

引言

随着对具有严格实时需求的移动通信和汽车系统等应用的需求不断增长，有必要在完成闪存擦除/编程操作之前访问有关系统功能的关键信息。

其他应用程序需要固件升级，但这可能存在风险，尤其是在更新过程中发生系统断电时。这可能导致许多问题，例如传输错误或信息丢失。

出于这些原因，ST提供嵌入双存储区闪存的STM32 MCU，旨在满足上述需求。

双存储区闪存允许在一个存储区正在被擦除或编程过程中，在另一个存储区执行代码。这可以避免在编程操作期间CPU失速，保护系统免受电源故障或其他错误的影响。

本应用笔记概述了STM32F7系列闪存双存储区功能，例如同时读写（RWW）和双启动功能。

本应用笔记随附X-CUBE-DBANK-F7嵌入式软件包，该软件包包含三个示例以及运行这些示例所需的所有嵌入式软件模块。

这些示例描述了闪存双存储区模式的主要特性：

- **同时读写示例：**通过示波器波形说明同时读写功能如何允许在写入闪存存储区2的同时从闪存存储区1执行代码，且不会停止执行。
- **双启动示例：**在闪存存储区1中启动并闪烁LED1，或在闪存存储区2中启动并闪烁LED2，实现双启动功能。
- **性能和功耗示例：**运行CMSIS ARM®图形均衡器算法，测量STM32F7系列器件的性能和功耗，对比双存储区模式与单存储区模式。

相关文档

意法半导体网站 www.st.com 提供以下文档：

- STM32F76xxx和STM32F77xxx基于32位MCU（RM0410）的高级ARM®
- STM32微控制器系统存储器启动模式（AN2606）。

目录

1	闪存单存储区和双存储区配置	6
1.1	1 MB闪存组织结构	6
1.2	2 MB闪存组织结构	7
1.3	如何激活双存储区模式	8
2	同时读写(RWW)	9
2.1	闪存单存储区与双存储区的RWW	10
3	从单存储区模式切换至双存储区模式（或相反）	11
3.1	存储器数据组织结构	11
3.2	切换示例	12
4	双启动	13
4.1	双启动流程图	14
4.2	闪存存储区交换：	16
4.2.1	1 MB闪存存储区交换	16
4.2.2	2 MB闪存存储区交换	17
4.3	使用CRC的安全固件升级：	18
4.4	写保护：	18
4.5	软件设置提示：	18
5	示例	19
5.1	同时读写 (RWW) 示例	19
5.1.1	说明	19
5.1.2	单存储区配置	20
5.1.3	双存储区配置	21
5.1.4	EEPROM仿真	21
5.2	双启动示例	21
5.2.1	说明	21
5.3	性能和功耗示例	23
5.3.1	硬件要求	23
5.3.2	测量结果	23

6	结论	26
7	版本历史	27

表格索引

表1. RWW操作可能性 9

表2. 性能指标 24

表3. 功耗指标 24

表4. 文档版本历史 27

表5. 中文文档版本历史 27



图片索引

图1.	1 MB闪存组织结构	6
图2.	2 MB闪存组织结构	7
图3.	如何激活双存储区模式	8
图4.	同时读写图	9
图5.	闪存单存储区与双存储区的RWW	10
图6.	从单存储区模式切换至双存储区模式	11
图7.	单存储区数据组织结构示例	12
图8.	双存储区数据组织结构示例	12
图9.	如何激活双启动模式	13
图10.	双启动流程图	15
图11.	1 MB闪存存储区交换	16
图12.	2 MB闪存存储区交换	17
图13.	RWW 示例	20
图14.	单存储区配置中的RWW	20
图15.	双存储区配置中的RWW	21
图16.	从存储区2启动	22
图17.	从存储区1启动	22
图18.	硬件连接	23
图19.	性能显示	24

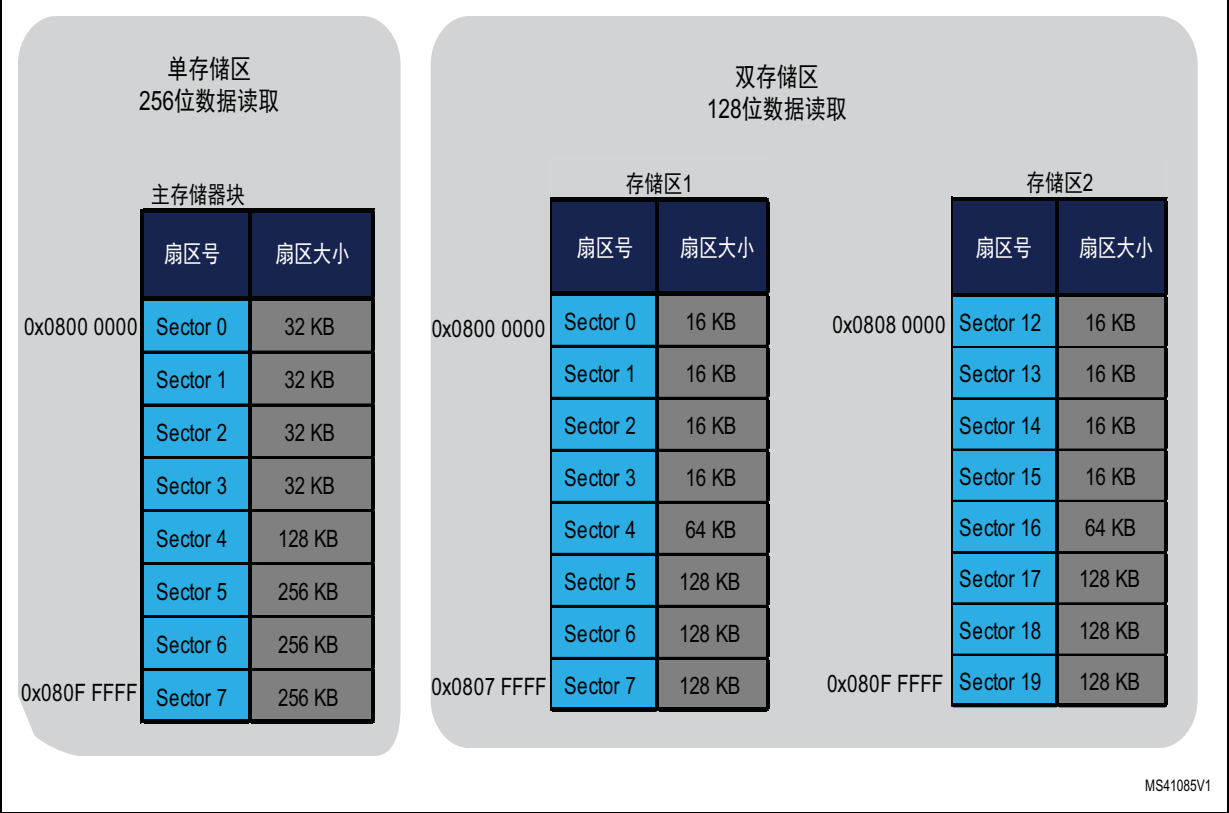
1 闪存单存储区和双存储区配置

STM32F7系列器件的闪存大小分别为1 MB和2 MB。
该闪存可以配置为单存储区或双存储区。

1.1 1 MB闪存组织结构

- 图 1介绍了适用于两种配置的1 MB闪存主块组织：单存储区和双存储区。
- 单存储区配置：主存储器块分为4个32 KB扇区、1个128 KB扇区和3个256 KB扇区。
 - 双存储区配置：在每个512KB存储区中，主存储器块分为4个16KB扇区、1个64KB扇区和3个128 KB扇区。

