STM32MP21xA/C/D/F device errata

Applicability

This document applies to the part numbers of STM32MP21xA/C/D/F devices and the device variants as stated in this page.

It gives a summary and a description of the device errata, with respect to the device datasheet and reference manual RM0506.

Deviation of the real device behavior from the intended device behavior is considered to be a device limitation. Deviation of the description in the reference manual or the datasheet from the intended device behavior is considered to be a documentation erratum. The term *"errata"* applies both to limitations and documentation errata.

Table 1. Device summary

Reference	Part numbers
STM32MP211x	STM32MP211A, STM32MP211C, STM32MP211D, STM32MP211F
STM32MP213x	STM32MP213A, STM32MP213C, STM32MP213D, STM32MP213F
STM32MP215x	STM32MP215A, STM32MP215C, STM32MP215D, STM32MP215F

Table 2. Device variants

Reference	Silicon revision codes	
	Device marking(1)	REV_ID(2)
STM32MP21	A	0x1000
	Z	0x1001

1. *Refer to the device datasheet for how to identify this code on different types of package.*
2. *REV_ID[15:0] bitfield of DBGMCU_IDCODE register.*

1 Summary of device errata

The following table gives a quick reference to the STM32MP21xA/C/D/F device limitations and their status:

A = limitation present, workaround available

N = limitation present, no workaround available

P = limitation present, partial workaround available

"-" = limitation absent

Applicability of a workaround may depend on specific conditions of target application. Adoption of a workaround may cause restrictions to target application. Workaround for a limitation is deemed partial if it only reduces the rate of occurrence and/or consequences of the limitation, or if it is fully effective for only a subset of instances on the device or in only a subset of operating modes, of the function concerned.

Table 3. Summary of device limitations

Function	Section	Limitation	Status	
			Rev. A	Rev. Z
Arm Cortex-A35	2.1.1	Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation	A	A
	2.1.2	Some AT instructions executed from EL3 might incorrectly report a domain fault	A	A
	2.1.3	ATB flush response may be delayed	A	A
	2.1.4	PMU counter might be inaccurate when monitoring BUS_ACCESS and BUS_ACCESS_ST	A	A
	2.1.5	Mismatch between EDPRSR.SR and EDPRSR.R	A	A
	2.1.6	ATS12NSOPR instruction might incorrectly translate when the HCR.TGE bit is set	A	A
Arm Cortex-M33	2.2.1	Access permission faults are prioritized over unaligned device memory faults	N	N
System	2.3.1	ADF1/MDF1 kernel clock not provided in autonomous mode	A	A
	2.3.2	The A35 is not able to access DBGMCU PID, CID registers	A	A
	2.3.3	STGEN is reset when D1 domain is in DStandby low power mode	N	-
	2.3.4	CPU2 (Cortex-M33) does not support debug in non-secure only	A	A
	2.3.5	ETHx kernel clock is gated if ETHxMACEN register bit is not set	A	A
	2.3.6	RISAF wrong SIDR value	N	N
	2.3.7	STOP and Standby entry failed when DDR is in shared mode	A	-
	2.3.8	AHB RISAB3/4/5 illegal access due to ghost CID0 detection	A	A
	2.3.9	ETM timestamp is exported with zero value	A	A
	2.3.10	Boot ROM writes in SYSRAM during LPLV-Stop2 wake-up	A	-
	2.3.11	Cortex-A35 not restarted after system reset in TDCID Cortex-M33 configuration	A	-
	2.3.12	Boot serial after System reset not OK if no power cycle on VBAT	A	-
	2.3.13	DEBUG AP0 port is not open after development boot	A	-
	2.3.14	Not possible to boot from SDMMC2 with 8x8 package	A	-
RCC	2.4.1	RCC semaphore restriction	A	A
FMC	2.5.1	NOR flash memory/PSRAM incorrect bus turnaround timing	A	A
	2.5.2	Incorrect FMC_CLK clock period when CLKDIV value is changed on-the-fly in Continuous clock mode	A	A

Function	Section	Limitation	Status	
			Rev. A	Rev. Z
OCTOSPI	2.6.1	Memory-mapped write error response when DQS output is disabled	P	P
	2.6.2	Memory wrap instruction not enabled when DQS is disabled	N	N
	2.6.3	Deadlock or write-data corruption after spurious write to a misaligned address in OCTOSPI_AR register	N	N
	2.6.4	Deadlock on consecutive out-of-range memory-mapped write operations	P	P
	2.6.5	Indirect write mode limited to 256 Mbytes	N	N
	2.6.6	Read-modify-write operation does not clear the MSEL bit	A	A
	2.6.8	Setting the ABORT bit does not generate an error on the AHB bus for undefined-length incremental burst transfers	P	P
	2.6.9	Read data corruption when a wrap transaction is followed by a linear read to the same MSB address	N	N
	2.6.10	Transactions are limited to 8 Mbytes in OctaRAM™ memories	N	N
	2.6.11	Variable latency is not supported when a refresh collision occurs during a write access to some OctaRAM™ memories	P	P
	2.6.12	In automatic status-polling and multiplexed modes, the controller does not request the port if less than two bytes are sent per cycle when OCTOSPI_DLR is cleared	A	A
SDMMC	2.7.1	Command response and receive data end bits not checked	N	N
ADC	2.8.1	JEOS may be set before the last injected data are available in ADC_JDRx	A	A
	2.8.2	In combined regular simultaneous plus alternate trigger mode, stopping injected conversion may delay regular conversion	A	A
	2.8.3	When the ADC clock is derived from the AHB clock, the injected conversion latency is not respected if the injected trigger coincides with the stopping of the regular conversion	A	A
LTDC	2.9.1	Layers cannot read YUV420 multibuffer data	N	N
TIM	2.10.1	Unexpected PWM output when using ocref_clr	N	N
LPTIM	2.11.1	Device may remain stuck in LPTIM interrupt when entering Stop mode	A	A
	2.11.2	ARRM and CMPM flags are not set when APB clock is slower than kernel clock	A	A
	2.11.3	Interrupt status flag is cleared by hardware upon writing its corresponding bit in LPTIM_DIER register	N	N
RTC	2.12.1	Alarm flag may be repeatedly set when the core is stopped in debug	N	N
I2C	2.13.1	Wrong data sampling when data setup time ($t_{SU;DAT}$) is shorter than one I2C kernel clock period	P	P
	2.13.2	Spurious bus error detection in controller mode	A	A
USART	2.14.1	Wrong data received by SPI slave receiver in autonomous mode with CPOL = 1	A	A
	2.14.2	Received data may be corrupted upon clearing the ABREN bit	A	A
	2.14.3	Noise error flag set while ONEBIT is set	N	N
LPUART	2.15.1	Possible LPUART transmitter issue when using low BRR[15:0] value	P	P
SPI	2.16.1	RDY output failure at high serial clock frequency	N	N
FDCAN	2.17.1	Desynchronization under specific condition with edge filtering enabled	A	A
	2.17.2	Tx FIFO messages inverted under specific buffer usage and priority setting	A	A
	2.17.3	DAR mode transmission failure due to lost arbitration	A	A

Function	Section	Limitation	Status	
			Rev. A	Rev. Z
OTG_HS	2.18.1	Potential unexpected transfer on the USB bus instead of a zero-length packet	A	A
	2.18.2	False detection of chirp-K when a FS device is connected	A	A
ETH	2.19.1	Incorrect gate control list switching for intermediate cycles when CTR is less than the GCL execution time	A	A

The following table gives a quick reference to the documentation errata.

