

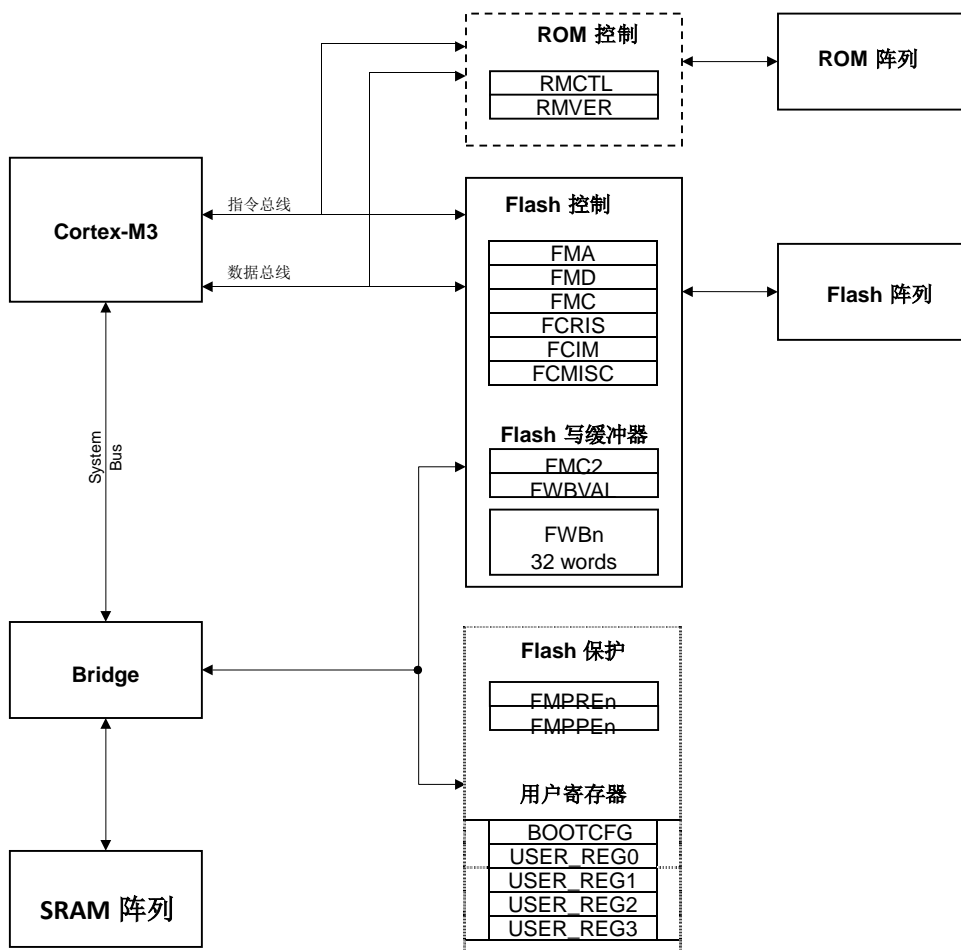
6 内部存储器

LM3S9B96微控制器带有96 KB的位带 SRAM、内部ROM和256KB的Flash存储器。Flash存储器控制器提供了一个友好的用户接口，使Flash编程成为一项简单的任务。在Flash存储器中可应用Flash存储器保护，以2KB块大小为单位。

6.1 框图

第309页的图6-1说明了内部存储器的结构框图和控制逻辑。图中的虚线框表示处于系统控制模块中的寄存器。

图 6-1. 内部存储器方框图



6.2 功能描述

本节描述SRAM、ROM和Flash存储器的功能。

注意： μ DMA 控制器可以将数据转移到片上SRAM，也可以从片上SRAM将数据转移出。但是，由于Flash存储器和ROM位于不同的内部总线，所以 μ DMA不能从Flash存储器或ROM转移数据。

6.2.1 SRAM

注意： SRAM 使用两个32位的SRAW存储区来实现功能(分离的SRAM阵列)。存储区被这样分开，以便一个包含所有偶数字(偶存储区)，另一个包含所有奇数字(奇存储区)。对同一个存储区执行读访问后立即执行写访问，中间会引起一个单时钟周期的停顿。但是对一个存储区执行读访问后对另一个存储区执行写访问，则可以在连续时钟周期内进行而不引起延迟。

Stellaris[®] 设备的内部SRAM位于器件存储器的映射地址为0x2000.0000。为了减少读-修改-写的操作时间，ARM在Cortex-M3处理器中植入了**位带**技术。在位带使能的处理器中，存储器映射的特定区域(SRAM和外设空间)能够使用地址别名，在单个原子操作中访问各个位。位带基址位于0x2200.0000。

位带别名可以使用下面的公式计算：

位带别名 = 位带基址 + (字节偏移量 * 32) + (位编号 * 4)

例如，如果要修改地址0x2000.1000的第3位，位带别名的计算如下：

$0x2200.0000 + (0x1000 * 32) + (3 * 4) = 0x2202.000C$

通过计算得出的位带别名，对地址0x2202.000C执行读/写操作的指令可以直接访问地址0x2000.1000处字节的第3位。

关于位带的详细信息，可查看第99页的“位带”。

6.2.2 ROM

Stellaris[®] 设备的内部ROM位于器件存储器的映射地址为0x0100.0000。ROM包含下面几部分：

- Stellaris[®] 引导装载程序和向量表
- Stellaris[®] 外设驱动库 (DriverLib)，为产品特定的外设和接口而发行
- SafeRTOS代码
- 高级加密标准 (AES) 密码表
- 循环冗余检验 (CRC) 错误检测功能

引导装载程序用作初始化程序的装载器(当Flash存储器为空时)，也可以作为一种应用——初始的固件升级机制(通过回调引导装载程序)。应用程序可以调用ROM中外设驱动库的API，以减少对Flash存储器的需求，释放Flash存储器空间用于其它目的(如应用程序增加的特性)。SafeRTOS是一个低费用、微型、可优先购买的实时调度程序。高级加密标准(AES)是美国政府使用的公开定义的加密标准。循环冗余检验(CRC)是一项技术，用来确认一段数据的内容与先前检验的相同。

6.2.2.1 引导装载程序概述

Stellaris[®] 引导装载程序用来将代码下载到设备的Flash存储器中，而不需要使用调试接口。任何复位内核的复位中，通过使用启动配置寄存器(BOOTCFG)中配置好的端口A-H的GPIO信号，用户可以选择让内核直接执行ROM的引导装载程序或Flash存储器上的应用程序。

复位时，ROM映射在Flash存储器上，以便ROM的启动序列总能被执行。从ROM中执行的启动序列如下：

1. 清BA位(在下面)以使ROM映射在0x01x.xxxx，同时Flash存储器映射在0x0。
2. 读BOOTCFG寄存器。如果EN位被置位，那么将指定的GPIO管脚的状态与规定的极性相比较，如果管脚状态与规定的极性匹配，那么执行ROM的引导装载程序。
3. 如果管脚状态与规定极性不匹配，检查地址0x0000.0004来看Flash存储器是否被编程。如果该地址的数据是0xFFFF.FFFF，那么执行ROM的引导装载程序。
4. 如果地址0x0000.0004的数据是有效的，那么堆栈指针(SP)装载Flash存储器地址0x0000.0000的数据，程序计数器(PC)装载地址0x0000.0004的数据。用户应用程序开始执行。

引导装载程序使用一个简单的封装接口提供与设备的同步通信。由于引导装载程序不能使能PLL，所以它的速度由内部振荡器(PIOSC)频率决定。下面的串行接口可以使用：

- UART0
- SSI0
- I²C0
- 以太网

为简单起见，所有串行接口的数据格式和通信协议都是相同的。

注意：引导装载程序的Flash存储器驻留版本也支持CAN和USB。有关引导装载程序的软件信息可查看 **Stellaris® 引导装载程序使用指南**。

6.2.2.2 Stellaris® 外设驱动库

Stellaris® 外设驱动库包含一个叫做driverlib/rom.h的文件，它帮助调用ROM中的外设驱动库函数。每个函数的详细描述可在**Stellaris® ROM使用指南**中获得。更多关于调用ROM函数和使用driverlib/rom.h的详细信息，可查看**Stellaris® 外设驱动库使用指南**中的“使用ROM”一章。

ROM起始处的表格指示了ROM提供的API的入口指针。通过这些表格访问API提供了可扩展性，因为API的位置可能会在将来的ROM版本中改变，而API表格不会变。该表格被分为两级，主表格包含的每个指针对应一个外设，该外设指向二级表格。二级表格包含的每个指针对应一个与外设相关的API。主表格的位置在0x0100.0010，恰好在Cortex-M3的ROM向量表后面。

DriverLib函数在**Stellaris® 外设驱动库使用指南**中有详细描述。

增加的API可用于图像和USB功能，但不会预装载到ROM。Stellaris® 图像库提供了一系列图像原型，其中一部分用来在基于Stellaris®微处理器的有图形显示的板子上建立图形用户接口(更多信息可查看 **Stellaris® 图像库使用指南**)。Stellaris® 的USB库是一系列数据类型和函数，用来在基于Stellaris®微处理器的板子上建立USB设备，主机或On-The-Go (OTG) 应用(更多信息可查看 **Stellaris® USB库使用指南**)。

6.2.2.3 SafeRTOS

SafeRTOS 属于 SIL3 RTOS 版本，它已经被认证可作为安全关键的应用。在功能上与FreeRTOS相似，但是扩展了安全功能。完全符合 IEC 61508的开发和安全生命周期记录是可获得的(由TÜV SÜD做一致性认证，包括编译器校证明)。

SafeRTOS 由 WITTENSTEIN 为了满足 IEC 61508的高完整性系统而制作。SafeRTOS在LM3S9B96微控制器的集成ROM内，可以使用SafeRTOS头文件、初始化代码和例程。为获得完全的SIL认证，可以从WITTENSTEIN 购买一致性规范。