图1. 1 MB闪存组织结构



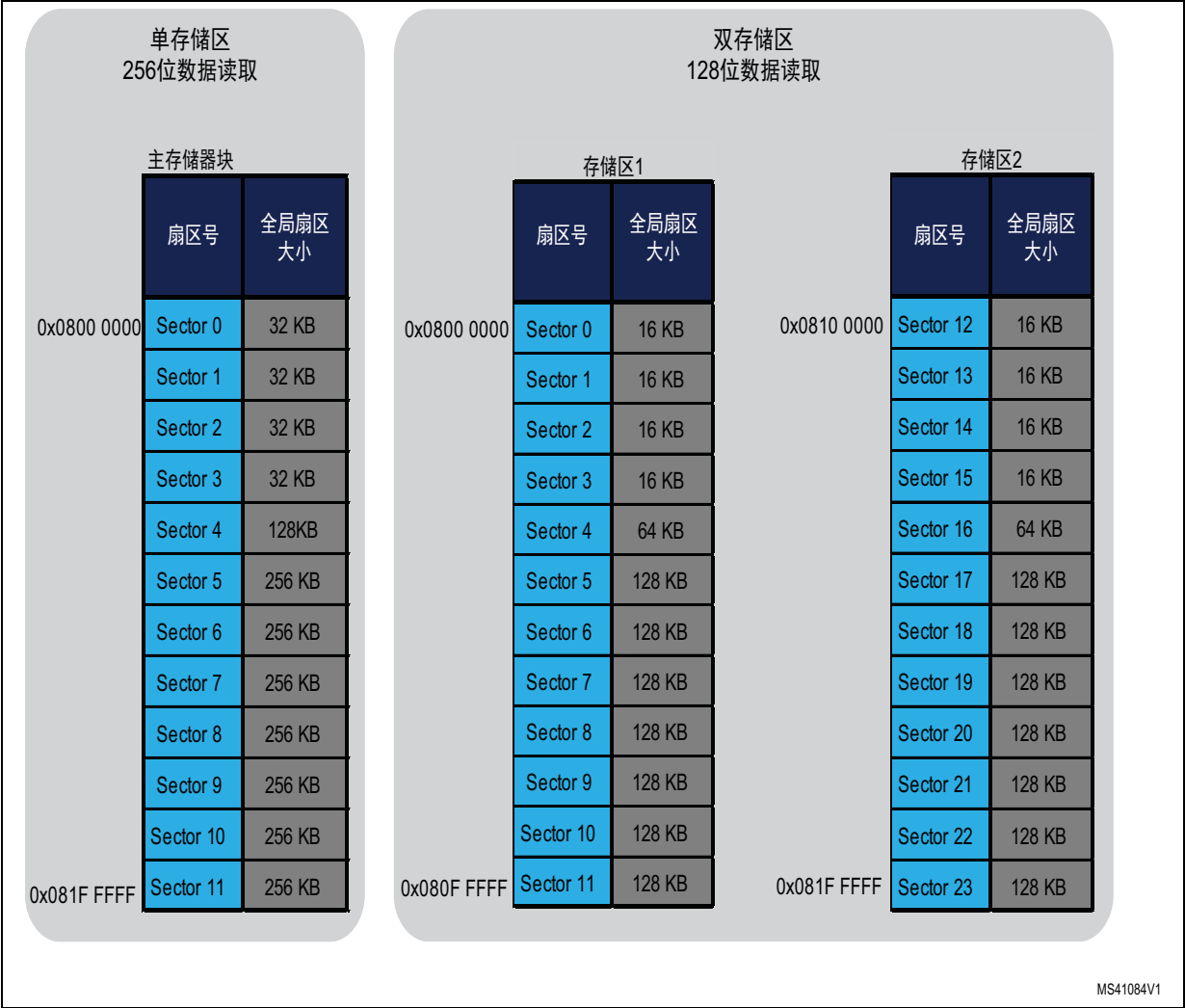
注：双存储区存储器组织的扇区编号与单存储区存储器组织不同：
单存储区存储器包含8个连续扇区号，而双存储区存储器包含16个扇区，扇区号不连续。

1.2 2 MB闪存组织结构

图 2介绍了适用于两种配置的2 MB闪存主块组织：单存储区和双存储区。

- 单存储区配置：
主存储器块分为4个32 KB扇区、1个128 KB扇区和7个256 KB扇区。
- 双存储区配置：
在每个1 MB存储区中，主存储器块分为4个16 KB扇区、1个64 KB扇区和7个128 KB扇区。

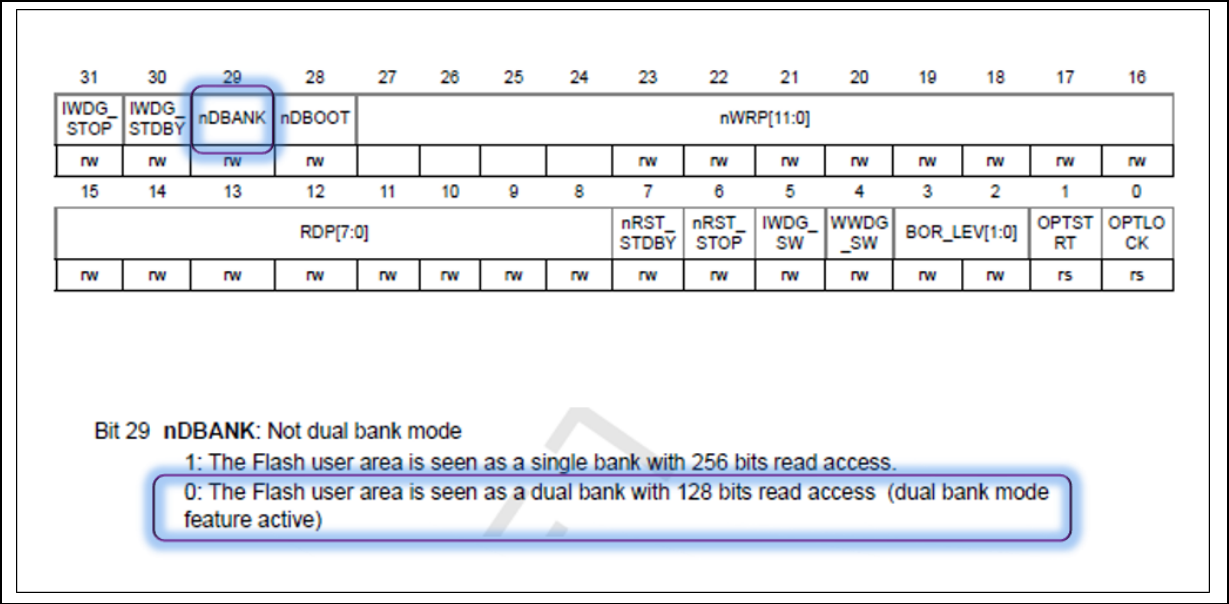
图2. 2 MB闪存组织结构



1.3 如何激活双存储区模式

使用FLASH_OPTCR寄存器，在用户选项字节中设置nDBANK=0，激活双存储区闪存模式。

图3. 如何激活双存储区模式



2同时读写(RWW)

双存储区闪存具有同时读写能力，可在继续操作的同时对系统进行编程。

但是，在擦除或编程另一个存储区时，不能在一个存储区上执行擦除或编程操作（同时擦除两个存储区的批量擦除除外）。图 4介绍了上述情况，其中，编程/擦除FSM的输出必须是程序或擦除操作。