Table 4. Summary of device documentation errata

Function	Section	Documentation erratum
OCTOSPI	2.6.7	Automatic status-polling mode cannot be used with HyperFlash™ memories

2 Description of device errata

The following sections describe the errata of the applicable devices with Arm® core and provide workarounds if available. They are grouped by device functions.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

2.1 Arm Cortex-A35

Reference manual and errata notice for the Arm® Cortex®-A35 core revision r1p0 is available from http://infocenter.arm.com.

2.1.1 Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation

This limitation is registered under Arm ID number 1608096 as Cat B, with significant impact to the silicon devices using this Arm core.

Description

A speculative *address translation* (AT) instruction translates using registers that are associated with an out-of-context translation regime and caches the resulting translation in the TLB. A subsequent translation request that is generated when the out-of-context translation regime is current uses the previous cached TLB entry producing an incorrect virtual to physical mapping.

The wrong speculative address translation may occur when the following conditions are all met:

1. A speculative AT instruction performs a table walk, translating a virtual address to a physical address using registers associated with an out-of-context translation regime.
2. Address translation data that is generated during the walk is cached in the TLB.
3. The out-of-context translation regime becomes current and a subsequent memory access is translated using previously cached address translation data in the TLB, resulting in an incorrect virtual to physical mapping.

When these conditions are met, the resulting translation is incorrect.

Workaround

When context-switching the register state for an out-of-context translation regime, system software at EL2 or above must ensure that all intermediate states during the context switch would report a level 0 translation fault in response to an AT instruction targeting the out-of-context translation regime.

A workaround is only required if the system software contains an AT instruction as part of an executable page.

2.1.2 Some AT instructions executed from EL3 might incorrectly report a domain fault

This limitation is registered under Arm ID number 799764 as Cat B, with significant impact to the silicon devices using this Arm core.

Description

Address translation instructions executed from EL3 and targeting EL1 or EL0 might report an incorrect result in the PAR when the HCR_EL2. DC bit is set.

The failure occurs when the following conditions are all met:

1. The core is executing at *exception* level 3 in AArch64.
2. The core executes one of the following address translation instructions:
   - AT S1E0R, AT S1E0W
   - AT S1E1R, AT S1E1W
   - AT S12E0R, AT S12E0W
   - AT S12E1R, AT S12E1W
3. The Exception level targeted by the address translation instruction is Non-secure and AArch32.
4. HCR_EL2.DC is set.

5. The DACR is programmed so that domain 0 would cause a domain fault if the HCR_EL2.DC bit had not been set. Note that this is the default value out of reset.

When these conditions are met, the PAR register incorrectly reports that a domain fault occurred.

Workaround

Secure software can set the DACR[1:0] to 0b01 before executing the address translation instruction. It should restore the previous DACR value before returning to a lower *exception* level.

2.1.3 ATB flush response may be delayed

This limitation is registered under Arm ID number 2252746 as Cat C, with minor impact to the silicon devices using this Arm core.

Description

The *embedded trace macrocell* (ETM) supports an external flush request for each ATB bus. Under certain timing conditions, an AFREADY response from the processor may be delayed until a new ATB transfer is generated by the processor.

The erratum occurs when the following conditions are met:

1. The ETM is enabled.
2. Trace data is generated on the ATB bus.
3. An external flush request is generated.
4. Trace data stops being generated due to filtering in the ETM or the ETM being disabled.

When these conditions are met, external trace infrastructure which is waiting for trace to be captured may stall forever (for example in a *flush and stop* scenario). All of the trace data are captured, but it is not possible to identify this by polling the *trace capture device*. If the flush is acknowledged, this can be treated as a reliable indication.

If the ETM is enabled again after the erratum has been triggered, the flush logic should become active again. If a flush is generated while trace is being generated, the only effect will be a delay in acknowledging the flush. This should not have any observable impact.

In a system with multiple trace sources, the delayed flush response may prevent other trace sources from accessing the ATB bus if they are generating trace while the flush is in progress. This could cause trace to be lost from these other sources.

Workaround

There is no workaround to reliably avoid this erratum.

As an alternative to waiting for the flush to be acknowledged, a sufficiently long timeout can be used if it is likely that trace generation has stopped. If ATB upsizers are present in the system, this workaround will not be effective.

2.1.4 PMU counter might be inaccurate when monitoring BUS_ACCESS and BUS_ACCESS_ST

This limitation is registered under Arm ID number 1010162 as Cat C, with minor impact to the silicon devices using this Arm core.

Description

The Cortex-A35 processor implements a *performance monitor unit* (PMU). The PMU allows programmers to gather statistics on the operation of the processor during runtime. Because of this erratum, the PMU counter values might be inaccurate when monitoring BUS_ACCESS and BUS_ACCESS_ST events.

This limitation occurs under the following conditions:

1. A performance counter is enabled and configured to count BUS_ACCESS or BUS_ACCESS_ST events.
2. A write or eviction occurs on the bus.

When these conditions are met, the PMU counter will erroneously increment or erroneously fail to increment. The inaccuracy varies between cores. Some bus accesses for core 0 might be attributed to core 1. The impact on the final counter value in each core is high.

This might lead to inaccurate results when using the PMU to debug or profile code.

Workaround

The BUS_ACCESS and BUS_ACCESS_ST events can be counted accurately for core 0 by enabling the counter on both core 0 and core 1 and taking the total. This workaround requires core 1 to be present in the configuration and idle during testing.

The event L2D_CACHE_WB counts write-backs from the L2 cache that are attributable to a core. For a test focused on cacheable memory bandwidth, this might be a suitable replacement for BUS_ACCESS_ST. This event includes:

- write-backs as a result of evictions and cache maintenance instructions
- writes in read allocate mode (write-streaming mode)

L2D_CACHE_WB does not include:

- write-backs as a result of snoop requests from outside of the cluster
- write-backs as a result of ACP interface accesses

2.1.5 Mismatch between EDPRSR.SR and EDPRSR.R

This limitation is registered under Arm ID number 857487 as Cat C, with minor impact to the silicon devices using this Arm core.

Description

The processor provides access to the EDPRSR through the APB interface. If this access is done at the same time as the core leaves a *warm* reset, then a subsequent read of the same register reads an incorrect value of the SR field.

The error occurs when the following conditions are met:

1. The core is in *warm* reset.
2. A debugger reads the EDPRSR register over the APB interface.
3. The core comes out of *warm* reset during the APB read.
4. A second APB read is made to the EDPRSR register. One or more exception types are routed to *monitor* mode by setting one or more of SCR.{EA, FIQ, IRQ} bits.

When these conditions are met then the exception mask bit CPSR.{A, F, I} is cleared for each exception type that meets the conditions 1. and 2.. The affected mask bits are cleared together regardless of the exception type in the condition 3..

The implications of this erratum are:

- The first read of the EDPRSR reads the SR field and R field both as 0b1.
- The second read of the EDPRSR.SR field reads 0b0 whereas the previous read of the EDPRSR.SR was 0b1 while in *warm* reset. Because the first read took place while in *warm* reset, the sticky bit must still be set on the second read.

Workaround

If the debugger reads the EDPRSR and sees both the SR and R fields set, then it must remember this result and on the next EDPRSR read treat the SR bit as if it was set.