更多的信息可参考 **SafeRTOS 使用说明**。

6.2.2.4 高级加密标准 (AES) 密码表

AES 是一种强大的加密方法，拥有不错的性能和大小。AES 在硬件和软件方面都很快，它非常容易使用，并且只需要很少的存储空间。AES 可理想的用于预先排列好密钥的应用，如在加工或配置过程中的设置好。XySSL AES 使用的4个数据表都在ROM中提供。第一个是正向的S-box代换表，第二个是反向的S-box代换表，第三个是正向的多项式表，最后一个是反向的多项式表。关于AES的更多信息可查看 **Stellaris® ROM 使用指南**。

6.2.2.5 循环冗余检验 (CRC) 错误检测

CRC 技术可用来确认信息的正确接收(在传送中没有丢失或改变)，用来确认解压后的数据，用来证实Flash存储器的内容没有更改，以及其它数据需要被确认的情况。CRC 优于简单的校验和(例如同异或所有的位)，因为它更容易捕捉到变化。关于CRC的更多信息可查看 **Stellaris® ROM 使用指南**。

6.2.3 Flash 存储器

在系统时钟速度为50MHz或以下时，Flash存储器是单周期读取的。Flash存储器由一系列1KB的块组织在一起，这些块可以被单独擦除。一个单独的32位字可以被编程将位从1变为0。另外，写缓冲器提供了Flash存储器中连续32个字同时编程的能力。擦除一个块将使块中的所有位变为1。1KB的块可以配成一系列2KB的块，2KB块可以被单独保护。该保护允许块被标记为只读或只执行，以提供不同等级的代码保护。只读块不能被擦除或编程，块的内容受保护不能修改。只执行块不能被擦除或编程，只能通过控制器取指机制来读取它的内容，块的内容受保护不能被控制器或调试器读取。

警告 – 在微控制器小于5分钟频繁上电的系统中，电源应该以一种可控的方式从微控制器去除以确保正常工作。软件应该使用Flash控制器寄存器 (FCTL) 的USDREQ位请求掉电允许，接着等待接收一个来自USDACK位的确认，然后去除电源。

6.2.3.1 预取指缓冲器

Flash存储器控制器有一个预取指缓冲器，当CPU频率大于50MHz时它将自动启用。在此模式下，Flash存储器以一半系统时钟的工作。每个时钟周期，预取指缓冲器获取两个32位字，这样在代码线性执行时允许取指不处于等待状态。取指缓冲器包含一个分支推断机制，它可以辨认出分支从而避免因读取下一对字而增加的额外等待状态。并且，短循环分支经常保持在缓冲器中。因此，一些分支可以没有等待状态而执行。其它分支会引起一个单独的等待状态。

6.2.3.2 Flash 存储器保护

在4对32位寄存器中，以2KB Flash块为基础，用户有两种形式的Flash保护。由FMPPEn 和 FMPREn寄存器的各个位来控制器每种形式的保护策略 (每个块一个策略)。

- **Flash 存储器保护编程使能 (FMPPEn):** 如果某个位被置位，那么对应的块可以被编程 (烧写) 或擦除。如果被清零，对应的块不能更改。
- **Flash 存储器保护读使能 (FMPREn):** 如果某个位被置位，那么对应的块可以被软件或调试器执行或读取。如果被清零，对应的块只能被执行，块的内容禁止被作为数据读取。

这些策略进行组合，如第313页的图6-1所示。

表 6-1. Flash 存储器保护策略组合

FMPPEn	FMPREn	Protection
0	0	只执行保护。块只能被执行不能被写或擦除。该模式用于保护代码
1	0	块可以被写、擦除或执行，但不能读。这个组合不可能被使用。
0	1	只读保护。块可以被读取或执行，但是不能被写或擦除。该模式用来锁定块不被进一步修改，但允许读和执行访问。
1	1	没有保护。块可以被写、擦除、执行或读。

对Flash存储器的读保护块 (FMPREn位被置位) 尝试读访问是被禁止的，会产生一次总线故障。对Flash存储器的编程保护块 (FMPPEn位被置位) 尝试编程或擦除访问是被禁止的。可以选择产生一个中断 (通过置位Flash控制器中断屏蔽寄存器 (FCIM) 的AMASK位) 提醒软件开发者在开发和调试阶段的错误软件操作。

对于所有使用的存储区，FMPREn和FMPPEn寄存器的出厂设置值为1。这种设置实现了一种开放式的访问和可编程性的策略。寄存器的位可通过清零特定寄存器的位来改变。这种改变不是永久的，直到寄存器被提交 (保存)，此时位的改变是永久性的。如果一个位从1变为0且没有提交，那么它可以通过执行一段上电复位序列来恢复。

这些更改使用Flash控制寄存器 (FMC) 来提交。关于编程这些位的详细信息在第316页的“非易失性寄存器编程”讨论。

6.2.3.3 中断

Flash 控制器在检测到下列状态时会产生中断：

- 编程中断 - 当1个编程或擦除动作完成时发出信号。
- 访问中断 - 当对受相应FMPPEn位保护的2KB块存储器尝试编程或擦除操作时发出信号。

能够触发控制器级中断的事件在Flash控制器可屏蔽中断状态寄存器 (FCMIS) (参阅第325页)定义，置位相应的位可以使能该中断。如果不使用中断，原始的中断状态总是在Flash控制器原始中断状态寄存器 (FCRIS) (参阅第324页)中可见。

对Flash控制器可屏蔽中断的状态和清除寄存器 (FCMISC) (参阅第326页)的位写1可以清除相应的中断 (对于FCMIS寄存器和FCRIS寄存器)。

6.3 Flash 存储器初始化和配置

6.3.1 Flash 存储器编程

Stellaris[®] 设备为Flash存储器编程提供了一个友好的用户接口。所有的擦除/编程操作都通过三个寄存器处理：Flash存储器地址 (FMA)，Flash存储器数据 (FMD) 和Flash存储器控制 (FMC)。注意，如果微控制器的调试功能没有激活而处在“锁死”状态，必须执行一段恢复序列来再激活调试模块。查阅第192页的“恢复一个“锁死”的微控制器”。

在Flash存储器操作 (写、页擦除或整体擦除) 过程中，对它进行访问是禁止的。所以指令和按字取指都将延迟到Flash存储器操作完成。如果在Flash存储器操作过程中需要执行指令，那么代码必须放置在SRAM上并在SRAM上执行。

警告 – Flash存储器被分隔为每块4KB的电气分离扇区，以4KB的范围对齐。在扇区内的1KB页上执行擦除/编程操作会对其它3个1KB的页有电气影响。一个特定1KB页的擦除必须在对它所在的4KB扇区的其它页执行6个完整的擦除/编程周期后进行。下面对Flash存储器4KB扇区 (第0..3页) 的操作序列提供一个示例：

- 第3页被擦除和编程数值。
 - 第0页、1页和2页都被擦除，然后编程数据。此时第3页已经被3个擦除/编程周期影响。
 - 第0页、1页和2页又被擦除，然后编程数据，此时第3页已经被6个擦除/编程周期影响。
 - 如果第3页的内容必须持续有效，第3页必须在该扇区的任意其它页进行另外的擦除或编程操作前被擦除并重新编程。
-

6.3.1.1 编程1个32位字

1. 将源数据写入**FMD**寄存器。
2. 将目标地址写入**FMA**寄存器。
3. 将Flash写密钥写入**FMC**寄存器，并置位**WRITE**位（写入0xA442.0001）。
4. 查询**FMC**寄存器，直到**WRITE**位被清零。

重要: 为确保正确的操作，对同一个字的两次写入必须插入一次清除。下面的两个序列是允许的：

- 擦除 -> 编程数值 -> 编程0x0000.0000
- 擦除 -> 编程数值 -> 擦除

下面的序列是不允许的：

- 擦除 -> 编程数值 -> 编程数值
-

6.3.1.2 执行一个1KB页的擦除

1. 将页地址写入 **FMA** 寄存器。
2. 将Flash写密钥写入**FMC**寄存器，并置位**ERASE**位（写入 0xA442.0002）。
3. 查询 **FMC** 寄存器，直到**ERASE**位被清零。

6.3.1.3 执行一次Flash存储器的整体擦除

1. 将Flash写密钥写入**FMC**寄存器，并置位**MERASE**位（写入 0xA442.0004）。
2. 查询 **FMC** 寄存器，直到**MERASE**位被清零。

6.3.2 32字Flash存储器写缓冲器

通过在一个单独被缓存的Flash存储器写操作中同时编程32个字，32字的写缓冲器提供了对Flash存储器执行更快的写访问的能力。被缓存的Flash存储器写操作与一个由**FMC**寄存器第0位控制的单独的字写入操作花费相同的时间。被缓存的数据写入Flash写缓冲器寄存器 (**FWBn**)。

该寄存器是由32个字在Flash存储器对齐组成的，所以**FWB0**在**FMA**对应的地址，其[6:0]位都是0。**FWB1**对应的地址在**FMA+0x4**，以此类推。**FWBn**寄存器只有经过之前被缓存的Flash存储器写操作后被更新才能被写入Flash存储器。**Flash**写缓冲器有效寄存器 (**FWBVAL**) 显示了从上次被缓存的Flash存储器写操作之后，哪个寄存器已经被更新。该寄存器包含的位对应32个**FWBn**寄存器，第[n]位对应**FWBn**。如果**FWBVAL**寄存器的位被置位，那么相应的**FWBn**寄存器已经被更新了。

6.3.2.1 用一次单独被缓冲的Flash存储器写操作来编程32个字

1. 将源数据写入FWBn寄存器。
2. 将目标地址写入FMA寄存器。该地址必须是一个32字对齐的地址 (即FMA的[6:0]必须是0)。
3. 将Flash写密钥写入FMC2寄存器，并置位WRBUF位 (写入 0xA442.0001)。
4. 查询 FMC2寄存器，直到WRBUF位被清零。

