 的配置控制着是哪条中断线发出中断。

偏移地址：0x0050

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ARA	PED	PEA	WDI	BO	EW	EP	ELO	Res.	Res.	DRX	TOO	MRAF	TSW
		rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29 **ARA**：访问保留地址 (Access to reserved address)
0：未对保留地址进行访问
1：已对保留地址进行访问

位 28 **PED**：数据阶段的协议错误（使用数据位时间）(Protocol error in data phase (data bit time is used))
0：数据阶段没有协议错误
1：检测到数据阶段有协议错误（PSR.DLEC 不是 0、7）

位 27 **PEA**：仲裁阶段中的协议错误（使用标称位时间）(Protocol error in arbitration phase (nominal bit time is used))
0：仲裁阶段中没有协议错误
1：检测到仲裁阶段有协议错误（PSR.LEC 不是 0、7）

位 26 **WDI**：看门狗中断 (Watchdog interrupt)
0：未发生消息 RAM 看门狗事件
1：因 READY 缺失而发生消息 RAM 看门狗事件

位 25 **BO**：Bus_Off 状态 (Bus_Off status)
0：Bus_Off 状态未更改
1：Bus_Off 状态已更改

位 24 **EW**：警告状态 (Warning status)
0：Error_Warning 状态未更改
1：Error_Warning 状态已更改

位 23 **EP**：错误被动 (Error passive)
0：Error_Passive 状态未更改
1：Error_Passive 状态已更改

位 22 **ELO**：错误记录溢出 (Error logging overflow)
0：CAN 错误记录计数器未溢出
1：CAN 错误记录计数器已溢出

位 21:20 保留，必须保持复位值。

位 19 **DRX**：消息存储到专用接收缓冲区 (Message stored to dedicated Rx Buffer)
接收到的消息已存储到专用接收缓冲区后，此标志会置 1。
0：接收缓冲区无更新
1：至少有一条已接收消息存储到接收缓冲区中

位 18 **TOO**：发生超时 (Timeout occurred)
0：无超时
1：达到超时

位 17 **MRAF**：消息 RAM 访问失败 (Message RAM access failure)
如果接收处理单元在接收以下消息仲裁字段之前没有对已接收消息进行验收过滤或存储，
- 此标志会置 1。在这种情况下，验收过滤或消息存储会中止，接收处理单元会开始处理以下消息。
- 无法向消息 RAM 写入消息。在这种情况下，会中止消息存储。

在这两种情况下，FIFO 放入索引不会更新，专用接收缓冲区的新数据标志也不会置 1。下一条消息存储到该位置时，部分存储的消息会被覆盖。
当发送处理单元无法及时从消息 RAM 中读取数据时，此标志也会置 1。在这种情况下，消息发送会中止。如果发送处理单元访问失败，FDCAN 会切换到受限工作模式（请参见*受限工作模式*）。要退出受限工作模式，用户必须复位 FDCAN_CCCR.ASM。
0：未发生消息 RAM 访问失败
1：发生消息 RAM 访问失败

位 16 **TSW**：时间戳回卷 (Timestamp wraparound)
0：无时间戳计数器回卷
1：时间戳计数器回卷

位 15 **TEFL**：发送事件 FIFO 元素丢失 (Tx event FIFO element lost)
0：发送事件 FIFO 元素未丢失
1：发送事件 FIFO 元素已丢失，尝试向大小为 0 的发送事件 FIFO 进行写入时也会置 1。

位 14 **TEFF**：发送事件 FIFO 已满
0：发送事件 FIFO 未满
1：发送事件 FIFO 已满

位 13 **TEFW**：达到发送事件 FIFO 阈值 (Tx event FIFO watermark reached)
0：发送事件 FIFO 填充级别低于阈值
1：发送事件 FIFO 填充级别达到阈值

位 12 **TEFN**：发送事件 FIFO 新条目 (Tx event FIFO new entry)
0：发送事件 FIFO 未更改
1：发送处理单元写入发送事件 FIFO 元素

位 11 **TFE**：发送 FIFO 为空 (Tx FIFO empty)
0：发送 FIFO 非空
1：发送 FIFO 为空 (Tx FIFO empty)

位 10 **TCF**：发送取消完成 (Transmission cancellation finished)
0：发送取消未完成
1：发送取消完成 (Transmission cancellation finished)

位 9 **TC**：发送完成 (Transmission completed)
0：发送未完成
1：发送完成 (Transmission completed)

位 8 **HPM**：高优先级消息 (High priority message)
0：未接收到任何高优先级消息
1：接收到高优先级消息

位 7 **RF1L**：接收 FIFO 1 消息丢失 (Rx FIFO 1 message lost)
0：接收 FIFO 1 消息未丢失
1：接收 FIFO 1 消息已丢失，尝试向大小为 0 的接收 FIFO 1 进行写入后也会置 1

位 6 **RF1F**：接收 FIFO 1 已满 (Rx FIFO 1 full)
0：接收 FIFO 1 未满
1：接收 FIFO 1 已满 (Rx FIFO 1 full)

位 5 **RF1W**：达到接收 FIFO 1 阈值 (Rx FIFO 1 watermark reached)
0：接收 FIFO 1 填充级别低于阈值
1：接收 FIFO 1 填充级别达到阈值

位 4 **RF1N**：接收 FIFO 1 新消息 (Rx FIFO 1 new message)
0：未向接收 FIFO 1 写入新消息
1：已向接收 FIFO 1 写入新消息

位 3 **RF0L**：接收 FIFO 0 消息丢失 (Rx FIFO 0 message lost)
0：接收 FIFO 0 消息未丢失
1：接收 FIFO 0 消息已丢失，尝试向大小为 0 的接收 FIFO 0 进行写入后也会置 1

位 2 **RF0F**：接收 FIFO 0 已满 (Rx FIFO 0 full)
0：接收 FIFO 0 未满
1：接收 FIFO 0 已满 (Rx FIFO 0 full)

位 1 **RF0W**：达到接收 FIFO 0 阈值 (Rx FIFO 0 watermark reached)
0：接收 FIFO 0 填充级别低于阈值
1：接收 FIFO 0 填充级别达到阈值

位 0 **RF0N**：接收 FIFO 0 新消息 (Rx FIFO 0 New message)
0：未向接收 FIFO 0 写入新消息
1：已向接收 FIFO 0 写入新消息

53.5.16 FDCAN 中断使能寄存器 (FDCAN_IE)

FDCAN interrupt enable register

中断使能寄存器中的设置决定了哪些中断状态寄存器中的状态变化会在中断线上被标记。

偏移地址：0x0054

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ARAE	PEDE	PEAE	WDIE	BOE	EWE	EPE	ELOE	Res.	Res.	DRXE	TOOE	MRAFE	TSWE
		rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29 **ARAE**：访问保留地址使能 (Access to Reserved address enable)

位 28 **PEDE**：数据阶段的协议错误使能 (Protocol error in data phase enable)

位 27 **PEAE**：仲裁阶段中的协议错误使能 (Protocol error in Arbitration phase enable)

位 26 **WDIE**：看门狗中断使能 (Watchdog interrupt enable)

0：禁止中断

1：使能中断

位 25 **BOE**：Bus_Off 状态 (Bus_Off status)

0：禁止中断

1：使能中断

位 24 **EWE**：警告状态中断使能 (Warning status interrupt enable)

0：禁止中断

1：使能中断

位 23 **EPE**：错误被动中断使能 (Error passive interrupt enable)

0：禁止中断

1：使能中断

位 22 **ELOE**：错误记录溢出中断使能 (Error logging overflow interrupt enable)

0：禁止中断

1：使能中断

位 21:20 保留，必须保持复位值。

位 19 **DRXE**：消息存储到专用接收缓冲区中断使能 (Message stored to dedicated Rx buffer interrupt enable)

0：禁止中断

1：使能中断

位 18 **TOOE**：超时已发生中断使能 (Timeout occurred interrupt enable)

0：禁止中断

1：使能中断

位 17 **MRAFE**：消息 RAM 访问失败中断使能 (Message RAM access failure interrupt enable)

0：禁止中断

1：使能中断

位 16 **TSWE**：时间戳回卷中断使能 (Timestamp wraparound interrupt enable)
0：禁止中断
1：使能中断

位 15 **TEFLE**：发送事件 FIFO 元素丢失中断使能 (Tx event FIFO element lost interrupt enable)
0：禁止中断
1：使能中断

位 14 **TEFFE**：发送事件 FIFO 已满中断使能 (Tx event FIFO full interrupt enable)
0：禁止中断
1：使能中断

位 13 **TEFWE**：达到发送事件 FIFO 阈值中断使能 (Tx event FIFO watermark reached interrupt enable)
0：禁止中断
1：使能中断

位 12 **TEFNE**：发送事件 FIFO 新条目中断使能 (Tx event FIFO new entry interrupt enable)
0：禁止中断
1：使能中断

位 11 **TFEE**：Tx FIFO 为空时中断使能 (Tx FIFO empty interrupt enable)
0：禁止中断
1：使能中断

位 10 **TCFE**：发送取消完成中断使能 (Transmission cancellation finished interrupt enable)
0：禁止中断
1：使能中断

位 9 **TCE**：发送完成中断使能 (Transmission completed interrupt enable)
0：禁止中断
1：使能中断

位 8 **HPME**：高优先级消息中断使能 (High priority message interrupt enable)
0：禁止中断
1：使能中断

位 7 **RF1LE**：接收 FIFO 1 消息丢失中断使能 (Rx FIFO 1 message lost interrupt enable)
0：禁止中断
1：使能中断

位 6 **RF1FE**：接收 FIFO 1 已满中断使能 (Rx FIFO 1 full interrupt enable)
0：禁止中断
1：使能中断

位 5 **RF1WE**：达到接收 FIFO 1 阈值中断使能 (Rx FIFO 1 watermark reached interrupt enable)
0：禁止中断
1：使能中断

位 4 **RF1NE**：接收 FIFO 1 新消息中断使能 (Rx FIFO 1 new message interrupt enable)
0：禁止中断
1：使能中断

位 3 **RF0LE**：接收 FIFO 0 消息丢失中断使能 (Rx FIFO 0 message lost interrupt enable)
0：禁止中断
1：使能中断

位 2 **RF0FE**：接收 FIFO 0 已满中断使能 (Rx FIFO 0 full interrupt enable)
0：禁止中断
1：使能中断

位 1 **RF0WE**：达到接收 FIFO 0 阈值中断使能 (Rx FIFO 0 watermark reached interrupt enable)
0：禁止中断
1：使能中断

位 0 **RF0NE**：接收 FIFO 0 新消息中断使能 (Rx FIFO 0 new message interrupt enable)
0：禁止中断
1：使能中断

53.5.17 FDCAN 中断线选择寄存器 (FDCAN_ILS)

FDCAN interrupt line select register

该寄存器将来自中断寄存器的特定中断标志分配到两个模块中断线之一。要生成中断，必须通过 FDCAN_ILE.EINT0 和 FDCAN_ILE.EINT1 使能相应的中断线。

偏移地址：0x0058

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ARAL	PEDL	PEAL	WDIL	BOL	EWL	EPL	ELOL	Res.	Res.	DRXL	TOOL	MRAFL	TSWL
		rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29 **ARAL**：访问保留地址线

位 28 **PEDL**：数据阶段的协议错误线

位 27 **PEAL**：仲裁阶段的协议错误线

位 26 **WDIL**：看门狗中断线

位 25 **BOL**：Bus_Off 状态 (Bus_Off status)

位 24 **EWL**：警告状态中断线

位 23 **EPL**：错误被动中断线 (Error passive interrupt line)

位 22 **ELOL**：错误记录溢出中断线

位 21:20 保留，必须保持复位值。

位 19 **DRXL**：消息存储到专用接收缓冲区中断线 (Message stored to dedicated Rx buffer interrupt line)

位 18 **TOOL**：超时已发生中断线

位 17 **MRAFL**：消息 RAM 访问失败中断线

位 16 **TSWL**：时间戳回卷中断线

位 15 **TEFLL**：发送事件 FIFO 元素丢失中断线 (Tx event FIFO element Lost interrupt line)

位 14 **TEFFL**：发送事件 FIFO 已满中断线

位 13 **TEFWL**：达到发送事件 FIFO 阈值中断线 (Tx event FIFO watermark reached interrupt line)

位 12 **TEFNL**：发送事件 FIFO 新条目中断线

位 11 **TFEL**：发送 FIFO 为空中断线

位 10 **TCFL**：发送取消完成中断线 (Transmission cancellation finished interrupt line)

位 9 **TCL**：发送完成中断线 (Transmission completed interrupt line)

位 8 **HPML**：高优先级消息中断线 (High priority message interrupt line)

位 7 **RF1LL**：接收 FIFO 1 消息丢失中断线 (Rx FIFO 1 message lost interrupt line)

位 6 **RF1FL**：接收 FIFO 1 已满中断线 (Rx FIFO 1 full interrupt line)

位 5 **RF1WL**：达到接收 FIFO 1 阈值中断线 (Rx FIFO 1 watermark reached interrupt line)

位 4 **RF1NL**：接收 FIFO 1 新消息中断线 (Rx FIFO 1 new message interrupt line)

位 3 **RF0LL**：接收 FIFO 0 消息丢失中断线 (Rx FIFO 0 message lost interrupt line)

位 2 **RF0FL**：接收 FIFO 0 已满中断线 (Rx FIFO 0 full interrupt line)

位 1 **RF0WL**：达到接收 FIFO 0 阈值中断线 (Rx FIFO 0 watermark reached interrupt line)

位 0 **RF0NL**：接收 FIFO 0 新消息中断线

53.5.18 FDCAN 中断线使能寄存器 (FDCAN_ILE)

FDCAN interrupt line enable register

通过对位 EINT0 和 EINT1 进行编程，可分别使能/禁止两条连接至 CPU 的中断线。

偏移地址：0x005C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EINT1	EINT0
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **EINT1**：使能中断线 1 (Enable interrupt line 1)

0：禁止中断线 fdcan_intr1_it

1：使能中断线 fdcan_intr1_it

位 0 **EINT0**：使能中断线 0 (Enable interrupt line 0)

0：禁止中断线 fdcan_intr0_it

1：使能中断线 fdcan_intr0_it

53.5.19 FDCAN 全局过滤器配置寄存器 (FDCAN_GFC)

FDCAN global filter configuration register

用于消息 ID 过滤的全局设置。全局过滤器配置控制着标准和扩展消息的过滤器路径，请参见*图 670* 和*图 671*。

偏移地址：0x0080

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持复位值。

位 5:4 **ANFS[1:0]**：接受非匹配标准帧 (Accept non-matching frames standard)
定义了如何处理 11 位 ID 与过滤器表不匹配的报文。
00：在接收 FIFO 0 中接受
01：在接收 FIFO 1 中接受
10：拒绝
11：拒绝
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 3:2 **ANFE[1:0]**：接受非匹配扩展帧 (Accept non-matching frames extended)
定义了如何处理 29 位 ID 与过滤器表不匹配的报文。
00：在接收 FIFO 0 中接受
01：在接收 FIFO 1 中接受
10：拒绝
11：拒绝
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 1 **RRFS**：拒绝标准远程帧 (Reject remote frames standard)
0：过滤采用 11 位标准 ID 的远程帧
1：拒绝所有采用 11 位标准 ID 的远程帧
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 0 **RRFE**：拒绝扩展远程帧 (Reject remote frames extended)
0：过滤采用 29 位标准 ID 的远程帧
1：拒绝所有采用 29 位标准 ID 的远程帧
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

53.5.20 FDCAN 标准 ID 过滤器配置寄存器 (FDCAN_SIDFC)

FDCAN standard ID filter configuration register

用于 11 位标准 ID 过滤的设置。标准 ID 过滤器配置寄存器控制着标准消息的过滤器路径，请参见*图 670*。

偏移地址：0x0084

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSS[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLSSA[13:0]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31:24 保留，必须保持复位值。

位 23:16 **LSS[7:0]**：标准列表大小 (List size standard)

0：无标准消息 ID 过滤器

1-128：标准消息 ID 过滤器元素数量

>128：大于 128 的值会被解析为 128。

这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 15:2 **FLSSA[13:0]**：标准过滤器列表起始地址 (Filter list standard start address)

标准 ID 过滤器列表的起始地址（32 位字地址，请参见*表 496*）。这些位受写保护，也就是说，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入这些位。

位 1:0 保留，必须保持复位值。

53.5.21 FDCAN 扩展 ID 过滤器配置寄存器 (FDCAN_XIDFC)

FDCAN extended ID filter configuration register

用于 29 位扩展 ID 过滤的设置。FDCAN 扩展 ID 过滤器配置寄存器控制着标准消息的过滤器路径，请参见*图 671*。

偏移地址：0x0088

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSE[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLESA[13:0]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31:24 保留，必须保持复位值。

位 23:16 **LSE[7:0]**：扩展列表大小 (List size extended)
0：无扩展消息 ID 过滤器
1-128：扩展 ID 过滤器元素数量
>128：大于 128 的值会被解析为 128。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 15:2 **FLESA[13:0]**：过滤器列表扩展起始地址 (Filter list extended start address)
扩展消息 ID 过滤器列表的起始地址（32 位字地址，请参见*表 498：扩展消息 ID 过滤器元素*）。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 1:0 保留，必须保持复位值。

53.5.22 FDCAN 扩展 ID 和掩码寄存器 (FDCAN_XIDAM)

FDCAN extended ID and mask register

偏移地址：0x0090

复位值：0x1FFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	EIDM[28:16]												
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIDM[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:0 **EIDM[28:0]**：扩展 ID 掩码 (Extended ID Mask)
要对扩展帧进行验收过滤，会将扩展 ID 与掩码与已接收帧的 ID 进行与运算。用于屏蔽 SAE J1939 中的 29 位 ID。如果所有位的复位值均设为 1，则掩码无效。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

53.5.23 FDCAN 高优先级消息状态寄存器 (FDCAN_HPMS)

FDCAN high priority message status register

每次消息 ID 过滤器元素配置为生成优先级事件匹配时，都会更新此寄存器。此寄存器可用于监控传入高优先级消息的状态，并可使能对这些消息的快速访问。

偏移地址：0x0094

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLST	FIDX[6:0]							MSI[1:0]		BIDX[5:0]					
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15 **FLST**：过滤器列表 (Filter list)
指示匹配过滤器元素的过滤器列表。
0：标准过滤器列表
1：扩展过滤器列表

位 14:8 **FIDX[6:0]**：过滤器索引 (Filter index)
匹配过滤器元素的索引。范围为 0 到 FDCAN_SIDFC[LSS] - 1 或 FDCAN_XIDFC[LSE] - 1。

位 7:6 **MSI[1:0]**：消息存储指示符 (Message storage indicator)
00：未选择 FIFO
01：FIFO 上溢
10：消息存储在 FIFO 0 中
11：消息存储在 FIFO 1 中

位 5:0 **BIDX[5:0]**：缓冲区索引 (Buffer index)
消息存储到的接收 FIFO 元素的索引。仅当 MSI[1] = 1 时有效。

53.5.24 FDCAN 新数据 1 寄存器 (FDCAN_NDAT1)

FDCAN new data 1 register

偏移地址：0x0098

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **ND[31:0]**：新数据 [31:0] (New data[31:0])

该寄存器保存接收缓冲区 0 到 31 的新数据标志。当相应的接收缓冲区通过已接收的帧进行更新时，这些标志会置 1。必须由用户清零，否则将一直保持置位。通过向对应位位置写入 1 将标志清零。写入 0 不会起任何作用。硬复位会将寄存器清零。

0：接收缓冲区无更新

1：接收缓冲区有新消息更新

53.5.25 FDCAN 新数据 2 寄存器 (FDCAN_NDAT2)

FDCAN new data 2 register

偏移地址：0x009C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **ND[63:32]**：新数据 [63:32] (New data[63:32])

该寄存器保存接收缓冲区 32 到 63 的新数据标志。当相应的接收缓冲区通过已接收的帧进行更新时，这些标志会置 1。必须由用户清零，否则将一直保持置位。通过向对应位位置写入 1 将标志清零。写入 0 不会起任何作用。硬复位会将寄存器清零。

0：接收缓冲区无更新

1：接收缓冲区有新消息更新

53.5.26 FDCAN 接收 FIFO 0 配置寄存器 (FDCAN_RXF0C)

FDCAN Rx FIFO 0 configuration register

偏移地址：0x00A0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
F0OM	F0WM[6:0]							Res.	F0S[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F0SA[13:0]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31 **F0OM**：FIFO 0 工作模式 (FIFO 0 Operation mode)

FIFO 0 可工作在阻止模式或覆盖模式下。

0：FIFO 0 处于阻止模式

1：FIFO 0 处于覆盖模式

位 30:24 **F0WM[6:0]**：FIFO 0 阈值 (FIFO 0 watermark)
0：禁止阈值中断
1-64：接收 FIFO 0 阈值中断 (FDCAN_IR.RF0W) 的级别
>64：禁止阈值中断
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 23 保留，必须保持复位值。

位 22:16 **F0S[6:0]**：接收 FIFO 0 大小 (Rx FIFO 0 size)
0：无接收 FIFO 0
1-64：接收 FIFO 0 元素数
>64：大于 64 的值会被解析为 64
接收 FIFO 0 元素的索引为 0 到 F0S-1。

位 15:2 **F0SA[13:0]**：接收 FIFO 0 起始地址 (Rx FIFO 0 start address)
消息 RAM 中接收 FIFO 0 的起始地址（32 位字地址，请参见*图 669*）。

位 1:0 保留，必须保持复位值。

53.5.27 FDCAN 接收 FIFO 0 状态寄存器 (FDCAN_RXF0S)

FDCAN Rx FIFO 0 status register

偏移地址：0x00A4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	RF0L	F0F	Res.	Res.	F0PI[5:0]					
						rw	rw			rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	F0GI[5:0]						Res.	F0FL[6:0]						
		rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25 **RF0L**：接收 FIFO 0 消息丢失 (Rx FIFO 0 message lost)
该位是中断标志 FDCAN_IR.RF0L 的副本。当 FDCAN_IR.RF0L 复位时，该位也会复位。
0：接收 FIFO 0 消息未丢失
1：接收 FIFO 0 消息已丢失，尝试向大小为 0 的接收 FIFO 0 进行写入后也会置 1

位 24 **F0F**：接收 FIFO 0 已满 (Rx FIFO 0 full)
0：接收 FIFO 0 未满
1：接收 FIFO 0 已满 (Rx FIFO 0 full)

位 23:22 保留，必须保持复位值。

位 21:16 **F0PI[5:0]**：接收 FIFO 0 放入索引 (Rx FIFO 0 put index)
接收 FIFO 0 写入索引指针，范围为 0 到 63。

位 15:14 保留，必须保持复位值。

位 13:8 **F0GI[5:0]**：接收 FIFO 0 获取索引 (Rx FIFO 0 get index)
接收 FIFO 0 读取索引指针，范围为 0 到 63。

位 7 保留，必须保持复位值。

位 6:0 **F0FL**[6:0]：Rx FIFO 0 填充级别 (Rx FIFO 0 fill level)
接收 FIFO 0 中存储的元素数，范围为 0 到 64。

53.5.28 FDCAN 接收 FIFO 0 确认寄存器 (FDCAN_RXF0A)

FDCAN Rx FIFO 0 acknowledge register

偏移地址：0x00A8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F0AI[5:0]					
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持复位值。

位 5:0 **F0AI[5:0]**：接收 FIFO 0 确认索引 (Rx FIFO 0 acknowledge index)
用户从接收 FIFO 0 读取消息或消息序列后，必须将从接收 FIFO 0 读取的最后一个元素的缓冲区索引写入 F0AI 中。此操作会将接收 FIFO 0 获取索引 FDCAN_RXF0S.F0GI 设为 F0AI + 1，并会更新 FIFO 0 填充级别 FDCAN_RXF0S.F0FL。

53.5.29 FDCAN 接收缓冲区配置寄存器 (FDCAN_RXBC)

FDCAN Rx buffer configuration register

偏移地址：0x00AC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBSA[13:0]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31:16 保留，必须保持复位值。

位 15:2 **RBSA[13:0]**：接收缓冲区起始地址 (Rx buffer start address)
配置消息 RAM 中接收缓冲区部分的起始地址（32 位字地址）。此外，也用于引用调试消息 A、B、C。这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 1:0 保留，必须保持复位值。

53.5.30 FDCAN 接收 FIFO 1 配置寄存器 (FDCAN_RXF1C)

FDCAN Rx FIFO 1 configuration register

偏移地址：0x00B0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
F1OM	F1WM[6:0]							Res.	F1S[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F1SA[13:0]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31 **F1OM**：FIFO 1 工作模式 (FIFO 1 operation mode)
FIFO 1 可工作在阻止模式或覆盖模式下。
0：FIFO 1 处于阻止模式
1：FIFO 1 处于覆盖模式

位 30:24 **F1WM[6:0]**：接收 FIFO 1 阈值 (Rx FIFO 1 watermark)
0：禁止阈值中断
1-64：接收 FIFO 1 阈值中断 (FDCAN_IR.RF1W) 的级别
>64：禁止阈值中断。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 23 保留，必须保持复位值。

位 22:16 **F1S[6:0]**：接收 FIFO 1 大小 (Rx FIFO 1 size)
0：无接收 FIFO 1
1-64: 接收 FIFO 1 元素数
>64: 大于 64 的值会被解析为 64
接收 FIFO 1 元素的索引为 0 到 F1S - 1。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 15:2 **F1SA[13:0]**：接收 FIFO 1 起始地址 (Rx FIFO 1 start address)
消息 RAM 中接收 FIFO 1 的起始地址（32 位字地址，请参见*图 669*）。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 1:0 保留，必须保持复位值。

53.5.31 FDCAN 接收 FIFO 1 状态寄存器 (FDCAN_RXF1S)

FDCAN Rx FIFO 1 status register

偏移地址：0x00B4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMS[1:0]		Res.	Res.	Res.	Res.	RF1L	F1F	Res.	Res.	F1PI[5:0]					
r	r					r	r			r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	F1GI[5:0]						Res.	F1FL[6:0]						
		r	r	r	r	r	r		r	r	r	r	r	r	r

位 31:30 **DMS[1:0]**：调试消息状态 (Debug message status)
00：空闲状态，等待接收调试消息
01：已接收调试消息 A
10：已接收调试消息 A、B
11：已接收调试消息 A、B、C

位 29:26 保留，必须保持复位值。

位 25 **RF1L**：接收 FIFO 1 消息丢失 (Rx FIFO 1 message lost)
该位是中断标志 FDCAN_IR.RF1L 的副本。当 FDCAN_IR.RF1L 复位时，该位也会复位。
0：接收 FIFO 1 消息未丢失
1：接收 FIFO 1 消息已丢失，尝试向大小为 0 的接收 FIFO 1 进行写入后也会置 1。

位 24 **F1F**：接收 FIFO 1 已满 (Rx FIFO 1 full)
0：接收 FIFO 1 未满
1：接收 FIFO 1 已满 (Rx FIFO 1 full)

位 23:22 保留，必须保持复位值。

位 21:16 **F1PI[5:0]**：接收 FIFO 1 放入索引 (Rx FIFO 1 put index)
接收 FIFO 1 写入索引指针，范围为 0 到 63。

位 15:14 保留，必须保持复位值。

位 13:8 **F1GI[5:0]**：接收 FIFO 1 获取索引 (Rx FIFO 1 get index)
接收 FIFO 1 读取索引指针，范围为 0 到 63。

位 7 保留，必须保持复位值。

位 6:0 **F1FL[6:0]**：Rx FIFO 1 填充级别 (Rx FIFO 1 fill level)
接收 FIFO 1 中存储的元素数，范围为 0 到 64

53.5.32 FDCAN 接收 FIFO 1 确认寄存器 (FDCAN_RXF1A)

FDCAN Rx FIFO 1 acknowledge register

偏移地址：0x00B8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F1AI[5:0]					
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持复位值。

位 5:0 **F1AI[5:0]**：接收 FIFO 1 确认索引 (Rx FIFO 1 acknowledge index)

用户从接收 FIFO 1 读取消息或消息序列后，必须将从接收 FIFO 1 读取的最后一个元素的缓冲区索引写入 F1AI 中。此操作会将接收 FIFO 1 获取索引 FDCAN_RXF1S.F1GI 设为 F1AI + 1，并会更新 FIFO 1 填充级别 FDCAN_RXF1S.F1FL。

53.5.33 FDCAN 接收缓冲区元素大小配置寄存器 (FDCAN_RXESC)

FDCAN Rx buffer element size configuration register

配置属于接收缓冲区/接收 FIFO 元素的数据字节数。如果数据字段大小大于 8 个字节，则该数据字段仅可用于 CAN FD 操作。

偏移地址：0x00BC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	RBDS[2:0]			Res.	F1DS[2:0]			Res.	F0DS[2:0]		
					r	r	r		r	r	r		r	r	r

位 31:11 保留，必须保持复位值。

位 10:8 **RBDS[2:0]**：接收缓冲区数据字段大小 (Rx buffer data field size)

000：8 字节数据字段
001：12 字节数据字段
010：16 字节数据字段
011：20 字节数据字段
100：24 字节数据字段
101：32 字节数据字段
110：48 字节数据字段
111：64 字节数据字段

位 7 保留，必须保持复位值。

位 6:4 **F1DS[2:0]**：接收 FIFO 0 数据字段大小 (Rx FIFO 0 data field size)
000：8 字节数据字段
001：12 字节数据字段
010：16 字节数据字段
011：20 字节数据字段
100：24 字节数据字段
101：32 字节数据字段
110：48 字节数据字段
111：64 字节数据字段

位 3 保留，必须保持复位值。

位 2:0 **F0DS[2:0]**：接收 FIFO 1 数据字段大小 (Rx FIFO 1 data field size)
000：8 字节数据字段
001：12 字节数据字段
010：16 字节数据字段
011：20 字节数据字段
100：24 字节数据字段
101：32 字节数据字段
110：48 字节数据字段
111：64 字节数据字段

53.5.34 FDCAN 发送缓冲区配置寄存器 (FDCAN_TXBC)

FDCAN Tx buffer configuration register

偏移地址：0x00C0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TFQM	TFQS[5:0]						Res.	Res.	NDTB[5:0]					
	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBSA[13:0]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31 保留，必须保持复位值。

位 30 **TFQM**：发送 FIFO/队列模式 (Tx FIFO/queue mode)
0：发送 FIFO 操作
1：发送队列操作
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 29:24 **TFQS[5:0]**：发送 FIFO/队列大小 (Transmit FIFO/queue size)。
0：无发送 FIFO/队列
1-32：用于发送 FIFO/队列的发送缓冲区数
>32：大于 32 的值会被解析为 32。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 23:22 保留，必须保持复位值。

位 21:16 **NDTB[5:0]**：专用发送缓冲区数 (Number of dedicated transmit buffers)
0：无专用发送缓冲区
1-32：专用发送缓冲区数
>32：大于 32 的值会被解析为 32。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 15:2 **TBSA[13:0]**：发送缓冲区起始地址 (Tx buffers start address)
消息 RAM 中发送缓冲区部分的起始地址（32 位字地址，请参见*图 669*）。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 1:0 保留，必须保持复位值。

注： *TFQS 与 NDTB 之和不能大于 32。不会检查配置是否存在错误。消息 RAM 中的发送缓冲区部分从专用发送缓冲区开始。*

53.5.35 FDCAN 发送 FIFO/队列状态寄存器 (FDCAN_TXFQS)

FDCAN Tx FIFO/queue status register

发送 FIFO/队列状态与寄存器 FDCAN_TXBRP 中列出的挂起发送请求相关。因此，添加/取消请求的作用可能因正在运行发送扫描而延迟（FDCAN_TXBRP 尚未更新）。

偏移地址：0x00C4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TFQF	TFQPI[4:0]				
										r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TFGI[4:0]					Res.	Res.	TFFL[5:0]					
			r	r	r	r	r			r	r	r	r	r	r

位 31:22 保留，必须保持复位值。

位 21 **TFQF**：发送 FIFO/队列已满 (Tx FIFO/queue full)
0：发送 FIFO/队列未满
1：发送 FIFO/队列已满

位 20:16 **TFQPI[4:0]**：发送 FIFO/队列放入索引 (Tx FIFO/queue put index)
发送 FIFO/队列写入索引指针，范围为 0 到 31

位 15:13 保留，必须保持复位值。

位 12:8 **TFGI[4:0]**：
发送 FIFO 获取索引 (Tx FIFO get index)。
发送 FIFO 读取索引指针，范围为 0 到 31。如果已配置发送队列操作 (FDCAN_TXBC.TFQM = 1)，则读为 0

位 7:6 保留，必须保持复位值。

位 5:0 **TFFL[5:0]**：发送 FIFO 空闲级别 (Tx FIFO free level)
从 TFGI 开始的连续空闲发送 FIFO 元素数，范围为 0 到 32。如果已配置发送队列操作 (FDCAN_TXBC.TFQM = 1)，则读为 0

注：　如果是专用发送缓冲区与发送 FIFO 或发送队列相结合的混合配置，放入和取出索引指示从第一个专用发送缓冲区开始的发送缓冲区数。例如：对于 12 个专用发送缓冲区与包含 20 个缓冲区的发送 FIFO 相结合的混合配置，放入索引 15 会指向发送 FIFO 的第四个缓冲区。

53.5.36 FDCAN 发送缓冲区元素大小配置寄存器 (FDCAN_TXESC)

FDCAN Tx buffer element size configuration register

配置属于发送缓冲区元素的数据字节数。如果数据字段大小大于 8 个字节，则该数据字段仅可用于 CAN FD 操作。

偏移地址：0x00C8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TBDS[2:0]		
													r	r	r

位 31:3 保留，必须保持复位值。

位 2:0 **TBDS[2:0]**：发送缓冲区数据字段大小 (Tx buffer data Field size)：
000：8 字节数据字段
001：12 字节数据字段
010：16 字节数据字段
011：20 字节数据字段
100：24 字节数据字段
101：32 字节数据字段
110：48 字节数据字段
111：64 字节数据字段

53.5.37 FDCAN 发送缓冲区请求挂起寄存器 (FDCAN_TXBRP)

FDCAN Tx buffer request pending register

偏移地址：0x00CC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRP[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRP[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TRP[31:0]**：发送请求挂起 (Transmission request pending)。

每个发送缓冲区都有自己的发送请求挂起位。这些位通过寄存器 FDCAN_TXBAR 置 1。请求的发送已完成或已通过寄存器 FDCAN_TXBCR 取消后，这些位会复位。

仅会为通过 FDCAN_TXBC 配置的发送缓冲区设置 FDCAN_TXBRP 位。FDCAN_TXBRP 位置 1 后，会启动发送扫描（请参见*过滤调试消息*）以检查优先级最高（消息 ID 最小的发送缓冲区）的挂起发送请求。

取消请求会使寄存器 FDCAN_TXBRP 的相应发送请求挂起位复位。如果请求取消时发送已开始进行，则无论发送是否成功，都会在发送结束时执行复位操作。相应的 FDCAN_TXBRP 位复位后，取消请求位也会立即复位。

在发出取消请求后，成功发送后通过 FDCAN_TXBCF 信号完成取消，

并与相应的 FDCAN_TXBTO 位一起发送

取消时发送尚未开始

发送因仲裁丢失而中止

帧发送过程中出错

在 DAR 模式下，所有发送在发送失败后都会自动取消。对于所有未成功的发送，相应的 FDCAN_TXBCF 位会置 1。

0：无发送请求挂起

1：发送请求挂起

注： *如果 FDCAN_TXBRP 位在正在进行发送扫描时置 1，则进行该特殊发送扫描期间不会考虑该位。如果请求取消此类发送缓冲区，则该添加请求会立即取消，相应的 TXBRP 位会复位。*

53.5.38 FDCAN 发送缓冲区添加请求寄存器 (FDCAN_TXBAR)

FDCAN Tx buffer add request register

偏移地址：0x00D0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **AR[31:0]**：添加请求 (Add request)

每个发送缓冲区都有自己的添加请求位。写入 1 会将相应的添加请求位置 1；写入 0 没有影响。这样，用户通过对 FDCAN_TXBAR 进行一次写操作便可为多个发送缓冲区设置发送请求。仅会为通过 FDCAN_TXBC 配置的发送缓冲区设置 FDCAN_TXBAR 位。如果没有运行发送扫描，这些位会立即复位，否则在发送扫描过程完成之前，这些位保持置 1 状态。

0：未添加发送请求

1：已添加发送请求。

注： *如果添加请求应用到具有挂起发送请求的发送缓冲区（相应的 FDCAN_TXBRP 位已置 1），则会忽略请求。*

53.5.39 FDCAN 发送缓冲区取消请求寄存器 (FDCAN_TXBCR)

FDCAN Tx buffer cancellation request register

偏移地址：0x00D4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CR[31:0]**：取消请求 (Cancellation request)

每个发送缓冲区都有自己的取消请求位。写入 1 会将相应的取消请求位置 1；写入 0 没有影响。这样，用户通过对 FDCAN_TXBCR 进行一次写操作便可为多个发送缓冲区设置取消请求。仅会为通过 FDCAN_TXBC 配置的发送缓冲区设置 FDCAN_TXBCR 位。在相应的 FDCAN_TXBRP 位复位之前，这些位保持置 1 状态。

0：无取消挂起

1：取消挂起

53.5.40 FDCAN 发送缓冲区发送已发生寄存器 (FDCAN_TXBTO)

FDCAN Tx buffer transmission occurred register

偏移地址：0x00D8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TO[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TO[31:0]**：已进行发送 (Transmission occurred)

每个发送缓冲区都有自己的发送已发生位。当相应的 FDCAN_TXBRP 位在发送成功后清零时，这些位会置 1。发出向寄存器 FDCAN_TXBAR 的相应位写入 1 请求进行新发送时，这些位会复位。

0：未进行发送

1：已进行发送

53.5.41 FDCAN 发送缓冲区取消完成寄存器 (FDCAN_TXBCF)

FDCAN Tx buffer cancellation finished register

偏移地址：0x00DC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CF[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CF[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CF[31:0]**：取消完成 (Cancellation finished)

每个发送缓冲区都有自己的取消完成位。当相应的 FDCAN_TXBRP 位在通过 FDCAN_TXBCR 请求取消后清零时，这些位会置 1。如果相应的 FDCAN_TXBRP 位在取消时未置 1，CF 会立即置 1。发出向寄存器 FDCAN_TXBAR 的相应位写入 1 请求进行新发送时，这些位会复位。

0：无发送缓冲区取消

1：发送缓冲区取消完成

53.5.42 FDCAN 发送缓冲区发送中断使能寄存器 (FDCAN_TXBTIE)

FDCAN Tx buffer transmission interrupt enable register

偏移地址：0x00E0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIE[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **TIE[31:0]**：发送中断使能 (Transmission interrupt enable)

每个发送缓冲区都有自己的发送中断使能位。

0：禁止发送中断

1：发送中断使能

53.5.43 FDCAN 发送缓冲区取消完成中断使能寄存器 (FDCAN_TXBCIE)

FDCAN Tx buffer cancellation finished interrupt enable register

偏移地址：0x00E4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CFIE[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CFIE[31:0]**：取消完成中断使能 (Cancellation finished interrupt enable)
每个发送缓冲区都有自己的取消完成中断使能位。
0：禁止取消完成中断
1：使能取消完成中断

53.5.44 FDCAN 发送事件 FIFO 配置寄存器 (FDCAN_TXEFC)

FDCAN Tx event FIFO configuration register

偏移地址：0x00F0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	EFWM[5:0]						Res.	Res.	EFS[5:0]					
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EFSA[13:0]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31:30 保留，必须保持复位值。

位 29:24 **EFWM[5:0]**：事件 FIFO 阈值 (Event FIFO watermark)
0：禁止阈值中断
1-32：发送事件 FIFO 阈值中断 (FDCAN_IR.TEFW) 的级别
>32：禁止阈值中断
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 23:22 保留，必须保持复位值。

位 21:16 **EFS[5:0]**：事件 FIFO 大小 (Event FIFO size)
0：禁止发送事件 FIFO
1-32：发送事件 FIFO 元素数
>32：大于 32 的值会被解析为 32
发送事件 FIFO 元素的索引为 0 到 EFS-1。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 15:2 **EFSA[13:0]**：事件 FIFO 起始地址 (Event FIFO start address)
消息 RAM 中发送事件 FIFO 的起始地址（32 位字地址，请参见*图 669*）。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 1:0 保留，必须保持复位值。

53.5.45 FDCAN 发送事件 FIFO 状态寄存器 (FDCAN_TXEFS)

FDCAN Tx event FIFO status register

偏移地址：0x00F4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	TEFL	EFF	Res.	Res.	Res.	EFPI[4:0]				
						r	r				r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	EFGI[4:0]					Res.	Res.	EFFL[5:0]					
			r	r	r	r	r			r	r	r	r	r	r

位 31:26 保留，必须保持复位值。

位 25 **TEFL**：发送事件 FIFO 元素丢失 (Tx event FIFO element lost)
该位是中断标志 FDCAN_IR.TEFL 的副本。当 FDCAN_IR.TEFL 复位时，该位也会复位。
0：发送事件 FIFO 元素未丢失
1：发送事件 FIFO 元素已丢失，尝试向大小为 0 的发送事件 FIFO 进行写入时也会置 1。

位 24 **EFF**：事件 FIFO 已满 (Event FIFO full)
0：发送事件 FIFO 未满
1：发送事件 FIFO 已满

位 23:21 保留，必须保持复位值。

位 20:16 **EFPI[4:0]**：事件 FIFO 放入索引 (Event FIFO put index)
发送事件 FIFO 写入索引指针，范围为 0 到 31。

位 15:13 保留，必须保持复位值。

位 12:8 **EFGI[4:0]**：事件 FIFO 获取索引 (Event FIFO get index)
发送事件 FIFO 读取索引指针，范围为 0 到 31。

位 7:6 保留，必须保持复位值。

位 5:0 **EFFL[5:0]**：事件 FIFO 填充级别 (Event FIFO fill level)
发送事件 FIFO 中存储的元素数，范围为 0 到 31。

53.5.46 FDCAN 发送事件 FIFO 确认寄存器 (FDCAN_TXEFA)

FDCAN Tx event FIFO acknowledge register

偏移地址：0x00F8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EFAI[4:0]				
											rw	rw	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4:0 **EFAI[4:0]**：事件 FIFO 确认索引 (Event FIFO acknowledge index)

用户从发送事件 FIFO 读取元素或元素序列后，必须将从发送事件 FIFO 读取的最后一个元素的索引写入 EFAI 中。此操作会将发送事件 FIFO 获取索引 FDCAN_TXEFS.EFGI 设为 EFAI + 1，并会更新 FIFO 0 填充级别 FDCAN_TXEFS.EFFL。

53.5.47 FDCAN 寄存器映射

表 502. FDCAN 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	FDCAN_CREL	REL[3:0]				STEP[3:0]				SUBSTEP [3:0]				YEAR[3:0]				MON[7:0]								DAY[7:0]							
	Reset value	0	0	1	1	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0	0	0
0x0004	FDCAN_ENDN	ETV[31:0]																															
	Reset value	1	0	0	0	0	1	1	1	0	1	1	0	0	1	1	1	0	1	0	0	0	0	1	1	0	0	1	0	0	0	0	1
0x0008	Reserved	Reserved																															
0x000C	FDCAN_DBTP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDC	Res.	Res.	DBRP[4:0]					Res.	Res.	Res.	DTSEG1[4:0]					DTSEG2 [3:0]				DSJW[3:0]			
	Reset value									0			0	0	0	0	0				0	1	0	1	0	0	0	1	1	0	0	1	1
0x0010	FDCAN_TEST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RX	TX [1:0]		LBCK	Res.	Res.	Res.	Res.
	Reset value																									0	0	0	0				
0x0014	FDCAN_RWD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDV[7:0]								WDC[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018	FDCAN_CCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NISO	TXP	EFBI	PXHD	Res.	Res.	BRSE	FDOE	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
	Reset value																	0	0	0	0			0	0	0	0	0	0	0	0	0	1
0x001C	FDCAN_NBTP	NSJW[6:0]							NBRP[8:0]									NTSEG1[7:0]								Res.	NTSEG2[6:0]							
	Reset value	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0		0	0	0	0	0	1	1

表 502. FDCAN 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0020	FDCAN_TSCC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCP[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSS [1:0]	
	Reset value													0	0	0	0															0	0
0x0024	FDCAN_TSCV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0028	FDCAN_TOCC	TOP[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TOS [1:0]	ETOC
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1														0	0	0
0x002C	FDCAN_TOCV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TOC[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0030-0x003F	Reserved	Reserved																															
0x0040	FDCAN_ECR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CEL[7:0]								RP	REC[6:0]							TEC[7:0]							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	FDCAN_PSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PXE	REDL	RBRS	RESI	DLEC [2:0]			BO	EW	EP	ACT [1:0]		LEC[2:0]		
	Reset value																		0	0	0	0	1	1	1	0	0	0	0	0	1	1	1
0x0048	FDCAN_TDCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDCO[6:0]							Res.	TDCF[6:0]						
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0050	FDCAN_IR	Res.	Res.	ARA	PED	PEA	WDI	BO	EW	EP	ELO	Res.	Res.	DRX	TOO	MRAF	TSW	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
	Reset value			0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0054	FDCAN_IE	Res.	Res.	ARAE	PEDE	PEAE	WDIE	BOE	EWE	EPE	ELOE	Res.	Res.	DRXE	TOOE	MRAFE	TSWE	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
	Reset value			0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0058	FDCAN_ILS	Res.	Res.	ARAL	PEDL	PEAL	WDIL	BOL	EWL	EPL	ELOL	Res.	Res.	DRXL	TOOL	MRAFL	TSWL	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x005C	FDCAN_ILE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EINT1	EINT0
	Reset value																															0	0
0x006-0x007F	Reserved	Reserved																															
0x0080	FDCAN_GFC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ANFS [1:0]		ANFE [1:0]		EINT1	EINT0
	Reset value																											0	0	0	0	0	0
0x0084	FDCAN_SIDFC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSS[7:0]								FLSSA[13:0]														Res.	Res.
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x0088	FDCAN_XIDFC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSE[7:0]								FLESA[13:0]														Res.	Res.
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

表 502. FDCAN 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0090	FDCAN_XIDAM	Res.	Res.	Res.	EIDM[28:0]																												
	Reset value				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0094	FDCAN_HPMS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLST	FIDX[6:0]							MISI [1:0]		MIDX[5:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0098	FDCAN_NDAT1	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x009C	FDCAN_NDAT2	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00A0	FDCAN_RXF0C	F0OM	F0WM[6:0]							Res.	F0S[6:0]							F0SA[13:0]														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x00A4	FDCAN_RXF0S	Res.	Res.	Res.	Res.	Res.	Res.	RF0	F0F	Res.	Res.	F0PI[5:0]						Res.	Res.	F0GI[5:0]						Res.	Reserved	F0FL[6:0]					
	Reset value							0	0			0	0	0	0	0	0			0	0	0	0	0	0		0	0	0	0	0	0	0
0x00A8	FDCAN_RXF0A	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F0AI[5:0]					
	Reset value																											0	0	0	0	0	0
0x00AC	FDCAN_RXBC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RBSA[13:0]														Res.	Res.
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x00B0	FDCAN_RXF1C	F1OM	F1WM[6:0]							Res.	F1S[6:0]							F1SA[13:0]														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x00B4	FDCAN_RXF1S	DMS [1:0]		Res.	Res.	Res.	Res.	RF1L	F1F	Res.	Res.	F1PI[5:0]						Res.	Res.	F1GI[5:0]						Res.	Reserved	F1FL[6:0]					
	Reset value	0	0					0	0			0	0	0	0	0	0			0	0	0	0	0	0		0	0	0	0	0	0	0
0x00B8	FDCAN_RXF1A	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F1AI[5:0]					
	Reset value																											0	0	0	0	0	0
0x00BC	FDCAN_RXESC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RBDS [2:0]			Res.		F1DS [2:0]			Res.	F0DS [2:0]			
	Reset value																				0	0	0			0	0	0			0	0	0
0x00C0	FDCAN_TXBC	Res.	TFQM	TFQS[5:0]						Res.	Res.	NDTB[5:0]						TBSA[13:0]														Res.	Res.
	Reset value		0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x00C4	FDCAN_TXFQS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TFQF	TFQPI[4:0]					Res.	Res.	Res.	TFGI[4:0]					Res.	Res.	TFFL[5:0]					
	Reset value											0	0	0	0	0	0				0	0	0	0	0			0	0	0	0	0	0
0x00C8	FDCAN_TXESC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TBDS [2:0]		
	Reset value																														0	0	0

表 502. FDCAN 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00CC	FDCAN _TXBRP	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00D0	FDCAN _TXBAR	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00D4	FDCAN _TXBCR	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00D8	FDCAN _TXBTO	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00DC	FDCAN _TXBCF	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00E0	FDCAN _TXBTIE	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00E4	FDCAN _TXBCIE	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00F0	FDCAN _TXEFC	Res.	Res.	EFWM[5:0]						Res.	Res.	EFS[5:0]						EFSA[15:2]														Res.	Res.
	Reset value			0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x00F4	FDCAN _TXEFS	Res.	Res.	Res.	Res.	Res.	Res.	TEFL	EFF	Res.	Res.	Res.	EFPI[4:0]					Res.	Res.	Res.	EFGI[4:0]					Res.	Res.	EFFL[5:0]					
	Reset value							0	0				0	0	0	0	0				0	0	0	0	0			0	0	0	0	0	0
0x00F8	FDCAN _TXEFA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EFAI[4:0]				
	Reset value																												0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 2.5 节*。

53.6 TTCAN 寄存器

这些寄存器仅适用于 FDCAN1。

53.6.1 FDCAN TT 触发内存配置寄存器 (FDCAN_TTTMC)

FDCAN TT trigger memory configuration register

偏移地址：0x100

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TME[6:0]						
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMSA[13:0]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31:23 保留，必须保持复位值。

位 22:16 **TME[6:0]**：触发内存元素 (Trigger memory elements)
0：无触发内存
1-64：触发内存元素数
>64：大于 64 的值会被解析为 64
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 15:2 **TMSA[13:0]**：触发内存起始地址 (Trigger memory start address)。
消息 RAM 中触发内存的起始地址（32 位字地址，请参见*图 669*）。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 1:0 保留，必须保持复位值。

53.6.2 FDCAN TT 参考消息配置寄存器 (FDCAN_TTRMC)

FDCAN TT reference message configuration register

偏移地址：0x0104

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RMPS	XTD	Res.	RID[28:16]												
rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RID[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **RMPS**：参考消息有效负载选择 (Reference message payload select)
时间被控节点会忽略此位。
0：参考消息没有额外的有效负载
1：会从发送缓冲区 0 获取以下元素：
消息标记 MM、
事件 FIFO 控制 EFC、
数据长度代码 DLC、
数据字节 DB（级别 1：字节 2-8，级别 0、2：字节 5-8）
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 30 **XTD**：扩展标识符 (Extended identifier)
0：11 位标准标识符
1：29 位扩展标识符
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 29 保留，必须保持复位值。

位 28:0 **RID[28:0]**：参考标识符 (Reference identifier)。
通过参考消息发送并用于参考消息过滤的标识符。标准参考标识符或扩展参考标识符取决于 XTD 位。必须将标准标识符写入 ID[28:18]。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

53.6.3 FDCAN TT 操作配置寄存器 (FDCAN_TTOCF)

FDCAN TT operation configuration register

偏移地址：0x0108

复位值：0x0001 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	EVTP	ECC	EGTF	AWL[7:0]							
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EECS	IRTO[6:0]							LDSDL[2:0]			TM	GEN	Res.	OM[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

位 31:27 保留，必须保持复位值。

位 26 **EVTP**：事件触发极性 (Event trigger polarity)。
0：上升沿触发
1：下降沿触发
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 25 **ECC**：使能时钟校准 (Enable clock calibration)。
0：FDCAN 0 级、2 级中禁止自动时钟校准
1：FDCAN 0 级、2 级中使能自动时钟校准
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 24 **EGTF**：使能全局时间过滤 (Enable global time filtering)。
0：FDCAN 0 级、2 级中禁止全局时间过滤
1：FDCAN 0 级、2 级中使能全局时间过滤
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 23:16 **AWL[7:0]**：应用看门狗限制 (Application watchdog limit)。
可通过将 AWL 编程为 0x00 禁止应用看门狗。
0x00 到 FF：应用可延迟处理应用看门狗的最长时间。应用看门狗每隔 256 个 NTU 递增一次。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 15 **EECS**：使能外部时钟同步 (Enable external clock synchronization)
若使能，TUR 配置 (FDCAN_TURCF.NCL only) 可在 FDCAN 操作期间更新。
0：FDCAN 0 级、2 级中禁止外部时钟同步
1：FDCAN 0 级、2 级中使能外部时钟同步
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 14:8 **IRTO[6:0]**：初始参考触发偏移 (Initial reference trigger offset)。
0x00 到 7F 正偏移，范围为 0 到 127
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 7:5 **LDSDL[2:0]**：同步偏差限值的 LD (LD of synchronization deviation limit)。
同步偏差限值 SDL 是通过其双对数 LDSDL 配置的，公式为 SDL= 2*(LDSDL + 5)。SDL 在 32 和 4096 之间。不应超过 CAN 位定时配置提供的时钟容差。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 4 **TM**：时间主控节点 (Time master)。
0：禁止时间主控节点功能
1：可作为时间主控节点
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 3 **GEN**：间隙使能 (Gap enable)。
0：严格的时间触发操作
1：外部事件同步时间触发操作
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 2 保留，必须保持复位值。

位 1:0 **OM[1:0]**：工作模式 (Operation mode)
00：事件驱动 CAN 通信（默认）
01：TTCAN 1 级
10：TTCAN 2 级
11：TTCAN 0 级
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

53.6.4 FDCAN TT 矩阵限值寄存器 (FDCAN_TTMLM)

FDCAN TT matrix limits register

偏移地址：0x010C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	ENTT[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TXEW[3:0]				CSS[1:0]		CCM[5:0]					
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **ENTT[11:0]**：预期发送触发数 (Expected number of Tx triggers)
0x000 到 FFF：一个矩阵周期中的预期发送触发数。
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 15:12 保留，必须保持复位值。

位 11:8 **TXEW[3:0]**：发送使能窗口 (Tx enable window)
0x0 到 F：发送使能窗口的长度，1-16 个 NTU 周期
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 7:6 **CSS[1:0]**：周期开始同步 (Cycle start synchronization)
使能同步脉冲输出。
00：无同步脉冲
01：基本周期开始时同步脉冲
10：矩阵周期开始时同步脉冲
11：Reserved
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 5:0 **CCM[5:0]**：周期计数最大值 (Cycle count Max)
0x00：每个矩阵周期 1 个基本周期
0x01：每个矩阵周期 2 个基本周期
0x03：每个矩阵周期 4 个基本周期
0x07：每个矩阵周期 8 个基本周期
0x0F：每个矩阵周期 16 个基本周期
0x1F：每个矩阵周期 32 个基本周期
0x3F：每个矩阵周期 64 个基本周期
其他值：保留
这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

注： *ISO 11898-4 第 5.2.1 部分要求仅配置上面列出的周期计数值。配置其他值也可以，但可能导致矩阵周期不一致。*

53.6.5 FDCAN TUR 配置寄存器 (FDCAN_TURCF)

FDCAN TUR configuration register

NTU 的长度由下式给出：NTU = CAN 时钟周期 x NC/DC。

NC 为 18 位值。其高位部分 NCH[17:16] 通过硬接线的方式写入 0b01。因此，NC 的范围从 0x10000 扩展到 0x1FFFF。NCL 配置的值为 FDCAN_TURNA.NAV[15:0] 的初始值。DC 通过硬件复位设为 0x1000，不得写入为 0x0000。

- 1 级：NC 4 × DC 和 NTU = CAN 位时间
- 0 级和 2 级：NC 8 × DC

FDCAN_TUR 的实际值可通过 TTCAN 0 级和 2 级的时钟偏移补偿功能进行，以便将 NTU 的节点本地视图调整为 NTU 的时间主控节点视图。自动偏移补偿不会自动更改 DC，FDCAN_TURNA.NAV 可在由 FDCAN_TTOCF.LDSDL 提供的同步偏差限值范围内在 NC 周围调整。NC 和 DC 应编程为最大的适用值，以便使偏移补偿过程达到最佳计算精度。

偏移地址：0x0110

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ELT	Res.	DC[13:0]													
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NCL[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **ELT**：使能本地时间 (Enable local time)。

0：停止本地时间（默认）

1：使能本地时间

这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

注意：本地时间通过将 ELT 置 1 来启动，之后会一直保持激活状态，直到 ELT 复位或下一次硬件复位为止。当 FDCAN_TURCF.ELT = 1 时，会锁定 FDCAN_TURCF.DC。如果向 ELT 写入 0，则在新值同步到 CAN 时钟域之前，可读值始终为 1。在此期间，对寄存器其他位的写访问将被锁定。

位 30 保留，必须保持复位值。

位 29:16 **DC[13:0]**：分母配置 (Denominator configuration)。

0x0000：非法值

0x0001 到 0x3FFF：分母配置。

这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

位 15:0 **NCL[15:0]**：分子配置下限 (Numerator configuration low)。

仅当配置为 FDCAN_TURCF.ELT = 0 或者 FDCAN_TTOCF.EECS（外部时钟同步使能）置 1 时，才能对 TUR 分子配置下限进行写访问。如果在 TT 配置模式以外写入 NCL 的新值，新值会在 FDCAN_TTOST.WECS 清零后生效。NCL 锁定，FDCAN_TTOST.WECS 为 1。

0X0000 到 FFFF 分子配置下限

这些位受写保护 (P)，仅当 FDCAN_CCCR 寄存器的位 CCE 和位 INIT 置 1 时，才能写入。

注：如果 TTCAN 1 级中的 NC < 7 × DC，则触发内存中的后续时间标记之间的差值必须至少为两个 NTU。

53.6.6 FDCAN TT 操作控制寄存器 (FDCAN_TTOCN)

FDCAN TT operation control register

偏移地址：0x0114

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCKC	Res.	ESCN	NIG	TMG	FGP	GCS	TTIE	TMC[1:0]		RTIE	SWS[1:0]		SWP	ECS	SGT
r		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **LCKC**：TT 操作控制寄存器锁定 (TT operation control register locked)。

对寄存器 FDCAN_TTOCN 进行写访问会将该位置 1。当更新的配置已同步到 CAN 时钟域时，该位会复位。

0：使能对 FDCAN_TTOCN 的写访问

1：锁定对 FDCAN_TTOCN 的写访问

位 14 保留，必须保持复位值。

位 13 **ESCN**：外部同步控制 (External synchronization control)

若使能该位，FDCAN 会将其周期时间相位同步到由事件触发引脚的上升沿所指示的外部事件（请参见*第 53.4.17 节：与外部时间调度同步*）。

0：禁止外部同步

1：使能外部同步

位 12 **NIG**：接下来是空闲 (Next is gap)。

仅当 FDCAN 为实际时间主控节点，并且配置为进行外部事件同步时间触发操作时 (FDCAN_TTOCF.GEN = 1)，该位才能置 1。

0：无操作，接收任何参考消息时复位

1：Next_is_Gap = 1 时发送下一条参考消息

位 11 **TMG**：时间标记间隙 (Time mark gap)。

0：通过每条参考消息复位

1：寄存器时间标记中断 FDCAN_TTIR.RTMI 激活时，开始发送下一条参考消息

位 10 **FGP**：结束空闲 (Finish gap)。
由 CPU 置 1，通过每条参考消息复位
0：未请求任何参考消息
1：应用请求开始发送参考消息

位 9 **GCS**：空闲控制选择 (Gap control select)
0：空闲控制与事件触发无关
1：通过输入事件触发引脚进行空闲控制

位 8 **TTIE**：触发时间标记中断脉冲使能 (Trigger time mark interrupt pulse enable)
外部时间标记事件是通过触发内存元素 TMEX 配置的。当触发内存元素生效且 FDCAN 处于同步状态 In_Schedule 或 In_Gap 时，会生成触发时间标记中断脉冲。
0：禁止触发时间标记中断输出 fdcan1_tmp（多个实例时，仅一个实例时为 fdcan_tmp）
1：使能触发时间标记中断输出 fdcan1_tmp（多个实例时，仅一个实例时为 fdcan_tmp）

位 7:6 **TMC[1:0]**：寄存器时间标记比较 (Register time mark compare)。
00：未生成寄存器时间标记中断
01：如果时间标记 = 周期时间，则会生成寄存器时间标记中断
10：如果时间标记 = 本地时间，则会生成寄存器时间标记中断
11：如果时间标记 = 全局时间，则会生成寄存器时间标记中断
注意：更改时间标记参考（周期时间、本地时间、全局时间）时，建议先写入 TMC = 00，然后重新配置 FDCAN_TTTMK，最后将 FDCAN_TMC 设为所需时间参考。

位 5 **RTIE**：寄存器时间标记中断脉冲使能 (Register time mark interrupt pulse enable)。
寄存器时间标记中断是通过寄存器 FDCAN_TTTMK 配置的。当 FDCAN_TTOCN.TMC 引用的时间（周期时间、本地时间或全局时间）等于 FDCAN_TTTMK.TM 时，会生成长度为一个 fdcan_tq_ck 周期的寄存器时间标记中断脉冲，与同步状态无关。
0：禁止寄存器时间标记中断输出
1：使能寄存器时间标记中断输出

位 4:3 **SWS[1:0]**：秒表源 (Stop watch source)。
00：禁止秒表
01：周期时间的实际值会复制到 FDCAN_TTCPT.SWV
10：本地时间的实际值会复制到 FDCAN_TTCPT.SWV
11：全局时间的实际值会复制到 FDCAN_TTCPT.SWV

位 2 **SWP**：秒表极性 (Stop watch polarity)。
0：上升沿触发
1：下降沿触发

位 1 **ECS**：外部时钟同步 (External clock synchronization)。
如果节点为实际时间主控节点，则向 ECS 写入 1 会将 FDCAN_TTOST.WECS 置 1。一个 APB 时钟周期后，ECS 会复位。外部时钟同步会在下一基本周期开始时生效。

位 0 **SGT**：设置全局时间 (Set global time)
如果节点为实际时间主控节点，则向 SGT 写入 1 会将 FDCAN_TTOST.WGDT 置 1。一个 APB 时钟周期后，SGT 会复位。当节点发送的下一条参考消息的 Master_Ref_Mark 被写入 FDCAN_TTGTP 的预设值修改时，全局时间预设会生效。

53.6.7 FDCAN TT 全局时间预设寄存器 (FDCAN_TTGTP)

FDCAN TT global time preset register

如果 TTOST.WGDT 置 1，则发送的下一条参考消息的 Master_Ref_Mark 会被预设值修改，并且 Disc_Bit = 1，从而会同时预设所有节点中的全局时间。

每次 Disc_Bit = 1 的参考消息生效、或者节点不是当前时间主控节点时，TP 会复位为 0x0000。FDCAN_TTOCN.SGT 置 1 后，如果 FDCAN_TTOST.WGTD = 1，则 TP 会锁定，直至 Disc_Bit = 1 的参考消息生效或节点不再是当前时间主控节点。

偏移地址：0x0118

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTP[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TP[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **CTP[15:0]**：周期时间目标相位 (Cycle time target phase)。
CTP 受写保护，而 FDCAN_TTOCN.ESCN 或 FDCAN_TTOST.SPL 置 1（请参见*第 53.4.17 节：与外部时间调度同步*）。
0X0000 到 FFFF 定义预期事件触发上升沿时周期时间的目标值

位 15:0 **TP[15:0]**：时间预设 (Time preset)
TP 受写保护，而 FDCAN_TTOST.WGTD 置 1。
0X0000 到 7FFF 下一主控节点参考标记 = 主控节点参考标记 + TP
0x8000 保留
0x8001-FFFF 下一主控节点参考标记 = 主控节点参考标记 - (0x10000 - TP)。

53.6.8 FDCAN TT 时间标记寄存器 (FDCAN_TTTMK)

FDCAN TT time mark register

当 FDCAN_TTOCN.TMC 指示的时基（周期时间、本地时间或全局时间）的值与 TM 相同时，会生成时间标记中断 (FDCAN_TTIR.TMI = 1)。

偏移地址：0x011C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCKM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TICC[6:0]						
r									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TM[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **LCKM**：TT 时间标记寄存器锁定 (TT time mark register locked)。
始终通过对寄存器 FDCAN_TTOCN 进行写访问来置 1。当 FDCAN_TTOCN.TMC 为 00 时，对寄存器 FDCAN_TTTMK 进行写访问会将该位置 1。寄存器已同步到 CAN 时钟域后，此位会复位。
0：使能对 FDCAN_TTTMK 的写访问
1：锁定对 FDCAN_TTTMK 的写访问

位 30:23 保留，必须保持复位值。

位 22:16 **TICC[6:0]**：时间标记周期代码 (Time mark cycle code)。
时间标记有效的周期计数。
0b000000x 对所有周期有效
周期计数 mod2 = c 时，每 2 个周期，0b000001c 有效
周期计数 mod4 = cc 时，每 4 个周期，0b00001cc 有效
周期计数 mod8 = ccc 时，每 8 个周期，0b0001ccc 有效
周期计数 mod16 = cccc 时，每 16 个周期，0b001cccc 有效
周期计数 mod32 = ccccc 时，每 32 个周期，0b01ccccc 有效
周期计数 mod64 = cccccc 时，每 64 个周期，0b1cccccc 有效

位 15:0 **TM[15:0]**：时间标记 (Time mark)
0X0000 到 FFFF 时间标记

注： *对寄存器 FDCAN_TTTMK 进行字节访问时，建议先禁止时间标记比较功能 (FDCAN_TTOCN.TMC = 00)，以免对不一致的寄存器值进行比较。*

53.6.9 FDCAN TT 中断寄存器 (FDCAN_TTIR)

FDCAN TT interrupt register

如果检测到其中一个列出的条件（边沿有效），标志会置 1。必须由用户清零，否则将一直保持置位。通过向对应位位置写入 1 将标志清零。写入 0 不会起任何作用。硬复位会将寄存器清零。

偏移地址：0x0120

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CER	AW	WT
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWTG	ELC	SE2	SE1	TXO	TXU	GTE	GTD	GTW	SWE	TTMI	RTMI	SOG	CSM	SMC	SBC
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:19 保留，必须保持复位值。

位 18 **CER**：配置错误 (Configuration error)
触发次序颠倒。
0：触发列表中未发现错误
1：触发列表中发现错误

位 17 **AW**：应用看门狗 (Application watchdog)
0：及时处理应用看门狗
1：未及时处理应用看门狗

位 16 **WT**：监视触发 (Watch trigger)
0：未丢失参考消息
1：丢失参考消息（0 级：周期时间 0xFF00）

位 15 **IWTG**：初始化监视触发 (Initialization watch trigger)
通过复位 IWT 重新开始初始化。
0：系统启动期间未丢失参考消息
1：由于参考消息丢失，系统未启动

位 14 **ELC**：错误级别已更改 (Error level changed)
如果初始化过程中错误级别发生了变化，此位不会置 1。
0：错误级别未更改
1：错误级别已更改

位 13 **SE2**：调度错误 2 (Scheduling error 2)
0：无调度错误 2
1：发生调度错误 2

位 12 **SE1**：调度错误 1 (Scheduling error 1)
0：无调度错误 1
1：发生调度错误 1

位 11 **TXO**：发送计数上溢 (Tx count overflow)
0：发送触发数与预期相同
1：一个周期中的发送触发数比预期多

位 10 **TXU**：发送计数下溢 (Tx count underflow)
0：发送触发数与预期相同
1：一个周期中的发送触发数比预期少

位 9 **GTE**：全局时间错误 (Global time error)
同步偏差 SD 超过 FDCAN_TTOCF.LDSDL 指定的限值，仅限 TTCAN 0 级、2 级。
0：同步偏差在限值范围内
1：同步偏差超出限值范围

位 8 **GTD**：全局时间不连续 (Global time discontinuity)
0：全局时间没有不连续的情况
1：全局时间不连续

位 7 **GTW**：全局时间回卷 (Global time wrap)
0：未发生全局时间回卷
1：发生了从 0xFFFF 到 0x0000 的全局时间回卷

位 6 **SWE**：秒表事件 (Stop watch event)
0：秒表触发引脚上未检测到上升沿/下降沿
1：秒表触发引脚上检测到上升沿/下降沿

位 5 **TTMI**：内部触发时间标记事件 (Trigger time mark event internal)
内部时间标记事件是通过触发内存元素 TMIN 配置的（请参见*第 53.4.23 节*）。当触发内存元素生效、且 FDCAN 处于同步状态 In_Gap 或 In_Schedule 时，此位会置 1。
0：未达到时间标记
1：已达到时间标记（0 级：周期时间 FDCAN_TTOCF.RTO x 0x200）

位 4 **RTMI**：寄存器时间标记中断 (Register time mark interrupt)
当 TTOCN.TMC 引用的时间（周期时间、本地时间或全局时间）等于 FDCAN_TTTMK.TM 时，该位会置 1，与同步状态无关。
0：未达到时间标记
1：已达到时间标记

位 3 **SOG**：空闲开始 (Start of gap)
0：未发现 Next_is_Gap 位置 1 的参考消息
1：Next_is_Gap 位置 1 的参考消息生效

位 2 **CSM**：同步模式更改 (Change of synchronization mode)
0：主控节点与被控节点的关系或调度同步没有变化
1：主控节点与被控节点的关系或调度同步发生了变化，FDCAN_TTOST.SPL 复位时也会置 1

位 1 **SMC**：矩阵周期开始 (Start of matrix cycle)
0：复位后未开始任何矩阵周期
1：矩阵周期已开始

位 0 **SBC**：基本周期开始 (Start of basic cycle)
0：复位后未开始任何基本周期
1：基本周期已开始

53.6.10 FDCAN TT 中断使能寄存器 (FDCAN_TTIE)

FDCAN TT interrupt enable register

TT 中断使能寄存器中的设置决定了 TT 中断寄存器中的哪些状态更改会产生中断。

偏移地址：0x0124

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CERE	AWE	WTE
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWTE	ELCE	SE2E	SE1E	TXOE	TXUE	GTEE	GTDE	GTWE	SWEE	TTMIE	RTMIE	SOGE	CSME	SMCE	SBCE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:19 保留，必须保持复位值。

位 18 **CERE**：配置错误中断使能 (Configuration error interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 17 **AWE**：应用看门狗中断使能 (Application watchdog interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 16 **WTE**：监视触发中断使能 (Watch trigger interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 15 **IWTE**：初始化监视触发中断使能 (Initialization watch trigger interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 14 **ELCE**：更改错误级别中断使能 (Change error level interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 13 **SE2E**：调度错误 2 中断使能 (Scheduling error 2 interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 12 **SE1E**：调度错误 1 中断使能 (Scheduling error 1 interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 11 **TXOE**：发送计数上溢中断使能 (Tx count overflow interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 10 **TXUE**：发送计数下溢中断使能 (Tx count underflow interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 9 **GTEE**：全局时间错误中断使能 (Global time error interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 8 **GTDE**：全局时间不连续中断使能 (Global time discontinuity interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 7 **GTWE**：全局时间回卷中断使能 (Global time wrap interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 6 **SWEE**：秒表事件中断使能 (Stop watch event interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 5 **TTMIE**：内部触发时间标记事件中断使能 (Trigger time mark event internal interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 4 **RTMIE**：寄存器时间标记中断使能 (Register time mark interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 3 **SOGE**：空闲开始中断使能 (Start of gap interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 2 **CSME**：同步模式更改中断使能 (Change of synchronization mode interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 1 **SMCE**：矩阵周期开始中断使能 (Start of matrix cycle interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

位 0 **SBCE**：基本周期开始中断使能 (Start of basic cycle interrupt enable)
0：禁止 TT 中断
1：使能 TT 中断

53.6.11 FDCAN TT 中断线选择寄存器 (FDCAN_TTILS)

FDCAN TT interrupt line select register

TT 中断线选择寄存器将特定中断标志生成的中断从 TT 中断寄存器分配到两个模块中断线之一。要生成中断，必须通过 FDCAN_ILE.EINT0 和 FDCAN_ILE.EINT1 使能相应的中断线。

偏移地址：0x0128

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CERL	AWL	WTL
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWTL	ELCL	SE2L	SE1L	TXOL	TXUL	GTEL	GTDL	GTWL	SWEL	TTMIL	RTMIL	SOGL	CSML	SMCL	SBCL
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:19 保留，必须保持复位值。

位 18 **CERL**：配置错误中断线 (Configuration error interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 17 **AWL**：应用看门狗中断线 (Application watchdog interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 16 **WTL**：监视触发中断线 (Watch trigger interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 15 **IWTL**：初始化监视触发中断线 (Initialization watch trigger interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 14 **ELCL**：更改错误级别中断线 (Change error level interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 13 **SE2L**：调度错误 2 中断线 (Scheduling error 2 interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 12 **SE1L**：调度错误 1 中断线 (Scheduling error 1 interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 11 **TXOL**：发送计数上溢中断线 (Tx count overflow interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 10 **TXUL**：发送计数下溢中断线 (Tx count underflow interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 9 **GTEL**：全局时间错误中断线 (Global time error interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 8 **GTDL**：全局时间不连续中断线 (Global time discontinuity interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 7 **GTWL**：全局时间回卷中断线 (Global time wrap interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 6 **SWEL**：秒表事件中断线 (Stop watch event interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 5 TTMIL：内部触发时间标记事件中断线 (Trigger time mark event internal interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 4 **RTMIL**：寄存器时间标记中断线 (Register time mark interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 3 **SOGL**：空闲开始中断线 (Start of gap interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 2 **CSML**：同步模式更改中断线 (Change of synchronization mode interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 1 **SMCL**：矩阵周期开始中断线 (Start of matrix cycle interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 0 **SBCL**：基本周期开始中断线 (Start of basic cycle interrupt line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

53.6.12 FDCAN TT 工作状态寄存器 (FDCAN_TTOST)

FDCAN TT operation status register

偏移地址：0x012C

复位值：0x0000 0080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPL	WECS	AWE	WFE	GSI	TMP[2:0]			GFI	WGTD	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r	r	r	r	r	r						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTO[7:0]								QCS	QGTP	SYS[1:0]		MS[1:0]		EL[1:0]	
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31 **SPL**：调度相位锁定 (Schedule phase lock)

仅当外部同步使能 (FDCAN TTOCN.ESCN=1) 时，该位才有效。在这种情况下，该位会指示由 FDCAN_TTGTP.CTP 配置的周期时间与事件触发引脚上升沿的周期时间之差小于还是等于 9 个 NTU（请参见*第 53.4.17 节：与外部时间调度同步*）。

0：相位超出范围

1：相位在范围内

位 30 **WECS**：等待外部时钟同步 (Wait for external clock synchronization)。

0：无外部时钟同步挂起

1：节点等待外部时钟同步生效。此位会在下一基本周期开始时复位。

位 29 **AWE**：应用看门狗事件 (Application watchdog event)

应用看门狗是通过读取 FDCAN_TTOST 来处理的。如果未及时处理看门狗，位 AWE 会置 1，所有 FDCAN 通信都会停止，FDCAN 会设为进入总线监控模式。

0：及时处理应用看门狗

1：未及时处理应用看门狗

位 28 **WFE**：等待事件 (Wait for event)

0：未发布间隙，由 Next_is_Gap = 0 的参考消息复位

1：接收到 Next_is_Gap = 1 的参考消息

位 27 **GSI**：空闲开始指示符 (Gap started indicator)

0：调度中没有间隙，由每条参考消息复位，用于所有时间被控节点

1：基本周期开始后的空闲时间

位 26:24 **TMP[2:0]**：时间主控节点优先级 (Time master priority)

0x0-7：实际时间主控节点的优先级

位 23 **GFI**：空闲结束指示符 (Gap finished indicator)

CPU 对 FDCAN_TTOCN.FGP 进行写操作时置 1，或者在 TMG = 1 时通过时间标记中断置 1，或者在 FDCAN_TTOCN.GCS = 1 时通过输入引脚置 1（事件触发）。不会由 Ref_Trigger_Gap 置 1，间隙由另一发送参考消息的节点结束时也不会置 1。

0：每条参考消息结束时复位

1：空闲由 FDCAN 结束

位 22 **WGTD**：等待全局时间不连续 (Wait for global time discontinuity)

0：没有全局时间预设挂起

1：节点等待全局时间预设生效节点已发送 Disc_Bit = 1 的参考消息或接收到参考消息后，该位会复位。

位 21:16 保留，必须保持复位值。

位 15:8 **RTO[7:0]**：参考触发偏移 (Reference trigger offset)。

参考触发偏移是有符号整型值，范围为 -127 (0x81) 到 127 (0x7F)。达到下限 -127 时，不会发出通知。如果 FDCAN 成为时间主控节点（MS[1:0] = 11），由于用户和 CAN 时钟域之间要进行同步，因此 RTO 的复位会延迟。对于时间被控节点，会读取由 FDCAN_TTOCF.IRTO 配置的值。

0x00-FF：实际参考触发偏移值

位 7 **QCS**：时钟速度质量 (Quality of clock speed)

仅在 TTCAN 0 级和 2 级中有意义，否则会固定为 1。

0：本地时钟速度未同步到时间主控节点时钟速度

1：同步偏差 ≤ SDL

位 6 **QGTP**：全局时间相位质量 (Quality of global time phase)

仅在 TTCAN 0 级和 2 级中有意义，否则会固定为 0。

0：全局时间无效

1：全局时间与时间主控节点同相

位 5:4 **SYS[1:0]**：同步状态 (Synchronization state)
00：不同步
01：同步到 FDCAN 通信
10：调度被空闲挂起 (In_Gap)
11：同步到调度 (In_Schedule)

位 3:2 **MS[1:0]**：主控节点状态 (Master state)。
00：Master_Off，无任何相关主控节点属性
01：作为时间被控节点工作
10：作为备份时间主控节点工作
11：作为当前时间主控节点工作

位 1:0 **EL[1:0]**：错误级别 (Error level)。
00：严重程度 0 - 无错误
01：严重程度 1 - 警告
10：严重程度 2 - 错误
11：严重程度 3 - 严重错误

53.6.13 FDCAN TUR 分子实际寄存器 (FDCAN_TURNA)

FDCAN TUR numerator actual register

TTCAN 1 级中没有偏移补偿 (NAV = NC)。在 TTCAN 0 级和 2 级中，会计算节点本地时钟与时间主控节点本地时钟之间的偏移。当同步偏差（NC 与计算出的 NAV 之差）小于 2*(FDCAN_TTOCF.LDSDL + 5) 时，会对偏移进行补偿。如果 FDCAN_TTOCF.LDSDL < 7 时，会得出 NAV 的最大范围为 (NC - 0x1000) ≤ NAV ≤ (NC + 0x1000)。

偏移地址：0x0130

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NAV[17:16]	
														r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAV[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:18 保留，必须保持复位值。

位 17:0 **NAV[17:0]**：分子实际值 (Numerator actual value)
0x0EFFF：非法值
0x0F000 到 20FFF：实际分子值
0x21000：非法值

53.6.14 FDCAN TT 本地和全局时间寄存器 (FDCAN_TTLGT)

FDCAN TT local and global time register

偏移地址：0x0134

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 **GT[15:0]**：全局时间 (Global time)
节点本地时间与其本地偏移之和的非小数部分（请参见*第 53.4.12 节*）。
0x0000 到 FFFF：FDCAN 网络的全局时间值

位 15:0 **LT[15:0]**：本地时间 (Local time)
本地时间的非小数部分，每隔一个本地 NTU 递增一次（请参见*第 53.4.12 节*）。
0x0000 到 FFFF：FDCAN 节点的本地时间值

53.6.15 FDCAN TT 周期时间和计数寄存器 (FDCAN_TTCTC)

FDCAN TT cycle time and count register

偏移地址：0x0138

复位值：0x003F 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC[5:0]					
										r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:22 保留，必须保持复位值。

位 21:16 **CC[5:0]**：周期计数 (Cycle count)
0X00 到 3F：系统矩阵中的实际基本周期计数

位 15:0 **CT[15:0]**：周期时间 (Cycle time)
节点本地时间与 Ref_Mark 之差的非小数部分（请参见*第 53.4.12 节*）。
0X0000 到 FFFF：FDCAN 基本周期的周期时间值

53.6.16 FDCAN TT 捕获时间寄存器 (FDCAN_TTCPT)

FDCAN TT capture time register

偏移地址：0x013C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWV[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCV[5:0]					
										r	r	r	r	r	r

位 31:16 **SWV[15:0]**：秒表值 (Stop watch value)

当秒表触发引脚上出现上升/下降沿时（通过 FDCAN_TTOCN.SWP 配置），如果 FDCAN_TTOCN.SWS 不为 00 且 FDCAN_TTIR.SWE 为 0，则由 FDCAN_TTOCN.SWS 选择的实际时间值（周期时间、本地时间和全局时间）会复制到 SWV，并且 TFDCAN_TIR.SWE 将置 1。通过复位 FDCAN_TTIR.SWE 使能捕获下一秒表值。

0X0000 到 FFFF：捕获的秒表值

位 15:6 保留，必须保持复位值。

位 5:0 **CCV[5:0]**：周期计数值 (Cycle count value)

与 SWV 一起捕获的周期计数值。

0X00 到 3F：捕获的周期计数值

53.6.17 FDCAN TT 周期同步标记寄存器 (FDCAN_TTCSM)

FDCAN TT cycle sync mark register

偏移地址：0x0140

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSM[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **CSM[15:0]**：周期同步标记 (Cycle sync mark)

周期同步标记以周期时间为单位进行测量。当参考消息生效时，此位会更新，并会在下一参考消息生效之前保持其值。

0X0000 到 FFFF：捕获的周期时间

53.6.18 FDCAN TT 触发选择寄存器 (FDCAN_TTTS)

FDCAN TT trigger select register

FDCAN_TTTS 寄存器中的设置会选择将用作事件触发和停止监控触发的输入。

偏移地址：0x0300

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EVTSEL[1:0]		Res.	Res.	SWTDEL[1:0]	
										rw	rw			rw	rw

位 31:6 保留，必须保持复位值。

位 5:4 **EVTSEL[1:0]**：事件触发输入选择 (Event trigger input selection)

这些位用于选择要用作事件触发的输入

00：fdcan1_evt0

01：fdcan1_evt1

10：fdcan1_evt2

11：fdcan1_evt3

位 3:2 保留，必须保持复位值。

位 1:0 **SWTDEL[1:0]**：秒表触发输入选择 (Stop watch trigger input selection)

这些位用于选择要用作停止监视触发的输入

00：fdcan1_swt0

01：fdcan1_swt1

10：fdcan1_swt2

11：fdcan1_swt3

53.6.19 FDCAN TT 寄存器映射

表 503. FDCAN TT 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0100	FDCAN_TTTMC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TME[6:0]							TMSA[13:0]														Res.	Res.
	Reset value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x0104	FDCAN_TTRMC	RMPS	XTD	Res.	RID[28:0]																												
	Reset value	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0108	FDCAN_TTOCF	Res.	Res.	Res.	Res.	Res.	EVTP	ECC	EGTF	AWL[7:0]								EECS	IRTO[6:0]							LDSDL [2:0]			TM	GEN	Res.	OM [1:0]	
	Reset value						0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
0x010C	FDCAN_TTMLM	Res.	Res.	Res.	Res.	ENTT[11:0]												Res.	Res.	Res.	Res.	TXEW[3:0]				CSS [1:0]		CCM[5:0]					
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 503. FDCAN TT 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0110	FDCAN _TURCF	ELT	Res.	DC[13:0]														NCL[15:0]															
	Reset value	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0114	FDCAN _TTOCN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKC	Res.	ESCN	NIG	TMG	FGP	GCS	TTIE	TMC[1:0]		RTIE	SWS[1:0]		SWP	ECS	SGT
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0118	FDCAN _TTGTP	CTP[15:0]																TP[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x011C	FDCAN _TTTMK	LCKM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TICC[6:0]							TM[15:0]															
	Reset value	0									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0120	FDCAN_TTIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CER	AW	WT	IWTG	ELC	SE2	SE1	TXO	TXU	GTE	GTD	GTW	SWE	TTMI	RTMI	SOG	CSM	SMC	SBC
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0124	FDCAN_TTIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CERE	AWE	WTE	IWTE	ELCE	SE2E	SE1E	TXOE	TXUE	GTEE	GTDE	GTWE	SWEE	TTMIE	RTMIE	SOGE	CSME	SMCE	SBCE
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0128	FDCAN_TTILS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CERL	AWL	WTL	IWTL	ELCL	SE2L	SE1L	TXOL	TXUL	GTEL	GTDL	GTWL	SWEL	TTMIL	RTMIL	SOGL	CSML	SMCL	SBCL
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x012C	FDCAN _TTOST	SPL	WECS	AWE	WFE	GSI	TMP [2:0]			GFI	WGTD	Res.	Res.	Res.	Res.	Res.	Res.	RTO[7:0]								QCS	QGTP	SYS [1:0]		ms [1:0]		EL [1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0x0130	FDCAN _TURNA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NAV[17:0]																	
	Reset value															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0134	FDCAN _TTLGT	GT[15:0]																LT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0138	FDCAN _TTCTC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC[5:0]						CT[15:0]															
	Reset value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x013C	FDCAN _TTCPT	SWV[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCV[5:0]					
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0											0	0	0	0	0	0
0x0140	FDCAN _TTCSM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSM[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0144 to 0x01FC	Reserved																																
	Reset value																																

表 503. FDCAN TT 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0300	FDCAN_TTTS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EVTSEL[1:0]		Res.	Res.	SWTSEL[1:0]	
	Reset value																											0	0			0	0

有关寄存器边界地址的信息，请参见第 147 页的第 2.5 节。

53.7 CCU 寄存器

53.7.1 时钟校准单元核心版本寄存器 (FDCAN_CCU_CREL)

Clock calibration unit core release register

偏移地址：0x0000

复位值：0x1114 1218

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REL[3:0]				STEP[3:0]				SUBSTEP[3:0]				YEAR[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MON[7:0]								DAY[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **REL[3:0]**：核心版本 = 1

位 27:24 **STEP[3:0]**：核心版本主阶段号 = 1

位 23:20 **SUBSTEP[3:0]**：核心版本子阶段号 = 1

位 19:16 **YEAR[3:0]**：发布年份 =

位 15:8 **MON[7:0]**：发布月份 = 12

位 7:0 **DAY[7:0]**：发布日期 = 18

53.7.2 校准配置寄存器 (FDCAN_CCU_CCFG)

Calibration configuration register

偏移地址：0x0004

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CDIV[3:0]			
rw												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCPM[7:0]								CFL	BCC	Res.	TQBT[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31 **SWR**：软件复位 (Software reset)

向该位写入 1 会将校准 FSM 复位为 Not_Calibrated 状态 (FDCAN_CCU_CSTAT.CALS = 00)。校准看门狗值 CWD.WDV 也会复位。寄存器 FDCAN_CCFG、FDCAN_CCU_CSTAT 和校准看门狗配置 CWD.WDC 不会更改。在复位完成之前，此位保持置 1 状态。

仅当 FDCAN 控制位 FDCAN_CCCR.CCE = 1 且 FDCAN_CCCR.INIT = 1 时，才能由用户对标有“P = 受写保护”的寄存器/位进行写访问。

位 30:20 保留，必须保持复位值。

位 19:16 **CDIV[3:0]**：时钟分频器 (Clock divider)

如果绕过时钟校准 (BCC = 1)，则必须配置时钟分频，以确保满足 FDCAN 要求。

0000：1 分频
0001：2 分频
0010：4 分频
0011：6 分频
0100：8 分频
0101：10 分频
0110：12 分频
0111：14 分频
1000：16 分频
1001: 18 分频
1010: 20 分频
1011: 22 分频
1100: 24 分频
1101: 26 分频
1110: 28 分频
1111：30 分频

仅当 FDCAN 控制位 FDCAN_CCCR.CCE = 1 且 FDCAN_CCCR.INIT = 1 时，才能由用户对标有“P = 受写保护”的寄存器/位进行写访问。

位 15:8 **OCPM[7:0]**：振荡器时钟周期最小值 (Oscillator clock periods minimum)

以两个 CAN 位时间为单位配置最小周期数。在基本校准中会使用 OCPM，以免总线上存在毛刺时出现测量错误。配置的周期数为 OCPM × 32。配置取决于频率以及在 FDCAN 模块中配置的比特率（125 kb/s 到 1 Mb/s）。建议将该值配置为略小于两个 CAN 位时间。当 fdcan_ker_ck 频率为 80 MHz 且 CAN 比特率为 1 Mb/s 时，复位值为 1.6 倍的位时间。

仅当 FDCAN 控制位 FDCAN_CCCR.CCE = 1 且 FDCAN_CCCR.INIT = 1 时，才能由用户对标有“P = 受写保护”的寄存器/位进行写访问。

位 7 **CFL**：校准字段长度 (Calibration field length)

0：校准字段长度为 32 位
1：校准字段长度为 64 位

仅当 FDCAN 控制位 FDCAN_CCCR.CCE = 1 且 FDCAN_CCCR.INIT = 1 时，才能由用户对标有“P = 受写保护”的寄存器/位进行写访问。

位 6 **BCC**：绕过时钟校准 (Bypass clock calibration)

如果该位置 1，时钟输入 fdcan_ker_ck 会通过可由 CDIV 配置的时钟分频器发送到时间片时钟。在这种情况下，必须配置已连接 FDCAN 的波特率预分频器，以生成 FDCAN 内部时间片时钟。

0：时钟校准单元生成时间片时钟
1：绕过时钟校准单元（默认配置）

位 5 保留，必须保持复位值。

位 4:0 **TQBT[4:0]**：每个位时间的时间片 (Time quanta per bit time)

配置每个位时间的时间片数，与在 FDCAN 模块中配置的值相同。得出的时间片时钟 fdcan_tq_ck 的范围为 0.5 MHz（比特率为 125 kb/s，每个位时间有 4 个 tq）到 25 MHz（比特率为 1 Mb/s，每个位时间有 25 个 tq）。有效值为 4 到 25。如果配置的值小于 4，则会被解析为 4，如果配置的值大于 25，则会被解析为 25。

仅当 FDCAN 控制位 FDCAN_CCCR.CCE = 1 且 FDCAN_CCCR.INIT = 1 时，才能由用户对标有“P = 受写保护”的寄存器/位进行写访问。

53.7.3 校准状态寄存器 (FDCAN_CCU_CSTAT)

Calibration status register

偏移地址：0x0008

复位值：0x0203 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CALS[1:0]		Res.	TQC[10:0]											OCPC[17:16]	
r	r		r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCPC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:30 **CALS[1:0]**：校准状态 (Calibration state)

00：Not_Calibrated

01：Basic_Calibrated

10：Precision_Calibrated

11：保留

位 29 保留，必须保持复位值。

位 28:18 **TQC[10:0]**：时间片计数器 (Time quanta counter)

在校准字段捕获的时间片数（32 位或 64 位）。仅当时钟校准单元处于 Precision_Calibrated 状态时，这些位才有效。

位 17:0 **OCPC[17:0]**：振荡器时钟周期计数器 (Oscillator clock period counter)

在校准字段捕获的振荡器时钟周期数（32 或 64 位）。仅当时钟校准单元处于 Precision_Calibrated 状态时，这些位才有效。

53.7.4 校准看门狗寄存器 (FDCAN_CCU_CWD)

Calibration watchdog register

偏移地址：0x000C

复位值：0x0000 0000

当校准 FSM 处于 Not_Calibrated 状态 (FDCAN_CCU_CSTAT.CALS = 00) 时，校准看门狗会在第一个下降沿后启动。在该状态下，校准看门狗会监控接收到的消息。如果在校准看门狗递减计数到 0 之前未接收到任何消息，校准 FSM 会保持 Not_Calibrated 状态 (FDCAN_CCU_CSTAT.CALS = 00)，计数器会重新载入 FDCAN_RWD.WDC，下一下降沿之后，基本校准会重新开始。

当校准看门狗处于 Basic_Calibrated 状态时 (FDCAN_CCU_CSTAT.CALS = 01)，每次接收到消息，校准看门狗都会重新启动。如果在校准看门狗递减计数到 0 之前未接收到任何消息，校准 FSM 会恢复 Not_Calibrated 状态 (FDCAN_CCU_CSTAT.CALS = 00)，计数器会重新载入 FDCAN_RWD.WDC，下一下降沿之后，基本校准会重新开始。

接收到晶振消息时，校准 FSM 会进入 Precision_Calibrated 状态 (FDCAN_CCU_CSTAT.CALS = 10)，校准看门狗会重新启动。在该状态下，校准看门狗会监控接收到的晶振消息。如果连接的 TTCAN 在校准看门狗递减计数到 0 之前未接收到任何由晶振控制的节点发送的消息，校准 FSM 会切换回 Basic_Calibrated 状态 (FDCAN_CCU_CSTAT.CALS = 01)。当连接的 TTCAN 上的 CAN 协议引擎启动，也就是 INIT 位复位时，信号有效。

校准看门狗事件也会将中断标志 FDCAN_CCU_IR.CWE 置 1。如果中断线通过 FDCAN_CCU_IE.CWEE 使能，则会被激活（设为高电平）。中断标志 FDCAN_CCU_IR.CWE 复位之前，中断线会保持有效状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDV[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **WDV[15:0]**：看门狗值 (Watchdog value)
实际校准看门狗计数器值。

位 15:0 **WDC[15:0]**：WDC
看门狗配置 (Watchdog configuration)
消息 RAM 看门狗计数器的起始值。如果使用复位值 00，计数器会禁止。
仅当 FDCAN 控制位 FDCAN_CCCR.CCE = 1 且 FDCAN_CCCR.INIT = 1 时，才能由用户对标有“P = 受写保护”的寄存器/位进行写访问。

53.7.5 时钟校准单元中断寄存器 (CCU_IR)

Clock calibration unit interrupt register

如果检测到其中一个列出的条件（边沿有效），标志会置 1。必须由用户手动清零，否则将一直保持置位。通过向对应位位置写入 1 将标志清零。写入 0 不会起任何作用。硬复位会将寄存器清零。FDCAN_CCU_IE 的配置控制着是否生成中断。

偏移地址：0x0010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSC	CWE
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **CSC**：校准状态已更改 (Calibration state changed)
0：校准状态未更改
1：校准状态已更改

位 0 **CWE**：校准看门狗事件 (Calibration watchdog event)
0：无校准看门狗事件
1：发生校准看门狗事件

53.7.6 时钟校准单元中断使能寄存器 (FDCAN_CCU_IE)

Clock calibration unit interrupt enable register

偏移地址：0x0014

复位值：0x0000 0000

CU 中断使能寄存器中的设置决定了 CU 中断寄存器中哪些状态更改将标示到中断线上。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSCE	CWEE
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **CSCE**：校准状态更改使能 (Calibration state changed enable)
0：禁止中断
1：使能中断

位 0 **CWEE**：校准看门狗事件使能 (Calibration watchdog event enable)
0：禁止中断
1：使能中断

53.7.7 CCU 寄存器映射

表 504. CCU 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	FDCAN_CCU_CREL	REL[3:0]				STEP[3:0]				SUBSTEP[3:0]				YEAR[3:0]				MON[7:0]								DAY[7:0]							
	Reset value	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0	0	0
0x0004	FDCAN_CCU_CCFG	SWR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CDIV[3:0]				OCPM[7:0]								CFL	BCC	Res	TQBT[4:0]				
	Reset value	0												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0008	FDCAN_CCU_CSTAT	CALS[1:0]		Res	TQC[10:0]											OCPC[17:0]																	
	Reset value	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 504. CCU 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000C	FDCAN_CCU_CWD	WDV[15:0]																WDC[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0010	FDCAN_CCU_IR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSC	CWE
	Reset value																															0	0
0x0014	FDCAN_CCU_IE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSCE	CWEE
	Reset value																															0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

54 USB on-the-go 高速 (OTG)

54.1 简介

Portions Copyright (c) Synopsys, Inc. 保留所有权利。使用须经许可。

本节介绍了 OTG 控制器的架构和编程模型。

本节中使用以下缩写：

FS	全速
LS	低速
HS	高速
MAC	介质访问控制器
OTG	On-the-go
PFC	数据包 FIFO 控制器
PHY	物理层
USB	通用串行总线
UTMI	USB 2.0 收发器宏单元接口 (UTMI)
LPM	链路层电源管理
HNP	主机协商协议
SRP	会话请求协议

参考文档如下：

- USB On-The-Go 补充标准，第 2.0 版
- 通用串行总线规范第 2.0 版
- 基于 USB 2.0 规范的 USB 2.0 链路层电源管理补充工程变更通知，2007 年 7 月 16 日
- USB 2.0 ECN 勘误表：链路层电源管理 (LPM)——2007 年 7 月

USB OTG 是一款双角色设备 (DRD) 控制器，同时支持从机功能和主机功能，完全符合 *USB 2.0 规范的 On-The-Go 补充标准*。此外，该控制器也可配置为“仅主机”模式或“仅从机”模式，完全符合 *USB 2.0 规范*。OTG 支持下面的*表 505：支持的 OTG 速度*中定义的速度。USB OTG 支持 HNP 和 SRP。OTG 模式下需要的唯一外部设备是提供 V_{BUS} 的电荷泵。

表 505. 支持的 OTG 速度

-	HS (480 Mb/s)	FS (12 Mb/s)	LS (1.5 Mb/s)
主机模式	X	X	X
从机模式	X	X	-

54.2 OTG 主要特性

主要特性可分为三类：通用特性、主机模式特性和从机模式特性。

54.2.1 通用特性

OTG 接口的通用特性如下：

- 经 USB-IF 认证，符合通用串行总线规范第 2.0 版
- OTG支持以下 PHY 接口：
  - 用于内部 HS PHY 的 UTMI 接口
- 模块内嵌的 PHY 还完全支持定义在标准规范 OTG 补充第 2.0 版中的 OTG 协议
  - 支持 A-B 设备识别（ID 线）
  - 支持主机协商协议 (HNP) 和会话请求协议 (SRP)
  - 允许主机关闭 V_{BUS} 以在 OTG 应用中节省电池电量
  - 支持通过内部比较器对 V_{BUS} 电平采取 OTG 监控
  - 支持主机到从机的角色动态切换
- 可通过软件配置为以下角色：
  - 具有 SRP 功能的 USB HS 从机（B 设备）
  - 具有 SRP 功能的 USB HS/LS 主机（A 设备）
  - USB On-The-Go 全速双角色设备
- 支持 HS SOF 和 LS Keep-alive 令牌
  - SOF 脉冲可通过 PAD 输出
  - SOF 脉冲通过内部连接到定时器 (TIMx)
  - 可配置的帧周期
  - 可配置的帧结束中断
- OTG 内嵌 DMA，并可软件配置 AHB 的批量传输类型。
- 它支持基于描述符的分散/收集 DMA 控制器，适用于从机模式和主机模式。（不支持基于描述符的一致顺序 DMA）。分散/收集 DMA 操作在从机模式和主机模式下均受支持。该功能将提高从机模式同步端点的性能。请注意，在主机分散/收集 DMA 工作模式下不支持集线器（拆分传输）。拆分传输只在主机缓冲 DMA（内部 DMA）工作模式下受支持。
- 具有省电功能，例如在 USB 挂起期间停止系统、关闭数字模块内部时钟域、对 PHY 和 DFIFO 电源加以管理。
- 具有采用高级 FIFO 控制的 4 KB 专用 RAM：
  - 可将 RAM 空间划分为不同 FIFO，以便灵活有效地使用 RAM
  - 每个 FIFO 可存储多个数据包
  - 动态分配存储区
  - FIFO 大小可配置为非 2 的幂次方值，以便连续使用存储单元
- 一帧 (1 ms) 之内可以无需要应用程序干预，以达到最大 USB 带宽。

54.2.2 主机模式特性

OTG 接口在主机模式下具有以下主要特性和要求：

- 通过外部电荷泵生成 V_{BUS} 电压。
- 多达 16 个主机通道（又称之为管道）：每个通道都可以动态实现重新配置，可支持任何类型的 USB 传输。
- 内置硬件调度器可：
  - 在周期性硬件队列中存储多达 16 个中断加同步传输请求
  - 在非周期性硬件队列中存储多达 16 个控制加批量传输请求
- 管理一个共享 Rx FIFO、一个周期性传输 Tx FIFO 和一个非周期性传输 Tx FIFO，以有效使用 USB 数据 RAM。

54.2.3 从机模式特性

OTG 接口在从机模式下具有以下特性：

- 1 个双向控制端点 0
- 8 个 IN 端点 (EP)，可配置为支持批量传输、中断传输或同步传输
- 8 个 OUT 端点，可配置为支持批量传输、中断传输或同步传输
- 管理一个共享 Rx FIFO 和一个 Tx-OUT FIFO，以高效使用 USB 数据 RAM
- 管理多达 9 个专用 Tx-IN FIFO（分别用于每个使能的 IN EP），以降低应用程序负荷
- 支持软断开功能。

54.3 OTG 实现

表 506. OTG 实现(1)

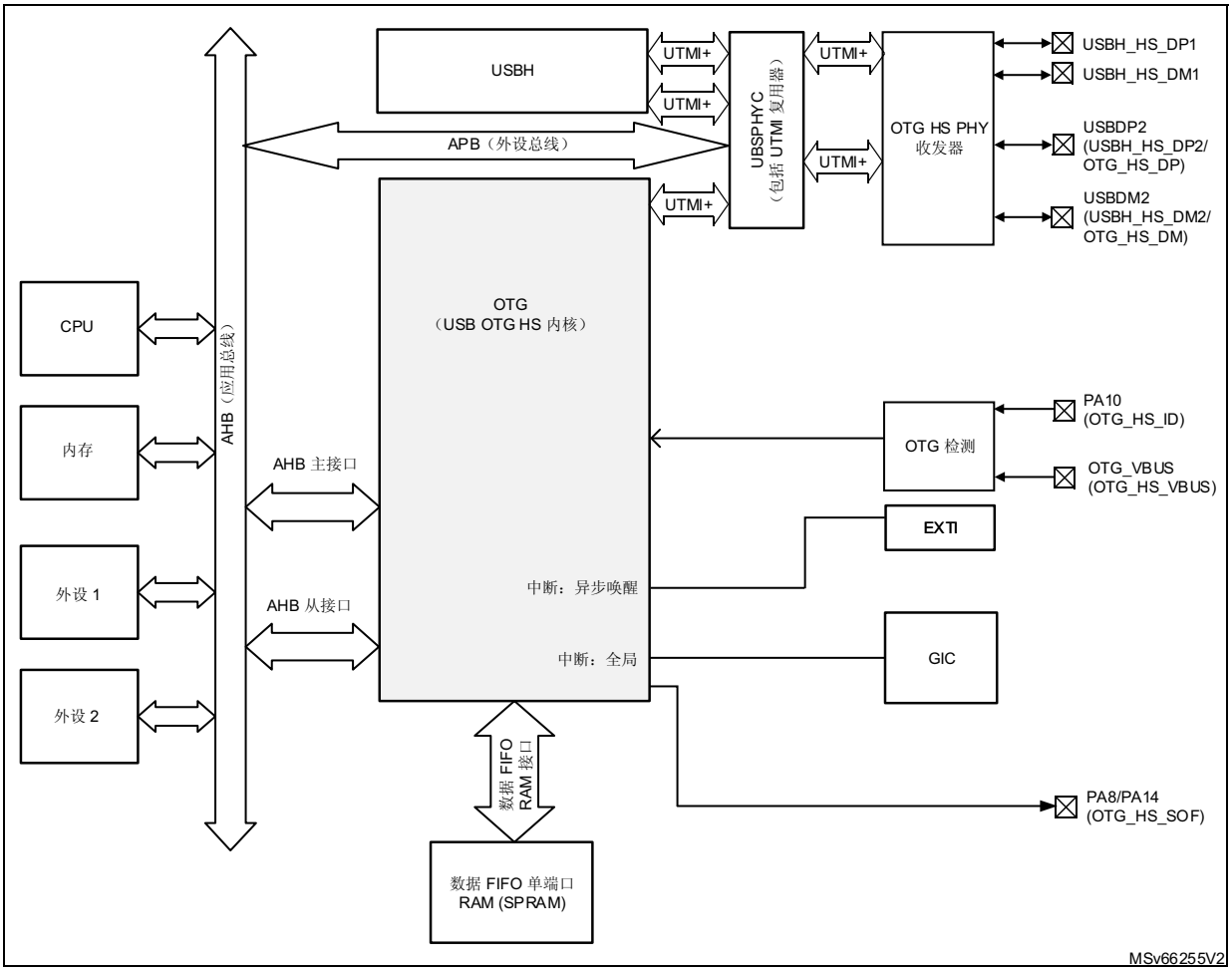
USB 特性	适用于 STM32MP13xx 的 OTG
设备双向端点（包括 EP0）	9
主机模式通道	16
专用 SRAM 的大小	4 KB
USB 2.0 链路层电源管理 (LPM) 支持	X
支持 OTG 版本	2.0
连接检测协议 (ADP) 支持	-
电池充电检测 (BCD) 支持	-
ULPI 可通过多路复用连接到主 IO	-
集成 PHY	HS
分散/收集 DMA	X
EP1 的专用中断线	-

1. “X”表示支持，“-”表示不支持，“FS”表示在 FS 模式下支持，“HS”表示在 HS 模式下支持。

54.4 OTG 功能描述

54.4.1 OTG 框图

图 681. 高速 OTG 模块框图



54.4.2 OTG 引脚和内部信号

表 507. OTG 输入/输出引脚

信号名称	信号类型	说明
OTG_HS_DP	数字输入/输出	USB OTG D+ 线
OTG_HS_DM	数字输入/输出	USB OTG D- 线
OTG_HS_ID	数字输入	USB OTG ID
OTG_HS_VBUS	模拟输入	USB OTG VBUS
OTG_HS_SOF	数字输出	USB OTG 帧起始位（可见）

表 508. OTG 输入/输出信号

信号名称	信号类型	说明
usb_sof	数字输出	USB OTG 帧起始事件（用于片上外设）
usb_wkup	数字输出	USB OTG 唤醒事件输出
usb_gbl_it	数字输出	USB OTG 全局中断

54.4.3 OTG 内核

OTG 通过复位和时钟控制器 (RCC) 接收 60 MHz 时钟。该时钟通常在与 HS PHY 关联的 PLL 中生成，并在 RCC 中使能。该时钟用于在高速 (480 Mbit/s) 通信时驱动 60 MHz 时钟域，必须在配置 OTG 模块前使能。

CPU 通过 AHB 外设总线对 OTG 模块寄存器进行读写操作，通过*第 54.12 节：OTG 中断*中所述的 USB OTG 中断线接收 USB 事件通知。

CPU 通过向特定的 OTG 单元（压栈寄存器）写入 32 位字来向 USB 提交数据。数据随即自动存储到 USB 数据 RAM 中配置的数据发送 FIFO 中。每个 IN 端点（从机模式）或 OUT 通道（主机模式）都有一个 Tx-FIFO 压栈寄存器。

CPU 从特定的 OTG 地址（出栈寄存器）读取 32 位字，以读取来自 USB 的数据。数据随即从在 4 KB USB 数据 RAM 内配置的共享 Rx-FIFO 中弹出。每个 OUT 端点或 IN 通道都有一个 Rx FIFO 出栈寄存器。

USB 协议层通过串行接口引擎 (SIE) 驱动，并通过片上物理层 (PHY) 中的收发器模块经由 USB 进行数据的串行通信。

小心： 为确保 USB OTG 模块正常工作，AHB 频率应大于 30 MHz。

54.4.4 OTG 检测

此外，OTG 还使用以下功能：

- 集成 ID 上拉电阻，用于对 ID 线进行采样，以便识别 A/B 设备。
- 带滞回功能的 V_{BUS} 感应比较器，用于检测 V_{BUS} 有效、A-B 会话有效和会话端电压阈值。执行 USB 操作期间，这些比较器用于驱动会话请求协议 (SRP)、检测会话的有效启动和结束条件，以及持续监视 V_{BUS} 供电情况。

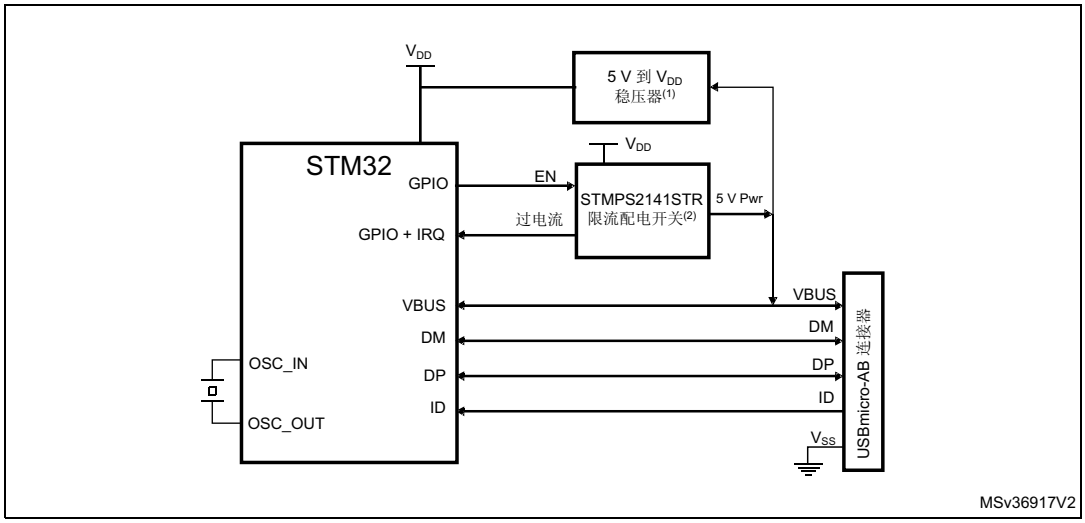
54.4.5 连接到 OTG_HS 的高速 OTG PHY

注： 请参见实现表以确定是否嵌入了 HS PHY。

USB OTG 模块内部包含一个 UTMI 接口，该接口连接到嵌入式 HS PHY（请参见*第 54.4.1 节：OTG 框图*）。

54.5 OTG 双角色设备 (DRD)

图 682. OTG A-B 设备连接



1. 只有在构建由 VBUS 供电的设备时才需要外部调压器。
2. 只有在应用必须支持由 VBUS 供电的设备时才需要 STMPS2141STR。如果应用电路板提供 5 V 电源，则可以使用基本电源开关。

54.5.1 ID 线检测

采取主机还是从机（默认设置）角色取决于 ID 输入引脚的电平。插入 USB 电缆时可根据是 MicroA 还是 MicroB 插头连接到 micro-AB 插座来确定 ID 线状态。

- 如果 USB 电缆的 B 端连入，其 ID 线悬空，则由于设备在 ID 线上的集成上拉电阻设备将检测到 ID 高电平并确认采取默认的从机角色。在此配置中，OTG 符合“USB2.0 On-The-Go 规范第 2.0 版补充标准中第 4.2.4 章节 ID 引脚”中所述的 FSM 标准。
- 如果 USB 电缆的 A 端连入，其 ID 线接地，则 OTG 将发出 ID 线状态更改中断（OTG_GINTSTS 中的 CIDSCHG 位）以告知应用程序初始化主机，并自动切换为主机角色。在此配置中，OTG 符合“USB2.0 On-The-Go 规范第 2.0 版补充标准中第 4.2.4 章节 ID 引脚”中所述的 FSM 标准。

54.5.2 HNP 双角色设备

全局 USB 配置寄存器中的 HNP 使能位（OTG_ GUSBCFG 中的 HNPCAP 位）可使 OTG 模块根据主机协商协议 (HNP) 动态切换角色，例如从 A 主机切换为 A 从机（反之亦然），或者从 B 从机切换为 B 主机（反之亦然）。通过全局 OTG 控制和状态寄存器中的连接器 ID 状态位（OTG_GOTGCTL 中的 CIDSTS 位）及全局中断和状态寄存器中的当前工作模式位（OTG_GINTSTS 中的 CMOD 位）二者的组合值可读取设备当前状态。

*第 54.15 节：OTG 编程模型*详细介绍了 HNP 编程模型。

54.5.3 SRP 双角色设备

全局 USB 配置寄存器中的 SRP 使能位（OTG_GUSBCFG 中的 SRPCAP 位）可使 OTG 模块关闭 V_{BUS} 供电，为 A 设备节省电能。注意，无论 OTG 采取主机角色还是从机角色，A 设备将始终负责 V_{BUS} 的提供。

*第 54.15 节：OTG 编程模型*详细介绍了 SRP A/B 设备编程模型。

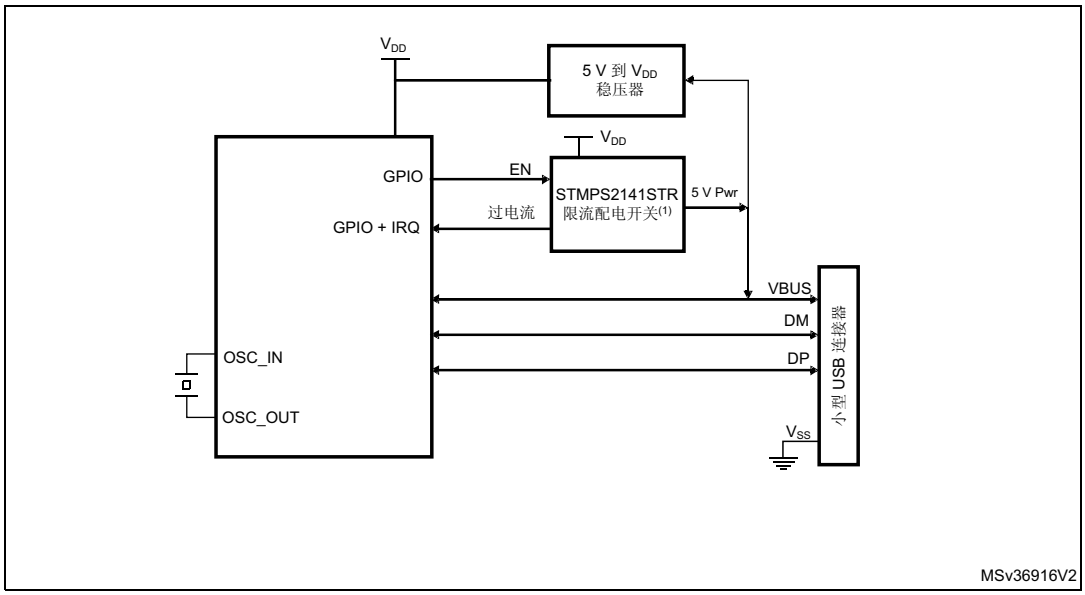
54.6 OTG 作为 USB 设备

本节介绍了 OTG 在 USB 从机模式下所具有的功能。在以下情形下，OTG 用作 USB 设备：

- OTG B 设备
  - OTG B 设备插入 USB 电缆 B 端时的默认状态
- OTG A 设备
  - HNP 将 OTG 切换到其设备角色后的 OTG A 设备状态
- B 设备
  - 如果 ID 线有连接，设备与 USB 电缆的 B 端相连，并且全局 USB 配置寄存器中的 HNP 功能位（OTG_GUSBCFG 中的 HNPCAP 位）清零。
- 纯设备
  - 将*第 54.14.4 节：OTG USB 配置寄存器 (OTG_GUSBCFG)* 中的强制从机模式位 (FDMOD) 置 1，从而将 OTG 模块强制用作纯 USB 设备。这种情况下，即使 USB 连接器上存在 ID 线，也会将该 ID 线忽略。

注： *要在 B 设备或仅作设备配置情形下构建总线供电的设备方案，需要添加一个外部调压器，用于从 V_{BUS} 生成所需电源。*

图 683. OTG 纯设备的连接



1. 使用调压器构建总线供电设备。

54.6.1 支持 SRP 功能的 USB 设备

全局 USB 配置寄存器中的 SRP 功能位（OTG_GUSBCFG 中的 SRPCAP 位）可使 OTG 支持会话请求协议 (SRP)。这样一来，远程 A 设备便可以在 USB 会话挂起时，通过关闭 V_{BUS} 来节省电能。

*B 设备会话请求协议*一节详细介绍了 SRP 设备的编程模型。

54.6.2 USB 设备状态

供电状态

V_{BUS} 输入检测到 B 会话有效电压，就会使 USB 设备进入供电状态（请参见 USB2.0 第 9.1 部分）。然后，OTG 自动连接 DP 上拉电阻，发出全速设备与主机相连的信号并生成会话请求中断（OTG_GINTSTS 中的 SRQINT 位），指示进入供电状态。

此外，V_{BUS} 输入还可确保主机在 USB 操作期间提供有效的 V_{BUS} 电平。如果检测到 V_{BUS} 降至 B 会话有效电压以下（例如，因电源干扰或主机端口关闭引发），OTG 将自动断开连接并生成检测到会话结束中断（OTG_GOTGINT 中的 SEDET 位），指示 OTG 已退出供电状态。

供电状态下，OTG 期望收到来自主机的复位信号。其他 USB 操作则无法执行。收到复位信号后，立即生成检测到复位中断（OTG_GINTSTS 中的 USBRST）。复位信号结束后，将生成枚举完成中断（OTG_GINTSTS 中的 ENUMDNE 位），OTG 随即进入默认状态。

软断开

供电状态可借助软断开功能通过软件退出。将设备控制寄存器中的软断开位（OTG_DCTL 中的 SDIS 位）置 1 即可移除 DP 上拉电阻，此时尽管没有从主机端口实际拔出 USB 电缆，但主机端仍会发生设备断开检测中断。

默认状态

默认状态下，OTG 期望从主机收到 SET_ADDRESS 命令。其他 USB 操作则无法执行。当 USB 上解码出有效 SET_ADDRESS 命令时，应用程序会将相应的数值写入设备配置寄存器中的设备地址位字段（OTG_DCFG 中的 DAD 位）。OTG 随即进入地址状态，并准备好以所配置的 USB 地址对主机事务进行应答。

挂起状态

OTG 设备持续监视 USB 活动。在 USB 空闲时间达到 3 ms 后，将发出早期挂起中断（OTG_GINTSTS 中的 ESUSP 位），并在 3 ms 后由挂起中断（OTG_GINTSTS 中的 USBSUSP 位）确认设备进入挂起状态。然后，设备状态寄存器中的设备挂起位（OTG_DSTS 中的 SUSPSTS 位）自动置 1，OTG 随即进入挂起状态。

可通过设备本身退出挂起状态。这种情况下，应用程序会将设备控制寄存器中的远程唤醒信号位（OTG_DCTL 中的 RWUSIG 位）置 1，并在 1 ms 到 15 ms 后将其清零。

但若设备检测到主机发出的恢复信号时，将生成恢复中断（OTG_GINTSTS 中的 WKUPINT 位），设备挂起位自动清零。

54.6.3 USB 设备端点

OTG 模块实现了以下 USB 端点：

- 控制端点 0：
  - 双向且仅处理控制消息
  - 使用一组单独的寄存器来处理 IN 和 OUT 事务
  - 专用控制 (OTG_DIEPCTL0/OTG_DOEPCTL0) 寄存器、传输配置 (OTG_DIEPTSIZ0/OTG_DOEPTSIZ0) 寄存器和状态中断 (OTG_DIEPINT0/OTG_DOEPINT0) 寄存器。控制和传输大小寄存器中可用的位组与其他端点中稍有不同
- 8 个 IN 端点
  - 每个端点都可配置为支持同步传输、批量传输或中断传输类型
  - 每个端点都有专用控制 (OTG_DIEPCTL*x*) 寄存器、传输配置 (OTG_DIEPTSIZ*x*) 寄存器和状态中断 (OTG_DIEPINT*x*) 寄存器
  - 设备 IN 端点通用中断屏蔽寄存器 (OTG_DIEPMSK) 可用于使能/禁止所有 IN 端点（包括 EP0）上的同一类端点中断源
  - 支持未完成的同步 IN 传输中断（OTG_GINTSTS 中的 IISOIXFR 位），该中断将在当前帧中至少有一个同步 IN 端点上的传输未完成时触发。该中断和周期性帧结束中断 (OTG_GINTSTS/EOPF) 一起触发。
- 8 个 OUT 端点
  - 每个端点都可配置为支持同步传输、批量传输或中断传输类型
  - 每个端点都有专用控制 (OTG_DOEPCTL*x*) 寄存器、传输配置 (OTG_DOEPTSIZ*x*) 寄存器和状态中断 (OTG_DOEPINT*x*) 寄存器
  - 设备 OUT 端点通用中断屏蔽寄存器 (OTG_DOEPMSK) 可用于使能/禁止所有 OUT 端点（包括 EP0）上的同一类端点中断源
  - 支持未完成的同步 OUT 传输中断（OTG_GINTSTS 中的 INCOMPISOOUT 位），该中断将在当前帧中至少有一个同步 OUT 端点上的传输未完成时触发。该中断和周期性帧结束中断 (OTG_GINTSTS/EOPF) 一起触发。

端点控制

- 应用程序可通过设备端点 *x* IN/OUT 控制寄存器 (OTG_DIEPCTL*x*/OTG_DOEPCTL*x*) 对端点采取以下控制：
  - 端点使能/禁止
  - 在当前配置下激活端点
  - 设置 USB 传输类型（同步、批量和中断）
  - 设置支持的数据包大小
  - 设置与 IN 端点相关的 Tx FIFO 编号
  - 设置希望收到的或发送时要使用到的 data0/data1 PID（仅限批量/中断传输）
  - 设置接收或发送事务时所对应的奇/偶帧（仅限同步传输）
  - 可以设置 NAK 位，从而不论此时 FIFO 的状态如何，都对主机的请求回复 NAK
  - 可以设置 STALL 位，使得主机对该端点的令牌都被硬件回复 STALL
  - 可以将 OUT 端点设置为侦听模式，即对接收到的数据不进行 CRC 检查

端点传输

设备端点 *x* 传输大小寄存器 (OTG_DIEPTSIZ*x*/OTG_DOEPTSIZ*x*) 允许应用程序对传输大小参数进行编程并读取传输状态。必须在端点控制寄存器中的端点使能位置 1 之前完成对此寄

存器的设置。使能端点后，这些位字段立即变为只读状态，同时 OTG 模块根据当前传输状态对这些位字段进行更新。

可对以下传输参数进行编程：

- 以字节为单位的传输大小
- 构成整个传输的数据包个数

端点状态/中断

设备端点 *x* 中断寄存器 (OTG_DIEPINT*x*/OTG_DOPEPINT*x*) 指示端点在出现 USB 和 AHB 相关事件时的状态。当模块中断寄存器中的 OUT 端点中断位或 IN 端点中断位（分别为 OTG_GINTSTS 中的 OEPINT 位或 OTG_GINTSTS 中的 IEPINT 位）置 1 时，应用程序必须读取这些寄存器以获得详细信息。在应用程序读取这些寄存器之前，必须先读取设备全体端点中断 (OTG_DAINT) 寄存器，以获取设备端点 *x* 中断寄存器的端点编号。应用程序必须将此寄存器中的相应位清零，才能将 OTG_DAINT 和 OTG_GINTSTS 寄存器中的相应位清零

模块提供以下状态检查和中断产生功能：

- 传输完成中断，指示应用程序 (AHB) 和 USB 端均已完成数据传输
- Setup 阶段已完成（仅针对控制传输类型的 OUT 端点）
- 相关的发送 FIFO 为半空或全空状态（IN 端点）
- NAK 应答已发送到主机（仅针对同步传输的 IN 端点）
- Tx FIFO 为空时接收到 IN 令牌（仅针对批量和中断传输类型的 IN 端点）
- 尚未使能端点时接收到 OUT 令牌
- 检测到串扰错误
- 应用程序关闭端点生效
- 应用程序对端点设置 NAK 生效（仅针对同步传输类型的 IN 端点）
- 接收到 3 个以上连续 setup 数据包（仅针对控制类型的 OUT 端点）
- 检测到超时状况（仅针对控制传输类型的 IN 端点）
- 同步传输类型的数据包未产生中断而丢失

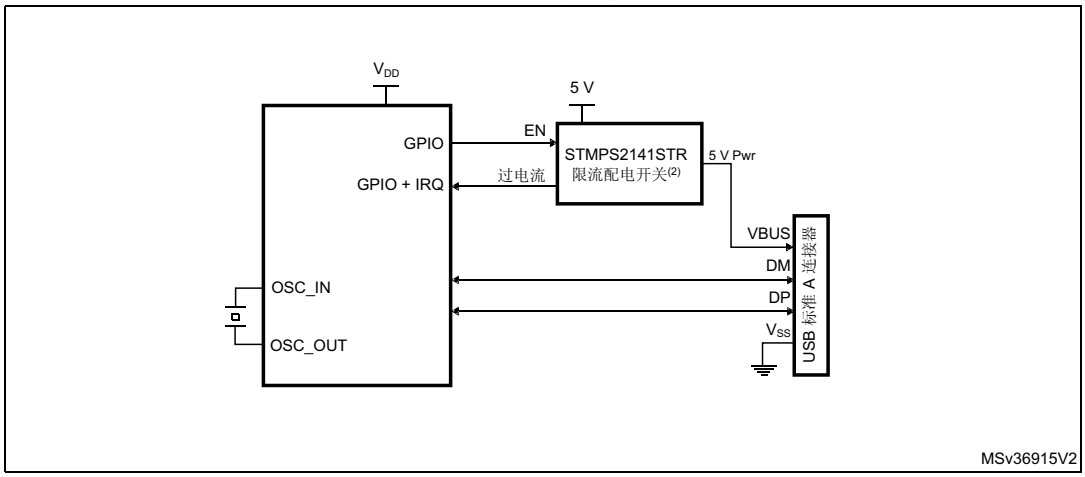
54.7 OTG 作为 USB 主机

本节介绍了 OTG 在 USB 主机模式下所具有的功能。在以下情形下，OTG 用作 USB 主机：

- OTG A 主机
  - OTG A 设备在插入 USB 电缆 A 端时的默认状态
- OTG B 主机
  - OTG B 设备被 HNP 切换为主机角色后的状态
- A 设备
  - 如果 ID 线有连接，设备与 USB 电缆的 A 端相连，全局 USB 配置寄存器中的 HNP 功能位（OTG_GUSBCFG 中的 HNPCAP 位）清零。DP/DM 线上的集成下拉电阻自动使能。
- 仅作主机
  - *OTG USB 配置寄存器 (OTG_GUSBCFG)* 中的强制主机模式位 (FHMOD) 将 OTG 模块强制用作纯 USB 主机。这种情况下，即使 USB 连接器上存在 ID 线，模块也会忽略 ID 线上的电平。DP/DM 线上的集成下拉电阻自动使能。

注： *微控制器不能输出 5 V 以提供 V_{BUS}。为此，必须在微控制器以外添加电荷泵或电源开关（如果应用电路板提供 5 V 电源）来驱动 5 V V_{BUS} 线。外部电荷泵可通过任何 GPIO 输出驱动。OTG A 主机、A 设备和仅作主机配置都需要使用电荷泵。*

图 684. OTG 仅做主机的连接



1. V_{DD} 范围介于 2 V 到 3.6 V 之间。

54.7.1 支持 SRP 功能的 USB 主机

全局 USB 配置寄存器中的 SRP 使能位（OTG_GUSBCFG 中的 SRPCAP 位）可提供 SRP 支持。使能 SRP 功能后，主机可在 USB 会话挂起时通过关闭 V_{BUS} 电源来节省电能。

*A 设备会话请求协议*一节详细介绍了 SRP 主机的编程模型。

54.7.2 USB 主机状态

给主机端口供电

微控制器不能输出 5 V 以提供 V_{BUS}。为此，必须在微控制器以外添加电荷泵或电源开关（如果应用电路板提供 5 V 电源）来驱动 5 V V_{BUS} 线。外部电荷泵可通过任何 GPIO 输出驱动，或者通过连接至外部 PMIC（电源管理 IC）的 I^2C 接口驱动。当应用程序确定控制 V_{BUS}，还必须将主机端口控制和状态寄存器中的端口电源位（OTG_HPRT 中的 PPWR 位）置 1。

V_{BUS} 有效

使能 HNP 或 SRP 后，必须将 VBUS 感应引脚连接到 V_{BUS}。V_{BUS} 输入可确保电荷泵在 USB 操作期间提供有效的 V_{BUS} 电平。如果 V_{BUS} 电压意外降至 V_{BUS} 有效阈值 (4.4 V) 以下，将通过会话结束检测位（OTG_GOTGINT 中的 SEDET 位）触发 OTG 中断。之后应用必须断开 V_{BUS} 电源并使端口电源位清零。

当 HNP 和 SRP 均禁止时，无需将 VBUS 感应引脚连接到 V_{BUS}，该引脚可用作 GPIO。

电荷泵过流标志也可用来防止电气损坏。将电荷泵的过流标志输出连接到任意 GPIO 输入，然后将其配置为出现有效电平时生成端口中断。过流 ISR 必须立即关闭 V_{BUS} 并清零端口电源位。

主机检测设备连接

如果使能 SRP 或 HNP，即使可以随时连接 USB 设备或 B 设备，OTG 也只有在 V_{BUS} 有效后 (5 V) 才能检测到设备的连接。当 V_{BUS} 处于有效电平且已连接远程 B 设备时，OTG 模块将发出主机端口中断信号，该中断由主机端口控制和状态寄存器中的设备连接位（OTG_HS_HPRT 中的 PCDET 位）触发。

在 HNP 和 SRP 同时关闭的情况下，USB 设备或 B 设备将在连接后立即被检测到。OTG 模块将发出主机端口中断信号，该中断由主机端口控制和状态寄存器中的设备连接位（OTG_HPRT 中的 PCDET 位）触发。

主机检测设备断开

设备断开事件将触发断开连接检测中断（OTG_GINTSTS 中的 DISCINT 位）。

主机枚举

检测到设备连接后，若又有新的设备连接进来，主机必须通过向新的设备发送 USB 复位和配置命令来启动枚举过程。

开始驱动 USB 复位前，应用程序必须等待去抖动完成位（OTG_GOTGINT 中的 DBCDNE 位）触发 OTG 中断，这表示由于在 DP (FS) 或 DM (LS) 上连接上拉电阻而发生电气抖动之后，总线恢复稳定状态。

应用程序通过将主机端口控制和状态寄存器中的端口复位位（OTG_HPRT 中的 PRST 位）置 1，使该过程最少持续 10 ms、最多持续 20 ms，以此通过 USB 驱动 USB 复位信号（单端零）。应用程序计算这个过程的持续时间，然后将端口复位位清零。

USB 复位序列完成后，端口使能/禁止更改位（OTG_HPRT 中的 PENCHNG 位）立即触发主机端口中断，进而向应用程序发出通知，指示可从主机端口控制和状态寄存器中的端口速度位字段（OTG_HPRT 中的 PSPD）读取枚举的设备速度，以及主机已经开始驱动 SOF (FS) 或 Keep-alive 令牌 (LS)。此时主机已就绪，可通过对设备发送命令来完成对设备的枚举。

主机挂起

应用程序通过将主机端口控制和状态寄存器中的端口挂起位（OTG_HPRT 中的 PSUSP）置 1 来挂起 USB 活动。OTG 模块停止发送 SOF 并进入挂起状态。

可由远程设备的自主活动（远程唤醒）使总线退出挂起状态。这种情况下，远程唤醒信号将触发远程唤醒中断（OTG_GINTSTS 中的 WKUPINT 位），硬件把主机端口控制和状态寄存器中的端口恢复位（OTG_HPRT 中的 PRES 位）自行复位，并通过 USB 自动驱动恢复信号。应用程序必须为恢复窗口定时，然后将端口恢复位清零以退出挂起状态并重新启动 SOF。

如果由主机发起退出挂起状态，则应用程序必须将端口恢复位置 1 以启动主机端口上的恢复信号，为恢复窗口定时并最终将端口恢复位清零。

54.7.3 主机通道

OTG 模块实现了 16 主机通道。每个主机通道均可用于 USB 主机传输（USB 管道）。主机最多能同时处理 16 个传输请求。如果应用程序有 16 个以上的传输请求挂起，则在通道从之前任务释放后（即，接收到传输完成和通道停止中断后），主机控制器驱动器 (HCD) 必须为未处理的传输请求重新对通道进行分配。

每个主机通道都可配置为支持输入/输出以及周期性/非周期性事务。每个主机通道都使用专用控制 (OTG_HCCHAR*x*) 寄存器、传输配置 (OTG_HCTSIZ*x*) 寄存器状态/中断 (OTG_HCINT*x*) 寄存器以及和其相关的中断屏蔽寄存器 (OTG_HCINTMSK*x*)。

主机通道控制

- 应用程序可通过主机通道 *x* 特性寄存器 (OTG_HCCHAR*x*) 对主机通道作以下控制：
  - 通道使能/禁止
  - 编程目标 USB 设备的 HS/FS/LS 速度
  - 设置目标 USB 设备的地址
  - 设置与该通道通信的目标 USB 设备上的端点的编号
  - 设置该通道上的传输方向：IN/OUT
  - 设置该通道上的 USB 传输的类型：控制/批量/中断/同步
  - 设置与该通道通信的设备端点的最大包长
  - 设置要进行周期传输的帧：奇帧/偶帧

主机通道传输

主机通道传输大小寄存器 (OTG_HCTSIZ*x*) 允许应用程序对传输大小参数进行编程并读取传输状态。必须在主机通道特性寄存器中的通道使能位置 1 之前完成对此寄存器的设置。使能端点后，数据包计数位字段立即变为只读状态，同时 OTG 模块根据当前传输状态对该位字段进行更新。

- 可对以下传输参数进行编程：
  - 以字节为单位的传输大小
  - 构成整个传输大小的数据包个数
  - 初始数据 PID

主机通道状态/中断

主机通道 *x* 中断寄存器 (OTG_HCINT*x*) 指示通道在出现 USB 和 AHB 相关事件时的状态。当模块中断寄存器中的主机通道中断位（OTG_GINTSTS 中的 HCINT 位）置 1 时，应用程序必须读取这些寄存器以获得详细信息。在读取这些寄存器之前，应用程序必须先读取主机全体通道中断 (OTG_HAINT) 寄存器，以获取主机通道 x 中断寄存器的通道编号。应用程序必须将该寄存器中的相应位清零，才能将 OTG_HAINT 和 OTG_GINTSTS 寄存器中的对应位清零。OTG_HCINTMSKx 寄存器还提供每个通道各中断源的屏蔽位。

- 主机模块提供以下状态检查和中断产生功能：
  - 传输完成中断，指示应用程序 (AHB) 和 USB 端均已完成数据传输
  - 通道因传输完成、USB 事务错误或应用程序发出禁止命令而停止
  - 相关的发送 FIFO 为半空或全空状态（IN 端点）
  - 接收到 ACK 响应
  - 接收到 NAK 响应
  - 接收到 STALL 响应
  - 由于 CRC 校验失败、超时、位填充错误和错误的 EOP 导致 USB 事务错误
  - 串扰错误
  - 帧上溢
  - 数据同步错误

54.7.4 主机调度器

主机模块内置硬件调度器，可自主对应用程序发出的 USB 事务请求重新排序和管理。每一帧开始时，主机都先执行周期性（同步和中断）事务，然后执行非周期性（控制和批量）事务，以符合 USB 规范对同步和中断传输高优先级的保证。

主机通过请求队列（一个周期性请求队列和一个非周期请求队列）处理 USB 事务。每个请求队列最多可存储 8 个条目。每个条目代表一个应用程序发起但还未得到响应的 USB 事务请求，并存储了执行该 USB 事务所用到的 IN 或 OUT 通道的编号，以及其他相关信息。USB 事务请求在队列中的写入顺序决定了事务在 USB 接口上的执行顺序。

每一帧开始时，主机都先处理周期性请求队列，然后处理非周期性请求队列。如果当前帧结束时，计划在当前帧执行的同步或中断类型的 USB 传输事务请求仍处于挂起状态，则主机将发出未完成周期性传输中断（OTG_GINTSTS 中的 IPXFR 位）。OTG 模块全面负责对周期性和非周期性请求队列的管理。周期性发送 FIFO 和队列状态寄存器 (OTG_HPTXSTS) 与非周期性发送 FIFO 和队列状态寄存器 (OTG_HNPTXSTS) 都为只读寄存器，应用程序可使用它们来读取各请求队列的状态。其中包括：

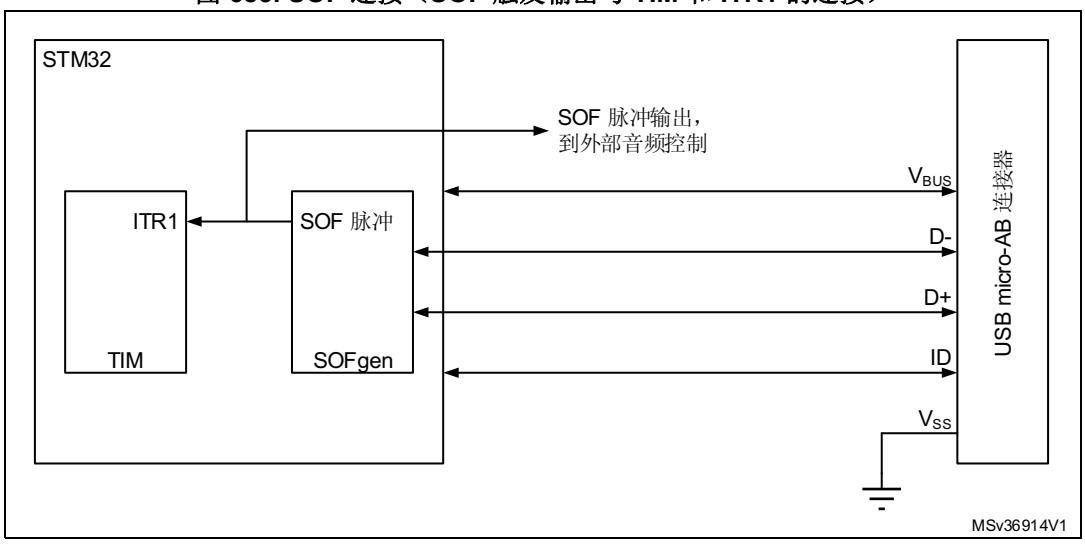
- 周期性（非周期性）请求队列中当前可用的空闲条目数（最多 8 个）
- 周期性（非周期性）Tx FIFO（OUT 事务）中当前可用的空闲空间
- IN/OUT 令牌、主机通道编号和其他状态信息。

由于每个请求队列最多可存储 8 个 USB 事务请求，因此应用程序可以把主机 USB 事务请求提前发送给调度器；实际的通信最晚会在调度器处理完已挂起的 8 个周期事务和 8 个非周期事务完成之后出现在 USB 总线上。

要向主机调度器（队列）发出事务请求，应用程序必须读取 OTG_HNPTXSTS 寄存器中的 PTXQSAV 位或 OTG_HNPTXSTS 寄存器中的 NPTQXSAV 位，确保周期性（非周期性）请求队列中至少有一个可用空间来存储当前请求。

54.8 OTG SOF 触发

图 685. SOF 连接（SOF 触发输出与 TIM 和 ITR1 的连接）



OTG 模块在主机和从机模式下都可以监视、跟踪和配置 SOF 帧并且还具备 SOF 脉冲输出连接功能。

此功能尤其适用于自适应音频时钟生成，其中音频设备需要与 PC 提供的同步音频数据流实现同步，或者主机需要根据音频设备的要求调整数据帧率。

54.8.1 主机 SOF

主机模式下，可以在主机帧间隔寄存器 (HFIR) 中对所产生的两个连续 SOF (HS/FS) 或 Keep-alive (LS) 令牌期间所出现的 PHY 时钟数进行编程，进而应用程序可对 SOF 帧周期进行控制。帧开始（OTG_GINTSTS 中的 SOF 位）时都将生成中断。当前帧编号和出现下一个 SOF 前剩余的时间应用程序在主机帧编号寄存器 (HFNUM) 中能够进行跟踪。

SOF 令牌发出的同时会产生 SOF 脉冲信号，并且宽度为 20 个 HCLK 周期。此外，SOF 脉冲还与定时器的输入触发相连，因此可通过 SOF 脉冲触发输入捕获功能、输出比较功能和定时器。

54.8.2 设备 SOF

在从机模式下，USB 每次接收到 SOF 令牌时，都将触发帧开始中断（OTG_GINTSTS 中的 SOF 位）。相应的帧编号可从设备状态寄存器（OTG_DSTS 中的 FNSOF 位）读取。还可以生成宽度为 20 个 HCLK 周期的 SOF 脉冲信号。此外，SOF 脉冲信号还在内部与 TIM 的输入触发相连，因此可通过 SOF 脉冲触发输入捕获功能、输出比较功能和定时器。

周期性帧结束中断 (OTG_GINTSTS/EOPF) 用于在经过了 80%、85%、90% 或 95% 的帧间隔时间时通知应用程序，具体取决于设备配置寄存器中的周期性帧间隔位字段（OTG_DCFG 中的 PFIVL 位）。此功能可用于确定该帧的所有同步通信是否完成。

54.9 OTG 低功耗模式

下文的*表 509* 定义了 STM32 低功耗模式及其与 OTG 的兼容性。

表 509. STM32 低功耗模式与 OTG 的兼容性

模式	说明	USB 兼容性
运行	MCU 完全激活	USB 不处于挂起状态时需要。
睡眠	USB 挂起退出可使设备退出睡眠模式。外设寄存器内容保持不变。	USB 处于挂起状态时可用。
Stop	USB 挂起退出可使设备退出停止模式。外设寄存器内容保持不变(1)。	USB 处于挂起状态时可用。
待机	掉电。在退出待机模式后，必须重新初始化外设。	与 USB 应用程序不兼容。

1. 在停止模式下，有不同的可能设置。也可能存在一些限制，请参见*第 9 章：电源控制 (PWR)* 以了解使用 OTG 的适用限制（如果存在）。

以下位和步骤降低了功耗。

OTG PHY 的功耗由通用模块配置寄存器中的两个或三个位控制，具体取决于 OTG 支持的版本。

- V_{BUS} 检测使能 (OTG_GCCFG/VBDEN)
  用于开启/关闭与 OTG 操作关联的 V_{BUS} 感应比较器。

USB 会话没有开始或设备未连接时，可以在 USB 挂起状态下使用功率降低技术。

- 停止 PHY 时钟（OTG_PCGCCTL 中的 STPPCLK 位）
  将时钟门控控制寄存器中的停止 PHY 时钟位置 1 时，会关掉收发器的大部分单元，只有负责检测异步恢复事件或远程唤醒事件的部分还保持工作状态。
- HCLK 门控（OTG_PCGCCTL 中的 GATEHCLK 位）
  将时钟门控控制寄存器中的 Gate HCLK 位置 1 时，OTG 模块内部的大多数系统时钟域均由时钟门控关闭。只有寄存器读取和写入接口保持活动状态。即使应用程序仍提供时钟输入，也会节省掉模块由于时钟信号翻转带来的动态功耗。
- USB 系统停止
  当 OTG 处于 USB 挂起状态时，应用程序可通过将 USB 系统中的所有时钟源全部关闭来显著降低总功耗。USB 系统停止可通过以下方式激活：首先将停止 PHY 时钟位置 1，然后在电源控制系统模块 (PWR) 中将系统配置为深度睡眠模式。
  OTG 模块通过对 USB 上的远程唤醒（作为主机）或恢复（作为设备）信号进行异步检测，自动重新激活系统时钟和 USB 时钟。

为了节省动态功耗，只在 USB 数据 FIFO 被 OTG 模块访问时为其提供时钟。

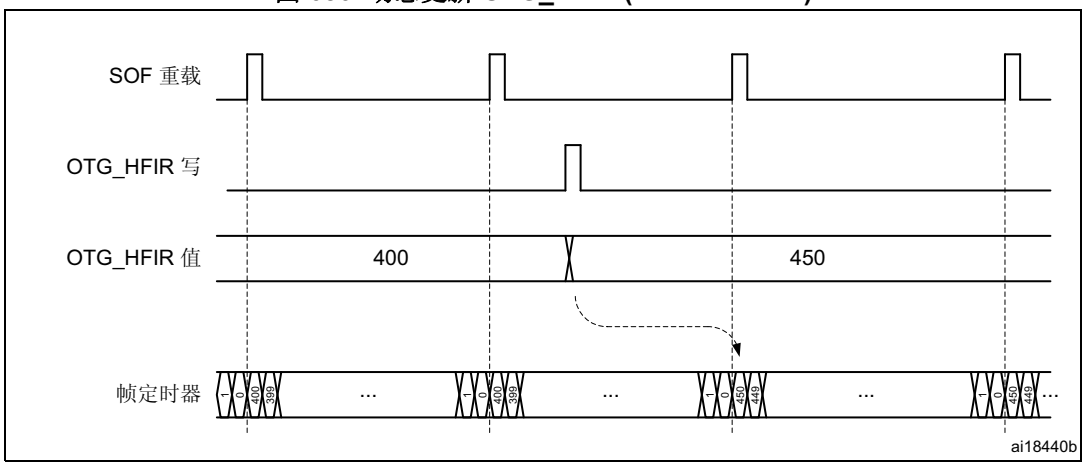
54.10 OTG 动态更新 OTG_HFIR 寄存器

主机模式下，USB 模块具有对 micro-SOF 帧周期进行动态微调的功能，能够将外部设备与 micro-SOF 帧进行同步。

如果 OTG_HS_HFIR 寄存器在当前 micro-SOF 帧内发生更改，则将在下一个帧中对 SOF 周期进行相应修正，具体说明请参见*图 686*。

对于动态更新，需要设置 RLDCTRL = 1。

图 686. 动态更新 OTG_HFIR (RLDCTRL = 1)

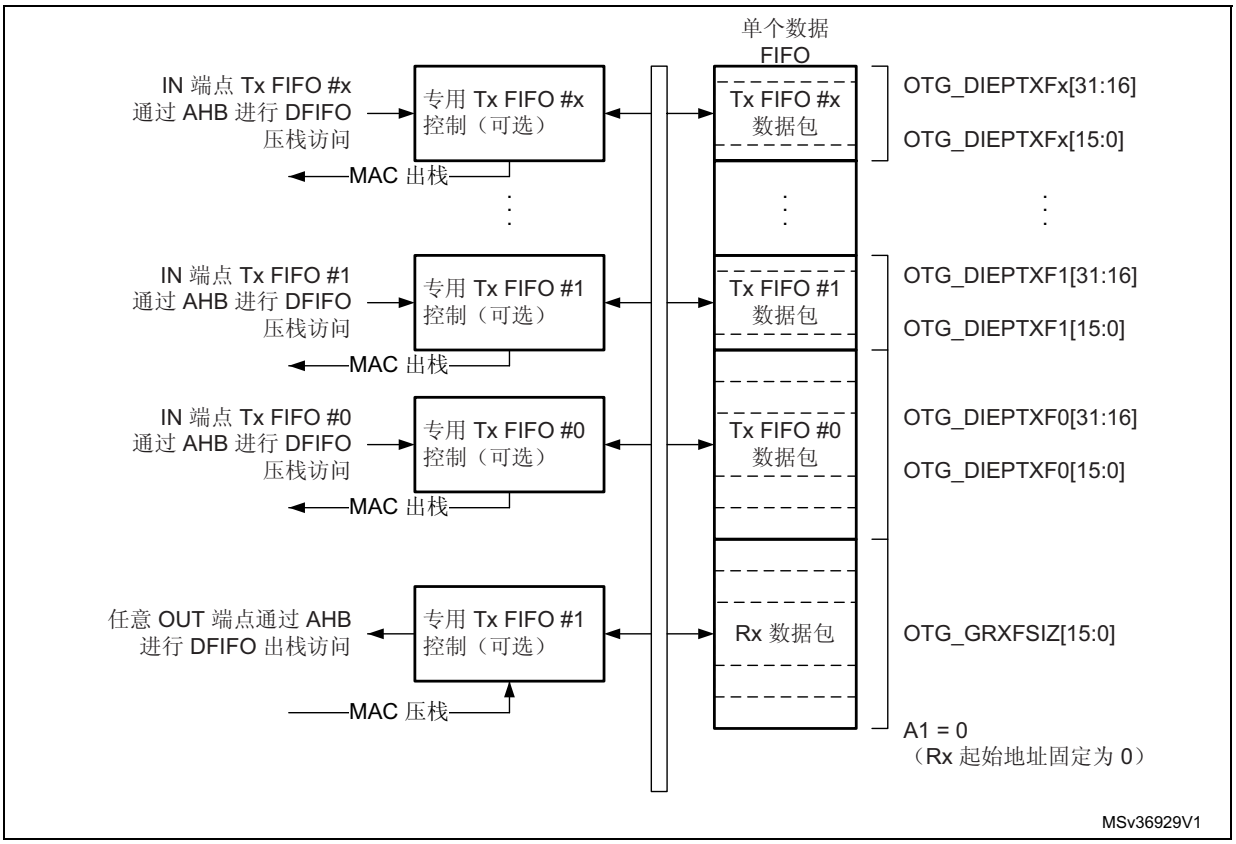


54.11 OTG 数据 FIFO

USB 系统具有 4 KB 专用 RAM，采用复杂的 FIFO 控制机制。OTG 模块中的数据包 FIFO 控制器模块将 RAM 空间划分为多个 Tx-FIFO（USB 传输前，应用程序将数据压入其中进行短暂存储）和单个 Rx FIFO（从 USB 接收到的数据被应用程序读取之前，在其中进行短暂存储）。RAM 中所构建的 FIFO 的数量与组织方式取决于设备的角色。从机模式下，为每个激活的 IN 端点配置一个 Tx FIFO。FIFO 的大小均由软件配置，以更好地满足应用要求。

54.11.1 设备 FIFO 架构

图 687. 从机模式下的 FIFO 地址映射和 AHB FIFO 访问映射



设备 Rx FIFO

OTG 设备使用单个接收 FIFO 接收发送到所有 OUT 端点的数据。只要 Rx FIFO 中有空余空间，收到的数据包就挨个填入 Rx FIFO。除了有效数据外，接收到的数据包状态（包含 OUT 端点目标编号、字节数、数据 PID 和对所接收数据的验证）也由模块进行存储。没有可用空间时，设备会回复主机事务 NACK 应答并在被寻址的端点上触发中断。接收 FIFO 的大小在接收 FIFO 大小寄存器 (OTG_GRXFSIZ) 中配置。

单个接收 FIFO 架构使得 USB 设备更高效地填充接收 RAM 缓冲区：

- 所有 OUT 端点共享同一个 RAM 缓冲区（共享 FIFO）
- OTG 模块可将主机发出的任何 OUT 通信序列填充到接收 FIFO，直到没有多余空闲空间

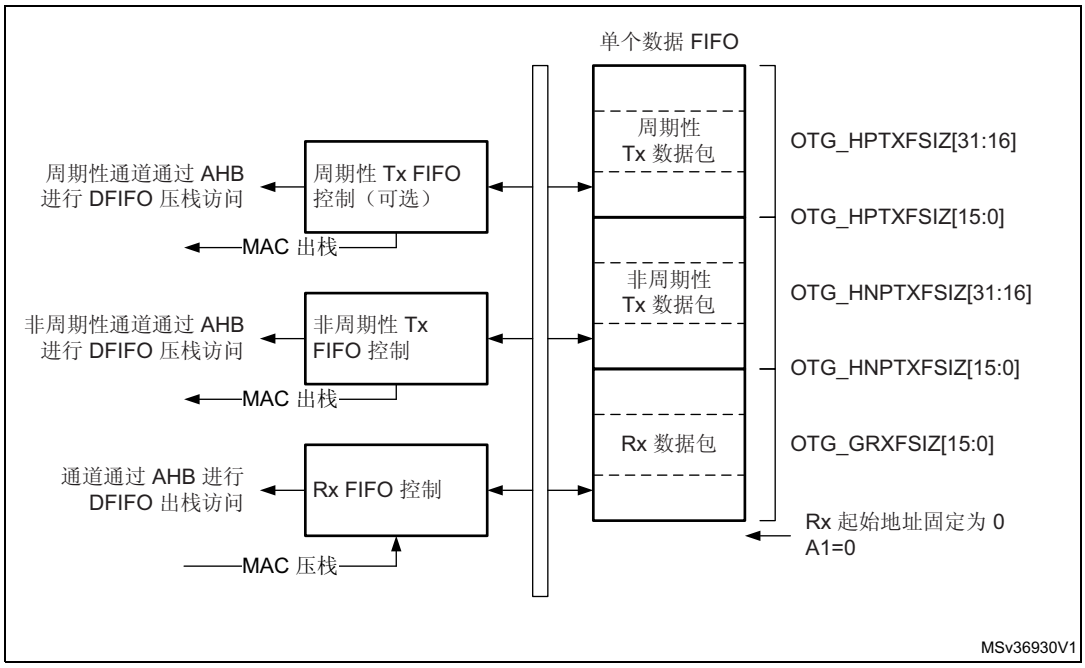
只要至少有一个数据包在 Rx FIFO 中可供读取，应用程序就会一直接收 Rx FIFO 非空中断（OTG_GINTSTS 中的 RXFLVL 位）。应用程序从接收状态读取和出栈寄存器 (OTG_GRXSTSP) 中读取数据包信息，最后通过读取与端点相关的出栈地址从接收 FIFO 读出相应数据。

设备 Tx FIFO

模块为各个 IN 端点提供了专用的 FIFO。应用程序通过端点 0 发送 FIFO 大小寄存器 (OTG_DIEPTXF0) 为 IN 端点 0 配置 FIFO 大小；通过设备 IN 端点发送 FIFOx 寄存器 (OTG_DIEPTXFx) 为 IN 端点 x 配置 FIFO 大小。

54.11.2 主机 FIFO 架构

图 688. 主机模式下的 FIFO 地址映射和 AHB FIFO 访问映射



主机 Rx FIFO

主机使用一个接收 FIFO 处理所有周期和非周期事务。FIFO 用作接收缓冲区以保存从 USB 接收到的数据（接收到的数据包的数据部分），直至这些数据传输到系统内存。只要 FIFO 中有空间，来自设备 IN 端点的数据包就接收进来并挨个存储。接收到的每个数据包的状态（包含主机目标通道、字节数、数据 PID 和对所接收数据的校验）也存储在 FIFO 中。接收 FIFO 的大小在接收 FIFO 大小寄存器 (OTG_GRXFSIZ) 中配置。

单个接收 FIFO 架构使得 USB 主机高效地填充接收数据缓冲区：

- 所有 IN 配置主机通道共享同一个 RAM 缓冲区（共享 FIFO）
- OTG 模块可将主机发出的任何 IN 通信序列带来的接收数据填充到接收 FIFO，直到没有多余空闲空间

只要至少有一个数据包在 Rx FIFO 中可供读取，应用程序就会接收 Rx FIFO 非空中断。应用程序从接收状态读取和出栈寄存器中读取数据包信息，最后从接收 FIFO 中读出数据。

主机 Tx FIFO

主机使用一个发送 FIFO 处理所有非周期（控制和批量）OUT 事务，使用另一个发送 FIFO 处理所有周期（同步和中断）OUT 事务。FIFO 用作发送缓冲区以保存要通过 USB 发送的数据（发送数据包）。周期（非周期）Tx FIFO 的大小在主机周期（非周期）发送 FIFO 大小 (OTG_HPTXFSIZ/OTG_HNPTXFSIZ) 寄存器中配置。

两个 Tx FIFO 按优先级实施操作，周期性通信的优先级较高，因此在 USB 一帧的时间内首先进行周期性通信。帧起始时，内置的主机调度器先处理周期请求队列，再处理非周期请求队列。

两个发送 FIFO 的架构使得 USB 主机能够对周期和非周期发送数据缓冲区分别进行优化管理：

- 配置为支持周期（非周期）OUT 事务的所有主机通道共享同一个 RAM 缓冲区（共享 FIFO）
- OTG 模块可将主机发出的任何 OUT 通信填充到周期性（非周期性）发送 FIFO，直到没有多余空闲空间

只要周期性 Tx FIFO 为半空或全空，OTG 模块就会发出周期性 Tx FIFO 空中断（OTG_GINTSTS 中的 PTXFE 位），具体取决于 AHB 配置寄存器中的周期性 Tx FIFO 空门限位（OTG_GAHBCFG 中的 PTXFELVL 位）的值。只要周期性 Tx FIFO 和周期性请求队列中均存在空闲空间，应用程序便可提前写入发送数据。可通过读取主机周期性发送 FIFO 和队列状态寄存器 (OTG_HPTXSTS) 来了解二者的可用空间。

只要非周期性 Tx FIFO 为半空或全空，OTG 模块就会发出非周期性 Tx FIFO 空中断（OTG_GINTSTS 中的 NPTXFE 位），具体取决于 AHB 配置寄存器中的非周期性 Tx FIFO 空门限位（OTG_GAHBCFG 中的 TXFELVL 位）。只要非周期性 Tx FIFO 和非周期性请求队列中均存在空闲空间，应用程序便可写入发送数据。可通过读取主机非周期性发送 FIFO 和队列状态寄存器 (OTG_HNPTXSTS) 来了解二者的可用空间。

54.11.3 FIFO RAM 分配

从机模式

接收 FIFO RAM 分配：应用程序必须为 SETUP 数据包分配 RAM：

- 接收 FIFO 中必须保留 10 个位置以在控制端点上接收 SETUP 数据包。模块不会向这些为 SETUP 数据包保留的位置写入任何其他数据。
- 将会为全局 OUT NAK 分配一个位置。
- 状态信息随各个接收数据包写入 FIFO。因此，必须至少为接收数据包分配（最大数据包大小/4）+ 1 的空间。如果使能了多个同步端点，则为接收连续数据包分配的空间必须至少为（最大数据包大小/4）的两倍 + 1。通常，推荐的空间为（最大数据包/4 + 1）的两倍，这样当上一个数据包向 CPU 传送时，USB 可同时接收后续的数据包。
- 传输完成状态信息和该端点收到的最后一个数据包会一起被推入 FIFO。推荐为每个 OUT 端点分配一个位置。

设备 RxFIFO =

（5 * 控制端点数量 + 8）+（（所使用的最大 USB 数据包/4）+ 用于状态信息的 1）+（2 * OUT 端点数量）+ 用于全局 NAK 的 1

示例：周期性 USB 数据包的 MPS 是 1024 个字节，非周期性 USB 数据包的 MPS 是 512 个字节。有三个 OUT 端点、三个 IN 端点、一个控制端点和三个主机通道。

则设备 RxFIFO = (5 * 1 + 8) + ((1024/4) +1) + (2 * 4) + 1 = 279

发送 FIFO RAM 分配：各个 IN 端点发送 FIFO 所需的最小 RAM 空间为该特定 IN 端点的最大数据包大小。

注： *为发送 IN 端点 FIFO 分配的空间越多，USB 的性能就越高。*

主机模式

接收 FIFO RAM 分配：

状态信息随各个接收数据包写入 FIFO。因此，必须至少为接收数据包分配（最大数据包大小/4）+ 1 的空间。如果使能了多个同步通道，则为接收连续数据包分配的空间必须至少为（最大数据包大小/4）的两倍 + 1。通常，推荐的空间为（最大数据包/4 + 1）的两倍，这样当上一个数据包向 CPU 传送时，USB 可同时接收后续的数据包。

传输完成状态信息和主机通道收到的最后一个数据包会一起被推入 FIFO。所以必须为此分配一个位置。

主机 RxFIFO =（（所用的最大 USB 数据包/4）+ 用于状态信息的 1）+ 用于传输完成的 1

示例：主机 RxFIFO = ((1024/4) + 1) + 1 = 258

发送 FIFO RAM 分配：

主机非周期性发送 FIFO 所需的最小 RAM 为所支持的所有非周期性 OUT 通道上传输的最大数据包的大小。

通常，推荐的空间为最大数据包大小的两倍，这样当 USB 正在发送当前数据包的同时，CPU 可获得下一个数据包。

非周期性 TxFIFO = 所用的最大非周期性 USB 数据包/4

示例：非周期性 TxFIFO = (512/4) = 128

主机周期性发送 FIFO 所需的最小 RAM 为所支持的所有周期性 OUT 通道上传输的最大数据包的大小。如果至少有一个同步 OUT 端点，则空间必须至少为该通道中最大数据包大小的两倍。

主机周期性 TxFIFO = 所用的最大周期性 USB 数据包/4

示例：主机周期性 TxFIFO = (1024/4) = 256

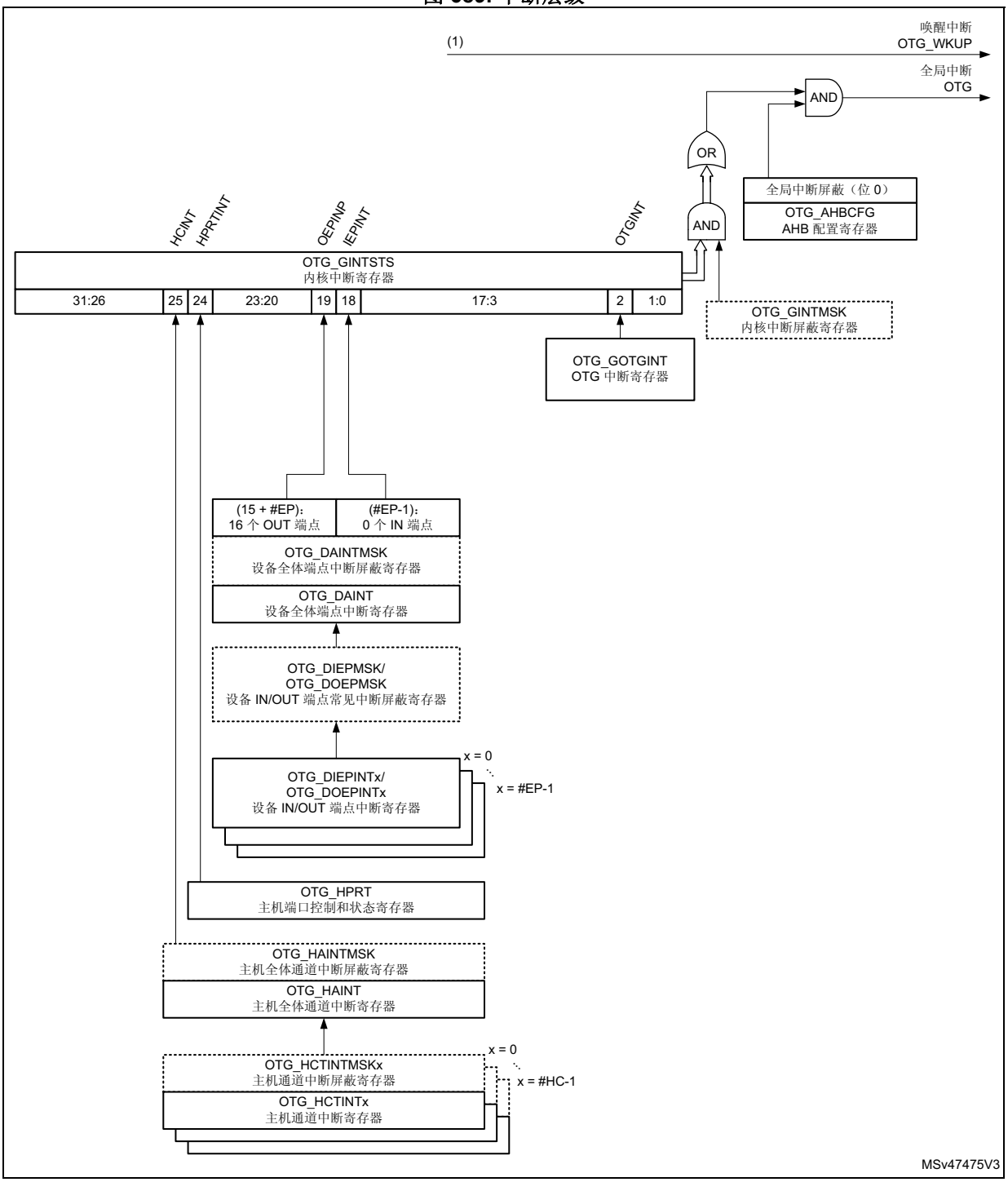
注： *为非周期性发送 FIFO 分配的空间越多，USB 的性能就越高。*

54.12 OTG 中断

当 OTG 控制器在一种模式下（从机模式或主机模式）工作时，应用程序不得以另一种角色模式访问寄存器。如果发生了非法访问，将会产生模式不匹配中断并在模块中断寄存器（OTG_GINTSTS 寄存器中的 MMIS 位）中反映。当模块从一种角色模式切换到另一种角色模式时，新工作模式下的寄存器必须重新编程为上电复位后的状态。

图 689 显示了中断层级。

图 689. 中断层级



1. 处于 L1 睡眠或 L2 挂起状态期间发生恢复条件时，OTG_WKUP 变为有效状态（高电平状态）。
2. 请参见*表 506：OTG 实现*。

54.13 OTG 控制和状态寄存器

应用程序通过 AHB 从接口对控制和状态寄存器 (CSR) 进行读写操作，以此来控制 OTG 模块。这些都是 32 位寄存器，其地址按 32 位对齐，因此只能以 32 位的方式访问 OTG 寄存器。

CSR 分为以下几类：

- 模块全局寄存器
- 主机模式寄存器
- 主机全局寄存器
- 主机端口 CSR
- 主机通道相关寄存器
- 从机模式寄存器
- 设备全局寄存器
- 设备端点相关寄存器
- 电源和时钟门控寄存器
- 数据 FIFO (DFIFO) 访问寄存器

只有模块全局寄存器、电源和时钟门控寄存器、数据 FIFO 访问寄存器以及主机端口控制和状态寄存器，可在主机模式和从机模式下进行访问。当 OTG 控制器在一种模式下（从机模式或主机模式）工作时，应用程序不得以另一种角色模式访问寄存器。如果发生了非法访问，将会产生模式不匹配中断并在模块中断寄存器（OTG_GINTSTS 寄存器中的 MMIS 位）中反映。当模块从一种角色模式切换到另一种角色模式时，新工作模式下的寄存器必须重新编程为上电复位后的状态。

54.13.1 CSR 内存映射

主机模式寄存器和从机模式寄存器占据不同的地址。所有寄存器均在 AHB 时钟域内实现。

全局 CSR 地址映射

这些寄存器在主机模式和从机模式下均可用。

表 510. 模块全局控制和状态寄存器 (CSR)

缩略语	偏移地址	寄存器名称
OTG_GOTGCTL	0x000	*第 54.14.1 节：OTG 控制和状态寄存器 (OTG_GOTGCTL)*
OTG_GOTGINT	0x004	*第 54.14.2 节：OTG 中断寄存器 (OTG_GOTGINT)*
OTG_GAHBCFG	0x008	*第 54.14.3 节：OTG AHB 配置寄存器 (OTG_GAHBCFG)*
OTG_GUSBCFG	0x00C	*第 54.14.4 节：OTG USB 配置寄存器 (OTG_GUSBCFG)*
OTG_GRSTCTL	0x010	*第 54.14.5 节：OTG 复位寄存器 (OTG_GRSTCTL)*
OTG_GINTSTS	0x014	*第 54.14.6 节：OTG 模块中断寄存器 (OTG_GINTSTS)*
OTG_GINTMSK	0x018	*第 54.14.7 节：OTG 中断屏蔽寄存器 (OTG_GINTMSK)*
OTG_GRXSTSR	0x01C	*第 54.14.8 节：OTG 接收状态调试读取寄存器 (OTG_GRXSTSR)*
		第 54.14.9 节：OTG 接收状态调试读取寄存器 [复用] (OTG_GRXSTSR)

表 510. 模块全局控制和状态寄存器 (CSR)（续）

缩略语	偏移地址	寄存器名称
OTG_GRXSTSP	0x020	*第 54.14.10 节：OTG 状态读取和弹出寄存器 (OTG_GRXSTSP)*
		第 54.14.11 节：OTG 状态读取和弹出寄存器 [复用] (OTG_GRXSTSP)
OTG_GRXFSIZ	0x024	*第 54.14.12 节：OTG 接收 FIFO 大小寄存器 (OTG_GRXFSIZ)*
OTG_HNPTXFSIZ/ OTG_DIEPTXF0(1)	0x028	*第 54.14.13 节：OTG 主机非周期性发送 FIFO 大小寄存器 (OTG_HNPTXFSIZ)/端点 0 发送 FIFO 大小 (OTG_DIEPTXF0)*
OTG_HNPTXSTS	0x02C	*第 54.14.14 节：OTG 非周期性发送 FIFO/队列状态寄存器 (OTG_HNPTXSTS)*
OTG_GCCFG	0x038	*第 54.14.15 节：OTG 通用模块配置寄存器 (OTG_GCCFG)*
OTG_CID	0x03C	*第 54.14.16 节：OTG 模块 ID 寄存器 (OTG_CID)*
OTG_GLPMCFG	0x54	*第 54.14.17 节：OTG 模块 LPM 配置寄存器 (OTG_GLPMCFG)*
OTG_HPTXFSIZ	0x100	*第 54.14.18 节：OTG 主机周期性发送 FIFO 大小寄存器 (OTG_HPTXFSIZ)*
OTG_DIEPTXFx	0x104 0x108 ... 0x120	*第 54.14.19 节：OTG 设备 IN 端点发送 FIFO x 大小寄存器 (OTG_DIEPTXFx)*

1. 通用规则是：主机模式下使用 OTG_HNPTXFSIZ；从机模式下使用 OTG_DIEPTXF0。

主机模式 CSR 地址映射

模块每次切换到主机模式时都必须对这些寄存器进行设置。

表 511. 主机模式控制和状态寄存器 (CSR)

缩略语	偏移地址	寄存器名称
OTG_HCFG	0x400	*第 54.14.21 节：OTG 主机配置寄存器 (OTG_HCFG)*
OTG_HFIR	0x404	*第 54.14.22 节：OTG 主机帧间隔寄存器 (OTG_HFIR)*
OTG_HFNUM	0x408	*第 54.14.23 节：OTG 主机帧编号/帧剩余时间寄存器 (OTG_HFNUM)*
OTG_HPTXSTS	0x410	*第 54.14.24 节：OTG_Host 周期性发送 FIFO/队列状态寄存器 (OTG_HPTXSTS)*
OTG_HAINT	0x414	*第 54.14.25 节：OTG 主机全体通道中断寄存器 (OTG_HAINT)*
OTG_HAINTMSK	0x418	*第 54.14.26 节：OTG 主机全体通道中断屏蔽寄存器 (OTG_HAINTMSK)*
OTG_HFLBADDR	0x41C	*第 54.14.27 节：OTG 主机帧列表基址寄存器 (OTG_HFLBADDR)*
OTG_HPRT	0x440	*第 54.14.28 节：OTG 主机端口控制和状态寄存器 (OTG_HPRT)*
OTG_HCCHARx	0x500 0x520 ... 0x6E0	*第 54.14.29 节：OTG 主机通道 x 特性寄存器 (OTG_HCCHARx)*

表 511. 主机模式控制和状态寄存器 (CSR)（续）

缩略语	偏移地址	寄存器名称
OTG_HCSPLTx	0x504 0x524 .... 0x6E4	*第 54.14.30 节：OTG 主机通道 x 分离控制寄存器 (OTG_HCSPLTx)*
OTG_HCINTx	0x508 0x528 .... 0x6E8	*第 54.14.31 节：OTG 主机通道 x 中断寄存器 (OTG_HCINTx)*
OTG_HCINTMSKx	0x50C 0x52C 0x6EC	*第 54.14.32 节：OTG 主机通道 x 中断屏蔽寄存器 (OTG_HCINTMSKx)*
OTG_HCTSIZx	0x510 0x530 .... 0x6F0	*第 54.14.33 节：OTG 主机通道 x 传输大小寄存器 (OTG_HCTSIZx)*
OTG_HCTSIZSGx	0x510 0x530 .... 0x6F0	*第 54.14.34 节：OTG 主机通道 x 传输大小寄存器 (OTG_HCTSIZSGx)*
OTG_HCDMAx	0x514 0x534 .... 0x6F4	*第 54.14.35 节：OTG 主机通道 x DMA 地址寄存器（缓冲 DMA 模式）[复用] (OTG_HCDMAx)*
OTG_HCDMASGx	0x514 0x534 .... 0x6F4	*第 54.14.36 节：OTG 主机通道 x DMA 地址寄存器（分散/收集 DMA）[复用] (OTG_HCDMASGx)*
OTG_HCDMABx	0x51C 0x53C 0x6FC	*第 54.14.37 节：OTG 主机通道 n DMA 地址缓冲区寄存器 (OTG_HCDMABx)*

从机模式 CSR 地址映射

模块每次切换到从机模式时都必须对这些寄存器进行编程。

表 512. 从机模式控制和状态寄存器

缩略语	偏移地址	寄存器名称
OTG_DCFG	0x800	*第 54.14.39 节：OTG 设备配置寄存器 (OTG_DCFG)*
OTG_DCTL	0x804	*第 54.14.40 节：OTG 设备控制寄存器 (OTG_DCTL)*
OTG_DSTS	0x808	*第 54.14.41 节：OTG 设备状态寄存器 (OTG_DSTS)*
OTG_DIEPMSK	0x810	*第 54.14.42 节：OTG 设备 IN 端点通用中断屏蔽寄存器 (OTG_DIEPMSK)*
OTG_DOEPMSK	0x814	*第 54.14.43 节：OTG 设备 OUT 端点通用中断屏蔽寄存器 (OTG_DOEPMSK)*
OTG_DAINT	0x818	*第 54.14.44 节：OTG 设备全体端点中断寄存器 (OTG_DAINT)*
OTG_DAINTMSK	0x81C	*第 54.14.45 节：OTG 全体端点中断屏蔽寄存器 (OTG_DAINTMSK)*
OTG_DVBUSDIS	0x828	*第 54.14.46 节：OTG 设备 V_{BUS} 放电时间寄存器 (OTG_DVBUSDIS)*
OTG_DVBUSPULSE	0x82C	*第 54.14.47 节：OTG 设备 V_{BUS} 脉冲时间寄存器 (OTG_DVBUSPULSE)*
OTG_DTHRCTL	0x830	*第 54.14.48 节：OTG 设备阈值控制寄存器 (OTG_DTHRCTL)*
OTG_DIEPEMPMSK	0x834	*第 54.14.49 节：OTG 设备 IN 端点 FIFO 空中断屏蔽寄存器 (OTG_DIEPEMPMSK)*
OTG_DIEPCTLx	0x900 0x920 ... 0xA00	*第 54.14.50 节：OTG 设备 IN 端点 x 控制寄存器 (OTG_DIEPCTLx)*
OTG_DIEPINTx	0x908 0x928 ... 0x9E8	*第 54.14.51 节：OTG 设备 IN 端点 x 中断寄存器 (OTG_DIEPINTx)*
OTG_DIEPTSIZ0	0x910	*第 54.14.52 节：OTG 设备 IN 端点 0 传输大小寄存器 (OTG_DIEPTSIZ0)*
OTG_DIEPDMAx	0x914 0x934 ... 0x9F4	*第 54.14.53 节：OTG 设备 IN 端点 x DMA 地址寄存器 (OTG_DIEPDMAx)*
OTG_DTXFSTSx	0x918 0x938 ..... 0x9F8	*第 54.14.54 节：OTG 设备 IN 端点发送 FIFO 状态寄存器 (OTG_DTXFSTSx)*

表 512. 从机模式控制和状态寄存器（续）

缩略语	偏移地址	寄存器名称
OTG_DIEPTSIZx	0x930 0x950 ... 0x9F0	*第 54.14.55 节：OTG 设备 IN 端点 x 传输大小寄存器 (OTG_DIEPTSIZx)*
OTG_DOEPCTL0	0xB00	*第 54.14.56 节：OTG 设备控制 OUT 端点 0 控制寄存器 (OTG_DOEPCTL0)*
OTG_DOEPINTx	0xB08 0XB28 ... 0xC08	*第 54.14.57 节：OTG 设备 OUT 端点 x 中断寄存器 (OTG_DOEPINTx)*
OTG_DOEPTSIZ0	0xB10	*第 54.14.58 节：OTG 设备 OUT 端点 0 传输大小寄存器 (OTG_DOEPTSIZ0)*
OTG_DOEPDMAx	0xB14 0xB34 ... 0xC14	*第 54.14.59 节：OTG 设备 OUT 端点 x DMA 地址寄存器 (OTG_DOEPDMAx)*
OTG_DOEPCTLx	0xB20 0xB40 ... 0xC00	*第 54.14.60 节：OTG 设备 OUT 端点 x 控制寄存器 (OTG_DOEPCTLx)*
OTG_DOEPTSIZx	0xB30 0xB50 .. 0xBF0	*第 54.14.61 节：OTG 设备 OUT 端点 x 传输大小寄存器 (OTG_DOEPTSIZx)*

数据 FIFO (DFIFO) 访问寄存器地址映射

这些寄存器在主机模式和从机模式下均可用，用于对给定方向的特定端点或通道的 FIFO 空间进行读写操作。如果主机通道类型为 IN，则只能对该通道上的 FIFO 进行读操作。同样地，如果主机通道类型为 OUT，则只能对该通道上的 FIFO 进行写操作。

表 513. 数据 FIFO (DFIFO) 访问寄存器地址映射

FIFO 访问寄存器段	偏移地址	访问
设备 IN 端点 0/主机 OUT 通道 0：DFIFO 写访问 设备 OUT 端点 0/主机 IN 通道 0：DFIFO 读访问	0x1000–0x1FFC	w r
设备 IN 端点 1/主机 OUT 通道 1：DFIFO 写访问 设备 OUT 端点 1/主机 IN 通道 1：DFIFO 读访问	0x2000–0x2FFC	w r
...	...	...
设备 IN 端点 x(1)/主机 OUT 通道 x(1)：DFIFO 写访问 设备 OUT 端点 x(1)/主机 IN 通道 x(1)：DFIFO 读访问	0xX000–0xXFFC	w r

1. 其中，x 为 8（从机模式）或 15（主机模式）。

电源和时钟门控 CSR 地址映射

由一个单独寄存器对电源和时钟门控进行管理。在主机模式和从机模式下均可使用。

表 514. 电源和时钟门控控制和状态寄存器

缩略语	偏移地址	寄存器名称
OTG_PCGCCTL	0xE00–0xE04	*第 54.14.62 节：OTG 电源和时钟门控控制寄存器 (OTG_PCGCCTL)*

54.14 OTG 寄存器

这些寄存器在主机模式和从机模式下都可用，且在这两个模式间切换时无需对其进行重新编程。

除非特别说明，否则寄存器描述中的位值以二进制表示。

54.14.1 OTG 控制和状态寄存器 (OTG_GOTGCTL)

OTG control and status register

偏移地址：0x000

复位值：0x0001 0000

OTG_GOTGCTL 寄存器控制模块的 OTG 功能并反映其状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CUR MOD	OTG VER	BSVLD	ASVLD	DBCT	CID STS
										r	rw	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	EHEN	DHNP EN	HSHNP EN	HNP RQ	HNG SCS	BVALO VAL	BVALO EN	AVALO VAL	AVALO EN	VBVAL OVAL	VBVAL OEN	SRQ	SRQ SCS
			rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw	rw	r

位 31:22 保留，必须保持复位值。

位 21 **CURMOD**：当前工作模式 (Current mode of operation)
该位指示当前模式（主机或设备）。
0：从机模式
1：主机模式

位 20 **OTGVER**：OTG 版本 (OTG version)
选择 OTG 版本。
0：OTG 版本 1.3。OTG1.3 在新产品开发中遭到淘汰。
1：OTG 版本 2.0。在此版本中，模块仅支持通过数据线脉冲实现 SRP。

位 19 **BSVLD**：B 会话有效 (B-session valid)
指示从机模式下收发器的状态。
0：B 会话无效。
1：B 会话有效。
在 OTG 模式下，该位可用于确定设备处于连接状态还是断开状态。
注意：仅可在从机模式下访问。

位 18 **ASVLD**：A 会话有效 (A-session valid)
指示主机模式下收发器的状态。
0：A 会话无效
1：A 会话有效
注意：仅可在主机模式下访问。

位 17 **DBCT**：长/短去抖动时间 (Long/short debounce time)
指示已检测到连接的去抖动时间。
0：长去抖动时间，用于物理连接 (100 ms + 2.5 µs)
1：短去抖动时间，用于软连接 (2.5 µs)
注意：仅可在主机模式下访问。

位 16 **CIDSTS**：连接器 ID 状态 (Connector ID status)
指示发生连接事件时的连接器 ID 状态。
0：OTG 控制器处于 A 从机模式
1：OTG 控制器处于 B 从机模式
注意：在从机模式和主机模式均可访问。

位 15:13 保留，必须保持复位值。

位 12 **EHEN**：嵌入式主机使能 (Embedded host enable)
用于在 OTG A 设备状态机和嵌入式主机状态机之间选择。
0：选择 OTG A 设备状态机
1：选择嵌入式主机状态机

位 11 **DHNPEN**：使能设备 HNP 特性 (Device HNP enabled)
从所连 USB 主机处成功接收到 SetFeature.SetHNPEnable 命令时，应用程序将该位置 1。
0：未在应用程序中使能 HNP
1：已在应用程序中使能 HNP
注意：仅可在从机模式下访问。

位 10 **HSHNPEN**：主机设置 HNP 使能 (Host set HNP enable)
（使用 SetFeature.SetHNPEnable 命令）在所连接设备上成功使能 HNP 后，应用程序将该位置 1。
0：未使能主机设置 HNP
1：已使能主机设置 HNP
注意：仅可在主机模式下访问。

位 9 **HNPRQ**：HNP 请求 (HNP request)

应用程序将该位置 1 时，将对所连接 USB 主机发起 HNP 请求。当 OTG_GOTGINT 寄存器中的主机协商成功状态更改位（OTG_GOTGINT 中的 HNSSCHG 位）置 1 时，应用程序可通过写 0 将该位清零。HNSSCHG 位清零时，模块会将该位清零。

0：不发送 HNP 请求

1：HNP 请求 (HNP request)

注意：仅可在从机模式下访问。

位 8 **HNGSCS**：主机协商成功 (Host negotiation success)

当主机协商成功时，模块会将该位置 1。当该寄存器中的 HNP 请求位 (HNPRQ) 置 1 时，模块会将该位清零。

0：主机协商失败

1：主机协商成功 (Host negotiation success)

注意：仅可在从机模式下访问。

位 7 **BVALOVAL**：B 设备会话有效覆盖值 (B-peripheral session valid override value)

该位用于在 BVALOEN 位置 1 时设置 Bvalid 信号的覆盖值。

0：BVALOEN = 1 时，Bvalid 值为“0”

1：BVALOEN = 1 时，Bvalid 值为“1”

注意：仅可在从机模式下访问。

位 6 **BVALOEN**：B 设备会话有效覆盖使能 (B-peripheral session valid override enable)

该位用于使能/禁止软件通过 BVALOVAL 位来覆盖 Bvalid 信号。

0：禁止覆盖，模块使用来自所选 PHY 的 Bvalid 信号

1：从 PHY 内部接收的 Bvalid 由 BVALOVAL 位的值覆盖

注意：仅可在从机模式下访问。

位 5 **AVALOVAL**：A 设备会话有效覆盖值 (A-peripheral session valid override value)

该位用于在 AVALOEN 位置 1 时设置 Avalid 信号的覆盖值。

0：AVALOEN = 1 时，Avalid 值为“0”

1：AVALOEN = 1 时，Avalid 值为“1”

注意：仅可在主机模式下访问。

位 4 **AVALOEN**：A 设备会话有效覆盖使能 (A-peripheral session valid override enable)

该位用于使能/禁止软件通过 AVALOVAL 位来覆盖 Avalid 信号。

0：禁止覆盖，模块使用来自所选 PHY 的 Avalid 信号

1：从 PHY 内部接收的 Avalid 由 AVALOVAL 位的值覆盖

注意：仅可在主机模式下访问。

位 3 **VBVALOVAL**：V_{BUS} 有效覆盖值 (V_{BUS} valid override value)。

该位用于在 VBVALOEN 位置 1 时设置 vbusvalid 信号的覆盖值。

0：VBVALOEN = 1 时，vbusvalid 值为“0”

1：VBVALOEN = 1 时，vbusvalid 值为“1”

注意：仅可在主机模式下访问。

位 2 **VBVALOEN**：V_{BUS} 有效覆盖使能 (V_{BUS} valid override enable)。

该位用于使能/禁止软件通过 VBVALOVAL 位来覆盖 vbusvalid 信号。

0：禁止覆盖，模块使用来自所选 PHY 的 vbusvalid 信号

1：从 PHY 内部接收的 vbusvalid 由 VBVALOVAL 位的值覆盖

注意：仅可在主机模式下访问。

位 1 **SRQ**：会话请求 (Session request)

应用程序将该位置 1 时，将在 USB 上发起会话请求。当 OTG_GOTGINT 寄存器中的主机协商成功状态更改位（OTG_GOTGINT 中的 HNSSCHG 位）置 1 时，应用程序可通过写 0 将该位清零。HNSSCHG 位清零时，模块会将该位清零。

0：无会话请求

1：会话请求 (Session request)

注意：仅可在从机模式下访问。

位 0 **SRQSCS**：会话请求成功 (Session request success)

当会话请求成功时，模块会将该位置 1。

0：会话请求失败

1：会话请求成功 (Session request success)

注意：仅可在从机模式下访问。

54.14.2 OTG 中断寄存器 (OTG_GOTGINT)

OTG interrupt register

偏移地址：0x04

复位值：0x0000 0000

应用程序会在出现 OTG 中断时读取该寄存器，并通过将该寄存器中的位清零来清除 OTG 中断。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBC DNE	ADTO CHG	HNG DET	Res.
												rc_w1	rc_w1	rc_w1	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	HNSS CHG	SRSS CHG	Res.	Res.	Res.	Res.	Res.	SEDET	Res.	Res.
						rc_w1	rc_w1						rc_w1		

位 31:20 保留，必须保持复位值。

位 19 **DBCDNE**：去抖动完成 (Debounce done)

模块会在设备连接后且去抖动结束时将该位置 1。应用程序会在看见该中断后，开始驱动 USB 复位。该位仅在 OTG_GUSBCFG 寄存器中的 HNP 功能位或 SRP 功能位（分别为 OTG_GUSBCFG 中的 HNPCAP 位或 SRPCAP 位）置 1 时有效。

注意：仅可在主机模式下访问。

位 18 **ADTOCHG**：A 设备超时更改 (A-device timeout change)

模块将该位置 1 时，指示 A 设备在等待 B 设备连接时超时。

注意：在从机模式和主机模式均可访问。

位 17 **HNGDET**：检测到主机协商 (Host negotiation detected)

当检测到 USB 上的主机协商请求时，模块会将该位置 1。

注意：在从机模式和主机模式均可访问。

位 16:10 保留，必须保持复位值。

位 9 **HNSSCHG**：主机协商成功状态更改 (Host negotiation success status change)

模块将在 USB 主机协商请求成功或失败时将该位置 1。应用程序必须读取 OTG_GOTGCTL 寄存器中的主机协商成功位（OTG_GOTGCTL 中的 HNGSCS 位）来检查请求成功还是失败。

注意：在从机模式和主机模式均可访问。

位 7:3 保留，必须保持复位值。

位 8 **SRSSCHG**：会话请求成功状态更改 (Session request success status change)

模块将在会话请求成功或失败时将该位置 1。应用程序必须读取 OTG_GOTGCTL 寄存器中的会话请求成功位（OTG_GOTGCTL 中的 SRQSCS 位）来检查请求成功还是失败。

注意：在从机模式和主机模式均可访问。

位 2 **SEDET**：检测到会话结束 (Session end detected)

模块将该位置 1 时，指示 V_{BUS} < 0.8 V 时，V_{BUS} 上的电压不再适用于 B 设备会话。

注意：在从机模式和主机模式均可访问。

位 1:0 保留，必须保持复位值。

54.14.3 OTG AHB 配置寄存器 (OTG_GAHBCFG)

OTG AHB configuration register

偏移地址：0x008

复位值：0x0000 0000

该寄存器可用于在上电后或更改角色模式时对模块进行配置。该寄存器主要包含 AHB 系统相关的配置参数。请勿在初始编程后更改该寄存器。应用程序必须在开始任何 AHB 或 USB 事务前对该寄存器进行编程。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PTXFE LVL	TXFE LVL	Res.	DMAEN	HBSTLEN[3:0]				GINT MSK
							rw	rw		rw	rw	rw	rw	rw	rw

位 31:9 保留，必须保持复位值。

位 8 **PTXFELVL**：周期性 Tx FIFO 空门限 (Periodic Tx FIFO empty level)

指示 OTG_GINTSTS 寄存器中的周期性 Tx FIFO 空中断位（OTG_GINTSTS 中的 PTXFE 位）何时触发。

0：PTXFE（位于 OTG_GINTSTS）中断表示周期性 Tx FIFO 为半空状态

1：PTXFE（位于 OTG_GINTSTS）中断表示周期性 Tx FIFO 为全空状态

注意：仅可在主机模式下访问。

位 7 **TXFELVL**：Tx FIFO 空门限 (Tx FIFO empty level)

在从机模式下，该位指示 IN 端点发送 FIFO 空中断（OTG_DIEPINTx 中的 TXFE）何时触发：

0：TXFE（位于 OTG_DIEPINTx）中断表示 IN 端点 Tx FIFO 为半空状态

1：TXFE（位于 OTG_DIEPINTx）中断表示 IN 端点 Tx FIFO 为全空状态

在主机模式下，该位指示非周期性 Tx FIFO 空中断（OTG_GINTSTS 中的 NPTXFE 位）何时触发。

0：NPTXFE（位于 OTG_GINTSTS）中断表示非周期性 Tx FIFO 为半空状态

1：NPTXFE（位于 OTG_GINTSTS）中断表示非周期性 Tx FIFO 为全空状态

位 6 保留，必须保持复位值。

位 5 **DMAEN**：使能 DMA (DMA enabled)

0：模块以从模式运行
1：模块以 DMA 模式运行

位 4:1 **HBSTLEN[3:0]**：批量长度/类型 (Burst length/type)

0000 单次：总线事务使用单次 32 位访问（不建议）
0001 INCR：总线事务使用未指定的长度访问（不建议使用 INCR AHB 总线命令）
0011 INCR4：总线事务指定 4x 32 位访问
0101 INCR8：总线事务指定 8x 32 位访问

0111 INCR16：总线事务基于 16x 32 位访问
其他值：保留

位 0 **GINTMSK**：全局中断屏蔽 (Global interrupt mask)

该位用于屏蔽全局中断或使能全局中断。中断状态寄存器由模块进行更新，与该位的设置无关。
0：屏蔽应用程序触发的中断。
1：取消对应用程序触发的中断的屏蔽。
注意：在从机模式和主机模式均可访问。

54.14.4 OTG USB 配置寄存器 (OTG_GUSBCFG)

OTG USB configuration register

偏移地址：0x00C

复位值：0x0000 1400

该寄存器可用于在上电或更改角色模式后对模块进行配置。其中包含与 USB 和 USB-PHY 相关的配置参数。应用程序必须在开始任何 AHB 或 USB 事务前对该寄存器进行编程。请勿在初始编程后更改该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	FD MOD	FH MOD	Res.	Res.	Res.	Res.	Res.	Res.	TSDPS	Res.	Res.	Res.	Res.	Res.	Res.
	rw	rw							rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHYL PC	Res.	TRDT[3:0]				HNP CAP	SRP CAP	Res.	Res.	Res.	Res.	Res.	TOCAL[2:0]		
rw		rw	rw	rw	rw	rw	rw						rw	rw	rw

位 31 保留，必须保持复位值。

位 30 **FDMOD**：强制从机模式 (Force device mode)

向该位写入 1 时，可将模块强制为从机模式，而无需考虑 OTG_ID 输入引脚。
0：正常模式
1：强制从机模式 (Force device mode)
将强制位置 1 后，应用程序必须等待至少 25 ms 后更改方可生效。
注意：在从机模式和主机模式均可访问。

位 29 **FHMOD**：强制主机模式 (Force host mode)

向该位写入 1 时，可将模块强制为主机模式，而无需考虑 OTG_ID 输入引脚。

0：正常模式

1：强制主机模式

将强制位置 1 后，应用程序必须等待至少 25 ms 后更改方可生效。

注意：在从机模式和主机模式均可访问。

位 28:26 保留，必须保持复位值。

位 25:23 保留，必须保持复位值。

位 22 **TSDPS**：TermSel DLine 脉冲选择 (TermSel DLine pulsing selection)

该位选择 utmi_termselect 来驱动 SRP（会话请求协议）期间的数据线脉冲。

0：使用 utmi_txvalid 驱动数据线脉冲（默认）

1：使用 utmi_termsel 驱动数据线脉冲

位 21:16 保留，必须保持复位值。

位 15 **PHYLPC**：PHY 低功耗时钟选择 (PHY Low-power clock select)

该位用于选择 480 MHz 或 48 MHz（低功耗）PHY 模式。在 FS 和 LS 模式下，PHY 通常可使用 48 MHz 时钟运行，从而节约功耗。

0：480 MHz 内部 PLL 时钟

1：48 MHz 外部时钟

在 480 MHz 模式下，UTMI 接口以 60 MHz 或 30 MHz 运行，具体取决于选择的是 8 位还是 16 位数据宽度。在 48 MHz 模式下，UTMI 接口在 FS 和 LS 模式下以 48 MHz 运行。

位 14 保留，必须保持复位值。

位 13:10 **TRDT[3:0]**：USB 周转时间 (USB turnaround time)

这些位用于以 PHY 时钟数为单位设置周转时间。必须根据*表 515：TRDT 值*来配置这些位，具体取决于应用程序 AHB 频率。TRDT 值越高，USB 对 IN 令牌的响应时间就越长，从而可以弥补 AHB 对数据 FIFO 的较长读访问延迟。

注意：仅可在从机模式下访问。

位 9 **HNPCAP**：HNP 使能 (HNP-capable)

应用程序使用该位控制 OTG 控制器的 HNP 功能。

0：不使能 HNP 功能。

1：使能 HNP 功能。

注意：在从机模式和主机模式均可访问。

位 8 **SRPCAP**：SRP 使能 (SRP-capable)

应用程序使用该位控制 OTG 控制器的 SRP 功能。如果模块作为无 SRP 功能的 B 设备，则无法请求连接的 A 设备（主机）激活 V_{BUS} 并开始会话。

0：不使能 SRP 功能。

1：使能 SRP 功能。

注意：在从机模式和主机模式均可访问。

位 7 保留，必须保持复位值。

位 6 保留，必须保持复位值。

位 5 保留，必须保持复位值。

位 4 保留，必须保持复位值。

位 3 保留，必须保持复位值。

位 2:0 **TOCAL[2:0]**：FS 超时校准 (FS timeout calibration)

PHY 引入的额外延迟包括应用程序在该位字段中设置的 PHY 时钟数，以及模块的全速数据包间超时间隔。不同 PHY 引入的延迟对数据线状态的影响是不同的。

全速操作的 USB 标准超时值为 16 到 18（含）个位时间。应用程序必须根据枚举速度编程该位字段。每个 PHY 时钟增加的位时间数为 0.25 个位时间。

表 515. TRDT 值

AHB 频率范围 (MHz)		TRDT 最小值
最小值	最大值	
30	-	0x9

54.14.5 OTG 复位寄存器 (OTG_GRSTCTL)

OTG reset register

偏移地址：0x10

复位值：0x8000 0000

应用程序通过此寄存器复位模块中的各项硬件特性。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AHB IDL	DMAREQ	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	TXFNUM[4:0]					TXF FLSH	RXF FLSH	Res.	FCRST	PSRST	CSRST
					rw	rw	rw	rw	rw	rs	rs		rs	rs	rs

位 31 **AHBIDL**：AHB 主设备空闲 (AHB master idle)

指示 AHB 主设备状态机处于空闲情况。

注意：在从机模式和主机模式均可访问。

位 30 **DMAREQ**：DMA 请求信号使能 (DMA request signal enabled)

该位指示 DMA 请求正在进行中。用于调试。

位 29:11 保留，必须保持复位值。

位 10:6 **TXFNUM[4:0]**：Tx FIFO 编号 (Tx FIFO number)

使用 Tx FIFO 刷新位进行刷新的 FIFO 编号。只有在模块将 Tx FIFO 刷新位清零后，方可更改该位字段。

00000：

– 主机模式下刷新非周期性 Tx FIFO

– 从机模式下刷新 Tx FIFO 0

00001：

– 主机模式下刷新周期性 Tx FIFO

– 从机模式下刷新 Tx FIFO 1

00010：从机模式下刷新 Tx FIFO 2

...