2.1.6 ATS12NSOPR instruction might incorrectly translate when the HCR.TGE bit is set

This limitation is registered under Arm ID number 801757 as Cat C, with minor impact to the silicon devices using this Arm core.

Description

An ATS12NSOPR address translation instruction executed from EL3 might report an incorrect result in the PAR when both the HCR.TGE bit is set and the SCTLR.M is set.

The error occurs when the following conditions are met:

1. The core is executing at *exception* level 3 in AArch32.
2. SCR.NS = 0
3. HCR.TGE = 1
4. SCTLR(ns).M = 1
5. The core executes an ATS12NSOPR instruction.

If the above conditions are met, the PAR register incorrectly reports the translation as if the stage 1 MMU was enabled.

Note: The combination of both HCR.TGE and SCTLR.M bits being set was unpredictable in earlier versions of the architecture, and was only given a defined behavior to reduce the unpredictable space. It is not expected to be a useful combination for software.

Workaround

Secure software can clear the SCTLR(ns).M bit before executing the address translation instruction, if the HCR.TGE bit is set. It should restore the previous SCTLR(ns).M value before returning to a lower *exception* level.

2.2 Arm Cortex-M33

Reference manual and errata notice for the Arm® Cortex®-M33 core revision r0p4 is available from http://infocenter.arm.com.

2.2.1 Access permission faults are prioritized over unaligned device memory faults

Description

A load or store which causes an unaligned access to device memory results in an UNALIGNED UsageFault exception. However, if the region is not accessible because of the MPU access permissions (as specified in MPU_RBAR.AP), then the resulting MemManage fault is prioritized over the UsageFault.

The failure occurs when the MPU is enabled and:

- A load/store access occurs to an address which is not aligned to the data type specified in the instruction.
- The memory access hits one region only.
- The region attributes (specified in the MAIR register) mark the location as device memory.
- The region access permissions prevent the access (that is, unprivileged or write not allowed).

The MemManage fault caused by the access permission violation is prioritized over the UNALIGNED UsageFault exception because of the memory attributes.

Workaround

None. However, it is expected that no existing software is relying on this behavior since it was permitted in Armv7-M.

2.3 System

2.3.1 ADF1/MDF1 kernel clock not provided in autonomous mode

Description

When hse_ker_ck/msi_ker_ck clock is selected and HSEKERON/MSIKERON bit is not set in RCC_OCENSETR/RCC_D3DCR registers then ADF1/MDF1 kernel clock is not provided when the peripheral is working in autonomous mode and the bus clock request is active.

Workaround

Set HSEKERON/MSIKERON bit in RCC_OCENSETR/RCC_D3DCR registers.

2.3.2 The A35 is not able to access DBGMCU PID, CID registers

Description

The reading of DBGMCU registers (PID @0xFE0 and CID @0xFF0) by the Arm Cortex®-A35 generates errors (RRESP)

Other DBGMCU registers read/write work well.

Accessing these registers are through JTAG for debugger, so the Arm Cortex®-A35 access is not required.

Workaround

Access these registers through JTAG for debugger.

2.3.3 STGEN is reset when D1 domain is in DStandby low power mode

Description

The STGEN is reset by a cpu1_rstn signal. As a consequence, STGEN is reset whenever CPU1 is woken-up from any of the RUN2, STOP2, LP-STOP2, or LPLV-STOP2 modes.

Workaround

None.

2.3.4 CPU2 (Cortex-M33) does not support debug in non-secure only

Description

The Cortex-M33 access port AHB-AP (AP8) is not accessible when “microcontroller debug / non-secure / full debug” profile is selected (BSEC_DENR[15:0] = 0x0398). This mean the debug and trace features integrated in the Cortex-M33 are not accessible. Debug activity of the device can still be done using system interconnect AXI-AP (AP4).

Workaround

Use “microcontroller debug / secure / full debug” profile (BSEC_DENR[15:0] = 0x0F98).

2.3.5 ETHx kernel clock is gated if ETHxMACEN register bit is not set

Description

In registers RCC_ETHxCFGR, when setting only ETHxEN or ETHxLPEN bits, the kernel clock is not enabled. The ETHxMACEN bit controlling the gating of the bus clock must also be set to allow enabling the kernel clock (x = 1, 2 or SW for respectively ETH1, ETH2 and ETHSW).

Workaround

Each time the ck_ker_ethx kernel clock is required, the corresponding ETHxMACEN bit must be set together with regular kernel clock enable bit.

2.3.6 RISAF wrong SIDR value

Description

The value of RISAF_SIDR is 0xA3C5 DD01 whereas it should be 0xA3C5 DD04. 4 Kbytes are allocated to RISAF in the memory map.

Workaround

None.

2.3.7 STOP and Standby entry failed when DDR is in shared mode

Description

Once RCC_DDRITFCFGR.DDRSHR is set to 1, the system cannot enter in low power mode (Stop and Standby modes) if the DDRSS bus clocks are kept enabled.

Workaround

The DDRSS bus clocks must be disabled by software before entering low power mode. The bus clocks are disabled by clearing RCC_DDRCPCFGR.DDRCPEN, RCC_DDRPHYCAPBCFGR.DDRPHYCAPBEN and RCC_DDRCFGR.DDRCFGEN registers bits.

2.3.8 AHB RISAB3/4/5 illegal access due to ghost CID0 detection

Description

When an AHB busy signal is inserted during a transaction, a ghost CID0 is generated on the bus. If the compartment filtering is enabled on RISAB3/4/5, this transient CID0 is interpreted as a fault access by RISAB3/4/5 which aborts current access and returns an IAC.

Workaround

CID0 must be “reserved”: not assigned to any master at RISFC RIMU level.

- RIFSC_RIMC_ATTRx.MCID must not be programmed with CID0 value.

CID0 must be “reserved”: not assigned to any master at RIFSC RISUP level.

- RIFSC_RISC_PERy_CIDCFGR.CFEN must be set to 1 (compartment filtering enabled) to avoid use of CID0 default at RISUP level.
- RIFSC_RISC_PERy_CIDCFGR.SCID must not be programmed with CID0 value.
- RIFSC_RISC_PERy_CIDCFGR.SEMWLC0 must not be set to 1.

CID0 must be “reserved” for HPDMAx channel.

- HPDMA_CxCIDCFGR.CFEN must be set to 1 (compartment filtering enabled) to avoid use of CID0 default value.
- HPDMA_CxCIDCFGR.SCID must not be programmed with CID0 value.
- HPDMA_CxCIDCFGR.SEM_WLIST_CID0 must not be set to 1.

For RISAB3/4/5, CID0 read and write accesses must be enabled for all blocks/pages.

- RISAB3/4/5_CID0RDCFGR = 0xFFFFFFFF.
- RISAB3/4/5_CID0WRCFGR = 0xFFFFFFFF.

2.3.9 ETM timestamp is exported with zero value

Description

When attempting to use ETM traces, the timestamp value exported by default is zero. This is because STM clock is needed for ETM trace timestamps.

Workaround

Set RCC_STMCFGR.STMEN bit to 1.

2.3.10 Boot ROM writes in SYSRAM during LPLV-Stop2 wake-up

Description

During LPLV-Stop2 wake-up, the boot ROM writes dummy 32-bits word values at address 0x0E000080 and 0x0E000084, thus modifying SYSRAM content.

Workaround

The firmware in SYSRAM shall not use these addresses.