6.3.3 非易失性寄存器编程

本节讨论如何更新Flash存储器自身中的寄存器。这些寄存器驻留在与主Flash存储器阵列分离的空间，并且不受擦除或整体擦除的影响。写操作可以将这些寄存器中的位由1变为0。除了上电复位会把寄存器的内容复位为0xFFFF.FFFF外，其它的复位都不会影响寄存器。在上电序列后，可以使用FMC寄存器的COMT位来提交确认寄存器的值，然后寄存器的值变为非易失性保持下去。一旦寄存器的内容被更改，唯一能恢复出厂默认值的办法就是执行第192页描述的“恢复一个“锁死”的微控制器”的序列。

除了启动配置寄存器 (BOOTCFG)，对其它寄存器的设置在提交Flash存储器前可以被测试。对于BOOTCFG寄存器，在它提交前先将要写入的数据载入FMD寄存器。在BOOTCFG寄存器提交到非易失性存储器之前，FMD寄存器是只读的，并且不允许尝试BOOTCFG操作。

重要： Flash存储器驻留寄存器只能被用户程序由1改为0，并且只能更改1次。更改后，唯一能恢复出厂默认值的办法就是执行第192页描述的“恢复一个“锁死”的微控制器”的序列。由该序列引起的对主Flash存储器阵列执行整体擦除优先于恢复这些寄存器。

另外，USER_REG0、USER_REG1、USER_REG2、USER_REG3 和 BOOTCFG 寄存器都用第31位 (NW) 来指示它们没有被提交，并且寄存器的位只能由1改为0。第316页的表6-2提供了每个要提交的寄存器的FMA地址，也提供了当FMC寄存器写入值0xA442.0088时寄存器要被写入的源数据。在写COMT位之后，用户可以查询FMC寄存器，等待提交操作完成。

表 6-2. 用户可编程的Flash存储器驻留寄存器

被提交的寄存器	FMA 值	数据源
FMPRE0	0x0000.0000	FMPRE0
FMPRE1	0x0000.0002	FMPRE1
FMPRE2	0x0000.0004	FMPRE2
FMPRE3	0x0000.0006	FMPRE3
FMPPE0	0x0000.0001	FMPPE0
FMPPE1	0x0000.0003	FMPPE1
FMPPE2	0x0000.0005	FMPPE2

表 6-2. 用户可编程的Flash存储器驻留寄存器 (续)

被提交的寄存器	FMA 值	数据源
FMPPE3	0x0000.0007	FMPPE3
USER_REG0	0x8000.0000	USER_REG0
USER_REG1	0x8000.0001	USER_REG1
USER_REG2	0x8000.0002	USER_REG2
USER_REG3	0x8000.0003	USER_REG3
BOOTCFG	0x7510.0000	FMD

6.4 寄存器映射

第317页的表 6-3 列出了ROM控制器寄存器和Flash存储器和控制寄存器。列出的偏移量是寄存器地址的16进制增量。**FMA**, **FMD**, **FMC**, **FCRIS**, **FCIM**, **FCMISC**, **FMC2**, **FWBVAL**和 **FWBn** 寄存器偏移量都是相对于Flash存储器控制基址0x400F.D000的。ROM和Flash存储器保护寄存器的偏移量是相对于系统控制基址0x400F.E000的。

Table 6-3. Flash 寄存器映射

偏移量	名称	类型	复位	描述	见页面
Flash 存储器寄存器 (Flash 控制偏移量)					
0x000	FMA	R/W	0x0000.0000	Flash 存储器地址	319
0x004	FMD	R/W	0x0000.0000	Flash 存储器数据	320
0x008	FMC	R/W	0x0000.0000	Flash 存储器控制	321
0x00C	FCRIS	RO	0x0000.0000	Flash 控制器原始中断状态	324
0x010	FCIM	R/W	0x0000.0000	Flash 控制器中断屏蔽	325
0x014	FCMISC	R/W1C	0x0000.0000	Flash 控制器可屏蔽的中断状态和清除	326
0x020	FMC2	R/W	0x0000.0000	Flash 存储器控制2	327
0x030	FWBVAL	R/W	0x0000.0000	Flash 写缓冲器有效	328
0x0F8	FCTL	R/W	0x0000.0000	Flash 控制	330
0x100-0x17C	FWBn	R/W	0x0000.0000	Flash 写缓冲器n	329
存储器寄存器 (系统控制偏移量)					
0x0F0	RMCTL	R/W1C	-	ROM 控制	331
0x130	FMPRE0	R/W	0xFFFF.FFFF	Flash 存储器保护读使能0	332
0x200	FMPRE0	R/W	0xFFFF.FFFF	Flash 存储器保护读使能0	332
0x134	FMPPE0	R/W	0xFFFF.FFFF	Flash 存储器保护编程使能0	333
0x400	FMPPE0	R/W	0xFFFF.FFFF	Flash 存储器保护编程使能0	333
0x1D0	BOOTCFG	R/W	0xFFFF.FFFE	启动配置	334
0x1E0	USER_REG0	R/W	0xFFFF.FFFF	用户寄存器0	337

Table 6-3. Flash 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x1E4	USER REG1	R/W	0xFFFF.FFFF	用户寄存器 1	338
0x1E8	USER REG2	R/W	0xFFFF.FFFF	用户寄存器2	339
0x1EC	USER REG3	R/W	0xFFFF.FFFF	用户寄存器 3	340
0x204	FMPRE1	R/W	0xFFFF.FFFF	Flash 存储器保护读使能1	341
0x208	FMPRE2	R/W	0xFFFF.FFFF	Flash 存储器保护读使能2	342
0x20C	FMPRE3	R/W	0xFFFF.FFFF	Flash 存储器保护读使能3	343
0x404	FMPPE1	R/W	0xFFFF.FFFF	Flash 存储器保护编程使能1	344
0x408	FMPPE2	R/W	0xFFFF.FFFF	Flash 存储器保护编程使能2	345
0x40C	FMPPE3	R/W	0xFFFF.FFFF	Flash 存储器保护编程使能3	346

6.5 Flash 存储器寄存器描述 (Flash 控制偏移量)

本节按照地址偏移的数字顺序排列和描述Flash寄存器。本节的寄存器都是相对于Flash控制基址0x400F.D000的。

寄存器 1: Flash 存储器地址 (FMA), 偏移量 0x000

在一次写操作过程中，该寄存器包含一个4字节对齐的地址并指定在哪里写入数据。在擦除操作过程中，该寄存器包含一个1KB对齐的地址并指定擦除哪一页。注意必须要符合对齐的要求，否则操作的结果将不可预知。

Flash存储器地址 (FMA)

基址 0x400F.D000
偏移量 0x000
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留														偏移量	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	偏移量															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:18	保留	RO	0x0	软件不能依赖保留位的值。为了兼容将来的器件，保留位的值应该通过1次读-修改-写操作来保持不变。
17:0	偏移量	R/W	0x0	地址偏移量。

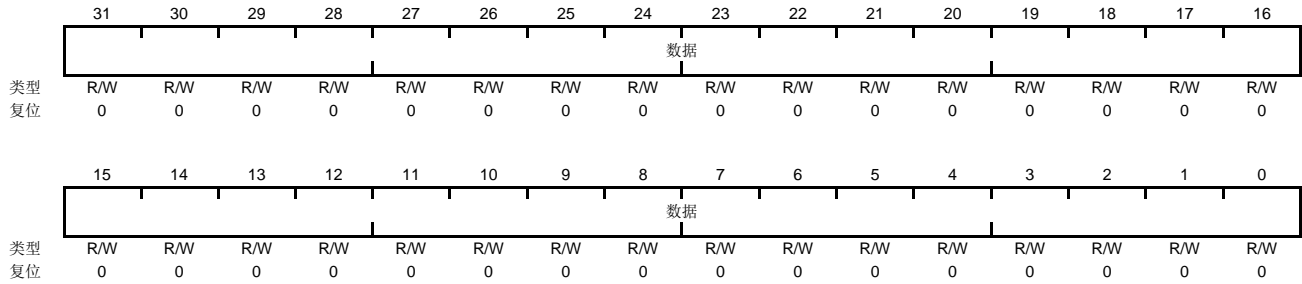
执行操作的Flash偏移量，非易失性寄存器除外（有关该域的值的具体内容见第316页的“非易失性寄存器编程”）。

寄存器 2: Flash 存储器数据 (FMD), 偏移量 0x004

该寄存器中包含的是编程周期中被写入的数据和读周期中被读出的数据。注意对只执行块进行读访问时，该寄存器的内容是没有定义的。该寄存器在擦除周期中不使用。

Flash 存储器数据 (FMD)

基址: 0x400F.D000
 偏移量: 0x004
 类型: R/W, 复位: 0x0000.0000



位/域	名称	类型	复位	描述
31:0	数据	R/W	0x0000.0000	数据值

写操作的数据值。

位/域	名称	类型	复位	描述
3	COMT	R/W	0	<p>提交寄存器值 该位用于提交对Flash存储器驻留寄存器的写入，并监视提交过程。</p> <p>值描述 1 置位该位可以向Flash存储器驻留寄存器提交 (写) 寄存器的值。 读取该位为1表示之前的提交访问没有完成。 0 写入0对该位的状态没有影响。 读取该位为0表示之前的提交访问完成。</p> <p>该操作最多花费50uS。 更过关于Flash存储器驻留寄存器的信息见第316页的“非易失性寄存器编程”。</p>
2	MERASE	R/W	0	<p>整体擦除Flash存储器 该位用于整体擦除Flash主存储器，并监视擦除过程。</p> <p>值描述 1 置位该位可以擦除Flash主存储器。 读取该位为1表示之前的整体擦除访问没有完成。 0 写入0对该位的状态没有影响。 读取该位为0表示之前的整体擦除访问完成。</p> <p>一次整体擦除最多花费16ms。</p>
1	ERASE	R/W	0	<p>擦除Flash存储器的页 该位用于擦除Flash存储器的页，并监视擦除过程。</p> <p>值描述 1 置位该位可以擦除Flash存储器中由FMA寄存器内容指定的页。 读取该位为1表示之前的页擦除访问没有完成。 0 写入0对该位的状态没有影响。 读取该位为0表示之前的页擦除访问完成。</p> <p>一次页擦除最多花费25ms。</p>