01111：从机模式下刷新 Tx FIFO 15

10000：在从机模式或主机模式下刷新所有的发送 FIFO。

注意：在从机模式和主机模式均可访问。

位 5 **TXFFLSH**：Tx FIFO 刷新 (Tx FIFO flush)

该位选择性地刷新一个或所有的发送 FIFO，但当模块处理通信事务时无法执行该操作。

只有在确认模块当前未对 Tx FIFO 执行读写操作后，应用程序方可对该位执行写操作。使用以下寄存器进行确认：

读——NAK 有效中断可确保模块当前未对 FIFO 执行读操作

写——OTG_GRSTCTL 中的 AHBIDL 位可确保模块当前未对 FIFO 执行任何写操作。

通常建议在重新配置 FIFO 时进行刷新。还建议在设备端点禁止期间进行 FIFO 刷新。应用程序必须等待模块将该位清零，才能执行任意操作。该位需要八个时钟来清零（使用较慢的 phy_clk 或 hclk 时钟）。

注意：在从机模式和主机模式均可访问。

位 4 **RXFFLSH**：Rx FIFO 刷新 (Rx FIFO flush)

应用程序可使用该位刷新整个 Rx FIFO，但必须首先确保模块当前未在处理事务。

只有在确认模块当前未对 Rx FIFO 执行读写操作后，应用程序方可对该位执行写操作。

应用程序必须等到该位清零后，方可执行其他操作。通常需要等待 8 个时钟周期（以 PHY 或 AHB 时钟中最慢的为准）。

注意：在从机模式和主机模式均可访问。

位 3 保留，必须保持复位值。

位 2 **FCRST**：主机帧计数器复位 (Host frame counter reset)

应用程序对该位执行写操作时，模块中的（微）帧数计数器复位。（微）帧计数器复位后，由模块发送的下一个 SOF 的帧号为 0。

当应用程序向该位写入“1”后，可能无法回读该值，原因是模块将在几个时钟周期内将该位清零。

注意：仅可在主机模式下访问。

位 1 **PSRST**：部分软复位 (Partial soft reset)

复位内部状态机，但保留枚举信息。可用于恢复一些特定的 PHY 错误。

注意：在从机模式和主机模式均可访问。

位 0 **CSRST**：模块软复位 (Core soft reset)

按如下所述将 HCLK 和 PHY 时钟域复位：

除以下各位外，将各个中断和所有 CSR 寄存器位清零：

–OTG_PCGCCTL 中的 GATEHCLK 位

–OTG_PCGCCTL 中的 STPPCLK 位

–OTG_HCFG 中的 FSLSPCS 位

–OTG_DCFG 中的 DSPD 位

–OTG_DCTL 中的 SDIS 位

–OTG_GCCFG 寄存器

将所有模块状态机（AHB 从设备除外）复位至空闲状态，并清空所有发送 FIFO 和接收 FIFO。

在 AHB 传输的最后数据阶段结束后，尽快终止 AHB 主设备上的所有事务。立即终止 USB 上的所有事务。

应用程序可在需要复位模块时随时对该位执行写操作。该位为自清零位，模块将在其中所有必要逻辑复位后将该位清零，该过程需要若干个时钟的时间，具体取决于模块的当前状态。该位一旦清零，软件必须等待至少 3 个 PHY 时钟后才可以访问 PHY 域（同步延迟）。此外，软件还必须在确定该寄存器中的位 31 置 1（AHB 主设备空闲）后方可开始运行。

软件复位通常在两种情况下使用，一是软件开发期间，二是用户动态更改以上所列 USB 配置寄存器中的 PHY 选择位后。用户更改 PHY 时，将为 PHY 选择相应的时钟并用于 PHY 域中。一旦选择了新的时钟，则必须复位 PHY 域，才能保证正常运行。

注意：在从机模式和主机模式均可访问。

54.14.6 OTG 模块中断寄存器 (OTG_GINTSTS)

OTG core interrupt register

偏移地址：0x014

复位值：0x0400 0020

该寄存器用于在当前模式（从机模式或主机模式）下借助系统级别的事件来中断应用程序。

该寄存器中的某些位仅在主机模式下有效，而其他位则仅在从机模式下有效。此外，该寄存器还可指示当前模式。要将 rc_w1 类型中断状态位清零，应用程序必须向该位写入 1。

FIFO 状态中断为只读；如果软件在处理这些中断期间对 FIFO 执行读写操作，则 FIFO 中断标志将自动清零。

使能中断位前，应用程序必须在初始化时将 OTG_GINTSTS 寄存器清零，才可以避免在初始化前产生任何中断。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUP INT	SRQ INT	DISC INT	CIDS CHG	LPM INT	PTXFE	HCINT	HPRT INT	RST DET	DATAF SUSP	IPXFR/ IN COMP ISO OUT	IISOI XFR	OEP INT	IEPINT	Res.	Res.
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	r	r	r	rc_w1	rc_w1	rc_w1	rc_w1	r	r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPF	ISOO DRP	ENUM DNE	USB RST	USB SUSP	ESUSP	Res.	Res.	GO NAK EFF	GI NAK EFF	NPTXF E	RXF LVL	SOF	OTG INT	MMIS	CMOD
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1			r	r	r	r	rc_w1	r	rc_w1	r

位 31 **WKUPINT**：恢复/远程唤醒检测到中断 (Resume/remote wake-up detected interrupt)

挂起 (L2) 或 LPM(L1) 状态期间的唤醒中断。

– 挂起 (L2) 期间：

在从机模式下，当 USB 总线上检测到恢复信号时，将触发该中断。在主机模式下，当 USB 上检测到远程唤醒时，将触发该中断。

– LPM(L1) 期间：

USB 上检测到主机发起的恢复信号或设备发起的远程唤醒时，将触发该中断。

注意：在从机模式和主机模式均可访问。

位 30 **SRQINT**：检测到会话请求/新会话中断 (Session request/new session detected interrupt)

在主机模式下，当检测到来自设备的会话请求时，将触发该中断。在从机模式下，当 V_{BUS} 在 B 设备的有效范围内时，将触发该中断。在从机模式和主机模式均可访问。

位 29 **DISCINT**：检测到断开连接中断 (Disconnect detected interrupt)

当检测到设备断开连接时触发该中断。

注意：仅可在主机模式下访问。

位 28 **CIDSCHG**：连接器 ID 线状态更改 (Connector ID status change)

当连接器 ID 线状态发生更改时，模块将该位置 1。

注意：在从机模式和主机模式均可访问。

位 27 **LPMINT**：LPM 中断 (LPM interrupt)

在从机模式下，当设备接收到 LPM 事务并以非 ERROR 应答时，将触发该中断。

在主机模式下，当设备以非 ERROR 应答 LPM 事务或主机模块完成所编程次数（OTG_GLPMCFG 中的 RETRYCNT 位）的 LPM 事务时，将触发该中断。

只有 OTG_GLPMCFG 中的 LPMEN 位置 1 时，该位字段才有效。

位 26 **PTXFE**：周期性 Tx FIFO 为空 (Periodic Tx FIFO empty)

当周期性发送 FIFO 为半空或全空状态，且周期性请求队列中存在可写入至少一个条目的空间时，将触发该中断。该 FIFO 为半空状态还是全空状态由 OTG_GAHBCFG 寄存器中的周期性 Tx FIFO 空门限位（OTG_GAHBCFG 中的 PTXFELVL 位）决定。

注意：仅可在主机模式下访问。

位 25 **HCINT**：主机通道中断 (Host channels interrupt)

模块将该位置 1 时，指示模块中一个通道上存在挂起的中断（在主机模式下）。应用程序必须读取 OTG_HAINT 寄存器，以确定发生中断的通道的准确编号，然后读取相应的 OTG_HCINTx 寄存器，以确定引发中断的确切原因。应用程序必须先将 OTG_HCINTx 寄存器的相应状态位清零，之后才能将该位清零。

注意：仅可在主机模式下访问。

位 24 **HPRTINT**：主机端口中断 (Host port interrupt)

模块将该位置 1 时，指示主机模式下 OTG 控制器端口的状态发生变化。应用程序必须读取 OTG_HPRT 寄存器，以确定引发此中断的确切事件。应用程序必须先将 OTG_HPRT 寄存器的相应状态位清零，之后才能将该位清零。

注意：仅可在主机模式下访问。

位 23 **RSTDET**：检测到复位中断 (Reset detected interrupt)

在从机模式下，若设备处于挂起状态，则在部分断电模式下的 USB 上检测到复位时，将触发该中断。

注意：仅可在从机模式下访问。

位 22 **DATAFSUSP**：数据获取挂起 (Data fetch suspended)

该中断仅在 DMA 模式下有效。该中断指示，模块因 TxFIFO 空间或请求队列空间不可用而停止为 IN 端点获取数据。应用程序进行端点不匹配处理时会用到该中断。例如，在检测到端点不匹配后，应用程序将执行以下操作：

- 将全局非周期性 IN NAK 握手信号置 1
- 禁止 IN 端点
- 清空 FIFO
- 根据 IN 令牌序列学习队列确定令牌序列
- 重新使能端点

如果全局非周期性 IN NAK 已清零但模块尚未为 IN 端点获取数据，同时又已接收到 IN 令牌，则清零全局非周期性 IN NAK 握手信号：模块将产生“FIFO 为空时接收到 IN 令牌”中断。然后，OTG 将 NAK 响应发送到主机。为避免这种情况的发生，应用程序可以检查 OTG_GINTSTS 中的 FetSusp 中断，该中断可确保在 FIFO 存满后再将全局 NAK 握手信号清零。或者，应用程序可以在将全局 IN NAK 握手信号清零时屏蔽“当 FIFO 为空时接收到 IN 令牌”中断。

位 21 **IPXFR**：未完成周期性传输 (Incomplete periodic transfer)

在主机模式下，如果存在仍处于挂起状态的未完成周期性事务，而这些事务计划在当前帧期间完成，则模块会将该中断位置 1。

INCOMPISOOUT：未完成 OUT 同步传输 (Incomplete isochronous OUT transfer)

在从机模式下，模块将该中断置 1 时，指示当前帧中至少有一个同步 OUT 端点上的传输未完成。该中断随该寄存器中的周期性帧结束中断 (EOPF) 位一同触发。

位 20 **IISOIXFR**：未完成 IN 同步传输 (Incomplete isochronous IN transfer)

模块将该中断置 1 时，指示当前帧中至少有一个同步 IN 端点上的传输未完成。该中断随该寄存器中的周期性帧结束中断 (EOPF) 位一同触发。

注意：仅可在从机模式下访问。

位 19 **OEPINT**：OUT 端点中断 (OUT endpoint interrupt)

模块将该位置 1 时，指示模块中一个 OUT 端点上存在挂起的中断（在从机模式下）。应用程序必须读取 OTG_DAINT 寄存器，以确定发生中断的 OUT 端点的准确编号，然后读取相应的 OTG_DOEPINTx 寄存器，以确定引发中断的确切原因。应用程序必须先将相应 OTG_DOEPINTx 寄存器的相应状态位清零，之后才能将该位清零。

注意：仅可在从机模式下访问。

位 18 **IEPINT**：IN 端点中断 (IN endpoint interrupt)

模块将该位置 1 时，指示模块中一个 IN 端点上存在挂起的中断（在从机模式下）。应用程序必须读取 OTG_DAINT 寄存器，以确定发生中断的 IN 端点的准确编号，然后读取相应的 OTG_DIEPINTx 寄存器，以确定引发中断的确切原因。应用程序必须先将相应 OTG_DIEPINTx 寄存器的相应状态位清零，之后才能将该位清零。

注意：仅可在从机模式下访问。

位 17:16 保留，必须保持复位值。

位 15 **EOPF**：周期性帧结束中断 (End of periodic frame interrupt)

指示当前帧已达到 OTG_DCFG 寄存器中周期性帧间隔位字段（OTG_DCFG 中的 PFIVL 位）所指定的周期。

注意：仅可在从机模式下访问。

位 14 **ISOODRP**：丢弃同步 OUT 数据包中断 (Isochronous OUT packet dropped interrupt)

如果由于 Rx FIFO 空间不足，无法容纳同步 OUT 端点的最大数据包，从而导致模块无法向 Rx FIFO 写入同步 OUT 数据包，模块会将该位置 1。

注意：仅可在从机模式下访问。

位 13 **ENUMDNE**：枚举完成 (Enumeration done)

模块将该位置 1 时，指示速度枚举已完成。应用程序必须读取 OTG_DSTS 寄存器来获取枚举速度。

注意：仅可在从机模式下访问。

位 12 **USBRST**：USB 复位 (USB reset)

模块将该位置 1 时，指示在 USB 上检测到复位信号。

注意：仅可在从机模式下访问。

位 11 **USBSUSP**：USB 挂起 (USB suspend)

模块将该位置 1 时，指示在 USB 上检测到挂起状态。当数据线上长时间没有活动时，模块便会进入挂起状态。

注意：仅可在从机模式下访问。

位 10 **ESUSP**：早期挂起 (Early suspend)

模块将该位置 1 时，指示已检测到 USB 处于空闲状态的时间达到 3 ms。

注意：仅可在从机模式下访问。

位 9:8 保留，必须保持复位值。

位 7 **GONAKEFF**：全局 OUT NAK 有效 (Global OUT NAK effective)

指示 OTG_DCTL 寄存器中由应用程序设置的“将全局 OUT NAK 置 1”位（OTG_DCTL 中的 SGONAK 位）已在模块中生效。通过写入 OTG_DCTL 寄存器中的“将全局 OUT NAK 清零”位（OTG_DCTL 中的 CGONAK 位），可将该位清零。

注意：仅可在从机模式下访问。

位 6 **GINAKEFF**：全局非周期性 IN NAK 有效 (Global IN nonperiodic NAK effective)

指示 OTG_DCTL 寄存器中由应用程序设置的“将全局非周期性 IN NAK 置 1”位（OTG_DCTL 中的 SGINAK 位）已在模块中生效。也就是说，模块已对应用程序设置的全局 IN NAK 位进行采样，结果已生效。通过清零 OTG_DCTL 寄存器中的“将全局非周期性 IN NAK 清零”位（OTG_DCTL 中的 CGINAK 位），可将该位清零。

此中断不一定表示 USB 上已发送了一个 NAK 握手信号。STALL 位优先级高于 NAK 位。

注意：仅可在从机模式下访问。

位 5 **NPTXFE**：非周期性 Tx FIFO 为空 (Non-periodic Tx FIFO empty)

当非周期性 Tx FIFO 为半空或全空状态，且非周期性发送请求队列中至少存在可写入一个条目的空间时，将触发该中断。该 FIFO 为半空状态还是全空状态由 OTG_GAHBCFG 寄存器中的非周期性 Tx FIFO 空门限位（OTG_GAHBCFG 中的 TXFELVL 位）决定。

注意：仅可在主机模式下访问。

位 4 **RXFLVL**：Rx FIFO 非空 (Rx FIFO non-empty)

指示 Rx FIFO 中至少有一个数据包等待读取。

注意：在主机模式和从机模式均可访问。

位 3 **SOF**：帧起始 (Start of frame)

在主机模式下，模块将该位置 1 时，指示 USB 上已发送一个 SOF (FS) 或 Keep-Alive (LS) 信号。应用程序必须将该位置 1 才可清除该中断。

在从机模式下，模块将该位置 1 时，指示 USB 上已接收到一个 SOF 令牌。应用程序可通过读取 OTG_DSTS 寄存器来获得当前的帧编号。只有在模块以 FS 模式运行时，才会出现此中断。

注意：如果上电复位后立即读取，该寄存器可能返回“1”。如果寄存器位在上电复位后立即读取“1”，则不表示已发送 SOF（主机模式下）或已接收 SOF（从机模式下）。只有在主机和设备之间建立有效连接后，此中断的读取值才有效。如果该位在上电复位后置 1，应用程序可把该位清零。

注意：在主机模式和从机模式均可访问。

位 2 **OTGINT**：OTG 中断 (OTG interrupt)

模块将该位置 1 时，指示出现 OTG 协议事件。应用程序必须读取 OTG 中断状态 (OTG_GOTGINT) 寄存器，以确定引发此中断的确切事件。应用程序必须先将 OTG_GOTGINT 寄存器的相应状态位清零，之后才能将该位清零。

注意：在主机模式和从机模式均可访问。

位 1 **MMIS**：模式不匹配中断 (Mode mismatch interrupt)

当应用程序尝试访问以下寄存器时，模块将该位置 1：

– 模块运行在从机模式下访问主机模式寄存器

– 模块运行在主机模式下访问从机模式寄存器

寄存器访问在 AHB 上以 OKAY 响应结束，但该访问在内部被模块忽略并且不会影响模块运行。

注意：在主机模式和从机模式均可访问。

位 0 **CMOD**：当前工作模式 (Current mode of operation)

指示当前模式。

0：从机模式

1：主机模式

注意：在主机模式和从机模式均可访问。

54.14.7 OTG 中断屏蔽寄存器 (OTG_GINTMSK)

OTG interrupt mask register

偏移地址：0x018

复位值：0x0000 0000

该寄存器与模块中断寄存器结合使用，以中断应用程序。如果将某个中断位屏蔽，则不会产生与该位相关的中断。但是，与该中断相对应的模块中断 (OTG_GINTSTS) 寄存器位仍会置 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WUIM	SRQIM	DISCINT	CIDSCHGM	LPMINTM	PTXFEM	HCIM	PRTIM	RSTDETM	FSUSPM	IPXFRM/IISOOXFRM	IISOIXFRM	OEPINT	IEPINT	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPFM	ISOODRPM	ENUMDNEM	USBRST	USBSUSPM	ESUSPM	Res.	Res.	GONAKEFFM	GINAKEFFM	NPTXFEM	RXFLVLM	SOFM	OTGINT	MMISM	Res.
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	

位 31 **WUIM**：检测到恢复/远程唤醒中断屏蔽 (Resume/remote wake-up detected interrupt mask)

0：屏蔽中断

1：使能中断

注意：在主机模式和从机模式均可访问。

位 30 **SRQIM**：检测到会话请求/新会话中断屏蔽 (Session request/new session detected interrupt mask)

0：屏蔽中断

1：使能中断

注意：在主机模式和从机模式均可访问。

位 29 **DISCINT**：检测到断开连接中断屏蔽 (Disconnect detected interrupt mask)
0：屏蔽中断
1：使能中断
注意：仅可在主机模式下访问。

位 28 **CIDSCHGM**：连接器 ID 状态更改屏蔽 (Connector ID status change mask)
0：屏蔽中断
1：使能中断
注意：在主机模式和从机模式均可访问。

位 27 **LPMINTM**：LPM 中断屏蔽 (LPM interrupt mask)
0：屏蔽中断
1：使能中断
注意：在主机模式和从机模式均可访问。

位 26 **PTXFEM**：周期性 Tx FIFO 空屏蔽 (Periodic Tx FIFO empty mask)
0：屏蔽中断
1：使能中断
注意：仅可在主机模式下访问。

位 25 **HCIM**：主机通道中断屏蔽 (Host channels interrupt mask)
0：屏蔽中断
1：使能中断
注意：仅可在主机模式下访问。

位 24 **PRTIM**：主机端口中断屏蔽 (Host port interrupt mask)
0：屏蔽中断
1：使能中断
注意：仅可在主机模式下访问。

位 23 **RSTDETM**：检测到复位中断屏蔽 (Reset detected interrupt mask)
0：屏蔽中断
1：使能中断
注意：仅可在从机模式下访问。

位 22 **FSUSPM**：数据获取挂起中断屏蔽 (Data fetch suspended mask)
0：屏蔽中断
1：使能中断
仅可在从机模式下访问。

位 21 **IPXFRM**：未完成周期性传输中断屏蔽 (Incomplete periodic transfer mask)
0：屏蔽中断
1：使能中断
注意：仅可在主机模式下访问。
IISOOXFRM：未完成 OUT 同步传输中断屏蔽 (Incomplete isochronous OUT transfer mask)
0：屏蔽中断
1：使能中断
注意：仅可在从机模式下访问。

位 20 **IISOIXFRM**：未完成 IN 同步传输中断屏蔽 (Incomplete isochronous IN transfer mask)
0：屏蔽中断
1：使能中断
注意：仅可在从机模式下访问。

位 19 **OEPINT**：OUT 端点中断屏蔽 (OUT endpoints interrupt mask)
0：屏蔽中断
1：使能中断
注意：仅可在从机模式下访问。

位 18 **IEPINT**：IN 端点中断屏蔽 (IN endpoints interrupt mask)
0：屏蔽中断
1：使能中断
注意：仅可在从机模式下访问。

位 17:16 保留，必须保持复位值。

位 15 **EOPFM**：周期性帧结束中断屏蔽 (End of periodic frame interrupt mask)
0：屏蔽中断
1：使能中断
注意：仅可在从机模式下访问。

位 14 **ISOODRPM**：丢弃同步 OUT 数据包中断屏蔽 (Isochronous OUT packet dropped interrupt mask)
0：屏蔽中断
1：使能中断
注意：仅可在从机模式下访问。

位 13 **ENUMDNEM**：枚举完成中断屏蔽 (Enumeration done mask)
0：屏蔽中断
1：使能中断
注意：仅可在从机模式下访问。

位 12 **USBRST**：USB 复位中断屏蔽 (USB reset mask)
0：屏蔽中断
1：使能中断
注意：仅可在从机模式下访问。

位 11 **USBSUSPM**：USB 挂起中断屏蔽 (USB suspend mask)
0：屏蔽中断
1：使能中断
注意：仅可在从机模式下访问。

位 10 **ESUSPM**：早期挂起中断屏蔽 (Early suspend mask)
0：屏蔽中断
1：使能中断
注意：仅可在从机模式下访问。

位 9:8 保留，必须保持复位值。

位 7 **GONAKEFFM**：全局 OUT NAK 生效中断屏蔽 (Global OUT NAK effective mask)
0：屏蔽中断
1：使能中断
注意：仅可在从机模式下访问。

位 6 **GINAKEFFM**：全局非周期性 IN NAK 生效中断屏蔽 (Global non-periodic IN NAK effective mask)
0：屏蔽中断
1：使能中断
注意：仅可在从机模式下访问。

位 5 **NPTXFEM**：非周期性 Tx FIFO 空中断屏蔽 (Non-periodic Tx FIFO empty mask)
0：屏蔽中断
1：使能中断
注意：仅可在主机模式下访问。

位 4 **RXFLVLM**：接收 FIFO 非空中断屏蔽 (Receive FIFO non-empty mask)
0：屏蔽中断
1：使能中断
注意：在从机模式和主机模式均可访问。

位 3 **SOFM**：帧起始中断屏蔽 (Start of frame mask)
0：屏蔽中断
1：使能中断
注意：在从机模式和主机模式均可访问。

位 2 **OTGINT**：OTG 中断屏蔽 (OTG interrupt mask)
0：屏蔽中断
1：使能中断
注意：在从机模式和主机模式均可访问。

位 1 **MMISM**：模式不匹配中断屏蔽 (Mode mismatch interrupt mask)
0：屏蔽中断
1：使能中断
注意：在从机模式和主机模式均可访问。

位 0 保留，必须保持复位值。

54.14.8 OTG 接收状态调试读取寄存器 (OTG_GRXSTSR)

OTG receive status debug read register

读取的偏移地址：0x01C

复位值：0x0000 0000

该说明针对从机模式下的寄存器 OTG_GRXSTSR。

读取接收状态调试读取寄存器将返回接收 FIFO 顶部的内容。

当接收 FIFO 为空时，模块会忽略对该寄存器的读取操作，并返回值 0x0000 0000。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	STSPHST	Res.	Res.	FRMNUM[3:0]				PKTSTS[3:0]				DPID[1]
				r			r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPID[0]	BCNT[10:0]											EPNUM[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27 **STSPHST**：起始状态阶段 (Status phase start)
指示控制写传输的起始状态阶段。该位与 OUT 传输完成的 PKTSTS 模式均置 1。

位 26:25 保留，必须保持复位值。

位 24:21 **FRMNUM[3:0]**：帧编号 (Frame number)
这是在 USB 上接收的数据包帧编号的 4 个最低有效位。只有当支持同步 OUT 端点时才支持该位字段。

位 20:17 **PKTSTS[3:0]**：数据包状态 (Packet status)
指示接收的数据包的状态
0001：全局 OUT NAK（触发中断）
0010：接收到 OUT 数据包
0011：OUT 传输完成（触发中断）
0100：SETUP 事务完成（触发中断）
0110：接收到 SETUP 数据包
其他值：保留

位 16:15 **DPID[1:0]**：数据 PID (Data PID)
指示接收的 OUT 数据包的数据 PID
00：DATA0
10：DATA1
01：DATA2
11：MDATA

位 14:4 **BCNT[10:0]**：字节计数 (Byte count)
指示接收的数据包的字节数。

位 3:0 **EPNUM[3:0]**：端点编号 (Endpoint number)
指示当前接收的数据包所属的端点编号。

54.14.9 OTG 接收状态调试读取寄存器 [复用] (OTG_GRXSTSR)

OTG receive status debug read

读取的偏移地址：0x01C

复位值：0x0000 0000

该说明针对主机模式下的寄存器 OTG_GRXSTSR。

读取接收状态调试读取寄存器将返回接收 FIFO 顶部的内容。

当接收 FIFO 为空时，模块会忽略对该寄存器的读取操作，并返回值 0x0000 0000。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTSTS[3:0]				DPID
											r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPID	BCNT[10:0]											CHNUM[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:21 保留，必须保持复位值。

位 20:17 **PKTSTS[3:0]**：数据包状态 (Packet status)
指示接收的数据包的状态
0010：接收到 IN 数据包
0011：IN 传输完成（触发中断）
0101：数据同步错误（触发中断）
0111：暂停通道（触发中断）
其他值：保留

位 16:15 **DPID[1:0]**：数据 PID (Data PID)
指示接收的数据包的数据 PID
00：DATA0
10：DATA1
01：DATA2
11：MDATA

位 14:4 **BCNT[10:0]**：字节计数 (Byte count)
指示接收的 IN 数据包的字节数。

位 3:0 **CHNUM[3:0]**：通道编号 (Channel number)
指示当前接收的数据包所属的通道编号。

54.14.10 OTG 状态读取和弹出寄存器 (OTG_GRXSTSP)

OTG status read and pop registers

出栈的偏移地址：0x020

复位值：0x0000 0000

该说明针对从机模式下的寄存器 OTG_GRXSTSP。

读取 OTG_GRXSTSR（接收状态调试读取寄存器）将返回接收 FIFO 顶部的内容，而读取 OTG_GRXSTSP（接收状态读取和出栈寄存器）将在其基础之上额外弹出 Rx FIFO 顶部的数据条目。

当接收 FIFO 为空时，模块会忽略对该寄存器的读取或出栈操作，并返回值 0x0000 0000。当模块中断寄存器的接收 FIFO 非空位（OTG_GINTSTS 中的 RXFLVL 位）置位时，应用程序必须仅弹出接收状态 FIFO。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	STSPHST	Res.	Res.	FRMNUM[3:0]				PKTSTS[3:0]				DPID[1]
				r			r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPID[0]	BCNT[10:0]											EPNUM[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27 **STSPHST**：起始状态阶段 (Status phase start)
指示控制写传输的起始状态阶段。该位与 OUT 传输完成的 PKTSTS 模式均置 1。

位 26:25 保留，必须保持复位值。

位 24:21 **FRMNUM[3:0]**：帧编号 (Frame number)
这是在 USB 上接收的数据包帧编号的 4 个最低有效位。只有当支持同步 OUT 端点时才支持该位字段。

位 20:17 **PKTSTS[3:0]**：数据包状态 (Packet status)
指示接收的数据包的状态
0001：全局 OUT NAK（触发中断）
0010：接收到 OUT 数据包
0011：OUT 传输完成（触发中断）
0100：SETUP 事务完成（触发中断）
0110：接收到 SETUP 数据包
其他值：保留

位 16:15 **DPID[1:0]**：数据 PID (Data PID)
指示接收的 OUT 数据包的数据 PID
00：DATA0
10：DATA1
01：DATA2
11：MDATA

位 14:4 **BCNT[10:0]**：字节计数 (Byte count)
指示接收的数据包的字节数。

位 3:0 **EPNUM[3:0]**：端点编号 (Endpoint number)
指示当前接收的数据包所属的端点编号。

54.14.11 OTG 状态读取和弹出寄存器 [复用] (OTG_GRXSTSP)

OTG status read and pop registers [alternate]

出栈的偏移地址：0x020

复位值：0x0000 0000

该说明针对主机模式下的寄存器 OTG_GRXSTSP。

读取 OTG_GRXSTSR（接收状态调试读取寄存器）将返回接收 FIFO 顶部的内容，而读取 OTG_GRXSTSP（接收状态读取和出栈寄存器）将在其基础之上额外弹出 Rx FIFO 顶部的数据条目。

当接收 FIFO 为空时，模块会忽略对该寄存器的读取或出栈操作，并返回值 0x0000 0000。当模块中断寄存器的接收 FIFO 非空位（OTG_GINTSTS 中的 RXFLVL 位）置位时，应用程序必须仅弹出接收状态 FIFO。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTSTS[3:0]				DPID
											r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPID	BCNT[10:0]											CHNUM[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:21 保留，必须保持复位值。

位 20:17 **PKTSTS[3:0]**：数据包状态 (Packet status)
指示接收的数据包的状态
0010：接收到 IN 数据包
0011：IN 传输完成（触发中断）
0101：数据同步错误（触发中断）
0111：暂停通道（触发中断）
其他值：保留

位 16:15 **DPID[1:0]**：数据 PID (Data PID)
指示接收的数据包的数据 PID
00：DATA0
10：DATA1
01：DATA2
11：MDATA

位 14:4 **BCNT[10:0]**：字节计数 (Byte count)
指示接收的 IN 数据包的字节数。

位 3:0 **CHNUM[3:0]**：通道编号 (Channel number)
指示当前接收的数据包所属的通道编号。

54.14.12 OTG 接收 FIFO 大小寄存器 (OTG_GRXFSIZ)

OTG receive FIFO size register

偏移地址：0x024

复位值：0x0000 0400

此应用程序可以对分配给 Rx FIFO 的 RAM 大小进行编程。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXFD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **RXFD[15:0]**：Rx FIFO 深度 (Rx FIFO depth)
以 32 位字为单位。
最大值为 1024
编程值必须遵循可分配的 FIFO 存储区，不得超过上电值。

54.14.13 OTG 主机非周期性发送 FIFO 大小寄存器 (OTG_HNPTXFSIZ)/端点 0 发送 FIFO 大小 (OTG_DIEPTXF0)

OTG host non-periodic transmit FIFO size register/Endpoint 0 Transmit FIFO size

偏移地址：0x028

复位值：0x0200 0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NPTXFD/TX0FD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NPTXFSA/TX0FSA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

主机模式

位 31:16 **NPTXFD[15:0]**：非周期性 Tx FIFO 深度 (Non-periodic Tx FIFO depth)
以 32 位字为单位。
最小值为 16
编程值必须遵循可分配的 FIFO 存储区，不得超过上电值。

位 15:0 **NPTXFSA[15:0]**：非周期性发送 RAM 起始地址 (Non-periodic transmit RAM start address)
该位字段配置非周期性发送 FIFO RAM 的内存起始地址。

从机模式

位 31:16 **TX0FD**：端点 0 Tx FIFO 深度 (Endpoint 0 Tx FIFO depth)
以 32 位字为单位。
最小值为 16
编程值必须遵循可分配的 FIFO 存储区，不得超过上电值。

位 15:0 **TX0FSA**：端点 0 发送 RAM 起始地址 (Endpoint 0 transmit RAM start address)
该位字段配置端点 0 发送 FIFO RAM 的内存起始地址。

54.14.14 OTG 非周期性发送 FIFO/队列状态寄存器 (OTG_HNPTXSTS)

OTG non-periodic transmit FIFO/queue status register

偏移地址：0x02C

复位值：0x0008 0400

注： *在从机模式下，此寄存器无效。*

此只读寄存器包含非周期性 Tx FIFO 和非周期性发送请求队列的空闲空间信息。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	NPTXQTOP[6:0]							NPTQXSAV[7:0]							
	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NPTXFSAV[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31 保留，必须保持复位值。

位 30:24 **NPTXQTOP[6:0]**：非周期性发送请求队列顶部 (Top of the non-periodic transmit request queue)

非周期性发送请求队列中 MAC 目前正在处理的条目。

位 30:27：通道/端点编号

位 26:25：

00：IN/OUT 令牌

01：长度为零的发送数据包（设备 IN/主机 OUT）

11：通道停止命令

位 24：结束（所选通道/端点的最后一个条目）

位 23:16 **NPTQXSAV[7:0]**：非周期性发送请求队列可用空间 (Non-periodic transmit request queue space available)

指示非周期性发送请求队列中的可用空闲空间大小。该队列既包含 IN 请求，又包含 OUT 请求。

0：非周期性发送请求队列已满

1：1 个位置可用

2：2 个位置可用

n：n 个位置可用 (0 ≤ n ≤ 8)

其他值：保留

位 15:0 **NPTXFSAV[15:0]**：非周期性 Tx FIFO 可用空间 (Non-periodic Tx FIFO space available)

指示非周期性 Tx FIFO 中的可用空闲空间大小。

以 32 位字为单位。

0：非周期性 Tx FIFO 已满

1：1 个字可用

2：2 个字可用

n：n 个位置可用（其中，0 ≤ n ≤ 512）

其他值：保留

54.14.15 OTG 通用模块配置寄存器 (OTG_GCCFG)

OTG general core configuration register

偏移地址：0x038

复位值：0x0000 XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDEN	VBDEN	Res.	Res.	Res.	Res.	Res.
									rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:23 保留，必须保持复位值。

位 22 **IDEN**：USB ID 检测使能 (USB ID detection enable)
使能 ID 引脚状态检测。
0：禁止 ID 引脚状态检测
1：使能 ID 引脚状态检测

位 21 **VBDEN**：USB V_{BUS} 检测使能 (USB V_{BUS} detection enable)
使能 V_{BUS} 感应比较器检测 USB 主机和设备工作模式下的 V_{BUS} 引脚上的 V_{BUS} 有效电平。
如果使能 HNP 和/或 SRP 支持，将自动使能 V_{BUS} 比较器，而与 VBDEN 值无关。
0 = 禁止 V_{BUS} 检测
1 = 使能 V_{BUS} 检测

位 20:19 保留，必须保持复位值。

位 18 保留，必须保持复位值。

位 17 保留，必须保持复位值。

位 16 保留，必须保持复位值。

位 15:4 保留，必须保持复位值。

位 3:1 保留，必须保持复位值。

位 0 保留，必须保持复位值。

54.14.16 OTG 模块 ID 寄存器 (OTG_CID)

OTG core ID register

偏移地址：0x03C

复位值：0x0000 4000

这是一个包含产品 ID 作为复位值的寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRODUCT_ID[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRODUCT_ID[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **PRODUCT_ID[31:0]**：产品 ID 字段 (Product ID field)
可通过应用程序编程的 ID 字段。

54.14.17 OTG 模块 LPM 配置寄存器 (OTG_GLPMCFG)

OTG core LPM configuration register

偏移地址：0x54

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	EN BESL	LPMRCNTSTS[2:0]			SND LPM	LPMRCNT[2:0]			LPMCHIDX[3:0]				L1RSM OK
			rw	r	r	r	rs	rw	rw	rw	rw	rw	rw	rw	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLP STS	LPMRSP[1:0]		L1DS EN	BESLTHRS[3:0]				L1SS EN	REM WAKE	BESL[3:0]				LPM ACK	LPM EN
r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28 **ENBESL**：使能尽力服务延迟 (Enable best effort service latency)
该位用于使能 LPM 勘误表中定义的 BESL 功能：
0：模块按照以下文档中的说明工作：
基于 USB 2.0 规范的 USB 2.0 链路层电源管理补充工程变更通知，2007 年 7 月 16 日
1：模块按照 LPM 勘误表中的说明工作：
USB 2.0 ECN 勘误表：链路层电源管理 (LPM)——2007 年 7 月
注意：本文档只考虑（LPM 勘误表中说明的）更新行为，因此 ENBESL 位必须由应用程序软件置 1。

位 27:25 **LPMRCNTSTS[2:0]**：LPM 重试计数状态 (LPM retry count status)
当前 LPM 序列仍要发送的 LPM 主机重试次数
注意：仅可在主机模式下访问。

位 24 **SNDLPM**：发送 LPM 事务 (Send LPM transaction)

当应用程序软件将该位置 1 时，将发送包含两个令牌 EXT 和 LPM 的 LPM 事务。从设备接收到有效响应（STALL、NYET 或 ACK）或模块发送完编程的 LPM 重试次数后，硬件即将该位清零。

注意：只有当主机连接到本地端口时，该位才可置 1。

注意：仅可在主机模式下访问。

位 23:21 **LPMRCNT[2:0]**：LPM 重试计数 (LPM retry count)

当设备发出 ERROR 响应后，该位表示接收到有效设备响应（STALL、NYET 或 ACK）之前，主机需额外执行的 LPM 重试次数。

注意：仅可在主机模式下访问。

位 20:17 **LPMCHIDX[3:0]**：LPM 通道索引 (LPM Channel Index)

向本地设备发送 LPM 事务时，必须应用到 LPM 事务的通道编号。基于 LPM 通道编号，模块可自动向 LPM 事务插入相应通道中编程的设备地址和端点号。

注意：仅可在主机模式下访问。

位 16 **L1RSMOK**：睡眠状态恢复正常 (Sleep state resume OK)

表示设备或主机可以从睡眠状态启动恢复。该位在 LPM 睡眠 (L1) 状态下有效。它在睡眠模式下经 50 µs ($T_{L1Residency}$) 延时后置 1。

该位在 SLPSTS 为 0 时复位。

1：应用程序或主机可以从睡眠状态启动恢复

0：应用程序或主机无法从睡眠状态启动恢复

位 15 **SLPSTS**：端口睡眠状态 (Port sleep status)

从机模式：

只要 USB 总线上存在睡眠条件，该位就会置 1。当 ACK 响应发送到 LPM 事务并且 $T_{L1TokenRetry}$ 定时器过期时，模块将进入睡眠模式。要停止 PHY 时钟，应用程序必须将 OTG_PCGCCTL 中的 STPPCLK 位置 1，这将触发 PHY 挂起输入信号。

应用程序必须依靠 LPMRSP 中的 SLPSTS 而不是 ACK 来确认切换到睡眠状态。

出现以下情况时，模块会退出睡眠状态：

– USB 线上有活动时

– 当应用程序对 OTG_DCTL 中的 RWUSIG 位执行写操作或者复位或软断开设备时。

主机模式：

模块通过来自设备的 ACK 响应成功将 LPM 事务发送给本地端口后，主机将切换到睡眠 (L1) 状态。该位的读取值反映该端口的当前睡眠状态。

在以下事件后，模块会将该位清零：

– 模块检测到远程 L1 唤醒信号，

– 应用程序将 OTG_HPRT 寄存器中的 PRST 位或 PRES 位置 1，或者

– 应用程序将模块中断寄存器中的“检测到 L1 恢复/远程唤醒中断”位或“检测到断开连接中断”位（分别为 OTG_GINTSTS 中的 WKUPINT 或 DISCINT 位）置 1。

0：模块未处于 L1

1：模块处于 L1

位 14:13 **LPMRSP[1:0]**：LPM 响应 (LPM response)

从机模式：

这两位反映模块对所接收的 LPM 事务的响应。

主机模式：

从本地设备接收、用于 LPM 事务的握手响应

11：ACK

10：NYET

01：STALL

00：ERROR（无握手响应）

位 12 **L1DSEN**：L1 深度睡眠使能 (L1 deep sleep enable)

在 L1 睡眠模式下使能挂起 PHY。为了在 L1 睡眠模式下最大程度节能，所有情况下该位均必须由应用程序软件置 1。

位 11:8 **BESLTHRS[3:0]**：BESL 阈值 (BESL threshold)

从机模式：

当 BESL 值大于或等于 BESL_Thres[3:0] 位于中定义的值时，模块将 PHY 置于 L1 下的深度低功耗模式。

主机模式：

模块将 PHY 置于 L1 下的深度低功耗模式。BESLTHRS[3:0] 用于指定要由主机在 USB 总线上反映的恢复信号的持续时间 ($T_{L1HubDrvResume2}$)。

在主机模式下，BESLTHRS 不得编程为大于 1100b 的值，原因是这将超出最大 $T_{L1HubDrvResume2}$。

Thres[3:0] 主机模式恢复信号的持续时间 (µs)：

0000：75

0001：100

0010：150

0011：250

0100：350

0101：450

0110：950

所有其他值：保留

位 7 **L1SSEN**：L1 浅度睡眠使能 (L1 Shallow Sleep enable)

在 L1 睡眠模式下使能挂起 PHY。为了在 L1 睡眠模式下最大程度节能，所有情况下该位均必须由应用程序软件置 1。

位 6 **REMWAKE**：bRemoteWake 值 (bRemoteWake value)

主机模式：

要在 LPM 事务的 wIndex 位字段中发送的远程唤醒值。

从机模式（只读）：

当 ACK、NYET 或 STALL 响应发送给 LPM 事务时，该位字段用接收的 LPM 令牌 bRemoteWake bmAttribute 进行更新。

位 5:2 **BESL[3:0]**：尽力服务延迟 (Best effort service latency)

主机模式：

要在 LPM 事务中发送的 BESL 值该值还用于发起持续时间为 $T_{L1HubDrvResume1}$ 的恢复信号，以实现主机发起的恢复操作。

从机模式（只读）：

当 ACK、NYET 或 STALL 响应发送给 LPM 事务时，该位字段用接收的 LPM 令牌 BESL bmAttribute 进行更新。

BESL[3:0]T_{BESL} (µs)

0000：125
0001：150
0010：200
0011：300
0100：400
0101：500
0110：1000
0111：2000
1000：3000
1001：4000
1010：5000
1011：6000
1100：7000
1101：8000
1110：9000
1111：10000

位 1 **LPMACK**：LPM 令牌应答使能 (LPM token acknowledge enable)

设备应用程序软件预编程的 LPM 令牌握手响应。

1：ACK

即使已预编程 ACK，也只有在成功完成 LPM 事务后，模块才能以 ACK 进行响应。LPM 事务成功的前提是：

– EXT 令牌或 LPM 令牌中没有 PID/CRC5 错误（否则会出现 ERROR）
– LPM 事务中接收到 Valid bLinkState = 0001B (L1)（否则会出现 STALL）
– 发送队列中没有待处理的数据（否则会出现 NYET）

0：NYET

在以下情况下，为响应 LPM 令牌，会重写预编程的软件位：

– 接收到的 bLinkState 不是 L1（STALL 响应），或者
– 由于任一 LPM 令牌数据包损坏而在其中检测到错误（ERROR 响应）。

注意：仅可在从机模式下访问。

位 0 **LPMEN**：LPM 支持使能 (LPM support enable)

应用程序使用该位控制 OTG 模块的 LPM 功能。

如果模块以不支持 LPM 功能的主机工作，则无法请求所连接的设备或集线器激活 LPM 模式。

如果模块以不支持 LPM 功能的设备工作，则无法响应任何 LPM 事务。

0：不使能 LPM 功能
1：使能 LPM 功能

54.14.18 OTG 主机周期性发送 FIFO 大小寄存器 (OTG_HPTXFSIZ)

OTG host periodic transmit FIFO size register

偏移地址：0x100

复位值：0x0400 0800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PTXFSIZ[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTXSA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **PTXFSIZ[15:0]**：主机周期性 Tx FIFO 深度 (Host periodic Tx FIFO depth)
以 32 位字为单位。
最小值为 16

位 15:0 **PTXSA[15:0]**：主机周期性 Tx FIFO 起始地址 (Host periodic Tx FIFO start address)
该位字段用于配置周期性发送 FIFO RAM 的内存起始地址。

54.14.19 OTG 设备 IN 端点发送 FIFO x 大小寄存器 (OTG_DIEPTXFx)

OTG device IN endpoint transmit FIFO x size register

偏移地址：0x104 + 0x04 * (x - 1)（x = 1 到 8）

复位值：块 1：0x0200 0400

复位值：块 2：0x0200 0600

复位值：块 3：0x0200 0800

复位值：块 4：0x0200 0A00

复位值：块 5：0x0200 0C00

复位值：块 6：0x0200 0E00

复位值：块 7：0x0200 1000

复位值：块 8：0x0200 1200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INEPTXFD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INEPTXSA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **INEPTXFD[15:0]**：IN 端点 Tx FIFO 深度 (IN endpoint Tx FIFO depth)
以 32 位字为单位。
最小值为 16

位 15:0 **INEPTXSA[15:0]**：IN 端点发送 FIFOx RAM 起始地址 (IN endpoint FIFOx transmit RAM start address)

该位字段包含 IN 端点发送 FIFOx 的内存起始地址。该地址必须与 32 位内存位置对齐。

54.14.20 主机模式寄存器

Host-mode registers

除非特别说明，否则寄存器描述中的位值以二进制表示。

主机模式寄存器会影响主机模式下的模块操作。在从机模式下不得访问主机模式寄存器，因为产生的结果不明确。主机模式寄存器可进行如下分类：

54.14.21 OTG 主机配置寄存器 (OTG_HCFG)

OTG host configuration register

偏移地址：0x400

复位值：0x0000 0000

此寄存器将在上电后对模块进行配置。请勿在初始化主机后更改此寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	PERSSCHEDENA	FRLSTEN[1:0]		DESCDMA	Res.	Res.	Res.	Res.	Res.	Res.	Res.
					rw	rw	rw	rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSLSS	FSLSPCS[1:0]	
													r	rw	rw

位 31:27 保留，必须保持复位值。

位 26 **PERSSCHEDENA**：使能定期调度 (Enable periodic scheduling)

仅适用于主机分散/收集 DMA 模式。在模块内使能定期调度。最初，该位是保留位，并且模块不处理任何周期性通道。一旦该位置 1，模块便会准备好开始调度周期性通道并将 OTG_HCFG.PERSCHEDSTAT 置 1。PERSCHEDSTAT 置 1 表示模块已使能定期调度。除非 PERSCHEDSTAT 置 1，否则应用程序不得在 PERSSCHEDENA 置 1 后将其复位。一旦该位置 1，模块便会准备好停止调度周期性通道并将 HCFG.PerSchedStat 复位。在非分散/收集 DMA 模式下，该位保留。

位 25:24 **FRLSTEN[1:0]**：帧列表条目 (Frame list entries)

寄存器中的值指定帧列表中的条目数。该位字段仅在分散/收集 DMA 模式下有效。

2'b00：保留

2'b01：8 个条目

2'b10：16 个条目

2'b11：32 个条目

位 23 **DESCDMA**：在主机模式下使能分散/收集 DMA (Enable scatter/gather DMA in host mode)
应用程序可以在初始化期间将该位置 1，以使能分散/收集 DMA 操作。该位必须在复位后只修改一次。可以进行以下组合编程：
OTG_GAHBCFG.DMAEN=0，OTG_HCFG.DESCDMA=0 => 从机模式
OTG_GAHBCFG.DMAEN=0，OTG_HCFG.DESCDMA=1 => 无效
OTG_GAHBCFG.DMAEN=1，OTG_HCFG.DESCDMA=0 => 缓冲 DMA 模式
OTG_GAHBCFG.DMAEN=1，OTG_HCFG.DESCDMA=1 => 分散/收集 DMA 模式

位 22:3 保留，必须保持复位值。

位 2 **FSLSS**：仅支持 FS 和 LS (FS- and LS-only support)
应用程序使用该位控制模块的枚举速度。使用该位，应用程序可使模块工作为 FS 主机，即使所连接的设备支持 HS 通信也是如此。请勿在初始编程后更改该位字段。

位 1:0 **FSLSPCS[1:0]**：FS/LS PHY 时钟选择 (FS/LS PHY clock select)
当模块处于 FS 主机模式时
01：PHY 时钟以 48 MHz 运行
其他值：保留
当模块处于 LS 主机模式时
00：保留
01：选择 48 MHz PHY 时钟频率
10：选择 6 MHz PHY 时钟频率
11：保留
注意：当设备连上主机时，必须依照所连接设备的速度设置 FSLSPCS（更改该位后，必须进行软件复位）。

54.14.22 OTG 主机帧间隔寄存器 (OTG_HFIR)

OTG host frame interval register

偏移地址：0x404

复位值：0x0000 EA60

此寄存器用于存储 OTG 控制器对已连接设备当前速度所设定的帧间隔信息。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RLD CTRL
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRIVL[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **RLDCTRL**：重载控制 (Reload control)
该位允许在运行期间动态重载 HFIR 寄存器。
0：无法动态重载 HFIR
1：运行期间可动态重载 HFIR。
该位需要在初始配置期间编程，并且不得在运行期间更改其值。

小心： 不建议 RLDCTRL = 0。

位 15:0 **FRIVL[15:0]**：帧间隔 (Frame interval)

应用程序在该位字段编程的值用于指定两个连续 micro-SOF (HS) 或 Keep-Alive 令牌 (LS) 之间的时间间隔。该位字段包含构成所需帧间隔的 PHY 时钟数。只有将主机端口控制和状态寄存器的端口使能位（OTG_HPRT 的 PENA 位）置 1 后，应用程序才能向此寄存器中写入值。如果未对值进行编程，模块将根据在主机配置寄存器的 FS/LS PHY 时钟选择位字段（OTG_HCFG 中的 FSLSPCS）中指定的 PHY 时钟来计算。除非 RLDCTRL 位置 1，否则请勿在初始配置后更改该位字段的值。在这种情况下，FRIVL 将在发生每个 SOF 事件时进行重载。

– 帧间隔 = 125 µs × (FRIVL - 1)高速操作

– 帧间隔 = 1 ms × (FRIVL - 1)低速/全速操作

54.14.23 OTG 主机帧编号/帧剩余时间寄存器 (OTG_HFNUM)

OTG host frame number/frame time remaining register

偏移地址：0x408

复位值：0x0000 3FFF

此寄存器用于指示当前帧编号。它还指示当前帧的剩余时间（以 PHY 时钟数为单位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FTREM[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRNUM[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 **FTREM[15:0]**：帧剩余时间 (Frame time remaining)

指示当前帧的剩余时间（以 PHY 时钟数为单位）。每过去 1 个 PHY 时钟，该位字段递减 1。当值达到零时，该位字段将重新装载帧间隔寄存器中的值，并由模块在 USB 上发送一个新 SOF。

位 15:0 **FRNUM[15:0]**：帧编号 (Frame number)

当在 USB 上发送 1 个新 SOF 时该位字段的值将递增 1，当达到 0x3FFF 时会清零。

54.14.24 OTG_Host 周期性发送 FIFO/队列状态寄存器 (OTG_HPTXSTS)

OTG_Host periodic transmit FIFO/queue status register

偏移地址：0x410

复位值：0x0008 0100

此只读寄存器包含周期性 Tx FIFO 和周期性发送请求队列的空闲空间信息。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PTXQTOP[7:0]								PTXQSAV[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTXFSAVL[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 **PTXQTOP[7:0]**：周期性发送请求队列顶部 (Top of the periodic transmit request queue)
指示周期性传输 Tx 请求队列中 MAC 当前正在处理的项。
该寄存器用于调试。
位 31：奇数/偶数帧
0：以偶数帧发送
1：以奇数帧发送
位 30:27：通道/端点编号
位 26:25：类型
00：IN/OUT
01：零长度数据包
11：禁止通道命令
位 24：结束（所选通道/端点的最后一个条目）

位 23:16 **PTXQSAV[7:0]**：周期性发送请求队列可用空间 (Periodic transmit request queue space available)
指示可供写入的周期性传输发送请求队列的空闲位置的数量。该队列既包含 IN 请求，又包含 OUT 请求。
00：周期性发送请求队列已满
01：1 个位置可用
10：2 个位置可用
bxn：n 个位置可用 (0 ≤ n ≤ 8)
其他值：保留

位 15:0 **PTXFSAVL[15:0]**：周期性发送数据 FIFO 可用空间 (Periodic transmit data FIFO space available)
指示可供写入的周期性传输 Tx FIFO 的空闲位置的数量。
以 32 位字为单位
0000：周期性 Tx FIFO 已满
0001：1 个字可用
0010：2 个字可用
bxn：n 个字可用（其中 0 ≤ n ≤ PTXFD）
其他值：保留

54.14.25 OTG 主机全体通道中断寄存器 (OTG_HAINT)

OTG host all channels interrupt register

偏移地址：0x414

复位值：0x0000 0000

当通道上有事件发生时，主机全体通道中断寄存器会使用模块中断寄存器中的主机通道中断位（OTG_GINTSTS 中的 HCINT 位）中断应用程序。相关内容如*图 689* 所示。每个通道对应 1 个中断位，最多有 16 个位。当应用程序通过相应主机通道 x 中断寄存器清零中断时，该寄存器中的位也会清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HAINT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **HAINT[15:0]**：通道中断 (Channel interrupt)
每个通道对应一位：通道 0 对应位 0，通道 15 对应位 15

54.14.26 OTG 主机全体通道中断屏蔽寄存器 (OTG_HAINTMSK)

OTG host all channels interrupt mask register

偏移地址：0x418

复位值：0x0000 0000

主机全体通道中断屏蔽寄存器与主机全体通道中断寄存器结合使用，进而在通道上发生事件时中断应用程序。每个通道对应 1 个中断屏蔽位，最多有 16 个位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HAINTM[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **HAINTM[15:0]**：通道中断屏蔽 (Channel interrupt mask)
0：屏蔽中断
1：使能中断
每个通道对应一位：通道 0 对应位 0，通道 15 对应位 15

54.14.27 OTG 主机帧列表基址寄存器 (OTG_HFLBADDR)

OTG host frame list base address register

偏移地址：0x41C

复位值：0x0000 0000

该寄存器保存帧列表信息（分散/收集模式）的起始地址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HFLBADDR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HFLBADDR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **HFLBADDR[31:0]**：帧列表的起始地址（分散/收集模式）。
该寄存器仅用于同步和中断通道。

54.14.28 OTG 主机端口控制和状态寄存器 (OTG_HPRT)

OTG host port control and status register

偏移地址：0x440

复位值：0x0000 0000

该寄存器仅在主机模式下可用。当前，OTG 主机仅支持一个端口。

该寄存器包含与 USB 端口相关的信息，例如 USB 复位、使能、挂起、恢复、连接状态以及测试模式。相关内容在*图 689* 中进行了说明。该寄存器中的 rc_w1 位可通过模块中断寄存器中的主机端口中断位（OTG_GINTSTS 中的 HPRTINT 位）触发应用程序中断。发生端口中断时，应用程序必须读取该寄存器，并将引起中断的位清零。对于 rc_w1 位，应用程序必须向该位写入 1 以清除该中断。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSPD[1:0]		PTCTL[3]
													r	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTCTL[2:0]			PPWR	PLSTS[1:0]		Res.	PRST	PSUSP	PRES	POC CHNG	POCA	PEN CHNG	PENA	PCDET	PCSTS
rw	rw	rw	rw	r	r		rw	rs	rw	rc_w1	r	rc_w1	rc_w1	rc_w1	r

位 31:19 保留，必须保持复位值。

位 18:17 **PSPD[1:0]**：端口速度 (Port speed)
指示连接到该端口的设备的速度。
01：全速
10：低速
11：保留
00：高速

位 16:13 **PTCTL[3:0]**：端口测试控制 (Port test control)
应用程序向该位字段写入一个非零值，以将端口置于测试模式，同时端口上会产生对应模式的信号。
0000：禁止测试模式
0001：Test_J 模式
0010：Test_K 模式
0011：Test_SE0_NAK 模式
0100：Test_Packet 模式
0101：Test_Force_Enable
其他值：保留

位 12 **PPWR**：端口电源 (Port power)
应用程序使用该位字段控制该端口的电源，且发生过流情况时，模块会将该位清零。
0：掉电
1：通电

位 11:10 **PLSTS[1:0]**：端口线状态 (Port line status)
指示 USB 数据线的当前逻辑电平
位 10：OTG_DP 的逻辑电平
位 11：OTG_DM 的逻辑电平

位 9 保留，必须保持复位值。

位 8 **PRST**：端口复位 (Port reset)

应用程序将该位置 1 时，会在该端口上启动复位序列。应用程序必须为复位周期定时，并在复位序列完成后将该位清零。

0：端口未处于复位状态

1：端口处于复位状态

应用程序必须将该位置 1 并最少保持 10 ms，以在端口上启动复位。除所需的最少持续时间之外，在将该位清零前，应用程序可将该位置 1 的状态再保持 10 ms，即使 USB 标准并没有设置最大限制。

高速：50 ms

全速/低速：10 ms

位 7 **PSUSP**：端口挂起 (Port suspend)

应用程序将该位置 1 以将此端口置于挂起模式。只有该位置 1 时，模块才会停止发送 SOF。要停止 PHY 时钟，应用程序必须将端口时钟停止位置 1，这会使能 PHY 的挂起输入引脚。

该位的读取值反映该端口的当前挂起状态。检测到远程唤醒信号，或者应用程序将此寄存器中的端口复位位或端口恢复位置 1 后，模块可将该位清零；或应用程序将模块中断寄存器中的检测到恢复/远程唤醒中断位或检测到断开连接中断位（分别为 OTG_GINTSTS 中的 WKUPINT 或 DISCINT）置 1，模块也可将该位清零。

0：端口未处于挂起模式

1：端口处于挂起模式

位 6 **PRES**：端口恢复 (Port resume)

应用程序将该位置 1 以在该端口上驱动恢复信号。模块会持续驱动恢复信号直到应用程序将该位清零。

如模块中断寄存器中的检测到端口恢复/远程唤醒中断位（OTG_GINTSTS 中的 WKUPINT 位）指示，如果模块检测到 USB 远程唤醒序列，则开始驱动恢复信号，而无需应用程序进行干预；如果模块检测到断开连接的情况，则将该位清零。该位的读取值指示当前模块是否正在驱动恢复信号。

0：不驱动恢复信号

1：驱动恢复信号

当 LPM 已使能且模块处于 L1 状态时，该位的影响如下：

1.应用程序将该位置 1 以在该端口上驱动恢复信号。

2.模块继续驱动恢复信号，直至达到 OTG_GLPMCFG 寄存器的 BESLTHRS[3:0] 位字段预定的时间。

3.如模块中断寄存器中的“检测到端口 L1 恢复/远程 L1 唤醒中断”位（OTG_GINTSTS 中的 WKUPINT 位）指示，如果模块检测到 USB 远程唤醒序列，则开始驱动恢复信号，而无需应用程序进行干预，并在恢复结束时，将该位清零。该位可以由模块和应用程序置 1 和清零。即使主机未连接任何设备，该位也可由模块清零。

位 5 **POCCHNG**：端口过流变化 (Port overcurrent change)

该寄存器中端口过流激活位（位 4）状态发生变化时，模块将该位置 1。

位 4 **POCA**：端口过流激活 (Port overcurrent active)

该位指示端口的过流状况。

0：无过流状况

1：有过流状况

位 3 **PENCHNG**：端口使能/禁止变化 (Port enable/disable change)

该寄存器中的端口使能位 2 的状态发生变化时，模块将该位置 1。

位 2 **PENA**：端口使能 (Port enable)

端口执行复位序列后，只能由模块使能，并且可以由过流状况、断开连接状况或应用程序将该位清零来禁止。应用程序无法通过对寄存器执行写操作将该位置 1。只能将该位清零来禁止端口。对该位的操作不会触发应用程序的任何中断。

0：禁止端口

1：使能端口

位 1 **PCDET**：检测到端口连接 (Port connect detected)

当检测到设备连接时，模块将该位置 1，以使用模块中断寄存器中的主机端口中断位（OTG_GINTSTS 中的 HPRTINT 位）触发应用程序的中断。应用程序必须将该位置 1 才可清除该中断。

位 0 **PCSTS**：端口连接状态 (Port connect status)

0：端口未连接设备

1：端口已连接设备

54.14.29 OTG 主机通道 x 特性寄存器 (OTG_HCCHARx)

OTG host channel x characteristics register

偏移地址：0x500 + 0x20 * x（x = 0 到 15）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHENA	CHDIS	ODD FRM	DAD[6:0]							MCNT[1:0]		EPTYP[1:0]		LSDEV	Res.
rs	rs	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPDIR	EPNUM[3:0]				MPSIZ[10:0]										
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **CHENA**：通道使能 (Channel enable)

使能分散/收集模式时：

1'b0：表示描述符结构尚未就绪

1'b1：表示描述符结构和包含数据的数据缓冲区已建立，且该通道可以访问描述符

禁止分散/收集模式时：此字段由应用程序软件置 1，并由 OTG 主机硬件清零。

0：禁止通道

1：使能通道

位 30 **CHDIS**：通道禁止 (Channel disable)

应用程序将该位置 1 以停止通过通道发送/接收数据，即使通过该通道的传输还未完成，停止操作仍然生效。应用程序必须等待禁止通道的中断以确认通道已经被禁止。

位 29 **ODDFRM**：奇数帧 (Odd frame)

该位字段由应用程序置 1（或复位），以分别指示 OTG 主机必须传输奇数帧或偶数帧。该位字段只适用于周期性（同步和中断）事务。

0：偶数帧

1：奇数帧 (Odd frame)

位 28:22 **DAD[6:0]**：设备地址 (Device address)

该位字段用于指定要与该主机通信的特定设备。

位 21:20 **MCNT[1:0]**：多重计数 (Multicount)

该位字段向主机指示该周期性端点每帧必须执行的事务数。该位字段不用于非周期性传输。

00：保留。对该位字段的操作会产生不明确的结果

01：1 个事务

10：该端点每帧需要发出 2 个事务

11：该端点每帧需要发出 3 个事务

注意：该位字段至少须置为 01。

位 19:18 **EPTYP[1:0]**：端点类型 (Endpoint type)
指示选择的传输类型。
00：控制
01：同步
10：批量
11：中断

位 17 **LSDEV**：低速设备 (Low-speed device)
该位字段由应用程序置 1，表示此通道正在与一个低速设备进行通信。

位 16 保留，必须保持复位值。

位 15 **EPDIR**：端点方向 (Endpoint direction)
指示通信事务的方向是输入还是输出。
0：OUT
1：IN

位 14:11 **EPNUM[3:0]**：端点编号 (Endpoint number)
指示要与该主机通道通信的 USB 设备的端点号。

位 10:0 **MPSIZ[10:0]**：最大数据包大小 (Maximum packet size)
指示与该主机通道通信的设备端点的最大数据包大小。

54.14.30 OTG 主机通道 x 分离控制寄存器 (OTG_HCSPLTx)

OTG host channel x split control register

偏移地址：0x504 + 0x20 * x（x = 0 到 15）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPLIT EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP LSPLT
rw															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XACTPOS[1:0]		HUBADDR[6:0]							PRTADDR[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **SPLITEN**：分离使能 (Split enable)
应用程序将该位置 1 时，指示允许该通道执行分离通信事务。

位 30:17 保留，必须保持复位值。

位 16 **COMPLSPLT**：执行完全分离 (Do complete split)
应用程序将该位置 1 时，可请求 OTG 主机执行完全分离通信事务。

位 15:14 **XACTPOS[1:0]**：事务位置 (Transaction position)
该位字段用于决定随各 OUT 事务发送全部、第一个、中间还是最后一个有效负载。
11：全部。指此事务的全部数据有效负载（小于或等于 188 字节）
10：起始。指此事务的第一个数据有效负载（大于 188 字节）
00：中间。指此事务的中间有效负载（大于 188 字节）
01：结尾。指此事务的最后一个有效负载（大于 188 字节）

位 13:7 **HUBADDR[6:0]**：集线器地址 (Hub address)
此字段存储事务转发器的集线器设备地址。

位 6:0 **PRTADDR[6:0]**：端口地址 (Port address)
该位字段是接收方事务转发器的端口编号。

54.14.31 OTG 主机通道 x 中断寄存器 (OTG_HCINTx)

OTG host channel x interrupt register

偏移地址：0x508 + 0x20 * x（x = 0 到 15）

复位值：0x0000 0000

该寄存器指示在出现 USB 和 AHB 相关事件时通道的状态。相关内容在*图 689* 中进行了说明。当模块中断寄存器中的主机通道中断位（OTG_GINTSTS 中的 HCINT 位）置 1 时，应用程序必须读取该寄存器。在对寄存器执行读操作之前，应用程序必须先读取主机全体通道中断 (OTG_HAINT) 寄存器，以获取主机通道 x 中断寄存器的准确通道编号。应用程序必须将该寄存器中的相应位清零，才能将 OTG_HAINT 和 OTG_GINTSTS 寄存器中的对应位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DESCLSTROLL	XCSXACTERR	BNA	DTERR	FRMOR	BBERR	TXERR	NYET	ACK	NAK	STALL	AHBERR	CHH	XFRC
		rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:14 保留，必须保持复位值。

位 13 **DESCLSTROLL**：描述符翻转中断 (Descriptor rollover interrupt)。
该位仅在使能了分散/收集 DMA 模式时有效。当相应的通道描述符列表翻转时，模块会将该位置 1。对于非分散/收集 DMA 模式，该位保留。

位 12 **XCSXACTERR**：事务错误过多 (Excessive transaction error)。
该位仅在使能了分散/收集 DMA 模式时有效。当 USB 总线上连续发生 3 个事务错误时，模块会将该位置 1。同步通道不会生成 XCS_XACT_ERR。对于非分散/收集 DMA 模式，该位保留。

位 11 **BNA**：缓冲区不可用中断 (Buffer not available interrupt)。
该位仅在使能了分散/收集 DMA 模式时有效。当所访问的描述符没有准备好供模块处理时，模块将生成该中断。同步通道不会生成 BNA 中断。对于非分散/收集 DMA 模式，该位保留。

位 10 **DTERR**：数据翻转错误 (Data toggle error)。在分散/收集 DMA 模式下，屏蔽因该位产生的中断。

位 9 **FRMOR**：帧上溢 (Frame overrun)。在分散/收集 DMA 模式下，屏蔽因该位产生的中断。

位 8 **BBERR**：串扰错误 (Babble error)。在分散/收集 DMA 模式下，屏蔽因该位产生的中断。

位 7 **TXERR**：事务错误 (Transaction error)。在分散/收集 DMA 模式下，屏蔽因该位产生的中断。
指示 USB 上发生下列错误之一：
CRC 校验失败
超时
位填充错误
错误的 EOP

位 6 **NYET**：尚未就绪响应已接收中断 (Not yet ready response received interrupt)。在分散/收集 DMA 模式下，屏蔽因该位产生的中断。

位 5 **ACK**：收到/发出 ACK 响应中断 (ACK response received/transmitted interrupt)。在分散/收集 DMA 模式下，屏蔽因该位产生的中断。

位 4 **NAK**：收到 NAK 响应中断 (NAK response received interrupt)。在分散/收集 DMA 模式下，屏蔽因该位产生的中断。

位 3 **STALL**：收到 STALL 响应中断 (STALL response received interrupt)。在分散/收集 DMA 模式下，屏蔽因该位产生的中断。

位 2 **AHBERR**：AHB 错误 (AHB error)

仅当处于内部 DMA 模式下且 AHB 读/写操作期间发生 AHB 错误时才生成此错误。应用程序可读取相应的 DMA 通道地址寄存器来获取错误地址。

位 1 **CHH**：通道停止 (Channel halted)

在非分散/收集 DMA 模式下，指示因任意 USB 事务错误或为响应应用程序的禁止请求而导致传输非正常结束。

在分散/收集 DMA 模式下，这表明传输由于以下任一原因完成：

– 描述符中设置了 EOL
– AHB 错误
– 事务错误过多。
– 响应应用程序发出的禁止请求
– 串扰
– Stall

位 0 **XFRC**：传输完成 (Transfer completed)。

未出现任何错误，正常完成传输。

54.14.32 OTG 主机通道 x 中断屏蔽寄存器 (OTG_HCINTMSKx)

OTG host channel x interrupt mask register

偏移地址：0x50C + 0x20 * x（x = 0 到 15）

复位值：0x0000 0000

此寄存器反映了先前部分中介绍的各通道状态的屏蔽情况。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DESCLSTROLLMSK	Res.	BNAMSK	DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	AHBERRM	CHHM	XFRCM
		rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:14 保留，必须保持复位值。

位 13 **DESCLSTROLLMSK**：描述符翻转中断屏蔽寄存器 (Descriptor rollover interrupt mask register)。

该位仅在使能了分散/收集 DMA 模式时有效。

在非分散/收集 DMA 模式下，该位保留。

位 12 保留，必须保持复位值。

位 11 **BNAMSK**：缓冲区不可用中断屏蔽寄存器 (Buffer not available interrupt mask register)。
该位仅在使能了分散/收集 DMA 模式时有效。
在非分散/收集 DMA 模式下，该位保留

位 10 **DTERRM**：数据翻转错误屏蔽 (Data toggle error mask)。在分散/收集 DMA 模式下，屏蔽因该位产生的中断。
0：屏蔽中断
1：使能中断

位 9 **FRMORM**：帧上溢屏蔽 (Frame overrun mask)。在分散/收集 DMA 模式下，屏蔽因该位产生的中断。
0：屏蔽中断
1：使能中断

位 8 **BBERRM**：串扰错误屏蔽 (Babble error mask)。在分散/收集 DMA 模式下，屏蔽因该位产生的中断。
0：屏蔽中断
1：使能中断

位 7 **TXERRM**：通信事务错误屏蔽 (Transaction error mask)。在分散/收集 DMA 模式下，屏蔽因该位产生的中断。
0：屏蔽中断
1：使能中断

位 6 **NYET**：响应接收中断屏蔽 (response received interrupt mask)。在分散/收集 DMA 模式下，屏蔽因该位产生的中断。
0：屏蔽中断
1：使能中断

位 5 **ACKM**：ACK 响应接收/发送中断屏蔽 (ACK response received/transmitted interrupt mask)。在分散/收集 DMA 模式下，屏蔽因该位产生的中断。
0：屏蔽中断
1：使能中断

位 4 **NAKM**：NAK 响应接收中断屏蔽 (NAK response received interrupt mask)。在分散/收集 DMA 模式下，屏蔽因该位产生的中断。
0：屏蔽中断
1：使能中断

位 3 **STALLM**：STALL 响应接收中断屏蔽 (STALL response received interrupt mask)。在分散/收集 DMA 模式下，屏蔽因该位产生的中断。
0：屏蔽中断
1：使能中断

位 2 **AHBERRM**：AHB 错误 (AHB error)。在主机的分散/收集 DMA 模式下，由于 OTG_HCINTx 中的相应位置 1，将不会生成中断。
0：屏蔽中断
1：使能中断

位 1 **CHHM**：通道停止中断屏蔽 (Channel halted mask)
0：屏蔽中断
1：使能中断

位 0 **XFRCM**：传输完成中断屏蔽 (Transfer completed mask)
0：屏蔽中断
1：使能中断

54.14.33 OTG 主机通道 x 传输大小寄存器 (OTG_HCTSIZx)

OTG host channel x transfer size register

偏移地址：0x510 + 0x20 * x（x = 0 到 15）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DOPNG	DPID[1:0]		PKTCNT[9:0]										XFRSIZ[18:16]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XFRSIZ[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **DOPNG**：执行 Ping 操作 (Do Ping)

该位仅用于 OUT 传输。将该位置 1 可指示主机执行 PING 协议。

注意：请勿针对 IN 传输将该位置 1。如果针对 IN 传输将该位置 1，将禁止通道。

位 30:29 **DPID[1:0]**：数据 PID (Data PID)

应用程序在该位字段设置数据通信的初始同步 PID。后续传输的数据 PID 由主机硬件维护。

00：DATA0

01：DATA2

10：DATA1

11：SETUP（控制传输）/MDATA（非控制传输）

位 28:19 **PKTCNT[9:0]**：数据包计数 (Packet count)

应用程序在该位字段中设置将要发送 (OUT) 或接收 (IN) 的数据包数。

主机每成功发送或接收一个 OUT/IN 数据包便递减一次计数值。此值达到 0 后，将中断应用程序来指示操作正常完成。

位 18:0 **XFRSIZ[18:0]**：传输数据量大小 (Transfer size)

对于 OUT 操作，该位字段为传输期间主机发送的数据字节数。

对于 IN 操作，该位字段为应用程序保留给传输的缓冲区大小。对于 IN 事务（周期性和非周期性），应用程序会将该位字段编程为最大数据包大小的整数倍。

54.14.34 OTG 主机通道 x 传输大小寄存器 (OTG_HCTSIZSGx)

OTG host channel x transfer size register

偏移地址：0x510 + 0x20 * x（x = 0 到 15）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DOPNG	PID[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NTD[7:0]								SCHED_INFO[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **DOPNG**：执行 Ping 操作 (Do Ping)

该位仅用于 OUT 传输。将该位置 1 可指示主机执行 PING 协议。

注意：请勿针对 IN 传输将该位置 1。如果针对 IN 传输将该位置 1，将禁止通道。

位 30:29 **PID[1:0]**：Pid

应用程序在该位字段设置数据通信的初始同步 PID。后续传输的数据 PID 由主机硬件维护。

00：DATA0
01：DATA2
10：DATA1
11：MDATA（非控制传输）/SETUP（控制传输）

位 28:16 保留，必须保持复位值。

位 15:8 **NTD[7:0]**：传输描述符的数量 (Number of transfer descriptors)

非同步：该值以描述符数量为单位。列表中可以显示的最大描述符数量为 64。值范围为 0 到 63。

0： 1 个描述符
1： 2 个描述符
...
63： 64 个描述符
其他值： 保留

该位字段指示列表中显示的描述符的总数。内核将在处理该列表的 NTD 描述符数量后回卷。

同步：该位字段指示列表中显示的描述符数量。

FS 的可能值为：

1： 2 个描述符
3： 4 个描述符
7： 8 个描述符
15： 16 个描述符
31： 32 个描述符
63： 64 个描述符
其他值：保留

HS 的可能值为：

7： 8 个描述符
15： 16 个描述符
31： 32 个描述符
63： 64 个描述符
127： 128 个描述符
255： 256 个描述符
其他值： 保留

位 7:0 **SCHED_INFO[7:0]**：调度信息 (Schedule information)

该 8 位寄存器中的每一位都指示相应微帧的调度。位 0 指示该帧中第 1 个微帧的调度，位 7 指示第 8 个微帧的调度。值 0b11111111 表示相应的中断通道调度为在该帧的每个微帧中发出一个令牌。值 0b10101010 表示相应的中断通道调度为从第二个微帧开始，每隔一个微帧发出一个令牌。请注意，该位字段仅适用于周期性（同步和中断）通道。

54.14.35 OTG 主机通道 x DMA 地址寄存器（缓冲 DMA 模式）[复用] (OTG_HCDMAx)

OTG host channel x DMA address register in buffer DMA

偏移地址：0x514 + 0x20 * x（x = 0 到 15）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAADDR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAADDR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DMAADDR[31:0]**：DMA 地址 (DMA address)

该位字段存储主机从设备端点获取数据或往设备端点发送数据所用 DMA 传输的内存的地址。每次进行 AHB 传输，该寄存器都会递增。

54.14.36 OTG 主机通道 x DMA 地址寄存器（分散/收集 DMA）[复用] (OTG_HCDMASGx)

OTG host channel x DMA address register in scatter/gather DMA

偏移地址：0x514 + 0x20 * x（x = 0 到 15）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMASG[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMASG[15:3]													Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			

位 31:3 **DMASG[31:3]**：DMA 分散/收集信息 (DMA scatter/gather information)

DMASG 信息由两部分组成，即 DMASG_ADDR = DMASG[31:N] 和 DMASG_CTD = DAMSG[N-1:3]。

非同步时 (N = 9)：

DAMSG_ADDR 位字段保存 512 字节页面的起始地址。列表中的第一个描述符应位于该地址。第一个描述符可能已准备好也可能未准备好。内核从 CTD 值开始处理列表。

DMASG_CTD 位字段保存对应于描述符数量的值。值范围为 0 (1 个描述符) 到 63 (64 个描述符)。该位字段指示列表中当前处理的描述符。应用和内核都可以更新该位字段。例如，如果应用在编程 CTD = 5 之后使能通道，则内核将开始处理第 6 个描述符。将 DMAAddr 加上 8 * 5 = 40 字节（十进制）即可获得地址。

同步时（N = 4 到 11）：

DAMSG_ADDR 位字段保存同步描述符所在位置的 2*(NTD+1) 字节地址，其中 N 基于 NTD，如下表所列：

HS ISOC; NTD = 7: N = 6
HS ISOC; NTD = 15: N = 7
HS ISOC; NTD = 31: N = 8
HS ISOC; NTD = 63: N = 9
HS ISOC; NTD = 127: N = 10
HS ISOC; NTD = 255: N = 11
FS ISOC; NTD = 1: N = 4
FS ISOC; NTD = 3: N = 5
FS ISOC; NTD = 7: N = 6
FS ISOC; NTD = 15: N = 7
FS ISOC; NTD = 31: N = 8
FS ISOC; NTD = 63: N = 9

DMASG_CTD 位字段基于当前帧/（微）帧值。需要由应用设置为零。

位 2:0 保留，必须保持复位值。

54.14.37 OTG 主机通道 n DMA 地址缓冲区寄存器 (OTG_HCDMABx)

OTG host channel-n DMA address buffer register

偏移地址：0x51C + 0x20 * x（x = 0 到 15）

复位值：0x0000 0000 (0x0000 0000)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HCDMAB[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCDMAB[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **HCDMAB[31:0]**：DMA 地址 (DMA address)

该寄存器保存当前缓冲区地址（分散/收集模式）。

54.14.38 从机模式寄存器

Device-mode registers

模块每次切换到从机模式时都必须对这些寄存器进行编程。

54.14.39 OTG 设备配置寄存器 (OTG_DCFG)

OTG device configuration register

偏移地址：0x800

复位值：0x0220 0000

此寄存器在上电、执行某些控制命令或枚举后，会将模块配置为从机模式。请勿在初始编程后更改该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	PERSCHIVL[1:0]		DESCDMA	Res.	Res.	Res.	Res.	Res.	Res.	Res.
						rw	rw	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRATIM	Res.	Res.	PFIVL[1:0]		DAD[6:0]							Res.	NZLSOHSK	DSPD[1:0]	
rw			rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25:24 **PERSCHIVL[1:0]**：周期性调度间隔 (Periodic schedule interval)

该位字段指定内部 DMA 引擎必须为获取周期性 IN 端点数据分配的时间。根据周期性端点数量的不同，必须将此值指定为 25%、50% 或 75%（微）帧。

- 只要存在活动的周期性端点，内部 DMA 引擎便会为获取周期性 IN 端点数据分配指定的时间
- 没有任何活动周期性端点时，内部 DMA 引擎将仅用于非周期性端点并忽略该位字段
- 经过一（微）帧中指定的时间后，DMA 切换为获取非周期性端点上的传输数据

00：25%（微）帧
01：50%（微）帧
10：75%（微）帧
11：保留

注意：周期性调度间隔 (PERSCHIVL) 必须针对分散/收集 DMA 模式进行编程。

位 23 **DESCDMA**：在从机模式下使能分散/收集 DMA (Enable scatter/gather DMA in device mode)

应用程序可以在初始化期间将该位置 1，以使能分散/收集 DMA 操作。该位必须在复位后只修改一次。可以进行以下组合编程：
OTG_GAHBCFG.DMAEN=0，OTG_DCFG.DESCDMA=0 => 从机模式
OTG_GAHBCFG.DMAEN=0，OTG_DCFG.DESCDMA=1 => 无效
OTG_GAHBCFG.DMAEN=1，OTG_DCFG.DESCDMA=0 => 缓冲区 DMA 模式
OTG_GAHBCFG.DMAEN=1，OTG_DCFG.DESCDMA=1 => 分散/收集 DMA 模式

位 22:16 保留，必须保持复位值。

位 15 **ERRATIM**：不定错误中断屏蔽 (Erratic error interrupt mask)
1：发送不定错误时不触发早期挂起中断
0：发生不定错误时生成早期挂起中断

位 14 保留，必须保持复位值。

位 13 保留，必须保持复位值。

位 12:11 **PFIVL[1:0]**：周期性帧间隔 (Periodic frame interval)
指示一帧内必须使用周期性帧中断通知应用程序的时间点。此功能可用于确定该帧的所有同步通信是否完成。
00：80% 帧间隔
01：85% 帧间隔
10：90% 帧间隔
11：95% 帧间隔

位 10:4 **DAD[6:0]**：设备地址 (Device address)
应用程序必须在执行每个 SetAddress 控制命令后根据命令参数对该位字段进行设置。

位 3 保留，必须保持复位值。

位 2 **NZLSOHSK**：非零长度状态 OUT 握手信号 (Non-zero-length status OUT handshake)
在控制传输状态阶段的 OUT 事务期间，当模块收到非零长度数据包后，应用程序可以使用该位字段选择要发送的握手信号。
1：收到非零长度状态 OUT 事务时，回复 STALL 握手信号，收到的 OUT 数据包不发送给应用程序。
0：将收到的 OUT 数据包（零长度或非零长度）发送给应用程序，并基于设备端点控制寄存器中端点的 NAK 和 STALL 位回复握手信号。

位 1:0 **DSPD[1:0]**：设备速度 (Device speed)
指示应用程序要求模块进行枚举所采用的速度，或应用程序支持的最大速度。但是，实际总线速度只有在完成 chirp 序列后才能确定，同时此速度基于与模块连接的 USB 主机的速度。
00：高速
01：全速
10：保留
11：保留

54.14.40 OTG 设备控制寄存器 (OTG_DCTL)

OTG device control register

偏移地址：0x804

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSBESLRJCT	ENCONTONBNA	Res.
													rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	POPRGDNE	CGONAK	SGONAK	CGINAK	SGINAK	TCTL[2:0]			GONSTS	GINSTS	SDIS	RWUSIG
				rw	w	w	w	w	rw	rw	rw	r	r	rw	rw

位 31:19 保留，必须保持复位值。

位 18 **DSBESLRJCT**：深度睡眠 BESL 拒绝 (Deep sleep BESL reject)
模块拒绝 BESL 值大于编程的 BESL 阈值的 LPM 请求。将为 BESL 值大于 BESL 阈值的 LPM 令牌发送 NYET 响应。默认情况下，禁用深度睡眠 BESL 拒绝功能。

位 17 ENCONTONBNA：使能在 BNA 时继续工作 (Enable continue on BNA)
该位用于使能模块在面对批量 OUT 和 INTR OUT 端点的 BNA 时继续工作。使能该功能后，当批量 OUT 或 INTR OUT 端点接收到 BNA 中断时，模块会在端点重新使能端点后开始处理导致 BNA 中断的描述符。
0：收到 BNA 中断后，模块会禁止端点。当应用程序重新使能端点时，模块从 DOEPDMA 描述符开始处理。
1：收到 BNA 中断后，模块会禁止端点。当应用程序重新使能端点时，内核从接收到 BNA 中断的描述符开始处理。该位在复位后为一次性可编程，与 OTG_DCTL 寄存器的所有其他位一样。

位 16:12 保留，必须保持复位值。

位 11 **POPRGDNE**：上电编程完成 (Power-on programming done)
应用程序使用该位指示寄存器从掉电模式唤醒后已完成编程。

位 10 **CGONAK**：将全局 OUT NAK 清零 (Clear global OUT NAK)
对该位字段写入 1 会将全局 OUT NAK 清零。

位 9 **SGONAK**：将全局 OUT NAK 置 1 (Set global OUT NAK)
对该位字段写入 1 会将全局 OUT NAK 置 1。
应用程序使用该位在所有 OUT 端点发送 NAK 握手信号。
应用程序只有确定模块中断寄存器中的全局 OUT NAK 有效位（OTG_GINTSTS 中 GONAKEFF 位）已清零时，才可以将该位置 1。

位 8 **CGINAK**：将全局 IN NAK 清零 (Clear global IN NAK)
对该位字段写入 1 会将全局 IN NAK 清零。

位 7 **SGINAK**：将全局 IN NAK 置 1 (Set global IN NAK)
对该位字段写入 1 会将全局非周期性 IN NAK 置 1。应用程序使用该位使所有非周期性 IN 端点发送 NAK 握手信号。
应用程序只有确定模块中断寄存器中的全局 IN NAK 有效位（OTG_GINTSTS 中的 GINAKEFF 位）已清零时，才可以将该位置 1。

位 6:4 **TCTL[2:0]**：测试控制 (Test control)
000：禁止测试模式
001：Test_J 模式
010：Test_K 模式
011：Test_SE0_NAK 模式
100：Test_Packet 模式
101：Test_Force_Enable
其他值：保留

位 3 **GONSTS**：全局 OUT NAK 状态 (Global OUT NAK status)
0：将根据 FIFO 状态和 NAK 和 STALL 位设置发送握手信号。
1：无论 Rx FIFO 中是否还有空闲空间都不接收数据。除 SETUP 事务之外，对所有收到的数据包回复 NAK 握手信号。所有同步类型的 OUT 数据包都将被丢弃。

位 2 **GINSTS**：全局 IN NAK 状态 (Global IN NAK status)
0：将根据发送 FIFO 中的数据可用性回复握手信号。
1：使所有非周期性 IN 端点回复 NAK 握手信号，无需考虑发送 FIFO 中的数据可用性。

位 1 **SDIS**：软断开

应用程序使用该位向 USB OTG 模块发出执行软断开的信号。该位置 1 时，主机不会看到设备已连接，且该设备也不会接收 USB 上的信号。在应用程序将该位清零之前，模块会保持断开状态。

0：正常工作。该位在软断开之后清零，会使主机收到设备已连接的事件。重新连接设备之后，USB 主机会重新启动设备枚举。

1：使主机收到设备断开连接的事件。

位 0 **RWUSIG**：远程唤醒信号 (Remote wake-up signaling)

应用程序将该位置 1 时，模块会启动远程发送信号，以唤醒 USB 主机。应用程序必须将该位置 1 以使模块退出挂起状态。根据 USB 2.0 规范，应用程序必须在将该位置 1 之后的 1 ms 到 15 ms 内将其清零。

如果 LPM 已使能并且模块处于 L1（睡眠）状态，则当应用程序将该位置 1 时，模块会启动 L1 远程信号来唤醒 USB 主机。应用程序必须将该位置 1 以使模块退出睡眠状态。按照 LPM 规范中的规定，在应用程序将该位置 1 后经过 50 µs ($T_{L1DevDrvResume}$)，硬件会自动将该位清零。当前一个 LPM 事务中的 bRemoteWake 为零时（请参见 GLPMCFG 寄存器中的 REMWAKE 位），应用程序不得将该位置 1。

表 516 显示了为使 USB 主机检测到设备断开连接所需的软断开 (SDIS) 位置 1 的最短时间（取决于设备状态）。为了协调时钟抖动，建议应用程序在指定的最小时间基础上再加入一段延迟。

表 516. 软断开的最小时间

运行速度	设备状态	最小时间
全速	挂起	1 ms + 2.5 µs
全速	空闲	2.5 µs
全速	非空闲或挂起（正在通信时）	2.5 µs
高速	非空闲或挂起（正在通信时）	125 µs

54.14.41 OTG 设备状态寄存器 (OTG_DSTS)

OTG device status register

偏移地址：0x808

复位值：0x0000 0010

此寄存器指示模块在出现 USB 相关事件时的状态。必须在发生设备全体中断寄存器 (OTG_DAINT) 中断时读取它。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEVLNSTS[1:0]		FNSOF[13:8]					
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FNSOF[7:0]								Res.	Res.	Res.	Res.	EERR	ENUMSPD[1:0]		SUSP STS
r	r	r	r	r	r	r	r					r	r	r	r

位 31:24 保留，必须保持复位值。

位 23:22 **DEVLNSTS[1:0]**：设备线状态 (Device line status)

指示 USB 数据线的当前逻辑电平。

位 [23]：D+ 的逻辑电平

位 [22]：D- 的逻辑电平

位 21:8 **FNSOF[13:0]**：接收 SOF 的帧编号 (Frame number of the received SOF)

位 7:4 保留，必须保持复位值。

位 3 **EERR**：不定错误 (Erratic error)

模块将该位置 1 以报告任何不定错误。

由于不定错误，OTG 控制器会进入挂起状态，并且会以 OTG_GINTSTS 寄存器的早期挂起位（OTG_GINTSTS 中的 ESUSP 位）为应用程序生成一个中断。如果早期挂起中断是由不定错误触发，则应用程序只能执行软断开以恢复通信。

位 2:1 **ENUMSPD[1:0]**：枚举速度 (Enumerated speed)

指示 OTG 控制器通过 chirp 序列检测速度后被枚举成的速度。

00：高速

01：全速

11：保留

其他值：保留

位 0 **SUSPSTS**：挂起状态 (Suspend status)

在从机模式下，只要在 USB 上检测到挂起状态，该位就会置 1。当 USB 总线上的空闲状态保持 3 ms，模块便会进入挂起状态。出现以下情况时，模块会退出挂起状态：

– USB 数据线上有活动

– 应用程序对 OTG_DCTL 寄存器的远程唤醒信号位（OTG_DCTL 中的 RWUSIG 位）执行写操作。

54.14.42 OTG 设备 IN 端点通用中断屏蔽寄存器 (OTG_DIEPMSK)

OTG device IN endpoint common interrupt mask register

偏移地址：0x810

复位值：0x0000 0000

此寄存器与全体端点的各个 OTG_DIEPINTx 寄存器配合使用，以便在每个 IN 端点上生成中断。通过对此寄存器的相应位执行写操作，可屏蔽 OTG_DIEPINTx 寄存器中特定状态的 IN 端点中断。默认情况下，状态中断都被屏蔽。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	NAKM	Res.	Res.	Res.	BNAM	TXFURM	Res.	INEPNEM	INEPNMM	ITTXFEMSK	TOM	AHBERRM	EPDM	XFRCM
		rw				rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:14 保留，必须保持复位值。

位 13 **NAKM**：NAK 中断屏蔽 (NAK interrupt mask)
0：屏蔽中断
1：使能中断

位 12:10 保留，必须保持复位值。

位 9 **BNAM**：BNA 中断屏蔽 (BNA interrupt mask)
0：屏蔽中断
1：使能中断

位 8 **TXFURM**：发送 FIFO 下溢中断屏蔽 (FIFO underrun mask)
0：屏蔽中断
1：使能中断

位 7 保留，必须保持复位值。

位 6 **INEPNEM**：IN 端点 NAK 有效中断屏蔽 (IN endpoint NAK effective mask)
0：屏蔽中断
1：使能中断

位 5 **INEPNMM**：EP 不匹配时接收到 IN 令牌中断屏蔽 (IN token received with EP mismatch mask)
0：屏蔽中断
1：使能中断

位 4 **ITTXFEMSK**：Tx FIFO 为空时接收到 IN 令牌中断屏蔽 (IN token received when Tx FIFO empty mask)
0：屏蔽中断
1：使能中断

位 3 **TOM**：超时中断屏蔽（非同步端点）(Timeout condition mask (Non-isochronous endpoints))
0：屏蔽中断
1：使能中断

位 2 **AHBERRM**：AHB 错误中断屏蔽 (AHB error mask)
0：屏蔽中断
1：使能中断

位 1 **EPDM**：端点禁止中断屏蔽 (Endpoint disabled interrupt mask)
0：屏蔽中断
1：使能中断

位 0 **XFRCM**：传输完成中断屏蔽 (Transfer completed interrupt mask)
0：屏蔽中断
1：使能中断

54.14.43 OTG 设备 OUT 端点通用中断屏蔽寄存器 (OTG_DOEPMSK)

OTG device OUT endpoint common interrupt mask register

偏移地址：0x814

复位值：0x0000 0000

此寄存器与所有端点的各个 OTG_DOEPINTx 寄存器配合使用，以便可以在每个 OUT 端点上生成中断。通过对此寄存器的相应位执行写操作，可屏蔽 OTG_DOEPINTx 寄存器中特定状态的 OUT 端点中断。默认情况下，状态中断都被屏蔽。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	NYET MSK	NAK MSK	BERR M	Res.	Res.	BNAM	OUT PKT ERRM	Res.	B2B STUPM	STS PHSR XM	OTEPD M	STUPM	AHB ERRM	EPDM	XFRC M
	rw	rw	rw			rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:15 保留，必须保持复位值。

位 14 **NYETMSK**：NYET 中断屏蔽 (NYET interrupt mask)
0：屏蔽中断
1：使能中断

位 13 **NAKMSK**：NAK 中断屏蔽 (NAK interrupt mask)
0：屏蔽中断
1：使能中断

位 12 **BERRM**：串扰错误中断屏蔽 (Babble error interrupt mask)
0：屏蔽中断
1：使能中断

位 11:10 保留，必须保持复位值。

位 9 **BNAM**：BNA 中断屏蔽 (BNA interrupt mask)
0：屏蔽中断
1：使能中断

位 8 **OUTPKTERRM**：输出数据包错误中断屏蔽 (Out packet error mask)
0：屏蔽中断
1：使能中断

位 7 保留，必须保持复位值。

位 6 **B2BSTUPM**：接收到连续的 SETUP 数据包中断屏蔽 (Back-to-back SETUP packets received mask)

仅适用于控制 OUT 端点。

0：屏蔽中断

1：使能中断

位 5 **STSPHSRXM**：接收到用于控制写的状态阶段中断屏蔽 (Status phase received for control write mask)

0：屏蔽中断

1：使能中断

位 4 **OTEPDM**：端点禁止时接收到 OUT 令牌中断屏蔽 (OUT token received when endpoint disabled mask)。仅适用于控制 OUT 端点。

0：屏蔽中断

1：使能中断

位 3 **STUPM**：STUPM：SETUP 阶段完成中断屏蔽 (SETUP phase done mask)。仅适用于控制端点。

0：屏蔽中断

1：使能中断

位 2 **AHBERRM**：AHB 错误中断屏蔽 (AHB error mask)

0：屏蔽中断

1：使能中断

位 1 **EPDM**：端点禁止中断屏蔽 (Endpoint disabled interrupt mask)

0：屏蔽中断

1：使能中断

位 0 **XFRCM**：传输完成中断屏蔽 (Transfer completed interrupt mask)

0：屏蔽中断

1：使能中断

54.14.44 OTG 设备全体端点中断寄存器 (OTG_DAINT)

OTG device all endpoints interrupt register

偏移地址：0x818

复位值：0x0000 0000

当端点上发生有效事件时，OTG_DAINT 寄存器将通过 OTG_GINTSTS 寄存器中的设备 OUT 端点中断位或设备 IN 端点中断位（分别为 OTG_GINTSTS 中的 OEPINT 或 IEPINT 位）来中断应用程序。每个端点对应一个中断位，OUT 端点和 IN 端点均最多有 16 个中断位。双向端点将使用相应的 IN 和 OUT 中断位。当应用程序将相应设备端点 x 中断寄存器 (OTG_DIEPINTx/OTG_DOEPINTx) 中的位置 1 和清零时，此寄存器中的相应位也将置 1 和清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OEPINT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPINT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 **OEPINT[15:0]**：OUT 端点中断位 (OUT endpoint interrupt bits)
每个 OUT 端点对应一位：
OUT 端点 0 对应位 16，而 OUT 端点 3 对应位 19。

位 15:0 **IEPINT[15:0]**：IN 端点中断位 (IN endpoint interrupt bits)
每个 IN 端点对应一位：
IN 端点 0 对应位 0，而 IN 端点 3 对应位 3。

54.14.45 OTG 全体端点中断屏蔽寄存器 (OTG_DAINTMSK)

OTG all endpoints interrupt mask register

偏移地址：0x81C

复位值：0x0000 0000

OTG_DAINTMSK 寄存器与设备端点中断寄存器结合使用，在设备端点上发生事件时中断应用程序。但是，与该中断相对应的 OTG_DAINT 寄存器位仍会置 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OEPM[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPM[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **OEPM[15:0]**：OUT EP 中断屏蔽位 (OUT EP interrupt mask bits)
每个 OUT 端点对应一位：
OUT EP 0 对应位 16，而 OUT EP 3 对应位 19
0：屏蔽中断
1：使能中断

位 15:0 **IEPM[15:0]**：IN EP 中断屏蔽位 (IN EP interrupt mask bits)
每个 IN 端点对应一位：
IN EP 0 对应位 0，而 IN EP 3 对应位 3
0：屏蔽中断
1：使能中断

54.14.46 OTG 设备 V_{BUS} 放电时间寄存器 (OTG_DVBUSDIS)

OTG device V_{BUS} discharge time register

偏移地址：0x0828

复位值：0x0000 17D7

该寄存器指定 SRP 期间 V_{BUS} 发出脉冲之后的 V_{BUS} 放电时间。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VBUSDT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **VBUSDT[15:0]**：设备 V_{BUS} 放电时间 (Device V_{BUS} discharge time)
指定 SRP 期间 V_{BUS} 发出脉冲之后的 V_{BUS} 放电时间。该时间值等于：
V_{BUS} 放电时间（PHY 时钟数）/1024
该值可基于不同的 V_{BUS} 负载根据需要进行调整。

54.14.47 OTG 设备 V_{BUS} 脉冲时间寄存器 (OTG_DVBUSPULSE)

OTG device V_{BUS} pulsing time register

偏移地址：0x082C

复位值：0x0000 05B8

该寄存器指定 SRP 期间的 V_{BUS} 脉冲时间。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DVBUSP[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **DVBUSP[15:0]**：设备 V_{BUS} 脉冲时间 (Device V_{BUS} pulsing time)。该功能仅与 OTG1.3 相关。
指定 SRP 期间的 V_{BUS} 脉冲时间。该时间值等于：
V_{BUS} 脉冲时间（PHY 时钟数）/1024

54.14.48 OTG 设备阈值控制寄存器 (OTG_DTHRCTL)

OTG device threshold control register

偏移地址：0x0830

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	ARPEN	Res.	RXTHRLEN[8:0]									RXTHREN
				rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	TXTHRLEN[8:0]									ISOTHREN	NONISOTHREN
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27 **ARPEN**：仲裁器驻留使能 (Arbiter parking enable)

该位用于控制内部 DMA 仲裁对 IN 端点的驻留。当阈值使能且该位置 1 后，DMA 会对收到令牌的 IN 端点驻留其仲裁。采用这种方式可以避免出现下溢情况。默认情况下使能驻留。

位 26 保留，必须保持复位值。

位 25:17 **RXTHRLEN[8:0]**：接收阈值长度 (Receive threshold length)

该位字段以 32 位字为单位指定接收阈值的大小。该位字段指定模块在 AHB 上启动传输之前在 USB 上接收的数据量。阈值长度最小值为八个 32 位字。推荐 RXTHRLEN 的值和设定的 AHB 批量传输长度（OTG_GAHBCFG 中的 HBSTLEN 位）相同。

位 16 **RXTHREN**：接收阈值使能 (Receive threshold enable)

该位置 1 时，模块会使能接收方向的阈值。

位 15:11 保留，必须保持复位值。

位 10:2 **TXTHRLEN[8:0]**：发送阈值长度 (Transmit threshold length)

该位字段以 32 位字为单位指定发送阈值的大小。该位字段指定模块在 USB 上启动传输之前相应端点发送 FIFO 中的数据量。阈值长度最小值为八个 32 位字。该位字段控制同步和非同步 IN 端点阈值。推荐 TXTHRLEN 的值和设定的 AHB 批量传输长度（OTG_GAHBCFG 中的 HBSTLEN 位）相同。

位 1 **ISOTHREN**：ISO IN 端点阈值使能 (ISO IN endpoint threshold enable)

该位置 1 时，模块会使能同步 IN 端点的阈值。

位 0 **NONISOTHREN**：非同步 IN 端点阈值使能 (Nonisochronous IN endpoints enable)

该位置 1 时，模块会使能非同步 IN 端点的阈值。

54.14.49 OTG 设备 IN 端点 FIFO 空中断屏蔽寄存器 (OTG_DIEPEMPMSK)

OTG device IN endpoint FIFO empty interrupt mask register

偏移地址：0x834

复位值：0x0000 0000

此寄存器用于控制 IN 端点 FIFO 空中断的生成 (TXFE_OTG_DIEPINTx)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INEPTXFEM[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **INEPTXFEM[15:0]**：IN EP Tx FIFO 空中断屏蔽位 (IN EP Tx FIFO empty interrupt mask bits)
这些位用作 OTG_DIEPINTx 的屏蔽位。
每个位对应一个 IN 端点的 TXFE 中断：
IN 端点 0 对应位 0，而 IN 端点 3 对应位 3
0：屏蔽中断
1：使能中断

54.14.50 OTG 设备 IN 端点 x 控制寄存器 (OTG_DIEPCTLx)

OTG device IN endpoint x control register

偏移地址：0x900 + 0x20 * x（x = 0 到 8）

复位值：0x0000 0000

应用程序使用此寄存器控制各个逻辑端点（端点 0 除外）的行为。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	SODD FRM	SD0 PID/ SEVN FRM	SNAK	CNAK	TXFNUM[3:0]				STALL	Res.	EPTYP[1:0]		NAK STS	EO NUM/ DPID
rs	rs	w	w	w	w	rw	rw	rw	rw	rw		rw	rw	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBA EP	Res.	Res.	Res.	Res.	MPSIZ[10:0]										
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **EPENA**：端点使能 (Endpoint enable)
应用程序将该位置 1 以在端点上启动数据发送。
在此端点上触发以下任一中断之前，模块会将该位清零：
– SETUP 阶段完成
– 端点禁止
– 传输完成

位 30 **EPDIS**：端点禁止 (Endpoint disable)

即使在该端点上的传送完成之前，应用程序也可将该位置 1，以停止端点上的数据发送/接收。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断位置 1 前，模块会将该位清零。只有在该端点的端点使能位置 1 后，应用程序才可将该位置 1。

位 29 **SODDFRM**：设置奇数帧 (Set odd frame)

仅适用于同步 IN 和 OUT 端点。

对该位字段进行写操作会将偶数/奇数帧 (EONUM) 位字段设置为奇数帧。

位 28 **SD0PID**：设置 DATA0 PID (Set DATA0 PID)

仅适用于中断/批量 IN 端点。

对该位字段进行写操作会将此寄存器中的端点数据 PID (DPID) 位字段设置为 DATA0。

SEVNFRM：设置偶数帧 (Set even frame)

仅适用于同步 IN 端点。

对该位字段进行写操作会将偶数/奇数帧 (EONUM) 位字段设置为偶数帧。

位 27 **SNAK**：将 NAK 置 1 (Set NAK)

对该位进行写操作会将端点的 NAK 位置 1。

通过该位，应用程序可以控制端点上 NAK 握手信号的发送。发生传输完成中断时或端点上接收到 SETUP 后，模块也可以将 OUT 端点的该位置 1。

位 26 **CNAK**：将 NAK 清零 (Clear NAK)

对该位进行写操作会将端点的 NAK 位清零。

位 25:22 **TXFNUM[3:0]**：Tx FIFO 编号 (Tx FIFO number)

这些位用于指定与此端点相关联的 FIFO 编号。必须为每个有效的 IN 端点设置单独的 FIFO 编号。

该位字段仅针对 IN 端点有效。

位 21 **STALL**：STALL 握手 (STALL handshake)

仅适用于非控制、非同步 IN 端点（访问类型为 rw）。

应用程序将该位置 1 使得设备对来自 USB 主机的所有令牌都回复 STALL。如果 NAK 位、全局 IN NAK 或全局 OUT NAK 与该位同时置 1，则 STALL 位优先。只有应用程序能够将该位清零，而模块则不能。

位 20 保留，必须保持复位值。

位 19:18 **EPTYP[1:0]**：端点类型 (Endpoint type)

以下是这个逻辑端点支持的传输类型。

00：控制

01：同步

10：批量

11：中断

位 17 **NAKSTS**：NAK 状态 (NAK status)

它指示以下结果：

0：模块根据 FIFO 状态回复非 NAK 握手。

1：模块在此端点上回复 NAK 握手。

当应用程序或模块将该位置 1 时：

对于非同步 IN 端点：即使 Tx FIFO 中存在可用数据，模块也会停止通过 IN 端点发送任何数据。

对于同步 IN 端点：即使 Tx FIFO 中存在可用数据，模块也会发送长度为零的数据包。

无论该位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 16 **EONUM**：偶数/奇数帧 (Even/odd frame)

仅适用于同步 IN 端点。

指示模块为此端点发送/接收同步的数据所在的帧的编号。应用程序必须通过此寄存器中的 SEVNFRM 和 SODDFRM 位字段对偶数/奇数帧编号进行编程，以便此端点发送/接收同步数据。

0：偶数帧

1：奇数帧 (Odd frame)

DPID：端点数据 PID (Endpoint data PID)

仅适用于中断/批量 IN 端点。

包含此端点上将要接收或发送的数据包的 PID。端点激活后，应用程序必须对要在此端点上接收或发送的首个数据包的 PID 进行编程。应用程序使用 SD0PID 寄存器位字段对 DATA0 或 DATA1 PID 进行编程。

0：DATA0

1：DATA1

位 15 **USBAEP**：USB 活动端点 (USB active endpoint)

指示此端点在当前配置和接口中是否激活。检测到 USB 复位后，模块会为所有端点（端点 0 除外）将该位清零。接收到 SetConfiguration 和 SetInterface 命令后，应用程序必须相应地对端点寄存器进行编程并将该位置 1。

位 14:11 保留，必须保持复位值。

位 10:0 **MPSIZ[10:0]**：最大数据包大小 (Maximum packet size)

应用程序必须将该位字段编程为当前逻辑端点的最大数据包大小。此值以字节为单位。

54.14.51 OTG 设备 IN 端点 x 中断寄存器 (OTG_DIEPINTx)

OTG device IN endpoint x interrupt register

偏移地址：0x908 + 0x20 * x（x = 0 到 8）

复位值：0x0000 0080

此寄存器指示端点在出现 USB 和 AHB 相关事件时的状态。相关内容在*图 689* 中进行了说明。当模块中断寄存器中的 IN 端点中断位（OTG_GINTSTS 中的 IEPINT 位）置 1 时，应用程序必须读取此寄存器。在应用程序能够读取此寄存器之前，必须先读取设备全体端点中断 (OTG_DAINT) 寄存器，以获取设备端点 x 中断寄存器的准确端点编号。应用程序必须将该寄存器中的相应位清零，才能将 OTG_DAINT 和 OTG_GINTSTS 寄存器中的对应位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	NAK	Res.	PKTD RPSTS	Res.	BNA	TXFIF OUD RN	TXFE	IN EPNE	IN EPNM	ITTXFE	TOC	AHB ERR	EP DISD	XFRC
		rc_w1		rc_w1		rc_w1	rc_w1	r	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:14 保留，必须保持复位值。

位 13 **NAK**：NAK 输入 (NAK input)

当设备发出或收到 NAK 时，模块将生成该中断。如果是同步 IN 端点，由于 Tx FIFO 中无数据可发而发送长度为零的数据包时
也会生成该中断。

位 12 保留，必须保持复位值。

位 11 **PKTDRPSTS**：数据包丢弃状态 (Packet dropped status)

该位用于向应用程序指示有 ISOC OUT 数据包被丢弃。该位没有相应的中断屏蔽位，也不会生成中断。

位 10 保留，必须保持复位值。

位 9 **BNA**：缓冲区不可用中断 (Buffer not available interrupt)

当所访问的描述符没有准备好供模块处理时（例如主机繁忙或 DMA 已完成），模块将生成该中断。
该位仅在使能了分散/收集 DMA 模式时有效。

位 8 **TXFIFOUDRN**：发送 FIFO 下溢 (TxfifoUndrn) (Transmit Fifo Underrun (TxfifoUndrn))

当模块检测到该端点的发送 FIFO 下溢时，将生成该中断。相关性：该中断仅在使能了阈值时有效

位 7 **TXFE**：发送 FIFO 为空 (Transmit FIFO empty)

当此端点的 Tx FIFO 为半空或全空时，此中断触发。Tx FIFO 为半空还是全空状态由 OTG_GAHBCFG 寄存器中的 Tx FIFO 空门限位（OTG_GAHBCFG 中的 TXFELVL 位）决定。

位 6 **INEPNE**：IN 端点 NAK 有效 (IN endpoint NAK effective)

当应用程序通过向 OTG_DIEPCTLx 中的 CNAK 位写入数据来将 IN 端点 NAK 清零时，该位可被清零。
该中断指示模块已对（由应用程序或模块）置 1 的 NAK 采样，结果已生效。该中断指示由应用程序置 1 的 IN 端点 NAK 位已在模块中起作用。
此中断不保证在 USB 上发送了 NAK 握手信号。STALL 位的优先级高于 NAK 位。

位 5 **INEPNM**：EP 不匹配时接收到 IN 令牌 (IN token received with EP mismatch)

指示非周期性 TxFIFO 顶部数据所属的端点不是收到 IN 令牌的端点。此中断会在接收到 IN 令牌的端点上产生。

位 4 **ITTXFE**：Tx FIFO 为空时接收到 IN 令牌 (IN token received when Tx FIFO is empty)

指示当和该端点相对应的 Tx FIFO（周期性/非周期性）为空时，接收到 IN 令牌。此中断会在接收到 IN 令牌的端点上产生。

位 3 **TOC**：超时条件 (Timeout condition)

指示该端点对最近收到的 IN 令牌响应超时。

位 2 **AHBERR**：AHB 错误 (AHB error)

仅当处于内部 DMA 模式下且 AHB 读/写操作期间发生 AHB 错误时才生成此错误。应用程序可通过读取相应端点 DMA 地址寄存器来获取错误地址。

位 1 **EPDISD**：端点禁止中断 (Endpoint disabled interrupt)

该位指示该端点已经由应用程序禁止掉。

位 0 **XFRC**：传输完成中断 (Transfer completed interrupt)

该位字段指示在此端点上设置的传输已经在 USB 和 AHB 上传输完成。

54.14.52 OTG 设备 IN 端点 0 传输大小寄存器 (OTG_DIEPTSIZ0)

OTG device IN endpoint 0 transfer size register

偏移地址：0x910

复位值：0x0000 0000

在使能端点 0 之前，应用程序必须修改此寄存器。通过设备控制端点 0 控制寄存器中的端点使能位（OTG_DIEPCTL0 中的 EPENA）使能端点 0 后，模块对此寄存器进行修改。当模块将端点使能位清零后，应用程序才能读取此寄存器。

非零端点使用端点 1-3 的寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTCNT[1:0]		Res.	Res.	Res.
											rw	rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	XFRSIZ[6:0]						
									rw	rw	rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值。

位 20:19 **PKTCNT[1:0]**：数据包计数 (Packet count)

指示端点 0 的一次数据传输包含的 USB 数据包个数。

每次从 Tx FIFO 读取数据包（最大大小数据包或短数据包）时，该位字段将递减。

位 18:7 保留，必须保持复位值。

位 6:0 **XFRSIZ[6:0]**：传输数据量大小(Transfer size)

指示端点 0 的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。

每次向 Tx FIFO 写入来自外部内存的数据包时，模块会使该位字段递减。

54.14.53 OTG 设备 IN 端点 x DMA 地址寄存器 (OTG_DIEPDMAx)

OTG device IN endpoint x DMA address register

偏移地址：0x914 + 0x20 * x（x = 0 到 8）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAADDR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAADDR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DMAADDR[31:0]**：DMA 地址 (DMA Address)

该位字段保存外部内存中的起始地址，必须从该地址获取端点的数据。每次进行 AHB 传输，该寄存器都会递增。

54.14.54 OTG 设备 IN 端点发送 FIFO 状态寄存器 (OTG_DTXFSTSx)

OTG device IN endpoint transmit FIFO status register

偏移地址：0x918 + 0x20 * x（x = 0 到 8）

复位值：0x0000 0200

此只读寄存器包含设备 IN 端点 Tx FIFO 的空闲空间信息。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INEPTFSAV[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **INEPTFSAV[15:0]**：IN 端点 Tx FIFO 可用空间 (IN endpoint Tx FIFO space available)

指示端点 Tx FIFO 中的可用空闲空间大小。
以 32 位字为单位：
0x0：端点 Tx FIFO 已满
0x1：1 个字可用
0x2：2 个字可用
0xn：n 个字可用
其他值：保留

54.14.55 OTG 设备 IN 端点 x 传输大小寄存器 (OTG_DIEPTSIZx)

OTG device IN endpoint x transfer size register

偏移地址：0x910 + 0x20 * x（x = 1 到 8）

复位值：0x0000 0000

在使能该端点之前，应用程序必须修改此寄存器。通过 OTG_DIEPCTLx 寄存器中的端点使能位（OTG_DIEPCTLx 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。当模块将端点使能位清零后，应用程序才能读取此寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	MCNT[1:0]		PKTCNT[9:0]										XFRSIZ[18:16]		
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XFRSIZ[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:29 **MCNT[1:0]**：多重计数 (Multi count)

对于周期性 IN 端点，该位字段指示在 USB 上每帧必须发送的数据包数。模块使用该位字段计算同步 IN 端点的数据 PID。

01：1 个数据包

10：2 个数据包

11：3 个数据包

位 28:19 **PKTCNT[9:0]**：数据包计数 (Packet count)

指示该端点上的一次数据传输包含的 USB 数据包个数。

每次从 Tx FIFO 读取数据包（最大大小数据包或短数据包）时，该位字段将递减。

位 18:0 **XFRSIZ[18:0]**：传输数据量大小 (Transfer size)

该位字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。

每次向 Tx FIFO 写入来自外部内存的数据包时，模块会使该位字段递减。

54.14.56 OTG 设备控制 OUT 端点 0 控制寄存器 (OTG_DOEPCTL0)

OTG device control OUT endpoint 0 control register

偏移地址：0xB00

复位值：0x0000 8000

本节介绍 OTG_DOEPCTL0 寄存器。非零控制端点使用编号 1-3 的寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	Res.	Res.	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP[1:0]		NAK STS	Res.
w	r			w	w					rs	rw	r	r	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBA EP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPSIZ[1:0]	
r														r	r

位 31 **EPENA**：端点使能 (Endpoint enable)

应用程序将该位置 1 以在端点 0 上启动数据发送。

在此端点上触发以下任一中断之前，模块会将该位清零：

– SETUP 阶段完成

– 端点禁止

– 传输完成

位 30 **EPDIS**：端点禁止 (Endpoint disable)

应用程序无法禁止控制 OUT 端点 0。

位 29:28 保留，必须保持复位值。

位 27 **SNAK**：将 NAK 置 1 (Set NAK)

对该位进行写操作会将端点的 NAK 位置 1。

通过该位，应用程序可以控制端点上 NAK 握手信号的发送。发生传输完成中断时或端点上接收到 SETUP 后，模块也可以将该位置 1。

位 26 **CNAK**：将 NAK 清零 (Clear NAK)
对该位进行写操作会将端点的 NAK 位清零。

位 25:22 保留，必须保持复位值。

位 21 **STALL**：STALL 握手 (STALL handshake)
此端点接收到 SETUP 令牌时，应用程序只能将该位置 1，而模块会将其清零。如果 NAK 位、全局 OUT NAK 与该位同时置 1，则 STALL 位优先。无论该位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 20 **SNPM**：监听模式 (Snoop mode)
该位用于将端点配置为监听模式。在监听模式下，模块不会在将 OUT 数据包传输到应用存储区前检查其是否正确。

位 19:18 **EPTYP[1:0]**：端点类型 (Endpoint type)
硬件固定为二进制 00，表示端点为控制传输类型。

位 17 **NAKSTS**：NAK 状态 (NAK status)
指示以下结果：
0：模块根据 FIFO 状态回复非 NAK 握手。
1：模块在此端点上回复 NAK 握手。
当应用程序或模块将该位置 1 时，即使 Rx FIFO 中存在空间可继续容纳收到的数据包，模块也会停止接收数据。无论该位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 16 保留，必须保持复位值。

位 15 **USBAEP**：USB 活动端点 (USB active endpoint)
该位总是置 1，指示在所有配置和接口中控制端点 0 始终处于激活状态。

位 14:2 保留，必须保持复位值。

位 1:0 **MPSIZ[1:0]**：最大数据包大小 (Maximum packet size)
控制 OUT 端点 0 的最大数据包大小与在控制 IN 端点 0 中进行编程的值相同。
00：64 字节
01：32 字节
10：16 字节
11：8 字节

54.14.57 OTG 设备 OUT 端点 x 中断寄存器 (OTG_DOEPINTx)

OTG device OUT endpoint x interrupt register

偏移地址：0xB08 + 0x20 * x（x = 0 到 8）

复位值：0x0000 0080

此寄存器指示端点在出现 USB 和 AHB 相关事件时的状态。相关内容在*图 689* 中进行了说明。当 OTG_GINTSTS 寄存器中的 OUT 端点中断位（OTG_GINTSTS 中的 OEPINT 位）置 1 时，应用程序必须读取此寄存器。在应用程序能够读取此寄存器之前，必须先读取 OTG_DAINT 寄存器，以获取 OTG_DOEPINTx 寄存器的准确端点编号。应用程序必须将该寄存器中的相应位清零，才能将 OTG_DAINT 和 OTG_GINTSTS 寄存器中的对应位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STPK TRX	NYET	NAK	BERR	Res.	Res.	BNA	OUT PKT ERR	Res.	B2B STUP	STSPH SRX	OTEP DIS	STUP	AHB ERR	EP DISD	XFRC
rc_w1	rc_w1	rc_w1	rc_w1			rc_w1	rc_w1		rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:16 保留，必须保持复位值。

位 15 **STPKTRX**：接收到设置数据包 (Setup packet received)

仅适用于缓冲区 DMA 模式下的控制 OUT 端点。该位由 OTG 设置，用于指示该缓冲区中有 8 个字节的设置数据。每个缓冲区中只有一个设置数据包。接收到设置数据包时，OTG 会在 Rx FIFO 呈现 SETUP_COMPLETE 状态后关闭缓冲区并禁用相应的端点。当 OTG_HS 在特定端点的 SETUP 数据包后发现第一个 IN 或 OUT 令牌时，OTG 会将 SETUP_COMPLETE 状态置于 Rx FIFO 中。应用程序随后必须重新使能端点以接收用于控制传输的 OUT 数据，并重新编程缓冲区起始地址。鉴于上述行为，OTG 可接收任意数量的背靠背设置数据包，每个设置数据包使用一个缓冲区。

位 14 **NYET**：NYET 中断 (NYET interrupt)

当非同步 OUT 端点回复 NYET 握手信号时将产生此中断。

位 13 **NAK**：NAK 输入 (NAK input)

当设备发出或收到 NAK 时，模块将生成该中断。如果是同步 IN 端点，由于 Tx FIFO 中无数据可发而发送长度为零的数据包时也会生成该中断。

位 12 **BERR**：串扰错误中断 (Babble error interrupt)

当端点收到串扰错误时，模块会生成此中断。

位 11:10 保留，必须保持复位值。

位 9 **BNA**：缓冲区不可用中断 (Buffer not available interrupt)

当所访问的描述符没有准备好供模块处理时（例如主机繁忙或 DMA 已完成），模块将生成该中断。

该位仅在使能了分散/收集 DMA 模式时有效。

位 8 **OUTPKTERR**：OUT 数据包错误 (OUT packet error)

当模块检测到 OUT 数据包发生上溢或 CRC 错误时会产生该中断。该中断仅在阈值已使能时有效。

位 7 保留，必须保持复位值。

位 6 **B2BSTUP**：接收到连续的 SETUP 数据包 (Back-to-back SETUP packets received)

仅适用于控制 OUT 端点。

该位指示该端点已接收到三个以上的连续 SETUP 数据包。

位 5 **STSPHSRX**：接收到控制写的状态阶段 (Status phase received for control write)

该中断仅对控制 OUT 端点有效。只有当 OTG 将主机在控制写传输的数据阶段发送的所有数据全部传输到系统内存缓冲区后，才会生成该中断。该中断向应用程序指示主机已从控制写传输的数据阶段切换到状态阶段。完成数据阶段的解码后，应用程序可使用该中断向状态阶段作出 ACK 或 STALL 响应。

位 4 **OTEPDIS**：端点禁止时接收到 OUT 令牌 (OUT token received when endpoint disabled)

仅适用于控制 OUT 端点。

指示在尚未使能端点时接收到 OUT 令牌，从而产生中断。

位 3 **STUP**：SETUP 阶段完成

仅适用于控制 OUT 端点。指示控制端点的 SETUP 阶段已完成，当前控制传输中不再接收到连续的 SETUP 数据包。在此中断上，应用程序可以对接收到的 SETUP 数据包进行解码。

位 2 **AHBERR**：AHB 错误 (AHB error)

仅当处于内部 DMA 模式下且 AHB 读/写操作期间发生 AHB 错误时才生成此错误。应用程序可通过读取相应端点 DMA 地址寄存器来获取错误地址。

位 1 **EPDISD**：端点禁止中断 (Endpoint disabled interrupt)

该位指示该端点已经由应用程序禁止掉。

位 0 **XFRC**：传输完成中断 (Transfer completed interrupt)

该位字段指示在此端点上设置的传输已经在 USB 和 AHB 上传输完成。

54.14.58 OTG 设备 OUT 端点 0 传输大小寄存器 (OTG_DOEPTSIZ0)

OTG device OUT endpoint 0 transfer size register

偏移地址：0xB10

复位值：0x0000 0000

在使能端点 0 之前，应用程序必须修改此寄存器。通过 OTG_DOEPCTL0 寄存器中的端点使能位（OTG_DOEPCTL0 中的 EPENA 位）使能端点 0 后，模块对此寄存器进行修改。当模块将端点使能位清零后，应用程序才能读取此寄存器。

非零端点使用端点 1–8 的寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	STUPCNT[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTCNT	Res.	Res.	Res.
	rw	rw										rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	XFRSIZ[6:0]						
									rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:29 **STUPCNT[1:0]**：SETUP 数据包计数 (SETUP packet count)

该位字段指定端点能连续接收的 SETUP 数据包数量。

01：1 个数据包

10：2 个数据包

11：3 个数据包

位 28:20 保留，必须保持复位值。

位 19 **PKTCNT**：数据包计数 (Packet count)

每向 Rx FIFO 写入一个数据包，该位字段递减。

位 18:7 保留，必须保持复位值。

位 6:0 **XFRSIZ[6:0]**：传输数据量大小 (Transfer size)

指示端点 0 的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。

每次从 Rx FIFO 读取数据包并将其写入外部内存时，模块会使该位字段递减。

54.14.59 OTG 设备 OUT 端点 x DMA 地址寄存器 (OTG_DOEPDMAx)

OTG device OUT endpoint x DMA address register

偏移地址：0xB14 + 0x20 * x（x = 0 到 8）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAADDR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAADDR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DMAADDR[31:0]**：DMA 地址 (DMA Address)

该位字段保存外部内存中的起始地址，必须从该地址获取端点的数据。每次进行 AHB 传输，该寄存器都会递增。

54.14.60 OTG 设备 OUT 端点 x 控制寄存器 (OTG_DOEPCTLx)

OTG device OUT endpoint x control register

偏移地址：0xB00 + 0x20 * x（x = 1 到 8）

复位值：0x0000 0000

应用程序使用此寄存器控制各个逻辑端点（端点 0 除外）的行为。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	SD1 PID/ SODD FRM	SD0 PID/ SEVN FRM	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP[1:0]		NAK STS	EO NUM/ DPID
rs	rs	w	w	w	w					rw	rw	rw	rw	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBA EP	Res.	Res.	Res.	Res.	MPSIZ[10:0]										
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **EPENA**：端点使能 (Endpoint enable)

适用于 IN 和 OUT 端点。

应用程序将该位置 1 以在端点上启动数据发送。

在此端点上触发以下任一中断之前，模块会将该位清零：

– SETUP 阶段完成

– 端点禁止

– 传输完成

位 30 **EPDIS**：端点禁止 (Endpoint disable)

即使在该端点上的传送完成之前，应用程序也可将该位置 1，以停止端点上的数据发送/接收。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断位置 1 前，模块会将该位清零。只有在该端点的端点使能位置 1 后，应用程序才可将该位置 1。

位 29 **SD1PID**：设置 DATA1 PID (Set DATA1 PID)

仅适用于中断/批量 IN 和 OUT 端点。对该位字段进行写操作会将此寄存器中的端点数据 PID (DPID) 位字段设置为 DATA1。

SODDFRM：设置奇数帧 (Set odd frame)

仅适用于同步 IN 和 OUT 端点。对该位字段进行写操作会将偶数/奇数帧 (EONUM) 位字段设置为奇数帧。

位 28 **SD0PID**：设置 DATA0 PID (Set DATA0 PID)

仅适用于中断/批量 OUT 端点。

对该位字段进行写操作会将此寄存器中的端点数据 PID (DPID) 位字段设置为 DATA0。

SEVNFRM：设置偶数帧 (Set even frame)

仅适用于同步 OUT 端点。

对该位字段进行写操作会将偶数/奇数帧 (EONUM) 位字段设置为偶数帧。

位 27 **SNAK**：将 NAK 置 1 (Set NAK)

对该位进行写操作会将端点的 NAK 位置 1。

通过该位，应用程序可以控制端点上 NAK 握手信号的发送。发生传输完成中断时或端点上接收到 SETUP 后，模块也可以将 OUT 端点的该位置 1。

位 26 **CNAK**：将 NAK 清零 (Clear NAK)

对该位进行写操作会将端点的 NAK 位清零。

位 25:22 保留，必须保持复位值。

位 21 **STALL**：STALL 握手 (STALL handshake)

仅适用于非控制、非同步 OUT 端点（访问类型为 rw）。

应用程序将该位置 1 使得设备对来自 USB 主机的所有令牌都回复 STALL。如果 NAK 位、全局 IN NAK 或全局 OUT NAK 与该位同时置 1，则 STALL 位优先。只有应用程序能够将该位清零，而模块则不能。

仅适用于控制端点（访问类型为 rs）

此端点接收到 SETUP 令牌时，应用程序只能将该位置 1，而模块会将其清零。如果 NAK 位、全局 IN NAK 或全局 OUT NAK 与该位同时置 1，则 STALL 位优先。无论该位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 20 **SNPM**：监听模式 (Snoop mode)

该位用于将端点配置为监听模式。在监听模式下，模块不会在将 OUT 数据包传输到应用存储区前检查其是否正确。

位 19:18 **EPTYP[1:0]**：端点类型 (Endpoint type)

以下是这个逻辑端点支持的传输类型。

00：控制

01：同步

10：批量

11：中断

位 17 **NAKSTS**：NAK 状态 (NAK status)

指示以下结果：

0：模块根据 FIFO 状态回复非 NAK 握手。

1：模块在此端点上回复 NAK 握手。

当应用程序或模块将该位置 1 时：

即使 Rx FIFO 存在空间可容纳传入数据包，模块也会停止在 OUT 端点上接收任何数据。

无论该位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 16 **EONUM**：偶数/奇数帧 (Even/odd frame)

仅适用于同步 IN 和 OUT 端点。

指示模块为此端点发送/接收同步的数据所在的帧的编号。应用程序必须通过此寄存器中的 SEVNFRM 和 SODDFRM 位字段对偶数/奇数帧编号进行编程，以便此端点发送/接收同步数据。

0：偶数帧

1：奇数帧

DPID：端点数据 PID (Endpoint data PID)

仅适用于中断/批量 OUT 端点。

包含此端点上将要接收或发送的数据包的 PID。端点激活后，应用程序必须对要在此端点上接收或发送的首个数据包的 PID 进行编程。应用程序使用 SD0PID 寄存器位字段对 DATA0 或 DATA1 PID 进行编程。

0：DATA0

1：DATA1

位 15 **USBAEP**：USB 活动端点 (USB active endpoint)

指示此端点在当前配置和接口中是否激活。检测到 USB 复位后，模块会为所有端点（端点 0 除外）将该位清零。接收到 SetConfiguration 和 SetInterface 命令后，应用程序必须相应地对端点寄存器进行编程并将该位置 1。

位 14:11 保留，必须保持复位值。

位 10:0 **MPSIZ[10:0]**：最大数据包大小 (Maximum packet size)

应用程序必须将该位字段编程为当前逻辑端点的最大数据包大小。此值以字节为单位。

54.14.61 OTG 设备 OUT 端点 x 传输大小寄存器 (OTG_DOEPTSIZx)

OTG device OUT endpoint x transfer size register

偏移地址：0xB10 + 0x20 * x（x = 1 到 8）

复位值：0x0000 0000

在使能该端点之前，应用程序必须修改此寄存器。通过 OTG_DOEPCTLx 寄存器中的端点使能位（OTG_DOEPCTLx 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。当模块将端点使能位清零后，应用程序才能读取此寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	RXDPID/ STUPCNT[1:0]		PKTCNT[9:0]										XFRSIZ[18:16]		
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XFRSIZ[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:29 **RXDPID[1:0]**：接收到的数据 PID (Received data PID)

仅适用于同步 OUT 端点。

这是此端点收到的上一个数据包的 PID。

00：DATA0

01：DATA2

10：DATA1

11：MDATA

STUPCNT[1:0]：SETUP 数据包计数 (SETUP packet count)

仅适用于控制 OUT 端点。

该位字段指定端点能连续接收的 SETUP 数据包数量。

01：1 个数据包

10：2 个数据包

11：3 个数据包

位 28:19 **PKTCNT[9:0]**：数据包计数 (Packet count)

指示该端点上的一次数据传输包含的 USB 数据包个数。

每次向 Rx FIFO 写入数据包（最大大小数据包或短数据包）后，该位字段将递减。

位 18:0 **XFRSIZ[18:0]**：传输数据量大小 (Transfer size)

该位字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。

每次从 Rx FIFO 读取数据包并将其写入外部内存时，模块会使该位字段递减。

54.14.62 OTG 电源和时钟门控控制寄存器 (OTG_PCGCCTL)

OTG power and clock gating control register

偏移地址：0xE00

复位值：0x200B 8000

此寄存器在主机模式和从机模式下均可用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUSP	PHY SLEEP	ENL1 GTG	PHY SUSP	Res.	Res.	GATE HCLK	STPP CLK
								r	r	rw	r			rw	rw

位 31:8 保留，必须保持复位值。

位 7 **SUSP**：深度睡眠 (Deep Sleep)

在 L1 状态下时，该位表示 PHY 处于深度睡眠状态。

位 6 **PHYSLEEP**：PHY 处于睡眠状态 (PHY in Sleep)

该位指示 PHY 处于睡眠状态。

位 5 **ENL1GTG**：使能睡眠时钟门控 (Enable sleep clock gating)

该位置 1 时，如果模块无法触发 utmi_l1_suspend_n，则会在睡眠状态下使能模块内部时钟门控。当该位不置 1 时，不会在睡眠状态下使能 PHY 时钟门控。

位 4 **PHYSUSP**：PHY 挂起 (PHY suspended)

指示 PHY 已挂起。应用程序将 STPPCLK 位置 1 后，一旦 PHY 挂起，该位就会更新。

位 3:2 保留，必须保持复位值。

位 1 **GATEHCLK**：门控 HCLK (Gate HCLK)

当 USB 通信挂起或会话无效时，应用程序会将该位置 1，以停止对除 AHB 总线从接口、主接口和唤醒逻辑之外的模块提供时钟。当 USB 恢复通信或新会话启动时，应用程序将该位清零。

位 0 **STPPCLK**：停止 PHY 时钟 (Stop PHY clock)

当 USB 通信挂起、会话无效或设备断开连接时，应用程序将该位置 1 以停止 PHY 时钟。当 USB 恢复通信或新会话启动时，应用程序将该位清零。

54.14.63 OTG 寄存器映射

下表提供了 USB OTG 寄存器映射和复位值。

表 517. OTG 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	OTG_ GOTGCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CURMOD	OTGVER	BSVLD	ASVLD	DBCT	CIDSTS	Res.	Res.	Res.	EHEN	DHNPEN	HSHNPEN	HNPRQ	HNGSCS	BVALOVAL	BVALOEN	AVALOVAL	AVALOEN	VBVALOVAL	VBVALOEN	SRQ	SRQSCS
	Reset value											0	0	0	0	0	1				0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	OTG_ GOTGINT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBCDNE	ADTOCHG	HNGDET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HNSSCHG	SRSSCHG	Res.	Res.	Res.	Res.	Res.	SEDET	Res.	Res.
	Reset value													0	0	0							-	0	0						0		
0x008	OTG_ GAHBCFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PTXFELVL	TXFELVL	Res.	DMAEN	HBSTLEN				GINTMSK
	Reset value																								0	0		0	0	0	0	0	0
0x00C	OTG_ GUSBCFG	Res.	FDMOD	FHMOD	Res.	Res.	Res.	Res.	Res.	Res.	TSDPS	Res.	Res.	Res.	Res.	Res.	Res.	PHYLPC	Res.	TRDT				HNPCAP	SRPCAP	Res.	Res.	Res.	Res.	Res.	TOCAL		
	Reset value		0	0														0		0	1	0	1	0	0						0	0	0
0x010	OTG_ GRSTCTL	AHBIDL	DMAREQ	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFNUM					TXFFLSH	RXFFLSH	Res.	FCRST	PSRST	CSRST
	Reset value	1	0																				0	0	0	0	0	0	0		0	0	0
0x014	OTG_ GINTSTS	WKUPINT	SRQINT	DISCINT	CIDSCHG	LPMINT	PTXFE	HCINT	HPRTINT	RSTDET	DATAFSUSP	IPXFR/INCOMPISOOUT	IISOIXFR	OEPINT	IEPINT	Res.	Res.	EOPF	ISOODRP	ENUMDNE	USBRST	USBSUSP	ESUSP	Res.	Res.	GONAKEFF	GINAKEFF	NPTXFE	RXFLVL	SOF	OTGINT	MMIS	CMOD
	Reset value	0	0	0	0	0	1	0	0	0	0	0	0	0	0			0	0	0	0	0	0			0	0	1	0	0	0	0	0

表 517. OTG 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x018	**OTG_ GINTMSK**	WUIM	SRQIM	DISCINT	CIDSCHGM	LPMINTM	PTXFEM	HCIM	PRTIM	RSTDETM	FSUSPM	IPXFRM/IISOOXFRM	IISOIXFRM	OEPINT	IEPINT	Res.	Res.	EOPFM	ISOODRPM	ENUMDNEM	USBRST	USBSUSPM	ESUSPM	Res.	Res.	GONAKEFFM	GINAKEFFM	NPTXFEM	RXFLVLM	SOFM	OTGINT	MMISM	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0			0	0	0	0	0	0	0	
0x01C	**OTG_ GRXSTSR (Device mode)**	Res.	Res.	Res.	Res.	STSPHST	Res.	Res.	FRMNUM				PKTSTS				DPID		BCNT											EPNUM			
	Reset value					0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	OTG_ GRXSTSR (host mode)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTSTS				DPID		BCNT												CHNUM		
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	**OTG_ GRXSTSP (Device mode)**	Res.	Res.	Res.	Res.	STSPHST	Res.	Res.	FRMNUM				PKTSTS				DPID		BCNT											EPNUM			
	Reset value					0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	OTG_ GRXSTSP (host mode)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTSTS				DPID		BCNT												CHNUM		
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	**OTG_ GRXFSIZ**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXFD															
	Reset value																	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x028	**OTG_ HNPTXFSIZ/ OTG_ DIEPTXF0**	NPTXFD/TX0FD																NPTXFSA/TX0FSA															
	Reset value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x02C	**OTG_ HNPTXSTS**	Res.	NPTXQTOP							NPTQXSAV								NPTXFSAV															
	Reset value		0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x038	**OTG_ GCCFG**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDEN	VBDEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value										0	0					0																
0x03C	**OTG_CID**	PRODUCT_ID																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x054	**OTG_ GLPMCFG**	Res.	Res.	Res.	ENBESL	LPMR CNTSTS			SNDLPM	LPM RCNT			LPMCHIDX				L1RSMOK	SLPSTS	LPM RSP		L1DSEN	BESLTHRS			L1SSEN	REMWAKE	BESL				LPMACK	LPMEN	
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x100	**OTG_ HPTXFSIZ**	PTXFSIZ																PTXSA															
	Reset value	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

表 517. OTG 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x104	**OTG_ DIEPTXF1**	INEPTXFD																INEPTXSA															
	Reset value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0x108	**OTG_ DIEPTXF2**	INEPTXFD																INEPTXSA															
	Reset value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
. . . .	. . . .	. . . .																															
. . . .	. . . .	. . . .																															
0x120	**OTG_ DIEPTXF7**	INEPTXFD																INEPTXSA															
	Reset value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
0x400	**OTG_ HCFG**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSLSS	FSLS PCS	
	Reset value																														0	0	0
0x404	**OTG_ HFIR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RLDCTRL	FRIVL															
	Reset value																0	1	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0
0x408	**OTG_ HFNUM**	FTREM																FRNUM															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x410	**OTG_ HPTXSTS**	PTXQTOP								PTXQSAV								PTXFSAVL															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0x414	**OTG_ HAINT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HAINT															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x418	**OTG_ HAINTMSK**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HAINTM															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x41C	**OTG_ HFLBADDR**	HFLBADDR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x440	**OTG_ HPRT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSPD	PTCTL					PPWR	PLSTS		Res.	PRST	PSUSP	PRES	POCCHNG	POCA	PENCHNG	PENA	PCDET	PCSTS
	Reset value														0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0
0x500	**OTG_ HCCHAR0**	CHENA	CHDIS	ODDFRM	DAD							MCNT		EPTYP	LSDEV	Res.	EPDIR	EPNUM						MPSIZ									
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 517. OTG 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x504	**OTG_ HCSPLT0**	SPLITEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMPLSPLT	XACTPOS		HUBADDR							PRTADDR						
	Reset value	0															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x508	**OTG_ HCINT0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERR	FRMOR	BBERR	TXERR	Res.	ACK	NAK	STALL	Res.	CHH	XFRC
	Reset value																						0	0	0	0		0	0	0		0	0
0x508	**OTG_ HCINT0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERR	FRMOR	BBERR	TXERR	NYET	ACK	NAK	STALL	AHBERR	CHH	XFRC
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x50C	**OTG_ HCINTMSK0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	Res.	CHHM	XFRCM
	Reset value																						0	0	0	0	0	0	0	0		0	0
0x510	**OTG_ HCTSIZ0**	DOPNG	DPID		PKTCNT										XFRSIZ																		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x510	**OTG_ HCTSIZSG0**	DOPNG	PID		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NTD								SCHED_INFO							
	Reset value	0	0	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x514	**OTG_ HCDMA0**	DMAADDR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x514	**OTG_ HCDMASG0**	DMASG[31:3]																													Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x51C	**OTG_ HCDMAB0**	HCDMAB																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x520	**OTG_ HCCHAR1**	CHENA	CHDIS	ODDFRM	DAD							MCNT		EPTYP	LSDEV	Res.	EPDIR	EPNUM					MPSIZ										
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x524	**OTG_ HCSPLT1**	SPLITEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMPLSPLT	XACTPOS		HUBADDR							PRTADDR						
	Reset value	0															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 517. OTG 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x528	**OTG_ HCINT1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERR	FRMOR	BBERR	TXERR	Res.	ACK	NAK	STALL	Res.	CHH	XFRC
	Reset value																						0	0	0	0		0	0	0		0	0
0x528	**OTG_ HCINT1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERR	FRMOR	BBERR	TXERR	NYET	ACK	NAK	STALL	AHBERR	CHH	XFRC
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x52C	**OTG_ HCINTMSK1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	Res.	CHHM	XFRCM
	Reset value																						0	0	0	0	0	0	0	0		0	0
0x530	**OTG_ HCTSIZ1**	DOPNG	DPID		PKTCNT										XFRSIZ																		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x530	**OTG_ HCTSIZSG1**	DOPNG	PID		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NTD								SCHED_INFO							
	Reset value	0	0	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x534	**OTG_ HCDMA1**	DMAADDR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x534	**OTG_ HCDMASG1**	DMASG[31:3]																													Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-
0x53C	**OTG_ HCDMAB1**	HCDMAB																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
. . . .	. . . .	. . . .																															
0x660	**OTG_ HCCHAR11**	CHENA	CHDIS	ODDFRM	DAD							MCNT		EPTYP	LSDEV	Res.	EPDIR	EPNUM					MPSIZ										
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x664	**OTG_ HCSPLT11**	SPLITEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMPLSPLT	XACTPOS		HUBADDR							PRTADDR						
	Reset value	0															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x668	**OTG_ HCINT11**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERR	FRMOR	BBERR	TXERR	Res.	ACK	NAK	STALL	Res.	CHH	XFRC
	Reset value																						0	0	0	0		0	0	0		0	0

表 517. OTG 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x66C	**OTG_ HCINTMSK11**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	Res.	CHHM	XFRCM
	Reset value																						0	0	0	0	0	0	0	0		0	0
0x670	**OTG_ HCTSIZ11**	DOPNG	DPID		PKTCNT										XFRSIZ																		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x670	**OTG_ HCTSIZSG11**	DOPNG	PID		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NTD								SCHED_INFO							
	Reset value	0	0	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x674	**OTG_ HCDMA11**	DMAADDR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x674	**OTG_ HCDMASG11**	DMASG[31:3]																													Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-
0x67C	**OTG_ HCDMAB11**	HCDMAB																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
. . . .	. . . .	. . . .																															
0x6E0	**OTG_ HCCHAR15**	CHENA	CHDIS	ODDFRM	DAD							MCNT		EPTYP	LSDEV	Res.	EPDIR	EPNUM					MPSIZ										
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x6E4	**OTG_ HCSPLT15**	SPLITEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMPLSPLT	XACTPOS		HUBADDR							PRTADDR						
	Reset value	0															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x6E8	**OTG_ HCINT15**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERR	FRMOR	BBERR	TXERR	Res.	ACK	NAK	STALL	Res.	CHH	XFRC
	Reset value																						0	0	0	0		0	0	0		0	0
0x6EC	**OTG_ HCINTMSK15**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	Res.	CHHM	XFRCM
	Reset value																						0	0	0	0	0	0	0	0		0	0
0x6F0	**OTG_ HCTSIZ15**	DOPNG	DPID		PKTCNT										XFRSIZ																		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 517. OTG 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x6F0	**OTG_ HCTSIZSG15**	DOPNG	PID		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NTD								SCHED_INFO							
	Reset value	0	0	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x6F4	**OTG_ HCDMA15**	DMAADDR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x6F4	**OTG_ HCDMASG15**	DMASG[31:3]																													Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-
0x6FC	**OTG_ HCDMAB15**	HCDMAB																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x800	**OTG_ DCFG**	Res.	Res.	Res.	Res.	Res.	Res.	PERS CHI VL		DESCDMA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERRATIM	XCVRDLY	Res.	PFIVL		DAD							Res.	NZLSOHSK	DSPD	
	Reset value							0	0	0								0	0		0	0	0	0	0	0	0	0	0		0	0	0
0x804	**OTG_ DCTL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSBESLRJCT	ENCONTONBNA	Res.	Res.	Res.	Res.	Res.	POPRGDNE	CGONAK	SGONAK	CGINAK	SGINAK	TCTL				GONSTS	GINSTS	SDIS	RWUSIG
	Reset value														0	0						0	0	0	0	0	0	0	0	0	0	1	0
0x808	**OTG_ DSTS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEV LN STS		FNSOF														Res.	Res.	Res.	Res.	EERR	ENUMSPD		SUSPSTS
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0
0x810	**OTG_ DIEPMSK**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		NAKM	Res.	Res.	Res.	BNAM	TXFURM	Res.	INEPNEM	INEPNMM	ITTXFEMSK	TOM	AHBERRM	EPDM	XFRCM
	Reset value																			0				0	0		0	0	0	0	0	0	0
0x814	**OTG_ DOEPMSK**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NYETMSK	NAKMSK	BERRM	Res.	Res.	BNAM	OUTPKTERRM	Res.	B2BSTUPM	Res.	OTEPDM	STUPM	AHBERRM	EPDM	XFRCM
	Reset value																		0	0	0			0	0		0		0	0	0	0	0
0x818	**OTG_ DAINT**	OEPINT																IEPINT															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x81C	**OTG_ DAINTMSK**	OEPM																IEPM															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 517. OTG 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x828	**OTG_ DVBUSDIS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBUSDT															
	Reset value																	0	0	0	1	0	1	1	1	1	1	0	1	0	1	1	1
0x82C	**OTG_DVB USPULSE**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DVBUSP															
	Reset value																	0	0	0	0	0	1	0	1	1	0	1	1	1	0	0	0
0x830	**OTG_ DTHRCTL**	Res.	Res.	Res.	Res.	ARPEN	Res.	RXTHRLEN									RXTHREN	Res.	Res.	Res.	Res.	TXTHRLEN									ISOTHREN	NONISOTHREN
	Reset value					0											0					0	0	0	0	0	0	0	0	0	0	0	0
0x834	**OTG_DIE PEMPMSK**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INEPTXFEM															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x900	**OTG_ DIEPCTL0**	EPENA	EPDIS	SODDFRM/SD1PID	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
	Reset value	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0
0x908	**OTG_ DIEPINT0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NAK	Res.	PKTDRPSTS	Res.	BNA	TXFIFOUDRN	TXFE	INEPNE	INEPNM	ITTXFE	TOC	AHBERR	EPDISD	XFRC
	Reset value																			0		0		0	0	1	0	0	0	0	0	0	0
0x910	**OTG_ DIEPTSIZ0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKT CNT		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	XFRSIZ						
	Reset value												0	0													0	0	0	0	0	0	0
0x914	**OTG_ DIEPDMA**	DMAADDR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x918	**OTG_ DTXFSTS0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INEPTFSAV															
	Reset value																	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x920	**OTG_ DIEPCTL1**	EPENA	EPDIS	SODDFRM/SD1PID	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
	Reset value	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0

表 517. OTG 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x928	**OTG_ DIEPINT1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NAK	Res.	PKTDRPSTS	Res.	BNA	TXFIFOUDRN	TXFE	INEPNE	INEPNM	ITTXFE	TOC	AHBERR	EPDISD	XFRC
	Reset value																			0		0		0	0	1	0	0	0	0	0	0	0
0x930	**OTG_ DIEPTSIZ1**	Res.	MCNT		PKTCNT										XFRSIZ																		
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x938	**OTG_ DTXFSTS1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INEPTFSAV															
	Reset value																	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x940	**OTG_ DIEPCTL2**	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
	Reset value	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0
. . . .	. . . .	.																															
. . . .	. . . .	.																															
. . . .	. . . .	.																															
. . . .	. . . .	.																															
0x9E0	**OTG_ DIEPCTL7**	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
	Reset value	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0
. . . .	. . . .	.																															
0x9E8	**OTG_ DIEPINT7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NAK	Res.	PKTDRPSTS	Res.	BNA	TXFIFOUDRN	TXFE	INEPNE	INEPNM	ITTXFE	TOC	AHBERR	EPDISD	XFRC
	Reset value																			0		0		0	0	1	0	0	0	0	0	0	0

表 517. OTG 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
. . . .	. . . .	. . . .																															
0x9F0	**OTG_ DIEPTSIZ7**	Res.	MCNT		PKTCNT										XFRSIZ																		
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
. . . .	. . . .	. . . .																															
0x9F8	**OTG_ DTXFSTS7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INEPTFSAV															
	Reset value																	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0xB00	**OTG_ DOEPCTL0**	EPENA	EPDIS	Res.	Res.	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP		NAKSTS	Res.	USBAEP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPSIZ	
	Reset value	0	0			0	0					0	0	0	0	0		1														0	0
0xB08	**OTG_ DOEPINT0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STPKTRX	NYET	NAK	BERR	Res.	Res.	BNA	OUTPKTERR	Res.	B2BSTUP	STSPHSRX	OTEPDIS	STUP	AHBERR	EPDISD	XFRC
	Reset value																	0	0	0	0			0	0		0	0	0	0	0	0	0
0xB10	**OTG_ DOEPTSIZ0**	Res.	STUPCNT		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTCNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	XFRSIZ						
	Reset value		0	0										0													0	0	0	0	0	0	0
0xB14	**OTG_ DOEPDMA0**	DMAADDR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xB20	**OTG_ DOEPCTL1**	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EP TYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
	Reset value	0	0	0	0	0	0					0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0
0xB28	**OTG_ DOEPINT1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STPKTRX	NYET	NAK	BERR	Res.	Res.	BNA	OUTPKTERR	Res.	B2BSTUP	STSPHSRX	OTEPDIS	STUP	AHBERR	EPDISD	XFRC
	Reset value																	0	0	0	0			0	0		0	0	0	0	0	0	0
0xB30	**OTG_ DOEPTSIZ1**	Res.	RXDPID/STUPCNT		PKTCNT										XFRSIZ																		
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 517. OTG 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xB34	OTG_ DOEPDMA1	DMAADDR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
. . . .	. . . .	. . . .																															
0xC00	OTG_ DOEPCTL8	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
	Reset value	0	0	0	0	0	0					0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0
0xC08	OTG_ DOEPINT8	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STPKTRX	NYET	NAK	BERR	Res.	Res.	BNA	OUTPKTERR	Res.	B2BSTUP	STSPHSRX	OTEPDIS	STUP	AHBERR	EPDISD	XFRC
	Reset value																	0	0	0	0		-	0	0		0	0	0	0	0	0	0
0xC10	OTG_ DOEPTSIZ8	Res.	RXDPID/ STUPCNT		PKTCNT										XFRSIZ																		
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xC14	OTG_ DOEPDMA8	DMAADDR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xE00	OTG_ PCGCCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUSP	PHYSLEEP	ENL1GTG	PHYSUSP	Res.	Res.	GATEHCLK	STPPCLK
	Reset value																									0	0	0	0			0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

54.15 OTG 编程模型

54.15.1 模块初始化

应用程序必须执行模块初始化序列。如果上电期间连接电缆，则 OTG_GINTSTS 中的当前工作模式位（OTG_GINTSTS 中的 CMOD 位）将指示模式。OTG 控制器在连接“A 型”插头后进入主机模式；连接“B 型”插头后进入从机模式。

本节介绍了 OTG 控制器在上电后的初始化过程。无论是以主机模式还是从机模式工作，应用程序都必须遵循初始化序列。根据模块配置对所有模块全局寄存器进行初始化：

1. 在 OTG_GAHBCFG 寄存器中编程以下位字段：
   - 全局中断屏蔽位 GINTMSK = 1
   - Rx FIFO 非空（OTG_GINTSTS 中的 RXFLVL 位）
   - 周期性 Tx FIFO 空门限
2. 在 OTG_GUSBCFG 寄存器中编程以下位字段：
   - HNP 功能位
   - SRP 功能位
   - OTG 超时校准位字段
   - USB 周转时间位字段
3. 软件必须使能 OTG_GINTMSK 寄存器中的以下位：

   OTG 中断屏蔽

   模式不匹配中断屏蔽
4. 通过读取 OTG_GINTSTS 中的 CMOD 位，软件可确定 OTG 控制器是在主机模式还是从机模式下工作。

54.15.2 主机初始化

要将模块作为主机进行初始化，应用程序必须执行以下步骤：

1. 编程 OTG_GINTMSK 寄存器中的 HPRTINT 以打开中断。
2. 编程 OTG_HCFG 寄存器以选择全速主机。
3. 将 OTG_HPRT 中的 PPWR 位编程为 1，给 USB 总线提供 V_{BUS}。
4. 等待 OTG_HPRT0 中的 PCDET 中断。这表示某设备已连接到主机端口。
5. 将 OTG_HPRT 中的 PRST 位编程为 1，在 USB 总线上发出复位信号。
6. 至少等待 10 ms，以便完成复位过程。
7. 将 OTG_HPRT 中的 PRST 位编程为 0，
8. 等待 OTG_HPRT 中的 PENCHNG 中断。
9. 读取 OTG_HPRT 中的 PSPD 位以获取枚举速度。
10. 使用所选 PHY 时钟 1，相应地设置 HFIR 寄存器。
11. 根据步骤 9 中检测到的设备速度编程 OTG_HCFG 寄存器中的 FSLSPCS 位字段。如果 FSLSPCS 发生更改，则必须执行端口复位。
12. 编程 OTG_GRXFSIZ 寄存器以选择接收 FIFO 的大小。

13. 编程 OTG_HNPTXFSIZ 寄存器，以选择用于非周期性通信事务的非周期性发送 FIFO 的大小和起始地址。
14. 编程 OTG_HPTXFSIZ 寄存器，以选择用于周期性通信事务的周期性发送 FIFO 的大小和起始地址。

要与设备通信，系统软件必须初始化并使能至少一个通道。

54.15.3 设备初始化

上电期间或者从主机模式切换为从机模式后，应用程序必须执行下列步骤来将模块作为设备进行初始化。

1. 在 OTG_DCFG 寄存器中编程以下位字段：
   - DESCDMA
   - 设备速度
   - 非零长度状态 OUT 握手信号
   - 周期性帧间隔
2. 编程设备阈值控制寄存器。仅当使用 DMA 模式并计划使能阈值时，才需要执行该操作。
3. 将 DCTL.SDIS 位清零。该位清零后，模块会发起连接。
4. 编程 OTG_GINTMSK 寄存器以使能以下中断：
   - USB 复位
   - 枚举完成
   - 早期挂起
   - USB 挂起
   - SOF
5. 等待 OTG_GINTSTS 中的 USBRST 中断。这表示已在 USB 上检测到复位信号，复位过程自接收到此中断后约持续 10 ms。
6. 等待 OTG_GINTSTS 中的 ENUMDNE 中断。此中断指示 USB 上复位过程结束。接收到此中断时，应用程序必须读取 OTG_DSTS 寄存器以确定枚举速度并执行 *第 2957 页的枚举完成时的端点初始化*中所列的步骤。

此时，设备已准备好接受 SOF 数据包并在控制端点 0 上执行控制传输。

54.15.4 DMA 模式

OTG 主机使用 AHB 主接口来获取发送数据包数据（AHB 到 USB）和接收数据更新（USB 到 AHB）。AHB 主接口使用经过编程的 DMA 地址（主机模式下的 OTG_HCDMAx 寄存器和外设模式下的 OTG_DIEPDMAx/OTG_DOEPDMAx 寄存器）来访问数据缓冲区。

分散/收集 DMA 模式

在分散/收集 DMA 模式中，内核实现了真正的分散/收集内存分布，其中数据缓冲区分散在整个系统内存中。但描述符本身是连续的。每个通道的内存结构均以连续描述符列表的形式实现；每个描述符均指向预定义大小的数据缓冲区。除了缓冲区指针（一个 32 位字）外，描述符还有一个状态四字节组（32 位字）。当列表作为环形缓冲区实现时，列表处理器在遇到最后一个位时切换到列表的第一个元素。所有通道（控制、批量、中断和同步）均在内存中实现这些结构。当在从机模式下使能分散/收集 DMA 时，将使用 OTG_DIEPDMAx 和 OTG_DOEPDMAx 寄存器访问基本描述符。

54.15.5 主机编程模型

通道初始化

应用程序必须初始化一个或多个通道，之后才能与所连接的设备通信。要初始化和使能通道，应用程序必须执行以下步骤：

1. 编程 OTG_GINTMSK 寄存器以取消对以下位的中断屏蔽：
2. 通道中断
   - 用于 OUT 事务的非周期性发送 FIFO 为空（在流水线事务级别工作且数据包计数位字段编程值大于 1 时适用）。
   - 用于 OUT 事务的非周期性发送 FIFO 为半空（在流水线事务级别工作且数据包计数位字段编程值大于 1 时适用）。
3. 编程 OTG_HAINTMSK 寄存器以使能所选通道中断。
4. 编程 OTG_HCINTMSK 寄存器，以使能主机通道中断寄存器中反映的和通信事务有关的中断。
5. 编程所选通道的 OTG_HCTSIZx 寄存器，指定以字节为单位的总传输大小和包括短数据包在内的预期数据包个数。应用程序必须使用初始数据 PID（用于第一个 OUT 事务或预期从第一个 IN 事务获取）编程 PID 位字段。
6. 编程所选通道的 OTG_HCCHARx 寄存器，指定设备的端点特性，例如类型、速度、方向等。（仅当应用程序准备好发送或接收数据包时，才能通过将通道使能位置 1 来使能通道）。
7. 使用集线器地址和端口地址在 OTG_HCSPLTx 寄存器中对所选通道进行编程（仅分离事务）。
8. 使用缓冲区起始地址在 OTG_HCDMAx 寄存器中对所选通道进行编程（仅 DMA 事务）。

通道的停止

应用程序可以通过编程 OTG_HCCHARx 寄存器将 CHDIS 和 CHENA 位置 1 来禁止任何通道。这会使 OTG 主机清空之前在该通道上发出的请求（如果有）并生成通道停止中断。应用程序在将通道重新分配给其他通信事务之前，必须等待 OTG_HCINTx 中的 CHH 中断。OTG 主机不会中断已在 USB 上启动的通信事务。

要禁止某个在 DMA 模式下工作的通道，应用程序无需检查请求队列中是否有可用空间。OTG 主机检查是否有空间，

在仲裁给要禁止的通道时，写入通道禁止的请求。同时，当 OTG_HCCHARx 中的 CHDIS 位置 1 时，将从请求队列中丢弃所有已发出的请求。

禁止通道前，应用程序必须确保非周期性请求队列（禁止非周期性通道时）或周期性请求队列（禁止周期性通道时）中至少有一个空闲空间。应用程序可以在请求队列已满时（禁止通道之前），通过编程 OTG_HCCHARx 寄存器将 CHDIS 位置 1（这会自动将 CHENA 位清零），清空已发出的请求。

出现以下任一情况时，应用程序将禁止通道：

1. IN 或 OUT 通道的 OTG_HCINTx 中接收到 STALL、TXERR、BBERR 或 DTERR 中断。应用程序在接收到通道停止信号之前，必须能够接收相同通道的其他中断（DTERR、Nak、data、TXERR）。
2. 在非周期性 IN 传输或高带宽中断 IN 传输期间 OTG_HCINTx 中接收到 XFRC 中断
3. 接收到 OTG_GINTSTS 中的 DISCINT（断开设备连接）中断。（应用程序将禁止所有已使能的通道）。
4. 应用程序在传输正常完成之前将其中止。

Ping 协议

当 OTG 主机工作在高速模式下时，如果应用程序要与高速批量或控制（数据和状态阶段）OUT 端点进行通信，则必须使用 ping 协议。当应用程序接收到 NAK/NYET/TXERR 中断时，必须使用 ping 协议。当 OTG 主机接收到上述其中一个响应时，它不会在该端点上继续执行任何通信事务，而是丢弃所有已发出或已获取的 OUT 请求（从请求队列中），然后清空发送 FIFO 中的相应数据。仅对从模式有效。在从模式下，应用程序可通过以下两种方式发送 ping 令牌：在使能通道之前，将 OTG_HCTSIZx 中的 DOPING 位置 1，或者在通道已使能之后，对 OTG_HCTSIZx 寄存器执行写操作时将 DOPING 位置 1。这将使能 OTG 主机将 ping 请求写入到请求队列中。应用程序必须等待设备对 ping 令牌的响应（NAK、ACK 或 TXERR 中断），然后才能继续执行通信事务或发送另一个 ping 令牌。仅当应用程序从设备的 OUT 端点接收到对 ping 令牌的 ACK 响应后，才能继续执行数据通信事务。在 DMA 模式下工作时，对于批量/控制 OUT 事务，应用程序不需要在收到 NAK/NYET 回复时将 OTG_HCTSIZx 中的 DOPING 位置 1。OTG 主机会自动将 OTG_HCTSIZx 中的 DOPING 位置 1，然后在批量/控制 OUT 传输时发出 ping 令牌。OTG 主机会持续发送 ping 令牌，直至收到 ACK 为止，随后自动切换到数据通信事务。

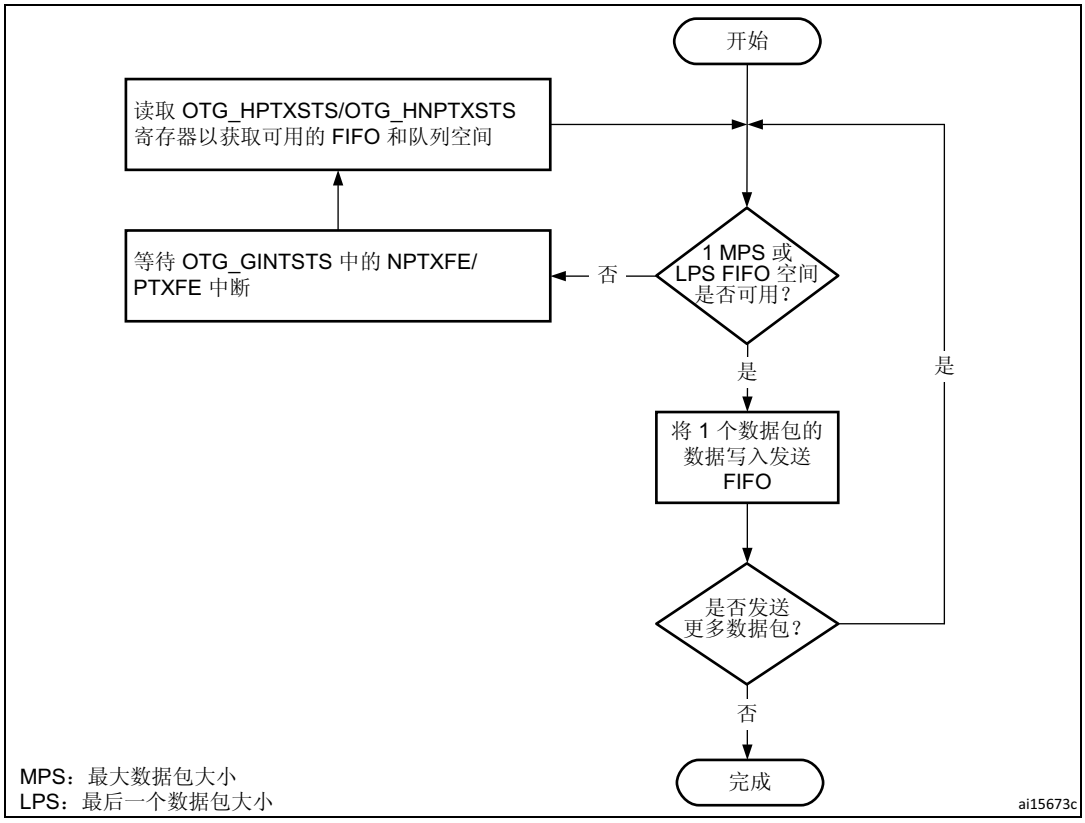
操作模型

应用程序必须初始化一个通道，之后才能与所连接的设备通信。本节介绍了针对不同 USB 事务类型要执行的操作序列。

- **写入发送 FIFO**

  应用程序对数据包执行最后一个 32 位字写操作的同时，OTG 主机自动向周期性/非周期性请求队列写入一个条目（OUT 请求）。因此开始向发送 FIFO 写入数据之前，应用程序必须确保周期性/非周期性请求队列中至少有一个空闲空间。应用程序必须始终以 32 位字的形式向发送 FIFO 写入数据。如果数据包大小不是 32 位字的整数倍，则应用程序必须将数据包填充到 32 位字的整数倍。OTG 主机根据设定的最大数据包大小和传输大小确定实际数据包大小。

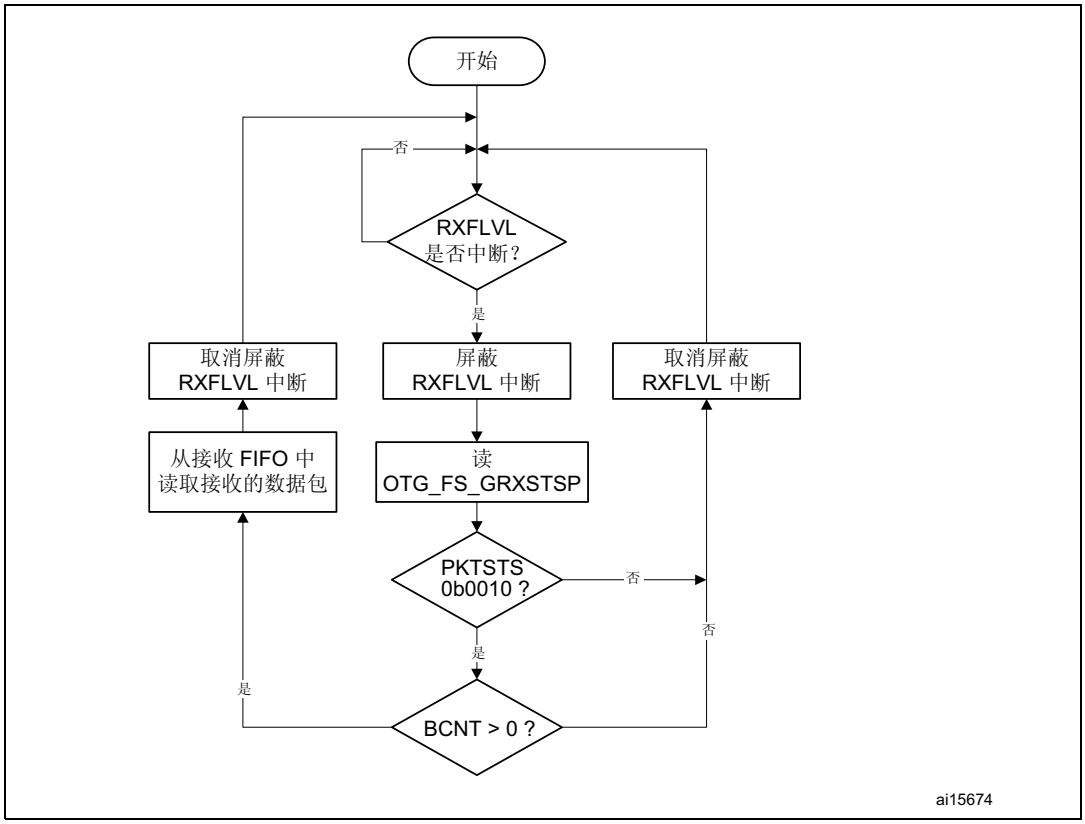
图 690. 发送 FIFO 写任务



- **读取接收 FIFO**

  应用程序必须忽略除 IN 数据包 (bx0010) 以外的所有数据包状态。

图 691. 接收 FIFO 读任务



- **批量和控制传输类型的 OUT/SETUP 通信事务**

  图 692 显示了典型的批量或控制 OUT/SETUP 流水线事务级操作。请参见通道 1 (ch_1)。该通道发送了两个批量 OUT 数据包。控制 SETUP 事务的工作方式相同，只不过只包含一个数据包。假设：

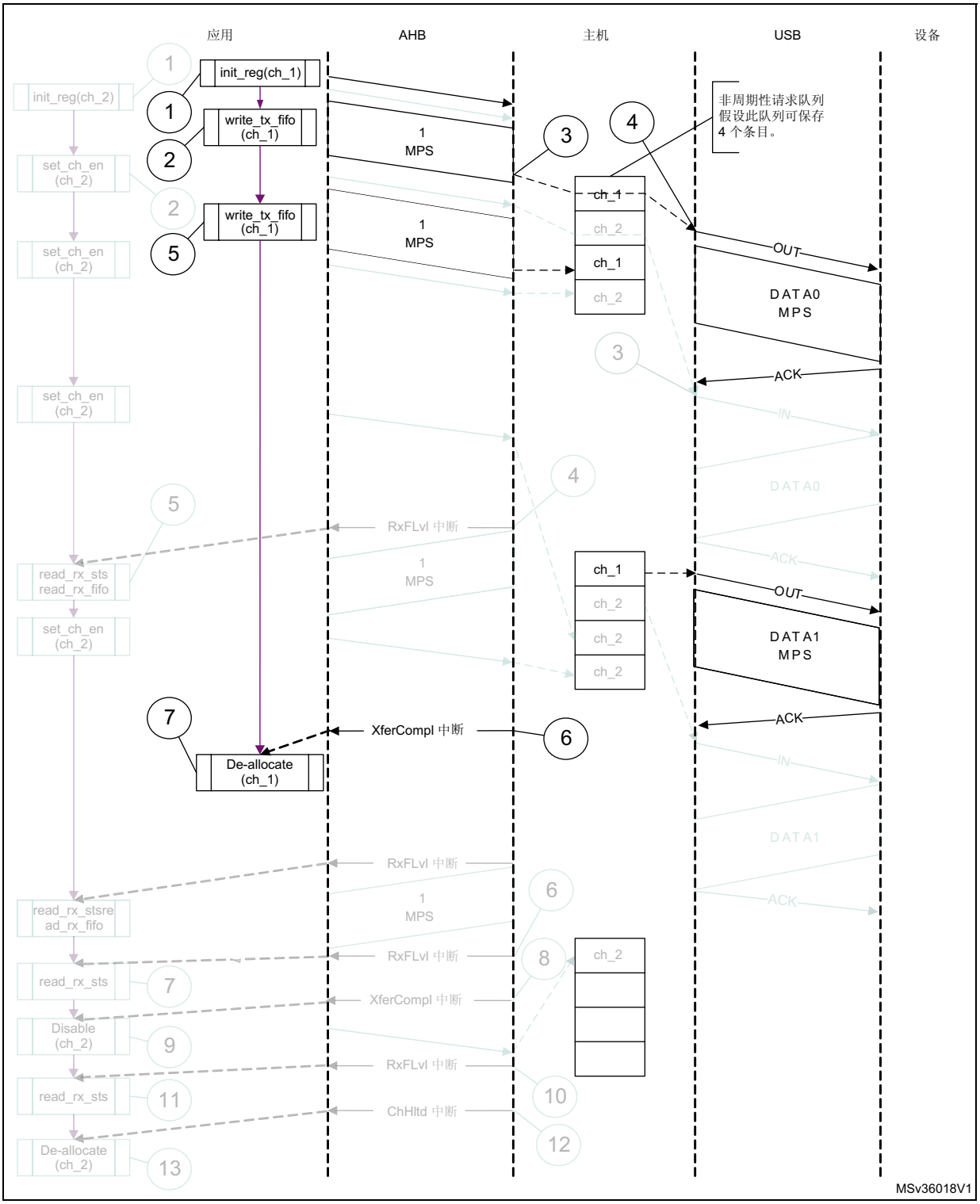
  – 应用程序尝试发送两个最大数据包大小的数据包（传输大小 = 1024 字节）。
  – 非周期性发送 FIFO 可存储两个数据包（HS 模式下为 1 KB）。
  – 非周期性请求队列深度 = 4。

- **正常批量和控制传输类型的 OUT/SETUP 操作**

  （通道 1）中的操作顺序如下：

1. 初始化通道 1
2. 写入通道 1 的第一个数据包
3. 在应用执行最后一次字写操作时，模块将向非周期性请求队列写入一个请求条目
4. 只要非周期性队列非空，模块即会尝试在当前帧内发送一个 OUT 令牌
5. 写入通道 1 的第二个（最后一个）数据包
6. 成功完成最后一个事务后，模块立即生成 XFRC 中断
7. 为了响应 XFRC 中断，将释放通道以供其他传输操作使用
8. 处理非 ACK 响应

图 692. 正常批量/控制 OUT/SETUP



1. 灰显元素与此图的上下文不相关。

以下代码示例说明了批量和控制 OUT/SETUP 传输类型的通信事务的通道相关中断服务程序。

- **批量/控制 OUT/SETUP 和批量/控制 IN 事务的中断服务程序**

  a) 批量/控制 OUT/SETUP

```
Unmask (NAK/TXERR/STALL/XFRC)
if (XFRC)
   {
   Reset Error Count
   Mask ACK
   De-allocate Channel
   }
else if (STALL)
   {
   Transfer Done = 1
   Unmask CHH
   Disable Channel
   }
else if (NAK or TXERR )
   {
   Rewind Buffer Pointers
   Unmask CHH
   Disable Channel
   if (TXERR)
      {
      Increment Error Count
      Unmask ACK
      }
   else
      {
      Reset Error Count
      }
      }
else if (CHH)
   {
   Mask CHH
   if (Transfer Done or (Error_count == 3))
      {
      De-allocate Channel
      }
   else
      {
      Re-initialize Channel
      }
   }
   else if (ACK)
```

```
      {
      Reset Error Count
      Mask ACK
      }
```

当发送 FIFO 和请求队列中有可用空间时，应用程序会将数据包写入发送 FIFO。应用程序可利用 OTG_GINTSTS 中的 NPTXFE 中断确定发送 FIFO 空间。

b) 批量/控制 IN

```
Unmask (TXERR/XFRC/BBERR/STALL/DTERR)
if (XFRC)
   {
   Reset Error Count
   Unmask CHH
   Disable Channel
   Reset Error Count
   Mask ACK
   }
else if (TXERR or BBERR or STALL)
   {
   Unmask CHH
   Disable Channel
   if (TXERR)
      {
      Increment Error Count
      Unmask ACK
      }
   }
else if (CHH)
   {
   Mask CHH
   if (Transfer Done or (Error_count == 3))
      {
      De-allocate Channel
      }
   else
      {
      Re-initialize Channel
      }
   }
else if (ACK)
   {
   Reset Error Count
   Mask ACK
   }
else if (DTERR)
   {
```

```
Reset Error Count
}
```

当请求队列空间可用时，应用程序会写入请求，直到接收到 XFRC 中断。

- **批量和控制 IN 事务**

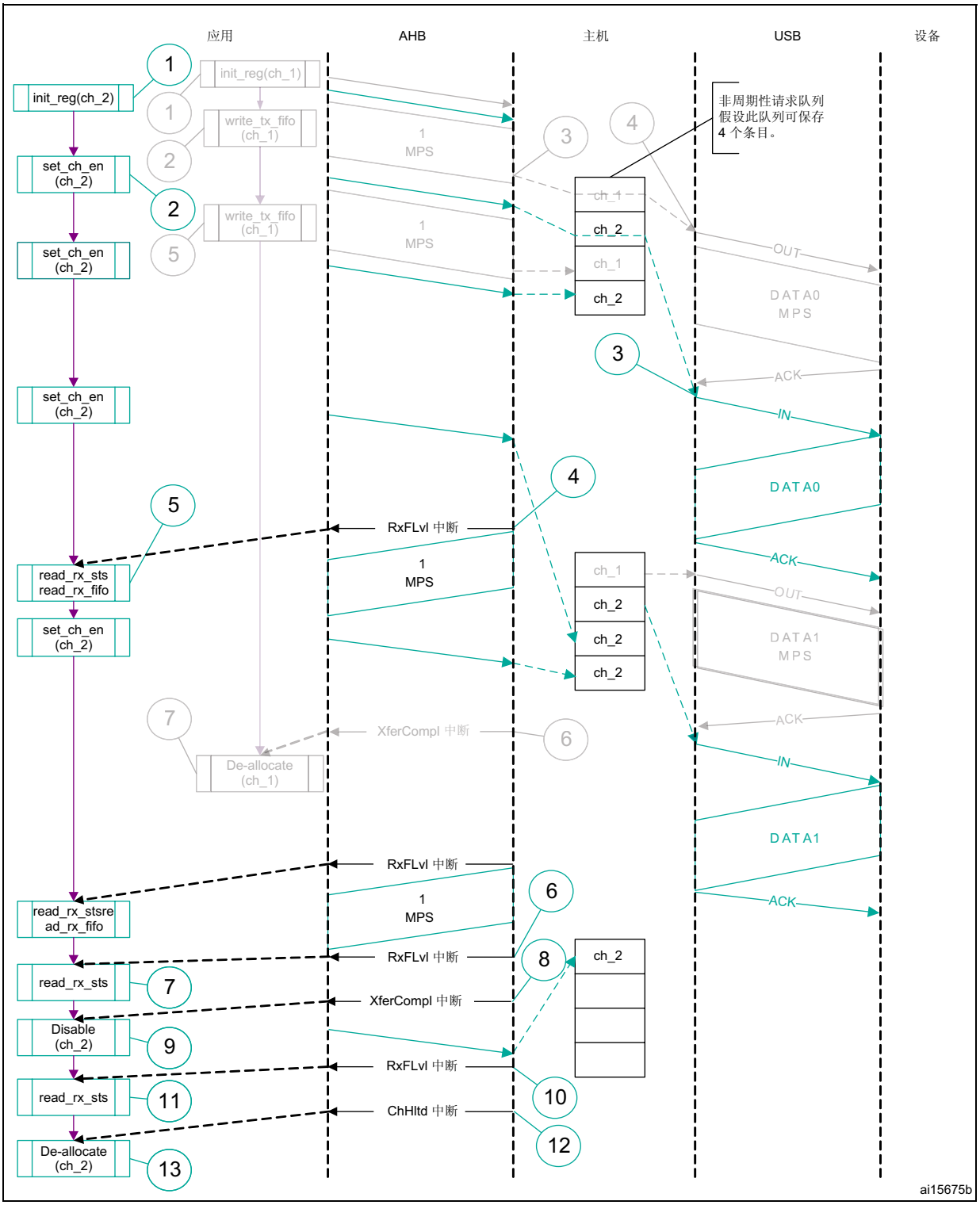
  图 693 显示了典型的批量或控制传输类型的 IN 流水线事务级操作。请参见通道 2 (ch_2)。假设：

  – 应用程序要接收两个最大数据包大小的数据包（传输大小 = 1024 字节）。

  – 接收 FIFO 可以包含至少一个最大数据包大小的数据包和每个数据包的两个状态字（HS 对应 520 字节）。

  – 非周期性请求队列深度 = 4。

图 693. 批量/控制 IN 事务



1. 灰显元素与此图的上下文不相关。

操作顺序如下：

1. 初始化通道 2。
2. 将 OTG_HCCHAR2 中的 CHENA 位置 1，以向非周期性请求队列写入 IN 请求。
3. 模块尝试在完成当前 OUT 事务后发送 IN 令牌。
4. 接收到的数据包写入接收 FIFO 后，模块立即生成 RXFLVL 中断。
5. 为了响应 RXFLVL 中断，将屏蔽 RXFLVL 中断并读取接收到的数据包状态，以此确定接收的字节数，然后相应地读取接收 FIFO。之后使能 RXFLVL 中断。
6. 模块针对接收 FIFO 中的传输完成状态条目生成 RXFLVL 中断。
7. 应用程序必须读取接收数据包状态，并在不是 IN 数据包（OTG_GRXSTSR 中的 PKTSTS ≠ 0b0010）时忽略它。
8. 读取接收数据包状态后，模块立即生成 XFRC 中断。
9. 为了响应 XFRC 中断，将禁止通道并停止针对其他请求向 OTG_HCCHAR2 写入数据。向 OTG_HCCHAR2 寄存器写入数据时，模块立即向非周期性请求队列写入通道禁止请求。
10. 停止状态写入接收 FIFO 后，模块立即生成 RXFLVL 中断。
11. 读取并忽略接收数据包状态。
12. 只要停止状态从接收 FIFO 弹出，模块立即生成 CHH 中断。
13. 为了响应 CHH 中断，将释放通道以供其他传输操作使用。
14. 处理非 ACK 响应

- **控制事务**

  控制传输的建立、数据和状态阶段必须作为三个独立的传输过程来执行。建立、数据和状态阶段的 OUT 事务与上文所述的批量 OUT 事务的执行方式类似。数据或状态阶段的 IN 事务与上文所述的批量 IN 事务的执行方式类似。在所有这三个阶段，应用程序都会将 OTG_HCCHAR1 中的 EPTYP 位字段设置为控制传输类型。在建立阶段期间，应用程序会将 OTG_HCTSIZ1 中的 PID 位字段设置为 SETUP。

- **中断 OUT 事务**

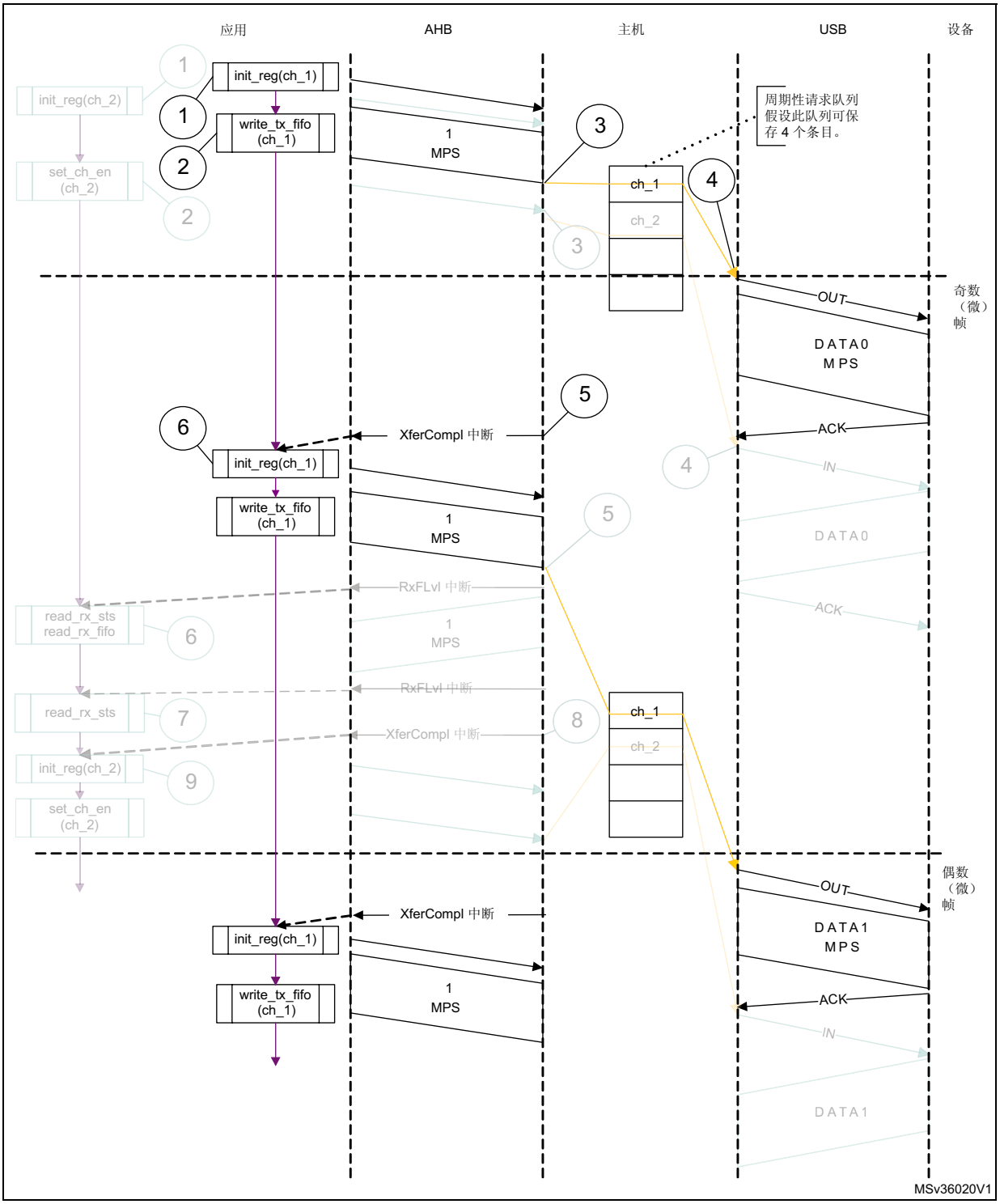
  图 694 显示了典型的中断 OUT 操作。假设：

  – 应用程序尝试从奇数帧开始，每帧发送一个数据包（高达 1 个最大数据包大小）（传输大小 = 1024 字节）
  – 周期性发送 FIFO 可存储一个数据包 (1 KB)
  – 周期性请求队列深度 = 4

  操作顺序如下：

1. 初始化和使能通道 1。应用程序必须将 OTG_HCCHAR1 中的 ODDFRM 位置 1。
2. 写入通道 1 的第一个数据包。
3. 应用程序在执行每个数据包的最后一次字写操作时，OTG 主机向周期性请求队列写入一个请求条目。
4. OTG 主机尝试在下一（奇数）帧发送 OUT 令牌。
5. 成功发送最后一个数据包后，OTG 主机立即生成 XFRC 中断。
6. 为了响应 XFRC 中断，将重新初始化通道以供下一传输操作使用。

图 694. 正常中断 OUT



1. 灰显元素与此图的上下文不相关。

- **中断 OUT/IN 事务的中断服务程序**

  a) 中断 OUT

```
Unmask (NAK/TXERR/STALL/XFRC/FRMOR)
```

```
if (XFRC)
   {
   Reset Error Count
   Mask ACK
   De-allocate Channel
   }
else
   if (STALL or FRMOR)
      {
      Mask ACK
      Unmask CHH
      Disable Channel
      if (STALL)
         {
         Transfer Done = 1
         }
      }
   else
      if (NAK or TXERR)
         {
         Rewind Buffer Pointers
         Reset Error Count
         Mask ACK
         Unmask CHH
         Disable Channel
         }
      else
         if (CHH)
         {
         Mask CHH
         if (Transfer Done or (Error_count == 3))
            {
            De-allocate Channel
            }
         else
            {
            Re-initialize Channel (in next b_interval - 1 Frame)
            }
         }
      else
         if (ACK)
         {
         Reset Error Count
         Mask ACK
         }
```

应用程序利用 OTG_GINTSTS 中的 NPTXFE 中断确定发送 FIFO 空间。

中断 IN

```
Unmask (NAK/TXERR/XFRC/BBERR/STALL/FRMOR/DTERR)
if (XFRC)
   {
   Reset Error Count
   Mask ACK
   if (OTG_HCTSIZx.PKTCNT == 0)
      {
      De-allocate Channel
      }
   else
      {
      Transfer Done = 1
      Unmask CHH
      Disable Channel
      }
   }
else
   if (STALL or FRMOR or NAK or DTERR or BBERR)
      {
      Mask ACK
      Unmask CHH
      Disable Channel
      if (STALL or BBERR)
         {
         Reset Error Count
         Transfer Done = 1
         }
      else
         if (!FRMOR)
         {
         Reset Error Count
         }
      }
else
   if (TXERR)
      {
      Increment Error Count
      Unmask ACK
      Unmask CHH
      Disable Channel
      }
else
   if (CHH)
```

```
      {
      Mask CHH
      if (Transfer Done or (Error_count == 3))
         {
         De-allocate Channel
         }
      else
         Re-initialize Channel (in next b_interval - 1 /Frame)
      }
   }
else
   if (ACK)
      {
      Reset Error Count
      Mask ACK
      }
```

- **中断 IN 事务**

  假设：

  – 应用程序要从奇数帧开始，每帧接收一个数据包（最大 1 个最大数据包大小）（传输大小 = 1024 字节）。

  – 接收 FIFO 可以保存至少一个最大数据包大小的数据包和每个数据包的两个状态字（1031 字节）。

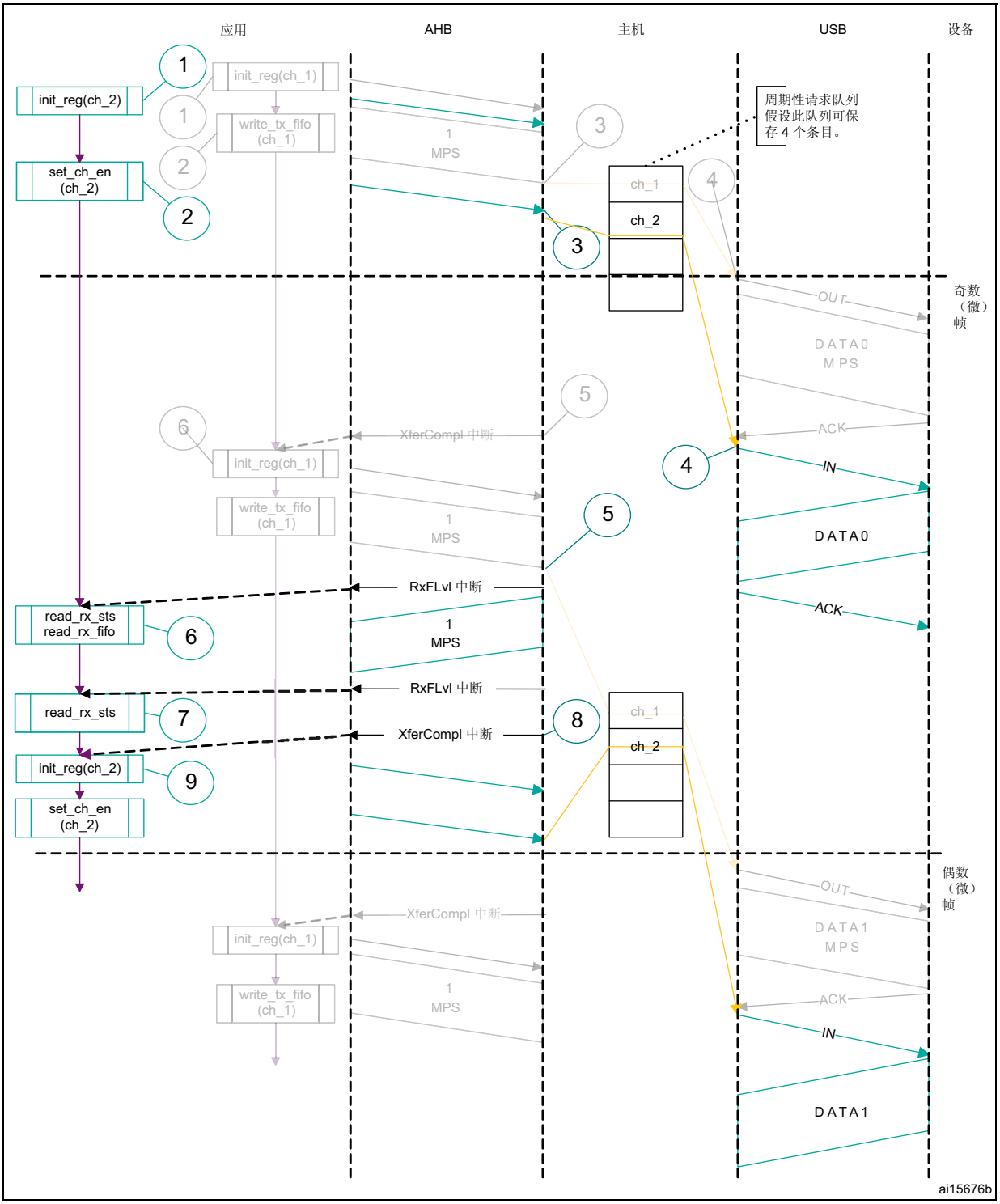
  – 周期性请求队列深度 = 4。

- **正常中断 IN 操作**

  操作顺序如下：

1. 初始化通道 2。应用程序必须将 OTG_HCCHAR2 中的 ODDFRM 位置 1。
2. 将 OTG_HCCHAR2 中的 CHENA 位置 1，以将 IN 请求写入周期性请求队列。
3. 只要 CHENA 置位，对于每次 OTG_HCCHAR2 寄存器写操作，OTG 主机都会将一个 IN 请求写入周期性请求队列。
4. OTG 主机尝试在下一（奇数）帧发送 IN 令牌。
5. 接收到 IN 数据包并写入接收 FIFO 后，OTG 主机便会立即生成 RXFLVL 中断。
6. 为响应 RXFLVL 中断，读取接收到的数据包状态以确定接收的字节数，然后相应地读取接收 FIFO。应用程序必须在读取接收 FIFO 前屏蔽 RXFLVL 中断，并且在读取整个数据包后使能。
7. 模块针对接收 FIFO 中的传输完成状态条目生成 RXFLVL 中断。应用程序必须读取接收数据包状态，并在不是 IN 数据包（GRXSTSR 中的 PKTSTS ≠ 0b0010）时忽略它。
8. 读取接收数据包状态后，模块立即生成 XFRC 中断。
9. 为响应 XFRC 中断，将读取 OTG_HCTSIZ2 中的 PKTCNT 位字段。如果 OTG_HCTSIZ2 中的 PKTCNT 位不等于 0，则在重新初始化通道以进行下次传输前（如果存在），禁止该通道。如果 OTG_HCTSIZ2 中的 PKTCNT 位等于 0，则重新初始化通道以进行下次传输。此时，应用程序必须复位 OTG_HCCHAR2 中的 ODDFRM 位。

图 695. 正常中断 IN



1. 灰显元素与此图的上下文不相关。

- **同步 OUT 事务**

  图 696 中显示了典型的同步 OUT 操作。假设：

  – 应用程序尝试从奇数帧开始，每帧发送一个数据包（最大 1 个最大数据包大小）。（传输大小 = 1024 字节）。

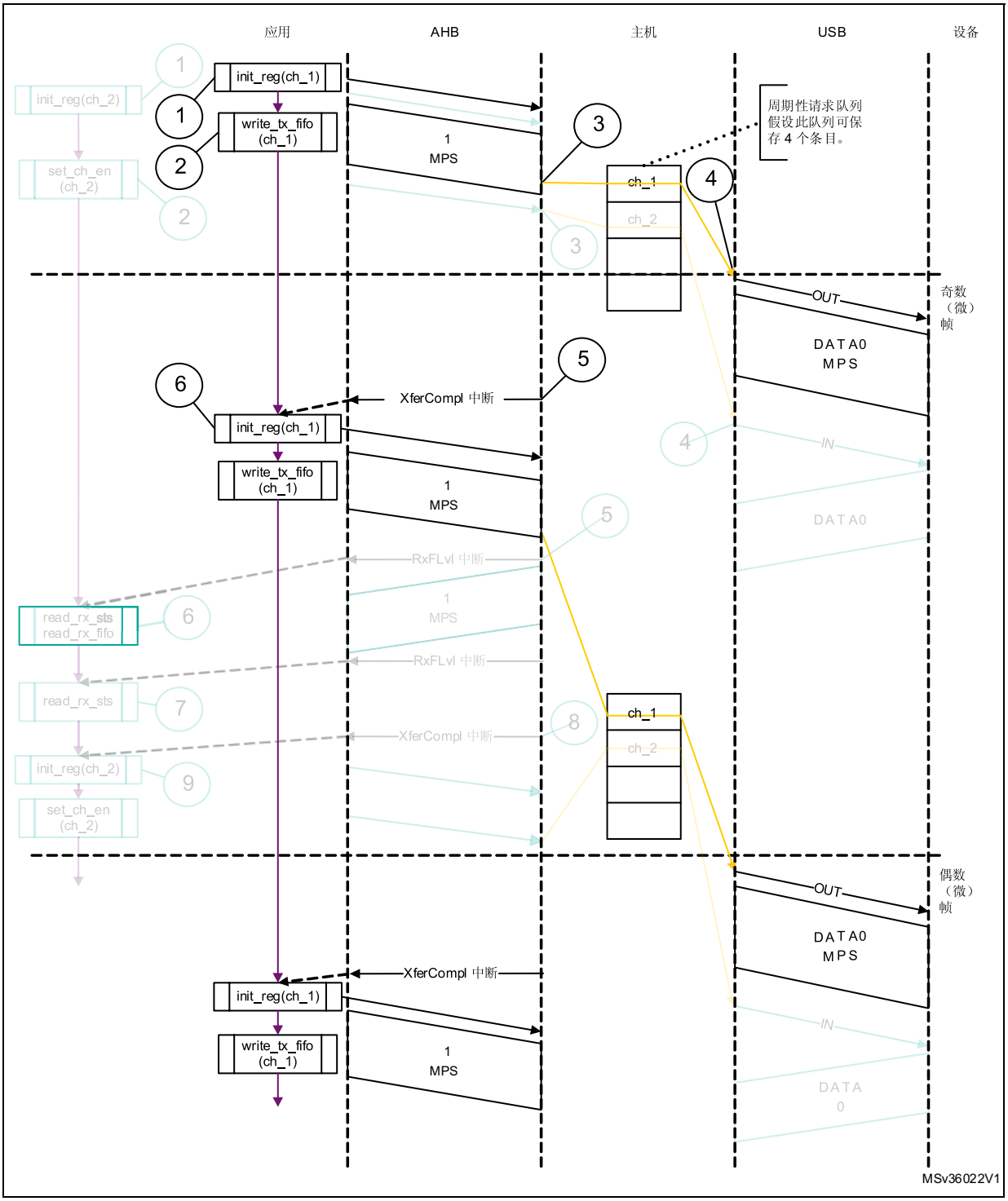
  – 周期性发送 FIFO 可存储一个数据包 (1 KB)。

  – 周期性请求队列深度 = 4。

  操作顺序如下：

1. 初始化和使能通道 1。应用程序必须将 OTG_HCCHAR1 中的 ODDFRM 位置 1。
2. 写入通道 1 的第一个数据包。
3. 应用程序在执行每个数据包的最后一次字写操作时，OTG 主机向周期性请求队列写入一个请求条目。
4. OTG 主机尝试在下一（奇数）帧发送 OUT 令牌。
5. 成功发送最后一个数据包后，OTG 主机立即生成 XFRC 中断。
6. 为了响应 XFRC 中断，将重新初始化通道以供下一传输操作使用。
7. 处理非 ACK 响应

图 696. 同步 OUT 事务



1. 灰显元素与此图的上下文不相关。

- **同步 OUT/IN 事务的中断服务程序**

  代码示例：同步 OUT

```
Unmask (FRMOR/XFRC)
```

```
if (XFRC)
   {
   De-allocate Channel
   }
else
   if (FRMOR)
      {
      Unmask CHH
      Disable Channel
      }
   else
   if (CHH)
      {
      Mask CHH
      De-allocate Channel
      }
```

代码示例：Isochronous IN

```
Unmask (TXERR/XFRC/FRMOR/BBERR)
if (XFRC or FRMOR)
   {
   if (XFRC and (OTG_HCTSIZx.PKTCNT == 0))
      {
      Reset Error Count
      De-allocate Channel
      }
   else
      {
      Unmask CHH
      Disable Channel
      }
   }
   else
      if (TXERR or BBERR)
         {
         Increment Error Count
         Unmask CHH
         Disable Channel
         }
      else
         if (CHH)
            {
            Mask CHH
            if (Transfer Done or (Error_count == 3))
               {
               De-allocate Channel
```

```
    }
  else
    {
    Re-initialize Channel
    }
  }
```

- **同步 IN 事务**

  假设：

  – 应用程序尝试从下一个奇数帧开始，每帧接收一个数据包（最大 1 个最大数据包大小）（传输大小 = 1024 字节）。

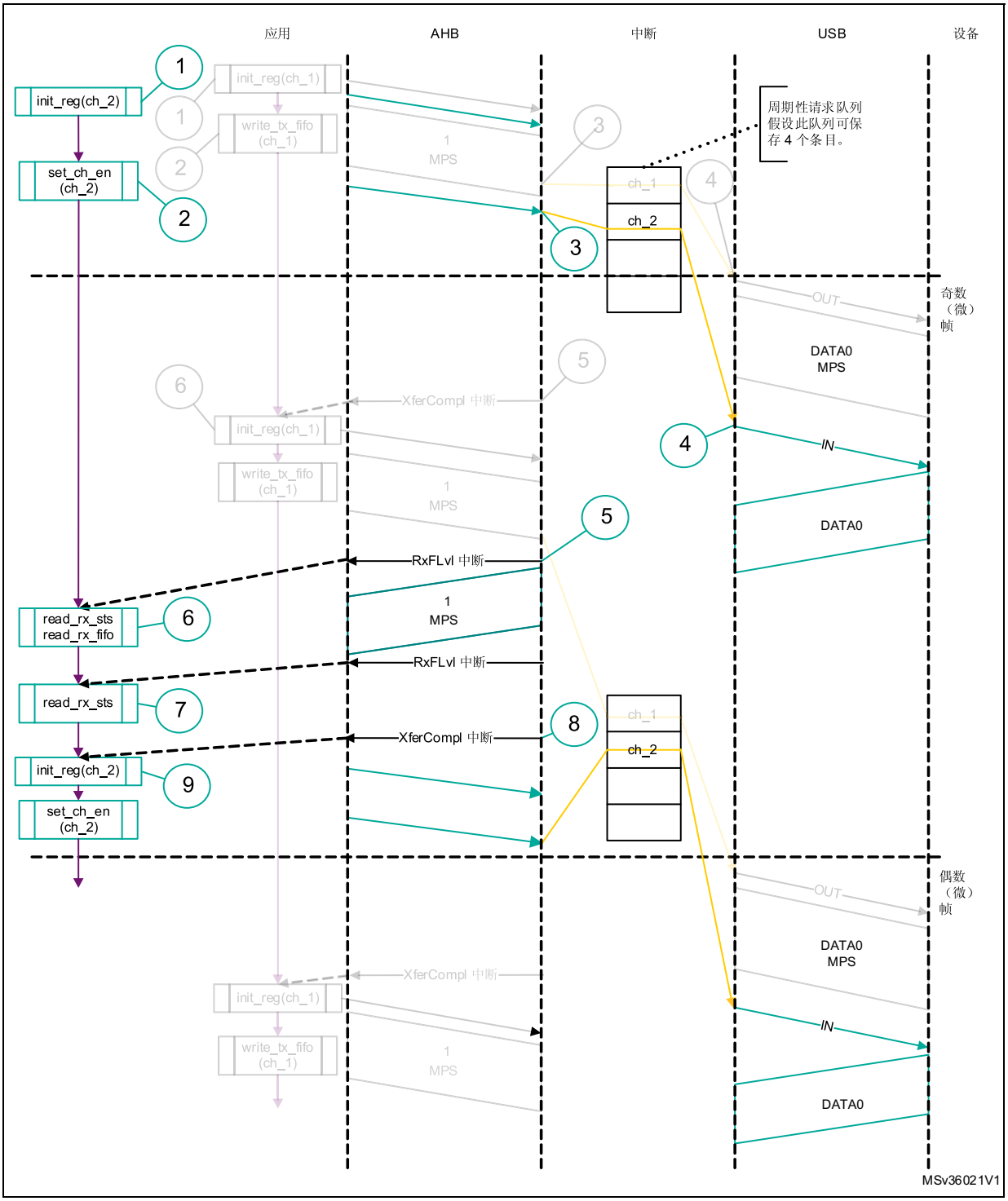
  – 接收 FIFO 可以保存至少一个最大数据包大小的数据包和每个数据包的两个状态字（1031 字节）。

  – 周期性请求队列深度 = 4。

  操作顺序如下：

1. 初始化通道 2。应用程序必须将 OTG_HCCHAR2 中的 ODDFRM 位置 1。
2. 将 OTG_HCCHAR2 中的 CHENA 位置 1，以将 IN 请求写入周期性请求队列。
3. 只要 CHENA 置位，对于每次 OTG_HCCHAR2 寄存器写操作，OTG 主机都会将一个 IN 请求写入周期性请求队列。
4. OTG 主机尝试在下一奇数帧发送 IN 令牌。
5. 接收到 IN 数据包并写入接收 FIFO 后，OTG 主机便会立即生成 RXFLVL 中断。
6. 为响应 RXFLVL 中断，读取接收到的数据包状态以确定接收的字节数，然后相应地读取接收 FIFO。应用程序必须在读取接收 FIFO 前屏蔽 RXFLVL 中断，并且在读取整个数据包后使能。
7. 模块针对接收 FIFO 中的传输完成状态条目生成 RXFLVL 中断。应用程序必须读取接收数据包状态，并在不是 IN 数据包（OTG_GRXSTSR 中的 PKTSTS 位 ≠ 0b0010）时忽略它。
8. 读取接收数据包状态后，模块立即生成 XFRC 中断。
9. 为响应 XFRC 中断，将读取 OTG_HCTSIZ2 中的 PKTCNT 位字段。如果在 OTG_HCTSIZ2 中的 PKTCNT ≠ 0，则在重新初始化通道以进行下次传输前（如果存在），禁止该通道。如果 OTG_HCTSIZ2 中的 PKTCNT = 0，则重新初始化通道以进行下次传输。此时，应用程序必须复位 OTG_HCCHAR2 中的 ODDFRM 位。

图 697. 同步 IN 事务



1. 灰显元素与此图的上下文不相关。

- **选择队列深度**

  请谨慎选择周期性和非周期性请求队列深度，以与要访问的周期性/非周期性端点数量相匹配。

  非周期性请求队列深度会影响非周期性传输的性能。队列越深（加上足够的 FIFO 大小），模块就更能对非周期性传输进行流水线处理。如果队列大小太小，则仅在队列空间释放时模块才能放入新请求。

  要按调度计划执行周期性传输，模块的周期性请求队列深度至关重要。根据一个微帧内要安排的周期性传输次数，选择周期性队列深度。如果周期性请求队列深度小于一个微帧内要安排的周期性传输数，则会发生帧溢出情况。

- **处理 babble 情况**

  OTG 控制器处理两种情况的 babble：数据包 babble 和端口 babble。如果设备发送的数据量超过通道的最大数据包大小，则会发生数据包串扰。如果模块从 EOF2（非常接近下一帧的 SOF）时刻还在从设备接收数据，则发生端口串扰。

  当 OTG 控制器检测到数据包 babble 时，它停止向 Rx 缓冲区中写入数据并等待数据包结束信号 (EOP)。当该控制器检测到 EOP 时，它会清空 Rx 缓冲区中的已写入数据并对应用程序生成串扰中断。

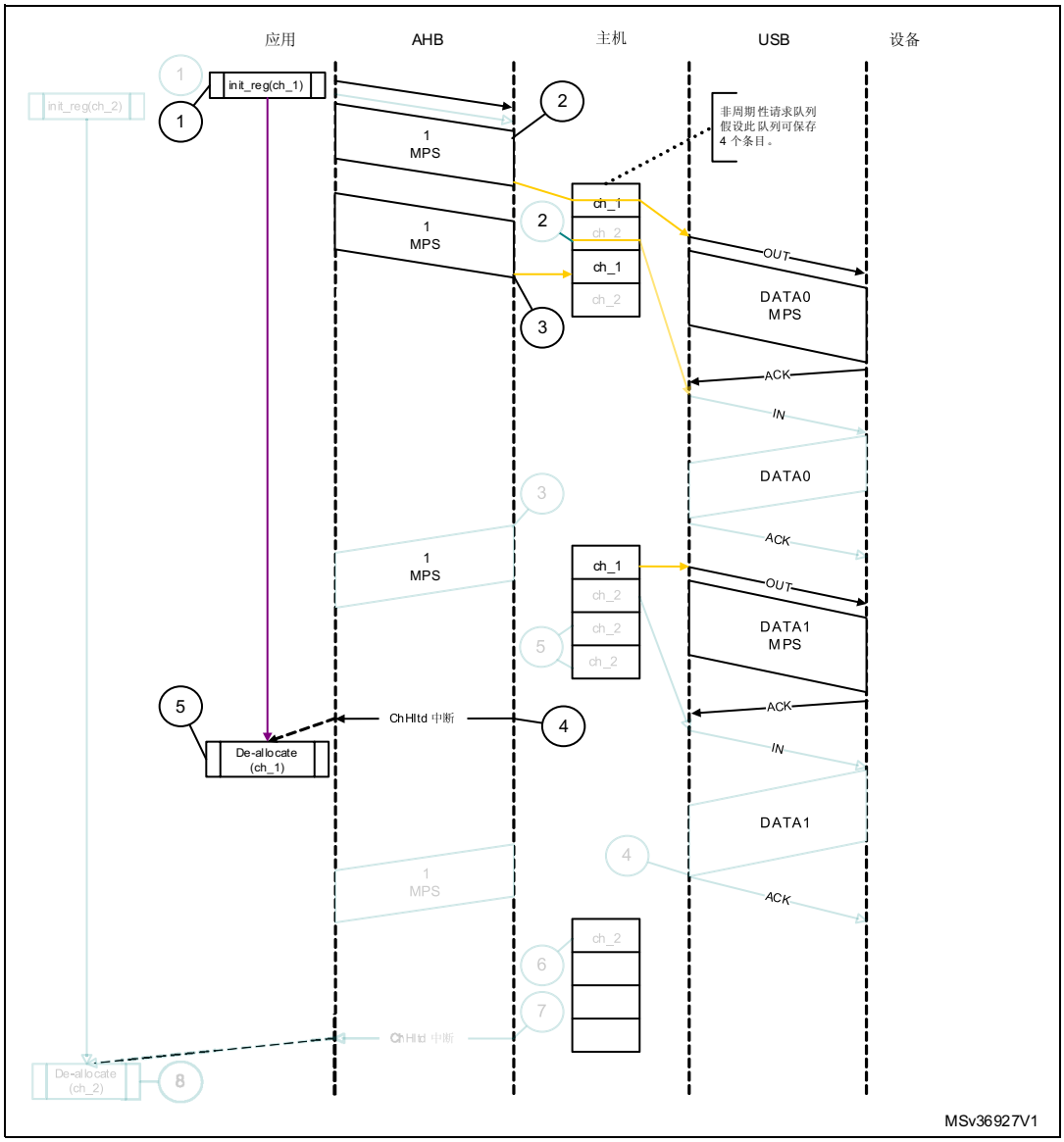
  当 OTG 控制器检测到端口 babble 时，它会清空 Rx FIFO 并禁止端口。模块随后将生成“端口已禁止”中断（OTG_GINTSTS 中的 HPRTINT 位和 OTG_HPRT 中的 PENCHNG 位）。在收到该中断时，应用程序必须通过检查 OTG_HPRT 中的 POCA 位，来确定该中断并非由过流（“端口已禁止”中断的另一个原因）所致，然后执行软复位。检测到端口串扰情况后，模块不再发送任何其他令牌。

- **DMA 模式下的批量和控制 OUT/SETUP 事务**

  操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 1。
2. 使能通道后，OTG 主机即开始获取第一个数据包以发送。对于内部 DMA 模式，OTG 主机使用经过编程的 DMA 地址来获取数据包。
3. 获取第二个（最后一个）数据包的最后一个 32 位字后，OTG 主机会在内部屏蔽通道 1，不参与仲裁。
4. 最后一个数据包发送出去后，OTG 主机立即生成一个 CHH 中断。
5. 为了响应 CHH 中断，将释放通道以供其他传输操作使用。

图 698. 正常批量/控制 OUT/SETUP 事务 - DMA



- **使用内部 DMA 处理 NAK 和 NYET：**

1. OTG 主机发送批量传输类型的 OUT 事务。
2. 设备回复 NAK 或 NYET。
3. 如果应用程序没有屏蔽 NAK 或 NYET 中断，模块将为应用程序生成相应的中断。应用程序并不需要处理这些中断，因为模块会负责调整缓冲区指针和重新初始化通道，而无需应用程序干预。
4. 模块自动发出一个 ping 令牌。
5. 当设备返回 ACK 后，模块会继续传输。应用程序也可以选择使用这些中断（在这种情况下，NAK 或 NYET 中断将被应用程序屏蔽）。

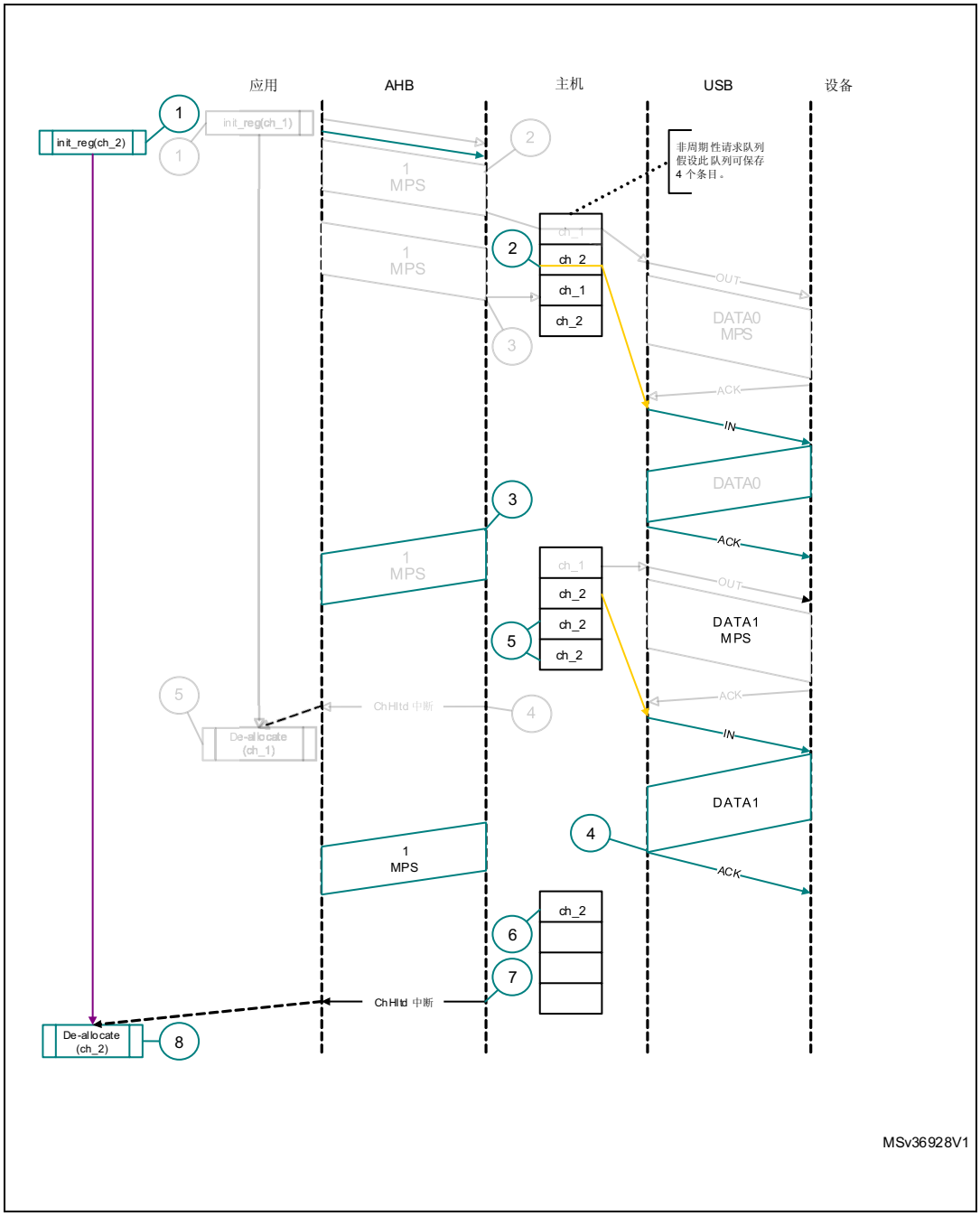
   当主机接收到 NAK 或 NYET 时，模块不会生成单独的中断。

- **DMA 模式下的批量和控制 IN 事务**

  操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能所使用的通道（通道 x）。
2. 当通道接收到来自仲裁器的授权后（以循环方式执行仲裁），OTG 会立即向请求队列写入一个 IN 请求。
3. 当正确接收到最后一个字节后，OTG 主机便开始将接收到的数据写入到系统内存中。
4. 当接收到最后一个数据包后，OTG 主机会将一个内部标志置 1，以便从请求队列中删除任何额外的 IN 请求。
5. OTG 主机会清空额外请求。
6. 向请求队列写入最后一个禁止通道的请求。此时，会在内部屏蔽通道 2，不参与仲裁。
7. 当禁止请求到达队列顶端时，OTG 主机即会生成 CHH 中断。
8. 为了响应 CHH 中断，将释放通道以供其他传输操作使用。

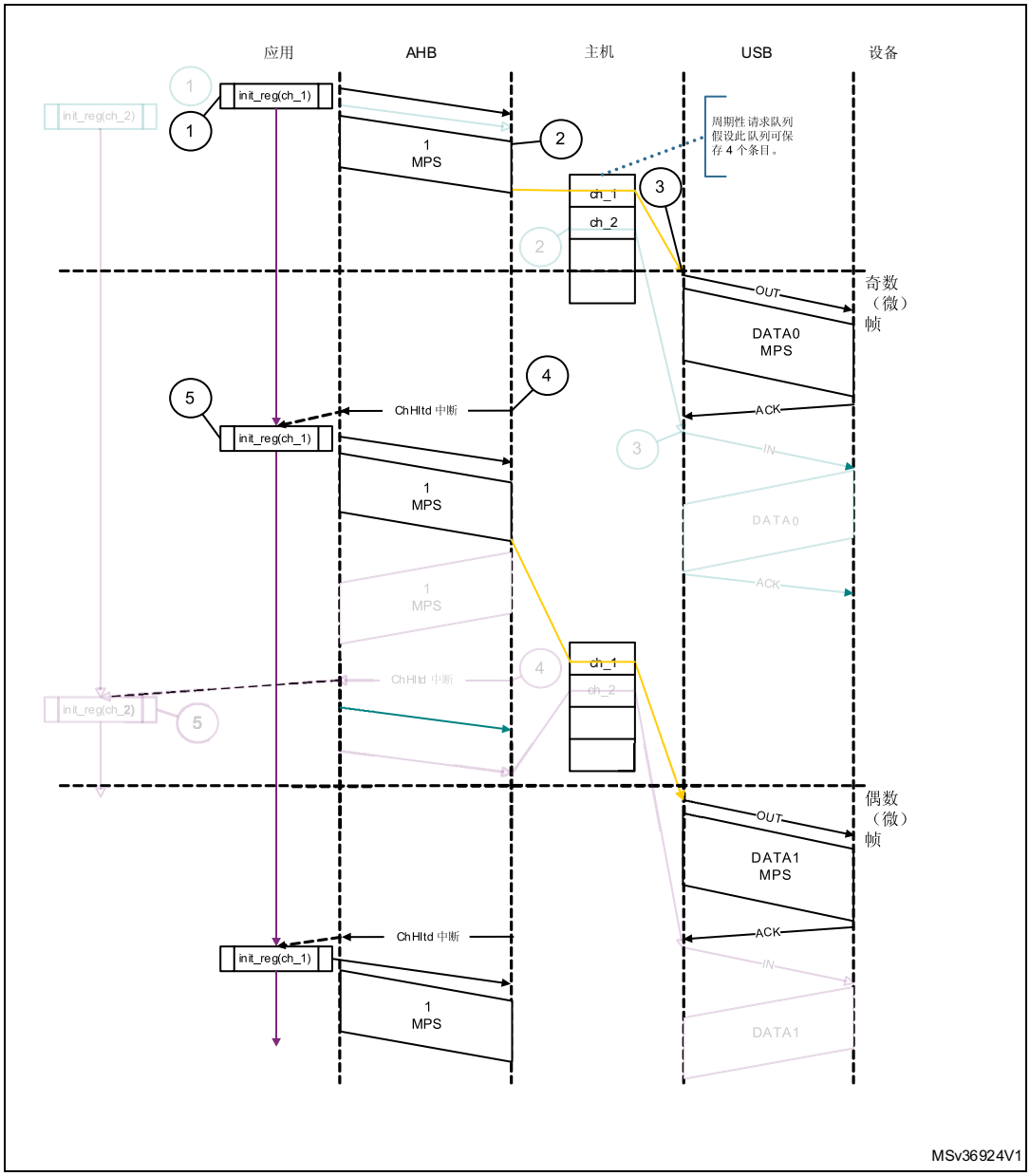
图 699. 正常批量/控制 IN 事务 - DMA



- **DMA 模式下的中断 OUT 事务**

1. 按照*通道初始化*一节中的说明初始化并使能通道 x。
2. 使能通道后，OTG 主机开始获取第一个数据包以发送，并在获取数据包的最后一个 32 位字后写入 OUT 请求。在高带宽传输模式下，OTG 主机会继续获取下一个数据包（直至数据包数量达到 MC 位字段中指定的数值），然后才能切换到下一个通道。
3. OTG 主机尝试在下一个奇数帧/微帧的起始位置发送 OUT 令牌。
4. 成功发送数据包后，OTG 主机将生成一个 CHH 中断。
5. 为了响应 CHH 中断，将重新初始化通道以供下一传输操作使用。

图 700. 正常中断 OUT 事务 - DMA 模式

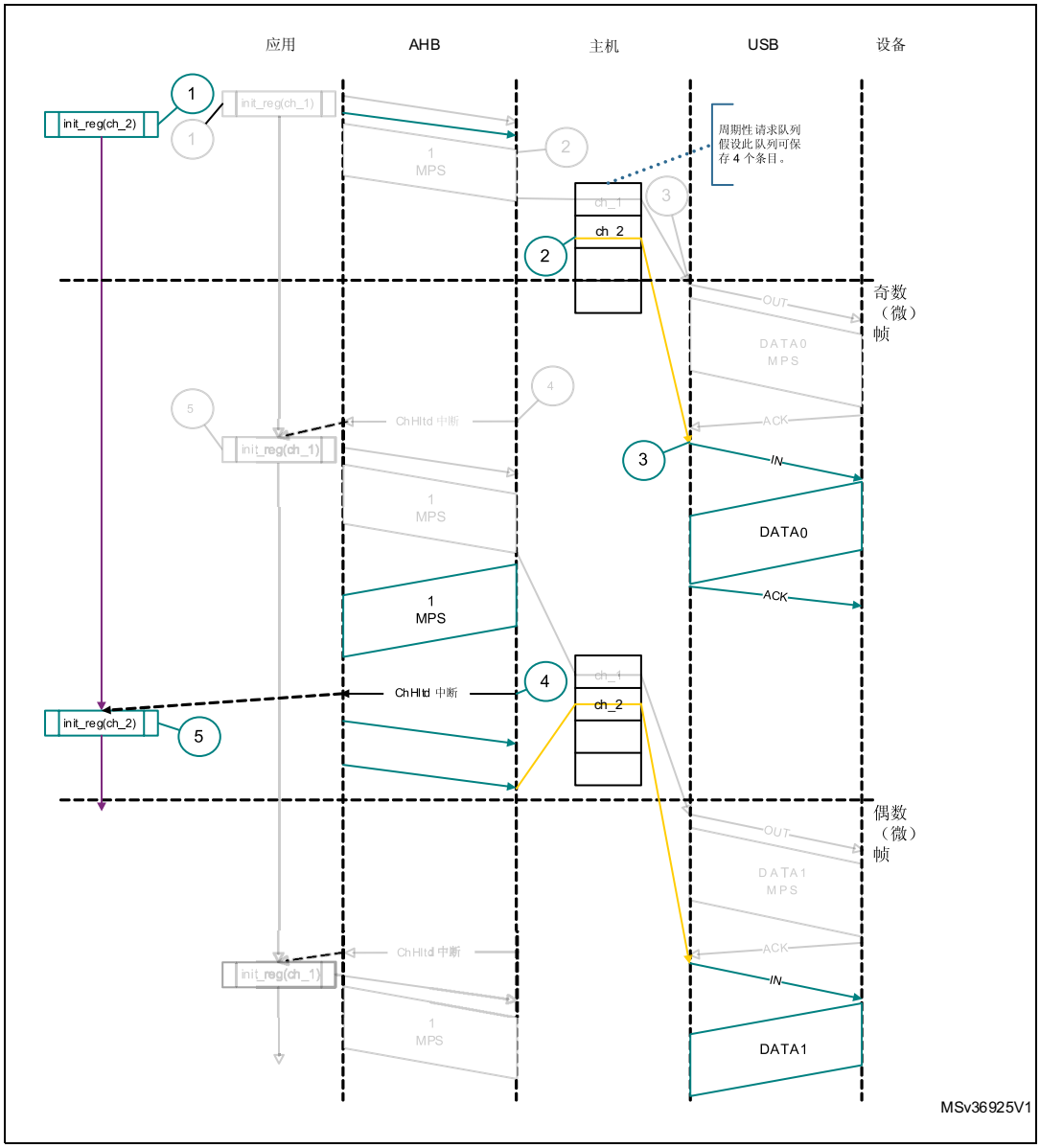


- **DMA 模式下的中断 IN 事务**

  通道 x 的操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 x。
2. 当通道获取了来自仲裁器的授权后（以循环方式执行仲裁），OTG 主机会立即向请求队列写入一个 IN 请求。在高带宽传输模式下，OTG 主机将执行连续写入操作，直到写入的数据包个数达到 MC。
3. OTG 主机尝试在下一个（奇数）帧/微帧的起始位置发送 IN 令牌。
4. 接收到 IN 数据包并写入接收 FIFO 后，OTG 主机便会立即生成 CHH 中断。
5. 为了响应 CHH 中断，将重新初始化通道以供下一传输操作使用。

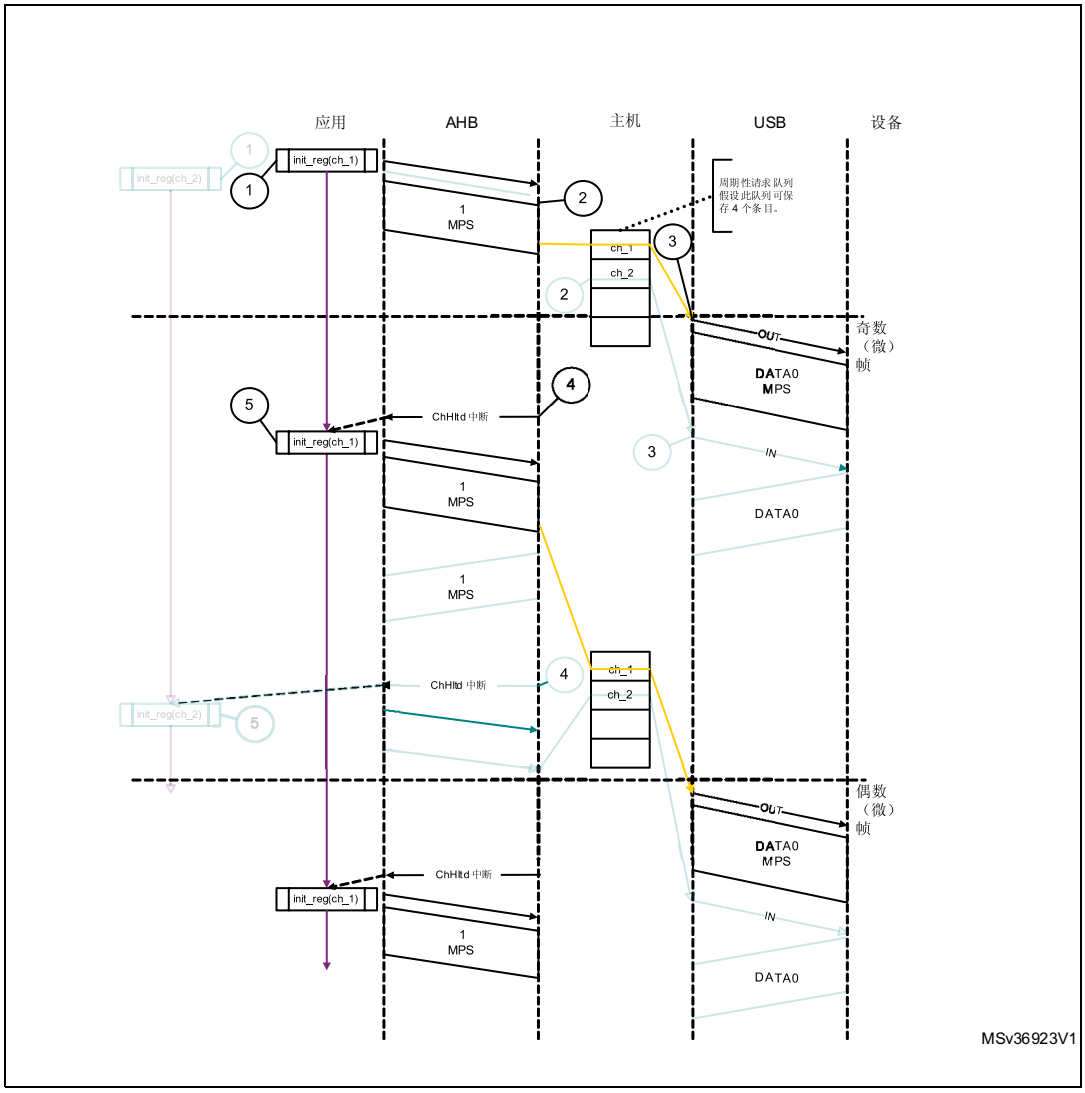
图 701. 正常中断 IN 事务 - DMA 模式



- **DMA 模式下的同步 OUT 事务**

1. 按照*第 2928 页的通道初始化*一节中的说明初始化并使能通道 x。
2. 使能通道后，OTG 主机开始获取第一个数据包，并在获取数据包的最后一个 32 位字后写入 OUT 请求。在高带宽传输模式下，OTG 主机会继续获取下一个数据包（直至数据包数量达到 MC 位字段中指定的数值），然后才能切换到下一个通道。
3. OTG 主机尝试在下一个（奇数）帧/微帧的起始位置发送 OUT 令牌。
4. 成功发送数据包后，OTG 主机将生成一个 CHH 中断。
5. 为了响应 CHH 中断，将重新初始化通道以供下一传输操作使用。