2.3.11 Cortex-A35 not restarted after system reset in TDCID Cortex-M33 configuration

Description

In TDCID = 0x2 (Cortex-M33) configuration, the first boot is successful (Cortex-M33 or Cortex-A35 flash boot), but if the TAMP_BKP11R has been set with a value corresponding to a SYSRAM address (intended for CPU1 CStandby exit), then in case of system reset, the Cortex-A35 does not restart. If the system reset is done with backup domain kept switched on during the power cycle, the system does not boot until the backup domain is switched off.

Workaround

The software must update the configuration after cold-boot (system reset applied), standby wakeup, Cortex-A35 reset initiated by Cortex-M33 or Watchdog reset with TAMP_BKP11R = 0x00000000.

2.3.12 Boot serial after System reset not OK if no power cycle on VBAT

Description

Serial boot is not working after a pad reset (nRST).

.

Workaround

- If a backup battery is used, the coin-cell battery should be removed from the connector.
- If VBAT is connected to VDD, a complete power cycle including VDD is required before each serial boot. All STPMIC2 regulators should be kept unmasked.

2.3.13 DEBUG AP0 port is not open after development boot

Description

The AP0 debug port is not open after development boot.

Workaround

Use an OpenOCD script to ensure a dummy write in DBGMCU_DBG_AUTH_DEV before using AP0 debug. First, enable the DBGMCU clock from CID1/Cortex-A.

2.3.14 Not possible to boot from SDMMC2 with 8x8 package

Description

It is not possible to boot from SDMMC2 with the 8x8 package.

Workaround

Need to boot from SDMMC1.

2.4 RCC

2.4.1 RCC semaphore restriction

Description

When the RCC_SEMxCR register is set up as a semaphore with two allowed CIDs, if one CID locks the semaphore, the other CID can unlock it without generating any IAC event. This applies to all resources except resource 103.

Workaround

Only use semaphore for system clock (only one CID in the allowlist) in the RCC.

2.5 FMC

2.5.1 NOR flash memory/PSRAM incorrect bus turnaround timing

Description

The delays between consecutive device accesses, programmed through the BUSTURN[3:0] bitfield of the FMC_BTRx and FMC_BWTRx registers, have no effect. Instead systematic delays are applied:

- t_{IDLE2}: 2 * fmc_ker_ck cycles between accesses to two NOR/PSRAM devices
- t_{IDLE1}: 1 * fmc_ker_ck cycle between accesses to a NOR/PSRAM and a NAND flash memory

Workaround

Extend the bus turnaround delays to satisfy bus turnaround constraints. Three cases need to be considered:

1. Two consecutive accesses to non-multiplexed NOR/PSRAM devices:
   Program t_{ADDSET} (NOR/PSRAM address setup phase) as needed (see Figure 1).
2. Access to a non-multiplexed NOR/PSRAM followed by an access either to a NOR/PSRAM device with multiplexed A/DQ signals or to a synchronous device:
   Decrease the FMC kernel clock frequency in order to meet the timing constraints (see Figure 2).
3. Access to a non-multiplexed NOR/PSRAM followed by an access to a NAND flash memory device
   Program t_{MEMSET} (NOR/PSRAM address setup phase) as needed (see Figure 3).

Figure 1. Bus turn timing recovery - asynchronous accesses to NOR/PSRAM devices (case 1)

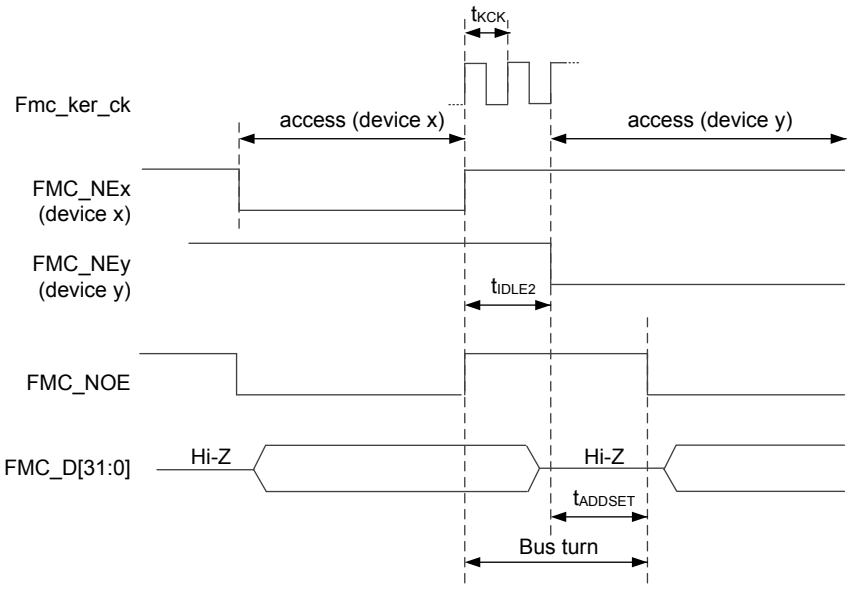

Figure 2. Bus turn timing recovery - access to NOR/PSRAM followed by access to multiplexed A/DQ synchronous device (case 2)

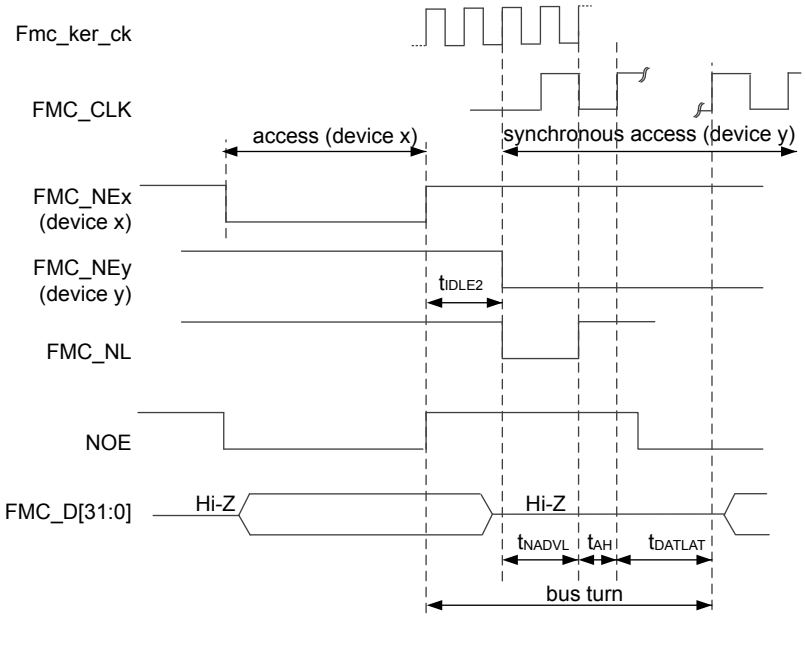

Figure 3. Bus turn timing recovery - access to NOR/PSRAM followed by access to NAND Flash device (case 3)

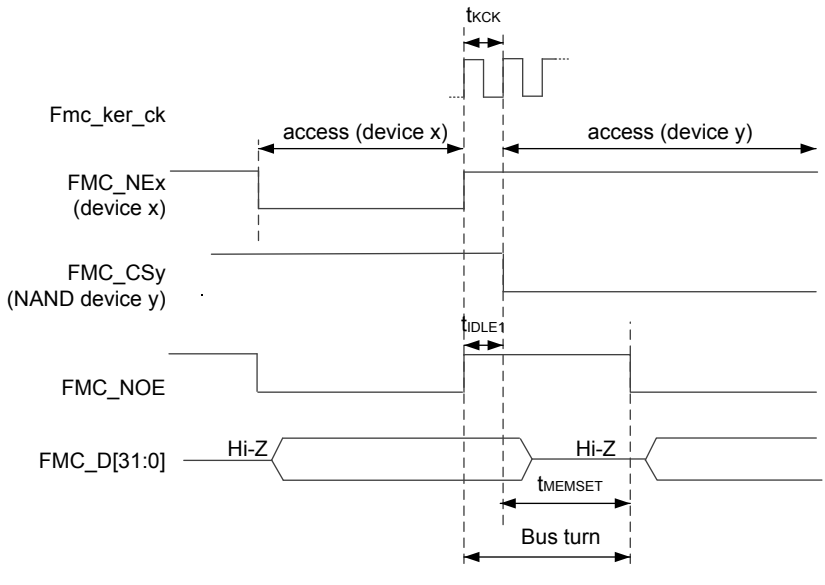

Table 5 gives the internal latencies that are not mentioned in the product datasheets. Refer to the datasheets for more details about the others timing values.