图4. 同时读写图

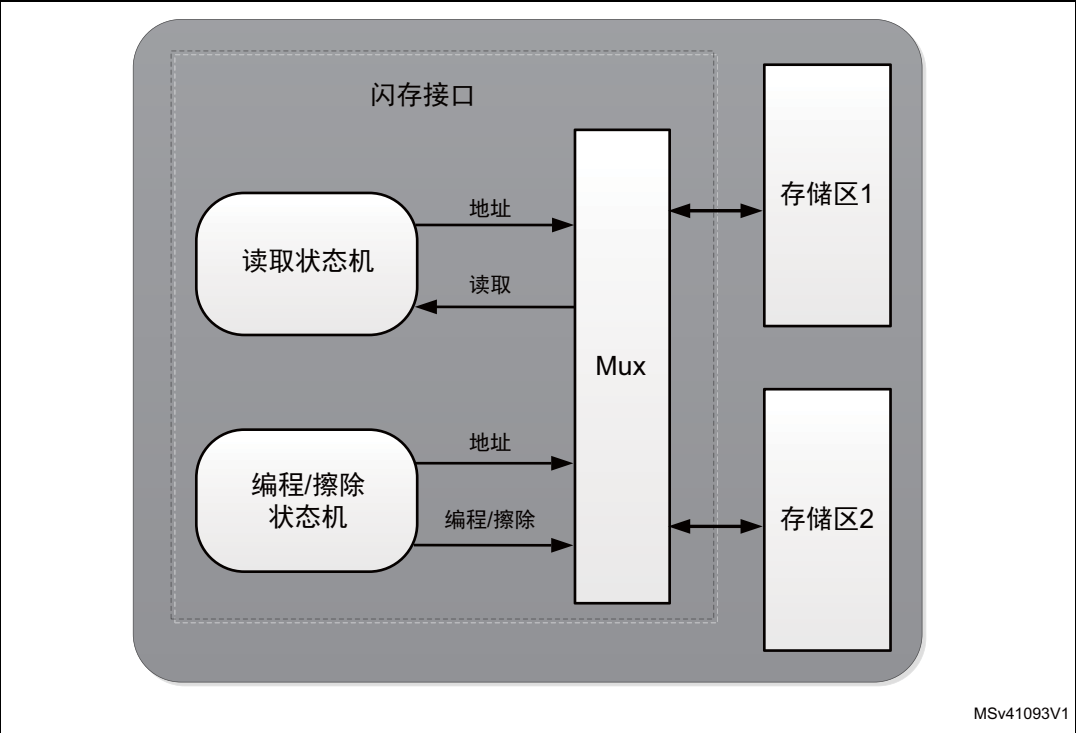


表 1总结了同时读写操作的可能性：

表 1的图例如下：

- A =允许，
- NA和灰色=不允许。

表1. RWW操作可能性

		存储区域2		
		读取	编程（Prog）	擦除
存储区域1	读取	NA	A	A
	编程（Prog）	A	NA	NA
	擦除	A	NA	A

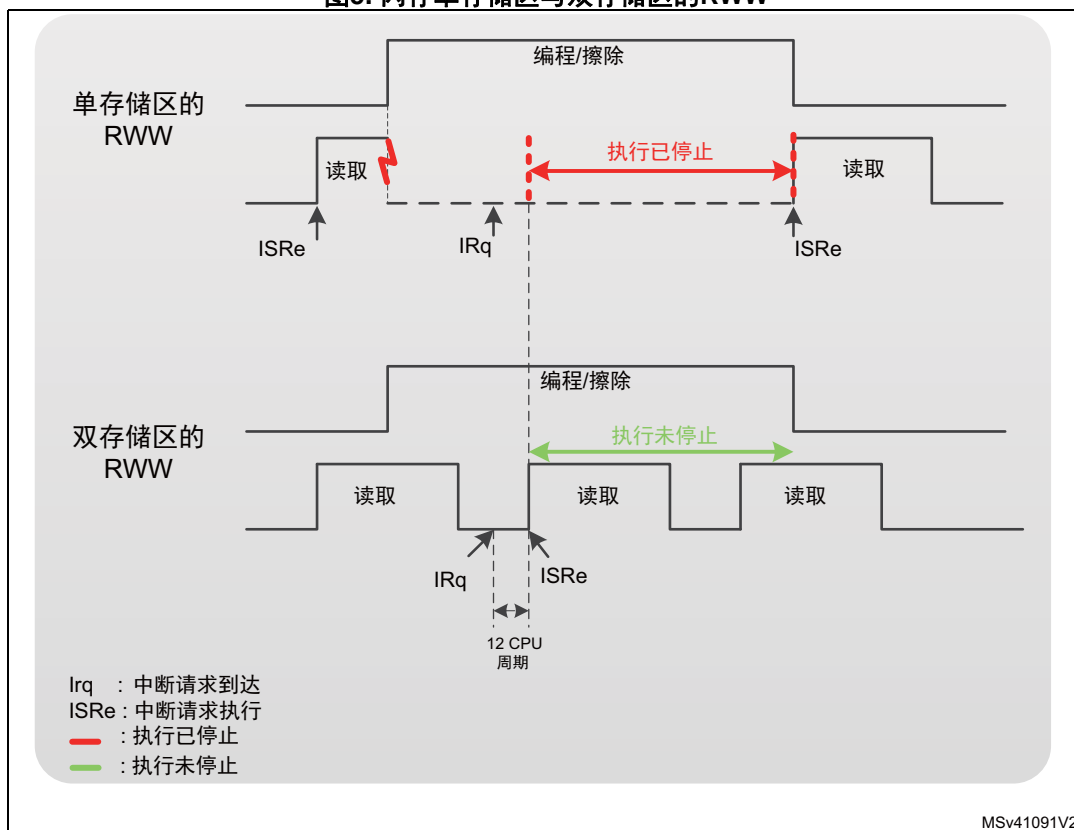
2.1 闪存单存储区与双存储区的RWW

图 5描述了CPU执行如何在闪存单存储区中失效，而在闪存双存储区中继续。

在单存储区模式中，当程序/擦除操作正在运行，读取中断请求到达时，CPU不能执行此中断，直到编程/擦除操作完成。

相反，在双存储区模式中，即使在编程/擦除操作期间（例如在存储区1中），CPU也可以同时执行来自存储区2的中断而不会失效。

图5. 闪存单存储区与双存储区的RWW

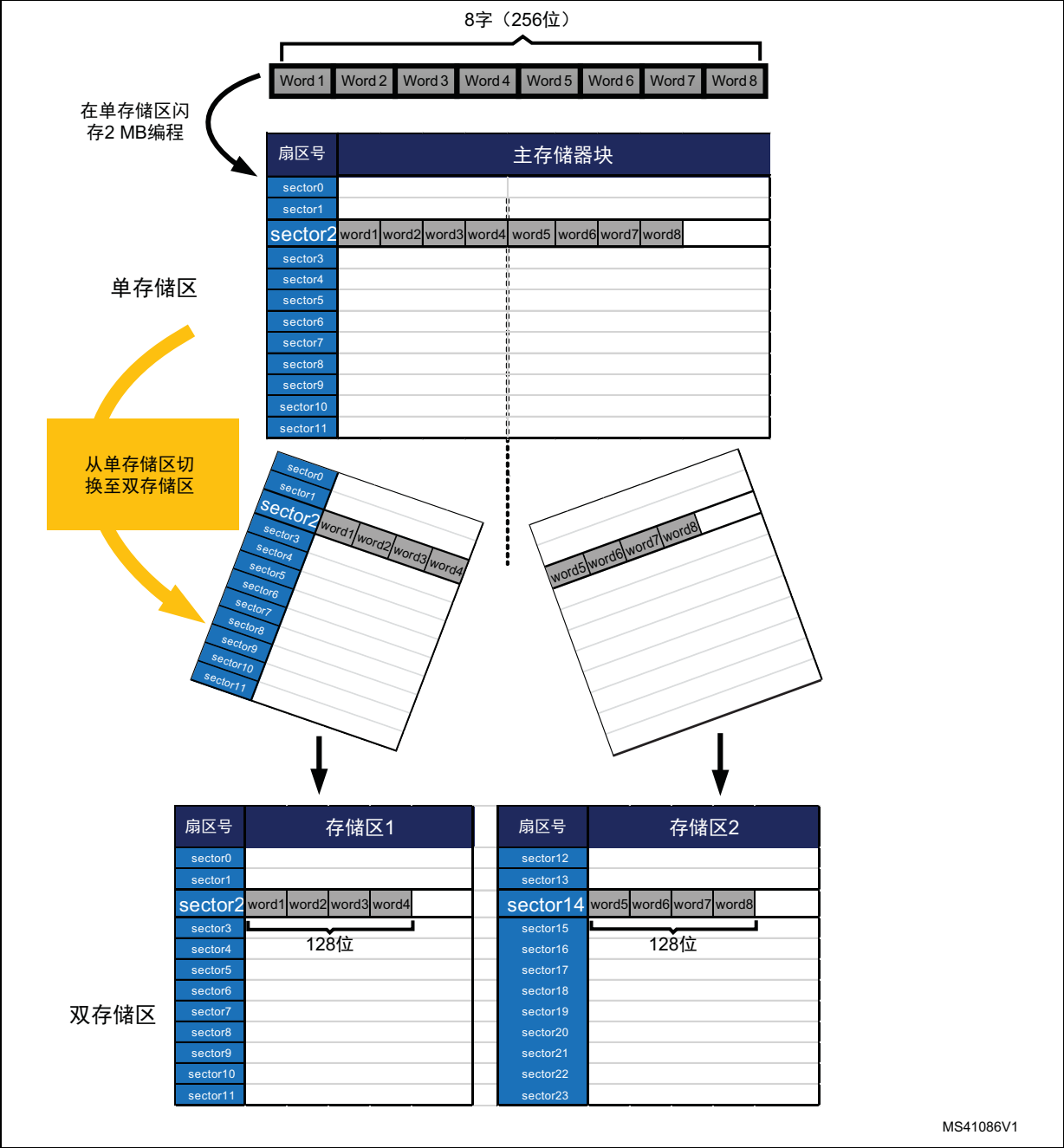


3 从单存储区模式切换至双存储区模式（或相反）

3.1 存储器数据组织结构

图 6描述了从单个存储区切换到双存储区模式时的内存数据组织结构：

图6. 从单存储区模式切换至双存储区模式

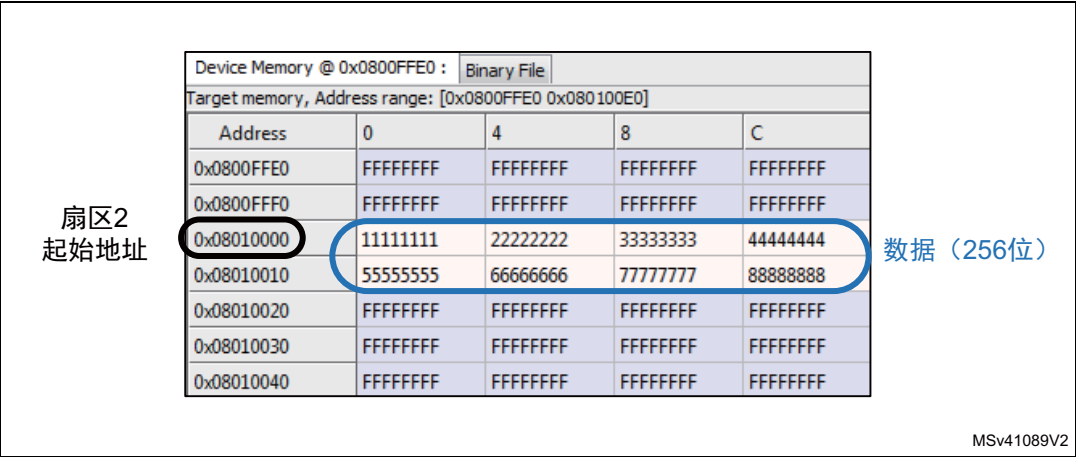


3.2 切换示例

图 7和图 8显示了从单存储区切换到双存储区模式并使用SLINK-Utility工具读取闪存内容的示例。