图 702. 正常同步 OUT 事务 - DMA 模式

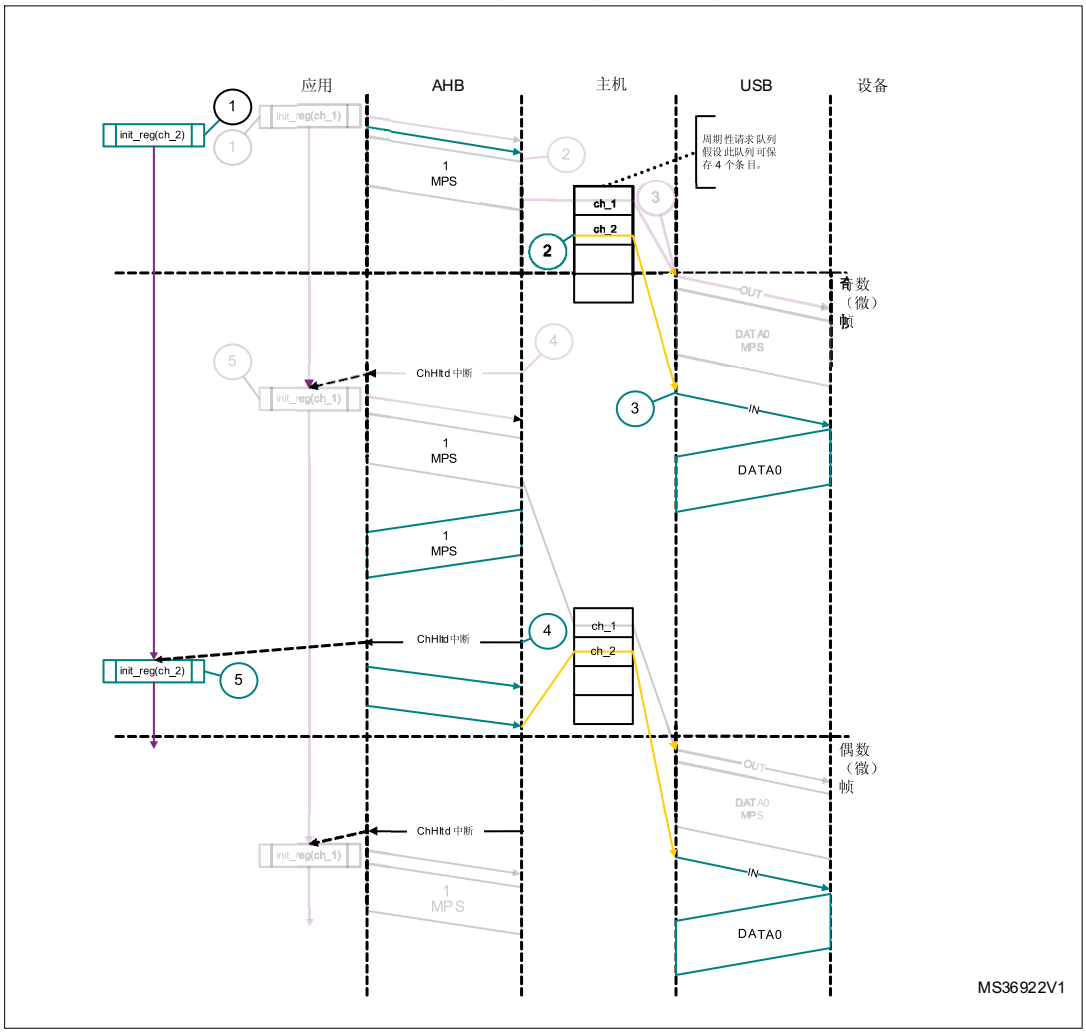


- **DMA 模式下的同步 IN 事务**

通道 x 的操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 x。
2. 当通道获取了来自仲裁器的授权后（以循环方式执行仲裁），OTG 主机会立即向请求队列写入一个 IN 请求。在高带宽传输模式下，OTG 主机将执行连续写入操作，直到写入的数据包个数达到 MC。
3. OTG 主机尝试在下一个（奇数）帧/微帧的起始位置发送 IN 令牌。
4. 接收到 IN 数据包并写入接收 FIFO 后，OTG 主机便会立即生成 CHH 中断。
5. 为了响应 CHH 中断，将重新初始化通道以供下一传输操作使用。

图 703. 正常同步 IN 事务 - DMA 模式



- **DMA 模式下的批量和控制 OUT/SETUP 分离事务**

（通道 x）中的操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 x 以用于启动分离。
2. 使能通道后，OTG 主机开始获取第一个数据包以发送，并在获取数据包的最后一个 32 位字后写入 OUT 请求。
3. 成功发送启动分离后，OTG 主机将生成 CHH 中断。
4. 作为对 CHH 中断的响应，将 OTG_HCSPLT1 中的 COMPLSPLT 位置 1，以发送完成分离。
5. 成功发送完成分离后，OTG 主机将生成 CHH 中断。
6. 为了响应 CHH 中断，将会释放通道。

- **DMA 模式下的批量/控制 IN 分离事务**

通道 x 的操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 x。
2. 获取了来自仲裁器的授权后，OTG 主机将会向非周期性请求队列写入启动分离请求。写入该请求后，OTG 主机将在内部屏蔽通道 x，不参与仲裁。
3. 发送 IN 令牌后，OTG 主机立即生成 CHH 中断。
4. 作为对 CHH 中断的响应，将 OTG_HCSPLT2 中的 COMPLSPLT 位置 1 并重新使能通道，以发送完成分离令牌。这将使能通道 x，并让它参与后续仲裁。
5. 接收到来自仲裁器的授权后，OTG 主机将会向非周期性请求队列写入完成分离请求。
6. 成功接收到数据包后，OTG 主机将开始向系统内存写入数据包。
7. 只要接收到的数据包写入系统内存，OTG 主机即会生成一个 CHH 中断。
8. 为了响应 CHH 中断，将会释放通道。

- **DMA 模式下的中断 OUT 分离事务**

（通道 x）中的操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 1 以用于启动分离。应用程序必须将 OTG_HCCHAR1 中的 ODDFRM 位置 1。
2. OTG 主机开始读取数据包。
3. OTG 主机尝试发送启动分离事务。
4. 成功发送启动分离后，OTG 主机将生成 CHH 中断。
5. 作为对 CHH 中断的响应，将 OTG_HCSPLT1 中的 COMPLSPLT 位置 1，以发送完成分离。
6. 成功结束完成分离事务后，OTG 主机将生成 CHH 中断。
7. 为了响应 CHH 中断，将会释放通道。

- **DMA 模式下的中断 IN 分离事务**

（通道 x）中的操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 x 以用于启动分离。
2. 当通道 x 接收到来自仲裁器的授权后，OTG 主机会立即向请求队列写入一个 IN 请求。
3. OTG 主机尝试在下一个奇数微帧的起始位置发送启动分离 IN 令牌。
4. 成功发送启动分离 IN 令牌后，OTG 主机将生成 CHH 中断。
5. 作为对 CHH 中断的响应，将 OTG_HCSPLT2 中的 COMPLSPLT 位置 1，以发送完成分离。
6. 成功接收到数据包后，OTG 主机即会开始向系统内存写入数据。
7. 将接收到的数据传输到系统内存后，OTG 主机会生成 CHH 中断。
8. 为了响应 CHH 中断，将释放或重新初始化通道以供下一启动分离使用。

- **DMA 模式下的同步 OUT 分离事务**

通道 x 的操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 x 以用于启动分离（开始）。应用程序必须将 OTG_HCCHAR1 中的 ODDFRM 位置 1。编程 MPS 位字段。
2. OTG 主机开始读取数据包。
3. 成功发送启动分离后（开始），OTG 主机将生成 CHH 中断。
4. 为了响应 CHH 中断，将重新初始化寄存器以发送启动分离（结束）。
5. 成功发送启动分离后（结束），OTG 主机将生成 CHH 中断。
6. 为了响应 CHH 中断，将会释放通道。

- **DMA 模式下的同步 IN 分离事务**

  通道 x 的操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 x 以用于启动分离。
2. 当通道 x 接收到来自仲裁器的授权后，OTG 主机会立即向请求队列写入一个 IN 请求。
3. OTG 主机尝试在下一个奇数微帧的起始位置发送启动分离 IN 令牌。
4. 成功发送启动分离 IN 令牌后，OTG 主机将生成 CHH 中断。
5. 作为对 CHH 中断的响应，将 OTG_HCSPLT2 中的 COMPLSPLT 位置 1，以发送完成分离。
6. 成功接收到数据包后，OTG 主机即会开始向系统内存写入数据。

   将接收到的数据传输到系统内存后，OTG 主机会生成 CHH 中断。为了响应 CHH 中断，将释放或重新初始化通道以供下一启动分离使用。

54.15.6 设备编程模型

USB 复位时的端点初始化

1. 为所有 OUT 端点将 NAK 位置 1
   - OTG_DOEPCTLx 中，SNAK = 1（对于所有 OUT 端点）
2. 使能以下中断位
   - OTG_DAINTMSK 中，INEP0 = 1（控制 0 IN 端点）
   - OTG_DAINTMSK 中，OUTEP0 = 1（控制 0 OUT 端点）
   - OTG_DOEPMSK 中的 STUPM = 1
   - OTG_DOEPMSK 中的 XFRCM = 1
   - OTG_DIEPMSK 中的 XFRCM = 1
   - OTG_DIEPMSK 中的 TOM = 1
3. 为每个 FIFO 设置数据 FIFO RAM
   - 对 OTG_GRXFSIZ 寄存器进行编程，以能够接收控制传输的 OUT 数据和设置数据。如果未使能阈值，则该寄存器必须至少等于控制端点 0 的 1 个最大数据包大小 + 2 个字（用于控制 OUT 数据包的状态）+ 10 个字（用于 SETUP 数据包）。
   - 对 OTG_DIEPTXF0 寄存器进行编程（取决于所选的 FIFO 编号），以能够发送控制 IN 数据。该寄存器至少必须等于控制端点 0 的 1 个最大数据包大小。
4. 对端点相关寄存器中的以下位字段进行编程，以使控制 OUT 端点 0 接收 SETUP 数据包
   - OTG_DOEPTSIZ0 中的 STUPCNT = 3（接收最多 3 个连续的 SETUP 数据包）
5. 在 DMA 模式下的 USB OTG，OTG_DOEPDMA0 寄存器应具有一个有效的内存地址，以存储接收到的任何 SETUP 数据包。

此时，接收 SETUP 数据包所需的所有初始化工作便已完成。

枚举完成时的端点初始化

1. 在枚举完成中断（OTG_GINTSTS 中的 ENUMDNE）中，读取 OTG_DSTS 寄存器以确定设备的枚举速度。
2. 对 OTG_DIEPCTL0 中的 MPSIZ 位字段进行编程以设置最大数据包大小。该步骤配置控制端点 0。控制端点的最大数据包大小取决于枚举速度。
3. 对于 DMA 模式下的 USB OTG，编程 OTG_DOEPCTL0 寄存器来使能控制 OUT 端点 0，以接收 SETUP 数据包。

此时，设备已准备好接收 SOF 数据包并配置为在控制端点 0 上执行控制传输。

收到 SetAddress 命令时的端点初始化

本节介绍了应用程序在 SETUP 数据包中接收到 SetAddress 命令时必须执行的操作。

1. 使用在 SetAddress 命令中接收到的设备地址来对 OTG_DCFG 寄存器进行编程
2. 对模块进行编程以发出状态阶段的 IN 数据包

收到 SetConfiguration/SetInterface 命令时的端点初始化

本节介绍了应用程序在 SETUP 包中接收 SetConfiguration 或 SetInterface 命令时必须执行的操作。

1. 接收到 SetConfiguration 命令时，应用程序必须对端点寄存器进行编程，以使用新配置中有效端点的特性来配置这些端点寄存器。
2. 接收到 SetInterface 命令时，应用程序必须对命令指定的端点的端点寄存器进行编程。
3. 在先前配置或其他设置中有效的端点在新的配置或其他设置中无效。必须停用这些无效端点。
4. 使用 OTG_DAINTMSK 寄存器使能有效端点的中断，屏蔽无效端点的中断。
5. 为每个 FIFO 设置数据 FIFO RAM。
6. 配置完所有必需的端点后，应用程序必须对模块进行编程以发送状态阶段的 IN 数据包。

此时，设备模块已可以接收和发送任何类型的数据包。

端点激活

本节介绍激活设备端点或者将现有设备端点配置为新类型所需的步骤。

1. 在 OTG_DIEPCTLx 寄存器（对于 IN 或双向端点）或 OTG_DOEPCTLx 寄存器（对于 OUT 或双向端点）的以下位字段中，对所需端点的特性进行编程。
   - 最大数据包大小
   - USB 活动端点位置 1
   - 端点初始数据同步位（对于中断和批量端点）
   - 端点类型
   - Tx FIFO 编号
2. 激活端点后，模块便开始解码发送到该端点的令牌，并在收到的令牌有效的情况下回复有效握手信号。

端点停用

本节介绍停用现有端点所需的步骤。

1. 在要停用的端点中，将 OTG_DIEPCTLx 寄存器（对于 IN 或双向端点）或 OTG_DOEPCTLx 寄存器（对于 OUT 或双向端点）中的 USB 活动端点位清零。
2. 停用端点后，模块便会忽略发送到该端点的令牌，从而导致 USB 超时。

注： *应用程序必须满足以下条件才能设置设备模块以处理通信：*
必须将 OTG_GINTMSK 寄存器中的 NPTXFEM 和 RXFLVLM 清零。

操作模型

SETUP 和 OUT 数据传输：

本节介绍了数据 OUT 传输和 SETUP 事务期间的内部数据流和应用程序操作步骤。

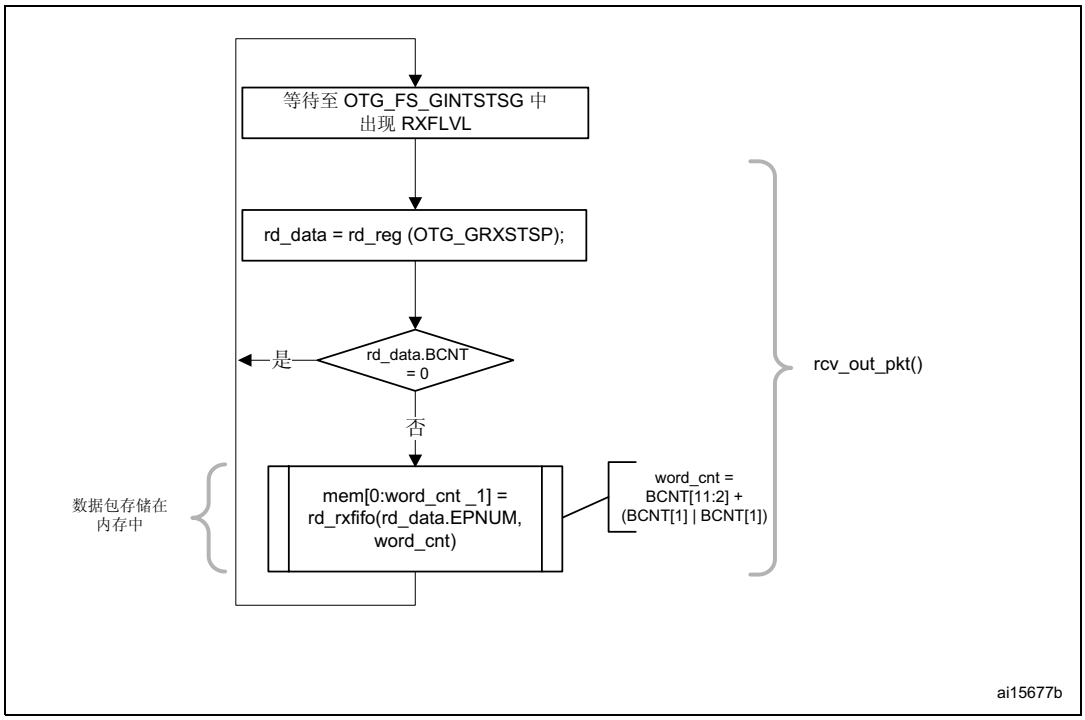
- **数据包读取**

本节介绍如何从接收 FIFO 读取数据包（OUT 数据和 SETUP 数据包）。

1. 捕获到 RXFLVL 中断（OTG_GINTSTS 寄存器）时，应用程序必须读取接收状态弹出寄存器 (OTG_GRXSTSP)。
2. 应用程序可以通过写入 RXFLVLM = 0（在 OTG_GINTMSK 中）来屏蔽 RXFLVL 中断（在 OTG_GINTSTS 中），直到它把数据包从接收 FIFO 中读取出来。
3. 如果已接收数据包的字节计数不是 0，则从接收数据 FIFO 中弹出数据的字节计数并存储在内存中。如果接收到的数据包字节计数为 0，则不会从接收数据 FIFO 中弹出任何数据。
4. 从接收 FIFO 读出的数据包状态有以下几种状态：
   a) 全局 OUT NAK：
   PKTSTS = 全局 OUT NAK，BCNT = 0x000，EPNUM = (0x0)，DPID = (0b00)。
   这些数据表示全局 OUT NAK 位已生效。
   b) SETUP 数据包：
   PKTSTS = SETUP，BCNT = 0x008，EPNUM = 控制端点编号，
   DPID = DATA0。这些数据表示指定端点上收到的 SETUP 数据包现在可从接收 FIFO 中读取。
   c) 建立阶段完成：
   PKTSTS = 建立阶段完成，BCNT = 0x0，EPNUM = 控制端点编号，
   DPID = (0b00)。
   这些数据表示指定端点的建立阶段完成并且数据阶段已启动。在此状态条目从接收 FIFO 中弹出后，模块将在指定控制 OUT 端点上产生建立中断。
   d) OUT 数据包：
   PKTSTS = DataOUT，BCNT = 接收的 OUT 数据包的大小 (0 ≤ BCNT ≤ 1 024)，
   EPNUM = 收到数据包的端点编号，DPID = 实际数据 PID。
   e) 数据传输完成：
   PKTSTS = OUT 数据传输完成，BCNT = 0x0，EPNUM = 完成数据传输的 OUT 端点编号，DPID = (0b00)。
   这些数据表示指定 OUT 端点的 OUT 数据传输完成。在此状态条目从接收 FIFO 中弹出后，模块将在指定的 OUT 端点上引发“传输完成”中断。
5. 从接收 FIFO 中弹出数据后，必须使能 RXFLVL 中断 (OTG_GINTSTS)。
6. 每次应用程序检测到 OTG_GINTSTS 中的 RXFLVL 中断时，都将重复步骤 1 到 5。读取空的接收 FIFO 可能导致未定义的模块行为。

图 704 提供了上述过程的流程图。

图 704. 接收 FIFO 数据包读取



SETUP 事务

本节介绍了模块处理 SETUP 数据包的方式以及应用程序处理 SETUP 事务的顺序。

- **应用程序要求**

1. 要接收 SETUP 数据包，必须将控制 OUT 端点中的 STUPCNT 位字段 (OTG_DOEPTSIZx) 编程为非零值。如果应用程序将 STUPCNT 字段编程为非零值，模块会接收 SETUP 数据包并将其写入接收 FIFO，而不考虑 NAK 状态和 OTG_FS_DOEPCTLx 中的 EPENA 位设置。控制端点每收到一个 SETUP 数据包后，STUPCNT 位字段都会递减。如果在接收 SETUP 数据包之前，未将 STUPCNT 位字段编程为适当值，模块仍能接收 SETUP 数据包并使 STUPCNT 位字段递减，但应用程序可能无法确定在控制传输的建立阶段中接收的 SETUP 数据包的正确数量。
   – 在 OTG_DOEPTSIZx 中，STUPCNT = 3
2. 应用程序必须始终在接收数据 FIFO 中分配一些额外空间，以便能够在控制端点上接收最多三个 SETUP 数据包。
   – 预留空间 10 个字。第一个 SETUP 数据包需要 3 个字，“建立阶段完成”状态双字需要 1 个字，还需要 6 个字以在所有控制端点间存储两个额外的 SETUP 数据包。
   – 每个 SETUP 数据包需要 3 个字以存储 8 个字节的 SETUP 数据和 4 个字节的 SETUP 状态（SETUP 数据包模式）。模块会在接收数据 FIFO 中保留这些空间，仅写入 SETUP 数据，而不会将此空间用于其他数据包。
3. 应用程序必须从接收 FIFO 中读取 SETUP 数据包的 2 个字。
4. 应用程序必须从接收 FIFO 中读取并丢弃“建立阶段完成”状态字。

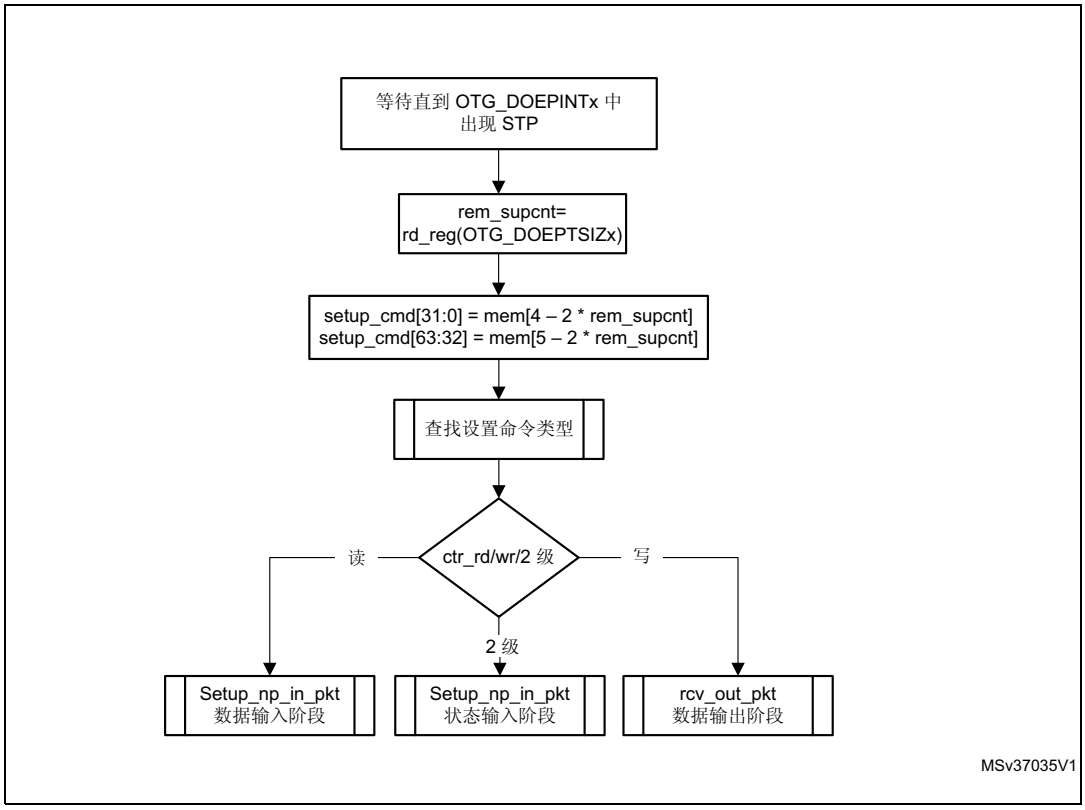
- **内部数据流**

1. 接收到 SETUP 数据包时，模块会将接收到的数据写入接收 FIFO，而不会检查接收 FIFO 中的可用空间，且不考虑端点的 NAK 和 STALL 位设置。
   – 模块会在内部将接收到 SETUP 数据包的控制 IN/OUT 端点的 IN NAK 和 OUT NAK 位置 1。
2. USB 上接收到的每个 SETUP 数据包，模块会将 3 个字的数据写入接收 FIFO，并且将 STUPCNT 位字段递减 1。
   – 第一个字包含由模块内部使用的控制信息
   – 第二个字包含 SETUP 命令的前 4 个字节
   – 第三个字包含 SETUP 命令的最后 4 个字节
3. 当建立阶段结束，数据 IN/OUT 阶段开始时，模块会将一个状态条目（“建立阶段完成”字）写入接收 FIFO，指示建立阶段完成。
4. 在 AHB 端，SETUP 数据包被应用程序读取。
5. 当应用程序从接收 FIFO 中弹出“建立阶段完成”字时，模块将使用 STUP 中断 (OTG_DOEPINTx) 来中断应用程序，表示其可以处理接收到的 SETUP 数据包。
6. 模块会将控制 OUT 端点的端点使能位清零。

- **应用程序编程顺序**

1. 对 OTG_DOEPTSIZx 寄存器进行编程。
   – STUPCNT = 3
2. 等待 RXFLVL 中断 (OTG_GINTSTS) 并且从接收 FIFO 中读取数据包。
3. STUP 中断的触发 (OTG_DOEPINTx) 表示 SETUP 数据传输成功完成。
   – 发生该中断时，应用程序必须读取 OTG_DOEPTSIZx 寄存器以确定接收的 SETUP 数据包数量并处理最后接收的 SETUP 数据包。

图 705. 处理 SETUP 数据包



- **处理三个以上连续的 SETUP 数据包**

根据 USB 2.0 规范，在 SETUP 数据包错误中，主机通常不会向同一个端点发送 3 个以上连续的 SETUP 数据包。但是，USB 2.0 规范并未限制主机可以向同一个端点发送的连续 SETUP 数据包数量。出现这种情况时，OTG 控制器将生成中断（OTG_DOEPINTx 中的 B2BSTUP）。

- **将全局 OUT NAK 置 1**

内部数据流：

1. 如果应用程序将全局 OUT NAK（OTG_DCTL 中的 SGONAK 位）置 1，模块将停止向接收 FIFO 中写入 SETUP 数据包以外的数据。无论接收 FIFO 中可用空间大小如何，设备都会对主机发送的非同步 OUT 令牌回复 NAK，而对同步 OUT 数据包直接予以忽略。
2. 模块将全局 OUT NAK 写入接收 FIFO。应用程序必须为此留出足够空间。
3. 当应用程序从接收 FIFO 中弹出全局 OUT NAK 字时，模块会将 GONAKEFF 中断 (OTG_GINTSTS) 置 1。
4. 应用程序检测到该中断后，会认为模块处于全局 OUT NAK 模式。应用程序可以通过将 OTG_DCTL 中的 SGONAK 位清零来清除该中断。

应用程序编程顺序：

1. 要停止接收任何类型的数据到接收 FIFO 中，应用程序必须通过编程以下位字段以将全局 OUT NAK 位置 1。
   – 在 OTG_DCTL 中，SGONAK = 1
2. 等待 OTG_GINTSTS 中的 GONAKEFF 中断触发。一旦被触发，该中断表示模块已停止接收 SETUP 数据包以外的任何类型数据。
3. 如果应用程序已将 OTG_DCTL 中的 SGONAK 位置 1，则在模块引发 GONAKEFF 中断 (OTG_GINTSTS) 之前，应用程序可以接收有效 OUT 数据包。
4. 应用程序可通过对 OTG_GINTMSK 寄存器中的 GONAKEFFM 位执行写操作来暂时屏蔽此中断。
   – 在 OTG_GINTMSK 寄存器中，GONAKEFFM = 0
5. 当应用程序准备退出全局 OUT NAK 模式时，必须将 OTG_DCTL 中的 SGONAK 位清零。此操作还会清除 GONAKEFF 中断 (OTG_GINTSTS)。
   – 在 OTG_DCTL 中，CGONAK = 1
6. 如果应用程序在之前已屏蔽此中断，则必须按以下方式使能该中断：
   – 在 OTG_GINTMSK 中，GONAKEFFM = 1

- **禁止 OUT 端点**

应用程序必须使用以下顺序禁止已使能的 OUT 端点。

应用程序编程顺序：

1. 禁止任何 OUT 端点前，应用程序必须在模块中使能全局 OUT NAK 模式。
   – 在 OTG_DCTL 中，SGONAK = 1
2. 等待 GONAKEFF 中断 (OTG_GINTSTS)
3. 通过编程以下位字段来禁止 OUT 端点：
   – 在 OTG_DOEPCTLx 中，EPDIS = 1
   – 在 OTG_DOEPCTLx 中，SNAK = 1
4. 等待 EPDISD 中断 (OTG_DOEPINTx)，该中断表示已完全禁止 OUT 端点。引发 EPDISD 中断时，模块还会将以下位清零：
   – 在 OTG_DOEPCTLx 中，EPDIS = 0
   – 在 OTG_DOEPCTLx 中，EPENA = 0
5. 应用程序必须将全局 OUT NAK 位清零，以开始从其他未禁止的 OUT 端点接收数据。
   – 在 OTG_DCTL 中，SGONAK = 0

- **OUT 端点的传输停止编程**

应用程序必须按照以下编程顺序来停止任何传输（由于是来自主机的中断，因此通常为复位）。

操作顺序：

1. 通过以下设置使能所有 OUT 端点
   – 所有 OTG_DOEPCTLx 寄存器中的 EPENA = 1。
2. 按如下方式刷新 RxFIFO
   – 轮询 OTG_GRSTCTL.AHBIDL 直到其为 1。这表明 AHB 主设备空闲。
   – 针对 OTG_GRSTCTL.RXFFLSH = 1 执行读修改写操作
   – 轮询 OTG_GRSTCTL.RXFFLSH 直到其为 0，但另外使用不到 10 毫秒的超时（对应于最小复位信号持续时间）。如果在超时结束之前变为 0，则 RxFIFO 刷新成功。如果发生超时时仍为 1（这可能是由于 EP0 上有来自主机的数据包），则返回（仅一次）到上一步（“执行读修改写操作”）。
3. 禁止任何 OUT 端点前，应用程序必须按照“*第 2962 页的将全局 OUT NAK 置 1*”中的说明在模块中使能全局 OUT NAK 模式。这样可确保 RxFIFO 中的数据成功发送到应用程序。将 OTG_DCTL 中的 SGONAK 置 1
4. 等待 GONAKEFF 中断 (OTG_GINTSTS)
5. 通过编程以下寄存器位来禁止所有活动 OUT 端点：
   – 在 OTG_DOEPCTLx 寄存器中，EPDIS = 1
   – 在 OTG_DOEPCTLx 寄存器中，SNAK = 1
6. 对于上一步中编程的每个 OUT 端点，等待 OTG_DOEPINTx 中的 EPDIS 中断。OTG_DOEPINTx 中的 EPDIS 中断表示已完全禁止相应的 OUT 端点。当 EPDIS 中断触发时，将清零以下位：
   – 在 OTG_DOEPCTLx 寄存器中，EPENA = 0
   – 在 OTG_DOEPCTLx 寄存器中，EPDIS = 0
   – 在 OTG_DOEPCTLx 寄存器中，SNAK = 0

- **通用非同步 OUT 数据传输**

本节介绍一种常规非同步 OUT 数据传输（控制、批量或中断）。

应用程序要求：

1. 建立 OUT 传输前，应用程序必须在内存中分配一个缓冲区，以容纳要作为 OUT 传输的一部分而接收的所有数据。
2. 对于 OUT 传输，端点的传输大小寄存器中的传输大小位字段必须是端点的最大数据包大小的倍数（且以字对齐）。
   – 传输大小[EPNUM] = n × (MPSIZ[EPNUM] + 4 – (MPSIZ[EPNUM] mod 4))
   – 数据包计数[EPNUM] = n
   – n > 0
3. 发生 OUT 端点中断时，应用程序必须读取端点的传输大小寄存器以计算内存中有效数据量。接收的有效数据量可能小于编程的传输大小。
   – 内存中的有效数据量 = 应用程序设置的初始传输量 – 模块更新后的剩余传输量
   – 接收到 USB 数据包数 = 应用程序设置的初始数据包数 – 模块更新后的剩余数据包数

内部数据流：

1. 应用程序必须在端点相关寄存器中设置传输大小和数据包计数位字段，将 NAK 位清零，并使能端点来接收数据。
2. NAK 位清零后，模块便开始接收数据并将数据写入接收 FIFO（只要接收 FIFO 中有空间）。对于 USB 上接收的每个数据包，数据包及其状态都会写入接收 FIFO。写入接收 FIFO 的每个数据包（数据量达到最大数据包大小的数据包或短数据包）都会使该端点的数据包计数位字段递减 1。
   - 收到的数据包若 CRC 无效，则自动被从接收 FIFO 中清除。
   - 在 USB 上为数据包回复 ACK 后，模块将丢弃主机因无法检测到 ACK 而重新发送的非同步 OUT 数据包。应用程序不会在具有相同数据 PID 的相同端点上检测到多个连续的 OUT 数据包。在这种情况下，数据包计数不会递减。
   - 如果接收 FIFO 中没有空间，则会忽略同步或非同步数据包并且不会将它们写入接收 FIFO。此外，非同步 OUT 令牌将会收到 NAK 握手应答。
   - 在上述所有三种情况中，数据包计数都不会递减，因为没有任何数据写入接收 FIFO。
3. 当数据包计数变为 0 或者在端点上接收到短数据包时，该端点的 NAK 位将置 1。NAK 位置 1 后，将忽略同步或非同步数据包并且不会将它们写入接收 FIFO，同时非同步 OUT 令牌会收到 NAK 握手应答。
4. 在数据写入接收 FIFO 后，应用程序将从接收 FIFO 中读取数据并将数据写入外部内存，一次一个数据包，逐个端点过来。
5. 在 AHB 上向外部内存写入完每个数据包后，端点的传输大小都会自动减去该数据包的大小。
6. 在以下情况时，OUT 端点的 OUT 数据传输完成状态将写入接收 FIFO：
   - 传输大小为 0 并且数据包计数为 0
   - 写入接收 FIFO 的最后一个 OUT 数据包是短数据包（0 ≤ 数据包大小 < 最大数据包大小）
7. 当应用程序弹出此状态条目（OUT 数据传输完成），并生成该端点的传输完成中断，同时清零端点使能位。

应用程序编程顺序：

1. 使用传输大小和相应数据包个数对 OTG_DOEPTSIZx 寄存器进行编程。
2. 使用端点特性对 OTG_DOEPCTLx 寄存器进行编程，并将 EPENA 和 CNAK 位置 1。
   - 在 OTG_DOEPCTLx 中，EPENA = 1
   - 在 OTG_DOEPCTLx 中，CNAK = 1
3. 等待 RXFLVL 中断（在 OTG_GINTSTS 中）并且从接收 FIFO 中读走数据包。
   - 此步骤可重复多次，具体取决于传输大小。
4. 触发 XFRC 中断 (OTG_DOEPINTx)，以表示非同步 OUT 数据传输成功完成。
5. 读取 OTG_DOEPTSIZx 寄存器，以确定有效数据量。

- **通用同步 OUT 数据传输**

本节介绍常规的同步 OUT 数据传输。

应用程序要求：

1. 非同步 OUT 数据传输的所有应用程序要求均适用于同步 OUT 数据传输。
2. 对于同步 OUT 数据传输中的传输大小和数据包计数位字段，必须始终将其设置为单个帧中可接收的最大数据包大小的数据包数目。同步类型的 OUT 数据传输事务必须在一个帧内完成。
3. 在周期性帧结束（OTG_GINTSTS 中的 EOPF 中断）之前，应用程序必须从接收 FIFO 中读取所有同步 OUT 数据包（数据条目和状态条目）。
4. 要接收下一帧中的数据，必须在 EOPF (OTG_GINTSTS) 之后 SOF (OTG_GINTSTS) 之前使能一个同步 OUT 端点。

内部数据流：

1. 同步 OUT 端点的内部数据流与非同步 OUT 端点的基本相同，但稍有差异。
2. 同步 OUT 端点通过将端点使能位置 1 并将 NAK 位清零来使能时，必须相应地将偶数/奇数帧位置 1。仅当符合以下条件时，模块才会在同步 OUT 端点上接收特定帧中的数据：
   – EONUM （在 OTG_DOEPCTLx 中）= FNSOF[0]（在 OTG_DSTS 中）
3. 当应用程序从接收 FIFO 中完整地读取一个同步 OUT 数据包（数据和状态）后，模块会根据从接收 FIFO 中读取的最后一个同步 OUT 数据包的数据 PID 更新 OTG_DOEPTSIZx 中的 RXDPID 位字段。

应用程序编程顺序：

1. 使用传输大小和相应数据包个数对 OTG_DOEPTSIZx 寄存器进行编程
2. 使用端点特性对 OTG_DOEPCTLx 寄存器进行编程，并将端点使能位、清除 NAK 位和奇数/偶数帧位置 1。
   – EPENA = 1
   – CNAK = 1
   – EONUM =（0：偶数/1：奇数）
3. 等待 RXFLVL 中断（在 OTG_GINTSTS 中）并且从接收 FIFO 中读走数据包
   – 此步骤可重复多次，具体取决于传输大小。
4. XFRC 中断（在 OTG_DOEPINTx 中）表示同步 OUT 数据传输完成。该中断不一定意味着内存中的数据是有效的。
5. 对于同步 OUT 传输，应用程序可能并不总会检测到该中断。而是可以检测到 OTG_GINTSTS 中的 INCOMPISOOUT 中断。
6. 读取 OTG_DOEPTSIZx 寄存器以确定接收的传输大小以及帧中所接收数据的有效性。仅当符合以下条件之一时，应用程序才必须将内存中接收的数据视为有效数据：
   – RXDPID = DATA0（在 OTG_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量 = 1
   – RXDPID = DATA1（在 OTG_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量 = 2

     RXDPID = D2（在 OTG_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量 = 3接收该有效数据的 USB 数据包数量 =
     应用程序编程的初始数据包个数 – 模块更新后的剩余数据包个数

   应用程序可将无效数据包丢弃。

- **未完成同步 OUT 数据传输**

本节介绍了同步 OUT 数据包出现丢包时应用程序编程顺序。

内部数据流：

1. 对于同步 OUT 端点，可能并不总是触发 XFRC 中断（在 OTG_DOEPINTx 中）。如果模块丢弃同步 OUT 数据包，则在以下情况下，应用程序可能无法检测到 XFRC 中断 (OTG_DOEPINTx)：
   – 在接收 FIFO 无法容纳完整的 ISO OUT 数据包时，模块将丢弃接收到的 ISO OUT 数据
   – 接收到的同步 OUT 数据包存在 CRC 错误
   – 模块接收到的同步 OUT 令牌损坏
   – 应用程序从接收 FIFO 中读取数据的速度非常缓慢
2. 如果模块在所有同步 OUT 端点的传输完成前检测到周期性帧结束，将触发未完成同步 OUT 数据中断（OTG_GINTSTS 中的 INCOMPISOOUT），指示至少有一个同步 OUT 端点上未触发 XFRC 中断（在 OTG_DOEPINTx 中）。此时，未完成传输的端点仍保持使能，但在 USB 的该端点上，没有进行中的有效传输。

应用程序编程顺序：

1. 硬件触发 INCOMPISOOUT 中断 (OTG_GINTSTS) 表示当前帧中至少有一个同步 OUT 端点具有未完成的传输。
2. 如果因未从端点完全读取同步 OUT 数据而发生这种情况，应用程序必须确保首先从接收 FIFO 读取走所有同步 OUT 数据（包括数据条目和状态条目），然后再继续处理。
   – 从接收 FIFO 读取所有数据后，应用程序即可检测到 XFRC 中断 (OTG_DOEPINTx)。在此情况下，应用程序必须重新使能端点以接收下一个帧中的同步 OUT 数据。
3. 当应用程序接收到 INCOMPISOOUT 中断（在 OTG_GINTSTS 中）时，应用程序必须读取所有同步 OUT 端点的控制寄存器 (OTG_DOEPCTLx)，以确定哪些端点在当前微帧中具有不完整的传输。同时满足以下两个条件时，表示端点传输未完成：
   – EONUM 位（在 OTG_DOEPCTLx 中）= FNSOF[0]（在 OTG_DSTS 中）
   – EPENA = 1（在 OTG_DOEPCTLx 中）
4. 在检测到 SOF 中断（在 OTG_GINTSTS 中）前，必须执行完成上一步操作，以确保当前帧编号未发生更改。
5. 对于具有未完成传输的同步 OUT 端点，应用程序必须丢弃内存中的数据，并通过将 OTG_DOEPCTLx 中的 EPDIS 位置 1 来禁止该端点。
6. 等待 EPDISD 中断（在 OTG_DOEPINTx 中），并使能该端点，以便接收下一个帧中的新数据。
   – 由于模块可能需要一些时间才能禁止端点，因此应用程序在接收到无效同步数据后，可能无法接收下一个帧中的数据。

- **停止非同步 OUT 端点**

本节介绍应用程序如何才能停止非同步端点。

1. 将模块置于全局 OUT NAK 模式。
2. 禁止所需的端点
   – 禁止端点时，请设置 STALL = 1（在 OTG_DOEPCTL 中），而不是将 OTG_DOEPCTL 中的 SNAK 位置 1。
   STALL 位的优先级始终高于 NAK 位。
3. 当应用程序不再需要端点回复 STALL 握手信号时，必须将 STALL 位（在 OTG_DOEPCTLx 中）清零。
4. 如果应用程序由于收到主机的 SetFeature.Endpoint Halt 或 ClearFeature.Endpoint Halt 命令来设置或清除端点的 STALL 状态，则必须在应用程序建立该控制端点上的状态阶段传输前，将 STALL 位置 1 或清零。

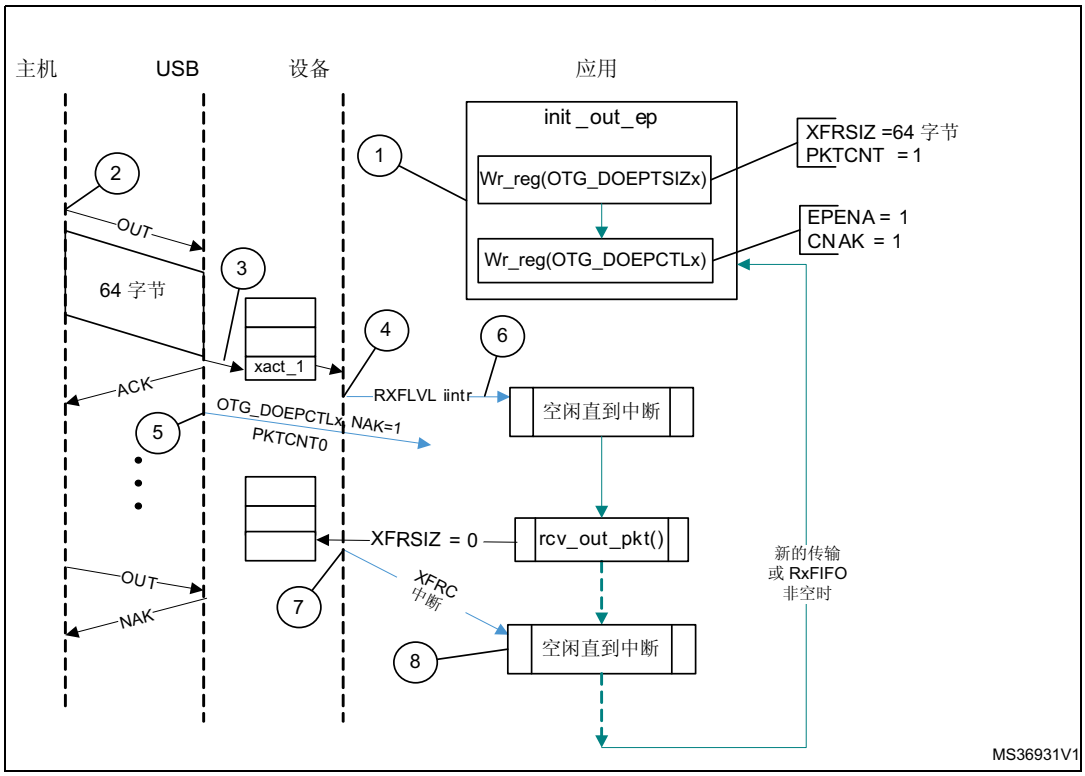
示例

本节介绍并描述了一些基本的传输类型和情景。

- 批量 OUT 事务

图 706 描述了将单个批量 OUT 数据包从 USB 接收到 AHB 中的过程，并介绍了这一过程中涉及到的事件。

图 706. 批量 OUT 事务



接收到 SetConfiguration/SetInterface 命令后，应用程序将初始化所有 OUT 端点：将 CNAK 和 EPENA 置 1（在 OTG_DOEPCTLx 中），并将 OTG_DOEPTSIZx 寄存器中的 XFRSIZ 和 PKTCNT 设置成合适的值。

1. 主机尝试将数据（OUT 令牌）发送到端点。
2. 当模块在 USB 上接收到 OUT 令牌时，模块会将数据包存储在 Rx FIFO 中，因为其中有可用空间。
3. 在 Rx FIFO 中写入完整数据包后，模块随后会触发 RXFLVL 中断（在 OTG_GINTSTS 中）。
4. 接收到 PKTCNT 所指定数量的 USB 数据包后，模块在内部将该端点的 NAK 位置 1，以避免其再接收任何数据包。
5. 应用程序处理中断并从 Rx FIFO 中读取数据。
6. 应用程序读取完所有数据后（相当于 XFRSIZ），模块将生成 XFRC 中断（在 OTG_DOEPINTx 中）。
7. 应用程序处理中断并通过 XFRC 中断位（在 OTG_DOEPINTx 中）的触发得知本次传输完成。

IN 数据传输

- **数据包写入**

本节介绍在已使能专用发送 FIFO 的情况下应用程序如何将数据包写入端点 FIFO。

1. 应用程序可以选择轮询模式或中断模式。
   - 在轮询模式下，应用程序通过读取 OTG_DTXFSTSx 寄存器来监视端点发送数据 FIFO 的状态，从而确定该数据 FIFO 中是否有足够空间。
   - 在中断模式中，应用程序等待 TXFE 中断（在 OTG_DIEPINTx 中），然后读取 OTG_DTXFSTSx 寄存器，以确定数据 FIFO 中是否有足够空间。
   - 要写入单个非零长度的数据包，数据 FIFO 中必须有足够的空间来容纳整个数据包。
   - 要写入零长度的数据包，应用程序不能查看 FIFO 空间。
2. 如果使用上述方法之一，当应用程序确定有足够空间来写入发送数据包时，应用程序必须首先对端点控制寄存器进行相应写操作，然后再将数据写入数据 FIFO。通常，应用程序必须对 OTG_HS_DIEPCTLx 寄存器执行读-修改-写操作，以避免在将端点使能位置 1 的同时，修改寄存器中的其它内容。

如果有足够空间，应用程序可将同一端点的多个数据包写入发送 FIFO。对于周期性 IN 端点，应用程序只能一次写入一个微帧内的多个数据包。只有先前一个微帧的通信事务传输完成之后，应用程序才会写入下一个微帧内要发送的所有数据包。

- **将 IN 端点 NAK 置 1**

内部数据流：

1. 当应用程序将特定端点的 IN NAK 置 1 时，模块将停止端点上的数据发送，而不考虑端点发送 FIFO 中的数据是否可用。
2. 非同步端点收到 IN 令牌，回复 NAK 握手应答
   - 同步端点收到 IN 令牌，返回零长度数据包
3. 模块在 OTG_DIEPINTx 中触发 INEPNE（IN 端点 NAK 有效）中断以响应 OTG_DIEPCTLx 中的 SNAK 位。
4. 应用程序检测到该中断后，便会认为端点处于 IN NAK 模式。应用程序可以通过将 OTG_DIEPCTLx 中的 CNAK 位置 1 来清除该中断。

应用程序编程顺序：

1. 要在特定 IN 端点上停止发送任何数据，应用程序必须将 IN NAK 位置 1。要将该位置 1，必须编程以下位字段。
   – 在 OTG_DIEPCTLx 中，SNAK = 1
2. 等待 OTG_DIEPINTx 中的 INEPNE 中断触发。该中断表示模块已在端点上停止发送数据。
3. 在应用程序将 NAK 位置 1 但“NAK 有效”中断尚未触发时，模块可以在端点上发送有效 IN 数据。
4. 应用程序可通过写入 OTG_DIEPMSK 中的 INEPNEM 位来临时屏蔽该中断。
   – 在 OTG_DIEPMSK 中，INEPNEM = 0
5. 要退出端点 NAK 模式，应用程序必须将 OTG_DIEPCTLx 中的 NAK 状态位 (NAKSTS) 清零。此操作还会将 INEPNE 中断位（在 OTG_DIEPINTx 中）清零。
   – 在 OTG_DIEPCTLx 中，CNAK = 1
6. 如果应用程序已将该中断屏蔽，则必须按以下方式使能：
   – 在 OTG_DIEPMSK 中，INEPNEM = 1

- **禁止 IN 端点**

使用以下顺序来禁止先前已使能的特定 IN 端点。

应用程序编程顺序：

1. 应用程序必须先停止在 AHB 上写入数据，之后才能禁止 IN 端点。
2. 应用程序必须将端点设置为 NAK 模式。
   – 在 OTG_DIEPCTLx 中，SNAK = 1
3. 等待 OTG_DIEPINTx 中的 INEPNE 中断。
4. 将必须禁止的端点的 OTG_DIEPCTLx 寄存器中的以下位置 1。
   – 在 OTG_DIEPCTLx 中，EPDIS = 1
   – 在 OTG_DIEPCTLx 中，SNAK = 1
5. OTG_DIEPINTx 中的 EPDISD 中断的触发表示模块已完全禁止指定的端点。在触发中断的同时，模块还会将以下位清零：
   – 在 OTG_DIEPCTLx 中，EPENA = 0
   – 在 OTG_DIEPCTLx 中，EPDIS = 0
6. 应用程序必须读取周期性 IN EP 的 OTG_DIEPTSIZx 寄存器，以计算该端点在 USB 上发送的数据量。
7. 应用程序必须通过将 OTG_GRSTCTL 寄存器中的以下位字段置 1，来清空端点发送 FIFO 中的数据：
   – TXFNUM（在 OTG_GRSTCTL 中）= 端点发送 FIFO 编号
   – TXFFLSH（在 OTG_GRSTCTL 中）= 1

应用程序必须轮询 OTG_GRSTCTL 寄存器，直至模块将 TXFFLSH 位清零，这表示 FIFO 清空操作结束。要在该端点上发送新数据，应用程序可以稍后重新使能该端点。

- **IN 端点的传输停止编程**

应用程序必须按照以下编程顺序来停止任何传输（由于是来自主机的中断，因此通常为复位）。

操作顺序：

1. 通过以下设置禁止 IN 端点：
   – 所有 OTG_DIEPCTLx 寄存器中的 EPDIS = 1
2. 等待 OTG_DIEPINTx 中的 EPDIS 中断，该中断表示已完全禁止 IN 端点。当 EPDIS 中断触发时，将清零以下位：
   – 在 OTG_DIEPCTLx 中，EPDIS = 0
   – 在 OTG_DIEPCTLx 中，EPENA = 0
3. 通过编程以下位来刷新 TxFIFO：
   – 在 OTG_GRSTCTL 中，TXFFLSH = 1
   – TXFNUM = OTG_GRSTCTL 中的“特定于端点的 FIFO 编号”
4. 应用程序可以开始轮询，直到 OTG_GRSTCTL 中的 TXFFLSH 清零。当该位清零时，可确保 Tx FIFO 中没有剩余数据。

- **通用非周期性 IN 数据传输**

应用程序要求：

1. 建立 IN 传输前，应用程序必须确保组成一次 IN 传输的每个数据包都可以容纳在单个缓冲区中。
2. 对于 IN 传输，端点传输大小寄存器中的传输大小位字段表示本次传输的有效数据量，它由多个最大数据包大小的数据包和单个短数据包组成。该短数据包在传输结束时发送。
   – 要发送多个最大数据包大小的数据包并在传输结束时外加一个短数据包：
     传输大小[EPNUM] = *x* × MPSIZ[EPNUM] + sp
     如果 (sp > 0)，数据包计数[EPNUM] = x + 1。
     否则，数据包计数[EPNUM] = *x*
   – 要发送单个零长度数据包：
     传输大小[EPNUM] = 0
     数据包计数[EPNUM] = 1
   – 要发送多个最大数据包大小的数据包并在传输结束时外加一个零长度数据包，应用程序必须将传输拆分为两个部分。第一部分发送最大数据包大小的数据包，第二部分仅发送零长度数据包。
     第一次传输：传输大小[EPNUM] = *x* × MPSIZ[epnum]；数据包计数 = *n*；
     第二次传输：传输大小[EPNUM] = 0；数据包计数 = 1；
3. 使能某个端点进行数据传输后，模块会更新传输大小寄存器。在 IN 传输结束时，应用程序必须读取传输大小寄存器，以确定送入发送 FIFO 中的数据已有多少通过 USB 发送出去。
4. 送入发送 FIFO 中的数据量 = 应用程序编程的初始传输大小 - 模块更新后的最终传输大小
   – 已经通过 USB 发送的数据量 = (应用程序编程的初始数据包个数 – 模块更新后的剩余数据包个数) × MPSIZ[EPNUM]
   – 要通过 USB 发送的剩余数据量 = (应用程序编程的初始传输大小 – 已通过 USB 发送的数据量)

内部数据流：

1. 应用程序必须在特定端点的寄存器中设置传输大小和数据包计数位字段，并使能该端点来发送数据。
2. 应用程序还必须向该端点的发送 FIFO 写入必需的数据。
3. 应用程序每向发送 FIFO 写入一个数据包，该端点的传输大小便会自动减去该数据包大小。应用程序持续从内存获取数据来写入发送 FIFO，直到该端点的传输大小变为 0。向 FIFO 写入数据后，“FIFO 中的数据包数”计数会递增（这是一个 3 位计数，由模块在内部进行维护，每个 IN 端点发送 FIFO 对应一个。在 IN 端点 FIFO 中，模块所维护的最大数据包数始终为八个）。对于零长度数据包，每个 FIFO 均另有一个单独的标志，FIFO 中没有任何数据。
4. 当数据写入发送 FIFO 后，模块会在接收到 IN 令牌时将这些数据送出。每个非同步 IN 数据包发送出去并收到回复的 ACK 握手信号后，该端点的数据包计数都会递减 1，直到数据包计数变 0 为止。发生超时时，数据包计数不会递减。
5. 对于零长度数据包（由内部零长度标志指示），模块会针对 IN 令牌发出一个零长度数据包，并递减数据包计数位字段的值。
6. 如果接收到 IN 令牌的端点对应的 FIFO 中无数据，且该端点的数据包计数位字段为零，则模块会针对该端点生成一个“Tx FIFO 为空时接收到 IN 令牌”(ITTXFE) 中断（前提是该端点的 NAK 位未置 1）。模块在该非同步端点上回复 NAK 握手信号。
7. 模块会在内部使 FIFO 指针重新返回到开头，并且不会生成超时中断。
8. 当传输大小为 0 且数据包计数为 0 时，将生成该端点的传输完成 (XFRC) 中断，同时将端点使能清零。

应用程序编程顺序：

1. 使用传输大小和相应数据包计数对 OTG_DIEPTSIZx 寄存器进行编程。
2. 使用端点特性对 OTG_DIEPCTLx 寄存器进行编程，并将 CNAK 和 EPENA（端点使能）位置 1。
3. 发送非零长度数据包时，应用程序必须轮询 OTG_DTXFSTSx 寄存器（其中 x 为与该端点相关联的 FIFO 编号）以确定数据 FIFO 中是否有足够的空间。写入数据前，应用程序可以选择使用 TXFE（在 OTG_DIEPINTx 中）。

- **通用周期性 IN 数据传输**

本节介绍典型的周期性 IN 数据传输。

应用程序要求：

1. *第 2971 页的通用非周期性 IN 数据传输*的应用程序要求 1、2、3 和 4 对周期性 IN 数据传输同样适用（只是对要求 2 稍加修改）。
   – 应用程序只能发送若干个最大数据包大小的数据包或若干个最大数据包大小的包，外加传输结束时的一个短数据包。要发送多个最大数据包大小的数据包并在传输结束时外加一个短数据包，必须满足以下条件：

     传输大小[EPNUM] = x × MPSIZ[EPNUM] + sp
     （其中 *x* 是 ≥ 0 的整数，且 0 ≤ sp < MPSIZ[EPNUM]）

     如果 (sp > 0)，数据包计数[EPNUM] = *x* + 1
     否则，数据包计数[EPNUM] = *x*；

     MCNT[EPNUM] = 数据包计数[EPNUM]

– 应用程序无法在传输结束时发送零长度数据包。应用程序可以单独发送一个零长度数据包。要发送单个零长度数据包：
– 传输大小[EPNUM] = 0
数据包计数[EPNUM] = 1
MCNT[EPNUM] = 数据包计数[EPNUM]

2. 应用程序一次只能安排一帧的数据传输。
   – (MCNT – 1) × MPSIZ ≤ XFERSIZ ≤ MCNT × MPSIZ
   – PKTCNT = MCNT（在 OTG_DIEPTSIZx 中）
   – 如果 XFERSIZ < MCNT × MPSIZ，则传输的最后一个数据包为短数据包。
   – 请注意：MCNT 在 OTG_DIEPTSIZx 中，MPSIZ 在 OTG_DIEPCTLx 中，PKTCNT 在 OTG_DIEPTSIZx 中，而 XFERSIZ 在 OTG_DIEPTSIZx 中
3. 接收到 IN 令牌前，应用程序必须将要在帧中发送的完整数据写入到发送 FIFO 中。在接收到 IN 令牌时，即使发送 FIFO 中该帧要发送的数据只差 1 个双字未写进来，模块也会执行 FIFO 为空时的操作。当发送 FIFO 为空时：
   – 同步端点上将回复零长度数据包
   – 中断端点上将回复 NAK 握手信号

内部数据流：

1. 应用程序必须在特定端点的寄存器中设置传输大小和数据包计数位字段，并使能该端点来发送数据。
2. 应用程序还必须向与该端点相关联的发送 FIFO 写入必需的数据。
3. 应用程序每向发送 FIFO 写入一个数据包，该端点的传输大小便会自动减去该数据包大小。应用程序持续从内存获取数据来写入发送 FIFO，直到该端点的传输大小变为 0。
4. 当周期性端点接收到 IN 令牌时，模块将开始发送 FIFO 中的数据（如果 FIFO 中有数据）。如果 FIFO 中没有该帧要发送的数据的完整数据包，则模块将为该端点生成一个“Tx FIFO 为空时接收到 IN 令牌”中断。
   – 同步端点上将回复零长度数据包
   – 中断端点上将回复 NAK 握手信号
5. 端点的数据包计数会在下列情况下递减 1：
   – 对于同步端点，发送一个零长度或非零长度的数据包时
   – 对于中断端点，在发送 ACK 握手信号时递减
   – 当传输大小和数据包计数均为 0 时，将生成该端点的传输完成中断，同时将端点使能位清零。
6. 在“周期性帧间隔”（由 OTG_DCFG 中的 PFIVL 位控制）内，当模块发现任何在当前帧内应为空的同步 IN 端点 FIFO 中的数据还未发送完成时，都会在 OTG_GINTSTS 中生成一个 IISOIXFR 中断。

应用程序编程顺序：

1. 使用端点特性对 OTG_DIEPCTLx 寄存器进行编程，并将 CNAK 和 EPENA 位置 1。
2. 将需要在下一帧中发送的数据写入发送 FIFO。
3. 硬件触发 ITTXFE 中断（在 OTG_DIEPINTx 中）表示应用程序尚未将需要发送的全部数据写入发送 FIFO。
4. 如果在检测到中断前已使能中断端点，则将忽略该中断。如果中断端点未使能，则使能此端点，以便数据能够在收到下一次 IN 令牌时发送出去。

5. 硬件触发 XFRC 中断（在 OTG_DIEPINTx 中）时如果 OTG_DIEPINTx 中未产生 ITTXFE 中断，则表示成功完成同步 IN 传输。读取 OTG_DIEPTSIZx 寄存器时始终得到传输大小 = 0 且数据包计数 = 0，表示所有数据都已通过 USB 发送完毕。
6. 无论是否产生 ITTXFE 中断（在 OTG_DIEPINTx 中），只要触发 XFRC 中断（在 OTG_DIEPINTx 中），即表示中断 IN 传输成功完成。读取 OTG_DIEPTSIZx 寄存器时始终得到传输大小 = 0 且数据包计数 = 0，表示所有数据都已通过 USB 发送完毕。
7. 如果在未产生任何前述中断的情况下在 OTG_GINTSTS 中触发未完成同步 IN 传输 (IISOIXFR) 中断，则表示模块在当前帧中至少未收到 1 个周期性 IN 令牌。

- **未完成同步 IN 数据传输**

本节介绍应用程序针对未完成同步 IN 数据传输必须执行的操作。

内部数据流：

1. 符合下列条件之一时，即认为同步 IN 传输未完成：
   a) 模块在至少一个同步 IN 端点上接收到损坏的同步 IN 令牌。此时，应用程序检测到未完成同步 IN 传输中断（OTG_GINTSTS 中的 IISOIXFR 位）。
   b) 应用程序向发送 FIFO 写入数据的速度过慢，在将完整数据写入 FIFO 之前便接收到 IN 令牌。此时，应用程序在 OTG_DIEPINTx 中检测到“Tx FIFO 为空时接收到 IN 令牌”中断。应用程序可忽略此中断，因为最终将在周期性帧结束时产生一个未完成同步 IN 传输中断（OTG_GINTSTS 中的 IISOIXFR 位）。

   模块会通过 USB 发送一个零长度数据包来响应接收到的 IN 令牌。
2. 应用程序必须尽快停止向发送 FIFO 写入数据。
3. 应用程序必须将端点的 NAK 位和禁止位置 1。
4. 模块会禁止该端点，将禁止位清零并触发端点的“端点禁止”中断。

应用程序编程顺序：

1. 应用程序可以在任何同步 IN 端点上忽略 OTG_DIEPINTx 中的“Tx FIFO 为空时接收到 IN 令牌”中断，因为最终这将产生一个未完成同步 IN 传输中断（在 OTG_GINTSTS 中）。
2. 硬件触发未完成同步 IN 传输中断（在 OTG_GINTSTS 中）表示在至少一个同步 IN 端点上存在未完成的同步 IN 传输。
3. 应用程序必须读取所有同步 IN 端点的“端点控制”寄存器来检测存在未完成 IN 数据传输的端点。
4. 应用程序必须停止向与这些端点相关联的“周期性发送 FIFO”写入数据。
5. 对 OTG_DIEPCTLx 寄存器中的下列位字段进行编程以禁止端点：
   – 在 OTG_DIEPCTLx 中，SNAK = 1
   – 在 OTG_DIEPCTLx 中，EPDIS = 1
6. 硬件触发 OTG_DIEPINTx 中的“端点禁止”中断表示模块已禁止该端点。
   – 此时，应用程序必须清空相关联的发送 FIFO 中的数据，或者通过在下一微帧中使能新传输的端点来覆盖 FIFO 中的现有数据。要刷新数据，应用程序必须使用 OTG_GRSTCTL 寄存器。

- **停止非同步 IN 端点**

本节介绍应用程序如何才能停止非同步端点。

应用程序编程顺序：

1. 禁止要停止的 IN 端点。同时将 STALL 位置 1。
2. OTG_DIEPCTLx 中的 EPDIS = 1（当端点已使能时）
   – OTG_DIEPCTLx 中的 STALL = 1
   – STALL 位的优先级始终高于 NAK 位
3. 硬件触发“端点禁止”中断（在 OTG_DIEPINTx 中），应用程序获知模块已禁止指定的端点。
4. 应用程序必须根据端点类型清空非周期性或周期性发送 FIFO。对于非周期性端点，应用程序必须重新使能另一个无需停止的非周期性端点来发送数据。
5. 当应用程序准备好结束该端点的 STALL 握手信号时，必须将 OTG_DIEPCTLx 的 STALL 位清零。
6. 如果应用程序因收到来自主机的 SetFeature.Endpoint Halt 命令或 ClearFeature.Endpoint Halt 命令来设置或清除端点的 STALL 状态，则必须在该控制端点的状态阶段传输之前将 STALL 位置 1 或清零。

特例：停止控制 OUT 端点

如果在控制传输的数据阶段，主机发送的 IN/OUT 令牌数超过 SETUP 数据包指定的值，则模块必须对这些多余的 IN/OUT 令牌回复 STALL。在这种情况下，在控制传输的数据阶段，当模块传输完 SETUP 数据包指定的数据量后，应用程序必须使能 OTG_DIEPINTx 中的 ITTXFE 中断和 OTG_DOEPINTx 中的 OTEPDIS 中断。随后，当应用程序收到此中断时，必须将相应端点控制寄存器中的 STALL 位置 1 并清除此中断。

54.15.7 最坏情况下的响应时间

当 OTG 控制器作为设备使用时，对于任何跟随在同步 OUT 之后的令牌，都存在一个最坏响应时间。这个最坏情况响应时间随 AHB 时钟频率的不同而异。

模块寄存器位于 AHB 域内，在更新这些寄存器值之前，模块不会接受新令牌。对于任何跟随在同步 OUT 之后的令牌，最坏的情况是：由于同步事务不需要握手信号，所以下一个令牌可能很快就到达。当 AHB 与 PHY 时钟频率相同时，这个最坏情况值为 7 个 PHY 时钟。AHB 时钟越快，此值越小。

如果发生这种最坏情况，模块将以 NAK 响应批量/中断令牌，并丢弃同步和 SETUP 令牌。对于 SETUP，主机会将这种情况视为超时，并尝试重新发送 SETUP 数据包。对于同步传输，会产生未完成同步 IN 传输中断 (IISOIXFR) 和未完成同步 OUT 传输中断 (IISOOXFR)，来通知应用程序同步 IN/OUT 数据包被丢弃。

选择 OTG_GUSBCFG 中的 TRDT 的值

TRDT 的值 (OTG_GUSBCFG) 是指在接收到 IN 令牌后以 PHY 时钟计算的时间长度，MAC 将在这段时间内获取 FIFO 状态并从 PFC 模块获取第一个数据。这段时间包含 PHY 和 AHB 时钟之间的同步延迟。最坏情况延迟是当 AHB 时钟与 PHY 时钟相等时的延迟。这种情况下的延迟为 5 个时钟周期。

MAC 接收到 IN 令牌后，此信息（接收到的令牌）将由 PFC（PFC 以 AHB 时钟运行）同步到 AHB 时钟。随后，PFC 从 SPRAM 中读取数据并将它们写入双时钟源缓冲区。MAC 再从源缓冲区读出数据（4 个深度）。

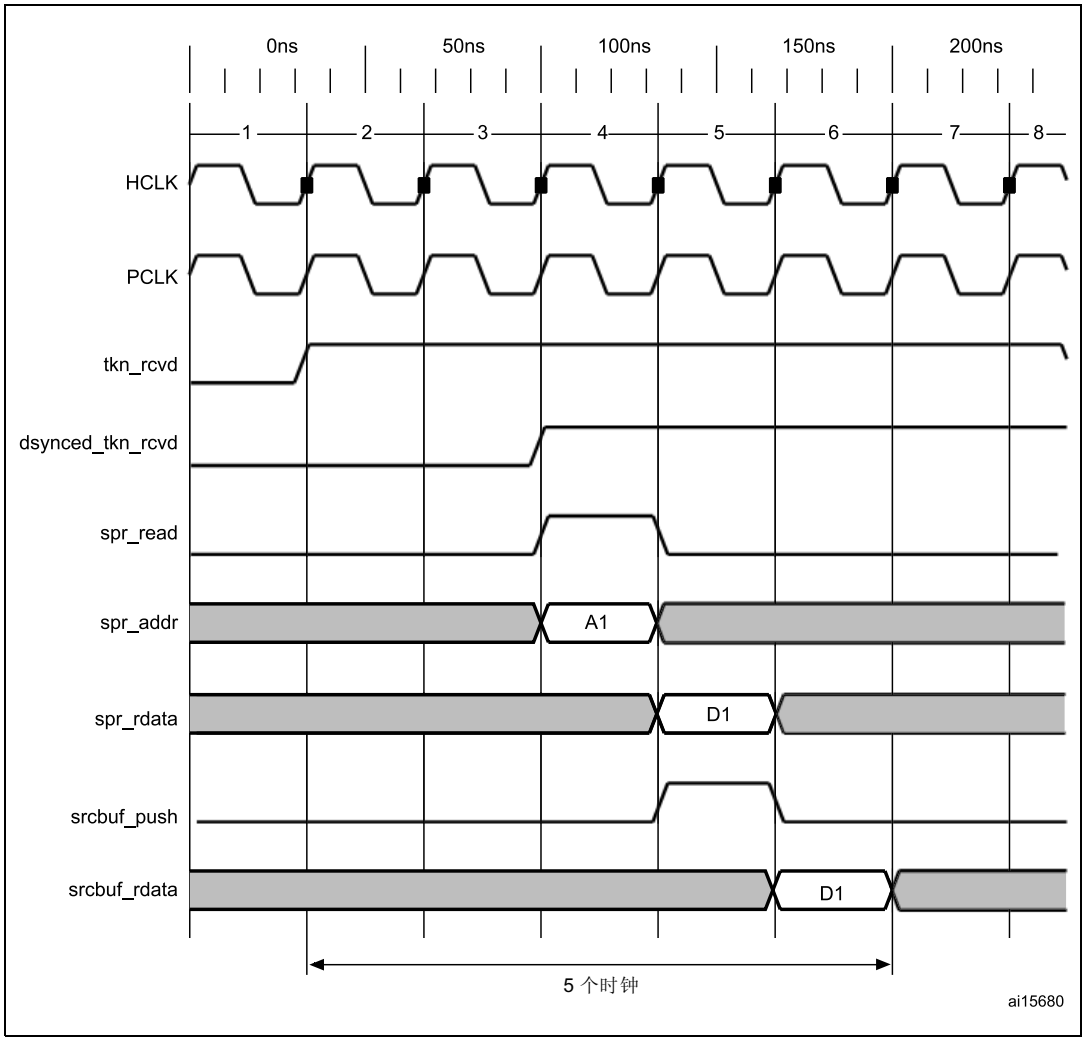
如果 AHB 的运行频率高于 PHY，应用程序可以使用较小的 TRDT 值（在 OTG_GUSBCFG 中）。

图 707 具有以下信号：

- tkn_rcvd：接收到令牌信息（从 MAC 到 PFC）
- dynced_tkn_rcvd：双倍同步 tkn_rcvd（从 PCLK 到 HCLK 域）
- spr_read：读取到 SPRAM
- spr_addr：寻址到 SPRAM
- spr_rdata：从 SPRAM 中读取数据
- srcbuf_push：压入源缓冲区
- srcbuf_rdata：从源缓冲区读取数据。MAC 看到的数据

要计算 TRDT 的值，请参见*表 515：TRDT 值*。

图 707. TRDT 最大时序情况



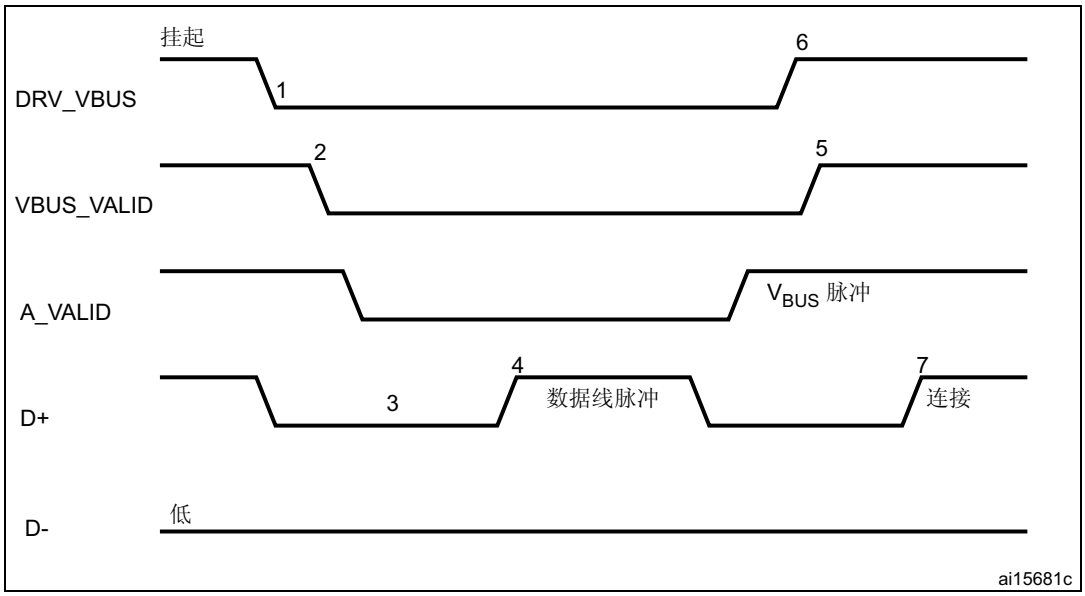
54.15.8 OTG 编程模型

OTG 控制器是一种支持 HNP 和 SRP 的 OTG 设备。该模块接口与“A 型”插头连接时，该模块称为 A 设备。该模块接口与“B 型”插头连接时，该模块称为 B 设备。在主机模式下，OTG 控制器将关闭 V_{BUS} 以节省电能。B 设备可以借助 SRP 请求 A 设备打开 V_{BUS} 电源。设备必须同时执行数据线脉冲和 V_{BUS} 脉冲，但主机可以检测数据线脉冲或者 V_{BUS} 脉冲中的一个用于 SRP。B 设备可以借助 HNP 协商并切换到主机角色。在协商模式下执行 HNP 后，B 设备会挂起总线并恢复到设备角色。

A 设备会话请求协议

应用程序必须将模块 USB 配置寄存器中的 SRP 功能位置 1。这将使能 OTG 控制器在 A 从机模式下检测到 SRP。

图 708. A 设备 SRP



1. DRV_VBUS = 发送到 PHY 的 V_{BUS} 驱动信号
   VBUS_VALID = 来自 PHY 的 V_{BUS} 有效信号
   A_VALID = 发送到 PHY 的 A 设备 V_{BUS} 电平信号
   D+ = 正向数据线
   D- = 负向数据线

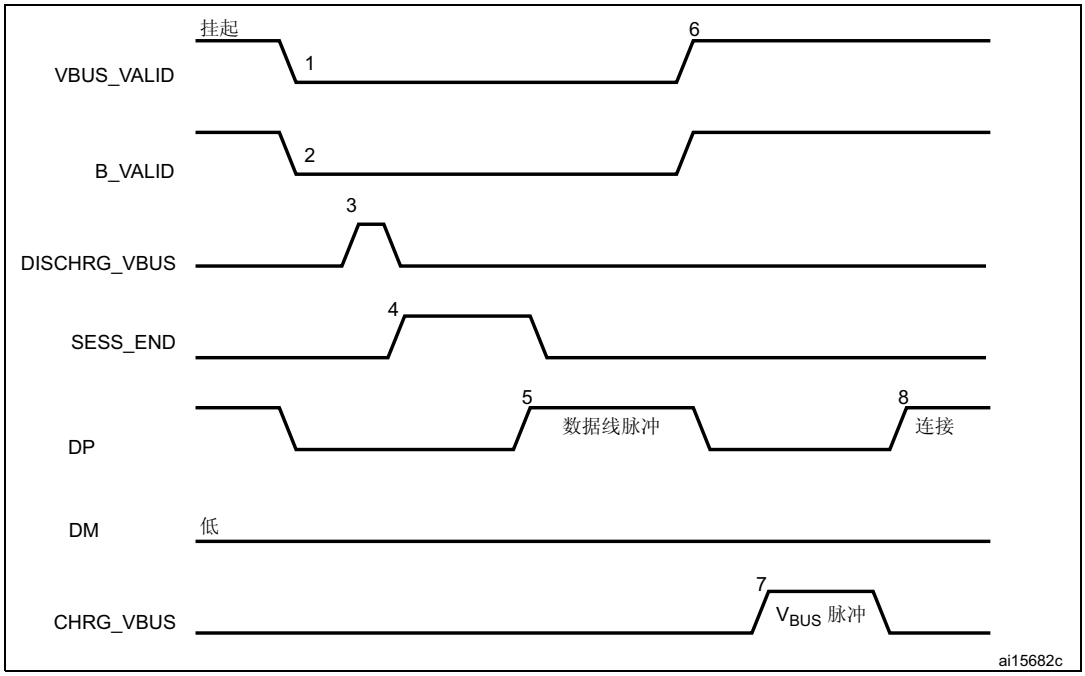
以下几点引用并描述了*图 708* 所示的信号过程：

1. 为节省电能，在总线空闲时，应用程序会通过在主机端口控制和状态寄存器中写入端口挂起位和端口电源位来挂起设备并关闭端口电源。
2. PHY 通过禁止 VBUS_VALID 信号来指示端口电源已关闭。
3. 当 V_{BUS} 电源关闭时，设备必须检测到至少 2 ms 的 SE0 信号才可以启动 SRP。
4. 要启动 SRP，设备需要打开其数据线的上拉电阻并维持 5 到 10 ms。OTG 控制器会检测到数据线脉冲。
5. 设备会驱动 V_{BUS} 到 A 设备会话有效电平（最小 2.0 V）以上，以产生 V_{BUS} 脉冲。OTG 控制器检测到 SRP 时将中断应用程序。在全局中断状态寄存器中，检测到会话请求位（OTG_GINTSTS 中的 SRQINT 位）将置 1。
6. 应用程序必须响应“检测到会话请求”中断，并通过在主机端口控制和状态寄存器中写入端口电源位来打开端口电源位。PHY 通过输出 VBUS_VALID 信号来指示端口已通电。
7. 当 USB 通电后，设备将连接，从而完成 SRP 过程。

B 设备会话请求协议

应用程序必须将模块 USB 配置寄存器中的 SRP 功能位置 1。这将使能 OTG 控制器在 B 从机模式下启动 SRP。OTG 控制器可以借助 SRP 向主机请求新会话。

图 709. B 设备 SRP



1. VBUS_VALID = 来自 PHY 的 V_{BUS} 有效信号
   B_VALID = 发送到 PHY 的 B 设备有效会话
   DISCHRG_VBUS = 发送到 PHY 的放电信号
   SESS_END = 发送到 PHY 的会话结束信号
   CHRG_VBUS = 发送到 PHY 的 V_{BUS} 充电信号
   DP = 正向数据线
   DM = 负向数据线

以下几点引用并描述了*图 709* 所示的信号过程：

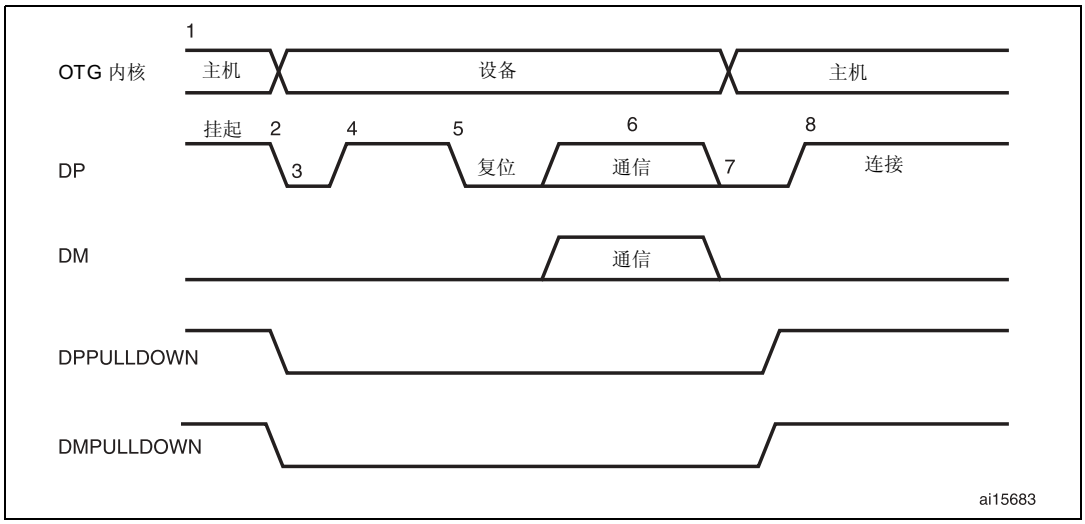
1. 为节省电能，在总线空闲时主机会挂起并关闭端口电源。
   OTG 控制器会在总线空闲 3 ms 后，将模块中断寄存器中的早期挂起位置 1。随后，OTG 控制器会将模块中断寄存器中的 USB 挂起位置 1。
   OTG 控制器会通知 PHY 对 V_{BUS} 进行放电。
2. PHY 指示会话结束。这是 SRP 的初始条件。启动 SRP 之前，OTG 控制器需要 2 ms 的 SE0 信号。
   对于 USB 1.1 全速串行收发器，应用程序必须在 BSVLD 位（位于 OTG_GOTGCTL）被禁止后，等待 V_{BUS} 放电至 0.2 V。此放电时间可从收发器供应商处获取，该值可能因收发器而异。
3. OTG 模块通知 PHY 对 V_{BUS} 加速放电。
4. 应用程序通过将 OTG 控制和状态寄存器中的会话请求位置 1 来启动 SRP。OTG 控制器先执行数据线脉冲，随后执行 V_{BUS} 脉冲。
5. 主机会从数据线脉冲或 V_{BUS} 脉冲检测到 SRP，然后打开 V_{BUS}。PHY 指示 V_{BUS} 对设备上电。

6. OTG 控制器执行 V_{BUS} 脉冲。
   主机通过打开 V_{BUS} 启动一个新会话，指示 SRP 已成功。OTG 控制器通过将 OTG 中断状态寄存器中的会话请求成功状态更改位置 1 来中断应用程序。应用程序读取 OTG 控制和状态寄存器中的会话请求成功位。
7. 当 USB 通电时，OTG 控制器将进行连接，并完成 SRP 过程。

A 设备主机协商协议

通过 HNP 可以将 USB 主机角色从 A 设备切换到 B 设备。应用程序必须将模块 USB 配置寄存器中的 HNP 功能位置 1，以使能 OTG 控制器在 A 从机模式下的 HNP 功能。

图 710. A 设备 HNP



1. DPPULLDOWN = 从模块发送到 PHY 的信号，用于使能/禁止 PHY 内部 DP 线的下拉电阻。
   DMPULLDOWN = 从模块发送到 PHY 的信号，用于使能/禁止 PHY 内部 DM 线的下拉电阻。

以下几点引用并描述了*图 710* 所示的信号过程：

1. OTG 控制器向 B 设备发送 SetFeature b_hnp_enable 描述符以使能 HNP 支持。B 设备的 ACK 响应表示 B 设备支持 HNP。应用程序必须将 OTG 控制和状态寄存器中的主机设置 HNP 使能位置 1，以向 OTG 控制器表明 B 设备支持 HNP。
2. 应用程序不再使用总线时，会通过在主机端口控制和状态寄存器中写入端口挂起位来挂起总线。
3. B 设备检测到 USB 挂起时，将断开连接，指示 HNP 的初始条件。B 设备只有在切换到主机角色时才会启动 HNP，否则总线会保持挂起状态。

   OTG 控制器将 OTG 中断状态寄存器中检测到的主机协商中断位置 1，指示 HNP 的启动。

   OTG 控制器将禁止 PHY 中 DP 线和 DM 线的下拉，以指示设备角色。PHY 使能 OTG_DP 上拉电阻，以告知 B 设备 A 设备的连接。

   应用程序必须读取 OTG 控制和状态寄存器中的当前模式位，以确定设备工作模式。
4. B 设备检测到连接，发出 USB 复位信号，并枚举 OTG 控制器进行数据通信。
5. B 设备继续担当主机角色，发起数据通信，并在结束时挂起总线。

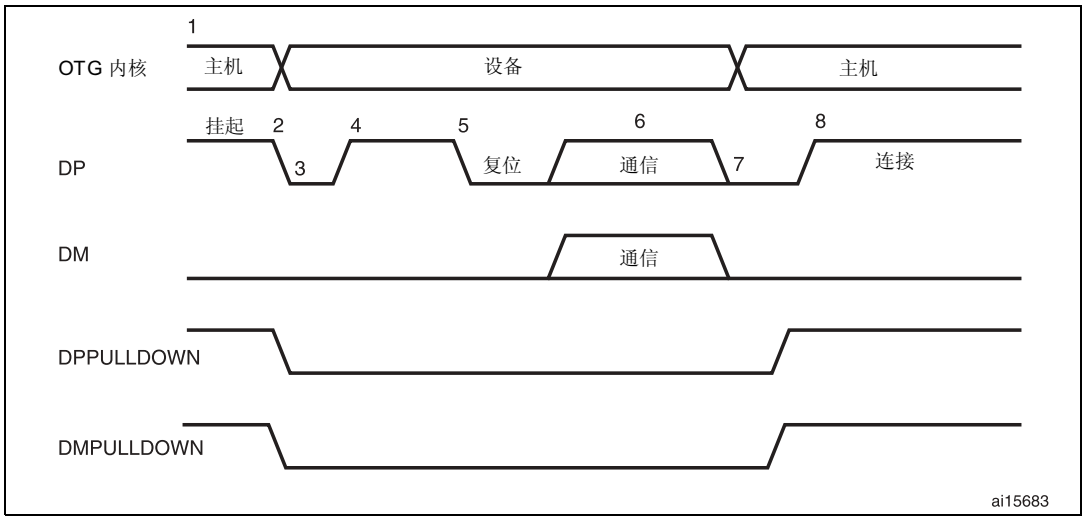
   OTG 控制器会在总线空闲 3 ms 后，将模块中断寄存器中的早期挂起位置 1。随后，OTG 控制器会将模块中断寄存器中的 USB 挂起位置 1。
6. 在协商模式下，OTG 控制器会检测到总线挂起，连接断开，然后切换回主机角色。OTG 控制器将使能 PHY 中 DM 和 DM 下拉电阻，以指示其承担主机角色。

7. OTG 控制器将 OTG 中断状态寄存器中的连接器 ID 状态更改中断置 1。应用程序必须读取 OTG 控制和状态寄存器中的连接器 ID 状态位，以确定 OTG 控制器在 A 从机模式下工作。这向应用程序表示 HNP 已经完成。应用程序必须读取 OTG 控制和状态寄存器中的当前模式位，以确定主机工作模式。
8. 连接 B 设备，完成 HNP 过程。

B 设备主机协商协议

通过 HNP 可以将 USB 主机角色从 B 设备切换到 A 设备。应用程序必须将模块 USB 配置寄存器中的 HNP 功能位置 1，以使能 OTG 控制器在 B 从机模式下的 HNP 功能。

图 711. B 设备 HNP



1. DPPULLDOWN = 从模块发送到 PHY 的信号，用于使能/禁止 PHY 内部 DP 线的下拉电阻。
   DMPULLDOWN = 从模块发送到 PHY 的信号，用于使能/禁止 PHY 内部 DM 线的下拉电阻。

以下几点引用并描述了*图 711* 所示的信号过程：

1. A 设备发出一个 SetFeature b_hnp_enable 描述符以使能 HNP 支持。OTG 控制器的 ACK 响应表示它支持 HNP。应用程序必须将 OTG 控制和状态寄存器中的设备 HNP 使能位置 1，以指示支持 HNP。

   应用程序将 OTG 控制和状态寄存器中的 HNP 请求位置 1，以向 OTG 控制器指示启动 HNP。
2. A 设备不再使用总线时，会通过在主机端口控制和状态寄存器中写入端口挂起位来挂起总线。

   OTG 控制器会在总线空闲 3 ms 后，将模块中断寄存器中的早期挂起位置 1。随后，OTG 控制器会将模块中断寄存器中的 USB 挂起位置 1。

   OTG 控制器会断开连接，A 设备在总线上检测到 SE0，指示 HNP 即将开始。OTG 控制器将使能 PHY 中 DP 和 DM 下拉电阻，以指示其承担主机角色。

   在检测到 SE0 的 3 ms 内，A 设备会通过激活 OTG_DP 上拉电阻进行响应。OTG 控制器检测到此信号时认为有设备接入。

   OTG 控制器将 OTG 中断状态寄存器中的主机协商成功状态更改中断位置 1，指示 HNP 的状态。应用程序必须读取 OTG 控制和状态寄存器中的主机协商成功位，以确定主机协商成功。应用程序必须读取模块中断寄存器 (OTG_GINTSTS) 中的当前模式位，以确定主机工作模式。

3. 应用程序将复位位（OTG_HS_HPRT 中的 PRST 位）置 1，同时 OTG 控制器发出 USB 复位信号，并枚举 A 设备进行数据通信。
4. OTG 控制器继续保持发起通信的主机角色，在通信结束后，会通过将主机端口控制和状态寄存器中的端口挂起位置 1 来挂起总线。
5. 在协商模式下，当 A 设备检测到挂起时，会断开连接，然后切换回主机角色。OTG 控制器将禁止 PHY 中 DP 和 DM 下拉电阻，以指示其承担的设备角色。
6. 应用程序必须读取模块中断寄存器 (OTG_GINTSTS) 中的当前模式位，以确定主机工作模式。
7. OTG 控制器进行连接，完成 HNP 过程。

55 USB HS PHY 控制器 (USBPHYC)

55.1 USBPHYC 简介

USBPHYC 块位于包含以下内容的系统环境中：

- 两个端口高速 PHY
- USB 高速 OTG 控制器（单端口）
- USB 高速 EHCI/OHCI 主机控制器（两个端口）

USBPHYC 允许控制和观察两个端口高速 USB PHY 配置和状态，以及在不同控制器之间共用其中的一个端口。

55.2 USBPHYC 主要特性

- 支持两个端口高速 PHY
- 允许一个端口在两个可用控制器之间切换

55.3 USBPHYC 功能描述

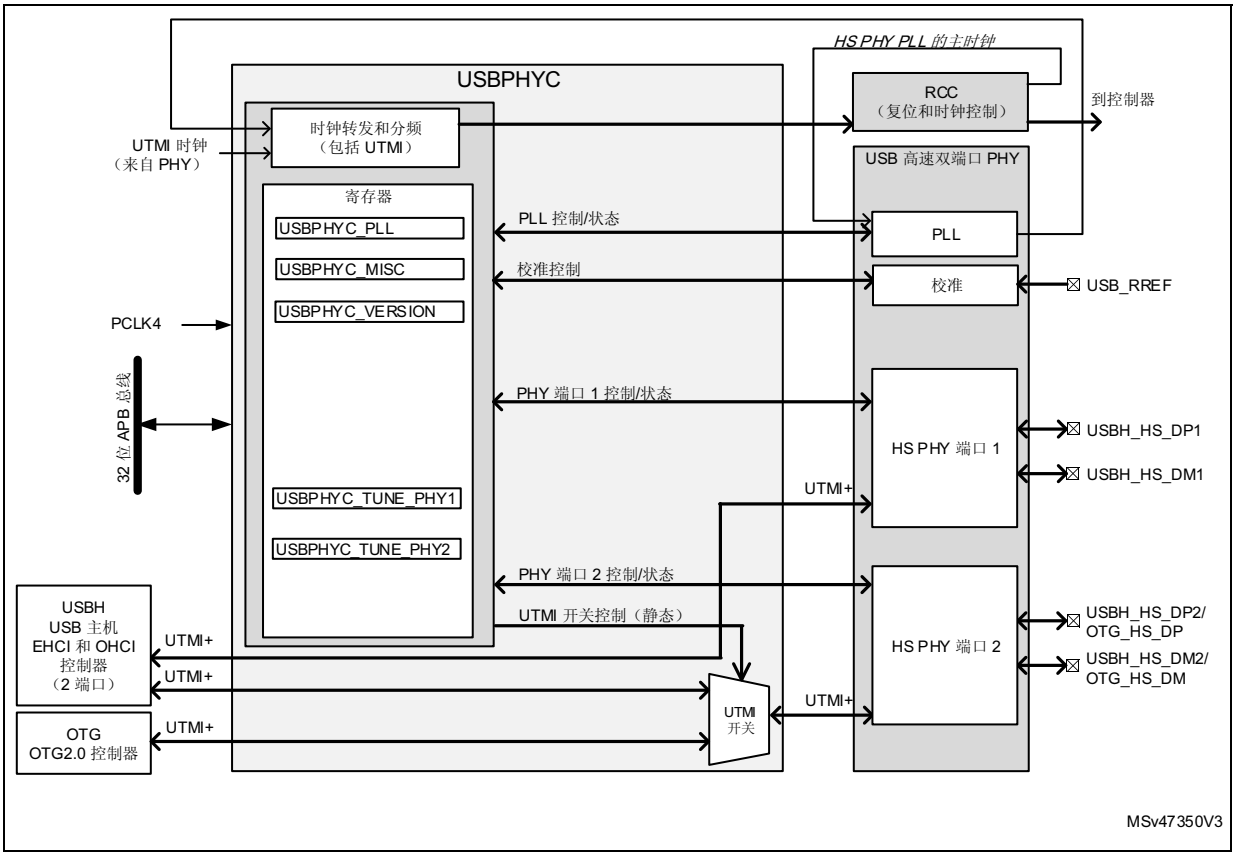
USB HS PHY 控制模块

- 控制 UTMI+ 开关，使得 PHY 的一个端口可以在两个控制器之间共享
- 设置 PLL 值
- 设置 PHY 上的其他控制器（和监视器）

55.3.1 USBPHYC 框图

USBPHYC 寄存器和主要功能如*图 712：USBPHYC 框图*所示。

图 712. USBPHYC 框图



55.3.2 USBPHYC 复位和时钟

USBPHYC 是一个 APB 外设。USBPHYC 逻辑由 APB 总线接口的 PCLK4 时钟直接提供时钟。寄存器通过 APB 总线复位实现复位。

下文提供 USBPHYC 内可用的其他时钟列表（不用于逻辑时钟）：

输入时钟（全部来自 PHY 的 PLL 部分）：

- 来自 PLL 的 60 Mhz 时钟（用作自由运行时钟）
- 来自 PLL 的 48 Mhz 时钟（用于 USB 全速 PHY 功能）
- 每个端口的 UTMI 时钟（用于控制器）

输出时钟用于与其他选项复用和门控（全部到达 RCC）：

- 60 MHz
- 48 MHz
- 12 MHz（派生自 48 MHz）
- UTMI 时钟（对应于每个端口），用于在 RCC 中分配给相关控制器（与每个 PHY 端口关联）

55.3.3 PLL 控制

通常应根据要求配置 PLL，然后设置 PLLEN = 1。

默认不使能小数分频模式。若要使用小数分频，应首先设置 PLLFRACCTL，然后可以使用不同的可行方法配置分频控制的小数部分。

使用第一种方法时，PLLSTRB 应在至少两个 PLL 输入时钟周期内保持高电平，因为在 FBCLK 跳变（无法监控这些跳变）时将发生控制位加载。此外，PLLSTRB 生效后，控制位应至少在输入时钟的 3 个时钟周期内保持稳定。

使用第二种方法时，应通过设置 PLLSTRBBYP = 1 绕过 PLLSTRB。这种情况下，控制位 PLLFRACCTL 和 PLLFRACIN 需要是静态的。仅当 PLL 处于掉电模式 (PLLEN = 0) 时，才能更改控制位。

只能在 PLL 处于粗锁定状态 (LOCKP = 1) 时使用 PLLSTRB 加载控制位。当 PLLSTRBBYP 为低电平时，只要 PLL 失锁，小数位就会复位。要加载小数控制位，只有在锁定后（延迟 100 微秒后）才能将 PLLSTRB 置为有效。

因此，小数分频的推荐方法是先将所有控制设置为所需值并且设置 PLLSTRBBYP = 1，然后再设置 PLLEN = 1 以启动 PLL。

55.4 USBPHYC 寄存器

55.4.1 USBPHYC PLL 控制寄存器 (USBPHYC_PLL)

USBPHYC PLL control register

该寄存器用于控制 HS PHY 的 PLL。

偏移地址：0x000

复位值：0xC000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLDITHEN1	PLLDITHEN0	PLLFRACCTL	PLLSTRBYP	PLLSTRB	PLLEN	PLLFRACIN[15:6]									
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLLFRACIN[5:0]						PLLODF[2:0]			PLLNDIV[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **PLLDITHEN1**：PLL 抖动 1（矩形）(PLL dither 1 (rectangular))

该位使能 PLL 的矩形抖动。

0：使能矩形 PDF 抖动输入到 PLL 的 SDM（不推荐）

1：禁止矩形 PDF 抖动输入到 PLL 的 SDM

位 30 **PLLDITHEN0**：PLL 抖动 2（三角形）(PLL dither 2 (triangular))

该位使能 PLL 的三角形抖动。

0：使能三角形 PDF 抖动输入到 PLL 的 SDM（不推荐）

1：禁止三角形 PDF 抖动输入到 PLL 的 SDM

位 29 **PLLFRACCTL**：PLL 小数模式控制 (PLL fractional mode control)

0：小数模式关闭

1：小数模式开启

位 28 **PLLSTRBYP**：PLL 选通旁路 (PLL strobe bypass)

旁路选通信号。如果在设置 PLLEN = 1 之前置 1，则 PLL 将根据 PLLFRACCTL 的设置以小数模式启动。

0：不旁路选通信号

1：旁路选通信号

位 27 **PLLSTRB**：PLL 选通 (PLL strobe)

该位允许控制作为输入提供给小数控制器的异步选通信号。小数控制器可通过 PLLSTRB 信号的上升沿获知要加载一个新小数字。

0：选通设置为 0

1：选通设置为 1

位 26 **PLLEN**：PLL 使能 (PLL enable)

该位使能 PLL。

0：禁止 PLL

1：使能 PLL

位 25:10 **PLLFRACIN[15:0]**：PLL 小数输入

PLLFRACIN 是 PLL 分频器的小数输入。

位 9:7 **PLLODF[2:0]**：PLL 输出分频系数 (PLL output division factor)

位 6:0 **PLLNDIV[6:0]**：PLL 的环路分频系数（整数部分）

55.4.2 USBPHYC 杂项控制寄存器 (USBPHYC_MISC)

USBPHYC misc control register

该寄存器用于控制 HS PHY 控制器之间的切换。

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PPCKDIS[1:0]		SWITHOST
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:1 **PPCKDIS[1:0]**：智能 HS PHY 端口时钟门控控制 (Intelligent per HS PHY port clock gating control)

该寄存器为静态寄存器。

0x0：无时钟门控PHY 专用 60 MHz 端口 1/端口 2 时钟始终送达目标控制器。PHY 48 MHz 输出时钟始终送达目标控制器。

0x1：端口 1 的智能时钟门控。端口 1 60 MHz 输入时钟已门控（端口 1 未使用）。如果目标控制器（主机或 OTG）请求并行数据通信，则提供端口 2 专用 60 MHz。如果目标控制器（SWITHOST = 1 为主机，否则为 OTG）请求串行通信，则提供 PHY 48 MHz。

0x2：端口 2 的智能时钟门控。端口 2 60 MHz 输入时钟已门控（端口 2 未使用）。如果目标控制器（主机）请求并行数据通信，则提供端口 1 专用 60 MHz。如果目标控制器（SWITHOST = 1 为主机，否则为 OTG）请求串行通信，则提供 PHY 48 MHz。

0x3：端口 1 和端口 2 的智能时钟门控。如果目标（主机）请求并行数据通信，则提供端口 1 专用 60 MHz。如果目标（SWITHOST = 1 为主机，否则为 OTG）请求并行数据通信，则提供端口 2 专用 60 MHz。如果目标控制器（主机或 OTG）请求串行通信，则提供 PHY 48 MHz 输出时钟。

位 0 **SWITHOST**：主机开关 (Switch host)

控制连接 USB 主机控制器于 USB OTG 控制器之间的 USB PHY 第二端口的主开关。该寄存器为静态寄存器。

0：为第二 PHY 端口选择 OTG 控制器

1：为第二 PHY 端口选择主机控制器

55.4.3 USBPHYC PHY x 监视寄存器 (USBPHYC_MONITORx)

USBPHYC PHY x MONITOR register

该寄存器用于控制和监视 HS PHY 的调试接口，端口 #x。

偏移地址：0x108 + 0x100 * (x-1)（x = 1 到 2）

复位值：0x0000 000X

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	MON_SEL[4:0]					LOCKP	Res.	Res.	Res.
							rw	rw	rw	rw	rw	r			

位 31:9 保留，必须保持复位值。

位 8:4 **MON_SEL[4:0]**：调试输出选择 (Choice of debug output)
31：使能锁定位可见性
其他值：保留

位 3 **LOCKP**：PLL 锁定可见性（当 MON_SEL 设置为 31 时）

位 2:0 保留，必须保持复位值。

55.4.4 USBPHYC PHY x 调整寄存器 (USBPHYC_TUNEx)

USBPHYC PHY x TUNE register

该寄存器用于控制 HS PHY 的调整接口，端口 #x。

偏移地址：0x10C + 0x100 * (x-1)（x = 1 到 2）

复位值：0x0407 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	STAGSEL	SHTCCTCTLPROT	HSFALLPREEM	HSRXOFF[1:0]		HDRXGNEQEN	SQLCHCTL[1:0]		OTPCOMP[4:1]			
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTP COMP [0]	HSDRVCHKZTRM[1:0]		HSDRVCHKITRM[3:0]				HSDRVRFRED	FSDRVRFADJ	HSDRVCURINCR	HSDRVDCLEV	HSDRVDCCUR	HSDRVSLEW	LFSCAPEN	INCURRINT	INCURREN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27 **STAGSEL**：交错选择 (Staggering selection)
该位使能 HS Tx 交错。
0：禁止 HS Tx 模式下的基本交错
1：使能 HS Tx 模式下的基本交错

位 26 **SHTCCTCTLPROT**：短路控制保护 (Short circuit control protection)
该位使能 LS/FS 驱动器中的短路保护电路。
0：禁止短路保护
1：使能短路保护

位 25 **HSFALLPREEM**：HS 下降时间预加重 (HS fall time pre-emphasis)
该位使能在预加重期间 HS 下降时间的单端信号控制：
0：开启控制
1：关闭控制

位 24:23 **HSRXOFF[1:0]**：HS 接收器偏移调整 (HS receiver offset adjustment)
0x0：无偏移
0x1：偏移 +5 mV
0x2：偏移 +10 mV
0x3：偏移 -5 mV

位 22 **HDRXGNEQEN**：使能 HS Rx 增益均衡器 (Enable HS Rx gain equalizer)
0：禁止增益均衡器
1：使能增益均衡器

位 21:20 **SQLCHCTL[1:0]**：静噪控制 (Squelch control)

该位字段调整静噪 DC 阈值。

0x0：阈值无偏移

0x1：静噪 DC 阈值偏移 +7 mV

0x2：静噪 DC 阈值偏移 -5 mV

0x3：静噪 DC 阈值偏移 +14 mV

位 19:15 **OTPCOMP[4:0]**：OTP 补偿代码 (OTP compensation code)

位 14:13 **HSDRVCHKZTRM[1:0]**：HS 驱动器扼流圈阻抗调整 (HS driver choke impedance trim)

该位字段控制 PHY 总线 HS 驱动器扼流圈补偿的阻抗调整。

0x0：无阻抗偏移

0x1：将阻抗减少 2 Ω

0x2：将阻抗减少 4 Ω

0x3：将阻抗减少 6 Ω

位 12:9 **HSDRVCHKITRM[3:0]**：HS 驱动器扼流圈电流调整 (HS driver choke current trim)

该位字段控制 HS 驱动器扼流圈补偿电流调整引脚。

0x0：18.87 mA 目标电流/标称 + 0%

0x1：19.165 mA 目标电流/标称 + 1.56%

0x2：19.46 mA 目标电流/标称 + 3.12%

0x3：19.755 mA 目标电流/标称 + 4.68%

0x4：20.05 mA 目标电流/标称 + 6.24%

0x5：20.345 mA 目标电流/标称 + 7.8%

0x6：20.64 mA 目标电流/标称 + 9.36%

0x7：20.935 mA 目标电流/标称 + 10.92%

0x8：21.23 mA 目标电流/标称 + 12.48%

0x9：21.525 mA 目标电流/标称 + 14.04%

0xA：21.82 mA 目标电流/标称 + 15.6%

0xB：22.115 mA 目标电流/标称 + 17.16%

0xC：22.458 mA 目标电流/标称 + 19.01%

0xD：22.755 mA 目标电流/标称 + 20.58%

0xE：23.052 mA 目标电流/标称 + 22.16%

0xF：23.348 mA 目标电流/标称 + 23.73%

位 8 **HSDRVRFRED**：高速升降减少使能 (High-speed rise-fall reduction enable)

0：默认上升/下降时间

1：将上升/下降时间增加 20%

位 7 **FSDRVRFADJ**：调整全速上升/下降时间的调节引脚 (Tuning pin to adjust the full speed rise/fall time)

0：禁止全速上升/下降调节选项

1：使能全速上升/下降调节选项

位 6 **HSDRVCURINCR**：使能 HS 驱动器电流增加功能 (Enable the HS driver current increase feature)

0：禁止 HSDRVDCLEV 功能

1：使能 HSDRVDCLEV 功能

位 5 **HSDRVDCLEV**：HS 驱动器直流电平 (HS driver DC level)

该位允许增加 HS 驱动器直流电平。在 HS 测试 J 和测试 K 数据传输期间无法增加。

0：如果 HSDRVCURINCR = 1，则 HS 驱动器直流电平增加 5 达到 7 mV

1：如果 HSDRVCURINCR = 1，则 HS 驱动器直流电平增加 10 达到 14 mV

位 4 **HSDRVDCCUR**：HS 驱动器直流电平 (HS driver DC level)
该位允许减少 HS 驱动器直流电平。
0：保持正常的 HS 驱动器直流电平
1：HS 驱动器直流电平减 5 达到 7 mV

位 3 **HSDRVSLEW**：HS 驱动器压摆率 (HS driver slew rate)
该位控制 HS 驱动器的压摆率。
0：保持正常压摆率
1：将驱动器压摆率减慢 10%

位 2 **LFSCAPEN**：低全速使能 (Low full speed enable)
该位使能低全速反馈电容。
0：禁止反馈电容。
1：使能反馈电容。

位 1 **INCURRINT**：电流增强值 (Low full speed enable)
该位控制 PHY 电流增强值。
0：如果 INCURREN = 1，提供 1mA 的电流增强
1：如果 INCURREN = 1，提供 2mA 的电流增强

位 0 **INCURREN**：
该位使能电流增强功能。
0：禁止电流增强
1：使能电流增强

55.4.5 USBPHYC 版本寄存器 (USBPHYC_VERR)

USBPHYC VERSION register

该寄存器定义了此 IP 的版本。

偏移地址：0xFFC

复位值：0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本 (Major revision)

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)

55.4.6 USBPHYC 寄存器映射

表 518. USBPHYC 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	USBPHYC_PLL	PLLDITHEN1	PLLDITHEN0	PLLFRACCTL	PLLSTRBYP	PLLSTRB	PLLEN	PLLFRACIN[15:0]																PLLODF[2:0]			PLLNDIV[6:0]						
	Reset value	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x008	USBPHYC_MISC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PPCKDIS[1:0]		SWITHOST
	Reset value																														0	0	0
0x00C-0x0104	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x108	USBPHYC_ MONITOR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MON_SEL [4:0]					LOCKP	Res.	Res.	Res.
	Reset value																								0	0	0	0	0	x			
0x10C	USBPHYC_TUNE1	Res.	Res.	Res.	Res.	STAGSEL	SHTCCTCTLPROT	HSFALLPREEM	HSRXOFF[1:0]		HDRXGNEQEN	SQLCHCTL[1:0]		OTPCOMP[4:0]					HSDRVCHKZTRM[1:0]		HSDRVCHKITRM[3:0]			HSDRVRFRED	FSDRVRFADJ	HSDRVCURINCR	HSDRVDCLEV	HSDRVDCCUR	HSDRVSLEW	LFSCAPEN	INCURRINT	INCURREN
	Reset value					0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0x110 - 0x204	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x208	USBPHYC_ MONITOR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MON_SEL [4:0]					LOCKP	Res.	Res.	Res.
	Reset value																								0	0	0	0	0	x			
0x20C	USBPHYC_TUNE2	Res.	Res.	Res.	Res.	STAGSEL	SHTCCTCTLPROT	HSFALLPREEM	HSRXOFF[1:0]		HDRXGNEQEN	SQLCHCTL[1:0]		OTPCOMP[4:0]					HSDRVCHKZTRM[1:0]		HSDRVCHKITRM[3:0]			HSDRVRFRED	FSDRVRFADJ	HSDRVCURINCR	HSDRVDCLEV	HSDRVDCCUR	HSDRVSLEW	LFSCAPEN	INCURRINT	INCURREN
	Reset value					0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0x210 - 0xFF8	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 518. USBPHYC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFFC	USBPHYC_VERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	0	0	1	0	0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

56 USB_EHCI/USB_OHCI 高速/全速链路控制器 (USBH)

56.1 简介

Portions Copyright (c) Synopsys, Inc. 保留所有权利。使用须经许可。

USBH 是 USB 主机控制器，支持用于全速（OHCI 控制器）和高速（EHCI 控制器）的标准寄存器。

56.2 USBH 主要特性

- 使用片上 2 端口高速 PHY 支持两个物理端口
- 支持 OHCI 模型进行全速和低速操作
- 支持 EHCI 模型进行高速操作
- LPM（链路层电源管理）支持
- 带本地 512 字节数据包缓冲区的 DMA 引擎

支持的标准包括：

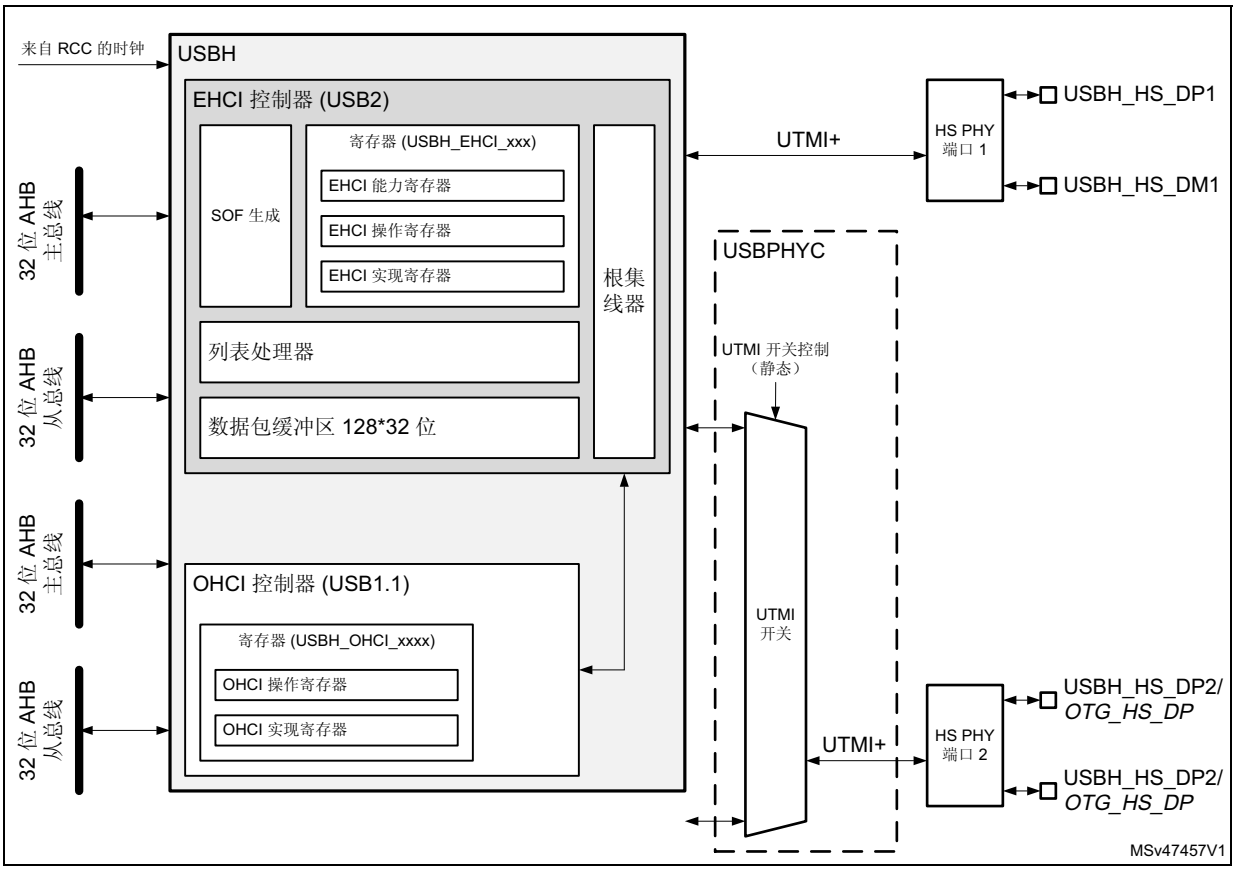
- 通用串行总线增强型主机控制器接口规范修订版 1.0（英特尔公司 2002 年 3 月 12 日）
  - http://www.intel.com
- EHCI 1.1 Addendum 版本 1.1（Intel Corp. 2008 年 8 月）
  - http://www.intel.com
- USB 开放式主机控制器接口规范 1.0a 版（Compaq, Inc.、Microsoft Corp.、National Semiconductor Corp. 1999 年 9 月 14 日）
  - http://www.usb.org
- 通用串行总线规范修订版 2.0（USB Implementers Forum, Inc. 2000 年 4 月 27 日）
  - http://www.usb.org
- USB 2.0 收发器宏单元接口 (UTMI) 规范修订版 1.05（Intel Corp. 2001 年 3 月 29 日）
  - http://www.intel.com
- UTMI+ 规范修订版 1.0（Philips Semiconductor 2004 年 2 月 25 日）
  - http://www.nxp.com

56.3 USBH 功能描述

56.3.1 USBH 框图

USBH 框图如下所示：

图 713. USBH 框图



56.3.2 USBH 复位和时钟

USB 需要正常运行的 AHB 总线时钟和 USB 时钟。USB 时钟在 USBPHYC 中生成（编程并使能 HS PHY 的专用 PLL）。RCC（复位和时钟控制器）控制时钟门控和一些复用选项。有关时钟和复位控制的完整详细信息，请参见*第 55 章：USB HS PHY 控制器 (USBPHYC)* 和*第 10 章：复位和时钟控制 (RCC)*。

注： *要访问 USBH_OHCI 寄存器，需通过使能 USBPHYC 控制的 PLL 来激活 USB 时钟。*

56.3.3 USBH 数据包缓冲区描述

数据包缓冲区提供 IN/OUT 数据的存储和控制。在 OUT 通信事务期间，系统存储控制器（在根集线器内部）从系统内存获取数据并将其写入数据包缓冲区。在 IN 通信事务期间，根集线器模块写入数据。数据包缓冲区的最大深度为 512 字节。

数据包缓冲区深度选择会影响系统延迟和分配给 EHCI 主机控制器的带宽。例如，如果数据包缓冲区深度设置为最大可能值，即 512 字节，那么对于 USB 数据包大小为 1024 字节的 IN 传输，AHB 总线上将会有两个（1024/512）数据传输。

56.3.4 USBH 根集线器的描述

USBH 根集线器将复位和恢复信号传播到下游端口，并处理端口的连接和断开。此外，根集线器实现了端口路由功能，可将端口路由至 EHCI 主机控制器或 OHCI 主机控制器。根集线器作用于本地 PHY 时钟（对自由运行的 60 MHz 时钟起作用）以及来自每个物理端口（通过 8 位接口以 60 MHz 运行）的时钟源。

56.4 USBH 中断

每个控制器都会产生独立的中断：

- OHCI：usbh_ohci_it
  - HCI 总线通用中断（涵盖标准 OHCI 寄存器 HcInterruptStatus 覆盖的所有 9 个中断条件）
- EHCI：usbh_ehci_it
  - 中断异步地址
  - 主机系统错误
  - 帧列表翻转
  - 端口变化
  - USB 错误
  - USB 中断

56.5 USBH 寄存器

注意，USBH 有两个物理 AHB 从设备：

- USBH_OHCI（映射到 USBH_OHCI_BASE）
- USBH_EHCI（映射到 USBH_OHCI_BASE + 0x1000 偏移）

此处所述的 USBH 涵盖两个从设备，为简单起见，所有偏移均相对于最低系统基址（即 USBH_OHCI 从设备的基址）而定义。

56.5.1 USBH OHCI 标准寄存器

OHCI 内核根据 USB 1.1 规范版本 1.0a 实现。标准 OHCI 操作寄存器在后文及*表 519：USBH 寄存器映射和复位值*中列出，但本文不对其进行详述。有关寄存器描述，请参见标准文档。在以下寄存器列表中，前缀表示相对于 USBH_OHCI 基址的偏移地址。

- [0x0000]: USBH_OHCI_HCREVISION
- [0x0004]: USBH_OHCI_HCCONTROL
- [0x0008]: USBH_OHCI_HCCOMMANDSTATUS
- [0x000C]: USBH_OHCI_HCINTERRUPTSTATUS
- [0x0010]: USBH_OHCI_HCINTERRUPTENABLE
- [0x0014]: USBH_OHCI_HCINTERRUPTDISABLE
- [0x0018]: USBH_OHCI_HCHCCA
- [0x001C]: USBH_OHCI_HCPERIODCURRENTED
- [0x0020]: USBH_OHCI_HCCONTROLHEADED
- [0x0024]: USBH_OHCI_HCCONTROLCURRENTED
- [0x0028]: USBH_OHCI_HCBULKHEADED
- [0x002C]: USBH_OHCI_HCBULKCURRENTED
- [0x0030]: USBH_OHCI_HCDONEHEAD
- [0x0034]: USBH_OHCI_HCFMINTERVAL
- [0x0038]: USBH_OHCI_HCFMREMAINING
- [0x003C]: USBH_OHCI_HCFMNUMBER
- [0x0040]: USBH_OHCI_PERIODICSTART
- [0x0044]: USBH_OHCI_HCLSTHRESHOLD
- [0x0048]: USBH_OHCI_HCRHDESCRIPTORA
- [0x004C]: USBH_OHCI_HCRHDESCRIPTORB
- [0x0050]: USBH_OHCI_HCRHSTATUS
- [0x0054]: USBH_OHCI_HCRHPORTSTATUS1
- [0x0058]: USBH_OHCI_HCRHPORTSTATUS2

56.5.2 USBH OHCI 实现寄存器 #6 (USBH_OHCI_INSNREG06)

USBH OHCI implementation register #6

偏移地址：0x0098

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AHB_ECAP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	HBURST[2:0]			EXP_BEATS[4:0]					SUC_BEATS[3:0]			
				r	r	r	r	r	r	r	r	r	r	r	r

位 31 **AHB_ECAP**：AHB 错误捕获 (AHB error captured)
指示遇到了 AHB 错误并且值被捕获。要将该位字段清零，应用必须向该寄存器写入 0。

位 30:12 保留，必须保持复位值。

位 11:9 **HBURST[2:0]**：发生 AHB 错误的控制阶段的 HBURST 值

位 8:4 **EXP_BEATS[4:0]**：预期节拍数 (Number of expected beats)
发生 AHB 错误的突发传输中的预期节拍数。有效值为 0 和 4。
0：如果没有错误，将向 **EXP_BEATS[4:0]** 写入 0。
4：如果 INCR4 并且发生错误，将向 **EXP_BEATS[4:0]** 写入 4。
除 4 之外的其他值不会写入 **EXP_BEATS[4:0]**。

位 3:0 **SUC_BEATS[3:0]**：成功节拍数 (Number of successful beats)
发生 AHB 错误之前，当前突发传输中以成功完成的节拍数。

56.5.3 USBH OHCI 实现寄存器 #7 (USBH_OHCI_INSNREG07)

USBH OHCI implementation register #7

偏移地址：0x009C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AHB_MST_ERR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AHB_MST_ERR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **AHB_MST_ERR[31:0]**：AHB 主设备错误标志 (AHB master error flag)
发生 AHB 错误的控制阶段的 AHB 地址。

56.5.4 USBH EHCI 标准寄存器

EHCI 内核按照 USB 1.1 规范版本 1.0a 和 EHCI 附录 1.1 实现。标准 EHCI 寄存器在后文及 *表 519：USBH 寄存器映射和复位值*中列出，但本文不对其进行详述。有关寄存器描述，请参见标准文档。在以下寄存器列表中，前缀表示相对于 USBH_OHCI 基址的偏移地址。

- EHCI 能力寄存器：
  - [0x1000]: USBH_EHCI_HCICAPLENGTH
  - [0x1004]: USBH_EHCI_HCSPARAMS
  - [0x1008]: USBH_EHCI_HCCPARAMS
- EHCI 功能寄存器
  - [0x1010]: USBH_EHCI_USBCMD
  - [0x1014]: USBH_EHCI_USBSTS
  - [0x1018]: USBH_EHCI_USBINTR
  - [0x101C]: USBH_EHCI_FRINDEX
  - [0x1024]: USBH_EHCI_PERIODICLISTBASE
  - [0x1028]: USBH_EHCI_ASYNCLISTADDR
  - [0x1050]: USBH_EHCI_CONFIGFLAG
  - [0x1054]: USBH_EHCI_PORTSC1
  - [0x1058]: USBH_EHCI_PORTSC2

56.5.5 USBH EHCI 实现寄存器 #1 (USBH_EHCI_INSNREG01)

USBH EHCI implementation register #1

该寄存器控制可编程数据包缓冲区的 OUT/IN 阈值。在该寄存器中指定的值表示 32 位字的数量。

只要从系统内存取得 OUT 阈值的数据量，就将使用该 OUT 阈值来启动 USB 传输。如果数据包缓冲区中没有足够空间来容纳达到该阈值的数据量，还会使用该寄存器来断开数据获取。

只要数据包缓冲区中的数据量达到 IN 阈值，就会使用此 IN 阈值来启动内存传输。如果数据包缓冲区中的数据量未达到阈值，还可以会使用该寄存器来断开数据写入。

可通过 INSN 寄存器编程的最小 OUT 和 IN 阈值为 16 字节。可编程的最小阈值是最高可能的 INCRX 突发传输值，即 INCR16。最小的 OUT 和 IN 阈值应为 64 字节（16×32 位字）。

仅当主机控制器未发起同步/中断通信事务时，OUT 和 IN 阈值才可以与数据包缓冲区深度相等。

注： *分解内存传输位始终使能。*

注： *视数据包缓冲区深度设置而定，可能未使用最高有效位。*

注： *上电复位将这些寄存器设为复位值，软复位不影响这些寄存器。*

偏移地址：0x1094

复位值：0x0020 0020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OUT_THRESHOLD[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IN_THRESHOLD[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **OUT_THRESHOLD[7:0]**：在该位字段中指定的值表示 32 位字的数量

只要从系统内存取得 OUT_THRESHOLD 数量的数据，就将使用 OUT 阈值来启动 USB 传输。如果数据包缓冲区中没有足够空间来容纳达到该阈值的数据量，还会使用该寄存器来断开数据获取。

位 15:8 保留，必须保持复位值。

位 7:0 **IN_THRESHOLD[7:0]**：在该位字段中指定的值表示 32 位字的数量

只要数据包缓冲区中的数据量达到 IN_THRESHOLD 值，就会使用此 IN 阈值来启动内存传输。如果数据包缓冲区中的数据量未达到阈值，还可以会使用该寄存器来断开数据写入。

56.5.6 USBH EHCI 实现寄存器 #2 (USBH_EHCI_INSNREG02)

USBH EHCI implementation register #2

可编程数据包缓冲区深度。

偏移地址：0x1098

复位值：0x0000 0080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKT_BUF[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **PKT_BUF[7:0]**：可编程数据包缓冲区深度 (Programmable packet buffer depth)

可编程数据包缓冲区深度。

在该位字段中指定的值表示 32 位字的数量。

最大有效数据包缓冲区深度为 128（32 位字）。

56.5.7 USBH EHCI 实现寄存器 #3 (USBH_EHCI_INSNREG03)

USBH EHCI implementation register #3

偏移地址：0x109C

复位值：0x0001 2001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENABLE_LS_GLITCH
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	EN_CLK256_CHECK	TESTSE_NAK	TX_TRA_DELAY[2:0]			FRM_LST_FETCH	TIME_AVAIL_OFF[7:0]								BRK_MEM_TRANS
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r

位 31:17 保留，必须保持复位值。

位 16 **ENABLE_LS_GLITCH**：使能/禁止线状态噪声增强 (Enable/disable enhancement for line state glitch)

默认使能此增强。

0：禁止

1：使能

位 15 保留，必须保持复位值。

位 14 **EN_CLK256_CHECK**：使能 256 时钟检查 (Enable 256 clock checking)

该位控制 EHCI 主机控制器恢复序列的结束。默认情况下，该位的值为 0，在恢复序列结束时，主机控制器会等待 SE0 线路状态，然后再将 PHY 切换为高速模式。

如果置 1，则在恢复序列结束时，控制器会等待 SE0 或 256 个时钟，然后再将 PHY 切换为高速模式。

将该位置 1 可使能 256 时钟检查。一些 UTMI PHY 在恢复序列结束时不会在线路上提供 SE0。对于这样的 PHY，应将该位置 1，以使内核不会一直等待 SE0。

该位只应在初始化期间置 1。

位 13 **TESTSE_NAK**：TestSE NAK

TestSE0 Nak 期间忽略线路状态

如果置 1（默认值），内核在 SE0_NAK 测试模式下发送 SOF 时会忽略线路状态检查。

设为 0 时，如果端口状态机在 SE0_NAK 测试期间发送 SOF 时未发现线路状态为 SE0，则它会禁止端口。在 SE0_NAK 测试期间执行阻抗测量时，线路状态可能会变为非 SE0，从而强制内核禁止该端口。该位用于控制此状态下的端口行为。

位 12:10 **TX_TRA_DELAY[2:0]**：Tx-Tx 转向时间额外延迟 (Tx-Tx turnaround delay add on)

该位字段指定要加到内核中保持的“发送到发送之间的转向时间延迟”值中的额外延迟（单位为 phy_clk 时钟数）。该寄存器位字段的默认值为 0。对于大多数 PHY 而言，默认值 0 已足够。但对于在令牌数据包期间设置等待周期的某些 PHY，可能需要编程一个大于 0 的值，以满足发送到发送之间的最小转向时间。建议使用默认值 0，仅在使用最小发送到发送转换时间存在问题时进行更改。该值应在内核初始化期间编程，之后不应进行更改。

位 9 **FRM_LST_FETCH**：

将该位置 1 会强制主控制器在帧的每个微帧中获取周期性帧列表。如果未置 1，则周期性帧列表只会在每个帧的微帧 0 中获取。

默认值为 0（未置 1）。该位只能在内核初始化期间更改，之后不应再进行更改。

位 8:1 **TIME_AVAIL_OFF[7:0]**：可用时间偏移 (Time available offset)

该值表示为计算可用时间需要容纳的额外字节数。

仅当有足够的时间在 EOF1 时间点完成数据包时，才能启动总线上的 USB 流量。有关 EOF1 点的详细信息，请参见 USB 2.0 规范。

该可用时间计算在硬件中完成，可以通过在该位置编程一个值来进一步偏移。

注意：加入可用时间计算的目的是提高未来的灵活性。默认情况下，应用无需编程该位字段。

位 0 **BRK_MEM_TRANS**：分解内存传输 (Break memory transfer)

与 EHCI_H_INSNREG01 结合使用，一旦达到 OUT/IN 阈值，就使能将内存事务分解成块。值固定为 1。

56.5.8 USBH EHCI 实现寄存器 #4 (USBH_EHCI_INSNREG04)

USBH EHCI implementation register #4

偏移地址：0x10A0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUSPEND_SIGNAL	Res.	Res.	Res.	HCCPARAMS_WRT	HCSPARAMS_WRT
										rw				rw	rw

位 31:6 保留，必须保持复位值。

位 5 **SUSPEND_SIGNAL**：挂起信号 (Suspend signal)

0：默认使能该自动功能。当软件将运行/停止复位时，挂起信号置为无效（逻辑电平 1），但 HCH (hchalted) 位尚未置 1。

1：禁止该自动功能，当软件清零运行/停止位时，使所有端口退出挂起状态。这是为了向后兼容。

对于主机已停止而未能将所有处于挂起状态的端口唤醒的系统，如果编程了停止，则端口可能会因来自 PHY 的时钟未运行而卡住。为避免出现这种情况，当主机被软件停止时，EHCI 会使端口自动从挂起状态退出。该位用于禁止这一自动功能。

位 4:2 保留，必须保持复位值。

位 1 **HCCPARAMS_WRT**：
当该位为 1 时，USBH_EHCI_HCCPARAMS 寄存器位 17、15:4 和 2:0 变为可写。

位 0 **HCSPARAMS_WRT**：
当该位为 1 时，USBH_EHCI_HCSPARAMS 寄存器变为可写。

56.5.9 USBH EHCI 实现寄存器 #5 (USBH_EHCI_INSNREG05)

USBH EHCI implementation register #5

偏移地址：0x10A4

复位值：0x0000 1000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBUSY	VPORT[3]
														r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VPORT[2:0]			VCONTROL_LDM	VCONTROL[3:0]				VSTATUS[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r

位 31:18 保留，必须保持复位值。

位 17 **VBUSY**：UTMI VBUSY
VBusy（只读）。硬件指示器，表示对该寄存器进行了写入，硬件当前正在处理由写入的数据所定义的操作。处理完成后，该位会清零。

位 16:13 **VPORT[3:0]**：UTMI VPORT
VPort（读写）。有效写入值仅为 1 和 2。不应将其他值写入该位字段。假设写入了一个非法值 4，之后对该寄存器的任何写操作都将被忽略，并且读取的值将始终是最初写入的非法值（在此例中为 4）。

位 12 **VCONTROL_LDM**：UTMI VCONTROLLoad（厂商控制载入）
有关详细用法，请参见 UTMI 规范。
0：负载
1：NOP

位 11:8 **VCONTROL[3:0]**：UTMI VCONTROL（厂商控制）
厂商定义的 4 位并行输入总线。有关详细用法，请参见 UTMI 规范。

位 7:0 **VSTATUS[7:0]**：UTMI VStatus（厂商状态）
有关详细用法，请参见 UTMI 规范。

56.5.10 USBH EHCI 实现寄存器 #6 (USBH_EHCI_INSNREG06)

USBH EHCI implementation register #6

偏移地址：0x10A8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AHB_ECAP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	HBURST[2:0]			EXP_BEATS[4:0]					SUC_BEATS[3:0]			
				r	r	r	r	r	r	r	r	r	r	r	r

位 31 **AHB_ECAP**：AHB 错误捕获 (AHB error captured)
指示遇到了 AHB 错误并且值被捕获。要清零该位字段，应用必须向该位写入 0。

位 30:12 保留，必须保持复位值。

位 11:9 **HBURST[2:0]**：发生 AHB 错误的控制阶段的 HBURST 值

位 8:4 **EXP_BEATS[4:0]**：预期节拍数 (Number of expected beats)
发生 AHB 错误的突发传输中的预期节拍数。有效值为 0 到 16。
0：如果没有错误，将向 **EXP_BEATS[4:0]** 写入 0。
4：如果 INCR4 并且发生错时，将向 **EXP_BEATS[4:0]** 写入 4。
8：如果 INCR8 并且发生错误，将向 **EXP_BEATS[4:0]** 写入 8。
16：如果 INCR16 并且发生错误，将向 **EXP_BEATS[4:0]** 写入16。
除了 4、8 和 16 之外的其他值不会写入 **EXP_BEATS[4:0]**

位 3:0 **SUC_BEATS[3:0]**：成功节拍数 (Number of successful beats)
AHB 错误发生之前当前突发传输中成功完成的节拍数。

56.5.11 USBH EHCI 实现寄存器 #7 (USBH_EHCI_INSNREG07)

USBH EHCI implementation register #7

偏移地址：0x10AC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AHB_MST_ERR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AHB_MST_ERR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **AHB_MST_ERR[31:0]**：AHB 主设备错误标志 (AHB master error flag)
发生 AHB 错误的控制阶段的 AHB 地址。

56.5.12 USBH 寄存器映射

表 519. USBH 寄存器映射和复位值

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**USBH_OHCI_HCREVISION**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REV[7:0]							
	Reset value																									0	0	0	1	0	0	0	0
0x004	**USBH_OHCI_HCCONTROL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWE	RWC	IR	HCFS[1:0]		BLE	CLE	IE	PLE	CBSR[1:0]	
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x008	**USBH_OHCI_HCCOMMANDSTATUS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOC[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCR	BLF	CLF	HCR
	Reset value															0	0													0	0	0	0
0x00C	**USBH_OHCI_HC INTERRUPT STATUS**	Res.	OC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RHSC	FNO	UE	RD	SF	WDH	SO
	Reset value		0																								0	0	0	0	0	0	0
0x010	**USBH_OHCI_HC INTERRUPT ENABLE**	MIE	OC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RHSC	FNO	UE	RD	SF	WDH	SO
	Reset value	0	0																								0	0	0	0	0	0	0
0x014	**USBH_OHCI_HC INTERRUPT DISABLE**	MIE	OC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RHSC	FNO	UE	RD	SF	WDH	SO
	Reset value	0	0																								0	0	0	0	0	0	0
0x018	**USBH_OHCI_HCHCCA**	HCCA[23:0]																								Res.							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0x01C	**USBH_OHCI_HCPERIODCURRENTED**	PCED[27:0]																												Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x020	**USBH_OHCI_HCCONTROLHEADED**	CHED[27:0]																												Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x024	**USBH_OHCI_HCCONTROLCURRENTED**	CCED[27:0]																												Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x028	**USBH_OHCI_HCBULKHEADED**	BHED[27:0]																												Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

表 519. USBH 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x02C	**USBH_OHCI_HCBULKCURRENTED**	BCED[27:0]																												Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x030	**USBH_OHCI_HCDONEHEAD**	DH[27:0]																												Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x034	**USBH_OHCI_HCFMINTERVAL**	Res.	FSMPS[14:0]															Res.	Res.	FI[13:0]													
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			1	0	1	1	1	0	1	1	0	1	1	1	1	1
0x038	**USBH_OHCI_HCFMREMAINING**	FRT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FR[13:0]													
	Reset value	0																		0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03C	**USBH_OHCI_HCFMNUMBER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FN[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x040	**USBH_OHCI_PERIODICSTART**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PS[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x044	**USBH_OHCI_HCLSTHRESHOLD**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LST[13:0]													
	Reset value																			0	0	0	1	1	0	0	0	1	0	1	0	0	0
0x048	**USBH_OHCI_HCRHDESCRIPTORA**	POTPGT[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NOCP	OCPM	DT	NPS	PSM	NDP[7:0]							
	Reset value	0	0	0	0	0	0	1	0												0	0	0	0	0	0	0	0	0	0	0	1	0
0x04C	**USBH_OHCI_HCRHDESCRIPTORB**	PPCM[15:0]																DR[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0050	**USBH_OHCI_HCRHSTATUS**	CRWE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCIC	LPSC	DRWE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCI	LPS
	Reset value	0														0	0	0													0	0
0x0054	**USBH_OHCI_HCRHPORTSTATUS1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRSC	OCIC	PSSC	PESC	CSC	Res.	Res.	Res.	Res.	Res.	Res.	LSDA	PPS	Res.	Res.	Res.	PRS	POCI	PSS	PES	CCS
	Reset value												0	0	0	0	0							0	0				0	0	0	0	0

表 519. USBH 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0058	**USBH_OHCI_HCRHPORTSTATUS2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRSC	OCIC	PSSC	PESC	CSC	Res.	Res.	Res.	Res.	Res.	Res.	LSDA	PPS	Res.	Res.	Res.	PRS	POCI	PSS	PES	CCS
	Reset value												0	0	0	0	0							0	0				0	0	0	0	0
0x0098	**USBH_OHCI_INSNREG06**	AHB_ECAP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	H BURST [2:0]			EXP_BEATS [4:0]					SUC_ BEATS [3:0]			
	Reset value	0																				0	0	0	0	0	0	0	0	0	0	0	0
0x009C	**USBH_OHCI_INSNREG07**	AHB_MST_ERR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1000	**USBH_EHCI_HCICAPLENGTH**	HCIVERSION[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CAPLENGTH[7:0]								
	Reset value	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0									0	0	0	1	0	0	0	0
0x1004	**USBH_EHCI_HCSPARAMS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DPN[3:0]				Res.	Res.	Res.	P_INDICATOR	N_CC[3:0]				N_PCC [3:0]			PRR	Res.	Res.	Res.	PPC	N_PCC [3:0]			
	Reset value									0	0	0	0				0	0	0	0	1	0	0	1	0	0			1	0	0	1	0
0x1008	**USBH_EHCI_HCCPARAMS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPMC	Res.	EECP[7:0]								ISO[3:0]				Res.	ASPC	PFLF	SFAC
	Reset value															1		1	0	1	0	0	0	0	0	0	0	1	0		1	1	0
0x1010	**USBH_EHCI_USBCMD**	Res.	Res.	Res.	Res.	HIRD[3:0]				ITC[7:0]								Res.	Res.	Res.	Res.	ASPME	Res.	ASP MC [1:0]		LHCR	IAAD	ASE	PSE	FLS[1:0]	HCR	RS
	Reset value					0	0	0	0	0	0	0	0	1	0	0	0					1		1	1	0	0	0	0	0	0	0	0
0x1014	**USBH_EHCI_USBSTS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ASS	PSS	REC	HCH	Res.	Res.	Res.	Res.	Res.	Res.	IAA	HSE	FLR	PCD	USBERRINT	USBINT
	Reset value																	0	0	0	1							0	0	0	0	0	0
0x1018	**USBH_EHCI_USBINTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IAAE	HSEE	FLRE	PCIE	USBERRINTE	USBINTE
	Reset value																											0	0	0	0	0	0
0x101C	**USBH_EHCI_FRINDEX**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRINDEX[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1024	**USBH_EHCI_PERIODICLISTBASE**	PERIODICLISTBASE[31:12]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												

表 519. USBH 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1028	**USBH_EHCI_ASYNCLISTADDR**	LPL[31:5]																											Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x1050	**USBH_EHCI_CONFIGFLAG**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CF
	Reset value																																0
0x1054	**USBH_EHCI_PORTSC1**	DEVADDR[6:0]							SS[1:0]		WKOC_E	WKDSCNNT_E	WKCNNT_E	PTC[3:0]				PIC[1:0]		PO	PP	LS[1:0]		SUSP_L1	PR	SUSPEND	FPR	OCC	OCA	PEDC	PED	CSC	CCS
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1058	**USBH_EHCI_PORTSC2**	DEVADDR[6:0]							SS[1:0]		WKOC_E	WKDSCNNT_E	WKCNNT_E	PTC[3:0]				PIC[1:0]		PO	PP	LS[1:0]		SUSP_L1	PR	SUSPEND	FPR	OCC	OCA	PEDC	PED	CSC	CCS
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1094	**USBH_EHCI_INSNREG01**	Res.	Res.	Res.	Res.	Res.	Res.	Res.		OUT_THRESHOLD[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IN_THRESHOLD[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0x1098	**USBH_EHCI_INSNREG02**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKT_BUF[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x109C	**USBH_EHCI_INSNREG03**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENABLE_LS_GLITCH	Res.	EN_CLK256_CHECK	TESTSE_NAK	TX_TRA_DELAY[2:0]			FRM_LST_FETCH	TIME_AVAIL_OFF[7:0]								BRK_MEM_TRANS
	Reset value																1		0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
0x10A0	**USBH_EHCI_INSNREG04**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUSPEND_SIGNAL	Res.	Res.	Res.	HCCPARAMS_WRT	HCSPARAMS_WRT
	Reset value																											0				0	0

表 519. USBH 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x10A4	**USBH_EHCI_INSNREG05**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBUSY	VPORT[3:0]				VCONTROL_LDM	VCONTROL[3:0]				VSTATUS[7:0]							
	Reset value															0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0x10A8	**USBH_EHCI_INSNREG06**	AHB_ECAP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HBURST[2:0]			EXP_BEATS[4:0]					SUC_BEATS[3:0]			
	Reset value	0																				0	0	0	0	0	0	0	0	0	0	0	0
0x10AC	**USBH_EHCI_INSNREG07**	AHB_MST_ERR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

57 以太网 (ETH)：带有 DMA 控制器的千兆介质访问控制 (GMAC)

57.1 以太网简介

Portions Copyright (c) Synopsys, Inc. 保留所有权利。使用须经许可。

借助以太网外设，可以通过以太网按照 IEEE 802.3-2008 标准发送和接收数据。

该外设支持配置，可满足各种消费类应用和工业应用的需求。

57.2 以太网主要特性

以太网外设嵌入了针对直接内存接口的专用 DMA、介质访问控制器 (MAC) 和支持多种格式的 PHY 接口模块。

57.2.1 标准合规

以太网外设遵守以下标准：

- IEEE 802.3-2008，用于以太网 MAC 和介质独立接口 (MII)
- IEEE 1588-2008，用于精密网络时钟同步 (PTP)
- IEEE 802.3az-2010，用于节能型以太网 (EEE)
- AMBA 2.0，用于 AHB 从端口
- AMBA4，用于 AXI 主端口
- HP/Marvell 推出的 RGMII 规范第 2.6 版
- RMII 联盟的 RMII 规范第 1.2 版

小心： 千兆介质独立接口 (GMII) 仅适用于内部，用于为 RGMII 适配器供电。GMII 信号不支持片外使用。

57.2.2 MAC 特性

MAC Tx 和 Rx 通用特性

- 为应用程序提供单独的发送、接收和控制接口
- 支持以下 PHY 接口实现 10、100 和 1000 Mbps 数据传输速率：
  - 通过符合 IEEE 802.3 的 MII 接口与外部快速以太网 PHY 进行通信
  - RGMII 接口，与外部千兆 PHY 进行通信
  - RMII 接口，与外部快速以太网 PHY 进行通信
- 半双工操作：
  - 支持 CSMA/CD 协议
  - 使用背压进行流控制（基于特定于实现的白皮书以及 UNH 以太网条款 4 MAC 测试套件 - 附录 D）
  - 在 1000 Mbps 半双工操作中支持数据包突发和数据包扩展
- IEEE 802.3az-2010 标准，用于 MII 和精简千兆介质独立接口 (RGMII) PHY 中的节能型以太网

- 应用侧有 32 位数据传输接口
- 全双工流控制操作（IEEE 802.3x 暂停数据包和优先级流控制）
- 通过 RMON 或 MIB 计数器进行的网络统计（部分支持 RFC2819/RFC2665）
- 以太网数据包时间戳如 IEEE 1588-2002 和 IEEE 1588-2008 规范所述（PTP 数据包的 Tx 或 Rx 状态中给出了 64 位时间戳）。Tx 方向支持单步和双步时间戳。
- 可灵活控制每秒脉冲数 (PPS) 输出信号（eth_ptp_pps_out 和 ETH_PPS_OUT）
- 用于配置和管理 PHY 设备的 MDIO（符合条款 22 和条款 45）主接口

MAC Tx 特性

- 支持插入前导码和帧起始数据 (SFD)
- 应用程序发送的每个数据包都有 32 位状态
- 可逐帧控制 CRC 和 pad 自动生成
- 可编程数据包长度，支持标准以太网数据包或高达 16 KB 的巨型以太网数据包
- 可编程数据包间隙（40-96 位时间，以 8 为步长）
- 当流控制输入（在全双工模式下）由生效转换为失效时，IEEE 802.3x 流控制会自动发送零量暂停数据包
- 源地址位字段插入或替换，以及发送数据包中的 VLAN 插入、替换和删除（按数据包控制或采用静态-全局控制）
- 插入、替换或删除最多两个 VLAN 标记
- 可在全双工模式下发送前导码更短的数据包
- 插入、替换或删除基于队列/通道的 VLAN 标记

MAC Rx 特性

- 自动 Pad 和 CRC 去除选项
- 可禁止自动 CRC 检查
- 支持删除前导码和 SFD
- 独立的 112 位或 128 位状态
- 可编程看门狗超时限制
- 灵活的地址过滤模式：
  - 四个 48 位完美 (DA) 地址过滤器，对每个字节进行屏蔽操作
  - 四次 48 位 SA 地址比较检查，对每个字节进行屏蔽操作
  - 64 位散列过滤器，适用于多播和单播 (DA) 地址
- 可传送所有多播地址数据包
- 支持混合模式，因此可传送所有数据包，无需为网络监视进行过滤
- 传送所有传入数据包时（每次过滤时）均附有一份状态报告
- 附加数据包过滤：
  - 基于 VLAN 标记：完美匹配和散列过滤（基于外部或内部 VLAN 标记）
  - 基于第 3 层和第 4 层：TCP/UDP over IPv4/IPv6
- IEEE 802.1Q VLAN 标记检测，并且可删除接收数据包中的 VLAN 标记
- 检测远程唤醒数据包和 AMD 魔术数据包
- （在全双工模式下）可选择将接收的暂停数据包转发到应用程序
- 已接收数据包的第 3 层/第 4 层校验和减荷
- 去除最多两个 VLAN 标记，以及在状态中提供标记

57.2.3 事务层 (MTL) 特性

MTL Tx 和 Rx 通用特性

- 32 位事务层模块（用于桥接应用程序和 MAC）
- 通过数据包分隔符优化以数据包为导向的传输
- 可编程突发长度可达 MTL Rx 队列或 Tx 队列大小的一半，用于支持 EQOS-MTL 配置中的突发数据传输
- 每个队列均支持可编程阈值功能（默认为 64 个字节）

MTL Tx 特性

- 2048 字节发送 FIFO，具有可编程阈值功能
- 发送路径上有两个 Tx 队列，所有发送队列共享一个内存
- 存储转发机制或阈值模式（直通），用于向 MAC 发送数据
- 多队列配置中支持可编程队列大小。每个队列大小均可以 256 字节为单位进行编程
- 在半双工模式下自动重新发送冲突数据包
- 丢弃延迟冲突、过度冲突、过度延迟和下溢条件下的数据包，并显示相应的状态
- 利用模块计算 IPv4 报头校验和与 TCP、UDP 或 ICMP 校验和并将其插入在存储转发模式下发送的帧
- 为 Tx FIFO 中（由于下溢）丢失的数据包生成脉冲，以此获得统计数据
- 数据包层面控制适用于
  - VLAN 标记插入或替换
  - 以太网源地址插入
  - 第 3 层/第 4 层校验和插入控制
  - 单步时间戳
  - 时间戳控制
  - CRC 和填充控制
- 在多队列配置中遵循以下调度算法
  - 加权轮询 (WRR)
  - 严格优先级 (SP)

MTL Rx 特性

- 2048 字节接收 FIFO，具有可配置阈值
- 接收路径上有两个 Rx 队列，所有 Rx 队列共享一个内存
- 在阈值（或直通）模式下支持可编程 Rx 队列阈值（默认固定为 64 字节）
- 在存储转发模式下，可以在接收时过滤所有的错误数据包，但不将这些错误数据包转发给应用程序
- 可以转发过小的良好数据包
- 为 Rx FIFO 中（由于上溢）丢失的数据包生成脉冲，以此获得统计数据
- 根据 Rx 队列填充级别自动生成要发送至 MAC 的暂停数据包控制或背压信号
- 存在多个队列时，可在队列之间执行仲裁。支持下列仲裁方案：
  - 加权轮询 (WRR)
  - 加权严格优先级 (WSP)
  - 严格优先级 (SP)

57.2.4 DMA 模块特性

DMA 模块在外设和系统内存之间交换数据。DMA 传输由软件描述符结构驱动。应用程序可使用一组寄存器（请参见*第 57.11.2 节：以太网 DMA 寄存器*）来控制 DMA 操作。DMA 模块支持以下特性：

- 32 位数据传输
- 在发送路径中为每个队列提供单独的 DMA 通道
- 在接收路径中为两个队列提供单一 DMA 通道
- 通过数据包分隔符优化以数据包为导向的 DMA 传输
- 支持对数据缓冲区进行字节对齐寻址
- 支持双缓冲区（环）描述符
- 采用描述符架构，可以在 CPU 几乎不干预的情况下传输大型数据块（每个描述符最多可传输 32 KB 的数据）
- 报告正常工作和传输错误的综合状态
- 可为 Tx DMA 和 Rx DMA 引擎单独编程突发长度，以充分利用主机总线
- 可编程中断选项，适用于不同的工作条件
- 按数据包控制发送或接收完成中断
- 接收引擎和发送引擎间采用循环调度仲裁或固定优先级仲裁
- 启动模式和停止模式
- 主机控制访问和主机数据接口 (AXI) 分别采用单独的端口
- 使能了 TCP 分段减荷 (TSO) 功能的 Tx DMA 通道
- 对发送描述符发布写入进行可编程控制，以提高吞吐量

57.2.5 总线接口功能

AXI 主接口

AXI 主接口的特性如下：

- 通过 AXI4 兼容接口与应用程序进行交互
- 32 位地址宽度
- 32 位数据
- OKAY、SLVERR 和 DECERR 响应
- 在 AXI 主接口中利用软件选择 AXI 突发类型（固定和可变长度突发）；将固定长度突发扩展为 32、64、128 和 256
- 通过 AXI 读写数据通道发送握手信号

AHB 从接口

AHB 从接口支持以下特性：

- 通过 AHB 与应用进行接口
- AHB 从接口（32 位），用于 CSR 访问
- 所有 AHB 突发类型

AHB 从接口不会生成以下响应：

- 拆分
- 重试
- 错误

57.2.6 通用排队特性

以太网外设支持以下通用排队特性：

- 利用可编程控制将带有多播/广播目标地址的接收数据包传送到可编程接收队列
- 支持将未标记的接收数据包传送到可编程接收队列
- 利用可编程控制将带有和未带有 VLAN 标记的 IEEE 1588 PTP over Ethernet 接收数据包传送到相同或单独的可编程接收队列
- 利用可编程控制将未通过目标地址过滤的单播/多播接收数据包传送到单独的可编程接收队列

57.3 以太网引脚和内部信号

表 520 列出了连接到封装引脚或焊球的以太网输入与输出信号。有效引脚取决于所选择的 PHY 类型（MII、RMII 或 RGMII）以及设备配置。

表 521 显示了内部以太网信号。

表 520. 以太网外设引脚

端口名称	数字端口类型	说明
ETH_COL	输入	冲突检测信号，仅限 MII
ETH_CRS	输入	载波侦听信号，仅限 MII
ETH_REF_CLK	输入	RMII 参考时钟
ETH_RX_CLK	输入	进行 Rx 数据传输时的 MII/RGMII 参考时序
ETH_RXD[3:0]	输入	接收的数据。 4 个引脚用于 MII 和 RGMII，2 个引脚用于 RMII。
ETH_RX_DV	输入	接收数据有效
ETH_CRS_DV	输入	RMII：在交替的时钟周期上复用载波侦听 (CRS) 和 RX_Data 有效 (RX_DV)。在 10 Mb/s 模式下，该操作每隔 10 个时钟周期交替一次。
ETH_RX_CTL	输入	解复用后，该引脚会在 ETH_RX_CLK 的上升沿提供 ETH_RX_DV，并提供 ETH_RX_ER（ETH_RX_CLK 上升沿和下降沿的 ETH_RX_CTL 值之间的逻辑异或运算结果）。
ETH_RX_ER	输入	接收错误
ETH_TX_CLK	输入	进行 Tx 数据传输时的 MII 参考时序
ETH_GTX_CLK	输出	进行 Tx 数据传输时的 RGMII 参考时序
ETH_TXD[3:0]	输出	发送数据。 4 个引脚用于 MII 和 RGMII，2 个引脚用于 RMII。
ETH_TX_EN	输出	发送数据使能
ETH_TX_CTL	输出	在 ETH_TX_CLK 的上升沿复用 ETH_TX_EN，并在 ETH_TX_CLK 下降沿的 ETH_TX_EN 和 ETH_TX_ER 值之间进行逻辑异或运算
ETH_TX_ER	输出	发送错误
ETH_MDC	输出	管理数据时钟
ETH_MDIO	输入/输出	管理数据

表 520. 以太网外设引脚（续）

端口名称	数字端口类型	说明
ETH_PHY_INTN	输入	PHY 中断
ETH_PPS_OUT	输出	PTP 每秒脉冲数输出

表 521. 以太网内部输入/输出信号

信号名称	信号类型	说明
eth_hclk	数字输入	AHB 时钟
eth_aclk	数字输入	AXI 时钟
eth_sbd_intr_it	数字输出	主要以太网中断
lpi_intr_o	数字输出	发送器或接收器进入或退出 LPI 状态时生成的边带信号
pmt_intr_o	数字输出	接收到有效远程唤醒数据包时生成的边带信号
eth_mii_tx_clk	数字输入	MII Tx 内核时钟
eth_mii_rx_clk	数字输入	MII Rx 内核时钟
eth_rmii_ref_clk	数字输入	RMII 参考内核时钟
eth_ptp_pps_out	数字输出	PTP 每秒脉冲数信号
mac_speed_o[1:0]	数字输出	RCC 使用的 MAC 速度信息
clk_ptp_ref_i	数字输入	PTP 参考时钟输入
eth_ptp_trig[4:1]	数字输入	PTP 系统时间辅助快照的触发输入

57.4 以太网架构

以太网外设由 4 个主要功能模块组成：

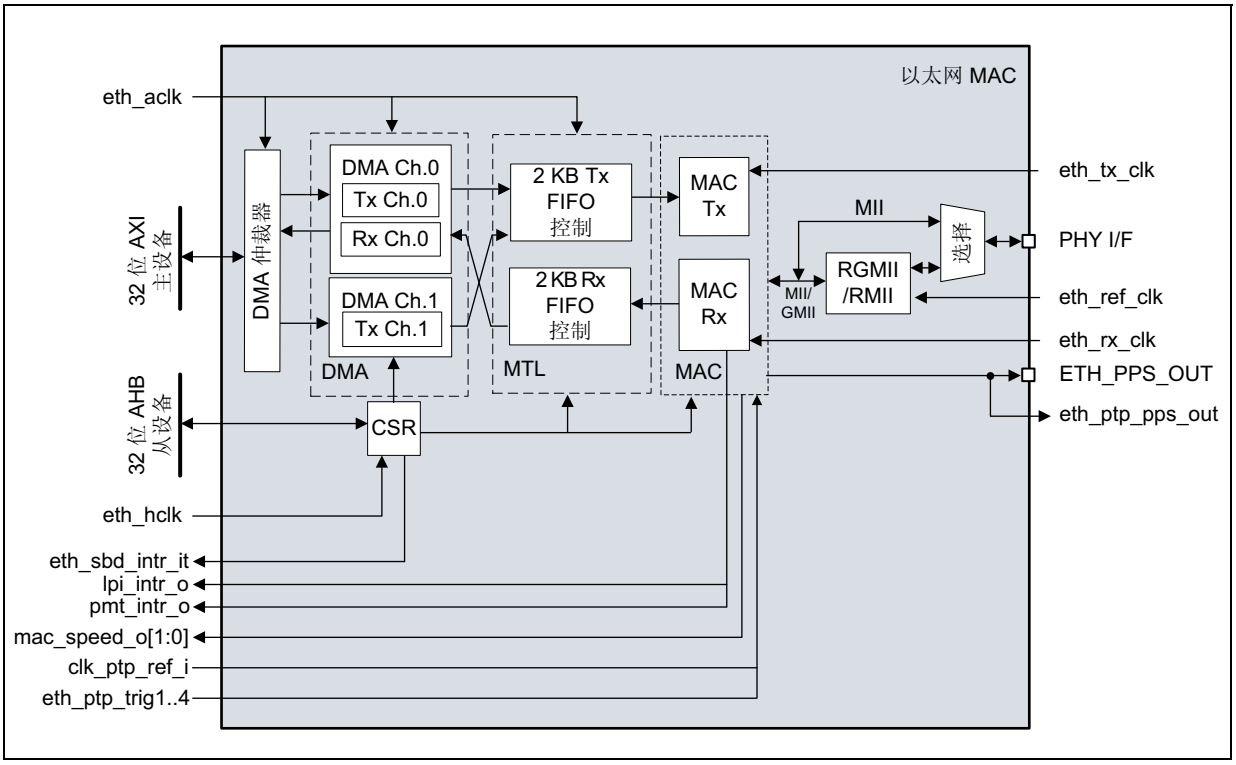
- **控制和状态寄存器模块 (CSR)**，用于控制通过 AHB 32 位从接口进行的寄存器访问
- 直接内存访问接口 (DMA)

  此为逻辑 DMA 模块，具有 1 条用于接收的物理通道和 2 条用于发送的物理通道。该模块用于控制通过 AMBA AXI4 64 位主接口在 MAC 和系统内存之间进行的数据传输。
- **介质访问控制模块 (MAC)**，负责实现以太网协议
- **MAC 事务层 (MTL)**，负责控制应用和 MAC 之间的数据流。

此外，还增添了协议适配模块以支持 RMII 或 RGMII PHY 介质独立接口。

注： 千兆介质独立接口 (GMII) 仅适用于内部，用于为 RGMII 适配器供电。GMII 信号不支持片外使用。

图 714. 以太网高级框图



1. 有关内部信号的定义，请参见*表 521*。
2. 有关以太网时钟架构的详细说明，请参见 RCC 一章的“以太网的时钟分配”。

57.4.1 DMA 控制器

DMA 具有独立的发送 (Tx) 和接收 (Rx) 引擎。Tx 引擎将数据从系统内存传输到 MAC 事务层 (MTL)，而 Rx 引擎将数据从设备端口 (PHY) 传输到系统内存。

此控制器可以在应用 CPU 完全不干预的情况下，通过描述符有效地将数据从源传送到目标。DMA 专为面向数据包的数据传送（如以太网中的数据包）而设计。该控制器经过编程后，可在完成数据包发送和接收传送操作时以及其他正常或错误条件下向应用 CPU 提供中断。

DMA 数据结构

DMA 和应用通过以下两种数据结构实现通信：

- 控制和状态寄存器 (CSR)
- 描述符列表和数据缓冲区

DMA 既可将 MAC 接收到的数据包传送到系统内存中的 Rx 缓冲区，也可传送来自系统内存的 Tx 缓冲区的 Tx 数据包。位于系统内存中的描述符包含指向这些缓冲区的指针。

每个列表的基址均写入到相应的 Tx 和 Rx 寄存器：*通道 x Tx 描述符列表地址寄存器 (ETH_DMACxTXDLAR)* 和*通道 0 Rx 描述符列表地址寄存器 (ETH_DMAC0RXDLAR)*。

描述符列表是一种前向链表，相邻两个描述符之间始终相差一个固定的偏移值。列表中的描述符数量在相应的 Tx/Rx（*通道 x Tx 描述符环长度寄存器 (ETH_DMACxTXRLR)* 和*通道 0 Rx 描述符环长度寄存器 (ETH_DMAC0RXRLR)*）中进行编程。

DMA 处理完列表中的最后一个描述符后，会自动跳转回列表地址寄存器中的相应描述符，以形成一个描述符环。描述符列表位于应用的物理存储地址空间内。每个描述符最多可指向两个缓冲区。因此，可使用并以物理方式寻址两个缓冲区，而非内存中的连续缓冲区。

数据缓冲区位于应用的物理存储空间，通常由整个数据包或部分数据包组成，但不会超过单个数据包。缓冲区中仅包含数据。其状态保存在描述符中。数据链接是指跨越多个数据缓冲区的数据包。但是，单个描述符不能跨越多个数据包。检测到 EOP（数据包结束）时，DMA 会跳到下一个数据包的数据缓冲区。

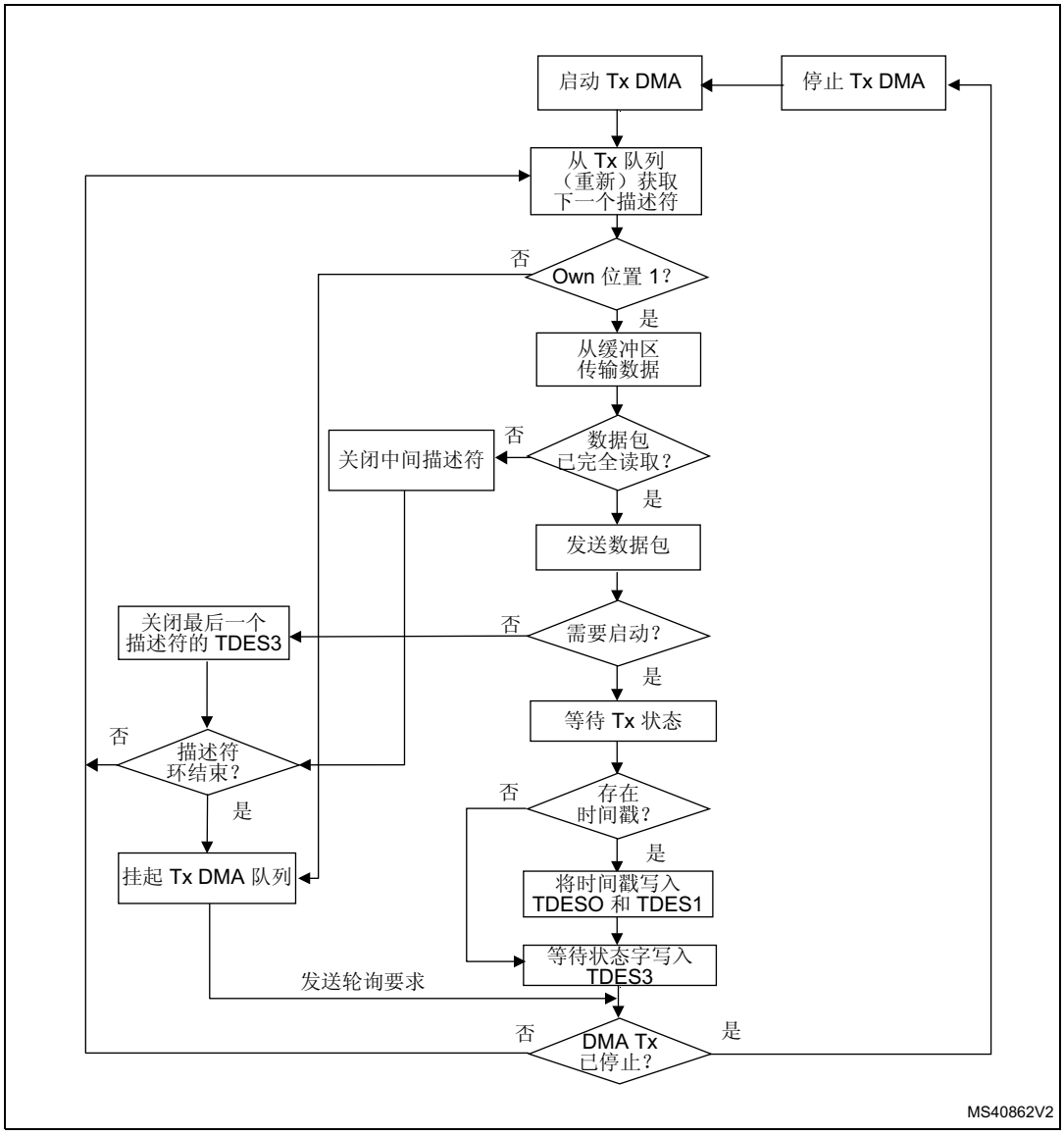
*第 57.10 节：描述符*中指定了描述符。

默认模式下的 DMA 发送

Tx DMA 引擎在默认模式下的操作顺序如下：

1. 在为数据缓冲区设置好相应的以太网数据包数据后，应用配置发送描述符 (TDES0–TDES3) 并将 Own 位 (TDES0[31]) 置 1。
2. 应用对发送通道的描述符尾指针偏移值进行移位。
3. DMA 处于运行状态时，会运行一个仲裁周期以选择下一个 Tx DMA 通道，以便处理其中需要发送的数据包。
4. DMA 从应用的内存中获取描述符。
5. 如果 DMA 检测到以下条件之一，则会挂起该通道的发送操作，相应 DMA 通道状态寄存器的位 2 和位 16 会置 1，Tx 引擎将执行步骤 11：
   - 描述符被标记为应用所有 (TDES3 [31] = 0)。
   - 在环描述符列表模式下，描述符尾指针与当前描述符指针相等。
   - 出现错误条件。
6. 如果获得的描述符标记为 DMA 所有 (TDES3[31] = 1)，则 DMA 将根据取得的描述符对发送数据缓冲区地址进行解码。
7. DMA 从系统内存中获取发送数据并将数据传送到 MTL 进行发送。
8. 如果以太网数据包保存在多个描述符的数据缓冲区中，则 DMA 将关闭中间描述符并获取下一个描述符。重复执行步骤 3 到步骤 7，直到以太网数据包的结束数据传送到 MTL。
9. 完成数据包发送后，如果已为数据包使能 IEEE 1588 时间戳功能（按 Tx 状态中的指示），则从 MTL 获得的时间戳值将写入包含 EOP 缓冲区的 Tx 描述符（TDES0 和 TDES1）。随后，状态信息将写入该 Tx 描述符 (TDES3)。该描述符现在为应用所有，因为 Own 位在此步骤中被清零。如果为该数据包禁止时间戳功能，则 DMA 不会更改 TDES0 和 TDES1 的内容。
10. 发送完在其最后一个描述符中将完成时中断 (TDES2[31]) 置 1 的数据包后，*通道 x 状态寄存器 (ETH_DMACxSR)* 的位 0 将置 1。随后，DMA 引擎返回步骤 3。
11. 在挂起状态下，DMA 尝试再次获取描述符（进而返回到步骤 3）。当接收到发送轮询要求并且下溢中断状态位清零时，通过对*通道 x Tx 描述符尾指针寄存器 (ETH_DMACxTXDTPR)* 写入任一值来触发轮询要求命令。如果应用通过将相应 DMA 通道的发送控制寄存器的位 0 清零来停止 DMA，DMA 将进入停止状态。

图 715. DMA 发送流程（标准模式）



OSP（处理第二个数据包）模式下的 DMA 发送

在运行状态下，如果将*通道 x 发送控制寄存器 (ETH_DMACxTXCR)* 的位 4 置 1，则发送过程可同时获取两个数据包，而无需关闭第一个数据包的状态描述符。发送过程完成第一个数据包的传送后，会立即轮询第二个数据包的发送描述符列表。如果第二个数据包有效，则发送过程会在写入第一个数据包的状态信息前发送此数据包。

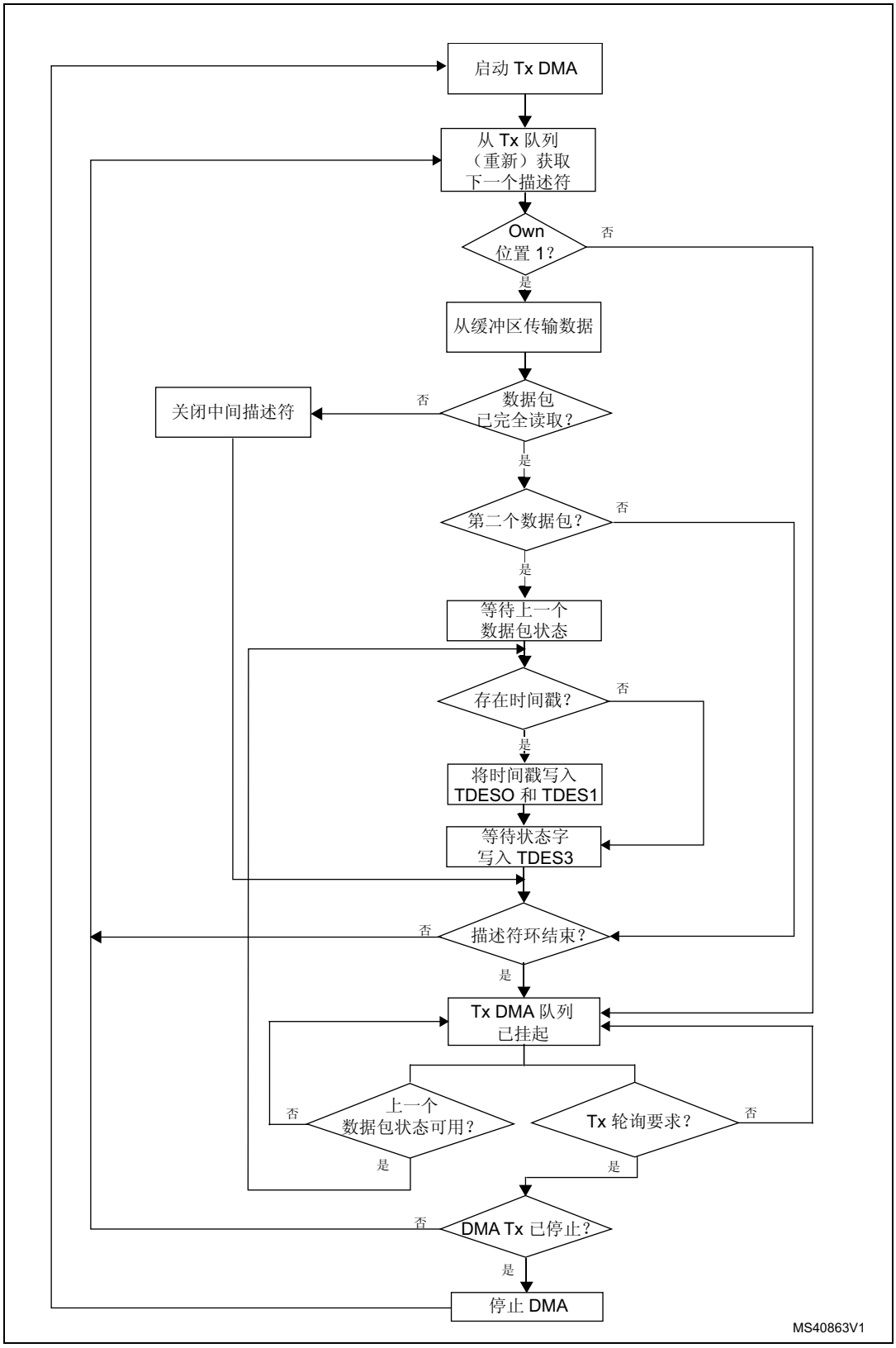
在 OSP 模式下，运行状态下的 DMA 按以下序列进行发送操作：

1. DMA 执行默认模式下 DMA 发送序列的步骤 1 到 7（请参见*默认模式下的 DMA 发送*一节）。
2. DMA 无需关闭前一数据包的最后一个描述符，即可获取下一个描述符。
3. 如果 DMA 拥有所需描述符，便会对该描述符中的发送缓冲区地址进行解码。如果 DMA 未拥有描述符，则会进入挂起模式并跳到步骤 7。
4. DMA 从系统内存中获取发送数据包并将该数据包发送至 MTL，直到发送完 EOP 数据，如果该数据包拆分到多个描述符中，则会同时关闭中间描述符。
5. DMA 等待前一个数据包的数据包发送状态和时间戳。当状态可用时，如果捕获到时间戳（由状态位指示），DMA 会将该时间戳写入 TDES0 和 TDES1。Own 位清零，DMA 将状态写入相应的 TDES3，进而关闭描述符。如果没有为前一个数据包使能时间戳功能，则 DMA 不会更改 TDES2 和 TDES3 的内容。
6. 发送中断将置 1（如果已使能）。DMA 获取下一个描述符，然后继续执行步骤 3（状态正常时）。如果上一个发送状态显示下溢错误，则 DMA 会进入挂起模式（步骤 7）。
7. 在挂起模式下，如果从 MTL 接收到挂起状态和时间戳，则 DMA 会执行以下操作：
   a) DMA 将时间戳（如果已为当前数据包使能）写入 TDES2 和 TDES3。
   a) DMA 将状态写入相应的 TDES3。
   a) DMA 将相关中断置 1 并返回挂起模式。

   如果没有挂起状态，且应用通过将相应 DMA 通道的发送控制寄存器的位 0 清零来停止 DMA，DMA 将进入停止状态。
8. 只有在接收到发送轮询要求（在相应通道的发送描述符尾指针寄存器中）后，DMA 才会退出挂起模式并进入运行状态（转到步骤 1 或步骤 2，具体取决于挂起状态）。

*图 717：接收 DMA 流程*给出了 OSP 模式下基本 DMA 发送流程的说明。

图 716. DMA 发送流程（OSP 模式）



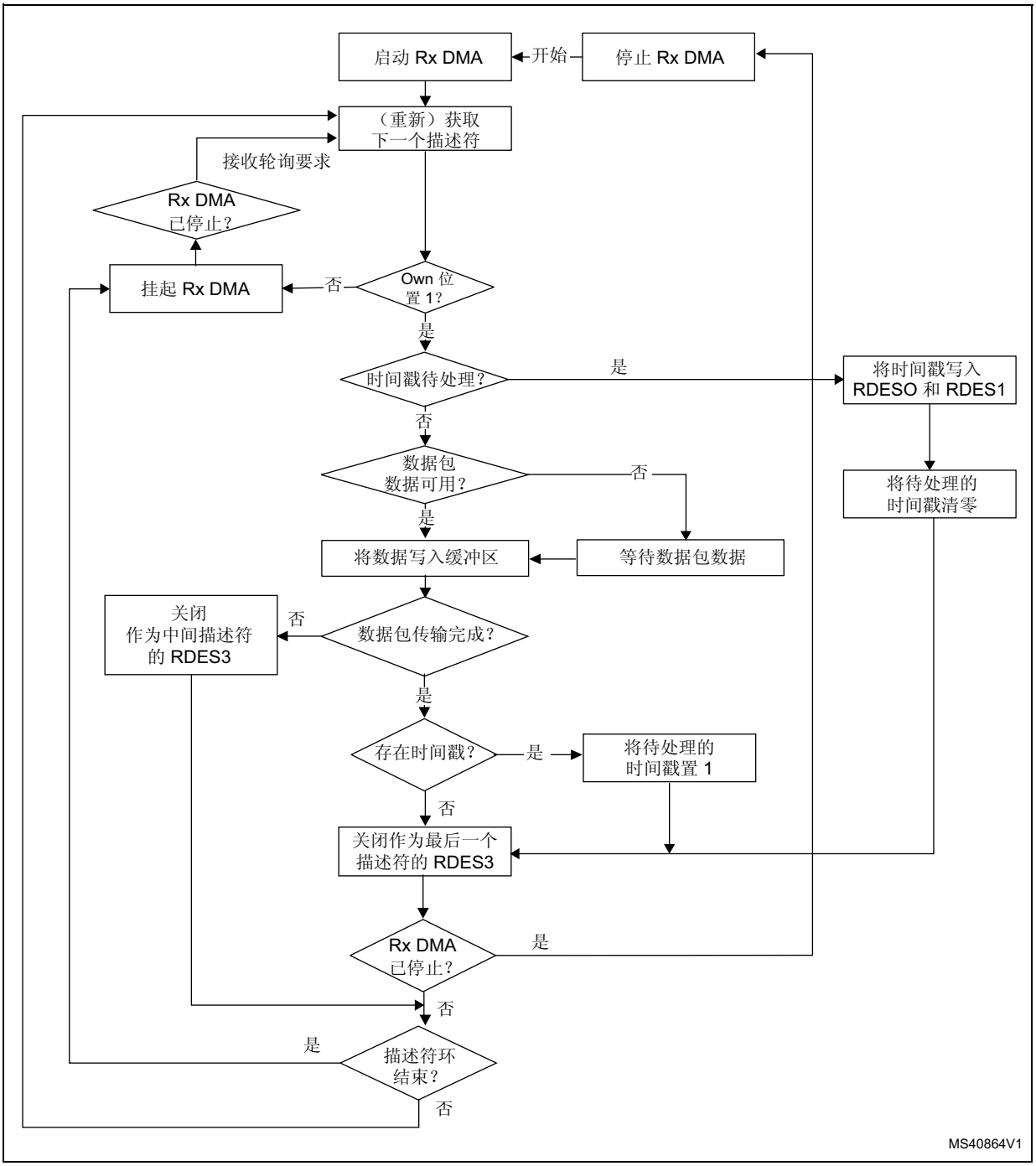
DMA 接收

在接收路径中，DMA 从 MTL 接收队列读取数据包，并将其写入相应 DMA 通道的数据包数据缓冲区。

Rx DMA 引擎的接收序列如下（另请参见*图 717：接收 DMA 流程*）：

1. 应用设置 Rx 描述符 (RDES0-RDES3) 并将 Own 位 (RDES3[31]) 置 1。应用必须在相应 DMA 通道的接收描述符尾指针寄存器中设置正确的值。
2. *通道 0 接收控制寄存器 (ETH_DMAC0RXCR)* 的位 0 置 1 时，DMA 进入运行状态。DMA 根据 Rx 当前描述符和描述符尾指针寄存器值查找空闲描述符。如果没有空闲描述符，则 DMA 通道会进入挂起状态并转到步骤 11。
3. DMA 在环中获取下一个可用描述符，并对来自所获取描述符的接收数据缓冲区地址进行解码。
4. 如果使能了 IEEE 1588 时间戳，并且该时间戳可用于前一数据包，则 DMA 会将该时间戳（如果可用）写入当前描述符的 RDES0 和 RDES1，并将 CTXT 位字段 (RDES3[30]) 置 1。
5. DMA 处理传入数据包并将其存储在所获取描述符的数据缓冲区中。
6. 如果当前数据包传输尚未完成，则 DMA 会将当前描述符作为中间描述符关闭并转至步骤 10。
7. DMA 从 MTL 获取接收帧的状态并将该状态字写入当前描述符，同时 Own 位清零且最后一个描述符位置 1。
8. DMA 将帧长度写入 RDES3，将 VLAN 标记写入 RDES0。DMA 还会将 MAC 控制帧操作码、OAM 控制帧代码和扩展状态信息（如果可用）写入最后一个描述符的 RDES1。
9. DMA 会存储此时间戳（如果可用）。DMA 在当前数据包的最后一个描述符后将上下文描述符写入下一个可用描述符中。
10. 如果 Rx DMA 描述符环中有多个描述符可用，请转到步骤 3，否则转到挂起状态（步骤 11）。
11. 当收到接收轮询要求命令并且应用使通道接收尾指针寄存器的值递增时，接收 DMA 将退出挂起状态。引擎继续执行步骤 2 并获取下一个描述符。

图 717. 接收 DMA 流程



Tx DMA 和 Rx DMA 的优先级方案

DMA 仲裁器将执行 DMA 通道 0 的 Tx 路径与 Rx 路径之间的仲裁，以访问描述符和数据缓冲区。

如果使能了给定通道的 Tx DMA 和 Rx DMA，则当该通道控制总线时，必须指定获取总线的 DMA。相应 Tx DMA 和 Rx DMA 之间的优先级可以通过 *DMA 模式寄存器 (ETH_DMAMR)* 的 TXPR 位字段进行配置。

57.4.2 MTL

MAC 事务层 (MTL) 提供 FIFO 内存接口，来缓冲和调整应用系统内存和 MAC 之间的数据包。它还可以在应用时钟和 MAC 时钟域之间实现数据传输。MTL 层具有两个 32 位宽的数据路径：发送路径和接收路径。

- 发送路径

  应用或内部 DMA 将从应用或系统内存中读取的以太网数据包推入 Tx FIFO 中的相应队列。当达到队列阈值（阈值模式）或完整数据包处于队列中（存储转发模式）时，相应数据包将被弹出并传输到 MAC。发送 EOP 后，将从 MAC 获取发送状态并传送回应用或内部 DMA。Tx 队列大小为 2048 个字节。

- 接收路径

  MTL Rx 模块从 MAC 接收数据包，并将其推入 Rx 队列。当队列超出配置的接收阈值（相应队列的工作模式寄存器中定义的 RTC 位 [1:0]）时，或者在接收到完整数据包后，会将队列的状态（填充级别）报告给应用或 DMA。MTL 还指示队列填充级别，以便 DMA 可向主接口发起预配置的突发传输。Rx 队列大小为 2048 个字节。

57.4.3 MAC

MAC 负责以太网协议处理。在发送模式下，它从 MTL 接收数据，然后再将其传送到 PHY 接口。在接收模式下，MAC 从 PHY 接口接收数据，然后再将其传送到 MTL 模块的 Rx FIFO。

本节简要介绍发送和接收序列。

MAC 发送

发送序列如下：

1. 当 MTL 应用在 SOP（数据包起始）信号置为有效的情况下推入数据时，会启动发送。
2. 检测到 SOP 信号时，MAC 接受数据并开始向 GMII 或 MII 发送。
3. 当 EOP（数据包结束）传送到 MAC 时，MAC 执行以下操作之一：
   - MAC 完成常规发送，并向 MTL 提供发送状态。
   - 如果在发送期间发生常规冲突（半双工模式下），则 MAC 会将发送状态提供给 MTL，同时重试位置 1。MAC 会提供重试请求，直到下列其中一个条件为真：

     数据包已成功发送；

     达到最大重试请求次数。在这种情况下，MAC 中止数据包发送，并处于过度冲突发送状态。MAC 接受并丢弃所有后续数据，直到接收到下一个 SOP。检测到来自 MAC 的重试请求（在状态中）时，MTL 模块应从 SOP 重新发送同一数据包。
   - 如果发生以下任一事件，则 MAC 会中止数据包发送：

     无载波（半双工模式）

     载波丢失（半双工模式）

     过度延迟（半双工模式）

     延迟冲突（半双工模式）

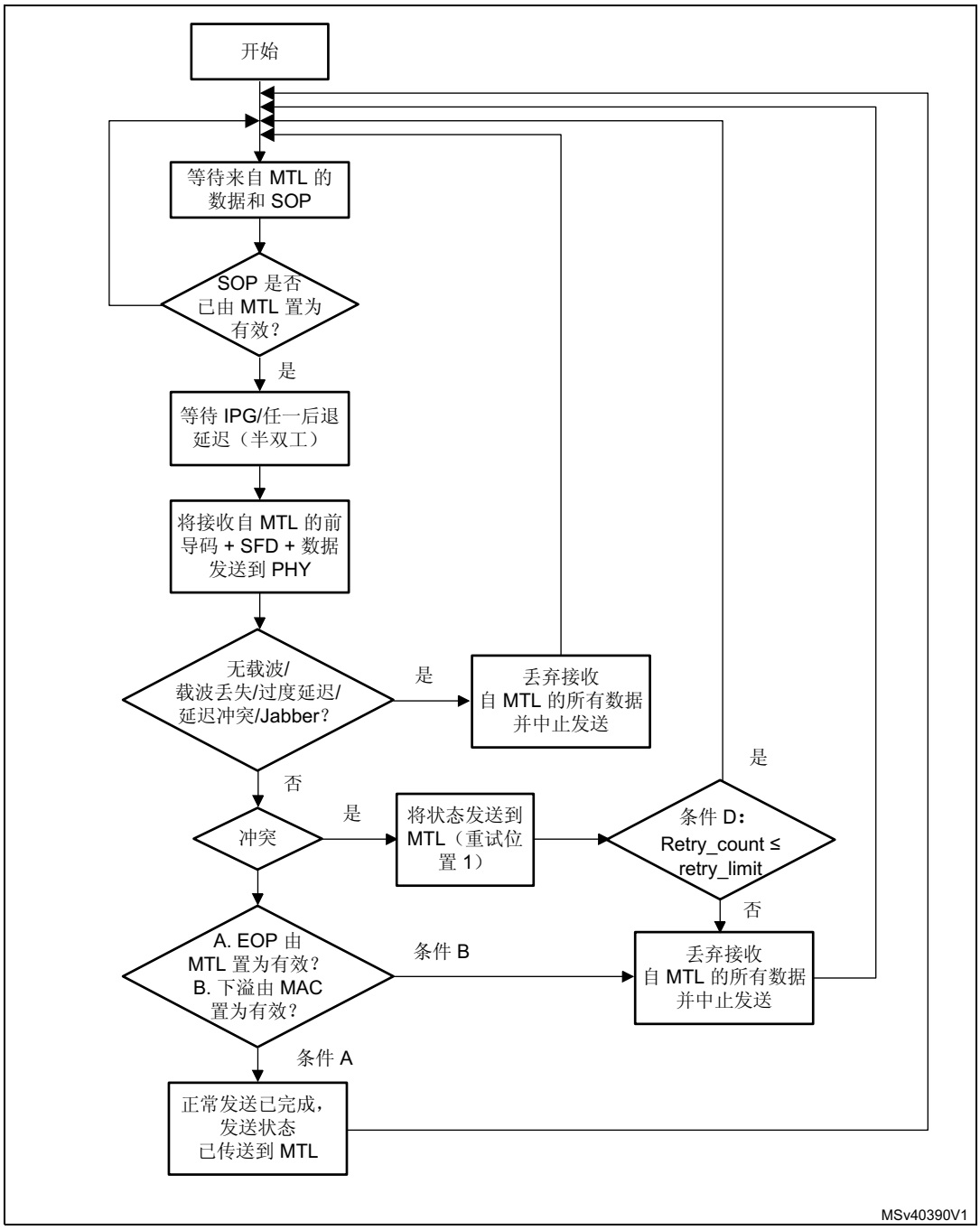
     Jabber

     MAC 接受并丢弃所有后续数据，直到接收到下一个 SOP。
4. 如果发送期间 MTL 无法连续提供数据，MAC 将发出下溢状态。MAC 接受并丢弃所有后续数据，直到接收到下一个 SOP。

5. 从 MTL 正常传输数据包期间，如果 MAC 在未获得前一数据包的 EOP 的情况下接收到 SOP，则会忽略该 SOP 并将新的数据包视为前一数据包的延续。

图 718：MAC 发送流程概览 所示为 MAC 发送过程流程。

图 718. MAC 发送流程概览



MAC 接收

当 MAC 在 GMII 上检测到 SFD 时，将启动接收操作。MAC 将去除前导码和 SFD，然后再继续处理数据包。检查报头位字段以进行过滤，FCS 位字段用于验证数据包的 CRC。接收的数据包存储在浅缓冲区中，直到执行地址过滤。如果数据包未通过地址过滤器，则会在 MAC 中将其丢弃。

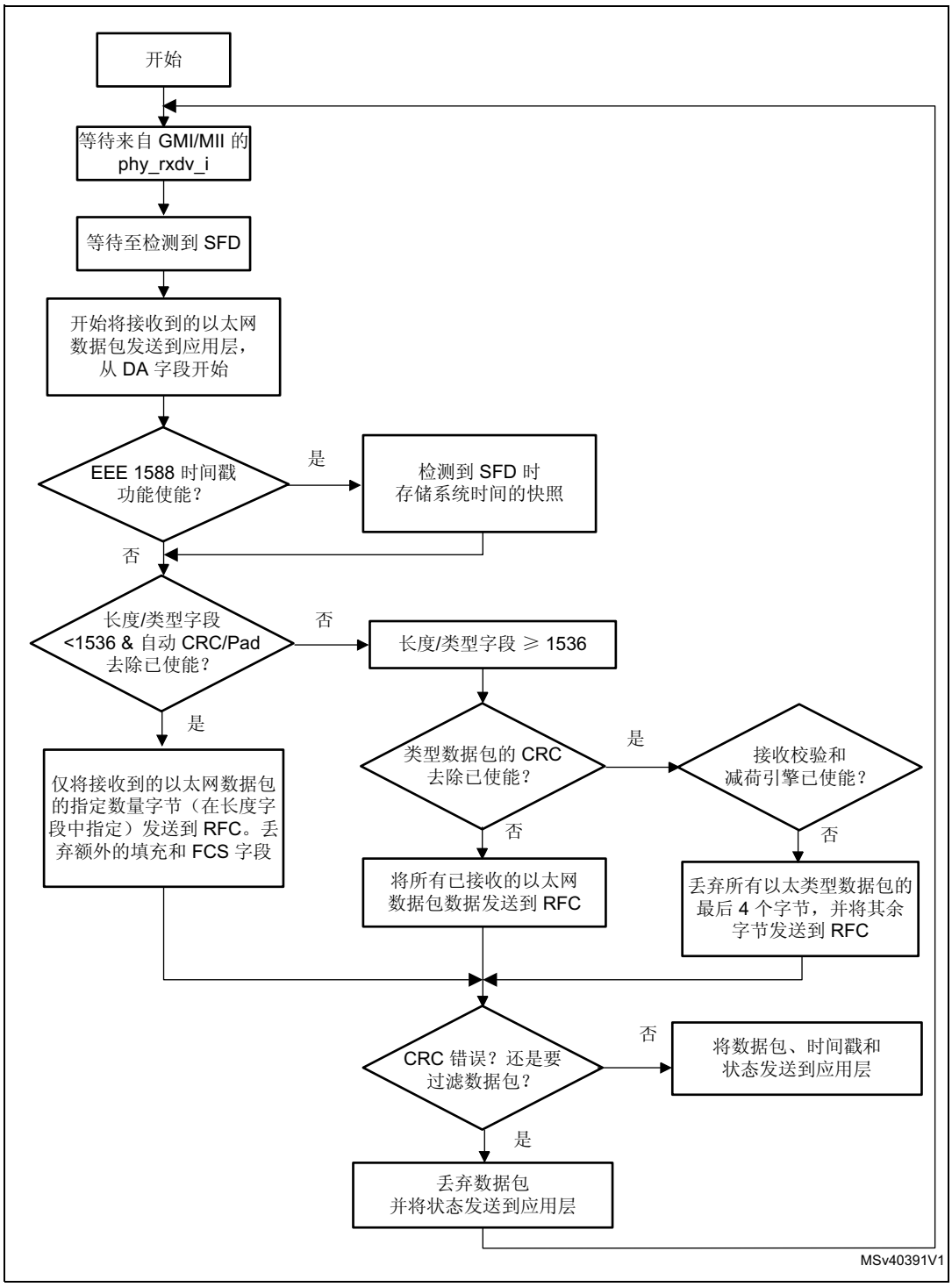
接收序列如下：

1. GMII 或 MII 的接收数据有效信号 (RxDV) 激活时，接收状态机 (RSM) 开始寻找 SFD 位字段（0x5D 字节）。

   状态机会丢弃接收到的数据包，直至检测到 SFD。
2. 检测到 SFD 时，状态机开始向 RPC 模块发送以太网数据包的数据，从 SFD 之后的第一个字节（目标地址）开始发送。
3. 如果使能了 IEEE 1588 时间戳功能，则在 GMII 或 MII 上检测到任何数据包的 SFD 时，MAC 都将获取系统时间的快照。如果在 MAC 过滤期间该数据包未被丢弃，则时间戳将传递给应用程序。接收状态机对正在接收的以太网数据包的长度/类型位字段进行解码。

   如果长度/类型位字段小于 1,536 并且为 MAC 编程了自动去除 CRC/Pad（*工作模式配置寄存器 (ETH_MACCR)* 的位 20），则状态机将发送数据包数据（数据量不超过长度/类型位字段中指定的数量），然后开始丢弃字节（包括 FCS 位字段）。状态机对长度/类型位字段进行解码并检查长度解析。
4. 当长度/类型位字段大于或等于 1,536 时，如果尚未使能*工作模式配置寄存器 (ETH_MACCR)* 位 21 中的类型数据包的 CRC 去除，则 RPE 模块会将所有接收到的以太网数据包数据发送到 RFC 模块。不过，如果已使能类型数据包的 CRC 去除，但未使能接收校验和减荷引擎，则 MAC 将去除并丢弃所有以太类型数据包的最后 4 个字节，然后将数据包转发到应用程序。
5. 默认情况下，将 MAC 编程为使能看门狗定时器，即在 RPE 模块中会切断大于 2,048（如果使能巨型数据包，则为 10,240）字节的数据包（DA + SA + LT + 数据 + PAD + FCS）。此外，还可以使用可编程看门狗定时器（*看门狗超时寄存器 (ETH_MACWTR)* 的位 16）覆盖 2,048 或 10,240 字节的固定超时。可通过编程*工作模式配置寄存器 (ETH_MACCR)* 的位 19 来禁止看门狗定时器。但是，即使禁止看门狗定时器，也仍将切断大于 32 KB 的数据包并给出看门狗超时状态。

图 719. MAC 接收流程



57.4.4 多通道和多队列支持

本章将介绍以太网外设内核支持多队列和多通道的具体方式。

系统将支持 Rx 端和 Tx 端的两个队列。Tx 端将实现两个 DMA 通道，每个队列对应一个通道，而在 Rx 路径上仅会实现一个面向两个队列的 DMA 通道。

发送路径中的多队列和多通道支持

Tx DMA 通道的数量等于 Tx 队列的数量，并且是直接的一对一映射。这使得主机/系统总线上多个 Tx DMA 的数据包/数据传输可以交错进行，且每个 Tx 队列都将被保留给相应的 Tx DMA。因此，系统将保持 Tx 队列中每个 Tx DMA 发起的数据包传输的完整性，并能高效利用主机总线进行多个 Tx DMA 之间的数据传输，而不会受到数据包长度、发送缓冲区可用性等因素的限制。

如果允许两个 Tx DMA 将数据传输到同一个 Tx 队列中，则当一个 Tx DMA 开始数据包传输时，除非该数据包传输完成，否则另一个 Tx DMA 将无法向同一个 Tx 队列传输数据。这意味着在第一个数据包传输完成之前，第二个 DMA 将保持空闲状态，这实际上就是 Tx DMA 到 Tx 队列的一对一映射。

以太网外设支持多个 Tx 通道。两个通道之间的优先级方案为固定优先级方案。在固定优先级方案中，优先级最高的通道总是能够赢得请求总线时的仲裁，如*表 522* 中所示。

表 522. DMA 通道的固定优先级方案

优先级	通道
0	通道 0
1	通道 1

由于 AXI 协议的功能和 TxDMA 的特性，固定优先级方案并不一定导致 AXI 总线较低优先级的通道“饥饿”，具体描述如下：

- 系统将在单独的 AXI 通道上驱动来自 TxDMA 且针对一次数据传输突发的读命令请求，其执行时间为一个时钟周期。必须在接受该请求后，实际的数据传输才会在单独的 AXI 通道上开始。完成这一过程需要花费多个时钟周期，且会出现相对较大的读取访问延迟。
- 即使先前的数据传输尚未完成，AXI 主设备也可以自由发出后续请求。
- 只有当与先前请求相对应的数据传输完成并传输到 MTL Tx 队列，且 Tx 队列中具备接受下一个请求所需的空间之后，TxDMA 才会请求另一次数据传输。
- 用户可对以太网外设进行编程，以控制未完成请求的最大数量、DMA 所请求的突发大小，以及 AXI 总线上每次数据传输的突发大小。

考虑到前述几点，用户可对以太网外设进行编程，从而在第一个 TxDMA 的数据传输完成并向仲裁器提出下一个请求之前，将所有 TxDMA 的请求放置在 AXI 总线上并予以执行。例如，在 64 位 AXI 数据总线中，应执行以下操作：

1. TxDMA PBL = 64（512 字节）。
2. 未完成 AXI 读请求的数量 = 16。
3. AXI 上的最大突发长度 (BLEN) = 32（256 字节）。
4. 有效 TxDMA 数量 = 8。

假设所有 TxDMA 同时提出请求并采用了固定优先级算法，则所有请求将在 32 个时钟周期的时间窗口内（每个 AXI 请求对应两个时钟周期，每个 TxDMA 请求对应两个 AXI 突发请求）放置在 AXI 总线上。第一个请求的数据传输需要花费 32 个时钟周期 + 读取访问延迟（假设存在数十个周期）。所以，在此数据传输完成之前，TxDMA 仲裁器已经处于空闲状态，并已完成了所有 TxDMA 请求的处理。

实际上，所有 TxDMA 通道都可以平等地访问 AXI 总线，且相应的 Tx 队列将会以相同的速率进行填充。系统将根据 MTL 中的流量类别调度器，从 Tx 队列中读取数据。因此在稳定状态下，每个通道的带宽都会受到 MTL 中流量调度器的优先级和设置的控制。

发送路径中多队列和多通道的编程指南

请参见*第 3106 页的第 57.9.8 节：关于多通道多队列操作的编程指南*。

接收路径中的多队列和多通道支持

以太网外设在 Rx 方向仅支持一个通道。

接收路径中多队列和多通道的编程指南

请参见*第 3106 页的第 57.9.8 节：关于多通道多队列操作的编程指南*。

分配给 Rx 队列的标记优先级选择

用户可在相应的 *Rx 队列控制 2 寄存器 (ETH_MACRXQC2R)* 中对 PSRQ 位字段进行编程，从而将 VLAN 标记优先级分配给 Rx 队列。用户可在 PSRQ 位字段中设置与 VLAN 标记优先级相对应的位，以便为 Rx 队列分配相应的优先级。例如，要将 VLAN 标记优先级 3 分配给 Rx 队列 0，请在 *Rx 队列控制 2 寄存器 (ETH_MACRXQC2R)* 中设置 PSRQ 位字段的位 3。分配给特定 Rx 队列的 VLAN 标记优先级必须唯一。这意味着不能将相同的 VLAN 标记优先级分配给多个 Rx 队列。然而，用户可以将多个 VLAN 标记优先级分配给同一个 Rx 队列。

PSRQ 位字段中的设置适用于传送到 Rx 队列的带 VLAN 标记的 Rx 数据包，以及基于 PFC 的 Tx 流控制。接收到的带 VLAN 标记的 Rx 数据包将传送到与 VLAN 标记优先级相匹配的 Rx 队列。在基于 PFC 的 Tx 流控制中，系统将使用与特定 Rx 队列相对应的 PSRQ 位字段，以使能在达到相应 Rx 队列阈值时发送的 PFC 数据包中的 VLAN 标记优先级。

Rx 端从 MAC 到队列的传送

MAC 将根据以下数据包类型将 Rx 数据包传送到 Rx 队列：

- 未通过目标地址过滤的单播/多播目标地址数据包
- 通过目标地址过滤的多播/广播目标地址数据包
- 带 VLAN 标记的 IEEE 1588 PTP over Ethernet 数据包
- 未标记的 IEEE1588 PTP over Ethernet 数据包
- 带 VLAN 标记的数据包中的 VLAN 标记优先级位字段
- 未标记的数据包

可按照相应 *Rx 队列控制 2 寄存器 (ETH_MACRXQC2R)* 的 PSRQ 位字段中分配给 Rx 队列的优先级来传送带 VLAN 标记的 Rx 数据包。可通过 *Rx 队列控制 0 寄存器 (ETH_MACRXQC0R)* 中的 RXQxEN 位字段来使能相应的 Rx 队列。

可根据 *Rx 队列控制 1 寄存器 (ETH_MACRXQC1R)* 的 PTPQ 位字段中指定的 Rx 队列号来传送未标记的 IEEE1588 PTP over Ethernet Rx 数据包。

可根据 *Rx 队列控制 2 寄存器 (ETH_MACRXQC2R)* 中通过 PSRQ[7:0] 位字段分配给 RX 队列的优先级，或是根据 *Rx 队列控制 1 寄存器 (ETH_MACRXQC1R)* 的 PTPQ[2:0] 位字段中指定的 Rx 队列号来传送带 VLAN 标记的 IEEE 1588 PTP over Ethernet Rx 数据包。可通过 *Rx 队列控制 0 寄存器 (ETH_MACRXQC0R)* 中的 RXQxEN[1:0] 位字段来使能相应的 Rx 队列。该操作需通过对 *Rx 队列控制 1 寄存器 (ETH_MACRXQC1R)* 的 TPQC[1:0] 位字段进行

编程来确定。只有当 802.1AS 模式处于禁止状态（*时间戳控制寄存器 (ETH_MACTSCR)* 中的 AV8021ASMEN 位设置为 0）时，系统才会检测到带 VLAN 标记的 IEEE 1588 PTP over Ethernet Rx 数据包，否则系统会将此类 Rx 数据包作为通用的带 VLAN 标记的 Rx 数据包进行传送。

当通过 *Rx 队列控制 1 寄存器 (ETH_MACRXQC1R)* 的 MCBCQEN 位使能时，通过目标地址过滤的多播/广播目标地址 Rx 数据包可根据 *Rx 队列控制 1 寄存器 (ETH_MACRXQC1R)* 的 MCBCQ[2:0] 位字段中指定的 Rx 队列号进行传送。可通过 *Rx 队列控制 0 寄存器 (ETH_MACRXQC0R)* 中的 RXQxEN[1:0] 位字段来使能相应的 Rx 队列。

未标记的 Rx 数据包可根据 *Rx 队列控制 1 寄存器 (ETH_MACRXQC1R)* 的 UPQ[2:0] 位字段中指定的 Rx 队列号进行传送。相应的 Rx 队列可通过 *Rx 队列控制 0 寄存器 (ETH_MACRXQC0R)* 中的 RXQxEN[1:0] 位字段进行使能。

当通过 *Rx 队列控制寄存器 (ETH_MACRXQCR)* 的 UFFQE 位和 *数据包过滤控制寄存器 (ETH_MACPFR)* 的 RA 位设置使能时，未通过目标地址过滤的单播目标地址 Rx 数据包可根据 *Rx 队列控制寄存器 (ETH_MACRXQCR)* 的 UFFQ 位字段中指定的 Rx 队列号进行传送。可通过 *Rx 队列控制 0 寄存器 (ETH_MACRXQC0R)* 中的 RXQxEN[1:0] 位字段来使能相应的 Rx 队列。

当通过 *Rx 队列控制寄存器 (ETH_MACRXQCR)* 的 MFFQE 位使能且 *数据包过滤控制寄存器 (ETH_MACPFR)* 的 RA 位为 1 时，未通过目标地址过滤的多播目标地址 Rx 数据包可根据 *Rx 队列控制寄存器 (ETH_MACRXQCR)* 的 MFFQ 位字段中指定的 Rx 队列号进行传送。相应的 Rx 队列可通过 *Rx 队列控制 0 寄存器 (ETH_MACRXQC0R)* 中的 RXQxEN[1:0] 位字段进行使能。

如果无法将 Rx 数据包作为任何已定义类型的数据包进行传送，则该数据包将通过 Rx 队列 0 进行传送。

DMA 和 MTL 之间的 Rx 端仲裁

在完成当前数据包处理后，DMA 通道控制器将获取下一个描述符。描述符获取完成后，DMA 通道控制器会根据编程的 PBL 和 Rx 缓冲区长度来评估要传输到 Rx 缓冲区的数据量，并请求 MTL 相应地传输数据。

处理完当前请求后，MTL Rx 队列仲裁方案将根据仲裁方案（*工作模式寄存器 (ETH_MTLOMR)* 的 RAA 位字段以及在 *Rx 队列 x 控制寄存器 (ETH_MTLRXQxCR)* 中编程的权重）选择一个 Rx 队列。仲裁将在 DMA 将要处理的队列之间进行。选择 Rx 队列后，系统将根据所选的 Rx 通道，从该队列读取 PBL（可编程突发长度）数量的数据，并传送到 Rx DMA 通道。

当每次 PBL 数据传输均已完成且至少有一个 DMA 通道的描述符做好了处理准备时，系统将开始仲裁。

队列模式

当为通用数据传输使能了多个 Tx 队列时，排队将基于 WRR 或 WSP 算法。

当选择多个 Rx 队列时，所有队列都基于 VLAN 标记优先级启用通用排队。VLAN 标记优先级应与 *Rx 队列控制 2 寄存器 (ETH_MACRXQC2R)* 的 PSRQ[7:0] 位字段匹配。默认情况下，系统会将未标记的数据包传送到 *Rx 队列控制 1 寄存器 (ETH_MACRXQC1R)* 的 UPQ 位字段中指定的接收队列。队列 0 是 UPQ[2:0] 位字段的默认值。可以使用任何其他值来覆盖 UPQ[2:0] 位字段的默认值。也可根据完美匹配的 MAC 地址寄存器 (*MAC 地址 x 高位寄存器 (ETH_MACAxHR)*) 的 DCS 位字段，将 Rx 数据包传送到特定的 DMA 通道。

57.5 以太网功能描述：MAC

57.5.1 双 VLAN 处理

以太网外设支持双 VLAN（虚拟 LAN）标记功能，通过该功能 MAC 可处理多达两个 VLAN 标记（内部和外部）。

MAC 支持以下操作：

- 在发送路径中插入、替换或删除最多两个 VLAN 标记
- 基于接收路径中两个 VLAN 标记的任意一个进行数据包过滤和去除。在接收路径中去除和提供最多两个 VLAN 标记（作为接收状态的一部分）

发送路径

*表 523：Tx 路径中的双 VLAN 处理功能*介绍了发送端 MAC 所支持的功能。

表 523. Tx 路径中的双 VLAN 处理功能

功能	说明
支持 C-VLAN 和 S-VLAN 标记类型	内部或外部 VLAN 标记可以是 C-VLAN 和 S-VLAN 类型。VLAN 类型分别通过 *VLAN 包含寄存器 (ETH_MACVIR)* 和*内部 VLAN 包含寄存器 (ETH_MACIVIR)* 的 CSVL 位指定。 以太网外设支持处理外部和内部 VLAN 标记的任何序列。但是，它不支持 C-VLAN S-VLAN 序列。 MAC 既不检查应用程序提供的数据包是否具有有效的 VLAN 标记类型序列，也不检查插入或替换操作是否会导致 VLAN 标记类型序列失效。因此，应用程序必须提供正确的 VLAN 标记类型序列，并对 MAC 进行编程以使发送的数据包中具有正确的 VLAN 标记类型序列。应用程序必须确保以下几点： – 使能外部 C-VLAN 标记插入时，内部标记不应为 S-VLAN。 – 使能内部 S-VLAN 标记插入时，外部标记不应为 C-VLAN。 – 用 C-VLAN 替换外部标记时，内部标记不应为 S-VLAN。 – 用 S-VLAN 替换内部标记时，外部标记不应为 C-VLAN。
VLAN 标记删除	可分别通过 *VLAN 包含寄存器 (ETH_MACVIR)* 或*内部 VLAN 包含寄存器 (ETH_MACIVIR)* 中的 VLC 位字段为外部或内部标记使能 VLAN 标记删除。使能 VLAN 删除时，MAC 会删除相应位置上的标记。数据包只有一个标记时，会将该标记视为外部标记。如果使能了内部标记删除，并且数据包只有一个标记，则 MAC 不会删除该标记。
VLAN 标记插入或替换	可分别通过 *VLAN 包含寄存器 (ETH_MACVIR)* 或*内部 VLAN 包含寄存器 (ETH_MACIVIR)* 中的 VLC 位字段为外部或内部标记使能 VLAN 标记插入或替换。使能 VLAN 标记插入或替换时，前一寄存器中的 VLTI 位用于确定是否应从寄存器或控制字中获取 VLAN 标记。

接收路径

*表 524：Rx 路径中的双 VLAN 处理功能*介绍了接收端 MAC 所支持的功能以及 *VLAN 标记寄存器 (ETH_MACVTR)* 中的相应位。

表 524. Rx 路径中的双 VLAN 处理功能

功能	说明
基于外部或内部 VLAN 标记的过滤	MAC 可基于外部或内部 VLAN 标记（通过 ERIVLT 位选择）对数据包进行过滤。
基于 C-VLAN 或 S-VLAN 标记的过滤	MAC 可基于 C-VLAN 或 S-VLAN 类型（通过 ERSVLM 位选择）对数据包进行过滤。
外部和内部 VLAN 标记去除	MAC 可基于 EVLS 和 EIVLS 位从接收到的帧中去除外部和内部 VLAN 标记。
Rx 状态中的 16 位外部和内部 VLAN 标记和类型	MAC 可分别基于 EVLRXS 和 EIVLRXS 位在 Rx 状态中提供 16 位外部和内部 VLAN 标记和类型。
禁止或跳过外部 VLAN 标记类型的检查	MAC 可基于 DOVLTC 位禁止或跳过外部 VLAN 标记类型的检查以匹配 C-VLAN 或 S-VLAN。

57.5.2 源地址和 VLAN 插入、替换或删除

源地址插入或替换

软件可使用 SA（源地址）插入或替换功能来指示 MAC 对 Tx 数据包执行以下操作：

- 在 SA 位字段中插入 MAC 地址寄存器的内容
- 使用 MAC 地址寄存器的内容替换 SA 位字段的内容

使能 SA 插入时，应用程序必须确保发送到 MAC 的数据包不包含 SA 位字段。MAC 不检查发送数据包中是否存在 SA 位字段，并且会在 SA 位字段中插入 MAC 地址寄存器的内容。类似地，使能 SA 替换时，应用程序必须确保发送到 MAC 的数据包中存在 SA 位字段。MAC 使用 MAC 地址寄存器的内容替换发送数据包中目标地址位字段后的六个字节。

可以针对所有发送数据包或选择性数据包使能 SA 插入或替换功能：

- 针对所有数据包使能 SA 插入或替换

  要针对所有数据包使能此功能，请编程*工作模式配置寄存器 (ETH_MACCR)* 的 SARC 位字段。
- 针对选择性数据包使能 SA 插入或替换

  要针对选择性数据包使能此功能，请按照如下所述对数据包第一个发送描述符的 SA 插入控制位字段（发送描述符字 3/TDES3 的位 [25:23]，请参见*第 57.10.3 节：发送描述符*）进行编程。TDES3 的位 25 置 1 时，SA 插入控制位字段指示基于 MAC 地址 1 寄存器进行的插入或替换。TDES3 的位 25 复位时，则指示基于 MAC 地址 0 寄存器进行的插入或替换。

如果未使能 MAC 地址 1 寄存器，则使用 MAC 地址 0 寄存器插入或替换 SA 插入控制位字段最高有效位的值。

VLAN 插入、替换或删除

软件可使用 VLAN 插入、替换或删除功能来指示 MAC 对 Tx 数据包执行以下操作：

- 删除 VLAN 类型和 VLAN 标记位字段
- 插入或替换 VLAN 类型和 VLAN 标记位字段

  基于 *VLAN 包含寄存器 (ETH_MACVIR)* 中 VLTI 位的设置执行插入或替换，如*表 525：基于 VLTI 位进行 VLAN 插入或替换*所述。

表 525. 基于 VLTI 位进行 VLAN 插入或替换

条件	说明
VLTI 位置 1	MAC 插入或替换以下位字段： VLAN 类型位字段（C-VLAN 或 S-VLAN，由 *VLAN 包含寄存器 (ETH_MACVIR)* 的 CSVL 位指示） VLAN 标记位字段（使用数据包发送上下文描述符的 VT 位字段）
VLTI 位复位	MAC 插入或替换以下位字段： VLAN 类型位字段（C-VLAN 或 S-VLAN，由 *VLAN 包含寄存器 (ETH_MACVIR)* 的 CSVL 位指示） VLAN 标记位字段（使用 *VLAN 包含寄存器 (ETH_MACVIR)* 的 VLT 位字段）

使能 VLAN 替换或删除时，MAC 会检查发送数据包中的 DA 和 SA 位字段之后是否存在 VLAN 类型位字段（0x8100 或 0x88A8）。如果在 DA 和 SA 位字段之后的两个字节中未检测到 VLAN 类型位字段，则不会进行替换或删除操作。不过，使能 VLAN 插入时，MAC 不会检查发送数据包中是否存在 VLAN 类型位字段，只会插入 VLAN 类型和 VLAN 标记位字段。

可针对所有 Tx 数据包或选择性数据包使能 VLAN 插入、替换或删除功能：

- 要针对所有数据包使能此功能，请编程 *VLAN 包含寄存器 (ETH_MACVIR)* 的 VLC 和 VLP 位字段。
- 要针对选择性数据包使能此功能，请对 TDES2 正常描述符的 VTIR 位字段进行编程（请参见*表 558：TDES2 正常描述符（读取格式）*）。

  此外，必须根据配置，将 *VLAN 包含寄存器 (ETH_MACVIR)*（针对外部 VLAN）和*内部 VLAN 包含寄存器 (ETH_MACIVIR)*（针对内部 VLAN）中的 VLP（VLAN 优先级控制）位复位，以使 MAC 从主机获取控制输入。

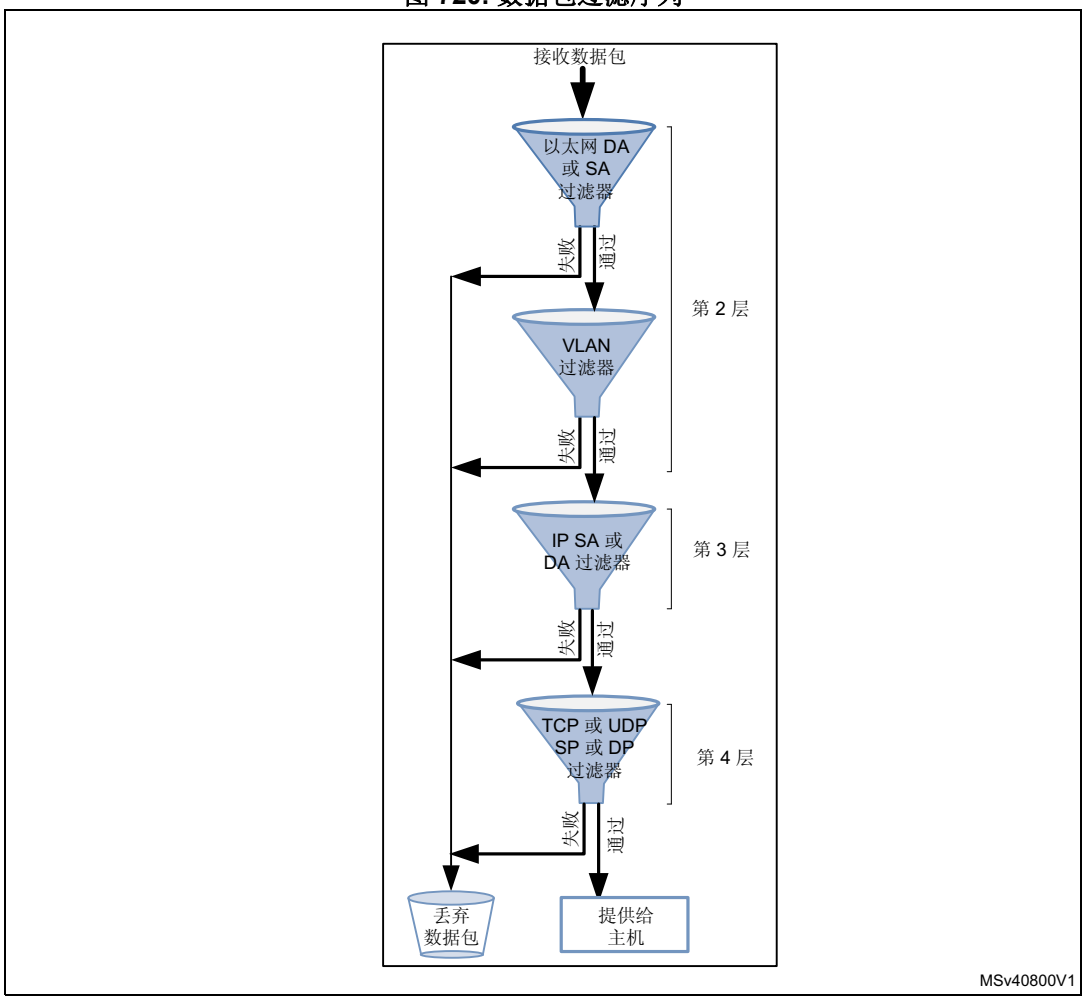
57.5.3 数据包过滤

MAC 支持对 Rx 数据包进行以下类型的过滤：

- *MAC 源地址或目标地址过滤*：地址过滤模块 (AFM) 检查每个传入数据包的源地址和目标地址位字段。
- *VLAN 过滤*：MAC 支持基于 VLAN 标记的过滤和 VLAN 散列过滤。
- *第 3 层和第 4 层过滤*：第 3 层过滤是指 IP 源地址和目标地址过滤。第 4 层过滤是指源端口和目标端口过滤。