位/域	名称	类型	复位	描述
0	WRITE	R/W	0	<p>写一个字到Flash存储器 该位用于写一个字到Flash存储器，并监视写入过程。</p> <p>值描述</p> <p>1 置位该位可以将FMD寄存器的值存储到Flash存储器中FMA寄存器内容指定的位置。 读取该位为1表示写入更新访问没有完成。</p> <p>0 写入0对该位的状态没有影响。 读取该位为0表示之前的写入更新访问完成。</p> <p>写入一个字最多花费50us。</p>

寄存器 4: Flash 控制器原始中断状态 (FCRIS), 偏移量 0x00C

该寄存器指示Flash控制器有一个中断状态。一个中断信号只有其对应的FCIM寄存器位被置位才能发送到中断控制器。

Flash 控制器原始中断状态 (FCRIS)

基址: 0x400F.D000
偏移量: 0x00C
类型: RO, 复位: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															PRIS	ARIS
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:2	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值应该通过1次读-修改-写操作来保持不变。
1	PRIS	RO	0	<p>编程的原始中断状态</p> <p>该位给出编程周期的状态，编程周期是指通过FMC或FMC2寄存器位产生的写入或擦除操作(见第321页和第327页)。</p> <p>值描述</p> <p>1 编程周期已经完成。</p> <p>0 编程周期没有完成。</p> <p>当FCIM寄存器的PMASK位置位时，该位状态会发送给中断控制器。清零该位可通过写1到FCMISC寄存器的PMISC位来实现。</p>
0	ARIS	RO	0	<p>访问的原始中断状态</p> <p>值描述</p> <p>1 试图对存储器块进行的编程或擦除行为与该存储器块由FMPPE_n寄存器置位的保护策略相矛盾。</p> <p>0 没有不合适的编程或擦除试图访问存储器。</p> <p>当FCIM寄存器的AMASK位置位时，该位状态会发送给中断控制器。清零该位可通过写1到FCMISC寄存器的AMISC位来实现。</p>

寄存器 5: Flash 控制器中断屏蔽 (FCIM), 偏移量 0x010

该寄存器控制 Flash 控制器是否向控制器产生中断。

Flash 控制器中断屏蔽 (FCIM)

基址: 0x400F.D000

偏移量: 0x010

类型: R/W, 复位: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														PMASK	AMASK
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值应该通过1次读-修改-写操作来保持不变。
1	PMASK	R/W	0	编程中断屏蔽 该位控制着编程原始中断状态向中断控制器的报告。 值描述 1 当PRIS位置位时，向中断控制器发送一个中断。 0 PRIS中断被阻止向中断控制器发送。
0	AMASK	R/W	0	访问中断屏蔽 该位控制着访问原始中断状态向中断控制器的报告。 值描述 1 当ARIS位置位时，向中断控制器发送一个中断。 0 ARIS中断被阻止向中断控制器发送。

寄存器 6: Flash控制器可屏蔽中断的状态和清除 (FCMISC), 偏移量 0x014

该寄存器提供两个功能。首先, 它可以通过指示哪个中断源或哪些中断源正在发出中断信号来报告中断产生的原因。其次, 它提供清除中断报告的办法。

Flash 控制器可屏蔽中断的状态和清除 (FCMISC)

基址: 0x400F.D000
偏移量: 0x014
类型: R/W1C, 复位: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														PMISC	AMISC
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值应该通过1次读-修改-写操作来保持不变。
1	PMISC	R/W1C	0	编程可屏蔽的中断状态和清除 值描述 1 当读取该位为1时, 表示一个非屏蔽中断信号被发出, 原因是一个编程周期完成。 对该位写1会清PMISC位和FCRIS寄存器的PRIS位 (见324页)。 0 当读取该位为0时, 表示没有发生编程周期完成中断。 对该位写0不影响它的状态。
0	AMISC	R/W1C	0	访问可屏蔽的中断状态和清除 值描述 1 当读取该位为1时, 表示一个非屏蔽中断信号被发出, 原因是试图对存储器块进行的编程或擦除行为与该存储器块由FMPPEn寄存器置位的保护策略相矛盾。 对该位写1会清AMISC位和FCRIS寄存器的ARIS位 (见324页)。 0 当读取该位为0时, 表示没有发生不合适的访问。 对该位写0不影响它的状态。

寄存器 7: Flash存储器控制2 (FMC2), 偏移量 0x020

当该寄存器被写入时，Flash控制器将对Flash存储器地址寄存器 (FMA) (见319页) 指定的位置启动适当的访问周期。如果访问是写操作，那么Flash写缓冲器寄存器 (FMB) 中的数据将会写入指定地址。

该寄存器是启动存储器操作过程中最后被写入的寄存器。

Flash存储器控制2 (FMC2)

基址 0x400FD000
偏移量 0x020
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	写密钥															
类型	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															WRBUF
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	写密钥	WO	0x0000	Flash存储器写密钥 该域包含一个写密钥，用于最大限度的减少对Flash的意外写入。必须将值0xA442写入该域来触发写操作。如果写入FMC2寄存器时不包含WRKEY的值，那么这次写入将被忽略。读取该域将返回0。
1	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值应该通过1次读-修改-写操作来保持不变。
0	WRBUF	R/W	0	被缓存的Flash存储器写入 该位用来启动一次对被缓存的Flash存储器写入。

值描述

1 置位该位将会把FWBn寄存器的数据存储到FMA寄存器内容所指定的位值。

当读取该位为1时，表示之前的被缓存的Flash存储器写访问没有完成。

0 对该位写0不影响它的状态。

当读取该位为0时，表示之前的被缓存的Flash存储器写访问完成。

一次被缓存的Flash存储器写最多花费4ms。

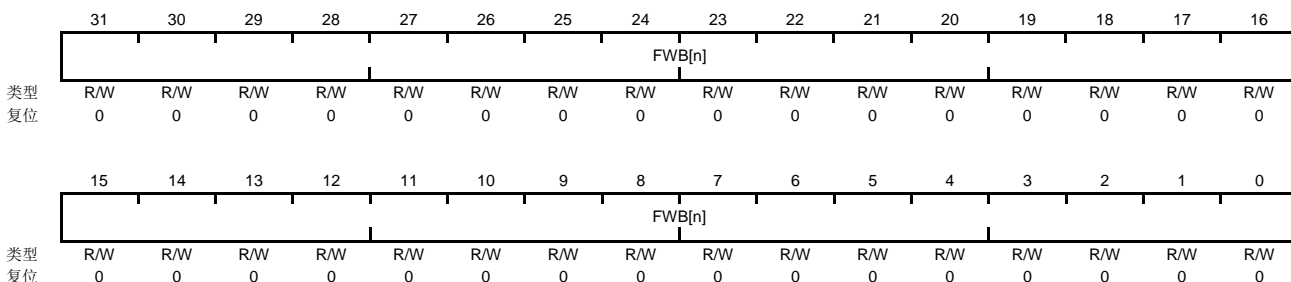
寄存器 8: Flash 写缓冲器有效 (FWBVAL), 偏移量 0x030

该寄存器逐位提供了哪个FWBn寄存器经过上次Flash存储器写缓冲器的写操作后, 已经被处理器写入的状态。值为1的寄存器会在下一次Flash存储器写缓冲器写操作中被写入。该寄存器在写操作后被硬件清零。写操作中违反保护也会清该状态。

在写操作清零FWB[n]位后, 软件可以通过置位FWB[n]位来将同样的32个字编程到不同的Flash存储器位置。于是接着的写操作将使用与上次写操作相同的数据。另外, 如果一个FWBn寄存器的改变不愿被写入Flash存储器, 当下一次写操作发生时, 软件可以清零相应的FWB[n]位来保留当前的数据。

Flash 写缓冲器有效 (FWBVAL)

基址 0x400F.D000
 偏移量 0x030
 类型 R/W, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:0	FWB[n]	R/W	0x0000	Flash存储器写缓冲器

值描述

- 1 相应的FWBn寄存器已经从上次缓冲器写操作之后更新, 并准备好写入Flash存储器。
- 0 相应的FWBn寄存器没有被写入新的数据。

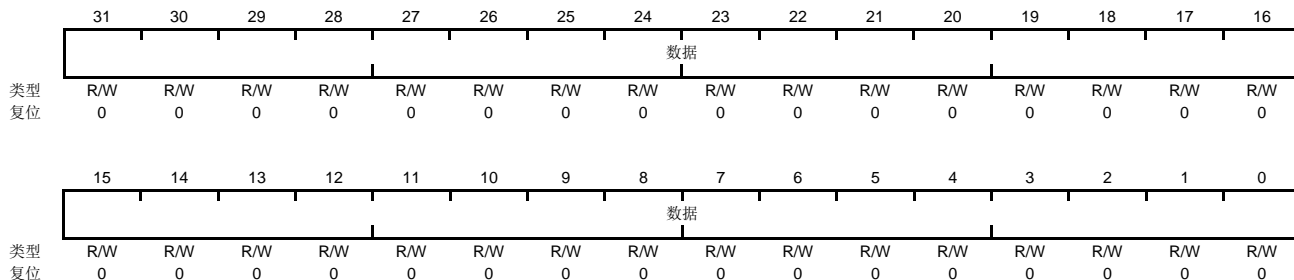
第0位对应FWB0, 偏移量0x100, 第31位对应FWB31, 偏移量0x17C。