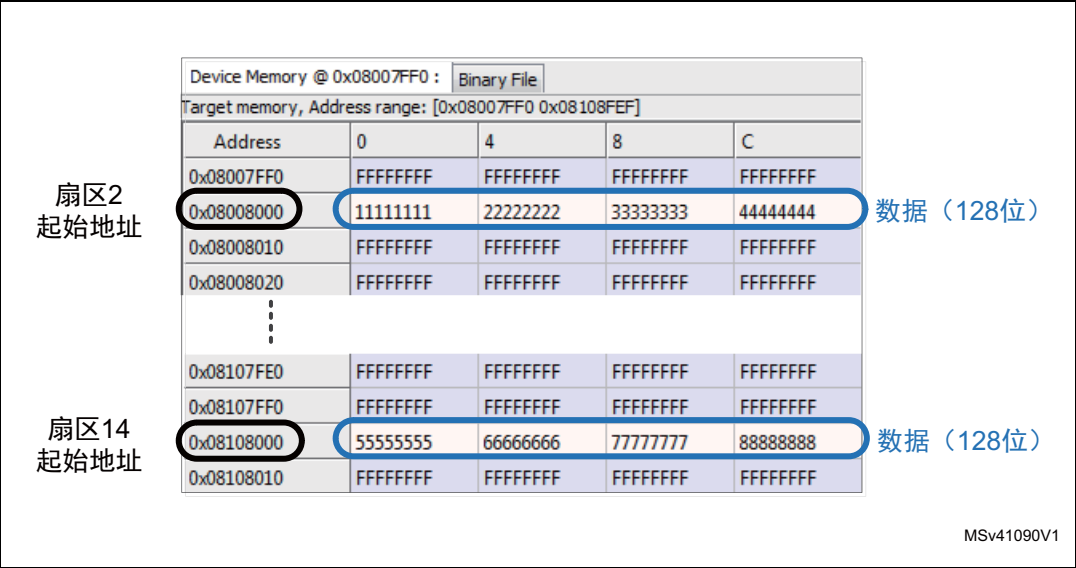
首先，用户在单存储区模式（nDBANK=1）中，对2 MB闪存扇区2（0x08010000）中的256位常量进行编程。

图7. 单存储区数据组织结构示例



接下来，用户切换到双存储区模式（nDBANK=0），获取闪存中的编程数据，结构如下：在存储区1和存储区2之间分配256位，前128位位于扇区2（0x0800 8000），后128位位于扇区1（0x0810 8000）。

图8. 双存储区数据组织结构示例



4 双启动

当STM32F7系列器件处于双存储区模式（nDBANK = 0）时，可以从存储区1或存储区2启动应用软件。

使用FLASH_OPTCR寄存器，在用户选项字节中设置nDBOOT = 0，激活双启动闪存模式。

图 9描述了如何在寄存器层面激活双启动模式。

图9. 如何激活双启动模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IWDG_STOP	IWDG_STDBY	nDBANK	nDBOOT	nWRP[11:0]											
rw	rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDP[7:0]								nRST_STOP	nRST_STOP	IWDG_SW	WWDG_SW	BOR_LEV[1:0]	OPTSTRT	OPTLOCK	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rs	rs