Table 5. Timing parameter description

Parameter	Description	Minimum value
t_{KCK}	FMC kernel clock period	See product datasheet

Parameter	Description	Minimum value
t_{IDLE1}	NEx high to CSy low (switching to a NAND flash memory device)	1 * t_{KCK}
t_{IDLE2}	NEx high to NEy low (NOR/PSRAM devices)	2 * t_{KCK}
t_{ADDSET}	NOR/PSRAM address setup phase	See Ref. Man.
t_{NADVL}	Address valid low pulse duration in synchronous mode	(CLKDIV+1) * t_{KCK}
t_{AH}	Address hold in synchronous mode	((CLKDIV+1) * t_{KCK}) / 2
t_{DATLAT}	NOR/PSRAM data latency in synchronous mode	See product datasheet
$t_{MEMxSET}$	NAND flash memory address setup phase	See product datasheet

2.5.2 Incorrect FMC_CLK clock period when CLKDIV value is changed on-the-fly in Continuous clock mode

Description

When the FMC operates in Continuous clock mode (CCLKEN is set in FMC_BCRx register), a new clock division factor is applied by changing the value of CLKDIV[3:0] in FMC_CFGR while the FMC is disabled (FMCEN cleared in FMC_BCRx), there is one FMC_CLK clock cycle during which the FMC_CLK period is not as expected: for example the clock low pulse duration matches the previous CLKDIV[3:0] value whereas the clock high pulse duration matches the new CLKDIV[3:0] value.

Workaround

Use the following sequence:

1. Stop the memory traffic for all devices.
2. Disable the FMC (refer to the disabling sequence described in the product reference manual).
3. Change CCLKEN for 1 to 0 in the FMC_BCRx register to stop the clock generation.
4. Program the desired CLKDIV[3:0] value in the FMC_CFGR register.
5. Change back CCLKEN from 0 to 1.
6. Enable the FMC.

2.6 OCTOSPI

2.6.1 Memory-mapped write error response when DQS output is disabled

Description

If the DQSE control bit of the OCTOSPI_WCCR register is cleared for memories without DQS pin, it results in an error response for every memory-mapped write request.

Workaround

When doing memory-mapped writes, set the DQSE bit of the OCTOSPI_WCCR register, even for memories that have no DQS pin.

2.6.2 Memory wrap instruction not enabled when DQS is disabled

Description

Memory wrap instruction (as configured in the OCTOSPI_WPxxx registers) is not generated when DQS is disabled. The memory wrap instruction is replaced by two regular successive read instructions to ensure the correct data ordering: this split has very limited impact on performance.

Workaround

None.

2.6.3 Deadlock or write-data corruption after spurious write to a misaligned address in OCTOSPI_AR register

Description

Upon writing a misaligned address to OCTOSPI_AR just before switching to memory-mapped mode (without first triggering the indirect write operation), with the OCTOSPI configured as follows:

- FMODE = 00 in OCTOSPI_CR (indirect write mode)
- DQSE = 1 in OCTOSPI_CCR (DQS active)

then, the OCTOSPI may be deadlocked on the first memory-mapped request or the first memory-mapped write to memory (and any sequential writes after it) may be corrupted.

An address is misaligned if the address is:

- Odd and the OCTOSPI is configured to send two bytes of data to the memory every cycle (octal-DTR mode or dual-quad-DTR mode), or
- Not a multiple of four when the OCTOSPI is configured to send four bytes of data to the memory (16-bit DTR mode or dual-octal DTR mode).

If the OCTOSPI_AR register is reprogrammed with an aligned address (without triggering the indirect write between the two writes to OCTOSPI register), the data sent to the memory during the indirect write operation are also corrupted.

Workaround

None.

2.6.4 Deadlock on consecutive out-of-range memory-mapped write operations

Description

The DEVSIZE[4:0] bitfield of the OCTOSPI_DCR1 register indicates that the size of the memory is 2 ^ [DEVSIZE + 1] bytes, and thus any memory-mapped access to address 2 ^ [DEVSIZE + 1] or above should get an error response.

However, no error response may be returned and the OCTOSPI may become deadlocked after the following sequence of events:

1. A memory-mapped write operation is ongoing on the AHB bus.
2. A second memory-mapped write is requested to an address close to the end of the memory but not consecutive to the address targeted by the first write operation.
3. A third memory-mapped write operation is requested, this time to an address consecutive to the address targeted by the second write, and the address of this third write is 2 ^ [DEVSIZE + 1] or an address consecutive to 2 ^ [DEVSIZE + 1].
   If the first write command has not completed writing data, then the write to 2 ^ [DEVSIZE + 1] does not return any error response and the next memory-mapped request gets stalled indefinitely.

Workaround

Ensure that no sequences of consecutive memory-mapped write operations pass the memory boundary.

2.6.5 Indirect write mode limited to 256 Mbytes

Description

In indirect write mode, if the address is greater than 256 Mbytes, the indirect write is not performed at the targeted address, even if it is located inside the allowed memory space configured through the device size (DEVSIZE[4:0] of OCTOSPI_DCR1). Actually, this write operation takes place within the 256-Mbyte memory space, thus corrupting the memory content.

Indirect read operations are not impacted.

Workaround

Indirect write operations have to be performed inside the first 256 Mbytes of the memory space.

2.6.6 Read-modify-write operation does not clear the MSEL bit

Description

When the MSEL bit of the OCTOSPI_CR register is set, it remains set even if the software attempts to clear it by performing a read-modify-write operation.

Workaround

To clear the MSEL bit, clear in a single write access bit 7 and bit 30 of the OCTOSPI_CR register, otherwise, the MSEL bit remains set.

2.6.7 Automatic status-polling mode cannot be used with HyperFlash™ memories

Description

Some reference manuals mention that the automatic status-polling mode can be used with the HyperBus™ protocol. This is not possible since HyperFlash™ memories require two steps to read the status register (a write operation followed by a read command), while the automatic status-polling mode, already implemented in the regular-command protocol, requires a single read instruction to read back the status register.

This is a documentation issue rather than a product limitation.

Workaround

None.

2.6.8 Setting the ABORT bit does not generate an error on the AHB bus for undefined-length incremental burst transfers

Description

An AHB error is expected to be generated when the ABORT bit of the OCTOSPI_CR register is set while a request is ongoing.