寄存器 9: Flash 写缓冲器n (FWBn), 偏移量 0x100 - 0x17C

这32个寄存器包含的数据将在一次被缓存的Flash存储器写操作中写入Flash存储器。偏移量选择其中的一个32位寄存器。只有经过之前的被缓存的Flash存储器写操作后被更新的FWBn寄存器才可以写入Flash存储器。所以没有必要为了写1或2个字而写满所有的寄存器存储区。FWBn 寄存器写入Flash存储器时, FWB0寄存器对应FMA中包含的地址, FWB1写入地址FMA+0x4, 以此类推。注意只有数据的位为0才会导致Flash存储器被修改。数据位为1将保持Flash存储器位的内容为它之前的值。

Flash写缓冲器n (FWBn)

基址: 0x400F.D000
偏移量: 0x100 - 0x17C
类型: R/W, 复位: 0x0000.0000



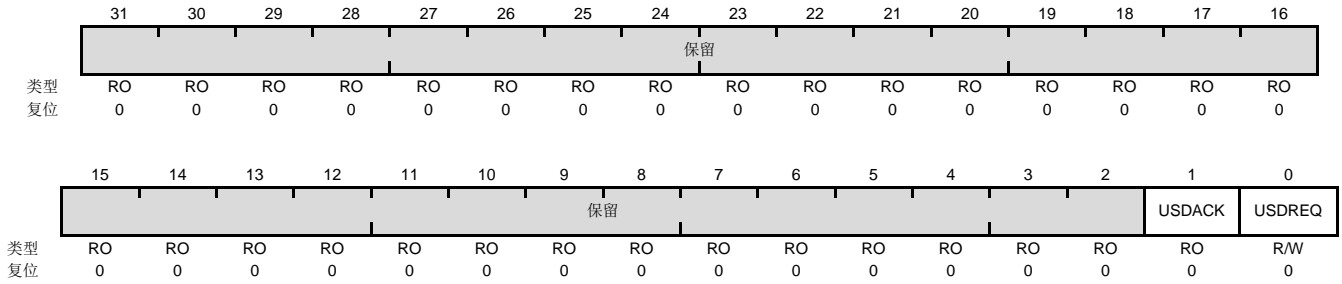
位/域	名称	类型	复位	描述
31:0	数据	R/W	0x0000.0000	数据 将要写入Flash存储器的数据。

寄存器 10: Flash控制 (FCTL), 偏移量 0x0F8

该寄存器用来确保在电源掉电频率大于每5分钟1次的系统里，微控制器的掉电处于可控的方式。电源将要关闭时应该置位USDREQ位。软件应该查询USDACK位来决定什么时候掉电是可接受的。

Flash 控制 (FCTL)

基址 0x400F.D000
 偏移量 0x0F8
 类型 R/W, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:2	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值应该通过1次读-修改-写操作来保持不变。
1	USDACK	RO	0	用户关闭确认 值描述 1 微控制器可以掉电。 0 微控制器仍然不能掉电。 该位会在置位USDREQ后的50ms之内置位。
0	USDREQ	R/W	0	用户关闭请求 值描述 1 请求允许关闭微控制器。 0 没有影响。

6.6 存储器寄存器描述 (系统控制偏移量)

本节的剩余部分按照地址偏移的数字顺序排列和描述Flash存储器寄存器。本节的寄存器都是相对于系统控制基址0x400F.E000的。

寄存器 11: ROM 控制 (RMCTL), 偏移量 0x0F0

该寄存器提供了对ROM控制器状态的控制。该寄存器的偏移量是相对于系统控制基址0x400F.E000的。

复位时，ROM被映射到Flash存储器以便ROM启动序列总可以被执行。从ROM执行的启动序列如下：

1. 清零BA位 (如下所示) 以便ROM映射到 0x01xx.xxxx 同时 Flash 存储器映射到地址0x0。
2. 读取 **BOOTCFG** 寄存器。如果EN位被置位，那么将指定的GPIO管脚状态与规定的极性比较，如果管脚状态与规定的极性匹配，那么执行ROM的引导装载程序。
3. 如果管脚状态与规定极性不匹配，检查地址 0x0000.0004来看Flash存储器是否被编程。如果该地址的数据是0xFFFF.FFFF，那么执行ROM的引导装载程序。
4. 如果地址0x0000.0004的数据是有效的，堆栈指针 (SP) 装载Flash存储器地址0x0000.0000的数据，程序计数器 (PC) 装载地址0x0000.0004的数据。用户应用程序开始执行。

ROM 控制 (RMCTL)

基址 0x400F.E000
偏移量 0x0F0
类型 R/W1C, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															BA
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值应该通过1次读-修改-写操作来保持不变。
0	BA	R/W1C	1	启动别名 值描述 1 微控制器的ROM出现在地址0x0。 0 Flash存储器在地址0x0。 对该位的位置写入1可以清零该位。

Register 12: Flash 存储器保护读使能 0 (FMPRE0), 偏移量 0x130 和 0x200

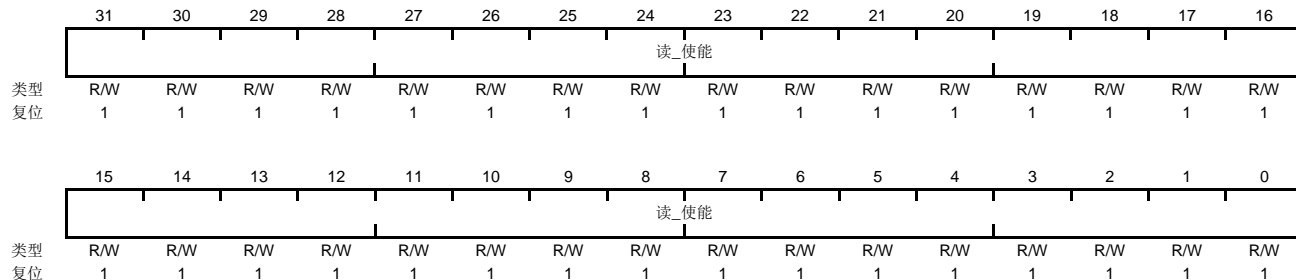
注意: 这个寄存器采用别名, 用来向后兼容。

注意: 偏移量是相对于系统控制基址0x400F.E000的。

这个寄存器存放的是每个2KB区块的只读保护位 (FMPPE_n存放只执行位)。总共多达64KB的Flash存储器由这个寄存器控制。其它的FMPPE_n寄存器 (如果有) 提供对其它64KB块的保护。这个寄存器在上电复位序列加载。对所有执行存储块来说, FMPPE_n 和 FMPPE_n 寄存器在出厂时都被设置为1。这样可以实现一种开放的访问和编程策略。可以通过写特定的寄存器位来更改这个寄存器的位。但是, 这个寄存器是R/W0, 用户只能将保护位从1变为0 (不可以从0变为1)。这种改变不是永久的, 直到寄存器被提交 (保存), 此时位的改变是永久性的。如果一个位从1变为0且没有提交, 那么它可以通过执行一段上电复位序列来恢复。显示的复位值只适用于上电复位, 任何其它类型的复位都不会影响这个寄存器。一旦提交, 恢复该寄存器出厂默认值的唯一方法是执行“恢复锁死的设备”序列, 详情见JTAG一章。更多信息见 "Flash 存储器保护" 一节。

Flash 存储器保护读使能 0 (FMPRE0)

基址 0x400F.E000
 偏移量 0x130 和 0x200
 类型 R/W, 复位 0xFFFF.FFFF



位/域	名称	类型	复位	描述
31:0	读_使能	R/W	0xFFFF.FFFF	Flash 读使能 将2KB的Flash块配置为读或只执行。这些策略可以如表 “Flash保护策略组合” 所示那样组合。 值 0xFFFF.FFFF 描述 位[31:0]中每一个位使能一个2KB的Flash块, 总共达到64KB。

寄存器 13: Flash 存储器保护编程使能 0 (FMPPE0), 偏移量0x134 和 0x400

注意: 这个寄存器采用别名, 用来向后兼容。

注意: 偏移量是相对于系统控制基址0x400F.E000的。

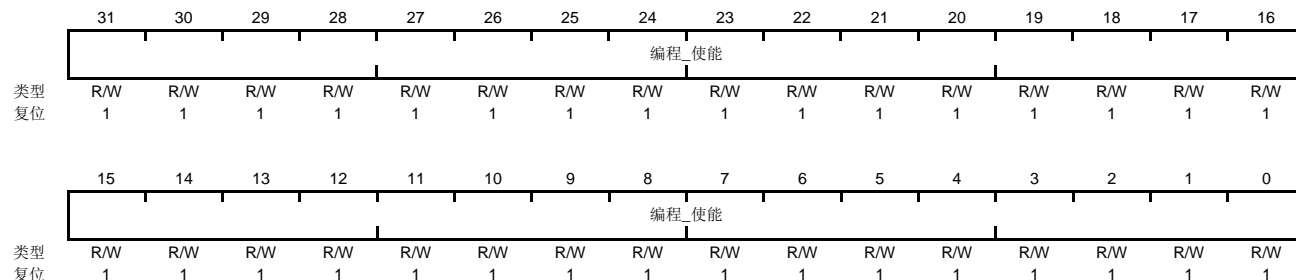
这个寄存器存放的是每个2KB区块的只执行保护位 (**FMPREn存放只读位**)。总共多达64KB的Flash存储器由这个寄存器控制。其它的FMPPEn寄存器 (如果有) 提供对其它64KB块的保护。这个寄存器在上电复位序列加载。对所有执行存储块来说, **FMPREn** 和 **FMPPEn** 寄存器在出厂时都被设置为1。这样可以实现一种开放的访问和编程策略。可以通过写特定的寄存器位来更改这个寄存器的位。但是, 这个寄存器是R/W0, 用户只能将保护位从1变为0 (不可以从0变为1)。这种改变不是永久的, 直到寄存器被提交 (保存), 此时位的改变是永久性的。如果一个位从1变为0且没有提交, 那么它可以通过执行一段上电复位序列来恢复。显示的复位值只适用于上电复位, 任何其它类型的复位都不会影响这个寄存器。一旦提交, 恢复该寄存器出厂默认值的唯一方法是执行“恢复锁死的设备”序列, 详情见JTAG一章。更多信息见 "Flash 存储器保护" 一节。