Bit 28 **nDBOOT**: Dual Boot mode (valid only when nDBANK=0)

Dual Boot mode (valid only when nDBANK=0)

1: Dual Boot disabled. Boot according to boot address option (Default)

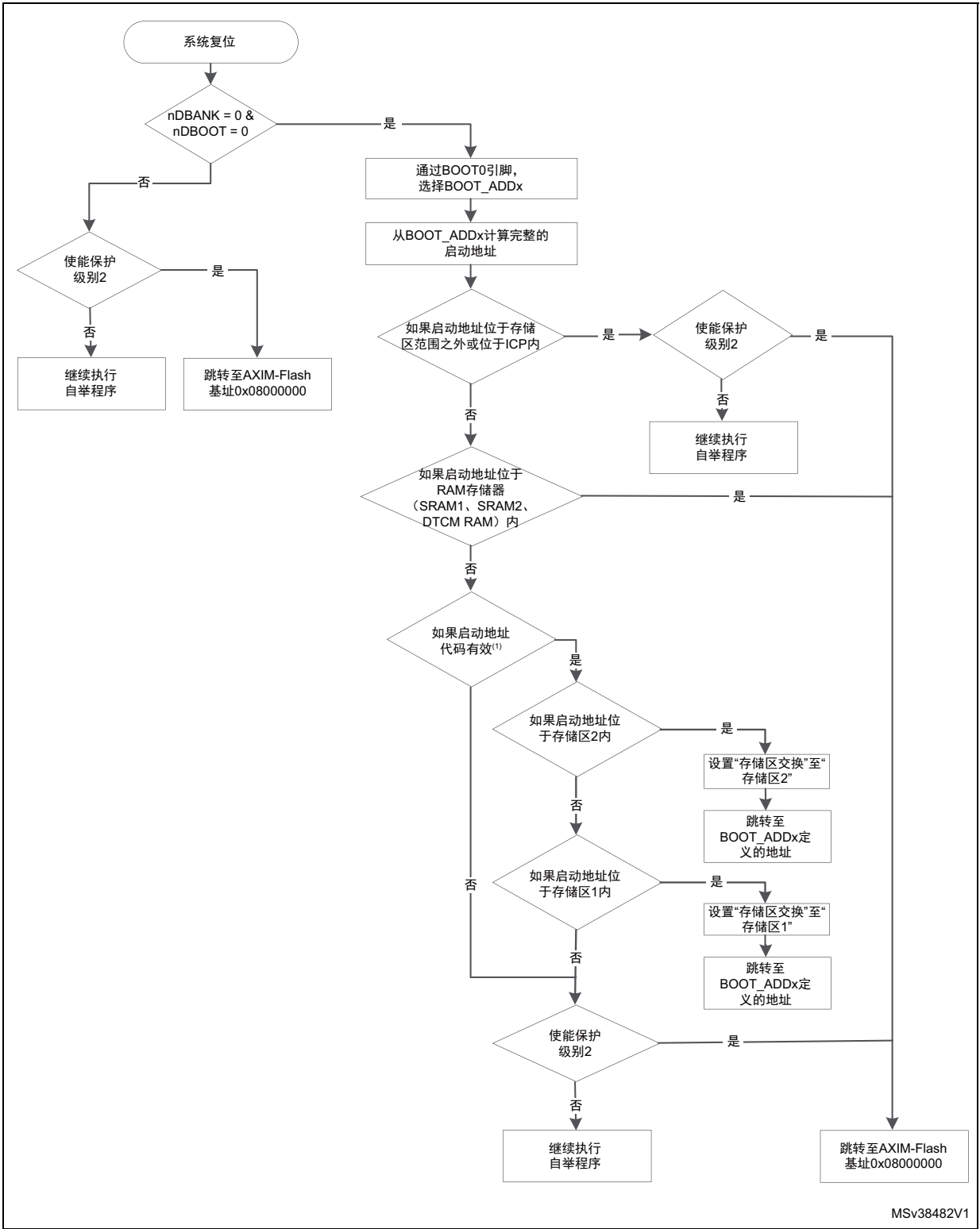
0: Dual Boot enabled. Boot always from system memory if boot address is in flash (Dual bank Boot mode), or RAM if Boot address option in RAM

4.1 双启动流程图

当STM32F7系列器件处于双存储区模式（nDBANK = 0），nDBOOT = 0，且BOOT引脚选择闪存存储区1或存储区2范围内的地址时，器件从系统存储器启动，自举程序跳转执行在Flash内存存储区1或存储区2中编程的用户应用程序。

[图 5](#)中的流程图解释了如何实现双启动。

图10. 双启动流程图



MSv38482V1

详情请参见应用笔记（AN2606）。



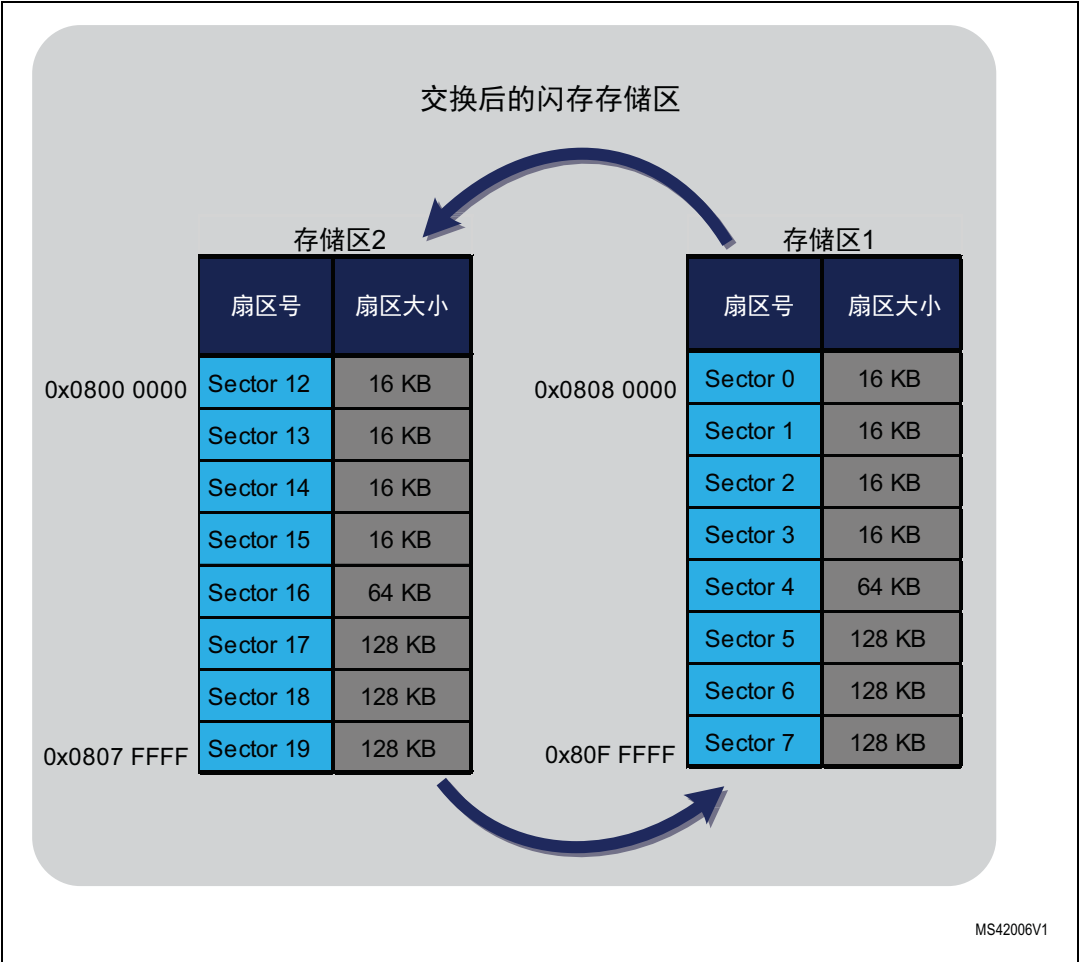
4.2 闪存存储区交换：

一旦激活双启动模式且启动地址有效，自举程序便会设置闪存存储区交换位（SWP_FB = 1），生成以下映射。

4.2.1 1 MB闪存存储区交换

- 闪存存储区2基址映射在0x08000000（AXI）
- 闪存存储区1基址映射在0x08000000（AXI）

图11. 1 MB闪存存储区交换



4.2.2 2 MB闪存存储区交换

- 闪存存储区2基址映射在0x08000000 (AXI)
- 闪存存储区1基址映射在0x08100000 (AXI)