这三种过滤类型可级联。*图 720* 所示为 Rx 数据包的过滤序列。

图 720. 数据包过滤序列



当所有过滤器（L2、VLAN、L3、L4）均已激活时，*图 720* 中显示的序列将处于有效状态。如果未使能任意一个层过滤器，则系统将绕过该过滤器，并应用后续过滤器。未通过任何过滤器的数据包都将被丢弃。然而，用户可将被丢弃的数据包转发到主机，具体取决于寄存器控制。

例如，当*数据包过滤控制寄存器 (ETH_MACPFR)* 的 RA 位置 1 时，所有被丢弃的数据包都将转发到主机，但其数据包状态会指示具体的过滤器故障。如果将 RA 位清零，则*数据包过滤控制寄存器 (ETH_MACPFR)* 的 VTFE 和 IPFE 位将控制是将未通过 VLAN 过滤器和第 3-4 层过滤器的数据包丢弃还是转发到主机。

MAC 源地址或目标地址过滤

MAC 地址过滤模块检查每个传入数据包的源地址 (SA) 和目标地址 (DA) 位字段。

单播目标地址过滤

MAC 支持 4 个用于单播完美过滤的 MAC 地址。如果选择完美过滤（复位*数据包过滤控制寄存器 (ETH_MACPFR)* 的 HUC 位），MAC 会将接收的单播地址的所有 48 位与编程的 MAC 地址进行比较来确定是否匹配。默认情况下，始终使能 MacAddr0。

MacAddr1 到 MacAddr3 地址则通过单独的使能位进行选择。与接收的相应 DA 字节进行比较时，可以将 *MAC 地址 x 高位寄存器 (ETH_MACAxHR)* 中相应的屏蔽字节控制位置 1 来屏蔽各个字节。这有助于 DA 的组地址过滤。

在散列过滤模式（HUC 位置 1）下，MAC 将使用 64 位散列表执行对单播地址的不完美过滤。对于散列过滤，MAC 将使用接收的目标地址的 6 个高 CRC 位来索引散列表的内容。值为 00000 时，选择所选寄存器的位 0；值为 11111 时，选择散列表寄存器的位 63。如果相应位（由 6 位 CRC 指示）已置 1，将认为单播数据包已通过散列过滤，否则认为数据包未能通过散列过滤。

多播目标地址过滤

要将 MAC 编程为让所有多播数据包通过，请将*数据包过滤控制寄存器 (ETH_MACPFR)* 的 PM 位置 1。如果 PM 位复位，MAC 将根据*数据包过滤控制寄存器 (ETH_MACPFR)* 的 HMC 位对多播地址执行过滤。

在完美过滤模式下，将多播地址与编程的 MAC 目标地址寄存器进行比较。组地址过滤也受到支持。

在散列过滤模式下，MAC 将使用 64 位散列表执行不完美过滤。MAC 使用接收的多播地址的 6 个高 CRC 位来索引散列表的内容。值为 000000 时，选择所选寄存器的位 0；值为 111111 时，选择散列表寄存器的位 63。如果相应的位置 1，则认为多播数据包已通过散列过滤。否则，认为数据包未通过散列过滤。

散列或完美地址过滤

要将 DA 过滤器配置为让 DA 与散列过滤器或完美过滤器匹配的数据包通过，请将*数据包过滤控制寄存器 (ETH_MACPFR)* 中的 HPF 位和相应的 HUC 或 HMC 位置 1。这对于单播数据包和多播数据包均适用。如果 HPF 位复位，则只有一种过滤器（散列或完美）应用于接收数据包。

广播地址过滤

默认情况下，MAC 不过滤任何广播数据包。要将 MAC 编程为拒绝所有广播数据包，请将*数据包过滤控制寄存器 (ETH_MACPFR)* 的 DBF 位置 1。

单播源地址过滤

MAC 可以根据所接收数据包的源地址位字段来执行完美过滤。默认情况下，MAC 将 SA 位字段与 SA 寄存器中编程的值进行比较。可通过将 *MAC 地址 x 高位寄存器 (ETH_MACAxHR)* 的位 30 置 1，来将 MAC 地址寄存器配置为使用 SA（而非 DA）进行比较。

MAC 还支持带 SA 的组地址过滤。可通过屏蔽地址的一个或多个字节来过滤一组地址。如果*数据包过滤控制寄存器 (ETH_MACPFR)* 中的 SAF 位置 1，则 MAC 会丢弃未通过 SA 过滤的数据包。否则，SA 过滤的结果将通过接收状态字中的状态位给出（参见*表 527*）。SAF 位置 1 时，对 SA 过滤和 DA 过滤结果进行与运算，以决定是否需要转发数据包。这意味着只要其中一种过滤失败，就会丢弃数据包。只有在数据包依次通过两种过滤时，才能将数据包转发到应用程序。

反向过滤

对于 DA 和 SA 过滤，可通过将*数据包过滤控制寄存器 (ETH_MACPFR)* 的 DAIF 和 SAIF 位置 1 来反转最终输出端的过滤匹配结果。DAIF 位同时适用于单播和多播 DA 数据包。在此模式下，将反转单播或多播目标地址过滤的结果。类似地，当 SAIF 位置 1 时，将反转单播 SA 过滤的结果。

表 526 和*表 527* 按照所接收数据包的类型总结了 DA 和 SA 过滤。

注： *当数据包过滤控制寄存器 (ETH_MACPFR) 的 RA 位置 1 时，所有数据包将连同 Rx 状态中正确的地址过滤结果一起转发到系统。*

表 526. 目标地址过滤

数据包类型	PR	HPF	HUC	DAIF	HMC	PM	DBF	DA 过滤操作
广播	1	X	X	X	X	X	X	通过
	0	X	X	X	X	X	0	通过
	0	X	X	X	X	X	1	不通过
单播	1	X	X	X	X	X	X	通过所有数据包
	0	X	0	0	X	X	X	完美/组过滤匹配时通过
	0	X	0	1	X	X	X	完美/组过滤匹配时不通过
	0	0	1	0	X	X	X	散列过滤匹配时通过
	0	0	1	1	X	X	X	散列过滤匹配时不通过
	0	1	1	0	X	X	X	散列或完美/组过滤匹配时通过
	0	1	1	1	X	X	X	散列或完美/组过滤匹配时不通过
多播	1	X	X	X	X	X	X	通过所有数据包
	X	X	X	X	X	1	X	通过所有数据包
	0	X	X	0	0	0	X	完美/组过滤匹配时通过，并在 PCF = 0x 时丢弃暂停数据包
	0	0	X	0	1	0	X	散列过滤匹配时通过，并在 PCF = 0x 时丢弃暂停数据包
	0	1	X	0	1	0	X	散列或完美/组过滤匹配时通过，并在 PCF = 0x 时丢弃暂停数据包
多播	0	X	X	1	0	0	X	完美/组过滤匹配时不通过，并在 PCF = 0x 时丢弃暂停数据包
	0	0	X	1	1	0	X	散列过滤匹配时不通过，并在 PCF = 0x 时丢弃暂停数据包
	0	1	X	1	1	0	X	散列或完美/组过滤匹配时不通过，并在 PCF = 0x 时丢弃暂停数据包

表 527. 源地址过滤

数据包类型	PR	SAIF	SAF	SA 过滤操作
单播	1	X	X	通过所有数据包。
	0	0	0	完美或组过滤匹配时为通过状态，但不丢弃未通过的数据包
	0	1	0	完美或组过滤匹配时为不通过状态，但不丢弃数据包
	0	0	1	完美或组过滤匹配时通过，并丢弃不通过的数据包
	0	1	1	完美或组过滤匹配时不通过，并丢弃不通过的数据包

VLAN 过滤

MAC 支持完美和散列 VLAN 过滤。有关详细的编程步骤，请参见*第 57.9.15 节：关于在接收端执行 VLAN 过滤的编程指南*。

VLAN 标记完美过滤

在 VLAN 标记完美过滤中，MAC 比较所接收数据包的 VLAN 标记，并向应用程序提供 VLAN 数据包状态。根据编程的模式，MAC 会选择比较所接收 VLAN 标记的低 12 位或全部 16 位，以确定完美匹配。

如果使能 VLAN 标记完美过滤，则 MAC 会转发 VLAN 标记的数据包以及 VLAN 标记匹配状态，并丢弃不匹配的 VLAN 数据包。还可以通过将 *VLAN 标记寄存器 (ETH_MACVTR)* 的 VTIM 位置 1 来使能 VLAN 数据包的反向匹配。此外，可通过将 *VLAN 标记寄存器 (ETH_MACVTR)* 的 ESVL 位置 1，来使能 S-VLAN 标记数据包与默认 C-VLAN 标记数据包的匹配操作。VLAN 数据包状态位（RDES0 的位 10）指示匹配数据包的 VLAN 标记匹配状态。

注： *源地址或目标地址（如果已使能）的优先级高于 VLAN 标记过滤器。这意味着无论 VLAN 标记过滤结果如何，未通过源地址或目标地址过滤的数据包都会被丢弃。*

VLAN 标记散列过滤

16 位 VLAN 散列表将用于基于 VLAN 标记进行组地址过滤。用户可使用 *VLAN 标记寄存器 (ETH_MACVTR)* 的 VTHM（VLAN 标记散列表匹配使能）位来使能 VLAN 标记散列过滤功能。如果 VTHM 位置 1，则使用 VLAN 标记的 CRC-32 的四个最高有效位来索引 VLAN 散列表寄存器的内容。VLAN 散列表寄存器的值 1 对应于索引，表示数据包的 VLAN 标记匹配，应转发数据包。值 0 则表示应丢弃 VLAN 标记的数据包。

注： *根据 ETH_MACVTR 寄存器中的 ETV 位，考虑使用 VLAN 标记的 16 位或 12 位进行 CRC-32 计算。*

ETV 位复位时，VLAN 标记的 CRC-32 的四个最高有效位被反转，并用于索引 VLAN 散列表寄存器 (ETH_MACVHTR) 的内容。

ETV 位置 1 时，VLAN 标记的 CRC-32 的四个最高有效位直接用于索引 VLAN 标记寄存器 (ETH_MACVTR) 的内容。

MAC 还支持 VLAN 数据包的反向匹配。在反向匹配模式下，当数据包的 VLAN 标记与完美或散列过滤匹配时，应丢弃该数据包。如果使能了 VLAN 完美匹配和 VLAN 散列匹配，则在 VLAN 散列过滤或 VLAN 完美过滤匹配时，会将数据包视为匹配。如果反向匹配置 1，则只有在完美和散列过滤指示不匹配时才转发数据包。

表 528 列出了 VLAN 匹配结果和最终 VLAN 匹配状态的各种可能情况。*数据包过滤控制寄存器 (ETH_MACPFR)* 的 RA 位置 1 时，即表示接收到所有数据包，并在 *RDES2 正常描述符（回写格式）*的 VF 位中指示 VLAN 匹配状态。RA 位未置 1 且*数据包过滤控制寄存器 (ETH_MACPFR)* 的 VTFE 位置 1 时，如果最终 VLAN 匹配状态为未通过，则会丢弃数据包。在*表 528* 中，值 *X* 表示该列可为任一值。

在 *VLAN 标记寄存器 (ETH_MACVTR)* 的 VL 位字段中将 VLAN VID 编程为 0 时，所有 VLAN 标记的数据包均被视为完美匹配，但 VLAN 散列匹配的状态取决于 *VLAN 标记寄存器 (ETH_MACVTR)* 中的 VTHM 和 VTIM 位。

表 528. VLAN 匹配状态(1)

VID	VLAN 完美过滤匹配结果	VTHM 位	VLAN 散列过滤匹配结果	VTIM 位	最终 VLAN 匹配状态
VID = 0	通过	0	X	X	通过
	通过	1	X	0	通过
	通过	1	不通过	1	通过
	通过	1	通过	1	不通过
VID!= 0	通过	X	X	0	通过
	不通过	0	X	0	不通过
	不通过	1	不通过	0	不通过
	不通过	1	通过	0	通过
	不通过	0	X	1	通过
	通过	X	X	1	不通过
	不通过	1	通过	1	不通过
	不通过	1	不通过	1	通过

1. 在此表中，X 表示任一值。

第 3 层和第 4 层过滤

MAC 支持基于第 3 层和第 4 层的数据包过滤。第 3 层过滤是指 IPv4 或 IPv6 数据包中的 IP 源地址或目标地址过滤，而第 4 层过滤则指 TCP 或 UDP 中的源端口号或目标端口号过滤。

第 3 层和第 4 层数据包过滤功能可自动使能接收端的 IPC 全校验和减荷引擎。对于第 3 层或第 4 层过滤操作，必须将 *工作模式配置寄存器 (ETH_MACCR)* 的 IPC 位置 1，以使能 Rx 校验和减荷引擎。

使能第 3 层和第 4 层过滤后，将按以下方式过滤数据包：

- 匹配的数据包

  MAC 将与所有已使能位字段匹配的数据包连同状态一起转发给应用程序。只有在 *工作模式配置寄存器 (ETH_MACCR)* 的 IPC 位置 1 且以下条件之一为真时，MAC 才会提供匹配的位字段状态：

  – 所有已使能的第 3 层和第 4 层位字段均匹配。

  – 至少有一个已使能的位字段匹配且其他位字段被绕过或禁止

  使能多个第 3 层和第 4 层过滤器时，任一过滤器匹配都将被视为匹配。如果多个过滤器匹配，则 MAC 会提供最低级过滤器的状态，其中过滤器 0 为最低级过滤器，过滤器 3 为最高级过滤器。例如，如果过滤器 0 和过滤器 1 均匹配，则 MAC 会提供过滤器 0 对应的状态。

注： *源地址或目标地址以及 VLAN 标记过滤器（如果已使能）的优先级高于第 3 层和第 4 层过滤器。这意味着无论第 3 层和第 4 层过滤结果如何，未通过源地址或目标地址或者 VLAN 标记过滤的数据包都会被丢弃。*

- 不匹配的数据包

  MAC 会丢弃与所有已使能位字段均不匹配的数据包。可使用反向匹配功能阻止或丢弃具有特定 TCP over IP 或 UDP over IP 位字段的数据包，并转发所有其他数据包。系统会将中止或部分数据包丢弃到 MTL Rx FIFO 中。如果 Rx FIFO 在阈值（直通）模式下运行，则系统会将阈值编程为一个较小值。系统会在得到未通过第 3 层和第 4 层过滤器的结果之前，开始将此类数据包传输到应用程序，且应用程序可能会接收到一个具有适当中止状态的部分数据包。

- 非 TCP 或 UDP IP 数据包

  默认情况下，所有非 TCP 或 UDP IP 数据包均从第 3 层和第 4 层过滤器绕过。可以将 MAC 编程为丢弃所有非 TCP over IP 或 UDP over IP 数据包。

第 3 层过滤

MAC 支持 IP 源地址和目标地址的完美匹配或反向匹配。此外，可以匹配完整的 IP 地址或屏蔽低位匹配，即比较地址中除指定低屏蔽位之外的所有位。

对于 IPv6 数据包过滤，可以使能寄存器组的后四个数据寄存器，以包含 128 位 IP 源地址或 IP 目标地址。IP 源地址或目标地址应按照 IPv6 规范中定义的顺序进行编程，即，所接收数据包中的 IP 源地址或目标地址的第一个字节位于寄存器的高字节中，后续寄存器中的顺序也同样如此。

对于 IPv4 数据包过滤，可以使能寄存器组的第二个和第三个数据寄存器，以包含 32 位 IP 源地址和 IP 目标地址。其余两个数据寄存器保留。IP 源地址或目标地址应按照 IPv4 规范中定义的顺序进行编程，即，所接收数据包中的 IP 源地址或目标地址的第一个字节位于相应寄存器的高字节中。

第 4 层过滤

MAC 支持 TCP 或 UDP 源端口号和目标端口号的完美匹配或反向匹配。不过，一次只能编程一种类型（TCP 或 UDP）。第一个数据寄存器包含 TCP 或 UDP 的 16 位源端口号和目标端口号，即，低 16 位用于源端口号，高 16 位用于目标端口号。

TCP 或 UDP 源端口号和目标端口号应按照 TCP 或 UDP 规范中定义的顺序进行编程，即，所接收数据包中的 TCP 或 UDP 源端口号和目标端口号的第一个字节位于寄存器的高字节中。

第 3 层和第 4 层过滤器寄存器组

MAC 为基于第 3 层和第 4 层的数据包过滤实现了两组寄存器。在其中一个寄存器组中，存在一个控制寄存器（例如，*L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R)*），用于控制数据包过滤。此外，还存在五个地址寄存器，用于编程要匹配的第 3 层和第 4 层位字段，例如：

- *第 4 层地址过滤器 0 寄存器 (ETH_MACL4A0R)*
- *第 3 层地址 0 过滤器 0 寄存器 (ETH_MACL3A00R)*
- *第 3 层地址 1 过滤器 0 寄存器 (ETH_MACL3A10R)*
- *第 3 层地址 2 过滤器 0 寄存器 (ETH_MACL3A20R)*
- *第 3 层地址 3 过滤器 0 寄存器 (ETH_MACL3A30R)*

另一个独立的寄存器组包括：*L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R)*、*第 4 层地址过滤器 1 寄存器 (ETH_MACL4A1R)*、*第 3 层地址 0 过滤器 1 寄存器 (ETH_MACL3A01R)*、*第 3 层地址 1 过滤器 1 寄存器 (ETH_MACL3A11R)*、*第 3 层地址 2 过滤器 1 寄存器 (ETH_MACL3A21R)* 和 *第 3 层地址 3 过滤器 1 寄存器 (ETH_MACL3A31R)*。

57.5.4 IEEE 1588 时间戳支持

IEEE 1588 标准定义了精密时间协议 (PTP)，此协议支持使用网络通信、局域计算和分布式对象等技术实现的测量和控制系统中的精密时钟同步。PTP 适用于利用支持多播消息传送的局域网（包括但不限于以太网）进行通信的系统。该协议可对非均匀系统进行同步，这类系统包含固有精度、分辨率和稳定性都不断变化的时钟。该协议支持亚秒范围的系统级同步精度，并且需要极少的网络和本地时钟计算资源。

本章包含以下章节：

- *IEEE 1588 时间戳支持*
- *IEEE 1588 系统时间源*
- *IEEE 1588 辅助快照*
- *每秒脉冲数灵活的输出*
- *PTP 时间戳减荷功能*
- *单步时间戳*

IEEE 1588 时间戳支持

以太网外设支持 IEEE 1588-2002（版本 1）和 IEEE 1588-2008（版本 2）。IEEE 1588-2002 支持通过 UDP/IP 传输的 PTP。IEEE 1588 2008 支持通过以太网传输的 PTP。外设为这两种标准提供了可编程支持。它支持下列功能：

- 支持两种时间戳格式
- 可获取所有数据包或仅 PTP 型数据包的快照
- 可获取仅事件消息的快照
- 可基于时钟类型获取快照：普通、边界、端对端透明和点对点透明
- 可将节点选作普通和边界时钟的主节点或从节点
- 识别数据包中直接通过以太网发送的 PTP 消息类型、版本和 PTP 有效负载，并发送状态
- 支持数字或二进制格式的测量亚秒级时间

时钟类型

MAC 支持 IEEE 1588-2008 规范中定义的以下时钟类型：

- 普通时钟

  域的普通时钟支持协议的单一副本。它具有一个 PTP 状态和一个物理端口。在典型工业自动化应用中，普通时钟与传感器或致动器等应用设备相关联。在电信应用中，普通时钟可与时序划分设备相关联。

  普通时钟可以是主时钟或从时钟。它支持下列功能：

  – 发送和接收 PTP 消息。时间戳快照可以按*时间戳控制寄存器 (ETH_MACTSCR)* 中所述进行控制。
  – 保持数据集，例如时间戳值。

  下表显示了可针对主节点和从节点在接收端拍摄时间戳快照时的消息。

表 529. 普通时钟：用于快照的 PTP 消息

主设备	从
Delay_Req	SYNC

对于普通时钟，可以拍摄以下 PTP 消息类型之一的快照：版本 1 或版本 2。不能同时拍摄两种 PTP 消息类型的快照。可通过在*时间戳控制寄存器 (ETH_MACTSCR)* 中将 TSVER2ENA 位置 1 并选择快照模式来拍摄快照。

- 边界时钟

  通常，边界时钟具有多个可与网络进行通信的物理端口。在边界时钟的协议引擎中，与同步、主从层级和信号传输结束相关的消息不会被转发。MAC 提供的 PTP 消息类型状态有助于识别消息的类型并采取适当的操作。

  除以下功能外，边界时钟与普通时钟类似：

  – 时钟数据集对边界时钟的所有端口通用。

  – 本地时钟对边界时钟的所有端口通用。

- 端对端透明时钟

  端对端透明时钟支持从时钟和主时钟之间的端对端延迟测量机制。端对端透明时钟转发所有消息，例如普通网桥、路由器或中继器。PTP 数据包的停留时间是指 PTP 数据包从入站端口传输到出站端口所用的时间。

  端对端透明时钟内的 SYNC 数据包的停留时间于发送前在相关 Follow_Up PTP 数据包的校准位字段中进行更新。类似地，端对端透明时钟内的 Delay_Req 数据包的停留时间于发送前在相关 Delay_Resp PTP 数据包的校准位字段中进行更新。因此，仅需在入站端口和出站端口为*表 530* 中所述的消息拍摄快照即可。可通过将*时间戳控制寄存器 (ETH_MACTSCR)* 中的 SNAPTYPSEL 位设为 10 来拍摄快照。

表 530. 端对端透明时钟：用于快照的 PTP 消息

PTP 消息
SYNC
Delay_Req

- 点对点透明时钟

  在点对点透明时钟中，链路延迟的计算基于与链路点的 Pdelay_Req、Pdelay_Resp 和 Pdelay_Resp_Follow_Up 消息的交换。

  点对点透明时钟对 PTP 时序消息的更正和处理方式与端对端透明时钟不同。除此之外，两类时钟基本相同。

  Pdelay_Req 和相关 Pdelay_Resp 数据包的停留时间相加并被插入到相关 Pdely_Resp_Followup 数据包的校准位字段中。因此，添加了针对 Pdelay 相关事件消息拍摄快照的支持，如*表 538* 所示。

表 531. 点对点透明时钟：用于快照的 PTP 消息

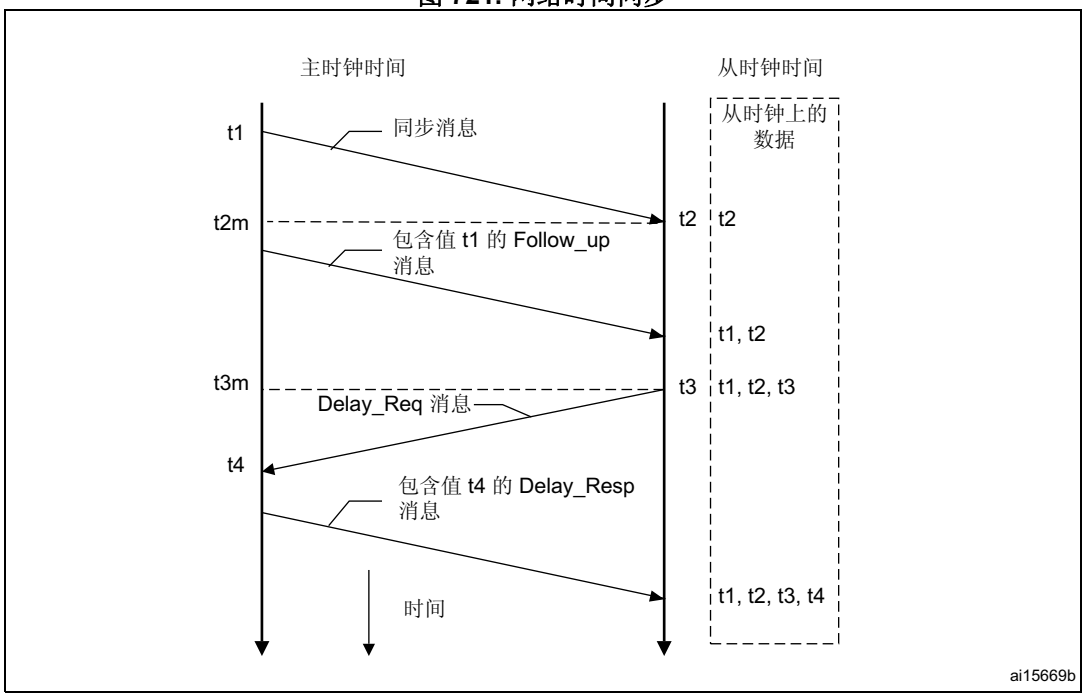
PTP 消息
SYNC
Pdelay_Req
Pdelay_Resp

可通过将*时间戳控制寄存器 (ETH_MACTSCR)* 中的 SNAPTYPESEL 位设为 11 来拍摄快照。

延迟请求响应机制

系统或网络归类为主节点和从节点，用于分配时序和时钟信息。*图 721* 显示了 PTP 通过交换 PTP 消息将从节点同步到主节点的过程。

图 721. 网络时间同步



如*图 721* 所示，PTP 使用以下过程：

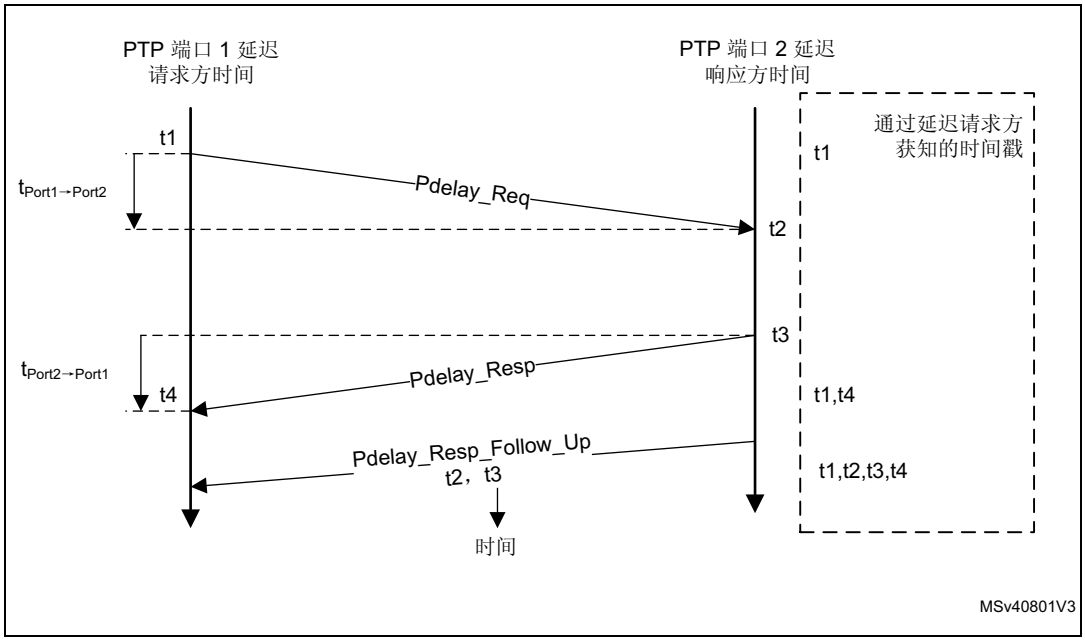
1. 主节点向其所有节点广播 PTP 同步消息。同步消息包含主节点的参考时间信息。该消息在时间 t_1 处离开主节点的系统。对于以太网端口，必须通过 GMII 或 MII 来捕获该时间。
2. 从节点接收同步消息并利用其参考时序捕获准确时间 t_2。
3. 主节点向从节点发送一个包含 t_1 信息的 Follow_up 消息以备后用。
4. 从节点向主节点发送一个 Delay_Req 消息，并标出该数据包离开 GMII 或 MII 接口的准确时间 t_3。
5. 主节点接收该消息，捕获消息进入其系统的准确时间 t_4。
6. 主节点将 Delay_Resp 消息中的 t_4 信息发送给从节点。
7. 从节点使用 t_1、t_2、t_3 和 t_4 这四个值来同步其本地参考时序与主节点的参考时序。

大多数 PTP 都在 UDP 层之上通过软件实现。不过，要通过 GMII 或 MII 接口来捕获特定 PTP 包进入或离开以太网端口时的准确时间，需要硬件支持。必须捕获该时序信息并将其返回给软件，以实现高精度的 PTP。

点对点 PTP 透明时钟 (P2P TC) 消息支持

除同步、延迟请求、后续消息和延迟响应消息外，IEEE 1588-2008 标准还支持点对点 PTP (Pdelay) 消息。*图 722* 所示为支持点对点路径校准的时钟的传播延迟计算方法。

图 722. 支持点对点路径校准的时钟的传播延迟计算



如*图 722* 所示，传播延迟的计算过程如下：

1. 端口 1 发出 Pdelay_Req 消息，并生成 Pdelay_Req 消息的时间戳 (t_1)。
2. 端口 2 接收 Pdelay_Req 消息，并生成该消息的时间戳 (t_2)。
3. 端口 2 返回 Pdelay_Resp 消息，并生成该消息的时间戳 (t_3)。
   为了最大限度减少因两个端口之间的频偏而引起的误差，端口 2 会在接收到 Pdelay_Req 消息之后尽快返回 Pdelay_Resp 消息。端口 2 返回以下任意一项：
   - Pdelay_Resp 消息中的时间戳 t_2 和 t_3 之差
   - Pdelay_Resp_Follow_Up 消息中的时间戳 t_2 和 t_3 之差
   - Pdelay_Resp 和 Pdelay_Resp_Follow_Up 消息中分别对应的时间戳 t_2 和 t_3
4. 端口 1 在接收到 Pdelay_Resp 消息时生成时间戳 (t_4)。
5. 端口 1 使用全部四个时间戳来计算平均链路延迟。

时间戳校准

根据 IEEE 1588 规范，必须在消息时间戳点（紧随起始帧分隔符八位字节之后的八位字节的第一个位的前沿）跨越节点和网络之间的边界时捕获时间戳。

因为 MAC 会在距离节点和网络的实际边界较远的内部点处捕获时间戳，所以系统会对该捕获的时间戳进行校准/更新，以反映入站/出站路径延迟（包括 PHY 层中的延迟）。为了纠正因捕获点的时钟（GMII Tx、Rx 时钟）与用于生成时间的 PTP 时钟 (clk_ptp_ref_i) 不同而导致的不精确/误差，可以进行进一步的校准。所得到的 CDC（时钟域跨越）电路将增加一个误差，具体取决于 GMII 和 PTP 时钟的时钟周期。

入站校准

在接收端，与在端口边界处接收该数据包 SFD 位的时间相比，在内部快照点捕获的时间戳会有所延迟（时间较晚）。因此，必须通过入站延迟和 CDC 采样中的误差来缩小捕获的时间戳。该校准值必须由软件进行确定/计算，并写入*时间戳入站校准纳秒寄存器 (ETH_MACTSICNR)*。

校准值由以下三个部分组成：

1. 边界点与内核输入之间 PHY 层中的外部延迟

   如果 PHY 满足符合 IEEE 802.3 条款 45 的 MMD 寄存器的相关要求，则表示 PHY 具备可指示最大和最小入站延迟的寄存器。软件可以读取这些寄存器，并确定 PHY 中的平均入站延迟。否则（如果 PHY 不支持这些寄存器），必须根据 PHY 数据手册或时序特性来确定入站延迟。

2. 从内核输入到内部捕获点的内部延迟

   延迟会根据激活的 PHY 接口（如 MII 或 RMII）和运行速度的不同而有所出入，如*表 532* 中所示。

3. CDC 同步

   CDC 同步误差几乎等同于 PTP 时钟 (clk_ptp_ref_i) 时钟周期的两倍。

应通过软件添加由这三个部分确定的值，且必须将其写入*时间戳入站校准纳秒寄存器 (ETH_MACTSICNR)* 的 TSIC 位字段。

注： *写入寄存器的值必须是负数（二进制补码），原因是必须将其从捕获的时间戳中减去。MAC 接收器会将此寄存器中的值添加到捕获的时间戳中，然后将得到的值作为已接收数据包的时间戳进行提供。*

当*时间戳控制寄存器 (ETH_MACTSCR)* 中的 TSCTRLSSR 位置 1 时，所捕获时间戳的纳秒位字段将采用十进制格式以 1 ns 的粒度表示。因此，必须将 TSIC 的位 31 置 1（对应负值），且必须使用以二进制表示的“10^9 – total ingress_correction_value[nanosecond part]”对位 30 到位 0 进行编程。例如，如果所需校准值为 -5 ns，则该值为 0xBB9A C9FB。

当*时间戳控制寄存器 (ETH_MACTSCR)* 中的 TSCTRLSSR 位复位时，所捕获时间戳的纳秒位字段将采用二进制格式以大约 0.466 ns 的粒度表示。因此，位[30:0] 必须使用以二进制表示的“2^{31} – total ingress_correction_value”进行写入，其中位[31] = 1。

出站校准

在发送端，与在端口边界处输出该数据包 SFD 位的时间相比，在内部快照点捕获的时间戳会更早（时间提前）。因此，必须通过出站延迟和 CDC 采样中的误差来补偿捕获的时间戳。该校准值必须由软件进行确定/计算，并写入*时间戳出站校准纳秒寄存器 (ETH_MACTSECNR)*。

校准值由以下三个部分组成：

1. 内核输出与网络和端口边界之间 PHY 层中的外部延迟

   如果 PHY 满足符合 IEEE 802.3 条款 45 的 MMD 寄存器的相关要求，则表示 PHY 具备可指示最大和最小出站延迟的寄存器。软件可以读取这些寄存器，并确定 PHY 中的平均出站延迟。否则（如果 PHY 不支持这些寄存器），必须根据 PHY 数据手册或时序特性来确定出站延迟。

2. 从内部捕获点到内核输出的内部延迟

   延迟会根据激活的 PHY 接口（RMII、MII 等）和运行速度的不同而有所出入，如*表 532* 中所示。

3. CDC 同步误差

   对于因同步而进行的时间戳校准，请通过添加以下公式来进行补偿：

   $$EGRESS_SYNC_CORR = (1 \times PTP_CLK_PER + 4 \times TX_CLK_PER)$$

表 532 列出了各种 PHY 接口的出站和入站延迟值：

表 532. PHY 接口的出站和入站延迟

PHY 接口		出站延迟	入站延迟
RGMII	1 Gbps	12	12
RGMII	100 Mbps	40	40
RGMII	10 Mbps	400	400
RMII	100 Mbps	40	120
RMII	10 Mbps	400	800

参考时序时钟的频率范围

时间戳信息会在异步时钟域之间传输，从 MAC 时钟域传输到应用程序时钟域。因此，需要在两次连续捕获时间戳之间应用最小延迟。该延迟为 GMII 或 MII 的四个时钟周期以及 PTP 时钟的三个时钟周期。如果两次捕获时间戳之间的延迟小于此延迟，则 MAC 将不会获取第二个数据包的时间戳快照。

PTP 时钟频率的限制如下：

- 最大 PTP 时钟频率

  最大 PTP 时钟频率将受到参考时间最大分辨率（100 MHz 下为 10 ns）的限制。此外，参考时间源的分辨率或粒度还将决定同步的精度。因此，更高的 PTP 时钟频率能够提供更好的系统性能。

- 最小 PTP 时钟频率

  最小 PTP 时钟频率取决于两个连续 SFD 字节之间所需的时间，以及与 GMII 或 MII 时钟域进行时间同步所需的时间。该关系可通过以下公式得出：

  3 * PTP 时钟周期 + 4 * GMII/MII 时钟周期 ≤ 两个 SFD 之间的最小间隙

  GMII 或 MII 时钟频率将由 IEEE 规范进行固定。因此，正常工作所需的最小 PTP 时钟频率将取决于 MAC 的工作模式和运行速度，如*表 533* 中所示。

表 533. 最小 PTP 时钟频率示例

模式	两个 SFD 之间的最小间隔	具有内部时间戳的最小 PTP 频率
10 Mbps 全双工	168 个 MII 时钟周期 （128 个 64 字节数据包的时钟周期 + 24 个最小 IFG 的时钟周期 + 16 个前导码的时钟周期）	5 MHz
10 Mbps 半双工	48 个 MII 时钟周期 （由于发生冲突而在 SFD 之后立即发送的 JAM 信号的 8 个时钟周期 + 24 个 IFG 的时钟周期 + 16 个前导码的时钟周期）	5 MHz
100 Mbps 全双工	168 个 MII 时钟周期 （128 个 64 字节数据包的时钟周期 + 24 个最小 IFG 的时钟周期 + 16 个前导码的时钟周期）	5 MHz
100 Mbps 半双工	48 个 MII 时钟周期 （由于发生冲突而在 SFD 之后立即发送的 JAM 信号的 8 个时钟周期 + 24 个 IFG 的时钟周期 + 16 个前导码的时钟周期）	5 MHz

表 533. 最小 PTP 时钟频率示例（续）

模式	两个 SFD 之间的最小间隔	具有内部时间戳的最小 PTP 频率
1000 Mbps 全双工	84 个 GMII 时钟周期 （64 个 64 字节数据包的时钟周期 + 12 个最小 IFG 的时钟周期 + 8 个前导码的时钟周期）	5 MHz
1000 Mbps 半双工	24 个 GMII 时钟周期 （由于发生冲突而在 SFD 之后立即发送的 JAM 信号的 4 个时钟周期 + 12 个 IFG 的时钟周期 + 8 个前导码的时钟周期）	18.75 MHz

PTP 处理和控制

表 534 中显示了 PTP 消息的通用消息报头。该格式取自 IEEE 1588-2008 规范。

表 534. IEEE 1588-2008 中定义的消息格式

位								八位字节	偏移
7	6	5	4	3	2	1	0		
transportSpecific				messageType				1	0
保留				versionPTP				1	1
messageLength								2	2
domainNumber								1	4
保留								1	5
flagField								2	6
correctionField								8	8
保留								4	16
sourcePortIdentity								10	20
sequenceId								2	30
controlField(1)								1	32
logMessageInterval								1	33

1. controlField 将在版本 1 中使用。在版本 2 中，messageType 位字段将用于检测不同的消息类型。

以太网有效负载的部分位字段可用于检测 PTP 数据包类型，以及控制要拍摄的快照。这些位字段在以下 PTP 数据包中有所不同：

- 通过 IPv4 发送的 PTP 数据包
- 通过 IPv6 发送的 PTP 帧
- 通过以太网发送的 PTP 数据包

通过 IPv4 发送的 PTP 数据包

表 535 中提供了用于控制通过 UDP over IPv4 发送的 PTP 数据包快照的匹配位字段相关信息，适用于 IEEE 1588 版本 1 和 2。带标记数据包的八位字节位置偏移为 4。该参数基于“IEEE 1588-2008，附录 D”以及*表 534* 中所定义的消息格式。

表 535. IEEE 1588-2008 中定义的消息格式

匹配位字段	八位字节位置	匹配值	说明
MAC 数据包类型	12、13	0x0800	IPv4 数据报
IP 版本和报头长度	14	0x45	IP 版本为 IPv4
第 4 层协议	23	0x11	UDP
IP 多播地址 （IEEE 1588 版本 1）	30、31、32、33	0xE0、0x00、0x01、0x81 （或是 0x82、0x83 或 0x84）	允许的多播 IPv4 地址： 224.0.1.129 224.0.1.130 224.0.1.131 224.0.1.132
IP 多播地址 （IEEE 1588 版本 2）	30、31、32、33	0xE0、0x00、0x01、0x81 （十六进制） 0xE0、0x00、0x00、0x6B （十六进制）	PTP 主多播地址：224.0.1.129 PTP Pdelay 多播地址：224.0.0.107
UDP 目标端口	36、37	0x013F、0x0140	0x013F：PTP 事件消息(1) 0x0140：PTP 通用消息
PTP 控制位字段 （IEEE 1588 版本 1）	74	0x00、0x01、0x02、0x03、0x04	0x00：SYNC 0x01：Delay_Req 0x02：Follow_Up 0x03：Delay_Resp 0x04：管理
PTP 消息类型位字段 （IEEE 1588 版本 2）	42（半字节）	0x0、0x1、0x2、0x3、0x8、0x9、0xB、0xC、0xD	0x0：SYNC 0x1：Delay_Req 0x2：Pdelay_Req 0x3：Pdelay_Resp 0x8：Follow_Up 0x9：Delay_Resp 0xA：Pdelay_Resp_Follow_Up 0xB：发布 0xC：信号发送 0xD：管理
PTP 版本	43（半字节）	0x1 或 0x2	0x1：支持 PTP 版本 1 0x2：支持 PTP 版本 2

1. PTP 事件消息为 SYNC、Delay_Req（IEEE 1588 版本 1 和 2）或是 Pdelay_Req、Pdelay_Resp（仅限 IEEE 1588 版本 2）

通过 IPv6 发送的 PTP 帧

表 536 中提供了用于控制通过 UDP over IPv6 发送的 PTP 数据包快照的匹配位字段相关信息，适用于 IEEE 1588 版本 1 和 2。带标记数据包的八位字节位置偏移为 4。该参数基于“IEEE 1588-2008，附录 D”以及*表 534* 中所定义的消息格式。

表 536. 控制和状态所需的 IPv6-UDP PTP 数据包位字段

匹配位字段	八位字节位置	匹配值	说明
MAC 数据包类型	12、13	0x86DD	IP 数据报
IP 版本	14（位 [7:4]）	0x6	IP 版本为 IPv6
第 4 层协议	20(1)	0x11	UDP
PTP 多播地址	38 – 53	FF0x:0:0:0:0:0:0:181（十六进制） FF02:0:0:0:0:0:0:6B（十六进制）	PTP 主多播地址： FF0x:0:0:0:0:0:0:0:181（十六进制） PTP Pdelay 多播地址： FF02:0:0:0:0:0:0:0:6B（十六进制）
UDP 目标端口	56、57a	0x013F、0x140	0x013F：PTP 事件消息 0x0140：PTP 通用消息
PTP 控制位字段（IEEE 1588 版本 1）	94a	0x00、0x01、0x02、0x03 或 0x04	0x00：SYNC 0x01：Delay_Req 0x02：Follow_Up 0x03：Delay_Resp 0x04：管理 (version1)
PTP 消息类型位字段（IEEE 1588 版本 2）	62a（半字节）	0x0、0x1、0x2、0x3、0x8、0x9、0xB、0xC 或 0xD	0x0：SYNC 0x1：Delay_Req 0x2：Pdelay_Req 0x3：Pdelay_Resp 0x8：Follow_Up 0x9：Delay_Resp 0xA：Pdelay_Resp_Follow_Up 0xB：发布 0xC：信号发送 0xD：管理
PTP 版本	63（半字节）	0x1 或 0x2	0x1：支持 PTP 版本 1 0x2：支持 PTP 版本 2

1. 尚未面向 PTP 数据包定义扩展报头。

通过以太网发送的 PTP 数据包

表 537 中提供了用于控制通过以太网发送的 PTP 数据包快照的匹配位字段相关信息，适用于 IEEE 1588 版本 1 和 2。带标记数据包的八位字节位置偏移为 4。该参数基于“IEEE 1588-2008，附录 D”以及消息格式。

表 537. 控制和状态所需的以太网 PTP 数据包位字段

匹配位字段	八位字节位置	匹配值	说明
MAC 目标多播地址(1)	0–5	01-1B-19-00-00-00 01-80-C2-00-00-0E	所有 PTP 消息都可以使用以下任意多播地址(2)： 01-1B-19-00-00-00 01-80-C2-00-00-0E(3)
MAC 数据包类型	12、13	0x88F7	PTP 以太网数据包
PTP 控制位字段（IEEE 1588 版本 1）	46	0x00、0x01、0x02、0x03 或 0x04	0x00：SYNC 0x01：Delay_Req 0x02：Follow_Up 0x03：Delay_Resp 0x04：管理
PTP 消息类型位字段（IEEE 1588 版本 2）	14（半字节）	0x0、0x1、0x2、0x3、0x8、0x9、0xB、0xC 或 0xD	0x0：SYNC 0x1：Delay_Req 0x2：Pdelay_Req 0x3：Pdelay_Resp 0x8：Follow_Up 0x9：Delay_Resp 0xA：Pdelay_Resp_Follow_Up 0xB：发布 0xC：信号发送 0xD：管理
PTP 版本	15（半字节）	0x1 或 0x2	0x1：支持 PTP 版本 1 0x2：支持 PTP 版本 2

1. 如果将*时间戳控制寄存器 (ETH_MACTSCR)* 的 TSENMACADDR 位置 1，则系统将使用需在 MAC 地址 0 到 31 中编程的目标地址 (DA) 的单播地址匹配。
2. IEEE 1588-2008，附录 F
3. MAC 不会将带有点延迟多播地址 (01-80-C2-00-00-0E) 的 PTP 版本 1 消息视为有效的 PTP 消息。

发送路径功能

通过 GMII 或 MII 接口发送数据包的起始数据包分隔符 (SFD) 时，MAC 会捕获时间戳。对于需要捕获其时间戳的数据包，可以按数据包进行控制。可对每个发送数据包进行标记，以指示是否应为其捕获时间戳。

MAC 不会处理发送的数据包来识别 PTP 数据包。必须指定需要捕获其时间戳的数据包。该数据包可通过使用发送描述符中的控制位进行定义（请参见*第 57.10.3 节：发送描述符*）。MAC 将时间戳返回到相应发送描述符内的软件，进而将时间戳自动连接到特定 PTP 数据包。

64 位时间戳信息会被写入到 TDES0 和 TDES1 位字段。TDES0 位字段会保持时间戳的 32 个最低有效位。

接收路径功能

可以将 MAC 编程为捕获通过 GMII 或 MII 接口接收到的所有数据包的时间戳，或者编程为处理数据包来识别有效 PTP 消息。可使用*时间戳控制寄存器 (ETH_MACTSCR)* 的以下选项来控制要发送到应用程序的时间快照：

- 使能所有数据包的快照
- 使能 IEEE 1588 版本 1 或版本 2 时间戳的快照
- 使能直接通过以太网或 UDP-IP-Ethernet 发送的 PTP 数据包的快照
- 使能所接收到的 IPv4 或 IPv6 数据包的时间戳快照
- 仅使能 EVENT 消息（SYNC、DELAY_REQ、PDELAY_REQ 或 PDELAY_RESP）的时间戳快照
- 将节点使能为主节点或从节点，并选择快照类型

  此功能控制要拍摄快照的消息类型。

注： *该外设也支持通过 VLAN 数据包发送的 PTP 消息。*

表 538 指示根据*时间戳控制寄存器 (ETH_MACTSCR)* 中的 SNAPTYPSEL 位字段拍摄快照的 PTP 消息。

表 538. 时间戳快照对 ETH_MACTSCR 位的相依性

SNAPTYPSEL	TSMSTRENA	TSEVNTENA	PTP 消息
00	X	0	SYNC、Follow_Up、Delay_Req、Delay_Resp
00	0	1	SYNC
00	1	1	Delay_Req
01	X	0	SYNC、Follow_Up、Delay_Req、Delay_Resp、Pdelay_Req、Pdelay_Resp、Pdelay_Resp_Follow_Up
01	0	1	SYNC、Pdelay_Req、Pdelay_Resp
01	1	1	Delay_Req、Pdelay_Req、Pdelay_Resp
10	X	X	SYNC、Delay_Req
11	X	X	Pdelay_Req、Pdelay_Resp

DMA 将时间戳返回到相应接收描述符内的软件应用程序。包含时间戳消息状态和 IPC 状态的扩展状态写入 RDES1 正常描述符中，时间戳的快照写入上下文描述符的 RDES0 和 RDES1 位字段中。RDES0 位字段会保持时间戳的 32 个最低有效位。

关于 IEEE 1588 时间戳的编程指南（系统时间校准）

请参见*第 3108 页的第 57.9.10 节：关于 IEEE 1588 时间戳的编程指南*中的*系统时间校准*一节。

IEEE 1588 系统时间源

为获取时间快照，MAC 需要一个 64 位格式的参考时间，如 IEEE 1588-2002 中定义（80 位格式如 IEEE 1588-2008 中定义）。

IEEE 1588 系统时间源的说明

该外设将使用参考时钟输入在内部生成参考时间（亦称作系统时间）以及捕获时间戳。

时间戳具有以下位字段：

- UInteger48 秒位字段

  秒位字段是以秒为单位的时间戳的整数部分。其宽度为 48 位。例如，2.000000001 秒表示为秒位字段 = 0x0000 0000 0002。
- UInteger32 nanosecondsField

  纳秒位字段是以纳秒为单位的时间戳的小数部分。例如，2.000000001 秒表示为纳秒 = 0x0000 0001。

  纳秒位字段支持以下两种模式：
  - **数字翻转模式**：在该模式下，纳秒位字段中的最大值为 0x3B9A C9FF，即 (10e9-1) 纳秒。
  - **二进制翻转模式**：在该模式下，纳秒位字段会翻转，并在值 0x7FFF FFFF 之后使秒位字段递增。精度约为 0.466 ns/位。

    这些模式可通过*时间戳控制寄存器 (ETH_MACTSCR)* 中的 TSCTRLSSR 位进行设置。

系统时间寄存器模块

使用 PTP 输入参考时钟 clk_ptp_ref_i 更新 64 位 PTP 时间。该 PTP 时间可用作时钟源，以获取 GMII 上发送或接收的以太网帧的快照（时间戳）。

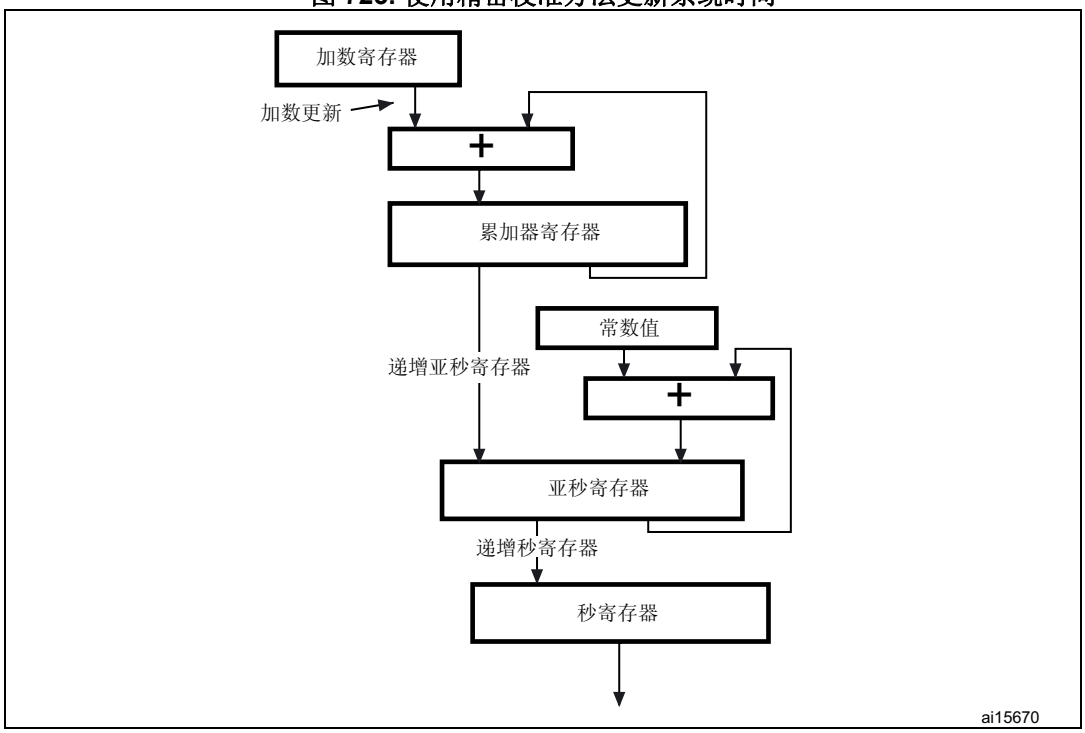
可使用粗略校准方法或精密校准方法对系统时间定时器进行初始化或校准。

使用粗略校准方法时，初始值或偏移值会写入时间戳更新寄存器。对于初始化，将使用时间戳更新寄存器中的值对系统时间计数器进行编程；对于系统时间校准，会将偏移值（时间戳更新寄存器）加到系统时间中或从系统时间中减去。

使用精密校准方法时，从时钟（参考时钟）频率相对于主时钟（如 IEEE 1588-2002 规范中定义）的偏移会在一段时间内进行校准，而不像粗略校准方法中那样，在单个时钟周期内进行校准。校准时间越长，越有助于保持线性时间，并且不会导致各 PTP 同步消息间隔之间的参考时间发生剧烈变化（或者大型抖动）。在此方法中，会使用一个累加器对加数寄存器中的内容求和，如*图 723* 所示。累加器生成的算数进位将用作使系统时间计数器递增的脉冲。累加器和加数寄存器均为 32 位寄存器。累加器用作高精度频率乘法器或除法器。

该系统时间更新算法如*图 723* 所示。

图 723. 使用精密校准方法更新系统时间



系统时间更新逻辑需要使用 50 MHz 的时钟频率，以达到 20 ns 的精度。分频是指参考时钟频率与所需时钟频率的比率。例如，如果参考时钟 (clk_ptp_ref_i) 的频率为 66MHz，则通过计算可知分频比为 66 MHz/50 MHz = 1.32。因此，寄存器中设置的默认加数值为 2^{32} / 1.32，相当于 0xC1F07C1F。

如果参考时钟向下偏移，例如降至 65 MHz，则分频比为 65 / 50（即，1.3），加数寄存器中要设置的值为 2^{32} / 1.30，相当于 0xC4EC 4EC4。

如果时钟向上偏移，例如升至 67 MHz，则必须将加数寄存器设置为 0xBF0B 7672。无时钟偏移时，必须编程的默认加法值为 0xC1F0 7C1F (2^{32} / 1.32)。

在*图 723* 中，用于累加亚秒寄存器的常数值为十进制 43，这可使系统时间精度达到 20 ns（换言之，增量步长时间为 20 ns）。

软件必须根据 SYNC 消息对频率偏移进行计算，并相应更新加数寄存器。

首先，使用加数寄存器中的 FreqCompensationValue0 设置从时钟。该值的计算公式如下：

$$FreqCompensationValue_0 = 2^{32} / FreqDivisionRatio$$

如果起初假定 MasterToSlaveDelay 对于连续的同步消息是相同的，则必须应用本节介绍的算法。数个同步周期之后，频率会锁定。从时钟随后可确定 MasterToSlaveDelay 的精确值，并使用新值重新与主时钟同步。

该算法如下：

1. 在 $MasterSyncTime_n$ 时刻，主时钟向从时钟发送同步消息。从时钟在其本地时钟为 $SlaveClockTime_n$ 时接收到该消息，并用如下公式计算 $MasterClockTime_n$：

$$MasterClockTime_n = MasterSyncTime_n + MasterToSlaveDelay_n$$

2. 当前同步周期的主时钟计数 $MasterClockCount_n$ 的计算公式如下：

$$MasterClockCount_n = MasterClockTime_n - MasterClockTime_{n-1}$$

（假定 MasterToSlaveDelay 对于同步周期 n 和 n – 1 是相同的）

3. 当前同步周期的从时钟计数 $SlaveClockCount_n$ 的计算公式如下：

$$SlaveClockCount_n = SlaveClockTime_n - SlaveClockTime_{n-1}$$

4. 当前同步周期的主从时钟计数差值 $ClockDiffCount_n$ 的计算公式如下：

$$ClockDiffCount_n = MasterClockTime_n - SlaveClockTime_n$$

5. 从时钟的分频系数 $FreqScaleFactor_n$ 的计算公式如下：

$$FreqScaleFactor_n = (MasterClockCount_n + ClockDiffCount_n)/SlaveClockCount_n$$

6. 加数寄存器的频率补偿值 $FreqCompensationValue_n$ 的计算公式如下：

$$FreqCompensationValue_n = FreqScaleFactor_n \times FreqCompensationValue_{n-1}$$

理论上，该算法可在一个同步周期内实现锁定。但是，由于网络传播延迟和工作条件会不断变化，因此可能需要多个周期。该算法可进行自校准。如果最初通过主时钟设置的从时钟不正确，则该算法会花费更多同步周期将其校准。

有关详细的编程步骤，请参见*第 57.9.10 节：关于 IEEE 1588 时间戳的编程指南*。

IEEE 1588 辅助快照

用户可使用辅助快照功能来基于外部事件存储系统时间的快照。该事件被视为 eth_ptp_trgx（其中 x = 1 到 4）边带信号的上升沿。

最多可配置四个辅助快照输入，因此最多可存储四个快照。FIFO 可通过以下寄存器进行访问：*辅助时间戳秒寄存器 (ETH_MACATSSR)* 以及 *辅助时间戳纳秒寄存器 (ETH_MACATSNR)*。

针对任意输入拍摄的快照均存储在一个公共 FIFO 中；只保留 64 位。应用程序可读取 *时间戳状态寄存器 (ETH_MACTSSR)*，以了解可在此 FIFO 顶部读取哪个输入的时间戳。

存储快照时，MAC 会向应用程序发送中断进行指示。快照的值通过 FIFO 寄存器访问进行读取。如果 FIFO 已满，且用于拍摄快照的外部触发信号置为有效，则 *时间戳状态寄存器 (ETH_MACTSSR)* 中的快照触发未命中状态 (ATSSTM) 会置 1。这表示，时间戳的最新辅助快照未存储在 FIFO 中。在 FIFO 已满时未向其中写入最新快照。

应用程序从 FIFO 中读取 64 位时间戳时，用于存储下一个快照的空间将可用。可通过将 *辅助控制寄存器 (ETH_MACACR)* 中的 ATSFC 位置 1 来清空 FIFO。FIFO 中存在多个快照时，*时间戳状态寄存器 (ETH_MACTSSR)* 的位 [27:25] 中会指示相应的计数。

每秒脉冲数灵活的输出

MAC 将支持每秒脉冲数固定的输出模式（也称为固定模式），或是适用于 ETH_PPS_OUT 和 eth_ptp_pps_out 输出的每秒脉冲数灵活的输出模式：

- 每秒脉冲数固定的输出

  在该模式下，只能通过将 *PPS 控制寄存器 (ETH_MACPPSCR)* 中的 PPSCTRL0 位字段置 1 来更改 PPS 输出的频率。

- 每秒脉冲数灵活的输出

  在该模式下，该软件可以灵活地编程在 eth_ptp_pps_out 输出端生成的脉冲的起始时间或停止时间、宽度和间隔。

  起始时间和停止时间通过 *PPS 目标时间秒寄存器 (ETH_MACPPSTTSR)* 和 *PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)* 编程。

  PPS 宽度和间隔通过相应 *PPS 间隔寄存器 (ETH_MACPPSIR)* 和 *PPS 宽度寄存器 (ETH_MACPPSWR)* 基于系统时间的粒度（亚秒增量值的单位数）进行编程。

注： *默认情况下，外设将处于固定模式，并表示一秒的时间间隔。当通过在 PPS 控制寄存器 (ETH_MACPPSCR) 中将 PPSEN0 清零来选择固定模式时：*

- *所有 PPS 输出上的输出将由在 PPSCTRL_PPSCMD 位字段中编程的值进行控制。固定模式不支持单独 PPS 输出的独立控制。*
- *PPS 目标时间秒寄存器 (ETH_MACPPSTTSR) 和 PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR) 仅用于生成达到目标时间中断，而不用于生成 PPS 输出。*
- *TRGTMODSEL0/1/2/3 必须编程为 0。*
- *可通过将 PPS 控制寄存器 (ETH_MACPPSCR) 中的 PPSCTRL0 位字段置 1 来更改 PPS 输出的频率。*

每秒脉冲数 (PPS) 灵活的输出说明

外设支持以下有关灵活 PPS 输出的特性：

- 根据系统时间对启动或停止时间进行编程。
- 根据 64 位系统时间对单脉冲的起始点以及脉冲串的启动点和停止点进行编程。目标时间寄存器用于对启动和停止时间进行编程。
- 提前对停止时间进行编程，也就是说，在实际启动时间结束之前就可以对停止时间进行编程。
- 根据在*亚秒增量寄存器 (ETH_MACSSIR)* 中编程的亚秒增量值单位数，对 PPS 信号输出的上升沿与相应下降沿之间的宽度进行编程。脉冲宽度的可编程范围为 1 至 232–1 个亚秒增量值单位。
- 根据亚秒增量值的单位数对 PPS 信号上升沿之间的间隔进行编程。用户可将脉冲之间的间隔编程为 1 至 232–1 个亚秒增量值单位。
- 可以取消已编程的 PPS 启动或停止请求。
- 如果已编程的启动或停止时间已结束，则系统会出错。

注： *以下章节提到的 PTP 参考时钟为更新系统时间的时钟。当时间戳控制寄存器 (ETH_MACTSCR) 的 TSCFUPDT 位置 0 时，该时钟将与 clk_ptp_ref_i 时钟类似。在精密校准模式下，此为更新系统时间（使用亚秒增量寄存器 (ETH_MACSSIR)，如图 723 中所示）的时钟节拍。*

有关如何配置灵活脉冲输出的更多详细信息，请参见*第 57.9.13 节：关于每秒脉冲数 (PPS) 灵活的输出的编程指南*。

PPS 启动和停止时间

初始启动时间可在目标时间寄存器中进行编程。

如果需要，可以重新编程启动或停止时间。但是，该操作必须在先前编程的值与 PTP 时钟域同步之后方可进行。*PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)* 的位 31 表示同步已完成。这使得用户可以在先前的停止或启动时间结束之前提前编程启动或停止时间。

为确保正确 PPS 信号输出，建议为起始或停止时间编程高级系统时间。如果应用程序编程的起始或停止时间已过时，则 MAC 会将指示编程错误的错误状态位置 1。MAC 还会产生“达到目标时间”中断事件（如果已使能）。只有在相应起始或停止时间尚未过时的情况下，应用程序才能取消起始或停止请求。如果时间已过时，则取消命令失效。

PPS 宽度和间隔

PPS 宽度和间隔通过基于系统时间的粒度（即亚秒增量值的单位数）进行编程。例如，要在 PTP 参考时钟频率为 50 MHz 的情况下获得 40 ns 的 PPS 脉冲宽度和 100 ns 的间隔，请分别将宽度和间隔值编程为 2 和 5。使用更快的 PTP 参考时钟可以获得更小的粒度。

在发出在 PPS 输出上触发脉冲或脉冲串的命令之前，请编程或更新 PPS 信号输出的间隔和宽度。

PTP 时间戳减荷功能

当 MAC 用作 PTP 网络中的特定节点时，使用该功能可自动生成特定 PTP 数据包。

这类数据包可定期生成，或者由主机软件触发生成。在其他模式下，该功能可解析接收器端传入的 PTP 数据包，并且自动生成和响应所需 PTP 数据包。它有助于以更高的精度和更低的响应延迟为某些 PTP 节点功能减荷。

PTP 减荷功能通过 *PTP 减荷控制寄存器 (ETH_MACPOCR)* 进行选择。80 位 PTP 节点标识需通过以下三个寄存器进行配置：*PTP 源端口标识 0 寄存器 (ETH_MACSPI0R)*、*PTP 源端口标识 1 寄存器 (ETH_MACSPI1R)* 和 *PTP 源端口标识 2 寄存器 (ETH_MACSPI2R)*。

PTP 减荷功能的说明

根据所编程模式的不同，MAC 会在不同的条件下生成 PTP 以太网消息：定期生成、由应用程序触发生成、或在接收到特定 PTP 消息时生成。*表 539* 列出了 PTP 消息的生成条件。

表 539. PTP 消息生成条件

编程			模式	生成 PTP 消息的条件	生成的 PTP 消息类型
SNAPTYPSEL	TSMSTRENA	TSEVNTENA			
00	0	1	普通或边界从模式	接收到 SYNC 消息	Delay_Req
00	1	1	普通或边界主模式	定期生成或由应用程序触发	SYNC
				接收到 Delay_Req 消息	Delay_Resp
01	0	1	透明从模式	定期生成或由应用程序触发	Pdelay_Req
				接收到 Pdelay_Req 消息	Pdelay_Resp
				接收到 SYNC 消息	Delay_Req

表 539. PTP 消息生成条件（续）

编程			模式	生成 PTP 消息的条件	生成的 PTP 消息类型
SNAPTYPSEL	TSMSTRENA	TSEVNTENA			
01	1	1	透明主模式	定期生成或由应用程序触发	Pdelay_Req
				接收到 Pdelay_Req 消息	Pdelay_Resp
				定期生成或由应用程序触发	SYNC
				接收到 Delay_Req 消息	Delay_Resp
11	X	X	点对点透明模式	定期生成或由应用程序触发	Pdelay_Req
				接收到 Pdelay_Req 消息	Pdelay_Resp
所有其他编程组合对于 PTP 减荷功能均无效。					

注： *根据 IEEE 1588-2008 规范，支持点延迟机制的时钟不能生成延迟请求/延迟响应消息。然而，外设控制器支持利用 PTP 减荷控制寄存器 (ETH_MACPOCR) 中的可编程控制位 (DRRDIS) 来实现此操作，以提升操作的灵活性。*

DRRDIS 位可用于控制延迟请求/延迟响应消息的响应生成。例如在透明从模式中，只有当该位复位时，系统才会生成延迟请求以响应接收到 SYNC。

当 MAC 设置为 PTP 网络中的普通或边界从时钟时，它可以通过自动生成并发送相应的 Delay_Req 消息来响应接收 SYNC 消息。同样，*表 539* 中也介绍了各种其他的工作模式。

MAC 支持多播通信模型，用于生成 SYNC 和 Pdelay_Req PTP 消息。例如，所生成 PTP over Ethernet 数据包中的目标地址位字段为已定义的特殊多播地址（除点延迟机制消息之外的所有消息为 0x011B 1900 0000，点延迟机制消息为 0x0180 C200 000E）。

当 MAC 对具有特殊多播目标地址的已接收 SYNC、Delay_Req 和 Pdelay_Req PTP 消息做出响应时，它也会分别在自动生成的 Delay_Req、Delay_Resp 和 Pdelay_Resp PTP 消息的 DA 位字段中使用相应的特殊多播地址。

当 MAC 对具有单播目标地址的已接收 SYNC、Delay_Req 和 Pdelay_Req PTP 消息做出响应时，它会获取已接收数据包的 SA 位字段，并将其用作自动生成的 Delay_Req、Delay_Resp 和 Pdelay_Resp PTP 消息的 DA 位字段。

同时，系统会将所有已接收 PTP 消息连同 Rx 状态一起转发到应用程序，以指示该响应是否由 MAC 生成，以及它是否满足 MAC 接收器的数据包过滤逻辑

当 MAC 自动生成 PdelayReq 或响应 Delay_Req 时，系统将在 Tx TS 状态中提供这两个 PTP 消息的出站时间戳（Tx 时间戳状态寄存器和所产生的中断）。

除了适用于基本 PTP over Ethernet 消息检测的 messageType 和 versionPTP 位字段之外，以下额外位字段也可用于确认已接收 PTP 消息的类型：

1. 将检查 domainNumber 位字段是否与 CSR 中编程的值匹配。
2. 系统会检查 flagField 位字段中 twoStepFlag 的一步指示 (0b0)。
3. 使能后，系统将检查 transportSpecific 位字段是否为默认的 PTP over Ethernet (0b0000) 或 802.1AS 模式 (0b1111)。

PTP 数据包生成

本节将介绍使能此模式时 MAC 自动生成的 PTP 数据包的格式和内容。文中给出了通用 PTP 消息报头的模板，以及生成的特定 PTP 数据包的位字段的详细说明。

表 540. 通用 PTP 消息报头位字段

位								八位字节	偏移
7	6	5	4	3	2	1	0		
transportSpecific				messageType				1	0
保留				versionPTP				1	1
messageLength								2	2
domainNumber								1	4
保留								1	5
flagField								2	6
correctionField								8	8
保留								4	16
sourcePortIdentity								10	20
sequenceId								2	30
sequenceId								2	30
controlField								1	32
logMessageInterval								1	33

PTP 消息报头位字段

- **messageType**

  以下编码值将用于 PTP 消息类型：

  – SYNC：0000
  – Delay_Req：0001
  – Pdelay_Req：0010
  – Pdelay_Resp：0011
  – Delay_Resp：1001

- **transportSpecific**

  将使用以下传输协议编码：

  – 默认的 PTP over Ethernet：0000
  – 802.1AS 模式：0001

- **versionPTP**

  该位字段始终设置为 2，原因是系统支持 PTP 版本 2。

- **domainNumber**

  该位字段包含来自 *PTP 减荷控制寄存器 (ETH_MACPOCR)* 的值。

- **flagField**

  系统将使用以下值：

  – alternateMasterFlag（八位字节 0 位 0）：SYNC 和 Delay_Resp 为 0
  – twoStepFlag（八位字节 0 位 1）：SYNC 和 Pdelay_Resp 为 0
  – unicastFlag（八位字节 0 位 2）：多播地址为 0，单播地址为 1

- **correctionField**

  有关更多信息，请参见*表 541*。

- **sourcePortIdentity**

  该位字段将获取在 *PTP 源端口标识 0 寄存器 (ETH_MACSPI0R)*、*PTP 源端口标识 1 寄存器 (ETH_MACSPI1R)* 和 *PTP 源端口标识 2 寄存器 (ETH_MACSPI2R)* 中编程的值。

- **sequenceId**

  Pdelay_Resp 和 Delay_Resp 将使用与接收到的 Pdelay_Req 和 Delay_Req PTP 消息相同的 sequenceId 位字段。对于 SYNC/Delay_Req 和 Pdelay_Req，系统将保留一个单独的 sequenceId 计数器。每次生成并发送相应的消息时，这些 sequenceId 计数器都会加 1。

- **controlField**

  以下编码值将用于 controlField：

  – SYNC：0000 0000
  – Delay_Req：0000 0001
  – Pdelay_Req：0000 0010
  – Pdelay_Resp：0000 0101
  – Delay_Resp：0000 0011

- **logMessageInterval**

  – SYNC：

    此位字段将包含对应的 MAC_Log_Message_Interval 寄存器中的 logSyncInterval。

  – Delay_Resp：

    此位字段将包含适用于多播 PTP 消息的，来自*日志消息间隔寄存器 (ETH_MACLMIR)* 的 DRSYNCR 和 logSyncInterval 值之和，以及适用于单播 PTP 消息的 0111 1111。

  – Delay_Req、Pdelay_Req 和 Pdelay_Resp：0111 1111

    其中 logSyncInterval = log2（间隔的平均值，以秒为单位）

MAC 支持 logSyncInterval 位字段的 –15 到 15 的值，其对应的范围为 32.768 微秒 (2–15) 到 215 秒。对于指定的日志同步间隔 (N) 值，两个 SYNC 数据包之间的时间间隔可通过以下公式得出：

- N 为负数（–1 至 –15）时为 $2^{(30+N)}$ ns
- N 为正数（0 至 15）时为 2^{N} 秒

例如：

- 当 logSyncInterval 编程为 1 时，间隔为 2^{1}；因此，SYNC 消息将会每 2 秒发送一次。
- 当 logSyncInterval 编程为 -1 时，间隔为 2^{-1} = 0.536 秒；因此，SYNC 消息将会每 536 毫秒发送一次。该值为 0.536 秒，因为 2^{-30} = 1 ns。
- 当 logSyncInterval 编程为 -5 时，间隔为 2^{-5} = 33.55 ms；因此，SYNC 消息将会每 33.55 ms 发送一次。

注： *MAC 会使用 PTP 系统时间来生成定期数据包发送间隔。对于所编程日志消息间隔的负值，生成的周期可能会偏离通过公式 $2^{(30+N)}$ 得出的值，原因是系统时间的纳秒位字段采用了非二进制格式。*

PTP 特定消息位字段

特定消息位字段如下：

- **messageLength**

  系统不支持后缀，因此此位字段所包含的 PTP 消息长度将包括 34 字节的 PTP 通用报头和特定于消息类型的正文。

  对于 SYNC 和 Delay_Req 数据包，此位字段将包含 44，而对于 Delay_Resp、Pdelay_Req 和 Pdelay_Resp，此位字段将包含 54。
- **originTimestamp**

  此位字段是为 SYNC、Delay_Req 和 Pdelay_Req PTP 消息捕获的出站时间戳。
- **receiveTimestamp**

  对于 Delay_Resp PTP 消息，此位字段为相应的已接收 Delay_Req PTP 消息的入站时间戳。
- **requestingPortIdentity**

  对于 Delay_Resp 和 Pdelay_Resp PTP 消息，此位字段为从相应的已接收 Delay_Req 和 Pdelay_Req PTP 消息中获取到的 sourcePortIdentity 位字段。
- **requestReceiptTimestamp**

  对于 Pdelay_Resp PTP 消息，此位字段将设置为 0。

单步时间戳

MAC 支持单步时间戳功能，该功能用于识别数据包中的偏移，并在该偏移处插入从应用程序接收到的时间戳。

面向单步时间戳的 MAC 发送 PTP 模式

MAC 将根据消息的类型和模式，更新发送 PTP 数据包的以下位字段：

- PTP 消息报头中的 correctionField
- SYNC、Delay_Req 和 Pdelay_Req 消息中的 originTimestamp

表 541 中显示了如何基于*时间戳控制寄存器 (ETH_MACTSCR)* 中 SNAPTYPSEL、TSMSTRENA 和 TSEVNTENA 位的设置来选择 PTP 模式，以及在单步时间戳操作期间，根据该模式中消息类型更新的传入 PTP 数据包的位字段。

表 541. MAC 发送 PTP 模式和单步时间戳操作

编程			模式	按数据包控制(1)			在发送中处理的消息
SNAPTYPSEL	TSMSTRENA	TSEVNTENA		TTSE(2)	OSTC(3)	TTS(4)	
X	X	X	N/A	1	X	X	捕获时间戳并将其返回到应用程序
X	X	X	N/A	X	0	X	不执行 OST 操作（未修改 PTP 数据包）
00	X	0	端对端透明	0	1	入站 TS	Sync（校准位字段，用于停留时间和入站不对称校准）
							Delay_Req（校准位字段，用于停留时间和出站不对称校准）

表 541. MAC 发送 PTP 模式和单步时间戳操作（续）

编程			模式	按数据包控制(1)			在发送中处理的消息
SNAPTYPSEL	TSMSTRENA	TSEVNTENA		TTSE(2)	OSTC(3)	TTS(4)	
00	0	1	普通或边界从模式	1	1	X	Delay_Req（originTimestamp 位字段） Delay_Req（校准位字段，用于出站不对称校准）
00	1	1	普通或边界主模式	0	1	X	Sync（originTimestamp 位字段） Sync（校准位字段，用于亚纳秒校准）
01	X	0	支持点延迟机制的端对端透明	0	1	入站 TS	Sync（校准位字段，用于停留时间和入站不对称校准）
						入站 TS	Pdelay_Req（校准位字段，用于停留时间和出站不对称校准）
						入站 TS	Pdelay_Resp（校准位字段，用于停留时间和入站不对称校准）
01	0	1	支持点延迟机制或点对点透明的普通或边界从模式	0	1	入站 TS	Sync（校准位字段，用于停留时间和入站不对称校准） （仅适用于点对点透明时钟操作）
				1	1	X	Delay_Req（originTimestamp 位字段） Delay_Req（校准位字段，用于出站不对称校准）
				1	1	X	Pdelay_Req（originTimestamp 位字段） Pdelay_Req（校准位字段，用于出站不对称校准）
				0	1	Pdelay_Req 的入站 TS	Pdelay_Resp（校准位字段，用于周转时间和入站不对称校准）
01	1	1	支持点延迟机制的普通或边界主模式	0	1	X	Sync（originTimestamp 位字段） Sync（校准位字段，用于亚纳秒校准）
				1	1	X	Pdelay_Req（originTimestamp 位字段） Pdelay_Req（校准位字段，用于出站不对称校准）
				0	1	Pdelay_Req 的入站 TS	Pdelay_Resp（校准位字段，用于周转时间和入站不对称校准）

表 541. MAC 发送 PTP 模式和单步时间戳操作（续）

编程			模式	按数据包控制(1)			在发送中处理的消息
SNAPTYPSEL	TSMSTRENA	TSEVNTENA		TTSE(2)	OSTC(3)	TTS(4)	
10	X	X	端对端透明	0	1	入站 TS	Sync（校准位字段，用于停留时间和入站不对称校准）
						入站 TS	Delay_Req（校准位字段，用于停留时间和出站不对称校准）
11	X	X	点对点透明	0	1	入站 TS	Sync（校准位字段，用于停留时间和入站不对称校准）
				1	1	X	Pdelay_Req（originTimestamp 位字段） Pdelay_Req（校准位字段，用于出站不对称校准）
				0	1	Pdelay_Req 的入站 TS	Pdelay_Resp（校准位字段，用于周转时间和入站不对称校准）

1. 此处给出的按数据包控制值为各类设备在典型 PTP 操作和已编程模式中使用的推荐设置。
2. TTSE 代表 TDES2 发送正常描述符的 TTSE 位。TTSE 功能与 OST 功能以及为 OST 编程的工作模式无关。当设置 TTSE 位时，MAC 会捕获并返回时间戳。
3. OSTC 代表 TDES3 发送上下文描述符的 OSTC 位
4. TTS 代表 TDES0 和 TDES1 发送正常描述符（回写格式）的 TTSH、TTSL 位字段中提供的时间戳值。

注： *停留时间/周转时间的计算方法为捕获的时间戳（出站时间戳）与入站时间戳之间的差值。支持点延迟机制的时钟不会使用延迟请求或响应，但会为了提升灵活性而包含在 OST 中。*

使能单步时间戳

用户可在 TDES3 上下文描述符中设置位 20 (OSTC)，以便为给定数据包使能单步时间戳功能。如需更新某些 PTP 数据包中的校准位字段，必须在 TSSL 和 TSSH 位字段中给出入站时间戳。

单步时间戳功能仅适用于 PTP over Ethernet 数据包，而并不适用于 PTP over IPv4/IPv6 数据包。

57.5.5 校验和减荷引擎

通信协议（例如 TCP 和 UDP）将实施校验和位字段，这有助于确定通过网络发送的数据的完整性。以太网最广泛的用途是通过 IP 数据报封装 TCP 和 UDP。MAC 具有校验和减荷引擎 (COE)，支持校验和计算、发送路径中的校验和插入以及接收路径中的错误检测。

发送校验和减荷引擎

在发送路径中，MAC 将计算校验和并将其插入 Tx 数据包。该功能有助于减轻软件的负载，并能提高整个系统的吞吐量。

COE 模块支持两种类型的校验和计算与插入。可以通过将 CIC 位（TDES3 位 [17:16]）置 1 来为各个数据包控制校验和引擎。

注： *通过完整的数据包来计算 TCP、UDP 或 ICMP 的校验和，然后将其插入相应的报头位字段。鉴于此要求，即使将 MAC 配置为阈值（直通）模式，Tx FIFO 也会自动进入存储转发模式工作。*

请确保 Tx FIFO 足够深，可在将数据包传输到 MAC 发送器之前存储完整的数据包；这是因为当没有空间用来接受已编程突发长度的数据时，MTL Tx FIFO 会开始读取以避免死锁。在这种情况下，由于在可以计算和插入有效负载校验和之前就已读取数据包报头的开始，COE 将会失败。因此，只能在小于通过以下公式得出的字节数的数据包中使能校验和插入：

$$\text{数据包大小} < \text{TxQSize} - (\text{PBL} + 7) \times 4$$

其中

TxQSize 对应于 *Tx 队列 x 工作模式寄存器 (ETH_MTLTXQxOMR)* 的 TQS 位字段

PBL 对应于*通道 x 发送控制寄存器 (ETH_DMACxTXCR)* 的 TxPBL 位字段

有关 IPv4、TCP、UDP、ICMP、IPv6 和 ICMPv6 数据包头规范的信息，请分别参见 IETF 规范 RFC 791、RFC 793、RFC 768、RFC 792、RFC 2460 和 RFC 4443。

IP 报头校验和引擎

在 IPv4 数据报中，报头位字段的完整性由 16 位头校验和位字段（IPv4 数据报的第十一个字节和第十二个字节）指示。当以太网数据包的类型位字段的值为 0x0800 且 IP 数据报的版本位字段的值为 0x4 时，COE 会检测到 IPv4 数据报。计算期间，将忽略输入数据包的校验和位字段并将其替换为计算出的值。

注： *IPv6 报头无校验和位字段。因此，COE 不会修改 IPv6 报头位字段。*

此 IP 报头校验和计算的结果由发送状态中的 IP 报头错误状态位（*表 563：TDES3 正常描述符（回写格式）*中的位 0）指示。

只要以太网类型位字段的值和 IP 报头版本位字段的值不一致，或当以太网数据包没有足够的数据（如 IP 报头长度位字段所指示）时，此状态位将置 1。换言之，在以下情况下发生 IP 报头错误时，此位将置 1：

- 对于 IPv4 数据报：
  - 接收到的以太网类型为 0x0800，但 IP 报头的版本位字段不等于 0x4。
  - IPv4 报头长度位字段指示小于 0x5（20 字节）的值。
  - 总的数据包长度小于 IPv4 报头长度位字段给定的值。
- 对于 IPv6 数据报：
  - 以太网类型为 0x86DD，但 IP 报头版本位字段不等于 0x6。
  - 数据包在 IPv6 报头（40 字节）之前结束，或已完全接收到扩展报头（如扩展报头中相应的报头长度位字段中给定）。

TCP/UDP/ICMP 校验和引擎

TCP/UDP/ICMP 校验和引擎对 IPv4 或 IPv6 报头（包括扩展报头）进行处理，并确定封装的有效负载是 TCP、UDP 还是 ICMP。计算 TCP、UDP 或 ICMP 有效负载的校验和，然后将其插入报头中相应的位字段。Tx COE 可工作在以下两种模式下：

- TCP、UDP 或 ICMPv6 伪报头并未包含在校验和计算中，并假定其存在于输入数据包的校验和位字段中。该引擎将校验和位字段包含在校验和计算中，然后将校验和位字段替换为最终计算得出的校验和。
- 引擎将忽略校验和位字段，并将 TCP、UDP 或 ICMPv6 伪报头数据包含在校验和计算中，然后使用最终计算出的值覆盖校验和位字段。

注： *对于 ICMP-over-IPv4 数据包，由于没有为其定义伪报头，因此在这两种模式下 ICMP 数据包中的校验和位字段都必须始终为 0x0000。如果不等于 0x0000，可能向数据包插入不正确的校验和。*

此操作的结果由发送状态向量中的有效负载校验和错误状态位（*表 563：TDES3 正常描述符（回写格式）*中的位 12）指示。如果该引擎检测到数据包已在存储转发模式下被转发到 MAC 发送器引擎但未向 FIFO 写入数据包结束 (EOP)，或者在接收到 IP 报头的有效负载长度位字段所指示的字节数之前数据包已结束，则会将有效负载校验和错误状态位置 1。当数据包的长度大于指示的有效负载长度时，COE 会将其作为填充字节忽略，并且不报告错误。该引擎检测到第一种类型的错误时，不会修改 TCP、UDP 或 ICMP 报头。对于第二种错误类型，仍会将计算得出的校验和插入相应的报头位字段。

表 542 介绍了发送校验和减荷引擎针对不同数据包类型所支持的功能。当 MAC 未插入校验和时，表中将以“否”进行表示。

注： *不要为大于发送校验和减荷引擎一节中指定的帧大小限制的 IPv4 或 IPv6 数据包使能校验和插入，因为这可能会导致校验和插入错误或意外行为。*

表 542. 针对不同数据包类型的发送校验和减荷引擎功能

数据包类型	硬件 IP 报头校验和插入	硬件 TCP/UDP 校验和插入
非 IPv4 或 IPv6 数据包	否	否
IPv4 数据包大于 1,522 字节（当在 MAC 中使能 IEEE 802.3ad 对 2K 数据包的支持时为 2,000 字节），但小于或等于*发送校验和减荷引擎*一节中指定的帧大小限制。	是	是
IPv6 数据包大于 1,522 字节（当在 MAC 中使能 IEEE 802.3ad 对 2K 数据包的支持时为 2,000 字节），但小于或等于*发送校验和减荷引擎*一节中指定的帧大小限制。	不适用	是
带 TCP、UDP 或 ICMP 的 IPv4	是	是
IPv4 数据包具有 IP 选项（IP 报头长度大于 20 字节）	是	是
数据包为 IPv4 段	是	否
主报头或扩展报头中包含以下后续报头位字段的 IPv6 数据包： – 逐跳选项（在 IPv6 主报头中） – 逐跳选项（在 IPv6 扩展报头中） – 目标选项 – 传送中（剩余段为 0） – 传送中（剩余段 > 0） – TCP、UDP 或 ICMP – 认证 – 主报头或扩展报头中的任何其他后续报头位字段	 – 不适用 – 不适用 – 不适用 – 不适用 – 不适用 – 不适用 – 不适用 – 不适用	 – 是 – 否 – 是 – 否 – 否 – 是 – 否 – 否
IPv4 数据包具有带选项位字段的 TCP 报头	是	是
IPv4 隧道： – IPv4 隧道中的 IPv4 数据包 – IPv4 隧道中的 IPv6 数据包	 – 是（IPv4 隧道报头） – 是（IPv4 隧道报头）	 – 否 – 否
IPv6 隧道： – IPv6 隧道中的 IPv4 数据包 – IPv6 隧道中的 IPv6 数据包	 – 不适用 – 不适用	 – 否 – 否
IPv4 数据包具有 802.3ac 标记（使能时带有 C-VLAN 标记或 S-VLAN 标记）。	是	是

表 542. 针对不同数据包类型的发送校验和减荷引擎功能（续）

数据包类型	硬件 IP 报头校验和插入	硬件 TCP/UDP 校验和插入
IPv6 数据包具有 802.3ac 标记（使能时带有 C-VLAN 标记或 S-VLAN 标记）。	不适用	是
带安全功能（例如封装的安全有效负载）的 IPv4 帧	是	否
带安全功能（例如封装的安全有效负载）的 IPv6 帧	不适用	否

接收校验和减荷引擎

接收校验和减荷引擎可计算报头校验和并将其与接收到的报头校验和进行匹配，以便检测 IP 数据包中的错误。该引擎还可以识别接收到的 IP 数据包中的 TCP、UDP 或 ICMP 有效负载，并正确计算这些负载的校验和。

接收校验和减荷引擎 (Rx COE) 可通过设置*工作模式配置寄存器 (ETH_MACCR)* 的 IPC 位进行使能。该位置 1 后，系统将对接收的以太网数据包中的 IPv4 和 IPv6 数据包进行检测和处理，以确保数据完整性。MAC 接收器通过检查所接收的以太网数据包的类型位字段中是否存在值 0x0800 或 0x86DD，来识别 IPv4 或 IPv6 数据包。此识别方法也适用于带单一 VLAN 标记的数据包。当设置了 *VLAN 标记寄存器 (ETH_MACVTR)* 的 EDVLP 位时，此方法还适用于带有双 VLAN 标记的数据包。

Rx COE 将计算 IPv4 报头校验和，并检查其是否与接收的 IPv4 报头校验和匹配。此操作的结果（通过或失败）将提供给 RFC 模块以插入到接收状态字中。如果指示的有效负载类型（以太网类型位字段）与 IP 报头版本之间有任何不匹配，或接收的数据包的字节数小于 IPv4 报头的长度位字段中指示的数量（IPv4 或 IPv6 报头中的可用字节数少于 20），IP 报头错误位将置 1。

当 *Rx 队列 x 工作模式寄存器 (ETH_MTLRXQxOMR)* 的 DIS_TCP_EF 位复位且设置了 FEP 位时，系统会将具有 TCP/IP 错误（报头或有效负载）的数据包丢弃在 MTL 层。

此引擎还可以识别接收的 IP 数据报（IPv4 或 IPv6）中的 TCP、UDP 或 ICMP 有效负载，并正确计算此类有效负载的校验和，如 TCP、UDP 或 ICMP 规范中所定义。该引擎包括用于校验和计算的 TCP、UDP 或 ICMPv6 伪报头字节，并检查接收的校验和位字段是否与计算的值匹配。此操作的结果由接收状态字中的有效负载校验和错误位给出。如果 TCP、UDP 或 ICMP 有效负载的长度与 IP 报头中给出的预期有效负载长度不匹配，此状态位也将置 1。

*表 543：针对不同数据包类型的接收校验和减荷引擎功能*介绍了 Rx COE 针对不同数据包类型所支持的功能。未处理 IP 数据包的有效负载（在表中以“否”进行表示）时，会在接收状态中给出是否绕过校验和引擎的信息。

注： *MAC 不会向接收的以太网数据包附加任何有效负载校验和字节。*

表 543. 针对不同数据包类型的接收校验和减荷引擎功能

数据包类型	硬件 IP 报头校验和检查	硬件 TCP/UDP/ICMP 校验和检查
非 IPv4 或 IPv6	否	否
IPv4 数据包大于 1,522 字节（当在 MAC 中使能 IEEE 802.3ad 对 2K 数据包的支持时为 2,000 字节）。	是	是
IPv6 数据包大于 1,522 字节（当在 MAC 中使能 IEEE 802.3ad 对 2K 数据包的支持时为 2,000 字节）。	不适用	是
带 TCP、UDP 或 ICMP 的 IPv4	是	是
IPv4 报头的协议位字段包含除 TCP、UDP 或 ICMP 之外的协议	是	否
IPv4 数据包具有 IP 选项（IP 报头长度大于 20 字节）	是	是
数据包为 IPv4 段	是	否
主报头或扩展报头中包含以下后续报头位字段的 IPv6 数据包： – 逐跳选项（在 IPv6 主报头中） – 逐跳选项（在 IPv6 扩展报头中） – 目标选项 – 传送中（剩余段为 0） – 传送中（剩余段 > 0） – TCP、UDP 或 ICMP – 主报头或扩展报头中的任何其他后续报头位字段	 – 不适用 – 不适用 – 不适用 – 不适用 – 不适用 – 不适用 – 不适用	 – 是 – 否 – 是 – 是 – 否 – 是 – 否
IPv4 数据包具有带选项位字段的 TCP 报头	是	是
IPv4 隧道： – IPv4 隧道中的 IPv4 数据包 – IPv4 隧道中的 IPv6 数据包	 – 是（IPv4 隧道报头） – 是（IPv4 隧道报头）	 – 否 – 否
IPv6 隧道： – IPv6 隧道中的 IPv4 数据包 – IPv6 隧道中的 IPv6 数据包	 – 不适用 – 不适用	 – 否 – 否
IPv4 数据包具有 802.3ac 标记（使能时带有 C-VLAN 标记或 S-VLAN 标记）。	是	是
IPv6 数据包具有 802.3ac 标记（使能时带有 C-VLAN 标记或 S-VLAN 标记）。	不适用	是
带安全功能（例如封装的安全有效负载）的 IPv4 帧	是	否
带安全功能（例如封装的安全有效负载）的 IPv6 帧	不适用	否

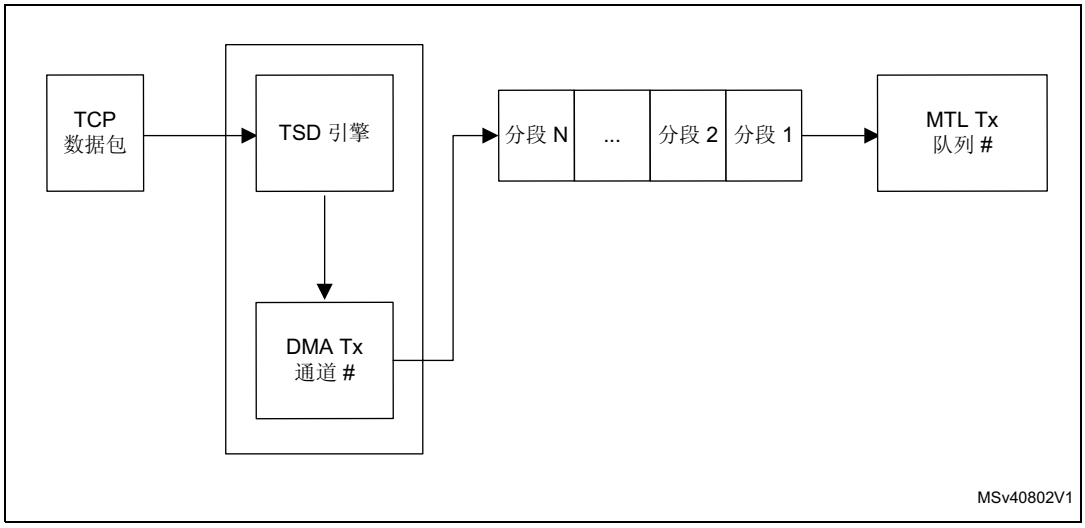
57.5.6 TCP 分段减荷

MAC 支持 TCP 分段减荷 (TSO) 功能，DMA 可使用该功能将大型 TCP 数据包拆分成多个小型数据包，并将这些数据包传递到 MTL，如*图 724* 所示。

通过对相应 ETH_DMACxCR 寄存器的 TSE 位进行编程来使能该功能（请参见*通道 x 发送控制寄存器 (ETH_DMACxTXCR)*）。只有 MAC 工作在全双工模式下时，才支持该功能。

有关详细的编程步骤，请参见*第 57.9.14 节：关于 TSO 的编程指南*。

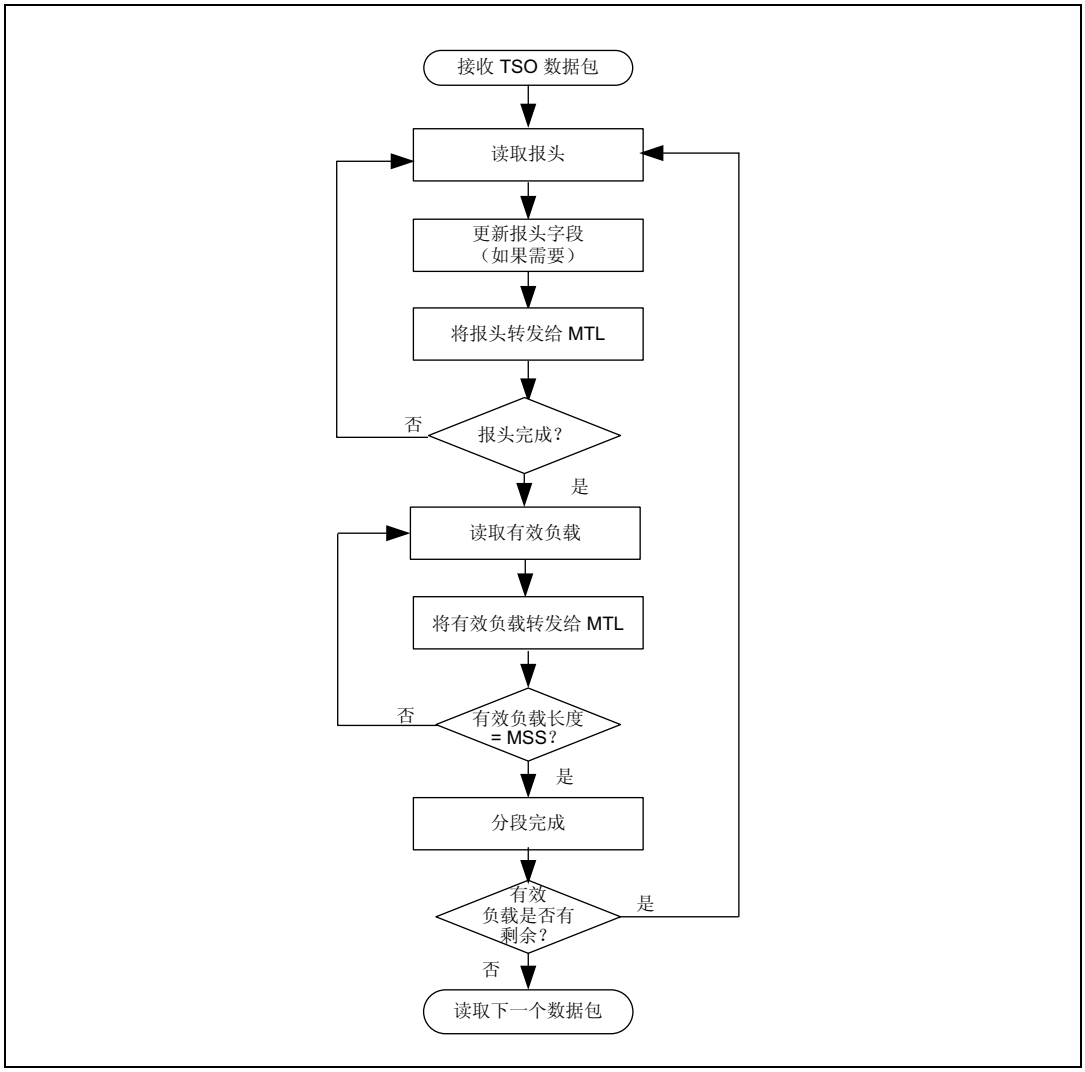
图 724. TCP 分段减荷概览



具有 TSO 功能的 DMA 操作

图 725 中显示了 TSO 流程。

图 725. TCP 分段减荷流程



对于 TSO 功能，Tx DMA 操作如下：

1. 在为数据缓冲区设置好相应的以太网数据包数据后，应用配置发送描述符 (TDES0–TDES3) 并将 Own 位 (TDES3[31]) 置 1。
2. 应用程序增大 DMA Tx 通道的描述符尾指针的偏移值。
3. 在运行状态下，DMA 会获取下一个可用描述符并执行下列其中一项操作：
   – 如果该描述符为上下文描述符，且上下文不在数据包的第一个描述符和最后一个描述符之间，则 DMA 会存储上下文值。
   – 如果该描述符为上下文描述符，且上下文位于数据包的第一个描述符和最后一个描述符之间，则 DMA 会关闭上下文描述符以表示上下文描述符错误 (TDES3[23])，并获取下一个描述符。
   – 如果该描述符为正常描述符，则 DMA 会检查此 TSE 位。如果 TSE 位未置 1，则 DMA 会继续使用默认的操作模式或执行 OSF 操作（如果使能）。
4. DMA 会根据 TCP 有效负载长度 (TDES3[17:0]) 和 MSS 值来计算段数。
5. DMA 会进行通道仲裁以获取数据缓冲区。DMA 会分别获取报头和有效负载。
6. 对于第一个段，DMA 将从系统内存中获取报头并将其存储在 TSO 内存中（前提是该报头存在且报头长度不超过 TSO 内存大小）。如果当前分段数据包并非第一个段，则该数据包会再次获取系统内存中的报头缓冲区，就像对第一个段所做的那样。在这种情况下，直到获取末段的报头之前，DMA 都不会关闭包含报头缓冲区的第一个描述符。
7. 系统会根据分段要求，修改/更新报头字节中的所需位字段，并写入相应的 MTL Tx 队列。
8. 然后 DMA 会获取有效负载缓冲区指针，从系统内存中获取 MSS 数量的有效负载字节，并直接将其推入 MTL Tx 队列。如果描述符中的缓冲区没有足够的数据用于 MSS 有效负载（末段除外），则 DMA 将关闭此描述符。
9. DMA 会跳到步骤 3 并重复这一过程，直到末段写入 Tx 队列为止。
10. DMA 关闭最后一个描述符和第一个描述符（如果未将其存储在 TSO 内存中，则其中将包含报头缓冲区），然后移动到下一个数据包传输。

如果有更多可用的描述符，则 DMA 会重复上述所有步骤。当没有更多可用的描述符时，DMA 会进入挂起状态。

注： *TSO 引擎会根据数据包第一个正常 Tx 描述符的 TDES3 中的 THL 位字段（TCP 报头长度）来决定是执行 TSO 操作还是 USO 操作。值 2 表示 USO，且任何大于或等于 5 的值均表示 TSO。*

TCP/IP 报头位字段

对 TCP 数据包进行分段时，DMA 会自动更新 TCP/IP 报头位字段。*表 544* 介绍了 TCP 和 IP 报头的更新过程。

表 544. TSO：TCP 和 IP 报头位字段

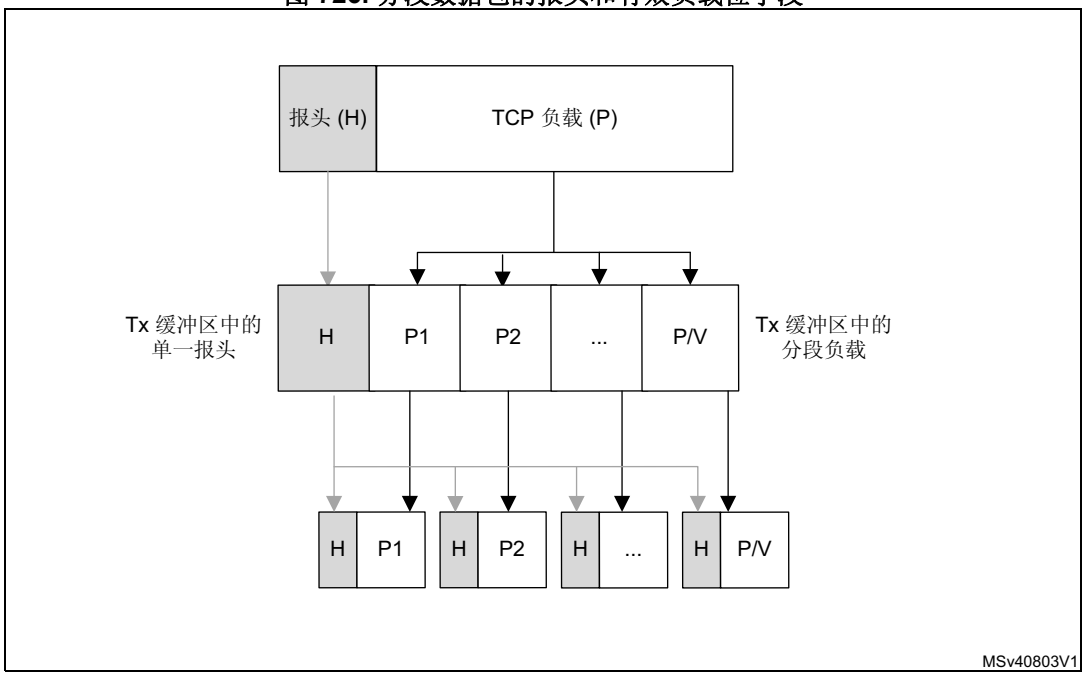
数据包序列	TCP 报头	IP 报头
第一个数据包	1. 不更新序列号。使用报头中提供的值。 2. 如果 FIN 和 PSH 标志置 1，则将其清零。 3. 重新计算 TCP 校验和。	IPv4 报头 – 总长度 = MSS + TCP 报头长度 + IP 报头长度 – 不修改标识位字段。按照软件提供的报头进行发送。 – 重新计算 IPv4 报头校验和。 IPv6 报头 – 有效负载长度 = MSS + TCP 报头长度 + IP 扩展报头长度
后续数据包	1. 更新序列号。将 MSS 值与前一段的序列号值相加。 2. 如果 FIN 和 PSH 标志置 1，则将其清零。 3. 重新计算 TCP 校验和。	IPv4 报头 – 总长度 = MSS + TCP 报头长度 + IP 报头长度 – 标识位字段 = 前一标识位字段 + 1 – 重新计算 IPv4 报头校验和 IPv6 报头 – 有效负载长度 = MSS + TCP 报头长度 + IP 扩展报头长度
最后一个数据包	1. 更新序列号。将 MSS 值与前一段的序列号值相加。 2. 如果 FIN 和 PSH 标志在原始报头中被置 1，则将这些标志置 1。 3. 重新计算 TCP 校验和。	IPv4 报头 – 总长度 = 剩余有效负载 + TCP 报头长度 + IP 报头长度 – 标识位字段 = 前一标识位字段 + 1 – 重新计算 IPv4 报头校验和 IPv6 报头 – 有效负载长度 = 剩余有效负载长度 + TCP 报头长度 + IP 扩展报头长度

分段数据包的报头和有效负载位字段

分段后，除了*表 544：TSO：TCP 和 IP 报头位字段*中所介绍的报头位字段以外，拆分数据包均与父 TCP 数据包使用相同的报头。*图 726：分段数据包的报头和有效负载位字段*显示了如何将同一报头用于各个分段数据包的报头位字段。

应用程序必须在要分段的数据包的第一个描述符的缓冲区 1 中创建报头，并在第一个描述符（FD = 1）的 TDES2 中提供报头长度。FD 位置 1 时，DMA 从 TDES0 指向的报头缓冲区读取报头。第一个描述符的缓冲区 2 可用于有效负载以及后续描述符的 TDES0 和 TDES1。对于后续描述符 (FD = 0)，TDES0 和 TDES1 所指向的地址被视为同一数据包的有效负载缓冲区地址。

图 726. 分段数据包的报头和有效负载位字段



上下文描述符序列

上下文描述符可提供用于分段的最大段大小 (MSS) 值。应用程序必须在使用相应 TCP 数据包的正常描述符之前提供上下文描述符。如果应用程序需要提供新的 MSS，则必须于描述符列表中在要使用新 MSS 值分段的数据包的第一个正常描述符之前，创建上下文描述符。只有在上下文描述符中 TDES3 的 TCMSSV 位置 1 且 OSTC 位复位（请参见*第 57.10.3 节：发送描述符*）时，上下文描述符中的 MSS 值才有效。

应用程序提供包含有效 MSS 值的上下文描述符时，DMA 会在内部存储 MSS 值，并通过 TDES3 正常描述符的 TSE 位将该值用于所有使能了 TSO 的后续数据包。

如果应用程序将一个上下文描述符置于数据包的中间位置（数据包的第一个描述符和最后一个描述符之间），则 DMA 会执行以下操作：

1. DMA 忽略上下文并关闭描述符。
2. DMA 在描述符状态中指示错误。
3. 如果在 DMA 通道对应的中断使能寄存器中将 CDEE 位置 1，则 DMA 会生成中断（请参见*通道 x 中断使能寄存器 (ETH_DMACxIER)*）。

应用程序可通过 DMA 通道对应的状态寄存器的 CDE 位来读取中断状态（请参见*通道 x 状态寄存器 (ETH_DMACxSR)*）。

构建用于 TSO 功能的描述符和数据包

要使能数据包分段功能，应用程序必须将第一个正常描述符（请参见*第 57.10.3 节：发送描述符*）的 TDES3 的 TSE 位置 1。如果在 TDES3 中针对非 TCP/IP 数据包将 TSE 位置 1，则 DMA 行为将无法预知。

应用程序必须在 TDES3[17:0] 中编程 TCP 数据包有效负载的长度，并在 TDES3[22:19] 中编程 TCP 报头的长度。可分段的 TCP 数据包有效负载的最大长度为 256 KB。

应在 TSO 数据包第一个正常描述符的缓冲区 1 中提供数据包报头，其中包括以太网报头、L3 报头和 L4 报头。只有为 TSO 使能的数据包的第一个正常描述符缓冲区 1 才被视为包含报头的缓冲区。

TCP 有效负载可从第一个正常描述符的缓冲区 2 开始，并持续到第二个正常描述符和后续描述符的缓冲区 1 和缓冲区 2。

TCP 有效负载可跨越多个缓冲区和多个描述符。包含 TCP 有效负载的缓冲区的总体大小应等于第一个正常描述符的 TDES3[17:0] 中提供的 TCP 有效负载长度。

MAC 始终会计算并附加 CRC，并为由 DMA 分段的所有数据包插入填充（如果需要）。如果使能 TDES3 的 TSE 位，则会保留 TDES3 的 CRC 填充控制 (CPC) 位字段。为确定分段后 TCP 数据包的大小，DMA 使用应用程序通过上下文描述符提供的最大段大小 (MSS)。DMA 会对有效负载大小大于 MSS 的数据包进行分段。应用程序必须通过编程 ETH_DMACxCR（请参见*通道 x 控制寄存器 (ETH_DMACxCR)*）中的 MSS 值或通过提供上下文描述符来提供 MSS。DMA 使用 MSS 的最后一个编程值或通过上下文提供的最后一个 MSS 值（以较后者为准）。

报头长度加上 MSS 大小（等于每个 TCP 段的大小）不应超过 16383 字节，否则 MAC 发送器会将数据包中 16383 字节之后的部分截断，以免导致出现 CRC 错误。

报头长度加上 MSS 大小再加上编程的 PBL 值（位于 ETH_DMACxTXCR 寄存器中，请参见*通道 x 发送控制寄存器 (ETH_DMACxTXCR)*）应小于在 ETH_MTLTxQxOMR 寄存器（请参见 *Tx 队列 x 工作模式寄存器 (ETH_MTLTXQxOMR)*）的 TQS 位字段中编程的 Tx 队列大小。建议使 MSS 加上报头的大小等于所编程 Tx 队列大小的一半。

DMA 也支持带 VLAN 标记的 TCP/IP 帧的分段。如果 TCP 数据包具有 VLAN 标记，则无论提供的是哪种 VLAN 标记类型（C-VLAN 或 S-VLAN），都会为所有段使用同一标记。VLAN 标记插入/替换控制位适用于所有段。

如果选择了双 VLAN 功能，则无论提供的是哪种 VLAN 标记类型（C-VLAN 或 S-VLAN），DMA 都会为所有段传递两个标记。两个标记的 VLAN 标记插入/替换控制位适用于所有段。如果未选择双 VLAN 功能，则应用程序不得针对带有两个标记的 TCP/IP 数据包将 TDES3 中的 TSE 位置 1，否则 DMA 将出现不可预测的行为。

如果数据包 TDES3 中的 TSE 位置 1 且所提供的 TCP 报头长度小于 5（这意味着它是无效的 TCP 报头，原因是它小于 20 字节），则 DMA 不会执行分段，而是将整个数据包作为单个数据包进行发送。在这种情况下，DMA 会将 CRC 填充控制位强制设置为 00（MAC 会执行 CRC 和填充）并将校验和插入控制位强制设置为 11（硬件对报头和有效负载都会进行校验和计算）。

57.5.7 IPv4 ARP 减荷

MAC 支持针对 IPv4 数据包的地址解析协议 (ARP) 减荷。使用该功能可在接收路径中处理 IPv4 ARP 请求数据包，并在发送路径中生成相应 ARP 响应数据包。

MAC 会为相应 ARP 请求数据包生成 ARP 响应数据包。IPv4 的 ARP 数据包为 L2 层数据包，其长度/类型为 0x0806。

ARP 减荷序列如下：

1. 如果请求目标协议地址与 MAC L3 寄存器中编程的 IPv4 地址匹配，则 MAC 接收器会获取 ARP 请求。
2. MAC 生成 ARP 响应数据包。
3. MAC 将 ARP 请求中的发送方硬件地址位字段复制到以下位字段：
   – 以太网数据包报头的 DA 位字段
   – ARP 响应数据包的目标硬件地址位字段
4. MAC 将 ARP 请求中的发送方协议地址位字段复制到 ARP 响应数据包中的目标协议地址位字段。
5. MAC 将其 MAC 地址置于以下位字段中：
   – 以太网数据包报头的 SA 位字段
   – ARP 响应数据包的发送方硬件地址位字段
6. MAC 将 ARP 请求中的目标协议地址位字段复制到 ARP 响应数据包中的发送方协议地址位字段。
7. MAC 将 ARP 响应数据包中的操作码位字段设为 2，以表示 ARP 响应。
8. MAC 重新计算 CRC，并对生成的 ARP 响应数据包进行填充。
9. MAC 发送器发送 ARP 响应

MAC 一次只会处理一个 ARP 请求。它不会存储多个 ARP 请求的位字段。如果 MAC 在收到 ARP 请求的时候已经开始处理先前的 ARP 请求，则 MAC 将不会为新的 ARP 请求生成 ARP 响应。MAC 会将新的 ARP 请求数据包转发给应用程序，并将“未生成 ARP 响应”状态位（位 34）置 1。然而，在掉电模式下，如果 MAC 在收到 ARP 请求的时候已经开始处理先前的 ARP 请求，则 MAC 会丢弃新的 ARP 请求。如果 *扩展工作模式配置寄存器 (ETH_MACECR)* 的禁止 CRC 检查 (DCRCC) 位置 1，则 MAC 不会检查 ARP 请求数据包的有效 CRC。如果其他条件有效，则它将可以生成 ARP 响应数据包。ARP 请求数据包必须始终具备有效的 CRC。

注： *当接收到的 ARP 请求小于 64 字节的数据包长度时，MAC 不会发送 ARP 响应。系统会将其视为正常数据包，并根据 MAC 过滤器设置将其转发到应用程序。*

57.5.8 回环

MAC 支持对发送到其接收器的数据包进行回环。

使用回环模式的指南

以下是一些有关使用回环模式的指南：

- 仅可在全双工模式下使能回环。在半双工模式下，系统会对载波侦听信号或冲突信号输入进行采样，这可能会导致诸如数据包丢弃等问题。
- 如果在不连接 PHY 芯片的情况下使能回环模式，则需要在外部生成 Tx 和 Rx 时钟，并将这些时钟提供给 MAC。
- 请勿回送大数据包，原因是它们可能会在回环 FIFO 中损坏。

发送时钟和接收时钟可能具有异步时序关系。因此，系统将使用异步 FIFO 将所发送数据的回环路径设为接收路径。FIFO 处于自由运行状态，可在写入时钟 (eth_mii_tx_clk) 上执行写入，并在每个读取时钟 (eth_mii_rx_clk) 上执行读取。开始从 FIFO 读取每个数据包时，系统会重新初始化写入和读取指针，以将偏移设为 2 或4（在 10/100 Mbps 模式下）。此举将避免数据包传输期间发生上溢或下溢，并确保上溢或下溢仅会在数据包之间的 IPG 阶段发生。将 FIFO 深度设为 5 或 9 就足以防止大小为 9,022 字节的数据包（其中 (G)MII 发送和接收时钟频率之间的差值为 200 ppm）发生数据损坏。

因此，用户不应回送较大的数据包，因为此类数据包可能会在回环 FIFO 中损坏。

在每个已接收数据包的末尾，接收协议引擎模块都会生成已接收数据包状态，并将其发送到接收数据包控制器模块。系统会将控制、丢失数据包和过滤失败状态添加到接收数据包控制器模块的接收状态中。MAC 不会处理回送的 ARP 或 PMT 数据包。

使能回环模式

要使能此功能，请对*工作模式配置寄存器 (ETH_MACCR)* 的 LM 位进行编程。用户可为所有 PHY 接口使能回环。无论选择哪个 PHY 接口，数据始终都会通过内部异步 FIFO 回送到内部接收 MII 或 GMII 接口。
回环数据也会通过相应的发送 PHY 接口模块传送到以太网线路上。

注： *在回环期间，数据/数据包会反射到 gmii_txd 信号上。*
回环模式不支持抢占。

57.5.9 流控

发送流控制包括在全双工模式下发送暂停数据包以及在半双工模式下提供背压功能，以便控制来自远程端的数据包流。本节介绍发送和接收路径的流控制。

全双工模式下的流控制

在全双工模式下，MAC 使用 IEEE 802.3x 暂停数据包进行流控制。*表 545* 介绍了暂停数据包的各个位字段。

表 545. 暂停数据包位字段

位字段	说明
DA	包含特殊多播地址
SA	包含 MAC 地址 0
类型	包含 8808
MAC 控制操作码	对于 IEEE 802.3x 暂停控制数据包，包含 0001
PT	包含在 *Tx 队列 0 流控制寄存器 (ETH_MACQ0TXFCR)* 的 PT 位字段中指定的暂停时间

FCB 位置 1 时，MAC 会生成并发送一个暂停数据包。如果在完成暂停数据包发送后再次将 FCB 位置 1，则无论暂停时间是否结束，MAC 都会发送另一个暂停数据包。要在先前发送的暂停数据包中指定的时间之前延长暂停时间或终止暂停，应用程序应使用合适的值对暂停时间寄存器进行编程，然后再次将 FCB 位置 1。

半双工模式下的流控制

在半双工模式下，MAC 使用延迟机制进行流控制（背压）。当应用程序请求停止接收数据包时，如果已使能发送流控制，则只要 MAC 检测到接收数据包，就会发送一个 32 字节的 JAM 信号。这会导致冲突并使远程站回退。如果应用程序请求发送数据包，则即使激活背压功能，也将按调度计划发送。如果背压功能长时间保持激活（发生的连续冲突事件超过 16 个），则远程站将因冲突过多而中止发送。

表 546 介绍了队列 0 的 Tx 路径中基于以下位设置的流控制：

- *Tx 队列 0 流控制寄存器 (ETH_MACQ0TXFCR)* 的 TFE 位
- *工作模式配置寄存器 (ETH_MACCR)* 的 DM 位

流控制对于所有队列均类似。

表 546. Tx MAC 流控制

TFE	DM	说明
0	X	MAC 发送器不执行流控制或背压操作。
1	0	将 *Tx 队列 0 流控制寄存器 (ETH_MACQ0TXFCR)* 的位 0 置 1 后，MAC 发送器会执行背压操作。
1	1	将 *Tx 队列 0 流控制寄存器 (ETH_MACQ0TXFCR)* 的位 0 置 1 后，MAC 发送器会发送暂停数据包。

发送流量控制

Tx 队列 0 流控制寄存器 (ETH_MACQ0TXFCR) 中的 TFE 位置 1 时，会使能发送流控制。

流控制触发

应用程序可通过将相应 *Tx 队列 0 流控制寄存器 (ETH_MACQ0TXFCR)* 中的 FCB 位置 1 来请求 MAC 发送暂停数据包或激活背压功能。

接收流量控制

在接收路径中，流控制仅在全双工模式下可用。如果在半双工模式下接收到暂停数据包，则该数据包会被视为正常控制数据包。

接收流控制的说明

表 547 介绍了 Rx 路径中基于以下位设置的流控制：

- *Rx 流控制寄存器 (ETH_MACRXFCR)* 的 RFE 位
- *工作模式配置寄存器 (ETH_MACCR)* 的 DM 位

表 547. Rx MAC 流控制

RFE	DM	说明
0	x	MAC 接收器未检测出接收的暂停数据包。
1	0	MAC 接收器未检测出接收的暂停数据包，而将这些数据包识别为控制数据包。
1	1	MAC 接收器检测出或已处理暂停数据包，并通过停止 MAC 发送器来响应这些数据包。

Rx 流控制的序列如下：

1. MAC 会基于以下任一参数检查已接收暂停数据包的目标地址 (DA)：
   - 多播目标地址：DA 将与为控制数据包指定的唯一多播地址 (0x0180 C200 0001) 相匹配。
   - 单播目标地址：DA 将与 MAC 地址 0 寄存器的内容（*MAC 地址 0 高位寄存器 (ETH_MACA0HR)* 和 *MAC 地址 x 低位寄存器 (ETH_MACAxLR)*）相匹配，并将 *Rx 流控制寄存器 (ETH_MACRXFCR)* 的 UP 位置 1。

     如果 UP 位置 1，则 MAC 除了会处理唯一多播地址之外，还会处理具有单播目标地址的暂停数据包。
2. MAC 将对已接收数据包的以下位字段进行解码：
   - 类型位字段：系统会检查此位字段是否为 0x8808。
   - 操作码位字段：系统会检查此位字段是否为 0x0001（暂停数据包）。
   - 暂停时间：系统将捕获暂停数据包中的暂停时间以确定发送器需要被中断的时间。
3. 如果状态的字节计数指示 64 个字节，并且不存在任何 CRC 错误，则 MAC 发送器将暂停任何数据包的发送，暂停时间为解码的暂停时间值乘以时隙（64 字节时间）。

如果系统在先前的暂停时间结束之前接收到了后续的暂停数据包，则 MAC 会用新值来更新暂停定时器。

使能接收流控制

设置 *Rx 流控制寄存器 (ETH_MACRXFCR)* 中的 RFE 位以使能暂停流控制。

57.5.10 MAC 管理计数器

外设支持在可通过应用程序访问的寄存器中存储关于已接收和已发送数据包的统计数据。

可将 MAC 管理计数器 (MMC) 模块中的计数器视为 CSR 模块的寄存器地址空间的扩展。MMC 模块保持一组寄存器来收集有关已接收数据包和已发送数据包的统计信息。这组寄存器包括一个用于控制各寄存器行为的控制寄存器、两个包含所生成中断的 32 位寄存器（接收和发送），以及两个包含中断寄存器屏蔽的 32 位寄存器（接收和发送）。这些寄存器可以通过 AHB 从接口从应用程序中进行访问，访问方式与 CSR 寄存器相同。这些寄存器的构成如*第 57.11.4 节：以太网 MAC 和 MMC 寄存器*中所示。

MMC 计数器自由运行。无需单独对计数器进行使能即可开始计数。当接收或发送相应数据包时，特定的 MMC 计数器就会开始计数。

接收 MMC 计数器会针对通过地址过滤器 (AFM) 模块传送的数据包进行更新。系统不会更新 AFM 模块丢弃的数据包的统计数据，除非该数据包为小于 6 字节的过短数据包（DA 字节未完全接收）。要获取所有数据包的统计数据，请将*数据包过滤控制寄存器 (ETH_MACPFR)* 中的位 0 置 1。MMC 模块会收集已接收以太网数据包中封装的 IPv4、IPv6、TCP、UDP 或 ICMP 有效负载的相关统计数据。

除了控制寄存器外，还实现了以下两组寄存器：

- 六个用于冲突、错误和良好数据包计数器的寄存器：
  - *Tx 单次冲突良好数据包寄存器 (ETH_TX_SINGLE_COLLISION_GOOD_PACKETS)*
  - *Tx 多次冲突良好数据包寄存器 (ETH_TX_MULTIPLE_COLLISION_GOOD_PACKETS)*
  - *Tx 良好数据包计数寄存器 (ETH_TX_PACKET_COUNT_GOOD)*
  - *Rx CRC 错误数据包寄存器 (ETH_RX_CRC_ERROR_PACKETS)*
  - *Rx 对齐错误数据包寄存器 (ETH_RX_ALIGNMENT_ERROR_PACKETS)*
  - *Rx 良好单播数据包寄存器 (ETH_RX_UNICAST_PACKETS_GOOD)*
- 四个用于记录 LPI 模式转变的寄存器：
  - *Tx LPI 微秒定时器寄存器 (ETH_TX_LPI_USEC_CNTR)*
  - *Tx LPI 转换计数器寄存器 (ETH_TX_LPI_TRAN_CNTR)*
  - *Rx LPI 微秒计数器寄存器 (ETH_RX_LPI_USEC_CNTR)*
  - *Rx LPI 转换计数器寄存器 (ETH_RX_LPI_TRAN_CNTR)*

定义

MMC 寄存器说明中将使用以下术语：

- 如果数据包成功发送，则将发送的数据包视为“好数据包”。换句话说，如果数据包发送过程没有因为以下错误而中止，则认为发送的数据包是好数据包：
  - Jabber 超时
  - 无载波或载波丢失
  - 延迟冲突
  - 数据包下溢
  - 过度延迟
  - 过度冲突

- 如果不存在以下错误，则认为接收数据包是“好数据包”：
  - CRC 错误 (CRC error)
  - 过短数据包（短于 64 字节）
  - 对齐错误（仅限 10/100 Mbps）
  - 长度错误（仅限非类型数据包）
  - 超出范围（仅限非类型数据包，长度超过 1518 字节）
  - GMII_RXER 输入错误
- 最大发送帧大小取决于帧类型，如下所示：
  - 无标记帧的最大大小 = 1,518
  - VLAN 帧的最大大小 = 1,522
  - 巨型帧的最大大小 = 9,018
  - 巨型 VLAN 帧的最大大小 = 9,022
- 最大接收数据包大小取决于数据包类型和控制位（JE、S2KP、GPSLCE 和 EDVLP），如*表 548* 中所示。

表 548. 最大接收数据包的大小

JE	S2KP	GPSLCE	EDVLP	未标记帧最大大小（以字节为单位）	单 VLAN 帧最大大小（以字节为单位）	双 VLAN 帧最大大小（以字节为单位）
1	X	X	1	9018	9022	9026
0	1	X	X	2000	2000	2000
0	0	1	1	GPSL	GPSL+4	GPSL+8
0	0	0	1	1518	1522	1526
1	X	X	0	9018	9022	9022
0	0	1	0	GPSL	GPSL+4	GPSL+4
0	0	0	0	1518	1522	1522

57.5.11 MAC 生成的中断

MAC 可由于各种事件而生成中断。这些中断事件与 DMA 中的事件结合用于 eth_sbd_intr_it 信号。MAC 中断为电平型中断，即在被应用程序或软件清除之前始终保持有效（高电平）。

中断状态寄存器 (ETH_MACISR) 描述了可导致 MAC 生成中断的事件。默认情况下使能 MAC 中断。可通过将 *中断使能寄存器 (ETH_MACIER)* 中的相应屏蔽位置 1，以阻止各个事件将 eth_sbd_intr_it 信号中断置为有效。

中断寄存器位仅指示报告事件的模块。必须读取相应的状态寄存器和其他寄存器才能清除中断。

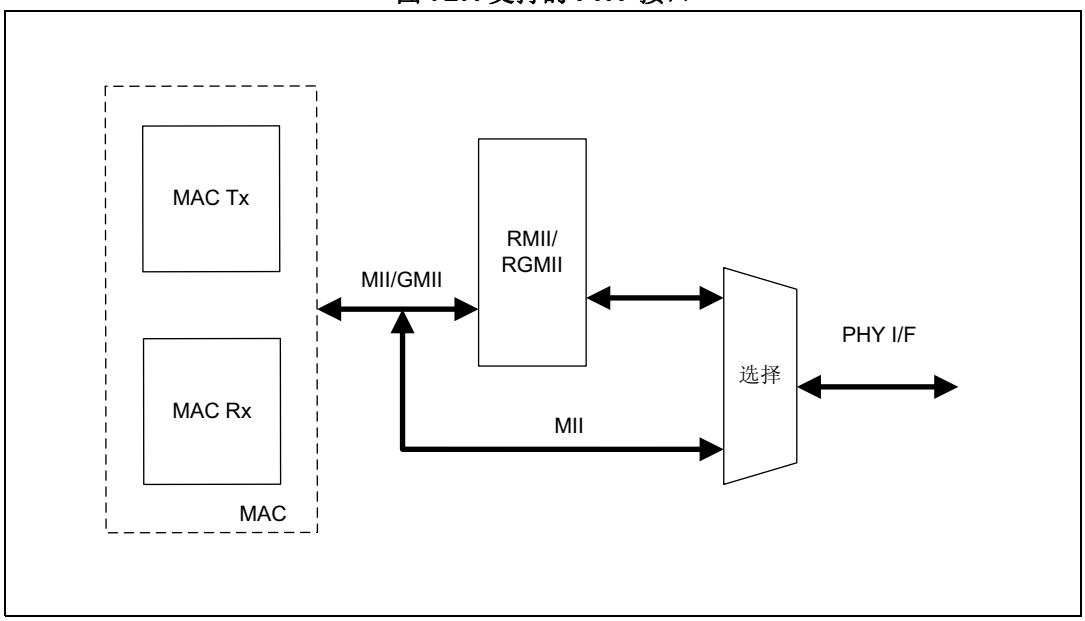
57.5.12 MAC 和 MMC 寄存器说明

请参见*第 57.11.4 节：以太网 MAC 和 MMC 寄存器*。

57.6 以太网功能描述：PHY 接口

以太网外设支持多个 PHY 接口。根接口为 MII/GMII 接口。所有其他接口均源自该接口，如*图 727* 所示。

图 727. 支持的 PHY 接口



本节介绍了用于 PHY 控制的 SMA 模块以及不同的 PHY 接口。它包含以下章节：

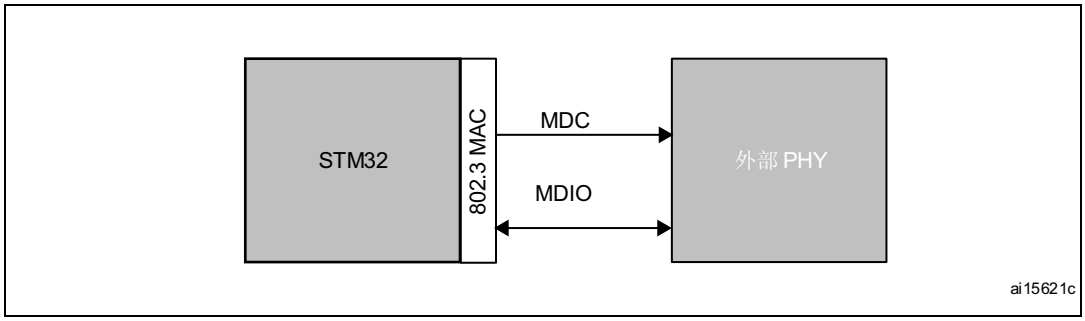
- *站管理代理 (SMA)*
- *介质独立接口 (MII)*
- *精简介质独立接口 (RMII)*
- *精简千兆介质独立接口 (RGMII)*

57.6.1 站管理代理 (SMA)

应用程序可通过站管理代理 (SMA) 模块访问 PHY 寄存器。SMA 包含一个双线站管理接口 (MIM)。

SMA 模块支持访问多达 32 个 PHY。应用程序可寻址 32 个 PHY 中的任意一个的 32 个寄存器之一。一次只能对一个 PHY 中的一个寄存器进行寻址。应用程序通过 SMA 模块向 PHY 发送控制数据以及从 PHY 接收状态信息，如*图 728* 所示。

图 728. SMA 接口模块



SMA 功能概述

MAC 基于 MDC 时钟触发管理写入操作或读取操作。MDC 时钟通过 CSR 时钟 (eth_hclk) 导出。按照 IEEE 802.3 规范中的规定，ETH_MDC 引脚的最大工作频率为 2.5 MHz。然而，如果系统支持更高的时钟频率，则可以选择其他的分频器。分频系数取决于 *MDIO 地址寄存器 (ETH_MACMDIOAR)* 寄存器中通过 CR[3:0] 设置的时钟范围。MDC 时钟的选择如下：

表 549. MCD 时钟选择

选择	MDC 时钟
0000	eth_hclk/42
0001	eth_hclk/62
0010	eth_hclk/16
0011	eth_hclk/26
0100	eth_hclk/102
0101	eth_hclk/124
0110、0111	-

MAC 与 PHY 之间的数据交换将通过三态缓冲器进行，并连接到 PHY 的 ETH_MDIO 线进行输出。

图 729 显示了条款 45 MDIO 数据包的结构，而*表 550* 则给出了数据包位字段的详细说明。

图 729. MDIO 数据包结构（条款 45）

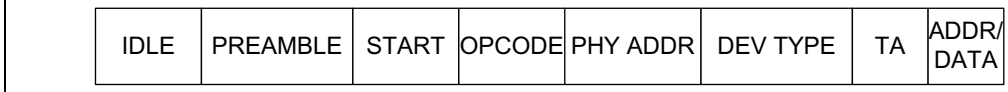
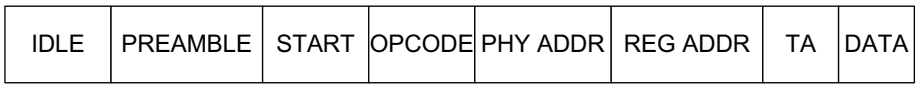


表 550. MDIO 条款 45 帧结构

位字段	说明
IDLE	ETH_MDIO 线为三态形式；ETH_MDC 无时钟。
PREAMBLE	32 个值为 1 的连续位
START	数据包起始为 0b00
OPCODE	– 0b00：地址 – 0b01：写 – 0b10：读取 + 地址 – 0b11：读取
PHY ADDR	32 个 PHY 之一的 5 位地址选择
DEV TYPE	5 位设备类型
TA	周转 – 0bZ0：读取和后读取增量地址 – 0b10：写入和地址 MDIO 访问 其中 Z 为三态电平
DATA	16 位值：对于地址周期 (OPCODE = 0b00)，此帧将包含下一周期要访问的寄存器的地址。对于写入帧的数据周期，该位字段将包含要写入寄存器的数据。对于读取或后读取增量地址帧，该位字段将包含从 PHY 读取的寄存器内容。 – 在地址和数据写入周期中，外设会在传输这 16 位期间驱动 ETH_MDIO 线。 – 在读取和后读取增量地址周期中，PHY 会在传输这 16 位期间驱动 ETH_MDIO 线。

此外，系统也会支持条款 22 帧的帧结构。可对 *MDIO 地址寄存器 (ETH_MACMDIOAR)* 中的 C45E 位进行编程，以使能符合条款 22 或条款 45 的操作模式。*图 730* 显示了条款 22 MDIO 数据包的结构，而*表 551* 则给出了数据包位字段的详细说明。

图 730. MDIO 数据包结构（条款 22）



MSv40806V1

表 551. MDIO 条款 22 帧结构

位字段	说明
IDLE	ETH_MDIO 线为三态形式；ETH_MDC 无时钟。
PREAMBLE	32 个值为 1 的连续位
START	数据包起始为 0b01
OPCODE	0b10 表示读取，0b01 表示写入
PHY ADDR	32 个 PHY 之一的 5 位地址选择
REG ADDR	所选 PHY 中的寄存器地址
TA	周转 – 0bZ0：读取和后读取增量地址 – 0b10：写入和地址 MDIO 访问 其中 Z 为三态电平
DATA	任意 16 位值。对于写操作，由 MAC 驱动 MDIO。对于读操作，则由 PHY 驱动。

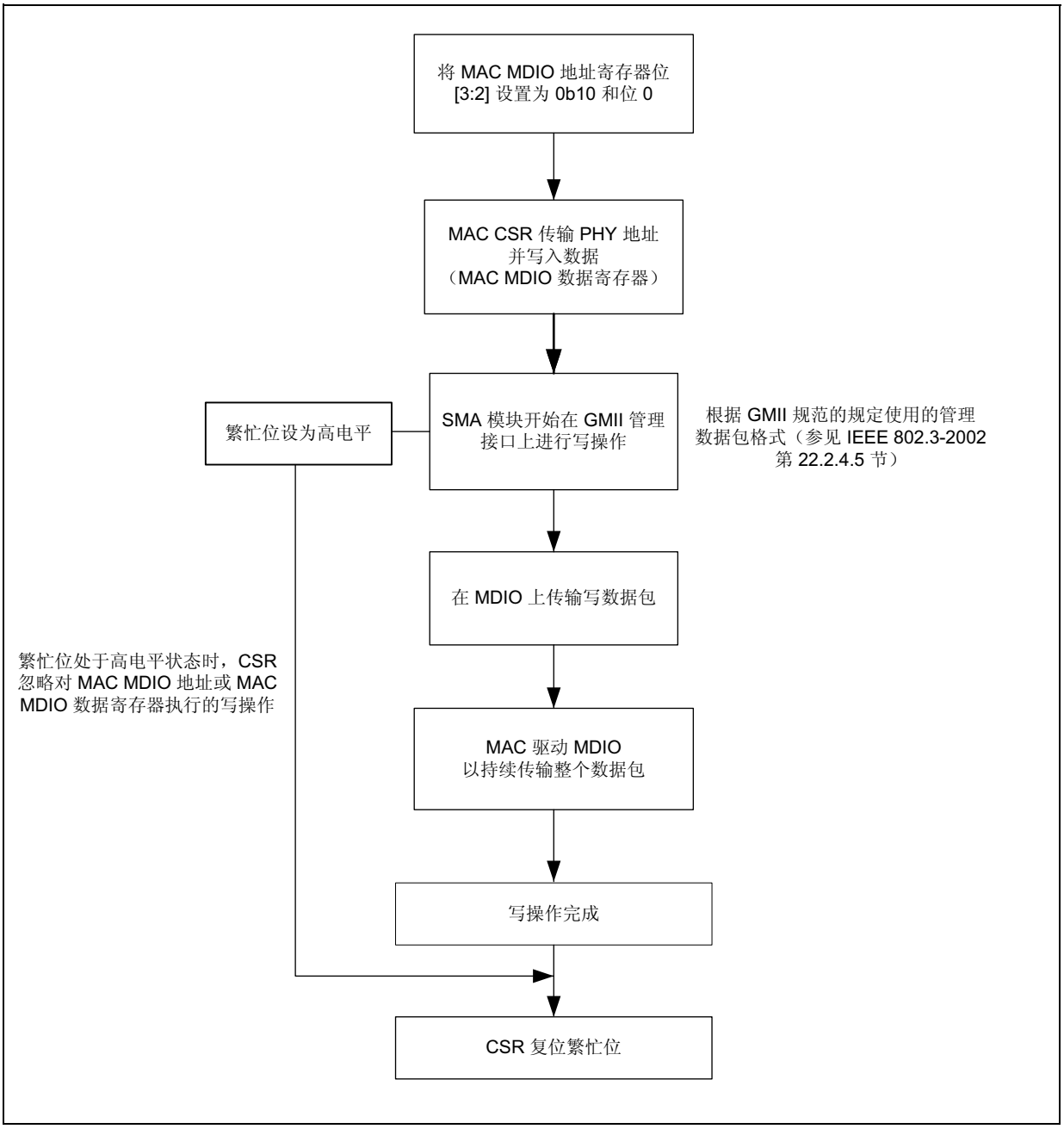
除了正常的读写操作外，SMA 在条款 45 模式下还支持后读取增量地址。

GMII/MII 管理写入操作

在站管理代理从 MAC CSR 模块接收到 PHY 地址和写入数据后，SMA 将开始对 PHY 寄存器进行写入操作。

图 731 中所示为从 SMA 模块到 PHY 寄存器的写入操作流程。

图 731. SMA 写入操作流程



将 *MDIO 地址寄存器 (ETH_MACMDIOAR)* 中的位 [3:2] 设为 01、位 0 设为 1 时，MAC CSR 模块会将 PHY 地址、PHY 中的寄存器地址和写数据（*MDIO 数据寄存器 (ETH_MACMDIODR)*）发送到 SMA，以触发对 PHY 寄存器的写入操作。此时，SMA 模块开始使用 GMII 规范（根据 IEEE 802.3-2002 规范的*第 22.2.4.5 节*）中指定的管理数据包格式通过 GMII 管理接口进行写入操作。

SMA 模块开始写入操作时，会通过 MDIO 线发送写数据包。MAC 会在数据包的整个持续时间内驱动 MDIO 线。写入操作完成之前，忙碌位始终处于高电平。在此期间（忙碌位为高电平时），CSR 会忽略对 *MDIO 地址寄存器 (ETH_MACMDIOAR)* 或 *MDIO 数据寄存器 (ETH_MACMDIODR)* 执行的写入操作。完成写入操作后，SMA 模块会向 CSR 提供相关指示，CSR 随后会将忙碌位复位。写入操作的数据包格式如下：

图 732. 写数据包

IDLE	PREAMBLE	START	OPCODE	PHY ADDR	REG ADDR	TA	DATA	IDLE
Z	1111..11	01	01	AAAAA	RRRRR	10	DDD...DDD	Z

MSv40807V1

GMII/MII 管理读取操作

将 *MDIO 地址寄存器 (ETH_MACMDIOAR)* 中的位 [3:2] 设为 11、位 0 设为 1 时，MAC CSR 模块会将 PHY 地址和 PHY 中的寄存器地址发送到 SMA，以触发对 PHY 寄存器的读取操作。此时，SMA 模块开始使用 GMII/MII 规范（根据 IEEE 802.3-2002 规范的 *第 22.2.4.5 节*）中指定的管理数据包格式通过 GMII/MII 管理接口进行读取操作。

当 SMA 模块开始在 MDIO 上进行读取操作时，CSR 会忽略在此期间对 *MDIO 地址寄存器 (ETH_MACMDIOAR)* 或 *MDIO 数据寄存器 (ETH_MACMDIODR)* 寄存器进行的写入操作（忙碌位为高电平），并且通过 MCI 接口无错完成事务。完成读取操作后，SMA 会向 CSR 提供相关指示。CSR 会将忙碌位复位，并会使用从 PHY 读取的数据更新 *MDIO 数据寄存器 (ETH_MACMDIODR)*。除了 PHY 驱动 MDIO 线时的相应数据位字段期间，MAC 会在帧的整个持续时间内驱动 MDIO 线。有关应用程序与 PHY 通信的更多信息，请参见 IEEE 802.3z 1000BASE 以太网的“协调子层和介质独立接口规范”部分。

读取操作的数据包格式如下：

图 733. 读取数据包

IDLE	PREAMBLE	START	OPCODE	PHY ADDR	REG ADDR	TA	DATA	IDLE
Z	1111..11	01	10	AAAAA	RRRRR	Z0	DDD...DDD	Z

MSv40808V1

前导码抑制

IEEE 标准将指定 MDIO 帧的 32 位前导码（全 1）。外设将提供控制，以支持前导码抑制。外设将发送仅包含 1 个前导码位的 MDIO 帧。可以将 *MDIO 地址寄存器 (ETH_MACMDIOAR)* 中的 PSE 位置 1，以使能前导码抑制。

尾部时钟和连续事务

外设会在 MDIO 帧持续期间驱动 ETH_MDC 时钟。空闲周期内不会驱动任何时钟。如果 PHY 需要在 MDIO 帧之后的一些周期内使 ETH_MDC 时钟保持激活状态，则可以使用尾部时钟功能。*MDIO 地址寄存器 (ETH_MACMDIOAR)* 中的 NTC[2:0] 位字段允许将尾部时钟从 0 编程到 7。

外设支持连续事务，即使前一个 MDIO 帧的尾部时钟尚未完成，也支持启动下一个 MDIO 帧。在尾部时钟功能处于使能状态的情况下，可通过将 *MDIO 地址寄存器 (ETH_MACMDIOAR)* 中的 BTB 位置 1 来使能此功能。在连续事务处于使能状态的情况下，GMII 忙碌位 (GB) 会在 MDIO 帧完成后立即清零。这样，软件便可发出下一个命令；该命令将由外设执行，同时前一个 MDIO 帧的尾部时钟仍然有效。在未使能 (GB) 事务的情况下，GMII 忙碌位会在 MDIO 帧的尾部时钟完成后清零。

MDIO 事务完成中断

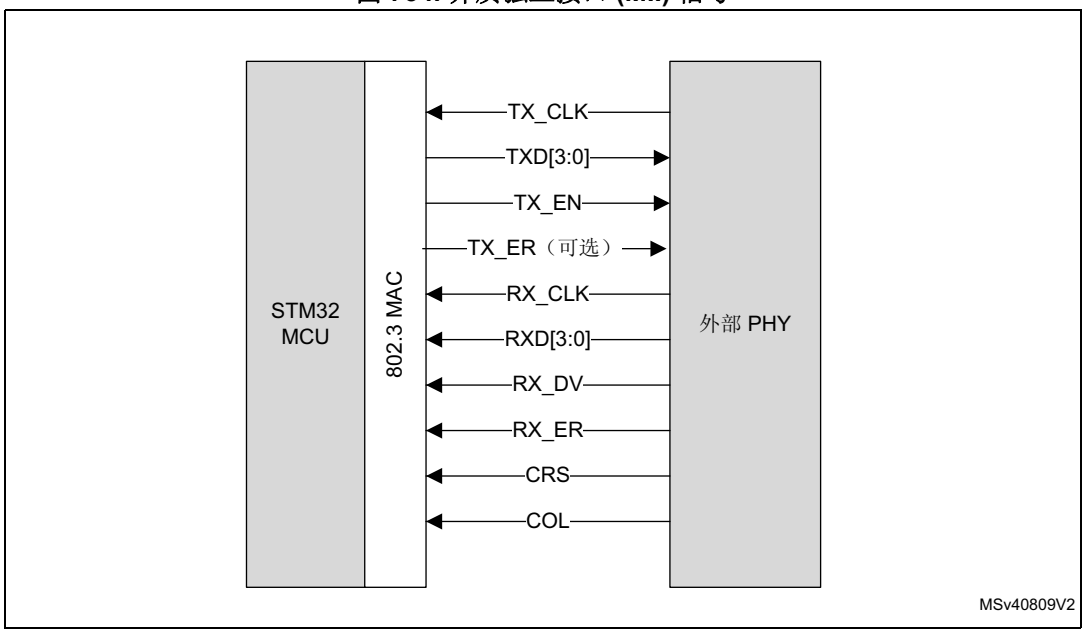
外设可在 MDIO 读取或写入事务完成时生成中断。因此，应用程序无需轮询 *MDIO 地址寄存器 (ETH_MACMDIOAR)* 的 GMII 忙碌位来了解 MDIO 命令的完成情况。

57.6.2 介质独立接口 (MII)

介质独立接口 (MII) 定义了 10 Mb/s 和 100 Mb/s 的数据传输速率下 MAC 子层与 PHY 之间的互连。

MII 信号如*图 734：介质独立接口 (MII) 信号*所示。

图 734. 介质独立接口 (MII) 信号



- TX_CLK：连续时钟信号，该信号提供进行 Tx 数据传输时的参考时序。速率为 10 Mb/s 时标称频率为 2.5 MHz；速率为 100 Mb/s 时标称频率为 25 MHz。
- TXD[3:0]：数据发送信号。

  TXD 是 4 个一组的数据信号，由 MAC 子层同步驱动，在 TX_EN 信号有效时才为有效信号（有效数据）。TXD[0] 为最低有效位，TXD[3] 为最高有效位。禁止 TX_EN 时，发送数据不会对 PHY 产生任何影响。
- TX_EN：发送使能信号，该信号表示 MAC 当前正针对 MII 发送半字节。该信号必须与前导码的前半字节进行同步 (TX_CLK)，并在所有待发送的半字节均发送到 MII 时必须保持同步。
- TX_ER（可选）：仅适用于节能型以太网 (EEE)。系统将通过反转 CRC 来指示发送错误。远程站可通过不正确的 CRC 来检测发送错误。
- RX_CLK：连续时钟信号，该信号提供进行 Rx 数据传输时的参考时序。速率为 10 Mb/s 时标称频率为 2.5 MHz；速率为 100 Mb/s 时标称频率为 25 MHz。
- RXD[3:0]：接收数据

  RXD 是 4 个一组的数据信号，由 PHY 同步驱动，在 RX_DV 信号有效时才为有效信号（有效数据）。RXD[0] 为最低有效位，RXD[3] 为最高有效位。当 RX_EN 禁止、RX_ER 使能时，特定的 RXD[3:0] 值用于传输来自 PHY 的特定信息。
- RX_DV：接收数据有效信号

  该信号表示 PHY 当前正针对 MII 接收已恢复并解码的半字节。该信号必须与恢复帧的头半字节进行同步 (RX_CLK)，并且一直保持同步到恢复帧的最后半字节。该信号必须在最后半字节随后的第一个时钟周期之前禁止。为了正确接收帧，RX_DV 信号必须在时间范围上涵盖要接收的帧，其开始时间不得迟于 SFD 位字段出现的时间。
- RX_ER：接收错误信号

  该信号必须保持一个或多个周期 (RX_CLK)，从而向 MAC 子层指示在帧的某处检测到错误。该错误条件必须通过 RX_DV 的有效情况进行验证。
- CRS：载波侦听信号。

  当发送或接收介质处于非空闲状态时，由 PHY 使能该信号。发送和接收介质均处于空闲状态时，由 PHY 禁止该信号。PHY 必须确保 CRS 信号在冲突条件下保持有效状态。该信号无需与 Tx 和 Rx 时钟保持同步。在全双工模式下，该信号没意义。
- COL：冲突检测信号

  检测到介质上存在冲突后，PHY 必须立即使能该信号，并且只要存在冲突条件，该信号就必须保持有效状态。该信号无需与 Tx 和 Rx 时钟保持同步。在全双工模式下，该信号没意义。

57.6.3 精简介质独立接口 (RMII)

精简介质独立接口 (RMII) 规范降低了以太网 PHY 与 STM32 MCU 之间的引脚数（仅限 10/100 模式）。根据 IEEE 802.3u，MII 包括 16 个数据和控制信号的引脚。RMII 规范将引脚数减少为 7 个。

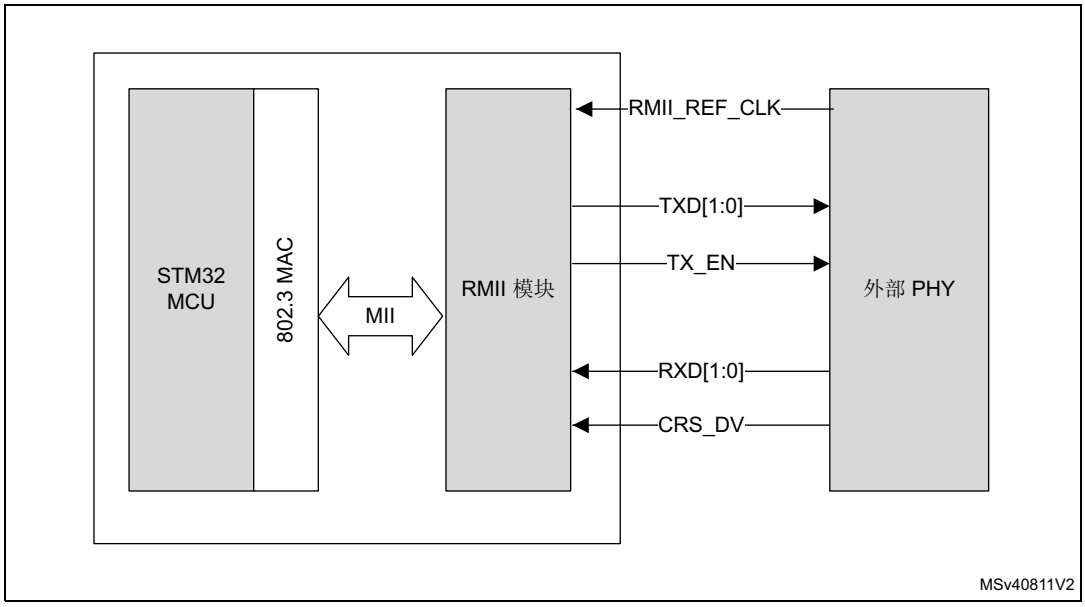
作为以太网外设的一部分，RMII 模块是 MAC 输出端的实例化对象。这有助于将 MAC 的 MII 转换为 RMII。RMII 具有以下特性：

- 支持 10 Mbps 和 100 Mbps 运行速率。不支持 1000 Mbps 运行速率。
- 通过在外部提供两个参考时钟来实现独立的 2 位宽发送和接收路径。

RMII 框图

*图 735：RMII 框图*显示了 RMII 模块相对于 MAC 和 RMII PHY 的位置。将 RMII 模块置于 MAC 之前，以将 MII 信号转换为 RMII 信号。

图 735. RMII 框图

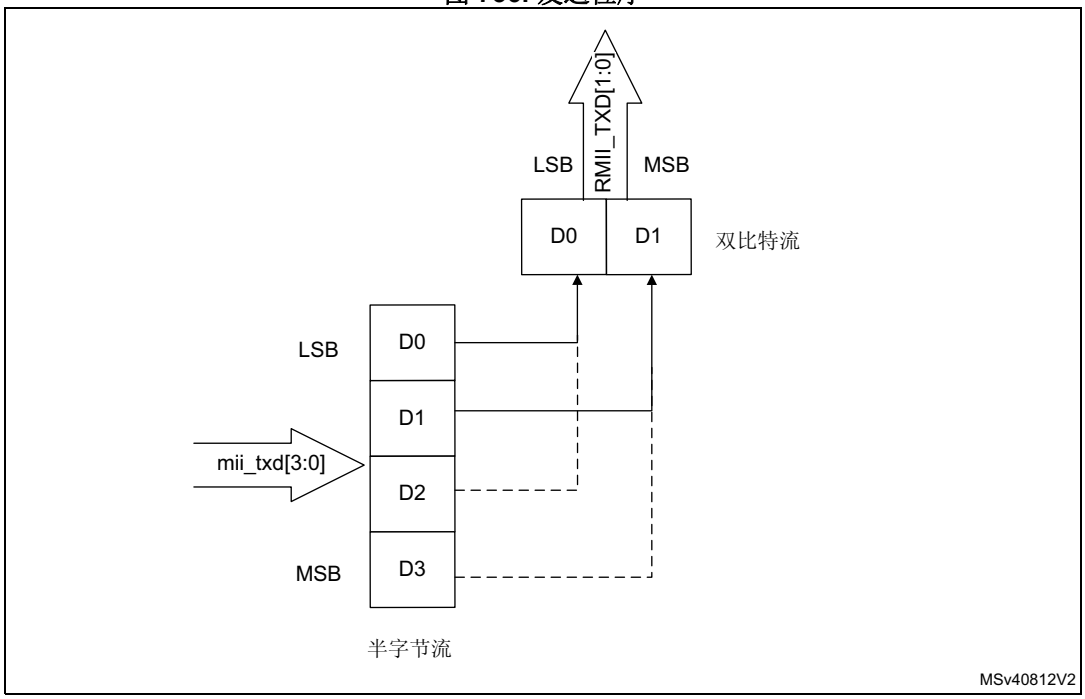


- RMII_REF_CLK：连续的 50 MHz 参考时钟输入
- TXD[1:0]：发送数据
- TX_EN：发送数据使能。
  该位为高电平时，表示正在 TXD[1:0] 上发送有效数据。
- RXD[1:0]：接收数据
- CRS_DV：在交替的时钟周期上复用载波侦听 (CRS) 和 RX_Data 有效 (RX_DV)。在 10 Mb/s 模式下，该操作每隔 10 个时钟周期交替一次。

发送位序

来自 MII 接口的每个半字节都在 RMII 接口上发送，一次发送双位，双位的发送顺序如*图 736：发送位序*所示。首先发送位序较低的位（D1 和 D0），再发送位序较高的位（D2 和 D3）。

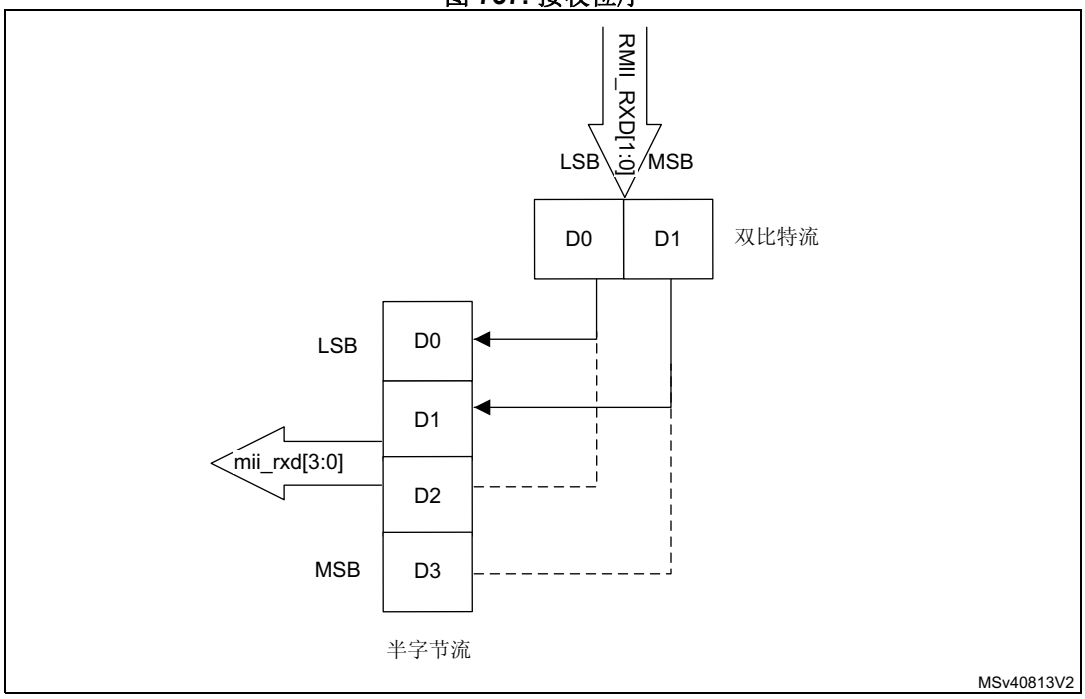
图 736. 发送位序



接收位序

每个半字节都从接收自 RMII 接口的双位发送到 MII 接口，半字节发送顺序如*图 737：接收位序*所示。先接收位序较低的位（D0 和 D1），再接收位序较高的位（D2 和 D3）。

图 737. 接收位序



57.6.4 精简千兆介质独立接口 (RGMII)

精简千兆介质独立接口 (RGMII) 规范降低了 MAC 与 GMII 和 MII 接口的 PHY 之间互连的引脚数。为此，系统将减少数据路径和控制信号，并与发送时钟和接收时钟的两个边沿一起复用。对于千兆运行，时钟将以 125 MHz 的频率运行；对于 10/100 运行，时钟频率将为 2.5/25 MHz。

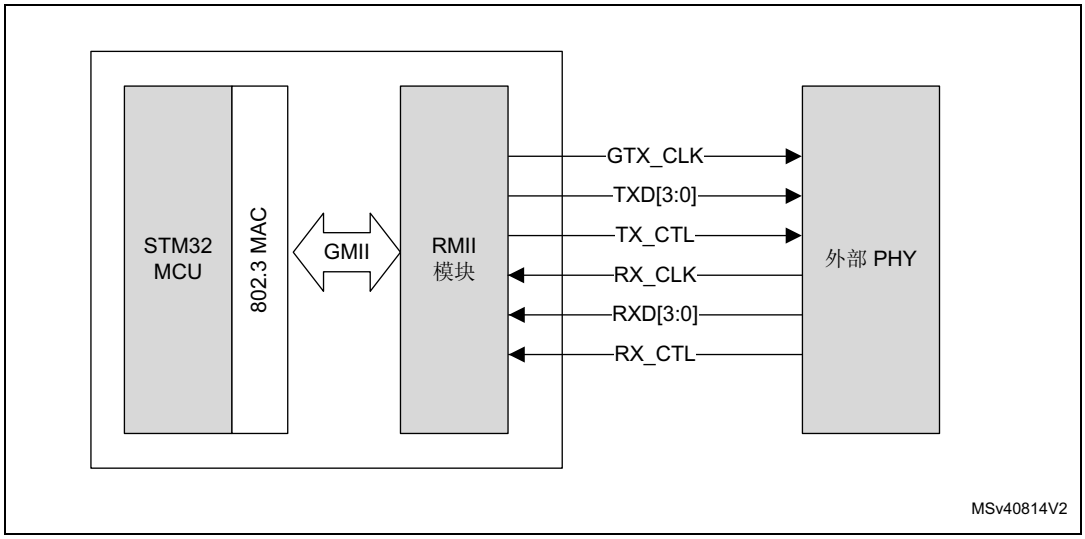
RGMII 模块将在 MAC 的 GMII 与 PHY 之间进行实例化，以便在 GMII 和 RGMII 协议之间转换控制信号和数据信号。RGMII 具有以下特性：

- 支持 10、100 和 1000 Mbps 的运行速率。
- 已经使用了传入时钟的两个边沿，因此不需要使用额外的时钟。
- 系统将从 PHY 中提取带内（链路速度、链路状态和双工模式）状态信号，并将其提供给 MAC 进行链路检测。

RGMII 框图

*图 738：RGMII 框图及接口信号*显示了 RGMII 模块相对于 MAC 和 RGMII PHY 的位置。RGMII 模块放置在 GMII 和 PHY 之间，用以将 GMII 信号转换为 RGMII 信号。

图 738. RGMII 框图及接口信号



- **发送路径**：该模块会将所有 GMII 发送信号转换为 RGMII 信号。该模块会在 eth_gmii_gtx_clk 的上升沿寄存所有 GMII 输入信号，并在 eth_gmii_gtx_clk 的上升沿和下降沿生成所有 RGMII 发送信号。
- **接收路径**：该模块会将所有 RGMII 接收信号转换为 GMII 接收信号。该模块会在 eth_gmii_rx_clk 的上升沿和下降沿寄存所有 RGMII 输入信号，并在 eth_gmii_rx_clk 的上升沿生成所有 GMII 接收信号。
- GTX_CLK：用于 Tx 数据传输的 RGMII 时钟，频率为 125 MHz、25 MHz 或 2.5 MHz
- TXD[3:0]：发送数据
- TX_CTL：在 GTX_CLK 的上升沿复用 TX_EN，并在 GTX_CLK 下降沿的 TX_EN 和 TX_ER 值之间进行逻辑异或运算
- RX_CLK：用于 Rx 数据传输的 RGMII 时钟
- RXD[3:0]：接收数据
- RX_CTL：解复用后，该引脚会在 ETH_RX_CLK 的上升沿提供 ETH_RX_DV，并提供 ETH_RX_ER（ETH_RX_CLK 上升沿和下降沿的 ETH_RX_CTL 值之间的逻辑异或运算结果）。

信号转换

RGMII 模块可执行双向数据转换。

发送数据转换

按照 RGMII 规范中的定义，千兆模式中的数据和控制复用将通过使用发送时钟的两个边沿来实现。RGMII 模块将在 eth_gmii_gtx_clk 的上升沿接受来自 MAC GMII (gmii_txd[7:0]) 的 8 位 GMII 发送数据。对于千兆运行，此后 RGMII 模块会将 gmii_txd[7:0] 转换为两个 4 位半字节，并在 RGMII 发送数据端口 (rgmii_txd_o[3:0]) 上发送数据，具体如下：

- 在 eth_gmii_gtx_clk 上升沿，rgmii_txd_o[3:0] 包含 gmii_txd[3:0]
- 在 eth_gmii_gtx_clk 下降沿，rgmii_txd_o[3:0] 包含 gmii_txd[7:4]

对于 10/100 模式，发送至 PHY 的数据是 4 位数据。因此，对于 10/100 模式，RGMII 模块会在 eth_gmii_gtx_clk 上升沿将 gmii_txd[3:0] 数据驱动至 rgmii_txd_o[3:0] 上的 PHY。

接收数据转换

在 1000-Mbps 模式下，RGMII 模块会在接收时钟 (eth_gmii_rx_clk) 的两个边沿上对 rgmii_rxd_i[3:0] 信号的 4 位数据进行寄存，并在 eth_gmii_rx_clk 时钟的上升沿向 MAC GMII 传输生成的 8 位数据，具体如下：

- gmii_rxd[3:0] 是在 eth_gmii_rx_clk 的上升沿于 rgmii_rxd_i[3:0] 上接收的值
- gmii_rxd[7:4] 是在 eth_gmii_rx_clk 的下降沿于 rgmii_rxd_i[3:0] 上接收的值

在 10/100 模式下，RGMII 模块仅会在 eth_mii_rx_clk 的上升沿上对 rgmii_rxd_i[3:0] 的 4 位数据进行寄存，并在 eth_gmii_rx_clk 的上升沿将数据传输到 MAC GMII，具体如下：

- gmii_rxd[3:0] 是在 eth_gmii_rx_clk 的上升沿于 rgmii_rxd_i[3:0] 上接收的值。
- gmii_rxd[7:4] 为 0x0。

57.7 以太网低功耗模式

57.7.1 低功耗管理

以太网外设支持以下节能技术：

- 魔术数据包
- 远程唤醒

当主机系统处于低功耗模式时，将使用魔术数据包和远程唤醒技术来节省电力，且系统只有在接收到来自以太网网络的特定数据包时才会唤醒。在低功耗模式下，大部分主机逻辑的时钟以及大多数外设（MAC 接收器逻辑除外）都可以禁止。接收到网络中的特定数据包后，MAC 会生成触发信号以唤醒主机系统并恢复到正常状态。请参阅参考手册中的电源控制 (PWR) 部分，以获取支持以太网低功耗模式的系统工作模式列表。

节能型以太网 (EEE) 模式符合 IEEE 802.3az-2010 标准。其主要用途是在线路上没有数据传输时节省以太网端口的电力消耗。在此模式下，主机会向远端指示自己在近期不会发送任何数据包，并将发送器端口（MAC 控制器、PCS 和 PHY 层）置于低功耗模式。同样，当远端主机指示自己没有任何数据需要传输时，也可将接收器端口置于低功耗模式。这样，以太网端口（主要在 PHY 中）就可以在间歇性和突发性数据传输配置下节省大量电能。将由 MAC 控制触发进入及退出 EEE 模式的操作，并在外设中予以支持。

此外，系统还支持将 EEE 模式与任意一种或两种其他节能模式同时运行。

魔术数据包模式说明

本节将介绍如何通过魔术数据包检测来节省电能。

注： *魔术数据包功能基于 AMD 公司推出的魔术数据包技术白皮书。*

魔术数据包的看门狗超时限制为 2,048 字节，该限制不受工作模式配置寄存器 (ETH_MACCR) 的 WD 位和看门狗超时寄存器 (ETH_MACWTR) 的 PWE 位中的编程值影响。

在基于魔术数据包的节能模式下，MAC 接收器接收到有效的魔术数据包时将会触发从低功耗模式退出的操作。当 *PMT 控制状态寄存器 (ETH_MACPCSR)* 的 PWRDWN 位编程为 1 时，MAC 将进入节能模式。用户可将 *PMT 控制状态寄存器 (ETH_MACPCSR)* 的 MGKPKTEN 位置 1，以使能从基于魔术数据包的节能模式退出。

魔术数据包在目标地址、源地址和长度/类型位字段后面的任何偏移中都包含了一个独特的模式。除了独特模式的匹配情况外，MAC 接收器还会检查以下内容，以便将接收到的数据包作为有效的魔术数据包进行检测：

- 必须将该数据包寻址到 MAC 接收器（已接收数据包的目标地址应与 *MAC 地址 0 高位寄存器 (ETH_MACA0HR)* 和 *MAC 地址 0 低位寄存器 (ETH_MACA0LR)* 完全匹配）或使用多播/广播地址。
- 该数据包不得存在任何长度错误、FCS 错误、dribble 位错误、GMII 错误和冲突。
- 该数据包不得过短（包含以太网报头和 FCS 的长度至少为 64 字节）。

魔术数据包数据格式

魔术数据包中独特模式的内容具体描述如下：

- 六个全 1 (0xFF FF FF FF FF FF) 字节，称为同步数据流。可能存在超过六个 0xFF 字节，但只考虑最后六个。
- 同步数据流之后紧跟着的是重复 16 次的数据包（*MAC 地址 0 高位寄存器 (ETH_MACA0HR)* 和 *MAC 地址 0 低位寄存器 (ETH_MACA0LR)*）目标地址位字段或多播/广播地址。
- 同步数据流与目标地址位字段的第一次重复之间或是在 16 次重复当中不得出现断开或中断。

如果节点的 MAC 地址为 0x00 11 22 33 44 55，则 MAC 扫描的数据序列如下：

目标地址 源地址 长度/类型 ..... FF FF FF FF FF FF
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
…CRC

远程唤醒数据包模式说明

本节将介绍基于远程唤醒数据包的节能模式。

注： *远程唤醒数据包功能的实现基于“设备类电源管理参考规范”以及各种特定于实现的白皮书。*

魔术数据包的看门狗超时限制为 2,048 字节，该限制不受工作模式配置寄存器 (ETH_MACCR) 的 WD 位和看门狗超时寄存器 (ETH_MACWTR) 的 PWE 位中的编程值影响。

在基于远程唤醒魔术数据包的节能模式下，MAC 接收器接收到预期的远程唤醒数据包时将会触发从低功耗模式退出的操作。当 *PMT 控制状态寄存器 (ETH_MACPCSR)* 中的 PWRDWN 位编程为 1 时，MAC 将进入节能模式。用户可将 *PMT 控制状态寄存器 (ETH_MACPCSR)* 的 RWKPKTEN 位编程为 1，以使能从基于远程唤醒魔术数据包的节能模式中退出。

MAC 会实现一个过滤器查找表（通过*远程唤醒数据包过滤寄存器 (ETH_MACRWKPFR)* 进行编程），其中编程了嵌入在远程唤醒数据包中的模式的 CRC、偏移和字节屏蔽以及过滤器操作命令。

嵌入在远程唤醒数据包中的模式将位于目标地址和源地址位字段后面的任何偏移中。除了该模式的 CRC 匹配情况外，MAC 接收器还将检查以下内容，以便将接收到的数据包作为有效的远程唤醒数据包进行检测：

- 必须将该数据包寻址到 MAC 接收器（已接收数据包的目标地址应与 *MAC 地址 0 高位寄存器 (ETH_MACA0HR)* 和 *MAC 地址 0 低位寄存器 (ETH_MACA0LR)* 完全匹配）或使用多播/广播地址。
- 该数据包不得存在任何长度错误、FCS 错误、dribble 位错误、GMII 错误和冲突。
- 该数据包不得过短（包含以太网报头和 FCS 的长度至少为 64 字节）。

当接收到有效的远程唤醒数据包时，MAC 接收器会将 *PMT 控制状态寄存器 (ETH_MACPCSR)* 中的 RWKPRCVD 位置 1，并在 pmt_intr_o 输出端口触发中断。当低功耗模式中未使能电源门控时，*中断状态寄存器 (ETH_MACISR)* 中的 PMTIS 位将置 1。当中断已使能（*中断使能寄存器 (ETH_MACIER)* 中的 PMTIE 位置 1）且未对低功耗模式中的 CSR 时钟进行门控关闭时，系统会在 sbd_intr_o 输出端口为应用程序触发中断。

远程唤醒数据包过滤器

当使能了基于远程唤醒的节能模式时，可以选择四个远程唤醒过滤器。远程唤醒过滤器的结构如*表 552：远程唤醒数据包过滤寄存器*中所示。

表 552. 远程唤醒数据包过滤寄存器

ETH_MACRWKPFR 值	位字段							
0	过滤器 0 字节屏蔽							
1	过滤器 1 字节屏蔽							
2	过滤器 2 字节屏蔽							
3	过滤器 3 字节屏蔽							
4	保留	过滤器 3 命令	保留	过滤器 2 命令	保留	过滤器 1 命令	保留	过滤器 0 命令
5	过滤器 3 偏移		过滤器 2 偏移		过滤器 1 偏移		过滤器 0 偏移	
6	过滤器 1 CRC - 16				过滤器 0 CRC - 16			
7	过滤器 3 CRC - 16				过滤器 2 CRC - 16			

*表 553：远程唤醒过滤器位字段的描述*中对远程唤醒过滤器的位字段进行了介绍。

表 553. 远程唤醒过滤器位字段的描述

寄存器	说明
过滤器 *i* 字节屏蔽	过滤器 *i* 字节屏蔽寄存器将定义过滤器 *i*（0、1、2 或 3）检测哪些数据包字节，以确定数据包是否为唤醒数据包。 – MSB（第 31 位）必须为零。 – 位 j[30:0] 为字节屏蔽。 – 如果字节屏蔽的位 j（字节号）置 1，则 CRC 模块会处理传入数据包的过滤器 *i* 偏移 + j；否则会忽略过滤器 *i* 偏移 + j。
过滤器 *i* 命令	4 位过滤器 *i* 命令控制过滤器 *i* 操作。 – 位 3 指定地址类型，用于定义模式的目标地址类型。 该位置 1 时，模式仅应用于多播数据包；该位复位时，模式仅应用于单播数据包。 – 位 2（反向模式）置 1 时，会反转 CRC16 散列函数信号的逻辑，以拒绝具有匹配 CRC_16 值的数据包。 位 2 与位 1 搭配使用，允许 MAC 通过创建过滤器逻辑（例如“模式 1 与模式 2 进行与非运算”）来拒绝远程唤醒数据包的子集。 – 位 1 (And_Previous) 实现布尔逻辑(1)。 该位置 1 时，当前条目的结果将与前一过滤器的结果进行逻辑与运算。使用此与逻辑运算，可以将屏蔽拆分到两个、三个或四个过滤器中，从而支持超过 32 个字节的过滤器模式。这具体取决于将 And_Previous 位置 1 的过滤器数目。 – 位 0 为过滤器 *i* 的使能位；如果位 0 未置 1，则将禁止过滤器 *i*。

表 553. 远程唤醒过滤器位字段的描述（续）

寄存器	说明
过滤器 *i* 偏移	此过滤器 *i* 偏移寄存器定义过滤器 *i* 要检测的数据包的偏移（在数据包内）。 – 该 8 位模式偏移是要检测的过滤器 *i* 的第一个字节的偏移。 – 允许的最小偏移为 12，指代数据包的第 13 个字节。 – 偏移值 0 指代数据包的第一个字节。
过滤器 *i* CRC-16	该过滤器 *i* CRC-16 寄存器包含根据模式计算的 CRC_16 值，以及对唤醒过滤寄存器模块编程的字节屏蔽。 – 使用以下多项式进行 16 位 CRC 计算： G(x) = x^16 + x^15 + x^2 + 1 散列函数计算中使用的每个屏蔽均与该屏蔽相关的 16 位值进行比较。每个过滤器均包含以下内容： – 32 位屏蔽：该屏蔽中的每个位均对应于所检测数据包中的一个字节。如果该位为 1'，则会将相应的字节用于 CRC16 计算。 – 8 位偏移指针：指定开始 CRC16 计算的字节。 指针和屏蔽一起用来定位用于 CRC16 计算的字节。

1. And_Previous 位设置适用于一组四个过滤器。
将未使能的过滤器的 And_Previous 位置 1 无效，也就是说，将一组四个过滤器中编号最小的过滤器的 And_Previous 位置 1 将会无效。例如，过滤器 0 的 And_Previous 位置 1 无效。
如果将给定过滤器的 And_Previous 位置 1 以形成与链形式的过滤器，则与链会在找到已禁止过滤器的时候断开。例如：如果过滤器 2 的 And_Previous 位置 1（过滤器 2 命令的位 1 置 1），但过滤器 1 未使能（过滤器 1 命令的位 0 复位），则只会考虑过滤器 2 的结果。如果过滤器 2 的 And_Previous 位置 1（过滤器 2 命令的位 1 置 1），过滤器 3 的 And_Previous 位置 1（过滤器 3 命令的位 1 置 1），但过滤器 1 未使能（过滤器 1 命令的位 0 复位），则只会考虑过滤器 2 结果与过滤器 3 结果的与运算结果。如果过滤器 2 的 And_Previous 位置 1（过滤器 2 命令的位 1 置 1），过滤器 3 的 And_Previous 位置 1（过滤器 3 命令的位 1 置 1），但过滤器 2 未使能（过滤器 2 命令的位 0 复位），则由于过滤器 2 的 And_Previous 位置 1 无效，只会考虑过滤器 1 结果与过滤器 3 结果的或运算结果。
如果通过将 And_Previous 位置 1 形成的过滤器链中存在互补性设定，则帧绝不会通过与链过滤器。例如，如果过滤器 2 的 And_Previous 位置 1（过滤器 2 命令的位 1 置 1），并且过滤器 1 的 Address_Type 位置 1（过滤器 1 命令的位 3 置 1，指示多播检测）、过滤器 2 的 Address_Type 位复位（过滤器 2 命令的位 3 复位，指示单播检测），或反之，则远程唤醒帧不会通过与链过滤器，因为远程唤醒帧不能同时属于单播和多播地址类型。

远程唤醒过滤寄存器将实现为四个远程唤醒过滤器的八个间接访问寄存器 (wkuppktfilter_reg#i)，并通过*远程唤醒数据包过滤寄存器 (ETH_MACRWKPFR)* 由应用程序进行访问。必须写入整套 wkuppktfilter_reg 寄存器，以编程远程唤醒过滤器。系统将分别针对 wkuppktfilter_reg0 至 wkuppktfilter_reg3 依次写入*远程唤醒数据包过滤寄存器 (ETH_MACRWKPFR)* 中的八个寄存器值，以此对 wkuppktfilter_reg 寄存器进行编程。读取 wkuppktfilter_reg 寄存器时也采用类似方式。MAC 会在 *PMT 控制状态寄存器 (ETH_MACPCSR)* 的 RWKPTR 位字段中更新 wkuppktfilter_reg 寄存器当前指针值。

注： *如果在字节模式或半字模式下访问远程唤醒数据包过滤寄存器 (ETH_MACRWKPFR)，则当 CPU 访问通道 3 时，用于访问相应 wkuppktfilter_reg 的内部计数器会递增。*

当写入远程唤醒数据包过滤寄存器 (ETH_MACRWKPFR) 后，内容将在执行写入操作后从 CSR 时钟域传输至 PHY 接收时钟域。在 PHY 接收时钟域中更新首次写入之前，不应再对远程唤醒数据包过滤寄存器 (ETH_MACRWKPFR) 进行进一步写入。否则，第二次写入操作将不会更新到 PHY 接收时钟域。因此，对远程唤醒数据包过滤寄存器 (ETH_MACRWKPFR) 进行两次写入操作之间的延迟应至少为 4 个 PHY 接收时钟周期。

PMT 中断

接收到有效远程唤醒数据包时，PMT 中断信号会置为有效。

表 554 列出了会生成 PMT 中断的远程唤醒情况。

表 554. 远程唤醒数据包和 PMT 中断生成(1)

过滤器 i 命令			帧类型和 CRC 状态		中断生成
CAST	INV	EN	接收到的帧播型	CRC 状态	RWK INTR
0	0	1	单播	匹配	检测到远程唤醒数据包并生成 PMT 中断
0	1	1	单播	不匹配	检测到远程唤醒数据包并生成 PMT 中断
1	0	1	多播	匹配	检测到远程唤醒数据包并生成 PMT 中断
1	1	1	多播	不匹配	检测到远程唤醒数据包并生成 PMT 中断

1. 在所有其他组合中，均未检测到远程唤醒数据包，并且不会生成 PMT 中断。

除了 sbd_intr_o 信号外，pmt_intr_o（与 Rx 时钟同步）信号也会生效。系统将提供与 Rx 时钟域同步的 pmt_intr_o 信号，以便在 MAC 进入掉电模式时，应用程序的时钟可以通过软件停止。

pmt_intr_o 信号会在 PHY Rx 时钟域中生成，因此该信号不会在读取 *PMT 控制状态寄存器 (ETH_MACPCSR)* 时立即清零。这是因为生成的清零信号必须跨越到 PHY Rx 时钟域，然后清除中断源。该延迟至少为 Rx 时钟的 4 个时钟周期，且当外设在 10 Mbps 模式下运行时可能具有重要意义。当应用程序将*远程唤醒数据包过滤寄存器 (ETH_MACRWKPFR)* 中的 PWRDWN 位清零时，MAC 会退出掉电模式，但此事件不会生成 PMT 中断。

掉电序列

软件必须执行以下任务以启动掉电序列：

- 通过将*通道 x 发送控制寄存器 (ETH_DMACxTXCR)* 的 ST 位清零来禁止发送 DMA（如适用）。
- 等待完成之前的所有帧发送过程。可通过读取所有 MTL Tx 队列的*调试寄存器 (ETH_MACDR)* 中的 TFCSTS[1:0] 和 TPESTS 位以及 *Tx 队列 x 调试寄存器 (ETH_MTLTXQxDR)* 中的 TXQSTS 位来对此进行检查。
- 通过将*工作模式配置寄存器 (ETH_MACCR)* 中的 TE 位和 RE 位清零来禁止 MAC 发送器和 MAC 接收器。
- 等待接收 DMA 清空 Rx FIFO 中的所有帧。可通过读取所有 Rx 队列的 *Rx 队列 x 调试寄存器 (ETH_MTLRXQxDR)* 中的 PRXQ[13:0] 来对此进行检查。如果这些位为零，则表示 Rx FIFO 为空。
- 在 *PMT 控制状态寄存器 (ETH_MACPCSR)* 中配置魔术数据包 (MGKPKTEN) 和/或远程唤醒 (RWKPKTEN) 检测。
- 将*工作模式配置寄存器 (ETH_MACCR)* 中的位 31 (ARPEN) 置 1。
- 通过将 RE 位置 1 来使能 MAC 接收器，然后将 *PMT 控制状态寄存器 (ETH_MACPCSR)* 中的 PWRDWN 位置 1 以启动 MAC 的掉电序列。

注： *如果使能了该功能，且 MAC 发送器在进入掉电模式时处于 LPI 模式，则 MII 接口将受到箝制，以将 LPI 模式置为有效。如果 MAC 发送器在进入掉电模式时未处于 LPI 模式，则 GMII 或 MII 接口将钳位至全零。*

上电序列

MAC 会在接收到魔术数据包或远程唤醒帧时唤醒。上电序列如下：

- MAC 将 pmt_intr_o 置为有效。当在低功耗模式中只使用时钟门控时，可以使用 pmt_intr_o 信号来启动在进入低功耗模式后门控关闭的时钟。
- 软件将执行以下任务：
  - 通过读取 *PMT 控制状态寄存器 (ETH_MACPCSR)* 将 pmt_intr_o 置为无效。
  - （使用复位值）对 *PMT 控制状态寄存器 (ETH_MACPCSR)* 和*远程唤醒数据包过滤寄存器 (ETH_MACRWKPFR)* 执行写入操作，以便对始终开启的模块中的相应值进行同步。否则，这些寄存器的值将会不同。
  - 对*工作模式配置寄存器 (ETH_MACCR)*、*MAC 地址 0 高位寄存器 (ETH_MACA0HR)* 和 *MAC 地址 0 低位寄存器 (ETH_MACA0LR)* 执行写入操作，以将 CSR 模块中的值与始终开启的模块中的相应位同步。否则，即使接收使能位置 0，MAC 接收器仍将处于开启状态。

完成这些步骤后，软件必须初始化所有寄存器，使能发送器，并（在 DMA 配置中）对 DMA 进行编程以恢复正常工作。

57.7.2 节能型以太网 (EEE)

EEE 是一种工作模式，可使 IEEE 802.3 介质访问控制 (MAC) 子层和一系列物理层在低功耗空闲 (LPI) 模式下运行。EEE 工作模式支持 IEEE 802.3 MAC 在 100 Mbps 和 1000 Mbps 的速率下运行。该外设支持面向 EEE 的 IEEE 802.3az-2010 标准。

当不存在要发送和接收的数据时，LPI 模式允许通过关闭一部分通信设备功能来实现节能。链路两端的系统可在链路利用率较低的时段禁止某些功能，以节省电能。MAC 会控制系统进入或退出 LPI 模式，并通知 PHY。

EEE 指定能力协商方法，链路伙伴可使用指定的方法来决定是否支持 EEE，然后可选择一组对于两个设备通用的参数。

发送路径功能

发送路径功能中包含了为使 PHY 进入 LPI 状态，MAC 必须执行的任务。

在发送路径中，软件必须将 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 的 LPIEN 位置 1，以指示 MAC 停止发送并启用 LPI 协议。如果链路在 *LPI 定时器控制寄存器 (ETH_MACLTCR)* 的 LPI LS TIMER LST[9:0] 位字段中指定的时间内一直保持连通状态，则 MAC 将完成正在进行的发送，为其生成发送状态，并开始发送 LPI 模式而非 IDLE 模式。*LPI 控制和状态寄存器 (ETH_MACLCSR)* 的 PHY 链路状态 PLS 位将指示 PHY 的链路状态。

注： *将 MAC 配置为使用 RMII 时，不支持 EEE 功能。*

按照标准 (IEEE 802.3az-2010)，在 MII（10 或 100）模式下，PHY 在处于 LPI 状态时不得停止 TxCLK 时钟。然而，在 GMII (1000) 代码中，MAC 可以在 LPI 状态下停止 GTX_CLK 时钟。

要使 PHY 进入 LPI 状态，MAC 应执行以下任务：

1. 将 TX_EN 信号置为无效。
2. 将 TX_ER 信号置为有效。
3. 将 TXD[3:0] 设置为 0x1（适用于 100 Mbps）或将 TXD[7:0] 设置为 0x01（适用于 1,000 Mbps）。
4. 更新状态（*LPI 控制和状态寄存器 (ETH_MACLCSR)* 的 TLPIEN 位），然后生成中断。

注： *在 PHY 保持 LPI 状态的整个时间内，MAC 都会将 TX_EN、TX_ER 和 TXD 信号保持在相同的状态。*

要使 PHY 退出 LPI 状态（即当软件复位 LPIEN 位时），MAC 会执行以下任务：

1. 停止发送 LPI 模式并开始发送 IDLE 模式。
2. 启动 LPI TW 定时器：

   MAC 需在为 PHY 指定的唤醒时间结束之后才能开始发送。自动协商的唤醒间隔在 *LPI 定时器控制寄存器 (ETH_MACLTCR)* 的 TWT 位字段中编程。
3. 更新 LPI 退出状态（*LPI 控制和状态寄存器 (ETH_MACLCSR)* 的 TLPIEX 位），然后生成中断。

图 739 显示了 LPI 模式转变期间，TX_EN、TX_ER 和 TXD[3:0] 信号的行为。

注： *MAC 不会停止 TX_CLK 时钟。如果 PHY 支持该时钟且 MAC 将 sbd_tx_clk_gating_ctrl_o 信号置 1，则应用程序可以停止该时钟（如图 739 中所示）。sbd_tx_clk_gating_ctrl_o 信号会在九个 Tx 时钟周期、一次脉冲同步器延迟和一个 CSR 时钟周期后生效。sbd_tx_clk_gating_ctrl_o 信号生效与否要取决于 LPI 控制和状态寄存器 (ETH_MACLCSR) 的 LPITCSE 位，且生效过程将自动完成（如图 740 中所示）。*

如果选择了 RGMII 接口，则需要通过 Tx 时钟来发送 LPI 模式，因此不能对 Tx 时钟进行门控。

如果 MAC 处于 Tx LPI 模式且 Tx 时钟已停止，则应用程序不应写入与 Tx 时钟域同步的 CSR 寄存器。

如果 MAC 处于 LPI 模式且应用程序发出软复位或硬复位，MAC 发送器将退出 LPI 模式。

图 739. LPI 转换（发送，100 Mbps）

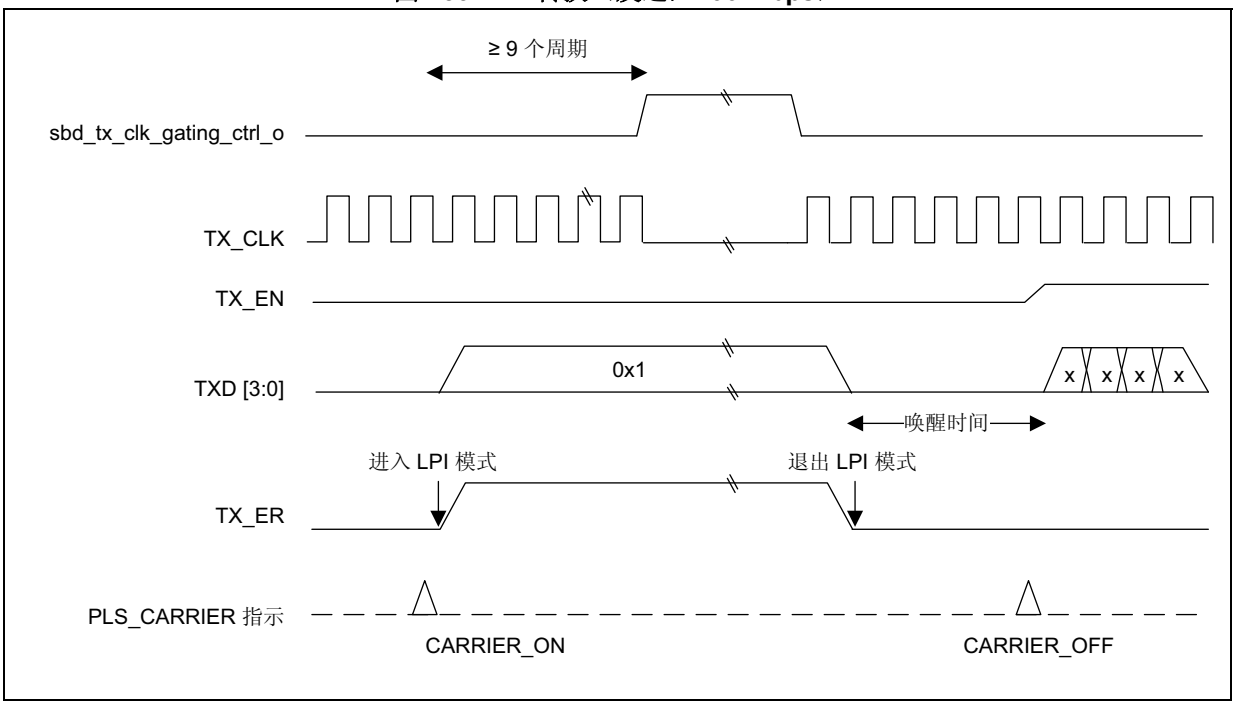
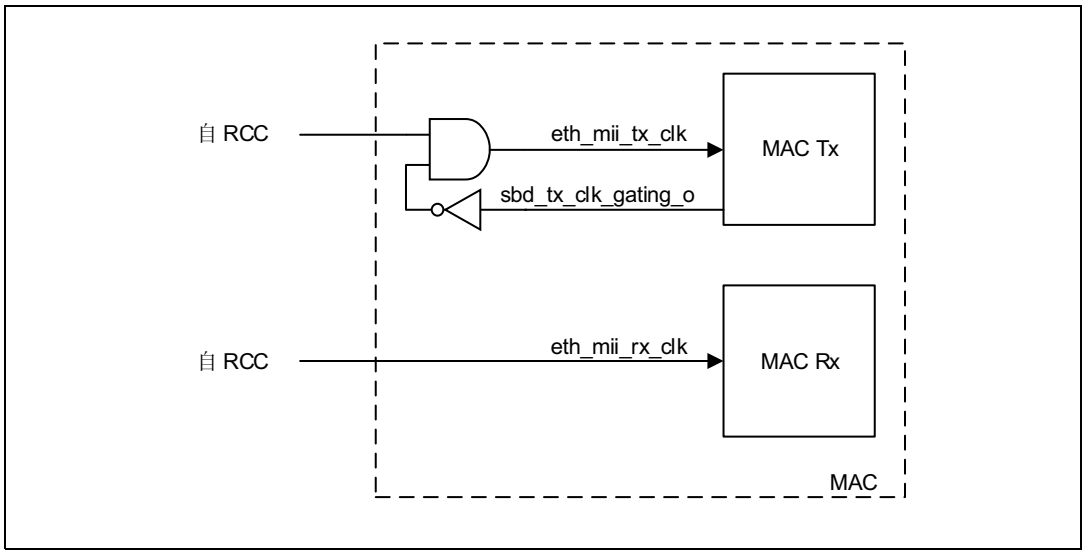


图 740. LPI Tx 时钟门控（LPITCSE = 1 时）



在发送路径中自动进入/退出 LPI 模式

MAC 发送器经过编程后，可在长时间处于空闲状态时自动进入 LPI 空闲模式，并在有数据包要传送时自动退出 LPI 空闲模式。这些模式通过 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 进行使能和控制。

如果 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 的 LPITXA 和 LPIEN 置 1，MAC 发送器会在 MAC 发送路径（包括 MTL 层和 DMA 层）处于空闲状态时进入 LPI 空闲状态。TX 路径中的任一功能（DMA、MTL 或 MAC）因触发数据包传送而变为非空闲状态后，MAC 发送器会立即退出 LPI 空闲状态并将 LPITXEN 位清零。

此外，如果 LPITE 也置 1，则只有在发送路径保持空闲状态（无活动）的时间达到 *LPI 进入定时器寄存器 (ETH_MACLETR)* 中的值所指示的时间时，MAC 发送器才会进入 LPI 空闲状态。在这种模式下，只要有任一功能变为非空闲状态，MAC 发送器就会立即退出 LPI 空闲状态。但是，LPIEN 位不会被清零，而是继续保持有效状态，以便在 MAC 再次空闲时无需任何软件干预即可重新进入 LPI 空闲状态。

LPITE 和 LPITXA 位均清零时，应用程序可通过编程 LPIEN 位直接控制进入和退出 LPI 空闲状态。

接收路径功能

接收路径功能中包含了在 PHY 从链路伙伴接收信号以退出 LPI 状态时，PHY 和 MAC 必须执行的任务。

在接收路径中，当 PHY 从链路伙伴接收到进入 LPI 状态的信号时，PHY 和 MAC 将执行以下任务：

1. PHY 将 RX_ER 置为有效。
2. PHY 将 RXD[3:0] 设置为 0x1（适用于 100 Mbps）或将 RXD[7:0] 设置为 0x01（适用于 1,000 Mbps）。
3. PHY 将 RX_DV 置为无效。
4. MAC 将更新 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 的 RLPIEN 位并立即生成中断。

注： *在 PHY 保持 LPI 状态的整个时间内，PHY 都会将 RX_ER、RXD 和 RX_DV 信号保持在相同的状态。*

如果检测到 LPI 模式持续时间非常短（即小于两个 Rx 时钟周期），则 MAC 不会进入 Rx LPI 模式。

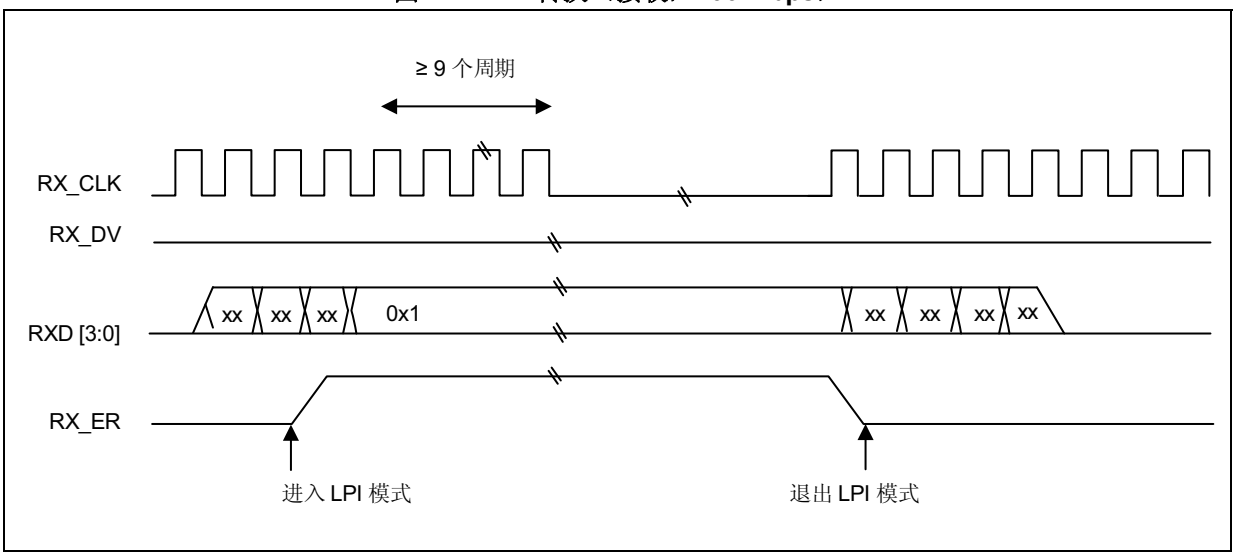
如果从当前 Rx LPI 模式结束到下一个 Rx LPI 模式开始的持续时间非常短（即小于两个 Rx 时钟周期），则 MAC 将退出并再次进入 Rx LPI 模式。MAC 不会触发 Rx LPI 退出和进入中断。

当 PHY 从链路伙伴接收到退出 LPI 状态的信号时，PHY 和 MAC 将执行以下任务：

1. PHY 将 RX_ER 置为无效并返回到正常数据包间状态。
2. MAC 会更新 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 的 RLPIEX 位并立即生成中断。边带信号 lpi_intr_o（与 Rx 时钟同步）也会生效。

图 741 显示了 LPI 模式转变期间，RX_ER、RX_DV 和 RXD[3:0] 信号的行为。

图 741. LPI 转换（接收，100 Mbps）



注： *如果当 PHY 向 MAC 指示 LPI 时 RX_CLK_stoppable 位（位于通过 MDIO 写入的 PHY 寄存器中）生效，则 PHY 可以在 LPI 状态开始超过九个时钟周期后的任意时间停止 RX_CLK，如图 741 所示。*

如果 MAC 处于 LPI 模式且应用程序发出软复位或硬复位，则 MAC 接收器将在复位期间退出 LPI 模式。如果将复位置为无效后仍然接收到 LPI 模式，则 MAC 接收器将再次进入 LPI 状态。

如果在 RX LPI 模式下停止了 RX 时钟，则应用程序不应写入正在与 RX 时钟域同步的 CSR 寄存器。

当 PHY 发送 LPI 模式时，如果使能了 EEE 功能，则 MAC 会自动进入 LPI 状态。没有软件控制可以阻止 MAC 进入 LPI 状态。

LPI 定时器

发送器将保留 LPI LS TIMER、LPI TW TIMER 和 LPI AUTO ENTRY TIMER 定时器。

以下 LPI 定时器加载了来自 *LPI 定时器控制寄存器 (ETH_MACLTCR)* 和 *LPI 进入定时器寄存器 (ETH_MACLETR)* 的相应值：

- LPI LS TIMER

  LPI LS TIMER 将以毫秒为单位计数自链路状态变为连通起的过期时间。可以将 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 的 PLSEN 位置 1，以使能对 *PHYIF 控制状态寄存器 (ETH_MACPHYCSR)* 的 LUD 位的监视。

  链路状态将通过 *PHYIF 控制状态寄存器 (ETH_MACPHYCSR)* 的 LNKSTS 位或软件在 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 的 PLS 位中编程的值来指示给 MAC。如果 *PHYIF 控制状态寄存器 (ETH_MACPHYCSR)* 中链路状态不可用，则软件应通过读取 PHY 寄存器来获取 PHY 链路状态，并相应更新 PLS 位。

  每次链路断开时，此定时器都会清零。一旦链路重新连通，该定时器就会开始持续递增，直到定时器的值与端子计数相等为止。一旦达到端子计数，则只要链路处于连通状态，定时器就会保持在同一个值。端子计数是在 *LPI 定时器控制寄存器 (ETH_MACLTCR)* 的 LST[9:0] 位字段中编程的值。除非达到端子计数，否则 GMII 接口不会将 LPI 模式置为有效。这就确保了在与远程站建立链路后，存在一个任何 LPI 模式都不会生效的最短时

间。IEEE 802.3-az-2010 标准中将这段时间定义为 1 秒。LPI LS TIMER 的数据宽度为 10 位。软件至多可以编程为 1023 毫秒。

- LPI TW TIMER

  LPI TW TIMER 将以微秒为单位计数自 LPI 置为无效起的过期时间。应在 *LPI 定时器控制寄存器 (ETH_MACLTCR)* 的位[15:0] 中对端子计数进行编程。定时器的端子计数是解析后的发送 TW 值，即 MAC 在此后可恢复正常发送操作的自动协商时间。退出 LPI 模式后，MAC 将在 TW 定时器达到端子计数后恢复正常工作。

  MAC 支持以微秒为单位表示 LPI TW TIMER。LPI TW TIMER 的数据宽度为 16 位。因此，软件最多可以编程至 65535 微秒。

- LPI AUTO ENTRY TIMER

  该定时器将以八微秒为步长进行计数，以统计在 MAC 发送器进入 LPI IDLE 状态并开始发送 LPI 模式之前，MAC 发送路径必须保持在空闲状态（无活动）下的时间。此定时器将在 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 中的 LPITE 位置 1 时使能。

注： *在 GMII 和 MII 模式之间切换之前，请将 LPI 控制和状态寄存器 (ETH_MACLCSR) 的 PLS 位编程为 1'b0。此举将复位内部定时器。如果在 LPI LS TIMER 或 LPI TW TIMER 启动后更改了该模式，则 Tx 时钟频率的变化可能会导致超时错误。*

LPI 中断

当 Tx 或 Rx 端进入或退出 LPI 状态时，MAC 会生成 LPI 中断。当 LPI 中断状态位置 1 时，中断 sbd_intr_o 将生效。通过对 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 进行读取操作，可将 LPI 中断清除。

当 MAC 退出 Rx LPI 状态时，除了 sbd_intr_o 之外，边带信号 lpi_intr_o（与 Rx 时钟同步）也会生效。lpi_intr_o 信号可用于触发外部时钟门控电路，以将应用程序时钟恢复至 MAC。系统将提供与 Rx 时钟域同步的 lpi_intr_o 信号，以便在 MAC 进入 LPI 状态时，应用程序的时钟可以通过软件停止。

lpi_intr_o 信号将在 Rx 时钟域中生成。在读取 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 之后，可能无法立即将其清除。这是因为在 CSR 时钟域中生成的清零信号必须跨越 Rx 时钟域，然后才能清除中断源。该延迟至少为 Rx 时钟的四个时钟周期，且当外设在 10 Mbps 模式下运行时可能具有重要意义。

关于节能型以太网的编程指南

有关编程指南的详细指导原则，请参见 *第 3113 页的第 57.9.12 节：关于节能型以太网 (EEE) 的编程指南*。

57.8 以太网中断

以太网外设可生成一个中断信号 (eth_sbd_intr_it)。该信号可作为各种事件的结果而生成。这些事件被捕获到状态寄存器中，并且每个中断源都配有一个中断使能位，因此事件的中断信号只有在相应的中断使能位置 1 时才会置为有效。

中断状态和相应的使能寄存器以层级结构进行组织，以便软件更加轻松地快速遍历和识别中断事件源。触发中断时，*中断状态寄存器 (ETH_DMAISR)* 寄存器将作为第一级指示，指示中断事件源对应的主要模块。该寄存器为只读寄存器，其中包含与每个 DMA 通道（TX & RX 对）、MTL 和 MAC 相对应的位。软件应用程序随后必须读取置 1 的位所对应的以下寄存器中的一个（或多个）：

- ETH_DMAC0SR：通道 0 状态（请参见*通道 x 状态寄存器 (ETH_DMACxSR)*）
- ETH_DMAC1SR：通道 1 状态（请参见*通道 x 状态寄存器 (ETH_DMACxSR)*）
- ETH_MTLISR：中断状态（请参见*中断状态寄存器 (ETH_MTLISR)*）
- ETH_MACISR：中断状态（请参见*中断状态寄存器 (ETH_MACISR)*）

57.8.1 DMA 中断

中断寄存器说明

ETH_DMAC0SR：通道 0 状态寄存器（请参见*通道 x 状态寄存器 (ETH_DMACxSR)*）捕获 TxDMA 和 RxDMA 通道的所有中断事件。ETH_DMAC0IER：通道 0 中断使能寄存器（请参见*通道 x 中断使能寄存器 (ETH_DMACxIER)*）包含对应于每个中断事件的使能位。

DMA 通道中有两组中断，即正常中断和异常中断。它们分别由 ETH_DMAC0SR 寄存器的位 [15:14] 进行指示。正常组对应于正常数据包传送期间发生的事件（TI：发送中断，RI：接收中断，TBU：发送缓冲区不可用），而异常中断事件则对应于错误事件。

中断不会进行排队。如果在驱动程序对上一个事件做出响应之前再次发生相同的中断事件，则不会再生成中断。对于同时发生的多个事件，只生成一次中断。驱动程序必须扫描*中断状态寄存器 (ETH_DMAISR)* 以明确中断原因，并清除相应状态寄存器中的中断源。只有在*中断状态寄存器 (ETH_DMAISR)* 的所有位均清零时才会清除中断。

定期调度发送和接收中断

出于系统吞吐量性能的原因，最好不要针对 DMA 传送的每个数据包都生成中断（RI 和 TI）。通过以太网外设可使用以下两种方法灵活地定期调度中断：

1. 每当要发送的数据包达到“所需”数量时，便将发送描述符中的完成时中断位（*表 558：TDES2 正常描述符（读取格式）*中的 TDES2[31]）置 1。
2. 类似地，仅以特定的间隔将接收描述符中的 IOC（*表 571：RDES3 正常描述符（读取格式）*中的 RDES3[30]）置 1。这样，只有在完成将接收到的数据包传送到系统内存，并且用于该数据包传送的所有描述符均将 IOC 位置 1 时，才会生成 RI 事件。

此外，还提供了中断定时器（ETH_DMAC0RXIWTR：通道 0 Rx 中断看门狗定时器），以供灵活控制和定期调度接收中断。当此中断定时器使用非零值编程时，只要 Rx DMA 完成将接收的数据包传送到系统内存并且未触发接收中断（因为相应的完成中断 IOC 位，即*表 571：RDES3 正常描述符（读取格式）*中的 RDES3[30] 未置 1），此定时器即会被激活。当此定时器按照编程值结束计时时，如果使能了 ETH_DMAC0IER 寄存器（请参见*通道 x 中断使能寄存器 (ETH_DMACxIER)*）中的相应 RIE，则 RI 位会置 1 并会引发中断。如果针对描述符的 IOC 已置 1 的数据包传送将 RI 置 1，则定时器会停止并清零。如果下一个数据包传送完成后未生成 RI 事件，则会自动重新激活定时器。

每通道传送完成中断

发送传送完成中断 (TI) 和接收传送完成中断 (RI) 通过通道状态寄存器（*通道 x 状态寄存器 (ETH_DMACxSR)*）进行指示。只要 TX DMA 通道关闭了 IOC 位置 1（完成时中断 - TDES2[31]）的描述符，TI 位便置 1。类似地，只要 Rx DMA 通道关闭了 LD 位置 1 的描述符，RI 位便置 1，并且用于传送该数据包的任何描述符中的 IOC 位均置 1（完成时中断使能 - RDES3[30]）。

只有在通道中断使能寄存器（*通道 x 中断使能寄存器 (ETH_DMACxIER)*）中使能了相应的中断时，才会针对传送完成中断将中断信号置为有效。

RI/TI 中断的行为随 ETH_DMAMR 寄存器（*DMA 模式寄存器 (ETH_DMAMR)*）中的 INTM 位字段（位[17:16]）设置而变化。*表 555* 对传送完成中断的行为进行了说明。

表 555. 传送完成中断行为

中断模式	TI/RI 和 eth_sbd_intr_it 的行为
INTM=0	检测到传送完成事件时，TI/RI 状态信号会被置 1。软件驱动程序向这些位写入 1 时，这些位会被清零。如果还在 ETH_DMACxIER 寄存器中使能了相应的中断，则 eth_sbd_intr_it 会被置为有效。
INTM=1	如上所述，TI/RI 置 1。不过，对于任一 RI/TI 事件，都不会触发 eth_sbd_intr_it。
INTM=2	RI/TI 状态位会在检测到传送完成事件时置 1，并且会在软件驱动程序通过写入 1 来将这些位清零时复位。不过，如果在软件将其清除（处理）之前检测到另一个传送完成事件，则会自动再次将这些状态位置 1。但是，并非基于 TI/RI 生成 eth_sbd_intr_it。

57.8.2 MTL 中断

MTL 中断事件与 DMA 中的事件结合用于生成中断信号。

寄存器 *中断状态寄存器 (ETH_MTLISR)* 报告负责事件的队列号。ETH_MTLQ0ICSR：队列 0 中断控制状态或 ETH_MTLQ1ICSR：队列 1 中断控制状态必须被读取以获取事件说明。

默认情况下使能 MTL 中断。可通过将 *中断状态寄存器 (ETH_MTLISR)* 寄存器中的相应屏蔽位置 1，以阻止各个事件将中断置为有效。

MTL 中断信号由队列 0 和队列 1 上的以下事件之一驱动：

- 接收队列上溢中断
- 发送队列下溢

57.8.3 MAC 中断

MAC 中断事件与 DMA 中的事件结合用于生成中断信号。

MAC 中断为电平型中断，即在被应用程序或软件清除之前始终保持有效（高电平）。

中断状态寄存器 (ETH_MACISR) 描述了可导致 MAC 生成中断的事件。默认情况下使能 MAC 中断。可通过将 *中断状态寄存器 (ETH_MACISR)* 中的相应屏蔽位置 1，以阻止各个事件将中断置为有效。

中断寄存器位仅指示报告事件的模块。必须读取相应的状态寄存器和其他寄存器才能清除中断。

MAC 中断信号由以下事件之一驱动：

- 接收状态中断
- 发送状态中断
- 时间戳中断状态
- MMC 中断状态
  - MMC 接收校验和减荷中断状态
  - MMC 发送中断状态
  - MMC 接收中断状态
- LPI 中断状态
- PMT 中断状态
- PHY 中断
- RGMII 中断状态

注： *以下两个边带信号与 LPI 和 PMT 中断一起生成：lpi_intr_o 和 pmt_intr_o。这两个信号用于在 EXTI 级进行唤醒事件检测。*

57.9 以太网编程模型

本章提供了按正确顺序初始化 DMA 或 MAC 寄存器的相关说明。它包含以下章节：

- DMA 初始化（请参见*第 57.9.1 节*）
- MTL 初始化（请参见*第 57.9.2 节*）
- MAC 初始化（请参见*第 57.9.3 节*）
- 执行正常接收和发送操作（请参见*第 57.9.4 节*）
- 停止和开始发送（请参见*第 57.9.5 节*）
- 关于多通道多队列操作的编程指南（请参见*第 57.9.8 节*）
- 关于 GMII 链路状态转换的编程指南（请参见*第 57.9.9 节*）
- 关于 IEEE 1588 时间戳的编程指南（请参见*第 57.9.10 节*）
- 关于节能型以太网的编程指南（请参见*第 57.9.12 节*）
- 关于每秒脉冲数 (PPS) 灵活的输出的编程指南（请参见*第 57.9.13 节*）
- 关于 TSO 的编程指南（请参见*第 57.9.14 节*）
- 关于接收端 VLAN 过滤的编程指南（请参见*第 57.9.15 节*）

57.9.1 DMA 初始化

要初始化 DMA，请完成以下步骤：

1. 提供软件复位，以复位所有 MAC 内部寄存器和逻辑（*DMA 模式寄存器 (ETH_DMAMR)* 的位 0）。
2. 等待完成复位过程（轮询 *DMA 模式寄存器 (ETH_DMAMR)* 的位 0，当复位操作完成时该位会清零）。
3. 编程以下位字段以初始化*系统总线模式寄存器 (ETH_DMASBMR)*：
   a) AAL
   b) 固定突发或未定义突发
   c) 未完成请求限制值 (OSR_LMT)。
   d) 如果使能固定长度值，请选择 AXI 总线上可能的最大突发长度（位 [7:1]）

4. 创建发送和接收描述符列表。此外，还应确保接收描述符为 DMA 所有（将 TDES3/RDES3 描述符的位 31 置 1）。有关描述符的更多信息，请参见*第 57.10 节：描述符*。

注： *从环起始到环结束的描述符地址不应超过 4GB 边界。*

5. 编程 ETH_DMAC0TXRLR 和 ETH_DMAC0RXRLR 寄存器（请参见*通道 x Tx 描述符环长度寄存器 (ETH_DMACxTXRLR)* 和*通道 0 Rx 描述符环长度寄存器 (ETH_DMAC0RXRLR)*）。编程的环长度必须至少为 4。
6. 使用发送和接收描述符的基址初始化接收和发送描述符列表地址（*通道 x Tx 描述符列表地址寄存器 (ETH_DMACxTXDLAR)*，*通道 0 Rx 描述符列表地址寄存器 (ETH_DMAC0RXDLAR)*）。此外，还应编程发送和接收尾指针寄存器，这类寄存器会将可用描述符通知给 DMA（请参见*通道 x Tx 描述符尾指针寄存器 (ETH_DMACxTXDTPR)* 和*通道 0 Rx 描述符尾指针寄存器 (ETH_DMAC0RXDTPR)*）。
7. 编程 ETH_DMAC0CR、ETH_DMAC0TXCR 和 ETH_DMAC0RXCR 寄存器（请参见*通道 x 控制寄存器 (ETH_DMACxCR)*、*通道 x 发送控制寄存器 (ETH_DMACxTXCR)* 和*通道 0 接收控制寄存器 (ETH_DMAC0RXCR)*）以配置以下参数：DMA 触发的最大突发长度 (PBL)、描述符跳过长度、用于 TxDMA 的 OSP 以及用于 RxDMA 的 RBSZ[13:0] 等等。
8. 通过编程 ETH_DMAC0IER 寄存器（请参见*通道 x 中断使能寄存器 (ETH_DMACxIER)*）使能中断。
9. 通过将*通道 0 接收控制寄存器 (ETH_DMAC0RXCR)* 的 SR（位 0）置 1 以及将 ETH_DMAC0TXCR（请参见*通道 x 发送控制寄存器 (ETH_DMACxTXCR)*）的 ST（位 0）置 1 来启动接收和发送 DMA。
10. 对硬件中选择的所有 Tx DMA 通道重复步骤 4 至 9。