Flash 存储器保护编程使能 0 (FMPPE0)

基址 0x400F.E000

偏移量 0x134 and 0x400

类型 R/W, 复位 0xFFFF.FFFF



位/域	名称	类型	复位	描述
31:0	编程_使能	R/W	0xFFFF.FFFF	Flash 编程使能 将2KB的Flash块配置为只执行。这些策略可以如表“Flash保护策略组合”所示那样组合。 值 0xFFFF.FFFF 描述 位[31:0]中每一个位使能一个2KB的Flash块, 总共达到64KB。

寄存器 14: 启动配置(BOOTCFG), 偏移量 0x1D0

注意: 偏移量是相对于系统控制基址 0x400F.E000 的。

该寄存器提供了对GPIO管脚的配置来使能ROM引导装载程序, 还提供了一次性写入机制来禁止外部调试器访问设备。复位时, 通过使用在该寄存器中配置的A-H端口GPIO信号, 用户可以选择让内核直接执行ROM引导装载程序或执行Flash存储器中的应用程序。如果EN位置位或者指定的管脚没有规定的极性, 系统控制模块将检测地址0x0000.0004, 查看Flash存储器是否有一个有效的向量。如果地址0x0000.0004的数据是0xFFFF.FFFF, 那么认为Flash存储器仍然没有被编程, 内核执行ROM引导装载程序。出厂时DBG0位(第0位)设为0, DBG1位(第1位)设为1, 这样可以使能外部调试器。清零DBG1位将永久禁止任何外部调试器访问设备, 从下一次设备上电周期开始。NW位(第31位)指示该寄存器仍然没有被提交, 该位通过硬件控制以确保这个寄存器只被提交一次。在提交前, 位只能由1变为0。显示的复位值只适用于上电复位, 任何其它类型的复位都不会影响这个寄存器。一旦提交, 恢复该寄存器出厂默认值的唯一方法是执行“恢复锁死的设备”序列, 详情见JTAG一章。更多信息见“Flash存储器保护”一节。

启动配置 (BOOTCFG)

基址 0x400F.E000

偏移量 0x1D0

类型 R/W, 复位 0xFFFF.FFFE

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NW	保留														
类型	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PORT			PIN		POL	EN	保留							DBG1	DBG0
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

位/域	名称	类型	复位	描述
31	NW	R/W	1	没有被写 该位为1表示这个32位寄存器没有被提交。该位为0说明这个寄存器已经提交且不可以再次提交。
30:16	保留	RO	0x7FFF	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值应该通过1次读-修改-写操作来保持不变。

位/域	名称	类型	复位	描述
15:13	端口	R/W	0x7	<p>启动GPIO端口</p> <p>该域选择复位时使能ROM引导装载程序的GPIO端口。</p> <p>值 描述</p> <p>0x0 Port A</p> <p>0x1 Port B</p> <p>0x2 Port C</p> <p>0x3 Port D</p> <p>0x4 Port E</p> <p>0x5 Port F</p> <p>0x6 Port G</p> <p>0x7 Port H</p>
12:10	PIN	R/W	0x7	<p>启动GPIO管脚</p> <p>该域选择复位时使能ROM引导装载程序的GPIO端口引脚编号。</p> <p>值 描述</p> <p>0x0 Pin 0</p> <p>0x1 Pin 1</p> <p>0x2 Pin 2</p> <p>0x3 Pin 3</p> <p>0x4 Pin 4</p> <p>0x5 Pin 5</p> <p>0x6 Pin 6</p> <p>0x7 Pin 7</p>
9	POL	R/W	0x1	<p>启动GPIO极性</p> <p>该位为1, 选择GPIO管脚的高电平在复位时使能ROM引导装载程序。 该位为0, 选择GPIO管脚的低电平。</p>
8	EN	R/W	0x1	<p>启动GPIO使能</p> <p>清零该位将使能GPIO管脚在复位时用于使能ROM引导装载程序。当该位置位时, 地址0x0000.0004的内容将被检测, 来查看Flash存储器是否被编程。如果内容不是0xFFFF.FFFF, 内核会从Flash存储器执行。如果Flash没有被编程, 内核会从ROM执行。</p>
7:2	保留	RO	0x3F	<p>软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值应该通过1次读-修改-写操作来保持不变。</p>
1	DBG1	R/W	1	<p>调试控制1</p> <p>为使调试可用, DBG1位必须为1并且DBG0必须为0。</p>

位/域	名称	类型	复位	描述
0	DBG0	R/W	0x0	调试控制0 为使调试可用，DBG1位必须为1并且DBG0必须为0。

寄存器 15: 用户寄存器 0 (USER_REG0), 偏移量 0x1E0

注意: 偏移量是相对于系统控制基址 0x400F.E000 的。

该寄存器提供31位用户定义的非易失性数据，且只能提交一次。第31位表示该寄存器已经准备好被提交，该位通过硬件控制以确保这个寄存器只被提交一次。在提交前，位只能由1变为0。显示的复位值只适用于上电复位，任何其它类型的复位都不会影响这个寄存器。该寄存器的一次性写入特性对于保持如通信地址这样每个部件唯一的静态信息是很有用的，否则就需要一个外部EEPROM或者其它非易失性设备。恢复该寄存器出厂默认值的唯一方法是执行“恢复锁死的设备”序列，详情见JTAG一章。

用户寄存器 0 (USER_REG0)

基址 0x400F.E000

偏移量 0x1E0

类型 R/W, 复位 0xFFFF.FFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	NW	数据															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	数据																
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

位/域	名称	类型	复位	描述
31	NW	R/W	1	没有被写 该位为1表示这个32位寄存器没有被提交。该位为0说明这个寄存器已经提交且不可以再次提交。
30:0	数据	R/W	0x7FFF.FFFF	用户数据 包含用户数据值。该域全部初始化为1，且只能被提交一次。

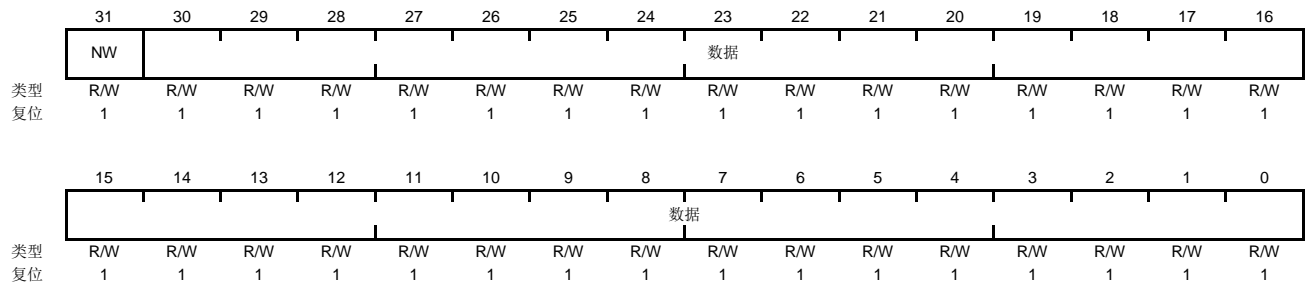
寄存器 16: 用户寄存器 1 (USER_REG1), 偏移量 0x1E4

注意: 偏移量是相对于系统控制基址 0x400F.E000 的。

该寄存器提供31位用户定义的非易失性数据，且只能写入一次。第31位表示该寄存器已经准备好被写入，该位通过硬件控制以确保这个寄存器只被写入一次。该寄存器的一次性写入特性对于保持如通信地址这样每个部件唯一的静态信息是很有用的，否则就需要一个外部EEPROM或者其它非易失性设备。

用户寄存器 1 (USER_REG1)

基址 0x400F.E000
 偏移量 0x1E4
 类型 R/W, 复位 0xFFFF.FFFF



位/域	名称	类型	复位	描述
31	NW	R/W	1	没有被写 该位为1表示这个32位寄存器没有被提交。该位为0说明这个寄存器已经提交且不可以再次提交。
30:0	数据	R/W	0x7FFF.FFFF	用户数据 包含用户数据值。该域全部初始化为1，且只能被提交一次。

寄存器 17: 用户寄存器 2 (USER_REG2), 偏移量 0x1E8

注意: 偏移量是相对于系统控制基址 0x400F.E000 的。

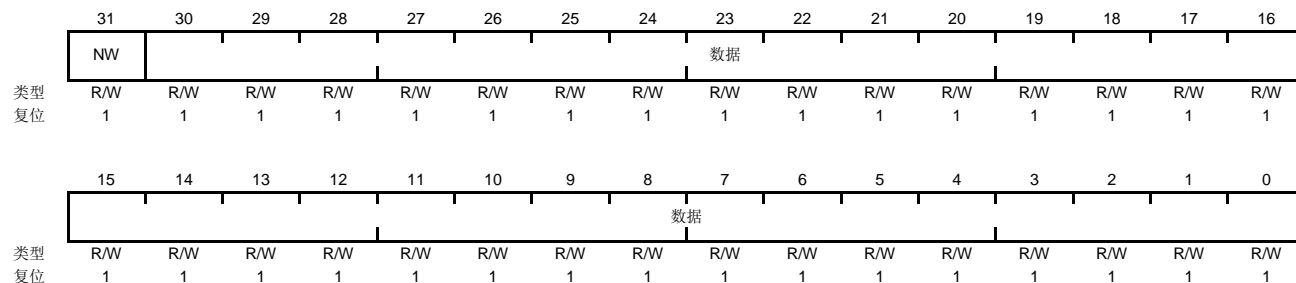
该寄存器提供31位用户定义的非易失性数据，且只能写入一次。第31位表示该寄存器已经准备好被写入，该位通过硬件控制以确保这个寄存器只被写入一次。该寄存器的一次性写入特性对于保持如通信地址这样每个部件唯一的静态信息是很有用的，否则就需要一个外部EEPROM或者其它非易失性设备。