图12. 2 MB闪存存储区交换



4.3 使用CRC的安全固件升级：

部分系统需要升级固件。使用自举程序即可完成。但是，在固件升级期间可能会出现一些问题，例如升级过程中发生断电。因此，新固件未正确安装，导致系统行为出现问题。

CRC是一种致力于实施安全性的技术，用于验证数据传输或存储完整性。它们可用于验证闪存的完整性。

CRC计算单元帮助计算待发送数据的签名。当从存储区1执行原始固件时，该签名与原始数据一起被发送到闪存存储区2（举例而言）。

当目标接收数据时，计算新签名，并将其与接收的签名进行比较。如果两者相等，则自举程序可以继续安全地升级进程，然后交换到存储区2，以执行新固件。

4.4 写保护：

- **单存储区配置：**

可以使用以下方案保护用户扇区（扇区0到扇区11）：

nWRP[i]位是扇区（i）的写保护位。

当其中一个扇区的非写保护有效时，不能擦除或编程扇区。因此，不能执行批量擦除。

- **双存储区配置：**

可以用以下方案保护存储区1（扇区0到扇区11）和存储区2（扇区12到扇区23）的用户扇区：

nWRP[i]位是扇区（2 * i）和扇区（2 * i + 1）的写保护位。

当其中一个扇区对的非写保护有效时，不能擦除或编程此扇区对。因此，不能执行批量擦除或存储区擦除。

4.5 软件设置提示：

用户必须牢记有关闪存单存储区或闪存双存储区编程的一些提示。

例如：关于使用20 KB代码的闪存编程：

- **第一种配置：在单存储区模式下配置项目，且nDBANK = 0**

由于在单存储区模式下配置项目，用户必须擦除1个扇区（32 KB），但在nDBANK = 0时，闪存接口配置为双存储区模式，因此1个扇区= 16 KB。这样一来，仅擦除了16 Kb，遗漏了4 KB。

因此，无法执行20 KB的编程，出现编程错误。

- **第二种配置：在双存储区模式下配置项目，且nDBANK = 1**

由于在双存储区模式下配置项目，用户必须擦除2个扇区（2x16KB），但在nDBANK=1时，闪存接口配置为单存储区模式，因此1个扇区= 32 KB。这样一来，擦除了2个扇区（2 x 32 Kb），但实际只需要20 KB。因此，擦除了额外的扇区（2 x 16 KB）。

编程闪存存储器时，用户必须注意项目配置和闪存选项字节是否匹配。

5 示例

本节介绍X-CUBE-DBANK-F7嵌入式软件示例，重点介绍闪存双存储区模式功能。

5.1 同时读写 (RWW) 示例

5.1.1 说明

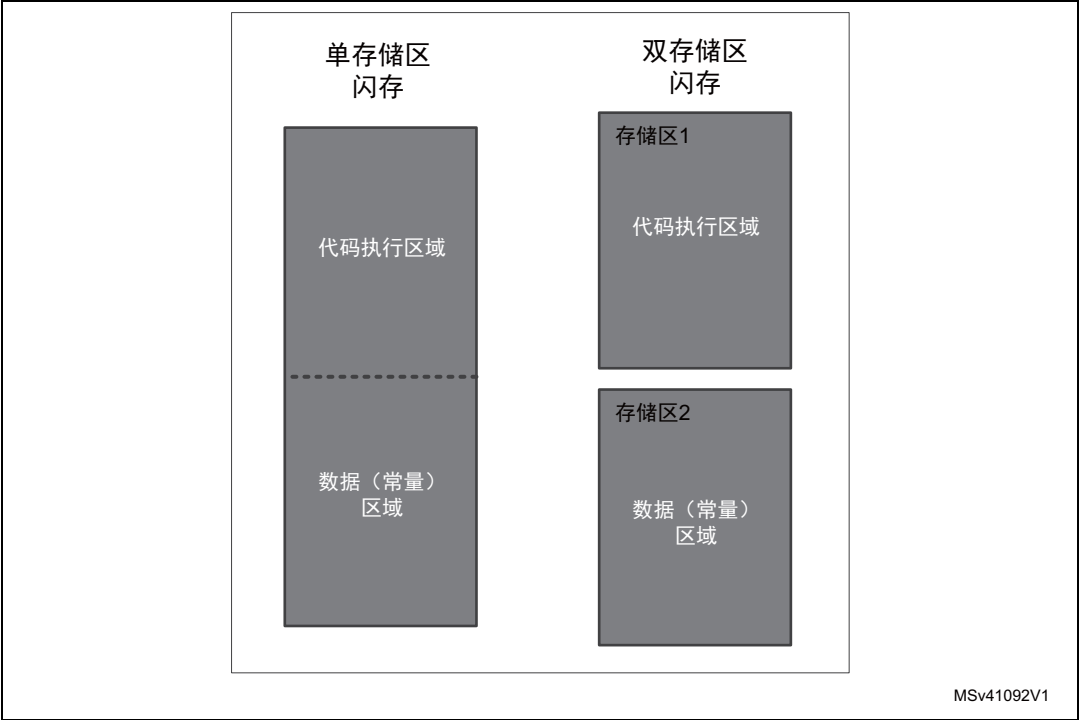
此示例演示了**同时读写**功能如何在双存储区闪存配置中正常运行。此特性不适用于单存储区闪存配置。

在此示例中，用户在执行LED闪烁时使用数据（常量）对闪存进行编程。

第一种情况显示了如何在单存储区配置中使用闪存，以使编程的常量和执行的代码位于同一个闪存存储区中。

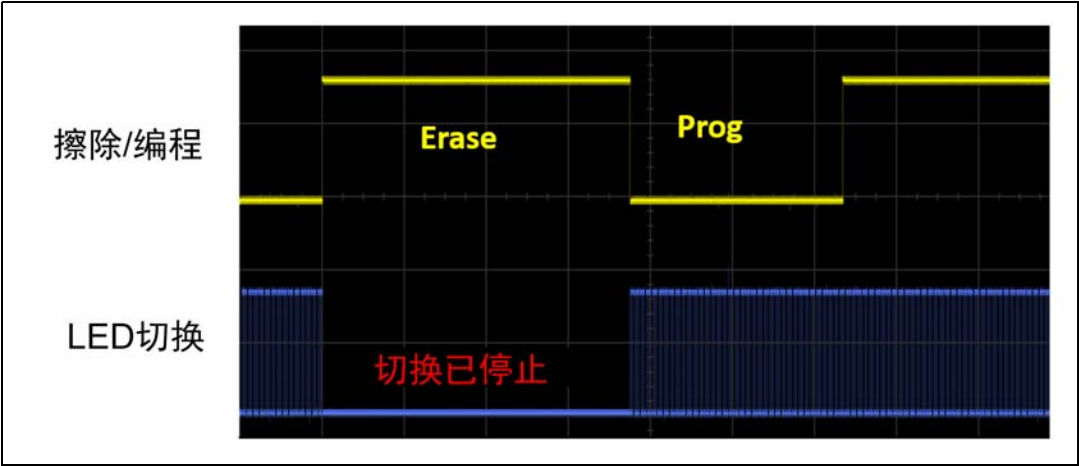
第二种情况显示了如何在双存储区配置中配置闪存，并让用户能够存储常量，在不同的存储区中执行代码。