57.9.2 MTL 初始化

要初始化 MTL 寄存器，请完成以下步骤：

1. 当使用多个 Tx 和 Rx 队列时，请在*工作模式寄存器 (ETH_MTLOMR)* 中对 Tx 调度 (SCHALG) 和接收仲裁算法 (RAA) 位字段进行编程，以初始化 MTL 操作。
2. 在 *Tx 队列 x 工作模式寄存器 (ETH_MTLTXQxOMR)* 中编程以下位字段，以初始化工作模式。
   a) 发送存储转发 (TSF) 或发送阈值控制 (TTC)（使用阈值模式时）。
   b) 将发送队列使能 (TXQEN) 编程为值 2‘b10，以使能发送队列 0。
   c) 发送队列大小 (TQS)。
3. 在 ETH_MTLRXQxOMR 寄存器（请参见 *Rx 队列 x 工作模式寄存器 (ETH_MTLRXQxOMR)*）中编程以下位字段，以初始化工作模式：
   a) 接收存储转发 (RSF) 或 RTC（使用阈值模式时）。
   b) 流控制激活和取消激活阈值，用于 MTL 接收 FIFO（RFA 和 RFD）。
   c) 错误数据包和过小的良好数据包转发使能（FEP 和 FUP）。
   d) 接收队列大小 (RQS)。
4. 对配置中选择的所有 MTL Tx 和 Rx 队列重复前两个步骤。

57.9.3 MAC 初始化

MAC 配置寄存器建立 MAC 的工作模式。如果可能，必须在初始化 DMA 之前对这些寄存器进行初始化。在完成 DMA 初始化之后，还可执行以下 MAC 初始化操作。如果在配置 DMA 之前已完成 MAC 初始化，则只有在 DMA 激活后才可使能 MAC 接收器（以下序列中的最后一步）。否则，接收到的帧将填充 Rx FIFO 并引发上溢。

1. 提供 MAC 地址寄存器：*MAC 地址 x 低位寄存器 (ETH_MACAxLR)*、*MAC 地址 0 高位寄存器 (ETH_MACA0HR)* 和 *MAC 地址 x 高位寄存器 (ETH_MACAxHR)*。
2. 在*数据包过滤控制寄存器 (ETH_MACPFR)* 中编程以下位字段，从而为传入的帧设置相应过滤器：
   a) 接收所有。
   b) 混合模式。
   c) 散列或完美过滤器。
   d) 单播、多播、广播和控制帧过滤设置。
3. 在 *Tx 队列 0 流控制寄存器 (ETH_MACQ0TXFCR)* 中编程以下位字段以实现适当的流控制：
   a) 暂停时间和其他暂停帧控制位。
   b) 发送流控制位。
   c) 流控制忙碌。
4. 根据需要编程*中断使能寄存器 (ETH_MACIER)*，前提是适用于您的配置。
5. 在*工作模式配置寄存器 (ETH_MACCR)* 寄存器中编程相应位字段。例如，发送和 jabber 禁止时的数据包间隙。
6. 将*工作模式配置寄存器 (ETH_MACCR)* 寄存器中的位 0 和位 1 置 1，以启动 MAC 发送器和接收器。

如需支持巨型发送/接收数据包，请遵循如下步骤：

- 在*工作模式配置寄存器 (ETH_MACCR)* 中
  a) 将 JE 位置 1。
  b) 将 JD 和 WD 位置 0 以避免发送大型数据包错误报告。
  c) 将 GPSLCE 位置 1
  d) 将*扩展工作模式配置寄存器 (ETH_MACECR)* 的 GPSL 位字段设置为大于 9026 的值

为了支持发送/接收数据包（最大为 16K），请按照以下步骤操作：

- 在*工作模式配置寄存器 (ETH_MACCR)* 中
  a) 将 JD 和 WD 位置 1 以避免发送大型数据包错误报告。
  b) 将 GPSLCE 位置 1。
  c) 将*扩展工作模式配置寄存器 (ETH_MACECR)* 的 GPSL 位字段设置为 16383。

57.9.4 执行正常接收和发送操作

为实现正常操作，请完成以下步骤：

1. 对于正常发送和接收中断，请读取中断状态。然后，通过读取主机所拥有的描述符的状态轮询描述符（发送或接收）。
2. 将描述符设为相应值。确保发送和接收描述符为 DMA 所有，以恢复数据的发送和接收。
3. 如果描述符并非为 DMA 所有（或无可用描述符），则 DMA 会进入挂起状态。可释放描述符并对 ETH_DMAC0TXDTPR（请参见*通道 x Tx 描述符尾指针寄存器 (ETH_DMACxTXDTPR)*）和 ETH_DMAC0RXDTPR（请参见*通道 0 Rx 描述符尾指针寄存器 (ETH_DMAC0RXDTPR)*）执行写入操作，以恢复发送或接收。
4. 在调试模式下，当前主机发送器或接收器描述符地址指针的值可在 ETH_DMAC0CATXDR 和 ETH_DMAC0CARXDR 寄存器（请参见*通道 x 当前应用程序发送描述符寄存器 (ETH_DMACxCATXDR)* 和*通道 0 当前应用程序接收描述符寄存器 (ETH_DMAC0CARXDR)*）中进行读取。
5. 在调试模式下，当前主机发送缓冲区地址指针和接收缓冲区地址指针的值可在 ETH_DMAC0CATXDR 和 ETH_DMAC0CARXDR 寄存器（请参见*通道 x 当前应用程序发送描述符寄存器 (ETH_DMACxCATXDR)* 和*通道 0 当前应用程序接收描述符寄存器 (ETH_DMAC0CARXDR)*）中进行读取。

57.9.5 停止和开始发送

要将发送暂停一段时间，请完成以下步骤：

1. 通过将 ETH_DMAC0TXCR 寄存器（请参见*通道 x 发送控制寄存器 (ETH_DMACxTXCR)*）的位 0 (ST) 清零来禁止发送 DMA（如适用）。
2. 等待完成之前的所有帧发送过程。可通过读取 *Tx 队列 x 调试寄存器 (ETH_MTLTXQxDR)* 的相应位（TRCSTS[1:0] 不为 01，TXQSTS = 0）对此进行检查。
3. 通过将*工作模式配置寄存器 (ETH_MACCR)* 寄存器的 RE 位和 TE 位清零来禁止 MAC 发送器和 MAC 接收器。
4. 确保 Rx FIFO 中的数据被传送到系统内存（通过读取 *Tx 队列 x 调试寄存器 (ETH_MTLTXQxDR)* 的相应位，PRXQ= 0 且 RXQSTS[1:0] = 00）后，禁止接收 DMA（如适用）。
5. 确保 Tx 队列和 Rx 队列均为空（*Tx 队列 x 调试寄存器 (ETH_MTLTXQxDR)* 中的 TXQSTS 为 0 且 RXQSTS[1:0] 被设为 00）。
6. 如要重新启动操作，请首先启动 DMA，然后再使能 MAC 发送器和接收器。

注： *当 MAC 正在主动执行发送或接收时，请勿更改配置（如双工模式、速度、端口或回环）。仅当 MAC 发送器和接收器未激活时，这些参数才由软件进行更改。*

同样，当发送和接收 DMA 激活时，请勿更改 DMA 相关配置。

57.9.6 关于在 RxDMA 中切换至新描述符列表的编程指南

在 Rx DMA 中切换至新的描述符列表与在 Tx DMA 中相比有所不同。当 RxDMA 处于挂起状态时，系统将允许切换至新的描述符列表，具体如下所述：

- 通常，RxDMA 会提前准备描述符。
- 如果由于描述符不可用而导致 RxDMA 进入挂起状态，则会发生重大故障（软件无法清空已填充的描述符/缓冲区）。如果不立即纠正这个问题，则系统会由于 RxFIFO 上溢而丢帧。因此，系统将允许软件创建新的描述符列表并立即开始使用该列表对 RxDMA 进行编程，而无需进入停止状态。

57.9.7 关于切换 AXI 时钟频率的编程指南

如需动态更改 AXI 时钟频率（无需应用软复位或硬复位），请按照以下步骤操作：

1. 禁止发送 DMA（如适用），并等待所有之前的帧发送完成。当帧发送完成时，Tx FIFO 会变为空，且 Tx DMA 会进入停止状态。Tx FIFO 的状态将在 *Tx 队列 x 调试寄存器 (ETH_MTLTXQxDR)* 中给出，DMA 的状态将在 *调试状态寄存器 (ETH_DMADSR)* 中给出。
2. 通过将 *工作模式配置寄存器 (ETH_MACCR)* 中的相应位清零来禁止 MAC 发送器和 MAC 接收器。
3. 确保 Rx FIFO 中的数据被传送到系统内存后，禁止接收 DMA（如适用）。*Rx 队列 x 调试寄存器 (ETH_MTLRXQxDR)* 中将给出 Rx FIFO 空状态。
4. 确保应用程序不会执行任何寄存器读写操作。
5. 更改 AXI 时钟的频率。
6. 使能 MAC 发送器或 MAC 接收器以及发送或接收 DMA。这些步骤可确保在切换时钟频率时，Tx FIFO 或 Rx FIFO 中没有有效数据，并可防止一切数据损坏。

57.9.8 关于多通道多队列操作的编程指南

发送路径

1. 在 ETH_MTLTXQ**x**OMR 寄存器的 TQS 位字段中，对发送队列大小进行编程。队列的大小由在 TQS 位字段中编程的值决定。

   在执行发送操作期间，通道的数量将等于队列的数量。因此，通道至队列映射将为固定映射。
2. 如需使用队列，必须通过将相应 ETH_MTLTXQ**x**OMR 寄存器中的 TXQEN 位置 1 来使能队列。必须使能 *通道 x 发送控制寄存器 (ETH_DMACxTXCR)* 中的 ST 位以及 ETH_MTLTXQ**x**OMR 寄存器中相应的 TXQEN 位。
3. 在 ETH_MTLOMR 寄存器的 SCHALG 位中对调度方法进行编程。
4. 如果在 AV 队列 1 上使能了 CBS 算法，则需要相应地对 ETH_MTLTXQ1ECR、ETH_MTLTXQ1SSCR、ETH_MTLTXQ1HCR 和 ETH_MTLTXQ1LCR 寄存器进行编程。

接收路径

1. 在 ETH_MTLRXQ**x**OMR 寄存器的 RQS 位字段中，对接收队列大小进行编程。队列的大小由在 RQS 位字段中编程的值决定。
2. 为 AV 在 ETH_MACRXQ0CR 寄存器中的 RXQ0EN 到 RXQ1EN 位字段使能接收队列 0 到 1。

   使能 *通道 0 接收控制寄存器 (ETH_DMAC0RXCR)* 中静态或动态映射的 SR 位以及 *Rx 队列控制 0 寄存器 (ETH_MACRXQC0R)* 中相应的 RXQiEN 位。
3. MAC 将根据以下数据包类型将 Rx 数据包传送到 Rx 队列：
   a) AV PTP 数据包将根据 ETH_MACRXQC1R 寄存器中的 PTPQ 值进行传送。
   b) 带 VLAN 标记的数据包中的 VLAN 标记优先级位字段

      对 ETH_MACRXQC2R 寄存器的 PSRQ1 和 PSRQ0 进行编程，以便依据已接收数据包的 USP（用户优先级）位字段将带标记的数据包配置为传送到 Rx 队列 0 到 1。
   c) AV 带标记的控制和数据包也将根据 ETH_MACRXQC2R 寄存器的 PSRQ 位字段进行传送。

注： *在 PSRQ0 和 PSRQ1 中设置的优先级必须唯一。*

关于 DMA 通道故障恢复的编程指南

当 DMA 通道出现总线错误时，请按照下述步骤来进行故障恢复。

从接收 DMA 通道故障中恢复

如果接收 DMA 通道中发生总线错误，请按照以下步骤操作：

1. 将 RPF 位置 1。此举将逐个刷新所有数据包。该步骤可选。然而，将此位置 1 可以防止当发送到 RXDMA 的数据包因总线错误而停止时，Rx 队列中出现 HOL（队头）阻塞。
2. 对 DMA 通道的特定寄存器进行重新编程。
3. 启动 DMA 通道。

从发送 DMA 通道故障中恢复

1. 停止特定的 DMA 通道，即使该通道已处于激活状态。
2. 刷新相应的 MTL 队列。
3. 对 DMA 通道的特定寄存器进行重新编程。
4. 启动 DMA 通道。

注： *由于已知限制的缘故，对 DMA 通道寄存器进行重新编程也许并不一定能够成功从总线错误中恢复。如果在重新编程 DMA 后外设并未进入完全正常工作的状态，则可以尝试一种变通方法，即通过发出软复位从总线错误中恢复。*

关于禁止接收队列的编程指南

请按照以下步骤禁止特定接收队列：

1. 将*工作模式配置寄存器 (ETH_MACCR)* 的 RE 位置 0，以禁止接收器。
2. 读取所有 Rx 队列的*调试寄存器 (ETH_MACDR)*（确保 RPESTS 和 RFCFCSTS 位字段为 0）和 *Rx 队列 x 调试寄存器 (ETH_MTLRXQxDR)*。
3. 当所有 *Rx 队列 x 调试寄存器 (ETH_MTLRXQxDR)* 都读作 0 时，请通过对 *Rx 队列控制 0 寄存器 (ETH_MACRXQC0R)* 中相应的位字段进行编程来禁止预期使用的 Rx 队列。
4. 将*工作模式配置寄存器 (ETH_MACCR)* 的 RE 位置 1，以使能接收器。

57.9.9 关于 GMII 链路状态转换的编程指南

链路断开时发送时钟和接收时钟处于运行状态

如果在发送和接收时钟处于运行状态时链路断开，请完成以下步骤：

1. 通过将*通道 x 控制寄存器 (ETH_DMACxCR)* 的位 0 (ST) 清零来禁止发送 DMA（如适用）。
2. 通过将*工作模式配置寄存器 (ETH_MACCR)* 的 RE 位清零来禁止 MAC 接收器。
3. 等待完成之前的所有帧发送过程。可通过读取 *Tx 队列 x 调试寄存器 (ETH_MTLTXQxDR)* 的相应位（TRCSTS[1:0] 不为 01）对此进行检查。

   或

   刷新 Tx FIFO 以加快清空操作。
4. 通过将*工作模式配置寄存器 (ETH_MACCR)* 寄存器的 TE 位清零来禁止 MAC 发送器。
5. 确保 Tx 和 Rx 队列均为空（*Tx 队列 x 调试寄存器 (ETH_MTLTXQxDR)* 中的 TXQSTS 被设为 0，且 *Rx 队列 x 调试寄存器 (ETH_MTLRXQxDR)* 中的 RXQSTS[1:0] 被设为 00）。
6. 链路接通后，读取 PHY 寄存器以识别最新配置，并相应地对 MAC 寄存器进行编程。
7. 如要重新启动相应操作，请首先启动 Tx DMA，然后再使能 MAC 发送器和接收器。

   Rx DMA 无需使能：因为接收器已被禁止，Rx FIFO 中没有数据。

链路断开时发送时钟和接收时钟处于停止状态

如果链路断开且发送时钟和接收时钟处于停止状态，请完成以下步骤：

1. 通过将*工作模式配置寄存器 (ETH_MACCR)* 中的 RE 位和 TE 位清零来禁止 MAC 发送器和接收器。由于不存在时钟，因此上述操作不会立即生效。
2. 等待链路接通以及时钟恢复。
3. 如果在发送/接收时钟停止时正在进行帧传送，请等待至所有部分帧全部完成传送。可通过读取 *调试寄存器 (ETH_MACDR)* 进行检查（所有位均应设为 0）。MAC 发送器停止时，一些旧数据包可能仍保留在 TXFIFO 中。
4. 读取 PHY 寄存器以识别最新工作模式，并相应地对 MAC 寄存器进行编程。
5. 通过将 RE 位和 TE 位置 1 来重启 MAC 发送器和接收器。

57.9.10 关于 IEEE 1588 时间戳的编程指南

初始化系统时间生成

可通过将*时间戳控制寄存器 (ETH_MACTSCR)* 的位 0 置 1 来使能时间戳功能。不过，必须在该位置 1 后初始化时间戳计数器。要执行外设初始化，请完成以下步骤：

1. 通过将 *中断使能寄存器 (ETH_MACIER)* 的位 12 清零来屏蔽时间戳触发中断。
2. 将*时间戳控制寄存器 (ETH_MACTSCR)* 的位 0 置 1 以使能时间戳。
3. 根据 PTP 时钟频率编程*亚秒增量寄存器 (ETH_MACSSIR)*。
4. 如果使用精密校准方法，请编程*时间戳加数寄存器 (ETH_MACTSAR)*，并将*时间戳控制寄存器 (ETH_MACTSCR)* 的位 5 置 1。
5. 轮询*时间戳控制寄存器 (ETH_MACTSCR)*，直到位 5 清零。
6. 编程*时间戳控制寄存器 (ETH_MACTSCR)* 的位 1 以选择精密更新方法（如果需要）。
7. 使用适当的时间值编程*系统时间秒更新寄存器 (ETH_MACSTSUR)* 和*系统时间纳秒更新寄存器 (ETH_MACSTNUR)*。
8. 将*时间戳控制寄存器 (ETH_MACTSCR)* 中的位 2 置 1。

   用时间戳更新寄存器中写入的值初始化时间戳计数器后，时间戳计数器便开始运行。如果需要单步时间戳：

   a) 通过编程 TDES3 上下文描述符的位 27 来使能单步时间戳。

   b) 编程*时间戳入站不对称校准寄存器 (ETH_MACTSIACR)* 以更新 PDelay_Req PTP 消息中的校准位字段。
9. 使能 MAC 接收器和发送器以使时间戳功能正常运行。

注： *如果通过将时间戳控制寄存器 (ETH_MACTSCR) 的位 0 清零来禁止时间戳操作，请重复所有上述步骤以重新启动时间戳操作。*

系统时间校准

要一次性（粗略校准方法）同步或更新系统时间，请完成以下步骤：

1. 在时间戳更新寄存器（*系统时间秒更新寄存器 (ETH_MACSTSUR)* 和 *系统时间纳秒更新寄存器 (ETH_MACSTNUR)*）中设置偏移（正偏移或负偏移）。
2. 将 *时间戳控制寄存器 (ETH_MACTSCR)* 的位 3 (TSUPDT) 置 1。

   当 TSUPDT 位清零时，时间戳更新寄存器中的值将加到系统时间中或者从系统时间中减去。

要同步或更新系统时间以减少系统时间抖动（精密校准方法），请完成以下步骤：

1. 借助 *系统时间寄存器模块* 一节中介绍的算法，计算预期的系统时间递增或递减速率。
2. 使用新值更新 *时间戳加数寄存器 (ETH_MACTSAR)*，并将 *时间戳控制寄存器 (ETH_MACTSCR)* 寄存器的位 5 置 1。
3. 等待加数寄存器中的新值生效。这可以通过在系统时间达到目标值后，使能时间戳触发中断来实现。
4. 在 *PPS 目标时间秒寄存器 (ETH_MACPPSTTSR)* 和 *PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)* 中编程所需目标时间。
5. 在 *中断使能寄存器 (ETH_MACIER)* 的位 12 中使能时间戳中断。
6. 将寄存器 *时间戳控制寄存器 (ETH_MACTSCR)* 中的位 4 置 1。
7. 当此触发生成中断时，请对 *中断状态寄存器 (ETH_MACISR)* 进行读取操作。
8. 使用旧值重新编程 *时间戳加数寄存器 (ETH_MACTSAR)*，并再次将位 5 置 1。

57.9.11 PTP 减荷功能的编程指南

关于使能自动定期生成 PTP 同步消息的编程指南

请按照以下步骤使能自动定期生成 PTP 同步消息：

1. 将 *时间戳控制寄存器 (ETH_MACTSCR)* 的 SNAPTYPSEL、TSMSTRENA 和 TSEVNTENA 位字段分别编程为 0、1 和 1，以将节点配置为普通或边界主模式（透明主模式为 1、1 和 1）。
2. 对 *PTP 减荷控制寄存器 (ETH_MACPOCR)* 的 PTOEN 位和 DN 位字段进行编程，以使能 PTP 减荷功能以及需在出站 PTP 同步消息中发送的域号。
3. 对 *PTP 减荷控制寄存器 (ETH_MACPOCR)* 的 ASYNCEN 位进行编程，以使能定期生成 PTP 同步消息。
4. 对 *PTP 源端口标识 0 寄存器 (ETH_MACSPI0R)*、*PTP 源端口标识 1 寄存器 (ETH_MACSPI1R)* 和 *PTP 源端口标识 2 寄存器 (ETH_MACSPI2R)* 中的 80 位源端口标识进行编程，以便在出站 PTP 同步消息中进行发送。
5. 对 *日志消息间隔寄存器 (ETH_MACLMIR)* 的 LSI 位字段进行编程，以设置 PTP 同步消息的周期。

   例如，值 1 对应于 2^1，即每 2 秒生成一个 PTP 同步消息，而值 0xFF（-1 的二进制补码）对应于 2^{-1}，即每 0.536 秒生成一个 PTP 同步消息。
6. 对 *中断使能寄存器 (ETH_MACIER)* 的 TSIE 位进行编程，以使能生成时间戳中断。
7. 等待通过将 *时间戳状态寄存器 (ETH_MACTSSR)* 中的 TXTSSIS 位置 1 生成的 sbd_intr_o 中断生效。这表明 *Tx 时间戳状态秒寄存器 (ETH_MACTXTSSSR)* 和 *Tx 时间戳状态纳秒寄存器 (ETH_MACTXTSSNR)* 中已捕获到 PTP 同步消息的时间戳。

关于使能定期生成 PTP Pdelay_Req 消息的编程指南

请按照以下步骤使能自动定期生成 PTP Pdelay_Req 消息

1. 将*时间戳控制寄存器 (ETH_MACTSCR)* 的 SNAPTYPSEL、TSMSTRENA 和 TSEVNTENA 位字段分别编程为 1、0 和 1，以将节点配置为透明从模式（透明主模式为 1、1 和 1，点对点透明模式为 3、X 和 X）。
2. 对 *PTP 减荷控制寄存器 (ETH_MACPOCR)* 的 PTOEN 位和 DN 位字段进行编程，以使能 PTP 减荷功能以及需在出站 PTP Pdelay_Req 消息中发送的域号。
3. 对 *PTP 减荷控制寄存器 (ETH_MACPOCR)* 的 APDREQEN 位进行编程，以使能定期生成 PTP Pdelay_Req 消息。
4. 对 *PTP 源端口标识 0 寄存器 (ETH_MACSPI0R)*、*PTP 源端口标识 1 寄存器 (ETH_MACSPI1R)* 和 *PTP 源端口标识 2 寄存器 (ETH_MACSPI2R)* 中的 80 位源端口标识进行编程，以便在出站 PTP Pdelay_Req 消息中进行发送。
5. 对*日志消息间隔寄存器 (ETH_MACLMIR)* 的 LMPDRI 位字段进行编程，以设置 PTP Pdelay_Req 消息的周期。

   例如，值 1 对应于 2^1，即每 2 秒生成一个 PTP Pdelay_Req 消息，而值 0xFF（-1 的二进制补码）对应于 2^{-1}，即每 0.536 秒生成一个 PTP Pdelay_Req 消息。
6. 对*中断使能寄存器 (ETH_MACIER)* 的 TSIE 位进行编程，以使能生成时间戳中断。
7. 等待通过将*时间戳状态寄存器 (ETH_MACTSSR)* 中的 TXTSSIS 位置 1 生成的 sbd_intr_o 中断生效。这表明 *Tx 时间戳状态秒寄存器 (ETH_MACTXTSSSR)* 和 *Tx 时间戳状态纳秒寄存器 (ETH_MACTXTSSNR)* 中已捕获到 PTP 同步消息的时间戳。

关于使能为普通或边界主模式生成 PTP 响应消息的编程指南

请按照以下步骤使能为普通或边界主模式生成 PTP 响应消息（定期生成 PTP 同步消息，并生成 PTP Delay_Req 消息以响应 PTP Delay_Resp 消息）：

1. 将*时间戳控制寄存器 (ETH_MACTSCR)* 的 SNAPTYPSEL、TSMSTRENA 和 TSEVNTENA 位字段分别编程为 0、1 和 1。
2. 对 *PTP 减荷控制寄存器 (ETH_MACPOCR)* 的 PTOEN 位和 DN 位字段进行编程，以使能 PTP 减荷功能以及需与入站 PTP Delay_Req 消息匹配，并在出站 PTP Delay_Resp 消息中发送的域号。
3. 对 *PTP 源端口标识 0 寄存器 (ETH_MACSPI0R)*、*PTP 源端口标识 1 寄存器 (ETH_MACSPI1R)* 和 *PTP 源端口标识 2 寄存器 (ETH_MACSPI2R)* 中的 80 位源端口标识进行编程，以便与入站 PTP Delay_Req 消息匹配，并在出站 PTP Delay_Resp 消息中发送。
4. 对*日志消息间隔寄存器 (ETH_MACLMIR)* 中的 DRSYNCR 和 LSI 位字段进行编程。这两个位字段之和将在 PTP Delay_Resp 消息的 logMinDelayReqInterval 位字段中更新。

关于使能为普通或边界从模式生成 PTP 响应消息的编程指南

请按照以下步骤使能为普通或边界从模式生成 PTP 响应消息（生成 PTP Delay_Req 消息以响应 PTP 同步消息）：

1. 将*时间戳控制寄存器 (ETH_MACTSCR)* 的 SNAPTYPSEL、TSMSTRENA 和 TSEVNTENA 位字段分别编程为 0、0 和 1。
2. 对 *PTP 减荷控制寄存器 (ETH_MACPOCR)* 的 PTOEN 位和 DN 位字段进行编程，以使能 PTP 减荷功能以及需与入站 PTP 同步消息匹配，并在出站 PTP Delay_Req 消息中发送的域号。
3. 对 *PTP 源端口标识 0 寄存器 (ETH_MACSPI0R)*、*PTP 源端口标识 1 寄存器 (ETH_MACSPI1R)* 和 *PTP 源端口标识 2 寄存器 (ETH_MACSPI2R)* 中的 80 位源端口标识进行编程，以便与入站 PTP 同步消息匹配，并在出站 PTP Delay_Req 消息中发送。
4. 对*日志消息间隔寄存器 (ETH_MACLMIR)* 中的 DRSYNCR 位字段进行编程，以指示生成一个 PTP Delay_Req 消息以响应已接收 PTP 同步消息的数量。

关于使能为透明从模式生成 PTP 响应消息的编程指南

请按照以下步骤使能为透明从模式生成 PTP 响应消息（生成 PTP Delay_Req 消息以响应 PTP 同步消息，生成 PTP Pdelay_Resp 消息以响应 PTP Pdelay_Req 消息，并定期生成 PTP Pdelay_Req 消息）

1. 将*时间戳控制寄存器 (ETH_MACTSCR)* 的 SNAPTYPSEL、TSMSTRENA 和 TSEVNTENA 位字段分别编程为 1、0 和 1。
2. 对 *PTP 减荷控制寄存器 (ETH_MACPOCR)* 的 PTOEN 位和 DN 位字段进行编程，以使能 PTP 减荷功能以及需与入站 PTP 同步或 Pdelay_Req 消息匹配，并在出站 PTP Delay_Req、Pdelay_Resp 或 Pdelay_Req 消息中发送的域号。
3. 对 *PTP 源端口标识 0 寄存器 (ETH_MACSPI0R)*、*PTP 源端口标识 1 寄存器 (ETH_MACSPI1R)* 和 *PTP 源端口标识 2 寄存器 (ETH_MACSPI2R)* 中的 80 位源端口标识进行编程，以便与入站 PTP 同步或 Pdelay_Req 消息匹配，并在出站 PTP Delay_Req、Pdelay_Resp 或 Pdelay_Req 消息中发送。
4. 对*日志消息间隔寄存器 (ETH_MACLMIR)* 中的 DRSYNCR 和 LMPDRI 位字段进行编程，以指示生成一个 PTP Delay_Req 消息以响应已接收 PTP 同步消息的数量，以及 PTP Pdelay_Req 消息的周期。
5. 对*中断使能寄存器 (ETH_MACIER)* 的 TSIE 位进行编程，以使能生成时间戳中断。
6. 等待通过将*时间戳状态寄存器 (ETH_MACTSSR)* 中的 TXTSSIS 位置 1 生成的 sbd_intr_o 中断生效。这表明 *Tx 时间戳状态秒寄存器 (ETH_MACTXTSSSR)* 和 *Tx 时间戳状态纳秒寄存器 (ETH_MACTXTSSNR)* 中已捕获到 PTP 同步消息的时间戳，以供出站 PTP Pdelay_Req 和 Pdelay_Resp 消息使用。

关于使能为透明主模式生成 PTP 响应消息的编程指南

请按照以下步骤使能为透明主模式生成 PTP 响应消息（生成 PTP Delay_Resp 消息以响应 PTP Delay_Req 消息，生成 PTP Pdelay_Resp 消息以响应 PTP Pdelay_Req 消息，并定期生成 PTP Pdelay_Req 或同步消息）：

1. 将*时间戳控制寄存器 (ETH_MACTSCR)* 的 SNAPTYPSEL、TSMSTRENA 和 TSEVNTENA 位字段分别编程为 1、1 和 1。
2. 对 *PTP 减荷控制寄存器 (ETH_MACPOCR)* 的 PTOEN 位和 DN 位字段进行编程，以使能 PTP 减荷功能以及需与入站 PTP Delay_Req 或 Pdelay_Req 消息匹配，并在出站 PTP Delay_Resp、Pdelay_Resp、Pdelay_Req 或同步消息中发送的域号。
3. 对 *PTP 源端口标识 0 寄存器 (ETH_MACSPI0R)*、*PTP 源端口标识 1 寄存器 (ETH_MACSPI1R)* 和 *PTP 源端口标识 2 寄存器 (ETH_MACSPI2R)* 中的 80 位源端口标识进行编程，以便与入站 PTP Delay_Req 或 Pdelay_Req 消息匹配，并在出站 PTP Delay_Resp、Pdelay_Resp、Pdelay_Req 或同步消息中发送。
4. 对*日志消息间隔寄存器 (ETH_MACLMIR)* 中的 DRSYNCR、LSI 和 LMPDRI 位字段进行编程，DRSYNCR 与 LSI 之和将在 PTP Delay_Resp 消息的 logMinDelayReqInterval 位字段中更新，并指示 PTP 同步或 Pdelay_Req 消息的周期。
5. 对*中断使能寄存器 (ETH_MACIER)* 的 TSIE 位进行编程，以使能生成时间戳中断。
6. 等待通过将*时间戳状态寄存器 (ETH_MACTSSR)* 中的 TXTSSIS 位置 1 生成的 sbd_intr_o 中断生效。这表明 *Tx 时间戳状态秒寄存器 (ETH_MACTXTSSSR)* 和 *Tx 时间戳状态纳秒寄存器 (ETH_MACTXTSSNR)* 中已捕获到 PTP 同步消息的时间戳，以供出站 PTP 同步、Pdelay_Req 和 Pdelay_Resp 消息使用。

关于使能为点对点透明模式生成 PTP 响应消息的编程指南

请按照以下步骤使能为点对点透明模式生成 PTP 响应消息（生成 PTP Pdelay_Resp 消息以响应 PTP Pdelay_Req 消息，并定期生成 PTP Pdelay_Req 消息）：

1. 将*时间戳控制寄存器 (ETH_MACTSCR)* 的 SNAPTYPSEL、TSMSTRENA 和 TSEVNTENA 位字段分别编程为 3、X 和 X。
2. 对 *PTP 减荷控制寄存器 (ETH_MACPOCR)* 的 PTOEN 位和 DN 位字段进行编程，以使能 PTP 减荷功能以及需与入站 PTP Pdelay_Req 消息匹配，并在出站 PTP Pdelay_Resp 消息中发送的域号。
3. 对 *PTP 源端口标识 0 寄存器 (ETH_MACSPI0R)*、*PTP 源端口标识 1 寄存器 (ETH_MACSPI1R)* 和 *PTP 源端口标识 2 寄存器 (ETH_MACSPI2R)* 中的 80 位源端口标识进行编程，以便与入站 PTP Pdelay_Req 消息匹配，并在出站 PTP Pdelay_Resp 消息中发送。
4. 对*日志消息间隔寄存器 (ETH_MACLMIR)* 中的 LMPDRI 位字段进行编程，以指示 PTP Pdelay_Req 消息的周期
5. 对*中断使能寄存器 (ETH_MACIER)* 的 TSIE 位进行编程，以使能生成时间戳中断
6. 等待通过将*时间戳状态寄存器 (ETH_MACTSSR)* 中的 TXTSSIS 位置 1 生成的 sbd_intr_o 中断生效。这表明 *Tx 时间戳状态秒寄存器 (ETH_MACTXTSSSR)* 和 *Tx 时间戳状态纳秒寄存器 (ETH_MACTXTSSNR)* 中已捕获到 PTP 同步消息的时间戳，以供出站 PTP Pdelay_Req 和 Pdelay_Resp 消息使用。

57.9.12 关于节能型以太网 (EEE) 的编程指南

进入和退出 Tx LPI 模式

EEE 可使 IEEE 802.3 介质访问控制 (MAC) 子层和一系列物理层在低功耗空闲 (LPI) 模式下运行。在发送路径中，软件必须将 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 的 LPIEN 位置 1，以指示 MAC 停止发送并启用 LPI 协议。

要执行 MAC 初始化，请完成以下步骤：

1. 通过 MDIO 接口对 PHY 寄存器进行读取操作，并检查远程端是否具有 EEE 功能。然后协商定时器值。
2. 通过 MDIO 接口编程 PHY 寄存器（包括 RX_CLK_stoppable 位，该位用于向 PHY 指示是否在 LPI 模式下停止 Rx 时钟）。
3. 对 *LPI 定时器控制寄存器 (ETH_MACLTCR)* 中的位 25 到 16 以及位 15 到 0 进行编程。
4. 使用 MDIO 接口读取 PHY 链路状态，并更新 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 的位 17。

   相应地更新 *LPI 控制和状态寄存器 (ETH_MACLCSR)*。PHY 芯片的链路状态发生变化时，应进行此更新。
5. 根据用于访问 CSR 从端口的时钟频率，编程*一微秒节拍计数器寄存器 (ETH_MAC1USTCR)*。
6. 使用 MAC 在自行进入 LPI 状态之前应等待的 IDLE 时间编程 *LPI 进入定时器寄存器 (ETH_MACLETR)* 中的 LPIET 位。
7. 将 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 的 LPITE 和 LPITXA（位 20 到 19）置 1 以使能 LPI 自动进入和 MAC 自动退出 LPI 状态。
8. 将 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 的位 16 置 1 以将 MAC 发送器置于 LPI 状态。

   所有计划的数据包均完成后，MAC 会进入 LPI 状态。它将在 LPIET 位所指示的时间内始终保持空闲。进入 LPI 状态后，它会将 TLPIEN（位 0）置 1。
9. 计划好数据包发送后（TxDMA 退出空闲状态，或者 ATI 或 MTI 接口上出现数据包时），MAC 发送器自动退出 LPI 状态。MAC 发送器在 TLPIEX 中断状态位置 1 之前会等待 TWT 时间，之后才会恢复数据包发送。
10. 如果 MAC 发送器在 LPIET 时间内保持空闲，则会再次进入 LPI 状态。然后，它会将 TLPIEN 位置 1，继续执行进入-退出循环。
11. 如果应用程序需要覆盖自动进入/退出模式，并使 MAC 发送器直接退出 LPI 状态，请复位 LPIEN 位。

注： *为确保 MAC 仅会在 Tx FIFO 中所有排队帧发送完成后方才进入 LPI 状态，请将 LPI 控制和状态寄存器 (ETH_MACLCSR) 中的 LPITXA 位置 1。*

要在 LPI 状态下关闭 GMII 发送时钟，请使用 sbd_tx_clk_gating_ctrl_o 信号对时钟输入进行门控。

要在 LPI 状态下关闭 CSR 时钟或切断系统其余部分的电源，请等待系统生成 LPI 控制和状态寄存器 (ETH_MACLCSR) 的 TLPIEN 中断。当您想要退出 LPI 状态时，请在执行步骤 6 之前将时钟恢复。

在 LPI 模式下门控关闭 CSR 时钟

当 MAC 处于低功耗空闲 (LPI) 模式时，用户可以门控关闭 CSR 时钟以节省电能。

在 Rx LPI 模式下门控关闭 CSR 时钟

当 MAC 从 PHY 接收到 LPI 模式时，系统将执行以下操作：

1. MAC RX 将进入 LPI 模式，且系统将设置 Rx LPI 进入中断状态（*LPI 控制和状态寄存器 (ETH_MACLCSR)* 的 RLPIEN 中断）。
2. 中断引脚 (sbd_intr_o) 生效。当主机读取 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 时，系统将清除 sbd_intr_o 中断。

在 sbd_intr_o 中断生效且 MAC Tx 也进入 LPI 模式后，可对 CSR 时钟进行门控关闭。如果在门控关闭 CSR 时钟时 MAC TX 未处于 LPI 模式，则 CSR 中将不会报告或更新 MAC 发送器上的事件。如需恢复 CSR 时钟，请等待来自 PHY 的 LPI 退出指示，随后 MAC 会在 lpi_intr_o（与 clk_rx_i 同步）上将 LPI 退出中断置为有效。读取 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 时，将清零 lpi_intr_o 中断。

在 Tx LPI 模式下门控关闭 CSR 时钟

当 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 的位 16 (LPIEN) 置 1 时，系统将执行以下操作：

1. 将发送 LPI 进入中断位（*LPI 控制和状态寄存器 (ETH_MACLCSR)* 的 TLPIEN 位）置 1。
2. 中断引脚 (sbd_intr_o) 生效。当主机读取 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 时，系统将清除 sbd_intr_o 中断。

在 sbd_intr_o 中断生效且 MAC RX 也进入 LPI 模式后，可对 CSR 时钟进行门控关闭。如果 CSR 时钟门控关闭时 MAC RX 未处于 LPI 模式，则 CSR 中将不会报告或更新 MAC 接收器上的事件。如需恢复 CSR 时钟，请在 MAC 退出 TX LPI 模式时打开 CSR 时钟。CSR 时钟恢复后，请将 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 的位 16 (LPIEN) 复位，从而使 MAC 退出 LPI 模式。

57.9.13 关于每秒脉冲数 (PPS) 灵活的输出的编程指南

基于 PPS 生成单脉冲

要基于 PPS 生成单脉冲：

1. 在 *PPS 控制寄存器 (ETH_MACPPSCR)* 中将 TRGTMODSEL[1:0] 位编程为 11 或 10（用于中断）。此操作会命令 MAC 将目标时间寄存器（*PPS 目标时间秒寄存器 (ETH_MACPPSTTSR)* 和 *PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)*）用作 PPS 信号输出的开始时间。
2. 在目标时间寄存器（寄存器 *PPS 目标时间秒寄存器 (ETH_MACPPSTTSR)* 和 *PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)*）中编程开始时间值。
3. 在 *PPS 宽度寄存器 (ETH_MACPPSWR)* 寄存器中编程 PPS 信号输出的宽度。
4. 将 *PPS 控制寄存器 (ETH_MACPPSCR)* 的 PPSCMD[3:0] 编程为 0001。此操作会命令 MAC 在目标时间寄存器中编程的时间内在 PPS 信号输出上生成单脉冲。

基于 PPS 生成下一个脉冲

执行 PPSCMD（PPSCMD 位 = 0）时，可通过在经过编程的开始时间之前给出取消开始命令 (PPSCMD=0011) 来取消脉冲生成。此外，还可以提前对下一个脉冲的行为进行编程。要编程下一个脉冲：

1. 在目标时间寄存器中编程下一个脉冲的开始时间。此时间应晚于上一个脉冲出现下降沿的时间。
2. 在 *PPS 宽度寄存器 (ETH_MACPPSWR)* 中编程下一个 PPS 信号输出的宽度。
3. 编程 *PPS 控制寄存器 (ETH_MACPPSCR)* 的 PPSCMD[3:0] 位，以在上一个脉冲无效后生成单个脉冲。此操作会命令 MAC 在目标时间寄存器中编程的时间内在 PPS 信号输出上生成单脉冲。

   如果在上一个脉冲变为低电平之前给出此命令，则新命令将覆盖上一个命令，并且外设可能只生成 1 个扩展脉冲。

基于 PPS 生成脉冲串

要基于 PPS 生成脉冲串：

1. 在 *PPS 控制寄存器 (ETH_MACPPSCR)* 中将 TRGTMODSEL[1:0] 位编程为 11 或 10（用于中断）。此操作会命令 MAC 将目标时间寄存器（*PPS 目标时间秒寄存器 (ETH_MACPPSTTSR)* 和 *PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)*）用于 PPS 信号输出的开始时间。
2. 在目标时间寄存器（寄存器 *PPS 目标时间秒寄存器 (ETH_MACPPSTTSR)* 和 *PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)*）中编程开始时间值。
3. 在 *PPS 间隔寄存器 (ETH_MACPPSIR)* 中编程 PPS 信号输出上的脉冲串之间的间隔值。
4. 在 *PPS 宽度寄存器 (ETH_MACPPSWR)* 中编程 PPS 信号输出的宽度。
5. 将 *PPS 控制寄存器 (ETH_MACPPSCR)* 中的 PPSCMD[3:0] 位编程为 0010。此操作会命令 MAC 在目标时间寄存器中编程的开始时间内在 PPS 信号输出上生成脉冲串。

   除非通过发出“经过一段时间后停止脉冲串”或“立即停止脉冲串”命令停止 PPS 脉冲串，否则默认情况下，PPS 脉冲串自由运行。
6. 在目标时间寄存器中编程停止值。确保在再次编程目标时间寄存器之前，*PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)* 中的 TSTRBUSY 位复位。
7. 将 *PPS 控制寄存器 (ETH_MACPPSCR)* 中的 PPSCMD[3:0] 位编程为 0100，以在第 6 步编程指定的停止时间过后，停止 PPS 信号输出上的脉冲串。

可通过在 PPSCMD[3:0] 位字段中编程 0101 随时停止脉冲串。

类似地，可通过在第 6 步中编程的时间结束之前，将 PPSCMD[3:0] 位编程为 0110 来取消停止脉冲串命令（在第 7 步中给出）。

可通过在第 2 步中编程的开始时间之前，将 PPSCMD[3:0] 编程为 0011 来停止生成脉冲串。

在不影响 PPS 的情况下生成中断

可通过 *PPS 控制寄存器 (ETH_MACPPSCR)* 中的 TRGTMODSEL[1:0] 位将目标时间寄存器（*PPS 目标时间秒寄存器 (ETH_MACPPSTTSR)* 和 *PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)*）编程为执行以下任一操作：

- 仅生成中断。
- 生成中断以及 PPS 开始和停止时间。
- 仅生成 PPS 开始和停止时间。

要将目标时间寄存器编程为仅生成中断事件：

1. 将 *PPS 控制寄存器 (ETH_MACPPSCR)* 的 TRGTMODSEL[1:0] 位编程为 00（用于中断）。此操作会命令 MAC 将目标时间寄存器用于目标时间中断。
2. 在目标时间寄存器中编程目标时间值。此操作会命令 MAC 在目标时间结束后生成中断。

   如果 TRGTMODSEL[1:0] 位发生更改（例如，用于控制 PPS），则中断生成操作将采用新模式和新编程的目标时间寄存器值。

注： *当所编程的目标时间小于系统时间秒寄存器 (ETH_MACSTSR) 和系统时间纳秒寄存器 (ETH_MACSTNR) 中的系统时间（即对应于过去的某个时间）时，时间戳状态寄存器 (ETH_MACTSSR) 中的 TSTRGTERR0 位将置 1。*

如果中断使能寄存器 (ETH_MACIER) 中的 TSIE 位置 1，则会产生中断 (sbd_intr_o)。

因此，为了避免不必要的中断，应采用以下正确的写入顺序：

1. *PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)。*
2. *PPS 目标时间秒寄存器 (ETH_MACPPSTTSR)。*
3. *PPS 间隔寄存器 (ETH_MACPPSIR)。*
4. *PPS 宽度寄存器 (ETH_MACPPSWR)。*
5. *PPS 控制寄存器 (ETH_MACPPSCR) 的 PPSCTRL[3:0]、PPSCTRL[3:0] 和 PPSEN0 位字段。*

57.9.14 关于 TSO 的编程指南

TCP 分段减荷 (TSO) 引擎用于将 TCP 分段功能减荷到硬件。要对 TSO 进行编程，请将 TSE 位置 1 以使能 TCP 数据包分段，并对描述符位字段进行编程，以便为当前数据包使能 TSO。

要编程 TSO，请按以下步骤操作：

1. 对相应*通道 x 发送控制寄存器 (ETH_DMACxTXCR)* 的 TSE 位进行编程，以使能该 DMA 中的 TCP 数据包分段。
2. 要为当前数据包使能 TSO，除了正常传输描述符设置外，还必须对以下描述符位字段进行编程：
   a) 使能 TDES3 的 TSE（位 18）。
   b) 在 TDES3 的位 17 到 0 中对未分段的 TCP/IP 数据包有效负载长度进行编程，并在 TDES3 的位 22 到 19 中对 TCP 报头长度进行编程。
   c) 在以下位字段中编程最大段大小：
   – *通道 x 控制寄存器 (ETH_DMACxCR)* 的 MSS[13:0]
   – 或上下文描述符中的 MSS

   如果在*通道 x 控制寄存器 (ETH_DMACxCR)* 和上下文描述符中均对 MSS[13:0] 位字段进行了编程，则会采用通过软件最新编程的序列。
3. 未分段的 TCP/IP 数据包报头应存储在第一个描述符的缓冲区 1 中。该缓冲区不能保存任何有效负载字节。有效负载被分配给缓冲区 2 和后续描述符的缓冲区。

小心： 如果在 TDES3 中针对非 TCP-IP 数据包使能了 TSE，则结果将不可预测。

57.9.15 关于在接收端执行 VLAN 过滤的编程指南

要在接收器端执行 VLAN 过滤，请按以下步骤操作：

1. 编程 *VLAN 标记寄存器 (ETH_MACVTR)* 中的以下位，以选择过滤方法：
   - ETV：使能 12 位 VLAN 标记比较或 16 位 VLAN 标记比较。
   - VTHM：VLAN 标记散列表匹配使能。
   - ERIVLT：使能内部 VLAN 标记或外部 VLAN 标记（以使能内部或外部 VLAN 标记过滤，双 VLAN 处理应通过将 EDVLP 置 1 来使能）
   - ERSVLM：使能接收 S-VLAN 匹配或 C-VLAN 匹配（对于要使能的 S-VLAN 处理，将 ESVL 置 1）
   - DOVLTC：忽略标记匹配的 VLAN 类型
   - VTIM：用于使能 VLAN 标记反向匹配，而非正常 VLAN 标记匹配
2. 针对 12 位或 16 位 VLAN 标记，编程 *VLAN 标记寄存器 (ETH_MACVTR)* 中的 VL 位。
3. 如果使能 VLAN 标记散列过滤，则编程 *VLAN 散列表寄存器 (ETH_MACVHTR)*。
   - ETV 位复位时，计算得出的 VLAN 标记 CRC-32 的高 4 位被反转，并用于索引 *VLAN 散列表寄存器 (ETH_MACVHTR)* 的内容。
   - ETV 位置 1 时，计算得出的 VLAN 标记 CRC-32 的高 4 位将用于索引 *VLAN 散列表寄存器 (ETH_MACVHTR)* 的内容。

   例如，ETV 位置 1 时，散列值 0b1000 用于选择 VLAN 散列表的位 8。当 ETV 位复位时，散列值 0b1000 用于选择 VLAN 散列表的位 7。

57.10 描述符

57.10.1 描述符概述

在以太网外设中，DMA 基于描述符的链表传输数据。应用程序在系统内存 (SRAM) 中创建描述符。支持以下两种描述符：

- **正常描述符**

  正常描述符用于数据包数据，以及用于提供适用于待发送数据包的控制信息。
- **上下文描述符**

  上下文描述符用于提供适用于待发送数据包的控制信息。

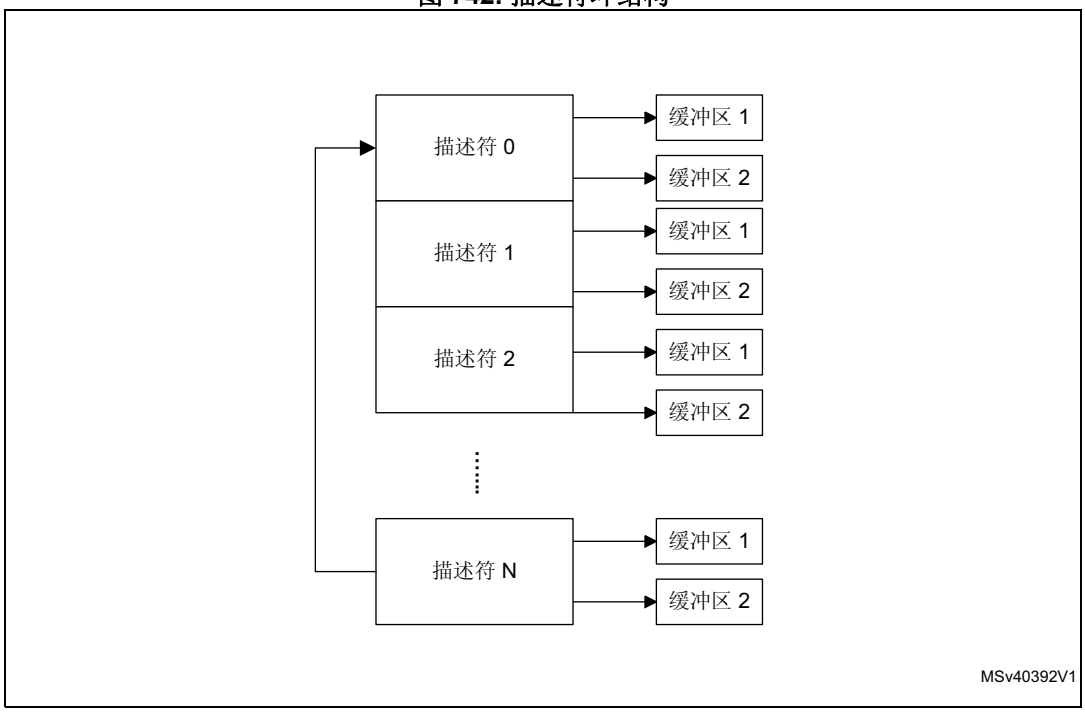
每个正常描述符均包含两个缓冲区和两个地址指针。这些缓冲区可使适配器端口与各种类型的内存管理方案兼容。

可将任意数量的描述符用于单个数据包。

57.10.2 描述符结构

以太网外设支持 DMA 描述符的环结构。

图 742. 描述符环结构



在环结构中，各描述符之间以*通道 x 控制寄存器 (ETH_DMACxCR)* 的 DSL 位字段中编程的 32 位字数为间隔彼此分开。应用程序需要在 DMA 通道的以下寄存器中编程总环长度，即，环结构内的描述符总数：

- *通道 x Tx 描述符环长度寄存器 (ETH_DMACxTXRLR)*)
- *通道 0 Rx 描述符环长度寄存器 (ETH_DMAC0RXRLR)*)

通道 x Tx 描述符尾指针寄存器 (ETH_DMACxTXDTPR) 或*通道 0 Rx 描述符尾指针寄存器 (ETH_DMAC0RXDTPR)* 包含指向描述符地址的指针 (*N*)。基址和当前描述符指针决定 DMA 可处理的当前描述符的地址。对于至少比描述符尾指针 (*N* – 1) 所指示的描述符少一个地址单元的描述符，均为 DMA 所有。DMA 会持续处理这些描述符，直至出现以下条件：

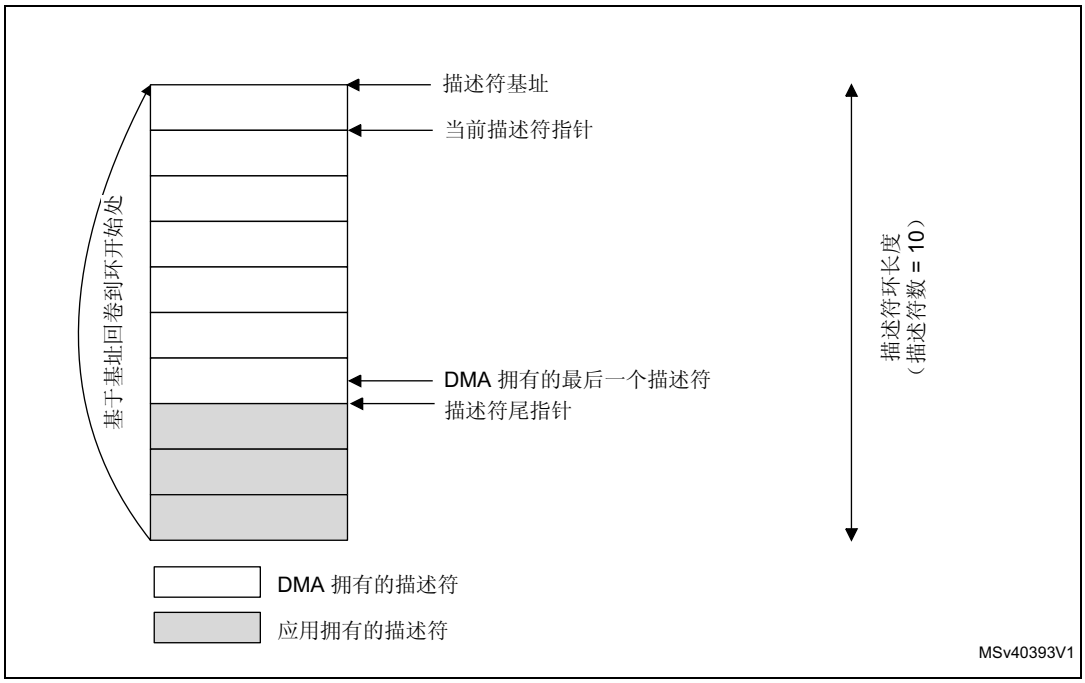
```
Current Descriptor Pointer == Descriptor Tail Pointer;
```

出现如上条件时，DMA 将进入挂起状态。应用程序必须对描述符尾指针寄存器执行写入操作，并更新尾指针，以满足以下条件：

```
Current Descriptor Pointer < Descriptor Tail Pointer;
```

到达环结束处时，DMA 会自动基于基址进行回卷，如*图 743：DMA 描述符环*所示。

图 743. DMA 描述符环



对于应用程序拥有的描述符，DES3 的 OWN 位复位为 0。

对于 DMA 拥有的描述符，OWN 位置 1。

在开始时，如果应用程序只有一个描述符，则会将最后一个描述符地址（尾指针）设为描述符基址 + 1。DMA 随即会处理第一个描述符，并等待应用程序对尾指针进行递增。

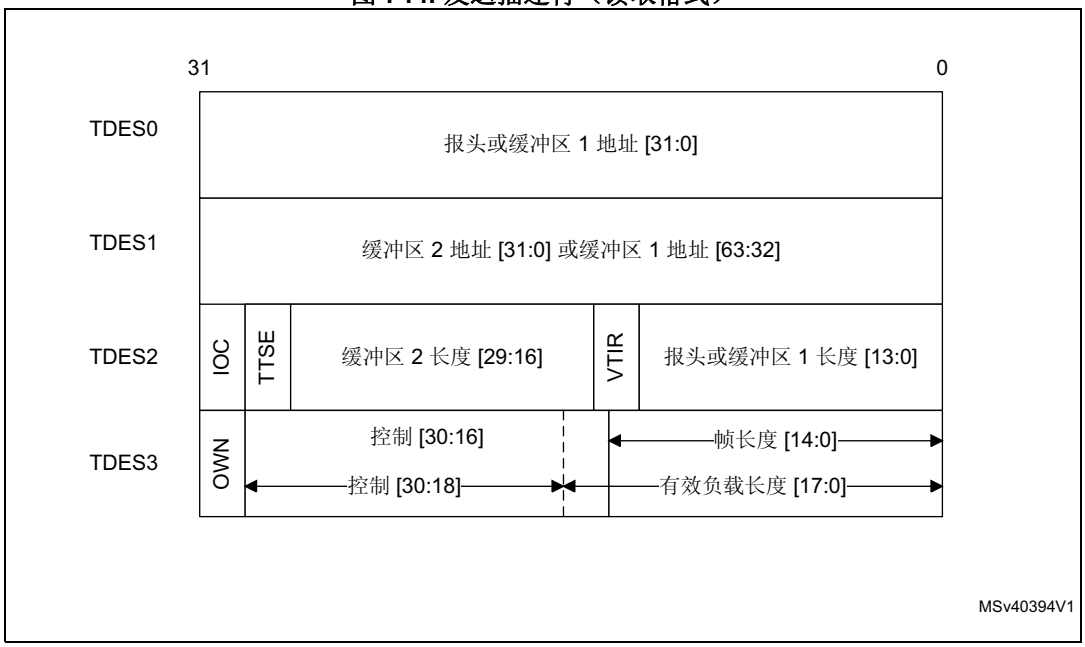
57.10.3 发送描述符

对于一个发送数据包，以太网外设 DMA 至少需要一个描述符。除了两个缓冲区、两个字节计数缓冲区和两个地址指针之外，发送描述符还具有控制位字段，此类位字段可用于按发送数据包来管理 MAC 操作。发送正常描述符具有以下两种格式：读取格式和回写格式

发送正常描述符（读取格式）

图 744 显示了发送正常描述符的读取格式。*表 556* 到*表 559* 给出了所有发送正常描述符（读取格式）的详细说明。

图 744. 发送描述符（读取格式）



- TDES0 正常描述符（读取格式）

表 556. TDES0 正常描述符（读取格式）

位	名称	说明
31:0	BUF1AP	**缓冲区 1 地址指针或 TSO 报头地址指针** 将以下位置 1 时，这些位指示缓冲区 1 的物理地址或 TSO 报头地址指针： – TDES3 的 TSE 位 – TDES3 的 FD 位

- TDES1 正常描述符（读取格式）

表 557. TDES1 正常描述符（读取格式）

位	名称	说明
31:0	BUF2AP	**缓冲区 2 或缓冲区 1 地址指针 (Buffer 2 or Buffer 1 Address Pointer)：** 使用描述符环结构时，这些位指示缓冲区 2 的物理地址。缓冲区地址对齐方式不限。

- TDES2 正常描述符（读取格式）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IOC	TTSE	B2L													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VTIR		HL 或 B1L													

表 558. TDES2 正常描述符（读取格式）

位	名称	说明
31	IOC	**完成时中断 (Interrupt on completion)：** 当前数据包发送完成时，该位会将*通道 x 状态寄存器 (ETH_DMACxSR)* 中的 TI 位置 1。
30	TTSE	**发送时间戳使能 (Transmit Timestamp Enable)** 该位将使能由描述符引用的发送数据包的 IEEE1588 时间戳。
29:16	B2L	**缓冲区 2 长度 (Buffer 2 Length)** 驱动程序会将该位字段置 1。该位字段置 1 时，即表示缓冲区 2 长度。
15:14	VTIR	**VLAN 标记插入或替换 (VLAN Tag Insertion or Replacement)：** 这些位请求 MAC 在发送数据包之前执行 VLAN 标记或取消标记操作。为数据包使能 VLAN 标记插入、替换或删除时，应用程序必须相应地设置 CRC 填充控制位。这些位的值如下： 00：不添加 VLAN 标记。 01：在发送前从数据包中删除 VLAN 标记。此选项应只与 VLAN 数据包一起使用。 10：使用在 *VLAN 包含寄存器 (ETH_MACVIR)* 或上下文描述符中编程的标记值插入 VLAN 标记。 11：使用在 *VLAN 包含寄存器 (ETH_MACVIR)* 或上下文描述符中编程的标记值替换数据包中的 VLAN 标记。此选项应只与 VLAN 数据包一起使用。
13:0	HL 或 B1L	**报头长度或缓冲区 1 长度 (Header length or buffer 1 length)** 对于报头长度，仅考虑位 [9:0]。位 13 到 0 仅应用于缓冲区 1 长度。 如果通过 TDES3 的 TSE 位使能 TCP 分段减荷功能，则该位字段等于报头长度。TDES3 中的 TSE 位置 1 时，报头长度包括从以太网源地址到 TCP 报头结束的长度（以字节为单位进行表示）。对于 TSO 功能，支持的最大报头长度为 1023 字节。 如果未使能 TCP 分段减荷功能，则该位字段等于缓冲区 1 长度。

- TDES3 正常描述符（读取格式）

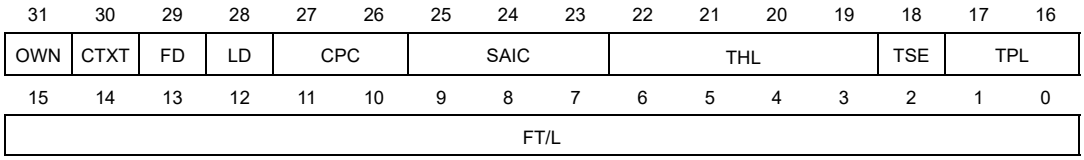


表 559. TDES3 正常描述符（读取格式）

位	名称	说明
31	OWN	**所有关系位 (Own bit)** 1：描述符为 DMA 所有。 0：描述符为应用程序所有。 在传输完相关缓冲区中给出的数据后，DMA 会将此位清零。
30	CTXT	**上下文类型 (Context Type)** 对于正常描述符，该位应设为 0。
29	FD	**第一个描述符 (First Descriptor)** 该位置 1 时，指示缓冲区中包含数据包的首段。
28	LD	**最后一个描述符 (Last Descriptor)** 该位置 1 时，指示缓冲区中包含数据包的末段。B1L 或 B2L 位字段应具有非零值。
27:26	CPC	**CRC 填充控制 (CRC Pad Control)** 该位字段控制针对 Tx 数据包的 CRC 和填充插入。只有第一个描述符位 (TDES3[29]) 置 1 时，该位字段才有效。位 [27:26] 的值如下： 00：CRC 和填充插入 MAC 会在长度大于或等于 60 字节的发送数据包末尾附加循环冗余校验 (CRC)。MAC 会自动将填充值和 CRC 附加到长度小于 60 字节的数据包。 01：CRC 插入（禁止填充插入） MAC 会在发送数据包的末尾附加 CRC，但不附加填充值。应用程序应确保从发送缓冲区传输的数据包中存在填充字节，即，从发送缓冲区传输的数据包长度大于或等于 60 字节。 10：禁止 CRC 插入 MAC 不会在发送数据包的末尾附加 CRC。应用程序应确保从发送缓冲区传输的数据包中存在填充和 CRC 字节。 11：CRC 替换 MAC 会用重新计算的 CRC 字节替换发送数据包的最后四个字节。应用程序应确保从发送缓冲区传输的数据包中存在填充和 CRC 字节。 TSE 位置 1 时，MAC 会忽略此位字段，因为将始终为分段进行 CRC 和填充插入。

表 559. TDES3 正常描述符（读取格式）（续）

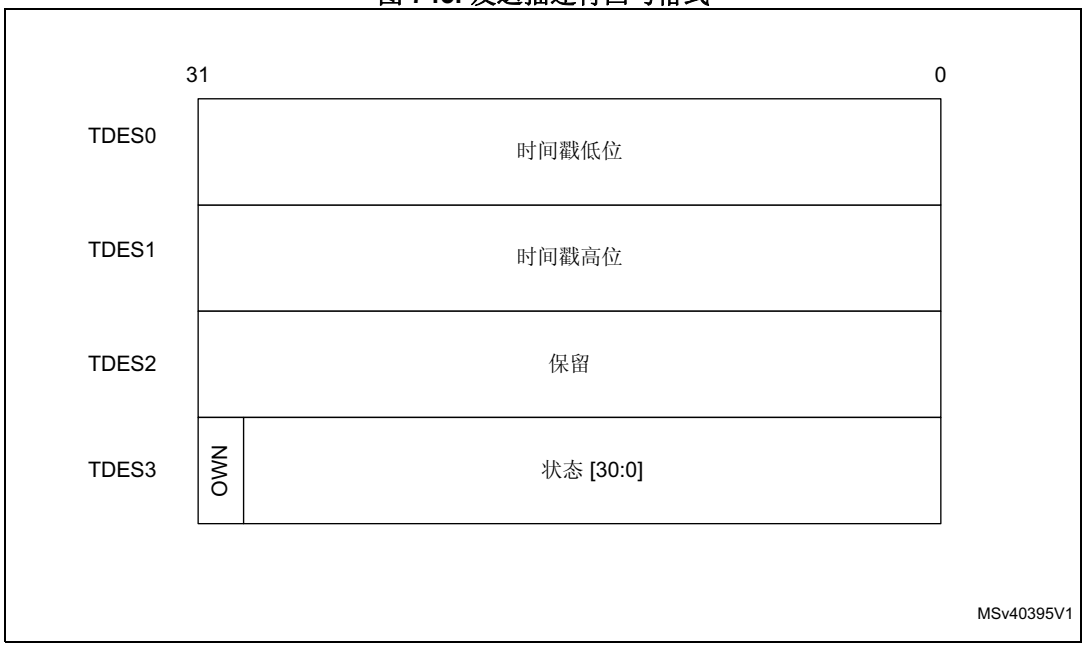
位	名称	说明
25:23	SAIC	**SA 插入控制 (SA Insertion Control)** 这些位请求 MAC 用 MAC 地址 0 寄存器中给出的值添加或替换以太网数据包中的源地址位字段。为数据包使能 SA 插入控制时，应用程序必须相应地设置 CRC 填充控制位。 位 25 指定用于源地址插入或替换的 MAC 地址寄存器（1 或 0）值。 下面对位 [24:23] 的值进行了说明： 00：不包含源地址 01：包含或插入源地址。为实现可靠发送，应用程序必须提供不包含源地址的帧。 10：替换源地址。为实现可靠发送，应用程序必须提供包含源地址的帧。 11：保留 这些位在首段控制位 (TDES3 [29]) 置 1 的情况下有效。
22:19	THL	**THL**：TCP 报头长度 (TCP Header Length) 如果 TSE 位置 1，则该位字段包含 TCP/UDP 报头的长度。对于 TCP 报头，该位字段的最小值必须为 5。UDP 报头的 THL 值必须等于 2。该位字段仅对第一个描述符有效。
18	TSE	**TCP 分段使能 (TCP Segmentation Enable)** 该位置 1 时，DMA 对数据包执行 TCP/UDP 分段。只有在 FD 位置 1 时，该位才有效。
17:16	CIC/TPL	**校验和插入控制或 TCP 有效负载长度 (Checksum Insertion Control or TCP Payload Length)** 这些位控制校验和的计算与插入。可采用以下值： 00：禁止插入校验和。 01：仅使能 IP 报头校验和的计算与插入。 10：使能 IP 报头校验和以及有效负载校验和的计算与插入，但不会在硬件中计算伪报头校验和。 11：使能 IP 报头校验和以及有效负载校验和的计算与插入，并在硬件中计算伪报头校验和。 TSE 位复位时，该位字段有效。TSE 位置 1 时，该位字段包含 TCP 有效负载长度的高位 [17:16]。这样可使 TCP 数据包长度位字段跨越 TDES3[17:0]，从而支持 256 KB 数据包长度。
15	TPL	**保留或 TCP 有效负载长度 (Reserved or TCP Payload Length)** TSE 位复位时，该位保留。TSE 位置 1 时，该位表示 TCP 有效负载长度 [17:0] 的位 15。
14:0	FL/TPL	**保留或 TCP 有效负载长度 (Reserved or TCP Payload Length)** TSE 位置 1 时，该位字段等于 TCP 有效负载长度的低 15 位。此长度不包括以太网报头或 TCP/UDP/IP 报头长度。 TSE 位复位时，该位保留。

发送正常描述符（回写格式）

回写格式仅适用于相应数据包的最后一个描述符。在 DMA 回写相应发送数据包的状态和时间戳信息的描述符中，LD 位 (TDES3[28]) 置 1。

图 745 显示了发送正常描述符的回写格式。*表 560* 到 *表 563* 给出了所有发送正常描述符（回写格式）的详细说明。

图 745. 发送描述符回写格式



- TDES0 正常描述符（回写格式）

表 560. TDES0 正常描述符（回写格式）(1)

位	名称	说明
31:0	TTSL	**发送数据包时间戳低位 (Transmit Packet Timestamp Low)** DMA 使用为对应发送数据包捕获的时间戳的 32 个最低有效位更新该位字段。只有在数据包第一个描述符中的 TDES2 的 TTSE 位置 1 时，DMA 才会写入时间戳。只有在描述符中的末段位 (LS) 置 1 并且时间戳状态 (TTSS) 位置 1 时，该位字段才保持时间戳。

1. 此格式仅适用于数据包的最后一个描述符。

- TDES1 正常描述符（回写格式）

表 561. TDES1 正常描述符（回写格式）(1)

位	名称	说明
31:0	TTSH	**发送数据包时间戳高位 (Transmit Packet Timestamp High)** DMA 使用为对应接收数据包捕获的时间戳的 32 个最高有效位更新该位字段。只有在数据包第一个描述符中的 TDES2 的 TTSE 位置 1 时，DMA 才会写入时间戳。只有在描述符中的末段位 (LS) 置 1 并且时间戳状态 (TTSS) 位置 1 时，该位字段才具有时间戳。

1. 此格式仅适用于数据包的最后一个描述符。

- TDES2 正常描述符（回写格式）

表 562. TDES2 正常描述符（回写格式）(1)

位	说明
31:0	保留

1. 此格式仅适用于数据包的最后一个描述符。

- TDES3 正常描述符（回写格式）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OWN	CTXT	FD	LD	保留										TTSS	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES	JT	FF	PCE	LoC	NC	LC	EC	CC				ED	UF	DB	IHE

表 563. TDES3 正常描述符（回写格式）(1)

位	名称	说明
31	OWN	**所有关系位 (Own bit)** 该位置 1 时，表示 DMA 拥有描述符。DMA 会在完成数据包发送时将该位清零。完成回写后，此位会置 0。
30	CTXT	**上下文类型 (Context Type)** 对于正常描述符，该位应置 0。
29	FD	**第一个描述符 (First Descriptor)** 该位指示缓冲区中包含数据包的首段。
28	LD	**最后一个描述符 (Last Descriptor)** 对于数据包的最后一个描述符，该位置 1。DMA 仅在数据包的最后一个描述符中才会对状态位字段进行写入操作。
27:18	保留	
17	TTSS	**Tx 时间戳状态 (Tx Timestamp Status)** 该状态位指示已经为相应的发送数据包捕获时间戳。该位置 1 时，TDES0 和 TDES1 将包含为发送数据包捕获到的时间戳值。该位字段仅在描述符末段控制位 (TDES3 [28]) 置 1 的情况下有效。
16	保留	
15	ES	**错误汇总 (Error Summary)** 该位指示以下位的逻辑或运算结果： TDES3[0]：IP 报头错误 TDES3[14]：Jabber 超时 TDES3[13]：数据包已刷新 TDES3[12]：有效负载校验和错误 TDES3[11]：载波丢失 TDES3[10]：无载波 TDES3[9]：延迟冲突 TDES3[8]：过度冲突 TDES3[3]：过度延迟 TDES3[2]：下溢错误

表 563. TDES3 正常描述符（回写格式）(1)（续）

位	名称	说明
14	JT	**Jabber 超时 (Jabber Timeout)** 该位指示 MAC 发送器经历了 jabber 超时。只有在*工作模式配置寄存器 (ETH_MACCR)* 的 JD 位未置 1 时，该位才会置 1。
13	FF	**数据包已刷新 (Packet Flushed)** 该位指示 DMA 或 MTL 已依照 CPU 发出的软件刷新命令刷新数据包。
12	PCE	**有效负载校验和错误 (Payload Checksum Error)** 该位指示校验和减荷引擎出现故障，并且未将任何校验和插入到封装的 TCP、UDP 或 ICMP 有效负载中。这种故障可能是由于字节数不足（即，未达到 IP 报头的有效负载长度位字段所指示的数量）而引起，或者因 MTL 在未计算出校验和的情况下开始在存储转发模式下将数据包转发到 MAC 发送器而引起。第二种错误条件仅发生在发送 FIFO 深度小于发送的以太网数据包的长度的情况下（以避免死锁），MTL 会在 FIFO 已满时开始转发数据包，即使在存储转发模式下也不例外。在数据包传输过程中检测到总线错误时，也可能发生此错误。
11	LoC	**载波丢失 (Loss of Carrier)** 该位指示数据包发送期间丢失载波（即，数据包发送期间有一个或多个发送时钟周期的 ETH_CRS 信号无效）。该位仅对在 MAC 处于半双工模式时实现无冲突发送的数据包有效。
10	NC	**无载波 (No Carrier)** 该位指示在发送期间未由 PHY 触发载波侦听信号。
9	LC	**延迟冲突 (Late Collision)** 该位指示数据包发送过程因冲突窗口（MII 模式下为 64 个字节时间，含前导码；GMII 模式下为 512 个字节时间，含前导码和载波扩展）后出现冲突而中止。如果下溢错误位置 1，则该位无效。
8	EC	**过度冲突 (Excessive Collision)** 该位指示尝试发送当前数据包时因出现 16 个连续冲突而中止发送。如果*工作模式配置寄存器 (ETH_MACCR)* 中的 DR 位置 1，则该位会在出现首个冲突后置 1，并且数据包发送过程将中止。
7:4	CC	**冲突计数 (Collision Count)** 此 4 位计数器值指示发送数据包之前出现的冲突个数。EC 位置 1 时，该计数无效。
3	ED	**过度延迟 (Excessive Deferral)** *工作模式配置寄存器 (ETH_MACCR)* 中的 DC 位置 1 时，该位将指示因延迟超过 24,288 个位时间（1000-Mbps 模式或使能巨型数据包模式下为 155,680 个位时间）而结束发送。
2	UF	**下溢错误 (Underflow Error)** 该位指示 MAC 因系统内存的数据未及时到达而中止了数据包的发送。在以下任意一种情况下，都可能会发生下溢错误： DMA 在数据包发送期间遇到发送缓冲区为空的情况。 应用程序填充 MTL Tx FIFO 的速率慢于 MAC 发送速率 发送过程进入挂起状态，并且 ETH_MTLISR 寄存器中对应于队列的下溢位置 1。

表 563. TDES3 正常描述符（回写格式）(1)（续）

位	名称	说明
1	DB	**延迟位 (Deferred Bit)** 该位指示 MAC 在发送前因存在载波而延迟。该位仅在半双工模式下有效。
0	IHE	**IP 报头错误 (IP Header Error)** IP 报头错误位置 1 时，该位指示校验和减荷引擎检测到 IP 报头错误。如果以太网类型位字段指示 IPv4 有效负载，即使 COE 检测到 IP 报头错误，也会插入 IPv4 报头校验和。

1. 此格式仅适用于数据包的最后一个描述符。

发送上下文描述符

可在数据包描述符之前的任何时间提供发送上下文描述符。上下文对当前数据包和后续数据包有效。上下文描述符用于提供时间戳（用于进行单步时间戳校准）和 VLAN 标记 ID（用于 VLAN 插入功能）。只能针对上下文描述符进行回写来复位 OWN 位。

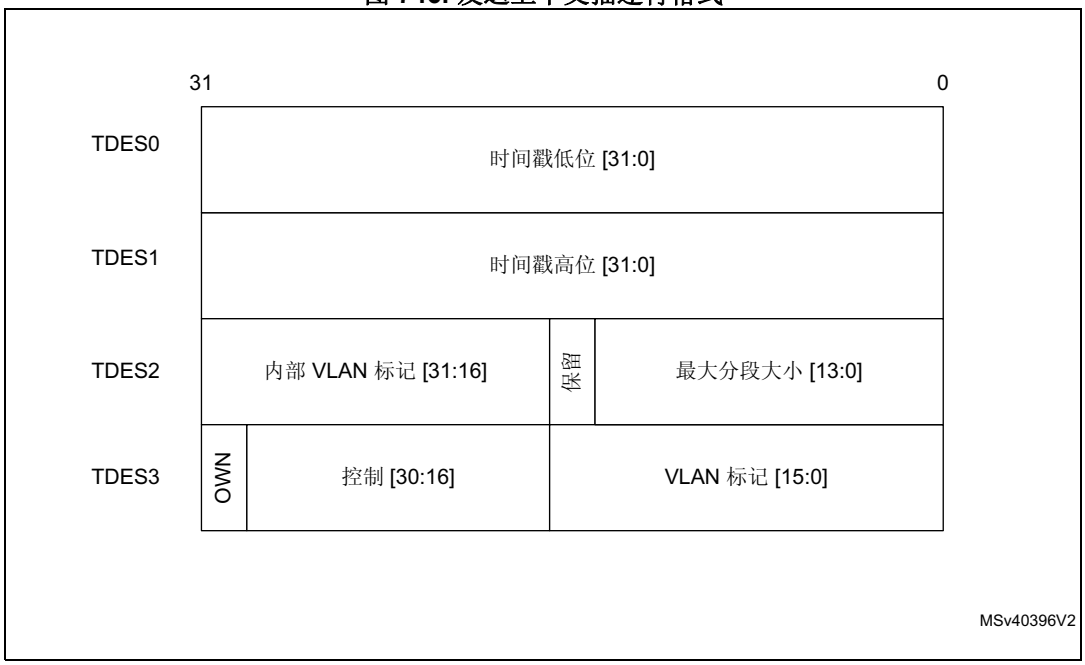
注： *应用程序在上下文描述符中提供的 VLAN 标记 ID 和 MSS 值（其相应的有效位置 1）由 DMA 在内部进行存储。*

所提供的外部或内部 VLAN 标记的有效位置 1 时，DMA 始终会将最后一个有效 VLAN 标记传递给 MTL。应用程序不能使 DMA 存储的有效 VLAN 标记失效。VLAN 标记将基于为数据包提供的控制输入进行插入或替换。

内部 VLAN 标记控制输入仅用于紧跟上下文描述符之后的数据包。如果数据包需要 DMA 为其使用内部 VLAN 标记控制输入，应用程序必须在数据包的正常描述符之前提供上下文描述符。

图 746 显示了发送上下文描述符的格式。*表 564* 到*表 567* 给出了所有发送上下文描述符的详细说明。

图 746. 发送上下文描述符格式



- TDES0 上下文描述符（读取格式）

表 564. TDES0 上下文描述符

位	名称	说明
31:0	TTSL	**发送数据包时间戳低位 (Transmit Packet Timestamp Low)** 对于单步校准，驱动程序可以在该描述符字中提供时间戳的低 32 位。DMA 将该值用作低位字来进行单步时间戳校准。只有在 TDES3 上下文描述符的 OSTC 和 TCMSSV 位置 1 时，该位字段才有效。

- TDES1 上下文描述符（读取格式）

表 565. TDES1 上下文描述符

位	名称	说明
31:0	TTSH	**发送数据包时间戳高位 (Transmit Packet Timestamp High)** 对于单步校准，驱动程序可以在该描述符中提供时间戳的高 32 位。DMA 将该值用作高位字来进行单步时间戳校准。只有在 TDES3 上下文描述符的 OSTC 和 TCMSSV 位置 1 时，该位字段才有效。

- TDES2 上下文描述符（读取格式）

表 566. TDES2 上下文描述符

位	名称	说明
31:16	IVT	**内部 VLAN 标记 (Inner VLAN Tag)** TDES3 上下文描述符的 IVLTV 位置 1 且 TDES3 上下文描述符的 TCMSSV 和 OSTC 位复位时，TDES2[31:16] 包含要插入到后续发送数据包中的内部 VLAN 标记。
15:14	保留	
13:0	MSS	**最大段大小 (Maximum Segment Size)** 对 TCP/IP 有效负载进行分段时使用此段大小。只有在 TDES3 上下文描述符的 TCMSSV 位置 1 且 TDES3 上下文描述符的 OSTC 位复位时，该位字段才有效。

- TDES3 上下文描述符（读取格式）

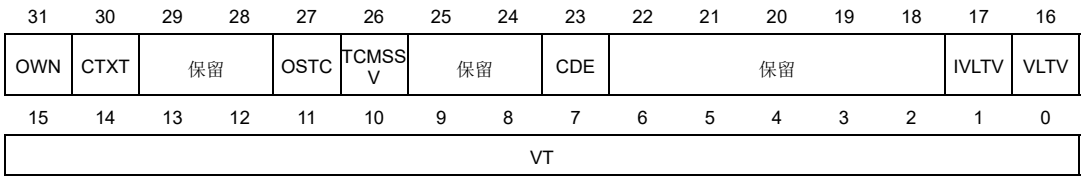


表 567. TDES3 上下文描述符

位	名称	说明
31	OWN	**所有关系位 (Own bit)** 1：描述符为 DMA 所有。 0：描述符为应用程序所有。 DMA 会在执行读取操作后立即将此位清零。
30	CTXT	**上下文类型 (Context Type)** 对于上下文描述符，该位应置 1。
29:28	保留	
27	OSTC	**单步时间戳校准使能 (One-Step Timestamp Correction Enable)** 该位置 1 时，DMA 将参考 TDES0 和 TDES1 中提供的时间戳值执行单步时间戳校准。
26	TCMSSV	**单步时间戳校准输入或 MSS 有效 (One-Step Timestamp Correction Input or MSS Valid)** 该位和 OSTC 位置 1 时，表示 TDES0 和 TDES1 中提供的时间戳校准输入有效。 OSTC 位复位，且后续正常描述符中的该位以及 TDES3 的 TSE 位置 1 时，表示 TDES2 中的 MSS 输入有效。
25:24	保留	
23	CDE	**上下文描述符错误 (Context Descriptor Error)** 该位置 1 时，表示上下文描述符不正确。关闭上下文描述符时，DMA 会在回写期间将此位置 1。 上下文描述符错误可能为： – 上下文描述符的序列错误。例如，某个位置位于数据包的第一个描述符之前。 – 所有位置 1。 – CD、LD 和 FD 位置 1。 *注意：当由于所有位置 1 或 CTXT、LD 和 FD 位置 1 而导致出现上下文描述符错误时，发送 DMA 将会关闭发送描述符，且 DE 和 LD 位将置 1。一旦将相应第一个描述符的 TDES2 中的 IOC 位置 1，发送 DMA 会将通道 x 状态寄存器 (ETH_DMACxSR) 中的 TI 位置 1。* *系统可能会根据发送描述符的 CTXT、LD 和 FD 位，将后续描述符视为第一个描述符（即使 FD 位未置 1），并发送部分数据包。* *此错误将归类为异常事件：只能通过发出软件复位命令来恢复（不支持“DMA 停止-重新配置-重新启动”的恢复机制）*
22:20	保留	

表 567. TDES3 上下文描述符 （续）

位	名称	说明
19:18	IVTIR	内部 VLAN 标记插入或替换 (Inner VLAN Tag Insert or Replace) 这些位置 1 时，会请求 MAC 在发送数据包之前执行内部 VLAN 标记或取消标记操作。如果修改了数据包的 VLAN 标记，则 MAC 会自动重新计算并替换 CRC 字节。 该位字段具有以下值： 00：不添加内部 VLAN 标记。 01：在发送前从数据包中删除内部 VLAN 标记。此选项应只与 VLAN 帧一起使用。 10：使用在*内部 VLAN 包含寄存器 (ETH_MACIVIR)* 或上下文描述符中编程的标记值插入内部 VLAN 标记。 11：使用在*内部 VLAN 包含寄存器 (ETH_MACIVIR)* 或上下文描述符中编程的标记值替换数据包中的内部 VLAN 标记。此选项应只与 VLAN 帧一起使用。
17	IVLTV	内部 VLAN 标记有效 (Inner VLAN Tag Valid) 该位置 1 时，表示 TDES2 的 IVT 位字段有效。
16	VLTV	VLAN 标记有效 (VLAN Tag Valid) 该位置 1 时，表示 TDES3 的 VT 位字段有效。
15:0	VT	VLAN 标记 (VLAN Tag) 该位字段包含要在数据包中插入或替换的 VLAN 标记。只有在 *VLAN 包含寄存器 (ETH_MACVIR)* 的 VLTI 位复位时，该位字段才用作 VLAN 标记。

57.10.4 接收描述符

只有在尾指针不同于基址指针或当前指针时，以太网外设中的 DMA 才会尝试读取描述符。建议使用长度至少可容纳 MAC 所接收到的两个完整数据包的描述符环；否则，会因描述符不可用而导致 DMA 的性能受到显著影响。在这种情况下，MTL RxFIFO 已满，将开始丢弃数据包。

存在以下两种接收描述符：

- 支持读取和回写格式的正常描述符
- 上下文描述符

接收到的所有描述符均由软件准备，并被作为“正常”描述符提供给 DMA（有关描述符内容的详细说明，请参见*图 747：接收正常描述符（读取格式）*）。DMA 读取该描述符，并且在将接收到的数据包（或数据包的一部分）传送到由描述符指示的缓冲区后，Rx DMA 会关闭具有相应数据包状态的描述符。状态格式如*图 748：接收正常描述符（回写格式）*所示。

对于某些数据包，正常描述符位不足以写入完整状态。对于这类数据包，Rx DMA 会将扩展状态写入下一个描述符（无需处理或使用嵌入到该描述符中的缓冲区指针）。此回写描述符的格式和内容如*图 749：接收上下文描述符*所示。

接收正常描述符（读取格式）

图 747 显示了接收正常描述符的读取格式。*表 568* 到*表 571* 给出了所有接收正常描述符（读取格式）的详细说明。

图 747. 接收正常描述符（读取格式）

注： *在接收描述符（读取格式）中，如果缓冲区地址位字段只包含 0，则 MAC 不会将数据传输到该缓冲区，并会跳到下一个缓冲区或下一个描述符。*

- RDES0 正常描述符（读取格式）

表 568. RDES0 正常描述符（读取格式）

位	名称	说明
31:0	BUF1AP	**缓冲区 1 地址指针 (Buffer 1 Address Pointer)** 这些位指示缓冲区 1 的物理地址。 应用程序可针对该缓冲区编程字节对齐地址，即，该位字段的 LS 位可以为非零值。不过，在传送数据包开始时，DMA 将在 64/128 位配置时 RDES2[1:0]=0 的情况下执行写入操作。但数据包数据会根据缓冲区地址指针中给出的实际偏移进行移位。 如果地址指针指向用于存储数据包的中间部分或最后一部分的缓冲区，则 DMA 将忽略偏移地址，并会写入完整位置（根据数据宽度的指示）。

- RDES1 正常描述符（读取格式）

表 569. RDES1 正常描述符（读取格式）

位	名称	说明
31:0	保留	保留位字段。

- RDES2 正常描述符（读取格式）

表 570. RDES2 正常描述符（读取格式）

位	名称	说明
31:0	BUF2AP	**缓冲区 2 地址指针 (Buffer 2 Address Pointer)** 这些位指示缓冲区 2 物理地址。 只有在传送数据包的起始字节时，RxDMA 才会使用指针地址的 LS 位。如果 BUF2AP 提供用于存储数据包的中间部分或最后一部分的缓冲区的地址，DMA 将忽略 RDES2[1:0]=0 并会写入完整位置。

- RDES3 正常描述符（读取格式）

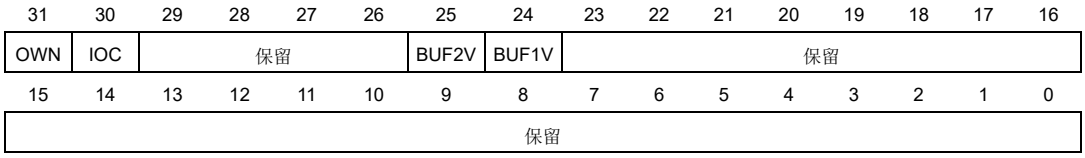


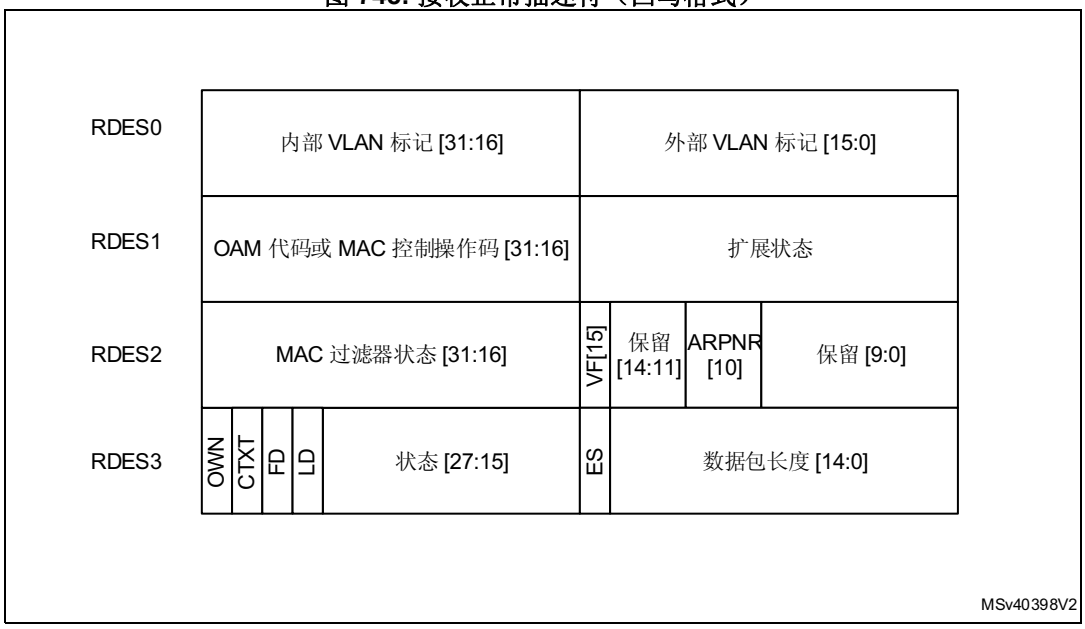
表 571. RDES3 正常描述符（读取格式）

位	名称	说明
31	OWN	**所有关系位 (Own bit)** 该位置 1 时，表示 DMA 拥有描述符。该位复位时，表示应用程序拥有描述符。以下任一条件为真时，DMA 会将该位清零： – DMA 完成数据包接收 – 与描述符相关的缓冲区已满
30	IOC	**完成时使能中断 (Interrupt Enabled on Completion)** 如果该位置 1，则在 DMA 关闭此描述符时会向应用程序发出中断。
29:26	保留	
25	BUF2V	**缓冲区 2 地址有效 (Buffer 2 Address Valid)** 该位置 1 时，向 DMA 指示 RDES2 中指定的缓冲区 2 地址有效。应用程序必须将该位置 1，以便 DMA 可使用 RDES2 中缓冲区 2 地址所指向的地址来写入接收到的数据包数据。
24	BUF1V	**缓冲区 1 地址有效 (Buffer 1 Address Valid)** 置 1 时，向 DMA 指示 RDES0 中指定的缓冲区 1 地址有效。 应用程序必须将该位置 1，以便 DMA 可使用 RDES0 中缓冲区 1 地址所指向的地址来写入接收到的数据包数据。
23:0	保留	

接收正常描述符（回写格式）

图 748 显示了接收正常描述符的回写格式。*表 572* 到*表 575* 给出了所有接收正常描述符（回写格式）的详细说明。

图 748. 接收正常描述符（回写格式）



- RDES0 正常描述符（回写格式）

表 572. RDES0 正常描述符（回写格式）

位	名称	说明
31:16	IVT	**内部 VLAN 标记 (Inner VLAN Tag)** 如果 RDES3 的 RS0V 位置 1，则该位字段包含接收到的数据包的内部 VLAN 标记。该位仅在使能了双 VLAN 标记处理和 VLAN 标记去除的情况下有效。
15:0	OVT	**外部 VLAN 标记 (Outer VLAN Tag)** 如果 RDES3 的 RS0V 位置 1，则该位字段包含接收到的数据包的外部 VLAN 标记。

- RDES1 正常描述符（回写格式）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPC															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TD	TSA	PV	PFT	PMT				IPCE	IPCB	IPV6	IPV4	IPHE	PT		

表 573. RDES1 正常描述符（回写格式）(1)

位	名称	说明
31:16	OPC	**OAM 子类型代码 (OAM Subtype Code) 或 MAC 控制数据包操作码 (MAC Control Packet opcode)** OAM 子类型代码 如果 RDES3 的位 [18:16] 置为 111，则该位字段包含 OAM 子类型和代码位字段。 MAC 控制数据包操作码 如果 RDES3 的位 [18:16] 置为 110，则该位字段包含 MAC 控制数据包操作码位字段。
15	TD	**时间戳被丢弃 (Timestamp Dropped)** 该位表示已为此数据包捕获时间戳，但因发生上溢而在 MTL Rx FIFO 中将时间戳丢弃。
14	TSA	**时间戳可用 (Timestamp Available)** 存在时间戳时，该位表示时间戳值在上下文描述符字 2 (RDES2) 和字 1 (RDES1) 中可用。只有在最后一个描述符位 (RDES3 [28]) 置 1 时，该位才有效。 上下文描述符被写入到紧跟数据包的最后一个正常描述符之后的下一个描述符。
13	PV	**PTP 版本 (PTP Version)** 1：接收到的 PTP 消息采用 IEEE 1588 版本 2 格式 0：接收到的 PTP 消息采用 IEEE 1588 版本 1 格式
12	PFT	**PTP 数据包类型 (PTP Packet Type)** 该位表示 PTP 消息直接通过以太网发送。
11:8	PMT	**PTP 消息类型 (PTP Message Type)** 将对这些位进行编码，以给出所接收消息的类型： 0000：未接收到任何 PTP 消息 0001：SYNC（所有时钟类型） 0010：Follow_Up（所有时钟类型） 0011：Delay_Req（所有时钟类型） 0100：Delay_Resp（所有时钟类型） 0101：Pdelay_Req（针对点对点透明时钟） 0110：Pdelay_Resp（针对点对点透明时钟） 0111：Pdelay_Resp_Follow_Up（针对点对点透明时钟） 1000：发布 1001：管理 1010：信号发送 1011–1110：保留 1111：采用保留消息类型的 PTP 数据包
7	IPCE	**IP 有效负载错误 (IP Payload Error)** 该位置 1 时，表示以下任一项： – 由 MAC 计算的 16 位 IP 有效负载校验和（即 TCP、UDP 或 ICMP 校验和）与接收段中对应的校验和位字段不匹配。 – TCP、UDP 或 ICMP 段长度与 IP 报头位字段中的有效负载长度值不匹配。 – TCP、UDP 或 ICMP 段长度小于 TCP、UDP 或 ICMP 的最小允许段长度。 该位置 1 时，RDES3 的位 15 (ES) 不置 1。

表 573. RDES1 正常描述符（回写格式）(1)（续）

位	名称	说明
6	IPCB	**绕过 IP 校验和 (IP Checksum Bypassed)** 该位表示绕过校验和减荷引擎。
5	IPV6	**存在 IPv6 报头 (IPv6 header Present)** 该位表示检测到 IPV6 报头。
4	IPV4	**存在 IPV4 报头 (IPV4 Header Present)** 该位表示检测到 IPV4 报头。
3	IPHE	**IP 报头错误 (IP Header Error)** – 该位置 1 时，表示以下任一项： – 由 MAC 计算的 16 位 IPv4 报头校验和与接收的校验和字节不匹配。 – IP 数据报版本与以太网类型值不一致。 – 以太网数据包不具有预期 IP 报头字节数。 该位在位 5 或位 4 置 1 时有效。
2:0	PT	**有效负载类型 (Payload Type)** 这些位表示由接收校验和减荷引擎 (COE) 处理的 IP 数据报中封装的有效负载类型： 000：类型未知，或未处理 IP/AV 有效负载 001：UDP 010：TCP 011：ICMP 100：如果存在 IPV4 报头位置 1，则为 IGMP 其他值：保留。 如果 COE 因存在 IP 报头错误或分段 IP 而未处理 IP 数据报的有效负载，则会将这些位设为 3'b000。

1. 回写格式的状态位字段仅对最后一个描述符（RDES3[28] 置 1）有效。

- RDES2 正常描述符（回写格式）

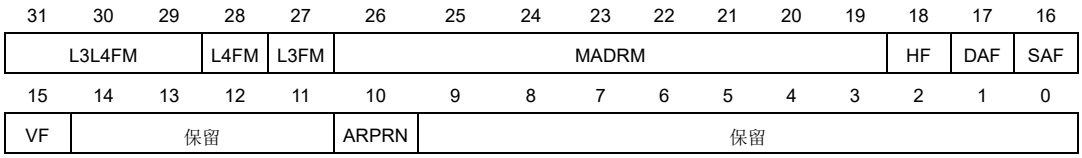


表 574. RDES2 正常描述符（回写格式）

位	名称	说明
31:29	L3L4FM	**第 3 层和第 4 层过滤器编号匹配 (Layer 3 and Layer 4 Filter Number Matched)** 这些位表示与接收到的数据包匹配的第 3 层和第 4 层过滤器的编号： – 000：过滤器 0 – 001：过滤器 1 – 010：过滤器 2 – 011：过滤器 3 – 100：过滤器 4 – 101：过滤器 5 – 110：过滤器 6 – 111：过滤器 7 只有在位 28 或位 27 置为高电平时，该位字段才有效。如果有多个过滤器匹配，则这些位会给出最低级过滤器的编号。
28	L4FM	**第 4 层过滤器匹配 (Layer 4 Filter Match)** 该位置 1 时，表示接收到的数据包与使能的第 4 层端口号位字段之一匹配。只有在下列条件之一为真时，才会给出该状态： – 第 3 层位字段未使能，所有使能的第 4 层位字段均匹配 – 所有已使能的第 3 层和第 4 层过滤器位字段均匹配 如果有多个过滤器匹配，该位会给出由位 [31:29] 指示的过滤器的第 4 层过滤器状态。
27	L3FM	**第 3 层过滤器匹配 (Layer 3 Filter Match)** 该位置 1 时，表示接收到的数据包与使能的第 3 层 IP 地址位字段之一匹配。只有在下列条件之一为真时，才会给出该状态： – 所有使能的第 3 层位字段均匹配，所有使能的第 4 层位字段均被绕过 – 所有使能的过滤器位字段均匹配 如果有多个过滤器匹配，该位会给出由位 [31:29] 指示的过滤器的第 3 层过滤器状态。
26:19	MADRM	**MAC 地址匹配或散列值 (MAC Address Match or Hash Value)** 当 HF 位复位时，该位字段包含与接收到的数据包的目标地址匹配的 MAC 地址寄存器编号。只有在 DAF 位复位时，该位字段才有效。 HF 位置 1 时，该位字段包含由 MAC 计算的散列值。散列过滤器寄存器中对应于散列值的位置 1 时，数据包会通过散列过滤器。
18	HF	**散列过滤器状态 (Hash Filter Status)** 该位置 1 时，表示数据包通过 MAC 地址散列过滤器。位 [26:19] 表示散列值。
17	DAF	**目标地址过滤失败 (Destination Address Filter Fail)** 该位置 1 时，表示数据包未通过 MAC 中的 DA 过滤。
16	SAF	**SA 地址过滤失败 (SA Address Filter Fail)** 该位置 1 时，表示数据包未通过 MAC 中的 SA 过滤。
15	VF	**VLAN 过滤器状态 (VLAN Filter Status)** 该位置 1 时，表示接收到的数据包的 VLAN 标记通过 VLAN 过滤。
14:11	保留	
10	ARPNR	**未生成 ARP 响应 (ARP Reply Not Generated)** 该位置 1 时，表示 MAC 未针对接收到的 ARP 请求数据包生成 ARP 响应。MAC 忙于针对较早 ARP 请求发送 ARP 响应时，该位置 1（一次只能处理一个 ARP 请求）。
9:0	保留	

- RDES3 正常描述符（回写格式）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OWN	CTXT	FD	LD	RS2V	RS1V	RS0V	CE	GP	RWT	OE	RE	DE	LT		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES	PL														

表 575. RDES3 正常描述符（回写格式）

位	名称	说明
31	OWN	**所有关系位 (Own bit)** 1：描述符为 DMA 所有。 0：描述符为应用程序所有。 以下任一条件为真时，DMA 会将该位清零： DMA 完成数据包接收 与描述符相关的缓冲区已满
30	CTXT	**接收上下文描述符 (Receive Context Descriptor)** 该位置 1 时，表示当前描述符为上下文类型描述符。对于正常接收描述符，DMA 向该位写入 0。 将 CTXT 与 FD 位搭配使用时，{CTXT, FD} 的可能值为： 00：中间描述符 01：第一个描述符 (First Descriptor) 10：保留 11：描述符错误（由于所有位置 1） *注意：当发生描述符错误时，接收 DMA 会关闭接收描述符，以指示发生了描述符错误。系统将跳过此接收描述符，且不使用缓冲地址来写入数据包数据。接收 DMA 会将通道 x 状态寄存器 (ETH_DMACxSR) 的 CDE 位字段置 1，但即使 IOC 位字段置 1，也不会将 RI 位字段置 1，原因是系统不会将其标记为数据包的最后一个接收描述符。系统将使用后续的有效接收描述符来写入数据包数据。*
29	FD	**第一个描述符 (First Descriptor)** 该位置 1 时，表示该描述符包含数据包的第一个缓冲区。如果第一个缓冲区的大小为 0，则第二个缓冲区将包含数据包的起始字节。如果第二个缓冲区的大小也为 0，则下一个描述符将包含数据包的起始字节。 有关如何将 CTXT 位与 FD 位搭配使用的详细信息，请参见 CTXT 位的说明。
28	LD	**最后一个描述符 (Last Descriptor)** 该位置 1 时，表示该描述符所指向的缓冲区为数据包的最后几个缓冲区。
27	RS2V	**接收状态 RDES2 有效 (Receive Status RDES2 Valid)** 该位置 1 时，表示 RDES2 中的状态有效，且由 DMA 写入。只有在 RDES3 的 LD 位置 1 时，该位才有效。
26	RS1V	**接收状态 RDES1 有效 (Receive Status RDES1 Valid)** 该位置 1 时，表示 RDES1 中的状态有效，且由 DMA 写入。只有在 RDES3 的 LD 位置 1 时，该位才有效。
25	RS0V	**接收状态 RDES0 有效 (Receive Status RDES0 Valid)** 该位置 1 时，表示 RDES0 中的状态有效，且由 DMA 写入。只有在 RDES3 的 LD 位置 1 时，该位才有效。

表 575. RDES3 正常描述符（回写格式）（续）

位	名称	说明
24	CE	**CRC 错误 (CRC Error)** 该位置 1 时，表示接收到的数据包发生循环冗余校验 (CRC) 错误。只有在 RDES3 的 LD 位置 1 时，该位字段才有效。
23	GP	**大型数据包 (Giant Packet)** 该位置 1 时，表示数据包长度超出指定的最大以太网大小，即，1518、1522 或 2000 字节（如果巨型数据包使能位置 1，则为 9018 或 9022 字节）。 大型数据包仅表示数据包长度，不会导致数据包截断。
22	RWT	**接收看门狗超时 (Receive Watchdog Timeout)** 该位置 1 时，表示接收看门狗定时器在接收当前数据包时已到期。看门狗超时后，当前数据包会被截断。
21	OE	**上溢错误 (Overflow Error)** 该位置 1 时，表示接收到的数据包因 Rx FIFO 中发生缓冲区上溢而损坏。 只有在 DMA 将部分数据包传输到应用程序时，该位才会置 1。仅当 Rx FIFO 工作在阈值模式下时才会出现这种情况。在存储转发模式下，所有部分数据包均会在 Rx FIFO 中被完全丢弃。
20	RE	**接收错误 (Receive Error)** 该位置 1 时，表示在数据包接收期间 ETH_RX_DV 信号置为有效时，ETH_RX_ER 信号置为有效。此错误还包括 GMII 和半双工模式下的载波扩展错误。错误可能是扩展不足或无扩展，或是在扩展期间出现错误 (rxd!= 0f)。
19	DE	**Dribble 位错误 (Dribble Bit Error)** 该位置 1 时，表示接收到的数据包具有非整数倍数的字节（奇数半字节）。该位仅在 MII 模式下有效。
18:16	LT	**长度/类型位字段 (Length/Type Field)** 该位字段指示接收到的数据包为长度数据包或类型数据包。这 3 个位的编码如下： 000：数据包为长度数据包 001：数据包为类型数据包。 011：数据包为 ARP 请求数据包类型 100：数据包为带有 VLAN 标记的类型数据包 101：数据包为带有双 VLAN 标记的类型数据包 110：数据包为 MAC 控制数据包类型 111：数据包为 OAM 数据包类型 010：保留
15	ES	**错误汇总 (Error Summary)** 该位置 1 时，表示以下位的逻辑或运算结果： RDES3[19]：Dribble 错误 RDES3[20]：接收错误 RDES3[21]：上溢错误 RDES3[22]：看门狗超时 RDES3[23]：大型数据包 RDES3[24]：CRC 错误 只有在 RDES3 的 LD 位置 1 时，该位字段才有效。

表 575. RDES3 正常描述符（回写格式）（续）

位	名称	说明
14:0	PL	**数据包长度 (Packet Length)** 这些位指示传输到系统内存中的接收数据包的字节长度（含 CRC）。 RDES3 的 LD 位置 1 且上溢错误位复位时，该位字段有效。使能 IP 校验和计算并且接收到的数据包不是 MAC 控制数据包时，数据包长度还包括两个附加到以太网数据包的字节。 当 RDES3 的 LD 位复位时，此位字段将包含已为当前数据包传输的（部分）累计字节数。

接收上下文描述符

该描述符对于应用程序是只读的。只能由 DMA 对该描述符进行写入操作。

上下文描述符提供与最后接收到的数据包有关的扩展状态信息。RDES3 的位 30 表示上下文类型描述符。

图 749 显示了接收上下文描述符的格式。*表 576* 到*表 579* 给出了所有接收上下文描述符的详细说明。

图 749. 接收上下文描述符

- RDES0 上下文描述符

表 576. RDES0 上下文描述符

位	名称	说明
31:0	RTSL	**接收数据包时间戳低位 (Receive Packet Timestamp Low)** DMA 使用为对应接收数据包捕获的时间戳的 32 个最低有效位更新该位字段。当该位字段和 RDES1 的 RTSH 位字段均显示值 1 时，必须视为时间戳已损坏。

- RDES1 上下文描述符

表 577. RDES1 上下文描述符

位	位字段	说明
31:0	RTSH	**接收数据包时间戳高位 (Receive Packet Timestamp High)** DMA 使用为对应接收数据包捕获的时间戳的 32 个最高有效位更新该位字段。当该位字段和 RDES0 的 RTSL 位字段均显示值 1 时，必须视为时间戳已损坏。

- RDES2 上下文描述符

表 578. RDES2 上下文描述符

位	说明
31:0	保留

- RDES3 上下文描述符

表 579. RDES3 上下文描述符

位	名称	说明
31	OWN	**所有关系位 (Own Bit)** 1：描述符为 DMA 所有 0：描述符为应用程序所有。 以下任一条件为真时，DMA 会将该位清零： DMA 完成数据包接收 与描述符相关的缓冲区已满
30	CTXT	**接收上下文描述符 (Receive Context Descriptor)** 该位置 1 时，表示当前描述符为上下文描述符。对于上下文描述符，DMA 向该位写入 1'b1。
29:0	保留	

57.11 以太网寄存器

57.11.1 以太网寄存器映射

本节提供以下寄存器映射：

- DMA 寄存器（请参见*第 57.11.2 节：以太网 DMA 寄存器*）
- MTL 寄存器（请参见*第 57.11.3 节：以太网 MTL 寄存器*）
- MAC 寄存器，其中包括 MMC 寄存器（请参见*第 57.11.4 节：以太网 MAC 和 MMC 寄存器*）

57.11.2 以太网 DMA 寄存器

DMA 模式寄存器 (ETH_DMAMR)

DMA mode register

偏移地址：0x1000

复位值：0x0000 0000

DMA 模式寄存器为 DMA 建立总线工作模式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTM[1:0]	
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TXPR	Res.	Res.	DSPW	Res.	Res.	Res.	TAA2	TAA1	TAA0	Res.	SWR
				rw			rw				r	r	r		rw

位 31:18 保留，必须保持复位值。

位 17:16 **INTM[1:0]**：中断模式

该位字段定义以太网外设的中断模式。

中断信号和 ETH_DMACSR 寄存器中 RI/TI 位的行为随 INTM 值而变化（请参见*表 555：传送完成中断行为*）。

位 15:12 保留，必须保持复位值。

位 11 **TXPR**：发送优先级

该位置 1 时，表示在系统侧总线仲裁期间，Tx DMA 的优先级高于 Rx DMA。

位 10:9 保留，必须保持复位值。

位 8 **DSPW**：描述符已发出写入 (Descriptor Posted Write)

当此位被设置为

0：描述符写入将始终为未发出写入。

1：仅当最后一个描述符中将 IOC（完成时中断）置 1 时，描述符写入才会是未发出写入。否则，描述符写入将始终为已发出写入。

位 7:5 保留，必须保持复位值。

位 4:2 **TAA[2:0]**：发送仲裁算法 (Transmit Arbitration Algorithm)

当选择多个 Tx DMA 时，此位字段将用于为发送端选择仲裁算法。

000：固定优先级。在固定优先级中，通道 0 的优先级最低，而最后一个通道的优先级最高。

001：加权严格优先级 (WSP)

010：加权轮询 (WRR)

011 到 111：保留，不得使用。

位 1 保留，必须保持复位值。

位 0 **SWR**：软件复位

该位置 1 时，MAC 和 DMA 控制器会复位 DMA、MTL 和 MAC 的逻辑和所有内部寄存器。在所有时钟域中完成复位操作后，该位自动清零。重新编程任何寄存器之前，都应在该位中读取到一个零值。

注意：仅当所有活动时钟域中的所有复位均无效时，才会完成复位操作。因此，必须存在所有 PHY 输入时钟（适用于所选 PHY 接口）才能完成软件复位。完成软件复位操作所用的时间取决于最慢的活动时钟的频率。

系统总线模式寄存器 (ETH_DMASBMR)

System bus mode register

偏移地址：0x1004

复位值：0x0101 0000

系统总线模式寄存器控制 AXI 主设备的行为。它主要用于控制突发拆分和未完成请求的数量。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN_LPI	LPI_XIT_PKT	Res.	Res.	Res.	Res.	WR_OSR_LMT[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	RD_OSR_LMT[1:0]	
rw	rw					rw	rw							rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ONEKBBE	AAL	Res.	AALE	Res.	Res.	BLEN256	BLEN128	BLEN64	BLEN32	BLEN16	BLEN8	BLEN4	FB
		rw	rw		rw			rw	rw	rw	rw	rw	rw	rw	rw

位 31 **EN_LPI**：使能低功耗接口 (LPI) (Enable Low Power Interface (LPI))

如果置 1，则此位将使能 LPI 模式并接受来自 AXI 系统时钟控制器的 LPI 请求。

如果置 0，则此位将禁止 LPI 模式并始终拒绝来自 AXI 系统时钟控制器的 LPI 请求。

位 30 **LPI_XIT_PKT**：对魔术数据包或远程唤醒数据包解锁 (Unlock on Magic Packet or Remote wakeup Packet)

如果置 1，则此位仅会在接收到魔术数据包或远程唤醒数据包时允许 AXI 主设备退出 LPI 模式。如果置 0，则此位将允许 AXI 主设备在接收到任何数据包时退出 LPI 模式。

位 29:26 保留，必须保持复位值。

位 25:24 **WR_OSR_LMT[1:0]**：AXI 最大写入未完成请求限制 (AXI Maximum Write Outstanding Request Limit)

此值将限制 AXI 写入接口上的最大未完成请求。最大未完成请求 = WR_OSR_LMT + 1。

位 23:18 保留，必须保持复位值。

位 17:16 **RD_OSR_LMT[1:0]**：AXI 最大读取未完成请求限制 (AXI Maximum Read Outstanding Request Limit)

此值将限制 AXI 读取接口上的最大未完成请求。最大未完成请求 = RD_OSR_LMT + 1

位 15:14 保留，必须保持复位值。

位 13 **ONEKBBE**：为 AXI 主设备使能 1 KB 边界跨越 (1 Kbyte Boundary Crossing Enable for the AXI Master)

当此位置 1 时，AXI 主设备执行的突发传输不会跨越 1 KB 边界。当此位复位时，AXI 主设备执行的突发传输不会跨越 4 KB 边界。

位 12 **AAL**：地址对齐的节拍 (Address-Aligned Beats)

该位置 1 时，主设备会在读取通道和写入通道上执行地址对齐的突发传输。

位 11 保留，必须保持复位值。

位 10 **AALE**：自动 AXI LPI 使能 (Automatic AXI LPI enable)

当此位置 1 时，如果以太网外设在 ETH_DMALPIEI 寄存器的 LPIEI 位字段中编程的若干系统时钟周期内无活动，则系统将允许 AXI 主设备进入 LPI 状态。

位 9:8 保留，必须保持复位值。

位 7 **BLEN256**：AXI 突发长度 256 (AXI Burst Length 256)

该位置 1 时，AXI 主设备可在 AXI 接口上选择 256 的突发长度。

位 6 **BLEN128**：AXI 突发长度 128 (AXI Burst Length 128)
该位置 1 时，AXI 主设备可在 AXI 接口上选择 128 的突发长度。

位 5 **BLEN64**：AXI 突发长度 64 (AXI Burst Length 64)
该位置 1 时，AXI 主设备可在 AXI 接口上选择 64 的突发长度。

位 4 **BLEN32**：AXI 突发长度 32 (AXI Burst Length 32)
该位置 1 时，AXI 主设备可在 AXI 接口上选择 32 的突发长度。

位 3 **BLEN16**：AXI 突发长度 16 (AXI Burst Length 16)
该位置 1 或 FB 位置 0 时，AXI 主设备可在 AXI 接口上选择 16 的突发长度。
当 FB 位置 0 时，将此位置 1 将不起任何作用。

位 2 **BLEN8**：AXI 突发长度 8 (AXI Burst Length 8)
该位置 1 或 FB 位置 0 时，AXI 主设备可在 AXI 接口上选择 8 的突发长度。
当 FB 位置 0 时，将此位置 1 将不起任何作用。

位 1 **BLEN4**：AXI 突发长度 4 (AXI Burst Length 4)
该位置 1 或 FB 位置 0 时，AXI 主设备可在 AXI 接口上选择 4 的突发长度。
当 FB 位置 0 时，将此位置 1 将不起任何作用。

位 0 **FB**：固定突发长度 (Fixed Burst Length)
该位置 1 时，AXI 主设备将发起特定长度的突发传输（如下文所示）。
- 按照 BLEN256、BLEN128、BLEN64、BLEN32、BLEN16、BLEN8 或 BLEN4 位字段指示的固定突发长度进行突发传输
- 长度为 1 的突发传输：当该位置 0 时，AXI 主设备将发起等于或小于在位[7:1] 中编程的最大允许突发长度的突发传输。

中断状态寄存器 (ETH_DMAISR)

Interrupt status register

偏移地址：0x1008

复位值：0x0000 0000

应用程序在中断服务程序或轮询期间读取该中断状态寄存器，以确定 DMA 通道、MTL 队列和 MAC 的中断状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MACIS	MTLIS
														r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DC1IS	DC0IS
														r	r

位 31:18 保留，必须保持复位值。

位 17 **MACIS**：MAC 中断状态 (MAC Interrupt Status)
此位指示 MAC 中发生的中断事件。要将该位复位为 1'b0，软件必须读取 MAC 中的相应寄存器，以获取产生中断的具体原因，然后将中断源清除。

位 16 **MTLIS**：MTL 中断状态 (MTL Interrupt Status)
此位指示 MTL 中发生的中断事件。要将该位复位为 1'b0，软件必须读取 MTL 中的相应寄存器，以获取产生中断的具体原因，然后将中断源清除。

位 15:2 保留，必须保持复位值。

位 1 **DC1IS**：DMA 通道 1 中断状态 (DMA Channel 1 Interrupt Status)
此位指示 DMA 通道 1 中发生的中断事件。要将该位复位为 0，软件必须读取 DMA 通道 1 中的相应寄存器，以获取产生中断的具体原因，然后将中断源清除。

位 0 **DC0IS**：DMA 通道 0 中断状态 (DMA Channel 0 Interrupt Status)
此位指示 DMA 通道 0 中发生的中断事件。要将该位复位为 0，软件必须读取 DMA 通道 0 中的相应寄存器，以获取产生中断的具体原因，然后将中断源清除。

调试状态寄存器 (ETH_DMADSR)

Debug status register

偏移地址：0x100C

复位值：0x0000 0000

调试状态寄存器提供 DMA 通道 0 到 1 的接收和发送过程状态，以供调试。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TPS1[3:0]				RPS1[3:0]			
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPS0[3:0]				RPS0[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	AXRH STS	AXWH STS
r	r	r	r	r	r	r	r							r	r

位 31:24 保留，必须保持复位值。

位 23:20 **TPS1[3:0]**：DMA 通道 1 发送过程状态 (DMA Channel 1 Transmit Process State)
该位字段指示通道 1 的 Tx DMA FSM 状态。此位字段与 TPS0 位字段相似。

位 19:16 **RPS1[3:0]**：DMA 通道 1 接收过程状态 (DMA Channel 1 Receive Process State)
该位字段指示通道 1 的 Rx DMA FSM 状态。此位字段与 RPS0 位字段相似。

位 15:12 **TPS0[3:0]**：DMA 通道 0 发送过程状态 (DMA Channel 0 Transmit Process State)
该位字段指示通道 0 的 Tx DMA FSM 状态：
000：停止（发出复位或停止发送命令）
001：运行中（正在获取 Tx 传输描述符）
010：运行中（正在等待状态）
011：运行中（正在读取系统内存缓冲区中的数据并将其加入 Tx 缓冲区 (Tx FIFO) 队列）
100：时间戳写状态
101：保留供将来使用
110：挂起（Tx 描述符不可用或 Tx 缓冲区下溢）
111：运行中（正在关闭 Tx 描述符）
该位字段的 MSB 始终返回 0。此位字段不会产生中断。

位 11:8 **RPS0[3:0]**：DMA 通道 0 接收过程状态 (DMA Channel 0 Receive Process State)

该位字段指示通道 0 的 Rx DMA FSM 状态：

000：停止（发出复位或停止接收命令）

001：运行中（正在获取 Rx 传输描述符）

010：保留供将来使用

011：运行中（正在等待 Rx 数据包）

100：已挂起（Rx 描述符不可用）

101：运行中（正在关闭 Rx 描述符）

110：时间戳写状态

111：运行中（正在将接收的数据包数据从 Rx 缓冲区传输到系统内存）

该位字段的 MSB 始终返回 0。此位字段不会产生中断。

位 7:2 保留，必须保持复位值。

位 1 **AXRHSTS**：AXI 主设备读取通道状态 (AXI Master Read Channel Status)

该位为高电平时，表示 AXI 主设备的读取通道处于有效状态，且正在传输数据。

位 0 **AXWHSTS**：AXI 主设备写入通道 (AXI Master Write Channel)

该位为高电平时，表示 AXI 主设备的写入通道处于有效状态，且正在传输数据。

AXI4 发送通道 ACE 控制寄存器 (ETH_DMAA4TXACR)

AXI4 transmit channel ACE control register

偏移地址：0x1020

复位值：0x0000 0000

该寄存器用于控制对应所有发送 DMA 通道的读取事务的 AXI4 缓存一致性信号。系统将通过面向相应访问类型（描述符、缓冲区 1、缓冲区 2）编程的不同值来驱动以下 AXI4 接口信号：

- arcache_m_o[3:0]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	THC[3:0]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TEC[3:0]				Res.	Res.	Res.	Res.	TDRC[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:20 保留，必须保持复位值。

位 19:16 **THC[3:0]**：发送 DMA 第一个数据包缓冲区或 TSO 报头缓存控制 (Transmit DMA First Packet Buffer or TSO Header Cache Control)

当未使能 TSO 时，此位字段将用于在发送 DMA 访问数据包的第一个缓冲区（第一个带有 FD 的有效缓冲区设置在描述符的 TDES3 中）时驱动 arcache_o[3:0] 信号。

当使能 TSO 时，此位字段将用于在发送 DMA 访问 TSO 报头数据时驱动 arcache_o[3:0] 信号。

位 15:12 保留，必须保持复位值。

位 11:8 **TEC[3:0]**：发送 DMA 扩展数据包缓冲区或 TSO 有效负载缓存控制 (Transmit DMA Extended Packet Buffer or TSO Payload Cache Control)

当未使能 TSO 时，此位字段将用于在发送 DMA 访问扩展缓冲区时（当数据包分布在多个缓冲区中时）驱动 arcache_o[3:0] 信号。

当使能 TSO 时，此位字段将用于在发送 DMA 访问 TSO 有效负载数据时驱动 arcache_o[3:0] 信号。

位 7:4 保留，必须保持复位值。

位 3:0 **TDRC[3:0]**：发送 DMA 读取描述符缓存控制 (Transmit DMA Read Descriptor Cache Control)
当发送 DMA 引擎访问描述符时，此位字段将用于驱动 arcache_o[3:0] 信号。

AXI4 接收通道 ACE 控制寄存器 (ETH_DMAA4RXACR)

AXI4 receive channel ACE control register

偏移地址：0x1024

复位值：0x0000 0000

该寄存器用于控制对应所有接收 DMA 通道的写入事务的 AXI4 缓存一致性信号。系统将通过面向相应访问类型（描述符、缓冲区 1、缓冲区 2）编程的不同值来驱动以下 AXI4 接口信号：

- awcache_m_o[3:0]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	RDC[3:0]				Res.	Res.	Res.	Res.	RHC[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	RPC[3:0]				Res.	Res.	Res.	Res.	RDWC[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **RDC[3:0]**：接收 DMA 缓冲区缓存控制 (Receive DMA Buffer Cache Control)
当接收 DMA 在报头和有效负载未分离的情况下访问缓冲区时，此位字段将用于驱动 awcache_o[3:0] 信号。

位 23:20 保留，必须保持复位值。

位 19:16 **RHC[3:0]**：接收 DMA 报头缓存控制 (Receive DMA Header Cache Control)
当接收 DMA 在报头和有效负载分离的情况下访问报头缓冲区时，此位字段将用于驱动 awcache_o[3:0] 信号。

位 15:12 保留，必须保持复位值。

位 11:8 **RPC[3:0]**：接收 DMA 有效负载缓存控制 (Receive DMA Payload Cache Control)
当接收 DMA 在报头和有效负载分离的情况下访问有效负载缓冲区时，此位字段将用于驱动 awcache_o[3:0] 信号。

位 7:4 保留，必须保持复位值。

位 3:0 **RDWC[3:0]**：接收 DMA 写入描述符缓存控制 (Receive DMA Write Descriptor Cache Control)
当接收 DMA 访问描述符时，此位字段将用于驱动 awcache_o[3:0] 信号。

AXI4 描述符 ACE 控制寄存器 (ETH_DMAA4DACR)

AXI4 descriptor ACE control register

偏移地址：0x1028

复位值：0x0000 0000

该寄存器用于控制对应所有 TxDMA 通道的描述符写入事务和所有 RxDMA 通道的描述符读取事务的 AXI4 缓存一致性信号。该寄存器还将控制要在 awprot_m_o 和 arprot_m_o 上驱动的值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	RDRC[3:0]				Res.	Res.	TDWD[1:0]		TDWC[3:0]			
				rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:8 **RDRC[3:0]**：接收 DMA 读取描述符缓存控制 (Receive DMA Read Descriptor Cache control)
当接收 DMA 引擎读取描述符时，此位字段将用于驱动 arcache_o[3:0] 信号。

位 7:6 保留，必须保持复位值。

位 5:4 **TDWD[1:0]**：发送 DMA 写入描述符域控制 (Transmit DMA Write Descriptor Domain control)
当发送 DMA 写入描述符时，此位字段将用于驱动 awdomain_o[1:0] 信号。

位 3:0 **TDWC[3:0]**：发送 DMA 写入描述符缓存控制 (Transmit DMA Write Descriptor Cache control)
当发送 DMA 写入描述符时，此位字段将用于驱动 awcache_o[3:0] 信号。

AXI4 LPI 进入间隔寄存器 (ETH_DMALPIEI)

AXI4 LPI Entry Interval register

偏移地址：0x1040

复位值：0x0000 0000

该寄存器用于控制 AXI LPI 进入间隔。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPIEI[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **LPIEI[3:0]**：LPI 进入间隔 (LPI Entry Interval)
包含外设中的活动进入 AXI 低功耗状态需等待的系统时钟周期数乘以 64 后的数量。
0 表示 64 个时钟周期。

通道 x 控制寄存器 (ETH_DMACxCR)

Channel x control register

偏移地址：0x1100 + 0x80 * x，（x = 0 到 1）

复位值：0x0000 0000

DMA 通道 x 控制寄存器指定用于分段的 MSS 值、两个描述符之间跳过的长度，以及报头拆分和 8xPBL 模式等功能。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSL[2:0]			Res.	PBLX8
											rw	rw	rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	MSS[13:0]													
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值。

位 20:18 **DSL[2:0]**：描述符跳过长度 (Descriptor Skip Length)

该位指定两个未链接描述符之间跳过的 32 位字数。地址从当前描述符结束处开始跳到下一个描述符起始处。

DSL 值等于零时，DMA 将描述符表视为连续的。

位 17 保留，必须保持复位值。

位 16 **PBLX8**：8xPBL 模式

该位置 1 时，在*通道 x 发送控制寄存器 (ETH_DMACxTXCR)* 的位 [21:16] 以及在*通道 0 接收控制寄存器 (ETH_DMAC0RXCR)* 的位 [21:16] 中编程的 PBL 值将被乘以 8 倍。因此，DMA 根据 PBL 值以 8、16、32、64、128 和 256 个节拍传输数据。

位 15:14 保留，必须保持复位值。

位 13:0 **MSS[13:0]**：最大段大小 (Maximum Segment Size)

此位字段指定在分段数据包时应使用的最大段大小。只有*通道 x 发送控制寄存器 (ETH_DMACxTXCR)* 的 TSE 位置 1 时，该位字段才有效。

在此位字段中编程的值必须大于配置的数据宽度（以字节为单位）。建议使用 64 字节或以上的 MSS 值。

通道 x 发送控制寄存器 (ETH_DMACxTXCR)

Channel x transmit control register

偏移地址：0x1104 + 0x80 * x，（x = 0 到 1）

复位值：0x0000 0000

DMA 通道 x 发送控制寄存器控制 Tx 功能，例如 PBL、TCP 分段和 Tx 通道权重。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TQOS[3:0]				Res.	Res.	TXPBL[5:0]					
				rw	rw	rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPBL	Res.	Res.	TSE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSF	TCW[2:0]			ST
rw			rw								rw	r	r	r	rw

位 31:29 保留，必须保持复位值。

位 28 保留，必须保持复位值。

位 27:24 **TQOS[3:0]**：发送 QOS (Transmit QOS)

此位字段用于为 DMA Tx 通道 x 的所有事务驱动 arqos_m_o[3:0] 或 awqos_m_o[3:0] 输出信号。

位 23:22 保留，必须保持复位值。

位 21:16 **TXPBL[5:0]**：发送可编程突发长度 (Transmit Programmable Burst Length)

这些位指示要在一个 DMA 数据传输过程中传输的最大节拍数。这是在单个块读取或块写入操作中使用的最大值。DMA 每次在应用总线上开始突发传输时，始终尝试按 PBL 中指定的方式进行突发。可使用以下任一值来编程 PBL：1、2、4、8、16 或 32。任何其他值都会产生未定义的行为。

要传输 32 个以上节拍，请按以下步骤操作：

a) 在 ETH_DMACxCR 中设置 PBLx8 模式。

b) 将 TXPBL[5:0] 置 1。

注意：TXPBL 的最大值必须小于或等于 Tx 队列大小（*Tx 队列 x 工作模式寄存器 (ETH_MTLTXQxOMR)* 的 TQS 位字段）的一半（以节拍计）。必须确保这一点，以便在 MTL Tx 队列控制器向 MAC 传输数据时，Tx 队列中的空间至少足以存储另一个 Tx PBL 大小的数据。2048 字节大小的 Tx 队列中的总位置数为 256，TXPBL 和 8xPBL 需要编程为小于或等于 256/2。

位 15 **IPBL**：忽略 PBL 要求 (Ignore PBL Requirement)

当此位置 1 时，DMA 在启动传输之前不会检查 MTL 中的 PBL 位置数。如果没有可用空间，则 MTL 可能会使用握手来减慢 DMA 的速度。

注意：当使能多个发送 DMA 通道时，不得使用此位/模式，原因是这样可能会导致在发送队列中有空间进行当前传输之前，阻止其他发送和接收 DMA 通道访问 AXI 总线的读取数据通道。

位 14:13 保留，必须保持复位值。

位 12 **TSE**：TCP 分段使能 (TCP Segmentation Enabled)

该位置 1 时，DMA 会对通道 x 中的数据包执行 TCP 分段。仅针对在 Tx 正常描述符中将 TSE 位 (TDES0[19]) 置 1 的数据包执行 TCP 分段。该位置 1 时，TxPBL 值必须大于或等于 4。

位 11:5 保留，必须保持复位值。

位 4 **OSF**：处理第二个数据包 (Operate on Second Packet)

该位置 1 时会命令 DMA 处理发送数据的第二个数据包，即使尚未获得首个数据包的状态。

位 3:1 **TCW[2:0]**：发送通道权重 (Transmit Channel Weight)

该位字段将指示分配给相应的发送通道的权重。当复位完成后，该位字段默认会将所有通道置 0，从而使所有通道的权重相等。

位 0 **ST**：启动或停止发送命令 (Start or Stop Transmission Command)

该位置 1 时，发送过程会进入运行状态。DMA 会检查当前位置的发送列表以查找待发送的数据包。

DMA 尝试从以下任一位置获取描述符：

– 列表中的当前位置：这是 ETH_DMACxTXDLAR 寄存器设置的发送列表的基址。

– 上次停止发送时的位置

如果当前描述符不为 DMA 所有，则发送过程将进入挂起状态，并且 ETH_DMACxSR 的 TBU 位会置 1。启动发送命令只有在已停止发送时才有效。如果该命令在设置 ETH_DMACxTXDLAR 寄存器之前发出，则 DMA 行为无法预知。

该位复位时，发送过程会在完成发送当前数据包的任务之后进入停止状态，并保存发送列表中的下一个描述符位置，该位置在重启发送后会成为当前位置。如要更改列表地址，则需要在该位复位时，使用新值编程 ETH_DMACxTXDLAR 寄存器。该位再次置 1 时会采用新值。停止发送命令只有在当前数据包发送过程结束或发送过程处于挂起状态时才有效。

通道 0 接收控制寄存器 (ETH_DMAC0RXCR)

Channel 0 receive control register

偏移地址：0x1108

复位值：0x0000 0000

DMA 通道 0 接收控制寄存器控制 Rx 功能，例如 PBL、缓冲区大小和扩展状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RPF	Res.	Res.	Res.	RQOS[3:0]				Res.	Res.	RXPBL[5:0]					
rw				rw	rw	rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RBSZ[13:0]														SR
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **RPF**：DMA Rx 通道 0 数据包刷新 (DMA Rx Channel 0 Packet Flush)

当此位置 1 时，以太网外设会在发生系统总线错误后 DMA Rx 通道 0 停止的情况下，自动将数据包从以 DMA Rx 通道 0 为目标的 Rx 队列中清除。当此位保持置 1 状态且 DMA 被软件驱动程序重新启动时，系统将清除在此 RxDMA 停止时接收并存放于 Rx 队列中的数据包。重新启动 RxDMA 后由 MAC 接收到的数据包将传送到 RxDMA。在 Rx 队列的读取端进行刷新。

当此位置 0 时，如果 DMA 因系统总线错误而停止，则以太网外设不会清除以 DMA Rx 通道 0 为目标的 Rx 队列中的数据包。

这可能会导致相应 Rx 队列中出现队头阻塞。

注意：通过设置 RPF 来停止从 Rx DMA 通道到应用程序的数据包流仅会当存在 Rx 队列到 Rx DMA 通道的一对一映射时有效。在动态映射模式下，在 ETH_DMAC0RXCR 寄存器中设置 RPF 位可能会清除掉那些原本不该清除的、目标为已停止的 Rx DMA 通道的 Rx 队列中的数据包。

位 30:28 保留，必须保持复位值。

位 27:24 **RQOS[3:0]**：Rx AXI4 QOS。

此位字段用于为 DMA Rx 通道0 的所有事务驱动 arqos_m_o[3:0] 或 awqos_m_o[3:0] 输出信号。

位 23:22 保留，必须保持复位值。

位 21:16 **RXPBL[5:0]**：接收可编程突发长度 (Receive Programmable Burst Length)

这些位指示要在一个 DMA 数据传输过程中传输的最大节拍数。这是在单个块读取或块写入操作中使用的最大值。DMA 每次在应用总线上开始突发传输时，始终尝试按 PBL 中指定的方式进行突发。可使用以下任一值来编程 PBL：1、2、4、8、16 或 32。任何其他值都会产生未定义的行为。

要传输 32 个以上节拍，请按以下步骤操作：

a) 在 ETH_DMAC0CR 中设置 PBLx8 模式。

b) 将 RXPBL[5:0] 置 1。

注意：RXPBL 的最大值必须小于或等于 Rx 队列大小（Rx 队列 x 工作模式寄存器 (ETH_MTLRXQxOMR) 的 RQS 位字段）的一半（以节拍计）。必须确保这一点，以便在 MTL Rx 队列控制器向 MAC 传输数据时，Rx 队列中的空间至少足以存储另一个 Rx PBL 大小的数据。2048 字节大小的 Rx 队列中的总位置数为 256，RXPBL 和 8xPBL 需要编程为小于或等于 256/2。

位 15 保留，必须保持复位值。

位 14:1 **RBSZ[13:0]**：接收缓冲区大小 (Receive Buffer size)

该位字段指示 Rx 缓冲区的大小（以字节为单位指定）。最大缓冲区大小被限制为 16 KB。

注意：缓冲区大小必须是 4 的倍数。即使缓冲区地址指针的值未与总线宽度对齐也不例外。如果缓冲区大小不是 4 的倍数，则可能会产生未定义的行为。

LSB 位 (1:0) 会被忽略，DMA 在内部将 LSB 位视为全零值。因此，这些 LSB 位为只读 (RO)。

位 0 **SR**：启动或停止接收 (Start or Stop Receive)

该位置 1 时，DMA 尝试从接收列表中获取描述符并处理传入的数据包。

DMA 尝试从以下任一位置获取描述符：

– 列表中的当前位置：这是*通道 0 Rx 描述符列表地址寄存器 (ETH_DMAC0RXDLAR)* 设置的地址。

– 上次停止 Rx 过程时的位置

如果当前描述符不为 DMA 所有，则接收过程将处于挂起状态，并且 ETH_DMAC0SR 的 RBU 位会置 1。启动接收命令只有在已停止接收时才有效。如果该命令在设置*通道 0 Rx 描述符列表地址寄存器 (ETH_DMAC0RXDLAR)* 之前发出，则 DMA 行为无法预知。

该位复位时，Rx DMA 会在传输当前数据包之后停止操作，并保存接收列表中的下一个描述符位置，该位置在 Rx 过程重启后会成为当前位置。停止接收命令只有在 Rx 过程处于运行（等待 Rx 数据包）或挂起状态时才有效。

通道 x Tx 描述符列表地址寄存器 (ETH_DMACxTXDLAR)

Channel x Tx descriptor list address register

偏移地址：0x1114 +0x80 * x，（x = 0 到 1）

复位值：0x0000 0000

通道 x Tx 描述符列表地址寄存器会将 DMA 指向发送描述符列表的起始处。描述符列表位于应用程序的物理内存空间，且必须为字对齐。DMA 会将描述符列表的对应 LSB 位置为低电平，进而在内部将其转换为总线宽度对齐地址。

只有当 Tx DMA 停止时，即 ETH_DMACxTXCR 寄存器中的 ST 位置 0 时，才能写入该寄存器。停止时，可以用新的描述符列表地址写入该寄存器。将 ST 位置 1 时，DMA 会使用新编程的描述符基址。如果在将 ST 位置为 0 时，该寄存器未更改，则 DMA 会使用先前停止时对应的描述符地址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TDESLA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDESLA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r

位 31:0 **TDESLA[31:0]**：发送列表的起始处 (Start of Transmit List)

此位字段包含发送描述符列表中的首个描述符的基址。DMA 会忽略 32 位总线宽度的 LSB 位 (1:0)，并在内部将这些位视为全零值。因此，这些 LSB 位为只读 (RO)。

通道 0 Rx 描述符列表地址寄存器 (ETH_DMAC0RXDLAR)

Channel 0 Rx descriptor list address register

偏移地址：0x111C

复位值：0x0000 0000

通道 0 Rx 描述符列表地址寄存器会将 DMA 指向接收描述符列表的起始处。

该寄存器指向接收描述符列表的起始处。描述符列表位于应用程序的物理内存空间，且必须为字对齐。DMA 会将描述符列表的对应 LS 位置为低电平，进而在内部将其转换为总线宽度对齐地址。仅当接收停止时，才允许对该寄存器执行写入操作。停止后，必须先对该寄存器执行写入操作，然后才能发出接收启动命令。只有当 Rx DMA 停止时，即 ETH_DMAC0RXCR 寄存器中的 SR 位置 0 时，才能写入该寄存器。停止时，可以用新的描述符列表地址写入该寄存器。

将 SR 位置 1 时，DMA 会使用新编程的描述符基址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDESLA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDESLA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r

位 31:0 **RDESLA[31:0]**：接收列表的起始处 (Start of Receive List)

此位字段包含 Rx 描述符列表中的首个描述符的基址。DMA 会忽略 32 位总线宽度的 LSB 位 (1:0)，并在内部将这些位视为全零值。因此，这些 LSB 位为只读 (RO)。

通道 x Tx 描述符尾指针寄存器 (ETH_DMACxTXDTPR)

Channel x Tx descriptor tail pointer register

偏移地址：0x1120 + 0x80 * x（x = 0 到 1）

复位值：0x0000 0000

通道 x Tx 描述符尾指针寄存器指向与基址的偏移，并指示最后一个有效描述符的位置。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TDT[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r

位 31:0 **TDT[31:0]**：发送描述符尾指针 (Transmit Descriptor Tail Pointer)

该位字段包含 Tx 描述符环的尾指针。软件对尾指针进行写入操作，以向 Tx 通道添加更多描述符。硬件尝试在头指针和尾指针寄存器之间发送由描述符引用的所有数据包。

通道 0 Rx 描述符尾指针寄存器 (ETH_DMAC0RXDTPR)

Channel 0 Rx descriptor tail pointer register

偏移地址：0x1128

复位值：0x0000 0000

通道 0 Rx 描述符尾指针指向与基址的偏移，并指示最后一个有效描述符的位置。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDT[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r

位 31:0 **RDT[31:0]**：接收描述符尾指针 (Receive Descriptor Tail Pointer)

该位字段包含 Rx 描述符环的尾指针。软件对尾指针进行写入操作，以向 Rx 通道添加更多描述符。硬件尝试在头指针和尾指针寄存器之间将接收到的所有数据包写入引用的描述符。

通道 x Tx 描述符环长度寄存器 (ETH_DMACxTXRLR)

Channel x Tx descriptor ring length register

偏移地址：0x112C + 0x80 * x，（x = 0 到 1）

复位值：0x0000 0000

Tx 描述符环长度寄存器包含发送描述符环的长度。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TDRL[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:10 保留，必须保持复位值。

位 9:0 **TDRL[9:0]**：发送描述符环长度 (Transmit Descriptor Ring Length)

该位字段设置循环描述符环中 Tx 描述符的最大数量。描述符的最大数量限制为 1K 个描述符。建议将最小描述符环长度设为 4。

例如，用户可将此位字段编程为任何不大于 0x3FF 的值。此位字段为 10 位宽，如果用户编程 0x3FF，则将得到 1024 个描述符。如果用户希望拥有 10 个描述符，则可将其编程为值 0x9。

通道 0 Rx 描述符环长度寄存器 (ETH_DMAC0RXRLR)

Channel 0 Rx descriptor ring length register

偏移地址：0x1130

复位值：0x0000 0000

通道 0 Rx 描述符环长度寄存器包含循环接收描述符环的长度。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARBS[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	RDRL[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **ARBS[7:0]**：交替接收缓冲区大小 (Alternate Receive Buffer Size)

用于指示当 ARBS[7:0] 编程为非零值时，缓冲区 1 的大小（以字节为单位）。

当 ARBS[7:0] = 0 时，Rx 缓冲区 1 和 Rx 缓冲区 2 的大小将基于 *通道 0 接收控制寄存器 (ETH_DMAC0RXCR)* 的 RBSZ[13:0] 位字段。

位 15:10 保留，必须保持复位值。

位 9:0 **RDRL[9:0]**：接收描述符环长度 (Receive Descriptor Ring Length)

该寄存器设置循环描述符环中 Rx 描述符的最大数量。描述符的最大数量限制为 1K 个描述符。例如，用户可将此位字段编程为任何不大于 0x3FF 的值。此位字段为 10 位宽。如果用户编程 0x3FF，则将得到 1024 个描述符。如果用户希望拥有 10 个描述符，则可将其编程为值 0x9。

通道 x 中断使能寄存器 (ETH_DMACxIER)

Channel x interrupt enable register

偏移地址：0x1134 + 0x80 * x，（x = 0 到 1）

复位值：0x0000 0000

通道 x 中断使能寄存器可使能状态寄存器报告的中断。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NIE	AIE	CDEE	FBEE	ERIE	ETIE	RWTE	RSE	RBUE	RIE	Res.	Res.	Res.	TBUE	TXSE	TIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **NIE**：正常中断汇总使能 (Normal Interrupt Summary Enable)
该位置 1 时，会使能正常中断汇总。该位可使能*通道 x 状态寄存器 (ETH_DMACxSR)* 中的以下中断：
位 0：发送中断
位 2：发送缓冲区不可用
位 6：接收中断
位 11：提前接收中断
当此位复位时，将禁止正常中断汇总。

位 14 **AIE**：异常中断汇总使能 (Abnormal Interrupt Summary Enable)
该位置 1 时，会使能异常中断汇总。该位可使能*通道 x 状态寄存器 (ETH_DMACxSR)* 中的以下中断：
位 1：发送过程停止
位 7：Rx 缓冲区不可用
位 8：接收过程停止
位 9：接收看门狗超时
位 10：提前发送中断
位 12：致命总线错误
当此位复位时，将禁止异常中断汇总。

位 13 **CDEE**：上下文描述符错误使能 (Context Descriptor Error Enable)
该位与 AIE 位同时置 1 时，会使能上下文描述符错误中断。该位复位时，会禁止上下文描述符错误中断。

位 12 **FBEE**：致命总线错误使能 (Fatal Bus Error Enable)
该位与 AIE 位同时置 1 时，会使能致命总线错误中断。该位复位时，会禁止致命总线错误中断。

位 11 **ERIE**：提前接收中断使能 (Early Receive Interrupt Enable)
该位与 NIE 位同时置 1 时，会使能提前接收中断。该位复位时，会禁止提前接收中断。

位 10 **ETIE**：提前发送中断使能 (Early Transmit Interrupt Enable)
该位与 AIE 位同时置 1 时，会使能提前发送中断。该位复位时，会禁止提前发送中断。

位 9 **RWTE**：接收看门狗超时使能 (Receive Watchdog Timeout Enable)
该位与 AIE 位同时置 1 时，会使能接收看门狗超时中断。该位复位时，会禁止接收看门狗超时中断。

位 8 **RSE**：接收停止使能 (Receive Stopped Enable)
该位与 AIE 位同时置 1 时，会使能接收停止中断。该位复位时，会禁止接收停止中断。

位 7 **RBUE**：接收缓冲区不可用使能 (Receive Buffer Unavailable Enable)
该位与 AIE 位同时置 1 时，会使能接收缓冲区不可用中断。该位复位时，会禁止接收缓冲区不可用中断。

位 6 **RIE**：接收中断使能 (Receive Interrupt Enable)
该位与 NIE 位同时置 1 时，会使能接收中断。该位复位时，会禁止接收中断。

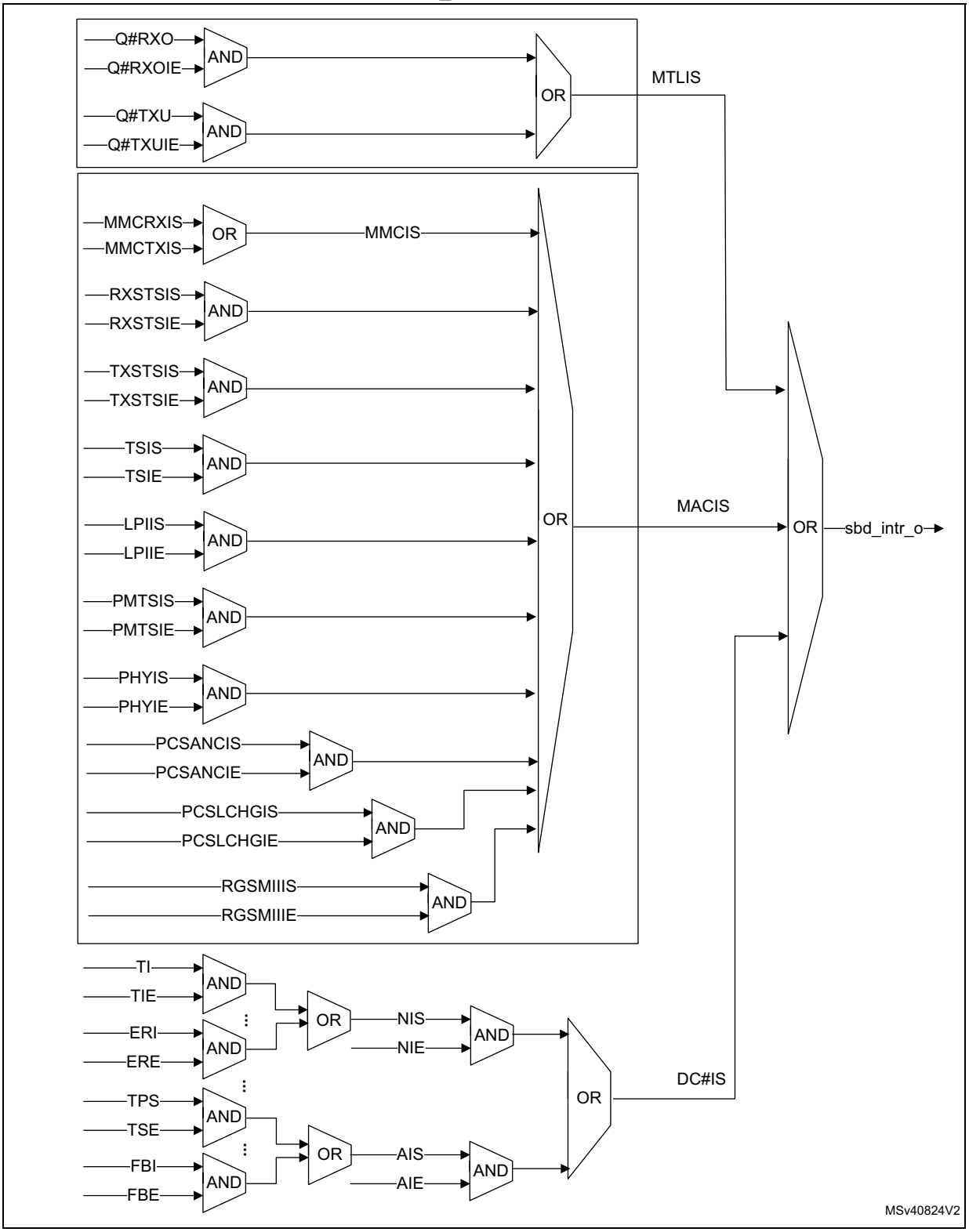
位 5:3 保留，必须保持复位值。

位 2 **TBUE**：发送缓冲区不可用使能 (Transmit Buffer Unavailable Enable)
该位与 NIE 位同时置 1 时，会使能发送缓冲区不可用中断。该位复位时，会禁止发送缓冲区不可用中断。

位 1 **TXSE**：发送停止使能 (Transmit Stopped Enable)
该位与 AIE 位同时置 1 时，会使能发送停止中断。该位复位时，会禁止发送停止中断。

位 0 **TIE**：发送中断使能 (Transmit Interrupt Enable)
该位与 NIE 位同时置 1 时，会使能发送中断。该位复位时，会禁止发送中断。

图 750. ETH_DMAISR 标志的生成



通道 0 Rx 中断看门狗定时器寄存器 (ETH_DMAC0RXIWTR)

Channel 0 Rx interrupt watchdog timer register

偏移地址：0x1138

复位值：0x0000 0000

接收中断看门狗定时器寄存器指示来自 DMA 的接收中断 (RI) 的看门狗超时。向该寄存器中写入非零值时，会针对*通道 x 状态寄存器 (ETH_DMACxSR)* 的 RI 位使能看门狗定时器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWTU[1:0]	
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:18 保留，必须保持复位值。

位 17:16 **RWTU[1:0]**：接收中断看门狗定时器计数单位 (Receive Interrupt Watchdog Timer Count Units)

此位字段指示 RWT[7:0] 位字段中的一个单位所对应的系统时钟周期数。

00：256

01：512

10：1024

11：2048

例如，当 RWT[7:0] = 2 且 RWTU[1:0] = 1 时，看门狗定时器将设置为 2 * 512 = 1024 个系统时钟周期。

位 15:8 保留，必须保持复位值。

位 7:0 **RWT[7:0]**：接收中断看门狗定时器计数 (Receive Interrupt Watchdog Timer Count)

该位字段指示系统时钟周期数乘以 RWTU 位字段中指示的系数之后的时间值，即看门狗定时器的设定时间值。

看门狗定时器会在 Rx DMA 结束数据包传输之后由编程设定的值触发，但 ETH_DMAC0SR 中的 RI 位未置 1，因为相应描述符中的中断使能位 RDES3[30] 已置 1。

当看门狗定时器计时结束时，RI 位置 1 且定时器停止。由于 RI 根据每个接收数据包的中断使能位 RDES3[30] 进行自动设置而使 RI 位设为高电平时，看门狗定时器复位。

通道 x 当前应用程序发送描述符寄存器 (ETH_DMACxCATXDR)

Channel x current application transmit descriptor register

偏移地址：0x1144 + 0x80 * x，（x = 0 到 1）

复位值：0x0000 0000

通道 x 当前应用程序发送描述符寄存器指向 DMA 所读取的当前发送描述符。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURTDESAPTR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURTDESAPTR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CURTDESAPTR[31:0]**：应用程序发送描述符地址指针 (Application Transmit Descriptor Address Pointer)

在 Tx 操作期间，DMA 会更新该指针。该指针在复位时清零。

通道 0 当前应用程序接收描述符寄存器 (ETH_DMAC0CARXDR)

Channel 0 current application receive descriptor register

偏移地址：0x114C

复位值：0x0000 0000

通道 0 当前应用程序接收描述符寄存器指向 DMA 所读取的当前接收描述符。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURRDESAPTR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRDESAPTR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CURRDESAPTR[31:0]**：应用程序接收描述符地址指针 (Application Receive Descriptor Address Pointer)

在 Rx 操作期间，DMA 会更新该指针。该指针在复位时清零。

通道 x 当前应用程序发送缓冲区寄存器 (ETH_DMACxCATXBR)

Channel x current application transmit buffer register

偏移地址：0x1154 + 0x80 * x，（x = 0 到 1）

复位值：0x0000 0000

通道 x 当前应用程序发送缓冲区地址寄存器指向 DMA 所读取的当前 Tx 缓冲区地址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURTBUFAPTR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURTBUFAPTR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CURTBUFAPTR[31:0]**：应用程序发送缓冲区地址指针 (Application Transmit Buffer Address Pointer)

在 Tx 操作期间，DMA 会更新该指针。该指针在复位时清零。

通道 0 当前应用程序接收缓冲区寄存器 (ETH_DMAC0CARXBR)

Channel 0 current application receive buffer register

偏移地址：0x115C

复位值：0x0000 0000

通道 0 当前应用程序接收缓冲区地址寄存器指向 DMA 所读取的当前 Rx 缓冲区地址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURRBUFAPTR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRBUFAPTR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CURRBUFAPTR[31:0]**：应用程序接收缓冲区地址指针 (Application Receive Buffer Address Pointer)

在 Rx 操作期间，DMA 会更新该指针。该指针在复位时清零。

通道 x 状态寄存器 (ETH_DMACxSR)

Channel x status register

偏移地址：0x1160 + 0x80 * x，（x = 0 到 1）

复位值：0x0000 0000

软件驱动程序（应用程序）在中断服务程序或轮询期间读取状态寄存器，以确定 DMA 的状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REB[2:0]			TEB[2:0]		
										r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Res.	Res.	Res.	TBU	TPS	TI
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rw	rc_w1	rc_w1	rc_w1				rc_w1	rc_w1	rc_w1

位 31:22 保留，必须保持复位值。

位 21:19 **REB[2:0]**：Rx DMA 错误位 (Rx DMA Error Bits)

该位字段指示导致总线错误的错误类型。例如，AXI 接口上的错误响应。
位 [2]：当此位置 1 时，Rx DMA 传输数据时出错；当此位置 0 时，Rx DMA 传输数据时不出错。
位 [1]：当此位置 1 时，访问描述符时出错；当此位置 0 时，访问数据缓冲区时出错
位 [0]：当此位置 1 时，读取传输时出错；当此位置 0 时，写入传输时出错
只有在 FBE 位置 1 时，该位字段才有效。此位字段不会产生中断。

位 18:16 **TEB[2:0]**：Tx DMA 错误位 (Tx DMA Error Bits)

该位字段指示导致总线错误的错误类型。例如，AXI 接口上的错误响应。
位 [2]：当此位置 1 时，Tx DMA 传输数据时出错；当此位置 0 时，Tx DMA 传输数据时不出错。
位 [1]：当此位置 1 时，访问描述符时出错；当此位置 0 时，访问数据缓冲区时出错
位 [0]：当此位置 1 时，读取传输时出错；当此位置 0 时，写入传输时出错
只有在 FBE 位置 1 时，该位字段才有效。此位字段不会产生中断。

位 15 **NIS**：正常中断汇总 (Normal Interrupt Summary)

当使能 ETH_DMACxIER 寄存器中对应的中断位时，正常中断汇总位的值是以下位的逻辑或运算结果：
位 0：发送中断
位 2：发送缓冲区不可用
位 6：接收中断
位 11：提前接收中断
只有未屏蔽位（在 ETH_DMACxIER 寄存器中将相应中断使能置 1 的中断）会影响正常中断汇总位。
它是黏着位。每当导致 NIS 置 1 的对应位被清零时，必须同时将该位也清零（通过向此位写入 1）。

位 14 **AIS**：异常中断汇总 (Abnormal Interrupt Summary)

当使能 ETH_DMACxIER 寄存器中对应的中断位时，异常中断汇总位的值是以下位的逻辑或运算结果：
位 1：发送过程停止
位 7：接收缓冲区不可用
位 8：接收过程停止
位 10：提前发送中断
位 12：致命总线错误
位 13：上下文描述符错误 (Context Descriptor Error)
只有未屏蔽的位会影响异常中断汇总位。
它是黏着位。每当导致 AIS 置 1 的对应位被清零时，必须同时将该位也清零（通过向此位写入 1）。

位 13 **CDE**：上下文描述符错误 (Context Descriptor Error)

此位指示 DMA Tx/Rx 引擎接收到一个描述符错误，表示数据包流中间存在无效上下文（中间描述符）或在 Tx 端存在全 1 的描述符，而在 Rx 端则表示 DMA 读取了一个描述符，其任一缓冲区地址为全 1（系统将此视为无效）。

位 12 **FBE**：致命总线错误
此位指示发生了总线错误（如 EB 位字段中所述）。此位置 1 后，对应的 DMA 通道引擎会禁止所有总线访问。

位 11 **ERI**：提前接收中断 (Early Receive Interrupt)
此位指示 DMA 已填满数据包的首个数据缓冲区。该寄存器的 RI 位自动清零此位。

位 10 **ETI**：提前发送中断
此位指示要发送的数据包已完全传输到 MTL Tx FIFO。

位 9 **RWT**：接收看门狗超时 (Receive Watchdog Timeout)
当接收到长度大于 2,048 字节（如果使能巨型数据包模式，则为 10,240 字节）的数据包时，该位将置 1。

位 8 **RPS**：接收过程停止 (Receive Process Stopped)
当 Rx 过程进入停止状态时，此位将置 1。

位 7 **RBU**：接收缓冲区不可用 (Receive Buffer Unavailable)
此位指示应用程序拥有接收列表中的下一个描述符，DMA 无法获取。Rx 过程进入挂起状态。要恢复处理 Rx 描述符，应用程序应更改描述符的拥有关系，然后发出接收轮询要求命令。如果未发出该命令，则当接收到下一个识别的传入数据包时，Rx 过程会恢复。在环形模式下，应用程序应将通道的接收描述符尾指针寄存器前移。只有当 DMA 拥有前一 Rx 描述符时，该位才会置 1。

位 6 **RI**：接收中断
此位指示数据包接收过程已完成。数据包接收过程完成时，最后一个描述符中的 RDES1 的位 31 复位，并且会在该描述符中更新特定数据包状态信息。
接收保持运行状态。

位 5:3 保留，必须保持复位值。

位 2 **TBU**：发送缓冲区不可用
此位指示应用程序拥有发送列表中的下一个描述符，DMA 无法获取。发送会进入挂起状态。*调试状态寄存器 (ETH_DMADSR)* 寄存器的 TPSi 位字段对发送过程状态转换进行说明。
要恢复处理发送描述符，应用程序应进行以下操作：
1. 通过将 TDES3 的位 31 置 1 来更改描述符的拥有关系。
2. 发出发送轮询要求命令。
对于环形模式，应用程序应将通道的发送描述符尾指针寄存器前移。

位 1 **TPS**：发送过程停止
当发送停止时，此位置 1。

位 0 **TI**：发送中断
此位指示数据包发送过程已完成。发送过程完成时，最后一个描述符中的 TDES3 的位 31 复位，并且会在该描述符中更新特定数据包状态信息。

通道丢失帧计数寄存器 (ETH_DMACMFCR)

Channel missed frame count register

偏移地址：0x116C

复位值：0x0000 0000

该寄存器包含 DMA 因总线错误或因编程*通道 0 接收控制寄存器 (ETH_DMAC0RXCR)* 寄存器中的 RPF 位字段而丢弃的数据包计数器计数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MFCO	Res.	Res.	Res.	Res.	MFC[10:0]										
rc_r					rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r

位 31:16 保留，必须保持复位值。

位 15 **MFCO**：MFC 计数器的上溢状态 (Overflow status of the MFC Counter)
该位置 1 时，MFC 计数器不会进一步递增。对此寄存器执行读取操作时此位清零。

位 14:11 保留，必须保持复位值。

位 10:0 **MFC[10:0]**：丢弃数据包计数器 (Dropped Packet Counters)
该计数器指示 DMA 因总线错误或因编程*通道 0 接收控制寄存器 (ETH_DMAC0RXCR)* 中的 RPF 位字段而丢弃的数据包计数器计数。对此寄存器执行读取操作时计数器清零。

以太网 DMA 寄存器映射和复位值

表 580. ETH_DMA 通用寄存器映射和复位值

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1000	ETH_DMAMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTM[1:0]		Res.	Res.	Res.	Res.	TXPR	Res.	Res.	DSPW	Res.	Res.	Res.	TAA[2:0]			Res.	SWR
	Reset value															0	0					0			0				0	0	0		0
0x1004	ETH_DMASBMR	EN_LPI	LPI_XIT_PKT	Res.	Res.	Res.	Res.	WR_OSR_LMT[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	RD_OSR_LMT[1:0]		Res.	Res.	ONEKBBE	AAL	Res.	AALE	Res.	Res.	BLEN256	BLEN128	BLEN64	BLEN32	BLEN16	BLEN8	BLEN4	FB
	Reset value	0	0					0	1							0	1			0	0		0			0	0	0	0	0	0	0	0
0x1008	ETH_DMAISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MACIS	MTLIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DC1IS	DC0IS
	Reset value															0	0															0	0
0x100C	ETH_DMADSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TPS1[3:0]				RPS1[3:0]				TPS0[3:0]				RPS0[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	AXRHSTS	AXWHSTS
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							0	0
0x1010 - 0x101C	Reserved																																
0x1020	ETH_ DMAA4TXACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	THC[3:0]				Res.	Res.	Res.	Res.	TEC[3:0]				Res.	Res.	Res.	Res.	TDRC[3:0]			
	Reset value													0	0	0	0					0	0	0	0					0	0	0	0
0x1024	ETH_ DMAA4RXACR	Res.	Res.	Res.	Res.	RDC[3:0]				Res.	Res.	Res.	Res.	RHC[3:0]				Res.	Res.	Res.	Res.	RPC[3:0]				Res.	Res.	Res.	Res.	RDWC[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0
0x1028	ETH_ DMAA4DACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDRC[3:0]				Res.	Res.	TDWD[1:0]		TDWC[3:0]			
	Reset value																					0	0	0	0			0	0	0	0	0	0
0x102C - 0x103C	Reserved																																
0x1040	ETH_DMALPIEI	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPIEI[3:0]			
	Reset value																													0	0	0	0

表 581. ETH_DMA_CH0 寄存器映射和复位值

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1100	ETH_DMAC0CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSL[2:0]			Res.	PBLX8	Res.	Res.	MSS[13:0]													
	Reset value												0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1104	ETH_ DMAC0TXCR	Res.	Res.	Res.	Res.	TQOS[3:0]				Res.	Res.	TXPBL[5:0]						IPBL	Res.	Res.	TSE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSF	TCW[2:0]			ST
	Reset value					0	0	0	0			0	0	0	0	0	0	0			0								0	0	0	0	0
0x1108	ETH_ DMAC0RXCR	RPF	Res.	Res.	Res.	RQOS[3:0				Res.	Res.	RXPBL[5:0]						Res.	RBSZ[13:0]											Res.	Res.	Res.	SR
	Reset value	0				0	0	0	0			0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0				0
0x110C - 01110	Reserved																																
0x1114	ETH_ DMAC0TXDLAR	TDESLA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1118	Reserved																																
0x111C	ETH_ DMAC0RXDLAR	RDESLA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1120	ETH_ DMAC0TXDTPR	TDT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1124	Reserved																																
0x1128	ETH_ DMAC0RXDTPR	RDT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x112C	ETH_ DMAC0TXRLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDRL[9:0]									
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x1130	ETH_ DMAC0RXRLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARBS[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	RDRL[9:0]									
	Reset value									0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x1134	ETH_DMAC0IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NIE	AIE	CDEE	FBEE	ERIE	ETIE	RWTE	RSE	RBUE	RIE	Res.	Res.	Res.	TBUE	TXSE	TIE
	Reset value																	0	0	0	0	0	0	0	0	0	0				0	0	0
0x1138	ETH_ DMAC0RXIWTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWTU[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWT[7:0]							
	Reset value															0	0									0	0	0	0	0	0	0	0

表 581. ETH_DMA_CH0 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x113C-0x1140	Reserved																																
0x1144	ETH_DMAC0CATXDR	CURTDESAPTR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1148	Reserved																																
0x0114C	ETH_DMAC0CARXDR	CURRDESAPTR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1150	Reserved																																
0x1154	ETH_DMAC0CATXBR	CURTBUFAPTR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1158	Reserved																																
0x115C	ETH_DMAC0CARXBR	CURRBUFAPTR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1160	ETH_DMAC0SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REB[2:0]			TEB[2:0]			NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Res.	Res.	Res.	TBU	TPS	TI
	Reset value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0
0x1164 - 0x1168	Reserved																																
0x116C	ETH_DMACMFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MFCO	Res.	Res.	Res.	Res.	MFC[10:0]										
	Reset value																	0					0	0	0	0	0	0	0	0	0	0	0

表 582. ETH_DMA_CH1 寄存器映射和复位值

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1180	ETH_DMAC1CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSL[2:0]			Res.	PBLX8	Res.	Res.	MSS[13:0]													
	Reset value												0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1184	ETH_DMAC1TXCR	Res.	Res.	Res.	TQOS[3:0]				Res.	Res.	TXPBL[5:0]						IPBL	Res.	Res.	TSE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSF	TCW[2:0]			ST	
	Reset value				0	0	0	0			0	0	0	0	0	0	0	0			0								0	0	0	0	0
0x1188 - 0x1190	Reserved																																

表 582. ETH_DMA_CH1 寄存器映射和复位值（续）

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1194	ETH_DMAC1TXDLAR	TDESLA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1198 - 0x119C	Reserved																																
0x11A0	ETH_DMAC1TXDTPR	TDT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x11A4 - 0x11A8	Reserved																																
0x11AC	ETH_DMAC1TXRLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDRL[9:0]									
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x11B0	Reserved																																
0x11B4	ETH_DMAC1IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NIE	AIE	CDEE	FBEE	ERIE	ETIE	RWTE	RSE	RBUE	RIE	Res.	Res.	Res.	TBUE	TXSE	TIE
	Reset value																	0	0	0	0	0	0	0	0	0	0				0	0	0
0x11B8	Reserved																																
0x11BC-0x11C0	Reserved																																
0x11C4	ETH_DMAC1CATXDR	CURTDESAPTR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x11C8 - 0x11D0	Reserved																																
0x11D4	ETH_DMAC1CATXBR	CURTBUFAPTR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
011D8 - 0x11DC	Reserved																																
0x11E0	ETH_DMAC1SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REB[2:0]			TEB[2:0]			NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Res.	Res.	Res.	TBU	TPS	TI
	Reset value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0
0x11E4-0x11E8	Reserved																																

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

57.11.3 以太网 MTL 寄存器

工作模式寄存器 (ETH_MTLOMR)

Operating mode Register

偏移地址：0x0C00

复位值：0x0000 0000

工作模式寄存器将建立发送和接收工作模式及命令。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CNT CLR	CNT PRST	Res.	SCHALG[1:0]		Res.	Res.	RAA	DTX STS	Res.
						rw	rw		rw	rw			rw	rw	

位 31:10 保留，必须保持复位值。

位 9 **CNTCLR**：计数器复位 (Counters Reset)
该位置 1 时，所有计数器都将复位。该位在 1 个时钟周期后自动清零。
如果该位和 CNTPRST 位同时置 1，则 CNTPRST 优先级较高。

位 8 **CNTPRST**：计数器预设 (Counters Preset)
当此位被设置为：
- *Tx 队列 x 下溢寄存器 (ETH_MTLTXQxUR)* 被初始化/预设为 0x7F0。
- *Rx 队列 x 丢失数据包和上溢计数器寄存器 (ETH_MTLRXQxMPOCR)* 中丢失的数据包和上溢数据包计数器被初始化/预设为 0x7F0

该位将自动清零。

位 7 保留，必须保持复位值。

位 6:5 **SCHALG[1:0]**：Tx 调度算法 (Tx Scheduling Algorithm)
该位字段指示 Tx 调度的算法：
00：加权轮询 (WRR) 算法
01 到 10：保留，不得使用。
11：严格优先级 (SP) 算法。

位 4:3 保留，必须保持复位值。

位 2 **RAA**：接收仲裁算法 (Receive Arbitration Algorithm)
该位字段用于选择 Rx 端的仲裁算法。
0：严格优先级 (SP)。队列 0 的优先级最低，而最后一个队列的优先级最高。
1：加权严格优先级 (WSP)

位 1 **DTXSTS**：丢弃发送状态 (Drop Transmit Status)
当此位置 1 时，系统会将从 MAC 接收到的 Tx 数据包状态丢弃到 MTL 中。当此位复位时，系统会将从 MAC 接收到的 Tx 数据包状态转发到应用程序。

位 0 保留，必须保持复位值。

中断状态寄存器 (ETH_MTLISR)

Interrupt status Register

偏移地址：0x0C20

复位值：0x0000 0000

软件驱动程序（应用程序）在中断服务程序或轮询期间读取该寄存器，以确定 MTL 队列和 MAC 的中断状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Q1IS	Q0IS
														r	r

位 31:2 保留，必须保持复位值。

位 1 **Q1IS**：队列 1 中断状态 (Queue 1 interrupt status)
此位指示队列 1 已生成中断。要复位该位，请读取 ETH_MTLQ1ICSR 寄存器，以识别中断原因并清除中断源。

位 0 **Q0IS**：队列 0 中断状态 (Queue 0 interrupt status)
此位指示队列 0 已生成中断。要复位该位，请读取 ETH_MTLQ0ICSR 寄存器，以识别中断原因并清除中断源。

Tx 队列 x 工作模式寄存器 (ETH_MTLTXQxOMR)

Tx queue x operating mode Register

偏移地址：0x0D00 + 0x40 * x，（x = 0 到 1）

复位值：0x0000 0000

队列 x 发送工作模式寄存器将建立发送队列工作模式及命令。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TQS[2:0]		
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TTC[2:0]			TXQEN[1:0]		TSF	FTQ
									rw	rw	rw	rw	rw	rw	rw

位 31:19 保留，必须保持复位值。

位 18:16 **TQS[2:0]**：发送队列大小 (Transmit queue size)
该位字段指示分配的发送队列大小（以 256 字节块计）。
队列大小范围为 256 字节 (TQS=0b000) 到 2048 字节 (TQS=0b111)。

位 15:7 保留，必须保持复位值。

位 6:4 **TTC[2:0]**：发送阈值控制 (Transmit Threshold Control)

这些位控制 MTL Tx 队列的阈值级别。当 MTL Tx 队列中的数据包大小大于阈值时启动发送。此外，还会发送长度小于阈值的完整数据包。这些位只有在 TSF 位复位后才能使用。

000：32
001：64
010：96
011：128
100：192
101：256
110：384
111：512

位 3:2 **TXQEN[1:0]**：发送队列使能 (Transmit Queue Enable)

此位字段用于使能/禁止发送队列 0。

00：未使能
10：使能
其他值：保留，不得使用。

注意：在多个 Tx 队列配置中，默认情况下所有队列均处于禁用状态。通过编程此位字段使能 Tx 队列。

位 1 **TSF**：发送存储并转发 (Transmit Store and Forward)

当此位置 1 时，如果 MTL Tx 队列中有一个完整数据包，则会启动发送。当此位置 1 时，会忽略在该寄存器的位 [6:4] 中指定的 TTC 值。该位只有在已停止发送时才能更改。

位 0 **FTQ**：刷新发送队列 (Flush Transmit Queue)

该位置 1 时，Tx 队列控制器逻辑被复位为其默认值。因此，Tx 队列中的所有数据都将丢失或被刷新。刷新操作结束时该位在内部复位。在该位复位之前，不应对 ETH_MTLTXQ1OMR 寄存器进行写入操作。MAC 发送器已接受的数据不会被刷新。其会被安排进行发送，并且会导致下溢且发送的数据包过短。

注意：仅当 Tx 队列为空并且应用程序已接受所有已发送数据包的挂起 Tx 状态时，才会完成刷新操作。要完成此刷新操作，PHY Tx 时钟 (eth_mii_tx_clk) 应处于活动状态。

Tx 队列 x 下溢寄存器 (ETH_MTLTXQxUR)

Tx queue x underflow register

偏移地址：0x0D04 + 0x40 * x，（x = 0 到 1）

复位值：0x0000 0000

队列 x 下溢计数器寄存器包含针对以下数据包的计数器：因发送队列下溢而被中止的数据包，以及因接收队列数据包刷新而丢失的数据包

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UFCNT OVF	UFFRMCNT[10:0]										
				rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r

位 31:12 保留，必须保持复位值。

位 11 **UFCNTOVF**：下溢数据包计数器的上溢位 (Overflow Bit for Underflow Packet Counter)
每次 Tx 队列下溢数据包计数器位字段上溢（即，已超出最大计数）时，该位都会置 1。在这种情况下，上溢数据包计数器将复位为全零值，该位指示发生翻转。

位 10:0 **UFFRMCNT[10:0]**：下溢数据包计数器 (Underflow Packet Counter)
此位字段指示控制器因 Tx 队列下溢而中止的数据包数。每次 MAC 因下溢而中止传出的数据包时，该计数器都会递增。对此寄存器执行读取操作时计数器清零。

Tx 队列 x 调试寄存器 (ETH_MTLTXQxDR)

Tx queue x debug register

偏移地址：0x0D08 + 0x40 * x，（x = 0 到 1）

复位值：0x0000 0000

队列 x 发送调试寄存器提供与发送队列相关的各个块的调试状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STXSTSF[2:0]			Res.	PTXQ[2:0]		
									r	r	r		r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXSTSFSTS	TXQSTS	TWCSTS	TRCSTS[1:0]		TXQPAUSED
										r	r	r	r	r	r

位 31:23 保留，必须保持复位值。

位 22:20 **STXSTSF[2:0]**：队列的 Tx 状态 FIFO 中的状态字数 (Number of Status Words in Tx Status FIFO of Queue)
此位字段指示该队列的 Tx 状态 FIFO 中的当前状态数。
当 ETH_MTLOMR 寄存器的 DTXSTS 位置 1 时，该位字段不反映 Tx 状态 FIFO 中的状态字数。

位 19 保留，必须保持复位值。

位 18:16 **PTXQ[2:0]**：发送队列中的数据包数 (Number of Packets in the Transmit Queue)
此位字段指示 Tx 队列中的当前数据包数。
当*工作模式寄存器 (ETH_MTLOMR)* 寄存器的 DTXSTS 位置 1 时，该位字段不反映发送队列中的数据包数。

位 15:6 保留，必须保持复位值。

位 5 **TXSTSFSTS**：MTL Tx 状态 FIFO 已满状态 (MTL Tx Status FIFO Full Status)
该位为高电平时，表示 MTL Tx 状态 FIFO 已满。因此，MTL 不能接受发送更多数据包。

位 4 **TXQSTS**：MTL Tx 队列非空状态 (MTL Tx Queue Not Empty Status)
该位为高电平时，表示 MTL Tx 队列不为空，还留有一些数据要发送。

位 3 **TWCSTS**：MTL Tx 队列写控制器状态 (MTL Tx Queue Write Controller Status)
该位为高电平时，表示 MTL Tx 队列写控制器有效且正在向 Tx 队列传输数据。

位 2:1 **TRCSTS[1:0]**：MTL Tx 队列读控制器状态 (MTL Tx Queue Read Controller Status)
此位字段指示 Tx 队列读控制器的状态：
00：空闲状态 (Idle state)
01：读取状态（将数据传输到 MAC 发送器）
10：等待来自 MAC 发送器的挂起 Tx 状态
11：因来自 MAC 的数据包中止请求而刷新 Tx 队列

位 0 **TXQPAUSED**：发送队列处于暂停状态 (Transmit Queue in Pause)
该位为高电平且使能 Rx 流控制时，表示 Tx 队列出于以下原因而处于暂停状态（仅在全双工模式下）：
– 当使能 PFC 时，接收到优先级分配给 Tx 队列的 PFC 数据包
– 当禁止 PFC 时，接收到 802.3x 暂停数据包

Tx 队列 x ETS 状态寄存器 (ETH_MTLTXQxESR)

Tx queue x ETS status Register

偏移地址：0x0D14 + 0x40 * x，（x = 0 到 1）

复位值：0x0000 0000

队列 x ETS 状态寄存器将提供队列 x 中发送的平均数据传输量。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ABS[23:16]							
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 保留，必须保持复位值。

位 23:0 **ABS[23:0]**：每时隙平均位数 (Average Bits per Slot)
该位字段包含每时隙发送的平均位数。

队列 x 中断控制状态寄存器 (ETH_MTLQxICSR)

Queue x interrupt control status Register

偏移地址：0x0D2C + 0x40 * x，（x = 0 到 1）

复位值：0x0000 0000

该寄存器包含队列 x 中断的中断使能和状态位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXOIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXOVFIS
							rw								rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	ABPSIE	TXUIE	Res.	Res.	Res.	Res.	Res.	Res.	ABPSIS	TXUNFIS
						rw	rw							rw	rc_w1

位 31:25 保留，必须保持复位值。

位 24 **RXOIE**：接收队列上溢中断使能 (Receive Queue Overflow Interrupt Enable)
该位置 1 时，会使能接收队列上溢中断。该位复位时，会禁止接收队列上溢中断。

位 23:17 保留，必须保持复位值。

位 16 **RXOVFIS**：接收队列上溢中断状态 (Receive Queue Overflow Interrupt Status)
此位指示在接收数据包期间，接收队列发生上溢。如果部分数据包已传输到应用程序，则 RDES3[21] 中的上溢状态位置 1。应用程序向该位写入 1 时会将其清零。

位 15:10 保留，必须保持复位值。

位 9 **ABPSIE**：每时隙平均位数中断使能 (Average Bits Per Slot Interrupt Enable)
如果该位置 1，则 MAC 会在每时隙平均位数状态更新时将 eth_sbd_intr_it 中断置为有效。
当该位清零时，不会因此事件而触发中断。

位 8 **TXUIE**：发送队列下溢中断使能 (Transmit Queue Underflow Interrupt Enable)
该位置 1 时，会使能发送队列下溢中断。该位复位时，会禁止发送队列下溢中断。

位 7:2 保留，必须保持复位值。

位 1 **ABPSIS**：每时隙平均位数中断状态 (Average Bits Per Slot Interrupt Status)
该位置 1 时，表示 MAC 已更新 ABS 值。应用程序向该位写入 1 时会将其清零。
注意：一旦复位释放，则无论以太网控制器配置如何，系统都将在进入 1000 Mbps 模式 10 ms（100 Mbps 模式下为 100 ms）后，自动将此位置 1。

位 0 **TXUNFIS**：发送队列下溢中断状态 (Transmit Queue Underflow Interrupt Status)
此位指示在发送数据包期间，发送队列发生下溢。发送会进入挂起状态，且下溢错误 TDES3[2] 置 1。应用程序向该位写入 1 时会将其清零。

Rx 队列 x 工作模式寄存器 (ETH_MTLRXQxOMR)

Rx queue x operating mode register

偏移地址：0x0D30 + 0x40 * x，（x = 0 到 1）

复位值：0x0000 0000

队列 x 接收工作模式寄存器将建立接收队列工作模式及命令。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RQS[2:0]			Res.	Res.	Res.	Res.
									rw	rw	rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIS_TCP_EF	RSF	FEP	FUP	Res.	RTC[1:0]	
									rw	rw	rw	rw		rw	rw

位 31:23 保留，必须保持复位值。

位 22:20 **RQS[2:0]**：接收队列大小 (Receive Queue Size)
该位字段指示分配的接收队列大小（以 256 字节块计）。队列的大小为 (RQS + 1) * 256 字节。

位 19:7 保留，必须保持复位值。

位 6 **DIS_TCP_EF**：禁止丢弃 TCP/IP 校验和错误数据包 (Disable Dropping of TCP/IP Checksum Error Packets)
当此位置 1 时，如果数据包中仅存在由接收校验和减荷引擎检测出来的错误，则 MAC 不会丢弃它。对于这类数据包，仅封装的有效负载中有错误。MAC 接收到的以太网数据包无错误（包括 FCS 错误）。
当此位复位时，如果 FEP 位进行了复位，则会丢弃所有错误数据包。

位 5 **RSF**：接收队列存储并转发 (Receive Queue Store and Forward)

当此位置 1 时，只有在向 Rx 队列写入完整数据包后，以太网外设才会从中读取数据包，同时会忽略该寄存器的 RTC 位字段。当此位复位时，Rx 队列工作在阈值（直通）模式下，具体取决于该寄存器的 RTC 位字段指定的阈值。

位 4 **FEP**：转发错误数据包 (Forward Error Packets)

该位复位时，Rx 队列会丢弃带有错误状态（CRC 错误、接收错误、看门狗超时或上溢）的数据包。不过，如果某个数据包的起始字节（写入）指针已传输到读控制器端（阈值模式下），则不会丢弃该数据包。

该位置 1 时，除过短错误数据包之外的所有数据包都将被转发到应用程序或 DMA。如果 RSF 位置 1，并且在写入部分数据包时 Rx 队列上溢，则无论该位的设置如何，该数据包都会被丢弃。不过，如果 RSF 位复位，并且在写入部分数据包时 Rx 队列上溢，则可能会将部分数据包转发到应用程序或 DMA。

位 3 **FUP**：转发过小的良好数据包 (Forward Undersized Good Packets)

当该位置 1 时，Rx 队列会转发过小的良好数据包（即，没有错误但长度小于 64 字节的数据包），其中包括填充字节和 CRC。当该位复位时，Rx 队列会丢弃所有小于 64 字节的数据包，除非因 Rx 阈值下限更低（例如 RTC = 01）而导致数据包已传输。

位 2 保留，必须保持复位值。

位 1:0 **RTC[1:0]**：接收队列阈值控制 (Receive Queue Threshold Control)

这些位控制 MTL Rx 队列的阈值级别（以字节为单位）：

00：64

01：32

10：96

11：128

当 MTL Rx 队列中的数据包大小大于阈值时，接收到的数据包将传输到应用程序或 DMA。此外，还会自动传输长度小于阈值的完整数据包。

只有在 RSF 位为零时，该位字段才有效。当 RSF 位置 1 时，该位字段将被忽略。

Rx 队列 x 丢失数据包和上溢计数器寄存器 (ETH_MTLRXQxMPOCR)

Rx queue x missed packet and overflow counter register

偏移地址：0x0D34 + 0x40 * x，（x = 0 到 1）

复位值：0x0000 0000

队列 x 丢失数据包和上溢计数器寄存器包含针对以下数据包的计数器：因接收队列数据包刷新而丢失的数据包，以及因接收队列上溢而丢弃的数据包。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	MISCNTOVF	MISPKTCNT[10:0]										
				rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	OVFCNTOVF	OVFPKTCNT[10:0]										
				rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r

位 31:28 保留，必须保持复位值。

位 27 **MISCNTOVF**：丢失数据包计数器上溢位 (Missed Packet Counter Overflow Bit)
该位置 1 时，表示 Rx 队列丢失数据包计数器超出最大限制。

位 26:16 **MISPKTCNT[10:0]**：丢失数据包计数器 (Missed Packet Counter)
该位字段指示以太网外设因应用程序请求针对该队列刷新数据包而丢失的数据包数。对此寄存器执行读取操作时，此计数器复位。
当 DMA 因缓冲区不可用而丢弃数据包时，该计数器会以 1 递增。

位 15:12 保留，必须保持复位值。

位 11 **OVFCNTOVF**：上溢计数器上溢位 (Overflow Counter Overflow Bit)
该位置 1 时，表示 Rx 队列上溢数据包计数器位字段超出最大限制。

位 10:0 **OVFPKTCNT[10:0]**：上溢数据包计数器 (Overflow Packet Counter)
此位字段指示以太网外设因接收队列上溢而丢弃的数据包数。每次以太网外设因上溢而丢弃一个传入数据包时，此计数器值都会递增。对此寄存器执行读取操作时，此计数器复位。

Rx 队列 x 调试寄存器 (ETH_MTLRXQxDR)

Rx queue x debug register

偏移地址：0x0D38 + 0x40 * x，（x = 0 到 1）

复位值：0x0000 0000

队列 x 接收调试寄存器提供与接收队列相关的各个块的调试状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	PRXQ[13:0]													
		r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXQSTS[1:0]		Res.	RRCSTS[1:0]		RWCSTS
										r	r		r	r	r

位 31:30 保留，必须保持复位值。

位 29:16 **PRXQ[13:0]**：接收队列中的数据包数 (Number of Packets in Receive Queue)

此位字段指示 Rx 队列中的当前数据包数。该位字段的理论最大值为 256KB/16B = 16K 个数据包，即，Max_Queue_Size/Min_Packet_Size。

位 15:6 保留，必须保持复位值。

位 5:4 **RXQSTS[1:0]**：MTL Rx 队列填充级别状态 (MTL Rx Queue Fill-Level Status)

该位字段给出了 Rx 队列填充级别的状态：

00：Rx 队列为空

01：Rx 队列填充级别低于流控制取消激活阈值

10：Rx 队列填充级别高于流控制激活阈值

11：Rx 队列已满

位 3 保留，必须保持复位值。

位 2:1 **RRCSTS[1:0]**：MTL Rx 队列读控制器状态 (MTL Rx Queue Read Controller State)

该位字段给出了 Rx 队列读控制器的状态：

00：空闲状态 (Idle state)

01：正在读取数据包数据

10：正在读取数据包状态（或时间戳）

11：正在刷新数据包数据和状态

位 0 **RWCSTS**：MTL Rx 队列写控制器有效状态 (MTL Rx Queue Write Controller Active Status)

该位为高电平时，表示 MTL Rx 队列写控制器有效且正在将接收到的数据包传输到 Rx 队列。

Rx 队列 x 控制寄存器 (ETH_MTLRXQxCR)

Rx queue x control register

偏移地址：0x0D3C + 0x40 * x，（x = 0 到 1）

复位值：0x0000 0000

队列接收控制寄存器将控制接收仲裁，以及将接收到的数据包传输到应用程序。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXQ_FRM_ARBIT	RXQ_WEGT[2:0]		
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3 **RXQ_FRM_ARBIT**：接收队列数据包仲裁 (Receive Queue Packet Arbitration)

当此位置 1 时，以太网外设会将数据包数据驱动至 DMA，以便在切换至其他队列之前发送当前选定队列的整个数据包数据。

当此位复位时，以太网外设会将数据包数据驱动至 DMA，以便在切换至其他队列之前发送以下数量的当前选定队列数据：

– PBL 数量的数据

或

– 一个数据包的完整数据

状态和时间戳并非 PBL 数据的组成部分。因此，以太网外设会在数据包的第一个 PBL 请求（在存储转发模式下）或最后一个 PBL 请求（在阈值模式下）期间，驱动完整的状态（包括时间戳状态）。

位 2:0 **RXQ_WEGT[2:0]**：接收队列权重 (Receive Queue Weight)

该位字段将指示分配给 Rx 队列 x 的权重。权重将用作在一个仲裁周期内分配给队列的连续 PBL 请求次数或连续数据包数量（具体取决于 RXQ_FRM_ARBIT）。

需将此位字段编程为比所需权重少一的值，即复位值 0 表示权重为 1，而复位值 1 则表示权重为 2，依此类推。

注意：RXQ_WEGT 值的变化仅在当前服务轮次完成后或当从 RAA=SP 变为 RAA=WSP 算法时才会生效。采取这种方法是为了确保平稳过渡。要使 RXQ_WEGT 值在开始时生效，必须先于 Rx 队列 x 工作模式寄存器 (ETH_MTLRXQxOMR) 对 Rx 队列 x 控制寄存器 (ETH_MTLRXQxCR) 进行编程。

Tx 队列 x 量子权重寄存器 (ETH_MTLTXQxQWR)

Tx queue x quantum weight register

偏移地址：0x0D18 + 0x40 * x，（x = 0 到 1）

复位值：0x0000 0000

此寄存器包含用于队列的加权轮询 (WRR) 或 IdleSlopeCredit 的权重。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ISCQW[13:0]													
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:14 保留，必须保持复位值。

位 13:0 **ISCQW[13:0]**：IdleSlopeCredit 或权重 (IdleSlopeCredit or Weights)

如果为队列 x 通用数据传输使能了 WRR 算法，则此位字段将包含队列 x 的权重。最大值为 0x64。位 [13:7] 必须写入为零。

注意：在多队列配置中，当使能两个队列时，必须将此位字段编程为某个非零值。当仅使能 Q0 时，无需对此位字段进行编程。一般而言，当选择 WRR 算法时，接收和发送端都必须编程为非零值。在接收端，寄存器将为工作模式寄存器 (ETH_MTLOMR)。

所编程的权重并不与数据包的数量相对应，而是与面向关于总 BW 或总时间的特定队列分配的带宽或时间的一部分相对应。

以太网 MTL 寄存器映射和复位值

表 583. ETH_MTL 寄存器映射和复位值

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0C00	ETH_MTLOMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNTCLR	CNTPRST	Res.	SCHALG[1:0]		Res.	Res.	RAA	DTXSTS	Res.
	Reset value																							0	0		0	0			0	0	
0x0C04 - 0x0C1C	Reserved																																
0x0C20	ETH_MTLISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Q1IS	Q0IS
	Reset value																															0	0
0x0C24 - 0x0CFC	Reserved																																
0x0C40	Reserved																																
0x0D00	ETH_ MTLTXQ0OMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TQS[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TTC[2:0]			TXQEN[1:0]		TSF	FTQ
	Reset value														1	1	1										0	0	0	1	0	0	0
0x0D04	ETH_ MTLTXQ0UR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UFCNTOVF.	UFFRMCNT[10:0]										
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x0D08	ETH_ MTLTXQ0DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STXSTSF[2:0]			Res.	PTXQ[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXSTSFSTS	TXQSTS	TWCSTS	TRCSTS[1:0]		TXQPAUSED
	Reset value										0	0	0		0	0	0											0	0	0	0	0	0
0x0D0C - 0x0D10	Reserved																																
0x0D14	ETH_ MTLTXQ0ESR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ABS[23:0]																							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0D18	ETH_ MTLTXQ0QWR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ISCQW[6:0]						
	Reset value																										0	0	0	0	0	0	0
0x0D28	Reserved																																

表 583. ETH_MTL 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0D2C	ETH_MTLQ0ICSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXOIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXOVFIS	Res.	Res.	Res.	Res.	Res.	Res.	ABPSIE	TXUIE	Res.	Res.	Res.	Res.	Res.	Res.	ABPSIS	TXUNFIS
	Reset value								0								0							0	0							0	0
0x0D30	ETH_ MTLRXQ0OMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RQS[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIS_TCP_EF	RSF	FEP	FUP	Res.	RTC[1:0]	
	Reset value										0	0	0														0	0	0	0		0	0
0x0D34	ETH_ MTLRXQ0MPO CR	Res.	Res.	Res.	Res.	MISCNTOVF	MISPKTCNT[10:0]											Res.	Res.	Res.	Res.	OVFCNTOVF	OVFPKTCNT[10:0]										
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x0D38	ETH_ MTLRXQ0DR	Res.	Res.	PRXQ[13:0]														Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXQSTS[1:0]		Res.	RRCSTS[1:0]		RWCSTS
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0										0	0		0	0	0
0x0D3C	ETH_ MTLRXQ0CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXQ_FRM_ARBIT	RXQ_WEGT[2:0]		
	Reset value																													0	0	0	0
0x0D40	ETH_ MTLTXQ1OMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TQS[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TTC[2:0]			TXQEN[1:0]		TSF	FTQ
	Reset value														0	0	0										0	0	0	0	0	0	0
0x0D44	ETH_ MTLTXQ1UR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UFCNTOVF	UFFRMCNT[10:0]										
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x0D48	ETH_ MTLTXQ1DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STXSTSF[2:0]			Res.	PTXQ[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXSTSFSTS	TXQSTS	TWCSTS	TRCSTS[1:0]		TXQPAUSED
	Reset value										0	0	0		0	0	0											0	0	0	0	0	0	0

表 583. ETH_MTL 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0D4C - 0x0D50	Reserved																																
0x0D54	ETH_ MTLTXQ1ESR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ABS[23:0]																							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0D58	ETH_ MTLTXQ1QWR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ISCQW[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0D5C- 0x0D68	Reserved																																
0x0D6C	ETH_MTLQ1ICSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXOIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXOVFIS	Res.	Res.	Res.	Res.	Res.	Res.	ABPSIE	TXUIE	Res.	Res.	Res.	Res.	Res.	Res.	ABPSIS	TXUNFIS
	Reset value								0								0							0	0							0	0
0x0D70	ETH_ MTLRXQ1OMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RQS[2:0]			Res.	Res.	Res.	RFD[2:0]			Res.	Res.	Res.	RFA[2:0]			EHFC	DIS_TCP_EF	RSF	FEP	FUP	Res.	RTC[1:0]	
	Reset value										0	0	0				0	0	0				0	0	0	0	0	0	0	0		0	0
0x0D74	ETH_MTLRXQ1M POCR	Res.	Res.	Res.	Res.	MISCNTOVF	MISPKTCNT[10:0]											Res.	Res.	Res.	Res.	OVFCNTOVF	OVFPKTCNT[10:0]										
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x0D78	ETH_ MTLRXQ1DR	Res.	Res.	PRXQ[13:0]														Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXQSTS[1:0]		Res.	RRCSTS[1:0]		RWCSTS
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0										0	0		0	0	0
0x0D7C	ETH_ MTLRXQ1CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXQ_FRM_ARBIT	RXQ_WEGT[2:0]		
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

57.11.4 以太网 MAC 和 MMC 寄存器

工作模式配置寄存器 (ETH_MACCR)

Operating mode configuration register

偏移地址：0x0000

复位值：0x0000 8000

MAC 配置寄存器建立 MAC 的工作模式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARPEN		SARC[2:0]		IPC		IPG[2:0]		GPSLCE	S2KP	CST	ACS	WD	BE	JD	JE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PS	FES	DM	LM	ECRSFD	DO	DCRS	DR	Res.	BL[1:0]		DC	PRELEN[1:0]		TE	RE
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31 **ARPEN**：ARP 减荷使能 (ARP Offload Enable)

该位置 1 时，MAC 可以识别传入的 ARP 请求数据包，并会安排发送 ARP 数据包。它会将 ARP 数据包转发到应用程序，还会在 RxStatus 中指示事件。

该位复位时，MAC 接收器不会识别任何 ARP 数据包，并会在 RxStatus 中将其指示为类型帧。

位 30:28 **SARC[2:0]**：源地址插入或替换控制 (Source Address Insertion or Replacement Control)

该位字段控制对所有已发送数据包的源地址插入或替换。位 30 指定用于源地址插入或替换的 MAC 地址寄存器（0 或 1）（基于位 [29:28] 的值）：

010：MAC 会将 MAC 地址 0 寄存器（*MAC 地址 0 高位寄存器 (ETH_MACA0HR)* 和 *MAC 地址 x 低位寄存器 (ETH_MACAxLR)*）的内容插入到所有已发送数据包的 SA 位字段中。

011：MAC 会替换所有已发送数据包 SA 位字段中的 MAC 地址 0 寄存器（*MAC 地址 0 高位寄存器 (ETH_MACA0HR)* 和 *MAC 地址 x 低位寄存器 (ETH_MACAxLR)*）的内容。

110：MAC 会将 MAC 地址 1 寄存器（*MAC 地址 x 高位寄存器 (ETH_MACAxHR)* 和 *MAC 地址 x 低位寄存器 (ETH_MACAxLR)*）的内容插入到所有已发送数据包的 SA 位字段中

111：MAC 会替换所有已发送数据包 SA 位字段中的 MAC 地址 1 寄存器（*MAC 地址 x 高位寄存器 (ETH_MACAxHR)* 和 *MAC 地址 x 低位寄存器 (ETH_MACAxLR)*）的内容。

其他值：保留，不得使用。

注意：对此位字段进行的更改仅在数据包起始时生效。如果在发送数据包时对该寄存器位字段进行写入操作，则只有后续数据包可使用更新值，即，当前数据包不使用更新值。

位 27 **IPC**：校验和减荷 (Checksum Offload)

该位置 1 时，会使能 IPv4 报头校验和检查以及 IPv4 或 IPv6 TCP、UDP 或 ICMP 有效负载校验和检查。该位复位时，将禁止接收器中的 COE 功能。

第 3 层和第 4 层数据包过滤功能可自动选择接收端的 IPC 全校验和减荷引擎。当此功能使能时，必须将 IPC 位置 1。

位 26:24 **IPG[2:0]**：数据包间隙 (Inter-Packet Gap)

这些位控制发送期间数据包间的最小 IPG。

000：96 位时间

001：88 位时间

010：80 位时间

...

111：40 位时间

此最小 IPG 范围在全双工模式下有效。

在半双工模式下，最小 IPG 仅能配置为 64 位时间 (IPG = 100)。不考虑更低的值。

因激活背压而发送 JAM 模式时，MAC 不考虑最小 IPG。

上述功能（IPG 小于 96 位时间）仅在 ETH_MACECR 寄存器中的 EIPGEN 位复位时有效。EIPGEN 置 1 时，将按照 ETH_MACECR 寄存器的 EIPG 位字段中给出的说明控制最小 IPG（大于 96 位时间）。

位 23 **GPSLCE**：大型数据包大小限制控制使能 (Giant Packet Size Limit Control Enable)

当该位置 1 时，MAC 会考虑 ETH_MACECR 寄存器的 GPSL 位字段中的值，以将接收到的数据包声明为大型数据包。必须将该位字段编程为大于 1,518 字节。否则，MAC 会将 1,518 字节视为大型数据包限值。

该位复位时，如果接收到的数据包大小大于 1,518 字节（对于带标记的数据包，为 1522 字节），则 MAC 会将其视为大型数据包。

看门狗超时限制、巨型数据包使能和 2K 数据包使能的优先级高于该位，即，对于接收到的数据包，如果在使能巨型数据包的情况下其大小大于 9,018 字节（对于带标记的数据包，为 9,022 字节）以及在使能 2K 数据包的情况下其大小大于 2,000 字节，则 MAC 会将其视为大型数据包。看门狗超时（如果使能）会在达到看门狗限值时终止接收到的数据包。因此，要获取大型数据包状态，编程的大型数据包限值应小于看门狗限值。

位 22 **S2KP**：2K 数据包的 IEEE 802.3as 支持 (IEEE 802.3as Support for 2K Packets)

该位置 1 时，MAC 将所有长度未超过 2,000 字节的数据包视为正常数据包。JE 位未置 1 时，MAC 将所有接收到的大小超过 2K 字节的数据包视为大型数据包。

该位复位且 JE 位未置 1 时，MAC 将所有接收到的大小超过 1,518 字节（对于带标记的数据包，为 1,522 字节）的数据包视为大型数据包。有关此位和 JE 位的设置会如何影响大型数据包状态的详细信息，请参见*表 584：基于 S2KP 和 JE 位的大型数据包状态*。

注意：JE 位置 1 时，将该位置 1 对大型数据包状态无影响。

位 21 **CST**：类型数据包的 CRC 去除 (CRC stripping for Type packets)

该位置 1 时，在将数据包转发到应用程序之前，所有以太类型的数据包（类型位字段大于 1,536）的最后四个字节 (FCS) 都会被去除和丢弃。在 MAC 接收器中使能 IP 校验和引擎（类型 1）时，此功能无效。只有在使能类型 2 校验和减荷引擎时，该功能才有效。

注意：有关 ACS 位和该位的设置会如何影响数据包长度的信息，请参见表 585：基于 CST 和 ACS 位的数据包长度。

位 20 **ACS**：自动 Pad 或 CRC 去除 (Automatic Pad or CRC Stripping)

该位置 1 时，MAC 仅在长度位字段值小于 1,536 字节时去除传入数据包上的 Pad 或 FCS 位字段。将所有已接收数据包传送给应用程序时，如果长度位字段大于或等于 1,536 字节，将都不去除 Pad 或 FCS 位字段。

该位复位时，MAC 将所有传入的数据包传送给应用程序，而不进行任何修改。

注意：有关 CST 位和该位的设置会如何影响数据包长度的信息，请参见表 585：基于 CST 和 ACS 位的数据包长度。

位 19 **WD**：看门狗禁止 (Watchdog Disable)

当该位置 1 时，MAC 禁止接收器上的看门狗定时器。MAC 可以接收多达 16,383 字节的数据包。

该位复位时，MAC 不允许接收超过 2,048 字节（JE 置为高电平时，为 10,240）的数据包。MAC 会截断在 2,048 字节之后接收到的所有字节。

位 18 **BE**：数据包突发使能 (Packet Burst Enable)

当此位置 1 时，MAC 将允许在 GMII 半双工模式下的发送过程中进行数据包突发。

位 17 **JD**：禁止 Jabber (Jabber Disable)

该位置 1 时，MAC 禁止发送器上的 jabber 定时器。MAC 可以发送多达 16,383 字节的数据包。

该位复位时，如果应用程序在发送期间发送超过 2,048 字节的数据（如果 JE 置为高电平，则为 10,240），则 MAC 不会发送该数据包中的剩余字节。

位 16 **JE**：巨型数据包使能 (Jumbo Packet Enable)

该位置 1 时，MAC 允许 9,018 字节的巨型数据包（对于带 VLAN 标记的数据包，为 9,022 字节），且不会在 Rx 数据包状态中报告大型数据包错误。

有关此位和 JE 位的设置会如何影响大型数据包状态的详细信息，请参见*表 584：基于 S2KP 和 JE 位的大型数据包状态*。

位 15 **PS**：端口选择 (Port Select)

此位将选择以太网线路速度。

0：适用于 1000 Mbps 操作

1：适用于 10 或 100 Mbps 操作

在 10 或 100 Mbps 操作中，此位将与位 FES 一同选择确切的线路速度。

位 14 **FES**：MAC 速度 (MAC Speed)

该位选择 10/100 Mbps 速度模式：

0：10 Mbps

1：100 Mbps

位 13 **DM**：双工模式 (Duplex Mode)

当该位置 1 时，MAC 在全双工模式下工作，此时它可以同时发送和接收。

位 12 **LM**：回环模式 (Loopback Mode)

当该位置 1 时，MAC 在 GMII 或 MII 回环模式下工作。回环正确工作需要 GMII Rx 时钟输入 (eth_mii_rx_clk)，因为 Tx 时钟不是内部回送的。

位 11 **ECRSFD**：在全双工模式下进行发送之前使能载波侦听 (Enable Carrier Sense Before Transmission in Full-duplex mode)

该位置 1 时，MAC 发送器会在全双工模式下发送数据包之前检查 CRS 信号。仅当 CRS 信号为低电平时，MAC 才开始发送。

当该位复位时，MAC 发送器将忽略 CRS 信号的状态。

位 10 **DO**：禁止接收自身 (Disable Receive Own)

如果该位置 1，则在半双工模式下将 ETH_TX_EN 置为有效时，MAC 禁止接收数据包。如果该位复位，MAC 将接收 PHY 提供的所有数据包。

该位不适用于全双工模式。在仅全双工配置中，该位保留，并处于只读 (RO) 状态且具有默认值。

位 9 **DCRS**：在发送期间禁止载波侦听 (Disable Carrier Sense During Transmission)

该位置 1 时，MAC 发送器会在半双工模式下发送数据包期间忽略 GMII CRS 信号。因此，不会因在发送期间载波丢失或无载波而生成错误。

该位复位时，MAC 发送器会因载波侦听而生成错误。MAC 甚至会中止发送。

位 8 **DR**：禁止重试 (Disable Retry)

该位置 1 时，MAC 仅尝试一次发送。当在 GMII 或 MII 接口发生冲突时，MAC 会忽略当前数据包发送，并以 Tx 数据包状态下过度冲突错误报告数据包中止。

该位复位时，MAC 会根据 BL 位字段的设置进行重试。该位仅在半双工模式下适用。

位 7 保留，必须保持复位值。

位 6:5 **BL[1:0]**：后退限制 (Back-Off Limit)

后退限制决定发生冲突后重试期间 MAC 在重新安排一次发送之前等待的随机整数 (r) 个时隙延迟（对于 1000 Mbps 为 4,096 位时间；对于 10/100 Mbps 为 512 位时间）：

00：k = min (n, 10)

01：k = min (n, 8)

10：k = min (n, 4)

11：k = min (n, 1)

其中 n = 重新发送尝试的次数

随机整数 r 的取值范围为 0 <= r < 2^k。

该位仅在半双工模式下适用。

位 4 **DC**：检查延迟 (Deferral Check)

当该位置 1 时，便在 MAC 中使能了检查延迟功能。当 Tx 状态机在 10 或 100 Mbps 模式下延迟超过 24,288 位时间时，MAC 将发出数据包中止状态，同时在 Tx 数据包状态中将过度延迟错误位置 1。

如果将 MAC 配置为 1000 Mbps 操作模式，则延迟阈值为 155,680 位时间。当发送器准备好发送但因 GMII 或 MII 上存在有效载波侦听信号 (CRS) 而被阻止时延迟开始。

延迟时间是非累积性的。例如，如果发送器因为 CRS 信号有效而延迟 10,000 位时间，之后 CRS 信号变为无效，则会导致发送器进行发送时出现冲突。由于发生冲突，因此发送器需要后退，然后在完成后退之后再次延迟。在这种情况下，延迟定时器将复位为 0，并会重新启动。

当该位复位时，禁止延迟检查功能，MAC 发生延迟，直到 CRS 信号变为无效信号。

该位仅在半双工模式下适用。

位 3:2 **PRELEN[1:0]**：发送数据包的前导码长度 (Preamble Length for Transmit packets)

这些位控制被添加到每个 Tx 数据包的开头的前导码字节数。仅当 MAC 工作在全双工模式下时，才可缩短前导码。

00：7 字节前导码

01：5 字节前导码

10：3 字节前导码

11：保留，不得使用

位 1 **TE**：发送器使能 (Transmitter Enable)

当该位置 1 时，使能 MAC 的 Tx 状态机，以便在 GMII 或 MII 接口上进行发送。该位复位时，MAC Tx 状态机会在完成当前数据包的发送后被禁止。Tx 状态机不会再发送任何数据包。

位 0 **RE**：接收器使能 (Receiver Enable)

当该位置 1 时，使能 MAC 的 Rx 状态机，以便从 GMII 或 MII 接口接收数据包。该位复位时，MAC Rx 状态机会在完成当前数据包的接收后被禁止。Rx 状态机不会再从 GMII 或 MII 接口接收任何数据包。

表 584 显示了 ETH_MACCR 寄存器的 S2KP 和 JE 位的设置对大型数据包状态的影响。

表 584. 基于 S2KP 和 JE 位的大型数据包状态(1)

长度/类型位字段	接收到的数据包长度	S2KP	JE	大型数据包状态
未标记的数据包	> 1518	0	0	1
	> 2000	1	0	1
	> 9018	x	1	1
带 VLAN 标记的数据包	> 1522	0	0	1
	> 2000	1	0	1
	> 9022	x	1	1

1. 对于所有其他组合，大型数据包状态均为 0。

表 585 显示了 ETH_MACCR 寄存器的 CST 和 ACS 位的设置如何影响数据包长度是否包含 CRC 长度。

表 585. 基于 CST 和 ACS 位的数据包长度

接收到的数据包长度	CST	ACS	已完成 FCS 去除
< 1536	x	0	否
	x	1	是（针对以太网数据包）
≥ 1536	0	x	否
	1	x	是（针对类型数据包）

扩展工作模式配置寄存器 (ETH_MACECR)

Extended operating mode configuration register

偏移地址：0x0004

复位值：0x0000 0000

MAC 扩展配置寄存器建立 MAC 的工作模式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	EIPG[4:0]					EIPGEN	Res.	Res.	Res.	Res.	Res.	USP	SPEN	DCRCC
		rw	rw	rw	rw	rw	rw						rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	GPSL[13:0]													
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29:25 **EIPG[4:0]**：扩展数据包间隙 (Extended Inter-Packet Gap)

EIPGEN 位置 1 时，此位字段中的值适用。该位字段（作为最高有效位）以及*工作模式配置寄存器 (ETH_MACCR)* 中的 IPG 位字段一起给出最小 IPG（大于 96 位时间，以 8 位时间为步长）。例如：

- EIPG = 0 且 IPG = 0 将提供 104 个位时间
- EIPG = 0 且 IPG = 1 将提供 112 个位时间
- EIPG = 0 且 IPG = 2 将提供 120 个位时间
- ..
- EIPG = 7 且 IPG = 31 将提供 2144 个位时间

位 24 **EIPGEN**：扩展数据包间隙使能 (Extended Inter-Packet Gap Enable)

该位置 1 时，MAC 将 EIPG 位字段和*工作模式配置寄存器 (ETH_MACCR)* 中的 IPG 位字段一起解析为最小 IPG（大于 96 位时间，以 8 位时间为步长）。

该位复位时，MAC 将忽略 EIPG 位字段，并将*工作模式配置寄存器 (ETH_MACCR)* 中的 IPG 位字段解析为最小 IPG（小于等于 96 位时间，以 8 位时间为步长）。

注意：必须仅在全双工模式下工作时使能扩展数据包间隙功能。如果在半双工模式下使能，则可能会对背压功能和帧发送产生不良影响。

位 23:19 保留，必须保持复位值。

位 18 **USP**：单播慢速协议数据包检测 (Unicast Slow Protocol Packet Detect)

该位置 1 时，MAC 会检测具有 *MAC 地址 0 高位寄存器 (ETH_MACA0HR)* 和 MAC 地址 0 低位寄存器 *MAC 地址 x 低位寄存器 (ETH_MACAxLR)* 所指定的站单播地址的慢速协议数据包。MAC 还会检测具有慢速协议多播地址 (01-80-C2-00-00-02) 的慢速协议数据包。

该位复位时，MAC 仅检测具有在 IEEE 802.3-2008 的第 5 节中所指定的慢速协议多播地址的慢速协议数据包。

位 17 **SPEN**：慢速协议检测使能 (Slow Protocol Detection Enable)

该位置 1 时，MAC 会处理慢速协议数据包（以太类型 0x8809）并提供 Rx 状态。MAC 会丢弃具有无效子类型的慢速协议数据包。

该位复位时，MAC 将所有无错误的慢速协议数据包转发到应用程序。MAC 将这类数据包视为正常类型数据包。

位 16 **DCRCC**：禁止对已接收数据包的 CRC 检查 (Disable CRC Checking for Received Packets)

该位置 1 时，MAC 接收器不会检查已接收数据包中的 CRC 位字段。该位复位时，MAC 接收器将始终检查已接收数据包中的 CRC 位字段。

位 15:14 保留，必须保持复位值。

位 13:0 **GPSL[13:0]**：大型数据包大小限制 (Giant Packet Size Limit)

如果接收到的数据包大小大于在此位字段中编程的值（以字节为单位），则 MAC 会将接收到的数据包声明为大型数据包。在此位字段中编程的值必须大于或等于 1,518 字节。任何其他编程值均被视为 1,518 字节。

对于带 VLAN 标记的数据包，MAC 会将编程的值加上 4 个字节。对于带双 VLAN 标记的数据包，MAC 会将编程的值加上 8 个字节。将 ETH_MACCR 寄存器中的 GPSLCE 位置 1 时，该位字段中的值适用。

数据包过滤控制寄存器 (ETH_MACPFR)

Packet filtering control register

偏移地址：0x0008

复位值：0x0000 0000

MAC 数据包过滤寄存器包含用于接收数据包的过滤控件。该寄存器的某些控件转到 MAC 的地址检查块，以执行第一级地址过滤。第二级过滤基于其他控件（例如，传送不良数据包和传送控制数据包）对传入数据包过滤。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DNTU	IPFE	Res.	Res.	Res.	VTFE
rw										rw	rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	HPF	SAF	SAIF	PCF[1:0]		DBF	PM	DAIF	HMC	HUC	PR
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **RA**：接收所有 (Receive All)

该位置 1 时，无论接收到的数据包是否通过地址过滤，MAC 接收器模块都会将它们全部传送至应用程序。SA 或 DA 过滤的结果在 Rx 状态字的相应位进行更新（通过或失败）。

该位复位时，接收器模块仅将通过 SA 或 DA 地址过滤的数据包传送至应用程序。

位 30:22 保留，必须保持复位值。

位 21 **DNTU**：丢弃非 TCP/UDP over IP 数据包 (Drop Non-TCP/UDP over IP Packets)

该位置 1 时，MAC 会丢弃非 TCP/UDP over IP 数据包。MAC 仅转发由第 4 层过滤器处理的数据包。该位复位时，MAC 会转发所有非 TCP/UDP over IP 数据包。

位 20 **IPFE**：第 3 层和第 4 层过滤器使能 (Layer 3 and Layer 4 Filter Enable)

该位置 1 时，MAC 会丢弃与使能的第 3 层和第 4 层过滤器不匹配的数据包。如果未使能第 3 层或第 4 层过滤器以进行匹配，则该位无任何影响。

该位复位时，无论第 3 层和第 4 层位字段的匹配状态如何，MAC 都会转发所有数据包。

位 19:17 保留，必须保持复位值。

位 16 **VTFE**：VLAN 标记过滤器使能 (VLAN Tag Filter Enable)

该位置 1 时，MAC 将丢弃与 VLAN 标记不匹配的带 VLAN 标记的数据包。该位复位时，无论 VLAN 标记的匹配状态如何，MAC 都会转发所有数据包。

位 15:11 保留，必须保持复位值。

位 10 **HPF**：散列或完美过滤器 (Hash or Perfect Filter)

此位置 1 时，如果数据包与完美过滤或散列过滤（通过 HMC 或 HUC 位设置）匹配，则地址过滤器会通过此数据包。

如果该位复位并且 HUC 或 HMC 位置 1，则只有在数据包与散列过滤器匹配时才会通过。

位 9 **SAF**：源地址过滤器使能 (Source Address Filter Enable)

该位置 1 时，MAC 会将所接收数据包的 SA 位字段与使能的 SA 寄存器中编程的值进行比较。如果比较结果为失败，MAC 会丢弃数据包。

该位复位时，MAC 会根据 SA 地址比较结果，通过更新 Rx 状态的 SAF 位来将接收到的数据包转发到应用程序。

注意：根据 IEEE 规范，将保留 SA 的位 47。不过，MAC 会比较所有 48 位。在针对 SA 编程 MAC 地址寄存器时，软件驱动程序应考虑到这一点。

位 8 **SAIF**：SA 反向过滤 (SA Inverse Filtering)

当该位置 1 时，地址检查块在反向过滤模式下进行 SA 地址比较。如果数据包的 SA 与 SA 寄存器中编程的值匹配，则会将其标记为 SA 地址过滤失败。

当该位复位时，如果数据包的 SA 与 SA 寄存器中编程的值不匹配，则会将其标记为 SA 地址过滤失败。

位 7:6 **PCF[1:0]**：通过控制数据包 (Pass Control Packets)

这些位控制所有控制数据包（包括单播和多播暂停数据包）的转发。

00：MAC 会过滤掉所有控制数据包，阻止它们到达应用程序。

01：即使所有控制数据包均未通过地址过滤，MAC 也仍将它们转发给应用程序（暂停数据包除外）。

10：即使所有控制数据包均未通过地址过滤，MAC 也仍将它们转发给应用程序。

11：MAC 转发通过地址过滤的控制数据包。

位 5 **DBF**：禁止广播数据包 (Disable Broadcast Packets)

该位置 1 时，AFM 模块会阻止所有传入的广播数据包。此外，它还会覆盖所有其他过滤设置。

该位复位时，AFM 模块会通过所有接收到的广播数据包。

位 4 **PM**：传送所有多播 (Pass All Multicast)

该位置 1 时，指示通过接收到的带多播目标地址（目标地址位字段的第一位是“1”）的所有数据包。该位复位时，多播数据包的过滤取决于 HMC 位。

位 3 **DAIF**：DA 反向过滤 (DA Inverse Filtering)

当该位置 1 时，地址检查块在反向过滤模式下对单播和多播数据包均进行 DA 地址比较。该位复位时，执行数据包的正常过滤。

位 2 **HMC**：散列多播 (Hash Multicast)

该位置 1 时，MAC 根据散列表对接收到的多播数据包执行目标地址过滤。

该位复位时，MAC 对多播数据包执行完美目标地址过滤，即，将 DA 位字段与 DA 寄存器中编程的值进行比较。

位 1 **HUC**：散列单播 (Hash Unicast)

该位置 1 时，MAC 根据散列表对单播数据包执行目标地址过滤。

该位复位时，MAC 对单播数据包执行完美目标地址过滤，即，将 DA 位字段与 DA 寄存器中编程的值进行比较。

位 0 **PR**：混合模式 (Promiscuous Mode)

当该位置 1 时，不论目标或源地址为何，地址过滤模块都会通过所有传入的数据包。PR 置 1 时，始终将 Rx 状态字的 SA 或 DA 过滤失败状态位清零。

看门狗超时寄存器 (ETH_MACWTR)

Watchdog timeout register

偏移地址：0x000C

复位值：0x0000 0000

看门狗超时寄存器用于控制已接收数据包的看门狗超时。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	PWE	Res	Res	Res	Res	WTO[3:0]			
							rw					rw	rw	rw	rw

位 31:9 保留，必须保持复位值。

位 8 **PWE**：可编程看门狗使能 (Programmable Watchdog Enable)

当该位置 1 且*工作模式配置寄存器 (ETH_MACCR)* 寄存器的 WD 位复位时，WTO 位字段用作已接收数据包的看门狗超时。当该位清零时，通过设置*工作模式配置寄存器 (ETH_MACCR)* 寄存器的 WD 和 JE 位来控制已接收数据包的看门狗超时。

位 7:4 保留，必须保持复位值。

位 3:0 **WTO[3:0]**：看门狗超时

当 PWE 位置 1 且 *工作模式配置寄存器 (ETH_MACCR)* 寄存器的 WD 位复位时，该位字段用作已接收数据包的看门狗超时。如果接收到的数据包的长度超过该位字段的值，则此数据包会被终止并被声明为错误数据包。

编码如下：

0x0：2 KB

0x1：3 KB

0x2：4 KB

0x3：5 KB

..