Instead, the controller does not trigger any AHB error if the ongoing request is an undefined-length incremental burst AHB transfer.

An AHB error is generated for all other transfer types.

Workaround

When possible, wait for the end of the transfer before setting the ABORT bit.

2.6.9 Read data corruption when a wrap transaction is followed by a linear read to the same MSB address

Description

If a wrap transaction is followed by a linear read having the same MSB start address as the wrap (), then the linear read is wrongly considered as a sequential transaction to the previous one, taking back the prefetched data and causing data corruption.

Notice that for a wrap transaction, the prefetch starts after the last address of the wrap window.

Workaround

As prefetch cannot be disabled, there is no workaround. However, the issue is seldom encountered since wrap operations are mostly initiated by the internal cache to refresh its cacheline. All the other masters must avoid retrieving data by using a linear read access to the same MSB address as the wrap, which has been just completed.

2.6.10 Transactions are limited to 8 Mbytes in OctaRAM™ memories

Description

When the controller is configured in Macronix OctaRAM™ mode, by setting the MTYP[2:0] bitfield of the OCTOSPI_DCR1 register to 011, only 13 bits of row address are decoded and sent to the memory, meaning that only 8 K of 1-Kbyte blocks can be accessed (8 Mbytes).

Workaround

None.

This limitation is not present for PSRAMs or HyperRAM™ memories.

2.6.11 Variable latency is not supported when a refresh collision occurs during a write access to some OctaRAM™ memories

Description

When the memory type (MTYP[2:0] bitfield of the OCTOSPI_CR register) is configured to 0b011 to target an OctaRAM™ memory, the host controller does not support the variable latency requested by the external memory if a refresh collision occurs during the write access. For example, some OctaRAM™ memories, such as ISSI memories, request extra latency cycles for write accesses during refresh collision. In this case, the controller does not sample the DQS input signal during the instruction phase, and cannot detect the extra latency requested by the external memory for the refresh operation.This results in data corruption.

Some OctaRAM™ memories do not request any additional latency for write access during refresh cycles. It is required only when the refresh occurs during a read access. In this case, no issue can be observed.

Workaround

When the application targets an OctaRAM™ memory that requests extra latency cycles for write access during refresh collision, force the fixed latency mode in the configuration register of the external memory. There is no constraint about read access, since both variable and fixed latency modes are supported.

2.6.12 In automatic status-polling and multiplexed modes, the controller does not request the port if less than two bytes are sent per cycle when OCTOSPI_DLR is cleared

Description

Due to FIFO RX pointer mismatches, the controller configured in automatic status-polling mode (FMODE[1:0] = 10) may not be able to request the port ownership to serve the status-polling request to the external memory. As a result, the FIFO is wrongly detected full, thus blocking the request from the controller to the I/O manager.

The issue happens in the following conditions:

- The I/O manager is used in multiplexed mode (to connect two controllers sharing the same port).
- The I/O manager is connected to a PHY.
- The controller is configured in automatic status-polling mode.
- Less than two bytes are sent per CLK cycle.
- Data length register (OCTOSPI_DLR) is cleared.

Workaround

Set the OCTOSPI_DLR to configure the number of bytes to 2 (OCTOSPI_DLR set to 0x0000 0001), and configure the OCTOSPI polling status mask register (OCTOSPI_PSMKR) to mask the second dummy byte

2.7 SDMMC

2.7.1 Command response and receive data end bits not checked

Description

The command response and receive data end bits are not checked by the SDMMC. A reception with only a wrong end bit value is not detected. This does not cause a communication failure since the received command response or data is correct.

Workaround

None.

2.8 ADC

2.8.1 JEOS may be set before the last injected data are available in ADC_JDRx

Description

When the ADC peripheral clock (adc_hclk) is slower than the ADC kernel clock (adc_ker_ck), the JEOS flag of the ADC_ISR register may be set before the last data of the injected sequence are available in the ADC_JDRx register.

Workaround

Apply one of the following measures:

- Ensure adc_hclk is faster than adc_ker_ck.
- Select the discontinuous conversion mode for regular channels by setting the DISCEN bit of the ADC_CFGR1 register.
- Use oversampling for injected conversions.
- Count the number of injected data conversions, by monitoring the JEOC flag of the ADC_ISR register.

2.8.2 In combined regular simultaneous plus alternate trigger mode, stopping injected conversion may delay regular conversion

Description

In dual ADC combined regular simultaneous plus alternate trigger mode, the resumption of an active regular conversion may be delayed by a few ADC clock cycles when an injected trigger event coincides with stopping, by application, an ongoing injected conversion.

Note: The dual ADC combined regular simultaneous plus alternate trigger mode is selected by setting the DUAL[4:0] bitfield of the ADCC_CCR register to 0x02. To stop an ongoing injected conversion, the software sets the JADSTP bit of the ADC_CR register.

Workaround

- Design the application so as to avoid injected trigger events from coinciding with stopping, by software or DMA, an ongoing regular conversion.
- Stop the regular conversion before stopping the injected conversion.

2.8.3 When the ADC clock is derived from the AHB clock, the injected conversion latency is not respected if the injected trigger coincides with the stopping of the regular conversion

Description

When the ADC clock is derived from the AHB clock, and the analog-to-digital conversion is triggered by a timer, the latency between the trigger and the start of the ADC sampling is fixed. When both injected and regular conversions are enabled, if the injected trigger coincides with the stopping of regular conversion, the latency of injected conversions becomes one clock cycle shorter than the expected latency.

Note: To stop an ongoing regular conversion, the software sets the ADSTP bit of the ADC_CR register.

Workaround

Apply one of the following measures:

- Avoid triggering injected conversions when the ADSTP bit is set.
- When an injected trigger is expected, keep the ADSTP bit cleared.

2.9 LTDC

2.9.1 Layers cannot read YUV420 multibuffer data

Description

The YUV semiplanar and full-planar modes of the layers are not functional.

The YUV coplanar and the RGB modes can be used.

Workaround

None.

2.10 TIM

2.10.1 Unexpected PWM output when using ocref_clr

Description

In combined PWM mode 1, asymmetric PWM mode 1, or asymmetric PWM mode 2, using ocref_clr can cause the tim_ocxrefc output to be unexpectedly re-enabled or disabled. This behavior depends on the timing of when ocref_clr is activated and deactivated.

Workaround

None.

2.11 LPTIM

2.11.1 Device may remain stuck in LPTIM interrupt when entering Stop mode

Description

This limitation occurs when disabling the low-power timer (LPTIM).

When the user application clears the ENABLE bit in the LPTIM_CR register within a small time window around one LPTIM interrupt occurrence, then the LPTIM interrupt signal used to wake up the device from Stop mode may be frozen in active state. Consequently, when trying to enter Stop mode, this limitation prevents the device from entering low-power mode and the firmware remains stuck in the LPTIM interrupt routine.

This limitation applies to all Stop modes and to all instances of the LPTIM. Note that the occurrence of this issue is very low.

Workaround

In order to disable a low power timer (LPTIMx) peripheral, do not clear its ENABLE bit in its respective LPTIM_CR register. Instead, reset the whole LPTIMx peripheral via the RCC controller by setting and resetting its respective LPTIMxRST bit in the relevant RCC register.

2.11.2 ARRM and CMPM flags are not set when APB clock is slower than kernel clock

Description

When LPTIM is configured in one shot mode and APB clock is lower than kernel clock, there is a chance that ARRM and CMPM flags are not set at the end of the counting cycle defined by the repetition value REP[7:0]. This issue can only occur when the repetition counter is configured with an odd repetition value.