图13. RWW 示例



5.1.2 单存储区配置

图14. 单存储区配置中的RWW



上述波形示例演示了当CPU从闪存执行代码时，如何在擦除操作开始时停止执行（LED闪烁）。如 [第 2 节：同时读写\(RWW\)](#) 所述，此示例证明，在单存储区模式下不允许从闪存执行和编程/擦除。

5.1.3 双存储区配置

图15. 双存储区配置中的RWW



在闪存双存储区模式下，从存储区1执行LED闪烁指令，同时在存储区2中进行常量编程。因此，CPU执行没有失效，如图15所示。

5.1.4 EEPROM仿真

EEPROM仿真涉及使用带有两个存储区闪存的STM32F7系列器件：一个作为闪存访问，另一个作为EEPROM访问。对于大多数器件，这无需使用仿真软件，还允许同时读写（RWW）操作，从而减少了访问延迟。

EEPROM仿真的应用示例已开发完毕，可以从STM32Cube_FW_F7封装加载。

5.2 双启动示例

5.2.1 说明

双启动示例描述了STM32F7系列器件如何根据有效地址，从存储区1或存储区2启动。

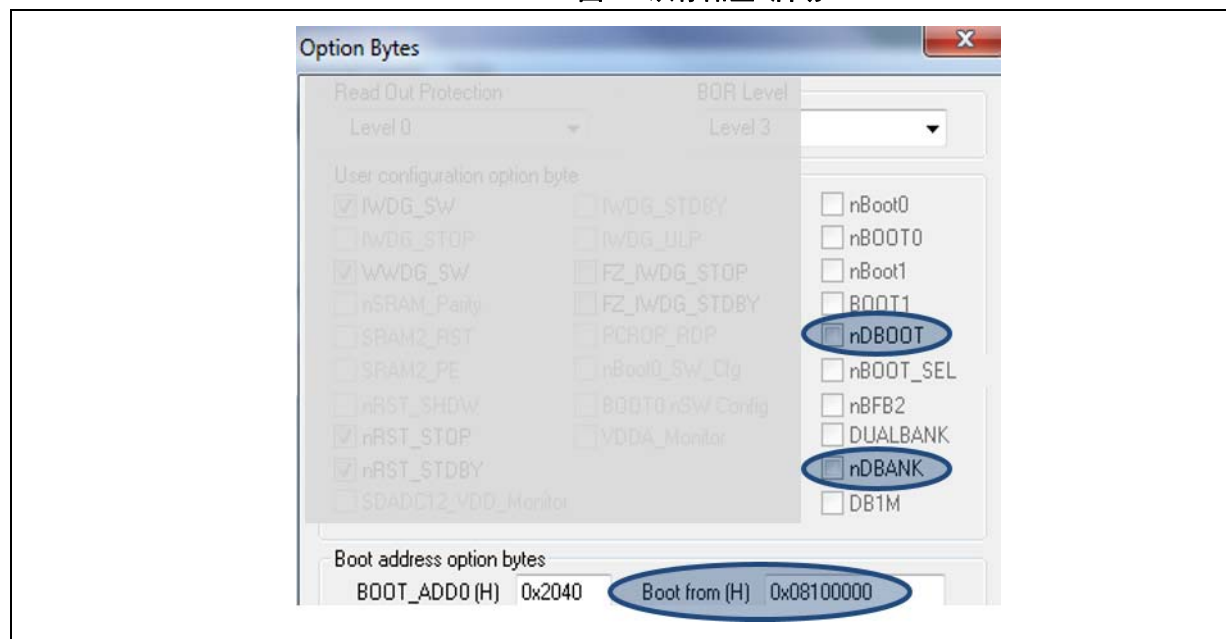
存储区1的代码闪烁LED1，存储区2中的代码闪烁LED2。

如要运行此示例，用户必须执行以下步骤：

- 使用STM32 ST-LINK Utility在双启动模式下配置闪存（nDBANK = nDBOOT = 0）。
- 构建FLASH_DualBoot_Bank2项目，使用STM32 ST-LINK Utility将二进制文件FLASH_DualBoot_Bank2.bin加载到地址0x08100000。
- 构建FLASH_DualBoot_Bank1项目，将其加载到地址0x08000000。

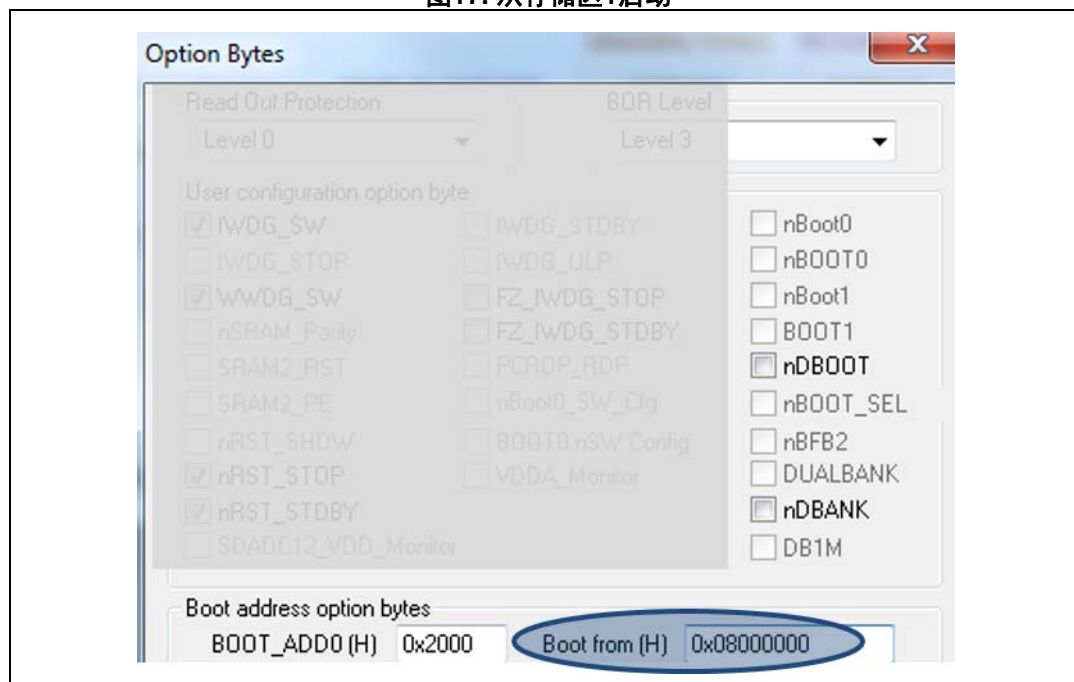
- 如果启动地址为0x081000000，则交换闪存存储区，自举程序跳转到存储区2，以运行LED2闪烁。