0xC：14 KB

0xD：15 KB

0xE：16383 字节

0xF：保留，不得使用

注意：当 PWE 位置 1 时，该位字段的值应大于 1,522 (0x05F2)。否则，IEEE 802.3 指定的带标记的有效数据包会被声明为错误数据包，然后被丢弃。

散列表 0 寄存器 (ETH_MACHT0R)

Hash Table 0 register

偏移地址：0x0010

复位值：0x0000 0000

散列表寄存器 0 包含散列表（64 位）的前低 32 位。

散列表用于组地址过滤。

对于散列过滤，传入数据包中目标地址的内容通过 CRC 逻辑传送，CRC 寄存器的高六位用于对散列表内容进行索引。最高有效位将决定要使用的寄存器（散列表寄存器 0 或 1），而五个最低有效位将决定寄存器中的位。例如，散列值 0b100000 用于选择散列表寄存器 1 的位 0。

目标地址的散列值的计算方式如下：

1. 计算 DA 的 32 位 CRC（有关计算 CRC32 的步骤，请参见 IEEE 802.3 的第 3.2.8 节）。
2. 对第 1 步中获得的值执行按位反转。
3. 取第 2 步获得的值的高 7 或 8 位。

如果寄存器的相应位值为 1，则接受该数据包。否则拒绝该帧。如果 ETH_MACPFR 中的 PM 位置 1，则不论多播散列值为何，都会接受所有多播数据包。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HT31T0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HT31T0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **HT31T0[31:0]**：MAC 散列表的前 32 位 (MAC Hash Table First 32 Bits)
该位字段包含散列表的前 32 位，即位 [31:0]。

散列表 1 寄存器 (ETH_MACHT1R)

Hash Table 1 register

偏移地址：0x0014

复位值：0x0000 0000

散列表 1 寄存器包含散列表（64 位）的高 32 位。

散列表用于组地址过滤。

对于散列过滤，传入数据包中目标地址的内容通过 CRC 逻辑传送，CRC 寄存器的高六位用于对散列表内容进行索引。最高有效位将决定要使用的寄存器（散列表寄存器 0 或 1），而五个最低有效位将决定寄存器中的位。例如，散列值 6‘b100000 用于选择散列表寄存器 1 的位 0。

目标地址的散列值的计算方式如下：

1. 计算 DA 的 32 位 CRC（有关计算 CRC32 的步骤，请参见 IEEE 802.3 的第 3.2.8 节）。
2. 对第 1 步中获得的值执行按位反转。
3. 取第 2 步获得的值的高 7 或 8 位。

如果寄存器的相应位值为 1，则接受该数据包。否则拒绝该帧。如果 ETH_MACPFR 中的 PM 位置 1，则不论多播散列值为何，都会接受所有多播数据包。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HT63T32[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HT63T32[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **HT63T32[31:0]**：MAC 散列表的后 32 位 (MAC Hash Table Second 32 Bits)
该位字段包含散列表的后 32 位，即位 [63:32]。

VLAN 标记寄存器 (ETH_MACVTR)

VLAN tag register

偏移地址：0x0050

复位值：0x0000 0000

VLAN 标记寄存器定义 IEEE 802.1Q VLAN 类型数据包。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EIVLRXS	Res.	EIVLS[1:0]		ERIVLT	EDVLP	VTHM	EVLRXS	Res.	EVLS[1:0]		DOVLTC	ERSVLM	ESVL	VTIM	ETV
rw		rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VL[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **EIVLRXS**：在 Rx 状态中使能内部 VLAN 标记 (Enable Inner VLAN Tag in Rx Status)
该位置 1 时，MAC 会在 Rx 状态中提供内部 VLAN 标记。该位复位时，MAC 不会在 Rx 状态中提供内部 VLAN 标记。

位 30 保留，必须保持复位值。

位 29:28 **EIVLS[1:0]**：在接收端使能内部 VLAN 标记去除 (Enable Inner VLAN Tag Stripping on Receive)
此位字段指示在接收到的数据包中对内部 VLAN 标记进行的去除操作：
00：不去除
01：VLAN 过滤通过时进行去除操作
10：VLAN 过滤失败时进行去除操作
11：始终进行去除操作

位 27 **ERIVLT**：使能内部 VLAN 标记 (Enable Inner VLAN Tag)
该位和 EDVLP 位字段均置 1 时，MAC 接收器会使能针对内部 VLAN 标记（如果存在）的操作。该位复位时，MAC 接收器会使能针对外部 VLAN 标记（如果存在）的操作。ERSVLM 位确定使能哪种 VLAN 类型进行过滤或匹配。ERSVLM 位和 DOVLTC 位确定使能哪种 VLAN 类型进行过滤。

位 26 **EDVLP**：使能双 VLAN 处理 (Enable Double VLAN Processing)
该位置 1 时，MAC 将支持在 Tx 和 Rx 端处理最多两个 VLAN 标记（如果存在）。该位复位时，MAC 将支持在 Tx 和 Rx 端处理最多一个 VLAN 标记（如果存在）。

位 25 **VTHM**：VLAN 标记散列表匹配使能 (VLAN Tag Hash Table Match Enable)

如果该位置 1，则使用 VLAN 标记的 CRC 的四个最高有效位来索引 ETH_MACVLANHTR 寄存器的内容。VLAN 散列表寄存器的值 1 对应于索引，表示数据包与 VLAN 散列表匹配。

ETV 位置 1 时，使用 12 位 VLAN 标识符 (VID) 的 CRC 进行比较。ETV 位复位时，使用 16 位 VLAN 标记的 CRC 进行比较。

该位复位时，不执行 VLAN 散列匹配操作。

位 24 **EVLRXS**：在 Rx 状态中使能 VLAN 标记 (Enable VLAN Tag in Rx status)

该位置 1 时，MAC 会在 Rx 状态中提供外部 VLAN 标记。该位复位时，MAC 不会在 Rx 状态中提供外部 VLAN 标记。

位 23 保留，必须保持复位值。

位 22:21 **EVLS[1:0]**：在接收端使能 VLAN 标记去除 (Enable VLAN Tag Stripping on Receive)

此位字段指示在接收到的数据包中对外部 VLAN 标记进行的去除操作：

00：不去除

01：VLAN 过滤通过时进行去除操作

10：VLAN 过滤失败时进行去除操作

11：始终进行去除操作

位 20 **DOVLTC**：禁止 VLAN 类型检查 (Disable VLAN Type Check)

该位置 1 时，MAC 不检查由 ERIVLT 位指定的 VLAN 标记是 S-VLAN 类型还是 C-VLAN 类型。该位复位时，只有当 VLAN 标记类型与 ERSVLM 位指定的类型相似时，MAC 才会对 ERIVLT 位指定的 VLAN 标记进行过滤或匹配。

位 19 **ERSVLM**：使能接收 S-VLAN 匹配 (Enable Receive S-VLAN Match)

该位置 1 时，MAC 接收器会使能针对 S-VLAN（类型 = 0x88A8）数据包的过滤或匹配。该位复位时，MAC 接收器会使能针对 C-VLAN（类型 = 0x8100）数据包的过滤或匹配。

ERIVLT 位确定用于过滤或匹配的 VLAN 标记位置。

位 18 **ESVL**：使能 S-VLAN (Enable S-VLAN)

该位置 1 时，MAC 发送器和接收器会将 S-VLAN 数据包（类型 = 0x88A8）视为带 VLAN 标记的有效数据包。

位 17 **VTIM**：VLAN 标记反向匹配使能 (VLAN Tag Inverse Match Enable)

该位置 1 时，会使能 VLAN 标记反向匹配。不含匹配 VLAN 标记的数据包被标记为匹配。该位复位时，会使能 VLAN 标记完美匹配。含有匹配 VLAN 标记的数据包被标记为匹配。

位 16 **ETV**：使能 12 位 VLAN 标记比较 (Enable 12-Bit VLAN Tag Comparison)

该位置 1 时，使用 12 位 VLAN 标识符进行比较和过滤，而不使用完整的 16 位 VLAN 标记。VLAN 标记的位 [11:0] 与所接收带 VLAN 标记的数据包中的相应位字段进行比较。类似地，使能时，只有所接收数据包中的 12 位 VLAN 标记用于基于散列的 VLAN 过滤。

该位复位时，所接收 VLAN 数据包的第 15 个和第 16 个字节的所有 16 位都用于比较和 VLAN 散列过滤。

位 15:0 **VL[15:0]**：用于接收数据包的 VLAN 标记标识符 (VLAN Tag Identifier for Receive Packets)

该位字段包含用于识别 VLAN 数据包的 802.1Q VLAN 标记。该 VLAN 标记标识符将与接收到的数据包的第 15 个和第 16 个字节进行比较，以识别 VLAN 数据包。下面对该位字段的各个位进行了说明：

位[15:13]：用户优先级

位 12：标准格式指示符 (CFI) 或丢弃合法指示符 (DEI)

位[11:0]：VLAN 标记的 VLAN 标识符 (VID) 位字段

ETV 位置 1 时，仅使用 VID 进行比较。

如果该位字段（ETV 置 1 时为 [11:0]）全部为零，则 MAC 不会检查用于 VLAN 标记比较的第 15 个和第 16 个字节，而是将类型位字段值为 0x8100 或 0x88a8 的所有数据包均声明为 VLAN 数据包。

VLAN 散列表寄存器 (ETH_MACVHTR)

VLAN Hash table register

偏移地址：0x0058

复位值：0x0000 0000

VLAN 标记寄存器 (ETH_MACVTR) 寄存器的 VTHM 位置 1 时，16 位 VLAN 散列表寄存器用于基于 VLAN 标记进行组地址过滤。对于散列过滤，传入数据包中的 16 位 VLAN 标记或 12 位 VLAN ID（取决于 *VLAN 标记寄存器 (ETH_MACVTR)* 寄存器的 ETV 位）的内容通过 CRC 逻辑传送。计算得出的 CRC 的高四位用于索引 VLAN 散列表的内容。例如，散列值 1000 用于选择 VLAN 散列表的位 8。

目标地址的散列值的计算方式如下：

1. 计算 VLAN 标记或 ID 的 32 位 CRC（有关计算 CRC32 的步骤，请参见 IEEE 802.3 的第 3.2.8 节）。
2. 对第 1 步中获得的值执行按位反转。
3. 取第 2 步获得的值的高四位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VLHT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **VLHT[15:0]**：VLAN 散列表 (VLAN Hash Table)
该位字段包含 16 位 VLAN 散列表。

VLAN 包含寄存器 (ETH_MACVIR)

VLAN inclusion register

偏移地址：0x0060

复位值：0x0000 0000

VLAN 标记包含或替换寄存器包含用于在发送数据包中进行插入或替换的 VLAN 标记。它还包含 VLAN 标记插入控件。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VLTI	CSVL	VLP	VLC[1:0]	
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VLT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值。

位 20 **VLTI**：VLAN 标记输入 (VLAN Tag Input)

该位置 1 时，表示应从 Tx 描述符中获取要在 Tx 数据包中插入或替换的 VLAN 标记。

位 19 **CSVL**：C-VLAN 或 S-VLAN (C-VLAN or S-VLAN)

该位置 1 时，会在所发送数据包的第 13 个和第 14 个字节中插入或替换 S-VLAN 类型 (0x88A8)。该位复位时，会在所发送数据包的第 13 个和第 14 个字节中插入或替换 C-VLAN 类型 (0x8100)。

0：C-LAN

1：S-LAN

位 18 **VLP**：VLAN 优先级控制 (VLAN Priority Control)

该位置 1 时，使用控制位 [17:16] 进行 VLAN 删除、插入或替换。当此位复位时，系统将忽略位 [17:16]。

位 17:16 **VLC[1:0]**：发送数据包中的 VLAN 标记控制 (VLAN Tag Control in Transmit Packets)

00：不进行 VLAN 标记删除、插入或替换

01：VLAN 标记删除。MAC 将删除所有带 VLAN 标记的已发送数据包的 VLAN 类型（字节 13 和 14）以及 VLAN 标记（字节 15 和 16）。

10：VLAN 标记插入。在字节 13 和 14 中插入类型值（0x8100 或 0x88a8）后，MAC 会在数据包的字节 15 和 16 中插入 VLT。无论已发送数据包是否已具有 VLAN 标记，都会对它们全部进行该操作。

11：VLAN 标记替换。MAC 替换所有 VLAN 类型已发送数据包的字节 15 和 16 中的 VLT（字节 13 和 14 的值为 0x8100 或 0x88a8）。

注意：对此位字段进行的更改仅在数据包起始时生效。如果在发送数据包时对该寄存器位字段进行写入操作，则只有后续数据包可使用更新值，即，当前数据包不使用更新值。

位 15:0 **VLT[15:0]**：发送数据包的 VLAN 标记 (VLAN Tag for Transmit Packets)

该位字段包含要插入或替换的 VLAN 标记的值。只有在发送线无效或处于初始化阶段时，才能更改该值。

下面对该位字段的各个位进行了说明：

位[15:13]：用户优先级

位 12：标准格式指示符 (CFI) 或丢弃合法指示符 (DEI)

位[11:0]：VLAN 标记的 VLAN 标识符 (VID) 位字段

内部 VLAN 包含寄存器 (ETH_MACIVIR)

Inner VLAN inclusion register

偏移地址：0x0064

复位值：0x0000 0000

内部 VLAN 标记包含或替换寄存器包含要在发送数据包中插入或替换的内部 VLAN 标记。它还包含内部 VLAN 标记插入控件。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VLTI	CSVL	VLP	VLC[1:0]	
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VLT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值。

位 20 **VLTI**：VLAN 标记输入 (VLAN Tag Input)

该位置 1 时，表示应从 Tx 描述符中获取要在 Tx 数据包中插入或替换的 VLAN 标记。

位 19 **CSVL**：C-VLAN 或 S-VLAN (C-VLAN or S-VLAN)

该位置 1 时，会在所发送数据包的第 13 个和第 14 个字节中插入或替换 S-VLAN 类型 (0x88A8)。该位复位时，会在所发送数据包的第 13 个和第 14 个字节中插入或替换 C-VLAN 类型 (0x8100)。

0：C-LAN

1：S-LAN

位 18 **VLP**：VLAN 优先级控制 (VLAN Priority Control)

该位置 1 时，使用 VLC 位字段进行 VLAN 删除、插入或替换。当此位复位时，系统会忽略 VLC 位字段。

位 17:16 **VLC[1:0]**：发送数据包中的 VLAN 标记控制 (VLAN Tag Control in Transmit Packets)

00：不进行 VLAN 标记删除、插入或替换

01：VLAN 标记删除

MAC 将删除所有带 VLAN 标记的已发送数据包的 VLAN 类型（字节 17 和 18）以及 VLAN 标记（字节 19 和 20）。

10：VLAN 标记插入

在字节 17 和 18 中插入类型值（0x8100 或 0x88a8）后，MAC 会在数据包的字节 19 和 20 中插入 VLT。无论已发送数据包是否已具有 VLAN 标记，都会对它们全部进行该操作。

11：VLAN 标记替换

MAC 替换所有 VLAN 类型已发送数据包的字节 19 和 20 中的 VLT（字节 17 和 18 的值为 0x8100 或 0x88a8）。

注意：对此位字段进行的更改仅在数据包起始时生效。如果在发送数据包时对该寄存器位字段进行写入操作，则只有后续数据包可使用更新值，即，当前数据包不使用更新值。

位 15:0 **VLT[15:0]**：发送数据包的 VLAN 标记 (VLAN Tag for Transmit Packets)

该位字段包含要插入或替换的 VLAN 标记的值。只有在发送线无效或处于初始化阶段时，才能更改该值。

下面对该位字段的各个位进行了说明：

位[15:13]：用户优先级

位 12：标准格式指示符 (CFI) 或丢弃合法指示符 (DEI)

位[11:0]：VLAN 标记的 VLAN 标识符 (VID) 位字段

Tx 队列 0 流控制寄存器 (ETH_MACQ0TXFCR)

Tx Queue 0 flow control register

偏移地址：0x0070

复位值：0x0000 0000

流控制寄存器通过 MAC 的流控制模块来对控制（暂停命令）数据包的生成和接收进行控制。忙碌位置 1 时对寄存器进行写入操作将触发流控制模块生成暂停数据包。控制数据包的位字段根据 802.3x 规范中的指定进行选择，此寄存器中的暂停时间值被用在控制数据包的暂停时间位字段中。忙碌位将保持置 1，直到将控制数据包发送到电缆。应用程序必须确保向该寄存器写入前已将忙碌位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DZPQ	PLT[2:0]			Res.	Res.	TFE	FCB_BPA
								rw	rw	rw	rw			rw	rw

位 31:16 **PT[15:0]**：暂停时间 (Pause time)
此位字段保留 Tx 控制数据包中暂停时间位字段要使用的值。

位 15:8 保留，必须保持复位值。

位 7 **DZPQ**：禁止零时间片暂停 (Disable Zero-Quanta Pause)
如果该位置 1，则系统将禁止自动生成零时间片暂停数据包。
该位复位时，会使能自动零时间片暂停数据包生成的正常操作。

位 6:4 **PLT[2:0]**：暂停阈值下限 (Pause Low Threshold)
该位字段配置暂停定时器的阈值，在该阈值处将检查输入流，以便自动重新发送暂停数据包。
该阈值应始终小于在位 [31:16] 中配置的暂停时间。例如，如果 PT = 100H（256 个时隙），而 PLT = 001，则在第一个暂停数据包发送完成后，第二个暂停数据包会在 228 (256-28) 个时隙启动时自动发送。
下面给出了针对不同值的阈值：
000：暂停时间减去 4 个时隙（PT - 4 个时隙）
001：暂停时间减去 28 个时隙（PT - 28 个时隙）
010：暂停时间减去 36 个时隙（PT - 36 个时隙）
011：暂停时间减去 144 个时隙（PT - 144 个时隙）
100：暂停时间减去 256 个时隙（PT - 256 个时隙）
101：暂停时间减去 512 个时隙（PT - 512 个时隙）
110 到 111：保留，不得使用
时隙定义为 GMII 或 MII 接口每发送 512 位（64 字节）所需的时间。
此（近似）计算基于数据包大小（64、1518、2000、9018、16384 或 32768）+ 2 个暂停数据包大小 + IPG（以时隙计）。

位 3:2 保留，必须保持复位值。

位 1 **TFE**：发送流控制使能 (Transmit Flow Control Enable)
全双工模式：当此位置 1 时，MAC 将使能流控制操作来发送暂停数据包。当此位复位时，将禁止 MAC 中的流控制操作，MAC 不会发送任何暂停数据包。
半双工模式：当此位置 1 时，MAC 将使能背压操作。当此位复位时，将禁止背压功能。

位 0 **FCB_BPA**：流控制忙碌或背压激活 (Flow Control Busy or Backpressure Activate)

此位在全双工模式下启动暂停数据包，在半双工模式下，TFE 位置 1 时，会激活背压功能。

全双工模式：向该寄存器写入数据前此位应读为 0。要启动暂停数据包，应用程序必须将此位置 1。在控制数据包传输过程中，此位保持置 1 以指示数据包发送正在进行中。当暂停数据包发送完成时，MAC 会将该位复位为 0。在该位清零之前，不应对该寄存器进行写入操作。

半双工模式：如果该位在半双工模式下置 1（且 TFE 位置 1），则 MAC 会激活背压。在背压操作期间，当 MAC 接收到新数据包时，发送器会开始发送一个导致冲突的 JAM 模式。当 MAC 配置为全双工模式时，将自动禁止 BPA。

Rx 流控制寄存器 (ETH_MACRXFCR)

Rx flow control register

偏移地址：0x0090

复位值：0x0000 0000

接收流控制寄存器根据接收到的暂停数据包来控制 MAC 发送的暂停。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	UP	RFE
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **UP**：单播暂停数据包检测 (Unicast Pause Packet Detect)

如果暂停数据包具有在 IEEE 802.3 中指定的唯一多播地址，则会对其进行处理。该位置 1 时，MAC 还可以检测具有站的单播地址的暂停数据包。该单播地址应为 *MAC 地址 0 高位寄存器 (ETH_MACA0HR)* 和 MAC 地址 0 低位寄存器 *MAC 地址 x 低位寄存器 (ETH_MACAxLR)* 所指定。

当该位复位时，MAC 仅检测具有唯一多播地址的暂停数据包。

注意：如果多播地址与该唯一多播地址不同，则 MAC 不会处理暂停数据包。这同样适用于使能优先级流控制 (PFC) 时接收到的 PFC 数据包。唯一多播地址 (0x01_80_C2_00_00_01) 如 IEEE 802.1 Qbb-2011 之规定。

位 0 **RFE**：接收流控制使能 (Receive Flow Control Enable)

当此位置 1 且 MAC 工作在全双工模式下时，MAC 对接收到的暂停数据包进行解码，并禁止其在指定（暂停）时间内发送。当该位复位或 MAC 工作在半双工模式下时，会禁止暂停数据包的解码功能。

使能 PFC 时，将为 PFC 数据包使能流控制。MAC 会对接收到的 PFC 数据包进行解码，并在接收到的暂停时间内禁止优先级匹配的发送队列。

Rx 队列控制寄存器 (ETH_MACRXQCR)

Rx Queue control register

偏移地址：0x0094

复位值：0x0000 0000

接收队列控制寄存器会控制将无法通过目标或源地址过滤器的单播和多播数据包传送到 Rx 队列

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VFFQ	VFFQE
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	MFFQ	MFFQE	Res.	Res.	Res.	Res.	Res.	Res.	UFFQ	UFFQE
						rw	rw							rw	rw

位 31:18 保留，必须保持复位值。

位 17 **VFFQ**：VLAN 标记过滤失败数据包队列 (VLAN Tag Filter Fail Packets Queue)

此位字段将存储必须将未通过目标或源地址过滤器（且未使能 UFFQE/MFFQE）或未通过 VLAN 标记过滤器的带标记数据包传送到的 Rx 队列号。只有在 VFFQE 位置 1 时，该位字段才有效。

0：队列 0

1：队列 1

位 16 **VFFQE**：VLAN 标记过滤失败数据包排队使能 (VLAN Tag Filter Fail Packets Queuing Enable)

当此位置 1 时，未通过目标或源地址过滤器或未通过 VLAN 标记过滤器的带标记数据包将传送到在 VFFQ 中编程的 Rx 队列号。当此位复位时，未通过目标或源地址过滤器或未通过 VLAN 标记过滤器的带标记数据包会根据其他路由选项进行传送。只有在*数据包过滤控制寄存器 (ETH_MACPFR)* 的 RA 位置 1 时，该位才有效。

0：禁止

1：使能

位 15:10 保留，必须保持复位值。

位 9 **MFFQ**：多播地址过滤失败数据包队列 (Multicast Address Filter Fail Packets Queue)。

此位字段将存储必须将未通过目标或源地址过滤器的多播数据包传送到的 Rx 队列号。只有在 MFFQE 位置 1 时，该位字段才有效。

0：队列 0

1：队列 1

位 8 **MFFQE**：多播地址过滤失败数据包排队使能 (Multicast Address Filter Fail Packets Queuing Enable)。

当此位置 1 时，未通过目标或源地址过滤器的多播数据包将传送到在 MFFQ 中编程的 Rx 队列号。

当此位复位时，未通过目标或源地址过滤器的多播数据包会根据其他路由选项进行传送。

只有在*数据包过滤控制寄存器 (ETH_MACPFR)* 的 RA 位置 1 时，该位才有效。

位 7:2 保留，必须保持复位值。

位 1 **UFFQ**：单播地址过滤失败数据包队列 (Unicast Address Filter Fail Packets Queue)。

此位字段将存储未通过目标或源地址过滤器的单播数据包将传送到的 Rx 队列号。只有在 UFFQE 位置 1 时，该位字段才有效。

0：队列 0

1：队列 1

位 0 **UFFQE**：单播地址过滤失败数据包排队使能 (Unicast Address Filter Fail Packets Queuing Enable)。

当此位置 1 时，未通过目标或源地址过滤器的单播数据包将传送到在 UFFQ 中编程的 Rx 队列号。

当此位复位时，未通过目标或源地址过滤器的单播数据包会根据其他路由选项进行传送。

只有在*数据包过滤控制寄存器 (ETH_MACPFR)* 的 RA 位置 1 时，该位才有效。

Rx 队列控制 0 寄存器 (ETH_MACRXQC0R)

Rx queue control 0 register

偏移地址：0x00A0

复位值：0x0000 0000

接收队列控制 0 寄存器将控制 MAC 接收器中的队列管理。

注： *在多个 Rx 队列配置中，默认情况下所有队列均处于禁止状态。可通过对此寄存器中的相应位字段进行编程来使能 Rx 队列。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXQ1EN[1:0]		RXQ0EN[1:0]	
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:2 **RXQ1EN[1:0]**：接收队列 1 使能 (Receive Queue 1 Enable)

此位字段指示是否针对 AV 或通用数据传输使能 Rx 队列 1。

00：未使能

01：为 AV 使能队列 1

10：为通用数据传输使能队列 1

11：保留，不得使用

位 1:0 **RXQ0EN[1:0]**：接收队列 0 使能 (Receive Queue 0 Enable)

此位字段指示是否针对 AV 或通用数据传输使能 Rx 队列 0。

00：未使能

01：为 AV 使能队列 0

10：为通用数据传输使能队列 0

11：保留，不得使用

Rx 队列控制 1 寄存器 (ETH_MACRXQC1R)

Rx queue control 1 register

偏移地址：0x00A4

复位值：0x0000 0000

接收队列控制 1 寄存器会控制将多播、广播及未标记的数据包传送到 Rx 队列。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TPQC	TACPQE	MCBCQEN	Res.	MCBCQ[2:0]		
									rw	rw	rw		rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	UPQ[2:0]			Res.	Res.	Res.	Res.	Res.	PTPQ[2:0]			Res.	Res.	Res.	Res.
	rw	rw	rw						rw	rw	rw				

位 31:24 保留，必须保持复位值。

位 23 保留，必须保持复位值。

位 22 **TPQC**：带标记的 PTP over Ethernet 数据包排队控制 (Tagged PTP over Ethernet Packets Queuing Control)

此位字段将控制带 VLAN 标记的 PTPoE 数据包的传送。

0：带 VLAN 标记的 PTPoE 数据包将作为带 VLAN 标记的通用数据包（基于使能通用的 Rx 队列的 PSRQ）进行传送

1：带 VLAN 标记的 PTPoE 数据包将传送到 PTPQ 位字段指定的 Rx 队列

位 21 **TACPQE**：带标记 AV 控制数据包排队使能 (Tagged AV Control Packets Queuing Enable)

当此位置 1 时，MAC 会将接收到的带标记 AV 控制数据包传送到 AVCPQ 位字段指定的 Rx 队列。

当此位复位时，MAC 会根据与 ETH_MACRXQC2R 寄存器中的 PSRQ 位字段匹配的标记优先级来传送接收到的带标记 AV 控制数据包。

如果未选择将 Rx 队列用于 AV 功能，则将保留此位字段。

位 20 **MCBCQEN**：多播和广播队列使能 (Multicast and Broadcast Queue Enable)

此位将指定使能将多播或广播数据包传送到 Rx 队列，且必须将多播或广播数据包传送到 MCBCQ 位字段中指定的 Rx 队列。

位 19 保留，必须保持复位值。

位 18:16 **MCBCQ[2:0]**：多播和广播队列 (Multicast and Broadcast Queue)

此位字段将指定要将多播或广播数据包传送到的 Rx 队列。任何使能了通用/AV 数据传输的 Rx 队列都可以用来传送多播或广播数据包。

000：Rx 队列 0

001：Rx 队列 1

其他值：保留，不得使用

位 15 保留，必须保持复位值。

位 14:12 **UPQ[2:0]**：未标记的数据包队列 (Untagged Packet Queue)

此位字段指示要将未标记的数据包传送到的 Rx 队列。任何使能了通用/AV 数据传输的 Rx 队列都可以用来传送未标记的数据包。

000：Rx 队列 0

001：Rx 队列 1

其他值：保留，不得使用

位 11:7 保留，必须保持复位值。

位 6:4 **PTPQ[2:0]**：PTP 数据包队列 (PTP Packets Queue)

此位字段指定通过以太网有效负载（而非 IPv4 或 IPv6）发送的 PTP 数据包将传送到的 Rx 队列。

000：Rx 队列 0

001：Rx 队列 1

其他值：保留，不得使用

当*时间戳控制寄存器 (ETH_MACTSCR)* 的 AV8021ASMEN 位置 1 时，只有未标记的 PTP over Ethernet 数据包会传送到 Rx 队列。如果此位未置 1，则根据 TPQC 位字段的编程情况，带有和未带有标记的 PTPoE 数据包都可以传送到此 Rx 队列。

位 3:0 保留，必须保持复位值。

Rx 队列控制 2 寄存器 (ETH_MACRXQC2R)

Rx queue control 2 register

偏移地址：0x00A8

复位值：0x0000 0000

该寄存器将根据已接收数据包的 USP（用户优先级）位字段来控制将带标记的数据包传送到 Rx 队列 0 和 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSRQ1[7:0]								PSRQ0[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:8 **PSRQ1[7:0]**：接收队列 1 中选择的优先级 (Priorities Selected in the Receive Queue 1)

此位字段将定义分配给 Rx 队列 1 的优先级。优先级与此位字段中设置的值相匹配的所有数据包都将传送到 Rx 队列 1。

例如，如果 PSRQ1[4] 置 1，则 USP 位字段等于 4 的数据包将传送到 Rx 队列 1。软件必须确保此位字段的内容与其他队列的 PSRQ 位字段互斥，即相同的优先级不会映射到多个 Rx 队列。

位 7:0 **PSRQ0[7:0]**：接收队列 0 中选择的优先级 (Priorities Selected in the Receive Queue 0)

此位字段将定义分配给 Rx 队列 0 的优先级。优先级与此位字段中设置的值相匹配的所有数据包都将传送到 Rx 队列 0。

例如，如果 PSRQ0[5] 置 1，则 USP 位字段等于 5 的数据包将传送到 Rx 队列 0。软件必须确保此位字段的内容与其他队列的 PSRQ 位字段互斥，即相同的优先级不会映射到多个 Rx 队列。

中断状态寄存器 (ETH_MACISR)

Interrupt status register

偏移地址：0x00B0

复位值：0x0000 0000

中断状态寄存器包含中断状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RXSTSIS	TXSTSIS	TSIS	Res.	MMCTXIS	MMCRXIS	MMCIS	Res.	Res.	LPIIS	PMTIS	PHYIS	Res.	Res.	RGSMIIIS
	rc_r	rc_r	rc_r		r	r	r			r	r	r			r

位 31:15 保留，必须保持复位值。

位 14 **RXSTSIS**：接收状态中断

此位指示所接收数据包的状态。将 *Rx Tx 状态寄存器 (ETH_MACRXTXSR)* 中的 RWT 位置 1 时，该位会置 1。在 ETH_MACISR 寄存器中读取相应的中断源位（或在 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 的 RCWE 位置 1 时向相应的中断源位写入 1）时，该位会清零。

位 13 **TXSTSIS**：发送状态中断

此位指示所发送数据包的状态。将 *Rx Tx 状态寄存器 (ETH_MACRXTXSR)* 中的以下任一位置 1 时，该位会置 1：

- 过度冲突 (EXCOL)
- 延迟冲突 (LCOL)
- 过度延迟 (EXDEF)
- 载波丢失 (LCARR)
- 无载波 (NCARR)
- Jabber 超时 (TJT)

在 ETH_MACISR 寄存器中读取相应的中断源位（或在 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 的 RCWE 位置 1 时向相应的中断源位写入 1）时，该位会清零。

位 12 **TSIS**：时间戳中断状态

如果使能了时间戳功能，则当以下任一条件为真时，该位会置 1：

- 系统时间值等于或超出目标时间高位和低位寄存器中指定的值。
- 秒寄存器发生上溢。
- 发生目标时间错误，即，编程的目标时间已结束。

如果使能辅助快照功能，则在将辅助快照触发信号置为有效时，该位会置 1。

如果在 MTL 中使能丢弃发送状态，则在 *Tx 时间戳状态纳秒寄存器 (ETH_MACTXTSSNR)* 和 *Tx 时间戳状态秒寄存器 (ETH_MACTXTSSSR)* 寄存器中更新捕获的发送时间戳时，该位会置 1。

如果使能 PTP 减荷功能，则在 *Tx 时间戳状态纳秒寄存器 (ETH_MACTXTSSNR)* 和 *Tx 时间戳状态秒寄存器 (ETH_MACTXTSSSR)* 寄存器中更新捕获的发送时间戳时，该位会置 1，以用于 PTO 生成的延迟请求和 Pdelay 请求数据包。

在*时间戳状态寄存器 (ETH_MACTSSR)* 中读取相应的中断源位（或在 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 的 RCWE 位置 1 时向相应的中断源位写入 1）时，此位将清零。

位 11 保留，必须保持复位值。

位 10 **MMCTXIS**：MMC 发送中断状态

当 *MMC Tx 中断寄存器 (ETH_MMC_TX_INTERRUPT)* 中生成中断时，此位就为高电平。当此中断寄存器中的所有位都清零时，此位将清零。

位 9 **MMCRXIS**：MMC 接收中断状态

当 *MMC Rx 中断寄存器 (ETH_MMC_RX_INTERRUPT)* 中生成中断时，此位就为高电平。当此中断寄存器中的所有位都清零时，此位将清零。

位 8 **MMCIS**：MMC 中断状态

当 MMCTXIS 或 MMCRXIS 置为高电平时，此位将置为高电平。仅当所有这些位均为低电平时，此位才会清零。

位 7:6 保留，必须保持复位值。

位 5 **LPIIS**：LPI 中断状态

系统将针对 MAC 发送器或接收器是进入还是退出 LPI 状态来设置此位。对 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 的 TLPIEN 位进行读取操作时，该位清零。

位 4 **PMTIS**：PMT 中断状态

在掉电模式下接收到魔术数据包或局域网唤醒数据包（ETH_MACPCSR 寄存器中的 RWKPRCVD 和 MGKPRCVD 位）时，该位置 1。当位 [6:5] 因对 *PMT 控制状态寄存器 (ETH_MACPCSR)* 执行读取操作而被清零时，此位也会清零。

位 3 **PHYIS**：PHY 中断

在 ETH_PHY_INTN 输入端检测到上升沿时，该位置 1。对此寄存器执行读取操作时此位清零。

位 2:1 保留，必须保持复位值。

位 0 **RGSMIIIS**：RGMII 中断状态

系统将根据 RGMII 接口链路状态值（*PHYIF 控制状态寄存器 (ETH_MACPHYCSR)* 寄存器中的 LNKSTS 位）的变化来设置此位。对 ETH_MACPHYCSR 寄存器执行读取操作时此位清零。

中断使能寄存器 (ETH_MACIER)

Interrupt enable register

偏移地址：0x00B4

复位值：0x0000 0000

中断使能寄存器包含用于生成中断的屏蔽位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RXSTSIE	TXSTSIE	TSIE	Res.	Res.	Res.	Res.	Res.	Res.	LPIIE	PMTIE	PHYIE	Res.	Res.	RGSMIIIE
	rw	rw	rw							rw	rw	rw			rw

位 31:15 保留，必须保持复位值。

位 14 **RXSTSIE**：接收状态中断使能 (Receive Status Interrupt Enable)
该位置 1 时，会因为 *中断状态寄存器 (ETH_MACISR)* 中的 RXSTSIS 位置 1 而将中断信号置为有效。

位 13 **TXSTSIE**：发送状态中断使能 (Transmit Status Interrupt Enable)
该位置 1 时，会因为 *中断状态寄存器 (ETH_MACISR)* 中的 TXSTSIS 位置 1 而将中断信号置为有效。

位 12 **TSIE**：使能时间戳中断
该位置 1 时，会因为 *中断状态寄存器 (ETH_MACISR)* 中的 TSIS 位置 1 而将中断信号置为有效。

位 11:6 保留，必须保持复位值。

位 5 **LPIIE**：LPI 中断使能 (LPI Interrupt Enable)
该位置 1 时，会因为 *中断状态寄存器 (ETH_MACISR)* 中的 LPIIS 位置 1 而将中断信号置为有效。

位 4 **PMTIE**：PMT 中断使能 (PMT Interrupt Enable)
该位置 1 时，会因为 *中断状态寄存器 (ETH_MACISR)* 中的 PMTIS 位置 1 而将中断信号置为有效。

位 3 **PHYIE**：PHY 中断使能 (PHY Interrupt Enable)
该位置 1 时，会因为 *中断状态寄存器 (ETH_MACISR)* 中的 PHYIS 位置 1 而将中断信号置为有效。

位 2:1 保留，必须保持复位值。

位 0 **RGSMIIIE**：RGMII 中断使能 (RGMII Interrupt Enable)
该位置 1 时，会因为 *中断状态寄存器 (ETH_MACISR)* 中的 RGSMIIIS 位置 1 而将中断信号置为有效。

Rx Tx 状态寄存器 (ETH_MACRXTXSR)

Rx Tx status register

偏移地址：0x00B8

复位值：0x0000 0000

接收发送状态寄存器包含接收和发送错误状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWT	Res.	Res.	EXCOL	LCOL	EXDEF	LCARR	NCARR	TJT
							rc_r			rc_r	rc_r	rc_r	rc_r	rc_r	rc_r

位 31:9 保留，必须保持复位值。

位 8 **RWT**：接收看门狗超时 (Receive Watchdog Timeout)

当接收到长度大于 2,048 字节（如果使能巨型数据包模式，则为 10,240 字节）的数据包，并且*工作模式配置寄存器 (ETH_MACCR)* 中的 WD 位复位时，该位置 1。当接收到长度大于 16,383 字节的数据包，并且*工作模式配置寄存器 (ETH_MACCR)* 中的 WD 位置 1 时，该位置 1。

此位会在读取时清零（或在将 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）。

位 7:6 保留，必须保持复位值。

位 5 **EXCOL**：过度冲突 (Excessive Collisions)

当*工作模式寄存器 (ETH_MTLOMR)* 中的 DTXSTS 位置 1 时，该位指示尝试发送当前数据包时因出现 16 个连续冲突而中止发送。如果*工作模式配置寄存器 (ETH_MACCR)* 中的 DR 位置 1，则该位在第一次冲突后置 1，并且数据包发送过程将中止。

此位会在读取时清零（或在将 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）。

位 4 **LCOL**：延迟冲突

当*工作模式寄存器 (ETH_MTLOMR)* 中的 DTXSTS 位置 1 时，该位指示因冲突窗口（MII 模式下为 64 个字节，含前导码；GMII 模式下为 512 个字节，含前导码和载波扩展）后出现冲突而中止数据包发送。

如果发生下溢错误，则该位无效。

此位会在读取时清零（或在将 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）。

位 3 **EXDEF**：过度延迟

当*工作模式寄存器 (ETH_MTLOMR)* 中的 DTXSTS 位置 1，且*工作模式配置寄存器 (ETH_MACCR)* 中的 DC 位置 1 时，该位指示因延迟超过 24,288 个位时间（在 1000 Mbps 模式下或在使能巨型数据包时，为 155,680）而结束发送。

此位会在读取时清零（或在将 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）。

位 2 **LCARR**：载波丢失

当*工作模式寄存器 (ETH_MTLOMR)* 中的 DTXSTS 位置 1 时，该位指示数据包发送期间丢失载波，即，数据包发送期间有一个或多个发送时钟周期的 ETH_CRS 信号无效。该位仅对未发生冲突的数据包有效。

此位会在读取时清零（或在将 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）。

位 1 **NCARR**：无载波

当*工作模式寄存器 (ETH_MTLOMR)* 中的 DTXSTS 位置 1 时，该位指示在前导码发送结束时不存在来自 PHY 的载波信号。

此位会在读取时清零（或在将 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）。

位 0 **TJT**：发送 Jabber 超时 (Transmit Jabber Timeout)

该位指示发送 Jabber 定时器已过期，在数据包大小超过 2,048 字节（使能巨型数据包时为 10,240 字节），并且*工作模式配置寄存器 (ETH_MACCR)* 中的 JD 位复位时，会出现这种情况。当数据包大小超过 16,383 字节并且*工作模式配置寄存器 (ETH_MACCR)* 中的 JD 位置 1 时，该位置 1。

此位会在读取时清零（或在将 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）。

PMT 控制状态寄存器 (ETH_MACPCSR)

PMT control status register

偏移地址：0x00C0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RWKFILTRST	Res.	Res.	RWKPTR[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw			r	r	r	r	r								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	RWKPFE	GLBLUCAST	Res.	Res.	RWKPRCVD	MGKPRCVD	Res.	Res.	RWKPKTEN	MGKPKTEN	PWRDWN
					rw	rw			r	rc_r			rw	rw	rw

位 31 **RWKFILTRST**：远程唤醒数据包过滤寄存器指针复位 (Remote wakeup Packet Filter Register Pointer Reset)

该位置 1 时，远程唤醒数据包过滤寄存器指针将复位为 0。它会在 1 个时钟周期后自动清零。

位 30:29 保留，必须保持复位值。

位 28:24 **RWKPTR[4:0]**：远程唤醒 FIFO 指针 (Remote wakeup FIFO Pointer)

该位字段给出了远程唤醒数据包过滤寄存器指针的当前值（0 到 7）。如果该指针的值等于 7，则对该寄存器进行写入操作时，远程唤醒数据包过滤寄存器的内容将被传输到 eth_mii_rx_clk 域。

位 23:11 保留，必须保持复位值。

位 10 **RWKPFE**：远程唤醒数据包转发使能 (Remote wakeup Packet Forwarding Enable)

该位与 RWKPKTEN 均置 1 时，MAC 接收器将丢弃所有接收到的帧，直到接收到预期唤醒帧。该事件之后的所有帧（包括接收到的唤醒帧）均被转发给应用程序。接收到唤醒数据包后，该位自动清零。

在接收到预期唤醒帧之前，应用程序也会将该位清零。在这种情况下，MAC 将恢复为默认行为，即，将接收到的数据包转发给应用程序。该位必须仅在 RWKPKTEN 置为高电平且 PWRDWN 置为低电平时置 1。当 PWRDWN 置为高电平时，该位的设置不起任何作用。

注意：如果魔术数据包使能和唤醒帧使能以及该位均置 1，并且在接收到唤醒帧之前接收到魔术数据包，则该位会在接收到魔术数据包时自动清零，接收到的魔术数据包被丢弃，接收到魔术数据包之后的所有帧均被转发给应用程序。

位 9 **GLBLUCAST**：全局单播 (Global Unicast)

当此位置 1 时，MAC (DAF) 地址识别所过滤的所有单播数据包均会被检测为远程唤醒数据包。

位 8:7 保留，必须保持复位值。

位 6 **RWKPRCVD**：接收到的远程唤醒数据包 (Remote wakeup Packet Received)

该位置 1 时，表示因接收到远程唤醒数据包而生成了电源管理事件。对此寄存器执行读取操作时此位清零。

位 5 **MGKPRCVD**：接收到魔术数据包 (Magic Packet Received)

该位置 1 时，表示因接收到魔术数据包而生成了电源管理事件。该位会在读取此寄存器时清零（或在 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）。

位 4:3 保留，必须保持复位值。

位 2 **RWKPKTEN**：远程唤醒数据包使能 (Remote wakeup Packet Enable)

如果该位置 1，则会在 MAC 接收到远程唤醒数据包时生成电源管理事件。

位 1 **MGKPKTEN**：魔术数据包使能 (Magic Packet Enable)

如果该位置 1，则会在 MAC 接收到魔术数据包时生成电源管理事件。

位 0 **PWRDWN**：掉电 (Power Down)

该位置 1 时，MAC 接收器将丢弃所有接收到的数据包，直到接收到预期魔术数据包或远程唤醒数据包。该位随后自动清零，并且将禁止掉电模式。在接收到预期魔术数据包或远程唤醒数据包之前，软件可将该位清零。此位清零后 MAC 接收到的数据包会被转发给应用程序。该位必须仅在魔术数据包使能、全局单播或远程唤醒数据包使能位置为高电平时置 1。

注意：在掉电模式下，可以门控关闭 CSR 时钟。但是，CSR 时钟门控关闭时，不能对该寄存器执行任何读取操作或写入操作。因此，软件不能将此位清零。

远程唤醒数据包过滤寄存器 (ETH_MACRWKPFR)

Remote wakeup packet filter register

偏移地址：0x00C4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MACRWKPFR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACRWKPFR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **MACRWKPFR[31:0]**：远程唤醒数据包过滤 (Remote wakeup packet filter)

如需了解有关寄存器内容和编程序列的详细信息，请参见*表 552*、*表 553* 和*表 554*。

地址为 0x00C4 的 ETH_MACRWKPFR 寄存器加载唤醒数据包过滤寄存器。

要在唤醒数据包过滤寄存器中加载值，必须写入整个寄存器 (ETH_MACRWKPFR)。通过分别针对 ETH_MACRWKPFR 值 0 到 7 在地址 (0x00C4) 中连续加载八个、十六个或三十二个寄存器值来加载 ETH_MACRWKPFR 寄存器。读取 ETH_MACRWKPFR 寄存器时也采用类似方式。以太网外设在 ETH_MACPCSR 寄存器的位 [26:24] 中更新 ETH_MACRWKPFR 寄存器当前指针值。

LPI 控制和状态寄存器 (ETH_MACLCSR)

LPI control and status register

偏移地址：0x00D0

复位值：0x0000 0000

LPI 控制和状态寄存器控制 LPI 功能并提供 LPI 中断状态。对此寄存器执行读取操作时状态位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPITCSE	LPITE	LPITXA	PLSEN	PLS	LPIEN
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	RLPIST	TLPIST	Res.	Res.	Res.	Res.	RLPIEX	RLPIEN	TLPIEX	TLPIEN
						r	r					r	r	r	r

位 31:22 保留，必须保持复位值。

位 21 **LPITCSE**：LPI Tx 时钟停止使能 (LPI Tx Clock Stop Enable)

该位置 1 时，MAC 在进入 Tx LPI 模式后会将 sbd_tx_clk_gating_ctrl_o 信号置为高电平，以指示可以停止向 MAC 提供 Tx 时钟。该位复位时，MAC 在进入 Tx LPI 模式后不会将 sbd_tx_clk_gating_ctrl_o 信号置为高电平。

如果选择了 RGMII 接口，则需要通过 Tx 时钟来发送 LPI 模式。无法对 Tx 时钟进行门控，因此无法对 LPITCSE 位进行编程。

位 20 **LPITE**：LPI 定时器使能 (LPI Timer Enable)

该位控制 MAC 发送器自动进入和退出 LPI 状态。如果 LPITE、LPITXA 和 LPIEN 位均置 1，则只有在整个 MAC TX 数据路径保持空闲的时长达到 ETH_MACLETR 寄存器所指示的时间后，MAC 发送器才会进入 LPI 状态。

进入 LPI 状态后，如果数据路径变为非空闲（由于接受了新的需要发送的数据包），则发送器会退出 LPI 状态，但不会将 LPIEN 位清零。这样一来，再次变为空闲时便可以重新返回 LPI 状态。

LPITE 为 0 时，LPI 自动定时器被禁止，MAC 发送器进入 LPI 状态（根据 LPITXA 和 LPIEN 位设置的说明）。

位 19 **LPITXA**：LPI Tx 自动化 (LPI Tx Automate)

该位控制 MAC 在发送端进入或退出 LPI 模式时的行为。

如果 LPITXA 和 LPIEN 位均置 1，则只有在已发送所有未完成的数据包（在内核中）和挂起数据包（在应用程序接口中）之后，MAC 才会进入 LPI 模式。应用程序发送任何数据包以供传送，或应用程序发出 Tx FIFO 刷新命令时，MAC 将退出 LPI 模式。此外，MAC 退出 LPI 状态时会自动将 LPIEN 位清零。当 MAC 处于 LPI 模式时，如果在 ETH_MTLTxQxOMR 的 FTQ 位中将 Tx FIFO 刷新置为有效，则 MAC 会退出 LPI 模式。

该位为 0 时，LPIEN 位直接控制 MAC 进入或退出 LPI 模式时的行为。

位 18 **PLSEN**：PHY 链路状态使能 (PHY Link Status Enable)

该位使能在 RGMII 接收路径上接收到的用于激活 LPI LS 定时器的链路状态。

当此位置 1 时，MAC 将使用 ETH_MACPHYCSR 寄存器的链路状态位和 PLS 位来触发 LPI LS 定时器。该位复位时，MAC 将忽略 ETH_MACPHYCSR 寄存器的链路状态位，并且仅使用 PLS 位。

位 17 **PLS**：PHY 链路状态 (PHY Link Status)

该位指示 PHY 的链路状态。只有在链路接通状态 (OKAY) 至少保持 LPI LS 定时器所指示的时间时，MAC 发送器才会将 LPI 模式置为有效。

该位置 1 时，将链路视为连通状态 (UP)；该位复位时，将链路视为断开状态。

位 16 **LPIEN**：LPI 使能 (LPI Enable)

该位置 1 时，会命令 MAC 发送器进入 LPI 状态。该位复位时，会命令 MAC 退出 LPI 状态并恢复正常发送。

当 LPITXA 位置 1 且由于新数据包到达等待发送而使 MAC 退出 LPI 状态时，该位清零。

位 15:10 保留，必须保持复位值。

位 9 **RLPIST**：接收 LPI 状态 (Receive LPI State)

该位置 1 时，表示 MAC 正在通过 GMII 或 MII 接口接收 LPI 模式。

位 8 **TLPIST**：发送 LPI 状态 (Transmit LPI State)

该位置 1 时，表示 MAC 正在通过 GMII 或 MII 接口发送 LPI 模式。

位 7:4 保留，必须保持复位值。

位 3 **RLPIEX**：接收 LPI 退出 (Receive LPI Exit)

该位置 1 时，表示 MAC 接收器已停止通过 GMII 或 MII 接口接收 LPI 模式，并且退出 LPI 状态和恢复正常接收。在读取此寄存器（或在 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）时，此位会清零。

注意：如果 MAC 在非常短的时间内（例如不到三个 CSR 时钟周期）停止接收 LPI 模式，则该位可能不会置 1。

位 2 **RLPIEN**：接收 LPI 进入 (Receive LPI Entry)

该位置 1 时，表示 MAC 接收器已接收到 LPI 模式并进入 LPI 状态。在读取此寄存器（或在 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）时，此位会清零。

注意：如果 MAC 在非常短的时间内（例如不到三个 CSR 时钟周期）停止接收 LPI 模式，则该位可能不会置 1。

位 1 **TLPIEX**：发送 LPI 退出 (Transmit LPI Exit)

该位置 1 时，表示应用程序将 LPIEN 位清零，且 LPI TW 定时器到期后，MAC 发送器退出了 LPI 状态。在读取此寄存器（或在 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）时，此位会清零。

位 0 **TLPIEN**：发送 LPI 进入 (Transmit LPI Entry)

该位置 1 时，表示由于 LPIEN 位置 1 而使 MAC 发送器进入了 LPI 状态。在读取此寄存器（或在 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）时，此位会清零。

LPI 定时器控制寄存器 (ETH_MACLTCR)

LPI timers control register

偏移地址：0x00D4

复位值：0x03E8 0000

LPI 定时器控制寄存器控制 LPI 状态下的超时值。它指定 MAC 发送 LPI 模式的时间以及 MAC 在恢复正常发送之前等待的时间。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	LST[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TWT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25:16 **LST[9:0]**：LPI LS 定时器 (LPI LS Timer)

该位字段指定在可以将 LPI 模式发送到 PHY 之前，来自 PHY 的链路状态 (OKAY) 应持续的最短时间（以毫秒为单位）。即使 LPIEN 位置 1，MAC 也不会发送 LPI 模式，除非 LPI LS 定时器达到编程的端子计数。LPI LS 定时器的默认值是 1000（1 秒），如 IEEE 标准中所定义。

位 15:0 **TWT[15:0]**：LPI TW 定时器 (LPI TW Timer)

该位字段指定 MAC 停止向 PHY 发送 LPI 模式之后至少需等待多长时间才能恢复正常发送（以微秒为单位）。TLPIEX 状态位在该定时器到期后置 1。

LPI 进入定时器寄存器 (ETH_MACLETR)

LPI entry timer register

偏移地址：0x00D8

复位值：0x0000 0000

此寄存器将控制 Tx LPI 进入定时器。只有当 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 寄存器的 LPITE 位置 1 时，才会使能此计数器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPIET[19:16]			
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPIET[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	r

位 31:20 保留，必须保持复位值。

位 19:0 **LPIET[19:0]**：LPI 进入定时器 (LPI Entry Timer)

此位字段指定 MAC 发送完所有帧后需等待多长时间才能进入 LPI 模式（以微秒为单位）。只有在 LPITE 和 LPITXA 均置 1 时，该位字段才有效且可用。

位 [2:0] 为只读位，因此该定时器的粒度以 8 微秒为步长。

一微秒节拍计数器寄存器 (ETH_MAC1USTCR)

One-microsecond-tick counter register

偏移地址：0x00DC

复位值：0x0000 0000

该寄存器控制所有 LPI 定时器的参考时间（一微秒节拍）的生成。该定时器最初必须由软件编程。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TIC_1US_CNTR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:0 TIC_1US_CNTR[11:0]：1 µs 节拍计数器 (1 µs tick Counter)

应用程序必须对该计数器进行编程，以使 CSR 时钟的时钟周期数为 1 µs（从编程前的值中减去 1）。

例如，如果 CSR 时钟为 100 MHz，则需要将该位字段编程为 100 - 1 = 99（即 0x63）。

要生成用于更新某些 EEE 相关计数器的 1 µs 事件，此项必不可少。

PHYIF 控制状态寄存器 (ETH_MACPHYCSR)

PHYIF control status register

偏移地址：0x00F8

复位值：0x0000 0000

PHY 接口控制和状态寄存器将指示 RGMII 接口从 PHY 接收到的状态信号。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LNKSTS	LNKSPEED[1:0]		LNKMOD
												r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LUD	TC
														rw	rw

位 31:20 保留，必须保持复位值。

位 19 **LNKSTS**：链路状态 (Link Status)

此位指示链路是连通还是断开。

0：链路断开

1：链路连通

位 18:17 **LNKSPEED[1:0]**：链路速度 (Link Speed)

此位指示链路的当前速度：

00：2.5 MHz

01：25 MHz

10：125 MHz

11：保留，不得使用

位 16 **LNKMOD**：链路模式 (Link Mode)

此位指示链路当前的工作模式：

0：半双工模式

1：全双工模式

位 15:2 保留，必须保持复位值。

位 1 **LUD**：链路连通或断开 (Link Up or Down)

此位指示在 RGMII 接口中发送配置期间链路是连通还是断开。

0：链路断开

1：链路连通

位 0 **TC**：在 RGMII 中发送配置

该位置 1 时，表示允许向 RGMII 端口中的 PHY 发送双工模式、链路速度以及链路连通或断开信息。当此位复位时，系统不会将此类信息驱动至 PHY（请参见*第 57.6.3 节：精简介质独立接口 (RMII)*）。

版本寄存器 (ETH_MACVR)

Version register

偏移地址：0x0110

复位值：0x0000 1242

版本寄存器标识以太网外设的版本。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERVER[7:0]								SNPSVER[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:8 **USERVER[7:0]**：ST 定义的版本 (ST-defined version)

位 7:0 **SNPSVER[7:0]**：IP 版本

调试寄存器 (ETH_MACDR)

Debug register

偏移地址：0x0114

复位值：0x0000 0000

调试寄存器提供各个 MAC 模块的调试状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TFCSTS[1:0]		TPESTS
													r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFCFCSTS[1:0]		RPESTS
													r	r	r

位 31:19 保留，必须保持复位值。

位 18:17 **TFCSTS[1:0]**：MAC 发送数据包控制器状态 (MAC Transmit Packet Controller Status)

此位字段指示 MAC 发送数据包控制器模块的状态：

00：空闲状态 (Idle state)

01：等待下列其中一项：

– 前一个数据包的状态

– IPG 或回退阶段结束

10：生成并发送暂停控制数据包（在全双工模式下）

11：传输要发送的输入数据包

位 16 **TPESTS**：MAC GMII 或 MII 发送协议引擎状态 (MAC GMII or MII Transmit Protocol Engine Status)

该位置 1 时，表示 MAC GMII 或 MII 发送协议引擎正在主动发送数据，而未处于空闲状态。

位 15:3 保留，必须保持复位值。

位 2:1 **RFCFCSTS[1:0]**：MAC 接收数据包控制器 FIFO 状态 (MAC Receive Packet Controller FIFO Status)

该位字段置 1 时，表示 MAC 接收数据包控制器模块的各个小 FIFO 读和写控制器的有效状态。

位 0 **RPESTS**：MAC GMII 或 MII 接收协议引擎状态 (MAC GMII or MII Receive Protocol Engine Status)

该位置 1 时，表示 MAC GMII 或 MII 接收协议引擎正在主动接收数据，而未处于空闲状态。

硬件功能 0 寄存器 (ETH_MACHWF0R)

HW feature 0 register

偏移地址：0x011C

复位值：0x0A0D 73F7

该寄存器指示存在第一组可选的以太网外设功能。软件驱动程序可使用该寄存器来动态使能或禁止与可选模块相关的程序。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	ACTPHYSEL[2:0]			SAVLANINS	TSSTSSEL[1:0]		MACADR64SEL	MACADR32SEL	ADDMACADRSEL[4:0]					Res.	RXCOESEL
	r	r	r	r	r	r	r	r	r	r	r	r	r		r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TXCOESEL	EEESEL	TSSEL	Res.	Res.	ARPOFFSEL	MMCSEL	MGKSEL	RWKSEL	SMASEL	VLHASH	PCSSEL	HDSEL	GMIISEL	MIISEL
	r	r	r			r	r	r	r	r	r	r	r	r	r

位 31 保留，必须保持复位值。

位 30:28 **ACTPHYSEL[2:0]**：已选择激活的 PHY (Active PHY Selected)

当用户配置中存在多个 PHY 接口时，此位字段将指示复位置为无效期间 phy_intf_sel_i 的采样值：

000：GMII 或 MII

001：RGMII

010：SGMII

011：TBI

100：RMII

101：RTBI

110：SMII

其他值：保留，不得使用

位 27 **SAVLANINS**：源地址或 VLAN 插入使能 (Source Address or VLAN Insertion Enable)

选择 Enable SA and VLAN Insertion on Tx（在 Tx 端使能 SA 和 VLAN 插入）选项时，该位置 1

位 26:25 **TSSTSSEL[1:0]**：时间戳系统时间源 (Timestamp System Time Source)

此位指示时间戳系统时间的源：

01：内部

10：外部

11：二者兼有

00：保留，不得使用

当选择了 Enable IEEE 1588 Timestamp Support（使能 IEEE 1588 时间戳支持）选项时，此位将置 1。

位 24 **MACADR64SEL**：已选择 MAC 地址 64-127 (MAC Addresses 64-127 Selected)

当选择了 Enable Additional 64 MAC Address Registers (64-127)（使能额外的 64 个 MAC 地址寄存器 (64-127)）选项时，此位将置 1

位 23 **MACADR32SEL**：已选择 MAC 地址 32-63 (MAC Addresses 32-63 Selected)

当选择了 Enable Additional 32 MAC Address Registers (32-63)（使能额外的 32 个 MAC 地址寄存器 (32-63)）选项时，此位将置 1

位 22:18 **ADDMACADRSEL[4:0]**：已选择 MAC 地址 1-31 (MAC Addresses 1-31 Selected)

当选择了 Enable Additional 1-31 MAC Address Registers（使能额外的 1-31 个 MAC 地址寄存器）选项时，此位将置 1

位 17 保留，必须保持复位值。

位 16 **RXCOESEL**：接收校验和减荷使能 (Receive Checksum Offload Enabled)

当选择了 Enable Receive TCP/IP Checksum Check（使能接收 TCP/IP 校验和检查）选项时，此位将置 1

位 15 保留，必须保持复位值。

位 14 **TXCOESEL**：发送校验和减荷使能 (Transmit Checksum Offload Enabled)

当选择了 Enable Transmit TCP/IP Checksum Insertion（使能发送 TCP/IP 校验和插入）选项时，此位将置 1

位 13 **EEESEL**：节能型以太网使能 (Energy Efficient Ethernet Enabled)

当选择了 Enable Energy Efficient Ethernet (EEE)（使能节能型以太网 (EEE)）选项时，该位置 1

位 12 **TSSEL**：IEEE 1588-2008 时间戳使能 (IEEE 1588-2008 Timestamp Enabled)

当选择了 Enable IEEE 1588 Timestamp Support（使能 IEEE 1588 时间戳支持）选项时，此位将置 1。

位 11:10 保留，必须保持复位值。

位 9 **ARPOFFSEL**：ARP 减荷使能 (ARP Offload Enabled)

选择 Enable IPv4 ARP Offload（使能 IPv4 ARP 减荷）选项时，该位置 1

位 8 **MMCSEL**：RMON 模块使能 (RMON Module Enable)
当选择了 Enable MAC Management Counters (MMC)（使能 MAC 管理计数器 (MMC)）选项时，该位置 1

位 7 **MGKSEL**：PMT 魔术数据包使能 (PMT Magic Packet Enable)
选择 Enable Magic Packet Detection（使能魔术数据包检测）选项时，此位将置 1

位 6 **RWKSEL**：PMT 远程唤醒数据包使能 (PMT Remote Wakeup Packet Enable)
选择 Enable Remote wakeup Packet Detection（使能远程唤醒数据包检测）选项时，此位将置 1

位 5 **SMASEL**：SMA (MDIO) 接口 (SMA (MDIO) Interface)
当选择了 Enable Station Management (MDIO Interface)（使能站管理（MDIO 接口））选项时，此位将置 1。

位 4 **VLHASH**：已选择 VLAN 散列过滤器 (VLAN Hash Filter Selected)
当选择了 Enable VLAN Hash Table Based Filtering（使能基于 VLAN 散列表的过滤）选项时，此位将置 1。

位 3 **PCSSEL**：PCS 寄存器（TBI、SGMII 或 RTBI PHY 接口）(PCS Registers (TBI, SGMII, or RTBI PHY interface))
当选择了 TBI, SGMII, or RTBI PHY interface（TBI、SGMII 或 RTBI PHY 接口）选项时，此位将置 1

位 2 **HDSEL**：半双工支持 (Half-duplex Support)
当选择了半双工模式时，此位将置 1

位 1 **GMIISEL**：1000 Mbps 支持 (1000 Mbps Support)
当选择了 1000 Mbps 作为工作模式时，此位将置 1。

位 0 **MIISEL**：10 或 100 Mbps 支持 (10 or 100 Mbps Support)
当选择了 10/100 Mbps 作为工作模式时，此位将置 1。

硬件功能 1 寄存器 (ETH_MACHWF1R)

HW feature 1 register

偏移地址：0x0120

复位值：0x1104 1904

该寄存器指示存在第二组可选的以太网外设功能。软件驱动程序可使用该寄存器来动态使能或禁止与可选模块相关的程序。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	L3L4FNUM[3:0]				Res.	HASHTBLSZ[1:0]		POUOST	Res.	RAVSEL	AVSEL	DBGMEMA	TSOEN	SPHEN	DCBEN
	r	r	r	r		r	r	r		r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR64[1:0]		ADVTHWORD	PTOEN	OSTEN	TXFIFOSIZE[4:0]					Res.	RXFIFOSIZE[4:0]				
r	r	r	r	r	r	r	r	r	r		r	r	r	r	r

位 31 保留，必须保持复位值。

位 30:27 **L3L4FNUM[3:0]**：L3 或 L4 过滤器总数 (Total number of L3 or L4 Filters)
该位字段指示 L3 或 L4 过滤器的总数：
0000：无 L3 或 L4 过滤器
0001：1 个 L3 或 L4 过滤器
0010：2 个 L3 或 L4 过滤器
..
1000：8 个 L3 或 L4 过滤器

位 26 保留，必须保持复位值。

位 25:24 **HASHTBLSZ[1:0]**：散列表大小 (Hash Table Size)
该位字段指示散列表的大小：
00：无散列表
01：64
10：128
11：256

位 23 **POUOST**：针对 PTP over UDP/IP 的单步时间戳功能使能 (One Step for PTP over UDP/IP Feature Enable)
当选择了 Enable one step timestamp for PTP over UDP/IP（使能针对 PTP over UDP/IP 的单步时间戳）功能时，此位将置 1。

位 22 保留，必须保持复位值。

位 21 **RAVSEL**：仅 Rx 端 AV 功能使能 (Rx Side Only AV Feature Enable)
当仅选择了 Rx 端的 Enable Audio Video Bridging（使能音视频桥接）选项时，此位将置 1。

位 20 **AVSEL**：AV 功能使能 (AV Feature Enable)
选择 Enable Audio Video Bridging（使能音视频桥接）选项时，该位置 1。

位 19 **DBGMEMA**：DMA 调试寄存器使能 (DMA Debug Registers Enable)
选择 Debug Mode Enable（调试模式使能）选项时，该位置 1

位 18 **TSOEN**：TCP 分段减荷使能 (TCP Segmentation Offload Enable)
选择 Enable TCP Segmentation Offloading for TCP/IP Packets（使能 TCP/IP 数据包的 TCP 分段减荷）选项时，该位置 1

位 17 **SPHEN**：拆分报头功能使能 (Split Header Feature Enable)
选择 Enable Split Header Structure（使能拆分报头结构）选项时，该位置 1

位 16 **DCBEN**：DCB 功能使能 (DCB Feature Enable)
选择 Enable Data Center Bridging（使能数据中心桥接）选项时，该位置 1

位 15:14 **ADDR64[1:0]**：地址宽度 (Address width)
此位字段指示所配置的地址宽度。
00：32 位
其他值：保留，不得使用

位 13 **ADVTHWORD**：IEEE 1588 高位字寄存器使能 (IEEE 1588 High Word Register Enable)
选择 Add IEEE 1588 Higher Word Register（添加 IEEE 1588 高位字寄存器）选项时，该位置 1

位 12 **PTOEN**：PTP 减荷使能 (PTP Offload Enable)
选择 Enable PTP Timestamp Offload Feature（使能 PTP 时间戳减荷功能）时，该位置 1。

位 11 **OSTEN**：单步时间戳使能 (One-Step Timestamping Enable)
选择 Enable One-Step Timestamp Feature（使能单步时间戳功能）时，该位置 1。

位 10:6 **TXFIFOSIZE[4:0]**：MTL 发送 FIFO 大小 (MTL Transmit FIFO Size)

该位字段包含 MTL Tx FIFO 的配置值（以字节表示），表示为以 2 为底的对数减去 7，即 Log2(TXFIFO_SIZE) -7：

00000：128 字节
00001：256 字节
00010：512 字节
00011：1,024 字节
00100：2,048 字节
00101：4,096 字节
00110：8,192 字节
00111：16,384 字节
01000：32 KB
01001：64 KB
01010：128 KB
01011 到 11111：保留，不得使用

位 5 保留，必须保持复位值。

位 4:0 **RXFIFOSIZE[4:0]**：MTL 接收 FIFO 大小 (MTL Receive FIFO Size)

该位字段包含 MTL Rx FIFO 的配置值（以字节表示），表示为以 2 为底的对数减去 7，即 Log2(RXFIFO_SIZE) -7：

00000：128 字节
00001：256 字节
00010：512 字节
00011：1,024 字节
00100：2,048 字节
00101：4,096 字节
00110：8,192 字节
00111：16,384 字节
01000：32 KB
01001：64 KB
01010：128 KB
01011：256 KB
01100 到 11111：保留，不得使用

硬件功能 2 寄存器 (ETH_MACHWF2R)

HW feature 2 register

偏移地址：0x0124

复位值：0x4104 0041

该寄存器指示存在第三组可选的以太网外设功能。软件驱动程序可使用该寄存器来动态使能或禁止与可选模块相关的程序。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AUXSNAPNUM[2:0]			Res.	PPSOUTNUM[2:0]			TDCSZ[1:0]		TXCHCNT[3:0]				RDCSZ[1:0]	
	r	r	r		r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXCHCNT[3:0]				Res.	Res.	TXQCNT[3:0]				Res.	Res.	RXQCNT[3:0]			
r	r	r	r			r	r	r	r			r	r	r	r

位 31 保留，必须保持复位值。

位 30:28 **AUXSNAPNUM[2:0]**：辅助快照输入数 (Number of Auxiliary Snapshot Inputs)

此位字段指示辅助快照输入数：
000：无辅助输入
001：1 个辅助输入
010：2 个辅助输入
011：3 个辅助输入
100：4 个辅助输入
101 到 111：保留，不得使用

位 27 保留，必须保持复位值。

位 26:24 **PPSOUTNUM[2:0]**：PPS 输出数 (Number of PPS Outputs)

此位字段指示 PPS 输出数：
000：无 PPS 输出
001：1 个 PPS 输出
010：2 个 PPS 输出
011：3 个 PPS 输出
100：4 个 PPS 输出
101 到 111：保留，不得使用

位 23:22 **TDCSZ[1:0]**：以 16 字节描述符为单位的 Tx DMA 描述符缓存大小 (Tx DMA Descriptor Cache Size in terms of 16-byte descriptors)

00：未配置缓存
01：四个 16 字节描述符
10：八个 16 字节描述符
11：十六个 16 字节描述符

位 21:18 **TXCHCNT[3:0]**：DMA 发送通道数 (Number of DMA Transmit Channels)

此位字段指示 DMA 发送通道数：
0000：1 个 DMA Tx 通道
0001：2 个 DMA Tx 通道
..
0111：8 个 DMA Tx 通道

位 17:16 **RDCSZ[1:0]**：以 16 字节描述符为单位的 Rx DMA 描述符缓存大小 (Rx DMA Descriptor Cache Size in terms of 16-byte descriptors)

00：未配置缓存
01：四个 16 字节描述符
10：八个 16 字节描述符
11：十六个 16 字节描述符

位 15:12 **RXCHCNT[3:0]**：DMA 接收通道数 (Number of DMA Receive Channels)

该位字段指示 DMA 接收通道数：
0000：1 个 DMA Rx 通道
0001：2 个 DMA Rx 通道
..
0111：8 个 DMA Rx 通道

位 11:10 保留，必须保持复位值。

位 9:6 **TXQCNT[3:0]**：MTL 发送队列数 (Number of MTL Transmit Queues)

该位字段指示 MTL 发送队列数：
0000：1 个 MTL Tx 队列
0001：2 个 MTL Tx 队列
..
0111：8 个 MTL Tx 队列

位 5:4 保留，必须保持复位值。

位 3:0 **RXQCNT[3:0]**：MTL 接收队列数 (Number of MTL Receive Queues)

此位字段指示 MTL 接收队列数：
0000：1 个 MTL Rx 队列
0001：2 个 MTL Rx 队列
..
0111：8 个 MTL Rx 队列

硬件功能 3 寄存器 (ETH_MACHWF3R)

HW feature 3 register

偏移地址：0x0128

复位值：0x0000 0020

该寄存器指示存在第四组可选的以太网外设功能。软件驱动程序可使用该寄存器来动态使能或禁止与可选模块相关的程序。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DVLAN	CBTISEL	Res.	NRVF[2:0]		
										r	r		r	r	r

位 31:6 保留，必须保持复位值。

位 5 **DVLAN**：双 VLAN 处理使能 (Double VLAN processing enable)
当使能双 VLAN 处理时，此位将置 1。

位 4 **CBTISEL**：在 Tx 端使能基于队列/通道的 VLAN 标记插入 (Queue/Channel based VLAN tag insertion on Tx enable)
当选择了 Enable Queue/Channel based VLAN tag insertion on Tx（在 Tx 端使能基于队列/通道的 VLAN 标记插入）功能时，此位将置 1。

位 3 保留，必须保持复位值。

位 2:0 **NRVF[2:0]**：扩展 VLAN 标记过滤器数量使能 (Number of Extended VLAN Tag Filters Enabled)
此位字段指示所选择的扩展 VLAN 标记过滤器数量：
000：无扩展 Rx VLAN 过滤器
001：4 个扩展 Rx VLAN 过滤器
010：8 个扩展 Rx VLAN 过滤器
011：16 个扩展 Rx VLAN 过滤器
100：24 个扩展 Rx VLAN 过滤器
101：32 个扩展 Rx VLAN 过滤器
110 到 111：保留，不得使用

MDIO 地址寄存器 (ETH_MACMDIOAR)

MDIO address register

偏移地址：0x0200

复位值：0x0000 0000

MDIO 地址寄存器通过管理接口控制外部 PHY 的管理周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	PSE	BTB	PA[4:0]					RDA[4:0]				
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	NTC[2:0]			CR[3:0]				Res	Res	Res	SKAP	GOC[1:0]		C45E	GB
	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27 **PSE**：前导码抑制使能 (Preamble Suppression Enable)

该位置 1 时，SMA 将抑制 32 位前导码，并发送仅包含 1 个前导码位的 MDIO 帧。

该位为 0 时，MDIO 帧始终具有如 IEEE 规范中定义的 32 位前导码。

位 26 **BTB**：连续事务 (Back to Back transactions)

该位置 1 且 NTC 的值大于 0 时，MAC 将在帧传输结束时（在发送尾部时钟之前）发出已完成读取命令或写入命令的通知。这样，软件便可启动下一条命令，而且无论为前一帧生成的尾部时钟数为多少，都会立即执行此命令。

该位复位时，只有在生成尾部时钟之后，才会完成读/写命令（GMII 忙碌位清零）。在该模式下，确保始终在每个帧之后生成 NTC。

NTC=0 时，该位不能置 1。

位 25:21 **PA[4:0]**：物理层地址 (Physical Layer Address)

该位字段指示 MAC 正在访问哪些符合条款 22 的 PHY 设备（32 个 PHY 设备中）。该位字段指示 MAC 正在访问哪些符合条款 45 的 PHY 设备（32 个 PHY 设备中）。

位 20:16 **RDA[4:0]**：寄存器/设备地址 (Register/Device Address)

这些位用于选择所选符合条款 22 的 PHY 设备中的 PHY 寄存器。这些位用于选择所选符合条款 45 的 PHY 的设备 (MMD)。

位 15 保留，必须保持复位值。

位 14:12 **NTC[2:0]**：尾部时钟数 (Number of Trailing Clocks)

该位字段控制 MDIO 帧发送结束后在 ETH_MDC 上生成的尾部时钟周期数。有效值范围为 0 到 7。将值编程为 011 表示 MDIO 帧完成传输后 MDC 线上存在三个额外时钟周期。

位 11:8 **CR[3:0]**：CSR 时钟范围 (CSR Clock Range)

CSR 时钟范围选择根据 CSR 时钟频率 (eth_hclk) 确定 MDC 时钟的频率：
0000：MDC 时钟 = eth_hclk / 42
0001：MDC 时钟 = eth_hclk / 62
0010：MDC 时钟 = eth_hclk / 16
0011：MDC 时钟 = eth_hclk / 26
0100：MDC 时钟 = eth_hclk / 102
0101：MDC 时钟 = eth_hclk / 124
0110 到 0111：保留，不得使用
适用于各个值的建议 CSR 时钟频率范围（位 11 = 0 时）可确保 MDC 时钟大致介于 1.0 MHz 到 2.5 MHz 频率范围之间。
位 11 置 1 时，可以实现高于 2.5 MHz 频率限制（在 IEEE 802.3 中规定）的 MDC 时钟频率，并编程较低值的时钟分频器。例如，当 CSR 时钟频率为 100 MHz，并且将这些位编程为 1010 时，所得到的 MDC 时钟为 12.5 MHz，此值高于 IEEE 802.3 中规定的范围。仅当接口芯片支持更快的 MDC 时钟时才编程以下值：
1000：eth_hclk / 4
1001：eth_hclk / 6
1010：eth_hclk / 8
1011：eth_hclk / 10
1100：eth_hclk / 12
1101：eth_hclk / 14
1110：eth_hclk / 16
1111：eth_hclk / 18

位 7:5 保留，必须保持复位值。

位 4 **SKAP**：跳过地址数据包 (Skip Address Packet)

该位置 1 时，SMA 在对增量地址数据包进行读取、写入或后读取操作之前不会发送地址数据包。只有在 C45E 置 1 时，该位才有效。

位 3:2 **GOC[1:0]**：GMII 操作命令 (GMII Operation Command)

该位指示 PHY 的操作命令。
00：保留，不得使用
01：写
10：对符合条款 45 的 PHY 的增量地址进行后读取操作
11：读
使能符合条款 22 的 PHY 时，只有读取命令和写入命令有效。

位 1 **C45E**：符合条款 45 的 PHY 使能 (Clause 45 PHY Enable)

该位置 1 时，符合条款 45 的 PHY 连接到 MDIO。该位复位时，符合条款 22 的 PHY 连接到 MDIO。

位 0 **GB**：GMII 忙碌 (GMII Busy)

应用程序将该位置 1 可命令 SMA 启动
对 MDIOS 的读访问或写访问。MAC 会在 MDIO 帧
传输完成后将该位清零。因此，只要该位置 1，软件就不能
写入或更改 *MDIO 地址寄存器 (ETH_MACMDIOAR)* 和 *MDIO 数据寄存器 (ETH_MACMDIODR)* 中的任何位字段。
对于写传输，应用程序必须在将该位置 1 前，在 *MDIO 数据寄存器 (ETH_MACMDIODR)* 寄存器的 GD 位字段（C45E 置 1 时，还包括 RA 位字段）中写入 16 位数据，C45E 置 1 时，还应在启动读传输之前，写入 *MDIO 数据寄存器 (ETH_MACMDIODR)* 的 RA 位字段。当读传输完成（GMII 忙碌=0 时），从 PHY 寄存器读取的数据将在 *MDIO 数据寄存器 (ETH_MACMDIODR)* 的 GD 位字段中处于有效状态。

注意：即使不存在寻址的 PHY，
该位的功能也不会改变。

MDIO 数据寄存器 (ETH_MACMDIODR)

MDIO data register

偏移地址：0x0204

复位值：0x0000 0000

MDIO 数据寄存器既存储要写入 PHY 寄存器（该寄存器的地址在 *MDIO 地址寄存器 (ETH_MACMDIOAR)* 中指定）的数据，也存储从 PHY 寄存器（该寄存器的地址由 MDIO 地址寄存器指定）读取的数据。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **RA[15:0]**：寄存器地址

只有在 C45E 置 1 时，该位字段才有效。它包含要使用 MDIO 帧的 PHY 中的寄存器地址。

位 15:0 **GD[15:0]**：GMII 数据 (GMII Data)

该位字段包含在某次管理读取操作之后从 PHY 中读取的 16 位数据值，或在某次管理写入操作之前要写入 PHY 的 16 位数据值。

ARP 地址寄存器 (ETH_MACARPAR)

ARP address register

偏移地址：0x0210

复位值：0x0000 0000

ARP 地址寄存器包含 MAC 的 IPv4 目标地址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARPPA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARPPA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **ARPPA[31:0]**：ARP 协议地址 (ARP Protocol Address)

此位字段包含 MAC 的 IPv4 目标地址。该地址用于与接收到的 ARP 数据包中的目标协议地址位字段进行完美匹配。

CSR 软件控制寄存器 (ETH_MACCSRSWCR)

CSR software control register

偏移地址：0x0230

复位值：0x0000 0000

此寄存器包含可用软件编程的控件，用于更改 CSR 访问响应和状态位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RCWE
							rw								rw

位 31:9 保留，必须保持复位值。

位 8 **SEEN**：从错误响应使能 (Slave Error Response Enable)
当此位置 1 时，MAC 会以从错误响应访问 CSR 空间中保留寄存器的操作。
当此位复位时，MAC 会向从 CSR 空间进行访问的任何寄存器发出 Okay 响应。

位 7:1 保留，必须保持复位值。

位 0 **RCWE**：寄存器写入 1 时清零使能 (Register Clear on Write 1 Enable)
当此位置 1 时，某些寄存器位字段的访问模式将变为 rc_w1（写入时清零），意味着应用程序需要将相应位置 1 以将其清零。
当此位复位时，这些寄存器位字段的访问模式将保持为 rc_r（读取时清零）。

MAC 地址 0 高位寄存器 (ETH_MACA0HR)

MAC Address 0 high register

偏移地址：0x0300

复位值：0x8000 FFFF

MAC 地址 0 高位寄存器可保存站的第一个 6 字节 MAC 地址的高 16 位。在 GMII 接口上接收到的第一个 DA 字节与 MAC 地址低位寄存器的 LS 字节（位 [7:0]）相对应。例如，如果在 GMII 上接收到 0x112233445566（第一列通道 0 中的 0x11）作为目标地址，则 MAC 地址 0 寄存器 [47:0] 会与 0x665544332211 进行比较。

如果将 MAC 地址寄存器配置为与 GMII 时钟域双同步，则只有在写入 MAC 地址 0 低位寄存器的位 [31:24]（小端模式）或位 [7:0]（大端模式）时，才会触发同步。为了正确进行同步更新，应在目标时钟域中的至少四个时钟周期之后，对该地址低位寄存器执行连续写入操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRHI[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **AE**：地址使能 (Address Enable)
该位始终置 1。

位 30:16 保留，必须保持复位值。

位 15:0 **ADDRHI[15:0]**：MAC 地址 0 [47:32] (MAC Address0[47:32])
此位字段包含第一个 6 字节 MAC 地址的高 16 位 [47:32]。MAC 使用此位字段过滤所接收到的数据包，以及将 MAC 地址插入到发送流控制（暂停）数据包中。

MAC 地址 x 低位寄存器 (ETH_MACAxLR)

MAC Address x low register

偏移地址：0x0304 + 0x8 * x，（x = 0 到 3）

复位值：0xFFFF FFFF

MAC 地址 x 低位寄存器可保存站的第一个 6 字节 MAC 地址的低 32 位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRLO[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRLO[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **ADDRLO[31:0]**：MAC 地址 x [31:0] (MAC Address x [31:0])（x = 0 到 3）

此位字段包含第一个 6 字节 MAC 地址的低 32 位。MAC 使用此位字段过滤所接收到的数据包，以及将 MAC 地址插入到发送流控制（暂停）数据包中。

MAC 地址 x 高位寄存器 (ETH_MACAxHR)

MAC Address x high register

偏移地址：0x0308 + 0x8 * (x-1)，（x = 1 到 3）

复位值：0x0000 FFFF

MAC 地址 x 高位寄存器可保存站的第二个 6 字节 MAC 地址的高 16 位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AE	SA	MBC[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRHI[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **AE**：地址使能 (Address Enable)

此位置 1 时，地址过滤器模块使用第二个 MAC 地址实现完美过滤。此位复位时，地址过滤器模块会忽略用于过滤的地址。

位 30 **SA**：源地址 (Source Address)

此位置 1 时，将使用 MAC 地址 x[47:0] 与所接收数据包的 SA 位字段进行比较。此位复位时，将使用 MAC 地址 x[47:0] 与所接收数据包的 DA 位字段进行比较。

0：DA

1：SA

位 29:24 **MBC[5:0]**：屏蔽字节控制 (Mask Byte Control)

这些位是用于比较每个 MAC 地址字节的屏蔽控制位。当将其设为高电平时，MAC 不会将所接收到的 DA 或 SA 的相应字节与 MAC 地址 1 寄存器的内容进行比较。每个位都用于控制字节的屏蔽，如下所示：

位 29：ETH_MACAxHR[15:8]
位 28：ETH_MACAxHR[7:0]
位 27：ETH_MACAxLR[31:24]
位 26：ETH_MACAxLR[23:16]
位 25：ETH_MACAxLR[15:8]
位 24：ETH_MACAxLR[7:0]

可通过屏蔽地址的一个或多个字节来过滤一组地址（被称为组地址过滤）。

位 23:16 保留，必须保持复位值。

位 15:0 **ADDRHI[15:0]**：MAC 地址 1 [47:32] (MAC Address1 [47:32])

此位字段包含第二个 6 字节 MAC 地址的高 16 位 [47:32]。

MMC 控制寄存器 (ETH_MMC_CONTROL)

MMC control register

偏移地址：0x0700

复位值：0x0000 0000

该寄存器配置 MMC 工作模式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	UCDBC	Res.	Res.	CNTPRSTLVL	CNTPRST	CNTFREEZ	RSTONRD	CNTSTOPRO	CNTRST
							rw			rw	rw	rw	rw	rw	rw

位 31:9 保留，必须保持复位值。

位 8 **UCDBC**：针对所丢弃广播数据包更新 MMC 计数器 (Update MMC Counters for Dropped Broadcast Packets)

CNTRST 位的优先级高于 CNTPRST 位。因此，软件尝试在同一写周期内将这两个位置 1 时，所有计数器都将清零，并且 CNTPRST 位不会置 1。

置 1 时，MAC 会针对因*数据包过滤控制寄存器 (ETH_MACPFR)* 的 DBF 位置 1 而被丢弃的广播数据包更新所有相关 MMC 计数器。

复位时，不会针对丢弃的广播数据包更新 MMC 计数器。

位 7:6 保留，必须保持复位值。

位 5 **CNTPRSTLVL**：全-半预设置 (Full-Half Preset)

该位为低电平且 CNTPRST 位置 1 时，所有 MMC 计数器均预设为几乎一半值。所有八位字节计数器均预设为 0x7FFF_F800（半个 2 KB），所有数据包计数器均预设为 0x7FFF_FFF0（半个 16）。

该位为高电平且 CNTPRST 位置 1 时，所有 MMC 计数器均预设为几乎全值。所有八位字节计数器均预设为 0xFFFF_F800（全 2 KB），所有数据包计数器均预设为 0xFFFF_FFF0（全 16）。

对于 16 位计数器，相应八位字节和数据包计数器的几乎一半预设值为 0x7800 和 0x7FF0。类似地，16 位计数器的几乎全预设值为 0xF800 和 0xFFF0。

位 4 **CNTPRST**：计数器预设 (Counters Preset)

该位置 1 时，所有计数器均初始化或预设为几乎全值或几乎一半值，具体取决于 CNTPRSTLVL 位。该位在 1 个时钟周期后自动清零。

该位与 CNTPRSTLVL 位一起用于调试和测试中断的有效情况，因为 MMC 计数器变为半-全或全值。

位 3 **CNTFREEZ**：MMC 计数器冻结 (MMC Counter Freeze)

该位置 1 时，会将所有 MMC 计数器冻结为当前值。

该位复位为 0 之前，因存在的任何已发送或已接收数据包而不会更新 MMC 计数器。在此模式下，“读取时复位”位置 1 时，如果读取任何 MMC 计数器，则该计数器也会清零。

位 2 **RSTONRD**：读取时复位 (Reset on Read)

该位置 1 时，读取 MMC 计数器后，该计数器会复位为零（复位后自清零）。读取最低有效字节通道（位 [7:0]）后，计数器会清零。

位 1 **CNTSTOPRO**：计数器停止翻转 (Counter Stop Rollover)

该位置 1 时，计数器达到其最大值后，不会返回到零。

位 0 **CNTRST**：计数器复位 (Counters Reset)

该位置 1 时，所有计数器都将复位。该位在 1 个时钟周期后自动清零。

MMC Rx 中断寄存器 (ETH_MMC_RX_INTERRUPT)

MMC Rx interrupt register

偏移地址：0x0704

复位值：0x0000 0000

该寄存器保存所有接收统计计数器生成的中断。

MMC 接收中断寄存器保存在发生以下情况时生成的中断：

- 接收统计计数器达到其最大值的一半（对于 32 位计数器为 0x8000_0000，对于 16 位计数器为 0x8000）。
- 接收统计计数器超过其最大值（对于 32 位计数器为 0xFFFF_FFFF，对于 16 位计数器为 0xFFFF）。

将 *MMC 控制寄存器 (ETH_MMC_CONTROL)* 中的 CNTSTOPRO 置 1 时，中断将置为有效，但计数器保持为全 1。MMC 接收中断寄存器为 32 位寄存器。当读取引发中断的各个 MMC 计数器时，会将中断位清零。必须读取相应计数器的最低有效字节通道（位 [7:0]），才能将中断位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	RXLPITRCIS	RXLPIUSCIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXUCGPIS	Res.
				rc_r	rc_r									rc_r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXALGNERPIS	RXCRCERPIS	Res.	Res.	Res.	Res.	Res.
									rc_r	rc_r					

位 31:28 保留，必须保持复位值。

位 27 **RXLPITRCIS**：MMC 接收 LPI 转换计数器中断状态 (MMC Receive LPI transition counter interrupt status)

当 *Rx LPI 转换计数器寄存器 (ETH_RX_LPI_TRAN_CNTR)* 计数器达到最大值的一半或最大值时，该位置 1。

位 26 **RXLPIUSCIS**：MMC 接收 LPI 微秒计数器中断状态 (MMC Receive LPI microsecond counter interrupt status)

当 *Rx LPI 微秒计数器寄存器 (ETH_RX_LPI_USEC_CNTR)* 计数器达到最大值的一半或最大值时，该位置 1。

位 25:18 保留，必须保持复位值。

位 17 **RXUCGPIS**：MMC 接收单播良好数据包计数器中断状态 (MMC Receive Unicast Good Packet Counter Interrupt Status)

当 *Rx 良好单播数据包寄存器 (ETH_RX_UNICAST_PACKETS_GOOD)* 计数器达到最大值的一半或最大值时，该位置 1。

位 16:7 保留，必须保持复位值。

位 6 **RXALGNERPIS**：MMC 接收对齐错误数据包计数器中断状态 (MMC Receive Alignment Error Packet Counter Interrupt Status)

当 *Rx 对齐错误数据包寄存器 (ETH_RX_ALIGNMENT_ERROR_PACKETS)* 计数器达到最大值的一半或最大值时，该位置 1。

位 5 **RXCRCERPIS**：MMC 接收 CRC 错误数据包计数器中断状态 (MMC Receive CRC Error Packet Counter Interrupt Status)

当 *Rx CRC 错误数据包寄存器 (ETH_RX_CRC_ERROR_PACKETS)* 计数器达到最大值的一半或最大值时，该位置 1。

位 4:0 保留，必须保持复位值。

MMC Tx 中断寄存器 (ETH_MMC_TX_INTERRUPT)

MMC Tx interrupt register

偏移地址：0x0708

复位值：0x0000 0000

该寄存器保存所有发送统计计数器生成的中断。

MMC 发送中断寄存器保存发送统计计数器达到其最大值的一半（对于 32 位计数器为 0x8000_0000，对于 16 位计数器为 0x8000）时生成的中断，以及发送统计计数器超过最大值（对于 32 位计数器为 0xFFFF_FFFF，对于 16 位计数器为 0xFFFF）时生成的中断。

将 *MMC 控制寄存器 (ETH_MMC_CONTROL)* 中的 CNTSTOPRO 置 1 时，中断将置为有效，但计数器保持为全 1。

MMC 发送中断寄存器为 32 位寄存器。当读取引发中断的各个 MMC 计数器时，会将中断位清零。

必须读取相应计数器的最低有效字节通道（位 [7:0]），才能将中断位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXLPITRCIS	TXLPIUSCIS	Res.	Res.	Res.	Res.	TXGPKTIS	Res.	Res.	Res.	Res.	Res.
				rc_r	rc_r					rc_r					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXMCOLGPIS	TXSCOLGPIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rc_r	rc_r														

位 31:28 保留，必须保持复位值。

位 27 **TXLPITRCIS**：MMC 发送 LPI 转换计数器中断状态 (MMC Transmit LPI transition counter interrupt status)

当 *Tx LPI 转换计数器寄存器 (ETH_TX_LPI_TRAN_CNTR)* 计数器达到最大值的一半或最大值时，该位置 1。

位 26 **TXLPIUSCIS**：MMC 发送 LPI 微秒计数器中断状态 (MMC Transmit LPI microsecond counter interrupt status)

当 *Tx LPI 微秒定时器寄存器 (ETH_TX_LPI_USEC_CNTR)* 计数器达到最大值的一半或最大值时，该位置 1。

位 25:22 保留，必须保持复位值。

位 21 **TXGPKTIS**：MMC 发送良好数据包计数器中断状态 (MMC Transmit Good Packet Counter Interrupt Status)

当 *Tx 良好数据包计数寄存器 (ETH_TX_PACKET_COUNT_GOOD)* 计数器达到最大值的一半或最大值时，该位置 1。

位 20:16 保留，必须保持复位值。

位 15 **TXMCOLGPIS**：MMC 发送多次冲突良好数据包计数器中断状态 (MMC Transmit Multiple Collision Good Packet Counter Interrupt Status)

当 *Tx 多次冲突良好数据包寄存器 (ETH_TX_MULTIPLE_COLLISION_GOOD_PACKETS)* 计数器达到最大值的一半或最大值时，该位置 1。

位 14 **TXSCOLGPIS**：MMC 发送单次冲突良好数据包计数器中断状态 (MMC Transmit Single Collision Good Packet Counter Interrupt Status)

当 *Tx 单次冲突良好数据包寄存器 (ETH_TX_SINGLE_COLLISION_GOOD_PACKETS)* 计数器达到最大值的一半或最大值时，该位置 1。

位 13:0 保留，必须保持复位值。

MMC Rx 中断屏蔽寄存器 (ETH_MMC_RX_INTERRUPT_MASK)

MMC Rx interrupt mask register

偏移地址：0x070C

复位值：0x0000 0000

MMC 接收中断屏蔽寄存器用于保存接收统计计数器达到其最大值的一半或最大值时所生成中断的屏蔽。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	RXLPITRCIM	RXLPIUSCIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXUCGPIM	Res.
				rw	rw									rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXALGNERPIM	RXCRCERPIM	Res.	Res.	Res.	Res.	Res.
									rw	rw					

位 31:28 保留，必须保持复位值。

位 27 **RXLPITRCIM**：MMC 接收 LPI 转换计数器中断屏蔽 (MMC Receive LPI transition counter interrupt Mask)

当 *Rx LPI 转换计数器寄存器 (ETH_RX_LPI_TRAN_CNTR)* 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 26 **RXLPIUSCIM**：MMC 接收 LPI 微秒计数器中断屏蔽 (MMC Receive LPI microsecond counter interrupt Mask)

当 *Rx LPI 微秒计数器寄存器 (ETH_RX_LPI_USEC_CNTR)* 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 25:18 保留，必须保持复位值。

位 17 **RXUCGPIM**：MMC 接收单播良好数据包计数器中断屏蔽 (MMC Receive Unicast Good Packet Counter Interrupt Mask)

当 *Rx 良好单播数据包寄存器 (ETH_RX_UNICAST_PACKETS_GOOD)* 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 16:7 保留，必须保持复位值。

位 6 **RXALGNERPIM**：MMC 接收对齐错误数据包计数器中断屏蔽 (MMC Receive Alignment Error Packet Counter Interrupt Mask)

当 *Rx 对齐错误数据包寄存器 (ETH_RX_ALIGNMENT_ERROR_PACKETS)* 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 5 **RXCRCERPIM**：MMC 接收 CRC 错误数据包计数器中断屏蔽 (MMC Receive CRC Error Packet Counter Interrupt Mask)

当 *Rx CRC 错误数据包寄存器 (ETH_RX_CRC_ERROR_PACKETS)* 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 4:0 保留，必须保持复位值。

MMC Tx 中断屏蔽寄存器 (ETH_MMC_TX_INTERRUPT_MASK)

MMC Tx interrupt mask register

偏移地址：0x0710

复位值：0x0000 0000

MMC 发送中断屏蔽寄存器用于保存发送统计计数器达到其最大值的一半或最大值时所生成中断的屏蔽。该寄存器为 32 位宽。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXLPITRCIM	TXLPIUSCIM	Res.	Res.	Res.	Res.	TXGPKTIM	Res.	Res.	Res.	Res.	Res.
				rw	rw					rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXMCOLGPIM	TXSCOLGPIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw														

位 31:28 保留，必须保持复位值。

位 27 **TXLPITRCIM**：MMC 发送 LPI 转换计数器中断屏蔽 (MMC Transmit LPI transition counter interrupt Mask)

当 *Tx LPI 转换计数器寄存器 (ETH_TX_LPI_TRAN_CNTR)* 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 26 **TXLPIUSCIM**：MMC 发送 LPI 微秒计数器中断屏蔽 (MMC Transmit LPI microsecond counter interrupt Mask)

当 *Tx LPI 微秒定时器寄存器 (ETH_TX_LPI_USEC_CNTR)* 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 25:22 保留，必须保持复位值。

位 21 **TXGPKTIM**：MMC 发送良好数据包计数器中断屏蔽 (MMC Transmit Good Packet Counter Interrupt Mask)

当 *Tx 良好数据包计数寄存器 (ETH_TX_PACKET_COUNT_GOOD)* 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 20:16 保留，必须保持复位值。

位 15 **TXMCOLGPIM**：MMC 发送多次冲突良好数据包计数器中断屏蔽 (MMC Transmit Multiple Collision Good Packet Counter Interrupt Mask)

当 *Tx 多次冲突良好数据包寄存器 (ETH_TX_MULTIPLE_COLLISION_GOOD_PACKETS)* 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 14 **TXSCOLGPIM**：MMC 发送单次冲突良好数据包计数器中断屏蔽 (MMC Transmit Single Collision Good Packet Counter Interrupt Mask)

当 *Tx 单次冲突良好数据包寄存器 (ETH_TX_SINGLE_COLLISION_GOOD_PACKETS)* 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 13:0 保留，必须保持复位值。

Tx 单次冲突良好数据包寄存器 (ETH_TX_SINGLE_COLLISION_GOOD_PACKETS)

Tx single collision good packets register

偏移地址：0x074C

复位值：0x0000 0000

该寄存器提供在半双工模式下于单次冲突后由以太网外设成功发送的数据包的数量。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXSNGLCOLG[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXSNGLCOLG[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TXSNGLCOLG[31:0]**：Tx 单次冲突良好数据包 (Tx Single Collision Good Packets)
该位字段指示在半双工模式下于单次冲突后成功发送的数据包的数量。

Tx 多次冲突良好数据包寄存器 (ETH_TX_MULTIPLE_COLLISION_GOOD_PACKETS)

Tx multiple collision good packets register

偏移地址：0x0750

复位值：0x0000 0000

该寄存器提供在半双工模式下于多次冲突后由以太网外设成功发送的数据包的数量。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXMULTCOLG[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXMULTCOLG[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TXMULTCOLG[31:0]**：Tx 多次冲突良好数据包 (Tx Multiple Collision Good Packets)
该位字段指示在半双工模式下于多次冲突后成功发送的数据包的数量。

Tx 良好数据包计数寄存器 (ETH_TX_PACKET_COUNT_GOOD)

Tx packet count good register

偏移地址：0x0768

复位值：0x0000 0000

该寄存器提供以太网外设发送的良好数据包的数量。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXPKTG[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXPKTG[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TXPKTG[31:0]**：Tx 良好数据包计数 (Tx Packet Count Good)
该位字段指示发送的良好数据包的数量。

Rx CRC 错误数据包寄存器 (ETH_RX_CRC_ERROR_PACKETS)

Rx CRC error packets register

偏移地址：0x0794

复位值：0x0000 0000

该寄存器提供以太网外设接收到的含 CRC 错误的数据包数量。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXCRCERR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXCRCERR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RXCRCERR[31:0]**：Rx CRC 错误数据包 (Rx CRC Error Packets)
该位字段指示接收到的含 CRC 错误的数据包数量。

Rx 对齐错误数据包寄存器 (ETH_RX_ALIGNMENT_ERROR_PACKETS)

Rx alignment error packets register

偏移地址：0x0798

复位值：0x0000 0000

该寄存器提供以太网外设接收到的含对齐 (dribble) 错误的数据包数量。它仅在 10/100 模式下有效。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXALGNERR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXALGNERR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RXALGNERR[31:0]**：Rx 对齐错误数据包 (Rx Alignment Error Packets)
该位字段指示接收到的含对齐 (dribble) 错误的数据包数量。它仅在 10/100 模式下有效。

Rx 良好单播数据包寄存器 (ETH_RX_UNICAST_PACKETS_GOOD)

Rx unicast packets good register

偏移地址：0x07C4

复位值：0x0000 0000

该寄存器提供以太网外设接收的良好单播数据包数量。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXUCASTG[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXUCASTG[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RXUCASTG[31:0]**：Rx 良好单播数据包 (Rx Unicast Packets Good)
该位字段指示接收的良好单播数据包数量。

Tx LPI 微秒定时器寄存器 (ETH_TX_LPI_USEC_CNTR)

Tx LPI microsecond timer register

偏移地址：0x07EC

复位值：0x0000 0000

该寄存器提供以太网外设将 Tx LPI 置为有效的微秒数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXLPIUSC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXLPIUSC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TXLPIUSC[31:0]**：Tx LPI 微秒计数器 (Tx LPI Microseconds Counter)
该位字段指示将 Tx LPI 置为有效的微秒数。对于每次 Tx LPI 进入和退出，定时器值均会有 +/- 1 微秒的误差。

Tx LPI 转换计数器寄存器 (ETH_TX_LPI_TRAN_CNTR)

Tx LPI transition counter register

偏移地址：0x07F0

复位值：0x0000 0000

该寄存器提供以太网外设已进入 Tx LPI 的次数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXLPITRC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXLPITRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TXLPITRC[31:0]**：Tx LPI 转换计数器 (Tx LPI Transition counter)

该位字段指示已进入 Tx LPI 的次数。即使在自动模式下进入 Tx LPI（因为 *LPI 控制和状态寄存器 (ETH_MACLCSR)* 中的 LPITXA 位置 1），计数器也会递增。

Rx LPI 微秒计数器寄存器 (ETH_RX_LPI_USEC_CNTR)

Rx LPI microsecond counter register

偏移地址：0x07F4

复位值：0x0000 0000

该寄存器提供以太网外设对 Rx LPI 进行采样的微秒数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXLPIUSC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXLPIUSC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RXLPIUSC[31:0]**：Rx LPI 微秒计数器 (Rx LPI Microseconds Counter)

该位字段指示将 Rx LPI 置为有效的微秒数。对于每次 Rx LPI 进入和退出，定时器值均会有 +/- 1 微秒的误差。

Rx LPI 转换计数器寄存器 (ETH_RX_LPI_TRAN_CNTR)

Rx LPI transition counter register

偏移地址：0x07F8

复位值：0x0000 0000

该寄存器提供以太网外设已进入 Rx LPI 的次数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXLPITRC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXLPITRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RXLPITRC[31:0]**：Rx LPI 转换计数器 (Rx LPI Transition counter)

该位字段指示已进入 Rx LPI 的次数。

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R)

L3 and L4 control 0 register

偏移地址：0x0900

复位值：0x0000 0000

第 3 层和第 4 层控制寄存器控制第 3 层和第 4 层的过滤器 0 的操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	L4DPIM0	L4DPM0	L4SPIM0	L4SPM0	Res.	L4PEN0
										rw	rw	rw	rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3HDBM0[4:0]					L3HSBM0[4:0]					L3DAIM0	L3DAM0	L3SAIM0	L3SAM0	Res.	L3PEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位 31:22 保留，必须保持复位值。

位 21 **L4DPIM0**：第 4 层目标端口反向匹配使能 (Layer 4 Destination Port Inverse Match Enable)

该位置 1 时，将使能第 4 层目标端口号位字段以进行反向匹配。该位复位时，将使能第 4 层目标端口号位字段以进行完美匹配。

只有在 L4DPM0 位置为高电平时，该位才有效且适用。

位 20 **L4DPM0**：第 4 层目标端口匹配使能 (Layer 4 Destination Port Match Enable)

该位置 1 时，将使能第 4 层目标端口号位字段以进行匹配。该位复位时，MAC 将忽略用于匹配的第 4 层目标端口号位字段。

位 19 **L4SPIM0**：第 4 层源端口反向匹配使能 (Layer 4 Source Port Inverse Match Enable)

该位置 1 时，将使能第 4 层源端口号位字段以进行反向匹配。该位复位时，将使能第 4 层源端口号位字段以进行完美匹配。

只有在 L4SPM0 位置为高电平时，该位才有效且适用。

位 18 **L4SPM0**：第 4 层源端口匹配使能 (Layer 4 Source Port Match Enable)

该位置 1 时，将使能第 4 层源端口号位字段以进行匹配。该位复位时，MAC 将忽略用于匹配的第 4 层源端口号位字段。

位 17 保留，必须保持复位值。

位 16 **L4PEN0**：第 4 层协议使能 (Layer 4 Protocol Enable)

该位置 1 时，将使用 UDP 数据包的源端口号和目标端口号位字段进行匹配。该位复位时，将使用 TCP 数据包的源端口号和目标端口号位字段进行匹配。

只有当 L4SPM0 或 L4DPM0 位置 1 时，才能完成第 4 层匹配。

位 15:11 **L3HDBM0[4:0]**：第 3 层 IP DA 高位匹配 (Layer 3 IP DA higher bits match)

条件：IPv4 数据包

该位字段包含在 IPv4 数据包中屏蔽的 IP 目标地址的高位数目：

0：未屏蔽任何位。

1：屏蔽 LSb[0]

2：屏蔽两个 LSb 位 [1:0]

..

31：屏蔽除了 MSb 之外的所有位。

条件：IPv6 数据包

该位字段的位 [12:11] 对应于 L3HSBM0 的位 [6:5]，表示在 IPv6 数据包中被屏蔽的 IP 源地址或目标地址的低位数目。屏蔽的位数由 L3HDBM0[1:0] 和 L3HSBM0 位的各个值给出：

0：未屏蔽任何位。

1：屏蔽 LSb[0]

2：屏蔽两个 LSb 位 [1:0]

..

31：屏蔽除了 MSb 之外的所有位。

只有在 L3DAM0 或 L3SAM0 位置 1 时，该位字段才有效且适用。

位 10:6 **L3HSBM0[4:0]**：第 3 层 IP SA 高位匹配 (Layer 3 IP SA higher bits match)

条件：IPv4 数据包

该位字段包含为在 IPv4 数据包中匹配而被屏蔽的 IP 源地址的低位数目。下面对该位字段的各个值进行了说明：

0：未屏蔽任何位。

1：屏蔽 LSb[0]

2：屏蔽两个 LSb 位 [1:0]

..

31：屏蔽除了 MSb 之外的所有位。

条件：IPv6 数据包：

该位字段包含 L3HSBM0 的位 [4:0]。这些位表示 IPv6 数据包中匹配的 IP 源地址或目标地址的高位数目。只有在 L3DAM0 或 L3SAM0 位置为高电平时，该位字段才有效且适用。

位 5 **L3DAIM0**：第 3 层 IP DA 反向匹配使能 (Layer 3 IP DA Inverse Match Enable)

该位置 1 时，将使能第 3 层 IP 目标地址位字段以进行反向匹配。该位复位时，将使能第 3 层 IP 目标地址位字段以进行完美匹配。

只有在 L3DAM0 位置为高电平时，该位才有效且适用。

位 4 **L3DAM0**：第 3 层 IP DA 匹配使能 (Layer 3 IP DA Match Enable)

该位置 1 时，将使能第 3 层 IP 目标地址位字段以进行匹配。该位复位时，MAC 将忽略用于匹配的第 3 层 IP 目标地址位字段。

注意：L3PEN0 位置 1 时，应将该位或 L3SAM0 位置 1，因为这样可检查 IPv6 DA 或 SA 以进行过滤。

位 3 **L3SAIM0**：第 3 层 IP SA 反向匹配使能 (Layer 3 IP SA Inverse Match Enable)

该位置 1 时，将使能第 3 层 IP 源地址位字段以进行反向匹配。该位复位时，将使能第 3 层 IP 源地址位字段以进行完美匹配。

只有在 L3SAM0 位置 1 时，该位才有效且适用。

位 2 **L3SAM0**：第 3 层 IP SA 匹配使能 (Layer 3 IP SA Match Enable)

该位置 1 时，将使能第 3 层 IP 源地址位字段以进行匹配。该位复位时，MAC 将忽略用于匹配的第 3 层 IP 源地址位字段。

注意：L3PEN0 位置 1 时，应将该位或 L3DAM0 位置 1，因为这样可检查 IPv6 SA 或 DA 以进行过滤。

位 1 保留，必须保持复位值。

位 0 **L3PEN0**：第 3 层协议使能 (Layer 3 Protocol Enable)

该位置 1 时，将为 IPv6 数据包使能第 3 层 IP 源地址或目标地址匹配。该位复位时，将为 IPv4 数据包使能第 3 层 IP 源地址或目标地址匹配。

只有当 L3SAM0 或 L3DAM0 位置 1 时，才能完成第 3 层匹配。

第 4 层地址过滤器 0 寄存器 (ETH_MACL4A0R)

Layer4 Address filter 0 register

偏移地址：0x0904

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L4DP0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L4SP0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **L4DP0[15:0]**：第 4 层目标端口号位字段 (Layer 4 Destination Port Number Field)

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R) 中的 L4PEN0 位复位且 L4DPM0 位置 1 时，该位字段包含要与 IPv4 或 IPv6 数据包中的 TCP 目标端口号位字段进行匹配的值。

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R) 中的 L4PEN0 和 L4DPM0 位置 1 时，该位字段包含要与 IPv4 或 IPv6 数据包中的 UDP 目标端口号位字段进行匹配的值。

位 15:0 **L4SP0[15:0]**：第 4 层源端口号位字段 (Layer 4 Source Port Number Field)

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R) 中的 L4PEN0 位复位且 L4DPM0 位置 1 时，该位字段包含要与 IPv4 或 IPv6 数据包中的 TCP 源端口号位字段进行匹配的值。

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R) 中的 L4PEN0 和 L4DPM0 位置 1 时，该位字段包含要与 IPv4 或 IPv6 数据包中的 UDP 源端口号位字段进行匹配的值。

第 3 层地址 0 过滤器 0 寄存器 (ETH_MACL3A00R)

Layer3 Address 0 filter 0 register

偏移地址：0x0910

复位值：0x0000 0000

对于 IPv4 数据包，第 3 层地址 0 过滤器 0 寄存器包含 32 位 IP 源地址位字段。对于 IPv6 数据包，它包含 128 位 IP 源地址或目标地址位字段的位 [31:0]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A00[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A00[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **L3A00[31:0]**：第 3 层地址 0 位字段 (Layer 3 Address 0 Field)

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R) 中的 L3PEN0 和 L3SAM0 位置 1 时，该位字段包含要与 IPv6 数据包中 IP 源地址位字段的位 [31:0] 进行匹配的值。

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R) 中的 L3PEN0 和 L3DAM0 位置 1 时，该位字段包含要与 IPv6 数据包中 IP 目标地址位字段的位 [31:0] 进行匹配的值。

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R) 中的 L3PEN0 位复位且 L3SAM0 位置 1 时，该位字段包含要与 IPv4 数据包中的 IP 源地址位字段进行匹配的值。

第 3 层地址 1 过滤器 0 寄存器 (ETH_MACL3A10R)

Layer3 Address 1 filter 0 register

偏移地址：0x0914

复位值：0x0000 0000

对于 IPv4 数据包，第 3 层地址 1 过滤器 0 寄存器包含 32 位 IP 目标地址位字段。对于 IPv6 数据包，它包含 128 位 IP 源地址或目标地址位字段的位 [63:32]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A10[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A10[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **L3A10[31:0]**：第 3 层地址 1 位字段 (Layer 3 Address 1 Field)

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R) 中的 L3PEN0 和 L3SAM0 位置 1 时，该位字段包含要与 IPv6 数据包中 IP 源地址位字段的位 [63:32] 进行匹配的值。

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R) 中的 L3PEN0 和 L3DAM0 位置 1 时，该位字段包含要与 IPv6 数据包中 IP 目标地址位字段的位 [63:32] 进行匹配的值。

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R) 中的 L3PEN0 位复位且 L3SAM0 位置 1 时，该位字段包含要与 IPv4 数据包中的 IP 目标地址位字段进行匹配的值。

第 3 层地址 2 过滤器 0 寄存器 (ETH_MACL3A20R)

Layer3 Address 2 filter 0 register

偏移地址：0x0918

复位值：0x0000 0000

对于 IPv4 数据包，第 3 层地址 2 过滤器 0 寄存器保留。对于 IPv6 数据包，它包含 128 位 IP 源地址或目标地址位字段的位 [95:64]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A20[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A20[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **L3A20[31:0]**：第 3 层地址 2 位字段 (Layer 3 Address 2 Field)

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R) 中的 L3PEN0 和 L3SAM0 位置 1 时，该位字段包含要与 IPv6 数据包中 IP 源地址位字段的位 [95:64] 进行匹配的值。

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R) 中的 L3PEN0 和 L3DAM0 位置 1 时，该位字段包含要与 IPv6 数据包中 IP 目标地址位字段的位 [95:64] 进行匹配的值。

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R) 中的 L3PEN0 位复位时，不使用该位字段。

第 3 层地址 3 过滤器 0 寄存器 (ETH_MACL3A30R)

Layer3 Address 3 filter 0 register

偏移地址：0x091C

复位值：0x0000 0000

对于 IPv4 数据包，第 3 层地址 3 过滤器 0 寄存器保留。对于 IPv6 数据包，它包含 128 位 IP 源地址或目标地址位字段的位 [127:96]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A30[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A30[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **L3A30[31:0]**：第 3 层地址 3 位字段 (Layer 3 Address 3 Field)

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R) 中的 L3PEN0 和 L3SAM0 位置 1 时，该位字段包含要与 IPv6 数据包中 IP 源地址位字段的位 [127:96] 进行匹配的值。

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R) 中的 L3PEN0 和 L3DAM0 位置 1 时，该位字段包含要与 IPv6 数据包中 IP 目标地址位字段的位 [127:96] 进行匹配的值。

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R) 中的 L3PEN0 位复位时，不使用该位字段。

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R)

L3 and L4 control 1 register

偏移地址：0x0930

复位值：0x0000 0000

第 3 层和第 4 层控制寄存器控制第 3 层和第 4 层的过滤器 1 的操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	L4DPIM1	L4DPM1	L4SPIM1	L4SPM1	Res.	L4PEN1
										rw	rw	rw	rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3HDBM1[4:0]					L3HSBM1[4:0]					L3DAIM1	L3DAM1	L3SAIM1	L3SAM1	Res.	L3PEN1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位 31:22 保留，必须保持复位值。

位 21 **L4DPIM1**：第 4 层目标端口反向匹配使能 (Layer 4 Destination Port Inverse Match Enable)
该位置 1 时，将使能第 4 层目标端口号位字段以进行反向匹配。该位复位时，将使能第 4 层目标端口号位字段以进行完美匹配。
只有在 L4DPM1 位置为高电平时，该位才有效且适用。

位 20 **L4DPM1**：第 4 层目标端口匹配使能 (Layer 4 Destination Port Match Enable)
该位置 1 时，将使能第 4 层目标端口号位字段以进行匹配。该位复位时，MAC 将忽略用于匹配的第 4 层目标端口号位字段。

位 19 **L4SPIM1**：第 4 层源端口反向匹配使能 (Layer 4 Source Port Inverse Match Enable)
该位置 1 时，将使能第 4 层源端口号位字段以进行反向匹配。该位复位时，将使能第 4 层源端口号位字段以进行完美匹配。
只有在 L4SPM1 位置为高电平时，该位才有效且适用。

位 18 **L4SPM1**：第 4 层源端口匹配使能 (Layer 4 Source Port Match Enable)
该位置 1 时，将使能第 4 层源端口号位字段以进行匹配。该位复位时，MAC 将忽略用于匹配的第 4 层源端口号位字段。

位 17 保留，必须保持复位值。

位 16 **L4PEN1**：第 4 层协议使能 (Layer 4 Protocol Enable)
该位置 1 时，将使用 UDP 数据包的源端口号和目标端口号位字段进行匹配。该位复位时，将使用 TCP 数据包的源端口号和目标端口号位字段进行匹配。
只有当 L4SPM1 或 L4DPM1 位置 1 时，才能完成第 4 层匹配。

位 15:11 **L3HDBM1[4:0]**：第 3 层 IP DA 高位匹配 (Layer 3 IP DA higher bits match)

条件：IPv4 数据包

该位字段包含为在 IPv4 数据包中匹配而被屏蔽的 IP 目标地址的低位数目。下面对该位字段的各个值进行了说明：

0：未屏蔽任何位。

1：屏蔽 LSb[0]

2：屏蔽两个 LSb 位 [1:0]

..

31：屏蔽除了 MSb 之外的所有位。

条件：IPv6 数据包

该位字段的位 [12:11] 对应于 L3HSBM1 的位 [6:5]，表示在 IPv6 数据包中被屏蔽的 IP 源地址或目标地址的低位数目。下面对 L3HDBM1[1:0] 和 L3HSBM1 位的各个值进行了说明：

0：未屏蔽任何位

1：屏蔽 LSb[0]

2：屏蔽两个 LSb 位 [1:0]

..

127：屏蔽除了 MSb 之外的所有位

只有在 L3DAM1 或 L3SAM1 位置 1 时，该位字段才有效且适用。

位 10:6 **L3HSBM1[4:0]**：第 3 层 IP SA 高位匹配 (Layer 3 IP SA Higher Bits Match)

条件：IPv4 数据包

该位字段包含为在 IPv4 数据包中匹配而被屏蔽的 IP 源地址的低位数目。下面对该位字段的各个值进行了说明：

0：未屏蔽任何位。

1：屏蔽 LSb[0]

2：屏蔽两个 LSb 位 [1:0]

..

31：屏蔽除了 MSb 之外的所有位。

条件：IPv6 数据包

该位字段包含 L3HSBM1 的位 [4:0]。这些位表示 IPv6 数据包中匹配的 IP 源地址或目标地址的高位数目。只有在 L3DAM1 或 L3SAM1 位置为高电平时，该位字段才有效且适用。

位 5 **L3DAIM1**：第 3 层 IP DA 反向匹配使能 (Layer 3 IP DA Inverse Match Enable)

该位置 1 时，将使能第 3 层 IP 目标地址位字段以进行反向匹配。该位复位时，将使能第 3 层 IP 目标地址位字段以进行完美匹配。

只有在 L3DAM1 位置为高电平时，该位才有效且适用。

位 4 **L3DAM1**：第 3 层 IP DA 匹配使能 (Layer 3 IP DA Match Enable)

该位置 1 时，将使能第 3 层 IP 目标地址位字段以进行匹配。该位复位时，MAC 将忽略用于匹配的第 3 层 IP 目标地址位字段。

注意：L3PEN1 位置 1 时，应将该位或 L3SAM1 位置 1，因为这样可检查 IPv6 DA 或 SA 以进行过滤。

位 3 **L3SAIM1**：第 3 层 IP SA 反向匹配使能 (Layer 3 IP SA Inverse Match Enable)

该位置 1 时，将使能第 3 层 IP 源地址位字段以进行反向匹配。该位复位时，将使能第 3 层 IP 源地址位字段以进行完美匹配。

只有在 L3SAM1 位置 1 时，该位才有效且适用。

位 2 **L3SAM1**：第 3 层 IP SA 匹配使能 (Layer 3 IP SA Match Enable)

该位置 1 时，将使能第 3 层 IP 源地址位字段以进行匹配。该位复位时，MAC 将忽略用于匹配的第 3 层 IP 源地址位字段。

注意：L3PEN01 位置 1 时，应将该位或 L3DAM1 位置 1，因为这样可检查 IPv6 SA 或 DA 以进行过滤。

位 1 保留，必须保持复位值。

位 0 **L3PEN1**：第 3 层协议使能 (Layer 3 Protocol Enable)

该位置 1 时，将为 IPv6 数据包使能第 3 层 IP 源地址或目标地址匹配。该位复位时，将为 IPv4 数据包使能第 3 层 IP 源地址或目标地址匹配。

只有当 L3SAM1 或 L3DAM1 位置 1 时，才能完成第 3 层匹配。

第 4 层地址过滤器 1 寄存器 (ETH_MACL4A1R)

Layer 4 address filter 1 register

偏移地址：0x0934

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L4DP1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L4SP1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **L4DP1[15:0]**：第 4 层目标端口号位字段 (Layer 4 Destination Port Number Field)

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R) 中的 L4PEN1 位复位且 L4DPM1 位置 1 时，该位字段包含要与 IPv4 或 IPv6 数据包中的 TCP 目标端口号位字段进行匹配的值。

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R) 中的 L4PEN1 和 L4DPM1 位置 1 时，该位字段包含要与 IPv4 或 IPv6 数据包中的 UDP 目标端口号位字段进行匹配的值。

位 15:0 **L4SP1[15:0]**：第 4 层源端口号位字段 (Layer 4 Source Port Number Field)

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R) 中的 L4PEN1 位复位且 L4DPM1 位置 1 时，该位字段包含要与 IPv4 或 IPv6 数据包中的 TCP 源端口号位字段进行匹配的值。

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R) 中的 L4PEN1 和 L4DPM1 位置 1 时，该位字段包含要与 IPv4 或 IPv6 数据包中的 UDP 源端口号位字段进行匹配的值。

第 3 层地址 0 过滤器 1 寄存器 (ETH_MACL3A01R)

Layer3 address 0 filter 1 Register

偏移地址：0x0940

复位值：0x0000 0000

对于 IPv4 数据包，第 3 层地址 0 过滤器 1 寄存器包含 32 位 IP 源地址位字段。对于 IPv6 数据包，它包含 128 位 IP 源地址或目标地址位字段的位 [31:0]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A01[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A01[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **L3A01[31:0]**：第 3 层地址 0 位字段 (Layer 3 Address 0 Field)

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R) 中的 L3PEN1 和 L3SAM1 位置 1 时，该位字段包含要与 IPv6 数据包中 IP 源地址位字段的位 [31:0] 进行匹配的值。

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R) 中的 L3PEN1 和 L3DAM1 位置 1 时，该位字段包含要与 IPv6 数据包中 IP 目标地址位字段的位 [31:0] 进行匹配的值。

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R) 中的 L3PEN1 位复位且 L3SAM1 位置 1 时，该位字段包含要与 IPv4 数据包中的 IP 源地址位字段进行匹配的值。

第 3 层地址 1 过滤器 1 寄存器 (ETH_MACL3A11R)

Layer3 address 1 filter 1 register

偏移地址：0x0944

复位值：0x0000 0000

对于 IPv4 数据包，第 3 层地址 1 过滤器 1 寄存器包含 32 位 IP 目标地址位字段。对于 IPv6 数据包，它包含 128 位 IP 源地址或目标地址位字段的位 [63:32]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A11[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A11[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **L3A11[31:0]**：第 3 层地址 1 位字段 (Layer 3 Address 1 Field)

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R) 中的 L3PEN1 和 L3SAM1 位置 1 时，该位字段包含要与 IPv6 数据包中 IP 源地址位字段的位 [63:32] 进行匹配的值。

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R) 中的 L3PEN1 和 L3DAM1 位置 1 时，该位字段包含要与 IPv6 数据包中 IP 目标地址位字段的位 [63:32] 进行匹配的值。

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R) 中的 L3PEN1 位复位且 L3SAM1 位置 1 时，该位字段包含要与 IPv4 数据包中的 IP 目标地址位字段进行匹配的值。

第 3 层地址 2 过滤器 1 寄存器 (ETH_MACL3A21R)

Layer3 address 2 filter 1 Register

偏移地址：0x0948

复位值：0x0000 0000

对于 IPv4 数据包，第 3 层地址 2 过滤器 1 寄存器保留。对于 IPv6 数据包，它包含 128 位 IP 源地址或目标地址位字段的位 [95:64]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A21[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A21[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **L3A21[31:0]**：第 3 层地址 2 位字段 (Layer 3 Address 2 Field)

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R) 中的 L3PEN1 和 L3SAM1 位置 1 时，该位字段包含要与 IPv6 数据包中 IP 源地址位字段的位 [95:64] 进行匹配的值。

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R) 中的 L3PEN1 和 L3DAM1 位置 1 时，该位字段包含要与 IPv6 数据包中 IP 目标地址位字段的位 [95:64] 进行匹配的值。

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R) 中的 L3PEN1 位复位时，不使用该位字段。

第 3 层地址 3 过滤器 1 寄存器 (ETH_MACL3A31R)

Layer3 address 3 filter 1 register

偏移地址：0x94C

复位值：0x0000 0000

对于 IPv4 数据包，第 3 层地址 3 过滤器 1 寄存器保留。对于 IPv6 数据包，它包含 128 位 IP 源地址或目标地址位字段的位 [127:96]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A31[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A31[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **L3A31[31:0]**：第 3 层地址 3 位字段 (Layer 3 Address 3 Field)

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R) 中的 L3PEN1 和 L3SAM1 位置 1 时，该位字段包含要与 IPv6 数据包中 IP 源地址位字段的位 [127:96] 进行匹配的值。

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R) 中的 L3PEN1 和 L3DAM1 位置 1 时，该位字段包含要与 IPv6 数据包中 IP 目标地址位字段的位 [127:96] 进行匹配的值。

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R) 中的 L3PEN1 位复位时，不使用该位字段。

时间戳控制寄存器 (ETH_MACTSCR)

Timestamp control Register

偏移地址：0x0B00

复位值：0x0000 2000

该寄存器控制系统时间发生器的工作以及针对接收器中时间戳的 PTP 数据包处理。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	AV8021ASMEN	Res.	Res.	Res.	TXTSSTSM	Res.	Res.	Res.	Res.	Res.	TSENMACADDR	SNAPTYPSEL[1:0]	
			rw				rw						rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSMSTRENA	TSEVNTENA	TSIPV4ENA	TSIPV6ENA	TSIPENA	TSVER2ENA	TSCTRLSSR	TSENALL	Res.	Res.	TSADDREG	Res.	TSUPDT	TSINIT	TSCFUPDT	TSENA
rw	rw	rw	rw	rw	rw	rw	rw			rw		rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28 **AV8021ASMEN**：AV 802.1AS 模式使能 (AV 802.1AS Mode Enable)

当此位置 1 时，MAC 仅会处理未标记的 PTP over Ethernet 数据包，以提供 PTP 状态并捕获时间戳快照（即 IEEE 802.1AS 工作模式）。

如果使能了 PTP 减荷功能，则为了实现 PTP 减荷，系统会根据此位的值生成并检查 PTP 报头中的传输特定位字段。

位 27:25 保留，必须保持复位值。

位 24 **TXTSSTSM**：发送时间戳状态模式 (Transmit Timestamp Status Mode)

该位置 1 时，MAC 会覆盖较早发送时间戳状态，即使软件未对其进行读取操作也如此。MAC 通过将 *Tx 时间戳状态纳秒寄存器 (ETH_MACTXTSSNR)* 寄存器的 TXTSSMIS 位置 1 对此进行指示。

该位复位时，如果软件未读取先前数据包的时间戳状态，则 MAC 将忽略当前数据包的时间戳状态。MAC 通过将 *Tx 时间戳状态纳秒寄存器 (ETH_MACTXTSSNR)* 的 TXTSSMIS 位置 1 对此进行指示。

位 23:19 保留，必须保持复位值。

位 18 **TSENMACADDR**：使能 MAC 地址以用于 PTP 数据包过滤 (Enable MAC Address for PTP Packet Filtering)

如果该位置 1，则在通过以太网直接发送 PTP 时，会使用 DA MAC 地址（与任一 MAC 地址寄存器匹配）过滤 PTP 数据包。

如果该位置 1，则在通过以太网直接发送 PTP 时，对于所接收到的具有 DA（包含特殊多播或单播地址，与 MAC 地址寄存器中编程的地址匹配）的 PTP 数据包，将被视为要进行处理的数据包，如下所述。

对于正常时间戳操作，会将 MAC 地址寄存器 0 到 31 用于单播目标地址匹配。

对于 PTP 减荷，仅将 MAC 地址寄存器 0 用于单播目标地址匹配。

位 17:16 **SNAPTYPSEL[1:0]**：选择 PTP 数据包以拍摄快照 (Select PTP packets for Taking Snapshots)

这些位与位 15 和位 14 共同决定需要拍摄快照的 PTP 数据包类型组。编码如*表 538：时间戳快照对 ETH_MACTSCR 位的相依性*所示。

位 15 **TSMSTRENA**：使能主节点相关消息的快照 (Enable Snapshot for Messages Relevant to Master)
此位置 1 时，仅拍摄主节点相关消息的快照。否则，将拍摄从节点相关消息的快照。

位 14 **TSEVNTENA**：使能事件消息的时间戳快照 (Enable Timestamp Snapshot for Event Messages)
此位置 1 时，仅拍摄事件消息的时间戳快照（SYNC、Delay_Req、Pdelay_Req 或 Pdelay_Resp）。此位复位时，将拍摄除 Announce、Management 和 Signaling 以外所有消息的快照。有关时间戳快照的更多信息，请参见*表 538：时间戳快照对 ETH_MACTSCR 位的相依性*。

位 13 **TSIPV4ENA**：使能对通过 IPv4-UDP 发送的 PTP 数据包的处理 (Enable Processing of PTP Packets Sent over IPv4-UDP)
该位置 1 时，MAC 接收器处理封装在 IPv4-UDP 数据包中的 PTP 数据包。该位复位时，MAC 将忽略通过 IPv4-UDP 数据包传送的 PTP。默认情况下，此位置 1。

位 12 **TSIPV6ENA**：使能对通过 IPv6-UDP 发送的 PTP 数据包的处理 (Enable Processing of PTP Packets Sent over IPv6-UDP)
该位置 1 时，MAC 接收器处理封装在 IPv6-UDP 数据包中的 PTP 数据包。该位清零时，MAC 将忽略通过 IPv6-UDP 数据包传送的 PTP。

位 11 **TSIPENA**：使能对 PTP over Ethernet 数据包的处理 (Enable Processing of PTP over Ethernet Packets)
该位置 1 时，MAC 接收器处理直接封装在以太网数据包中的 PTP 数据包。该位复位时，MAC 将忽略 PTP over Ethernet 数据包。

位 10 **TSVER2ENA**：针对版本 2 格式使能 PTP 数据包处理 (Enable PTP Packet Processing for Version 2 Format)
该位置 1 时，会使用 IEEE 1588 版本 2 格式来处理 PTP 数据包。该位复位时，会使用 IEEE 1588 版本 1 格式来处理 PTP 数据包。“PTP 处理和控制”中对 IEEE 1588 格式进行了介绍。

位 9 **TSCTRLSSR**：时间戳数字或二进制翻转控制 (Timestamp Digital or Binary Rollover Control)
该位置 1 时，时间戳低位寄存器会在 0x3B9A_C9FF 值之后翻转（即，1 纳秒精度），并递增时间戳（高位）秒数。该位复位时，亚秒寄存器的翻转值为 0x7FFF_FFFF。必须根据 PTP 参考时钟频率和该位的值，正确对亚秒增量进行编程。

位 8 **TSENALL**：针对所有数据包使能时间戳 (Enable Timestamp for All Packets)
此位置 1 时，会针对 MAC 所接收的全部数据包使能时间戳快照。

位 7:6 保留，必须保持复位值。

位 5 **TSADDREG**：更新加数寄存器 (Update Addend Register)
此位置 1 时，时间戳加数寄存器的内容会被更新到 PTP 块中以进行精密校准。更新结束时此位会清零。该位在置 1 之前应为零。

位 4 保留，必须保持复位值。

位 3 **TSUPDT**：更新时间戳 (Update Timestamp)
该位置 1 时，将使用*系统时间秒更新寄存器 (ETH_MACSTSUR)* 和*系统时间纳秒更新寄存器 (ETH_MACSTNUR)* 中指定的值更新（加或减）系统时间。
该位在更新之前应为零。通过硬件完成更新时，此位会复位。

位 2 **TSINIT**：初始化时间戳 (Initialize Timestamp)
当此位置 1 时，系统时间将通过在*系统时间秒更新寄存器 (ETH_MACSTSUR)* 和*系统时间纳秒更新寄存器 (ETH_MACSTNUR)* 中指定的值进行初始化（覆盖）。

该位在更新之前应为零。该位在完成初始化时会复位。

位 1 **TSCFUPDT**：时间戳精密或粗略更新 (Fine or Coarse Timestamp Update)
该位置 1 时，使用精密方法更新系统时间戳。该位复位时，使用粗略方法更新系统时间戳。

位 0 **TSENA**：使能时间戳 (Enable Timestamp)
该位置 1 时，为发送和接收数据包添加时间戳。该位被禁止时，不会为发送和接收数据包添加时间戳，并且时间戳发生器也会被挂起。使能此模式后，需要初始化时间戳（系统时间）。
在接收端，MAC 只在该位置 1 时才处理 1588 个数据包。

亚秒增量寄存器 (ETH_MACSSIR)

Subsecond increment register

偏移地址：0x0B04

复位值：0x0000 0000

在粗略更新模式下（*时间戳控制寄存器 (ETH_MACTSCR)* 中的位 TSCFUPDT），此寄存器中的值在每个 clk_ptp_ref_i 时钟周期后都会被添加到系统时间。在精密更新模式下，此寄存器中的值将在累加器溢出时被添加到系统时间。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SSINC[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:24 保留，必须保持复位值。

位 23:16 **SSINC[7:0]**：亚秒增量值 (Subsecond Increment Value)
在该位字段中编程的值在（clk_ptp_i 的）每个时钟周期都会与亚秒寄存器的内容进行累加。例如，当 PTP 时钟为 50 MHz（周期为 20 ns）时，如果系统时间纳秒寄存器的精度为 1 ns [*时间戳控制寄存器 (ETH_MACTSCR)* 中的 TSCTRLSSR 位置 1]，则应编程 20 (0x14)。当 TSCTRLSSR 清零时，纳秒寄存器的分辨率约为 0.465 ns。在这种情况下，应编程值 43 (0x2B)（由 20 ns/0.465 得出）。

位 15:0 保留，必须保持复位值。

系统时间秒寄存器 (ETH_MACSTSR)

System time seconds register

偏移地址：0x0B08

复位值：0x0000 0000

系统时间秒寄存器与系统时间纳秒寄存器一起指示由 MAC 保持的系统时间的当前值。虽然该值是连续更新的，但由于存在时钟域传输延迟（从 clk_ptp_ref_i 到 CSR 时钟），实际时间有一些延迟。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSS[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TSS[31:0]**：时间戳秒 (Timestamp Second)

此位字段中的值表示由 MAC 保持的以秒为单位的系统时间当前值。

系统时间纳秒寄存器 (ETH_MACSTNR)

System time nanoseconds register

偏移地址：0x0B0C

复位值：0x0000 0000

系统时间纳秒寄存器与系统时间秒寄存器一起指示由 MAC 保持的系统时间的当前值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TSSS[30:16]														
	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSSS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31 保留，必须保持复位值。

位 30:0 **TSSS[30:0]**：时间戳亚秒 (Timestamp subseconds)

该位字段中的值包含亚秒时间表示，精度为 0.46 ns。当*时间戳控制寄存器 (ETH_MACTSCR)* 中的 TSCTRLSSR 置 1 时，每个位表示 1 ns。最大值为 0x3B9A_C9FF，之后会翻转为零。

系统时间秒更新寄存器 (ETH_MACSTSUR)

System time seconds update register

偏移地址：0x0B10

复位值：0x0000 0000

系统时间秒更新寄存器与系统时间纳秒更新寄存器一起用于初始化或更新 MAC 保持的系统时间。在将*时间戳控制寄存器 (ETH_MACTSCR)* 中的 TSINIT 或 TSUPDT 位置 1 前，必须写入这两个寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSS[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSS[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **TSS[31:0]**：时间戳秒 (Timestamp Seconds)

此位字段中的值是更新的秒部分。

ADDSUB 复位时，该位字段必须用更新值的秒部分进行编程。

ADDSUB 置 1 时，该位字段必须用更新值秒部分的补码进行编程。

例如，如需从系统时间中减去 2.000000001 秒，则 ETH_MACSTSUR 寄存器中的 TSS 位字段必须为 0xFFFF_FFFE（即 2^32 – 2）。

小心： 当 ADDSUB 位置 1 时，*系统时间纳秒更新寄存器 (ETH_MACSTNUR)* 中的 TSSS[30:0] 位字段不能置 0。即使必须减去 0 ns，TSSS 位字段也必须编程为 0x7FFF FFFF（结果为 -0.46 ns）。

例如，如需从系统时间中减去 2.000000000 秒，则*系统时间秒更新寄存器 (ETH_MACSTSUR)* 中的 TSS 位字段必须为 0xFFFF FFFE（即 2^32 – 1），并且*系统时间纳秒更新寄存器 (ETH_MACSTNUR)* 必须为 0xFFFF FFFF（ADDSUB = 1 且 TSSS[30:0] 位字段 = 0x7FFF_FFFF）

系统时间纳秒更新寄存器 (ETH_MACSTNUR)

System time nanoseconds update register

偏移地址：0x0B14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD SUB	TSSS[30:16]														
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSSS[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **ADDSUB**：加上或减去时间 (Add or Subtract Time)

该位置 1 时，用更新寄存器的内容减去时间值。该位复位时，时间值与更新寄存器的内容相加。

位 30:0 **TSSS[30:0]**：时间戳亚秒 (Timestamp subseconds)

此位字段中的值是更新的亚秒部分。

– ADDSUB 为 1：该位字段必须用更新值亚秒部分的补码进行编程，如文中所述。

– ADDSUB 为 0：该位字段必须用更新值的亚秒部分进行编程，其精度取决于*时间戳控制寄存器 (ETH_MACTSCR)* 的 TSCTRLSSR 位。

– *时间戳控制寄存器 (ETH_MACTSCR)* 中的 TSCTRLSSR 位字段为 1：
- 编程值必须为 10^9 - <亚秒值>。
- 每个位均表示 1 ns，且编程值不应超过 0x3B9A_C9FF。

– *时间戳控制寄存器 (ETH_MACTSCR)* 中的 TSCTRLSSR 位字段为 0：
- 编程值必须为 2^31 - <亚秒值>。
- 每个位均表示 0.46 ns 的精度。

例如，如需从系统时间中减去 2.000000001 秒，则当*时间戳控制寄存器 (ETH_MACTSCR)* 中的 TSCTRLSSR 位复位时，ETH_MACSTNUR 寄存器中的 TSSS 位字段必须为 0x7FFF_FFFF（即 2^31 – 1），而当*时间戳控制寄存器 (ETH_MACTSCR)* 中的 TSCTRLSSR 位置 1 时，该位字段必须为 0x3B9A_C9FF（即 10^9 –1）。

小心： 当 ADDSUB 位置 1 时，TSSS[30:0] 位字段不能置 0。即使必须减去 0 ns，TSSS 位字段也必须编程为 0x7FFF FFFF（结果为 -0.46 ns）。
例如，如需从系统时间中减去 2.000000000 秒，则*系统时间纳秒更新寄存器 (ETH_MACSTNUR)* 必须为 0xFFFF FFFF（ADDSUB = 1 且 TSSS[30:0] = 0），并且*系统时间秒更新寄存器 (ETH_MACSTSUR)* 中的 TSS 位字段必须为 0xFFFF FFFE（即 2^32 – 1）。

时间戳加数寄存器 (ETH_MACTSAR)

Timestamp addend register

偏移地址：0x0B18

复位值：0x0000 0000

仅当系统时间配置为精密更新模式（*时间戳控制寄存器 (ETH_MACTSCR)* 中的 TSCFUPDT 位）时，才使用该寄存器值。该寄存器的内容在 clk_ptp_ref_i 的每个时钟周期添加到一个 32 位累加器中，系统时间在该累加器发生上溢时更新。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSAR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSAR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **TSAR[31:0]**：时间戳加数寄存器 (Timestamp Addend Register)

该位字段指示要添加到累加器寄存器以实现时间同步的 32 位时间值。

时间戳状态寄存器 (ETH_MACTSSR)

Timestamp status register

偏移地址：0x0B20

复位值：0x0000 0000

当应用程序读取该寄存器时，除位 [27:25] 之外的所有位都将被清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ATSNS[4:0]					ATSSTM	Res.	Res.	Res.	Res.	ATSSTN[3:0]			
		r	r	r	r	r	rc_r					rc_r	rc_r	rc_r	rc_r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXTSSIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSTRGTERR0	AUXTSTRIG	TSTARGT0	TSSOVF
rc_r												rc_r	rc_r	rc_r	rc_r

位 31:30 保留，必须保持复位值。

位 29:25 **ATSNS[4:0]**：辅助时间戳快照数 (Number of Auxiliary Timestamp Snapshots)
该位字段指示 FIFO 中可用的快照数。值等于 FIFO 的深度 (4) 即表示辅助快照 FIFO 已满。当辅助快照 FIFO 清除位置 1 时，这些位清零 (00000)。

位 24 **ATSSTM**：辅助时间戳快照触发未命中 (Auxiliary Timestamp Snapshot Trigger Missed)
辅助时间戳快照 FIFO 已满并且设置了外部触发时，该位置 1。这表明最新的快照未存储在 FIFO 中。

位 23:20 保留，必须保持复位值。

位 19:16 **ATSSTN[3:0]**：辅助时间戳快照触发标识符 (Auxiliary Timestamp Snapshot Trigger Identifier)
这些位标识辅助快照寄存器中的时间戳所适用的辅助触发输入。多个位同时置 1 时，即表示相应的辅助触发采用相同时钟进行采样。只有辅助快照数超过一个时，这些位才适用。为每个触发分配一个位，如下所述：
位 16：辅助触发 0
位 17：辅助触发 1
位 18：辅助触发 2
位 19：辅助触发 3
软件可以读取该寄存器以找到在获取时间戳时设置的触发。

位 15 **TXTSSIS**：Tx 时间戳状态中断状态 (Tx Timestamp Status Interrupt Status)
如果在 MTL 中使能丢弃发送状态，则在 *Tx 时间戳状态纳秒寄存器 (ETH_MACTXTSSNR)* 和 *Tx 时间戳状态秒寄存器 (ETH_MACTXTSSSR)* 中更新捕获的发送时间戳时，该位会置 1。
如果使能 PTP 减荷功能，则在 *Tx 时间戳状态纳秒寄存器 (ETH_MACTXTSSNR)* 和 *Tx 时间戳状态秒寄存器 (ETH_MACTXTSSSR)* 中更新捕获的发送时间戳时，该位会置 1，以用于 PTO 生成的延迟请求和 Pdelay 请求数据包。
该位会在读取 *Tx 时间戳状态秒寄存器 (ETH_MACTXTSSSR)* 时清零（或在 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）。

位 14:4 保留，必须保持复位值。

位 3 **TSTRGTERR0**：时间戳目标时间错误 (Timestamp Target Time Error)
在 ETH_MACPPSTTSR 和 ETH_MACPPSTTSNR 中编程的最新目标时间结束后，此位将置 1（请参见 *PPS 目标时间秒寄存器 (ETH_MACPPSTTSR)* 和 *PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)*）。此位会在应用程序读取此位（或在 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）时清零。

位 2 **AUXTSTRIG**：辅助时间戳触发快照 (Auxiliary Timestamp Trigger Snapshot)
当辅助快照写入 FIFO 时，此位将置为高电平。
此位会在应用程序读取此位（或在 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）时清零。

位 1 **TSTARGT0**：达到时间戳目标时间 (Timestamp Target Time Reached)
该位置 1 时，表示系统时间值大于或等于在 ETH_MACPPSTTSR 和 ETH_MACPPSTTSNR 寄存器中指定的值（请参见 *PPS 目标时间秒寄存器 (ETH_MACPPSTTSR)* 和 *PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)*）。
此位会在应用程序读取此位（或在 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）时清零。

位 0 **TSSOVF**：时间戳秒上溢 (Timestamp Seconds Overflow)
该位置 1 时，表示时间戳的秒值（支持版本 2 格式时）已超过 32'hFFFF_FFFF。
此位会在应用程序读取此位（或在 *CSR 软件控制寄存器 (ETH_MACCSRSWCR)* 中的 RCWE 位置 1 时写入 1）时清零

Tx 时间戳状态纳秒寄存器 (ETH_MACTXTSSNR)

Tx timestamp status nanoseconds register

偏移地址：0x0B30

复位值：0x0000 0000

该寄存器包含在 Tx 状态被禁止时为发送数据包捕获的时间戳的纳秒部分。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXTSSMIS	TXTSSLO[30:16]														
r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXTSSLO[15:0]															
rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r

位 31 **TXTSSMIS**：发送时间戳状态丢失 (Transmit Timestamp Status Missed)
该位置 1 时，表示以下任一项：
– 如果*时间戳控制寄存器 (ETH_MACTSCR)* 的 TXTSSTSM 位复位，则会忽略当前数据包的时间戳
– 如果*时间戳控制寄存器 (ETH_MACTSCR)* 的 TXTSSTSM 位置 1，则前一数据包的时间戳将被当前数据包的时间戳覆盖。

位 30:0 **TXTSSLO[30:0]**：发送时间戳状态低位 (Transmit Timestamp Status Low)
该位字段包含发送数据包捕获的时间戳纳秒位字段的 31 位。

Tx 时间戳状态秒寄存器 (ETH_MACTXTSSSR)

Tx timestamp status seconds register

偏移地址：0x0B34

复位值：0x0000 0000

该寄存器包含发送 PTP 数据包时捕获的时间戳（以秒为单位）的高 32 位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXTSSHI[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXTSSHI[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TXTSSHI[31:0]**：发送时间戳状态高位 (Transmit Timestamp Status High)

该位字段包含发送数据包捕获的时间戳秒位字段的低 32 位。

辅助控制寄存器 (ETH_MACACR)

Auxiliary control register

偏移地址：0x0B40

复位值：0x0000 0000

辅助时间戳控制寄存器控制辅助时间戳快照。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ATSEN3	ATSEN2	ATSEN1	ATSEN0	Res.	Res.	Res.	ATSFC
								rw	rw	rw	rw				rw

位 31:8 保留，必须保持复位值。

位 7 **ATSEN3**：辅助快照 3 使能 (Auxiliary Snapshot 3 Enable)

– 该位控制辅助快照触发 3 的捕获。该位置 1 时，会使能 eth_ptp_trg3 输入上的事件辅助快照。该位复位时，此输入上的事件将被忽略。

位 6 **ATSEN2**：辅助快照 2 使能 (Auxiliary Snapshot 2 Enable)

– 该位控制辅助快照触发 2 的捕获。该位置 1 时，会使能 eth_ptp_trg2 输入上的事件辅助快照。该位复位时，此输入上的事件将被忽略。

位 5 **ATSEN1**：辅助快照 1 使能 (Auxiliary Snapshot 1 Enable)

– 该位控制辅助快照触发 1 的捕获。该位置 1 时，会使能 eth_ptp_trg1 输入上的事件辅助快照。该位复位时，此输入上的事件将被忽略。

位 4 **ATSEN0**：辅助快照 0 使能 (Auxiliary Snapshot 0 Enable)

该位控制辅助快照触发 0 的捕获。该位置 1 时，会使能 eth_ptp_trg0 输入上的事件辅助快照。该位复位时，此输入上的事件将被忽略。

位 3:1 保留，必须保持复位值。

位 0 **ATSFC**：辅助快照 FIFO 清除 (Auxiliary Snapshot FIFO Clear)
该位置 1 时，会复位辅助快照 FIFO 的指针。当指针复位并且 FIFO 为空时，该位清零。该位为高电平时，辅助快照存储在 FIFO 中。

辅助时间戳纳秒寄存器 (ETH_MACATSNR)

Auxiliary timestamp nanoseconds register

偏移地址：0x0B48

复位值：0x0000 0000

辅助时间戳纳秒寄存器 (ETH_MACATSNR) 与 *辅助时间戳秒寄存器 (ETH_MACATSSR)* 一起提供以辅助快照形式存储的 64 位时间戳。这两个寄存器构成深度为 4 个字的 64 位宽 FIFO 的读端口。

可将多个快照存储在此 FIFO 中。*时间戳状态寄存器 (ETH_MACTSSR)* 中的位 [29:25] 表示 FIFO 的填充级别。仅当读取 *辅助时间戳秒寄存器 (ETH_MACATSSR)* 时，才会将 FIFO 的顶部删除。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AUXTSLO[30:16]														
	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AUXTSLO[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31 保留，必须保持复位值。

位 30:0 **AUXTSLO[30:0]**：辅助时间戳 (Auxiliary Timestamp)
包含辅助时间戳的低 31 位（纳秒位字段）。

辅助时间戳秒寄存器 (ETH_MACATSSR)

Auxiliary timestamp seconds register

偏移地址：0x0B4C

复位值：0x0000 0000

辅助时间戳秒寄存器包含辅助时间戳寄存器秒位字段的低 32 位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AUXTSHI[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AUXTSHI[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **AUXTSHI[31:0]**：辅助时间戳 (Auxiliary Timestamp)
包含辅助时间戳秒位字段的低 32 位。

时间戳入站不对称校准寄存器 (ETH_MACTSIACR)

Timestamp Ingress asymmetric correction register

偏移地址：0x0B50

复位值：0x0000 0000

MAC 时间戳入站不对称校准寄存器包含要在更新 PDelay_Resp PTP 消息中的校准位字段时使用的入站不对称校准值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSTIAC[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSTIAC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **OSTIAC[31:0]**：单步时间戳入站不对称校准 (One-Step Timestamp Ingress Asymmetry Correction)

该位字段包含要添加到 Pdelay_Resp PTP 数据包的校准位字段的入站路径不对称值。编程值应以纳秒为单位，并乘以 2^16。例如，2.5 ns 表示为 0x00028000。

该值也可以是负数，它以 2 的补码形式表示，位 31 表示符号位。

时间戳出站不对称校准寄存器 (ETH_MACTSEACR)

Timestamp Egress asymmetric correction register

偏移地址：0x0B54

复位值：0x0000 0000

MAC 时间戳出站不对称校准寄存器包含要在更新 PDelay_Req PTP 消息中的校准位字段时使用的出站不对称校准值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSTEAC[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSTEAC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **OSTEAC[31:0]**：单步时间戳出站不对称校准 (One-Step Timestamp Egress Asymmetry Correction)

该位字段包含要从 Pdelay_Resp PTP 数据包的校准位字段中减去的出站路径不对称值。编程值必须为负值（以纳秒为单位），并乘以 2^16。

例如，如果所需校准值为 +2.5 ns，则编程值必须为 0xFFFD_8000，它是 0x0002_8000 (2.5 * 2^16) 的 2 的补码。类似地，如果所需校准值为 -3.3 ns，则编程值为 0x0003_4CCC (3.3 * 2^16)。

时间戳入站校准纳秒寄存器 (ETH_MACTSICNR)

Timestamp Ingress correction nanosecond register

偏移地址：0x0B58

复位值：0x0000 0000

该寄存器包含要与入站路径中捕获的时间戳值一起使用的校准值（以纳秒为单位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSIC[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **TSIC[31:0]**：时间戳入站校准 (Timestamp Ingress Correction)
该位字段包含由入站校准表达式定义的入站路径校准值。

时间戳出站校准纳秒寄存器 (ETH_MACTSECNR)

Timestamp Egress correction nanosecond register

偏移地址：0x0B5C

复位值：0x0000 0000

该寄存器包含要与出站路径中捕获的时间戳值一起使用的校准值（以纳秒为单位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSEC[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSEC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **TSEC[31:0]**：时间戳出站校准 (Timestamp Egress Correction)
该位字段包含由出站校准表达式定义的出站路径校准值的纳秒部分。

PPS 控制寄存器 (ETH_MACPPSCR)

PPS control register

偏移地址：0x0B70

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRGTMODSEL0[1:0]		PPSEN0	PPSCTRL[3:0]			
									rw	rw	rw	rw	rw	rw	rw

位 31:7 保留，必须保持复位值。

位 6:5 **TRGTMODSEL0[1:0]**：PPS 输出的目标时间寄存器模式 (Target Time Register Mode for PPS Output)

该位字段指示 PPS 输出信号的目标时间寄存器（*PPS 目标时间秒寄存器 (ETH_MACPPSTTSR)* 和 *PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)*）模式：

00：仅为生成中断事件而编程目标时间寄存器。

01：保留，不得使用

10：为生成中断事件以及开始或停止 PPS 输出信号生成而编程目标时间寄存器。

11：仅为开始或停止 PPS 输出信号生成而编程目标时间寄存器。未将任何中断置为有效。

位 4 **PPSEN0**：灵活 PPS 输出模式使能 (Flexible PPS Output Mode Enable)

该位置 1 时，PPSCTRL[3:0] 用作 PPSCMD[3:0]。该位复位时，PPSCTRL[3:0] 用作 PPSCTRL（固定 PPS 模式）。

位 3:0 **PPSCTRL[3:0]**：PPS 输出频率控制 (PPS Output Frequency Control)

该位字段控制 PPS 输出 (eth_ptp_pps_out) 信号的频率。PPSCTRL 的默认值为 0000，PPS 输出为每秒 1 个脉冲（宽度为 clk_ptp_i）。当 PPSCTRL 为其他值时，PPS 输出则为所生成的以下频率的时钟：

0001：二进制翻转为 2 Hz，数字翻转为 1 Hz。

0010：二进制翻转为 4 Hz，数字翻转为 2 Hz。

0011：二进制翻转为 8 Hz，数字翻转为 4 Hz。

0100：二进制翻转为 16 Hz，数字翻转为 8 Hz。

..

1111：二进制翻转为 32.768 KHz，数字翻转为 16.384 KHz。

注意：对于这些频率，在二进制翻转模式下，PPS 输出 (eth_ptp_pps_out) 的占空比为 50%。在数字翻转模式下，PPS 输出频率为平均数。实际时钟的频率不同，每秒同步一次。例如：

- *PPSCTRL = 0001 时，PPS (1 Hz) 具有 537 ms 的低电平周期和 463 ms 的高电平周期。*
- *PPSCTRL = 0010 时，PPS (2 Hz) 为如下序列：*
  第一个时钟的占空比为 50%，周期为 537 ms
  第二个时钟的周期为 463 ms（268 ms 低电平，195 ms 高电平）。
- *PPSCTRL = 0011 时，PPS (4 Hz) 为如下序列：*
  前三个时钟的占空比为 50%，周期为 268 ms
  第四个时钟的周期为 195 ms（134 ms 低电平，61 ms 高电平）

此行为是因为在 ETH_MACSTNR 寄存器中于数字翻转模式下对位进行非线性翻转而引起。

PPS 控制寄存器 [复用] (ETH_MACPPSCR)

PPS control register

偏移地址：0x0B70

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRGTMODSEL0[1:0]		PPSEN0	PPSCMD[3:0]			
									rw	rw	rw	rw	rw	rw	rw

位 31:7 保留，必须保持复位值。

位 6:5 **TRGTMODSEL0[1:0]**：PPS 输出的目标时间寄存器模式 (Target Time Register Mode for PPS Output)

该位字段指示 PPS 输出信号的目标时间寄存器（*PPS 目标时间秒寄存器 (ETH_MACPPSTTSR)* 和 *PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)*）模式：

00：仅为生成中断事件而编程目标时间寄存器。

01：保留，不得使用

10：为生成中断事件以及开始或停止 PPS 输出信号生成而编程目标时间寄存器。

11：仅为开始或停止 PPS 输出信号生成而编程目标时间寄存器。未将任何中断置为有效。

位 4 **PPSEN0**：灵活 PPS 输出模式使能 (Flexible PPS Output Mode Enable)

该位置 1 时，位 [3:0] 用作 PPSCMD[3:0]。该位复位时，位 [3:0] 用作 PPSCTRL（固定 PPS 模式）。

位 3:0 **PPSCMD[3:0]**：灵活 PPS 输出 (eth_ptp_pps_out) 控制 (Flexible PPS Output (eth_ptp_pps_out) Control)

为这些位编程非零值将命令 MAC 发起事件。当命令被传送或同步到 PTP 时钟域时，这些位将自动清零。软件应确保仅在这些位为“全零”时才进行编程。下面对 PPSCMD0 的值进行了说明：

0000：无命令

0001：启动单脉冲。

该命令会生成单脉冲，该脉冲在目标时间寄存器（寄存器 455 和 456）中定义的起始点上升，持续时间为 PPS 宽度寄存器中所定义。

0010：启动脉冲串。

该命令会生成脉冲串，该脉冲串在目标时间寄存器中定义的起始点上升，持续时间为 PPS 宽度寄存器中所定义，并以 PPS 间隔寄存器中定义的间隔进行重复。除非通过“经过一段时间后停止脉冲串”或“立即停止脉冲串”命令停止 PPS 脉冲串，否则默认情况下，PPS 脉冲串自由运行。

0011：取消启动。

如果系统时间未超过编程的启动时间，该命令将取消启动单脉冲和启动脉冲串命令。

0100：经过一段时间后停止脉冲串。

目标时间寄存器中编程的时间结束后，该命令将停止由启动脉冲串命令 (PPSCMD[3:0] = 0010) 启动的脉冲串。

0101：立即停止脉冲串。

该命令会立即停止由启动脉冲串命令 (PPSCMD[3:0] = 0010) 启动的脉冲串。

0110：取消停止脉冲串。

如果编程的停止时间尚未结束，该命令将取消经过一段时间后停止脉冲串命令。PPS 脉冲串在该命令成功执行后将自由运行。

0111 到 1111：保留，不得使用

PPS 目标时间秒寄存器 (ETH_MACPPSTTSR)

PPS target time seconds register

偏移地址：0x0B80

复位值：0x0000 0000

PPS 输出目标时间秒寄存器与 *PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)* 一起，用于在系统时间超过这些寄存器中编程的值时规划中断事件（*时间戳状态寄存器 (ETH_MACTSSR)* 的位 TSSOVF）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSTRH0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTRH0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **TSTRH0[31:0]**：PPS 目标时间秒寄存器 (PPS Target Time Seconds Register)

该位字段存储以秒为单位的时间。当时间戳值匹配或超过两个目标时间戳寄存器时，MAC 会启动或停止 PPS 信号输出，并根据在 *PPS 控制寄存器 (ETH_MACPPSCR)* 中为相应 PPS 输出选择的目标时间模式生成中断（如果使能）。

PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)

PPS target time nanoseconds register

偏移地址：0x0B84

复位值：0x0000 0000

仅在选择多个灵活 PPS 输出时，才存在 PPS 目标时间纳秒寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRGTBUSY0	TTSL0[30:16]														
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTSL0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **TRGTBUSY0**：PPS 目标时间寄存器忙 (PPS Target Time Register Busy)

将 *PPS 控制寄存器 (ETH_MACPPSCR)* 中的 PPSCMD0 位字段编程为 010 或 011 时，MAC 会将该位置 1。将 PPSCMD0 位字段编程为 010 或 011 时，会命令 MAC 将目标时间寄存器同步到 PTP 时钟域。

在将目标时间寄存器同步到 PTP 时钟域之后，MAC 会将该位清零。当该位读为 1 时，应用程序不得更新目标时间寄存器。否则，先前编程的时间同步会损坏。

位 30:0 **TTSL0[30:0]**：PPS 的目标时间低位寄存器 (Target Time Low for PPS Register)

该寄存器存储以纳秒为单位的时间（带符号）。当时间戳的值与两个目标时间戳寄存器中的值匹配时，MAC 将启动或停止 PPS 信号输出，并根据 *PPS 控制寄存器 (ETH_MACPPSCR)* 中的 TRGTMODSEL0 位字段（位 [6:5]）生成中断（如果使能）。

时间戳控制寄存器 (ETH_MACTSCR) 中的 TSCTRLSSR 位置 1 时，该值不应超过 0x3B9A_C9FF。PPS 信号输出的实际启动或停止时间最高可能具有一个单位的亚秒增量值的误差。

PPS 间隔寄存器 (ETH_MACPPSIR)

PPS interval register

偏移地址：0x0B88

复位值：0x0000 0000

PPS 间隔寄存器包含 PPS 输出的上升沿之间的亚秒增量值的单位数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PPSINT0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPSINT0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **PPSINT0[31:0]**：PPS 输出信号间隔 (PPS Output Signal Interval)

这些位存储 PPS 信号输出的上升沿之间的间隔。间隔以亚秒增量值的单位数进行存储。

需要编程一个小于所需间隔的值。例如，如果 PTP 参考时钟为 50 MHz（周期为 20 ns），并且 PPS 信号输出的上升沿之间的预期间隔为 100 ns（即，5 个单位的亚秒增量值），则应在该寄存器中编程值 4 (5-1)。

PPS 宽度寄存器 (ETH_MACPPSWR)

PPS width register

偏移地址：0x0B8C

复位值：0x0000 0000

PPS 宽度寄存器包含 PPS 输出的上升沿和相应下降沿之间的亚秒增量值的单位数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PPSWIDTH0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPSWIDTH0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **PPSWIDTH0[31:0]**：PPS 输出信号宽度 (PPS Output Signal Width)

这些位存储 PPS 信号输出的上升沿和相应下降沿之间的宽度。宽度以亚秒增量值的单位数进行存储。

需要编程一个小于所需间隔的值。例如，如果 PTP 参考时钟为 50 MHz（周期为 20 ns），并且 PPS 信号输出的上升沿和相应下降沿之间的宽度为 80 ns（即，4 个单位的亚秒增量值），则应在该寄存器中编程值 3 (4-1)。

注意：该寄存器中编程的值必须小于 PPS 间隔寄存器 (ETH_MACPPSIR) 中编程的值。

PTP 减荷控制寄存器 (ETH_MACPOCR)

PTP Offload control register

偏移地址：0x0BC0

复位值：0x0000 0000

该寄存器控制 PTP 减荷引擎操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DN[7:0]								Res.	DRRDIS	APDREQTRIG	ASYNCTRIG	Res.	APDREQEN	ASYNCEN	PTOEN
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw		rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:8 **DN[7:0]**：域号 (Domain Number)

该位字段指示 PTP 节点正在使用的域号。

位 7 保留，必须保持复位值。

位 6 **DRRDIS**：禁止 PTO 延迟请求/响应生成 (Disable PTO Delay Request/Response response generation)

如果该位置 1，则会按照所编程模式的要求，分别在接收到 SYNC 和延迟请求数据包时不生成延迟请求和延迟响应。

位 5 **APDREQTRIG**：自动 PTP Pdelay_Req 消息触发 (Automatic PTP Pdelay_Req message Trigger)
该位置 1 时，会发送一条 PTP Pdelay_Req 消息。发送 PTP Pdelay_Req 消息后，该位自动清零。为实现此操作，应用程序应将 APDREQEN 位置 1。

位 4 **ASYNCTRIG**：自动 PTP SYNC 消息触发 (Automatic PTP SYNC message Trigger)
该位置 1 时，会发送一条 PTP SYNC 消息。发送 PTP SYNC 消息后，该位自动清零。为实现此操作，应用程序应将 ASYNCEN 位置 1。

位 3 保留，必须保持复位值。

位 2 **APDREQEN**：自动 PTP Pdelay_Req 消息使能 (Automatic PTP Pdelay_Req message Enable)
该位置 1 时，如果将 MAC 编程为点对点透明模式，则会基于编程设定的间隔或在应用程序的触发下周期性地生成 PTP Pdelay_Req 消息。

位 1 **ASYNCEN**：自动 PTP SYNC 消息使能 (Automatic PTP SYNC message Enable)
该位置 1 时，如果将 MAC 编程为时钟主模式，则会基于编程设定的间隔或在应用程序的触发下周期性地生成 PTP SYNC 消息。

位 0 **PTOEN**：PTP 减荷使能 (PTP Offload Enable)
此位置 1 时，会使能 PTP 减荷功能。

PTP 源端口标识 0 寄存器 (ETH_MACSPI0R)

PTP Source Port Identity 0 Register

偏移地址：0x0BC4

复位值：0x0000 0000

该寄存器包含 PTP 节点的 80 位源端口标识的位 [31:0]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPI0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **SPI0[31:0]**：源端口标识 0 (Source Port Identity 0)
该位字段指示 PTP 节点的源端口标识的位 [31:0]。

PTP 源端口标识 1 寄存器 (ETH_MACSPI1R)

PTP Source port identity 1 register

偏移地址：0x0BC8

复位值：0x0000 0000

该寄存器包含 PTP 节点的 80 位源端口标识的位 [63:32]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPI1[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **SPI1[31:0]**：源端口标识 1 (Source Port Identity 1)
该位字段指示 PTP 节点的源端口标识的位 [63:32]。

PTP 源端口标识 2 寄存器 (ETH_MACSPI2R)

PTP Source port identity 2 register

偏移地址：0x0BCC

复位值：0x0000 0000

该寄存器包含 PTP 节点的 80 位源端口标识的位 [79:64]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **SPI2[15:0]**：源端口标识 2 (Source Port Identity 2)
该位字段指示 PTP 节点的源端口标识的位 [79:64]。

日志消息间隔寄存器 (ETH_MACLMIR)

Log message interval register

偏移地址：0x0BD0

复位值：0x0000 0000

该寄存器包含用于自动生成 PTP 数据包的周期性间隔。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LMPDRI[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	DRSYNCR[2:0]			LSI[7:0]							
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **LMPDRI[7:0]**：日志最小 Pdelay_Req 间隔 (Log Min Pdelay_Req Interval)

该位字段指示 PTP 节点的 logMinPdelayReqInterval。它用于调度周期性的 Pdelay 请求数据包发送。允许值为 -15 到 15。负值必须以 2 的补码形式表示。例如，如果所需值为 -1，则编程的值必须为 0xFF。

位 23:11 保留，必须保持复位值。

位 10:8 **DRSYNCR[2:0]**：

Delay_Req 与 SYNC 之比 (Delay_Req to SYNC Ratio)
在从模式下，它用于控制发送的 Delay_Req 消息的频率。
0：每次接收到 SYNC 都会生成一条 DelayReq 消息
1：每接收到 2 个 SYNC 会生成一条 DelayReq 消息
2：每接收到 4 个 SYNC 会生成一条 DelayReq 消息
3：每接收到 8 个 SYNC 会生成一条 DelayReq 消息
4：每接收到 16 个 SYNC 会生成一条 DelayReq 消息
5：每接收到 32 个 SYNC 会生成一条 DelayReq 消息
其他值：保留，不得使用
主节点将 DelayResp PTP 消息中的此信息 (logMinDelayReqInterval) 发送到从节点。接收端会对接收到的 DelayResp 消息中的此值进行处理，并相应地更新此位字段。在从模式下，主机不得对该寄存器进行写入/更新操作，除非其必须覆盖接收到的值。在主模式下，该位字段与 logSyncInterval (LSI) 位字段的总和在生成的多播 Delay_Resp PTP 消息的 logMinDelayReqInterval 位字段中提供。

位 7:0 **LSI[7:0]**：

日志同步间隔
该位字段指示 PTP 节点为主节点时自动生成 SYNC 消息的周期。允许值为 -15 到 15。负值必须以 2 的补码形式表示。例如，如果所需值为 -1，则编程的值必须为 0xFF。

以太网 MAC 寄存器映射和复位值

表 586. 以太网 MAC 寄存器映射和复位值

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	ETH_MACCR	ARPEN	SARC[2:0]			IPC	IPG[2:0]			GPSLCE	S2KP	CST	ACS	WD	BE	JD	JE	PS	FES	DM	LM	ECRSFD	DO	DCRS	DR	Res.	BL[1:0]		DC	PRELEN[1:0]		TE	RE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x0004	ETH_MACECR	Res.	Res.	EIPG[4:0]					EIPGEN	Res.	Res.	Res.	Res.	Res.	USP	SPEN	DCRCC	Res.	Res.	GPSL[13:0]													
	Reset value			0	0	0	0	0	0						0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0008	ETH_MACPFR	RA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DNTU	IPFE	Res.	Res.	Res.	VTFE	Res.	Res.	Res.	Res.	Res.	HPF	SAF	SAIF	PCF[1:0]		DBF	PM	DAIF	HMC	HUC	PR
	Reset value	0										0	0				0						0	0	0	0	0	0	0	0	0	0	0
0x000C	ETH_MACWTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PWE	Res.	Res.	Res.	Res.	WTO[3:0]			
	Reset value																								0					0	0	0	0
0x0010	ETH_MACHT0R	HT31T0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0014	ETH_MACHT1R	HT63T32[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018 - 0x004C	Reserved																																
0x0050	ETH_MACVTR	EIVLRXS	Res.	EIVLS[1:0]		ERIVLT	EDVLP	VTHM	EVLRXS	Res.	EVLS[1:0]		DOVLTC	ERSVLM	ESVL	VTIM	ETV	VL[15:0]																
	Reset value	0		0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0054	Reserved																																
0x0058	ETH_MACVHTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VLHT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x005C	Reserved																																
0x0060	ETH_MACVIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VLTI	CSVL	VLP	VLC[1:0]		VLT[15:0]															
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 586. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0064	ETH_MACiVIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VLTI	CSVL	VLP	VLC[1:0]		VLT[15:0]															
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0068 - 0x006C	Reserved																																
0x0070	ETH_ MACQ0TXFCR	PT[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DZPQ	PLT[2:0]			Res.	Res.	TFE	FCB_BPA
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									0	0	0	0			0	0
0x0074 - 0x008C	Reserved																																
0x0090	ETH_MACRXFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UP	RFE
	Reset value																															0	0
0x0094	ETH_MACRXQCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VFFQ	VFFQE	Res.	Res.	Res.	Res.	Res.	Res.	MFFQ	MFFQE	Res.	Res.	Res.	Res.	Res.	Res.	UFFQ	UFFQE
	Reset value															0	0							0	0							0	0
0x0098- 0x009C	Reserved																																
0x00A0	ETH_ MACRXQC0R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXQ1EN[1:0]		RXQ0EN[1:0]	
	Reset value																													0	0	0	0
0x00A4	ETH_ MACRXQC1R	Res.	Res.	Res.	Res.	Res.	Res.			Res.	TPQC	TACPQE	MCBCQEN	Res.	MCBCQ[2:0]			Res.	UPQ[2:0]			Res.	Res.	Res.	Res.	Res.	PTPQ[2:0]			Res.	Res.		
	Reset value										0	0	0		0	0	0		0	0	0						0	0	0				
0x00A8	ETH_ MACRXQC2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSRQ1[7:0]								PSRQ0[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00AC	Reserved																																
0x00B0	ETH_MACISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXSTSIS	TXSTSIS	TSIS	Res.	MMCTXIS	MMCRXIS	MMCIS	Res.	Res.	LPIIS	PMTIS	PHYIS	Res.	Res.	RGSMIII
	Reset value																		0	0	0		0	0	0			0	0	0			0

表 586. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00B4	ETH_MACIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXSTSIE	TXSTSIE	TSIE	Res.	Res.	Res.	Res.	Res.	Res.	LPIIE	PMTIE	PHYIE	Res.	Res.	RGSMIIIE
	Reset value																		0	0	0							0	0	0			0
0x00B8	ETH_ MACRXTXSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWT	Res.	Res.	EXCOL	LCOL	EXDEF	LCARR	NCARR	TJT
	Reset value																								0			0	0	0	0	0	0
0x00BC	Reserved																																
0x00C0	ETH_MACPCSR	RWKFILTRST	Res.	Res.	RWKPTR[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWKPFE	GLBLUCAST	Res.	Res.	RWKPRCVD	MGKPRCVD	Res.	Res.	RWKPKTEN	MGKPKTEN	PWRDWN
	Reset value	0			0	0	0	0	0														0	0			0	0			0	0	0
0x00C4	ETH_ MACRWKPFR	MACRWKPFR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C8 - 0x00CC	Reserved																																
0x00D0	ETH_MACLCSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPITCSE	LPITE	LPITXA	PLSEN	PLS	LPIEN	Res.	Res.	Res.	Res.	Res.	Res.	RLPIST	TLPIST	Res.	Res.	Res.	Res.	RLPIEX	RLPIEN	TLPIEX	TLPIEN
	Reset value											0	0	0	0	0	0							0	0					0	0	0	0
0x00D4	ETH_MACLTCR	Res.	Res.	Res.	Res.	Res.	Res.	LST[9:0]										TWT[15:0]															
	Reset value							1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00D8	ETH_MACLETR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPIET[19:0]																			
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00DC	ETH_ MAC1USTCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIC_1US_CNTR[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x00E0 - 0x00F4	Reserved																																
0x00F8	ETH_ MACPHYCSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LNKSTS	LNKSPEED[1:0]		LNKMOD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LUD	TC
	Reset value													0	0	0	0															0	0
0x00FC - 0x010C	Reserved																																

表 586. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0110	ETH_MACVR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USERVER[7:0]								SNPSVER[7:0]							
	Reset value																	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0
0x0114	ETH_MACDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TFCSTS[1:0]		TPESTS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFCFCSTS[1:0]		RPESTS
	Reset value														0	0	0														0	0	0
0x0118	Reserved																																
0x011C	ETH_MACHWF0R	Res.	ACTPHYSEL[2:0]			SAVLANINS	TSTSSEL[1:0]		MACADR64SEL	MACADR32SEL	ADDMACADRSEL[4:0]					Res.	RXCOESEL	Res.	TXCOESEL	EEESEL	TSSEL	Res.	Res.	ARPOFFSEL	MMCSEL	MGKSEL	RWKSEL	SMASEL	VLHASH	PCSSEL	HDSEL	GMIISEL	MIISEL
	Reset value		0	0	0	1	0	1	0	0	0	0	0	1	1		1		1	1	1			1	1	1	1	1	1	0	1	1	1
0x0120	ETH_MACHWF1R	Res.	L3L4FNUM[3:0]				Res.	HASHTBLSZ[1:0]		POUOST	Res.	RAVSEL	AVSEL	DBGMEMA	TSOEN	SPHEN	DCBEN	ADDR64[1:0]		ADVTHWORD	PTOEN	OSTEN	TXFIFOSIZE[4:0]					Res.	RXFIFOSIZE[4:0]				
	Reset value		0	0	1	0		0	1	0		0	0	0	1	0	0	0	0	0	1	1	0	0	1	0	0		0	0	1	0	0
0x0124	ETH_MACHWF2R	Res.	AUXSNAPNUM[2:0]			Res.	PPSOUTNUM[2:0]			TDCSZ[1:0]		TXCHCNT[3:0]				RDCSZ[1:0]		RXCHCNT[3:0]				Res.	Res.	TXQCNT[3:0]				Res.	Res.	RXQCNT[3:0]			
	Reset value		1	0	0		0	0	1	0	0	0	0	0	1	0	0	0	0	0	0			0	0	0	1			0	0	0	1
0x0128	ETH_MACHWF3R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DVLAN	CBTISEL	Res.	NRVF[2:0]		
	Reset value																											1	0		0	0	0
0x012C - 0x01FC	Reserved																																
0x0200	ETH_ MACMDIOAR	Res.	Res.	Res.	Res.	PSE	BTB	PA[4:0]					RDA[4:0]					Res.	NTC[2:0]			CR[3:0]				Res.	Res.	Res.	SKAP	GOC[1:0]		C45E	GB
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0				0	0	0	0	0

表 586. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0204	ETH_ MACMDIODR	RA[15:0]																GD[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0208 - 0x020C	Reserved																																
0x0210	ETH_MACARPAR	ARPPA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0214 - 0x022C	Reserved																																
0x0230	ETH_MACCSRSW CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RCWE
	Reset value																								0								0
0x0234 - 0x02FC	Reserved																																
0x0300	ETH_MACA0HR	AE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRHI[15:0]															
	Reset value	1																1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0304	ETH_MACA0LR	ADDRLO[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0308	ETH_MACA1HR	AE	SA	MBC[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRHI[15:0]															
	Reset value	0	0	0	0	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30C	ETH_MACA1LR	ADDRLO[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0310	ETH_MACA2HR	AE	SA	MBC[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRHI[15:0]															
	Reset value	0	0	0	0	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0314	ETH_MACA2LR	ADDRLO[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

表 586. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0318	ETH_MACA3HR	AE	SA	MBC[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRHI[15:0]															
	Reset value	0	0	0	0	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x031C	ETH_MACA3LR	ADDRLO[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0320 - 0x06FC	Reserved																																
0x0700	ETH_MMC_ CONTROL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UCDBC	Res.	Res.	CNTPRSTLVL	CNTPRST	CNTFREEZ	RSTONRD	CNTSTOPRO	CNTRST
	Reset value																								0			0	0	0	0	0	0
0x0704	ETH_MMC_RX_ INTERRUPT	Res.	Res.	Res.	Res.	RXLPITRCIS	RXLPIUSCIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXUCGPIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXALGNERPIS	RXCRCERPIS	Res.	Res.	Res.	Res.	Res.
	Reset value					0	0									0											0	0					
0x0708	ETH_MMC_TX_ INTERRUPT	Res.	Res.	Res.	Res.	TXLPITRCIS	TXLPIUSCIS	Res.	Res.	Res.	TXGPKTIS	Res.	Res.	Res.	Res.	Res.	Res.	TXMCOLGPIS	TXSCOLGPIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value					0	0				0							0	0														
0x070C	ETH_MMC_RX_ INTERRUPT_ MASK	Res.	Res.	Res.	Res.	RXLPITRCIM	RXLPIUSCIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXUCGPIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXALGNERPIM	RXCRCERPIM	Res.	Res.	Res.	Res.	Res.
	Reset value					0	0									0											0	0					
0x0710	ETH_MMC_TX_ INTERRUPT_ MASK	Res.	Res.	Res.	Res.	TXLPITRCIM	TXLPIUSCIM	Res.	Res.	Res.	TXGPKTIM	Res.	Res.	Res.	Res.	Res.	Res.	TXMCOLGPIM	TXSCOLGPIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value					0	0				0							0	0														
0x0714 - 0x0748	Reserved																																

表 586. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x074C	ETH_TX_SINGLE_COLLISION_GOOD_PACKETS	TXSNGLCOLG[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0750	ETH_TX_MULTIPLE_COLLISION_GOOD_PACKETS	TXMULTCOLG[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0754 - 0x0764	Reserved																																
0x0768	ETH_TX_PACKET_COUNT_GOOD	TXPKTG[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x076C - 0x0790	Reserved																																
0x0794	ETH_RX_CRC_ERROR_PACKETS	RXCRCERR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0798	ETH_RX_ALIGNMENT_ERROR_PACKETS	RXALGNERR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x079C - 0x07C0	Reserved																																
0x07C4	ETH_RX_UNICAST_PACKETS_GOOD	RXUCASTG[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07C8 - 0x07E8	Reserved																																
0x07EC	ETH_TX_LPI_USEC_CNTR	TXLPIUSC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07F0	ETH_TX_LPI_TRAN_CNTR	TXLPITRC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07F4	ETH_RX_LPI_USEC_CNTR	RXLPIUSC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07F8	ETH_RX_LPI_TRAN_CNTR	RXLPITRC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 586. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x07FC - 0x08FC	Reserved																																
0x0900	ETH_ MACL3L4C0R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	L4DPIM0	L4DPM0	L4SPIM0	L4SPM0	Res.	L4PEN0	L3HDBM0[4:0]					L3HSBM0[4:0]					L3DAIM0	L3DAM0	L3SAIM0	L3SAM0	Res.	L3PEN0
	Reset value											0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x0904	ETH_MACL4A0R	L4DP0[15:0]																L4SP0[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0908 - 0x090C	Reserved																																
0x0910	ETH_MACL3A00R	L3A00[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0914	ETH_MACL3A10R	L3A10[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0918	ETH_MACL3A20R	L3A20[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x091C	ETH_MACL3A30R	L3A30[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0920 - 0x092C	Reserved																																
0x0930	ETH_ MACL3L4C1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	L4DPIM1	L4DPM1	L4SPIM1	L4SPM1	Res.	L4PEN1	L3HDBM1[4:0]					L3HSBM1[4:0]					L3DAIM1	L3DAM1	L3SAIM1	L3SAM1	Res.	L3PEN1
	Reset value											0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x0934	ETH_MACL4A1R	L4DP1[15:0]																L4SP1[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0938 - 0x093C	Reserved																																
0x0940	ETH_MACL3A01R	L3A01[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0944	ETH_MACL3A11R	L3A11[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0948	ETH_MACL3A21R	L3A21[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x094C	ETH_MACL3A31R	L3A31[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 586. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0950 - 0x0AFC	Reserved																																
0x0B00	ETH_MACTSCR	Res.	Res.	Res.	AV8021ASMEN	Res.	Res.	Res.	TXTSSTSM	Res.	Res.	Res.	Res.	Res.	TSENMACADDR	SNAPTYPSEL[1:0]		TSMSTRENA	TSEVNTENA	TSIPV4ENA	TSIPV6ENA	TSIPENA	TSVER2ENA	TSCTRLSSR	TSENALL	Res.	Res.	TSADDREG	Res.	TSUPDT	TSINIT	TSCFUPDT	TSENA
	Reset value				0				0						0	0	0	0	0	1	0	0	0	0	0			0		0	0	0	0
0x0B04	ETH_MACSSIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SSINC[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value									0	0	0	0	0	0	0	0																
0x0B08	ETH_MACSTSR	TSS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B0C	ETH_MACSTNR	Res.	TSSS[30:0]																														
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B10	ETH_MACSTSUR	TSS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B14	ETH_MACSTNUR	ADDSUB	TSSS[30:0]																														
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B18	ETH_MACTSAR	TSAR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B1C	Reserved																																
0x0B20	ETH_MACTSSR	Res.	Res.	ATSNS[4:0]					ATSSTM	Res.	Res.	Res.	Res.	ATSSTN[3:0]				TXTSSIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSTRGTERR0	AUXTSTRIG	TSTARGT0	TSSOVF
	Reset value			0	0	0	0	0	0					0	0	0	0	0												0	0	0	0
0x0B24 - 0x0B2C	Reserved																																
0x0B30	ETH_ MACTXTSSNR	TXTSSMIS	TXTSSLO[30:0]																														
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B34	ETH_ MACTXTSSSR	TXTSSHI[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 586. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0B38 - 0x0B3C	Reserved																																
0x0B40	ETH_MACACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ATSEN3	ATSEN2	ATSEN1	ATSEN0	Res.	Res.	Res.	ATSFC
	Reset value																									0	0	0	0				0
0x0B44	Reserved																																
0x0B48	ETH_MACATSNR	Res.	AUXTSLO[30:0]																														
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B4C	ETH_MACATSSR	AUXTSHI[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B50	ETH_MACTSIACR	OSTIAC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B54	ETH_MACTSEACR	OSTEAC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B58	ETH_ MACTSICNR	TSIC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B5C	ETH_ MACTSECNR	TSEC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B60 - 0x0B6C	Reserved																																
0x0B70	ETH_MACPPSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRGTMODSEL0[1:0]		PPSEN0	PPSCTRL[3:0]			
	Reset value																										0	0	0	0	0	0	0
0x0B70	ETH_MACPPSCR (alternate)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRGTMODSEL0[1:0]		PPSEN0	PPSCMD[3:0]			
	Reset value																										0	0	0	0	0	0	0
0x0B74 - 0x0B7C	Reserved																																

表 586. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0B80	ETH_ MACPPSTTSR	TSTRH0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B84	ETH_ MACPPSTTNR	TRGTBUSY0	TTSL0[30:0]																														
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B88	ETH_MACPPSIR	PPSINT0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B8C	ETH_MACPPSWR	PPSWIDTH0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B90 - 0x0BBC	Reserved																																
0x0BC0	ETH_MACPOCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DN[7:0]								Res.	DRRDIS	APDREQTRIG	ASYNCTRIG	Res.	APDREQEN	ASYNCEN	PTOEN
	Reset value																	0	0	0	0	0	0	0	0		0	0	0		0	0	0
0x0BC4	ETH_MACSPI0R	SPI0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0BC8	ETH_MACSPI1R	SPI1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0BCC	ETH_MACSPI2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPI2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0BD0	ETH_MACLMIR	LMPDRI[7:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DRSYNCR[2:0]			LSI[7:0]							
	Reset value	0	0	0	0	0	0	0	0														0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

58 硬件调试端口 (HDP)

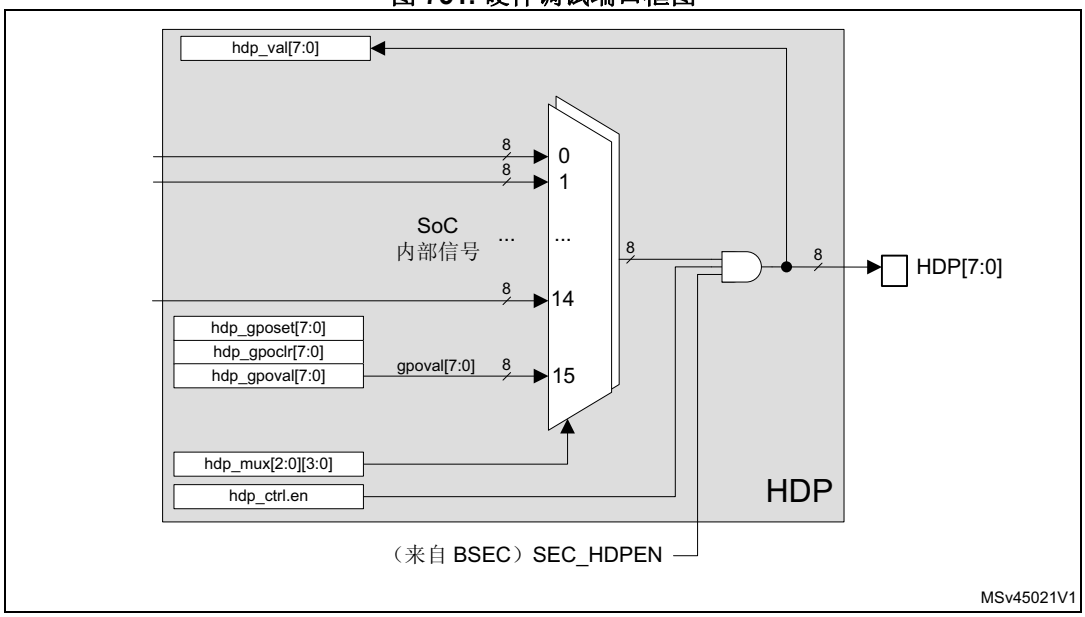
硬件调试端口允许观察内部信号。通过使用复用器，可以观察每一个 8 位输出的最多 16 个信号。

58.1 特性

- 8 个输出信号
- 15 个内部信号中有一个可以独立控制
- 8 个软件可编程信号，用于与引脚排列无关的代码调试
- 通过安全信号禁止输出

58.2 框图

图 751. 硬件调试端口框图



58.3 功能描述

主复用器通过 HDP_CTRL 寄存器进行控制，允许选择 8 信号调试集进行输出。

主复用器输出的 HDP 值可以通过软件读取 HDP_VAL 寄存器。

在调试过程中，输出一些由软件控制的信号通常很有用；这些信号可用来观察精确的代码序列或指示当前执行的任务。为此专门保留了调试集 #15；它在 HDP 输出上提供 HDP_GPOVAL 寄存器的值。该寄存器与设置和清零寄存器（HDP_GPOSET 和 HDP_GPOCLR）相关，无需读-改-写序列即可切换信号。

一些调试集会公开需要保护的内部数据；当 SEC_HDPEN 信号为低电平时，会将所有 HDP 模块输出强制为低电平。

注： *内部信号的切换速率可能会高于引脚所能支持的速率，在这种情况下，它们的可观测性可能会受到影响。用户应注意使用 HDP 端口时的这一限制。*

表 587. HDP 信号复用

HDP_MUX	HDP7	HDP6	HDP5	HDP4	HDP3	HDP2	HDP1	HDP0
	MUX7[3:0]	MUX6[3:0]	MUX5[3:0]	MUX4[3:0]	MUX3[3:0]	MUX2[3:0]	MUX1[3:0]	MUX0[3:0]
0	CA7 STANDBYWFI0	CA7 STANDBYWFE0	CA7 STANDBYWFIL2	-	PWR SEL_VTH_ VDDCORE	PWR pwrwake_mpu	PWR SEL_VTH_ VDDCPU	PWR pwrwake_sys
1	PWR RCC_VCPU_ RDY	PWR VTH_VDDCPU_ ACK	PWR VTH_VDDCORE _ACK	PWR STOP2_ACTIVE	PWR MPU_CLOCK_ DISABLE_REQ	PWR MPU_CLOCK_ DISABLE_ACK	PWR MPU_RAM_ LOWSPEED	PWR STOP_ FORBIDDEN
2	CA7 EVENTI	CA7 EVENTO	CA7 ncorereset_i	CA7 nl2reset_i	CA7 nPMUIRQ0	CA7 ndbgreset_i	CA7 nAXIERRIRQ	PWR STDBY_ WAKEUP
3	CA7 DBGACK0	-	CA7 nIRQOUT0	CA7 nporeset_varm_i	CA7 nFIQOUT0	-	PWR OKIN_MR	PWR ENCOMP_ VDDCORE
4	BSEC 输出 fuse_ok	BSEC 输入 tamper_det	BSEC 输入 pwrok	BSEC 输出 sec_dften	BSEC 输出 sec_dftlock	BSEC 输入 rstcore_n	BSEC 输出 sec_dbgen	BSEC 输出 sec_niden
5	BSEC 输出 sec_spiden	BSEC 输出 sec_spniden	BSEC 输出 sec_deviceen	BSEC 输出 sec_dbgswenable	BSEC 输出 sec_jtag_dis	BSEC 输出 sec_bsc_dis	AIEC c1_wakeup	AIEC sys_wakeup
6	ETH1 输出 mac_speed_o0	ETH1 输出 mac_speed_o1	ETH1 输出 lpi_intr_o	ETH1 输出 pmt_intr_o	RCC pwrds_sys	-	RCC pwrds_mpu	-
7	ETH2 输出 mac_speed_o0	ETH2 输出 mac_speed_o1	ETH2 输出 lpi_intr_o	ETH2 输出 pmt_intr_o	SRAM3CTRL TAMP_ERASE_ REQ	-	-	-
8	DDRCTRL_ csysreq_ddrc	DDRCTRL_ csysack_ddrc	DDRCTRL_ cactive_ddrc	DDRCTRL_stat_ ddrc_reg_selfref_ type[1]	DDRCTRL_stat _ddrc_reg_ selfref_type[0]	DDRCTRL_dfi_ init_complete	DDRCTRL_dfi_ ctrlupd_req	DDRCTRL_lp_ req

表 587. HDP 信号复用（续）

HDP_MUX	HDP7	HDP6	HDP5	HDP4	HDP3	HDP2	HDP1	HDP0
	MUX7[3:0]	MUX6[3:0]	MUX5[3:0]	MUX4[3:0]	MUX3[3:0]	MUX2[3:0]	MUX1[3:0]	MUX0[3:0]
9	DDRCTRL_hpr_credit_cnt[4:0]==0	DDRCTRL_lpr_credit_cnt[4:0]==0	DDRCTRL_wr_credit_cnt[4:0]==0	DDRCTRL_cactive_0	-	DDRCTRL_perf_op_is_refresh	DDRCTRL cactive_ddrc_asr	PWR DDR_RET_ENABLE_N
10	DTS valobus1_4	DTS valobus1_3	DTS valobus1_2	DTS valobus1_1	DTS valobus1_0	DDRCTRL gskp_dfi_lp_req	-	DTS CLK_PTAT
11	DTS valobus2_4	DTS valobus2_3	DTS valobus2_2	DTS valobus2_1	DTS valobus2_0	-	-	-
12	RNG tamper_out	SAES tamper_out	PKA pka_itamp_out	TAMP nreset_sram_ercfg	TAMP potential_tamp_erfcfg	SRAM3CTRL SW_ERASE_ACT	SRAM3CTRL HW_ERASE_ACT	SRAM3CTRL TAMP_ERASE_ACT
13	NIC400_S0 AWAVALID	NIC400_S0 AWREADY	NIC400_S0 WVALID	NIC400_S0 WLAST	NIC400_S0 WREADY	NIC400_S0 BVALID	NIC400_S0 BREADY	-
14	NIC400_S0 ARAVALID	NIC400_S0 ARREADY	NIC400_S0 RVALID	NIC400_S0 RLAST	NIC400_S0 RREADY	-	-	-
15	gpoval[7]	gpoval[6]	gpoval[5]	gpoval[4]	gpoval[3]	gpoval[2]	gpoval[1]	gpoval[0]

58.4 HDP 寄存器

58.4.1 HDP 控制寄存器 (HDP_CTRL)

HDP control register

偏移地址：0x000

复位值：0x0000 0000

HDP 控制

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
															rw

位 31:1 保留，必须保持复位值。

位 0 **EN**：使能 HDP (Enable HDP)
如果在 BSEC 中使能则有效。

58.4.2 HDP 复用器控制寄存器 (HDP_MUX)

HDP multiplexers control register

偏移地址：0x004

复位值：0x0000 0000

HDP 复用

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX7[3:0]				MUX6[3:0]				MUX5[3:0]				MUX4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX3[3:0]				MUX2[3:0]				MUX1[3:0]				MUX0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 **MUX7[3:0]**：从 16 个可用信号中选择 HDP7 输出

位 27:24 **MUX6[3:0]**：从 16 个可用信号中选择 HDP6 输出

位 23:20 **MUX5[3:0]**：从 16 个可用信号中选择 HDP5 输出

位 19:16 **MUX4[3:0]**：从 16 个可用信号中选择 HDP4 输出

位 15:12 **MUX3[3:0]**：从 16 个可用信号中选择 HDP3 输出

位 11:8 **MUX2[3:0]**：从 16 个可用信号中选择 HDP2 输出

位 7:4 **MUX1[3:0]**：从 16 个可用信号中选择 HDP1 输出

位 3:0 **MUX0[3:0]**：从 16 个可用信号中选择 HDP0 输出

58.4.3 HDP 回读值寄存器 (HDP_VAL)

HDP read back value register

偏移地址：0x010

复位值：0x0000 0000

HDP 值

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDPVAL[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **HDPVAL[7:0]**：HDP 回读值
提供 HDP 信号的值。

58.4.4 HDP 通用输出设置寄存器 (HDP_GPOSET)

HDP general purpose output set register

偏移地址：0x014

复位值：0xXXXX XXXX

HDP GPO 设置

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDPGPOSET[7:0]							
								w	w	w	w	w	w	w	w

位 31:8 保留，必须保持复位值。

位 7:0 **HDPGPOSET[7:0]**：HDP 通用输出设置
当在一个位中写入 1 时，相应的 HDP GPO 置 1。若在该位中写入 0 则不会产生影响。

58.4.5 HDP 通用输出清零寄存器 (HDP_GPOCLR)

HDP general purpose output clear register

偏移地址：0x018

复位值：0xXXXX XXXX

HDP GPO 清零

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDPGPOCLR[7:0]							
								w	w	w	w	w	w	w	w

位 31:8 保留，必须保持复位值。

位 7:0 **HDPGPOCLR[7:0]**：HDP 通用输出清除零 (HDP general purpose output clear)
当在一个位中写入 1 时，相应的 HDP GPO 清零。若在该位中写入 0 则不会产生影响。

58.4.6 HDP 通用输出值寄存器 (HDP_GPOVAL)

HDP general purpose output value register

偏移地址：0x01C

复位值：0x0000 0000

HDP GPO 值

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDPGPOVAL[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **HDPGPOVAL[7:0]**：HDP 通用输出值 (HDP general purpose output value)
写入时，定义 HDP GPO 值。读取时，提供当前 GPO 值。

58.4.7 HDP 版本寄存器 (HDP_VERR)

HDP version register

偏移地址：0x3F4

复位值：0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：IP 的主版本 (Major revision of the IP)

位 3:0 **MINREV[3:0]**：IP 的次版本 (Minor revision of the IP)

58.4.8 HDP IP 标识寄存器 (HDP_IPIDR)

HDP IP identification register

偏移地址：0x3F8

复位值：0x0003 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：IP 标识符 (IP Identifier)

58.4.9 HDP 大小标识寄存器 (HDP_SIDR)

HDP size identification register

偏移地址：0x3FC

复位值：0xA3C5 DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **SID[31:0]**：大小标识符 (Size identifier)

58.4.10 HDP 寄存器映射

表 588. HDP 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**HDP_CTRL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
	Reset value																																0
0x004	**HDP_MUX**	MUX7[3:0]				MUX6[3:0]				MUX5[3:0]				MUX4[3:0]				MUX3[3:0]				MUX2[3:0]				MUX1[3:0]				MUX0[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x008 - 0x00C	Reserved																																
0x010	**HDP_VAL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDPVAL[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x014	**HDP_GPOSET**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDPGPOSET[7:0]							
	Reset value																									X	X	X	X	X	X	X	X
0x018	**HDP_GPOCLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDPGPOCLR[7:0]							
	Reset value																									X	X	X	X	X	X	X	X
0x01C	**HDP_GPOVAL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDPGPOVAL[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x020 - 0x3F0	Reserved																																
0x3F4	**HDP_VERR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV [3:0]				MINREV [3:0]			
	Reset value																									0	0	0	1	0	0	0	0
0x3F8	**HDP_IPIDR**	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0x3FC	**HDP_SIDR**	SID[31:0]																															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 147 页的第 2.5 节*。

59 调试支持 (DBG)

59.1 简介

为支持软件开发和系统集成，提供了以下一整套调试功能：

- 针对系统中每个 CPU 内核的独立断点调试
- 代码执行跟踪
- 软件指令
- 交叉触发
- 调试功能可以通过 JTAG/串行线调试访问端口，使用行业标准调试工具进行控制。跟踪端口允许捕获数据以进行记录和分析。

调试功能基于 Arm® CoreSight™ 组件。

- SWJ-DP：JTAG/串行线调试端口
- AXI-AP：AXI 访问端口
- APB-AP：APB 访问端口
- ITM：指令跟踪宏单元
- DWT：数据监测点和跟踪
- ETM：嵌入式跟踪宏单元
- ETF：嵌入式跟踪 FIFO
- TPIU：跟踪端口接口单元
- CTI：交叉触发接口
- CTM：交叉触发矩阵
- 时间戳发生器

这些组件在*第 59.10 节*和*第 59.11 节*中进行介绍。更多相关信息，请参见*第 59.12 节*中引用的 Arm® 文档。

59.2 调试用例

跟踪和调试系统被设计为支持各种典型用例：

- 外部调试

  处理器可以使用连接到 JTAG/SWD 调试端口的设备进行调试，可以支持断点和监测点设置、逐步执行代码以及内存访问等。

- 自托管调试

  可通过在内核上运行软件来调试处理器。可以支持断点和监测点设置、逐步执行代码以及内存访问等。

- 通过跟踪端口跟踪代码执行

  跟踪信息合并为单个跟踪流并实时输出到跟踪端口分析器。

- 在循环缓冲区中连续捕获跟踪

  跟踪信息可以存储到片上循环缓冲区中，而不是输出到片外。跟踪存储可以通过调试器命令、软件命令、外部触发信号或内部事件等来启动和停止。

- 将缓冲区内容清空到跟踪端口

  存储的跟踪信息可以转存到片外的跟踪端口分析器。清空缓冲区可以通过调试器、软件、外部触发和内部事件等启动。

- 使用调试器读取缓冲区

  调试器可以通过调试端口读取跟踪缓冲区的内容。这与通过跟踪端口相比，虽然读取速度要慢一些，但是可以在不使用跟踪端口分析器的情况下实现基本跟踪功能。

- 自托管跟踪分析

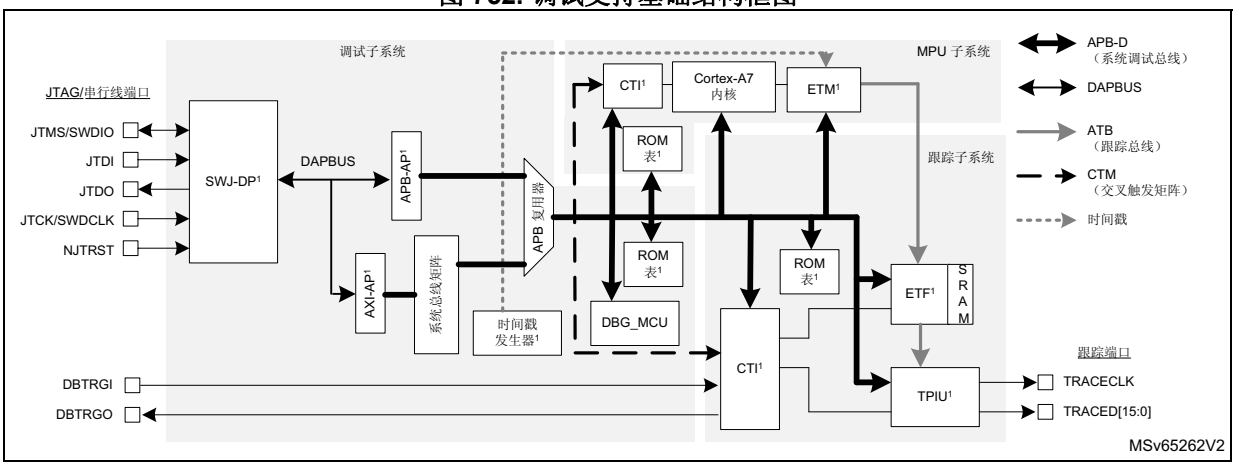
  跟踪缓冲区可通过处理器读取，或通过 DMA 传输到系统内存。借助这一强大的功能，可以通过内置的测试软件来实时监视代码执行，分析并识别故障以及自动处理异常等。

- 上传存储的跟踪信息

  存储的跟踪信息也可以使用 MPU 的各种通信接口（USB、USART、SPI、I2C、以太网以及 CAN 等）上传到主机。如果跟踪端口不可访问，例如对部署产品进行远程监视和故障分析时，这一点尤其有用。

59.3 DBG 框图

图 752. 调试支持基础结构框图



1. Arm® CoreSight™ 组件

59.4 DBG 引脚

表 589. JTAG/串行线调试端口引脚

引脚名称	JTAG 调试端口		SW 调试端口		引脚分配
	类型	说明	类型	说明	
JTMS/SWDIO	输入	JTAG 测试模式选择	I/O	串行线数据输入/输出	请参见数据手册
JTCK/SWCLK	输入	JTAG 测试时钟	输入	串行线时钟	
JTDI	输入	JTAG 测试数据输入	-	-	
JTDO	输出	JTAG 测试数据输出	-	-	
NJTRST	输入	JTAG 测试复位	-	-	

表 590. 跟踪端口引脚

引脚名称	类型	说明	引脚分配
TRACED[15:0]	输出	跟踪同步数据输出（可以是 1、2、4、8 或 16 个引脚）	请参见数据手册
TRACECLK	输出	跟踪时钟	

表 591. 触发引脚

引脚名称	类型	说明	引脚分配
DBTRGI	输入	CTI 的外部触发输入	请参见数据手册
DBTRGO	输出	CTI 的外部触发输出	

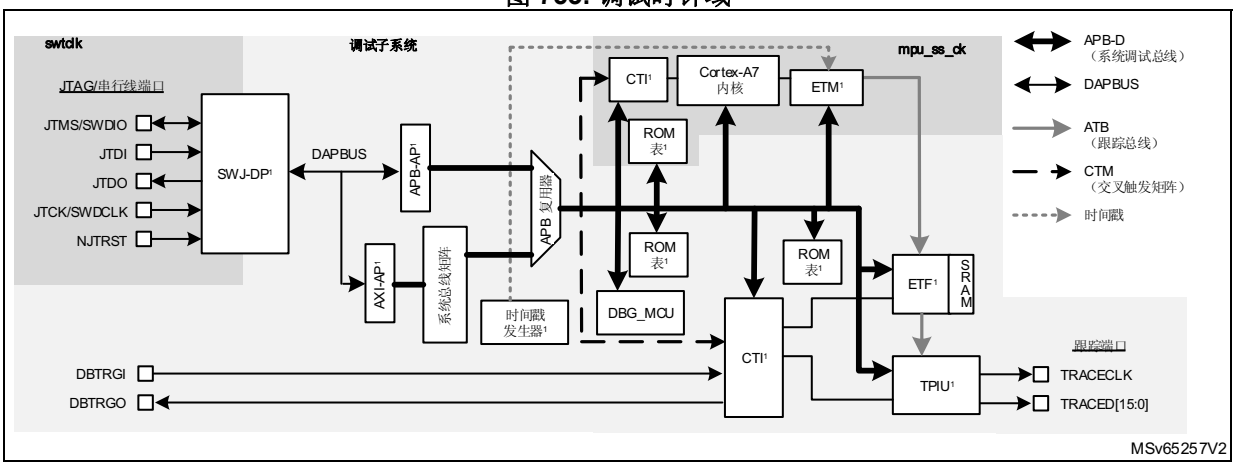
59.5 DBG 电源和时钟

59.5.1 DBG 电源域

所有调试组件（包括 SWJ-DP）均位于内核电源域中。这意味着调试器无法将设备从待机模式中唤醒。

59.5.2 DBG 时钟

图 753. 调试时钟域



1. Arm® CoreSight™ 组件

调试器通过调试接口引脚 JTCK/SWCLK 为调试端口 SWTCLK 提供时钟。该时钟用于在串行线和 JTAG 模式下寄存串行输入数据，同时操作状态机和调试端口的内部逻辑。因此，为确保调试端口返回到空闲状态，在访问结束之后，该时钟还必须持续几个周期。

SWJ-DP 包含一个异步接口，用来连接到系统调试时钟域 ck_sys_dbg。调试子系统的其余部分以及跟踪子系统均位于该域中。ck_sys_dbg 是对 AXI 总线时钟 aclk 进行门控而得到的。

该时钟既可以通过调试器使用调试端口 CTRL/STAT 寄存器中的 CDBGPWRUPREQ 位来使能，也可以通过软件写入 RCC 调试配置寄存器 (RCC_DBGCFGR) 中的 DBGCKEN 位来使能。必须先使能时钟，然后调试器或软件才能访问设备上的所有调试功能。不使用调试和跟踪功能时应禁止时钟，以免浪费能源。

APB-D（调试 APB 总线）时钟 (ck_apb_dbg) 的运行频率为 ck_sys_dbg 的一半。该总线上的一些 CoreSight™ 组件以 APB-D 频率运行，特别是时间戳发生器 (TSGEN)。

跟踪端口 (TPIU) 输出时钟 ck_trace 使用 RCC 调试配置寄存器 (RCC_DBGCFGR) 中的可编程分频比 TRACEDIV 生成。其最大频率为 ck_sys_dbg 的一半。当 TRACECLK GPIO 引脚配置为跟踪功能时，trace_ck 信号在该引脚上输出。

注： *通过将 RCC 调试配置寄存器 (RCC_DBGCFGR) 中的 TRACECKEN 置 1 来使能 TRACECLK。只要需要跟踪功能就应该激活 TRACECLK。*

处理器中包含的调试和跟踪组件（ETM、CTI 等）由内核时钟 (mpu_ss_ck) 提供时钟信号。

59.5.3 调试和低功耗模式

STM32MP13xx 设备具有节能特性，允许在不需要的时候关闭各个域时钟，甚至关闭内核电源。如果内核电源关闭（待机模式）或者系统调试域没有时钟信号（停止模式），则调试器将无法访问所有调试组件。为了避免这种情况，设计了节能模式仿真这一解决方案。如果使能了仿真功能，设备仍然会进入编程设定的低功耗模式，但其时钟和电源保持不变。换句话说，该域的行为就像处于低功耗模式，但调试器仍然可以对其进行访问。

仿真模式在 MCU 调试 (DBGMCU) 单元中编程。有关更多信息，请参见*第 59.10.6 节*。

如果设备处于停止模式（即，所有时钟关闭）且没有仿真功能，而调试器希望唤醒系统，则可以通过将调试端口 CTRL/STAT 寄存器中的 CDBGPWRUPREQ 位置为有效来实现。这将重启时钟（并根据需要接通高功率稳压器），但不会产生任何中断或事件。当系统完全唤醒后，CTRL/STAT 寄存器中的 CDBGPWRUPACK 位将变为高电平。在该状态下，调试器可以访问内存和外设寄存器，并根据需要触发唤醒 CPU。

59.5.4 DBG 复位

除了调试端口之外，调试组件将通过其各自的时钟域复位进行复位。调试端口 (SWJ-DP) 通过 nTRST 或 vcore_rst 复位进行复位。在调试器控制下，可以使用调试端口的 CTRL/STAT 寄存器中的 CDBGRSTREQ/ACK 位来复位访问端口。

调试器通过将 CDBGRSTREQ 位置 1 来请求复位，然后轮询 CDBGRSTACK 位，直到该位变为高电平（表示复位已完成）之后将 CDBGRSTREQ 位置为无效。

59.6 安全性

跟踪和调试组件允许在产品开发期间对处理器和系统进行深度访问。为了保护用户代码，并确保无法使用调试功能来改变或损害成品的正常操作，可以禁止这些功能或限制其范围。例如，可以禁止安全软件调试和跟踪，但不阻止调试非安全代码。

59.6.1 认证信号

系统使用六个认证信号来确定使能或禁止哪些功能。这些信号在 BSEC 调试配置 (BSEC_DENABLE) 中进行控制，具体包括：

- **deviceen**：控制通过外部调试端口对调试组件的访问。
  0：外部调试器无法访问调试组件。调试器对系统互连的访问不受影响。
  1：外部调试器可以访问调试组件
- **dbgen**：全局使能，用于所有调试和跟踪功能
  0：禁止所有调试功能。外部和软件仍可以访问调试组件。禁止外部访问系统互连。
  1：使能非安全模式下的调试功能。允许外部访问非安全系统互连。安全模式下的调试功能以及对安全系统互连的外部访问取决于 **spiden** 信号的状态。
- **spiden**：在 **dbgen** = 1 时使能安全特权模式下的调试功能
  0：在安全特权模式下禁止调试功能。禁止外部访问安全系统互连。
  1：在安全特权模式下使能调试功能。允许外部访问安全系统互连。
- **niden**：使能跟踪和性能监视（非侵入式调试）
  0：禁止跟踪生成。
  1：使能非安全模式下的跟踪生成。安全特权模式下的跟踪生成取决于 **spiden** 信号的状态。

- **spniden**：在 **niden** = 1 时使能安全特权模式下的跟踪和性能监视。
  0：在安全特权模式下禁止跟踪生成。
  1：在安全特权模式下使能跟踪生成。
- **dbgswenable**：使能自托管调试
  0：禁止软件访问所有调试组件
  1：允许软件访问所有调试组件

有关哪些信号可以与哪些调试组件和跟踪组件相连接的信息，请参见*表 592*。

表 592. 认证信号连接

	deviceen	dbgen	spiden	niden	spniden	dbgswenable
调试 APB-AP	X	-	-	-	-	X
系统总线 AXI-AP	-	X	X	-	-	-
Cortex®-A7 MPCore 调试单元	-	X	X	X	X	X
Cortex®-A7 MPCore ETM	-	X	-	X	-	-
Cortex®-A7 MPCore CTI	-	X	-	X	-	-

有关各个组件根据认证信号状态的行为的详细信息，请参见相关组件的章节或相关 Arm® 技术文档。

验证信号的初始状态在上电复位时设置，如*表 593* 所示。启动后，具有安全特权访问权限的软件可以在 BSEC 调试配置 (BSEC_DENABLE) 寄存器中修改验证信号的状态。

表 593. 认证信号初始状态

设备状态（通过 OTP 熔丝设置）	认证信号默认状态	说明
开发设备	全 1	启动时使能所有调试和跟踪功能。
生产设备	全 0	启动时禁止所有调试和跟踪功能。安全软件应在启动后根据需要使能调试。

59.6.2 JTAG/SWD Tamper 检测

JTAG 端口包含访问检测机制，如果将该机制配置为 Tamper，则会阻止通过调试端口进行未经授权的访问。

当尝试通过 JTAG 端口访问设备时，会被检测为 Tamper（内部 Tamper 6—— 请参见*第 47.3.2 节：TAMP 引脚和内部信号*）并通知 TAMP 单元。如果该 Tamper 源配置为通过 tamp_erase 信号擦除机密信息，则一旦检测到 Tamper，便会在 BSEC 中将所有认证位设为零，以禁止调试和跟踪功能（参见*第 3.4.3 节：BSEC Tamper 保护*）。

注： *如果 SWJ-DP 处于串行线调试 (SWD) 模式，将禁止访问检测。因此，如果在 SWJ-DP 处于 SWD 模式时（例如，如果连接了调试器）使能了 Tamper，则必须进行一次掉电再上电或 JTAG 测试复位，才能将 SWJ-DP 切换回 JTAG 模式。任何后续尝试访问 JTAG 端口或将其切换到 SWD 模式的行为都会触发 Tamper。*

59.7 芯片级 TAP 控制器 (CLTAPC)

JTAG 接口连接到 IC 的内部扫描通道。该接口用于测试和调试。扫描通道中的第一个组件是芯片级 TAP 控制器，它设计了基于 IEEE 标准 1149.1-1990 的 TAP 状态机。该控制器允许访问内置的制造测试功能。

必须绕过 CLTAPC 才能访问 SWJ-DP 提供的调试支持功能。这是通过在 SWJ-DP 指令前加上 CLTAPC 绕过指令序列 11111 来实现。

CLTAPC 还包含三个 32 位寄存器，这些寄存器可使用 CTRL 指令序列来访问：首先是 01000，接着是 1111 以绕过 SWJ-DP。接下来，寄存器地址被移入 DR（LSB 在前）。最后，根据是读取还是写入 32 位寄存器，将该寄存器的内容移入或移出 DR。

表 594 中列出了 CLTAPC 寄存器。

表 594. CLTAPC 寄存器

名称	地址	说明	读/写
SECSTS	001	BSEC_DENABLE 寄存器的状态	R
DBGCTL	010	连接到 BSEC_JTAGIN 寄存器	W
DBGSTS	011	连接到 BSEC_JTAGOUT 寄存器	R

有关 BSEC 寄存器位的详细信息，请参见*第 3 章：启动、安全性和 OTP 控制 (BSEC)*。

59.8 串行线和 JTAG 调试端口 (SWJ-DP)

SWJ-DP 是一个 CoreSight™ 组件，提供用来连接调试设备的外部访问端口。

可配置两种类型的接口：

- 一个 5 引脚标准 JTAG 接口 (JTAG-DP)
- 一个 2 引脚（时钟 + 数据）串行线调试端口 (SW-DP)

这两种模式共用相同的 IO 引脚，因此两者互斥。

默认情况下，系统复位或上电复位后选择 JTAG-DP 模式。JTDI、JTDO、JTMS/SWDIO 和 JTCK/SWCLK 这四个 I/O 引脚在调试复用功能模式下由硬件进行配置（NJTRST 是专用引脚）。在 SWJ-DP 模式下，在 JTDI、JTDO、JTMS/SWDIO 和 NJTRST 上集成了上拉电阻，在 JTCK/SWCLK 上集成了下拉电阻。

调试器可以通过在 JTMS/SWDIO 上传送以下串行数据序列来选择 SW-DP 模式：

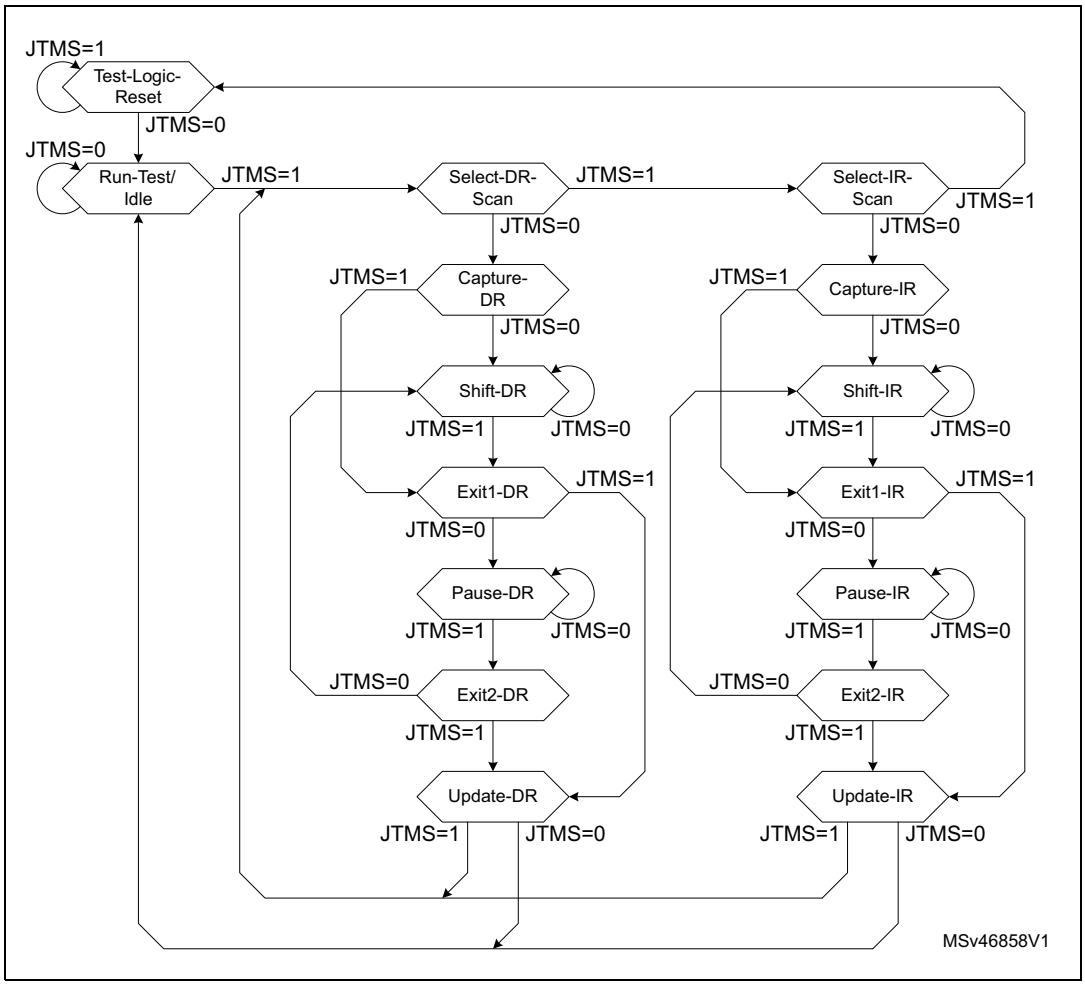
...(50 个或更多的 1)...,0,1,1,1,1,0,0,1,1,1,1,0,0,1,1,1,...(50 个或更多的 1)...