用户寄存器 2 (USER_REG2)

基址 0x400F.E000

偏移量 0x1E8

类型 R/W, 复位 0xFFFF.FFFF



位/域	名称	类型	复位	描述
31	NW	R/W	1	没有被写 该位为1表示这个32位寄存器没有被提交。该位为0说明这个寄存器已经提交且不可以再次提交。
30:0	数据	R/W	0x7FFF.FFFF	用户数据 包含用户数据值。该域全部初始化为1，且只能被提交一次。

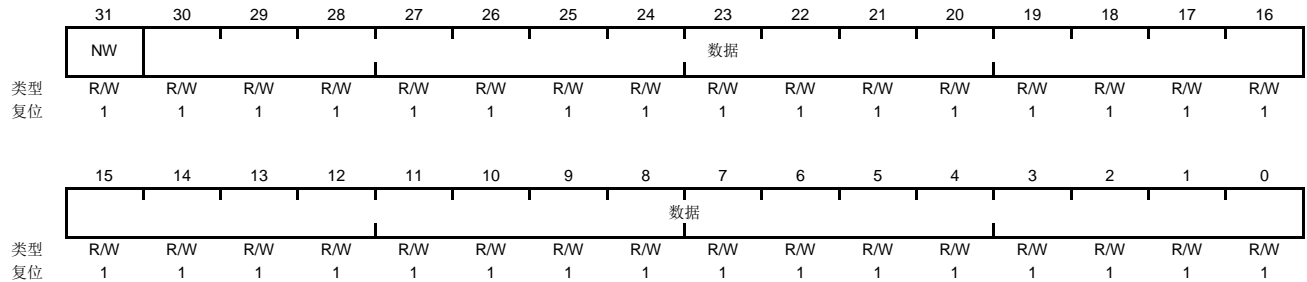
寄存器 18: 用户寄存器 3 (USER_REG3), 偏移量 0x1EC

注意: 偏移量是相对于系统控制基址 0x400F.E000 的。

该寄存器提供31位用户定义的非易失性数据，且只能写入一次。第31位表示该寄存器已经准备好被写入，该位通过硬件控制以确保这个寄存器只被写入一次。该寄存器的一次性写入特性对于保持如通信地址这样每个部件唯一的静态信息是很有用的，否则就需要一个外部EEPROM或者其它非易失性设备。

用户寄存器 3 (USER_REG3)

基址 0x400F.E000
 偏移量 0x1EC
 类型 R/W, 复位 0xFFFF.FFFF



位/域	名称	类型	复位	描述
31	NW	R/W	1	没有被写 该位为1表示这个32位寄存器没有被提交。该位为0说明这个寄存器已经提交且不可以再次提交。
30:0	数据	R/W	0x7FFF.FFFF	用户数据 包含用户数据值。该域全部初始化为1，且只能被提交一次。

寄存器 19: Flash 存储器保护读使能 1 (FMPRE1), 偏移量 0x204

注意: 偏移量是相对于系统控制基址 0x400F.E000 的。

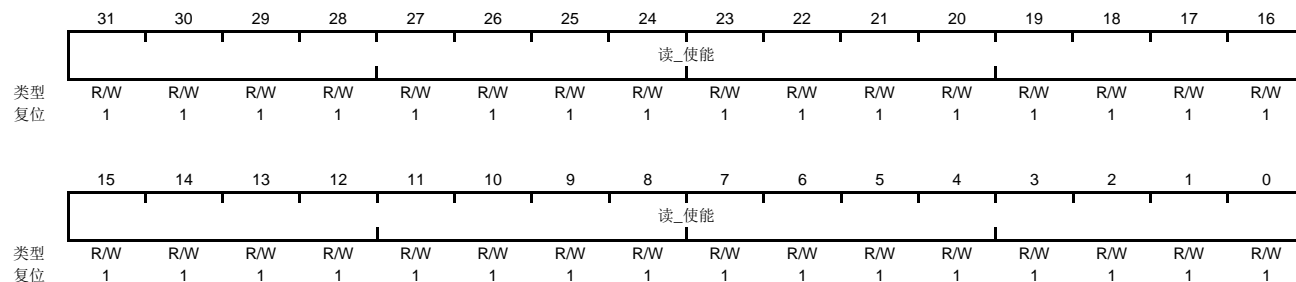
这个寄存器存放的是每个2KB区块的只读保护位 (FMPPEn存放只执行位)。总共多达64KB的Flash存储器由这个寄存器控制。其它的FMPREN寄存器 (如果有) 提供对其它64KB块的保护。这个寄存器在上电复位序列加载。对所有执行存储块来说, **FMPREN** 和 **FMPPEn** 寄存器在出厂时都被设置为1。这样可以实现一种开放的访问和编程策略。可以通过写特定的寄存器位来更改这个寄存器的位。但是, 这个寄存器是R/W0, 用户只能将保护位从1变为0 (不可以从0变为1)。这种改变不是永久的, 直到寄存器被提交 (保存), 此时位的改变是永久性的。如果一个位从1变为0且没有提交, 那么它可以通过执行一段上电复位序列来恢复。显示的复位值只适用于上电复位, 任何其它类型的复位都不会影响这个寄存器。一旦提交, 恢复该寄存器出厂默认值的唯一方法是执行“恢复锁死的设备”序列, 详情见JTAG一章。如果Flash存储器空间小于64KB, 读取该寄存器通常为0, 但是软件不能依赖这些位作为0。更多信息见 "Flash 存储器保护" 一节。

Flash 存储器保护读使能 1 (FMPRE1)

基址 0x400F.E000

偏移量 0x204

类型 R/W, 复位 0xFFFF.FFFF



位/域	名称	类型	复位	描述
31:0	读_使能	R/W	0xFFFF.FFFF	Flash 读使能

将2KB的Flash块配置为读或只执行。这些策略可以如表“Flash保护策略组合”所示那样组合。

值	描述
0xFFFF.FFFF	位[31:0]中每一个位使能一个2KB的Flash块, Flash存储器的范围从65到128KB。

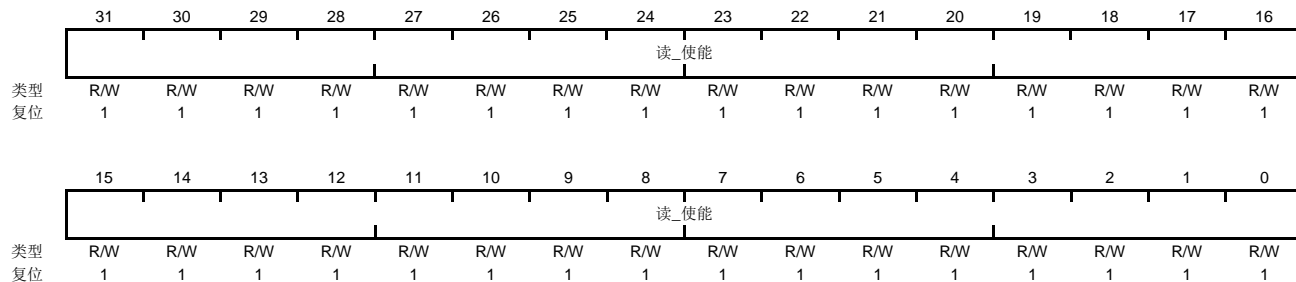
寄存器 20: Flash 存储器保护读使能 2 (FMPRE2), 偏移量 0x208

注意: 偏移量是相对于系统控制基址 0x400F.E000 的。

这个寄存器存放的是每个2KB区块的只读保护位 (FMPPE_n存放只执行位)。总共多达64KB的Flash存储器由这个寄存器控制。其它的FMPPE_n寄存器 (如果有) 提供对其它64KB块的保护。这个寄存器在上电复位序列加载。对所有执行存储块来说, FMPPE_n 和 FMPPE_n 寄存器在出厂时都被设置为1。这样可以实现一种开放的访问和编程策略。可以通过写特定的寄存器位来更改这个寄存器的位。但是, 这个寄存器是R/W0, 用户只能将保护位从1变为0 (不可以从0变为1)。这种改变不是永久的, 直到寄存器被提交 (保存), 此时位的改变是永久性的。如果一个位从1变为0且没有提交, 那么它可以通过执行一段上电复位序列来恢复。显示的复位值只适用于上电复位, 任何其它类型的复位都不会影响这个寄存器。一旦提交, 恢复该寄存器出厂默认值的唯一方法是执行“恢复锁死的设备”序列, 详情见JTAG一章。如果Flash存储器空间小于128KB, 读取该寄存器通常为0, 但是软件不能依赖这些位作为0。更多信息见 "Flash 存储器保护" 一节。

Flash 存储器保护读使能 2 (FMPRE2)

基址 0x400F.E000
 偏移量 0x208
 类型 R/W, 复位 0xFFFF.FFFF



位/域	名称	类型	复位	描述
-----	----	----	----	----

31:0	读_使能	R/W	0xFFFF.FFFF	Flash 读使能
------	------	-----	-------------	-----------