图16. 从存储区2启动



- 如果启动地址为0x080000000，则自举程序跳转到启动地址，以运行LED1切换。

图17. 从存储区1启动



5.3 性能和功耗示例

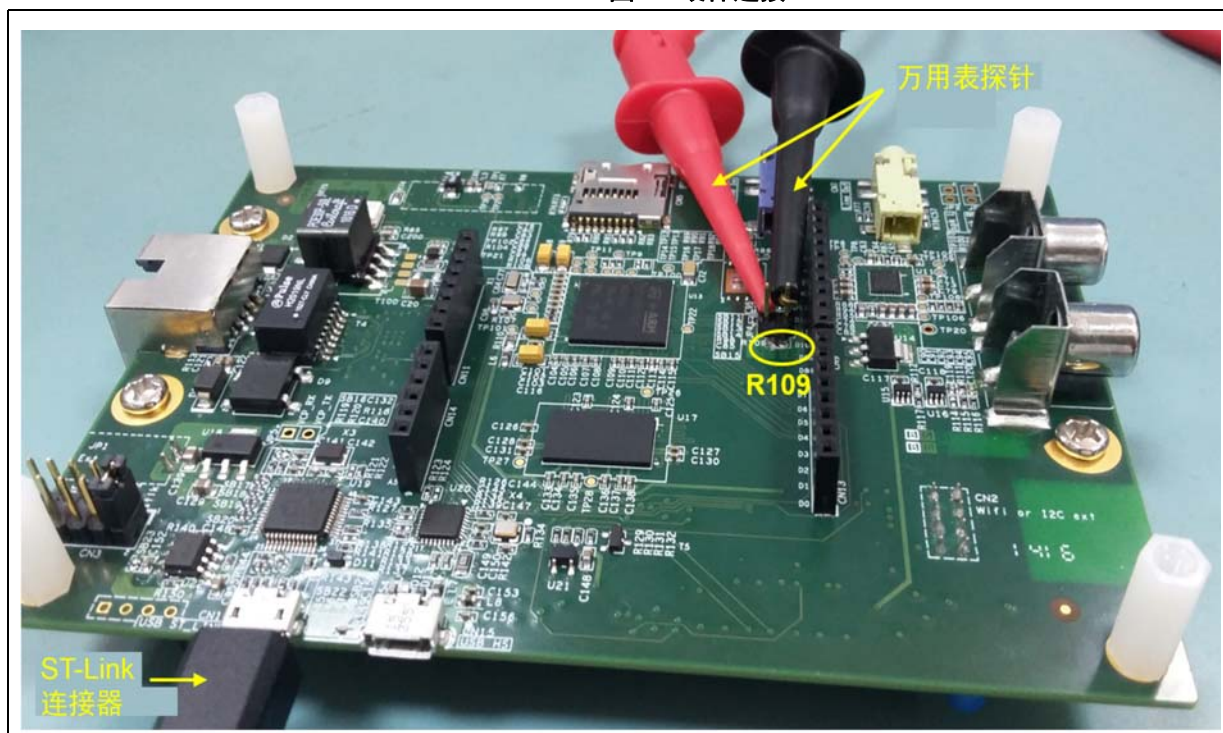
5.3.1 硬件要求

如要运行示例，用户必须：

- STM32F769I-DISCO板。
- 使用带微型USB的ST-Link，将探索板连接到PC，进行编程和调试。
- 使用万用表来测量电流消耗。

为了测量微控制器的电流消耗，用户需要从探索板上移除电阻R109，如图 18所示：

图18. 硬件连接



5.3.2 测量结果

STM32F7系列器件带有单存储区模式和双存储区模式的闪存，可通过运行“CMSIS^{ARM®}图形均衡器”算法，测量其性能和功耗。使用的工具链是MDK-RAM。

- 测量结果：

性能值显示在终端上，如图 19所示：

图19. 性能显示

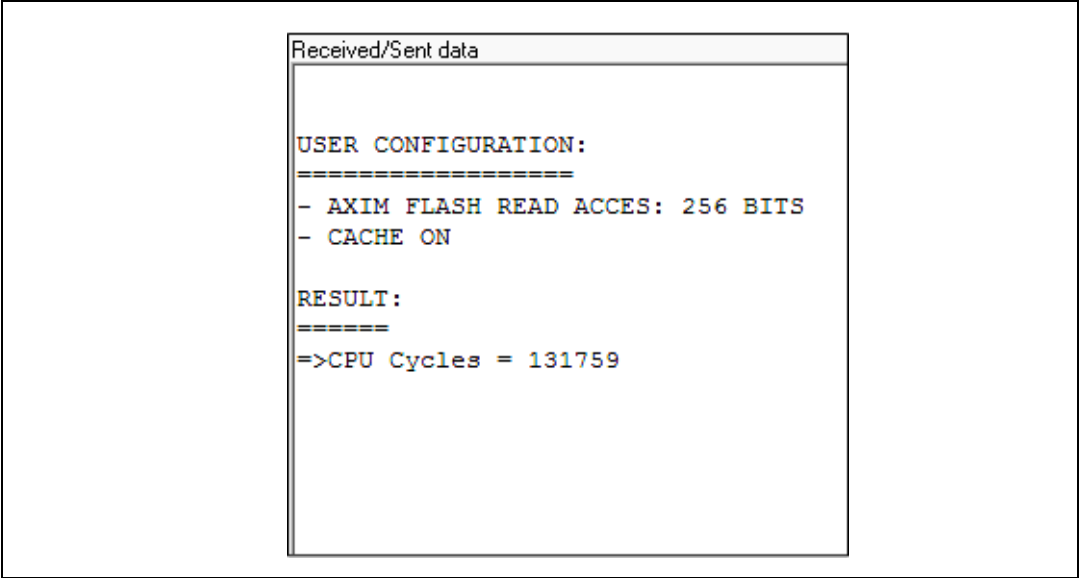


表 2总结了CMSIS ARM®图形均衡器的STM32F7性能指标:

表2. 性能指标

闪存配置		性能指标		
		单存储区 (CPU 周期)	双存储区 (CPU 周期)	增益
Flash_AXI	缓存开	131759	131760	0%
	缓存关	365656	442481	- 21%
Flash_ITCM	ART开	137989	154146	- 12%
	ART关	236302	320732	- 36%

- 功耗测量结果

表 3总结了STM32F7的功耗指标:

表3. 功耗指标

闪存配置		功耗指标		
		单存储区 (mA)	双存储区 (mA)	增益
Flash_AXI	缓存开	111	111	0%
	缓存关	96	85	11%
Flash_ITCM	ART开	117	114	3%
	ART关	118	95	19%

根据表 3和表 4 的指标，从单存储区模式切换到双存储区模式表明：

- 如果缓存和ART为“关”，则性能下降，且双存储区模式中功耗增益。
- 如果缓存和ART为“开”，则性能不会降低，且在双存储区模式下仍然存在功耗增益。

在双存储区模式下，可以根据用户应用节省功耗，由于STM32F7ART和缓存，单和双存储区模式的性能大致相同。

6 结论

本应用笔记演示了闪存双存储区模式如何响应广泛应用的严格实时需求，解决众多问题，例如通过同时读写功能减少访问延迟。

使用双存储区存储器可以随时在STM32F7系列器件中使用固件的至少一个工作版本。如果在升级期间出现问题（例如掉电或连接丢失），上述功能有助于避免固件损坏。

7 版本历史

表4. 文档版本历史

日期	版本	变更
2016年6月10日	1	初始版本。

表5. 中文文档版本历史

日期	版本	变更
2018年11月21日	1	中文初始版本。

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和 / 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。本文档的中文版本为英文版本的翻译件，仅供参考之用；若中文版本与英文版本有任何冲突或不一致，则以英文版本为准。

© 2018 STMicroelectronics - 保留所有权利