对于每一位数据，必须在 JTCK/SWCLK 上进行时钟循环。

在 SW-DP 模式下，未使用的 JTAG I/O（JTDI 和 JTDO）可用于其他功能。应注意，除了 NJTRST 之外的所有 SWJ 端口 IO 均可通过软件重新配置为其他功能，但不能再进行调试。

59.8.1 JTAG 调试端口

图 754. JTAG TAP 状态机



JTAG-DP 设计了一个基于 IEEE 标准 1149.1-1990 的 TAP 状态机 (TAPSM)。状态机如*图 754* 所示。该状态机控制两个扫描通道，一个与指令寄存器 (IR) 相关，一个与多个数据寄存器 (DR) 相关。

JTAG-DP 的工作原理如下：

当 TAPSM 通过 Capture-IR 状态时，0001 被传送到指令寄存器 (IR) 扫描通道。IR 扫描通道连接在 JTDI 和 JTDO 之间。

当 TAPSM 处于 Shift-IR 状态时，IR 扫描通道在 JTCK 的每个上升沿移位一位。这意味，在第一个节拍上：

– IR 扫描通道的 LSB 是 JTDO 上的输出。
– IR 扫描通道的位 [n] 被传送到位 [n-1]。
– JTDI 上的值会传送到 IR 扫描通道的 MSB 中。

当 TAPSM 通过 Update-IR 状态时，扫描进 IR 扫描通道中的值被传送到指令寄存器。

当 TAPSM 通过 Capture-DR 状态时，值从其中一个数据寄存器传送到其中一个 DR 扫描通道，DR 扫描通道连接在 JTDI 和 JTDO 之间。

指令寄存器中保存的值决定了数据寄存器和相关 DR 扫描通道的选择。

当 TAPSM 处于 Shift-DR 状态时，数据进行移位，与处于 IR 状态时的 IR 移位情形相同。

当 TAPSM 通过 Update-DR 状态时，扫描进 DR 扫描通道中的值被传送到所选择的数据寄存器中。

当 TAPSM 处于 Run-Test/Idle 状态时，不会发生特殊操作。IDCODE 指令加载在 IR 中。

NJTRST 信号在激活后将状态机异步复位为 Test-Logic-Reset 状态。

表 595 中列出与 4 位 IR 指令对应的数据寄存器。

表 595. JTAG-DP 数据寄存器

IR 指令	DR 寄存器	扫描通道长度	说明
0000 到 0111	(BYPASS)	1	未实现：选择 BYPASS
1000	ABORT	35	ABORT 寄存器 - 位 34:3 = APABORT：写入 0x0000 0001 可中止正在进行的 AP 访问 - 位 2:1 = A[3:2]：设为 00 - 位 0 = RnW：设为 0（只写）
1001	(BYPASS)	1	保留：选择 BYPASS
1010	DPACC	35	调试端口访问寄存器 初始化调试端口，并允许访问调试端口寄存器。 – 传输 IN 数据时： 位 34:3 = DATA[31:0] = 为写请求传输的 32 位数据 位 2:1 = A[3:2] = 调试端口寄存器的 2 位地址。 位 0 = RnW = 读请求 (1) 或写请求 (0)。 – 传输 OUT 数据时： 位 34:3 = DATA[31:0] = 读请求后读取的 32 位数据 位 2:0 = ACK[2:0] = 3 位确认： 010 = OK/FAULT 001 = WAIT 其他值 = 保留
1011	APACC	35	访问端口寄存器 初始化访问端口并允许对访问端口寄存器进行访问。 – 传输 IN 数据时： 位 34:3 = DATA[31:0] = 为写请求移入的 32 位数据 位 2:1 = A[3:2] = 访问端口寄存器的 2 位子地址。 位 0 = RnW = 读请求 (1) 或写请求 (0)。 – 传输 OUT 数据时： 位 34:3 = DATA[31:0] = 读请求后读取的 32 位数据 位 2:0 = ACK[2:0] = 3 位确认： 010 = OK/FAULT 001 = WAIT 其他值 = 保留
1100	(BYPASS)	1	保留：选择 BYPASS
1101	(BYPASS)	1	保留：选择 BYPASS

表 595. JTAG-DP 数据寄存器（续）

IR 指令	DR 寄存器	扫描通道长度	说明
1110	IDCODE	32	ID 代码 0x6BA0 0477：Arm® JTAG 调试端口 ID 代码
1111	BYPASS	1	旁路 在 JTDI 和 JTDO 之间插入单个 JTCK 周期延迟

有关 DR 寄存器的详细说明，请参见《Arm® 调试接口架构规范》*[1]*。

59.8.2 串行线调试端口

串行线调试协议使用两个引脚：

- SWCLK：从主机到目标的时钟
- SWDIO：双向串行数据（需要 100 kΩ 上拉电阻）

串行数据传输与时钟同步，首先传输 LSB。传输过程包括三个阶段：

1. 主机发送的数据包请求（8 位）。
2. 目标发送的确认响应（3 位）。
3. 主机（写操作时）或目标（读操作时）发送的传输数据（33 位）。

只有在确认响应为“OK”时才进行数据传输。

在每一个传输阶段之间，如果数据传输方向发生改变，则插入一个单个时钟周期的周转时间。

表 596. 数据包请求

位字段	名称	说明
0	开始	必须为 1
1	APnDP	0：DP 寄存器访问——有关 DP 寄存器的列表，请参见*表 599* 1：AP 寄存器访问——请参见*第 59.9 节*
2	RnW	0：写请求 1：读请求
4:3	A(3:2)	DP 或 AP 寄存器的地址位字段（请参见*表 599*）
5	奇偶校验	前面几位的单位奇偶校验
6	Stop	0
7	驻留	不接受主机指令控制。必须由目标读为 1。

表 597. ACK 响应

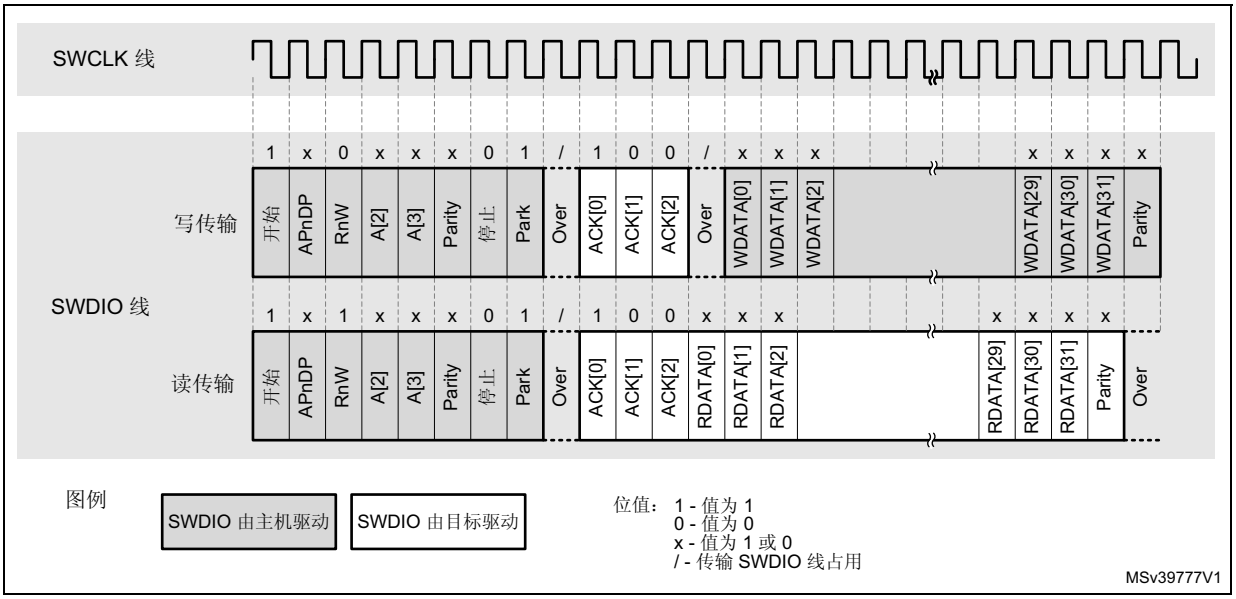
位字段	名称	说明
2:0	ACK	000：FAULT 010：WAIT 100：OK

表 598. 数据传输

位字段	名称	说明
31:0	WDATA 或 RDATA	写入或读取数据
32	奇偶校验	32 个数据位的单位奇偶校验

图 755 显示了成功的写操作和读操作传输过程示例。

图 755. SWD 成功的数据传输过程



如果目标响应为 FAULT 或 WAIT ACK，则数据传输过程将取消，除非使能了上溢检测，在这种情况下，数据会被目标忽略（写操作时）或不接受指令控制（读操作时）。

当引线第一次接通时，必须由主机生成引线复位信号，否则会出现协议错误。引线复位信号由 50 或以上个 SWDIO 为高电平的 SWCLK 周期和紧随的两个 SWDIO 为低电平的 SWCLK 周期组成。

有关串行线调试协议的更多详细信息，请参见《Arm® 调试接口架构规范》*[1]*。

注： *SWJ-DP 适用 SWD 协议版本 2*

59.8.3 调试端口寄存器

SW-DP 和 JTAG-DP 都可以访问调试端口 (DP) 寄存器。*表 599* 中列出了这些寄存器。

调试器可访问 DP 寄存器，如下所示：

1. 对 DP_SELECT 数据端口寄存器中的 DPBANKSEL 位字段编程，选择要访问的寄存器组（请参见*表 599*）。
2. 如果使用 JTAG，则使用组中的寄存器地址对 DPACC 寄存器中的 A(3:2) 位字段编程。
3. 对 RnW 位编程，选择读操作或写操作。
4. 如果选择写操作，则使用写入数据对 DATA 位字段进行编程。

如果使用 SWD，则 A(3:2) 和 RnW 位字段包含在发送到 SW-DP 的数据包请求字中，数据包请求字的 APnDP 位复位（请参见*表 596*）。写入数据在数据阶段发送。

表 599. 调试端口寄存器

地址	A(3:2) 值	R/W	说明
0x0	00	R	DP_DPIDR 寄存器。包含调试端口的 IDCODE
		W	DP_ABORT 寄存器(1)。中止当前的 AP 事务。该寄存器还用于清除 DP_CTRLSTAT 寄存器中的错误标志。
0x4	01	R/W	如果 DP_SELECT 寄存器的 DPBANKSEL 位字段为 0：DP_CTRLSTAT 寄存器。 控制 DP 并提供状态信息。
			如果 DP_SELECT 寄存器的 DPBANKSEL 位字段为 1：DP_DLCR 寄存器(2)。 控制 SWD 数据链路的工作模式。
			如果 DP_SELECT 寄存器的 DPBANKSEL 位字段为 2：DP_TARGETID 寄存器。 提供目标标识信息。
			如果 DP_SELECT 寄存器的 DPBANKSEL 位字段为 3：DP_DLPIDR 寄存器(3)。 提供 SWD 协议版本。
0x8	10	R	DP_RESEND 寄存器(4)。返回由最后一个 AP 读操作或 DP RDBUFF 读操作返回的值。在读传输损坏的情况下使用。
		W	DP_SELECT 寄存器。选择访问端口、访问端口寄存器组和地址为 0x4 的 DP 寄存器。
0xC	11	R	DP_RDBUFF 寄存器： 通过 JTAG-DP——用于允许调试器在执行一系列操作后获取最后结果（无需请求新的 JTAG-DP 操作）。 通过 SW-DP——包含以前的 AP 读访问的结果，从而避免了重新进行 AP 访问。
		W	DP_TARGETSEL 寄存器(5)。在引线复位序列后立即对 DP_TARGETSEL 执行写入操作时，如果同时满足以下两个条件，则目标就会被选择： 位 [31:28] 与 DP_DLPIDR 中的位 [31:28] 匹配。 位 [27:0] 与 DP_TARGETID 寄存器中的位 [27:0] 匹配。 写入其他任意值取消选择目标。调试工具必须写入 0xFFFF FFFF 以取消选择所有目标。这是一个无效的 TARGETID 值。所有其他无效的 TARGETID 值都被保留。

1. 使用 ABORT 指令实现从 JTAG-DP 访问 AP DP_ABORT 寄存器。
2. 只能通过 SW-DP 访问。寄存器通过 JTAG-DP“保留”。
3. 只能通过 SW-DP 访问。寄存器通过 JTAG-DP“保留”。
4. 只能通过 SW-DP 访问。寄存器通过 JTAG-DP“保留”。
5. 只能通过 SW-DP 访问。寄存器通过 JTAG-DP“保留”。

DP 调试端口标识寄存器 (DP_DPIDR)

DP debug port identification register

偏移地址：0x0

复位值：0x6BA0 2477

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REVISION[3:0]				PARTNO[7:0]								Res.	Res.	Res.	MIN
r	r	r	r	r	r	r	r	r	r	r	r				r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VERSION[3:0]				DESIGNER[10:0]											Res.
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	

位 31:28 **REVISION[3:0]**：版本代码 (revision code)
0x6

位 27:20 **PARTNO[7:0]**：调试端口产品编号 (debug port part number)
0xBA

位 19:17 保留，必须保持复位值。

位 16 **MIN**：最少调试端口设计 (minimal debug port implementation)
0x0：未设计最少调试端口 (MINDP)（支持事务计数器和推入操作）

位 15:12 **VERSION[3:0]**：DP 架构版本 (DP architecture version)
0x2：DPv2

位 11:1 **DESIGNER[10:0]**：JEDEC 设计人员标识代码 (JEDEC designer identity code)
0x23B：Arm®

位 0 保留，必须保持复位值。

DP 中止寄存器 (DP_ABORT)

DP abort register

偏移地址：0x0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ORUNERRCLR	WDERRCLR	STKERRCLR	STKCMPCLR	DAPABORT
											w	w	w	w	w

位 31:5 保留，必须保持复位值。

位 4 **ORUNERRCLR**：上溢错误清零 (overrun error clear)
0：不起作用
1：清零 DP_CTRLSTAT 寄存器的 STICKYORUN 位

位 3 **WDERRCLR**：写入数据错误清零 (write data error clear)
0：不起作用
1：清零 DP_CTRLSTAT 寄存器的 WDATAERR 位

位 2 **STKERRCLR**：粘滞错误清零 (sticky error clear)
0：不起作用
1：清零 DP_CTRLSTAT 寄存器的 STICKYERR 位

位 1 **STKCMPCLR**：粘滞比较清零 (sticky compare clear)
0：不起作用
1：清零 DP_CTRLSTAT 寄存器的 STICKYCMP 位

位 0 **DAPABORT**：当前的 AP 事务中止 (current AP transaction abort)
如果返回的 WAIT 响应数量过多（表示事务已停止），则当前的 AP 事务中止。
0：不起作用
1：中止事务

调试端口控制/状态寄存器 (DP_CTRLSTAT)

Debug port control/status register

偏移地址：0x4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	CDBGPWRUPACK	CDBGPWRUPREQ	CDBGRSTACK	CDBGRSTREQ	Res.	Res.	TRNCNT[11:4]							
		r	rw	r	rw			rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRNCNT[3:0]				MASKLANE[3:0]				WDATAERR	READOK	STICKYERR	STICKYCMP	TRNMODE[1:0]		STICKYORUN	ORUNDETECT
rw	rw	rw	rw	rw	rw	rw	rw	r	r	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29 **CDBGPWRUPACK**：调试上电确认 (debug power up acknowledge)
从电源控制器返回调试域上电确认信号的状态。
0：域掉电
1：域上电

位 28 **CDBGPWRUPREQ**：调试上电请求 (debug power up request)
控制到电源控制器的调试域上电/掉电请求信号。
0：请求掉电
1：请求上电

位 27 **CDBGRSTACK**：调试复位确认 (debug reset acknowledge)
来自复位控制器的调试域复位确认信号的状态。
0：域复位未激活/挂起
1：域复位完成

位 26 **CDBGRSTREQ**：调试复位请求 (debug reset request)
控制到复位控制器的调试域复位请求信号。
0：将复位请求置为无效/请求未激活
1：将复位请求置为有效/请求激活

位 25:24 保留，必须保持复位值。

位 23:12 **TRNCNT[11:0]**：事务计数器 (transaction counter)
若要通过 AP 将事务序列编程为增量地址，TRNCNT 需载有要执行的事务数。每个事务成功完成时，事务数会递减。

位 11:8 **MASKLANE[3:0]**：通道比较屏蔽 (lane comparison mask)
指示推入比较和推入验证操作（DP_CTRLSTAT 寄存器的 TRNMODE 位为 1 或 2 时）中要屏蔽的字节。在推入操作中，在 AP 写入事务中提供的字与目标 AP 地址中的当前值进行比较。
1XXX：比较中包含字节通道 3
X1XX：比较中包含字节通道 2
XX1X：比较中包含字节通道 1
XXX1：比较中包含字节通道 0

位 7 **WDATAERR**：SW-DP 模式下写入数据错误 (write data error in SW-DP)
指示以下任一情况：
– 写入操作的数据阶段存在奇偶校验错误或成帧错误
– 已被 DP 接受的写入操作被丢弃，而不是被提交给 AP
通过向 DP_ABORT 寄存器的 WDERRCLR 位写入 1 可将该位复位。
0：无错误
1：发生错误
注意：在 JTAG-DP 模式下，该位保留。

位 6 **READOK**：SW-DP 模式下 AP 读操作响应 (AP read response in SW-DP)
表示对最后一个 AP 读访问的响应。
0：读操作不正常
1：读操作正常
注意：在 JTAG-DP 模式下，该位保留。

位 5 **STICKYERR**：事务错误 (transaction error)(1)
指示在 AP 事务期间发生了错误。
0：无错误
1：发生错误
在 SW-DP 模式下，通过向 DP_ABORT 寄存器的 STKERRCLR 位写入 1 可将该位复位。在 JTAG-DP 模式下，通过向该位写入 1 可将该位复位。

位 4 **STICKYCMP**：比较匹配 (compare match)(1)
指示在推入操作中发生了匹配。
0：匹配（当 TRNMODE = 0x1 时）；不匹配（当 TRNMODE = 0x2 时）
1：不匹配（当 TRNMODE = 0x1 时）；匹配（当 TRNMODE = 0x2 时）
在 SW-DP 模式下，通过向 DP_ABORT 寄存器的 STKCMPCLR 位写入 1 可将该位复位。在 JTAG-DP 模式下，通过向该位写入 1 可将该位复位。

位 3:2 **TRNMODE[1:0]**：AP 写操作传输模式 (transfer mode for AP write operations)
对于读操作，该位字段必须设置为 0。
0：正常工作。AP 事务直接传递到 AP。
1：推入验证操作。DP 存储写入数据并执行目标 AP 地址的读取事务。将读取的结果与存储的数据进行比较，如果二者不匹配，则 STICKYCMP 位置 1。
2：推入比较操作。DP 存储写入数据并执行目标 AP 地址的读取事务。将读取的结果与存储的数据进行比较，如果二者匹配，则 STICKYCMP 位置 1。
3：保留
在推入操作中，只将 MASKLANE 位字段表示的数据字节进行比较。

位 1 **STICKYORUN**：上溢 (overrun)(1)
指示发生了上溢（在完成上一个事务之前接收到新的事务）。该位仅在 ORUNDETECT 位置 1 时才置 1。
0：无上溢
1：发生上溢
在 SW-DP 模式下，通过向 DP_ABORT 寄存器的 ORUNERRCLR 位写入 1 可将该位复位。
在 JTAG-DP 模式下，通过向该位写入 1 可将该位复位。

位 0 **ORUNDETECT**：上溢检测模式使能 (overrun detection mode enable)
0：禁止上溢检测。
1：使能上溢检测。如果发生上溢，则 STICKYORUN 位将置 1，同时后续事务将被阻止，直到 STICKYORUN 位清零。

1. SW-DP 模式下的访问类型为“r”，JTAG-DP 模式下的访问类型为“rw”。

DP 数据链路控制寄存器 (DP_DLCR)

DP data link control register

偏移地址：0x4

复位值：0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TURNROUND[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
						rw	rw								

位 31:10 保留，必须保持复位值。

位 9:8 **TURNROUND[1:0]**：SWDIO 的三态周期 (tristate period for SWDIO)
0x0：1 个数据位周期
0x1：2 个数据位周期
0x2：3 个数据位周期
0x3：4 个数据位周期

位 7 保留，必须保持复位值。

位 6 保留，必须保持复位值。

位 5:0 保留，必须保持复位值。

DP 目标标识寄存器 (DP_TARGETID)

DP target identification register

偏移地址：0x4

复位值：0x1501 0041

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TREVISION[3:0]				TPARTNO[15:4]											
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPARTNO[3:0]				TDESIGNER[10:0]											Res.
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	

位 31:28 **TREVISION[3:0]**：目标版本 (target revision)
0x1：版本 A

位 27:12 **TPARTNO[15:0]**：目标产品编号 (target part number)
0x5010：STM32MP13x

位 11:1 **TDESIGNER[10:0]**：目标设计人员 JEDEC 代码 (target designer JEDEC code)
0x020：意法半导体

位 0 保留，必须保持复位值。

DP 数据链路协议标识寄存器 (DP_DLPIDR)

DP data link protocol identification register

偏移地址：0x4

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TINSTANCE[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PROTSVN[3:0]			
												r	r	r	r

位 31:28 **TINSTANCE[3:0]**：目标实例编号 (target instance number)
用于在多点系统中定义此设备的实例编号。
0x0：实例 0

位 27:4 保留，必须保持复位值。

位 3:0 **PROTSVN[3:0]**：串行线调试协议版本 (Serial Wire Debug protocol version)
0x1：版本 2

DP 重新发送寄存器 (DP_RESEND)

DP resend register

偏移地址：0x8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESEND[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESEND[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RESEND[31:0]**：最后一个 AP 读操作或 DP_RDBUFF 读操作值 (last AP read or DP_RDBUFF read value)

返回由最后一个 AP 读操作或 DP_RDBUFF 读操作返回的值。在读传输损坏的情况下使用。

DP 访问端口选择寄存器 (DP_SELECT)

DP access port select register

偏移地址：0x8

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
APSEL[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
w	w	w	w	w	w	w	w								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APBANKSEL[3:0]				DPBANKSEL[3:0]			
								w	w	w	w	w	w	w	w

位 31:24 **APSEL[7:0]**：访问端口选择 (access port select)

选择下一个事务的访问端口。

0x0：AP0——系统总线访问端口 (AXI-AP)

0x1：AP1——调试总线访问端口 (APB-AP)

位 23:8 保留，必须保持复位值。

位 7:4 **APBANKSEL[3:0]**：AP 寄存器组选择 (AP register bank select)

选择活动 AP 上的 4 字寄存器组，以用于下一个事务。

位 3:0 **DPBANKSEL[3:0]**：DP 寄存器组选择 (DP register bank select)

选择调试端口地址为 0x4 的寄存器。

0x0：DP_CTRLSTAT 寄存器

0x1：DP_DLCR 寄存器

0x2：DP_TARGETID 寄存器

0x3：DP_DLPIDR 寄存器

其他值：保留

DP 读缓冲区寄存器 (DP_RDBUFF)

DP read buffer register

偏移地址：0xC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDBUFF[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDBUFF[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RDBUFF[31:0]**：最后一个 AP 读访问值 (last AP read access value)

包含由最后一个 AP 读访问返回的值。AP 读访问返回的值可通过对同一地址进行第二次读访问来获得，这将在相应的总线上启动新的事务，或者还可以从该寄存器读取，在这种情况下不会发生新的 AP 事务。

DP 目标标识寄存器 (DP_TARGETSEL)

DP target identification register

偏移地址：0xC

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TINSTANCE[3:0]				TPARTNO[15:4]											
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPARTNO[3:0]				TDESIGNER[10:0]											Res.
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	

位 31:28 **TINSTANCE[3:0]**：目标实例编号 (target instance number)

用于在多点系统中定义目标设备的实例编号。必须将 DP_DLPIDR 寄存器中的 TINSTANCE 位字段的值写入这些位才能选择此设备。

位 27:12 **TPARTNO[15:0]**：目标产品编号 (target part number)

定义目标设备的产品编号。必须将 DP_TARGETID 寄存器中的 TPARTNO 位字段的值写入这些位才能选择此设备。

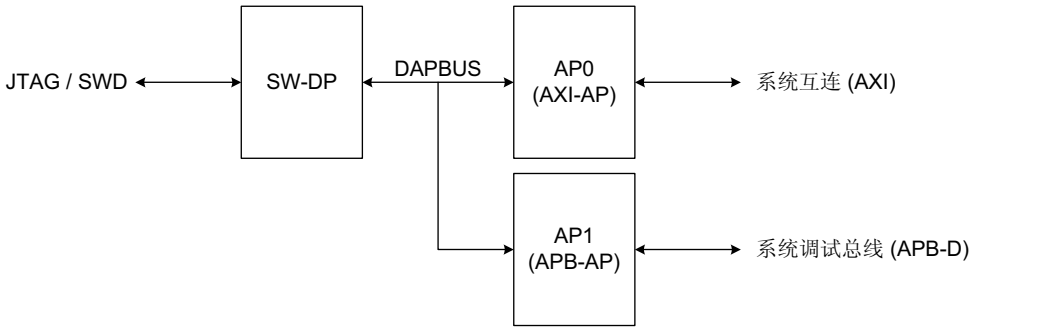
位 11:1 **TDESIGNER[10:0]**：目标设计人员 JEDEC 代码 (target designer JEDEC code)

定义目标设备的 JEDEC 代码。必须将 DP_TARGETID 寄存器中的 TDESIGNER 位字段的值写入这些位才能选择此设备。

位 0 保留，必须保持复位值。

59.9 访问端口

图 756. 调试和访问端口连接



DP 上附有两个访问端口 (AP)：

- AP0：系统总线访问端口 (AXI-AP)。允许访问系统总线矩阵。因此，能够看到可通过 AXI 和 AHB 互连访问的所有内存和外设，但系统 APB 调试总线上的内存和外设除外。无法通过该端口访问 CoreSight™ 组件。
- AP1：调试访问端口 (APB-AP)。允许访问调试 APB 总线上的 CoreSight™ 组件，包括 Cortex®-A7。DBGMCU 也位于该总线上。

两个访问端口都是 MEM-AP 类型，即调试和跟踪组件寄存器都映射在相关调试总线的地址空间中。调试器将 AP 视为一组 32 位寄存器，每四个寄存器组成一个 bank。其中一些寄存器用于配置或监视 AP 本身，而其他寄存器则用于执行总线传输。*表 602* 列出了 AP 寄存器。

AP 寄存器的地址包括：

- 位 [7:4]：DP 中 DP_SELECT 寄存器 APBANKSEL 位字段的内容（请参见 *DP 访问端口选择寄存器 (DP_SELECT)*）
- 位 [3:2]：JTAG-DP 模式下 APACC 数据寄存器（请参见*表 595*）或 SW-DP 数据包请求（请参见*表 596*）中的 A(3:2) 位字段的内容，具体取决于使用的调试接口
- 位 [1:0]：始终设为 0

DP 中 DP_SELECT 寄存器 APSEL 位字段的内容定义访问的 MEM-AP。

调试器可访问 AP 寄存器，如下所示：

1. 对 DP 中的 DP_SELECT 寄存器 APSEL 位字段编程，以选择其中一个 AP；对 APBANKSEL 位字段编程，以选择要访问的寄存器组（请参见 *DP 访问端口选择寄存器 (DP_SELECT)*）。
2. 如果使用 JTAG，则使用组中的寄存器地址对 APACC 寄存器中的 A(3:2) 位字段编程。对 RnW 位编程，选择读操作或写操作。如果选择写操作，则使用写入数据对 DATA 位字段进行编程。如果使用 SWD，则 A(3:2) 和 RnW 位字段包含在发送到 SW-DP 的数据包请求字当中，数据包请求字的 APnDP 位置 1（请参见*表 596*）。写入数据在数据阶段发送。

调试器可通过 MEM-AP 寄存器访问内存映射的调试组件寄存器（即，使用上述 AP 寄存器访问过程），如下所示：

1. 对 TAR 寄存器中的事务目标地址编程。
2. 对 CSW 寄存器编程，必要时，请使用传输参数（如 AddrInc）。

3. 针对 TAR 寄存器中保存的地址，写入或读取 DRW 寄存器，以启动总线事务。或者，读取或写入分组数据寄存器 BDn，以触发访问地址 TAR[31:4] + n（无需更改 TAR 寄存器中的地址即可访问多达四个连续的地址）。

有关 MEM-AP 的更多详细信息，请参见《Arm® 调试接口架构规范》*[1]*。

59.9.1 认证

表 600 至*表 601* 列出了每个访问端口的认证信号的作用。

表 600. AP0（系统互连）认证行为

dbgen	spiden	其他	行为
0	X	X	阻止所有传输。任何尝试执行传输的行为都将引发错误。
1	0		阻止安全传输。任何尝试执行安全传输的行为都将引发错误。使能非安全传输。
1	1		使能非安全传输和安全传输。

表 601. AP1（调试 APB）认证行为

deviceen	其他	行为
0	X	阻止所有传输。
1		使能传输。注：访问某些组件可能会返回错误，具体取决于其他身份认证信号的状态。

59.9.2 MEM-AP 寄存器

表 602. MEM-AP 寄存器

地址	APBANKSEL	A(3:2)	名称	说明
0x00	0x0	0	CSW	控制/状态字寄存器——请参见以下部分 *AP0 控制/状态字寄存器 (AP0_CSW)* *AP1 控制/状态字寄存器 (AP1_CSW)*
0x04	0x0	1	TAR	传输地址寄存器——请参见以下部分 *AP 传输地址寄存器 (AP_TAR)*
0x08	-	-	-	保留
0x0C	0x0	3	DRW	数据读/写寄存器——请参见以下部分 *AP 数据读/写寄存器 (AP_DRW)*
0x10	0x1	0	BD0	分组数据 0 寄存器——请参见以下部分 *AP 分组数据寄存器 0 (AP_BD0)*
0x14	0x1	1	BD1	分组数据 1 寄存器——请参见以下部分 *AP 分组数据寄存器 1 (AP_BD1)*
0x18	0x1	2	BD2	分组数据 2 寄存器——请参见以下部分 *AP 分组数据寄存器 2 (AP_BD2)*
0x1C	0x1	3	BD3	分组数据 3 寄存器——请参见以下部分 *AP 分组数据寄存器 3 (AP_BD3)*

表 602. MEM-AP 寄存器（续）

地址	APBANKSEL	A(3:2)	名称	说明
0x20	0x2	0	ACE	ACE 屏障事务寄存器（仅限 AXI-AP）——请参见以下部分 *AP ACE 屏障事务寄存器 (AP_ACE)*
0x24-0xEC	-	-	-	保留
0xF0	-	-	-	保留
0xF4	0xF	1	CFG	配置寄存器（仅限 AXI-AP）——请参见以下部分 *AP 配置寄存器 (AP_CFG)*
0xF8	0xF	2	BASE	调试基址寄存器 (RO)——请参见以下部分 *AP x 基址寄存器 (APx_BASE)（x = 0 到 1）*
0xFC	0xF	3	IDR	标识寄存器 (RO)——请参见以下部分 *AP x 标识寄存器 (APx_IDR)（x = 0 到 1）*

AP0 控制/状态字寄存器 (AP0_CSW)

AP0 control/status word register

偏移地址：0x0

复位值：0x3080 6042

有关每个位字段的详细说明，请参见相应总线的 *Arm AMBA 协议规范 [7.,8.,9.]*。

AP0 (AXI-AP)：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	PROT[2:0]			CACHE[3:0]				SPISTATUS	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw	rw	rw	rw	rw	rw	rw	r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DOMAIN[1:0]		ACEENABLE	MODE[3:0]				TRINPROG	DBGSTATUS	ADDRINC[1:0]		Res.	SIZE[2:0]		
	rw	rw	rw	rw	rw	rw	rw	r	r	rw	rw		rw	rw	rw

位 31 保留，必须保持复位值。

位 30:28 **PROT[2:0]**：总线事务保护属性 (bus transaction protection attributes)

该位字段设置总线事务的 AxPROT[2:0] 保护属性。

XX0：非特权

XX1：特权

X0X：安全

X1X：非安全

0XX：数据

1XX：指令

位 27:24 **CACHE[3:0]**：总线事务缓存属性 (bus transaction cache attributes)
该位字段设置总线事务的 AxCACHE[3:0] 缓存属性。
XXX0：不可缓冲
XXX1：可缓冲
XX0X：不可修改（不可缓存）
XX1X：可修改（可缓存）
X0XX：不读分配
X1XX：读分配
0XXX：不写分配
1XXX：写分配

位 23 **SPISTATUS**：**spiden** 认证信号的状态(status of **spiden** authentication signal)（只读）
该信号确定了调试器是否可访问安全内存。
0：阻止安全 AXI 传输
1：允许安全 AXI 传输

位 22:15 保留，必须保持复位值。

位 14:13 **DOMAIN[1:0]**：可共享事务属性 (shareable transaction attributes)
该位字段定义 ACE 事务的可共享属性。
0x0：不可共享
0x1：可共享，内部域，包括额外的主设备
0x2：可共享，外部域，也包括内部或额外的主设备
0x3：可共享，系统域，包括所有主设备

位 12 **ACEENABLE**：使能 ACE 事务，包括屏障 (ACE transactions enable, including barriers)
0：禁止
1：使能

位 11:8 **MODE[3:0]**：工作模式 (mode of operation)
0x0：正常下载或上传
0x1：屏障事务
0x2-0xF：保留

位 7 **TRINPROG**：传输正在进行 (transfer in progress)（只读）
表示正在 AP 上进行总线传输。
0：无正在进行的总线传输
1：正在进行总线传输

位 6 **DBGSTATUS**：DBGEN 认证信号状态 (status of DBGEN authentication signal)（只读）
该信号决定是否可以访问 AXI 总线。
0：阻止 AXI 事务
1：允许 AXI 事务

位 5:4 **ADDRINC[1:0]**：RW 数据访问时的自动递增和打包模式 (auto-increment and packing mode on RW data access)
仅在当前事务在未发送错误响应的情况下完成并且事务未被中止时递增。
访问分组数据寄存器 0x10-0x1C 时，不执行自动地址递增和打包数据传输。在这些情况下，忽略这些位的值。
如果使能递增，则地址在 1 KB 地址边界处回卷。
0x0：不自动递增
0x1：地址按事务字节（SIZE 位字段）的大小递增
0x2：使能打包传输。32 位 AP 访问产生与编程的事务大小相对应的 1 个 32 位、2 个 16 位或 4 个 8 位总线事务。数据相应地被打包或解包。
0x3：保留

位 3 保留，必须保持复位值。

位 2:0 **SIZE[2:0]**：下一个内存访问事务的大小 (size of next memory access transaction)
0x0：8 位
0x1：16 位
0x2：32 位
0x3：64 位
其他值：保留

AP1 控制/状态字寄存器 (AP1_CSW)

AP1 control/status word register

偏移地址：0x0

复位值：0x8000 0042

有关每个位字段的详细说明，请参见相应总线的 *Arm AMBA 协议规范 [7.,8.,9.]*。

AP1 (APB-AP)：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DBGSWENABLE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	MODE[3:0]				TRINPROG	DEVICEEN	ADDRINC[1:0]		Res.	SIZE[2:0]		
				rw	rw	rw	rw	r	r	rw	rw		r	r	r

位 31 **DBGSWENABLE**：软件访问使能 (software access enable)
定义软件是否可以访问系统调试 APB。该位与来自 BSEC 的 **dbgswenable** 信号进行逻辑与运算。
0：禁止软件访问
1：如果来自 BSEC 的 **dbgswenable** 信号也为高电平，则使能软件访问

位 30:12 保留，必须保持复位值。

位 11:8 **MODE[3:0]**：使能屏障支持 (barrier support enabled)
定义是否支持内存屏障操作。
0x0：不支持

位 7 **TRINPROG**：传输正在进行 (transfer in progress)（只读）
表示正在 AP 上进行总线传输。
0x0：无正在进行的传输
0x1：正在进行总线传输

位 6 **DEVICEEN**：使能设备 (device enabled)（只读）
定义调试器是否可以访问系统调试 APB。该位反映来自 BSEC 的 **deviceen** 信号的状态。
0x0：禁止 APB 访问。任何尝试在 AP 上发起事务的行为都将返回错误。
0x1：使能 APB 访问

位 5:4 **ADDRINC[1:0]**：自动递增模式 (auto-increment mode)
定义了在事务结束之后 TAR 地址是否会自动递增。
0x0：不自动递增
0x1：地址按事务字节（SIZE 位字段）的大小递增
其他值：保留

位 3 保留，必须保持复位值。

位 2:0 **SIZE[2:0]**：下一个内存事务访问的大小 (size of next memory transaction access)（只读）
0x2：字（32 位）

AP 传输地址寄存器 (AP_TAR)

AP transfer address register

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRESS[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **ADDRESS[31:0]**：当前传输的地址 (address of the current transfer)（位 31:0）

AP 数据读/写寄存器 (AP_DRW)

AP data read/write register

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DATA[31:0]**：当前传输的读/写数据 (read/write data for current transfer)

在写模式下，这是写数据值。在读模式下，这是读数据值。
对于 64 位访问（仅限 AXI-AP），必须对 DRW 发起多次访问才能实现单次 AXI 访问。

读：序列中第一次读取 DRW 会启动内存访问。第一次读访问返回数据的低 32 位。随后的读访问返回数据的高 32 位。如果在序列完成前启动对 CSW 或 TAR 的写操作，则将终止读访问，并且读数据不再可用。

写：第一次写入 DRW 指定要写入的低 32 位数据。随后的写访问指定要写入的高 32 位数据。如果在序列完成前启动对 CSW 的写操作，则不会在 AXI 接口上启动写访问。

序列中的部分读写访问组合会终止先前的访问。另外，也不会识别最新的访问。
序列中对 CSW 寄存器、TAR 或 AP 中任何其他寄存器的任何写访问都会终止正在进行的访问。另外，也不会识别当前的访问。
如果终止写序列，则不会在 AXI 接口上进行写操作。

AP 分组数据寄存器 0 (AP_BD0)

AP banked data register

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DATA[31:0]**：当前传输的读/写数据 (read/write data for current transfer)

事务地址为 TAR[31:4] << 4 + 0x0。

AP 分组数据寄存器 1 (AP_BD1)

AP banked data register 1

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DATA[31:0]**：当前传输的读/写数据 (read/write data for current transfer)

事务地址为 TAR[31:4] << 4 + 0x4。

AP 分组数据寄存器 2 (AP_BD2)

AP banked data register 2

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DATA[31:0]**：当前传输的读/写数据 (read/write data for current transfer)
事务地址为 TAR[31:4] << 4 + 0x8。

AP 分组数据寄存器 3 (AP_BD3)

AP banked data register 3

偏移地址：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DATA[31:0]**：当前传输的读/写数据 (read/write data for current transfer)
事务地址为 TAR[31:4] << 4 + 0xC。

AP ACE 屏障事务寄存器 (AP_ACE)

AP ACE barrier transaction register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BARTRAN[1:0]		TRGBARTRAN
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:1 **BARTRAN[1:0]**：屏障事务 (barrier transactions)

0x0：正常访问屏障

0x1：内存屏障

0x2：保留

0x3：同步屏障

位 0 **TRGBARTRAN**：使能屏障事务 (enable barrier transaction)

0：禁止

1：使能

AP 配置寄存器 (AP_CFG)

AP configuration register

偏移地址：0xF4

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LD	Res.	LA
													r		r

位 31:3 保留，必须保持复位值。

位 2 **LD**：大数据 (large data)

表示支持大于 32 位的数据项（仅限 AXI-AP）。

1：支持 64 位数据

位 1 保留，必须保持复位值。

位 0 **LA**：长地址 (long address)

表示支持大于 32 位的寻址。

0：32 位或更少位的地址

AP x 基址寄存器 (APx_BASE)（x = 0 到 1）

AP x base address register

偏移地址：0xF8

复位值：块 0：0x0000 0002

复位值：块 1：0xE008 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BASEADDR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASEADDR[15:12]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FORMAT	ENTRYPRESENT
r	r	r	r											r	r

位 31:12 **BASEADDR[31:12]**：基址 (base address)

对应于 AP 的 ROM 表地址的位 31 到 12。由于 ROM 表必须与 4 KB 边界对齐，因此 12 个 LSB 都为零。

AP0（AXI 系统总线）：N/A（无 ROM 表）

AP1（系统调试 APB）：0xE0080

位 11:2 保留，必须保持复位值。

位 1 **FORMAT**：基址寄存器格式 (base address register format)

1：Arm® 调试接口 v5

位 0 **ENTRYPRESENT**：调试组件存在 (debug component presence)

指示访问端口总线上有无调试组件。

0：无调试组件

1：有调试组件

AP x 标识寄存器 (APx_IDR)（x = 0 到 1）

AP x identification register

偏移地址：0xFC

复位值：块 0：0x3477 0004

复位值：块 1：0x4477 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REVISION[3:0]				JEDECBANK[3:0]				JEDECCODE[6:0]							MEMAP
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDENTITY[7:0]							
								r	r	r	r	r	r	r	r

位 31:28 **REVISION[3:0]**：版本代码 (revision code)

0x3：r1p0

0x4：r0p5

0x6：r0p7

位 27:24 **JEDECBANK[3:0]**：JEDEC bank

0x4：Arm®

位 23:17 **JEDECCODE[6:0]**：JEDEC 代码 (JEDEC code)
0x3B：Arm®

位 16 **MEMAP**：内存访问端口 (memory access port)
0x1：标准寄存器映射

位 15:8 保留，必须保持复位值。

位 7:0 **IDENTITY[7:0]**：AP 类型 (AP type)
0x04：AXI-AP
0x03：APB-AP

59.10 系统调试功能

系统调试功能包括以下 CoreSight™ 组件：

- 系统 ROM 表
- 全局跟踪时间戳发生器 (TSGEN)
- 系统交叉触发接口 (CTI)
- 交叉触发矩阵 (CTM)
- 跟踪端口接口单元 (TPIU)
- 嵌入式跟踪 FIFO (ETF)

调试器可通过系统 APB-AP 及其相关的 APB 调试总线访问这些组件。Cortex®-A7 处理器也可访问这些组件。

MCU 调试单元 (DBG_MCU) 也可通过 APB-D 访问，该单元不是 CoreSight™ 组件，其包含的寄存器用于在调试模式下配置设备的行为。

59.10.1 系统 ROM 表

调试 APB 总线上有三个 ROM 表。ROM 表是一款 CoreSight™ 组件，包含 APB 总线上的 CoreSight™ 调试组件的基址。这些表允许调试器自动发现 CoreSight™ 系统的拓扑。

第一个表指向位于调试子系统中的 CoreSight™ 组件：DBGMCU、TSGEN，以及跟踪子系统 ROM 表和 Cortex-A7 子系统 ROM 表。该表占用一个 4 KB、32 位宽的 APB 地址空间块——从 0xE008 0000 到 0xE008 0FFC（调试器访问时）或者从 0x5008 0000 到 0x5008 0FFC（从系统总线访问时）。

表 603. 调试子系统 ROM 表

ROM 表中的偏移地址	组件名称	组件基址（调试器）	组件基址（系统总线）	组件偏移地址	大小	数据
0x000	DBGMCU	0xE008 1000	0x5008 1000	0x0 1000	4 KB	0x0000 1003
0x004	时间戳发生器	0xE008 2000	0x5008 2000	0x0 2000	4 KB	0x0000 2003
0x008	跟踪子系统 ROM 表	0xE009 0000	0x5009 0000	0x1 0000	4 KB	0x0001 0003
0x00C	Cortex-A7 子系统 ROM 表	0xE00C 0000	0x500C 0000	0x4 0000	4 KB	0x0004 0003
0x010	表顶部	-	-	-	-	0x0000 0000

表 603. 调试子系统 ROM 表（续）

ROM 表中的偏移地址	组件名称	组件基址（调试器）	组件基址（系统总线）	组件偏移地址	大小	数据
0x014 到 0xFC8	保留	-	-	-	-	0x0000 0000
0xFCC 到 0xFFC	ROM 表寄存器	-	-	-	-	请参见*表 606*

第两个表占用一个 4 KB、32 位宽的 APB 地址空间块——从 0xE009 0000 到 0xE009 0FFC（调试器访问时）或者从 0x5009 0000 到 0x5009 0FFC（从系统总线访问时）。它指向位于跟踪子系统中的 CoreSight™ 组件。

表 604. 跟踪子系统 ROM 表

ROM 表中的偏移地址	组件名称	组件基址（调试器）	组件基址（系统总线）	组件偏移地址	大小	入口
0x000	ETF	0xE009 2000	0x5009 2000	0x2000	4 KB	0x0000 2003
0x004	TPIU	0xE009 3000	0x5009 3000	0x3000	4 KB	0x0000 3003
0x008	跟踪 CTI	0xE009 4000	0x5009 4000	0x4000	4 KB	0x0000 4003
0x00C	表顶部	-	-	-	-	0x0000 0000
0x010 到 0xFC8	保留	-	-	-	-	0x0000 0000
0xFCC 到 0xFFC	ROM 表寄存器	-	-	-	-	请参见*表 606*

第三个表占用一个 4 KB、32 位宽的 APB 地址空间块——从 0xE00C 0000 到 0xE00C 0FFC（调试器访问时）或者从 0x500C 0000 到 0x500C 0FFC（从系统总线访问时）。它指向位于 Cortex-A7 子系统中的 CoreSight™ 组件。

表 605. Cortex-A7 子系统 ROM 表

ROM 表中的偏移地址	组件名称	组件基址（调试器）	组件基址（系统总线）	组件偏移地址	大小	数据
0x000	C-A7 DBG	0xE00D 0000	0x500D 0000	0x1 0000	4 KB	0x0001 0003
0x004	C-A7 PMU	0xE00D 1000	0x500D 1000	0x1 1000	4 KB	0x0001 1003
0x008	保留	-	-	-	-	0x0001 2002
0x00C	保留	-	-	-	-	0x0001 3002
0x010	保留	-	-	-	-	0x0001 4002
0x014	保留	-	-	-	-	0x0001 5002
0x018	保留	-	-	-	-	0x0001 6002
0x01C	保留	-	-	-	-	0x0001 7002

表 605. Cortex-A7 子系统 ROM 表（续）

ROM 表中的偏移地址	组件名称	组件基址（调试器）	组件基址（系统总线）	组件偏移地址	大小	数据
0x020	C-A7 CTI	0xE00D 8000	0x500D 8000	0x1 8000	4 KB	0x0001 8003
0x024	保留	-	-	-	-	0x0001 9002
0x028	保留	-	-	-	-	0x0001 A002
0x02C	保留	-	-	-	-	0x0001 B002
0x030	C-A7 ETM	0xE00D C000	0x500D C000	0x1 C000	4 KB	0x0001 C003
0x034	保留	-	-	-	-	0x0001 D002
0x038	保留	-	-	-	-	0x0001 E002
0x03C	保留	-	-	-	-	0x0001 F002
0x040	表顶部	-	-	-	-	0x0000 0000
0x044 到 0xFC8	保留	-	-	-	-	0x0000 0000
0xFCC 到 0xFFC	ROM 表寄存器	-	-	-	-	请参见*表 606*

每个 ROM 表的顶部包含多个只读寄存器，包括标准的 CoreSight™ 组件和外设标识寄存器，请参见*表 606*。

每个调试组件占用一个或多个 4 KB 地址空间块。这个地址空间块被称为组件的调试寄存器文件。

ROM 表项的组件偏移地址指向组件地址空间的最后一个 4 KB 块的起始处。该块始终包含组件及组件的外设标识寄存器，其起始地址为块起始地址 + 偏移地址 0xFD0。4 KB 计数位字段 PIDR4[7:4] 指定了组件的 4 KB 块的数量。因此，查找组件地址空间起始地址的过程如下：

1. 读取组件的 ROM 表项并提取组件的偏移地址。偏移地址为 ROM 表项的位 [31:12]。
2. 使用该偏移地址与 ROM 表的基址一起计算组件的基址。组件基址的计算公式如下：

   组件基址 (Component_Base_Address) = ROM 基址 (ROM_Base_Address) + (偏移地址 (Address_Offset) << 12)

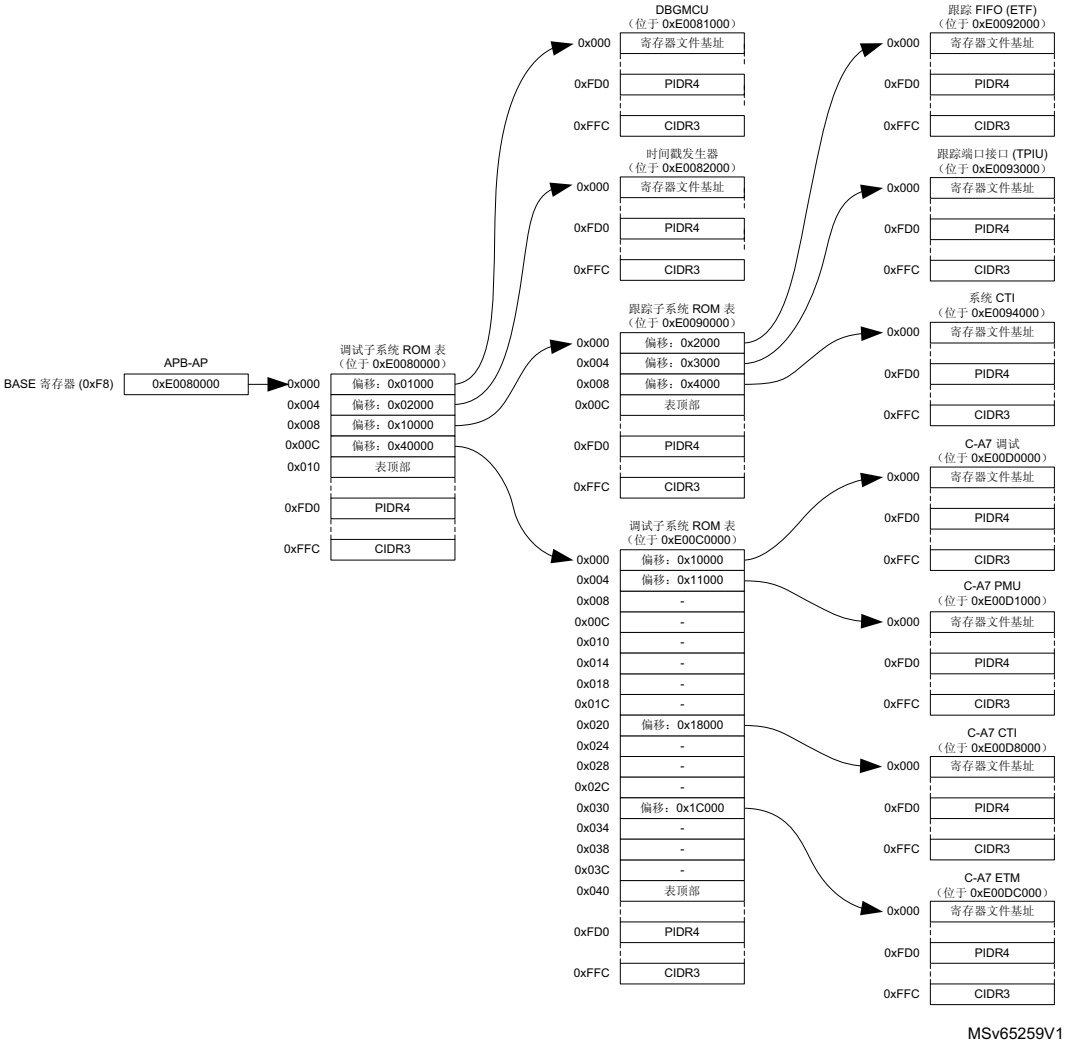
   组件基址 (Component_Base_Address) 就是组件地址空间的最后 4 KB 块的起始地址。
3. 读取组件的外设 ID4 寄存器。该寄存器的地址为：

   外设 ID4 寄存器地址 (Peripheral_ID4_address) = 组件基址 (Component_Base_Address) + 0xFD0
4. 从外设 ID4 寄存器的地址值中提取 4 KB 计数位字段 [7:4]。
5. 使用该 4 KB 计数值计算组件地址空间的起始地址。如果 4 KB 计数位字段为 0000（对应于计数值 1），则组件的地址空间的起始地址为步骤 2 中得到的组件基址 (Component_Base_Address)。

调试 APB 上的 CoreSight™ 组件拓扑如*图 757* 所示。

有关 ROM 表使用的更多信息，请参见《Arm® 调试接口架构规范》*[1]*。

图 757. 调试 APB CoreSight™ 拓扑



MSv65259V1

ROM 表寄存器

SYSROM 内存类型寄存器 (SYSROM_MEMTYPE)

SYSROM memory type register

偏移地址：0xFCC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSMEM
															r

位 31:1 保留，必须保持复位值。

位 0 **SYSMEM**：系统内存 (system memory)
0x0：该总线无系统内存

SYSROM CoreSight 外设标识寄存器 4 (SYSROM_PIDR4)

SYSROM CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SIZE[3:0]**：寄存器文件大小 (register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码
0x0：意法半导体 JEDEC 连续代码

SYSROM CoreSight 外设标识寄存器 0 (SYSROM_PIDR0)

SYSROM CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位 [7:0] (part number bits [7:0])
0x00：ROM 表产品编号

SYSROM CoreSight 外设标识寄存器 1 (SYSROM_PIDR1)

SYSROM CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位 [3:0] (JEP106 identity code bits [3:0])
0x0：意法半导体 JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位 [11:8] (part number bits [11:8])
0x5：ROM 表产品编号

SYSROM CoreSight 外设标识寄存器 2 (SYSROM_PIDR2)

SYSROM CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (component revision number)
0x0：r0p0

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
0x1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位 [6:4] (JEP106 identity code bits [6:4])
0x2：意法半导体 JEDEC 代码

SYSROM CoreSight 外设标识寄存器 3 (SYSROM_PIDR3)

SYSROM CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (customer modified)
0x0：无客户修改

SYSROM CoreSight 组件标识寄存器 0 (SYSROM_CIDR0)

SYSROM CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位 [7:0] (component ID bits [7:0])
0x0D：公共 ID 值

SYSROM CoreSight 外设标识寄存器 1 (SYSROM_CIDR1)

SYSROM CoreSight peripheral identity register 1

偏移地址：0xFF4

复位值：0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位 [15:12] (component ID bits [15:12])——组件类
0x1：ROM 表组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位 [11:8] (component ID bits [11:8])
0x0：公共 ID 值

SYSROM CoreSight 组件标识寄存器 0 (SYSROM_CIDR2)

SYSROM CoreSight component identity register 0

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位 [23:16] (component ID bits [23:16])
0x05：公共 ID 值

SYSROM CoreSight 组件标识寄存器 3 (SYSROM_CIDR3)

SYSROM CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位 [31:24] (component ID bits [31:24])
0xB1：公共 ID 值

ROM 表寄存器

表 606. ROM 表寄存器映射和复位值

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFCC	**SYSROM_MEMTYPE**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSMEM
	Reset value																																0
0xFD0	**SYSROM_PIDR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFD4	**SYSROM_PIDR5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	**SYSROM_PIDR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	**SYSROM_PIDR7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	**SYSROM_PIDR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0xFE4	**SYSROM_PIDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
	Reset value																									0	0	0	0	0	1	0	1
0xFE8	**SYSROM_PIDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
	Reset value																									0	0	0	1	1	0	1	0
0xFEC	**SYSROM_PIDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND [3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	**SYSROM_CIDR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	**SYSROM_CIDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
	Reset value																									0	0	0	1	0	0	0	0

表 606. ROM 表寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFF8	**SYSROM_CIDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	**SYSROM_CIDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

59.10.2 CoreSight™ 全局时间戳发生器 (TSGEN)

CoreSight™ 全局时间戳发生器包含一个 64 位计数器，该计数器为系统中的所有跟踪源提供公共的参考时序，即 ETM。ETM 在跟踪数据流中插入时间戳，以允许跟踪分析器恢复跟踪数据包的时序。

TSG 寄存器可通过调试 APB 访问。这允许调试器或调试软件：

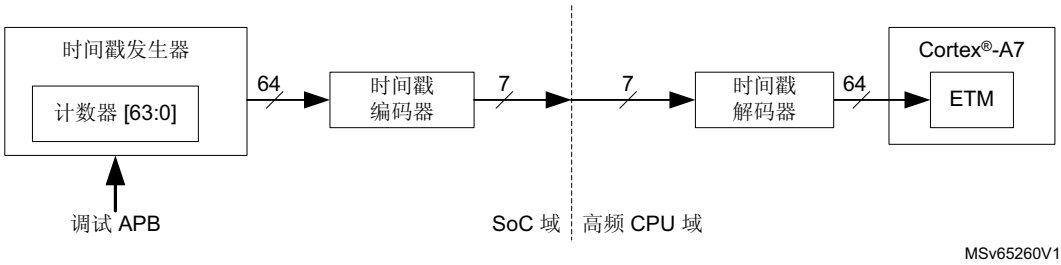
- 启动和停止时间戳递增。
- 使时间戳计数器在进入调试状态时停止计数。若要使能该功能，必须配置跟踪子系统 CTI 以通知时间戳发生器何时进入了系统级调试状态（即停止所有 CPU 内核）。
- 读取当前的时间戳值。
- 更改当前的时间戳值。在更改时，必须停止时间戳计数器。当时间戳值被更改时，时间戳发生器重新同步所有跟踪源。
- 更改报告的时间戳增量。

有关 CoreSight™ 时间戳发生器 CoreSight™ 组件的更多信息，请参见《Arm® CoreSight™ SoC-400 技术参考手册》*[2]*。

时间戳发生器位于 ck_apb_dbg 域，而时间戳分布到 Cortex®-A7。为了简化时钟域边界上的分布，64 位时间戳采用 7 位编码，然后在目标电源域中解码。时间戳分布如*图 758* 所示。

只要 TSGEN 或 Cortex®-A7 的频率发生变化，处理器的 ETM 就会接收到 TSCLKCHANGE 输入上的脉冲。

图 758. CoreSight™ 时间戳分布



时间戳发生器寄存器

TSG 计数器控制寄存器 (TSG_CNTCR)

TSG counter control register

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDBG	EN
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **HDBG**：遇调试停止 (halt on debug)
- 0：正常工作
- 1：（通过 CTI）检测到系统级调试状态时停止计数器

位 0 **EN**：使能 (enable)
- 0：禁止计数器
- 1：使能计数器并递增计数

TSG 计数器状态寄存器 (TSG_CNTSR)

TSG counter status register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGH	Res.
														r	

位 31:2 保留，必须保持复位值。

位 1 **DBGH**：因调试停止 (debug halted)
- 0：正常工作
- 1：因系统级调试状态而停止计数器

位 0 保留，必须保持复位值。

TSG 当前计数器值低位寄存器 (TSG_CNTCVL)

TSG current counter value lower register

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNTCVU_L_32[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNTCVU_L_32[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CNTCVU_L_32[31:0]**：时间戳计数器当前值（低 32 位）(timestamp counter current value (least significant 32 bits))

要更改当前的时间戳值，需先将新值的低 32 位写入该寄存器，再将高 32 位写入 TSG_CNTCVU。在写入 TSG_CNTCVU 寄存器之前，时间戳值保持不变。

注意：在写入 TSG_CNTCVL 寄存器之前，必须清零 TSG_CNTCR 寄存器的 EN 位。

TSG 当前计数器值高位寄存器 (TSG_CNTCVU)

TSG current counter value upper register

偏移地址：0x00C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNTCVU_U_32[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNTCVU_U_32[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CNTCVU_U_32[31:0]**：时间戳计数器当前值（高 32 位）(timestamp counter current value (most significant 32 bits))

要更改当前的时间戳值，需先将新值的低 32 位写入 TSG_CNTCVL，再将高 32 位写入该寄存器。当写入该寄存器时，64 位时间戳值会更新为两次写入的值。

注意：在写入 TSG_CNTCVU 寄存器之前，必须清零 TSG_CNTCR 寄存器的 EN 位。

TSG 基频 ID 寄存器 (TSG_CNTFID0)

TSG base frequency ID register

偏移地址：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FREQ[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREQ[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **FREQ[31:0]**：以 Hz 为单位的 TSG 寄存器的增量频率 (increment frequency of TSG counter in Hz)

只要跟踪发生器时钟频率发生变化，就使用其对该位字段编程。

TSG CoreSight 外设标识寄存器 4 (TSG_PIDR4)

TSG CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SIZE[3:0]**：寄存器文件大小 (register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码(JEP106 continuation code)
0x4：Arm® JEDEC 代码

TSG CoreSight 外设标识寄存器 0 (TSG_PIDR0)

TSG CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位 [7:0] (part number bits [7:0])
0x01：TSG 产品编号

TSG CoreSight 外设标识寄存器 1 (TSG_PIDR1)

TSG CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位 [3:0] (JEP106 identity code bits [3:0])
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位 [11:8] (part number bits [11:8])
0x1：TSG 产品编号

TSG CoreSight 外设标识寄存器 2 (TSG_PIDR2)

TSG CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 001B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (component revision number)
0x1：r0p1

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
0x1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位 [6:4] (JEP106 identity code bits [6:4])
0x3：Arm® JEDEC 代码

TSG CoreSight 外设标识寄存器 3 (TSG_PIDR3)

TSG CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (customer modified)
0x0：无客户修改

TSG CoreSight 组件标识寄存器 0 (TSG_CIDR0)

TSG CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位 [7:0] (component ID bits [7:0])
0x0D：公共 ID 值

TSG CoreSight 外设标识寄存器 1 (TSG_CIDR1)

TSG CoreSight peripheral identity register 1

偏移地址：0xFF4

复位值：0x0000 00F0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位 [15:12] (component ID bits [15:12])——组件类
0xF：CoreSight™ Soc-400 组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位 [11:8] (Component ID bits [11:8])
0x0：公共 ID 值

TSG CoreSight 组件标识寄存器 2 (TSG_CIDR2)

TSG CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位 [23:16] (component ID bits [23:16])
0x05：公共 ID 值

TSG CoreSight 组件标识寄存器 3 (TSG_CIDR3)

TSG CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位 [31:24] (component ID bits [31:24])
0xB1：公共 ID 值

时间戳发生器寄存器映射

表 607. TSGEN 寄存器映射和复位值

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**TSG_CNTCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDBG	EN
	Reset value																															0	0
0x004	**TSG_CNTSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGH	Res.
	Reset value																															0	

表 607. TSGEN 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x008	**TSG_CNTCVL**	CNTCVU_L_32[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	**TSG_CNTCVU**	CNTCVU_U_32[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	**TSG_CNTFID0**	FREQ[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xFD0	**TSG_PIDR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON[3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	**TSG_PIDR5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	**TSG_PIDR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	**TSG_PIDR7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	**TSG_PIDR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	0	0	0	0	0	0	1
0xFE4	**TSG_PIDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
	Reset value																									1	0	1	1	0	0	0	1
0xFE8	**TSG_PIDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
	Reset value																									0	0	0	1	1	0	1	1
0xFEC	**TSG_PIDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND [3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	**TSG_CIDR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1

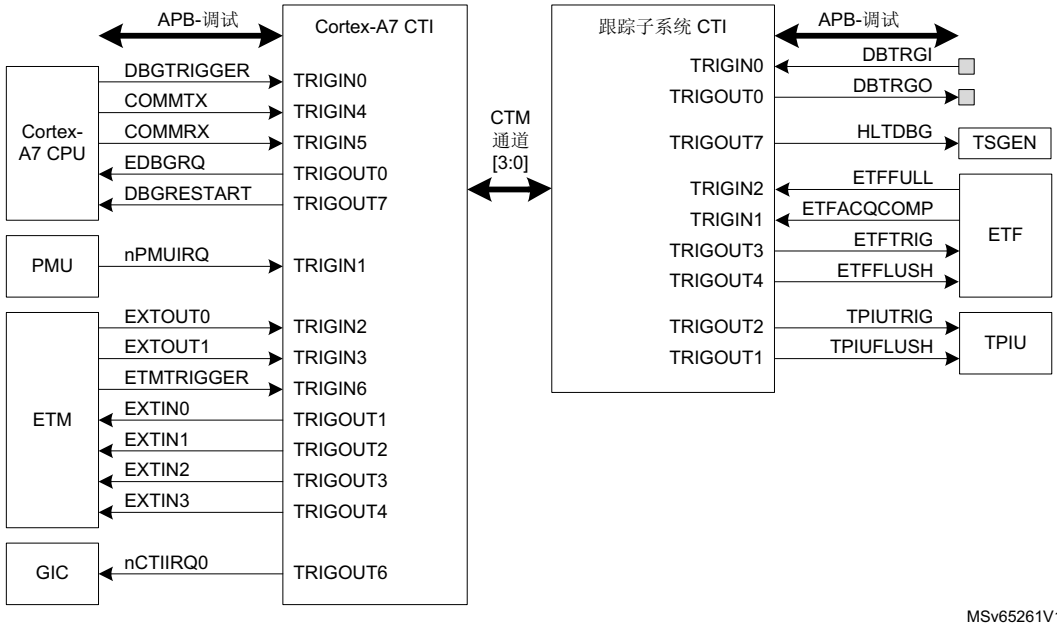
表 607. TSGEN 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFF4	**TSG_CIDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
	Reset value																									1	1	1	1	0	0	0	0
0xFF8	**TSG_CIDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	**TSG_CIDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

59.10.3 交叉触发接口 (CTI) 和矩阵 (CTM)

交叉触发接口 (CTI) 和交叉触发矩阵 (CTM) 一起形成了 CoreSight™ 嵌入式交叉触发功能。有两个 CTI 组件，一个在跟踪子系统中，一个用于 Cortex®-A7。两个 CTI 通过 CTM 相互连接。调试器可以通过调试 APB 访问跟踪子系统 CTI 和 Cortex®-A7 CTI.

图 759. 嵌入式交叉触发



MSv65261V1

CTI 允许来自各种来源的事件触发调试和/或跟踪活动。例如，如果检测到外部触发输入信号发生变换，则可以启动代码跟踪。

每个 CTI 最多拥有 8 个触发输入和 8 个触发输出。任何输入都可以连接到同一个 CTI 的任何输出，也可通过 CTM 连接到另一个 CTI 上的任何输出。

表 608 到*表 611* 中列出了各个 CTI 的触发输入和输出信号。

表 608. 跟踪子系统 CTI 输入

	源信号	源组件	备注
0	DBTRIGI	GPIO	外部触发输入——允许外部信号生成调试事件
1	ETFACQCOMP	ETF	ETF 捕获完成——当跟踪 FIFO 为空时，允许生成调试事件
2	ETFFULL	ETF	ETF 已满标志——当跟踪 FIFO 已满时，允许生成调试事件
3	-	-	未使用
4	-	-	未使用
5	-	-	未使用
6	-	-	未使用
7	-	-	未使用

表 609. 跟踪子系统 CTI 输出

	输出信号	目标组件	备注
0	DBTRIGO	GPIO	外部 I/O 触发输出——允许监视外部 DBTRIGO 引脚上的事件
1	TPIUFLUSH	TPIU	跟踪端口刷新触发——使 TPIU FIFO 被刷新
2	TPIUTRIG	TPIU	跟踪端口使能触发——启动外部跟踪端口上的跟踪输出
3	ETFTRIG	ETF	ETF 使能触发——开始填充跟踪 FIFO
4	ETFFLUSH	ETF	ETF 刷新触发——使跟踪 FIFO 被刷新
5	-	-	未使用
6	-	-	未使用
7	HLTDBG	TSGEN	在调试中停止时间戳——冻结 TSGEN 时间戳计数器

表 610. Cortex®-A7 CTI 输入

	源信号	源组件	备注
0	DBGTRIGGER	Cortex®-A7 CPU	当内核进入调试状态时为高电平脉冲
1	nPMUIRQ	PMU	性能监视中断
2	EXTOUT0	ETM	ETM 外部触发输出
3	EXTOUT1	ETM	ETM 外部触发输出
4	COMMTX	Cortex®-A7 CPU	调试通信端口发送
5	COMMRX	Cortex®-A7 CPU	调试通信端口接收
6	ETMTRIGGER	ETM	ETM 触发
7	-	-	未使用

表 611. Cortex®-A7 CTI 输出

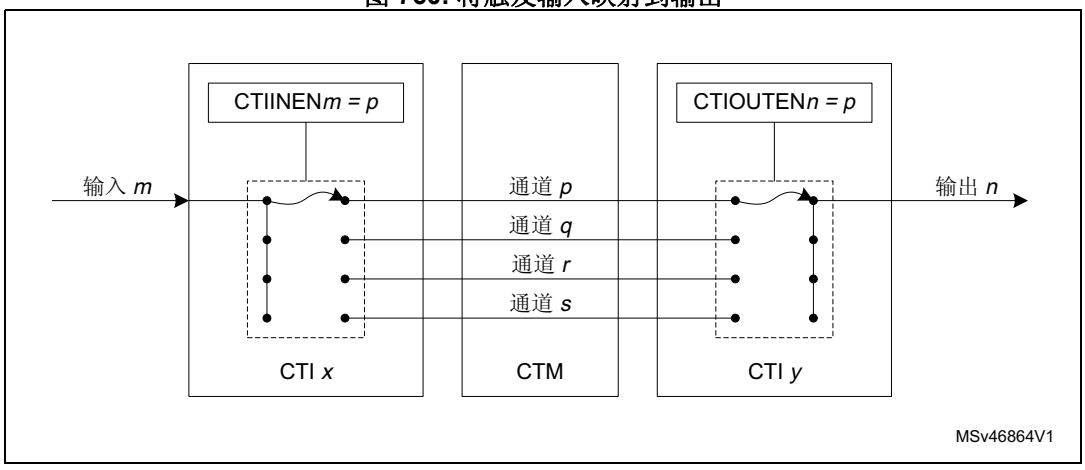
	输出信号	目标组件	备注
0	EDBGRQ	Cortex®-A7 CPU	CPU 停止请求——将 CPU 置于调试模式
1	EXTIN0	ETM	ETM 外部输入
2	EXTIN1	ETM	ETM 外部输入
3	EXTIN2	ETM	ETM 外部输入
4	EXTIN3	ETM	ETM 外部输入
5	-	-	未使用
6	nCTIIRQ	GIC	CTI 中断
7	DBGRESTART	Cortex®-A7 CPU	CPU 重启请求——CPU 退出调试模式

交叉触发矩阵中有四个事件通道，在不同 CTI 的触发输入和输出之间，最多支持四个并行双向连接。要将 CTI *x* 上的输入编号 *m* 连接到 CTI *y* 上的输出编号 *n*，则必须使用 CTI *x* 的 CTI_INEN*m* 寄存器将输入连接到事件通道 *p*。同一通道 *p* 必须使用 CTI *y* 的 CTI_OUTEN*n* 寄存器连接到输出。

注： *即使输入和输出属于同一个 CTI，上述连接设置也同样有效。*

一个输入可连接到多个通道（最多四个）。因此，一个输入可被发送到多个输出。类似地，一个输出可连接到多个输入。还可以将多个输入/输出连接到同一通道。

图 760. 将触发输入映射到输出



有关交叉触发接口 CoreSight™ 组件的更多信息，请参见《Arm® CoreSight™ SoC-400 技术参考手册》*[2]*。

交叉触发接口寄存器

每个 CTI 的寄存器文件基址通过其所连接总线的 ROM 表定义。每个 CTI 的寄存器是相同的。

CTI 控制寄存器 (CTI_CONTROL)

CTI control register

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GLBEN
															rw

位 31:1 保留，必须保持复位值。

位 0 **GLBEN**：全局使能 (global enable)
0：禁止交叉触发
1：使能交叉触发

CTI 触发确认寄存器 (CTI_INTACK)

CTI trigger acknowledge register

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTACK[7:0]							
								w	w	w	w	w	w	w	w

位 31:8 保留，必须保持复位值。

位 7:0 **INTACK[7:0]**：触发确认 (trigger acknowledge)
每个 CTITRIGOUT 输出都使用一个寄存器位。当向该寄存器的某位写入 1 时，则确认相应的 CTITRIGOUT 输出，使其被清零。

CTI 应用程序触发设置寄存器 (CTI_APPSET)

CTI application trigger set register

偏移地址：0x014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPSET[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **APPSET[3:0]**：设置通道事件 (channel event set)

读：

XXX0：通道 0 事件处于未激活状态

XXX1：通道 0 事件处于激活状态

XX0X：通道 1 事件处于未激活状态

XX1X：通道 1 事件处于激活状态

X0XX：通道 2 事件处于未激活状态

X1XX：通道 2 事件处于激活状态

0XXX：通道 3 事件处于未激活状态

1XXX：通道 3 事件处于激活状态

写：

XXX0：无影响

XXX1：设置通道 0 事件

XX0X：无影响

XX1X：设置通道 1 事件

X0XX：无影响

X1XX：设置通道 2 事件

0XXX：无影响

1XXX：设置通道 3 事件

CTI 应用程序触发清除寄存器 (CTI_APPCLEAR)

CTI application trigger clear register

偏移地址：0x018

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPCLEAR[3:0]			
												w	w	w	w

位 31:4 保留，必须保持复位值。

位 3:0 **APPCLEAR[3:0]**：清除通道事件 (channel event clear)

XXX0：无影响

XXX1：清除通道 0 事件

XX0X：无影响

XX1X：清除通道 1 事件

X0XX：无影响

X1XX：清除通道 2 事件

0XXX：无影响

1XXX：清除通道 3 事件

CTI 应用程序脉冲寄存器 (CTI_APPPULSE)

CTI application pulse register

偏移地址：0x01C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPULSE[3:0]			
												w	w	w	w

位 31:4 保留，必须保持复位值。

位 3:0 **APPULSE[3:0]**：脉冲通道事件 (pulse channel event)

该寄存器立即将自己清零。

XXX0：无影响

XXX1：生成通道 0 脉冲

XX0X：无影响

XX1X：生成通道 1 脉冲

X0XX：无影响

X1XX：生成通道 2 脉冲

0XXX：无影响

1XXX：生成通道 3 脉冲

CTI 触发输入使能 x 寄存器 (CTI_INENx)

CTI trigger in enable x registers

偏移地址：0x020 + 0x04 * x（x = 0 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **TRIGINEN[3:0]**：交叉触发事件使能/禁止 (cross-trigger event enable / disable)

当 CTITRIGINn（n = 0 到 7）处于激活状态时，使能或禁止四个通道中的每个通道上的交叉触发事件。

XXX0：触发 n 不生成通道 0 事件
XXX1：触发 n 生成通道 0 事件
XX0X：触发 n 不生成通道 1 事件
XX1X：触发 n 生成通道 1 事件
X0XX：触发 n 不生成通道 2 事件
X1XX：触发 n 生成通道 2 事件
0XXX：触发 n 不生成通道 3 事件
1XXX：触发 n 生成通道 3 事件

CTI 触发输出使能 x 寄存器 (CTI_OUTENx)

CTI Trigger out enable x registers

偏移地址：0x0A0 + 0x04 * x（x = 0 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **TRIGOUTEN[3:0]**：输出触发使能/禁止 (output trigger enable / disable)

对于每个通道，该位字段定义了该通道上的事件是否能够生成 CTITRIGOUTn（n = 0 到 7）触发。

XXX0：通道 0 事件不生成触发输出 n 触发
XXX1：通道 0 事件生成触发输出 n 触发
XX0X：通道 1 事件不生成触发输出 n 触发
XX1X：通道 1 事件生成触发输出 n 触发
X0XX：通道 2 事件不生成触发输出 n 触发
X1XX：通道 2 事件生成触发输出 n 触发
0XXX：通道 3 事件不生成触发输出 n 触发
1XXX：通道 3 事件生成触发输出 n 触发

CTI 触发输入状态寄存器 (CTI_TRGISTS)

CTI trigger in status register

偏移地址：0x130

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINSTATUS[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **TRIGINSTATUS[7:0]**：触发输入状态 (trigger input status)

每个 CTITRIGIN 输入都使用一个寄存器位。如果某位设置为 1，则表示相应触发输入处于激活状态。如果某位设置为 0，则表示相应触发输入处于未激活状态。

CTI 触发输出状态寄存器 (CTI_TRGOSTS)

CTI Trigger out status register

偏移地址：0x134

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTSTATUS[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **TRIGOUTSTATUS[7:0]**：触发输出状态 (trigger output status)

每个 CTITRIGOUT 输出都使用一个寄存器位。如果某位设置为 1，则表示相应触发输出处于激活状态。如果某位设置为 0，则表示相应触发输出处于未激活状态。

CTI 通道输入状态寄存器 (CTI_CHINSTS)

CTI channel in status register

偏移地址：0x138

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHINSTATUS[3:0]			
												r	r	r	r

位 31:4 保留，必须保持复位值。

位 3:0 **CHINSTATUS[3:0]**：通道输入状态 (channel input status)

每个通道输入都使用一个寄存器位。如果某位设置为 1，则表示相应通道输入处于激活状态。如果某位设置为 0，则表示相应通道输入处于未激活状态。

CTI 通道输出状态寄存器 (CTI_CHOUTSTS)

CTI channel out status register

偏移地址：0x13C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHOUTSTATUS[3:0]			
												r	r	r	r

位 31:4 保留，必须保持复位值。

位 3:0 **CHOUTSTATUS[3:0]**：通道输出状态 (channel output status)
每个通道输出都使用一个寄存器位。如果某位设置为 1，则表示相应通道输出处于激活状态。如果某位设置为 0，则表示相应通道输出处于未激活状态。

CTI 通道门控寄存器 (CTI_GATE)

CTI channel gate register

偏移地址：0x140

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GATEEN[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **GATEEN[3:0]**：通道输出使能 (channel output enable)
对于每个通道，定义了该通道事件能否通过 CTM 传播到其他 CTI。
XXX0：通道 0 事件不能传播
XXX1：通道 0 事件能够传播
XX0X：通道 1 事件不能传播
XX1X：通道 1 事件能够传播
X0XX：通道 2 事件不能传播
X1XX：通道 2 事件能够传播
0XXX：通道 3 事件不能传播
1XXX：通道 3 事件能够传播

CTI 声明标记设置寄存器 (CTI_CLAIMSET)

CTI claim tag set register

偏移地址：0xFA0

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMSET[3:0]**：设置声明标记位 (sets claim tag bits)

写：

- 0000：无影响
- xxx1：将位 0 置 1
- xx1x：将位 1 置 1
- x1xx：将位 2 置 1
- 1xxx：将位 3 置 1

读：

- 0xF：表示声明标记使用四个位

CTI 声明标记清零寄存器 (CTI_CLAIMCLR)

CTI claim tag clear register

偏移地址：0xFA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMCLR[3:0]**：复位声明标记位 (resets claim tag bits)

写：

- 0000：无影响
- xxx1：将位 0 清零
- xx1x：将位 1 清零
- x1xx：将位 2 清零
- 1xxx：将位 3 清零

读：

- 返回声明标记的当前值

CTI 锁定访问寄存器 (CTI_LAR)

CTI lock access register

偏移地址：0xFB0

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ACCESS_W[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS_W[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **ACCESS_W[31:0]**：CTI 寄存器写访问使能 (CTI register write access enable)
使能处理器内核写访问某些 CTI 寄存器（调试器无需解锁组件）。
0xC5ACCE55：使能写访问
其他值：禁止写访问

CTI 锁定状态寄存器 (CTI_LSR)

CTI lock status register

偏移地址：0xFB4

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCKTYPE	LOCKGRANT	LOCKEXIST
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **LOCKTYPE**：CTI_LAR 寄存器大小 (CTI_LAR register size)
指示 CTI_LAR 寄存器的大小。
0：32 位

位 1 **LOCKGRANT**：当前锁定状态 (current lock status)
当外部调试器读取该位时，始终读为零。
0：允许写访问
1：阻止写访问。只允许读操作。

位 0 **LOCKEXIST**：锁定控制存在 (lock control existence)
表示是否存在锁定控制机制。当外部调试器读取该位时，始终读为零。
0：不存在锁定控制机制
1：存在锁定控制机制

CTI 认证状态寄存器 (CTI_AUTHSTAT)

CTI authentication status register

偏移地址：0xFB8

复位值：0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:6 **SNID[1:0]**：安全非侵入式调试的安全等级 (security level for secure non-invasive debug)
0x0：未设计

位 5:4 SID**[1:0]**：安全侵入式调试的安全等级 (security level for secure invasive debug)
0x0：未设计

位 3:2 **NSNID[1:0]**：非安全非侵入式调试的安全等级 (security level for non-secure non-invasive debug)
0x2：禁止
0x3：使能

位 1:0 **NSID[1:0]**：非安全侵入式调试的安全等级 (security level for non-secure invasive debug)
0x2：禁止
0x3：使能

CTI 设备配置寄存器 (CTI_DEVID)

CTI device configuration register

偏移地址：0xFC8

复位值：0x0004 0800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NUMCH[3:0]			
												r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMTRIG[7:0]								Res.	Res.	Res.	EXTMUXNUM[4:0]				
r	r	r	r	r	r	r	r				r	r	r	r	r

位 31:20 保留，必须保持复位值。

位 19:16 **NUMCH[3:0]**：可用 ECT 通道数量 (number of ECT channels available)
0x4：4 个通道

位 15:8 **NUMTRIG[7:0]**：可用 ECT 触发数量 (number of ECT triggers available)
0x8：8 个触发输入和 8 个触发输出

位 7:5 保留，必须保持复位值。

位 4:0 **EXTMUXNUM[4:0]**：触发输入/输出多路复用器数量 (number of trigger input/output multiplexers)
0x0：无

CTI 设备类型标识符寄存器 (CTI_DEVTYPE)

CTI device type Identifier register

偏移地址：0xFCC

复位值：0x0000 0014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SUBTYPE[3:0]**：子分类 (sub-classification)
0x1：表示该组件为交叉触发组件

位 3:0 **MAJORTYPE[3:0]**：主分类 (major classification)
0x4：表示该组件允许调试器控制 CoreSight™ SoC-400 系统中的其他组件

CTI CoreSight 外设标识寄存器 4 (CTI_PIDR4)

CTI CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SIZE[3:0]**：寄存器文件大小 (register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：Arm® JEDEC 代码

CTI CoreSight 外设标识寄存器 0 (CTI_PIDR0)

CTI CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0006

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位 [7:0] (part number bits [7:0])
0x06：CTI 产品编号

CTI CoreSight 外设标识寄存器 1 (CTI_PIDR1)

CTI CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位 [3:0] (JEP106 identity code bits [3:0])
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位 [11:8] (part number bits [11:8])
0x9：CTI 产品编号

CTI CoreSight 外设标识寄存器 2 (CTI_PIDR2)

CTI CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 004B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (component revision number)
0x4：r0p5

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
0x1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位 [6:4] (JEP106 identity code bits [6:4])
0x3：Arm® JEDEC 代码

CTI CoreSight 外设标识寄存器 3 (CTI_PIDR3)

CTI CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (customer modified)
0x0：无客户修改

CTI CoreSight 组件标识寄存器 0 (CTI_CIDR0)

CTI CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位 [7:0] (component ID bits [7:0])
0x0D：公共 ID 值

CTI CoreSight 外设标识寄存器 1 (CTI_CIDR1)

CTI CoreSight peripheral identity register 1

偏移地址：0xFF4

复位值：0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位 [15:12] (component ID bits [15:12])——组件类
0x9：CoreSight™ 组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位 [11:8] (component ID bits [11:8])
0x0：公共 ID 值

CTI CoreSight 组件标识寄存器 0 (CTI_CIDR2)

CTI CoreSight component identity register 0

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位 [23:16] (component ID bits [23:16])
0x05：公共 ID 值

CTI CoreSight 组件标识寄存器 3 (CTI_CIDR3)

CTI CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位 [31:24] (component ID bits [31:24])
0xB1：公共 ID 值

交叉触发接口寄存器映射

表 612. CTI 寄存器映射和复位值

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**CTI_CONTROL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GLBEN
	Reset value																																0
0x010	**CTI_INTACK**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTACK[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x014	**CTI_APPSET**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPSET[3:0]			
	Reset value																													0	0	0	0
0x018	**CTI_APPCLEAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPCLEAR[3:0]			
	Reset value																													0	0	0	0
0x01C	**CTI_APPPULSE**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APULSE[3:0]			
	Reset value																													0	0	0	0
0x020	**CTI_INEN0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN[3:0]			
	Reset value																													0	0	0	0
0x024	**CTI_INEN1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN[3:0]			
	Reset value																													0	0	0	0
0x028	**CTI_INEN2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN[3:0]			
	Reset value																													0	0	0	0
0x02C	**CTI_INEN3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN[3:0]			
	Reset value																													0	0	0	0

表 612. CTI 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x030	**CTI_INEN4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN[3:0]			
	Reset value																													0	0	0	0
0x034	**CTI_INEN5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN[3:0]			
	Reset value																													0	0	0	0
0x038	**CTI_INEN6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN[3:0]			
	Reset value																													0	0	0	0
0x03C	**CTI_INEN7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN[3:0]			
	Reset value																													0	0	0	0
0x0A0	**CTI_OUTEN0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN[3:0]			
	Reset value																													0	0	0	0
0x0A4	**CTI_OUTEN1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN[3:0]			
	Reset value																													0	0	0	0
0x0A8	**CTI_OUTEN2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN[3:0]			
	Reset value																													0	0	0	0
0x0AC	**CTI_OUTEN3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN[3:0]			
	Reset value																													0	0	0	0

表 612. CTI 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0B0	**CTI_OUTEN4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN[3:0]			
	Reset value																													0	0	0	0
0x0B4	**CTI_OUTEN5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN[3:0]			
	Reset value																													0	0	0	0
0x0B8	**CTI_OUTEN6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN[3:0]			
	Reset value																													0	0	0	0
0x0BC	**CTI_OUTEN7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN[3:0]			
	Reset value																													0	0	0	0
0x130	**CTI_TRIGISTS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINSTATUS[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x134	**CTI_TRIGOSTS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTSTATUS[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x138	**CTI_CHINSTS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHISTATUS[3:0]			
	Reset value																													0	0	0	0
0x13C	**CTI_CHOUTSTS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHOSTATUS[3:0]			
	Reset value																													0	0	0	0
0x140	**CTI_GATE**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GATEEN [3:0]			
	Reset value																													1	1	1	1

表 612. CTI 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFA0	**CTI_CLAIMSET**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
	Reset value																													1	1	1	1
0xFA4	**CTI_CLAIMCLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
	Reset value																													0	0	0	0
0xFB0	**CTI_LAR**	ACCESS_W[31:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0xFB4	**CTI_LSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCKTYPE	LOCKGRANT	LOCKEXIST
	Reset value																														0	1	1
0xFB8	**CTI_AUTHSTAT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
	Reset value																									0	0	0	0	0	1	0	1
0xFC8	**CTI_DEVID**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NUMCH[3:0]				NUMTRIG[7:0]								Res.	Res.	Res.	EXMUXNUM[4:0]				
	Reset value													0	1	0	0	0	0	0	0	1	0	0	0				0	0	0	0	0
0xFCC	**CTI_DEVTYPE**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUB[3:0]				MAJOR[3:0]			
	Reset value																									0	0	0	1	0	1	0	0
0xFD0	**CTI_PIDR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON[3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	**CTI_PIDR5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	**CTI_PIDR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	**CTI_PIDR7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																

表 612. CTI 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFE0	**CTI_PIDR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	0	0	0	0	1	1	0
0xFE4	**CTI_PIDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
	Reset value																									1	0	1	1	1	0	0	1
0xFE8	**CTI_PIDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
	Reset value																									0	1	0	0	1	0	1	1
0xFEC	**CTI_PIDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND [3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	**CTI_CIDR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	**CTI_CIDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
	Reset value																									1	0	0	1	0	0	0	0
0xFF8	**CTI_CIDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	**CTI_CIDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

59.10.4 嵌入式跟踪 FIFO (ETF)

ETF 是一款 4 KB 内存，用来从 ETM 捕获跟踪数据。ETF 是 CoreSight™ 跟踪存储控制器组件的配置。

ETF 可用于三种模式（在模式寄存器中选择）：

1. 硬件 FIFO 模式

   跟踪内存作为 FIFO 使用，通过 ATB 主接口排空。跟踪数据被捕获到跟踪 RAM 中，当跟踪 RAM 已满时，传入的跟踪流将停止。当跟踪缓冲区非空时，跟踪数据流从 ATB 主接口送到 TPIU。

   在该模式下，FIFO 的作用是平滑到达跟踪端口的跟踪信息流。由于跟踪数据本身具有突发性，峰值数据速率可以轻松超出端口容量，从而导致上溢。ETF 可在跟踪端口实现稳定的数据速率，其大小可以根据平均速率而非峰值速率来调整。跟踪信息由跟踪端口分析器工具实时片外存储，因此跟踪日志能够非常大。

2. 软件 FIFO 模式

   跟踪内存作为 FIFO 使用，当跟踪被捕获时，可通过 RRD 寄存器读取。跟踪数据被捕获到跟踪 RAM 中，当跟踪 RAM 已满时，传入的跟踪流将停止。

   该模式允许通过软件（在其中一个内核上运行）由 DMA 将跟踪信息传输到系统内存、或传输到一个高速接口（SPI、USB 等等），甚至对跟踪信息进行监视。注意，与硬件 FIFO 模式不同，该模式是侵入式的，因为其使用由处理器共享的系统资源。

3. 循环缓冲区模式

   跟踪内存作为循环缓冲区使用。跟踪数据被捕获到跟踪内存，该跟踪内存起始地址为写指针寄存器指向的位置。即使跟踪内存已满，传入的跟踪数据仍继续覆盖到跟踪内存，直到出现停止条件。

   在该模式下，ETF 在片上存储跟踪数据，所以跟踪日志的大小限制为 ETF SRAM 的大小，本案例中为 4 KB。作为一个循环缓冲区，当 FIFO 变满时，传入的跟踪数据会覆盖最早存储的数据，最早存储的数据将丢失。因此，在缓冲区停止之前，跟踪缓冲区中的内容都表示最新的处理器活动，而不是跟踪开始后的所有活动。

   有三种方法可以在跟踪停止后读取缓冲区内容：

   – 通过跟踪端口——在使能了 TPIU 的情况下，缓冲区的内容通过跟踪端口输出。可通过将 ETF_FFCR 寄存器中的排空缓冲区位置 1 实现此操作。
   – 通过调试端口——调试器可以通过 RRD 寄存器读取缓冲区，该寄存器可通过调试 APB 访问。
   – 通过软件——由于调试 APB 可以通过系统总线访问，所以处理器可以通过 RRD 寄存器读取缓冲区。这也可以通过软件触发的 DMA 传输来执行。由于无法使用硬件 DMA 请求，因此可以使用 FULL 标志通过交叉触发器来触发中断。然后，软件可以读取 ETF_CBUFLVL 寄存器来确定要读取的数据数量。

有关跟踪存储控制器 CoreSight™ 组件的更多信息，请参见《Arm® CoreSight™ 跟踪存储控制器技术参考手册》*[3]*。

嵌入式跟踪 FIFO 寄存器

ETF RAM 大小寄存器 (ETF_RSZ)

ETF RAM size register

偏移地址：0x004

复位值：0x0000 0800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	RSZ[30:16]														
	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSZ[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31 保留，必须保持复位值。

位 30:0 **RSZ[30:0]**：以 32 位字表示的 RAM 大小 (size of the RAM in 32-bit words)
0x400：4 KB

ETF 状态寄存器 (ETF_STS)

ETF status register

偏移地址：0x00C

复位值：0x0000 001C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EMPTY	FTEMPTY	READY	TRIGD	完全
											r	r	r	r	r

位 31:5 保留，必须保持复位值。

位 4 **EMPTY**：跟踪 FIFO 为空 (trace FIFO empty)

该位仅当 ETF_CTL 寄存器的 TCEN 位为高电平时有效。当 TCEN 为低电平时，该位读为零。

0：跟踪 FIFO 含有数据

1：跟踪 FIFO 为空

注意：EMPTY 不指示 ETF 管道为空状态，该状态由 FTEMPTY 指示。

位 3 **FTEMPTY**：格式化器为空 (formatter empty)

当跟踪捕获停止时，该位置 1，所有内部管道和缓冲区都已排空。与 READY 不同，该位不受缓冲区排空影响。ACQCOMP 输出反映了该位的值。

位 2 **READY**：ETF 就绪 (ETF ready)

当跟踪捕获停止时，该位置 1，所有内部管道和缓冲区都已排空（STOPPED 或 DISABLED 模式）。

位 1 **TRIGD**：触发 (triggered)

当跟踪捕获正在进行并且 TMC 已检测到触发事件时，该位置 1。当退出禁止状态时，该位清零。

该位仅在循环缓冲区模式下运行。在其他模式下，该位始终为低电平。

该位并不表示触发已嵌入在 TMC 的格式化输出跟踪数据中。输出跟踪流上的触发指示由格式化器和刷新控制寄存器 (ETF_FFCR) 的编程确定。

位 0 **FULL**：跟踪缓冲区已满 (trace buffer full)

在循环缓冲区模式下，当 RAM 写指针回卷在缓冲区的顶部时，该标志置 1，并保持置 1，直到 ETF_CTL 寄存器的 TCEN 位清零并置 1。

在软件和硬件 FIFO 模式下，该标志表示跟踪内存中的当前空间小于或等于在 ETF_BUFWM 寄存器中编程的值，即填充级别 >= MEM_SIZE - BUFWM。

当退出禁止状态时，该位清零。FULL 输出反映了该寄存器位的值。

ETF RAM 读数据寄存器 (ETF_RRD)

ETF RAM read data register

偏移地址：0x010

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RRD[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RRD[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RRD[31:0]**：RAM 读数据 (RAM read data)

循环缓冲区模式：当处于停止状态且缓冲区非空时，读取该寄存器将从跟踪缓冲区返回下一个数据。读取所有的跟踪缓冲区后，ETF_STS 寄存器的 EMPTY 位置 1，后续读操作返回 0xFFF FFFF。在非停止状态下读取该寄存器将返回 0xFFFF FFFF。

软件 FIFO 模式：读取该寄存器将返回 FIFO 中的数据。如果在 FIFO 为空时读取该寄存器，则会返回数据 0xFFFF FFFF。

硬件 FIFO 模式：读取该寄存器返回 0xFFFF FFFF。

ETF RAM 读指针寄存器 (ETF_RRP)

ETF RAM read pointer register

偏移地址：0x014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	RRP[12:0]												
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:13 保留，必须保持复位值。

位 12:0 **RRP[12:0]**：RAM 读指针 (RAM read pointer)

RAM 读指针寄存器包含读指针的值，用于通过 ETF_RRD 寄存器和 APB 接口读取跟踪内存条目。指针可以用一个字节地址编程，64 位对齐（位 0 到 3 应为零）。每写入一个完整的 64 位 FIFO 条目，该指针就会增加 8。当该指针达到其最大值时，就会回卷。

只有在 DISABLED 状态下，才能写入该寄存器，但可在 DISABLED 状态、循环缓冲区模式和 SW FIFO 模式的 STOPPED 状态以及 SW FIFO 模式的 RUNNING 和 STOPPING 状态下读取。

ETF RAM 写指针寄存器 (ETF_RWP)

ETF RAM write pointer register

偏移地址：0x018

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	RWP[12:0]												
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:13 保留，必须保持复位值。

位 12:0 **RWP[12:0]**：RAM 写指针 (RAM write pointer)

RAM 写指针寄存器包含写指针的值，用于通过 ETF_RWD 寄存器和 APB 接口写入跟踪内存条目。指针可以用一个字节地址编程，64 位对齐（即，位 0 到 3 应为零）。每读取一个完整的 64 位 FIFO 条目，该指针就会增加 8。当该指针达到其最大值时，就会回卷。

只有在 DISABLED 状态下，才能写入该寄存器，但可在 DISABLED 状态、循环缓冲区模式和 SW FIFO 模式的 STOPPED 状态以及 SW FIFO 模式的 RUNNING 和 STOPPING 状态下读取。

ETF 触发计数器寄存器 (ETF_TRG)

ETF trigger counter register

偏移地址：0x01C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	TRG[10:0]										
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10:0 **TRG[10:0]**：触发计数器 (trigger counter)

在循环缓冲模式中，在检测到 TRIGIN 输入上的上升沿或输入跟踪流中的触发数据包 (ATID = 7'h7D) 之后，指定要在跟踪 RAM 中捕获的 32 位字的数量。一旦捕获了指定数量的数据字，就发生触发事件。触发事件对 ETF 行为的影响由 ETF_FFCR 寄存器控制。

触发后写入跟踪 RAM 的 32 位字的数量为存储在寄存器中的值加 1。当 ETF 处于软件 FIFO 模式或硬件 FIFO 模式时，该寄存器被忽略。当触发计数器开始计数时，在计数器达到零之前，所有其他触发都被忽略，不管是 TRIGIN 上的触发还是传入跟踪流中的触发。当触发计数器达到零时，它将保持为零，直到通过写入该寄存器对其重新编程为止。

当 READY 变为高电平时，该寄存器会清零，因此，跟踪捕获停止时计数器的状态不会影响后续跟踪捕获会话。在非 DISABLED 状态下写入该寄存器将引发不可预测的行为。

在 DISABLED 状态下或在循环缓冲区模式下，允许随时读取该寄存器。读访问返回触发计数器的当前值。

ETF 控制寄存器 (ETF_CTL)

ETF control register

偏移地址：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCEN
															rw

位 31:1 保留，必须保持复位值。

位 0 **TCEN**：跟踪捕获使能 (trace capture enable)

写入时：

0：禁止跟踪捕获（从 RUNNING、STOPPING 或 STOPPED 状态转为 DISABLING 或 DISABLED 状态）

1：使能跟踪捕获（从 DISABLED 状态转为 RUNNING 状态）

读取时，该位为低电平（DISABLING 或 DISABLED 状态下）或未高电平（其他状态下）。

ETF RAM 写数据寄存器 (ETF_RWD)

ETF RAM write data register

偏移地址：0x024

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RWD[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RWD[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **RWD[31:0]**：RAM 写入数据 (RAM write data)

在 DISABLED 状态下，写入该寄存器可将数据存储在 RWP 指向的位置。在非 DISABLED 状态下，忽略对该寄存器的写操作。当写入完整内存宽度（64 位）的数据时，数据被写入内存，同时 RAM 写指针递增到下一个内存字。

该寄存器用于测试目的。

ETF 模式寄存器 (ETF_MODE)

ETF mode register

偏移地址：0x028

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE[1:0]	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **MODE[1:0]**：工作模式 (operating mode)

0：循环缓冲区模式。在该模式下，跟踪内存作为循环缓冲区使用。跟踪数据被捕获到跟踪内存，该跟踪内存起始地址为写指针寄存器指向的位置。即使跟踪内存已满，传入的跟踪数据仍继续覆盖到跟踪内存，直到出现了停止条件。

1：软件 FIFO 模式。在该模式下，跟踪内存作为 FIFO 使用，当跟踪被捕获时，可通过 ETF_RRD 寄存器读取。跟踪数据被捕获到跟踪 RAM 中，当跟踪 RAM 已满时，传入的跟踪流将停止。

2：硬件 FIFO 模式。在该模式下，跟踪内存作为 FIFO 使用，通过 ATB 主接口排空。跟踪数据被捕获到跟踪 RAM 中，当跟踪 RAM 已满时，传入的跟踪流将停止。当跟踪缓冲区非空时，通过 ATB 主接口到 TPIU 来排空跟踪数据。

3：保留

ETF 锁存缓冲区填充级别寄存器 (ETF_LBUFLVL)

ETF latched buffer fill level register

偏移地址：0x02C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	LBUFLEVEL[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12 保留，必须保持复位值。

位 11:0 **LBUFLEVEL[11:0]**：锁存缓冲区填充级别 (latched buffer fill level)

读取该寄存器将以 32 位字的形式返回上一次读取该寄存器之后跟踪内存的最大填充级别。读取该寄存器还会使其内容更新为当前的填充级别。

当进入禁止状态时，该寄存器将保持其最近一次的值。当处于禁止状态时，读取该寄存器不会影响其值。当退出禁止状态时，ETF_LBUFLVL 寄存器清零。

该寄存器用于跟踪系统的性能分析。

ETF 当前缓冲区填充级别寄存器 (ETF_CBUFLVL)

ETF current buffer fill level register

偏移地址：0x030

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CBUFLEVEL[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12 保留，必须保持复位值。

位 11:0 **CBUFLEVEL[11:0]**：当前缓冲区填充级别 (current buffer fill level)
读取该寄存器将以 32 位字的形式返回跟踪内存的当前填充级别。
当 ETF_CTL.TCEN 为低电平时，该寄存器清零。

ETF 缓冲级水印寄存器 (ETF_BUFWM)

ETF buffer level watermark register

偏移地址：0x034

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	BUFWM[10:0]										
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10:0 **BUFWM[10:0]**：缓冲级水印 (buffer level watermark)
编程到该寄存器中的值以 32 位字的形式表示跟踪内存中所需的阈值空位级别。当 FIFO 的空间小于或等于该值时（即填充级别 >= MEM_SIZE - BUFWM），FULL 输出被拉高，同时 ETF_STS 寄存器中的 FULL 位置 1。
该寄存器仅在软件 FIFO 和硬件 FIFO 模式下使用。在循环缓冲区模式下，可将 ETF_RWP 编程到所需空位触发级别来实现此功能，因此，当指针回卷时，FULL 位置 1，表示空位级别已经降到所需级别以下。
可写入该寄存器的最大值为 MEM_SIZE - 1；这种情况下，当第一个 32 位字写入跟踪内存时，FULL 位输出被置为有效。
在非 DISABLED 状态下写入该寄存器会引发不可预测的行为。

ETF 格式化器和刷新状态寄存器 (ETF_FFSR)

ETF formatter and flush status register

偏移地址：0x300

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FTSTOPPED	FLINPROG
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **FTSTOPPED**：格式化器停止 (formatter stopped)

该位的工作方式与 ETF_STS 寄存器中的 FTEMPTY 位相同。

位 0 **FLINPROG**：刷新正在进行 (flush in progress)

指示 ATB 从端口上是否有正在进行的刷新。该位反映 AFVALIDS 输出的状态。刷新可由 ETF_FFCR 寄存器中的刷新控制位启动，或者由 ATB 主端口请求。

0：无正在进行的刷新

1：正在进行刷新

ETF 格式化器和刷新控制寄存器 (ETF_FFCR)

ETF formatter and flush control register

偏移地址：0x304

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DRAINBUF	STPONTRGEV	STOPONFL	Res.	TRIGONFL	TRGONTRGEV	TRGONTRGIN	Res.	FLUSHMAN	FONTRGEV	FONFLIN	Res.	Res.	ENTI	ENFT
	rw	rw	rw		rw	rw	rw		rw	rw	rw			rw	rw

位 31:15 保留，必须保持复位值。

位 14 **DRAINBUF**：排空缓冲区 (drain buffer)

该位用于使能在格式化器停止后通过 ATB 主接口排空跟踪数据。这在循环缓冲区模式下，可用于先将跟踪数据捕获到跟踪内存，再通过 ATB 主接口排空捕获的跟踪数据。

在停止状态下，向该位写入 1 可启动通过 ATB 主接口排空跟踪缓冲区的内容。该位始终读为零。当排空正在进行时，ETF_STS 寄存器的 READY 位变为低电平。

该位仅当 ETF 处于循环缓冲区模式且使能了格式化（即 ETF_FFCR 寄存器的 ENFT 位置 1）时有效。如果 ETF 处于任何其他模式或处于非停止状态，将该位置 1 会引发不可预测的行为。

当在循环缓冲模式下跟踪捕获完成时，必须通过相同的机制从跟踪内存检索所有捕获的跟踪数据，要么通过 ETF_RRD 读操作读取所有跟踪数据，要么通过将 DRAINBUF 位置 1 来排空所有跟踪数据。在通过 ETF_RRD 读取一些捕获的跟踪数据后将 DRAINBUF 位置 1 会引发不可预测的行为。

位 13 **STPONTRGEV**：通过触发事件的停止指示 (stop on trigger event)

0：不起作用

1：当发生触发事件时，停止跟踪捕获

注意：在软件 FIFO 模式或硬件 FIFO 模式且该位置 1 的情况下，使能 ETF 会引发不可预测的行为。

位 12 **STOPONFL**：通过刷新的停止指示 (stop on flush)

0：不起作用

1：当完成刷新时，停止跟踪捕获

如果刷新是由 ATB 主接口启动，则完成刷新时不会导致格式化器停止，无论在该位中编程的值是多少。

位 11 保留，必须保持复位值。

位 10 **TRIGONFL**：通过刷新的触发指示 (trigger on flush)

0：不起作用

1：指示刷新完成时跟踪流存在触发

如果 ENFT 和 ENTI 都清零，则忽略该位且跟踪流中未插入触发。

如果刷新是由 ATB 主接口启动，则完成刷新时不会导致触发指示，无论在该位中编程的值是多少。

位 9 **TRGONTRGEV**：通过触发事件的触发指示 (trigger on trigger event)

0：不起作用

1：指示发生触发事件时跟踪流中存在触发

如果 ENFT 和 ENTI 都清零，则忽略该位且跟踪流中未插入触发。

该位不能在软件 FIFO 模式或硬件 FIFO 模式下使用。

位 8 **TRGONTRGIN**：通过触发输入的触发指示 (trigger on trigger in)

0：不起作用

1：指示在 TRIGIN 输入上检测到上升沿时跟踪流中存在触发

如果 ENFT 和 ENTI 都清零，则忽略该位且跟踪流中未插入触发。

位 7 保留，必须保持复位值。

位 6 **FLUSHMAN**：手动刷新 (manual flush)

0：不起作用

1：刷新跟踪 FIFO 和管道

当刷新完成时，该位自动清零。如果 ETF_CTL 寄存器的 TCEN 位为 0，则忽略对该位的写操作。

位 5 **FONTRGEV**：通过触发事件的刷新指示 (flush on trigger event)

0：不起作用

1：如果发生触发事件，则刷新跟踪 FIFO 和管道

该位不能在软件 FIFO 模式或硬件 FIFO 模式下使用。如果 STPONTRGEV 置 1，则忽略该位。

位 4 **FONFLIN**：通过刷新输入的刷新指示 (flush on flush in)

0：不起作用

1：如果在 FLUSHIN 输入上检测到上升沿，则刷新跟踪 FIFO 和管道

位 3:2 保留，必须保持复位值。

位 1 **ENTI**：使能触发插入 (enable trigger insertion)

将该位置 1 使能在格式化的跟踪流中插入触发。如果在跟踪流中插入了一个字节的数据 8'h00，且 ATID = 7'h7D，则指示存在触发。另外，跟踪流上的触发指示还受 ETF_FFCR 寄存器中的寄存器位 TRIGONFL、TRGONTRGEV 和 TRGONTRGIN 控制。只有当 READY 为高电平且 TCEN 为低电平时，该位才能更改。该位仅在该寄存器中的 ENFT 寄存器位置 1 时才有效。如果在 ENFT 为低电平时将 ENTI 位设置为高电平，则会导致使能格式化。

位 0 **ENFT**：使能格式化 (enable formatting)

0：禁止格式化。假设传入的跟踪数据源于单个跟踪源

1：使能格式化

当使能了跟踪捕获且禁止了格式化器时，如果 ETF 接收了多个 ATID，则会引起跟踪数据的交错。仅在循环缓冲模式下才支持禁止格式化。如果在非循环缓冲区模式且 ENFT 为低电平的情况下使能 ETF，则会导致使能格式化。如果在 ENFT 为低电平时将 ENTI 位设置为高电平，则会导致使能格式化。

当处于禁止状态时，忽略该位。

ETF 周期性同步计数寄存器 (ETF_PSCR)

ETF formatter and flush status register

偏移地址：0x308

复位值：0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSCOUNT[4:0]				
											rw	rw	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4:0 **PSCOUNT[4:0]**：同步计数器重载值 (synchronization counter reload value)

PSCOUNT 确定同步计数器的重载值。重载值在下次计数器达到零时生效。读取该寄存器会返回编程到该寄存器中的重载值。一旦复位，该寄存器会被置为 0xA，对应于 1024 个字节的同步周期。

0x0：禁止同步

0x1-0x6：保留

0x7-0x1B：同步周期为 $2^{PSCOUNT}$ 个字节

其他值：保留

ETF 声明标记设置寄存器 (ETF_CLAIMSET)

ETF claim tag set register

偏移地址：0xFA0

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMSET[3:0]**：设置声明标记位 (sets claim tag bits)

写：

- 0000：无影响
- xxx1：将位 0 置 1
- xx1x：将位 1 置 1
- x1xx：将位 2 置 1
- 1xxx：将位 3 置 1

读：

- 0xF：表示声明标记使用四个位

ETF 声明标记清零寄存器 (ETF_CLAIMCLR)

ETF claim tag clear register

偏移地址：0xFA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMCLR[3:0]**：复位声明标记位 (resets claim tag bits)

写：

- 0000：无影响
- xxx1：将位 0 清零
- xx1x：将位 1 清零
- x1xx：将位 2 清零
- 1xxx：将位 3 清零

读：

- 返回声明标记的当前值

ETF 锁定访问寄存器 (ETF_LAR)

ETF lock access register

偏移地址：0xFB0

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ACCESS_W[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS_W[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **ACCESS_W[31:0]**：ETF 寄存器写访问使能 (CTI register write access enable)
使能处理器内核写访问某些 ETF 寄存器（调试器无需解锁组件）。
0xC5ACCE55：使能写访问
其他值：禁止写访问

ETF 锁定状态寄存器 (ETF_LSR)

ETF lock status register

偏移地址：0xFB4

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCKTYPE	LOCKGRANT	LOCKEXIST
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **LOCKTYPE**：ETF_LAR 寄存器大小 (ETF_LAR register size)
指示 ETF_LAR 寄存器的大小
0：32 位

位 1 **LOCKGRANT**：当前锁定状态 (current lock status)
当外部调试器读取该位时，始终读为零。
0：允许写访问
1：阻止写访问。只允许读操作。

位 0 **LOCKEXIST**：锁定控制存在 (lock control existence)
表示是否存在锁定控制机制。当外部调试器读取该位时，始终读为零。
0：不存在锁定控制机制
1：存在锁定控制机制

ETF 认证状态寄存器 (ETF_AUTHSTAT)

ETF authentication status register

偏移地址：0xFB8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:6 **SNID[1:0]**：安全非侵入式调试的安全等级 (security level for secure non-invasive debug)
0x0：未设计

位 5:4 SID**[1:0]**：安全侵入式调试的安全等级 (security level for secure invasive debug)
0x0：未设计

位 3:2 **NSNID[1:0]**：非安全非侵入式调试的安全等级 (security level for non-secure non-invasive debug)
0x0：未设计

位 1:0 **NSID[1:0]**：非安全侵入式调试的安全等级 (security level for non-secure invasive debug)
0x0：未设计

ETF 设备配置寄存器 (ETF_DEVID)

ETF device configuration register

偏移地址：0xFC8

复位值：0x0000 0380

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	MEMWIDTH[2:0]			CONFIGTYP[1:0]		CLKSCHEM	ATBINPORTCNT[4:0]				
					r	r	r	r	r	r	r	r	r	r	r

位 31:11 保留，必须保持复位值。

位 10:8 **MEMWIDTH[2:0]**：内存接口数据总线宽度 (memory interface data bus width)
0x3：64 位（对应于 32 位 ATB 数据）

位 7:6 **CONFIGTYP[1:0]**：组件（ETB、ETR 或 ETF）的配置类型 (configuration type of component (ETB, ETR or ETF))
0x2：ETF

位 5 **CLKSCHEM**：RAM 时钟方案（同步或异步）(RAM clocking scheme (synchronous or asynchronous))
0x0：同步

位 4:0 **ATBINPORTCNT[4:0]**：ATB 输入端口多路复用的数量/类型 (number/type of ATB input port multiplexing)
0x0：无

ETF 设备类型标识寄存器 (ETF_DEVTYPE)

ETF device type identifier register

偏移地址：0xFCC

复位值：0x0000 0032

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SUBTYPE[3:0]**：子分类 (sub-classification)

0x3：将跟踪数据从 ATB 从接口捕获到可通过 ATB 主接口排空的 RAM 中

位 3:0 **MAJORTYPE[3:0]**：主分类 (major classification)

0x2：组件为一个跟踪链路，因为它具有一个 ATB 主接口，在硬件 FIFO 模式下可通过该接口排空跟踪数据

ETF CoreSight 外设标识寄存器 4 (ETF_PIDR4)

ETF CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SIZE[3:0]**：寄存器文件大小 (register file size)

0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)

0x4：Arm® JEDEC 代码

ETF CoreSight 外设标识寄存器 0 (ETF_PIDR0)

ETF CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0061

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位 [7:0] (part number bits [7:0])
0x61：ETF 产品编号 (ETF part number)

ETF CoreSight 外设标识寄存器 1 (ETF_PIDR1)

ETF CoreSight Peripheral Identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位 [3:0] (JEP106 identity code bits [3:0])
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位 [11:8] (part number bits [11:8])
0x9：ETF 产品编号 (ETF part number)

ETF CoreSight 外设标识寄存器 2 (ETF_PIDR2)

ETF CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 001F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (component revision number)
0x1：r0p1

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
0x1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位 [6:4] (JEP106 identity code bits [6:4])
0x3：Arm® JEDEC 代码

ETF CoreSight 外设标识寄存器 3 (ETF_PIDR3)

ETF CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (customer modified)
0x0：无客户修改

ETF CoreSight 组件标识寄存器 0 (ETF_CIDR0)

ETF CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位 [7:0] (component ID bits [7:0])
0x0D：公共 ID 值

ETF CoreSight 外设标识寄存器 1 (ETF_CIDR1)

ETF CoreSight peripheral identity register 1

偏移地址：0xFF4

复位值：0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位 [15:12] (component ID bits [15:12])——组件类
0x9：CoreSight™ 组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位 [11:8] (component ID bits [11:8])
0x0：公共 ID 值

ETF CoreSight 组件标识寄存器 2 (ETF_CIDR2)

ETF CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位 [23:16] (component ID bits [23:16])
0x05：公共 ID 值

ETF CoreSight 组件标识寄存器 3 (ETF_CIDR3)

ETF CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位 [31:24] (component ID bits [31:24])
0xB1：公共 ID 值

嵌入式跟踪 FIFO 寄存器映射

表 613. ETF 寄存器映射和复位值

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x004	**ETF_RSZ**	Res.	RSZ[30:0]																														
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0x00C	**ETF_STS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EMPTY	FTEMPTY	READY	TRIGD	FULL
	Reset value																												1	1	1	0	0
0x010	**ETF_RRD**	RRD[31:0]																															
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0x014	**ETF_RRP**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RRP[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	**ETF_RWP**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWP[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x01C	**ETF_TRG**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRG[10:0]										
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x020	**ETF_CTL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCEN
	Reset value																																0
0x024	**ETF_RWD**	RWD[31:0]																															
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0x028	**ETF_MODE**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE[1:0]	
	Reset value																															0	0
0x02C	**ETF_LBUFLVL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LBUFLEVEL[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x030	**ETF_CBUFLVL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CBUFLEVEL[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x034	**ETF_BUFWM**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUFWM[10:0]										
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x300	**ETF_FFSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FTSTOPPED	FLINPROG
	Reset value																															1	0

表 613. ETF 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x304	**ETF_FFCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DRAINBUF	STPONTRGEV	STOPONFL	Res.	TRGONFL	TRGONTRGEV	TRGONTRGIN	Res.	FLUSHMAN	FONTRGEV	FONFLIN	Res.	Res.	ENTI	ENFT
	Reset value																		0	0	0		0	0	0		0	0	0			0	0
0x308	**ETF_PSCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSCOUNT[3:0]				
	Reset value																												0	1	0	1	0
0xFA0	**ETF_CLAIMSET**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
	Reset value																													1	1	1	1
0xFA4	**ETF_CLAIMCLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
	Reset value																													0	0	0	0
0xFB0	**ETF_LAR**	ACCESS_W[31:0]																															
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0xFB4	**ETF_LSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCKTYPE	LOCKGRANT	LOCKEXIST
	Reset value																														0	1	1
0xFB8	**ETF_AUTHSTAT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0xFC8	**ETF_DEVID**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEMWDTH[2:0]			CONFIGTYP[1:0]		CLKSCHEM	ATBINPORTCNT[4:0]				
	Reset value																						0	1	1	1	0	0	0	0	0	0	0
0xFD0	**ETF_DEVTYPE**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
	Reset value																									0	0	1	1	0	0	1	0

表 613. ETF 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFD0	**ETF_PIDR4**	Res.																								SIZE[3:0]				JEP106CON[3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	**ETF_PIDR5**	Res.																								Res.				Res.			
	Reset value																																
0xFD8	**ETF_PIDR6**	Res.																								Res.				Res.			
	Reset value																																
0xFDC	**ETF_PIDR7**	Res.																								Res.				Res.			
	Reset value																																
0xFE0	**ETF_PIDR0**	Res.																								PARTNUM[7:0]							
	Reset value																									0	1	1	0	0	0	0	1
0xFE4	**ETF_PIDR1**	Res.																								JEP106ID[3:0]				PARTNUM[11:8]			
	Reset value																									1	0	1	1	1	0	0	1
0xFE8	**ETF_PIDR2**	Res.																								REVISION[3:0]				JEDEC	JEP106ID[6:4]		
	Reset value																									0	0	0	1	1	0	1	1
0xFEC	**ETF_PIDR3**	Res.																								REVAND [3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	**ETF_CIDR0**	Res.																								PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	**ETF_CIDR1**	Res.																								CLASS[3:0]				PREAMBLE[11:8]			
	Reset value																									1	0	0	1	0	0	0	0
0xFF8	**ETF_CIDR2**	Res.																								PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	**ETF_CIDR3**	Res.																								PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

59.10.5 跟踪端口接口单元 (TPIU)

TPIU 是一款 CoreSight™ 组件，用来对跟踪流进行格式化处理并将其输出到外部跟踪端口信号上。TPIU 具有单个 ATB 从端口，用于传入跟踪数据。跟踪端口为同步并行端口，包括一个时钟输出 TRACECLK 和十六个数据输出 TRACED(15:0)。跟踪端口宽度的可编程范围为 1 到 16。使用较小的端口宽度可减少所需测试点/连接器引脚的数量，可将腾出的 I/O 用于其他用途。然而，这限制了跟踪端口的带宽，从而限制了可以实时输出的跟踪信息的数量。

有关跟踪端口接口 CoreSight™ 组件的更多信息，请参见《Arm® CoreSight™ SoC-400 技术参考手册》*[2]*。

跟踪端口接口单元寄存器

TPIU 支持的端口大小寄存器 (TPIU_SUPPSIZE)

TPIU supported port size register

偏移地址：0x000

复位值：0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PORTSIZE[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PORTSIZE[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **PORTSIZE[31:0]**：支持的跟踪端口大小 (supported trace port sizes)
指示支持的跟踪端口大小，从 1 到 32 个引脚。位 n-1 置 1 指示支持端口大小 n。
0x0000 FFFF：支持端口大小 1 到 16

TPIU 当前端口大小寄存器 (TPIU_CURPSIZE)

TPIU current port size register

偏移地址：0x004

复位值：0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PORTSIZE[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PORTSIZE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **PORTSIZE[31:0]**：当前跟踪端口大小 (current trace port size)
指示当前跟踪端口大小。位 n-1 置 1 指示当前端口大小为 n 个引脚。n 的值必须在支持的端口大小范围 (1-4) 内。只能将一个位置 1，否则可能导致不可预测的行为。只有当格式化器停止时，才应修改该寄存器。

TPIU 支持的触发模式寄存器 (TPIU_SUPTRGM)

TPIU supported trigger modes register

偏移地址：0x100

复位值：0x0000 011F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRGRUN	TRIGD
														r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCOUNT8	Res.	Res.	Res.	MULT64K	MULT256	MULT16	MULT4	MULT2
							r				r	r	r	r	r

位 31:18 保留，必须保持复位值。

位 17 **TRGRUN**：触发运行中 (Trigger running)
0：未发生触发或计数器为 0
1：已发生触发且计数器不为 0

位 16 **TRIGD**：触发 (triggered)
0：未发生触发
1：已发生触发且计数器已达到 0

位 15:9 保留，必须保持复位值。

位 8 **TCOUNT8**：8 位计数器寄存器 (8-bit counter register)
1：已设计

位 7:5 保留，必须保持复位值。

位 4 **MULT64K**：支持将触发计数器乘以 65536 (multiplying the trigger counter by 65536 support)
1：支持

位 3 **MULT256**：支持将触发计数器乘以 256 (multiplying the trigger counter by 256 support)
1：支持

位 2 **MULT16**：支持将触发计数器乘以 16 (multiplying the trigger counter by 16 support)
1：支持

位 1 **MULT4**：支持将触发计数器乘以 4 (multiplying the trigger counter by 4 support)
1：支持

位 0 **MULT2**：支持将触发计数器乘以 2 (multiplying the trigger counter by 2 support)
1：支持

TPIU 触发计数器值寄存器 (TPIU_TRGCNT)

TPIU trigger counter value register

偏移地址：0x104

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGCOUNT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **TRIGCOUNT[7:0]**：使能触发延迟到外部 (trigger delay to external enable)

使能将触发指示延迟到任意外部连接的跟踪捕获或存储设备。

该计数器只有八位宽，并且仅与 TPIU_TRGMULT 触发乘数寄存器中的计数器乘数一起使用。当触发启动时，该值（连同乘数一起）为指示触发之前的字数。当触发计数器达到 0 时，在此处写入的值将被重新加载。写入该寄存器会使触发计数器值复位，但不会复位乘数上的任何值。读取该寄存器会返回预设值，而不是当前的计数。

TPIU 触发乘数寄存器 (TPIU_TRGMULT)

TPIU trigger multiplier register

偏移地址：0x108

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MULT64K	MULT256	MULT16	MULT4	MULT2
											rw	rw	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4 **MULT64K**：将触发计数器乘以 65536 (multiply the trigger counter by 65536)

0：禁止

1：使能

位 3 **MULT256**：将触发计数器乘以 256 (multiply the trigger counter by 256)

0：禁止

1：使能

位 2 **MULT16**：将触发计数器乘以 16 (multiply the trigger counter by 16)

0：禁止

1：使能

位 1 **MULT4**：将触发计数器乘以 4 (multiply the trigger counter by 4)

0：禁止

1：使能

位 0 **MULT2**：将触发计数器乘以 2 (multiply the trigger counter by 2)

0：禁止

1：使能

TPIU 支持的测试模式寄存器 (TPIU_SUPTPM)

TPIU supported test patterns/modes register

偏移地址：0x200

复位值：0x0003 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCONTEN	PTIMEEN
														r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PATF0	PATA5	PATW0	PATW1
												r	r	r	r

位 31:18 保留，必须保持复位值。

位 17 **PCONTEN**：连续模式支持 (continuous mode support)
指示是否支持连续模式。
1：支持

位 16 **PTIMEEN**：计时模式支持 (timed mode support)
指示是否支持计时模式。
1：支持

位 15:4 保留，必须保持复位值。

位 3 **PATF0**：FF/00 模式支持 (FF/00 pattern support)
指示是否支持 FF/00 模式的跟踪端口输出。
1：支持

位 2 **PATA5**：AA/55 模式支持 (AA/55 pattern support)
指示是否支持 AA/55 模式的跟踪端口输出。
1：支持

位 1 **PATW0**：走 0 模式支持 (walking 0's pattern support)
指示是否支持走 0 模式的跟踪端口输出。
1：支持

位 0 **PATW1**：走 1 模式支持 (walking 1's pattern support)
指示是否支持走 1 模式的跟踪端口输出。
1：支持

TPIU 当前测试模式寄存器 (TPIU_CURTPM)

TPIU current test pattern/mode register

偏移地址：0x204

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCONTEN	PTIMEEN
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PATF0	PATA5	PATW0	PATW1
												rw	rw	rw	rw

位 31:18 保留，必须保持复位值。

位 17 **PCONTEN**：连续模式使能 (continuous mode enable)
指示是否使能连续模式。
0：禁止
1：使能

位 16 **PTIMEEN**：计时模式使能 (timed mode enable)
指示是否使能计时模式。
0：禁止
1：使能

位 15:4 保留，必须保持复位值。

位 3 **PATF0**：FF/00 模式使能 (FF/00 pattern enable)
指示是否使能 FF/00 模式的跟踪端口输出。
0：禁止
1：使能

位 2 **PATA5**：AA/55 模式使能 (AA/55 pattern enable)
指示是否使能 AA/55 模式的跟踪端口输出。
0：禁止
1：使能

位 1 **PATW0**：走 0 模式使能 (walking 0’s pattern enable)
指示是否使能走 0 模式的跟踪端口输出。
0：禁止
1：使能

位 0 **PATW1**：走 1 模式使能 (walking 1’s pattern enable)
指示是否使能走 1 模式的跟踪端口输出。
0：禁止
1：使能

TPIU 测试模式重复计数器寄存器 (TPIU_TPRCR)

TPIU test pattern repeat counter register

偏移地址：0x208

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PATTCOUNT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **PATTCOUNT[7:0]**：模式 TRACECLKIN 周期数 (pattern number of TRACECLKIN cycles)
8 位计数器值，用于指示某个模式在切换到下一个模式之前运行的 TRACECLKIN 周期数。

TPIU 格式化器和刷新状态寄存器 (TPIU_FFSR)

TPIU formatter and flush status register

偏移地址：0x300

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCPRESENT	FTSTOPPED	FLINPROG
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **TCPRESENT**：TRACECTL 输出引脚可用性 (TRACECTL output pin availability)
指示可选的 TRACECTL 输出引脚是否可用。
0：该设备中不存在 TRACECTL 引脚

位 1 **FTSTOPPED**：格式化器停止 (formatter stopped)
格式化器收到了停止请求信号和所有跟踪数据，并发送了后置码。ATB 接口上的任何其他跟踪数据都被忽略。
0：格式化器未停止
1：格式化器已停止

位 0 **FLINPROG**：刷新正在进行 (flush in progress)
指示 ATB 从端口上是否有正在进行的刷新。该位反映 AFVALIDS 输出的状态。刷新可由 TPIU_FFCR 寄存器中的刷新控制位启动。
0：无正在进行的刷新
1：正在进行刷新

TPIU 格式化器和刷新控制寄存器 (TPIU_FFCR)

TPIU formatter and flush control register

偏移地址：0x304

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	STOPTRIG	STOPFL	Res.	TRIGFL	TRIGEVT	TRIGIN	Res.	FONMAN	FONTRIG	FONFLIN	Res.	Res.	ENFCONT	ENFTC
		rw	rw		rw	rw	rw		rw	rw	rw			rw	rw

位 31:14 保留，必须保持复位值。

位 13 **STOPTRIG**：通过触发事件的停止指示 (stop on trigger event)

0：不起作用

1：当触发事件发生时，停止格式化器

位 12 **STOPFL**：通过刷新的停止指示 (stop on flush)

0：不起作用

1：当完成刷新时，停止格式化器

位 11 保留，必须保持复位值。

位 10 **TRIGFL**：通过刷新的触发指示 (trigger on flush)

0：不起作用

1：指示刷新完成时跟踪流存在触发

位 9 **TRIGEVT**：通过触发事件的触发指示 (trigger on trigger event)

0：不起作用

1：指示发生触发事件时跟踪流中存在触发

位 8 **TRIGIN**：通过触发输入的触发指示 (trigger on trigger in)

0：不起作用

1：指示当系统 CTI 的 TRIGIN 输入被置为有效时，跟踪流中存在触发。

位 7 保留，必须保持复位值。

位 6 **FONMAN**：生成手动刷新 (generates a manual flush)

0：不起作用

1：刷新跟踪

当刷新完成时，该位自动清零。

位 5 **FONTRIG**：通过触发事件的刷新指示 (flush on trigger event)

当触发计数器达到 0 时，或者触发计数器为 0，且系统 CTI 的 TRIGIN 输入为高电平时，触发事件会发生。

0：不起作用

1：如果触发事件发生，则刷新跟踪

位 4 **FONFLIN**：通过刷新输入的刷新指示 (flush on flush in)

0：不起作用

1：当系统 CTI 的 FLUSHIN 输入被置为有效时，刷新跟踪

位 3:2 保留，必须保持复位值。

位 1 **ENFCONT**：使能连续格式化 (enable continuous formatting)

0：禁止连续格式化

1：使能连续格式化

位 0 **ENFTC**：使能在格式化的跟踪数据中嵌入触发 (enables the embedding of triggers in formatted trace)

0：禁止格式化

1：使能格式化

TPIU 格式化器同步计数器寄存器 (TPIU_FSCR)

TPIU formatter synchronisation counter register

偏移地址：0x308

复位值：0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CYCCOUNT[4:0]				
											rw	rw	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4:0 **CYCCOUNT[4:0]**：格式化帧数 (formatted frame number)

使能有效使用不同大小的 TPA，而不会浪费捕获设备的大量存储容量。

该计数器包含自最上一个 128 位同步数据包以来的格式化帧数。它是一个 12 位计数器，最大计数器值为 4096。相当于每 65536 个字节（即 4096 个数据包 x 16 字节/数据包）同步一次。默认设置为每 1024 个字节（即每 64 个格式化帧）传输一个同步数据包。如果格式化器配置为连续模式，则在正常运行情况下插入全字和半字同步帧。在这些情况下，计数器值为完全同步数据包间的最大完整帧数。

TPIU 声明标记设置寄存器 (TPIU_CLAIMSET)

TPIU claim tag set register

偏移地址：0xFA0

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMSET[3:0]**：设置声明标记位 (sets claim tag bits)

写：

- 0000：无影响
- xxx1：将位 0 置 1
- xx1x：将位 1 置 1
- x1xx：将位 2 置 1
- 1xxx：将位 3 置 1

读：

- 0xF：表示声明标记使用四个位

TPIU 声明标记清零寄存器 (TPIU_CLAIMCLR)

TPIU claim tag clear register

偏移地址：0xFA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMCLR[3:0]**：复位声明标记位 (resets claim tag bits)

写：

- 0000：无影响
- xxx1：将位 0 清零
- xx1x：将位 1 清零
- x1xx：将位 2 清零
- 1xxx：将位 3 清零

读：

- 返回声明标记的当前值

TPIU 锁定访问寄存器 (TPIU_LAR)

TPIU lock access register

偏移地址：0xFB0

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ACCESS_W[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS_W[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **ACCESS_W[31:0]**：TPIU 寄存器写访问使能 (TPIU register write access enable)
使能处理器内核写访问某些 TPIU 寄存器（调试器无需解锁组件）。
0xC5ACCE55：使能写访问
其他值：禁止写访问

TPIU 锁定状态寄存器 (TPIU_LSR)

TPIU lock status register

偏移地址：0xFB4

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCKTYPE	LOCKGRANT	LOCKEXIST
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **LOCKTYPE**：TPIU_LAR 寄存器大小 (TPIU_LAR register size)
指示 TPIU_LAR 寄存器的大小。
0：32 位

位 1 **LOCKGRANT**：当前锁定状态 (current lock status)
当外部调试器读取该位时，始终读为零。
0：允许写访问
1：阻止写访问。只允许读操作。

位 0 **LOCKEXIST**：锁定控制存在 (lock control existence)
表示是否存在锁定控制机制。当外部调试器读取该位时，始终读为零。
0：不存在锁定控制机制
1：存在锁定控制机制

TPIU 认证状态寄存器 (TPIU_AUTHSTAT)

TPIU authentication status register

偏移地址：0xFB8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:6 **SNID[1:0]**：安全非侵入式调试的安全等级 (security level for secure non-invasive debug)
0x0：未设计

位 5:4 SID**[1:0]**：安全侵入式调试的安全等级 (security level for secure invasive debug)
0x0：未设计

位 3:2 **NSNID[1:0]**：非安全非侵入式调试的安全等级 (security level for non-secure non-invasive debug)
0x0：未设计

位 1:0 **NSID[1:0]**：非安全侵入式调试的安全等级 (security level for non-secure invasive debug)
0x0：未设计

TPIU 设备配置寄存器 (TPIU_DEVID)

TPIU device configuration register

偏移地址：0xFC8

复位值：0x0000 00A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SWOUARTNRZ	SWOMAN	TCLKDATA	FIFOSIZE[2:0]			CLKRELAT	MAXNUM[4:0]				
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12 保留，必须保持复位值。

位 11 **SWOUARTNRZ**：串行线输出，UART 或 NRZ 支持 (Serial Wire Output, UART or NRZ support)
指示是否支持串行线输出 UART 或 NRZ。
0x0：不支持

位 10 **SWOMAN**：串行线输出，曼彻斯特支持 (Serial-wire output, Manchester support)
表示是否支持曼彻斯特编码格式的串行线输出。
0x0：不支持

位 9 **TCLKDATA**：支持跟踪时钟加数据 (trace clock plus data support)
指示是否支持跟踪时钟加数据。
0x0：不支持

位 8:6 **FIFOSIZE[2:0]**：以 2 的幂表示的 FIFO 大小 (FIFO size in powers of 2)
0x2：FIFO 大小 = 4（16 字节）

位 5 **CLKRELAT**：ATB 时钟和 TRACECLKIN 同步 (ATB clock and TRACECLKIN sync)
指示 ATB 时钟和 TRACECLKIN 之间的关系（同步或异步）。
0x1：异步

位 4:0 **MAXNUM[4:0]**：ATB 输入端口多路复用的数量/类型 (number / type of ATB input port multiplexing)
0x0：无

TPIU 设备类型标识符寄存器 (TPIU_DEVTYPE)

TPIU device type identifier register

偏移地址：0xFCC

复位值：0x0000 0032

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SUBTYPE[3:0]**：子分类 (sub-classification)
0x1：跟踪端口组件

位 3:0 **MAJORTYPE[3:0]**：主分类 (major classification)
0x1：跟踪 Sink 组件

TPIU CoreSight 外设标识寄存器 4 (TPIU_PIDR4)

TPIU CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SIZE[3:0]**：寄存器文件大小 (register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：Arm® JEDEC 代码

TPIU CoreSight 外设标识寄存器 0 (TPIU_PIDR0)

TPIU CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0012

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位 [7:0] (part number bits [7:0])
0x12：TPIU 产品编号

TPIU CoreSight 外设标识寄存器 1 (TPIU_PIDR1)

TPIU CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位 [3:0] (JEP106 identity code bits [3:0])
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位 [11:8] (part number bits [11:8])
0x9：TPIU 产品编号

TPIU CoreSight 外设标识寄存器 2 (TPIU_PIDR2)

TPIU CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 004B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (component revision number)
0x4：r0p5

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
0x1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位 [6:4] (JEP106 identity code bits [6:4])
0x3：Arm® JEDEC 代码

TPIU CoreSight 外设标识寄存器 3 (TPIU_PIDR3)

TPIU CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (customer modified)
0x0：无客户修改

TPIU CoreSight 组件标识寄存器 0 (TPIU_CIDR0)

TPIU CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位 [7:0] (component ID bits [7:0])
0x0D：公共 ID 值

TPIU CoreSight 外设标识寄存器 1 (TPIU_CIDR1)

TPIU CoreSight peripheral identity register 1

偏移地址：0xFF4

复位值：0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位 [15:12] (component ID bits [15:12])——组件类
0x9：CoreSight™ 组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位 [11:8] (component ID bits [11:8])
0x0：公共 ID 值

TPIU CoreSight 组件标识寄存器 2 (TPIU_CIDR2)

TPIU CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位 [23:16] (component ID bits [23:16])
0x05：公共 ID 值

TPIU CoreSight 组件标识寄存器 3 (TPIU_CIDR3)

TPIU CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位 [31:24] (component ID bits [31:24])
0xB1：公共 ID 值

跟踪端口接口单元寄存器映射

表 614. TPIU 寄存器映射和复位值

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**TPIU_SUPPSIZE**	PORTSIZE[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x004	**TPIU_CURPSIZE**	PORTSIZE[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x100	**TPIU_SUPTRGM**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRGRUN	TRIGD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCOUNT8	Res.	Res.	Res.	MULT64K	MULT256	MULT16	MULT4	MULT2
	Reset value															0	0								1				1	1	1	1	1
0x104	**TPIU_TRGCNT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGCOUNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x108	**TPIU_TRGMULT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MULT64K	MULT256	MULT16	MULT4	MULT2
	Reset value																												0	0	0	0	0
0x200	**TPIU_SUPTPM**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCONTEN	PTIMEEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PATF0	PATA5	PATW0	PATW1
	Reset value															1	1													1	1	1	1
0x204	**TPIU_CURTPM**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCONTEN	PTIMEEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PATF0	PATA5	PATW0	PATW1
	Reset value															0	0													0	0	0	0
0x104	**TPIU_TPRCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PATTCOUNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x300	**TPIU_FFSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCPRESENT	FTSTOPPED	FLINPROG
	Reset value																														0	1	0

表 614. TPIU 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x304	**TPIU_FFCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STOPTRIG	STOPFL	Res.	TRIGFL	TRIGEVt	TRIGIN	Res.	FONMAN	FONTRIG	FONFLIN	Res.	Res.	ENFCONT	ENFTC
	Reset value																			0	0		0	0	0		0	0	0			0	0
0x308	**TPIU_FSCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CYCCOUNT[11:0]											
	Reset value																					0	0	0	0	0	1	0	0	0	0	0	0
0xFA0	**TPIU_CLAIMSET**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
	Reset value																													1	1	1	1
0xFA4	**TPIU_CLAIMCLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
	Reset value																													0	0	0	0
0xFB0	**TPIU_LAR**	ACCESS_W[31:0]																															
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0xFB4	**TPIU_LSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCKTYPE	LOCKGRANT	LOCKEXIST	
	Reset value																													0	1	1	
0xFB8	**TPIU_AUTHSTAT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0xFC8	**TPIU_DEVID**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWOUARTNRZ	SWOMAN	TCLKDATA	FIFOSIZE[2:0]			CLKRELAT	MUXNUM[4:0]				
	Reset value																					0	0	0	0	1	0	1	0	0	0	0	0
0xFD0	**TPIU_DEVTYPE**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
	Reset value																									0	0	0	1	0	0	0	1

表 614. TPIU 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFD0	**TPIU_PIDR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON[3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	**TPIU_PIDR5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	**TPIU_PIDR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	**TPIU_PIDR7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	**TPIU_PIDR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	0	0	1	0	0	1	0
0xFE4	**TPIU_PIDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
	Reset value																									1	0	1	1	1	0	0	1
0xFE8	**TPIU_PIDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
	Reset value																									0	1	0	0	1	0	1	1
0xFEC	**TPIU_PIDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND [3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	**TPIU_CIDR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	**TPIU_CIDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
	Reset value																									1	0	0	1	0	0	0	0
0xFF8	**TPIU_CIDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	**TPIU_CIDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

59.10.6 微处理器调试单元 (DBGMCU)

DBGMCU 组件包含多个寄存器，用来在调试模式下控制电源和时钟行为。具体地说，它允许调试器或调试软件：

- 在低功耗模式（睡眠、停止或待机）下，保持处理器的时钟和电源
- 在低功耗模式下，保持系统调试和跟踪组件的时钟和电源
- 当处理器在调试模式下停止时，停止某些外设（CAN、SMBUS 超时、看门狗、定时器、RTC）的时钟

调试模式下的低功耗模式仿真

当进入低功耗模式（CSleep、Stop、LP-Stop、LPLV-Stop、LPLV-Stop2 或待机）时，处理器的时钟、子系统时钟或待机模式下的内核电源可能会被关闭。这样可以阻止对无时钟的域进行任何调试访问。为了简化对使用这些节能特性的应用程序的调试，可以根据 DBGMCU_CR 寄存器中的 DBGLP 位设置来决定是否保留时钟和电源。该位可直接通过调试器或调试软件进行访问，这意味着无需修改应用软件。

如果系统进入其中一种节能模式，并且相应的寄存器位置 1，则系统会照常进入和退出低功耗状态。但是时钟和电源保持工作状态，允许调试器进行完全访问。

注： *使用跟踪端口接口 (TPIU) 时，如果系统进入待机模式并在 TPIU FIFO 为空之前关闭内核电源，则跟踪数据可能会丢失。为此，可以通过软件检查 FIFO 的状态来加以避免。但是，如果在使用跟踪端口时 DBGLP 位置 1，则跟踪将在待机模式下继续刷新，无需执行特定的软件操作。*

调试模式下的外设时钟“冻结”

当处理器在调试状态下停止时，默认外设继续工作。这意味着在触发断点时的系统状态将被覆盖，且无法恢复。此外，某些外设需要定期的软件干预，否则可能会发生超时并产生复位（例如看门狗）。

通过将 DBGMCU 外设冻结寄存器中的相应位置 1，一旦处理器进入调试状态，便可自动停止某些外设时钟。这些时钟将保持停止状态，直到处理器重新启动为止。

以下外设支持该功能：

- 所有定时器（TIMn、LPTIMn）
- 所有 I2C 外设（用于防止 SMBus 超时）
- 所有看门狗定时器 (IWDGn)
- 实时时钟 (RTC)

每个外设总线 (APB) 都有一个冻结寄存器，用于控制该总线上的外设。该寄存器控制外设是否对 Cortex®-A7 调试状态敏感。

当 CA7 内核进入调试状态并且外设冻结寄存器中的相应位置 1 时，外设时钟将被冻结。

调试器能否访问冻结寄存器位取决于认证信号 SPIDEN 的状态。

当 SPIDEN = 1（使能安全特权调试）时，可以修改和读取所有位。

当 SPIDEN = 0（禁止安全特权调试）时，只能修改与非安全外设对应的位，但可以读取所有位。

外设的安全状态由来自 ETZPC 的相应 DECPROTn(1:0) 信号指示，但以下情况例外：如果通过 RTC_SECCFGR 寄存器将至少一个 RTC/TAMP 功能配置为安全，则 RTC/TAMP 外设就是安全的。

微处理器调试单元寄存器

DBGMCU 寄存器不通过系统复位 **nreset** 而复位，只能通过以下方式复位：通过上电复位 **vcore_rst** 而复位、由调试器使用 CDBGRSTREQ/ACK 进行复位，或者通过写入 RCC 调试配置寄存器 (RCC_DBGCFGR) 的 DBGRST 位设置 dbg_rstn 进行复位。调试器可以通过基址为 0xE0081000 的 APB 总线对其进行访问。另外，基址为 0x50081000 的处理器也可以对其进行访问。

DBGMCU 标识代码寄存器 (DBGMCU_IDC)

DBGMCU identity code register

偏移地址：0x000

复位值：0xXXXX 6501

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REV_ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DEV_ID[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 **REV_ID[15:0]：版本 (revision)**

位 15:12 保留，必须保持复位值。

位 11:0 **DEV_ID[11:0]：设备 ID (Device ID)**

0x501：STM32MP13xx

DBGMCU 配置寄存器 (DBGMCU_CR)

DBGMCU configuration register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGLP	Res.	Res.
													rw		

位 31:3 保留，必须保持复位值。

位 2 **DBGLP**：低功耗模式调试使能 (Low power mode debug enable)

0：正常工作。在 Stop、LP-Stop、LPLV-Stop、LPLV-Stop2 和 Standby 模式下，调试访问将会丢失。

1：禁止自动停止时钟/掉电。所有活动的时钟和振荡器将在低功耗模式下继续运行，并将保持 V_{DDCORE} 和 V_{DDCPU} 电源，因此支持全面的调试功能。退出 Standby 或 LPLV-Stop2 模式时，Cortex®-A7 将会复位。

位 1:0 保留，必须保持复位值。

DBGMCU APB4 外设冻结寄存器 (DBGMCU_APB4FZ)

DBGMCU APB4 peripheral freeze register

偏移地址：0x02C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IWDG2	Res.	Res.
													rw		

位 31:3 保留，必须保持复位值。

位 2 **IWDG2**：IWDG2 在调试模式下停止 (IWDG2 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，IWDG2 继续运行。

1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，IWDG2 被冻结。不允许在其 APB 接口上进行任何访问。

位 1:0 保留，必须保持复位值。

DBGMCU APB1 外设冻结寄存器 (DBGMCU_APB1FZ)

DBGMCU APB1 peripheral freeze register

偏移地址：0x034

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C2	I2C1	Res.	Res.
												rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	LPTIM1	Res.	Res.	Res.	TIM7	TIM6	TIM5	TIM4	TIM3	TIM2
						rw				rw	rw	rw	rw	rw	rw

位 31:20 保留，必须保持复位值。

位 19 **I2C2**：在调试模式下，I2C2 SMBUS 超时停止 (I2C2 SMBUS timeout stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，I2C2 SMBUS 超时继续运行。

1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，I2C2 SMBUS 超时被冻结。

位 18 **I2C1**：在调试模式下，I2C1 SMBUS 超时停止 (I2C1 SMBUS timeout stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，I2C1 SMBUS 超时继续运行。

1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，I2C1 SMBUS 超时被冻结。

位 17:10 保留，必须保持复位值。

位 9 **LPTIM1**：LPTIM1 在调试模式下停止 (LPTIM1 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，LPTIM1 继续运行。

1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，LPTIM1 被冻结。

位 8:6 保留，必须保持复位值。

位 5 **TIM7**：TIM7 在调试模式下停止 (TIM7 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，TIM7 继续运行。
1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，TIM7 被冻结。

位 4 **TIM6**：TIM6 在调试模式下停止 (TIM6 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，TIM6 继续运行。
1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，TIM6 被冻结。

位 3 **TIM5**：TIM5 在调试模式下停止 (TIM5 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，TIM5 继续运行。
1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，TIM5 被冻结。

位 2 **TIM4**：TIM4 在调试模式下停止 (TIM4 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，TIM4 继续运行。
1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，TIM4 被冻结。

位 1 **TIM3**：TIM3 在调试模式下停止 (TIM3 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，TIM3 继续运行。
1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，TIM3 被冻结。

位 0 **TIM2**：TIM2 在调试模式下停止 (TIM2 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，TIM2 继续运行。
1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，TIM2 被冻结。

DBGMCU APB2 外设冻结寄存器 (DBGMCU_APB2FZ)

DBGMCU APB2 peripheral freeze register

偏移地址：0x03C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FDCAN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM8	TIM1
rw														rw	rw

位 31:16 保留，必须保持复位值。

位 15 **FDCAN**：在调试模式下 FDCAN 停止 (FDCAN stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，FDCAN 继续运行。
1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，FDCAN 被冻结。

位 14:9 保留，必须保持复位值。

位 8:2 保留，必须保持复位值。

位 1 **TIM8**：TIM8 在调试模式下停止 (TIM8 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，TIM8 继续运行。
1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，TIM8 被冻结。

位 0 **TIM1**：TIM1 在调试模式下停止 (TIM1 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，TIM1 继续运行。
1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，TIM1 被冻结。

DBGMCU APB3 外设冻结寄存器 (DBGMCU_APB3FZ)

DBGMCU APB3 peripheral freeze register

偏移地址：0x044

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM5	LPTIM4	LPTIM3	LPTIM2	Res.
											rw	rw	rw	rw	

位 31:5 保留，必须保持复位值。

位 4 **LPTIM5**：LPTIM5 在调试模式下停止 (LPTIM5 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，LPTIM5 继续运行。
1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，LPTIM5 被冻结。

位 3 **LPTIM4**：LPTIM4 在调试模式下停止 (LPTIM4 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，LPTIM4 继续运行。
1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，LPTIM4 被冻结。

位 2 **LPTIM3**：LPTIM3 在调试模式下停止 (LPTIM3 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，LPTIM3 继续运行。
1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，LPTIM3 被冻结。

位 1 **LPTIM2**：LPTIM2 在调试模式下停止 (LPTIM2 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，LPTIM2 继续运行。
1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，LPTIM2 被冻结。

位 0 保留，必须保持复位值。

DBGMCU APB5 外设冻结寄存器 (DBGMCU_APB5FZ)

DBGMCU APB5 peripheral freeze register

偏移地址：0x04C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTC	IWDG1	Res.	Res.	Res.
											rw	rw			

位 31:5 保留，必须保持复位值。

位 4 **RTC**：RTC 在调试模式下停止 (RTC stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，RTC 继续运行。
1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，RTC 被冻结。

位 3 **IWDG1**：独立看门狗 1 在调试模式下停止 (independent watchdog 1 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，看门狗继续计数。
1：在调试模式下停止。当 Cortex®-A7 内核处于调试模式时，IWGD1 被冻结。不允许在其 APB 接口上进行任何访问。

位 2:0 保留，必须保持复位值。

DBGMCU APB6 外设冻结寄存器 (DBGMCU_APB6FZ)

DBGMCU APB6 peripheral freeze register

偏移地址：0x054

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TIM17	TIM16	TIM15	TIM14	TIM13	TIM12	I2C5	I2C4	I2C3	Res.	Res.	Res.	Res.
			rw	rw	rw	rw	rw	rw	rw	rw	rw				

位 31:13 保留，必须保持复位值。

位 12 **TIM17**：TIM17 在调试模式下停止 (TIM17 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，TIM17 继续运行。
1：在调试模式下停止。当 Cortex®-A7 处于调试模式时，TIM17 被冻结。

位 11 **TIM16**：TIM16 在调试模式下停止 (TIM16 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，TIM16 继续运行。
1：在调试模式下停止。当 Cortex®-A7 处于调试模式时，TIM16 被冻结。

位 10 **TIM15**：TIM15 在调试模式下停止 (TIM15 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，TIM15 继续运行。
1：在调试模式下停止。当 Cortex®-A7 处于调试模式时，TIM15 被冻结。

位 9 **TIM14**：TIM14 在调试模式下停止 (TIM14 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，TIM14 继续运行。
1：在调试模式下停止。当 Cortex®-A7 处于调试模式时，TIM14 被冻结。

位 8 **TIM13**：TIM13 在调试模式下停止 (TIM13 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，TIM13 继续运行。
1：在调试模式下停止。当 Cortex®-A7 处于调试模式时，TIM13 被冻结。

位 7 **TIM12**：TIM12 在调试模式下停止 (TIM12 stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，TIM12 继续运行。
1：在调试模式下停止。当 Cortex®-A7 处于调试模式时，TIM12 被冻结。

位 6 **I2C5**：I2C5 SMBUS 在调试模式下停止 (I2C5 SMBUS stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，I2C5 SMBUS 超时继续运行。
1：在调试模式下停止。当 Cortex®-A7 处于调试模式时，I2C5 SMBUS 超时被冻结。

位 5 **I2C4**：I2C4 SMBUS 在调试模式下停止 (I2C4 SMBUS stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，I2C4 SMBUS 超时继续运行。
1：在调试模式下停止。当 Cortex®-A7 处于调试模式时，I2C4 SMBUS 超时被冻结。

位 4 **I2C3**：I2C3 SMBUS 在调试模式下停止 (I2C3 SMBUS stop in debug)

0：正常工作。当 Cortex®-A7 处于调试模式时，I2C3 SMBUS 超时继续运行。
1：在调试模式下停止。当 Cortex®-A7 处于调试模式时，I2C3 SMBUS 超时被冻结。

位 3:0 保留，必须保持复位值。

DBGMCU 外设 ID4 寄存器 (DBGMCU_PIDR4)

DBGMCU peripheral ID4 register

偏移地址：0xFD0

复位值：0x0000 0000

该寄存器属于外设标识寄存器组。其中包含设计人员标识的一部分和内存占用指示符。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				DES_2[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SIZE[3:0]**：组件内存大小指示符 (component memory size indicator)
指示组件使用的连续内存窗口大小（2 的幂，标准大小为 4 KB）。如果组件只需要标准大小 4 KB，则该位字段必须读为 0x0。对于 8 KB，设为 0x1。对于 16 KB，设为 0x2。对于 32 KB 及更大的内存窗口，设为 0x3。
0x0：设备仅占用 4 KB 存储空间

位 3:0 **DES_2[3:0]**：JEDEC 连续代码 (JEDEC continuation code)
表示组件的设计人员以及标识代码。
0x0：表示 STMicroelectronics JEDEC 标识代码在第一个 bank

DBGMCU 外设 ID0 寄存器 (DBGMCU_PIDR0)

DBGMCU peripheral ID0 register

偏移地址：0xFE0

复位值：0x0000 0000

该寄存器属于外设标识寄存器组。其中包含设计人员特定产品编号的一部分。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PART_0[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PART_0[7:0]**：组件产品编号的位 [7:0] (bits [7:0] of the component part number)
该位字段由组件的设计人员指定。
0x00：CoreSight™ DBGMCU 产品编号[7:0]。产品编号的低八位 (0x000)。

DBGMCU 外设 ID1 寄存器 (DBGMCU_PIDR1)

DBGMCU peripheral ID1 register

偏移地址：0xFE4

复位值：0x0000 0000

该寄存器属于外设标识寄存器组。其中包含设计人员特定产品编号的一部分和设计人员标识的一部分。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DES_0[3:0]				PART_1[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **DES_0[3:0]**：JEDEC 标识代码的位 [3:0] (bits [3:0] of the JEDEC identity code)
表示组件的设计人员以及连续代码。
0x0：JEP106 标识代码的低四位（0x20 = 意法半导体）

位 3:0 **PART_1[3:0]**：组件产品编号的位 [11:8] (bits [11:8] of the component part number)
由组件的设计人员指定。
0x0：CoreSight™ DBGMCU 产品编号[11:8]。产品编号的高四位 (0x000)。

DBGMCU 外设 ID2 寄存器 (DBGMCU_PIDR2)

DBGMCU peripheral ID2 register

偏移地址：0xFE8

复位值：0x0000 000A

该寄存器属于外设标识寄存器组。其中包含设计人员标识的一部分和产品版本。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	DES_1[2:0]		
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：递增组件设计版本 (incremental component design version)
从 0x0（对应于该组件的第一版设计）开始的递增值。无论是主版本更新还是次版本更新，该值都会增加 1，因此查看该值即可得知确切的主版本和次版本。
0x0：该组件为版本 0

位 3 **JEDEC**：JEDEC 分配的值使用情况 (JEDEC assigned value usage)
指示是否使用 JEDEC 指定的值。该位始终置 1。
1：设计人员 ID 由 JEDEC 指定（请参见 http://www.jedec.org）

位 2:0 **DES_1[2:0]**：JEDEC 标识代码的位 [6:4] (bits [6:4] of the JEDEC identity code)
表示组件的设计人员以及连续代码。
0x2：JEP106 标识代码的高三位（0x20 = 意法半导体）

DBGMCU 外设 ID3 寄存器 (DBGMCU_PIDR3)

DBGMCU peripheral ID3 register

偏移地址：0xFEC

复位值：0x0000 0000

该寄存器属于外设标识寄存器组。其中包含 RevAnd 和 Customer_Modified 位字段。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：次要修复指示符 (minor fix indicator)
指示针对设计的次要勘误修复，例如设计后的金属修复。在大多数情况下，该位字段为 0x0。Arm® 建议组件设计人员应确保可根据需要通过金属修复更改位字段，例如通过驱动寄存器中复位为零的位字段。
0x0：组件中无金属修复

位 3:0 **CMOD[3:0]**：客户修改指示符 (customer modification indicator)
当组件为可重复使用的 IP 时，该值指示客户是否已修改组件的行为。在大多数情况下，该位字段为 0x0。
0x0：未做修改

DBGMCU 组件 ID0 寄存器 (DBGMCU_CIDR0)

DBGMCU component ID0 register

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_0[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PRMBL_0[7:0]**：组件标识的位 [31:24] (bits [31:24] of component identification)
组件标识寄存器，用于指示存在标识寄存器。
0x0D：标识值

DBGMCU 组件 ID1 寄存器 (DBGMCU_CIDR1)

DBGMCU component ID1 register

偏移地址：0xFF4

复位值：0x0000 00F0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PRMBL_1[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件类 (component class)
指示组件的类别，例如 ROM 表或 CoreSight™ 组件。
0xF：指示该组件为非 CoreSight™ 组件

位 3:0 **PRMBL_1[3:0]**：组件标识的位 [19:16] (bits [19:16] of component identification)
组件标识寄存器，用于指示存在标识寄存器。
0x0：标识值

DBGMCU 组件 ID2 寄存器 (DBGMCU_CIDR2)

DBGMCU component ID2 register

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_2[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PRMBL_2[7:0]**：组件标识的位 [15:8] (bits [15:8] of component identification)
组件标识寄存器，用于指示存在标识寄存器。
0x05：标识值

DBGMCU 组件 ID3 寄存器 (DBGMCU_CIDR3)

DBGMCU component ID3 register

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_3[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PRMBL_3[7:0]**：组件标识的位 [7:0] (bits [7:0] of component identification)

组件标识寄存器，用于指示存在标识寄存器。

0xB1：标识值

微处理器控制器调试单元寄存器映射

表 615. DBGMCU 寄存器映射和复位值

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**DBGMCU_IDC**	REV_ID[15:0]																Res.	Res.	Res.	Res.	DEV_ID[11:0]											
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x					0	1	0	1	0	0	0	0	0	0	0	1
0x004	**DBGMCU_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGLP	Res.	Res.
	Reset value																														0		
0x008 to 0x028	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x02C	**DBGMCU_ APB4FZ**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IWDG2	Res.	Res.
	Reset value																														0		
0x030	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x034	**DBGMCU_ APB1FZ**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C2	I2C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM1	Res.	Res.	Res.	TIM7	TIM6	TIM5	TIM4	TIM3	TIM2
	Reset value													0	0									0				0	0	0	0	0	0
0x038	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x03C	**DBGMCU_ APB2FZ**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCAN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM8	TIM1
	Reset value																	0														0	0
0x040	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 615. DBGMCU 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x044	**DBGMCU_APB3FZ**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM5	LPTIM4	LPTIM3	LPTIM2	Res.
	Reset value																												0	0	0	0	
0x048	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x04C	**DBGMCU_APB5FZ**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTC	IWDG1	Res.	Res.	Res.
	Reset value																												0	0			
0x050	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x054	**DBGMCU_APB6FZ**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM17	TIM16	TIM15	TIM14	TIM13	TIM12	I2C5	I2C4	I2C3	Res.	Res.	Res.	Res.
	Reset value																				0	0	0	0	0	0	0	0	0				
0x058-0xFCC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xFD0	**DBGMCU_PIDR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				DES_2[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFD4-0xFDC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xFE0	**DBGMCU_PIDR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PART_0[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0xFE4	**DBGMCU_PIDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DES_0[3:0]				PART_1[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFE8	**DBGMCU_PIDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	DES_1[2:0]		
	Reset value																									0	0	0	0	1	0	1	0
0xFEC	**DBGMCU_PIDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	**DBGMCU_CIDR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_0[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	**DBGMCU_CIDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PRMBL_1[3:0]			
	Reset value																									1	1	1	1	0	0	0	0

表 615. DBGMCU 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFF8	**DBGMCU_CIDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_2[7:0]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	**DBGMCU_CIDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRMBL_3[7:0]							
	Reset value																									1	0	1	1	0	0	0	1

59.11 Cortex®-A7 调试功能

Cortex®-A7 子系统集成有以下 CoreSight™ 组件：

- 调试单元 (DBG)
- 性能监视单元 (PMU)
- 嵌入式跟踪宏单元 (ETM)
- 交叉触发接口 (CTI)

调试器可通过 APB-AP 及其相关的调试 APB 总线访问这些组件。此外，Cortex®-A7 通过系统互连也可以访问这些组件。

59.11.1 Cortex®-A7 调试单元

通过使用调试单元，调试器可以访问 Cortex®-A7 MPCore 内置的调试功能。外部调试器可使用这些调试功能：

- 停止执行程序
- 检查并更改过程和协处理器状态
- 检查并更改内存和输入/输出外设状态
- 重启处理器

有关调试单元的详细信息，请参见《Cortex®-A7 MPCore 技术参考手册》*[6]*。

调试单元寄存器

调试单元包含多个寄存器，调试器可以通过调试 APB 在 0xE00D 0000 到 0xE00D 0FFC 的地址范围内访问这些寄存器。软件可以在 0x500D 0000 到 0x500D 0FFC 的地址范围内访问这些寄存器。

Cortex®-A7 内核可以通过 CP14（基线和扩展）寄存器，使用 MCR、MRC、MCCR 和 MRRC 指令访问其自身的某些调试单元寄存器。

DBG 标识寄存器 (DBG_DIDR)

DBG identification register

偏移地址：0x000

复位值：0x3515 F005

指定设计哪个版本的调试架构以及哪些特性。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRPS[3:0]				BRPS[3:0]				CTX_CMPS[3:0]				VERSION[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEVIDIMP	NSUHDIMP	PCSRIMP	SEIMP	Res.	Res.	Res.	Res.	VARIANT[3:0]				REVISION[3:0]			
r	r	r	r					r	r	r	r	r	r	r	r

位 31:28 **WRPS[3:0]**：监测点寄存器对 (Watchpoint Registers Pairs)。
指示设计的监测点寄存器对数量。
0x3：处理器设计 4 个 WRP

位 27:24 **BRPS[3:0]**：断点寄存器对 (Breakpoint Registers Pairs)。
指示设计的断点寄存器对数量。
0x5：处理器设计 6 个 BRP

位 23:20 **CTX_CMPS[3:0]**：上下文 ID 比较器 (Context ID comparators)。
指示可用于上下文 ID 比较的 BRP 数量。
0x1：处理器设计 2 个可用于上下文 ID 比较的断点

位 19:16 **VERSION[3:0]**：调试架构版本 (Debug Architecture version)。
0x5：Arm®v7.1 调试架构

位 15 **DEVIDIMP**：已设计调试设备 ID 寄存器 (DBGDEVID) (Debug Device ID register (DBGDEVID) implemented)。
1：已设计

位 14 **NSUHDIMP**：已设计安全用户停止调试 (Secure User Halting Debug implemented)。
1：未设计

位 13 **PCSRIMP**：已设计程序计数器采样寄存器 (DBGPCSR)，即寄存器 33 (Program Counter Sampling Register (DBGPCSR) implemented as register 33)。
1：已设计

位 12 **SEIMP**：已设计安全扩展 (Security Extensions implemented)。
1：已设计

位 11:8 保留，必须保持复位值。

位 7:4 **VARIANT[3:0]**：处理器变体编号 (Processor Variant number)。
该编号会随功能变更而递增。该位字段与 CP15 主 ID 寄存器的位 [23:20] 一致。
0x0：变体编号 = 0

位 3:0 **REVISION[3:0]**：处理器版本号 (Processor Revision number)。
该编号会随错误修复而递增。该位字段与 CP15 主 ID 寄存器的位 [3:0] 一致。
0x5：版本号 = 5

DBG 监测点故障地址寄存器 (DBG_WFAR)

DBG watchpoint fault address register

偏移地址：0x018

复位值：0x0000 0000

返回有关访问监测点地址的指令的地址信息。

在该版本的调试架构中未使用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRESS[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ADDRESS[31:0]**：指令地址 (Instruction address)。
该寄存器未使用。

DBG 向量捕获寄存器 (DBG_VCR)

DBG vector catch register

偏移地址：0x01C

复位值：0x0000 0000

控制向量捕获调试事件。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NSF	NSI	Res.	NSD	NSP	NSC	NSU	Res.	NSHF	NSHI	NSHE	NSHD	NSHP	NSHC	NSHU	Res.
rw	rw		rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MF	MI	Res.	MD	MP	MS	Res.	Res.	SF	SI	Res.	SD	SP	SS	SU	R
rw	rw		rw	rw	rw			rw	rw		rw	rw	rw	rw	rw

位 31 **NSF**：使能 FIQ 中断异常非安全本地向量捕获 (FIQ interrupt exception non-secure local Vector catch enable)。
0：禁止
1：使能

位 30 **NSI**：使能 IRQ 中断异常非安全本地向量捕获 (IRQ interrupt exception non-secure local Vector catch enable)。
0：禁止
1：使能

位 29 保留，必须保持复位值。

位 28 **NSD**：使能数据中止异常非安全本地向量捕获 (Data Abort exception non-secure local Vector catch enable)。
0：禁止
1：使能

位 27 **NSP**：使能预取中止异常非安全本地向量捕获 (Prefetch Abort exception non-secure local Vector catch enable)。
0：禁止
1：使能

位 26 **NSC**：使能监控器调用异常非安全本地向量捕获 (Supervisor Call exception non-secure local Vector catch enable)。
0：禁止
1：使能

位 25 **NSU**：使能未定义指令异常非安全本地向量捕获 (Undefined Instruction exception non-secure local Vector catch enable)。
0：禁止
1：使能

位 24 保留，必须保持复位值。

位 23 **NSHF**：使能 FIQ 中断异常非安全 Hyp 向量捕获 (FIQ interrupt exception non-secure Hyp Vector catch enable)。
0：禁止
1：使能

位 22 **NSHI**：使能 IRQ 中断异常非安全 Hyp 向量捕获 (IRQ interrupt exception non-secure Hyp Vector catch enable)。
0：禁止
1：使能

位 21 **NSHE**：使能 Hyp 陷阱或 Hyp 模式进入异常非安全 Hyp 向量捕获 (Hyp Trap or Hyp mode entry exception non-secure Hyp Vector catch enable)。
0：禁止
1：使能

位 20 **NSHD**：使能数据中止异常非安全 Hyp 向量捕获 (Data Abort exception non-secure Hyp Vector catch enable)。
0：禁止
1：使能

位 19 **NSHP**：使能预取中止异常非安全 Hyp 向量捕获 (Prefetch Abort exception non-secure Hyp Vector catch enable)。
0：禁止
1：使能

位 18 **NSHC**：使能管理程序调用异常非安全 Hyp 向量捕获 (Hypervisor Call exception non-secure Hyp Vector catch enable)。
0：禁止
1：使能

位 17 **NSHU**：使能未定义指令异常非安全 Hyp 向量捕获 (Undefined Instruction exception non-secure Hyp Vector catch enable)。
0：禁止
1：使能

位 16 保留，必须保持复位值。

位 15 **MF**：使能 FIQ 中断异常安全监视模式向量捕获 (FIQ interrupt exception secure monitor mode Vector catch enable)。
0：禁止
1：使能

位 14 **MI**：使能 IRQ 中断异常安全监视模式向量捕获 (IRQ interrupt exception secure monitor mode Vector catch enable)。
0：禁止
1：使能

位 13 保留，必须保持复位值。

位 12 **MD**：使能数据中止异常安全监视模式向量捕获 (Data Abort exception secure monitor mode Vector catch enable)。
0：禁止
1：使能

位 11 **MP**：使能预取中止异常安全监视模式向量捕获 (Prefetch Abort exception secure monitor mode Vector catch enable)。
0：禁止
1：使能

位 10 **MS**：使能安全监视调用异常向量捕获 (Secure Monitor Call exception Vector catch enable)。
0：禁止
1：使能

位 9:8 保留，必须保持复位值。

位 7 **SF**：使能 FIQ 中断异常安全本地向量捕获 (FIQ interrupt exception secure local Vector catch enable)。
0：禁止
1：使能

位 6 **SI**：使能 IRQ 中断异常安全本地向量捕获 (IRQ interrupt exception secure local Vector catch enable)。
0：禁止
1：使能

位 5 保留，必须保持复位值。

位 4 **SD**：使能数据中止异常安全本地向量捕获 (Data Abort exception secure local Vector catch enable)。
0：禁止
1：使能

位 3 **SP**：使能预取中止异常安全本地向量捕获 (Prefetch Abort exception secure local Vector catch enable)。
0：禁止
1：使能

位 2 **SS**：使能监控器调用异常安全本地向量捕获 (Supervisor Call exception secure local Vector catch enable)。
0：禁止
1：使能

位 1 **SU**：使能未定义指令异常安全本地向量捕获 (Undefined Instruction exception secure local Vector catch enable)。
0：禁止
1：使能

位 0 **R**：使能复位向量捕获 (Reset Vector catch enable)。
0：禁止
1：使能

DBG 事件捕获寄存器 (DBG_ECR)

DBG event catch register

偏移地址：0x024

复位值：0x0000 0000

配置调试逻辑以在 OS 锁定位清零时生成 OS 解锁捕获调试事件。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSUCE
															rw

位 31:1 保留，必须保持复位值。

位 0 **OSUCE**：OS 解锁捕获使能 (OS Unlock Catch enable)。
当通过写入 DBGOSLAR 清零 OS 锁定位时，将生成 OS 解锁捕获调试事件。
0：禁止
1：使能

DBG 主机到目标数据传输寄存器 (DBG_DTRRX)

DBG host to target data transfer register

偏移地址：0x080

复位值：0x0000 0000

将数据从外部主机传输到 Arm® 处理器。例如，调试器使用该寄存器将命令和数据传输到调试目标。该寄存器是调试通信通道 (DCC) 的组成部分。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DATA[31:0]**：主机到目标的数据 (Host to Target Data)。
从调试主机传输到调试目标的一个数据字。

DBG 指令传输寄存器 (DBG_ITR)

DBG instruction transfer register

偏移地址：0x084

复位值：0x0000 0000

当处理器处于调试状态时，将 Arm® 指令传输到处理器来执行。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INSTRUCTION[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INSTRUCTION[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **INSTRUCTION[31:0]**：指令 (Instruction)。
在处理器上执行的 32 位编码的 Arm® 指令

DBG 状态和控制寄存器 (DBG_DSCR)

DBG status and control register

偏移地址：0x088

复位值：0x0000 0000

用于调试的主控制寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	RXFULL	TXFULL	Res.	RXFULL_L	TXFULL_L	PIPEADV	INSTRCOMPL_L	Res.	Res.	EXTDCCMODE[1:0]		ADADISCARD	NS	SPNIDDIS	SPIDDIS
	rw	rw		rw	rw	rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MDBGEN	HDBGEN	ITREN	UDCCDIS	INTDIS	DBGACK	FS	UND_L	ADABORT_L	SDABORT_L	MOE[3:0]				RESTARTED	HALTED
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30 **RXFULL**：DBGDTRRX 寄存器已满 (DBGDTRRX register full)。

0：寄存器为空

1：寄存器已满

位 29 **TXFULL**：DBGDTRTX 寄存器已满 (DBGDTRTX register full)。

0：寄存器为空

1：寄存器已满

位 28 保留，必须保持复位值。

位 27 **RXFULL_L**：锁存 RXfull (Latched RXfull)。

该位控制在非阻塞模式 (DBGDSCR.EXTDCCMODE = 0x0) 下写入 DBGDTRRX 时的处理器行为

0：执行对 DBGDTRRX 的写操作并将 RXFULL 和 RXFULL_L 位置 1

1：忽略对 DBGDTRRX 的写操作

位 26 **TXFULL_L**：锁存 TXfull (Latched TXfull)。

该位控制在非阻塞模式 (DBGDSCR.EXTDCCMODE = 0x0) 下读取 DBGDTRTX 时的处理器行为

0：读取 DBGDTRTX 时返回一个未定义的值

1：读取 DBGDTRTX 时返回寄存器的内容并将 TXFULL 和 TXFULL_L 位复位为 0

位 25 **PIPEADV**：粘滞流水线前进 (Sticky Pipeline Advance)（只读）。

只要处理器流水线通过退出一条或多条指令而前进，该位就会置 1。该位只能通过 DBGDRCR.CSPA 来清零。调试器可以使用该位来检测处理器是否处于空闲状态。在某些情况下，该位可能指示处理器发生死锁。

位 24 **INSTRCOMPL_L**：锁存指令完成 (Latched Instruction Complete)（只读）。

该位是内部 InstrCompl 标志的副本，在每次读取 DBGDSCR InstrCompl 信号时获取，与处理器是否已执行完通过 DBGITR 发出的指令无关。InstrCompl 在任何寄存器中都无法直接访问。在处理器处于调试状态的情况下读取 DBGDSCR 时，InstrCompl_l 始终返回 InstrCompl 的当前值。InstrCompl_l 值的含义如下：

0：之前通过 DBGITR 发出的指令尚未完成其对处理器架构状态的更改。

1：之前通过 DBGITR 发出的所有指令都已完成其对处理器架构状态的更改。

位 23:22 保留，必须保持复位值。

位 21:20 **EXTDCCMODE[1:0]**：外部 DCC 访问模式 (External DCC Access Mode)。
该位字段控制 DCC 寄存器的访问模式和 DBG_ITR。
0x0：非阻塞模式。始终在不阻塞总线的情况下完成访问。
0x1：阻塞模式。在阻塞总线的情况下执行访问，直到成功完成访问为止。
0x2：快速模式。可以将指令锁存在 DBG_ITR 中，并通过连续读取 DBG_DTRTX 或连续写入 DBG_DTRRX 来反复执行指令。
0x3：保留

位 19 **ADADISCARD**：丢弃异步中止 (Asynchronous Aborts Discarded)。
0：正常处理异步中止
1：发生该位适用的异步中止时，处理器会将粘滞异步中止位 ADABORT_L 置 1，但该位不适用的中止会被丢弃。

位 18 **NS**：非安全状态 (Non-Secure state status)（只读）。
0：处理器处于安全状态
1：处理器处于非安全状态

位 17 **SPNIDDIS**：安全 PL1 非侵入式调试已禁止 (Secure PL1 Non-Invasive Debug Disabled)（只读）。
该位显示是否允许在安全 PL1 模式下进行非侵入式调试。
0：允许在安全 PL1 模式下进行非侵入式调试
1：不允许在安全 PL1 模式下进行非侵入式调试

位 16 **SPIDDIS**：安全 PL1 侵入式调试已禁止 (Secure PL1 Invasive Debug Disabled)（只读）。
该位显示是否允许在安全 PL1 模式下进行侵入式调试。
0：允许在安全 PL1 模式下进行侵入式调试
1：不允许在安全 PL1 模式下进行侵入式调试

位 15 **MDBGEN**：使能监视器调试模式 (Monitor Debug mode enable)。
0：禁止监视器调试模式
1：使能监视器调试模式

位 14 **HDBGEN**：使能暂停调试模式 (Halting Debug mode enable)。
0：禁止暂停调试模式
1：使能暂停调试模式

位 13 **ITREN**：允许执行 Arm® 指令 (Enable Arm® instruction execution)。
该位允许通过 DBGITR 执行 Arm® 指令。当处理器不处于调试状态时，该位应始终写为低电平。
0：禁止 ITR 机制
1：使能 ITR 机制

位 12 **UDCCDIS**：禁止在用户模式下访问调试通信通道 (DCC) (User mode access to Debug Communications Channel (DCC) disable)。
0：允许在用户模式下访问 DCC
1：在用户模式下通过 CP14 访问任何 DCC 寄存器都将生成未定义的指令异常

位 11 **INTDIS**：禁止中断 (Interrupt disable)。
0：使能 FIQ 和 IRQ
1：屏蔽 FIQ 和 IRQ

位 10 **DBGACK**：强制调试确认 (Force Debug Acknowledge)。
该位允许调试器仿真在内核保持非调试模式时通过交叉触发进入调试模式。
0：正常工作
1：将 DBGACK 置为有效

位 9 **FS**：故障状态 (Fault Status)（只读）。

该位在调试状态下生成数据中止异常时更新。该位指示保存故障信息的位置。

0：故障状态信息可能保存在 PL1 故障报告寄存器（DFSR 和 DFAR）或 PL2 故障表征寄存器（HSR、HDFAR）中，具体取决于当前状态、模式和 HCR.TGE 寄存器值

1：故障状态信息写入 PL2 故障表征寄存器

位 8 **UND_L**：粘滞未定义指令 (Sticky Undefined Instruction)（只读）。

在调试状态下，如果发送到处理器的指令生成任何未定义的指令异常，则该位置 1。

0：自该位上次清零之后未生成未定义的指令异常

1：自该位上次清零之后生成了未定义的指令异常

位 7 **ADABORT_L**：粘滞异步中止 (Sticky Asynchronous Abort)（只读）。

如果 ADADISCARD 位置 1，则在调试状态下发生任何异步中止时，该位将置 1。

0：自该位上次清零之后未生成异步中止异常

1：自该位上次清零之后生成了异步中止异常

位 6 **SDABORT_L**：粘滞同步数据中止 (Sticky Synchronous Data Abort)（只读）。

在调试状态下，如果生成任何同步数据中止异常，则该位置 1。

0：自该位上次清零之后未生成同步数据中止异常

1：自该位上次清零之后生成了同步数据中止异常

位 5:2 **MOE[3:0]**：进入调试模式的方法 (Method of Debug Entry)（只读）。

指示触发处理器进入调试模式的调试事件。

0x0：暂停请求

0x1：断点

0x2：异步监测点

0x3：BKPT 指令

0x4：外部调试请求

0x5：向量捕获

0x6 到 0x7：保留

0x8：OS 解锁捕获

0x9：保留

0xA：同步监测点

0xB 到 0xF：保留

位 1 **RESTARTED**：处理器已重启 (Processor Restarted)（只读）。

发出重启请求后，调试器可以轮询该位，直到该位置 1 为止。那时即可确定重启请求已生效，处理器已退出调试状态。

0：处理器正在退出调试状态。该位仅在接收到重启请求之后、但在重启非调试状态操作之前读为 0

1：处理器已退出调试状态。如果处理器重新进入调试状态，该位保持置 1。

位 0 **HALTED**：处理器已暂停 (Processor Halted)（只读）。

在编程调试事件后，外部调试器可以轮询该位，直到该位置 1 为止。那时即可确定处理器已进入调试状态。

0：处理器处于非调试状态

1：处理器处于调试状态

DBG 目标到主机数据传输寄存器 (DBG_DTRTX)

DBG target to host data transfer register

偏移地址：0x08C

复位值：0x0000 0000

将数据从 Arm® 处理器传输到外部主机。例如，调试目标使用该寄存器将命令和数据传输到调试器。该寄存器是调试通信通道 (DCC) 的组成部分。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DATA[31:0]**：目标到主机的数据 (Target to Host Data)。
从调试目标传输到调试主机的一个数据字。

DBG 运行控制寄存器 (DBG_DRCR)

DBG run control register

偏移地址：0x090

复位值：0x0000 0000

软件使用该寄存器请求处理器进入或退出调试状态，并清零 DBG_DSCR 中的粘滞异常位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CBRRQ	CSPA	CSE	RRQ	HRQ
											rw	rw	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4 **CBRRQ**：取消总线请求 (Cancel Bus Requests)。
0：无操作
1：请求取消挂起的访问

位 3 **CSPA**：清零粘滞流水线前进位 (Clear Sticky Pipeline Advance bit)。
0：无操作
1：将 DBG_DSCR.PIPEADV 位复位为 0

位 2 **CSE**：清零粘滞异常位 (Clear Sticky Exception bits)。
0：无操作
1：将 DBG_DSCR[8:6] 位复位为 0

位 1 **RRQ**：重启请求 (Restart Request)。

该请求将一直保持到处理器退出调试状态为止。发出请求后，调试器可以轮询 DBG_DSCR.RESTARTED 位，直到该位读为 1 为止。

0：无操作

1：请求退出调试模式。

位 0 **HRQ**：暂停请求 (Halt Request)。

该请求将一直保持到处理器进入调试状态为止。发出请求后，调试器可以轮询 DBG_DSCR.HALTED 位，直到该位读为 1 为止。

0：无操作

1：请求进入调试模式。

DBG 外部辅助控制寄存器 (DBG_EACR)

DBG external auxiliary control register

偏移地址：0x094

复位值：0x0000 0000

提供由设计定义的配置和控制选项。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CDRSTAT	Res.	Res.	Res.
												r			

位 31:4 保留，必须保持复位值。

位 3 **CDRSTAT**：内核调试复位状态 (Core Debug Reset Status)（只读）。

反映 CPU 电源域中的调试逻辑的当前复位状态。

0：CPU 电源域中的调试逻辑未处于复位状态

1：CPU 电源域中的调试逻辑处于复位状态

位 2:0 保留，必须保持复位值。

DBG 程序计数器采样寄存器 (DBG_PCSR)

DBG program counter sampling register

偏移地址：0x0A0

复位值：0x0000 0000

该寄存器提供由设计定义的配置和控制选项。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PCS[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCS[15:1]															T
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:1 **PCS[31:1]**：程序计数器采样值 (Program Counter Sample value)。

PC 的位 [31:1] 的采样值。该采样值为指令的虚拟地址或指令的虚拟地址加上偏移量，具体取决于处理器指令集状态。DBG_DEVID1.PCSROFFSET 指示是否对采样地址应用偏移量。

位 0 **T**：Thumb 指令 (Thumb instruction)。

该位指示采样地址是 Arm® 指令，还是 Thumb 或 ThumbEE 指令：

0：Arm® 指令

1：Thumb 或 ThumbEE 指令

DBG 上下文 ID 采样寄存器 (DBG_CIDSR)

DBG context ID sampling register

偏移地址：0x0A4

复位值：0x0000 0000

只要 DBG_PCSR 对程序计数器进行采样，就会对 CONTEXTIDR 进行采样。因此，调试器可以将程序计数器采样与处理器上运行的进程相关联。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CONTEXTIDR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONTEXTIDR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CONTEXTIDR[31:0]**：上下文 ID 寄存器采样值 (Context ID Register sample value)。

与从 DBG_PCSR 读取的最后一个 PC 采样相关联的上下文 ID 寄存器 CONTEXTIDR 的值。

DBG 虚拟化 ID 采样寄存器 (DBG_VIDSR)

DBG virtualisation id sampling register

偏移地址：0x0A8

复位值：0x0000 0000

只要 DBGPCSR 对程序计数器进行采样，就会对 VMID、Hyp 模式状态和 NS 状态进行采样。因此，调试器可以将程序计数器采样与处理器上运行的虚拟机相关联。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NS	H	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VMID[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31 **NS**：采样 NS 状态 (Sampled NS state)。

指示与从 DBGPCSR 读取的最后一个 PC 采样相关联的安全或非安全状态。

0：安全状态

1：非安全状态

位 30 **H**：采样 Hyp 模式 (Sampled Hyp mode)。指示从 DBG_PCSR 读取的最后一个 PC 采样是否与 Hyp 模式相关联。

0：非 Hyp 模式

1：Hyp 模式

位 29:8 保留，必须保持复位值。

位 7:0 **VMID[7:0]**：VMID 采样 (VMID sample)。

VTTBR 中与从 DBG_PCSR 读取的最后一个 PC 采样相关联的 VMID 位字段的值。

DBG 断点值 x 寄存器 (DBG_BVRx)

DBG breakpoint value x register

偏移地址：0x100 + 0x4 * x（x = 0 到 5）

复位值：0x0000 0000

保存用于断点匹配的值，具体可以是指令地址或上下文 ID。每个 DBG_BVR 与一个 DBG_BCR 相关联，以形成断点寄存器对 (BRP)。例如，DBG_BVR[n] 与 DBG_BCR[n] 相关联以形成断点 n。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRESSCONTEXTID[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESSCONTEXTID[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **ADDRESSCONTEXTID[31:0]**：指令地址或上下文 ID (Instruction Address or Context ID)。

如果寄存器用于地址比较，该位字段包含要进行匹配的指令地址（位 31:2；位 1:0 必须设置为零）。如果寄存器用于上下文 ID 比较，该位字段包含要进行匹配的上下文 ID。

DBG 断点控制 x 寄存器 (DBG_BCRx)

DBG breakpoint control x register

偏移地址：0x140 + 0x4 * x（x = 0 到 5）

复位值：0x0000 0000

保存断点的控制信息。每个 DBG_BCR 与一个 DBG_BVR 相关联，以形成断点寄存器对 (BRP)。例如，DBG_BCR[n] 与 DBG_BVR[n] 相关联以形成断点 n。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BT[3:0]				LBN[3:0]			
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSC[1:0]		HMC	Res.	Res.	Res.	Res.	BAS[3:0]				Res.	Res.	PMC[1:0]		E
rw	rw	rw					rw	rw	rw	rw			rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:20 **BT[3:0]**：断点类型 (Breakpoint Type)。
决定断点进行的比较类型。
0x0：非链接指令地址匹配
0x1：链接指令地址匹配
0x2：非链接上下文 ID 匹配
0x3：链接上下文 ID 匹配
0x4：非链接指令地址不匹配
0x5：链接指令地址不匹配
0x6-0x7：保留
0x8：非链接 VMID 匹配
0x9：链接 VMID 匹配
0xA：非链接 VMID 匹配和上下文 ID 匹配
0xB：链接 VMID 匹配和上下文 ID 匹配
0xC-0xF：保留

位 19:16 **LBN[3:0]**：链接的断点编号 (Linked Breakpoint Number)。
如果该断点设定为链接指令地址匹配或不匹配，则该位字段必须设定为存放与指令地址和上下文组合比较结合使用的上下文匹配的断点编号。否则，该位字段必须设定为 0。

位 15:14 **SSC[1:0]**：安全状态控制 (Security State Control)。
该位字段允许断点以处理器的安全状态为条件触发。如果 DBG_BCR.BT 设定为链接上下文 ID 匹配，则该位字段必须设定为 0。
0：安全和非安全状态
1：仅非安全状态
2：仅安全状态
3：仅 Hyp 模式（HMC = 1 且 PMC = 0）

位 13 **HMC**：Hyp 模式控制 (yp Mode Control)。
该位字段控制断点在 Hyp 模式下是否匹配。
如果 DBG_BCR.BT 设定为链接上下文 ID 匹配，则该位字段必须设定为 0。
0：断点在 Hyp 模式下不匹配
1：断点可在 Hyp 模式下匹配

位 12:9 保留，必须保持复位值。

位 8:5 **BAS[3:0]**：字节地址选择 (Byte Address Select)。
该位字段允许只对 DBG_BVR 中保存的字地址的特定字节进行匹配或不匹配比较。该位字段的操作还取决于：
断点类型位字段设定为指令地址匹配还是不匹配。
处理器的指令集状态，由 CPSR.J 位和 CPSR.T 位指示。
如果 DBG_BVR.BT 设定为链接或非链接上下文 ID 匹配，则该位字段必须设定为 0xF。

位 4:3 保留，必须保持复位值。

位 2:1 **PMC[1:0]**：特权模式控制 (Privileged Mode Control)。
该位字段允许断点以处理器的模式为条件触发。如果 DBG_BCR.BT 设定为链接上下文 ID 匹配，则该位字段必须设定为 3。
0：仅 PL0、监控器和系统模式
1：仅 PL1 和 PL2 模式
2：仅 PL0 模式
3：所有模式

位 0 **E**：使能断点 (Breakpoint Enable)。
0：禁止断点
1：使能断点

DBG 监测点值 x 寄存器 (DBG_WVRx)

DBG watchpoint value x register

偏移地址：0x180 + 0x4 * x（x = 0 到 3）

复位值：0x0000 0000

包含用于监测点匹配的值，使用的地址必须是数据的虚拟地址。每个 DBG_WVR 与一个 DBG_WCR 相关联，以形成断点寄存器对 (BRP)。例如，DBG_WVR[n] 与 DBG_WCR[n] 相关联以形成断点 n。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRESS[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS[15:2]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31:2 **ADDRESS[31:2]**：数据地址 (Data Address)。
该位字段指示进行比较的地址值的位[31:2]。

位 1:0 保留，必须保持复位值。

DBG 监测点控制 x 寄存器 (DBG_WCRx)

DBG watchpoint control xregister

偏移地址：0x1C0 + 0x4 * x（x = 0 到 3）

复位值：0x0000 0000

保存监测点的控制信息。每个 DBG_WCR 与一个 DBG_WVR 相关联，以形成断点寄存器对 (BRP)。例如，DBG_WCR[n] 与 DBG_WVR[n] 相关联以形成断点 n。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	MASK[4:0]					Res.	Res.	Res.	WT	LBN[3:0]			
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSC[1:0]		HMC	BAS[7:0]								LSC[1:0]		PAC[1:0]		E
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **MASK[4:0]**：地址屏蔽 (Address Mask)。
该位字段可通过在监测点比较中屏蔽低位地址位，以设置针对某个地址范围的监测点。该位字段的值是地址中被屏蔽的低位位数，但值 1 和 2 保留。
0x0：不屏蔽
0x1：保留
0x2：保留
0x3：屏蔽三位（数据地址掩码 = 0x00000007）
0x4：屏蔽四位（数据地址掩码 = 0x0000000F）
0x5：屏蔽五位（数据地址掩码 = 0x0000001F）
..
0x1F：屏蔽 31 位（数据地址掩码 = 0x7FFFFFFF）

位 23:21 保留，必须保持复位值。

位 20 **WT**：监测点类型 (Watchpoint Type)。

该位置 1 时可将监测点与断点链接，以创建一个需要同时进行数据地址匹配和上下文 ID 匹配的链接监测点。链接的断点编号位字段指示该监测点所链接的断点。

0：禁止链接。

1：使能链接。

位 19:16 **LBN[3:0]**：链接的断点编号 (Linked Breakpoint Number)。

如果该监测点类型设定为使能链接，则该位字段必须设定为定义与数据地址比较结合使用的上下文匹配的断点编号。否则，该位字段必须设定为 0。

位 15:14 **SSC[1:0]**：安全状态控制 (Security State Control)。

该位字段允许监测点以处理器的安全状态为条件触发。

0：安全和非安全状态

1：仅非安全状态

2：仅安全状态

3：仅 Hyp 模式（HMC = 1 且 PAC = 0）

位 13 **HMC**：Hyp 模式控制 (yp Mode Control)。

该位字段控制监测点在 Hyp 模式下是否匹配。

0：监测点在 Hyp 模式下不匹配

1：监测点可在 Hyp 模式下匹配

位 12:5 **BAS[7:0]**：字节地址选择 (Byte Address Select)。

DBG_WVR 使用字对齐地址进行编程。该位字段允许仅在访问了所寻址字的特定字节时触发监测点。即使在以下情况下，只要访问命中了任何正被观察的字节，监测点就会触发：

访问大小超过了正被观察的区域大小。

访问是未对齐访问，并且访问的基址与 DBG_WVR 中的地址不在同一个内存字中。

在一个字中被观察的字节应该是连续的。例如，不得编程为 11100001。

BAS 的示例值：

00000001：观察字节 0

00000011：观察字节 1 和 0

00000010：观察字节 1

00000111：观察字节 2、1 和 0

11110000：观察字节 7、6、5 和 4

以此类推。

位 4:3 **LSC[1:0]**：加载/存储访问控制 (Load/store access control)。

该位字段允许监测点匹配以当前访问的类型为条件触发。

0：保留

1：在发生任何加载、独占加载或交换访问时触发匹配

2：在发生任何存储、独占存储或交换访问时触发匹配

3：在发生任何类型的访问时均触发匹配

位 2:1 **PAC[1:0]**：特权访问保护 (Privileged Access Control)。

该位字段允许监测点匹配以处理器的模式为条件触发。

0：仅 PL0、监控器和系统模式

1：仅 PL1 和 PL2 模式

2：仅 PL0 模式

3：所有模式

位 0 **E**：监测点使能 (Watchpoint Enable)。

0：禁止监测点

1：使能监测点

DBG 断点扩展值 x 寄存器 (DBG_BXVRx)

DBG breakpoint extended value x register

偏移地址：0x250 + 0x4 * x（x = 0 到 1）

复位值：0x0000 0000

保存用于支持上下文 ID 比较的 BRP 进行 VMID 匹配的值。每个 DBG_BXVR 与一个 DBG_BCR/DBG_BVR 相关联，以形成断点寄存器对 (BRP)。例如，DBG_BXVR[n] 与 DBG_BCR[n] 和 DBG_BVR[n] 相关联以形成断点 n。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VMID[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **VMID[7:0]**：虚拟机 ID (Virtual Machine ID)。
用于与虚拟化转换表基址寄存器 (VTTBR) 中的虚拟机 ID (VMID) 位字段进行比较。

DBG OS 锁定访问寄存器 (DBG_OSLAR)

DBG OS lock access register

偏移地址：0x300

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSLOCKACCESS[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSLOCKACCESS[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **OSLOCKACCESS[31:0]**：DBG 寄存器锁定访问 (DBG registers lock access)
0xC5ACCE55：禁止访问
其他值：使能访问

DBG OS 锁定状态寄存器 (DBG_OSLSR)

DBG OS lock status register

偏移地址：0x304

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSLM [1]	NTT	OSLK	OSLM [0]
												r	r	r	r

位 31:4 保留，必须保持复位值。

位 2 **NTT**：需要 32 位 (Need 32-bit)。

0：将密钥写入 OS 锁定访问寄存器需要采用 32 位访问。

位 1 **OSLK**：OS 锁定状态 (OS lock status)

0：未设置 LOCK。

1：已设置 LOCK。禁止访问 DBG 寄存器

位 3 和 0 **OSLM[1:0]**：OS 锁定模型 (OS lock model)

该位字段标识设计的 OS 保存和恢复机制形式：

10：处理器设计了 OS 锁定模型但未设计 DBGOSSRR

其他值：保留

DBG 设备掉电和复位控制寄存器 (DBG_PRCR)

DBG device power-down and reset control register

偏移地址：0x310

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CORE PURQ	HCWR	CWRR	CORE NPDR Q
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3 **COREPURQ**：上电请求 (Power-up request)

该位允许调试器请求电源控制器为
处理器上电，以便能够访问处理器电源域中的调试寄存器：

0：DBGPWRUPREQ 为低电平

1：DBGPWRUPREQ 为高电平

位 2 **HCWR**：保持复位 (Hold reset)

0：发生上电或热复位时，非调试逻辑不保持复位状态。

1：发生上电或热复位时，处理器的非调试逻辑保持复位状态，
直到该位设置为 0 为止。

位 1 **CWRR**：复位请求 (Reset request)

0：无操作

1：使用内存映射接口或 CP14 请求内部复位

位 0 **CORENPDRQ**：保持上电请求 (Hold power-up request)
0：DBGNOPWRDWN 为低电平
1：DBGNOPWRDWN 为高电平

DBG 设备掉电和复位状态寄存器 (DBG_PRSR)

DBG device power-down and reset status register

偏移地址：0x314

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DLK	OSLK	HALTED	SR	R	SPD	PU
									r	r	r	r	r	r	r

位 31:7 保留，必须保持复位值。

位 6 **DLK**：OS 双重锁定状态 (OS double lock status)
0：未设置 OS 双重锁定
1：已设置 OS 双重锁定

位 5 **OSLK**：OS 锁定状态 (OS lock status)
0：未设置 OS 锁定
1：已设置 OS 锁定

位 4 **HALTED**：停止状态 (Halted state)
0：处理器处于非调试状态
1：处理器处于调试状态

位 3 **SR**：粘滞复位 (Sticky reset)
0：自上次读取该寄存器之后，处理器的非调试逻辑尚未复位。
1：自上次读取该寄存器之后，处理器的非调试逻辑已经复位。

位 2 **R**：复位 (Reset)
0：非调试逻辑当前未处于复位状态
1：非调试逻辑当前处于复位状态

位 1 **SPD**：粘滞掉电状态 (Sticky power-down status)
0：自上次读取该寄存器之后，内核电源域尚未掉电
1：自上次读取该寄存器之后，内核电源域已经掉电

位 0 **PU**：上电状态 (Power-up status)
0：内核电源域处于掉电状态
1：内核电源域处于上电状态

DBG 处理器主 ID 寄存器 (DBG_MIDR)

DBG processor main ID register

偏移地址：0xD00

复位值：0x410F C075

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMPLEMENTER[7:0]								VARIANT[3:0]				ARCH[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PART[11:0]												REVISION[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 **IMPLEMENTER[7:0]**：设计人员代码 (Implementer code)
0x41：Arm® Limited

位 23:20 **VARIANT[3:0]**：变体（主版本号）(Variant (major revision number))
0x0：r0pn

位 19:16 **ARCH[3:0]**：架构 (Architecture)
0xF：Arm® v7

位 15:4 **PART[11:0]**：主产品编号 (Primary part number)
0xC07：Cortex-A7 MPCore

位 3:0 **REVISION[3:0]**：修订（次版本号）(Revision (minor revision number))
0x5：rnp5

DBG 处理器缓存类型寄存器 (DBG_CTR)

DBG processor cache type register

偏移地址：0xD04

复位值：0x8444 8003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FORMAT[2:0]			Res.	CWG[3:0]				ERG[3:0]				DMINLINE[3:0]			
r	r	r		r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L1IP[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IMINLINE[3:0]			
r	r											r	r	r	r

位 31:29 **FORMAT[2:0]**：CTR 格式 (CTR format)
0x4：Arm®v7 格式

位 28 保留，必须保持复位值。

位 27:24 **CWG[3:0]**：缓存回写粒度 (Cache write-back granule)
因回收修改了存储单元的缓存条目而可以覆盖的最大内存大小对应的字数。
0x4：16 字

位 23:20 **ERG[3:0]**：独占保留粒度 (Exclusives reservation granule)。
针对独占加载指令和独占存储指令设计的保留粒度最大大小对应的字数。
0x4：16 字

位 19:16 **DMINLINE[3:0]**：最小数据缓存行 (Minimum data cache line)。
处理器控制的所有数据缓存和统一缓存中最小缓存行的字数。
0x4：16 字

位 15:14 **L1IP[1:0]**：L1 指令缓存策略 (L1 instruction cache policy)
指示 L1 指令缓存的索引和标记策略。
0x2：虚拟索引物理标记 (VIPT)

位 13:4 保留，必须保持复位值。

位 3:0 **IMINLINE[3:0]**：最小指令高速缓存行 (Minimum instruction cache line)。
处理器控制的所有指令缓存中最小缓存行的字数。
0x3：8 字

DBG 处理器 TLB 类型寄存器 (DBG_TLBTR)

DBG processor TLB type register

偏移地址：0xD0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NU
															r

位 31:1 保留，必须保持复位值。

位 0 **NU**：统一 TLB (Unified TLB)。
0：处理器具有统一 TLB

DBG 处理器多处理器绑核寄存器 (DBG_MPIDR)

DBG processor multi-processor affinity register

偏移地址：0xD14

复位值：0x8000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	U	Res.	Res.	Res.	Res.	Res.	MT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	r						r								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CLUSTERID[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	CPUID[1:0]	
				r	r	r	r							r	r

位 31 保留，必须保持复位值。

位 30 **U**：单处理器系统 (Uniprocessor system)
0：处理器是多处理器系统的一部分

位 29:25 保留，必须保持复位值。

位 24 **MT**：多线程 (Multi-threading)。
指示最低级别的绑核是否包括采用多线程类型方法设计的逻辑处理器。
0：处理器是多处理器系统的一部分

位 23:12 保留，必须保持复位值。

位 11:8 **CLUSTERID[3:0]**：CLUSTERID 配置 (CLUSTERID configuration)。
指示在 CLUSTERID 配置引脚中读取的值。该值标识多处理器系统中的每个 Cortex-A7 MPCore 处理器。

位 7:2 保留，必须保持复位值。

位 1:0 **CPUID[1:0]**：处理器数量 (Processor number)。
如果有一个处理器，CPU ID 为 0x0。

DBG 版本 ID 寄存器 (DBG_REVIDR)

DBG revision ID register

偏移地址：0xD18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ID[31:0]**：特定于设计的版本信息 (Implementation-specific revision information)。

DBG 处理器特性 0 寄存器 (DBG_PFR0)

DBG processor feature 0 register

偏移地址：0xD20

复位值：0x0000 1131

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STATE3[3:0]				STATE2[3:0]				STATE1[3:0]				STATE0[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:12 **STATE3[3:0]**：支持 Thumb 执行环境 (ThumbEE) 指令集 (Support for Thumb execution environment (ThumbEE) instruction set)。

1：已设计 ThumbEE 指令集

位 11:8 **STATE2[3:0]**：支持 Jazelle 扩展 (Support for Jazelle extension)。

1：处理器支持 Jazelle 扩展的简单设计

位 7:4 **STATE1[3:0]**：支持 Thumb 指令集 (Support for Thumb instruction set)。

3：处理器在引入 Thumb-2 技术后支持 Thumb 编码，适用于所有 16 位和 32 位 Thumb 基本指令。

位 3:0 **STATE0[3:0]**：支持 Arm® 指令集 (Support for Arm® instruction set)。

1：处理器支持 Arm® 指令集

DBG 处理器特性 1 寄存器 (DBG_PFR1)

DBG processor feature 1 register

偏移地址：0xD24

复位值：0x0001 1011

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GENTIMEXT[3:0]			
												r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VIRTEXT[3:0]				MPPM[3:0]				SECEXT[3:0]				PROGMOD[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:20 保留，必须保持复位值。

位 19:16 **GENTIMEXT[3:0]**：通用定时器 (Generic timer)。

1：处理器支持通用定时器

位 15:12 **VIRTEXT[3:0]**：虚拟化扩展 (Virtualisation extensions)。

1：处理器支持虚拟化扩展

位 11:8 **MPPM[3:0]**：M 配置文件程序员模型 (M profile programmers model)。

0：处理器不支持微处理器程序员模型

位 7:4 **SECEXT[3:0]**：安全扩展 (Security extensions)。

1：处理器支持安全扩展

位 3:0 **PROGMOD[3:0]**：程序员模型 (Programmers model)

1：处理器支持 Arm®v4 及更高版本的标准程序员模型

DBG 特性 0 寄存器 (DBG_DFR0)

DBG feature 0 register

偏移地址：0xD28

复位值：0x0201 0555

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	PMUEXT[3:0]				MPDBGMOD[3:0]				MMTMOD[3:0]			
				r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTMOD[3:0]				MMDBGMOD[3:0]				CSDBGMOD[3:0]				CDBGMOD[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27:24 **PMUEXT[3:0]**：性能监视模型 (Performance monitor model)。
2：处理器支持性能监视单元版本 2 (PMUv2) 架构

位 23:20 **MPDBGMOD[3:0]**：调试模型，M 配置文件 (Debug model, M profile)。
0：处理器不支持 M 配置文件调试架构

位 19:16 **MMTMOD[3:0]**：内存映射跟踪模型 (Memory mapped trace model)。
1：处理器支持 Arm® 跟踪架构，内存映射访问

位 15:12 **CTMOD[3:0]**：协处理器跟踪模型 (Coprocessor trace model)
0：处理器不支持 Arm® 跟踪架构，CP14 访问

位 11:8 **MMDBGMOD[3:0]**：内存映射调试模型 (Memory mapped debug model)。
5：处理器支持 v7.1 调试架构，内存映射访问

位 7:4 **CSDBGMOD[3:0]**：协处理器安全调试模型 (Coprocessor secure debug model)。
1：处理器支持 v7.1 调试架构，CP14 访问

位 3:0 **CDBGMOD[3:0]**：协处理器调试模型 (Coprocessor debug model)。
1：处理器支持 v7.1 调试架构，CP14 访问

DBG 处理器内存模型特性 0 寄存器 (DBG_MMFR0)

DBG processor memory model feature 0 register

偏移地址：0xD30

复位值：0x1010 1105

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INNERSH[3:0]				FCSE[3:0]				AUXREGS[3:0]				TCM[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SHLVLS[3:0]				OUTERSH[3:0]				PMSA[3:0]				VMSA[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **INNERSH[3:0]**：最内层可共享性 (Innermost shareability)。
1：处理器设计了硬件一致性支持

位 27:24 **FCSE[3:0]**：快速上下文切换扩展 (Fast context switch extension)。
0：处理器不支持 FCSE

位 23:20 **AUXREGS[3:0]**：辅助寄存器 (Auxiliary registers)。
1：处理器只支持辅助控制寄存器

位 19:16 **TCM[3:0]**：TCM 和相关 DMA (TCM and associated DMA)。
0：处理器不支持 TCM

位 15:12 **SHLVLS[3:0]**：可共享性级别 (Shareability levels)
1：处理器设计了两个级别的可共享性

位 11:8 **OUTERSH[3:0]**：最外层可共享性 (Outermost shareability)
1：处理器支持硬件一致性

位 7:4 **PMSA[3:0]**：受保护内存系统架构 (Protected memory system architecture)。
1：处理器不支持 PMSA

位 3:0 **VMSA[3:0]**：虚拟内存系统架构 (Virtual memory system architecture)。
1：处理器支持：支持重映射和访问标志的 VMSAv7；第一级描述符内的永不特权执行 (PXN)；64 位地址转换描述符

DBG 处理器内存模型特性 1 寄存器 (DBG_MMFR1)

DBG processor memory model feature 1 register

偏移地址：0xD34

复位值：0x4000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCHPRED[3:0]				L1CTSTCLN[3:0]				L1UNIC[3:0]				L1HVDC[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L1UNICSW[3:0]				L1HVDCSW[3:0]				L1UNICVA[3:0]				L1HVDCVA[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **BCHPRED[3:0]**：分支预测器管理要求 (Branch predictor management requirements)。
4：分支预测器在任何时候都不需要清空

位 27:24 **L1CTSTCLN[3:0]**：支持 L1 缓存测试和清理操作 (L1 cache test and clean operations supported)。
0：不支持

位 23:20 **L1UNIC[3:0]**：L1 统一缓存维护操作 (L1 unified cache maintenance operations)。
0：不支持

位 19:16 **L1HVDC[3:0]**：L1 哈佛缓存维护操作 (L1 Harvard cache maintenance operations)。
0：不支持