Workaround

To avoid this issue, the following formula must be respected:

{ARR, CMP} ≥ KER_CLK / (2* APB_CLK),

where APB_CLK is the LPTIM APB clock frequency, and KER_CLK is the LPTIM kernel clock frequency. ARR and CMP are expressed in decimal value.

Example: The following example illustrates a configuration where the issue can occur:

- APB clock source (MSI) = 1 MHz, kernel clock source (HSI) = 16 MHz
- The repetition counter is set with REP[7:0] = 0x3 (odd value)

The above example is subject to issues, unless the user respects:

{CMP, ARR} ≥ 16 MHz / (2 * 1 MHz)

→ ARR must be ≥ 8 and CMP must be ≥ 8

Note: REP set to 0x3 means that effective repetition is REP+1 (= 4) but the user must consider the parity of the value loaded in the LPTIM_RCR register (=3, odd) to assess the risk of issue.

2.11.3 Interrupt status flag is cleared by hardware upon writing its corresponding bit in LPTIM_DIER register

Description

When any interrupt bit of the LPTIM_DIER register is modified, the corresponding flag of the LPTIM_ISR register is cleared by hardware.

Workaround

None.

2.12 RTC

2.12.1 Alarm flag may be repeatedly set when the core is stopped in debug

Description

When the core is stopped in debug mode, the clock is supplied to subsecond RTC alarm downcounter even when the device is configured to stop the RTC in debug.

As a consequence, when the subsecond counter is used for alarm condition (the MASKSS[3:0] bitfield of the RTC_ALRMASSR and/or RTC_ALRMBSSR register set to a non-zero value) and the alarm condition is met just before entering a breakpoint or printf, the ALRAF and/or ALRBF flag of the RTC_SR register is repeatedly set by hardware during the breakpoint or printf, which makes any attempt to clear the flag(s) ineffective.

Workaround

None.

2.13 I2C

2.13.1 Wrong data sampling when data setup time ($t_{SU;DAT}$) is shorter than one I2C kernel clock period

Description

The I^2C-bus specification and user manual specify a minimum data setup time ($t_{SU;DAT}$) as:

- 250 ns in Standard mode
- 100 ns in Fast mode
- 50 ns in Fast mode Plus

The device does not correctly sample the I^2C-bus SDA line when $t_{SU;DAT}$ is smaller than one I2C kernel clock (I^2C-bus peripheral clock) period: the previous SDA value is sampled instead of the current one. This can result in a wrong receipt of target address, data byte, or acknowledge bit.

Workaround

Increase the I2C kernel clock frequency to get I2C kernel clock period within the transmitter minimum data setup time. Alternatively, increase transmitter's minimum data setup time. If the transmitter setup time minimum value corresponds to the minimum value provided in the I^2C-bus standard, the minimum I2CCLK frequencies are as follows:

- In Standard mode, if the transmitter minimum setup time is 250 ns, the I2CCLK frequency must be at least 4 MHz.
- In Fast mode, if the transmitter minimum setup time is 100 ns, the I2CCLK frequency must be at least 10 MHz.
- In Fast-mode Plus, if the transmitter minimum setup time is 50 ns, the I2CCLK frequency must be at least 20 MHz.

2.13.2 Spurious bus error detection in controller mode

Description

In controller mode, a bus error can be detected spuriously, with the consequence of setting the BERR flag of the I2C_SR register and generating bus error interrupt if such interrupt is enabled. Detection of bus error has no effect on the I^2C-bus transfer in controller mode and any such transfer continues normally.

Workaround

If a bus error interrupt is generated in controller mode, the BERR flag must be cleared by software. No other action is required and the ongoing transfer can be handled normally.

2.14 USART

2.14.1 Wrong data received by SPI slave receiver in autonomous mode with CPOL = 1

Description

The SPI slave receiver device receives wrong data when all the following conditions are met:

- The USART is used in SPI master transmitter mode
- The autonomous mode is used
- The CPOL bit of the USART_CR2 register is set

Workaround

When the autonomous mode is used, do not set the CPOL bit in USART_CR2.

2.14.2 Received data may be corrupted upon clearing the ABREN bit

Description

The USART receiver may miss data or receive corrupted data when the auto baud rate feature is disabled by software (ABREN bit cleared in the USART_CR2 register) after an auto baud rate detection, while a reception is ongoing.

Workaround

Do not clear the ABREN bit.

2.14.3 Noise error flag set while ONEBIT is set

Description

When the ONEBIT bit is set in the USART_CR3 register (one sample bit method is used), the noise error (NE) flag must remain cleared. Instead, this flag is set upon noise detection on the START bit.

Workaround

None.

Note: Having noise on the START bit is contradictory with the fact that the one sample bit method is used in a noise free environment.

2.15 LPUART

2.15.1 Possible LPUART transmitter issue when using low BRR[15:0] value

Description

The LPUART transmitter bit length sequence is not reset between consecutive bytes, which could result in a jitter that cannot be handled by the receiver device. As a result, depending on the receiver device bit sampling sequence, a desynchronization between the LPUART transmitter and the receiver device may occur resulting in data corruption on the receiver side.

This happens when the ratio between the LPUART kernel clock and the baud rate programmed in the LPUART_BRR register (BRR[15:0]) is not an integer, and is in the three to four range. A typical example is when the 32.768 kHz clock is used as kernel clock and the baud rate is equal to 9600 baud, resulting in a ratio of 3.41.

Workaround

Apply one of the following measures:

- On the transmitter side, increase the ratio between the LPUART kernel clock and the baud rate. To do so:
  - Increase the LPUART kernel clock frequency, or
  - Decrease the baud rate.
- On the receiver side, generate the baud rate by using a higher frequency and applying oversampling techniques if supported.

2.16 SPI

2.16.1 RDY output failure at high serial clock frequency

Description

When acting as slave with RDY alternate function enabled through setting the RDIOM bit of the SPI_CFG2 register, the device may fail to indicate its *Not ready* status in time through the RDY output signal to suspend communication. This may then lead to data overrun and/or underrun on the device side. The failure occurs when the serial clock frequency exceeds:

- Twice the APB clock frequency, with data sizes from 8 to 15 bits
- Six times the APB clock frequency, with data sizes from 16 to 23 bits
- Fourteen times the APB clock frequency, with data sizes from 24 to 32 bits

Workaround

None.

2.17 FDCAN

2.17.1 Desynchronization under specific condition with edge filtering enabled

Description

FDCAN may desynchronize and incorrectly receive the first bit of the frame if:

- the edge filtering is enabled (the EFBI bit of the FDCAN_CCCR register is set), and
- the end of the integration phase coincides with a falling edge detected on the FDCAN_Rx input pin

If this occurs, the CRC detects that the first bit of the received frame is incorrect, flags the received frame as faulty and responds with an error frame.

Note: This issue does not affect the reception of standard frames.

Workaround

Disable edge filtering or wait for frame retransmission.

2.17.2 Tx FIFO messages inverted under specific buffer usage and priority setting

Description

Two consecutive messages from the Tx FIFO may be inverted in the transmit sequence if:

- FDCAN uses both a dedicated Tx buffer and a Tx FIFO (the TFQM bit of the FDCAN_TXBC register is cleared), and
- the messages contained in the Tx buffer have a higher internal CAN priority than the messages in the Tx FIFO.