位 15:12 **L1UNICSW[3:0]**：L1 统一缓存组/路行维护操作 (L1 unified cache set/way line maintenance operations)。
0：不支持

位 11:8 **L1HVDCSW[3:0]**：L1 哈佛缓存组/路行维护操作 (L1 Harvard cache set/way line maintenance operations)。
0：不支持

位 7:4 **L1UNICVA[3:0]**：通过 MVA 进行的 L1 统一缓存行维护操作 (L1 unified cache line maintenance operations by MVA)
0：不支持

位 3:0 **L1HVDCVA[3:0]**：通过 MVA 进行的 L1 哈佛缓存行维护操作 (L1 Harvard cache line maintenance operations by MVA)

0：不支持

DBG 处理器内存模型特性 2 寄存器 (DBG_MMFR2)

DBG processor memory model feature 2 register

偏移地址：0xD38

复位值：0x0124 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HWACFLG[3:0]				WFISTALL[3:0]				MEMBARR[3:0]				UNIFTLB[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HVDTLB[3:0]				L1HRANGE[3:0]				L1HBGFTCH[3:0]				L1HFGFTCH[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **HWACFLG[3:0]**：硬件访问标志 (Hardware access flag)。

0：不支持

位 27:24 **WFISTALL[3:0]**：等待中断停止 (Wait for interrupt stalling)。

1：处理器支持 WFI 停止

位 23:20 **MEMBARR[3:0]**：内存屏障 (Memory barrier)。

2：处理器支持：数据同步屏障 (DSB)；指令同步屏障 (ISB)；数据内存屏障 (DMB)

位 19:16 **UNIFTLB[3:0]**：统一 TLB 维护操作 (Unified TLB maintenance operations)。

4：处理器支持：使 TLB 中的所有条目失效；通过 MVA 使 TLB 条目失效；通过 ASID 匹配使 TLB 条目失效；通过 MVA 所有 ASID 使 TLB 条目失效；通过 MVA 使统一 hyp TLB 条目失效；使整个非安全非 hyp 统一 TLB 失效；使整个 hyp 统一 TLB 失效

位 15:12 **HVDTLB[3:0]**：哈佛 TLB 维护操作 (Harvard TLB maintenance operations)。

0：不支持

位 11:8 **L1HRANGE[3:0]**：L1 哈佛缓存维护操作范围 (L1 Harvard cache maintenance operation range)

0：不支持

位 7:4 **L1HBGFTCH[3:0]**：L1 哈佛缓存后台预取操作 (L1 Harvard cache background prefetch operations)

0：不支持

位 3:0 **L1HFGFTCH[3:0]**：L1 哈佛缓存前台预取操作 (L1 Harvard cache foreground prefetch operations)

DBG 处理器内存模型特性 3 寄存器 (DBG_MMFR3)

DBG processor memory model feature 3 register

偏移地址：0xD3C

复位值：0x0210 2211

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SUPERSECT[3:0]				CMEMSIZE[3:0]				COHWALK[3:0]				Res.	Res.	Res.	Res.
r	r	r	r	r	r	r	r	r	r	r	r				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAINTBCST[3:0]				BPMAINT[3:0]				CMAINTSW[3:0]				CMAINTMVA[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **SUPERSECT[3:0]**：支持超级段 (Supersection support)。
0：处理器支持超级段

位 27:24 **CMEMSIZE[3:0]**：缓存物理内存大小 (Cache physical memory size)。
2：缓存支持 40 位内存地址

位 23:20 **COHWALK[3:0]**：一致性遍历 (Coherent walk)。
1：更新转换表时不需要清理到统一点，以确保后续转换表遍历的可见性。

位 19:16 保留，必须保持复位值。

位 15:12 **MAINTBCST[3:0]**：维护广播 (Maintenance broadcast)
2：缓存、TLB 和分支预测器操作根据可共享性和指令定义的行为来作用于结构。

位 11:8 **BPMAINT[3:0]**：分支预测器维护 (Branch predictor maintenance)。
2：处理器支持：使整个分支预测器数组失效；通过 MVA 使分支预测器失效。

位 7:4 **CMAINTSW[3:0]**：按组/路进行缓存维护 (Cache maintenance by set/way)。
1：处理器支持：按组/路使数据缓存失效；按组/路清理数据缓存；按组/路清理数据缓存并使其失效。

位 3:0 **CMAINTMVA[3:0]**：通过 MVA 进行缓存维护 (Cache maintenance by MVA)
1：处理器支持：通过 MVA 使数据缓存失效；通过 MVA 清理数据缓存；通过 MVA 清理数据缓存并使其失效；通过 MVA 使指令缓存失效；使所有指令缓存条目失效。

DBG 处理器指令集属性 0 寄存器 (DBG_ISAR0)

DBG processor instruction set attribute 0 register

偏移地址：0xD40

复位值：0x0210 1110

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	DIVINST[3:0]				DBGINST[3:0]				COPROINST[3:0]			
				r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPBRINST[3:0]				BITFLDINST[3:0]				BITCNTINST[3:0]				SWAPINST[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27:24 **DIVINST[3:0]**：除法指令 (Divide instructions)。
2：处理器支持：Thumb 指令集中的 SDIV 和 UDIV； Arm® 指令集中的 SDIV 和 UDIV。

位 23:20 **DBGINST[3:0]**：调试指令 (Debug instructions)。
1：处理器支持 BKPT 指令

位 19:16 **COPROINST[3:0]**：协处理器指令 (Coprocessor instructions)。
0：不支持，除非归属独立架构，包括 CP15、CP14、高级 SIMD 和 VFP。

位 15:12 **CMPBRINST[3:0]**：比较和分支组合指令 (Combined compare and branch instructions)。
1：处理器支持 CBNZ 和 CBZ 指令

位 11:8 **BITFLDINST[3:0]**：位字段指令 (Bit field instructions)。
1：处理器支持 BFC、BFI、SBFX 和 UBFX 指令

位 7:4 **BITCNTINST[3:0]**：位计数指令 (Bit count instructions)。
1：处理器支持 CLZ 指令

位 3:0 **SWAPINST[3:0]**：交换指令 (Swap instructions)。
0：支持一部分 SWP 和 SWPB 指令

DBG 处理器指令集属性 1 寄存器 (DBG_ISAR1)

DBG processor instruction set attribute 1 register

偏移地址：0xD44

复位值：0x1311 2111

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JAZINST[3:0]				INTWKINST[3:0]				IMMINST[3:0]				IFTHENINST[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTNDINST[3:0]				EXCARINST[3:0]				EXCINST[3:0]				ENDIANINST[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **JAZINST[3:0]**：Jazelle 扩展指令 (Jazelle extension instructions)。
1：处理器支持：BXJ 指令以及 PSR 中的 J 位

位 27:24 **INTWKINST[3:0]**：交互工作指令 (Interworking instructions)。
3：处理器支持：BX 指令以及 PSR 中的 T 位；BLX 指令，PC 加载行为类似于 BX；Arm® 指令集中的数据处理指令，以 PC 为目标，S 位清零，行为类似于 BX

位 23:20 **IMMINST[3:0]**：立即数指令 (Immediate instructions)。
1：处理器支持：MOVT 指令；MOV 指令编码（零扩展的 16 位立即数）；Thumb ADD 和 SUB 指令编码（零扩展的 12 位立即数），以及其他 ADD、ADR 和 SUB 编码（通过这些编码的伪代码进行交叉引用）。

位 19:16 **IFTHENINST[3:0]**：Thumb If-then 指令 (Thumb If-then instructions)。
1：处理器支持 IT 指令以及 PSR 中的 IT 位

位 15:12 **EXTNDINST[3:0]**：扩展指令 (Extend instructions)。
2：处理器支持：SXTB、SXTH、UXTB 和 UXTH 指令；SXTB16、SXTAB、SXTAB16、SXTAH、UXTB16、UXTAB、UXTAB16 和 UXTAH 指令

位 11:8 **EXCARINST[3:0]**：A 和 R 配置文件异常处理指令。
1：处理器支持 SRS、RFE 和 CPS 指令

位 7:4 **EXCINST[3:0]**：异常处理指令 (Exception-handling instructions)。
1：处理器支持 LDM（异常返回）、LDM（用户寄存器）和 STM（用户寄存器）指令

位 3:0 **ENDIANINST[3:0]**：字节序指令 (Endian instructions)。
1：处理器支持 SETEND 指令以及 PSR 中的 E 位

DBG 处理器指令集属性 2 寄存器 (DBG_ISAR2)

DBG processor instruction set attribute 2 register

偏移地址：0xD48

复位值：0x2123 2041

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REVSALINST[3:0]				PSRINST[3:0]				MULTUINST[3:0]				MULTSINST[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MULTINST[3:0]				MULACINST[3:0]				MEMHTINST[3:0]				LDSTRINST[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **REVSALINST[3:0]**：反转指令 (Reversal instructions)。
2：处理器支持 REV、REV16、REVSH 和 RBIT 指令

位 27:24 **PSRINST[3:0]**：PSR 指令 (PSR instructions)。
1：处理器支持 MRS 和 MSR 指令，以及异常返回形式的数据处理指令

位 23:20 **MULTUINST[3:0]**：高级无符号乘法指令 (Advanced unsigned multiply instructions)。
2：处理器支持 UMULL、UMLAL 和 UMAAL 指令

位 19:16 **MULTSINST[3:0]**：高级有符号乘法指令 (Advanced signed multiply instructions)。
3：处理器支持：SMULL 和 SMLAL 指令；SMLABB、SMLABT、SMLALBB、SMLALBT、SMLALTB、SMLALTT、SMLATB、SMLATT、SMLAWB、SMLAWT、SMULBB、SMULBT、SMULTB、SMULTT、SMULWB、SMULWT 指令，以及
PSR 中的 Q 位：SMLAD、SMLADX、SMLALD、SMLALDX、SMLSD、SMLSDX、SMLSLD、SMLSLDX、SMMLA、SMMLAR、SMMLS、SMMLSR、SMMUL、SMMULR、SMUAD、SMUADX、SMUSD 和 SMUSDX 指令

位 15:12 **MULTINST[3:0]**：附加乘法指令 (Additional multiply instructions)。
2：处理器支持 MUL、MLA 和 MLS 指令

位 11:8 **MULACINST[3:0]**：可中断的多重访问指令 (Interruptible multi-access instructions)。
0：不支持。这意味着处理器不支持可中断的 LDM 和 STM 指令

位 7:4 **MEMHTINST[3:0]**：内存提示指令 (Memory hint instructions)。
4：处理器支持 PLD、PLI (NOP) 和 PLDW 指令

位 3:0 **LDSTRINST[3:0]**：附加加载/存储指令 (Additional load/store instructions)。
1：处理器支持 LDRD 和 STRD 指令

DBG 处理器指令集属性 3 寄存器 (DBG_ISAR3)

DBG processor instruction set attribute 3 register

偏移地址：0xD4C

复位值：0x1111 2131

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEEXTINST[3:0]				TRNOPINST[3:0]				THCPYINST[3:0]				TABBRINST[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNPRIINST[3:0]				SVCINST[3:0]				SIMDINST[3:0]				SATINST[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **TEEXTINST[3:0]**：Thumb 执行环境 (ThumbEE) 指令 (Thumb Execution Environment (ThumbEE) instructions)。
1：处理器支持 ENTERX 和 LEAVEX 指令，并修改了加载行为以包含为空检查

位 27:24 **TRNOPINST[3:0]**：真正 NOP 指令 (True NOP instructions)。

1：处理器支持 Arm® 和 Thumb 指令集中的真正 NOP 指令，以及其他兼容 NOP 的提示

位 23:20 **THCPYINST[3:0]**：Thumb 复制指令 (Thumb copy instructions)。

1：处理器支持 MOV（寄存器）指令的 Thumb 指令集编码 T1，从低位寄存器复制到低位寄存器

位 19:16 **TABBRINST[3:0]**：表分支指令 (Table branch instructions)。

1：处理器支持 TBB 和 TBH 指令

位 15:12 **SYNPRIINST[3:0]**：同步原语指令 (Synchronisation primitive instructions)。

2：处理器支持：LDREX 和 STREX 指令；CLREX、LDREXB、LDREXH、STREXB 和 STREXH 指令；LDREXD 和 STREXD 指令

位 11:8 **SVCINST[3:0]**：SVC 指令 (SVC instructions)。

1：处理器支持 SVC 指令

位 7:4 **SIMDINST[3:0]**：单指令多数据 (SIMD) 指令 (Single instruction multiple data (SIMD) instructions)。

3：处理器支持：SSAT 和 USAT 指令，以及 PSR 中的 Q 位；PKHBT、PKHTB、QADD16、QADD8、QASX、QSUB16、QSUB8、QSAX、SADD16、SADD8、SASX、SEL、SHADD16、SHADD8、SHASX、SHSUB16、SHSUB8、SHSAX、SSAT16、SSUB16、SSUB8、SSAX、SXTAB16、SXTB16、UADD16、UADD8、UASX、UHADD16、UHADD8、UHASX、UHSUB16、UHSUB8、UHSAX、UQADD16、UQADD8、UQASX、UQSUB16、UQSUB8、UQSAX、USAD8、USADA8、USAT16、USUB16、USUB8、USAX、UXTAB16、UXTB16 指令，以及 PSR 中的 GE[3:0] 位

位 3:0 **SATINST[3:0]**：饱和指令 (Saturate instructions)。

1：处理器支持 QADD、QDADD、QDSUB、QSUB 以及 PSR 中的 Q 位

DBG 处理器指令集属性 4 寄存器 (DBG_ISAR4)

DBG processor instruction set attribute 4 register

偏移地址：0xD50

复位值：0x1001 1142

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWPFRAC[3:0]				PSRMINST[3:0]				SPINSFRAC[3:0]				BARRINST[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMCINST[3:0]				WRBKINST[3:0]				WSHFTINST[3:0]				UNPRIVINST[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **SWPFRAC[3:0]**：支持内存系统为 SWP 或 SWPB 指令锁定总线 (Support for the memory system locking the bus for SWP or SWPB instructions)。

1：处理器支持 SWP 和 SWPB 指令，但仅限在单处理器上下文中。SWP 和 SWPB 不保证其他主设备是否可以在 SWP 或 SWPB 指令的加载内存访问与存储内存访问之间进行内存访问

位 27:24 **PSRMINST[3:0]**：用于修改 PSR 的 M 配置文件指令 (M profile instructions to modify the PSRs)。

0：不支持

位 23:20 **SPINSFRAC[3:0]**：同步原语指令 (Synchronisation primitive instructions)。

0：处理器支持：LDREX 和 STREX 指令；CLREX、LDREXB、LDREXH、STREXB 和 STREXH 指令；LDREXD 和 STREXD 指令

位 19:16 **BARRINST[3:0]**：屏障指令 (Barrier instructions)。
1：处理器支持 DMB、DSB 和 ISB 指令

位 15:12 **SMCINST[3:0]**：SMC 指令 (SMC instructions)。
1：处理器支持 SMC 指令

位 11:8 **WRBKINST[3:0]**：回写寻址模式 (Write-back addressing modes)。
1：处理器支持 Arm®v7 架构中定义的所有回写寻址模式

位 7:4 **WSHFTINST[3:0]**：带移位的指令 (Instructions with shifts)。
4：处理器支持：LSL 0-3 范围内的加载和存储操作的移位；加载/存储和其他指令上的常量移位选项；寄存器控制的移位选项。

位 3:0 **UNPRIVINST[3:0]**：非特权指令 (Unprivileged instructions)。
2：处理器支持：LDRBT、LDRT、STRBT 和 STRT 指令；LDRHT、LDRSBT、LDRSHT 和 STRHT 指令。

DBG 声明标记设置寄存器 (DBG_CLAIMSET)

DBG claim tag set register

偏移地址：0xFA0

复位值：0x0000 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SET[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **SET[7:0]**：设置声明标记位 (Set claim tag bits)
写：
00000000：无影响
xxxxxxx1：将位 0 置 1
xxxxxx1x：将位 1 置 1
xxxxx1xx：将位 2 置 1
以此类推。
读：
0xFF：表示声明标记使用八个位

DBG 声明标记清零寄存器 (DBG_CLAIMCLR)

DBG claim tag clear register

偏移地址：0xFA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **CLR[7:0]**：复位声明标记位 (Reset claim tag bits)

写：

00000000：无影响

xxxxxxx1：将位 0 清零

xxxxxx1x：将位 1 清零

xxxxx1xx：将位 2 清零

xxxx1xxx：将位 3 清零

以此类推。

读：返回声明标记的当前值

DBG 锁定访问寄存器 (DBG_LAR)

DBG lock access register

偏移地址：0xFB0

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **KEY[31:0]**：DBG 寄存器写访问 (DBG register write access)

使能处理器内核写访问某些 DBG 寄存器（调试器无需解锁组件）

0xC5ACCE55：使能写访问

其他值：禁止写访问

DBG 锁定状态寄存器 (DBG_LSR)

DBG lock status register

偏移地址：0xFB4

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NTT	SLK	SLI
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **NTT**：DBG_LAR 寄存器的大小 (Size of the DBG_LAR register)
0：32 位

位 1 **SLK**：当前锁定状态 (Current status of lock)
当外部调试器读取该位时，该位始终返回零。
0：允许写访问
1：阻止写访问。仅允许读访问。

位 0 **SLI**：存在锁定控制机制 (Existence of lock control mechanism)
该位指示是否存在锁定控制机制。当外部调试器读取该位时，该位始终返回零。
0：不存在锁定控制机制
1：存在锁定控制机制

DBG 认证状态寄存器 (DBG_AUTHSTAT)

DBG authentication status register

偏移地址：0xFB8

复位值：0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNI	SNE	SI	SE	NSNI	NSNE	NSI	NSE
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7 **SNI**：安全非侵入式调试 (Secure non-invasive debug)
1：已设计安全非侵入式调试功能

位 6 **SNE**：使能安全非侵入式调试 (Secure non-invasive debug enabled)
0：不允许在安全 PL1 模式下进行非侵入式调试
1：允许在安全 PL1 模式下进行非侵入式调试

位 5 **SI**：安全侵入式调试 (Secure invasive debug)
1：已设计安全侵入式调试功能

位 4 **SE**：使能安全侵入式调试 (Secure invasive debug enabled)
0：不允许在安全 PL1 模式下进行侵入式暂停调试
1：允许在安全 PL1 模式下进行侵入式暂停调试

位 3 **NSNI**：非安全非侵入式调试 (Non-secure non-invasive debug)
1：已设计非安全非侵入式调试功能

位 2 **NSNE**：使能非安全非侵入式调试 (Non-secure non-invasive debug enabled)
0：不允许进行非安全非侵入式调试
1：允许进行非安全非侵入式调试

位 1 **NSI**：非安全侵入式调试 (Non-secure invasive debug)
1：已设计非安全侵入式调试功能

位 0 **NSE**：使能非安全侵入式调试 (Non-secure invasive debug enabled)
0：不允许进行非安全侵入式调试
1：允许进行非安全侵入式调试

DBG CoreSight 设备 ID 1 寄存器 (DBG_DEVID1)

DBG CoreSight device ID 1 register

偏移地址：0xFC4

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCSROFFSET[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PCSROFFSET[7:0]**：DBGPCSR 偏移 (DBGPCSR offset)
0x1：未应用偏移

DBG Coresight 设备配置寄存器 (DBG_DEVID)

DBG Coresight device configuration register

偏移地址：0xFC8

复位值：0x0111 0F13

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CIDMASK[3:0]				AUXREGS[3:0]				DOUBLELOCK[3:0]				VIREXTNS[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VECTORCATCH[3:0]				BPADDRMASK[3:0]				WPADDRMASK[3:0]				PCSAMPLE[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 **CIDMASK[3:0]**：上下文 ID 匹配断点屏蔽 (Context ID matching breakpoint masking)
0x0：未设计

位 27:24 **AUXREGS[3:0]**：调试外部辅助控制寄存器 (Debug external auxiliary control register)。
0x1：处理器支持调试外部辅助控制寄存器

位 23:20 **DOUBLELOCK[3:0]**：调试 OS 双重锁定寄存器 (Debug OS double lock register)。
0x1：处理器支持调试 OS 双重锁定寄存器

位 19:16 **VIREXTNS[3:0]**：虚拟化扩展 (Virtualisation extensions)。
0x1：处理器设计了对调试架构的虚拟化扩展

位 15:12 **VECTORCATCH[3:0]**：向量捕获事件 (Vector catch event)。
0x0：处理器设计了地址匹配形式的向量捕获

位 11:8 **BPADDRMASK[3:0]**：立即数虚拟地址 (IVA) 匹配断点屏蔽 (Immediate Virtual Address (IVA) matching breakpoint masking)。
0xF：未设计

位 7:4 **WPADDRMASK[3:0]**：DVA 匹配监测点屏蔽 (DVA matching watchpoint masking)。
0x1：已设计监测点地址屏蔽

位 3:0 **PCSAMPLE[3:0]**：使用调试寄存器 40、41 和 42 进行程序计数器采样 (Program Counter sampling using debug registers 40, 41, and 42)。
0x3：DBGPCSR、DBGCIDSR 和 DBGVIDSR 用作调试寄存器 40、41 和 42

DBG CoreSight 设备类型寄存器 (DBG_DEVTYPE)

DBG CoreSight device type register

偏移地址：0xFCC

复位值：0x0000 0015

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SUBTYPE[3:0]**：设备子类型标识符 (Device sub-type identifier)
0x1：处理器

位 3:0 **MAJORTYPE[3:0]**：设备主类型标识符 (Device main type identifier)
0x5：调试逻辑

DBG CoreSight 外设标识寄存器 4 (DBG_PIDR4)

DBG CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SIZE[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：Arm® JEDEC 代码

DBG CoreSight 外设标识寄存器 0 (DBG_PIDR0)

DBG CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0007

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位字段 (Part number field)，位 [7:0]
0x07：DBG 产品编号

DBG CoreSight 外设标识寄存器 1 (DBG_PIDR1)

DBG CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00BC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位字段 (JEP106 identity code field)，位 [3:0]
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位字段 (Part number field)，位 [11:8]
0xC：DBG 产品编号

DBG CoreSight 外设标识寄存器 2 (DBG_PIDR2)

DBG CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 005B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x5：r0p5

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位字段 (JEP106 identity code field)，位 [6:4]
0x3：Arm® JEDEC 代码

DBG CoreSight 外设标识寄存器 3 (DBG_PIDR3)

DBG CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

DBG CoreSight 组件标识寄存器 0 (DBG_CIDR0)

DBG CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位字段 (Component ID field)，位 [7:0]
0x0D：公共 ID 值

DBG CoreSight 组件标识寄存器 1 (DBG_CIDR1)

DBG CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位字段 (Component ID field)，位 [15:12]——组件类
0x9：调试组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位字段 (Component ID field)，位 [11:8]
0x0：公共 ID 值

DBG CoreSight 组件标识寄存器 2 (DBG_CIDR2)

DBG CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位字段 (Component ID field)，位 [23:16]
0x05：公共 ID 值

DBG CoreSight 组件标识寄存器 3 (DBG_CIDR3)

DBG CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位字段 (Component ID field)，位 [31:24]
0xB1：公共 ID 值

调试单元寄存器映射

表 616. 调试单元寄存器映射和复位值

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**DBG_DIDR**	WRPS[3:0]				BRPS[3:0]				CTX_CMPS [3:0]				VERSION [3:0]				DEVIDIMP	NSUHDIMP	PCSRIMP	SEIMP.	Res.	Res.	Res.	Res.	VARIANT [3:0]				REVISION [3:0]			
	Reset value	0	0	1	1	0	1	0	1	0	0	0	1	0	1	0	1	1	1	1	1					0	0	0	0	0	1	0	1
0x018	**DBG_WFAR**	ADDRESS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x01C	**DBG_VCR**	NSF	NSI	Res.	NSD	NSP	NSC	NSU	Res.	NSHF	NSHI	NSHE	NSHD	NSHP	NSHC	NSHU	Res.	MF	MI	Res.	MD	MP	MS	Res.	Res.	SF	SI	Res.	SD	SP	SS	SU	R
	Reset value	0	0		0	0	0	0		0	0	0	0	0	0	0		0	0		0	0	0			0	0		0	0	0	0	0
0x024	**DBG_ECR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSUCE
	Reset value																																0
0x080	**DBG_DTRRX**	DATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	**DBG_ITR**	INSTRUCTION[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x088	**DBG_DSCR**	Res.	RXFULL	TXFULL	Res.	RXFULL_L	TXFULL_L	PIPEADV	INSTRCOMPL_L	Res.	Res.	EXTDCCMODE[1:0]		ADADISCARD	NS	SPNIDDIS	SPIDDIS	MDBGEN	HDBGEN	ITREN	UDCCDIS	INTDIS	DBGACK	FS	UND_I	ADABORT_I	SDABORT_I	MOE[3:0]				RESTARTED	HALTED
	Reset value		0	0		0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08C	**DBG_DTRTX**	DATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 616. 调试单元寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x090	**DBG_DRCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CBRRQ	CSPA	CSE	RRQ	HRQ
	Reset value																												0	0	0	0	0
0x094	**DBG_DEACR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CDRSTAT	Res.	Res.	Res.
	Reset value																													0			
0x0A0	**DBG_PCSR**	PCS[31:1]																															T
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0A4	**DBG_CIDSR**	CONTEXTIDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0A8	**DBG_VIDSR**	NS	H	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VMID[7:0]							
	Reset value	0	0																							0	0	0	0	0	0	0	0
0x100	**DBG_BVR0**	ADDRESSCONTEXTID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x104	**DBG_BVR1**	ADDRESSCONTEXTID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x108	**DBG_BVR2**	ADDRESSCONTEXTID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10C	**DBG_BVR3**	ADDRESSCONTEXTID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x110	**DBG_BVR4**	ADDRESSCONTEXTID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x114	**DBG_BVR5**	ADDRESSCONTEXTID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x140	**DBG_BCR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BT[3:0]				LBN[3:0]				SSC [1:0]		HMC	Res.	Res.	Res.	Res.	BAS[3:0]				Res.	Res.	PMC [1:0]		E
	Reset value									0	0	0	0	0	0	0	0	0	0	0					0	0	0	0			0	0	0
0x144	**DBG_BCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BT[3:0]				LBN[3:0]				SSC [1:0]		HMC	Res.	Res.	Res.	Res.	BAS[3:0]				Res.	Res.	PMC [1:0]		E
	Reset value									0	0	0	0	0	0	0	0	0	0	0					0	0	0	0			0	0	0
0x148	**DBG_BCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BT[3:0]				LBN[3:0]				SSC [1:0]		HMC	Res.	Res.	Res.	Res.	BAS[3:0]				Res.	Res.	PMC [1:0]		E
	Reset value									0	0	0	0	0	0	0	0	0	0	0					0	0	0	0			0	0	0
0x14C	**DBG_BCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BT[3:0]				LBN[3:0]				SSC [1:0]		HMC	Res.	Res.	Res.	Res.	BAS[3:0]				Res.	Res.	PMC [1:0]		E
	Reset value									0	0	0	0	0	0	0	0	0	0	0					0	0	0	0			0	0	0
0x150	**DBG_BCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BT[3:0]				LBN[3:0]				SSC [1:0]		HMC	Res.	Res.	Res.	Res.	BAS[3:0]				Res.	Res.	PMC [1:0]		E
	Reset value									0	0	0	0	0	0	0	0	0	0	0					0	0	0	0			0	0	0
0x154	**DBG_BCR5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BT[3:0]				LBN[3:0]				SSC [1:0]		HMC	Res.	Res.	Res.	Res.	BAS[3:0]				Res.	Res.	PMC [1:0]		E
	Reset value									0	0	0	0	0	0	0	0	0	0	0					0	0	0	0			0	0	0

表 616. 调试单元寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x180	**DBGWVR0**	ADDRESS[31:0]																														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x184	**DBGWVR1**	ADDRESS[31:0]																														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x188	**DBGWVR2**	ADDRESS[31:0]																														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x18C	**DBGWVR3**	ADDRESS[31:0]																														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x1C0	**DBG_WCR0**	Res.	Res.	Res.	MASK[4:0]					Res.	Res.	Res.	WT	LBN[3:0]				SSC [1:0]		HMC	Res.	Res.	Res.	Res.	BAS[3:0]				LSC [1:0]		PAC [1:0]		E
	Reset value				0	0	0	0	0				0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x1C4	**DBG_WCR1**	Res.	Res.	Res.	MASK[4:0]					Res.	Res.	Res.	WT	LBN[3:0]				SSC [1:0]		HMC	Res.	Res.	Res.	Res.	BAS[3:0]				LSC [1:0]		PAC [1:0]		E
	Reset value				0	0	0	0	0				0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x1C8	**DBG_WCR2**	Res.	Res.	Res.	MASK[4:0]					Res.	Res.	Res.	WT	LBN[3:0]				SSC [1:0]		HMC	Res.	Res.	Res.	Res.	BAS[3:0]				LSC [1:0]		PAC [1:0]		E
	Reset value				0	0	0	0	0				0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x1CC	**DBG_WCR3**	Res.	Res.	Res.	MASK[4:0]					Res.	Res.	Res.	WT	LBN[3:0]				SSC [1:0]		HMC	Res.	Res.	Res.	Res.	BAS[3:0]				LSC [1:0]		PAC [1:0]		E
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x250	**DBG_BXVR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VMID[7::0]							
	Reset value																									0	0	0	0	0	0	0	0
0x254	**DBG_BXVR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VMID[7::0]							
	Reset value																									0	0	0	0	0	0	0	0
0x300	**DBG_OSLAR**	OSLOCKACCESS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x304	**DBG_OSLSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	0SLM[1]	NTT	0SLK	0SLM[0]
	Reset value																													0	0	0	0
0x310	**DBG_PRCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COREPURQ	HCWR	CWRR	CORENPDRQ
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x314	**DBG_PRSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DLK	OSLK	HALTED	SR	R	SPD	PU
	Reset value																										0	0	0	0	0	0	0
0xD00	**DBG_MIDR**	IMPLEMENTER[7:0]								VARIANT [3:0]				ARCH[3:0]				PART[11:0]												REVISION [3:0]			
	Reset value	0	1	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	0	1	0	1
0xD04	**DBG_CTR**	FORMAT [2:0]			Res.	CWG[3:0]				ERG[3:0]				DMINLINE [3:0]				L1IP [1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IMINLINE [3:0]			
	Reset value	1	0	0		0	1	0	0	0	1	0	0	0	1	0	0	1	0											0	0	1	1
0xD0C	**DBG_TLBTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NU
	Reset value																																0

表 616. 调试单元寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xD14	**DBG_MPIDR**	Res.	U	Res.	Res.	Res.	Res.	Res.	MT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLUSTERID [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	CPUID [1:0]	
	Reset value		0						0													0	0	0	0							0	0
0xD18	**DBG_REVIDR**	ID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD20	**DBG_PFR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STATE3[3:0]				STATE2[3:0]				STATE1[3:0]				STATE0[3:0]			
	Reset value																	0	0	0	1	0	0	1	0	0	0	1	1	0	0	0	1
0xD24	**DBG_PFR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GENTIMEXT [3:0]				VIRTEXT [3:0]				MPPM [3:0]				SECEXT [3:0]				PROGMOD [3:0]			
	Reset value													0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1
0xD28	**DBG_DFR0**	Res.	Res.	Res.	Res.	PMUEXT [3:0]				MPDBGMOD[3:0]				MMTMOD [3:0]				CTMOD [3:0]				MMDBGMOD [3:0]				CSDBGMOD [3:0]				CDBGMOD [3:0]			
	Reset value					0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1
0xD30	**DBG_MMFR0**	INNERSH [3:0]				FCSE [3:0]				AUXREGS [3:0]				TCM [3:0]				SHLVLS [3:0]				OUTERSH [3:0]				PMSA [3:0]				VMSA [3:0]			
	Reset value	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	1
0xD34	**DBG_MMFR1**	BCHPRED [3:0]				L1CTSTCLN [3:0]				L1UNIC[3:0]				L1HVDC[3:0]				L1UNICSW [3:0]				L1HVDCSW [3:0]				L1UNICVA [3:0]				L1HVDCVA [3:0]			
	Reset value	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD38	**DBG_MMFR2**	HWACFLG [3:0]				WFISTALL [3:0]				MEMBARR [3:0]				UNIFTLB [3:0]				HVDTLB [3:0]				L1HRANGE [3:0]				L1HBGFTCH [3:0]				L1HFGFTCH[3:0]			
	Reset value	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD3C	**DBG_MMFR3**	SUPERSECT[3:0]				CMEMSIZE [3:0]				COHWALK [3:0]				Res.	Res.	Res.	Res.	MAINTBCST [3:0]				BPMAINT [3:0]				CMAINTSW [3:0]				CMAINTMVA [3:0]			
	Reset value	0	0	0	0	0	0	1	0	0	0	0	1					0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1
0xD40	**DBG_ISAR0**	Res.	Res.	Res.	Res.	DIVINST [3:0]				DBGINST [3:0]				COPROINST [3:0]				CMPBRINST [3:0]				BITFLDINST [3:0]				BITCNTINST [3:0]				SWAPINST [3:0]			
	Reset value					0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
0xD44	**DBG_ISAR1**	JAZINST [3:0]				INTWKINST [3:0]				IMMINST [3:0]				IFTHENINST [3:0]				EXTNDINST [3:0]				EXCARINST [3:0]				EXCINST [3:0]				ENDIANINST[3:0]			
	Reset value	0	0	0	1	0	0	1	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1
0xD48	**DBG_ISAR2**	REVSALINST [3:0]				PSRMINST [3:0]				MULTUINST [3:0]				MULTSINST [3:0]				MULTINST [3:0]				MULACINST [3:0]				MEMHTINST[3:0]				LDSTRINST [3:0]			
	Reset value	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1
0xD4C	**DBG_ISAR3**	TEEXTINST [3:0]				TRNOPINST [3:0]				THCPYINST [3:0]				TABBRINST [3:0]				SYNPRIINST[3:0]				SVCINST [3:0]				SIMDINST [3:0]				SATINST [3:0]			
	Reset value	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1
0xD50	**DBG_ISAR4**	SWPFRAC [3:0]				PSRMINST [3:0]				SPINSFRAC [3:0]				BARRINST [3:0]				SMCINST [3:0]				WRBKINST [3:0]				WSHFTINST [3:0]				UNPRIVINST[3:0]			
	Reset value	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	0	0	1	0
0xFA0	**DBG_CLAIMSET**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SET[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0xFA4	**DBG_CLAIMCLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0xFB0	**DBG_LAR**	KEY[31:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

表 616. 调试单元寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFB4	**DBG_LSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NTT	SLK	SLI
	Reset value																													0	1	1
0xFB8	**DBG_AUTHSTAT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNI	SNE	SI	SE	NSNI	NSNE	NSI	NSE
	Reset value																									1	0	1	0	1	0	1	0
0xFC4	**DBG_DEVID1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCSROFFSET [3:0]			
	Reset value																													0	0	0	1
0xFC8	**DBG_DEVID**	CIDMASK [3:0]				AUXREGS [3:0]				DOUBLELOCK [3:0]				VIREXTNS [3:0]				VECTORCATCH [3:0]				BPADDRMASK [3:0]				WPADDRMASK [3:0]				PCSAMPLE [3:0]			
	Reset value	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1
0xFCC	**DBG_DEVTYPE**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE [3:0]				MAJORTYPE[3:0]			
	Reset value																									0	0	0	1	0	1	0	1
0xFD0	**DBG_PIDR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	**DBG_PIDR5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	**DBG_PIDR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	**DBG_PIDR7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	**DBG_PIDR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM [7:0]							
	Reset value																									0	0	0	0	0	1	1	1
0xFE4	**DBG_PIDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	1	1	0	0
0xFE8	**DBG_PIDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	1	0	1	1	0	1	1
0xFEC	**DBG_PIDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	**DBG_CIDR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	**DBG_CIDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE			
	Reset value																									1	0	0	1	0	0	0	0
0xFF8	**DBG_CIDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE							
	Reset value																									0	0	0	0	0	1	0	1

表 616. 调试单元寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFFC	DBG_CIDR3	reserved																								PREAMBLE							
	Reset value																									1	0	1	1	0	0	0	1

有关调试单元寄存器的说明，请参见《Cortex®-A7 MPCore 技术参考手册》*[6]*。

59.11.2 Cortex®-A7 性能监视单元 (PMU)

性能监视单元包含四个 32 位事件计数器和一个处理器时钟周期计数器。事件计数器可配置为在每次发生指定事件时递增。可以计数的事件示例包括：

- 缓存命中/未命中
- 数据加载/存储
- 总线访问
- 中断异常
- ETM 外部触发输出
- 监听事务
- 软件递增

当计数器回卷时，可以在 PMUIRQ 上生成中断。该中断连接到 CTI，因此可以停止处理器、开始或停止跟踪等。

大多数事件通过 PMUEVENTS 总线连接到 ETM。因此，可以使用 ETM 的资源来过滤事件。经过过滤的事件通过 ETM 的两个 EXTOUT 输出返回到 PMU。

有关 PMU 事件及其对应 PMUEVENTS 总线的列表，请参见《Cortex®-A7 MPCore 技术参考手册》*[6]*。

性能监视单元寄存器

调试器可以通过调试 APB 在 0xE00D 1000 到 0xE00D 1FFC 的地址范围内访问性能监视单元寄存器。软件可以在 0x500D 1000 到 0x500D 1FFC 的地址范围内访问这些寄存器。

此外，CPU 也可以通过 CP15 寄存器访问其自身 PMU 的寄存器。

有关 PMU 寄存器的说明，请参见《Cortex®-A7 MPCore 技术参考手册》*[6]*。

PMU 事件计数 x 寄存器 (PM_XEVCNTRx)

PMU event count x register

偏移地址：0x000 + 0x4 * x（x = 0 到 3）

复位值：0xXXXX XXXX

读取或写入所选事件计数器 PMNn（其中 n = 0..3）的值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PMN[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMN[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **PMN[31:0]**：事件计数器值 (Event counter value)。

PMU 周期计数寄存器 (PM_CCNTR)

PMU cycle count register

偏移地址：0x07C

复位值：0x0000 0000

保存处理器周期计数器的值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCNT[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CCNT[31:0]**：周期计数 (Cycle count)。

周期计数。该位字段每个处理器时钟周期或每 64 个处理器时钟周期递增一次，具体取决于 PM_CR.D 的值。

PMU 事件类型选择 x 寄存器 (PM_XEVTYPERx)

PMU event type select x register

偏移地址：0x400 + 0x4 * x（x = 0 到 3）

复位值：0x0XXX XXXX

控制哪些事件在哪些模式下触发相应的事件计数器递增。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P	U	NSK	NSU	NSH	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EVTCOUNT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31 **P**：特权执行过滤 (Privileged execution filtering)。

0：在 PL1 模式下执行时对事件计数

1：在 PL1 模式下执行时不对事件计数

位 30 **U**：非特权执行过滤 (Unprivileged execution filtering)。

0：在 PL0 模式下执行时对事件计数

1：在 PL0 模式下执行时不对事件计数

位 29 **NSK**：非安全 PL1 控制位 (Non-secure PL1 control bit)。
控制在非安全 PL1 模式下执行时的计数。行为取决于 P 位与 NSK 位的组合值：
P=NSK：在非安全 PL1 模式下执行时对事件计数
P!=NSK：在非安全 PL1 模式下执行时不对事件计数

位 28 **NSU**：非安全非特权控制位 (Non-secure unprivileged control bit)。
控制在非安全 PL0 模式下执行时的计数。行为取决于 U 位与 NSU 位的组合值：
U=NSU：在非安全 PL0 模式下执行时对事件计数
U!=NSU：在非安全 PL0 模式下执行时不对事件计数

位 27 **NSH**：非安全 PL2 使能 (Non-secure PL2 enable)。
0：在非安全 PL2 模式下执行时不对事件计数
1：在非安全 PL2 模式下执行时对事件计数

位 26:8 保留，必须保持复位值。

位 7:0 **EVTCOUNT[7:0]**：要计数的事件 (Event to count)。
由相应事件计数器 PMNn（其中 n = 0..3）计数的事件的事件编号

PMU 事件类型选择 31 寄存器 (PM_XEVTYPER31)

PMU event type select 31 register

偏移地址：0x47C

复位值：0x0XXX XXXX

控制周期计数器在哪些模式下递增。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P	U	NSK	NSU	NSH	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31 **P**：特权执行过滤 (Privileged execution filtering)。
0：在 PL1 模式下执行时对周期计数
1：在 PL1 模式下执行时不对周期计数

位 30 **U**：非特权执行过滤 (Unprivileged execution filtering)。
0：在 PL0 模式下执行时对周期计数
1：在 PL0 模式下执行时不对周期计数

位 29 **NSK**：非安全 PL1 控制位 (Non-secure PL1 control bit)。
控制在非安全 PL1 模式下执行时的计数。行为取决于 P 位与 NSK 位的组合值：
P=NSK：在非安全 PL1 模式下执行时对周期计数
P!=NSK：在非安全 PL1 模式下执行时不对周期计数

位 28 **NSU**：非安全非特权控制位 (Non-secure unprivileged control bit)。
控制在非安全 PL0 模式下执行时的计数。行为取决于 U 位与 NSU 位的组合值：
U=NSU：在非安全 PL0 模式下执行时对周期计数
U!=NSU：在非安全 PL0 模式下执行时不对周期计数

位 27 **NSH**：非安全 PL2 使能 (Non-secure PL2 enable)。
0：在非安全 PL2 模式下执行时不对周期计数
1：在非安全 PL2 模式下执行时对周期计数

位 26:0 保留，必须保持复位值。

PMU 计数使能设置寄存器 (PM_CNTENSET)

PMU count enable set register

偏移地址：0xC00

复位值：0xXXXX XXXX

使能周期计数器和事件计数器 PMN0..3。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	P3	P2	P1	P0
												rw	rw	rw	rw

位 31 **C**：使能周期计数器 (Cycle counter enable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读操作时），使能周期计数器（写操作时）

位 30:4 保留，必须保持复位值。

位 3 **P3**：使能事件计数器 3 (Event counter 3 enable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读操作时），使能事件计数器（写操作时）

位 2 **P2**：使能事件计数器 2 (Event counter 2 enable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读操作时），使能事件计数器（写操作时）

位 1 **P1**：使能事件计数器 1 (Event counter 1 enable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读操作时），使能事件计数器（写操作时）

位 0 **P0**：使能事件计数器 0 (Event counter 0 enable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读操作时），使能事件计数器（写操作时）

PMU 计数使能清除寄存器 (PM_CNTENCLR)

PMU count enable clear register

偏移地址：0xC20

复位值：0xXXXX XXXX

禁止周期计数器和事件计数器 PMN0..3。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	P3	P2	P1	P0
												rw	rw	rw	rw

位 31 **C**：禁止周期计数器 (Cycle counter disable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读操作时），禁止周期计数器（写操作时）

位 30:4 保留，必须保持复位值。

位 3 **P3**：禁止事件计数器 3 (Event counter 3 disable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读操作时），禁止事件计数器（写操作时）

位 2 **P2**：禁止事件计数器 2 (Event counter 2 disable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读操作时），禁止事件计数器（写操作时）

位 1 **P1**：禁止事件计数器 1 (Event counter 1 disable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读操作时），禁止事件计数器（写操作时）

位 0 **P0**：禁止事件计数器 0 (Event counter 0 disable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读操作时），禁止事件计数器（写操作时）

PMU 中断使能设置寄存器 (PM_INTENSET)

PMU interrupt enable set register

偏移地址：0xC40

复位值：0xXXXX XXXX

使能通过周期计数器和事件计数器 PMN0..3 生成中断请求。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	P3	P2	P1	P0
												rw	rw	rw	rw

位 31 **C**：使能周期计数器中断请求 (Cycle counter interrupt request enable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读操作时），使能在周期计数器溢出时生成中断请求（写操作时）

位 30:4 保留，必须保持复位值。

位 3 **P3**：使能事件计数器 3 中断请求 (Event counter 3 interrupt request enable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读操作时），使能在事件计数器溢出时生成中断请求（写操作时）

位 2 **P2**：使能事件计数器 2 中断请求 (Event counter 2 interrupt request enable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读操作时），使能在事件计数器溢出时生成中断请求（写操作时）

位 1 **P1**：使能事件计数器 1 中断请求 (Event counter 1 interrupt request enable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读操作时），使能在事件计数器溢出时生成中断请求（写操作时）

位 0 **P0**：使能事件计数器 0 中断请求 (Event counter 0 interrupt request enable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读操作时），使能在事件计数器溢出时生成中断请求（写操作时）

PMU 中断使能清除寄存器 (PM_INTENCLR)

PMU interrupt enable clear register

偏移地址：0xC60

复位值：0xXXXX XXXX

禁止通过周期计数器和事件计数器 PMN0..3 生成中断请求。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	P3	P2	P1	P0
												rw	rw	rw	rw

位 31 **C**：禁止周期计数器中断 (Cycle counter interrupt disable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读操作时），禁止在周期计数器溢出时生成中断请求（写操作时）

位 30:4 保留，必须保持复位值。

位 3 **P3**：禁止事件计数器 3 中断 (Event counter 3 interrupt disable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读取），禁止在事件计数器溢出时生成中断请求（写入）

位 2 **P2**：禁止事件计数器 2 中断 (Event counter 2 interrupt disable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读取），禁止在事件计数器溢出时生成中断请求（写入）

位 1 **P1**：禁止事件计数器 1 中断 (Event counter 1 interrupt disable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读取），禁止在事件计数器溢出时生成中断请求（写入）

位 0 **P0**：禁止事件计数器 0 中断 (Event counter 0 interrupt disable)。
0：禁止（读操作时），无影响（写操作时）
1：使能（读取），禁止在事件计数器溢出时生成中断请求（写入）

PMU 溢出标志状态寄存器 (PM_OVSR)

PMU overflow flag status register

偏移地址：0xC80

复位值：0xXXXX XXXX

保存周期计数器和事件计数器 PMN0..3 的溢出位状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	P3	P2	P1	P0
												rw	rw	rw	rw

位 31 **C**：周期计数器溢出 (Cycle counter overflow)。

0：未发生溢出（读操作时），无影响（写操作时）

1：发生溢出（读操作时），复位溢出位（写操作时）

位 30:4 保留，必须保持复位值。

位 3 **P3**：事件计数器 3 溢出 (Event counter 3 overflow)。

0：未发生溢出（读操作时），无影响（写操作时）

1：发生溢出（读操作时），复位溢出位（写操作时）

位 2 **P2**：事件计数器 2 溢出 (Event counter 2 overflow)。

0：未发生溢出（读操作时），无影响（写操作时）

1：发生溢出（读操作时），复位溢出位（写操作时）

位 1 **P1**：事件计数器 1 溢出 (Event counter 1 overflow)。

0：未发生溢出（读操作时），无影响（写操作时）

1：发生溢出（读操作时），复位溢出位（写操作时）

位 0 **P0**：事件计数器 0 溢出 (Event counter 0 overflow)。

0：未发生溢出（读操作时），无影响（写操作时）

1：发生溢出（读操作时），复位溢出位（写操作时）

PMU 软件递增寄存器 (PM_SWINC)

PMU software increment register

偏移地址：0xCA0

复位值：0xXXXX XXXX

使对软件递增事件（事件 0x00）进行计数的计数器递增。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	P3	P2	P1	P0
												w	w	w	w

位 31:4 保留，必须保持复位值。

位 3 **P3**：事件计数器 3 递增 (Event counter 3 increment)。

0：无影响

1：计数器递增 1

位 2 **P2**：事件计数器 2 递增 (Event counter 2 increment)。

0：无影响

1：计数器递增 1

位 1 **P1**：事件计数器 1 递增 (Event counter 1 increment)。
0：无影响
1：计数器递增 1

位 0 **P0**：事件计数器 0 递增 (Event counter 0 increment)。
0：无影响
1：计数器递增 1

PMU 配置寄存器 (PM_CFGR)

PMU configuration register

偏移地址：0xE00

复位值：0xXXX9 DF04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UEN	Res.	Res.	EX
												r			r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCD	CC	SIZE[5:0]						N[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:20 保留，必须保持复位值。

位 19 **UEN**：用户模式使能寄存器 (User-mode enable register)。
1：已设计用户模式使能寄存器

位 18:17 保留，必须保持复位值。

位 16 **EX**：导出 (Export)。
1：支持导出

位 15 **CCD**：周期计数器分频器 (Cycle counter divider)。
1：已设计周期计数器分频器

位 14 **CC**：周期计数器 (Cycle counter)。
1：已设计周期计数器

位 13:8 **SIZE[5:0]**：计数器大小 (Counter size)。
0x1F：32 位计数器

位 7:0 **N[7:0]**：事件计数器数量 (Number of event counters)。
0x04：四个事件计数器

PMU 控制寄存器 (PM_CR)

PMU control register

偏移地址：0xE04

复位值：0x4107 2000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMP[7:0]								IDCODE[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N[4:0]					Res.	Res.	Res.	Res.	Res.	DP	X	D	C	P	E
r	r	r	r	r						rw	rw	rw	w	w	rw

位 31:24 **IMP[7:0]**：设计人员代码 (Implementer code)。
0x41：Arm®

位 23:16 **IDCODE[7:0]**：标识码 (Identification code)。
0x07：Cortex-A7 MP 内核

位 15:11 **N[4:0]**：事件计数器数量 (Number of event counters)。
在安全 Hyp 模式下，该位字段返回 0x4 以指示设计的计数器数量。
在 Hyp 模式以外的非安全模式中，该位字段读取处理器 HDCR.HPMN 寄存器字段的值。

位 10:6 保留，必须保持复位值。

位 5 **DP**：在禁止区域禁止周期计数器 (Disable cycle counter in prohibited regions)。
0：在禁止区域使能计数器
1：在禁止区域禁止计数器

位 4 **X**：使能导出 (Export enable)。
该位允许通过事件总线将事件导出到另一个调试设备（例如跟踪宏单元）：
0：禁止导出事件
1：允许导出事件

位 3 **D**：时钟分频器 (Clock divider)。
0：周期计数器每个时钟周期计数一次
1：周期计数器每 64 个时钟周期计数一次

位 2 **C**：定时器计数器复位 (Clock counter reset)。
0：无影响
1：将周期计数器复位为 0

位 1 **P**：事件计数器复位 (Event counter reset)。
0：无影响
1：将所有事件计数器复位为 0

位 0 **E**：使能 (Enable)。
0：禁止所有计数器，包括周期计数器
1：使能所有计数器

PMU 用户使能寄存器 (PM_USERENR)

PMU user enable register

偏移地址：0xE08

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
															rw

位 31:1 保留，必须保持复位值。

位 0 **EN**：使能用户模式访问 (User mode access enable)。
0：禁止用户模式访问
1：使能用户模式访问

PMU 通用事件 ID 0 寄存器 (PM_CEID0)

PMU common event ID 0 register

偏移地址：0xE20

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	EVENT ID29	EVENT ID28	EVENT ID27	EVENT ID26	EVENT ID25	EVENT ID24	EVENT ID23	EVENT ID22	EVENT ID21	EVENT ID20	EVENT ID19	EVENT ID18	EVENT ID17	EVENT ID16
		r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT ID15	EVENT ID14	EVENT ID13	EVENT ID12	EVENT ID11	EVENT ID10	EVENT ID9	EVENT ID8	EVENT ID7	EVENT ID6	EVENT ID5	EVENT ID4	EVENT ID3	EVENT ID2	EVENT ID1	EVENT ID0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:30 保留，必须保持复位值。

位 29 **EVENTID29**：使能事件 ID 0x1D 用户模式访问 (Event ID 0x1D user mode access enable)
事件编号 0x1D 对应以下事件：
“总线周期”。
0：已设计该事件
1：未设计该事件

位 28 **EVENTID28**：使能事件 ID 0x1C 用户模式访问 (Event ID 0x1C user mode access enable)
事件编号 0x1C 对应以下事件：
“在架构上执行指令，通过条件代码检查，写入 TTBR”。
0：已设计该事件
1：未设计该事件

位 27 **EVENTID27**：使能事件 ID 0x1B 用户模式访问 (Event ID 0x1B user mode access enable)
事件编号 0x1B 对应以下事件：
“推测性地执行指令”。
0：已设计该事件
1：未设计该事件

位 26 **EVENTID26**：使能事件 ID 0x1A 用户模式访问 (Event ID 0x1A user mode access enable)
事件编号 0x1A 对应以下事件：
“本地内存错误”。
0：已设计该事件
1：未设计该事件

位 25 **EVENTID25**：使能事件 ID 0x19 用户模式访问 (Event ID 0x19 user mode access enable)
事件编号 0x19 对应以下事件：
“总线访问”。
0：已设计该事件
1：未设计该事件

位 24 **EVENTID24**：使能事件 ID 0x18 用户模式访问 (Event ID 0x18 user mode access enable)
事件编号 0x18 对应以下事件：
“二级数据缓存回写”。
0：已设计该事件
1：未设计该事件

位 23 **EVENTID23**：使能事件 ID 0x17 用户模式访问 (Event ID 0x17 user mode access enable)
事件编号 0x17 对应以下事件：
“二级数据缓存重新填充”。
0：已设计该事件
1：未设计该事件

位 22 **EVENTID22**：使能事件 ID 0x16 用户模式访问 (Event ID 0x16 user mode access enable)
事件编号 0x16 对应以下事件：
“二级数据缓存访问”。
0：已设计该事件
1：未设计该事件

位 21 **EVENTID21**：使能事件 ID 0x15 用户模式访问 (Event ID 0x15 user mode access enable)
事件编号 0x15 对应以下事件：
“一级数据缓存回写”。
0：已设计该事件
1：未设计该事件

位 20 **EVENTID20**：使能事件 ID 0x14 用户模式访问 (Event ID 0x14 user mode access enable)
事件编号 0x14 对应以下事件：
“一级指令缓存访问”。
0：已设计该事件
1：未设计该事件

位 19 **EVENTID19**：使能事件 ID 0x13 用户模式访问 (Event ID 0x13 user mode access enable)
事件编号 0x13 对应以下事件：
“数据内存访问”。
0：已设计该事件
1：未设计该事件

位 18 **EVENTID18**：使能事件 ID 0x12 用户模式访问 (Event ID 0x12 user mode access enable)
事件编号 0x12 对应以下事件：
“推测性地执行可预测的分支”。
0：已设计该事件
1：未设计该事件

位 17 **EVENTID17**：使能事件 ID 0x11 用户模式访问 (Event ID 0x11 user mode access enable)
事件编号 0x11 对应以下事件：
“周期”。
0：已设计该事件
1：未设计该事件

位 16 **EVENTID16**：使能事件 ID 0x10 用户模式访问 (Event ID 0x10 user mode access enable)
事件编号 0x10 对应以下事件：
“推测性地执行错误预测或未预测到的分支”。
0：已设计该事件
1：未设计该事件

位 15 **EVENTID15**：使能事件 ID 0xF 用户模式访问 (Event ID 0xF user mode access enable)
事件编号 0xF 对应以下事件：
“在架构上执行指令，通过条件代码检查，非对齐加载或存储”。
0：已设计该事件
1：未设计该事件

位 14 **EVENTID14**：使能事件 ID 0xE 用户模式访问 (Event ID 0xE user mode access enable)
事件编号 0xE 对应以下事件：
“在架构上执行指令，通过条件代码检查，程序返回”。
0：已设计该事件
1：未设计该事件

位 13 **EVENTID13**：使能事件 ID 0xD 用户模式访问 (Event ID 0xD user mode access enable)
事件编号 0xD 对应以下事件：
“在架构上执行指令，立即分支”。
0：已设计该事件
1：未设计该事件

位 12 **EVENTID12**：使能事件 ID 0xC 用户模式访问 (Event ID 0xC user mode access enable)
事件编号 0xC 对应以下事件：
“在架构上执行指令，通过条件代码检查，软件更改 PC”。
0：已设计该事件
1：未设计该事件

位 11 **EVENTID11**：使能事件 ID 0xB 用户模式访问 (Event ID 0xB user mode access enable)
事件编号 0xB 对应以下事件：
“在架构上执行指令，通过条件代码检查，写入 CONTEXTIDR”。
0：已设计该事件
1：未设计该事件

位 10 **EVENTID10**：使能事件 ID 0xA 用户模式访问 (Event ID 0xA user mode access enable)
事件编号 0xA 对应以下事件：
“在架构上执行指令，通过条件代码检查，异常返回”。
0：已设计该事件
1：未设计该事件

位 9 **EVENTID9**：使能事件 ID 0x9 用户模式访问 (Event ID 0x9 user mode access enable)
事件编号 0x9 对应以下事件：
“发生异常”。
0：已设计该事件
1：未设计该事件

位 8 **EVENTID8**：使能事件 ID 0x8 用户模式访问 (Event ID 0x8 user mode access enable)
事件编号 0x8 对应以下事件：
“在架构上执行指令”。
0：已设计该事件
1：未设计该事件

位 7 **EVENTID7**：使能事件 ID 0x7 用户模式访问 (Event ID 0x7 user mode access enable)
事件编号 0x7 对应以下事件：
“在架构上执行指令，通过条件代码检查，存储”。
0：已设计该事件
1：未设计该事件

位 6 **EVENTID6**：使能事件 ID 0x6 用户模式访问 (Event ID 0x6 user mode access enable)
事件编号 0x6 对应以下事件：
“在架构上执行指令，通过条件代码检查，加载”。
0：已设计该事件
1：未设计该事件

位 5 **EVENTID5**：使能事件 ID 0x5 用户模式访问 (Event ID 0x5 user mode access enable)
事件编号 0x5 对应以下事件：
“一级数据 TLB 重新填充”。
0：已设计该事件
1：未设计该事件

位 4 **EVENTID4**：使能事件 ID 0x4 用户模式访问 (Event ID 0x4 user mode access enable)
事件编号 0x4 对应以下事件：
“一级数据缓存访问”。
0：已设计该事件
1：未设计该事件

位 3 **EVENTID3**：使能事件 ID 0x3 用户模式访问 (Event ID 0x3 user mode access enable)
事件编号 0x3 对应以下事件：
“一级数据缓存重新填充”。
0：已设计该事件
1：未设计该事件

位 2 **EVENTID2**：使能事件 ID 0x2 用户模式访问 (Event ID 0x2 user mode access enable)
事件编号 0x2 对应以下事件：
“一级指令 TLB 重新填充”。
0：已设计该事件
1：未设计该事件

位 1 **EVENTID1**：使能事件 ID 0x1 用户模式访问 (Event ID 0x1 user mode access enable)
事件编号 0x1 对应以下事件：
“一级指令缓存重新填充”。
0：已设计该事件
1：未设计该事件

位 0 **EVENTID0**：使能事件 ID 0x0 用户模式访问 (Event ID 0x0 user mode access enable)
事件编号 0x0 对应以下事件：
“在架构上执行指令，通过条件代码检查，软件递增”。
0：已设计该事件
1：未设计该事件

PMU 锁定访问寄存器 (PM_LAR)

PMU lock access register

偏移地址：0xFB0

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **KEY[31:0]**：PMU 寄存器写访问 (PMU register write access)
允许处理器内核对某些 PMU 寄存器进行写访问
0xC5ACCE55：使能写访问
其他值：禁止写访问

PMU 锁定状态寄存器 (PM_LSR)

PMU lock status register

偏移地址：0xFB4

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NTT	SLK	SLI
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **NTT**：PM_LAR 寄存器的大小 (Size of the PM_LAR register)
0：32 位

位 1 SLK：当前锁定状态 (Current status of lock)
0：允许写访问
1：阻止写访问。仅允许读访问。

位 0 SLI：存在锁定控制机制 (Existence of lock control mechanism)
该位指示是否存在锁定控制机制。
0：不存在锁定控制机制
1：存在锁定控制机制

PMU 认证状态寄存器 (PM_AUTHSTAT)

PMU authentication status register

偏移地址：0xFB8

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNI	SNE	SI	SE	NSNI	NSNE	NSI	NSE
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7 **SNI**：安全非侵入式调试 (Secure non-invasive debug)
1：已设计安全非侵入式调试功能

位 6 **SNE**：使能安全非侵入式调试 (Secure non-invasive debug enabled)
0：不允许在安全 PL1 模式下进行非侵入式调试
1：允许在安全 PL1 模式下进行非侵入式调试

位 5 SI：安全侵入式调试 (Secure invasive debug)
0：未设计安全侵入式调试功能

位 4 SE：使能安全侵入式调试 (Secure invasive debug enabled)
0：不允许在安全 PL1 模式下进行侵入式暂停调试

位 3 **NSNI**：非安全非侵入式调试 (Non-secure non-invasive debug)
1：已设计非安全非侵入式调试功能

位 2 **NSNE**：使能非安全非侵入式调试
0：不允许进行非安全非侵入式调试
1：允许进行非安全非侵入式调试

位 1 NSI：非安全侵入式调试 (Non-secure invasive debug)
0：未设计非安全侵入式调试功能

位 0 NSE：使能非安全侵入式调试 (Non-secure invasive debug enabled)
0：不允许进行非安全侵入式调试

PMU CoreSight 设备类型寄存器 (PM_DEVTYPE)

PMU CoreSight device type register

偏移地址：0xFCC

复位值：0x0000 0016

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SUBTYPE[3:0]：设备子类型标识符 (Device sub-type identifier)**
0x1：处理器

位 3:0 **MAJORTYPE[3:0]：设备主类型标识符 (Device main type identifier)**
0x6：性能监视器

PMU CoreSight 外设标识寄存器 4 (PM_PIDR4)

PMU CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SIZE[3:0]：寄存器文件大小 (Register file size)**
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]：JEP106 连续代码 (JEP106 continuation code)**
0x4：Arm® JEDEC 代码

PMU CoreSight 外设标识寄存器 0 (PM_PIDR0)

PMU CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 00A7

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]：产品编号位字段 (Part number field)，位 [7:0]**
0xA7：PMU 产品编号

PMU CoreSight 外设标识寄存器 1 (PM_PIDR1)

PMU CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 JEP106ID[3:0]：JEP106 标识代码位字段 (JEP106 identity code field)，位 [3:0]
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]：产品编号位字段 (Part number field)，位 [11:8]**
0x9：PMU 产品编号

PMU CoreSight 外设标识寄存器 2 (PMU_PIDR2)

PMU CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 005B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 REVISION[3:0]：组件版本号 (Component revision number)
0x5：r0p5

位 3 **JEDEC：JEDEC 分配的值 (JEDEC assigned value)**
1：JEDEC 指定的设计人员 ID

位 2:0 JEP106ID[6:4]：JEP106 标识代码位字段 (JEP106 identity code field)，位 [6:4]
0x3：Arm® JEDEC 代码

PMU CoreSight 外设标识寄存器 3 (PMU_PIDR3)

PMU CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 REVAND[3:0]：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]：客户修改 (Customer modified)**
0x0：无客户修改

PMU CoreSight 组件标识寄存器 0 (PMU_CIDR0)

PMU CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]：组件 ID 位字段 (Component ID field)，位 [7:0]**
0x0D：公共 ID 值

PMU CoreSight 组件标识寄存器 1 (PMU_CIDR1)

PMU CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 CLASS[3:0]:组件 ID 位字段 (Component ID field)，位 [15:12]——组件类
0x9：PMU 组件

位 3:0 **PREAMBLE[11:8]：组件 ID 位字段 (Component ID field)，位 [11:8]**
0x0：公共 ID 值

PMU CoreSight 组件标识寄存器 2 (PMU_CIDR2)

PMU CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]：组件 ID 位字段 (Component ID field)，位 [23:16]**
0x05：公共 ID 值

PMU CoreSight 组件标识寄存器 3 (PMU_CIDR3)

PMU CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]：组件 ID 位字段 (Component ID field)，位 [31:24]**
0xB1：公共 ID 值

PMU 寄存器映射

表 617. 性能监视单元寄存器映射和复位值

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**PM_XEVCNTR0**	PMN[31:0]																															
	Reset value																																
0x004	**PM_XEVCNTR1**	PMN[31:0]																															
	Reset value																																

表 617. 性能监视单元寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x008	**PM_XEVCNTR2**	PMN[31:0]																															
	Reset value																																
0x00C	**PM_XEVCNTR3**	PMN[31:0]																															
	Reset value																																
0x07C	**PM_CCNTR**	CCNT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x400	**PM_XEVTYPER0**	P	U	NSK	NSU	NSH	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EVTCOUNT[7:0]							
	Reset value	0	0	0	0	0																											
0x404	**PM_XEVTYPER1**	P	U	NSK	NSU	NSH	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EVTCOUNT[7:0]							
	Reset value	0	0	0	0	0																											
0x408	**PM_XEVTYPER2**	P	U	NSK	NSU	NSH	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EVTCOUNT[7:0]							
	Reset value	0	0	0	0	0																											
0x40C	**PM_XEVTYPER3**	P	U	NSK	NSU	NSH	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EVTCOUNT[7:0]							
	Reset value	0	0	0	0	0																											
0x47C	**PM_XEVTYPER31**	P	U	NSK	NSU	NSH	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0																											
0xC00	**PM_CNTENSET**	C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	P3	P2	P1	P0
	Reset value																																
0xC20	**PM_CNTENCLR**	C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	P3	P2	P1	P0
	Reset value																																
0xC40	**PM_INTENSET**	C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	P3	P2	P1	P0
	Reset value																																
0xC60	**PM_INTENCLR**	C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	P3	P2	P1	P0
	Reset value																																
0xC80	**PM_OVSR**	C	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	P3	P2	P1	P0
	Reset value																																
0xCA0	**PM_SWINC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	P3	P2	P1	P0
	Reset value																																
0xE00	**PM_CFGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UEN	Res.	Res.	EX	CCD	CC	SIZE[5:0]						N[7:0]							
	Reset value													1			0	1	1	0	1	1	1	1	1	0	0	0	0	0	1	0	0
0xE04	**PM_CR**	IMP[7:0]								IDCODE[7:0]								N[4:0]					Res.	Res.	Res.	Res.	Res.	DP	X	D	C	P	E
	Reset value	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	0	1	1						0	0	0	1	0	0

表 617. 性能监视单元寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xE08	**PM_USERENR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
	Reset value																																
0xE20	**PM_CEID0**	EVENTID[31:0]																															
	Reset value	0	0												1	1	1												1	1			1
0xFB0	**PM_LAR**	KEY[31:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0xFB4	**PM_LSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NTT	SLK	SLI
	Reset value																														0	1	1
0xFB8	**PM_AUTHSTAT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNI	SNE	SI	SE	NSNI	NSNE	NSI	NSE
	Reset value																									1	0	0	0	1	0	0	0
0xFCC	**PM_DEVTYPE**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE [3:0]				MAJORTYPE[3:0]			
	Reset value																									0	0	0	1	0	1	1	0
0xFD0	**PM_PIDR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON[3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	**PM_PIDR5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	**PM_PIDR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	**PM_PIDR7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	**PM_PIDR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM [7:0]							
	Reset value																									0	0	0	0	0	1	1	1
0xFE4	**PM_PIDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	1	1	0	0
0xFE8	**PM_PIDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106 ID[6:4]		
	Reset value																									0	1	0	1	1	0	1	1
0xFEC	**PM_PIDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND [3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	**PM_CIDR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	**PM_CIDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE			
	Reset value																									1	0	0	1	0	0	0	0

表 617. 性能监视单元寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFF8	**PM_CIDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	**PM_CIDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE							
	Reset value																									1	0	1	1	0	0	0	1

59.11.3 嵌入式跟踪宏单元 (ETM)

Cortex®-A7 ETM 是与 CPU 紧密耦合的 CoreSight™ 组件。ETM 生成跟踪数据包，可用于对 Cortex®-A7 内核的执行进行跟踪。指令执行和数据访问都可以跟踪。

ETM 通过处理器跟踪接口从 CPU 接收信息，包括：

- 在同一周期内执行的指令数量
- 程序流的变化
- 当前处理器指令状态
- 由加载和存储指令访问的内存位置的地址
- 传输的类型、方向和大小
- 条件代码信息
- 异常信息
- 等待中断状态信息

有关更多信息，请参见《Arm® CoreSight™ ETM™-A7 技术参考手册》*[5]*。

嵌入式跟踪宏单元寄存器

调试器可以通过调试 APB 在 0xE00D C000 到 0xE00D CFFC 的地址范围内访问 ETM 寄存器。软件可以在 0x500D C000 到 0x500D CFFC 的地址范围内访问这些寄存器。

有关 ETM 寄存器的说明，请参见《Arm® CoreSight™ ETM™-A7 技术参考手册》*[5]*。

ETM 控制寄存器 (ETM_CR)

ETM control register

偏移地址：0x000

复位值：0x0000 0461

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	VMIDTRCEN	Res.	TSEN	PROCSEL[2:0]			Res.	Res.	Res.	PORTSIZE[3]	DATAONLY	FILTERCPRT	SUPPDATA	PORTMODE[1:0]	
	rw		rw	rw	rw	rw				r	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONTXTIDSIZE[1:0]		PORTMODE[2]	CYCLEACC	ETMPORTSEL	ETMPROG	DBGRQCTL	BRANCHOUTPUT	Res.	PORTSIZE[2:0]			DATAACC[1:0]		MONITORCPRT	ETMPDN
rw	rw	rw	rw	rw	rw	rw	rw		r	r	r	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30 **VMIDTRCEN**：使能 VMID 跟踪 (VMID trace enable)
0：禁止 VMID 跟踪
1：使能 VMID 跟踪

位 29 保留，必须保持复位值。

位 28 **TSEN**：使能时间戳 (Timestamping enable)
0：禁止时间戳
1：使能时间戳

位 27:25 **PROCSEL[2:0]**：处理器选择 (Processor select)。
0：选择 CPU0 进行跟踪
1：选择 CPU1 进行跟踪
2：保留
3：保留

位 24:22 保留，必须保持复位值。

位 20 **DATAONLY**：仅数据模式 (Data-only mode)
0：使能指令跟踪
1：使能仅数据跟踪

位 19 **FILTERCPRT**：过滤协处理器寄存器传输 (Filter co-processor register transfers)
如果 MONITORCPRT = 0，该位应设置为 0。
如果 MONITORCPRT = 1，该位具有以下作用：
0：跟踪所有 CPRT
1：仅在 ViewData 工作时跟踪 CPRT。

位 18 **SUPPDATA**：使能数据抑制 (Enable data suppression)
0：不抑制数据
1：在 FIFO 为空计数 < ETMFFLR.FIFOFULLLVL 时抑制数据

位 15:14 **CONTXTIDSIZE[1:0]**：上下文 ID 大小 (Context ID size)。
0：不跟踪上下文 ID
1：跟踪上下文 ID 位 [7:0]
2：跟踪上下文 ID 位 [15:0]
3：跟踪上下文 ID 位 [31:0]

位 13，17:16 **PORTMODE[2:0]**：设置跟踪端口时钟模式 (Set trace port clocking mode)
将 PORTMODE[2:0] 设置为任意值将选择动态模式。

位 12 **CYCLEACC**：周期精确跟踪 (Cycle accurate tracing)
0：禁止
1：使能。包含对已执行指令的精确周期计数

位 11 **ETMPORTSEL**：控制 ETMEN 输出状态 (Control state of ETMEN output)
0：ETMEN 为低电平
1：ETMEN 为高电平
必须将该位置 1，以便使能跟踪输出

位 10 **ETMPROG**：ETM 编程 (ETM programming)
在编程 ETM 之前，必须先将该位置 1。当该位置 1 时，将阻止跟踪。
0：ETM 处于工作状态
1：ETM 处于编程状态

位 9 **DBGRQCTL**：调试请求控制 (Debug request control)。

0：在发生触发事件时不设置 DBGRQ 输出。

1：在发生触发事件时设置 DBGRQ 输出以强制处理器进入调试状态

位 8 **BRANCHOUTPUT**：分支输出 (Branch output)

0：不跟踪分支地址

1：输出所有分支地址，即使分支是因某个直接分支指令引起也如此。将该位置 1 时，无需访问正在执行的代码的内存映像也可以重建程序流

位 7 保留，必须保持复位值。

位 21，6:4 **PORTSIZE[3:0]**：端口大小 (Port size)

将 PORTSIZE[3:0] 设置为任意值将选择 64 位端口大小。

位 3:2 **DATAACC[1:0]**：数据访问跟踪模式 (Data access tracing mode)。

0：不跟踪数据

1：仅跟踪访问的数据部分

2：仅跟踪访问的地址部分

3：跟踪访问的地址和数据

位 1 **MONITORCPRT**：监视协处理器寄存器传输 (Monitor co-processor register transfers)

0：不跟踪 CPRT

1：跟踪 CPRT。如果 FILTERCPRT = 1，则仅在 ViewData 工作时跟踪 CPRT。

位 0 **ETMPDN**：ETM 掉电 (ETM power down)

0：使能 ETM 跟踪

1：禁止 ETM 跟踪，并且 ETM 处于低功耗模式

ETM 配置代码寄存器 (ETM_CCR)

ETM configuration code register

偏移地址：0x004

复位值：0x8D29 4024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDREG	Res.	Res.	Res.	SWACC	START STOP	CONTXTIDCMP [1:0]		FIFOFULL	Res.	EXTOUTPUTS [1:0]		EXTINPUTS[2:0]			SEQUENCER
r				r	r	r	r	r		r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTERS[2:0]			MMAPDECS[4:0]					DATACOMPS[3:0]				ADCOMPPRS[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31 **IDREG**：ETM ID 寄存器存在 (ETM ID register present)

1：ETMIDR 寄存器存在并定义了使用的 ETM 架构版本

位 30:28 保留，必须保持复位值。

位 27 **SWACC**：协处理器和内存访问 (Co-processor and memory access)

1：支持对寄存器进行内存映射访问

位 26 **STARTSTOP**：跟踪开始/停止模块存在 (Trace start/stop block present)

1：跟踪开始/停止模块存在

位 25:24 **CONTXTIDCMP[1:0]**：上下文 ID 比较器数量 (Number of context ID comparators)

0x1：设计了一个上下文 ID 比较器

位 23 **FIFOFULL**：FIFOFULL 逻辑存在 (FIFOFULL logic present)
0：ETM 中不存在 FIFOFULL 逻辑

位 22 保留，必须保持复位值。

位 21:20 **EXTOUTPUTS[1:0]**：外部输出数量 (Number of external outputs)
0x2：支持两个外部输出

位 19:17 **EXTINPUTS[2:0]**：外部输入数量 (Number of external inputs)
0x4：支持四个外部输入

位 16 **SEQUENCER**：定序器存在 (Sequencer present)
1：ETM 中存在定序器。

位 15:13 **COUNTERS[2:0]**：计数器数量 (Number of counters)
0x2：设计了两个计数器

位 12:8 **MMAPDECS[4:0]**：内存映射解码器数量 (Number of memory map decoders)
0x0：未设计内存映射解码器输入。

位 7:4 **DATACOMPS[3:0]**：数据值比较器数量 (Number of data value comparators)
0x2：已设计两个数据值比较器。

位 3:0 **ADCOMPPRS[3:0]**：地址比较器对数量 (Number of address comparator pairs)
0x4：已设计四个地址比较器对。

ETM 触发寄存器 (ETM_TRIGGER)

ETM trigger register

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCN[1:0]		RESOURCEB[6:0]							RESOURCEA[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:14 **FCN[2:0]**：布尔函数 (Boolean function)
如果 A 定义为第一资源匹配，B 定义为第二资源匹配，则事件定义为 A 与 B 的函数。

0x0：A
0x1：NOT A
0x2：A AND B
0x3：NOT A AND B
0x4：NOT A AND NOT B
0x5：A OR B
0x6：NOT A OR B
0x7：NOT A OR NOT B

位 13:7 **RESOURCEB[6:0]**：第二资源标识符 (Second resource identifier)
关于位说明，请参见 RESOURCEA[6:0] 位字段。

位 6:0 **RESOURCEA[6:0]**：第一资源标识符 (First resource identifier)

位 [6:4] 定义资源类型，位 [3:0] 定义索引。下面列出了支持的资源标识符。编程任何其他值都可能导致不可预测的行为。

0x00：单地址比较器 1
...
0x0F：单地址比较器 16
0x10：地址范围比较器 1
...
0x17：地址范围比较器 8
0x18：仪表资源 1
0x19：仪表资源 2
0x1A：仪表资源 3
0x1B：仪表资源 4
0x20：嵌入式 ICE 监测点比较器 1（来自 DWT）
...
0x27：嵌入式 ICE 监测点比较器 8（来自 DWT）
0x40：值为 0 的计数器 1
0x41：值为 0 的计数器 2
0x42：值为 0 的计数器 3
0x43：值为 0 的计数器 4
0x50：状态 1 的定序器
0x51：状态 2 的定序器
0x52：状态 3 的定序器
0x58：上下文 ID 比较器 1
0x5B：VMID 比较器
0x5F：跟踪开始/停止资源
0x60：外部输入 1
0x61：外部输入 2
0x62：外部输入 3
0x63：外部输入 4
0x6D：非安全状态的处理器
0x6E：处理器禁止的跟踪
0x6F：硬接线输入，始终为真
其他值：保留

ETM 状态寄存器 (ETM_SR)

ETM status register

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIG	STSTP RESSTAT	PROG BIT	UNTRC DOVFL
												rw	rw	r	r

位 31:4 保留，必须保持复位值。

位 3 **TRIG**：触发标志 (Trigger flag)
发生触发时置 1，可防止在再次编程 ETM 之前输出触发信号。

位 2 **STSTPRESSTAT**：触发开始/停止资源状态 (Trigger start/stop resource status)
保存跟踪开始/停止资源的当前状态。如果置 1，则表示已匹配跟踪开启地址，无相应的跟踪关闭地址匹配。

位 1 **PROGBIT**：编程位值 (Programming bit value)（只读）
ETM 编程位的当前有效值，ETM_CR 的位 [10]。在编程 ETM 之前，必须将编程位置 1 并等待该位变为 1。
在 FIFO 为空之前，或者 ETM_OSLSR 寄存器中的 OS 锁定位置 1 时，该位始终保持为 0。

位 0 **UNTRCDOVFL**：未跟踪溢出 (Untraced overflow)（只读）
如果置 1，则表示存在尚未跟踪的溢出。在重新开始跟踪时，或在 ETM 掉电位（ETM_CR 寄存器的位 [0]）置 1 时，该位清零。

ETM 配置寄存器 (ETM_SCR)

ETM configuration register

偏移地址：0x014

复位值：0x0002 0D09

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NOFTCHCOMP	Res.
														r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SUPPROC[2:0]			PM SUPP	PS SUPP	MAXPS [3]	FF SUPP	Res.	Res.	Res.	Res.	Res.	MAXPS[2:0]		
	r	r	r	r	r	r	r						r	r	r

位 31:18 保留，必须保持复位值。

位 17 **NOFTCHCOMP**：无取指令比较 (No fetch comparisons)
1：未设计取指令比较

位 16:15 保留，必须保持复位值。

位 14:12 **SUPPROC[2:0]**：支持的处理器数 (Number of supported processors)
0x0：支持一个处理器

位 11 **PMSUPP**：端口模式支持 (Port mode support)
0：不支持当前选择的端口模式
1：支持当前选择的端口模式

位 10 **PSSUPP**：端口大小支持 (Port size support)
0：不支持当前选择的端口大小
1：支持当前选择的端口大小

位 8 **FFSUPP**：FIFOFULL 支持 (FIFOFULL support)
0：不支持 FIFOFULL

位 7:3 保留，必须保持复位值。

位 9、2、1、0 **MAXPS[3:0]**：最大 ETM 端口大小位字段 (Maximum ETM port size field)
这些位指示支持的最大 ETM 端口大小。
0x4：最大端口大小 = 4
其他值：保留

ETM 跟踪使能开始/停止控制寄存器 (ETM_SSCR)

ETM trace enable start/stop control register

偏移地址：0x018

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STOPADDRCOMPSEL[16:1]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STARTADDRCOMPSEL[16:1]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **STOPADDRCOMPSEL[16:1]**：停止地址比较器选择位 (Stop address comparator select bits)
当位 n 置 1 时，选择单地址比较器 n 作为停止地址。

位 15:0 **STARTADDRCOMPSEL[16:1]**：起始地址比较器选择位 (Start address comparator select bits)
当位 n 置 1 时，选择单地址比较器 n 作为起始地址。

ETM 跟踪使能控制 2 寄存器 (ETM_TECR2)

ETM trace enable control 2 register

偏移地址：0x01C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INCEXCCTLCOMPSEL[16:1]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **INCEXCCTLCOMPSEL[16:1]**：包括/不包括控制比较器选择位 (Include/exclude control comparator select bits)
当位 n 置 1 时，选择单地址比较器 n 进行包括/不包括控制。

ETM 跟踪使能事件寄存器 (ETM_TEEVR)

ETM trace enable event register

偏移地址：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCN[1:0]		RESOURCEB[6:0]							RESOURCEA[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:14 **FCN[2:0]**：布尔函数 (Boolean function)

如果 A 定义为第一资源匹配，B 定义为第二资源匹配，则事件定义为 A 与 B 的函数。

0x0：A
0x1：NOT A
0x2：A AND B
0x3：NOT A AND B
0x4：NOT A AND NOT B
0x5：A OR B
0x6：NOT A OR B
0x7：NOT A OR NOT B

位 13:7 **RESOURCEB[6:0]**：第二资源标识符 (Second resource identifier)

关于位说明，请参见 RESOURCEA[6:0] 位字段。

位 6:0 **RESOURCEA[6:0]**：第一资源标识符 (First resource identifier)

位 [6:4] 定义资源类型，位 [3:0] 定义索引。下面列出了支持的资源标识符。编程任何其他值都可能导致不可预测的行为。

0x00：单地址比较器 1
...
0x0F：单地址比较器 16
0x10：地址范围比较器 1
...
0x17：地址范围比较器 8
0x18：仪表资源 1
0x19：仪表资源 2
0x1A：仪表资源 3
0x1B：仪表资源 4
0x20：嵌入式 ICE 监测点比较器 1（来自 DWT）
...
0x27：嵌入式 ICE 监测点比较器 8（来自 DWT）
0x40：值为 0 的计数器 1
0x41：值为 0 的计数器 2
0x42：值为 0 的计数器 3
0x43：值为 0 的计数器 4
0x50：状态 1 的定序器
0x51：状态 2 的定序器
0x52：状态 3 的定序器
0x58：上下文 ID 比较器 1
0x5B：VMID 比较器
0x5F：跟踪开始/停止资源
0x60：外部输入 1
0x61：外部输入 2
0x62：外部输入 3
0x63：外部输入 4
0x6D：非安全状态的处理器
0x6E：处理器禁止的跟踪
0x6F：硬接线输入，始终为真
其他值：保留

ETM 跟踪使能控制 1 寄存器 (ETM_TECR1)

ETM trace enable control 1 register

偏移地址：0x024

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	TSSEN	INCEX CCTL	INCEXCMMDSEL[16:9]							
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INCEXCMMDSEL[8:1]								INCEXCADCOMPSEL[8:1]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25 **TSSEN**：跟踪开始/停止跟踪使能控制 (Trace start/stop trace enable control)

0：跟踪开始/停止模块不控制跟踪使能
1：跟踪开始/停止模块控制跟踪使能

位 24 **INCEXCCTL**：跟踪开始/停止跟踪使能控制 (Trace start/stop trace enable control)

0：包括。指定的资源表示可以进行跟踪的区域。当位于此区域之外时，不可以进行跟踪

1：不包括。在位 23:0 和 ETMTECR2 寄存器中指定的资源表示不可以进行跟踪的区域。当位于不包括区域之外时，可以进行跟踪。

位 23:8 **INCEXCMMDSEL[16:1]**：包括/不包括控制的内存映射解码选择位 (Memory map decode select bits for include/exclude control)。

未使用，没有内存映射解码器

位 7:0 **INCEXCADCOMPSEL[8:1]**：包括/不包括控制的地址比较器选择位 (Address comparator select bits for include/exclude control)。

当位 n 置 1 时，选择地址范围比较器 n 进行包括/不包括控制

ETM FIFOFULL 级别寄存器 (ETM_FFLR)

ETM FIFOFULL level register

偏移地址：0x028

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FIFOFULLLVL[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **FIFOFULLLVL[7:0]**：FIFOFULL 信号的阈值 (Threshold for FIFOFULL signal)

FIFO 中剩余的字节数 (The number of bytes left in the FIFO)，如果低于该字节数，将触发 SuppressData 信号。例如，如果将此值设置为 15，而 FIFO 中的空闲字节少于 15 个，则会抑制数据跟踪（如果已使能）。

ETM 查看数据事件寄存器 (ETM_VDEVR)

ETM view data event register

偏移地址：0x030

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2]
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCN[1:0]		RESOURCEB[6:0]							RESOURCEA[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:14 **FCN[2:0]**：布尔函数 (Boolean function)

如果 A 定义为第一资源匹配，B 定义为第二资源匹配，则事件定义为 A 与 B 的函数。

0x0：A
0x1：NOT A
0x2：A AND B
0x3：NOT A AND B
0x4：NOT A AND NOT B
0x5：A OR B
0x6：NOT A OR B
0x7：NOT A OR NOT B

位 13:7 **RESOURCEB[6:0]**：第二资源标识符 (Second resource identifier)

关于位说明，请参见 RESOURCEA[6:0] 位字段。

位 6:0 **RESOURCEA[6:0]**：第一资源标识符 (First resource identifier)

位 [6:4] 定义资源类型，位 [3:0] 定义索引。下面列出了支持的资源标识符。编程任何其他值都可能导致不可预测的行为。

0x00：单地址比较器 1
...
0x0F：单地址比较器 16
0x10：地址范围比较器 1
...
0x17：地址范围比较器 8
0x18：仪表资源 1
0x19：仪表资源 2
0x1A：仪表资源 3
0x1B：仪表资源 4
0x20：嵌入式 ICE 监测点比较器 1（来自 DWT）
...
0x27：嵌入式 ICE 监测点比较器 8（来自 DWT）
0x40：值为 0 的计数器 1
0x41：值为 0 的计数器 2
0x42：值为 0 的计数器 3
0x43：值为 0 的计数器 4
0x50：状态 1 的定序器
0x51：状态 2 的定序器
0x52：状态 3 的定序器
0x58：上下文 ID 比较器 1
0x5B：VMID 比较器
0x5F：跟踪开始/停止资源
0x60：外部输入 1
0x61：外部输入 2
0x62：外部输入 3
0x63：外部输入 4
0x6D：非安全状态的处理器
0x6E：处理器禁止的跟踪
0x6F：硬接线输入，始终为真
其他值：保留

ETM 查看数据控制 1 寄存器 (ETM_VDCR1)

ETM view data control 1 register

偏移地址：0x034

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VDEXSADCOMPSEL[16:1]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDINCSADCOMPSEL[16:1]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **VDEXSADCOMPSEL[16:1]**：ViewData 不包括控制的单地址比较器选择位 (Single address comparator select bits for ViewData exclude control)

当位 n 置 1 时，选择单地址比较器 n 进行不包括控制。

位 15:0 **VDINCSADCOMPSEL[16:1]**：ViewData 包括控制的单地址比较器选择位 (Single address comparator select bits for ViewData include control)

当位 n 置 1 时，选择单地址比较器 n 进行包括控制。

ETM 查看数据控制 3 寄存器 (ETM_VDCR3)

ETM view data control 3 register

偏移地址：0x03C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXCONLY
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDEXCADCOMPSEL[8:1]								VDINCADCOMPSEL[8:1]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **EXCONLY**：仅不包括 (Exclude only)

0：混合模式。ViewData 以混合模式工作，可以编程包括资源和不包括资源

1：仅不包括模式。ViewData 只能编程为不包括模式。如果没有不包括资源发生匹配，则可以进行跟踪

位 15:8 **VDEXCADCOMPSEL[8:1]**：ViewData 不包括控制的地址范围比较器选择位 (Address range comparator select bits for ViewData exclude control)

当位 n 置 1 时，选择地址范围比较器 n 进行不包括控制。

位 7:0 **VDINCADCOMPSEL[8:1]**：ViewData 包括控制的地址范围比较器选择位 (Address range comparator select bits for ViewData include control)

当位 n 置 1 时，选择单地址比较器 n 进行包括控制。

ETM 地址比较器值 x 寄存器 (ETM_ACVRx)

ETM address comparator value x register

偏移地址：0x040 + 0x4 * (x - 1)（x = 1 到 8）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRESS[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **ADDRESS[31:0]**：比较地址 (Address for comparison)

ETM 地址比较器类型 x 寄存器 (ETM_ACTRx)

ETM address comparator type x register

偏移地址：0x080 + 0x4 * (x - 1)（x = 1 到 8）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VMID COMP	HYP MODE	STMODE[3:0]				CTXTIDCOMP [1:0]		EXACT MATCH	DCOMP[1:0]		COMPACCSIZE [1:0]		ACCTYPE[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **VMIDCOMP**：虚拟机 ID (VMID) 比较使能 (Virtual machine ID (VMID) comparison enable)
值为 1 时表示地址比较器仅在当前 VMID 与
存储在 ETM_VMIDCVR 寄存器中的值匹配时发生匹配

位 14 **HYPMODE**：Hyp 模式比较使能 (Hyp mode comparison enable)
值为 1 时表示地址比较器在处理器工作于
Hyp 模式时也会发生匹配

位 13:10 **STMODE[3:0]**：状态和模式比较控制 (State and mode comparison control)
位 [3,1] 控制非安全状态比较。位 [2,0] 控制安全状态比较。
X0X0：在安全状态的所有模式下匹配
X0X1：在安全状态的任何模式下都不匹配
X1X0：在安全状态除用户模式外的所有模式下匹配
X1X1：仅在安全状态的用户模式下匹配
0X0X：在非安全状态的所有模式下匹配
0X1X：在非安全状态的任何模式下都不匹配
1X0X：在非安全状态除用户模式外的所有模式下匹配
1X1X：仅在非安全状态的用户模式下匹配

位 9:8 **CTXTIDCOMP[1:0]**：上下文 ID 比较器控制 (Context ID comparator control)。
- 0：忽略上下文 ID 比较器
- 1：地址比较器仅在上下文 ID 比较器值 1 匹配时发生匹配
- 2：地址比较器仅在上下文 ID 比较器值 2 匹配时发生匹配
- 3：地址比较器仅在上下文 ID 比较器值 3 匹配时发生匹配

位 7 **EXACTMATCH**：精确匹配 (Exact match)
- 0：比较器甚至在取消指令、数据传输中止或存储失败时也会发生匹配
- 1：比较器在取消指令、数据传输中止或存储失败时不会发生匹配

位 6:5 **DCOMP[1:0]**：数据值比较控制 (Data value comparison control)。
- 0：不进行数据值比较
- 1：比较器只能在数据值匹配时发生匹配
- 2：保留
- 3：比较器只能在数据值不匹配时发生匹配

位 4:3 **COMPACCSIZE[1:0]**：比较访问大小 (Comparison access size)。
- 0：Java 指令或字节数据
- 1：Thumb 指令或半字数据
- 2：保留
- 3：Arm 指令或字数据

位 2:0 **ACCTYPE[2:0]**：访问类型 (Access type)
- 0：取指（不支持）
- 1：执行指令
- 2：执行指令且通过条件代码测试
- 3：执行指令但未通过条件代码测试
- 4：数据加载或存储
- 5：数据加载
- 6：数据存储
- 7：保留

ETM 数据比较器值 x 寄存器 (ETM_DCVRx)

ETM data comparator value x register

偏移地址：0x0C0 + 0x4 * (x - 1)（x = 1 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DATA[31:0]**：进行比较的数据值 (Data value for comparison)

ETM 数据比较器掩码 x 寄存器 (ETM_DCMRx)

ETM data comparator mask x register

偏移地址：0x100 + 0x4 * (x - 1)（x = 1 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MASK[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **MASK[31:0]**：数据掩码 (Data mask)

当某个数据掩码位置 1 时，在比较过程中将忽略 ETM_DCVRn 寄存器中对应的位，并且必须将该位设置为零

ETM 计数器重载值 x 寄存器 (ETM_CNTRLDVRx)

ETM counter reload value x register

偏移地址：0x140 + 0x4 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INITIALCOUNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **INITIALCOUNT[15:0]**：初始计数 (Initial count)

指定计数器重载值。

ETM 计数器使能 x 寄存器 (ETM_CNTENRx)

ETM counter enable x register

偏移地址：0x150 + 0x4 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2]
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCN[1:0]		RESOURCEB[6:0]							RESOURCEA[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:14 **FCN[2:0]**：布尔函数 (Boolean function)

如果 A 定义为第一资源匹配，B 定义为第二资源匹配，则事件定义为 A 与 B 的函数。

0x0：A
0x1：NOT A
0x2：A AND B
0x3：NOT A AND B
0x4：NOT A AND NOT B
0x5：A OR B
0x6：NOT A OR B
0x7：NOT A OR NOT B

位 13:7 **RESOURCEB[6:0]**：第二资源标识符 (Second resource identifier)

关于位说明，请参见 RESOURCEA[6:0] 位字段。

位 6:0 **RESOURCEA[6:0]**：第一资源标识符 (First resource identifier)

位 [6:4] 定义资源类型，位 [3:0] 定义索引。下面列出了支持的资源标识符。编程任何其他值都可能导致不可预测的行为。

0x00：单地址比较器 1
...
0x0F：单地址比较器 16
0x10：地址范围比较器 1
...
0x17：地址范围比较器 8
0x18：仪表资源 1
0x19：仪表资源 2
0x1A：仪表资源 3
0x1B：仪表资源 4
0x20：嵌入式 ICE 监测点比较器 1（来自 DWT）
...
0x27：嵌入式 ICE 监测点比较器 8（来自 DWT）
0x40：值为 0 的计数器 1
0x41：值为 0 的计数器 2
0x42：值为 0 的计数器 3
0x43：值为 0 的计数器 4
0x50：状态 1 的定序器
0x51：状态 2 的定序器
0x52：状态 3 的定序器
0x58：上下文 ID 比较器 1
0x5B：VMID 比较器
0x5F：跟踪开始/停止资源
0x60：外部输入 1
0x61：外部输入 2
0x62：外部输入 3
0x63：外部输入 4
0x6D：非安全状态的处理器
0x6E：处理器禁止的跟踪
0x6F：硬接线输入，始终为真
其他值：保留

ETM 计数器重载事件 x 寄存器 (ETM_CNTRLDEVRx)

ETM counter reload event x register

偏移地址：0x160 + 0x4 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCN[1:0]		RESOURCEB[6:0]							RESOURCEA[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:14 **FCN[2:0]**：布尔函数 (Boolean function)

如果 A 定义为第一资源匹配，B 定义为第二资源匹配，则事件定义为 A 与 B 的函数。

0x0：A
0x1：NOT A
0x2：A AND B
0x3：NOT A AND B
0x4：NOT A AND NOT B
0x5：A OR B
0x6：NOT A OR B
0x7：NOT A OR NOT B

位 13:7 **RESOURCEB[6:0]**：第二资源标识符 (Second resource identifier)

关于位说明，请参见 RESOURCEA[6:0] 位字段。

位 6:0 **RESOURCEA[6:0]**：第一资源标识符 (First resource identifier)

位 [6:4] 定义资源类型，位 [3:0] 定义索引。下面列出了支持的资源标识符。编程任何其他值都可能导致不可预测的行为。

0x00：单地址比较器 1
...
0x0F：单地址比较器 16
0x10：地址范围比较器 1
...
0x17：地址范围比较器 8
0x18：仪表资源 1
0x19：仪表资源 2
0x1A：仪表资源 3
0x1B：仪表资源 4
0x20：嵌入式 ICE 监测点比较器 1（来自 DWT）
...
0x27：嵌入式 ICE 监测点比较器 8（来自 DWT）
0x40：值为 0 的计数器 1
0x41：值为 0 的计数器 2
0x42：值为 0 的计数器 3
0x43：值为 0 的计数器 4
0x50：状态 1 的定序器
0x51：状态 2 的定序器
0x52：状态 3 的定序器
0x58：上下文 ID 比较器 1
0x5B：VMID 比较器
0x5F：跟踪开始/停止资源
0x60：外部输入 1
0x61：外部输入 2
0x62：外部输入 3
0x63：外部输入 4
0x6D：非安全状态的处理器
0x6E：处理器禁止的跟踪
0x6F：硬接线输入，始终为真
其他值：保留

ETM 计数器值 x 寄存器 (ETM_CNTVRx)

ETM counter value x register

偏移地址：0x170 + 0x4 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INITIALCOUNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **INITIALCOUNT[15:0]**：初始计数 (Initial count)
指定计数器重载值。

ETM 定序器状态 1-2 转换事件寄存器 (ETM_SQ12EVR)

ETM sequencer state 1-2 transition event register

偏移地址：0x180

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCN[1:0]		RESOURCEB[6:0]							RESOURCEA[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:14 **FCN[2:0]**：布尔函数 (Boolean function)
如果 A 定义为第一资源匹配，B 定义为第二资源匹配，则事件定义为 A 与 B 的函数。

0x0：A
0x1：NOT A
0x2：A AND B
0x3：NOT A AND B
0x4：NOT A AND NOT B
0x5：A OR B
0x6：NOT A OR B
0x7：NOT A OR NOT B

位 13:7 **RESOURCEB[6:0]**：第二资源标识符 (Second resource identifier)
关于位说明，请参见 RESOURCEA[6:0] 位字段。

位 6:0 **RESOURCEA[6:0]**：第一资源标识符 (First resource identifier)

位 [6:4] 定义资源类型，位 [3:0] 定义索引。下面列出了支持的资源标识符。编程任何其他值都可能导致不可预测的行为。

0x00：单地址比较器 1
...
0x0F：单地址比较器 16
0x10：地址范围比较器 1
...
0x17：地址范围比较器 8
0x18：仪表资源 1
0x19：仪表资源 2
0x1A：仪表资源 3
0x1B：仪表资源 4
0x20：嵌入式 ICE 监测点比较器 1（来自 DWT）
...
0x27：嵌入式 ICE 监测点比较器 8（来自 DWT）
0x40：值为 0 的计数器 1
0x41：值为 0 的计数器 2
0x42：值为 0 的计数器 3
0x43：值为 0 的计数器 4
0x50：状态 1 的定序器
0x51：状态 2 的定序器
0x52：状态 3 的定序器
0x58：上下文 ID 比较器 1
0x5B：VMID 比较器
0x5F：跟踪开始/停止资源
0x60：外部输入 1
0x61：外部输入 2
0x62：外部输入 3
0x63：外部输入 4
0x6D：非安全状态的处理器
0x6E：处理器禁止的跟踪
0x6F：硬接线输入，始终为真
其他值：保留

ETM 定序器状态 2-1 转换事件寄存器 (ETM_SQ21EVR)

ETM sequencer state 2-1 transition event register

偏移地址：0x184

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCN[1:0]		RESOURCEB[6:0]							RESOURCEA[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:14 **FCN[2:0]**：布尔函数 (Boolean function)

如果 A 定义为第一资源匹配，B 定义为第二资源匹配，则事件定义为 A 与 B 的函数。

0x0：A
0x1：NOT A
0x2：A AND B
0x3：NOT A AND B
0x4：NOT A AND NOT B
0x5：A OR B
0x6：NOT A OR B
0x7：NOT A OR NOT B

位 13:7 **RESOURCEB[6:0]**：第二资源标识符 (Second resource identifier)

关于位说明，请参见 RESOURCEA[6:0] 位字段。

位 6:0 **RESOURCEA[6:0]**：第一资源标识符 (First resource identifier)

位 [6:4] 定义资源类型，位 [3:0] 定义索引。下面列出了支持的资源标识符。编程任何其他值都可能导致不可预测的行为。

0x00：单地址比较器 1
...
0x0F：单地址比较器 16
0x10：地址范围比较器 1
...
0x17：地址范围比较器 8
0x18：仪表资源 1
0x19：仪表资源 2
0x1A：仪表资源 3
0x1B：仪表资源 4
0x20：嵌入式 ICE 监测点比较器 1（来自 DWT）
...
0x27：嵌入式 ICE 监测点比较器 8（来自 DWT）
0x40：值为 0 的计数器 1
0x41：值为 0 的计数器 2
0x42：值为 0 的计数器 3
0x43：值为 0 的计数器 4
0x50：状态 1 的定序器
0x51：状态 2 的定序器
0x52：状态 3 的定序器
0x58：上下文 ID 比较器 1
0x5B：VMID 比较器
0x5F：跟踪开始/停止资源
0x60：外部输入 1
0x61：外部输入 2
0x62：外部输入 3
0x63：外部输入 4
0x6D：非安全状态的处理器
0x6E：处理器禁止的跟踪
0x6F：硬接线输入，始终为真
其他值：保留

ETM 定序器状态 2-3 转换事件寄存器 (ETM_SQ23EVR)

ETM sequencer state 2-3 transition event register

偏移地址：0x188

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCN[1:0]		RESOURCEB[6:0]							RESOURCEA[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:14 **FCN[2:0]**：布尔函数 (Boolean function)

如果 A 定义为第一资源匹配，B 定义为第二资源匹配，则事件定义为 A 与 B 的函数。

0x0：A
0x1：NOT A
0x2：A AND B
0x3：NOT A AND B
0x4：NOT A AND NOT B
0x5：A OR B
0x6：NOT A OR B
0x7：NOT A OR NOT B

位 13:7 **RESOURCEB[6:0]**：第二资源标识符 (Second resource identifier)

关于位说明，请参见 RESOURCEA[6:0] 位字段。

位 6:0 **RESOURCEA[6:0]**：第一资源标识符 (First resource identifier)

位 [6:4] 定义资源类型，位 [3:0] 定义索引。下面列出了支持的资源标识符。编程任何其他值都可能导致不可预测的行为。

0x00：单地址比较器 1
...
0x0F：单地址比较器 16
0x10：地址范围比较器 1
...
0x17：地址范围比较器 8
0x18：仪表资源 1
0x19：仪表资源 2
0x1A：仪表资源 3
0x1B：仪表资源 4
0x20：嵌入式 ICE 监测点比较器 1（来自 DWT）
...
0x27：嵌入式 ICE 监测点比较器 8（来自 DWT）
0x40：值为 0 的计数器 1
0x41：值为 0 的计数器 2
0x42：值为 0 的计数器 3
0x43：值为 0 的计数器 4
0x50：状态 1 的定序器
0x51：状态 2 的定序器
0x52：状态 3 的定序器
0x58：上下文 ID 比较器 1
0x5B：VMID 比较器
0x5F：跟踪开始/停止资源
0x60：外部输入 1
0x61：外部输入 2
0x62：外部输入 3
0x63：外部输入 4
0x6D：非安全状态的处理器
0x6E：处理器禁止的跟踪
0x6F：硬接线输入，始终为真
其他值：保留

ETM 定序器状态 3-1 转换事件寄存器 (ETM_SQ31EVR)

ETM sequencer state 3-1 transition event register

偏移地址：0x18C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCN[1:0]		RESOURCEB[6:0]							RESOURCEA[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:14 **FCN[2:0]**：布尔函数 (Boolean function)

如果 A 定义为第一资源匹配，B 定义为第二资源匹配，则事件定义为 A 与 B 的函数。

0x0：A
0x1：NOT A
0x2：A AND B
0x3：NOT A AND B
0x4：NOT A AND NOT B
0x5：A OR B
0x6：NOT A OR B
0x7：NOT A OR NOT B

位 13:7 **RESOURCEB[6:0]**：第二资源标识符 (Second resource identifier)

关于位说明，请参见 RESOURCEA[6:0] 位字段。

位 6:0 **RESOURCEA[6:0]**：第一资源标识符 (First resource identifier)

位 [6:4] 定义资源类型，位 [3:0] 定义索引。下面列出了支持的资源标识符。编程任何其他值都可能导致不可预测的行为。

0x00：单地址比较器 1
...
0x0F：单地址比较器 16
0x10：地址范围比较器 1
...
0x17：地址范围比较器 8
0x18：仪表资源 1
0x19：仪表资源 2
0x1A：仪表资源 3
0x1B：仪表资源 4
0x20：嵌入式 ICE 监测点比较器 1（来自 DWT）
...
0x27：嵌入式 ICE 监测点比较器 8（来自 DWT）
0x40：值为 0 的计数器 1
0x41：值为 0 的计数器 2
0x42：值为 0 的计数器 3
0x43：值为 0 的计数器 4
0x50：状态 1 的定序器
0x51：状态 2 的定序器
0x52：状态 3 的定序器
0x58：上下文 ID 比较器 1
0x5B：VMID 比较器
0x5F：跟踪开始/停止资源
0x60：外部输入 1
0x61：外部输入 2
0x62：外部输入 3
0x63：外部输入 4
0x6D：非安全状态的处理器
0x6E：处理器禁止的跟踪
0x6F：硬接线输入，始终为真
其他值：保留

ETM 定序器状态 3-2 转换事件寄存器 (ETM_SQ32EVR)

ETM sequencer state 3-2 transition event register

偏移地址：0x190

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCN[1:0]		RESOURCEB[6:0]							RESOURCEA[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:14 **FCN[2:0]**：布尔函数 (Boolean function)

如果 A 定义为第一资源匹配，B 定义为第二资源匹配，则事件定义为 A 与 B 的函数。

0x0：A
0x1：NOT A
0x2：A AND B
0x3：NOT A AND B
0x4：NOT A AND NOT B
0x5：A OR B
0x6：NOT A OR B
0x7：NOT A OR NOT B

位 13:7 **RESOURCEB[6:0]**：第二资源标识符 (Second resource identifier)

关于位说明，请参见 RESOURCEA[6:0] 位字段。

位 6:0 **RESOURCEA[6:0]**：第一资源标识符 (First resource identifier)

位 [6:4] 定义资源类型，位 [3:0] 定义索引。下面列出了支持的资源标识符。编程任何其他值都可能导致不可预测的行为。

0x00：单地址比较器 1
...
0x0F：单地址比较器 16
0x10：地址范围比较器 1
...
0x17：地址范围比较器 8
0x18：仪表资源 1
0x19：仪表资源 2
0x1A：仪表资源 3
0x1B：仪表资源 4
0x20：嵌入式 ICE 监测点比较器 1（来自 DWT）
...
0x27：嵌入式 ICE 监测点比较器 8（来自 DWT）
0x40：值为 0 的计数器 1
0x41：值为 0 的计数器 2
0x42：值为 0 的计数器 3
0x43：值为 0 的计数器 4
0x50：状态 1 的定序器
0x51：状态 2 的定序器
0x52：状态 3 的定序器
0x58：上下文 ID 比较器 1
0x5B：VMID 比较器
0x5F：跟踪开始/停止资源
0x60：外部输入 1
0x61：外部输入 2
0x62：外部输入 3
0x63：外部输入 4
0x6D：非安全状态的处理器
0x6E：处理器禁止的跟踪
0x6F：硬接线输入，始终为真
其他值：保留

ETM 定序器状态 1-3 转换事件寄存器 (ETM_SQ13EVR)

ETM sequencer state 1-3 transition event register

偏移地址：0x194

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCN[1:0]		RESOURCEB[6:0]							RESOURCEA[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:14 **FCN[2:0]**：布尔函数 (Boolean function)

如果 A 定义为第一资源匹配，B 定义为第二资源匹配，则事件定义为 A 与 B 的函数。

0x0：A
0x1：NOT A
0x2：A AND B
0x3：NOT A AND B
0x4：NOT A AND NOT B
0x5：A OR B
0x6：NOT A OR B
0x7：NOT A OR NOT B

位 13:7 **RESOURCEB[6:0]**：第二资源标识符 (Second resource identifier)

关于位说明，请参见 RESOURCEA[6:0] 位字段。

位 6:0 **RESOURCEA[6:0]**：第一资源标识符 (First resource identifier)

位 [6:4] 定义资源类型，位 [3:0] 定义索引。下面列出了支持的资源标识符。编程任何其他值都可能导致不可预测的行为。

0x00：单地址比较器 1
...
0x0F：单地址比较器 16
0x10：地址范围比较器 1
...
0x17：地址范围比较器 8
0x18：仪表资源 1
0x19：仪表资源 2
0x1A：仪表资源 3
0x1B：仪表资源 4
0x20：嵌入式 ICE 监测点比较器 1（来自 DWT）
...
0x27：嵌入式 ICE 监测点比较器 8（来自 DWT）
0x40：值为 0 的计数器 1
0x41：值为 0 的计数器 2
0x42：值为 0 的计数器 3
0x43：值为 0 的计数器 4
0x50：状态 1 的定序器
0x51：状态 2 的定序器
0x52：状态 3 的定序器
0x58：上下文 ID 比较器 1
0x5B：VMID 比较器
0x5F：跟踪开始/停止资源
0x60：外部输入 1
0x61：外部输入 2
0x62：外部输入 3
0x63：外部输入 4
0x6D：非安全状态的处理器
0x6E：处理器禁止的跟踪
0x6F：硬接线输入，始终为真
其他值：保留

ETM 当前定序器状态寄存器 (ETM_SQR)

ETM current sequencer state register

偏移地址：0x19C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SQSTATE[1:0]	
														r	r

位 31:2 保留，必须保持复位值。

位 1:0 **SQSTATE[1:0]**：当前定序器状态 (Current sequencer state)

0：状态 1
1：状态 2
2：状态 3
3：保留

ETM 外部输出事件 x 寄存器 (ETM_EXTOUTEVRx)

ETM external output event x register

偏移地址：0x1A0 + 0x4 * (x - 1)（x = 1 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCN[1:0]		RESOURCEB[6:0]							RESOURCEA[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:14 **FCN[2:0]**：布尔函数 (Boolean function)

如果 A 定义为第一资源匹配，B 定义为第二资源匹配，则事件定义为 A 与 B 的函数。

0x0：A
0x1：NOT A
0x2：A AND B
0x3：NOT A AND B
0x4：NOT A AND NOT B
0x5：A OR B
0x6：NOT A OR B
0x7：NOT A OR NOT B

位 13:7 **RESOURCEB[6:0]**：第二资源标识符 (Second resource identifier)

关于位说明，请参见 RESOURCEA[6:0] 位字段。

位 6:0 **RESOURCEA[6:0]**：第一资源标识符 (First resource identifier)

位 [6:4] 定义资源类型，位 [3:0] 定义索引。下面列出了支持的资源标识符。编程任何其他值都可能导致不可预测的行为。

0x00：单地址比较器 1
...
0x0F：单地址比较器 16
0x10：地址范围比较器 1
...
0x17：地址范围比较器 8
0x18：仪表资源 1
0x19：仪表资源 2
0x1A：仪表资源 3
0x1B：仪表资源 4
0x20：嵌入式 ICE 监测点比较器 1（来自 DWT）
...
0x27：嵌入式 ICE 监测点比较器 8（来自 DWT）
0x40：值为 0 的计数器 1
0x41：值为 0 的计数器 2
0x42：值为 0 的计数器 3
0x43：值为 0 的计数器 4
0x50：状态 1 的定序器
0x51：状态 2 的定序器
0x52：状态 3 的定序器
0x58：上下文 ID 比较器 1
0x5B：VMID 比较器
0x5F：跟踪开始/停止资源
0x60：外部输入 1
0x61：外部输入 2
0x62：外部输入 3
0x63：外部输入 4
0x6D：非安全状态的处理器
0x6E：处理器禁止的跟踪
0x6F：硬接线输入，始终为真
其他值：保留

ETM 上下文 ID 比较器值 1 寄存器 (ETM_CIDCVR1)

ETM context ID comparator value 1 register

偏移地址：0x1B0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CONTEXTID[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONTEXTID[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CONTEXTID[31:0]**：进行比较的上下文 ID 值 (Context ID value for comparison)

ETM 上下文 ID 比较器掩码寄存器 (ETM_CIDCMR)

ETM context ID comparator mask register

偏移地址：0x1BC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MASK[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **MASK[31:0]**：上下文 ID 掩码 (Context ID mask)

当某个掩码位置 1 时，在比较过程中将忽略 ETM_CIDCVR1 寄存器中对应的位，并且必须将该位设置为零

ETM 同步频率寄存器 (ETM_SYNCFR)

ETM synchronization frequency register

偏移地址：0x1E0

复位值：0x0000 0400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	FREQUENCY[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:0 **FREQUENCY[11:0]**：同步频率 (Synchronisation frequency)。

此值表示 ETM 使用的跟踪中相邻同步点的时间间隔（单位为字节）（工具只能在同步点开始解压）。

ETM ID 寄存器 (ETM_IDR)

ETM ID register

偏移地址：0x1E4

复位值：0x410C F250

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMPLEMENTER[7:0]								Res.	Res.	Res.	BPENC	SECEXT	T32	Res.	LDPC1ST
r	r	r	r	r	r	r	r				r	r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PROCFAM[3:0]				ARCHVMAJ[3:0]				ARCHVMIN[3:0]				REVISION[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 **IMPLEMENTER[7:0]**：设计人员代码 (Implementer code)
0x41：Arm®

位 23:21 保留，必须保持复位值。

位 20 **BPENC**：分支数据包编码支持 (Branch packet encoding support)
0：支持原始分支数据包编码

位 19 **SECEXT**：安全扩展支持 (Security extensions support)
1：支持安全扩展

位 18 **T32**：32 位 Thumb 指令跟踪 (32-bit Thumb instruction tracing)
1：支持 32 位 Thumb 指令

位 17 保留，必须保持复位值。

位 16 **LDPC1ST**：先加载 PC (Load PC first)
0：如果 LSM 加载操作包含 PC，则 PC 不会排在列表中的前面位置先进行加载

位 15:12 **PROCFAM[3:0]**：处理器系列 (Processor family)
0xF：此寄存器中未标识处理器系列

位 11:8 **ARCHVMAJ[3:0]**：主要 ETM 架构版本 (Major ETM architecture version)
0x2：版本 3.x

位 7:4 **ARCHVMIN[3:0]**：次要 ETM 架构版本 (Minor ETM architecture version)
0x5：版本 x.5

位 3:0 **REVISION[3:0]**：设计版本 (Implementation revision)
0x0：版本 r0p0

ETM 配置代码扩展寄存器 (ETM_CCER)

ETM configuration code extension register

偏移地址：0x1E8

复位值：0x3440 08F2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	TSSIZE	TSENC	CNTFNC	VIRTEXT	Res.	Res.	Res.	TIMSTMP	ETMEIBCR	TSSUWP	EMBICEWPS[3:0]			
		r	r	r	r				r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INSTRES[2:0]			DACOMPS	ARR	PMUEVENTREGS[7:0]								EEISELS[2:0]		
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:30 保留，必须保持复位值。

位 29 **TSSIZE**：时间戳大小 (Timestamp size)
1：64 位。

位 28 **TSENC**：时间戳编码 (Timestamp encoding)
1：时间戳编码为自然二进制数

位 27 **CNTFNC**：计数器功能 (Counter function)
0：计数器为全功能计数器

位 26 **VIRTEXT**：虚拟扩展 (Virtual extensions)
1：支持虚拟扩展

位 25:23 保留，必须保持复位值。

位 22 **TIMSTMP**：时间戳 (Timestamping)
1：支持时间戳

位 21 **ETMEIBCR**：设计嵌入式 ICE 行为控制 (Embedded ICE behavior control implemented)
0：未设计

位 20 **TSSUWP**：跟踪开始/停止使用嵌入式 ICE 监测点输入 (Trace start/stop uses Embedded ICE watchpoint inputs)
0：不支持嵌入式 ICE 监测点输入

位 19:16 **EMBICEWPS[3:0]**：嵌入式 ICE 监测点输入 (Embedded ICE watchpoint inputs)
0：不支持嵌入式 ICE 监测点输入

位 15:13 **INSTRES[2:0]**：仪表资源 (Instrumentation resources)
0：None

位 12 **DACOMPS**：数据地址比较支持 (Data address comparison support)
0：不支持数据地址比较

位 11 **ARR**：可读寄存器 (Readable registers)
1：所有寄存器均可读

位 10:3 **PMUEVENTREGS[7:0]**：可用的 PMU 事件寄存器数量 (Number of PMU event registers available)
0x1E：30 个寄存器

位 2:0 **EEISELS[2:0]**：扩展外部输入选择器数量 (Number of extended external input selectors)
0x2：2 个选择器

ETM 扩展外部输入选择寄存器 (ETM_EXTINSELR)

ETM extended external input selection register

偏移地址：0x1EC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTINSEL2[7:0]								EXTINSEL1[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:8 **EXTINSEL2[7:0]**：扩展外部输入选择器 2 (Extended external input selector 2)

位 7:0 **EXTINSEL1[7:0]**：扩展外部输入选择器 1 (Extended external input selector 1)

ETM 时间戳事件寄存器 (ETM_TSEVR)

ETM timestamp event register

偏移地址：0x1F8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCN[1:0]		RESOURCEB[6:0]							RESOURCEA[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:14 **FCN[2:0]**：布尔函数 (Boolean function)

如果 A 定义为第一资源匹配，B 定义为第二资源匹配，则事件定义为 A 与 B 的函数。

0x0：A
0x1：NOT A
0x2：A AND B
0x3：NOT A AND B
0x4：NOT A AND NOT B
0x5：A OR B
0x6：NOT A OR B
0x7：NOT A OR NOT B

位 13:7 **RESOURCEB[6:0]**：第二资源标识符 (Second resource identifier)

关于位说明，请参见 RESOURCEA[6:0] 位字段。

位 6:0 **RESOURCEA[6:0]**：第一资源标识符 (First resource identifier)

位 [6:4] 定义资源类型，位 [3:0] 定义索引。下面列出了支持的资源标识符。编程任何其他值都可能导致不可预测的行为。

0x00：单地址比较器 1
...
0x0F：单地址比较器 16
0x10：地址范围比较器 1
...
0x17：地址范围比较器 8
0x18：仪表资源 1
0x19：仪表资源 2
0x1A：仪表资源 3
0x1B：仪表资源 4
0x20：嵌入式 ICE 监测点比较器 1（来自 DWT）
...
0x27：嵌入式 ICE 监测点比较器 8（来自 DWT）
0x40：值为 0 的计数器 1
0x41：值为 0 的计数器 2
0x42：值为 0 的计数器 3
0x43：值为 0 的计数器 4
0x50：状态 1 的定序器
0x51：状态 2 的定序器
0x52：状态 3 的定序器
0x58：上下文 ID 比较器 1
0x5B：VMID 比较器
0x5F：跟踪开始/停止资源
0x60：外部输入 1
0x61：外部输入 2
0x62：外部输入 3
0x63：外部输入 4
0x6D：非安全状态的处理器
0x6E：处理器禁止的跟踪
0x6F：硬接线输入，始终为真
其他值：保留

ETM 辅助控制寄存器 (ETM_AUXCR)

ETM auxiliary control register

偏移地址：0x1FC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DPKTTRCDIS	TSPKTGENSUP	DNFOVFL
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **DPKTTRCDIS**：禁止跟踪数据包 (Disable tracing of data packets)
0：不禁止
1：禁止

位 1 **TSPKTGENSUP**：抑制生成时间戳数据包 (Suppress generation of timestamp packets)
0：不抑制
1：执行 ISB 指令后抑制生成时间戳数据包

位 0 **DNFOVFL**：不强制 FIFO 上溢 (Do not force FIFO overflow)
0：如果已有两个完整同步周期没有输出周期性同步信号，则强制 FIFO 上溢
1：不强制 FIFO 上溢 (Do not force FIFO overflow)

ETM 跟踪 ID 寄存器 (ETM_TRACEIDR)

ETM trace ID register

偏移地址：0x200

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRACEID[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **TRACEID[7:0]**：输出到跟踪总线的跟踪 ID (Trace ID to output onto the trace bus)
该位字段应编程为唯一值，以将其与系统中的其他跟踪源区分。

ETM ID 2 寄存器 (ETM_IDR2)

ETM ID 2 register

偏移地址：0x208

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWP TO	RFE TO
														r	r

位 31:2 保留，必须保持复位值。

位 1 **SWPTO**：SWP 或 SWPB 指令顺序 (SWP or SWPB instruction order)
标识 SWP 或 SWPB 指令的顺序。
0：在存储传输之前跟踪加载传输

位 0 **RFETO**：RFE 指令顺序 (RFE instruction order)
标识 RFE 指令的顺序。
0：在 CPSR 传输之前跟踪 PC 传输

ETM VMID 比较器值寄存器 (ETM_VMIDCVR)

ETM VMID comparator value register

偏移地址：0x240

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VMID[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **VMID[7:0]**：与当前虚拟机 ID 进行比较的虚拟机 ID (Virtual machine ID for comparison with current one)

ETM OS 锁定访问寄存器 (ETM_OSLAR)

ETM OS lock access register

偏移地址：0x300

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEYVALUE[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYVALUE[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **KEYVALUE[31:0]**：ETM 寄存器锁定访问 (ETM register lock access)
0xC5ACCE55：禁止访问
其他值：使能访问

ETM OS 锁定状态寄存器 (ETM_OSLSR)

ETM OS lock status register

偏移地址：0x304

复位值：0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETM OSLAR	NTT	DBGR LKED	DBGR LKSUP
												r	r	r	r

位 31:4 保留，必须保持复位值。

位 3 **ETMOSLAR**：设计 ETM OS 锁定访问寄存器 (ETM OS lock access register implemented)
1：已设计

位 2 **NTT**：需要 32 位访问 (32-bit access required)
0：需要通过 32 位访问来操作 ETM_OSLAR

位 1 **DBGRLKED**：寄存器已锁定 (Registers locked)
0：ETM 寄存器未锁定
1：ETM 寄存器已锁定。任何访问都将返回从设备生成的错误响应

位 0 **DBGRLKSUP**：ETM 掉电支持 (ETM power down support)
0：完全支持

ETM OS 保存和恢复寄存器 (ETM_OSSRR)

ETM OS save & restore register

偏移地址：0x308

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VALUE[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **VALUE[31:0]**：OS 保存/恢复寄存器值 (OS save/restore register value)
在锁定寄存器后进行第一次读操作时，此寄存器包含保存或恢复 ETM 寄存器所需的额外访问次数。
在 OS 保存期间的后续读操作将返回每个寄存器的内容，以便存储在内存中。在 OS 恢复期间，后续写操作会依次将写入的值加载到每个寄存器中。

ETM 设备掉电控制寄存器 (ETM_PDCR)

ETM device power-down control register

偏移地址：0x310

复位值：0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRCPWREN	Res.	Res.	Res.
												rw			

位 31:4 保留，必须保持复位值。

位 3 **TRCPWREN**：使能跟踪电源 (Trace power enable)

1：必须为包含跟踪寄存器的域提供电源。

位 2:0 保留，必须保持复位值。

ETM 设备掉电状态寄存器 (ETM_PDSR)

ETM device power-down status register

偏移地址：0x314

复位值：0x0000 0023

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSLOCKST	Res.	Res.	Res.	STKYREGST	ETMPWRD
										r				r	r

位 31:6 保留，必须保持复位值。

位 5 **OSLOCKST**：OS 锁定状态 (OS lock status)

该位的值与 ETM_OSLSR 的位 [1] 的值相同。

位 4:2 保留，必须保持复位值。

位 1 **STKYREGST**：粘滞寄存器状态 (Sticky register state)

0：表示在复位后已读取此寄存器

1：表示在复位后未读取此寄存器

位 0 **ETMPWRD**：ETM 上电 (ETM powered)

1：ETM 已上电；可访问跟踪寄存器。

ETM 声明标记设置寄存器 (ETM_CLAIMSET)

ETM claim tag set register

偏移地址：0xFA0

复位值：0x0000 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **CLAIMSET[7:0]**：设置声明标记位 (Set claim tag bits)

写：

00000000：无影响

xxxxxxx1：将位 0 置 1

xxxxxx1x：将位 1 置 1

xxxxx1xx：将位 2 置 1

以此类推。

读：

0xFF：表示声明标记使用八个位

ETM 声明标记清零寄存器 (ETM_CLAIMCLR)

ETM claim tag clear register

偏移地址：0xFA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **CLAIMCLR[7:0]**：复位声明标记位 (Reset claim tag bits)

写：

00000000：无影响

xxxxxxx1：将位 0 清零

xxxxxx1x：将位 1 清零

xxxxx1xx：将位 2 清零

xxxx1xxx：将位 3 清零

以此类推。

读：返回声明标记的当前值

ETM 锁定访问寄存器 (ETM_LAR)

ETM lock access register

偏移地址：0xFB0

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ACCESS_W[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS_W[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **ACCESS_W[31:0]**：ETM 寄存器写访问 (ETM register write access)
使能处理器内核写访问某些 ETM 寄存器（调试器无需解锁组件）
0xC5ACCE55：使能写访问
其他值：禁止写访问

ETM 锁定状态寄存器 (ETM_LSR)

ETM lock status register

偏移地址：0xFB4

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCK TYPE	LOCK GRANT	LOCK EXIST
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **LOCKTYPE**：ETM_LAR 寄存器的大小 (Size of the ETM_LAR register)
0：32 位

位 1 **LOCKGRANT**：当前锁定状态 (Current status of lock)
当外部调试器读取该位时，该位始终返回零。
0：允许写访问
1：阻止写访问。仅允许读访问。

位 0 **LOCKEXIST**：存在锁定控制机制 (Existence of lock control mechanism)
该位指示是否存在锁定控制机制。当外部调试器读取该位时，该位始终返回零。
0：不存在锁定控制机制
1：存在锁定控制机制

ETM 认证状态寄存器 (ETM_AUTHSTAT)

ETM authentication status register

偏移地址：0xFB8

复位值：0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:6 **SNID[1:0]**：安全非侵入式调试的安全等级 (Security level for secure non-invasive debug)
0x：未使用
10：禁止安全非侵入式调试
11：使能安全非侵入式调试

位 5:4 **SID[1:0]**：安全侵入式调试的安全等级 (security level for secure invasive debug)
00：未设计

位 3:2 **NSNID[1:0]**：非安全非侵入式调试的安全等级 (Security level for non-secure non-invasive debug)
0x：未使用
10：禁止非安全非侵入式调试
11：使能非安全非侵入式调试

位 1:0 **NSID[1:0]**：非安全侵入式调试的安全等级 (Security level for non-secure invasive debug)
00：未设计

ETM Coresight 设备配置寄存器 (ETM_DEVID)

ETM Coresight device configuration register

偏移地址：0xFC8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DEVICEID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEVICEID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **DEVICEID[31:0]**：设备配置 (Device configuration)
0x0：无配置信息

ETM CoreSight 设备类型寄存器 (ETM_DEVTYPE)

ETM CoreSight device type register

偏移地址：0xFCC

复位值：0x0000 0013

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SUBTYPE[3:0]**：设备子类型标识符 (Device sub-type identifier)
0x1：处理器跟踪

位 3:0 **MAJORTYPE[3:0]**：设备主类型标识符 (Device main type identifier)
0x3：跟踪源

ETM CoreSight 外设标识寄存器 4 (ETM_PIDR4)

ETM CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **SIZE[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：Arm® JEDEC 代码

ETM CoreSight 外设标识寄存器 0 (ETM_PIDR0)

ETM CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0056

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位字段 (Part number field)，位 [7:0]
0x56：ETM 产品编号

ETM CoreSight 外设标识寄存器 1 (ETM_PIDR1)

ETM CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位字段 (JEP106 identity code field)，位 [3:0]
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位字段 (Part number field)，位 [11:8]
0x9：ETM 产品编号

ETM CoreSight 外设标识寄存器 2 (ETM_PIDR2)

ETM CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 000B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x0：r0p0

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位字段 (JEP106 identity code field)，位 [6:4]
0x3：Arm® JEDEC 代码

ETM CoreSight 外设标识寄存器 3 (ETM_PIDR3)

ETM CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

ETM CoreSight 组件标识寄存器 0 (ETM_CIDR0)

ETM CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位字段 (Component ID field)，位 [7:0]
0x0D：公共 ID 值

ETM CoreSight 组件标识寄存器 1 (ETM_CIDR1)

ETM CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位字段 (Component ID field)，位 [15:12]——组件类
0x9：跟踪发生器组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位字段 (Component ID field)，位 [11:8]
0x0：公共 ID 值

ETM CoreSight 组件标识寄存器 2 (ETM_CIDR2)

ETM CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位字段 (Component ID field)，位 [23:16]
0x05：公共 ID 值

ETM CoreSight 组件标识寄存器 3 (ETM_CIDR3)

ETM CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位字段 (Component ID field)，位 [31:24]
0xB1：公共 ID 值

嵌入式跟踪宏单元寄存器映射

表 618. ETM 寄存器映射和复位值

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**ETM_CR**	Res.	VMIDTRCEN	Res.	TSEN.	PROCSEL[2:0]			Res.	Res.	Res.	PORTSIZE[3]	DATAONLY	FILTERCPRT	SUPPDATA.	PORTMODE[1:0]		CONTXTIDSIZE[1:0]		PORTMODE[2]	CYCLEACC	ETMPORTSEL.	ETMPROG	DBGRQCTL	BRANCHOUTPUT	Res.	PORTSIZE[2:0]			DATAACC[1:0]		MONITORCPRT	ETMPDN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	1
0x004	**ETM_CCR**	IDREG	Res.	Res.	Res.	SWACC	STARTSTOP	CONTXTIDCMP[1:0]		FIFOFULL	Res.	EXTOUTPUTS[1:0]		EXTINPUTS[2:0]			SEQUENCER	COUNTERS[2:0]			MMAPDECS[4:0]					DATACOMPS[3:0]				ADCOMPPRS[3:0]			
	Reset value	1	0	0	0	1	1	0	1	0	0	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0
0x008	**ETM_TRIGGER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2:0]			RESOURCEB[6:0]						RESOURCEA[6:0]							
	Reset value																																
0x010	**ETM_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIG	STSTPRESSTAT	PROGBIT	UNTRCDOVFL
	Reset value																																
0x014	**ETM_SCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NOFTCHCMP	Res.	Res.	SUPPROC[2:0]		PMSUPP	PSSUPP	MAXPS[3]	FFSUPP	Res.	Res.	Res.	Res.	Res.	MAXPS[2:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0
0x018	**ETM_SSCR**	STOPADDRCOMPSEL[16:1]																STARTADDRCOMPSEL[16:1]															
	Reset value																																
0x01C	**ETM_TECR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INCEXCCTLCOMPSEL[16:1]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	**ETM_TEEVR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2:0]			RESOURCEB[6:0]						RESOURCEA[6:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	**ETM_TECR1**	Res.	Res.	Res.	Res.	Res.	Res.	TSSEN	INCEXCCTL	INCEXCMMDSEL[16:1]																	INCEXCADCOMPSEL[8:1]							
	Reset value							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x02C	**ETM_FFLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FIFOFULLLVL[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

表 618. ETM 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x030	**ETM_VDEVR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]							
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x034	**ETM_VDCR1**	VDEXCSADCOMPSEL[16:1]																VDINCSADCOMPSEL[16:1]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03C	**ETM_VDCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXCONLY	VDEXCADCOMPSEL[8:1]								VDINCADCOMPSEL[8:1]							
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x040	**ETM_ACVR1**	ADDRESS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x044	**ETM_ACVR2**	ADDRESS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x048	**ETM_ACVR3**	ADDRESS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04C	**ETM_ACVR4**	ADDRESS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x050	**ETM_ACVR5**	ADDRESS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x054	**ETM_ACVR6**	ADDRESS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x058	**ETM_ACVR7**	ADDRESS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x05C	**ETM_ACVR8**	ADDRESS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x080	**ETM_ACTR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VMIDCOMP	HYPMODE	STMODE[3:0]				CTXTIDCOMP[1:0]		EXACTMATCH	DCOMP[1:0]		COMPACCSIZE[1:0]		ACCTYPE		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	**ETM_ACTR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VMIDCOMP	HYPMODE	STMODE[3:0]				CTXTIDCOMP[1:0]		EXACTMATCH	DCOMP[1:0]		COMPACCSIZE[1:0]		ACCTYPE[2:0]		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 618. ETM 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x088	**ETM_ACTR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VMIDCOMP	HYPMODE	STMODE[3:0]				CTXTIDCOMP[1:0]		EXACTMATCH	DCOMP[1:0]		COMPACCSIZE[1:0]		ACCTYPE		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08C	**ETM_ACTR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VMIDCOMP	HYPMODE	STMODE[3:0]				CTXTIDCOMP[1:0]		EXACTMATCH	DCOMP[1:0]		COMPACCSIZE[1:0]		ACCTYPE		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x090	**ETM_ACTR5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VMIDCOMP	HYPMODE	STMODE[3:0]				CTXTIDCOMP[1:0]		EXACTMATCH	DCOMP[1:0]		COMPACCSIZE[1:0]		ACCTYPE		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x094	**ETM_ACTR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VMIDCOMP	HYPMODE	STMODE[3:0]				CTXTIDCOMP[1:0]		EXACTMATCH	DCOMP[1:0]		COMPACCSIZE[1:0]		ACCTYPE		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x098	**ETM_ACTR7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VMIDCOMP	HYPMODE	STMODE[3:0]				CTXTIDCOMP[1:0]		EXACTMATCH	DCOMP[1:0]		COMPACCSIZE[1:0]		ACCTYPE		
	Reset value																																
0x09C	**ETM_ACTR8**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VMIDCOMP	HYPMODE	STMODE[3:0]				CTXTIDCOMP[1:0]		EXACTMATCH	DCOMP[1:0]		COMPACCSIZE[1:0]		ACCTYPE		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C0	**ETM_DCVR1**	DATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C8	**ETM_DCVR3**	DATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 618. ETM 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x100	**ETM_DCMR1**	MASK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x108	**ETM_DCMR3**	MASK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x140	**ETM_CNTRLDVR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INITIALCOUNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x144	**ETM_CNTRLDVR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INITIALCOUNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x150	**ETM_CNTENR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN [2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]							
	Reset value															1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x154	**ETM_CNTENR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN [2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]							
	Reset value															1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x160	**ETM_ CNTRLDEVR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN [2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]							
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x164	**ETM_ CNTRLDEVR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN [2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]							
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x170	**ETM_CNTVR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COUNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x174	**ETM_CNTVR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COUNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x180	**ETM_SQ12EVR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN [2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]							
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x184	**ETM_SQ21EVR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN [2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]							
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x188	**ETM_SQ23EVR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN [2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]							
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18C	**ETM_SQ31EVR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN [2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]							
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x190	**ETM_SQ32EVR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN [2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]							
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x194	**ETM_SQ13EVR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN [2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]							
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 618. ETM 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x19C	**ETM_SQR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SQSTATE[1:0]	
	Reset value																															0	0
0x1A0	**ETM_ EXTOUTEVR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN [2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]						
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1A4	**ETM_ EXTOUTEVR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN [2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]						
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1B0	**ETM_CIDCVR1**	CONTEXTID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1BC	**ETM_CIDCMR**	MASK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1E0	**ETM_SYNCFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYNCFREQ[11:0]											
	Reset value																					0	1	0	0	0	0	0	0	0	0	0	0
0x1E4	**ETM_IDR**	IMPLEMENTER[7:0]								Res.	Res.	Res.	BPENC	SECEXT	T32	Res.	LDPC1ST	PROCFAM [3:0]				ARCHVMAJ [3:0]				ARCHVMIN [3:0]				REVISION [3:0]			
	Reset value	0	1	0	0	0	0	0	1				0	1	1		0	1	1	1	1	0	0	1	0	0	1	0	1	0	0	0	0
0x1E8	**ETM_CCER**	Res.	Res.	TSSIZE	TSENC	CNTFNC	VIRTEXT	Res.	Res.	Res.	TIMSTMP	ETMEIBCR	TSSBLK	EMBICEWPS[3:0]				INSTRES[2:0]			DACOMPS	ARR	PMUEVENTREGS[7:0]								EEISELS		
	Reset value			1	1	0	1				1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	0	1	0
0x1EC	**ETM_EXTINSELR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTINSEL2[7:0]								EXTINSEL1[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1F8	**ETM_TSEVR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN [2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]						
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1FC	**ETM_AUXCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DPKTTRCDIS	TSPKTGENSUP	DNFOVFL
	Reset value																														0	0	0
0x200	**ETM_TRACEIDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRACEID[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x208	**ETM_IDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWPTO	RFETO
	Reset value																															0	0

表 618. ETM 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x240	**ETM_VMIDCVR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VMID[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x300	**ETM_OSLAR**	KEYVALUE[31:0]																															
	Reset value																																
0x304	**ETM_OSLSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETMOSLAR	NTT	DBGRLKED	DBGRLKSUP
	Reset value																													1	0	1	0
0x308	**ETM_OSSRR**	VALUE[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x310	**ETM_PDCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRCPWREN	Res.	Res.	Res.
	Reset value																													1			
0x314	**ETM_PDSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSLOCKST	Res.	Res.	Res.	STKYREGST	ETMPWRD
	Reset value																											1				1	1
0xFA0	**ETM_CLAIMSET**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SET							
	Reset value																									1	1	1	1	1	1	1	1
0xFA4	**ETM_CLAIMCLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR							
	Reset value																									0	0	0	0	0	0	0	0
0xFB0	**ETM_LAR**	KEY																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0xFB4	**ETM_LSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NTT	SLK	SLI
	Reset value																																
0xFB8	**ETM_AUTHSTAT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID		SID		NSNID		NSID	
	Reset value																									0	0	0	0	1	1	0	0
0xFC8	**ETM_DEVID**	DEVICEID																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xFD0	**ETM_DEVTYPE**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUB TYPE				MAJOR TYPE			
	Reset value																									0	0	0	1	0	0	1	1
0xFD0	**ETM_PIDR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE				JEP106 CON			
	Reset value																									0	0	0	0	0	1	0	0

表 618. ETM 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFD4	**ETM_PIDR5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	**ETM_PIDR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	**ETM_PIDR7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	**ETM_PIDR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM [7:0]							
	Reset value																									0	1	0	1	0	1	1	0
0xFE4	**ETM_PIDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	1	0	0	1
0xFE8	**ETM_PIDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	0	0	1	0	1	1
0xFEC	**ETM_PIDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND				CMOD			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	**ETM_CIDR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	**ETM_CIDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS				PREAMBLE			
	Reset value																									1	0	0	1	0	0	0	0
0xFF8	**ETM_CIDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	**ETM_CIDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE							
	Reset value																									1	0	1	1	0	0	0	1

59.11.4 交叉触发接口 (CTI)

请参见*第 59.10.3 节*。

59.12 参考资料

1. IHI 0031C (ID080813) ——《Arm® 调试接口架构规范 ADIv5.0 到 ADIv5.2》，发布版本 C，2013 年 8 月 8 日。
2. DDI 0480F (ID100313) ——《Arm® CoreSight™ SoC-400 r3p1 技术参考手册》，发布版本 F，2013 年 9 月 26 日。
3. DDI 0461B (ID010111) ——《Arm® CoreSight™ 跟踪存储控制器 r0p1 技术参考手册》，发布版本 B，2010 年 12 月 10 日
4. DDI 0314H ——《Arm® CoreSight™ 组件技术参考手册》，发布版本 H，2009 年 7 月 10 日
5. DDI 0468A (ID101712) ——《Arm® CoreSight™ ETM™-A7 r0p0》，发布版本 A，2011 年 9 月 12 日
6. DDI 0464F (ID051113) ——《Arm® Cortex®-A7 MPCore™ r0p5 技术参考手册》，发布版本 F，2013 年 4 月 11 日
7. IHI 0022E (ID022613) ——《Arm AMBA AXI 和 ACE 协议规范》，发布版本 E，2013 年 2 月 22 日
8. Arm IHI 0033A ——《Arm AMBA 3 AHB-Lite 协议 v1.0》，发布版本 A，2006 年 6 月 6 日
9. Arm IHI 0024C (ID041610) ——《Arm AMBA 协议 v2.0》，发布版本 C，2010 年 4 月 13 日

60 设备电子签名

设备电子签名在硬件中设置并存储在 OTP 中（在 BSEC 寄存器中可见），可以使用调试接口读取，或由 CPU 读取。它包含出厂前编程的标识数据，这些数据允许用户固件或其他外部设备与组件的特性自动匹配。

60.1 唯一设备 ID 寄存器（96 位）(UID)

Unique device ID register

唯一设备标识符最适合：

- 用作序列号（USB 字符串序列号或其他终端应用程序）
- 在对内存进行编程前将唯一 ID 与软件加密原语和协议结合使用时用作安全密钥的一部分以提高 Flash 中代码的安全性
- 在安全启动等过程中。

96 位的唯一设备标识符提供了一个对于给定设备和任何上下文都唯一的参考号码。用户不能更改这些位。

基址：0x5C00 5000（APB5 上的 BSEC 基址）

偏移地址：0x234

复位值：0xXXXX XXXX

注： *X 是出厂前编程的。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **UID[31:0]**：31:0 唯一 ID 位 (31:0 unique ID bits)

偏移地址：0x238

复位值：0xXXXX XXXX

注： *X 是出厂前编程的。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[63:48]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[47:32]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **UID[63:32]**：63:32 唯一 ID 位 (63:32 unique ID bits)

偏移地址：0x23C

复位值：0xXXXX XXXX

注： *X 是出厂前编程的。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[95:80]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[79:64]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **UID[95:64]**：95:64 唯一 ID 位 (95:64 unique ID bits)

60.2 设备产品编号 (RPN)

基址：0x5C00 5000（APB5 上的 BSEC 基址）

偏移地址：0x204

复位值：0x0000 8XXX

注： *X 是出厂前编程的。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	RPN_coding[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12 保留，必须保持复位值。

位 11:0 **RPN_coding[11:0]**：

0x6C9：STM32MP131A
0x6C8：STM32MP131C
0xEC9：STM32MP131D
0xEC8：STM32MP131F
0x0C1：STM32MP133A
0x0C0：STM32MP133C
0x8C1：STM32MP133D
0x8C0：STM32MP133F
0x001：STM32MP135A
0x000：STM32MP135C
0x801：STM32MP135D
0x800：STM32MP135F
其他值：保留

60.3 设备版本

基址：0x5008 1000（Debug_APB 上的 DBGMCU 基址）

偏移地址：0x000

复位值：0xXXXX 6501

注： *X 是出厂前编程的。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REV_ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DEV_ID[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 **REV_ID[15:0]**：芯片版本 (Silicon revision)

位 15:12 保留，必须保持复位值。

位 11:0 **DEV_ID[11:0]**：设备 ID (Device ID)

0x501：STM32MP13xx

61 重要安全说明

意法半导体集团公司 (ST) 高度重视产品安全性，因此，本文档中所述的 ST 产品可能已通过各种安全认证机构的认证和/或可能已实施我们内部的安全措施（如本文所述）。但是，任何级别的安全认证和/或 ST 内部的安全措施都不能保证 ST 产品能够抵御所有形式的攻击。因此，ST 的每位客户都需自行负责确定 ST 产品中提供的安全级别是否满足自身的需求，而且除了 ST 产品本身的安全级别之外，还需要考虑在最终产品或应用中与其他组件和/或软件结合使用时的情况。特别要注意以下几点：

- ST 产品可能已通过一家或多家安全认证机构的认证，例如平台安全架构 (www.psacertified.org) 和/或物联网平台安全评估标准 (www.trustcb.com)。有关此处引用的 ST 产品是否已获得此类安全认证以及认证级别和当前认证状态的详细信息，请访问相关认证标准网站或访问 *www.st.com* 上的相关产品页以获取最新信息。由于 ST 产品的安全认证状态和/或级别可能会不时发生变化，客户应根据需要随时检查 ST 产品的安全认证状态/级别。如果未显示 ST 产品已获得特定安全标准认证，客户不应主观认为 ST 产品已通过认证。
- 认证机构有权评估、授予和撤销与 ST 产品相关的安全认证。因此，这些认证机构独立负责授予或撤销 ST 产品的安全认证，针对认证机构对任何 ST 产品所做的误判、评估、测试或其他活动，ST 不承担任何责任。
- 行业通用的加密算法（如 AES、DES 或 MD5）以及其他可与 ST 产品结合使用的开放标准技术均基于非 ST 开发的标准。对于此类加密算法或开放技术所存在的任何缺陷，或者可能甚至已经开发出的能够绕过、解密或破解此类算法或技术的任何方法，ST 不承担任何责任。
- 尽管 ST 产品已完成可靠的安全测试，但任何级别的认证都无法绝对保证抵御所有形式的攻击，其中包括未经测试的高级攻击、新的或未识别形式的攻击、在规范或预期用途之外使用 ST 产品时存在的任何形式攻击，或者与客户用来创建最终产品或应用的其他组件或软件结合使用时存在的攻击等。对于此类攻击，ST 不承担任何责任。因此，无论集成的安全功能和/或 ST 可能提供的任何信息或支持如何，每位客户都需独自负责确定测试的攻击抵御级别是否满足自身的需求，而且除了 ST 产品本身的安全级别之外，还需要考虑将 ST 产品整合到最终产品或应用中的情况。
- ST 产品（包括任何硬件、软件、文档等）的所有安全功能，包括但不限于 ST 添加的任何增强安全功能，均按“原样”提供。因此，除非适用的书面和签署的合同条款另有明确规定，否则在适用法律允许的范围内，ST 不提供任何明示或暗示的担保，包括但不限于对适销性或特定用途适用性的暗示担保。

62 版本历史

表 619. 文档版本历史

日期	版本	变更
2023 年 3 月 1 日	1	初始版本。
2023 年 6 月 19 日	2	**PWR：** 更新了*第 9.3.3 节：VSW 域*。 更新了*表 30：存在 VDD 时 VSW 域的功能*。 更新了*表 31：存在 VBAT 时 VSW 域的功能*。 **GPIO：** 更新了*第 11.3.16 节：在备份电源域中使用 GPIO 引脚*。 **DCMIPP：** 更新了*第 30.7 节：影子寄存器*。 更新了*第 30.10.3 节：DCMIPP IP-Plug 全局寄存器 3 (DCMIPP_IPGR3)*。 更新了*第 30.10.6 节：DCMIPP IP-Plug 客户端 x 寄存器 2 (DCMIPP_IPCxR2)*。 更新了*第 30.10.23 节：DCMIPP Pipe0 流控制配置寄存器 (DCMIPP_P0FCTCR)*。 更新了*第 30.10.40 节：DCMIPP 硬件配置寄存器 2 (DCMIPP_HWCFGR2)*。 **LTDC：** 更新了*第 31.4.5 节：YUV 到 RGB 颜色转换*。 **RNG：** 更新了*第 32.2 节：RNG 主要特性*。 更新了*第 32.3.3 节：随机数生成*。 更新了*第 32.3.8 节：RNG 低功耗使用*。 更新了*表 268：RNG 中断请求*。 更新了*第 32.5 节：RNG 处理时间*。 更新了*第 32.6.1 节：简介*。 更新了*第 32.6.2 节：验证条件*。 更新了*表 269：RNG 配置*。 更新了*第 32.6.3 节：数据采集*。 更新了*第 32.7.1 节：RNG 控制寄存器 (RNG_CR)*。 更新了*第 32.7.3 节：RNG 数据寄存器 (RNG_DR)*。 **TIM1/TIM8：** 更新了*第 38.3.22 节：编码器接口模式*。 更新了*第 38.4.20 节：TIMx 刹车和死区寄存器 (TIMx_BDTR)（x = 1、8）*。 **RTC：** 更新了*第 46.6.20 节：RTC 状态寄存器 (RTC_SR)*。

表 619. 文档版本历史（续）

日期	版本	变更
2023 年 6 月 19 日	2 （续）	更新了*第 46.6.21 节：RTC 非安全屏蔽中断状态寄存器 (RTC_MISR)*。 更新了*第 46.6.22 节：RTC 安全屏蔽中断状态寄存器 (RTC_SMISR)*。 **TAMP：** 更新了*第 47.3.11 节：Tamper 检测配置和初始化*。 **SPI：** 更新了*第 50.5.3 节：CRC 计算*。 **ETH：** 更新了*图 715：DMA 发送流程（标准模式）*。 更新了*第 57.4.4 节：多通道和多队列支持*。 更新了*第 57.5.10 节：MAC 管理计数器*。 更新了*表 549：MCD 时钟选择*。 更新了*第 57.7.1 节：低功耗管理*。 更新了*通道 x 控制寄存器 (ETH_DMACxCR)*一节。 更新了*通道丢失帧计数寄存器 (ETH_DMACMFCR)*一节。 已删除发送队列 0 量子权重寄存器。 更新了*Tx 队列 x 量子权重寄存器 (ETH_MTLTXQxQWR)*一节。 更新了*MDIO 地址寄存器 (ETH_MACMDIOAR)*一节。

索引

A

ADC_AWD2CR 1465
ADC_AWD3CR 1466
ADC_CALFACT 1467
ADC_CCR 1468
ADC_CFGR 1449
ADC_CFGR2 1453
ADC_CR 1446
ADC_DIFSEL 1466
ADC_DR 1462
ADC_HWCFGR0 1469
ADC_IER 1444
ADC_IPDR 1471
ADC_ISR 1442
ADC_JDRy 1465
ADC_JSQR 1462
ADC_OFRy 1464
ADC_OR 1467
ADC_SIDR 1471
ADC_SMPR1 1455
ADC_SMPR2 1455
ADC_SQR1 1458
ADC_SQR2 1459
ADC_SQR3 1460
ADC_SQR4 1461
ADC_TR1 1456
ADC_TR2 1457
ADC_TR3 1458
ADC_VERR 1470
AP_ACE 3314
AP_BD0 3313
AP_BD1 3313
AP_BD2 3314
AP_BD3 3314
AP_CFG 3315
AP_DRW 3312
AP_TAR 3312
AP0_CSW 3309
AP1_CSW 3311
APx_BASE 3315
APx_IDR 3316
AXIMC_COMP_ID_0 126
AXIMC_COMP_ID_1 127
AXIMC_COMP_ID_2 127
AXIMC_COMP_ID_3 128
AXIMC_M4_FN_MOD_LB 136
AXIMC_Mx_FN_MOD . .130, 133-134, 136, 138
AXIMC_Mx_FN_MOD_AHB 129, 131
AXIMC_Mx_FN_MOD2 . . . . . .128, 131
AXIMC_Mx_READ_QOS 129, 132-133, 135, 137
AXIMC_Mx_WRITE_QOS . . .130, 132, 134-135, 137
AXIMC_PERIPH_ID_0 124
AXIMC_PERIPH_ID_1 125
AXIMC_PERIPH_ID_2 125
AXIMC_PERIPH_ID_3 126
AXIMC_PERIPH_ID_4 122
AXIMC_PERIPH_ID_5 123
AXIMC_PERIPH_ID_6 123
AXIMC_PERIPH_ID_7 124

B

BSEC_DENABLE 172
BSEC_HWCFGR 179
BSEC_IPIDR 180
BSEC_JTAGIN 177
BSEC_JTAGOUT 178
BSEC_OTP_CONFIG 168
BSEC_OTP_CONTROL 169
BSEC_OTP_DATAx 178
BSEC_OTP_DISTURBEDx 173
BSEC_OTP_ERRORx 174
BSEC_OTP_LOCK 171
BSEC_OTP_SPLOCKx 175
BSEC_OTP_SRLOCKx 177
BSEC_OTP_STATUS 170
BSEC_OTP_SWLOCKx 176
BSEC_OTP_WRDATA 170
BSEC_OTP_WRLOCKEDx 174
BSEC_SCRATCH 178
BSEC_SIDR 180
BSEC_VERR 179

C

CRC_CR 1176
CRC_DR 1175
CRC_HWCFGR 1178
CRC_IDR 1175
CRC_INIT 1177
CRC_IPIDR 1179
CRC_POL 1177
CRC_SIDR 1179
CRC_VERR 1178
CRYP_CR 1797
CRYP_CSGCMCCMxR 1811

CRYP_CSGCMxR 1811
CRYP_DIN 1801
CRYP_DMACR 1802
CRYP_DOUT 1802
CRYP_HWCFGR 1812
CRYP_IMSCR 1803
CRYP_IPIDR 1813
CRYP_IV0LR 1809
CRYP_IV0RR 1809
CRYP_IV1LR 1810
CRYP_IV1RR 1810
CRYP_K0LR 1805
CRYP_K0RR 1805
CRYP_K1LR 1806
CRYP_K1RR 1806
CRYP_K2LR 1807
CRYP_K2RR 1807
CRYP_K3LR 1808
CRYP_K3RR 1808
CRYP_MISR 1804
CRYP_RISR 1803
CRYP_SIDR 1813
CRYP_SR 1800
CRYP_VERR 1812
CTI_APPCLEAR 3338
CTI_APPPULSE 3339
CTI_APPSET 3337
CTI_AUTHSTAT 3344
CTI_CHINSTS 3341
CTI_CHOUTSTS 3342
CTI_CIDR0 3348
CTI_CIDR1 3348
CTI_CIDR2 3349
CTI_CIDR3 3349
CTI_CLAIMCLR 3343
CTI_CLAIMSET 3342
CTI_CONTROL 3337
CTI_DEVID 3345
CTI_DEVTYPE 3346
CTI_GATE 3342
CTI_INENx 3339
CTI_INTACK 3337
CTI_LAR 3343
CTI_LSR 3344
CTI_OUTENx 3340
CTI_PIDR0 3346
CTI_PIDR1 3347
CTI_PIDR2 3347
CTI_PIDR3 3348
CTI_PIDR4 3346
CTI_TRGISTS 3340
CTI_TRGOSTS 3341

D

DBG_AUTHSTAT 3437
DBG_BCRx 3417
DBG_BVRx 3417
DBG_BXVRx 3421
DBG_CIDR0 3441
DBG_CIDR1 3442
DBG_CIDR2 3442
DBG_CIDR3 3443
DBG_CIDSR 3416
DBG_CLAIMCLR 3435
DBG_CLAIMSET 3435
DBG_CTR 3424
DBG_DEVID 3438
DBG_DEVID1 3438
DBG_DEVTYPE 3439
DBG_DFR0 3428
DBG_DIDR 3405
DBG_DRCR 3414
DBG_DSCR 3410
DBG_DTRRX 3410
DBG_DTRTX 3414
DBG_EACR 3415
DBG_ECR 3409
DBG_ISAR0 3431
DBG_ISAR1 3432
DBG_ISAR2 3432
DBG_ISAR3 3433
DBG_ISAR4 3434
DBG_ITR 3410
DBG_LAR 3436
DBG_LSR 3436
DBG_MIDR 3424
DBG_MMFR0 3428
DBG_MMFR1 3429
DBG_MMFR2 3430
DBG_MMFR3 3430
DBG_MPIDR 3425
DBG_OSLAR 3421
DBG_OSLSR 3421
DBG_PCSR 3415
DBG_PFR0 3426
DBG_PFR1 3427
DBG_PIDR0 3440
DBG_PIDR1 3440
DBG_PIDR2 3441
DBG_PIDR3 3441
DBG_PIDR4 3439
DBG_PRCR 3422
DBG_PRSR 3423
DBG_REVIDR 3426
DBG_TLBTR 3425

DBG_VCR 3407
DBG_VIDSR 3416
DBG_WCRx 3419
DBG_WFAR 3406
DBG_WVRx 3419
DBGMCU_APB1FZ 3395
DBGMCU_APB2FZ 3396
DBGMCU_APB3FZ 3397
DBGMCU_APB4FZ 3395
DBGMCU_APB5FZ 3397
DBGMCU_APB6FZ 3398
DBGMCU_CIDR0 3401
DBGMCU_CIDR1 3402
DBGMCU_CIDR2 3402
DBGMCU_CIDR3 3403
DBGMCU_CR 3394
DBGMCU_IDC 3394
DBGMCU_PIDR0 3399
DBGMCU_PIDR1 3400
DBGMCU_PIDR2 3400
DBGMCU_PIDR3 3401
DBGMCU_PIDR4 3399
DCMIPP_CMCR 1585
DCMIPP_CMFCR 1589
DCMIPP_CMFRCR 1585
DCMIPP_CMHWCFGR 1584
DCMIPP_CMIER 1586
DCMIPP_CMSR1 1587
DCMIPP_CMSR2 1588
DCMIPP_HWCFGR1 1604
DCMIPP_HWCFGR2 1603
DCMIPP_IPCxR1 1578
DCMIPP_IPCxR2 1578
DCMIPP_IPCxR3 1579
DCMIPP_IPGR1 1576
DCMIPP_IPGR2 1576
DCMIPP_IPGR3 1577
DCMIPP_IPGR8 1577
DCMIPP_IPIDR 1606
DCMIPP_P0CFCTCR 1599
DCMIPP_P0CPPCR 1601
DCMIPP_P0CPPM0AR1 1602
DCMIPP_P0CPPM0AR2 1603
DCMIPP_P0CSCSTR 1600
DCMIPP_P0CSCSZR 1600
DCMIPP_P0DCCNTR 1594
DCMIPP_P0DCLMTR 1594
DCMIPP_P0FCR 1598
DCMIPP_P0FCTCR 1592
DCMIPP_P0FSCR 1591
DCMIPP_P0HWCFGR 1590
DCMIPP_P0IER 1597
DCMIPP_P0PPCR 1595
DCMIPP_P0PPM0AR1 1596
DCMIPP_P0PPM0AR2 1596
DCMIPP_P0SCSTR 1592
DCMIPP_P0SCSZR 1593
DCMIPP_P0SR 1597
DCMIPP_PRCR 1579
DCMIPP_PRESCR 1581
DCMIPP_PRESUR 1582
DCMIPP_PRFCR 1584
DCMIPP_PRIER 1582
DCMIPP_PRSR 1583
DCMIPP_SIDR 1606
DCMIPP_VERR 1605
DDRCTRL_ADDRMAP1 254
DDRCTRL_ADDRMAP10 265
DDRCTRL_ADDRMAP11 266
DDRCTRL_ADDRMAP2 255
DDRCTRL_ADDRMAP3 257
DDRCTRL_ADDRMAP4 260
DDRCTRL_ADDRMAP5 261
DDRCTRL_ADDRMAP6 262
DDRCTRL_ADDRMAP9 264
DDRCTRL_CRCPARCTL0 222
DDRCTRL_CRCPARSTAT 223
DDRCTRL_DBG0 273
DDRCTRL_DBG1 274
DDRCTRL_DBGCAM 274
DDRCTRL_DBGCMD 276
DDRCTRL_DBGSTAT 277
DDRCTRL_DERATEEN 213
DDRCTRL_DERATEINT 214
DDRCTRL_DFILPCFG0 247
DDRCTRL_DFIMISC 252
DDRCTRL_DFIPHYMSTR 254
DDRCTRL_DFISTAT 253
DDRCTRL_DFITMG0 245
DDRCTRL_DFITMG1 246
DDRCTRL_DFIUPD0 250
DDRCTRL_DFIUPD1 251
DDRCTRL_DFIUPD2 252
DDRCTRL_DIMMCTL 228
DDRCTRL_DRAMTMG0 229
DDRCTRL_DRAMTMG1 230
DDRCTRL_DRAMTMG14 240
DDRCTRL_DRAMTMG15 241
DDRCTRL_DRAMTMG2 231
DDRCTRL_DRAMTMG3 234
DDRCTRL_DRAMTMG4 235
DDRCTRL_DRAMTMG5 236
DDRCTRL_DRAMTMG6 238
DDRCTRL_DRAMTMG7 239

DDRCTRL_DRAMTMG8 240
DDRCTRL_HWLPCTL 217
DDRCTRL_INIT0 224
DDRCTRL_INIT1 225
DDRCTRL_INIT2 225
DDRCTRL_INIT3 226
DDRCTRL_INIT4 227
DDRCTRL_INIT5 227
DDRCTRL_MRCTRL0 211
DDRCTRL_MRCTRL1 212
DDRCTRL_MRSTAT 213
DDRCTRL_MSTR 208
DDRCTRL_ODTCFG 266
DDRCTRL_ODTMAP 268
DDRCTRL_PCCFG 281
DDRCTRL_PCFGQOS0_0 286
DDRCTRL_PCFGQOS1_0 287
DDRCTRL_PCFGR_0 283
DDRCTRL_PCFGW_0 284
DDRCTRL_PCFGWQOS0_0 287
DDRCTRL_PCFGWQOS1_0 289
DDRCTRL_PCTRL_0 285
DDRCTRL_PERFHPR1 271
DDRCTRL_PERFLPR1 272
DDRCTRL_PERFWR1 272
DDRCTRL_POISONCFG 279
DDRCTRL_POISONSTAT 280
DDRCTRL_PSTAT 281
DDRCTRL_PWRCTL 215
DDRCTRL_PWRTMG 216
DDRCTRL_RFSHCTL0 218
DDRCTRL_RFSHCTL3 220
DDRCTRL_RFSHTMG 221
DDRCTRL_SCHED 269
DDRCTRL_SCHED1 270
DDRCTRL_STAT 210
DDRCTRL_SWCTL 278
DDRCTRL_SWSTAT 278
DDRCTRL_ZQCTL0 242
DDRCTRL_ZQCTL1 243
DDRCTRL_ZQCTL2 244
DDRCTRL_ZQSTAT 244
DDRPERFM_CCR 411
DDRPERFM_CFG 410
DDRPERFM_CNTx 414
DDRPERFM_CTL 410
DDRPERFM_HWCFG 414
DDRPERFM_ICR 413
DDRPERFM_ID 415
DDRPERFM_IER 412
DDRPERFM_ISR 412
DDRPERFM_SID 415
DDRPERFM_STATUS 411
DDRPERFM_TCNT 413
DDRPERFM_VER 414
DDRPHYC_ACDLLCR 368
DDRPHYC_ACIOCR 371
DDRPHYC_DCR 376
DDRPHYC_DDR3_MR0 380
DDRPHYC_DDR3_MR1 382
DDRPHYC_DDR3_MR2 384
DDRPHYC_DDR3_MR3 386
DDRPHYC_DLLGCR 366
DDRPHYC_DSGCR 373
DDRPHYC_DTAR 387
DDRPHYC_DTDR0 388
DDRPHYC_DTDR1 388
DDRPHYC_DTPR0 377
DDRPHYC_DTPR1 378
DDRPHYC_DTPR2 380
DDRPHYC_DXCCR 372
DDRPHYC_DXnDLLCR 396
DDRPHYC_DXnDQSTR 399
DDRPHYC_DXnDQTR 398
DDRPHYC_DXnGCR 393
DDRPHYC_DXnGSR0 395
DDRPHYC_DXnGSR1 395
DDRPHYC_GPR0 389
DDRPHYC_GPR1 389
DDRPHYC_LPDDR2_MR1 383
DDRPHYC_LPDDR2_MR2 385
DDRPHYC_LPDDR3_MR1 384
DDRPHYC_LPDDR3_MR2 386
DDRPHYC_ODTCR 387
DDRPHYC_PGCR 363
DDRPHYC_PGSR 365
DDRPHYC_PIR 360
DDRPHYC_PTR0 369
DDRPHYC_PTR1 369
DDRPHYC_PTR2 370
DDRPHYC_RIDR 360
DDRPHYC_ZQ0CR0 389
DDRPHYC_ZQ0CR1 390
DDRPHYC_ZQ0SR0 391
DDRPHYC_ZQ0SR1 392
DFSDM_CHyAWSCDR 1525
DFSDM_CHyCFGR1 1522
DFSDM_CHyCFGR2 1525
DFSDM_CHyDATINR 1527
DFSDM_CHyDLYR 1528
DFSDM_CHyWDATR 1526
DFSDM_FLTxAWCFR 1539
DFSDM_FLTxAWHTR 1538
DFSDM_FLTxAWLTR 1538

DFSDM_FLTxAWSR 1539
DFSDM_FLTxCNVTIMR 1541
DFSDM_FLTxCR1 1528
DFSDM_FLTxCR2 1531
DFSDM_FLTxEXMAX 1540
DFSDM_FLTxEXMIN 1540
DFSDM_FLTxFCR 1535
DFSDM_FLTxICR 1534
DFSDM_FLTxISR 1532
DFSDM_FLTxJCHGR 1535
DFSDM_FLTxJDATAR 1536
DFSDM_FLTxRDATAR 1537
DFSDM_HWCFGR 1542
DFSDM_IPIDR 1543
DFSDM_SIDR 1543
DFSDM_VERR 1542
DLYB_CFGR 1377
DLYB_CR 1377
DMA_HIFCR 1039
DMA_HISR 1038
DMA_HWCFGR1 1048
DMA_HWCFGR2 1047
DMA_IPDR 1050
DMA_LIFCR 1039
DMA_LISR 1037
DMA_SIDR 1050
DMA_SxCR 1040
DMA_SxFCR 1045
DMA_SxM0AR 1044
DMA_SxM1AR 1045
DMA_SxNDTR 1043
DMA_SxPAR 1044
DMA_VERR 1049
DMAMUX_CFR 1069
DMAMUX_CSR 1068
DMAMUX_CxCR 1066-1067
DMAMUX_HWCFGR1 1072
DMAMUX_HWCFGR2 1073
DMAMUX_IPIDR 1071
DMAMUX_RGCFR 1070
DMAMUX_RGSR 1070
DMAMUX_RGxCR 1069
DMAMUX_SIDR 1071
DMAMUX_VERR 1072
DP_ABORT 3300
DP_CTRLSTAT 3301
DP_DLCR 3303
DP_DLPIDR 3304
DP_DPIDR 3300
DP_RDBUFF 3306
DP_RESEND 3305
DP_SELECT 3305
DP_TARGETID 3304
DP_TARGETSEL 3306
DTS_CFGR1 1482
DTS_DR 1485
DTS_ICIFR 1488
DTS_ITENR 1487
DTS_ITR1 1485
DTS_OR 1489
DTS_RAMPVALR 1484
DTS_SR 1486
DTS_T0VALR1 1484

E

ETF_AUTHSTAT 3366
ETF_BUFWM 3361
ETF_CBUFLVL 3360
ETF_CIDR0 3370
ETF_CIDR1 3370
ETF_CIDR2 3371
ETF_CIDR3 3371
ETF_CLAIMCLR 3365
ETF_CLAIMSET 3364
ETF_CTL 3359
ETF_DEVID 3367
ETF_DEVTYPE 3367
ETF_FFCR 3362
ETF_FFSR 3361
ETF_LAR 3365
ETF_LBUFLVL 3360
ETF_LSR 3366
ETF_MODE 3359
ETF_PIDR0 3368
ETF_PIDR1 3369
ETF_PIDR2 3369
ETF_PIDR3 3370
ETF_PIDR4 3368
ETF_PSCR 3364
ETF_RRD 3357
ETF_RRP 3357
ETF_RSZ 3355
ETF_RWD 3359
ETF_RWP 3358
ETF_STS 3356
ETF_TRG 3358
ETH_DMAA4DACR 3147
ETH_DMAA4RXACR 3146
ETH_DMAA4TXACR 3145
ETH_DMAC0CARXBR 3160
ETH_DMAC0CARXDR 3159
ETH_DMAC0RXCR 3150
ETH_DMAC0RXDLAR 3153

ETH_DMAC0RXDTPR 3154
ETH_DMAC0RXIWTR 3158
ETH_DMAC0RXRLR 3155
ETH_DMACMFCR 3163
ETH_DMACxCATXBR 3160
ETH_DMACxCATXDR 3159
ETH_DMACxCR 3148
ETH_DMACxIER 3155
ETH_DMACxSR 3160
ETH_DMACxTXCR 3149
ETH_DMACxTXDLAR 3152
ETH_DMACxTXDTPR 3153
ETH_DMACxTXRLR 3154
ETH_DMADSR 3144
ETH_DMAISR 3143
ETH_DMALPIEI 3147
ETH_DMAMR 3141
ETH_DMASBMR 3142
ETH_MAC1USTCR 3212
ETH_MACA0HR 3226
ETH_MACACR 3256
ETH_MACARPAR 3225
ETH_MACATSNR 3257
ETH_MACATSSR 3257
ETH_MACAxHR 3227
ETH_MACAxLR 3227
ETH_MACCR 3182
ETH_MACCSRSWCR 3225
ETH_MACDR 3214
ETH_MACECR 3187
ETH_MACHT0R 3191
ETH_MACHT1R 3191
ETH_MACHWF0R 3215
ETH_MACHWF1R 3217
ETH_MACHWF2R 3220
ETH_MACHWF3R 3222
ETH_MACIER 3205
ETH_MACISR 3203
ETH_MACIVIR 3196
ETH_MACL3A00R 3241
ETH_MACL3A01R 3246
ETH_MACL3A10R 3241
ETH_MACL3A11R 3246
ETH_MACL3A20R 3242
ETH_MACL3A21R 3247
ETH_MACL3A30R 3242
ETH_MACL3A31R 3247
ETH_MACL3L4C0R 3238
ETH_MACL3L4C1R 3243
ETH_MACL4A0R 3240
ETH_MACL4A1R 3245
ETH_MACLCSR 3209
ETH_MACLETR 3212
ETH_MACLMIR 3267
ETH_MACLTCR 3211
ETH_MACMDIOAR 3223
ETH_MACMDIODR 3225
ETH_MACPCSR 3207
ETH_MACPFR 3188
ETH_MACPHYCSR 3213
ETH_MACPOCR 3264
ETH_MACPPSCR 3259, 3261
ETH_MACPPSIR 3263
ETH_MACPPSTTNR 3263
ETH_MACPPSTTSR 3262
ETH_MACPPSWR 3264
ETH_MACQ0TXFCR 3197
ETH_MACRWKPFR 3209
ETH_MACRXFCR 3198
ETH_MACRXQC0R 3200
ETH_MACRXQC1R 3201
ETH_MACRXQC2R 3202
ETH_MACRXQCR 3199
ETH_MACRXTXSR 3206
ETH_MACSPI0R 3265
ETH_MACSPI1R 3266
ETH_MACSPI2R 3266
ETH_MACSSIR 3250
ETH_MACSTNR 3251
ETH_MACSTNUR 3252
ETH_MACSTSR 3251
ETH_MACSTSUR 3252
ETH_MACTSAR 3253
ETH_MACTSCR 3248
ETH_MACTSEACR 3258
ETH_MACTSECNR 3259
ETH_MACTSIACR 3258
ETH_MACTSICNR 3259
ETH_MACTSSR 3254
ETH_MACTXTSSNR 3255
ETH_MACTXTSSSR 3256
ETH_MACVHTR 3194
ETH_MACVIR 3195
ETH_MACVR 3214
ETH_MACVTR 3192
ETH_MACWTR 3190
ETH_MMC_CONTROL 3228
ETH_MMC_RX_INTERRUPT 3229
ETH_MMC_RX_INTERRUPT_MASK 3232
ETH_MMC_TX_INTERRUPT 3230
ETH_MMC_TX_INTERRUPT_MASK 3233
ETH_MTLISR 3169
ETH_MTLOMR 3168
ETH_MTLQxICSR 3172

ETH_MTLRXQxCR 3177
ETH_MTLRXQxDR 3176
ETH_MTLRXQxMPOCR 3175
ETH_MTLRXQxOMR 3173
ETH_MTLTXQ1ECR 3178
ETH_MTLTXQ1HCR 3179
ETH_MTLTXQ1LCR 3179
ETH_MTLTXQ1SSCR 3179
ETH_MTLTXQxDR 3171
ETH_MTLTXQxESR 3172
ETH_MTLTXQxOMR 3169
ETH_MTLTXQxQWR 3178
ETH_MTLTXQxUR 3170
ETH_RX_ALIGNMENT_ERROR_PACKETS . . . 3235
ETH_RX_CRC_ERROR_PACKETS 3235
ETH_RX_LPI_TRAN_CNTR 3238
ETH_RX_LPI_USEC_CNTR 3237
ETH_RX_UNICAST_PACKETS_GOOD . . .3236
ETH_TX_LPI_TRAN_CNTR 3237
ETH_TX_LPI_USEC_CNTR 3236
ETH_TX_MULTIPLE_COLLISION_GOOD_PAC KETS 3234
ETH_TX_PACKET_COUNT_GOOD 3235
ETH_TX_SINGLE_COLLISION_GOOD_PACKE TS 3234
ETM_ACTRx 3481
ETM_ACVRx 3481
ETM_AUTHSTAT 3509
ETM_AUXCR 3502
ETM_CCER 3499
ETM_CCR 3471
ETM_CIDCMR 3498
ETM_CIDCVR1 3497
ETM_CIDR0 3513
ETM_CIDR1 3513
ETM_CIDR2 3514
ETM_CIDR3 3514
ETM_CLAIMCLR 3507
ETM_CLAIMSET 3507
ETM_CNTENRx 3483
ETM_CNTRLDEVRx 3485
ETM_CNTRLDVRx 3483
ETM_CNTVRx 3486
ETM_CR 3469
ETM_DCMRx 3483
ETM_DCVRx 3482
ETM_DEVID 3509
ETM_DEVTYPE 3510
ETM_EXTINSELR 3500
ETM_EXTOUTEVRx 3496
ETM_FFLR 3478
ETM_IDR 3498
ETM_IDR2 3503
ETM_LAR 3508
ETM_LSR 3508
ETM_OSLAR 3504
ETM_OSLSR 3505
ETM_OSSRR 3505
ETM_PDCR 3506
ETM_PDSR 3506
ETM_PIDR0 3511
ETM_PIDR1 3511
ETM_PIDR2 3512
ETM_PIDR3 3512
ETM_PIDR4 3510
ETM_SCR 3474
ETM_SQ12EVR 3487
ETM_SQ13EVR 3494
ETM_SQ21EVR 3488
ETM_SQ23EVR 3490
ETM_SQ31EVR 3491
ETM_SQ32EVR 3493
ETM_SQR 3496
ETM_SR 3473
ETM_SSCR 3475
ETM_SYNCFR 3498
ETM_TECR1 3477
ETM_TECR2 3475
ETM_TEEVR 3476
ETM_TRACEIDR 3503
ETM_TRIGGER 3472
ETM_TSEVR 3501
ETM_VDCR1 3480
ETM_VDCR3 3480
ETM_VDEVR 3478
ETM_VMIDCVR 3504
ETZPC_DECPROT_LOCKx 973
ETZPC_DECPROTx 972
ETZPC_HWCFGR 976
ETZPC_IDR 977
ETZPC_SIDR 977
ETZPC_TZMA0_SIZE 970
ETZPC_TZMA1_SIZE 971
ETZPC_VERR 976
EXTI_EMR1 1162
EXTI_EMR2 1163
EXTI_EMR3 1164
EXTI_EXTICR1 1154
EXTI_EXTICR2 1156
EXTI_EXTICR3 1158
EXTI_EXTICR4 1159
EXTI_FPR1 1147
EXTI_FPR2 1150

EXTI_FPR3 1153
EXTI_FTSR1 1146
EXTI_FTSR2 1149
EXTI_FTSR3 1152
EXTI_HWCFGR1 1166
EXTI_HWCFGRx 1164-1166
EXTI_IMR1 1161
EXTI_IMR2 1162
EXTI_IMR3 1163
EXTI_IPIDR 1167
EXTI_RPR1 1147
EXTI_RPR2 1150
EXTI_RPR3 1153
EXTI_RTSR1 1145
EXTI_RTSR2 1149
EXTI_RTSR3 1152
EXTI_SIDR 1167
EXTI_SWIER1 1146
EXTI_SWIER2 1149
EXTI_SWIER3 1153
EXTI_TZENR1 1148
EXTI_TZENR2 1151
EXTI_TZENR3 1154
EXTI_VERR 1167

F

FDCAN_CCCR 2754
FDCAN_CCU_CCFG 2812
FDCAN_CCU_CREL 2812
FDCAN_CCU_CSTAT 2814
FDCAN_CCU_CWD 2814
FDCAN_CCU_IE 2816
FDCAN_CCU_IR 2815
FDCAN_CREL 2751
FDCAN_DBTP 2752
FDCAN_ECR 2760
FDCAN_ENDN 2751
FDCAN_GFC 2770
FDCAN_HPMS 2773
FDCAN_IE 2766
FDCAN_ILE 2769
FDCAN_ILS 2768
FDCAN_IR 2763
FDCAN_NBTP 2755
FDCAN_NDAT1 2773
FDCAN_NDAT2 2774
FDCAN_PSR 2760
FDCAN_RWD 2753
FDCAN_RXBC 2776
FDCAN_RXESC 2779
FDCAN_RXF0A 2776
FDCAN_RXF0C 2774
FDCAN_RXF0S 2775
FDCAN_RXF1A 2779
FDCAN_RXF1C 2777
FDCAN_RXF1S 2778
FDCAN_SIDFC 2771
FDCAN_TDCR 2762
FDCAN_TEST 2753
FDCAN_TOCC 2758
FDCAN_TOCV 2759
FDCAN_TSCC 2757
FDCAN_TSCV 2758
FDCAN_TTCPT 2809
FDCAN_TTCSM 2809
FDCAN_TTCTC 2808
FDCAN_TTGTP 2799
FDCAN_TTIE 2802
FDCAN_TTILS 2804
FDCAN_TTIR 2800
FDCAN_TTLGT 2808
FDCAN_TTMLM 2795
FDCAN_TTOCF 2793
FDCAN_TTOCN 2797
FDCAN_TTOST 2805
FDCAN_TTRMC 2792
FDCAN_TTTMC 2792
FDCAN_TTTMK 2799
FDCAN_TTTS 2810
FDCAN_TURCF 2796
FDCAN_TURNA 2807
FDCAN_TXBAR 2783
FDCAN_TXBC 2780
FDCAN_TXBCF 2785
FDCAN_TXBCIE 2786
FDCAN_TXBCR 2784
FDCAN_TXBRP 2782
FDCAN_TXBTIE 2785
FDCAN_TXBTO 2784
FDCAN_TXEFA 2788
FDCAN_TXEFC 2786
FDCAN_TXEFS 2787
FDCAN_TXESC 2782
FDCAN_TXFQS 2781
FDCAN_XIDAM 2772
FDCAN_XIDFC 2771
FMC_BCHDSR0 1262
FMC_BCHDSR1 1262
FMC_BCHDSR2 1263
FMC_BCHDSR3 1263
FMC_BCHDSR4 1264
FMC_BCHICR 1260
FMC_BCHIER 1258

FMC_BCHISR 1259
FMC_BCHPBR1 1260
FMC_BCHPBR2 1261
FMC_BCHPBR3 1261
FMC_BCHPBR4 1261
FMC_BCRx 1217
FMC_BTRx 1220
FMC_BWTRx 1222
FMC_CSQAR1 1254
FMC_CSQAR2 1255
FMC_CSQCFGR1 1250
FMC_CSQCFGR2 1251
FMC_CSQCFGR3 1253
FMC_CSQCR 1249
FMC_CSQEMSR 1258
FMC_CSQICR 1257
FMC_CSQIER 1255
FMC_CSQISR 1256
FMC_HECCR 1246
FMC_HPR 1246
FMC_HWCFGR1 1226
FMC_HWCFGR2 1227
FMC_ICR 1249
FMC_IER 1247
FMC_IPIDR 1225
FMC_ISR 1248
FMC_PATT 1245
FMC_PCR 1241
FMC_PCSCNTR 1224
FMC_PMEM 1244
FMC_SIDR 1225
FMC_SR 1243
FMC_VERR 1225

G

GICC_ABPR 1116
GICC_AEOIR 1117
GICC_AHPPIR 1117
GICC_AIAR 1116
GICC_APR0 1118
GICC_BPR 1113
GICC_BPRNS 1113
GICC_CTLR 1111
GICC_CTLRNS 1112
GICC_DIR 1119
GICC_EOIR 1114
GICC_HPPIR 1115
GICC_IAR 1114
GICC_IIDR 1118
GICC_NSAPR0 1118
GICC_PMR 1113
GICC_RPR 1115
GICD_CIDR0 1106
GICD_CIDR1 1106
GICD_CIDR2 1107
GICD_CIDR3 1107
GICD_CPENDSGIRx 1102
GICD_CTLR 1088
GICD_CTLRNS 1089
GICD_ICACTIVERx 1093
GICD_ICENABLER0 1091
GICD_ICENABLERx 1092
GICD_ICFGR0 1097
GICD_ICFGR1 1098
GICD_ICFGRx 1099
GICD_ICPENDRx 1092
GICD_IGROUPRx 1090
GICD_IIDR 1090
GICD_IPRIORITYRx 1094
GICD_ISACTIVERx 1093
GICD_ISENABLER0 1090
GICD_ISENABLERx 1091
GICD_ISPENDRx 1092
GICD_ITARGETSRx 1095-1096
GICD_PIDR0 1105
GICD_PIDR1 1105
GICD_PIDR2 1105
GICD_PIDR3 1106
GICD_PIDR4 1104
GICD_PIDRx 1104
GICD_PPISR 1100
GICD_SGIR 1101
GICD_SPENDSGIRx 1103
GICD_SPISRx 1100
GICD_TYPER 1089
GICH_APR0 1125
GICH_EISR0 1125
GICH_ELSR0 1125
GICH_HCR 1122
GICH_LRx 1126
GICH_MISR 1124
GICH_VMCR 1123
GICH_VTR 1123
GICV_ABPR 1131
GICV_AEOIR 1132
GICV_AHPPIR 1133
GICV_AIAR 1132
GICV_APR0 1133
GICV_BPR 1129
GICV_CTLR 1128
GICV_DIR 1134
GICV_EOIR 1130
GICV_HPPIR 1131

GICV_IAR 1129
GICV_IIDR 1134
GICV_PMR 1128
GICV_RPR 1130
GPIOx_AFRH 924
GPIOx_AFRL 923
GPIOx_BRR 925
GPIOx_BSRR 922
GPIOx_HWCFGR0 931
GPIOx_HWCFGR1 930
GPIOx_HWCFGR10 926
GPIOx_HWCFGR2 930
GPIOx_HWCFGR3 930
GPIOx_HWCFGR4 929
GPIOx_HWCFGR5 929
GPIOx_HWCFGR6 928
GPIOx_HWCFGR7 928
GPIOx_HWCFGR8 927
GPIOx_HWCFGR9 926
GPIOx_IDR 921
GPIOx_IPIDR 932
GPIOx_LCKR 922
GPIOx_MODER 919
GPIOx_ODR 921
GPIOx_OSPEEDR 920
GPIOx_OTYPER 919
GPIOx_PUPDR 920
GPIOx_SIDR 932
GPIOx_VERR 931

H

HASH_CR 1831
HASH_CSRx 1839
HASH_DIN 1833
HASH_HRAx 1836
HASH_HRx 1836
HASH_HWCFGR 1839
HASH_IMR 1837
HASH_IPIDR 1840
HASH_SIDR 1840
HASH_SR 1837
HASH_STR 1834
HASH_VERR 1840
HDP_CTRL 3283
HDP_GPOCLR 3284
HDP_GPOSET 3284
HDP_GPOVAL 3285
HDP_IPIDR 3286
HDP_MUX 3283
HDP_SIDR 3286
HDP_VAL 3284
HDP_VERR 3285

I

I2C_CR1 2418
I2C_CR2 2420
I2C_HWCFGR 2431
I2C_ICR 2428
I2C_IPIDR 2432
I2C_ISR 2426
I2C_OAR1 2422
I2C_OAR2 2423
I2C_PECR 2429
I2C_RXDR 2430
I2C_SIDR 2432
I2C_TIMEOUTR 2425
I2C_TIMINGR 2424
I2C_TXDR 2430
I2C_VERR 2431
IWDG_EWCR 2263
IWDG_HWCFGR 2263
IWDG_IDR 2265
IWDG_KR 2259
IWDG_PR 2260
IWDG_RLR 2260
IWDG_SIDR 2265
IWDG_SR 2261
IWDG_VERR 2264
IWDG_WINR 2262

L

LPTIM_ARR 2226
LPTIM_CFGR 2222
LPTIM_CFGR2 2228
LPTIM_CMP 2226
LPTIM_CNT 2227
LPTIM_CR 2225
LPTIM_ICR 2221
LPTIM_IER 2221
LPTIM_ISR 2220
LPTIM_PIDR 2230
LPTIM_SIDR 2230
LPTIM_VERR 2229
LPTIM3_OR 2227
LPTIMx_HWCFGR 2229
LTDC_AWCR 1634
LTDC_BCCR 1640
LTDC_BPCR 1633
LTDC_CCRCR 1648
LTDC_CDSR 1644
LTDC_CPSR 1643
LTDC_ECRCR 1648

LTDC_EDCR 1644
LTDC_FUTR 1649
LTDC_GC1R 1637
LTDC_GC2R 1638
LTDC_GCCR 1639
LTDC_GCR 1635
LTDC_ICR 1642
LTDC_ICR2 1647
LTDC_IDR 1632
LTDC_IER 1640
LTDC_IER2 1645
LTDC_ISR 1641
LTDC_ISR2 1646
LTDC_L1AFBA0R 1660
LTDC_L1AFBA1R 1660
LTDC_L1AFBLNR 1661
LTDC_L1AFBLR 1661
LTDC_LCR 1632
LTDC_LIPCR 1643
LTDC_LIPCR2 1648
LTDC_LxBFCR 1656
LTDC_LxBLCR 1657
LTDC_LxC0R 1649
LTDC_LxC1R 1650
LTDC_LxCACR 1655
LTDC_LxCFBAR 1658
LTDC_LxCFBLNR 1659
LTDC_LxCFBLR 1659
LTDC_LxCKCR 1654
LTDC_LxCLUTWR 1662
LTDC_LxCR 1652
LTDC_LxCYR0R 1662
LTDC_LxCYR1R 1663
LTDC_LxDCCR 1655
LTDC_LxFPF0R 1663
LTDC_LxFPF1R 1664
LTDC_LxPCR 1657
LTDC_LxPFCR 1654
LTDC_LxRCR 1651
LTDC_LxWHPCR 1653
LTDC_LxWVPCR 1653
LTDC_SRCR 1639
LTDC_SSCR 1633
LTDC_TWCR 1634

M

MCE_CR 1881
MCE_EADDR 1885
MCE_HWCFGR1 1887
MCE_HWCFGR2 1886
MCE_HWCFGR3 1886
MCE_IACR 1882
MCE_IAIER 1883
MCE_IASR 1882
MCE_IPIDR 1888
MCE_MKEYRx 1885
MCE_PRIVCFGR 1883
MCE_REGCR 1884
MCE_SADDR 1884
MCE_SIDR 1889
MCE_SR 1881
MCE_VERR 1888
MDMA_CxBNDTR 1011
MDMA_CxBRUR 1013
MDMA_CxCR 1006
MDMA_CxDAR 1013
MDMA_CxESR 1005
MDMA_CxIFCR 1004
MDMA_CxISR 1003
MDMA_CxLAR 1014
MDMA_CxMAR 1015
MDMA_CxMDR 1016
MDMA_CxSAR 1012
MDMA_CxTBR 1015
MDMA_CxTCR 1008
MDMA_GISR0 1002
MDMA_SGISR0 1002

O

OTG_CID 2869
OTG_DAINT 2897
OTG_DAINTMSK 2898
OTG_DCFG 2890
OTG_DCTL 2891
OTG_DIEPCTLx 2901
OTG_DIEPDMAx 2905
OTG_DIEPEMPMSK 2901
OTG_DIEPINTx 2903
OTG_DIEPMSK 2895
OTG_DIEPTSIZ0 2905
OTG_DIEPTSIZx 2906
OTG_DIEPTXF0 2866
OTG_DIEPTXFx 2873
OTG_DOEPCTL0 2907
OTG_DOEPCTLx 2911
OTG_DOEPDMAx 2911
OTG_DOEPINTx 2908
OTG_DOEPMSK 2896
OTG_DOEPTSIZ0 2910
OTG_DOEPTSIZx 2913
OTG_DSTS 2894
OTG_DTHRCTL 2900

OTG_DTXFSTSx 2906
OTG_DVBUSDIS 2899
OTG_DVBUSPULSE 2899
OTG_GAHBCFG 2849
OTG_GCCFG 2868
OTG_GINTMSK 2858
OTG_GINTSTS 2854
OTG_GLPMCFG 2869
OTG_GOTGCTL 2845
OTG_GOTGINT 2848
OTG_GRSTCTL 2852
OTG_GRXFSIZ 2865
OTG_GRXSTSP 2863-2864
OTG_GRXSTSR 2861-2862
OTG_GUSBCFG 2850
OTG_HAINT 2877
OTG_HAINTMSK 2878
OTG_HCCHARx 2881
OTG_HCDMABx 2889
OTG_HCDMASGx 2888
OTG_HCDMAx 2888
OTG_HCFG 2874
OTG_HCINTMSKx 2884
OTG_HCINTx 2883
OTG_HCSPLTx 2882
OTG_HCTSIZSGx 2886
OTG_HCTSIZx 2886
OTG_HFIR 2875
OTG_HFLBADDR 2878
OTG_HFNUM 2876
OTG_HNPTXFSIZ 2866
OTG_HNPTXSTS 2867
OTG_HPRT 2879
OTG_HPTXFSIZ 2873
OTG_HPTXSTS 2876
OTG_PCGCCTL 2914

P

PKA_CLRFR 1873
PKA_CR 1870
PKA_HWCFGR 1874
PKA_IPIDR 1875
PKA_SIDR 1875
PKA_SR 1872
PKA_VERR 1874
PM_AUTHSTAT 3462
PM_CCNTR 3449
PM_CEID0 3457
PM_CFGR 3455
PM_CNTENCLR 3451
PM_CNTENSET 3451
PM_CR 3456
PM_DEVTYPE 3462
PM_INTENCLR 3453
PM_INTENSET 3452
PM_LAR 3461
PM_LSR 3461
PM_OVSR 3453
PM_PIDR0 3463
PM_PIDR1 3464
PM_PIDR4 3463
PM_SWINC 3454
PM_USERENR 3457
PM_XEVCNTRx 3448
PM_XEVTYPER31 3450
PM_XEVTYPERx 3449
PMU_CIDR0 3465
PMU_CIDR1 3465
PMU_CIDR2 3466
PMU_CIDR3 3466
PMU_PIDR2 3464
PMU_PIDR3 3464
PWR_CR1 461
PWR_CR2 463
PWR_CR3 465
PWR_CSR1 463
PWR_ID 471
PWR_MPUCR 467
PWR_MPUWKUPENR 470
PWR_SID 472
PWR_VERR 471
PWR_WKUPCR 468
PWR_WKUPFR 470

Q

QUADSPI_ABR 1292
QUADSPI_AR 1291
QUADSPI_CCR 1289
QUADSPI_CR 1283
QUADSPI_DCR 1286
QUADSPI_DLR 1288
QUADSPI_DR 1292
QUADSPI_FCR 1288
QUADSPI_HWCFGR 1295
QUADSPI_IPIDR 1296
QUADSPI_LPTR 1294
QUADSPI_PIR 1294
QUADSPI_PSMAR 1293
QUADSPI_PSMKR 1293
QUADSPI_SIDR 1296
QUADSPI_SR 1287
QUADSPI_VERR 1295

R

RCC_ADC12CKSELR 691
RCC_AHB2RSTCLRR 727
RCC_AHB2RSTSETR 725
RCC_AHB2SECSR 876
RCC_AHB4RSTCLRR 731
RCC_AHB4RSTSETR 729
RCC_AHB5RSTCLRR 735
RCC_AHB5RSTSETR 733
RCC_AHB5SECSR 877
RCC_AHB6RSTCLRR 739
RCC_AHB6RSTSETR 737
RCC_AHB6SECSR 879
RCC_APB1DIVR 660
RCC_APB1RSTCLRR 704
RCC_APB1RSTSETR 701
RCC_APB2DIVR 661
RCC_APB2RSTCLRR 708
RCC_APB2RSTSETR 707
RCC_APB3DIVR 662
RCC_APB3RSTCLRR 712
RCC_APB3RSTSETR 710
RCC_APB3SECSR 871
RCC_APB4DIVR 663
RCC_APB4RSTCLRR 715
RCC_APB4RSTSETR 714
RCC_APB4SECSR 871
RCC_APB5DIVR 664
RCC_APB5RSTCLRR 718
RCC_APB5RSTSETR 717
RCC_APB5SECSR 872
RCC_APB6DIVR 665
RCC_APB6RSTCLRR 722
RCC_APB6RSTSETR 719
RCC_APB6SECSR 874
RCC_ASSCKSELR 653
RCC_AXIDIVR 658
RCC_BDCR 608
RCC_BR_RSTSCLRR 594
RCC_CPERCKSELR 655
RCC_CSICFGR 619
RCC_DBGCFGR 622
RCC_DCMIPPCKSELR 700
RCC_DDRITFCR 669
RCC_ETH12CKSELR 693
RCC_FDCANCKSELR 689
RCC_FMCCKSELR 697
RCC_HSICFGR 618
RCC_I2C12CKSELR 674
RCC_I2C345CKSELR 675
RCC_IDR 881
RCC_LPTIM1CKSELR 684
RCC_LPTIM23CKSELR 685
RCC_LPTIM45CKSELR 686
RCC_MCO1CFGR 620
RCC_MCO2CFGR 621
RCC_MLAHBDIVR 659
RCC_MP_AHB2ENCLRR 778
RCC_MP_AHB2ENSETR 776
RCC_MP_AHB2LPENCLRR 839
RCC_MP_AHB2LPENSETR 837
RCC_MP_AHB5ENCLRR 790
RCC_MP_AHB5ENSETR 788
RCC_MP_AHB5LPENCLRR 851
RCC_MP_AHB5LPENSETR 849
RCC_MP_AHB6ENCLRR 795
RCC_MP_AHB6ENSETR 792
RCC_MP_AHB6LPENCLRR 857
RCC_MP_AHB6LPENSETR 853
RCC_MP_APB1ENCLRR 744
RCC_MP_APB1ENSETR 741
RCC_MP_APB1LPENCLRR 805
RCC_MP_APB1LPENSETR 802
RCC_MP_APB2ENCLRR 749
RCC_MP_APB2ENSETR 747
RCC_MP_APB2LPENCLRR 810
RCC_MP_APB2LPENSETR 808
RCC_MP_APB3ENCLRR 752
RCC_MP_APB3ENSETR 751
RCC_MP_APB3LPENCLRR 813
RCC_MP_APB3LPENSETR 812
RCC_MP_APB4ENCLRR 760
RCC_MP_APB4ENSETR 758
RCC_MP_APB4LPENCLRR 821
RCC_MP_APB4LPENSETR 819
RCC_MP_APB5ENCLRR 768
RCC_MP_APB5ENSETR 766
RCC_MP_APB5LPENCLRR 829
RCC_MP_APB5LPENSETR 827
RCC_MP_APB6ENCLRR 773
RCC_MP_APB6ENSETR 770
RCC_MP_APB6LPENCLRR 834
RCC_MP_APB6LPENSETR 831
RCC_MP_APRSTCR 590
RCC_MP_APRSTSR 591
RCC_MP_CIER 602
RCC_MP_CIFR 605
RCC_MP_GRSTCSETR 593
RCC_MP_IWDGFZCLRR 601
RCC_MP_IWDGFZSETR 600
RCC_MP_MLAHBLPENCLRR 870
RCC_MP_MLAHBLPENSETR 869
RCC_MP_NS_AHB4ENCLRR 786
RCC_MP_NS_AHB4ENSETR 784

RCC_MP_NS_AHB4LPENCLRR 847
RCC_MP_NS_AHB4LPENSETR 845
RCC_MP_NS_AHB6ENCLRR 801
RCC_MP_NS_AHB6ENSETR 800
RCC_MP_NS_AHB6LPENCLRR 864
RCC_MP_NS_AHB6LPENSETR 863
RCC_MP_NS_APB3ENCLRR 757
RCC_MP_NS_APB3ENSETR 756
RCC_MP_NS_APB3LPENCLRR 818
RCC_MP_NS_APB3LPENSETR 817
RCC_MP_NS_APB4ENCLRR 765
RCC_MP_NS_APB4ENSETR 764
RCC_MP_NS_APB4LPENCLRR 826
RCC_MP_NS_APB4LPENSETR 825
RCC_MP_NS_AXIMLPENCLRR 868
RCC_MP_NS_AXIMLPENSETR 867
RCC_MP_RSTSCLRR 598
RCC_MP_RSTSSETR 596
RCC_MP_S_AHB4ENCLRR 782
RCC_MP_S_AHB4ENSETR 780
RCC_MP_S_AHB4LPENCLRR 843
RCC_MP_S_AHB4LPENSETR 841
RCC_MP_S_AHB6ENCLRR 799
RCC_MP_S_AHB6ENSETR 798
RCC_MP_S_AHB6LPENCLRR 862
RCC_MP_S_AHB6LPENSETR 861
RCC_MP_S_APB3ENCLRR 755
RCC_MP_S_APB3ENSETR 754
RCC_MP_S_APB3LPENCLRR 816
RCC_MP_S_APB3LPENSETR 815
RCC_MP_S_APB4ENCLRR 763
RCC_MP_S_APB4ENSETR 762
RCC_MP_S_APB4LPENCLRR 824
RCC_MP_S_APB4LPENSETR 823
RCC_MP_S_AXIMLPENCLRR 866
RCC_MP_S_AXIMLPENSETR 865
RCC_MP_SREQCLRR 589
RCC_MP_SREQSETR 588
RCC_MPCKDIVR 657
RCC_MPCKSELR 652
RCC_MSSCKSELR 654
RCC_OCENCLRR 614
RCC_OCENSETR 612
RCC_OCRDYR 616
RCC_PLL1CFGR1 628
RCC_PLL1CFGR2 629
RCC_PLL1CR 626
RCC_PLL1CSGR 631
RCC_PLL1FRACR 630
RCC_PLL2CFGR1 634
RCC_PLL2CFGR2 635
RCC_PLL2CR 632
RCC_PLL2CSGR 637
RCC_PLL2FRACR 636
RCC_PLL3CFGR1 640
RCC_PLL3CFGR2 642
RCC_PLL3CR 638
RCC_PLL3CSGR 644
RCC_PLL3FRACR 643
RCC_PLL4CFGR1 647
RCC_PLL4CFGR2 649
RCC_PLL4CR 645
RCC_PLL4CSGR 651
RCC_PLL4FRACR 650
RCC_PWRLPDLYCR 592
RCC_QSPICKSELR 696
RCC_RCK12SELR 623
RCC_RCK3SELR 624
RCC_RCK4SELR 625
RCC_RDLSICR 610
RCC_RNG1CKSELR 698
RCC_RTCDIVR 656
RCC_SAESCKSELR 701
RCC_SAI1CKSELR 687
RCC_SAI2CKSELR 688
RCC_SDMMC12CKSELR 692
RCC_SECCFGR 586
RCC_SIDR 881
RCC_SPDIFCKSELR 690
RCC_SPI2S1CKSELR 676
RCC_SPI2S23CKSELR 677
RCC_SPI45CKSELR 678
RCC_STGENCKSELR 699
RCC_TIMG1PRER 666
RCC_TIMG2PRER 667
RCC_TIMG3PRER 668
RCC_UART12CKSELR 679
RCC_UART35CKSELR 680
RCC_UART4CKSELR 681
RCC_UART6CKSELR 682
RCC_UART78CKSELR 683
RCC_USBCKSELR 695
RCC_VERR 880
RNG_CR 1680
RNG_DR 1683
RNG_HTCR 1684
RNG_HWCFGR 1684
RNG_IPIDR 1685
RNG_NSCR 1683
RNG_SIDR 1686
RNG_SR 1682
RNG_VERR 1685
RPN 3524
RTC_ALRABINR 2314

RTC_ALRBBINR 2314
RTC_ALRMAR 2305
RTC_ALRMASSR 2306
RTC_ALRMBR 2307
RTC_ALRMBSSR 2308
RTC_CALR 2301
RTC_CR 2294
RTC_DR 2289
RTC_HWCFGR1 2315
RTC_HWCFGR2 2315
RTC_ICSR 2291
RTC_IPIDR 2317
RTC_MISR 2310
RTC_OR 2313
RTC_PRER 2293
RTC_PRIVCFGR 2298
RTC_SCR 2312
RTC_SECCFGR 2299
RTC_SHIFTR 2302
RTC_SIDR 2317
RTC_SMISR 2311
RTC_SR 2309
RTC_SSR 2290
RTC_TR 2288
RTC_TSDR 2304
RTC_TSSSR 2304
RTC_TSTR 2303
RTC_VERR 2316
RTC_WPR 2300
RTC_WUTR 2294

S

SAES_CR 1731
SAES_DINR 1736
SAES_DOUTR 1736
SAES_HWCFGR 1745
SAES_ICR 1744
SAES_IER 1742
SAES_IPIDR 1746
SAES_ISR 1743
SAES_IVR0 1738
SAES_IVR1 1739
SAES_IVR2 1739
SAES_IVR3 1740
SAES_KEYR0 1737
SAES_KEYR1 1737
SAES_KEYR2 1738
SAES_KEYR3 1738
SAES_KEYR4 1740
SAES_KEYR5 1740
SAES_KEYR6 1741
SAES_KEYR7 1741
SAES_SIDR 1746
SAES_SR 1734
SAES_SUSPxR 1741
SAES_VERR 1745
SAI_ACLRFR 2647
SAI_ACR1 2629
SAI_ACR2 2634
SAI_ADR 2649
SAI_AFRCR 2637
SAI_AIM 2641
SAI_ASLOTR 2639
SAI_ASR 2644
SAI_BCLRFR 2648
SAI_BCR1 2631
SAI_BCR2 2635
SAI_BDR 2650
SAI_BFRCR 2638
SAI_BIM 2643
SAI_BSLOTR 2640
SAI_BSR 2646
SAI_GCR 2628
SAI_HWCFGR 2654
SAI_IPIDR 2655
SAI_PDMCR 2650
SAI_PDMDLY 2652
SAI_SIDR 2656
SAI_VERR 2655
SDMMC_ACKTIMER 1366
SDMMC_ARGR 1352
SDMMC_CLKCR 1350
SDMMC_CMDR 1352
SDMMC_DCNTR 1358
SDMMC_DCTRL 1356
SDMMC_DLENR 1355
SDMMC_DTIMER 1355
SDMMC_FIFORx 1366
SDMMC_ICR 1361
SDMMC_IDMABAR 1369
SDMMC_IDMABASER 1368
SDMMC_IDMABSIZER 1367
SDMMC_IDMACTRLR 1367
SDMMC_IDMALAR 1368
SDMMC_IPIDR 1370
SDMMC_MASKR 1363
SDMMC_POWER 1349
SDMMC_RESPCMDR 1354
SDMMC_RESPxR 1354
SDMMC_SIDR 1370
SDMMC_STAR 1358
SDMMC_VERR 1370
SPDIFRX_CR 2680

SPDIFRX_CSR 2687
SPDIFRX_DIR 2688
SPDIFRX_FMT0_DR 2685
SPDIFRX_FMT1_DR 2686
SPDIFRX_FMT2_DR 2687
SPDIFRX_IFCR 2685
SPDIFRX_IMR 2682
SPDIFRX_IPIDR 2689
SPDIFRX_SIDR 2690
SPDIFRX_SR 2683
SPDIFRX_VERR 2689
SPI_CFG1 2573
SPI_CFG2 2576
SPI_CR1 2571
SPI_CR2 2573
SPI_CRCPOLY 2583
SPI_I2S_HWCFGR 2587
SPI_I2SCFGR 2585
SPI_IER 2578
SPI_IFCR 2581
SPI_IPIDR 2589
SPI_RXCRC 2584
SPI_RXDR 2583
SPI_SIDR 2589
SPI_SR 2579
SPI_TXCRC 2584
SPI_TXDR 2582
SPI_UDRDR 2585
SPI_VERR 2588
STGENC_CIDR0 2241
STGENC_CIDR1 2241
STGENC_CIDR2 2242
STGENC_CIDR3 2242
STGENC_CNTCR 2235
STGENC_CNTCVL 2236
STGENC_CNTCVU 2236
STGENC_CNTFID0 2237
STGENC_CNTSR 2235
STGENC_PIDR0 2239
STGENC_PIDR1 2239
STGENC_PIDR2 2240
STGENC_PIDR3 2240
STGENC_PIDR4 2237
STGENC_PIDR5 2238
STGENC_PIDR6 2238
STGENC_PIDR7 2238
STGENR_CIDR0 2247
STGENR_CIDR1 2248
STGENR_CIDR2 2248
STGENR_CIDR3 2249
STGENR_CNTCVL 2243
STGENR_CNTCVU 2243
STGENR_PIDR0 2245
STGENR_PIDR1 2246
STGENR_PIDR2 2246
STGENR_PIDR3 2247
STGENR_PIDR4 2244
STGENR_PIDR5 2244
STGENR_PIDR6 2245
STGENR_PIDR7 2245
SYSCFG_BOOTCR 945
SYSCFG_BOOTR 939
SYSCFG_CBR 949
SYSCFG_CMPCR 947
SYSCFG_CMPENCLRR 949
SYSCFG_CMPENSETR 948
SYSCFG_CMPSD1CR 950
SYSCFG_CMPSD1ENCLRR 952
SYSCFG_CMPSD1ENSETR 951
SYSCFG_CMPSD2CR 952
SYSCFG_CMPSD2ENCLRR 955
SYSCFG_CMPSD2ENSETR 954
SYSCFG_HSLVENxR 956
SYSCFG_IDC 956
SYSCFG_IPIDR 957
SYSCFG_PMCCLRR 942
SYSCFG_PMCSETR 940
SYSCFG_SIDR 958
SYSCFG_SRAM3ERASER 945
SYSCFG_SRAM3KR 946
SYSCFG_VERR 957
SYSROM_CIDR0 3323
SYSROM_CIDR1 3323
SYSROM_CIDR2 3324
SYSROM_CIDR3 3324
SYSROM_MEMTYPE 3321
SYSROM_PIDR0 3321
SYSROM_PIDR1 3322
SYSROM_PIDR2 3322
SYSROM_PIDR3 3323
SYSROM_PIDR4 3321

T

TAMP_ATCR1 2343
TAMP_ATCR2 2347
TAMP_ATOR 2346
TAMP_ATSEEDR 2346
TAMP_BKPxR 2364
TAMP_COUNT1R 2361
TAMP_COUNT2R 2362
TAMP_CR1 2336
TAMP_CR2 2338
TAMP_CR3 2341

TAMP_ERCFGR 2363
TAMP_FLTCR 2342
TAMP_HWCFGR1 2365
TAMP_HWCFGR2 2364
TAMP_IER 2352
TAMP_IPIDR 2366
TAMP_MISR 2356
TAMP_OR 2362
TAMP_PRIVCFGR 2351
TAMP_SCR 2360
TAMP_SECCFGR 2350
TAMP_SIDR 2367
TAMP_SMISR 2358
TAMP_SR 2354
TAMP_VERR 2366
TIM1_AF1 1977
TIM1_AF2 1978
TIM1_TISEL 1981
TIM12_ARR 2096
TIM12_CCER 2094
TIM12_CCMR1 . . . . . .2090, 2092
TIM12_CCR1 2096
TIM12_CCR2 2097
TIM12_CNT 2095
TIM12_CR1 2084
TIM12_CR2 2086
TIM12_DIER 2088
TIM12_EGR 2089
TIM12_PSC 2096
TIM12_SMCR 2086
TIM12_SR 2088
TIM12_TISEL 2097
TIM13_TISEL 2108
TIM14_TISEL 2109
TIM15_AF1 2167
TIM15_ARR 2161
TIM15_BDTR 2163
TIM15_CCER 2158
TIM15_CCMR1 2155-2156
TIM15_CCR1 2162
TIM15_CCR2 2163
TIM15_CNT 2161
TIM15_CR1 2147
TIM15_CR2 2148
TIM15_DCR 2166
TIM15_DIER 2151
TIM15_DMAR 2166
TIM15_EGR 2154
TIM15_PSC 2161
TIM15_RCR 2162
TIM15_SMCR 2150
TIM15_SR 2152
TIM15_TISEL 2167
TIM16_AF1 2186
TIM16_TISEL 2187
TIM17_AF1 2187
TIM17_TISEL 2188
TIM2_AF1 2052
TIM2_TISEL 2054
TIM3_AF1 2053
TIM3_TISEL 2055
TIM4_AF1 2053
TIM4_TISEL 2056
TIM5_AF1 2054
TIM5_TISEL 2056
TIM8_AF1 1979
TIM8_AF2 1980
TIM8_TISEL 1982
TIMx_ARR 1967, 2048, 2107, 2181, 2202
TIMx_BDTR 1970, 2182
TIMx_CCER 1964, 2046, 2106, 2178
TIMx_CCMR1 . . 1957, 1959, 2040, 2042, 2103-2104, ... 2175-2176
TIMx_CCMR2 . . . . . .1961-1962, 2044-2045
TIMx_CCMR3 1975
TIMx_CCR1 . . . . . .1968, 2049, 2108, 2182
TIMx_CCR2 . . . . . .1969, 2049
TIMx_CCR3 . . . . . .1969, 2050
TIMx_CCR4 . . . . . .1970, 2050
TIMx_CCR5 1976
TIMx_CCR6 1977
TIMx_CNT . 1967, 2047-2048, 2107, 2180, 2201
TIMx_CR1 . . . . . .1946, 2032, 2100, 2171, 2198
TIMx_CR2 . . . . . .1947, 2033, 2172, 2200
TIMx_DCR 1974, 2051, 2185
TIMx_DIER 1952, 2037, 2101, 2173, 2200
TIMx_DMAR 1975, 2052, 2185
TIMx_EGR 1956, 2039, 2102, 2175, 2201
TIMx_PSC 1967, 2048, 2107, 2181, 2202
TIMx_RCR 1968, 2181
TIMx_SMCR 1950, 2034
TIMx_SR 1954, 2038, 2102, 2174, 2201
TPIU_AUTHSTAT 3384
TPIU_CIDR0 3388
TPIU_CIDR1 3389
TPIU_CIDR2 3389
TPIU_CIDR3 3389
TPIU_CLAIMCLR 3383
TPIU_CLAIMSET 3382
TPIU_CURPSIZE 3375
TPIU_CURTPM 3378
TPIU_DEVID 3385
TPIU_DEVTYPE 3386
TPIU_FFCR 3381

TPIU_FFSR 3380
TPIU_FSCR 3382
TPIU_LAR 3383
TPIU_LSR 3384
TPIU_PIDR0 3386
TPIU_PIDR1 3387
TPIU_PIDR2 3387
TPIU_PIDR3 3388
TPIU_PIDR4 3386
TPIU_SUPPSIZE 3375
TPIU_SUPTPM 3378
TPIU_SUPTRGM 3376
TPIU_TPRCR 3380
TPIU_TRGCNT 3376
TPIU_TRGMULT 3377
TSG_CIDR0 3331
TSG_CIDR1 3331
TSG_CIDR2 3332
TSG_CIDR3 3332
TSG_CNTCR 3327
TSG_CNTCVL 3327
TSG_CNTCVU 3328
TSG_CNTFID0 3328
TSG_CNTSR 3327
TSG_PIDR0 3329
TSG_PIDR1 3329
TSG_PIDR2 3330
TSG_PIDR3 3330
TSG_PIDR4 3329
TZC_ACTION 313
TZC_BUILD_CONFIG 313
TZC_CID0 326
TZC_CID1 326
TZC_CID2 327
TZC_CID3 327
TZC_FAIL_ADDRESS_HIGH 317
TZC_FAIL_ADDRESS_LOW 316
TZC_FAIL_CONTROL 317
TZC_FAIL_ID 318
TZC_GATE_KEEPER 314
TZC_INT_CLEAR 316
TZC_INT_STATUS 315
TZC_PID0 324
TZC_PID1 325
TZC_PID2 325
TZC_PID3 325
TZC_PID4 323
TZC_PID5 323
TZC_PID6 323
TZC_PID7 324
TZC_REGION_ATTRIBUTE0 320
TZC_REGION_ATTRIBUTEx 322
TZC_REGION_BASE_HIGHx 319
TZC_REGION_BASE_LOW0 318
TZC_REGION_BASE_LOWx 321
TZC_REGION_ID_ACCESSx 321
TZC_REGION_TOP_HIGHx 320
TZC_REGION_TOP_LOW0 319
TZC_REGION_TOP_LOWx 322
TZC_SPECULATION_CTRL 315

U

UID 3523
USART_BRR 2495
USART_CR1 . . . . . 2480, 2484
USART_CR2 2487
USART_CR3 2491
USART_GTPR 2495
USART_HWCFGR1 2511
USART_HWCFGR2 2510
USART_ICR 2507
USART_IPIDR 2512
USART_ISR 2498, 2503
USART_PRESC 2510
USART_RDR 2509
USART_RQR 2497
USART_RTOR 2496
USART_SIDR 2513
USART_TDR 2509
USART_VERR 2512
USBH_EHCI_INSNREG01 2998
USBH_EHCI_INSNREG02 2999
USBH_EHCI_INSNREG03 3000
USBH_EHCI_INSNREG04 3001
USBH_EHCI_INSNREG05 3002
USBH_EHCI_INSNREG06 3003
USBH_EHCI_INSNREG07 3003
USBH_OHCI_INSNREG06 2997
USBH_OHCI_INSNREG07 2997
USBPHYC_MISC 2986
USBPHYC_MONITORx 2987
USBPHYC_PLL 2985
USBPHYC_TUNEx 2988
USBPHYC_VERR 2990

V

VREFBUF_CCR 1493
VREFBUF_CSR 1492

重要通知——请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和/或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对ST产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST不对任何知识产权进行任何明示或默示的授权或许可。

转售的ST产品如有不同于此处提供的信息的规定，将导致ST针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。若需 ST 商标的更多信息，请参考 www.st.com/trademarks。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。

© 2025 STMicroelectronics - 保留所有权利