Workaround

Apply one of the following measures:

- Ensure that only one Tx FIFO element is pending for transmission at any time:
  The Tx FIFO elements may be filled at any time with messages to be transmitted, but their transmission requests are handled separately. Each time a Tx FIFO transmission has completed and the Tx FIFO gets empty (TFE bit of FDACN_IR set to 1) the next Tx FIFO element is requested.
- Use only a Tx FIFO:
  Send both messages from a Tx FIFO, including the message with the higher priority. This message has to wait until the preceding messages in the Tx FIFO have been sent.
- Use two dedicated Tx buffers (for example, use Tx buffer 4 and 5 instead of the Tx FIFO). The following pseudo-code replaces the function in charge of filling the Tx FIFO:

```
Write message to Tx Buffer 4
Transmit Loop:
    Request Tx Buffer 4 - write AR4 bit in FDCAN_TXBAR
    Write message to Tx Buffer 5
    Wait until transmission of Tx Buffer 4 complete (IR bit in FDCAN_IR),
    read TO4 bit in FDCAN_TXBTO
    Request Tx Buffer 5 - write AR5 bit of FDCAN_TXBAR
    Write message to Tx Buffer 4
    Wait until transmission of Tx Buffer 5 complete (IR bit in FDCAN_IR),
    read TO5 bit in FDCAN_TXBTO
```

2.17.3 DAR mode transmission failure due to lost arbitration

Description

In DAR mode, the transmission may fail due to lost arbitration at the first two identifier bits.

Workaround

Upon failure, clear the corresponding Tx buffer transmission request bit TRPx of the FDCAN_TXBRP register and set the corresponding cancellation finished bit CFx of the FDCAN_TXBCF register, then restart the transmission.

2.18 OTG_HS

2.18.1 Potential unexpected transfer on the USB bus instead of a zero-length packet

Description

In both buffer DMA and slave modes, when a zero-length packet must be transmitted on the USB bus, an incorrect data packet can be sent on the USB bus.

Workaround

In buffer DMA mode:

1. Program DIEPCTLx.EPENA=1, DIEPCTLx.SNAK=1, DIEPCTLx.CNAK=0.
2. Wait for 15 AHB clock cycles.
3. Program DIEPCTLx.CNAK=1, DIEPCTLx.SNAK=0.

The above steps must be performed for all IN transfers that involve a zero-length packet transmission.

In target mode:

1. Program DIEPCTLx.SNAK=1, DIEPCTLx.CNAK=0.
2. Wait for 20 AHB clock cycles (this value is determined by considering the fastest AHB clock and slowest phy_clk combination).
3. Program DIEPCTLx.EPENA=1.
4. Wait for 15 AHB clock cycles (fixed safe delay).
5. Program DIEPCTLx.CNAK=1 and DIEPCTLx.SNAK=0.

The above steps must be performed for all IN transfers that involve a zero-length packet transmission in device mode (DIEPTSIZx.XFRSIZ=0 and DIEPTSIZx.PKTCNT=1).

2.18.2 False detection of chirp-K when a FS device is connected

Description

When the host controller is programmed in high-speed mode and the port reset (OTG_HPRT.PRST) is programmed within 150 PHY clock cycles after a full-speed device connection is detected, the controller can detect a false chirp-K. This causes the controller to switch to high-speed operation even though no chirp is received from the connected device.

Workaround

1. Program the OTG_GINTMSK.PRTIM bit to unmask.
2. Configure the OTG_HCFG register to select either the full-speed host or high-speed host.
3. Set the OTG_HCFG.PPWR bit to 1.
4. Wait for the OTG_HPRT.PCDET interrupt, which indicates that a device is connected to the port.
5. Wait for 150 PHY clock cycles.
6. Set the OTG_HPRT.PRST bit to 1 to start the reset process.

2.19 ETH

2.19.1 Incorrect gate control list switching for intermediate cycles when CTR is less than the GCL execution time

Description

The EST (enhancements to scheduled traffic) scheduler switches to the next gate control list (GCL) after executing the current GCL, regardless of the difference between the start time of the next GCL and the time when the current GCL rows are completely executed.

If the GCL execution takes longer than the cycle time, the GCL is truncated at the cycle time, and the subsequent loop begins at BTR + N × cycle time, where N is an integer that represents the iteration number.

However, the GCL incorrectly updates the internal BTR twice, and skips the execution of the next GCL loop. This issue arises if one of the following conditions is met:

- CTR value < sum of time intervals of fully executed GCL rows, or
- CTR value > sum of time intervals of fully executed GCL rows, and
  CTR value < sum of time intervals of fully executed GCL rows + 8 PTP clock periods

Note: The CTR is the value initially configured by the software.

Workaround

Apply one of the following measures:

- CTR value > sum of time intervals of fully executed GCL rows + 8 PTP clock periods
- CTR = sum of time intervals of fully executed GCL rows

Important security notice

The STMicroelectronics group of companies (ST) places a high value on product security, which is why the ST product(s) identified in this documentation may be certified by various security certification bodies and/or may implement our own security measures as set forth herein. However, no level of security certification and/or built-in security measures can guarantee that ST products are resistant to all forms of attacks. As such, it is the responsibility of each of ST's customers to determine if the level of security provided in an ST product meets the customer needs both in relation to the ST product alone, as well as when combined with other components and/or software for the customer end product or application. In particular, take note that:

- ST products may have been certified by one or more security certification bodies, such as Platform Security Architecture (www.psacertified.org) and/or Security Evaluation standard for IoT Platforms (www.trustcb.com). For details concerning whether the ST product(s) referenced herein have received security certification along with the level and current status of such certification, either visit the relevant certification standards website or go to the relevant product page on www.st.com for the most up to date information. As the status and/or level of security certification for an ST product can change from time to time, customers should re-check security certification status/level as needed. If an ST product is not shown to be certified under a particular security standard, customers should not assume it is certified.
- Certification bodies have the right to evaluate, grant and revoke security certification in relation to ST products. These certification bodies are therefore independently responsible for granting or revoking security certification for an ST product, and ST does not take any responsibility for mistakes, evaluations, assessments, testing, or other activity carried out by the certification body with respect to any ST product.
- Industry-based cryptographic algorithms (such as AES, DES, or MD5) and other open standard technologies which may be used in conjunction with an ST product are based on standards which were not developed by ST. ST does not take responsibility for any flaws in such cryptographic algorithms or open technologies or for any methods which have been or may be developed to bypass, decrypt or crack such algorithms or technologies.
- While robust security testing may be done, no level of certification can absolutely guarantee protections against all attacks, including, for example, against advanced attacks which have not been tested for, against new or unidentified forms of attack, or against any form of attack when using an ST product outside of its specification or intended use, or in conjunction with other components or software which are used by customer to create their end product or application. ST is not responsible for resistance against such attacks. As such, regardless of the incorporated security features and/or any information or support that may be provided by ST, each customer is solely responsible for determining if the level of attacks tested for meets their needs, both in relation to the ST product alone and when incorporated into a customer end product or application.
- All security features of ST products (inclusive of any hardware, software, documentation, and the like), including but not limited to any enhanced security features added by ST, are provided on an "AS IS" BASIS. AS SUCH, TO THE EXTENT PERMITTED BY APPLICABLE LAW, ST DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, unless the applicable written and signed contract terms specifically provide otherwise.

Revision history

Table 6. **Document revision history**

Date	Version	Changes
22-Aug-2025	1	Initial release.

Contents