将2KB的Flash块配置为读或只执行。这些策略可以如表 “Flash保护策略组合” 所示那样组合。

值	描述
0xFFFF.FFFF	位[31:0]中每一个位使能一个2KB的Flash块, Flash存储器的范围从129到192KB。

寄存器 21: Flash 存储器保护读使能 3 (FMPRE3), 偏移量 0x20C

注意: 偏移量是相对于系统控制基址 0x400F.E000 的。

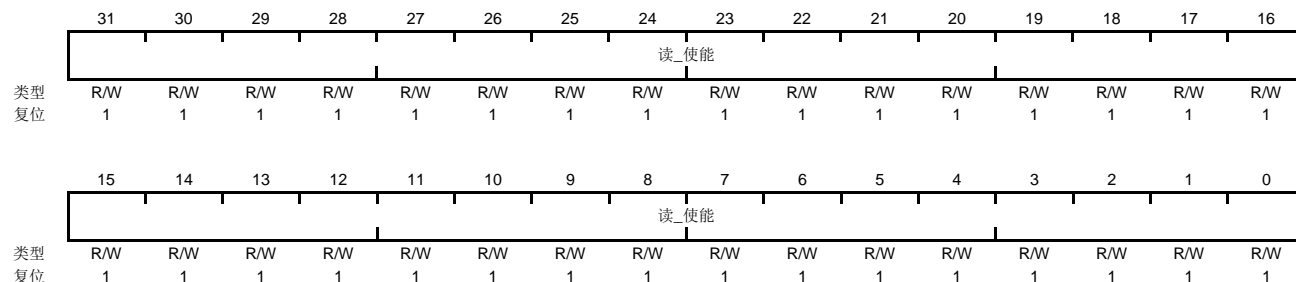
这个寄存器存放的是每个2KB区块的只读保护位 (FMPPE_n存放只执行位)。总共多达64KB的Flash存储器由这个寄存器控制。其它的FMPPE_n寄存器 (如果有) 提供对其它64KB块的保护。这个寄存器在上电复位序列加载。对所有执行存储块来说, FMPPE_n 和 FMPPE_n 寄存器在出厂时都被设置为1。这样可以实现一种开放的访问和编程策略。可以通过写特定的寄存器位来更改这个寄存器的位。但是, 这个寄存器是R/W0, 用户只能将保护位从1变为0 (不可以从0变为1)。这种改变不是永久的, 直到寄存器被提交 (保存), 此时位的改变是永久性的。如果一个位从1变为0且没有提交, 那么它可以通过执行一段上电复位序列来恢复。显示的复位值只适用于上电复位, 任何其它类型的复位都不会影响这个寄存器。一旦提交, 恢复该寄存器出厂默认值的唯一方法是执行“恢复锁死的设备”序列, 详情见JTAG一章。如果Flash存储器空间小于192KB, 读取该寄存器通常为0, 但是软件不能依赖这些位作为0。更多信息见 "Flash 存储器保护" 一节。

Flash 存储器保护读使能 3 (FMPRE3)

基址 0x400F.E000

偏移量 0x20C

类型 R/W, 复位 0xFFFF.FFFF



位/域	名称	类型	复位	描述
31:0	读_使能	R/W	0xFFFF.FFFF	Flash 读使能

将2KB的Flash块配置为读或只执行。这些策略可以如表“Flash保护策略组合”所示那样组合。

值	描述
0xFFFF.FFFF	位[31:0]中每一个位使能一个2KB的Flash块, Flash存储器的范围从193到256KB。

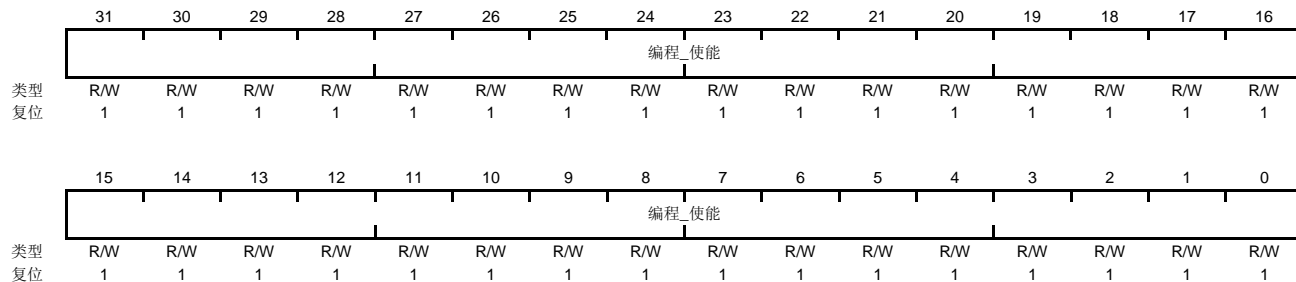
寄存器 22: Flash 存储器保护编程使能 1 (FMPPE1), 偏移量0x404

注意: 偏移量是相对于系统控制基址 0x400F.E000 的。

这个寄存器存放的是每个2KB区块的只执行保护位 (**FMPREn存放只读位**)。总共多达64KB的Flash存储器由这个寄存器控制。其它的FMPPEn寄存器 (如果有) 提供对其它64KB块的保护。这个寄存器在上电复位序列加载。对所有执行存储块来说, **FMPREn** 和 **FMPPEn** 寄存器在出厂时都被设置为1。这样可以实现一种开放的访问和编程策略。可以通过写特定的寄存器位来更改这个寄存器的位。但是, 这个寄存器是R/W0, 用户只能将保护位从1变为0 (不可以从0变为1)。这种改变不是永久的, 直到寄存器被提交 (保存), 此时位的改变是永久性的。如果一个位从1变为0且没有提交, 那么它可以通过执行一段上电复位序列来恢复。显示的复位值只适用于上电复位, 任何其它类型的复位都不会影响这个寄存器。一旦提交, 恢复该寄存器出厂默认值的唯一方法是执行“恢复锁死的设备”序列, 详情见JTAG一章。如果Flash存储器空间小于64KB, 读取该寄存器通常为0, 但是软件不能依赖这些位作为0。更多信息见 "Flash 存储器保护" 一节。

Flash 存储器保护编程使能 1 (FMPPE1)

基址 0x400F.E000
 偏移量 0x404
 类型 R/W, 复位 0xFFFF.FFFF



位/域	名称	类型	复位	描述
31:0	编程_使能	R/W	0xFFFF.FFFF	Flash 编程使能 将2KB的Flash块配置为只执行。这些策略可以如表“Flash保护策略组合”所示那样组合。 值 0xFFFF.FFFF 描述 位[31:0]中每一个位使能一个2KB的Flash块, Flash存储器的范围从65到128KB。

寄存器 23: Flash 存储器保护编程使能 2 (FMPPE2), 偏移量 0x408

注意: 偏移量是相对于系统控制基址 0x400F.E000 的。

这个寄存器存放的是每个 2KB 区块的只执行保护位 (**FMPREn 存放只读位**)。总共多达 64KB 的 Flash 存储器由这个寄存器控制。其它的 FMPPEn 寄存器 (如果有) 提供对其它 64KB 块的保护。这个寄存器在上电复位序列加载。对所有执行存储块来说, FMPREN 和 FMPPEn 寄存器在出厂时都被设置为 1。这样可以实现一种开放的访问和编程策略。可以通过写特定的寄存器位来更改这个寄存器的位。但是, 这个寄存器是 R/W0, 用户只能将保护位从 1 变为 0 (不可以从 0 变为 1)。这种改变不是永久的, 直到寄存器被提交 (保存), 此时位的改变是永久性的。如果一个位从 1 变为 0 且没有提交, 那么它可以通过执行一段上电复位序列来恢复。显示的复位值只适用于上电复位, 任何其它类型的复位都不会影响这个寄存器。一旦提交, 恢复该寄存器出厂默认值的唯一方法是执行“恢复锁死的设备”序列, 详情见 JTAG 一章。如果 Flash 存储器空间小于 128KB, 读取该寄存器通常为 0, 但是软件不能依赖这些位作为 0。更多信息见 "Flash 存储器保护" 一节。

Flash 存储器保护编程使能 2 (FMPPE2)

基址 0x400F.E000

偏移量 0x408

类型 R/W, 复位 0xFFFF.FFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	编程_使能															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	编程_使能															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:0	编程_使能	R/W	0xFFFF.FFFF	Flash 编程使能

将 2KB 的 Flash 块配置为只执行。这些策略可以如表“Flash 保护策略组合”所示那样组合。

值	描述
0xFFFF.FFFF	位 [31:0] 中每一个位使能一个 2KB 的 Flash 块, Flash 存储器的范围从 129 到 192KB。

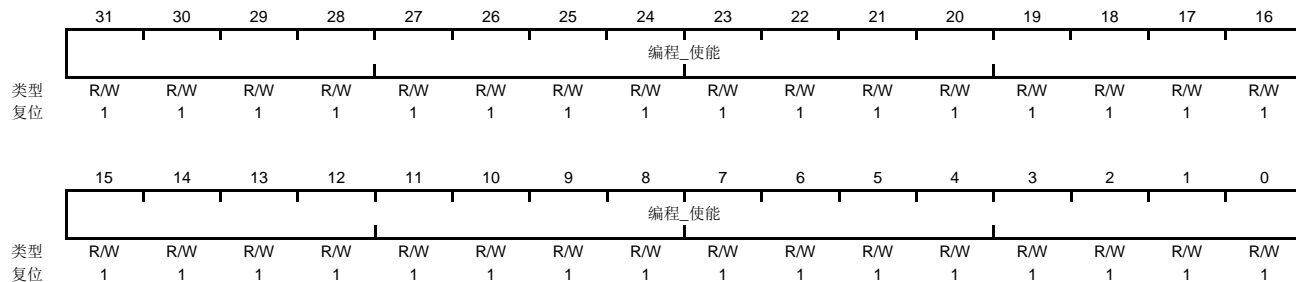
寄存器 24: Flash 存储器保护编程使能 3 (FMPPE3), 偏移量0x40C

注意: 偏移量是相对于系统控制基址 0x400F.E000 的。

这个寄存器存放的是每个2KB区块的只执行保护位 (FMPREn存放只读位)。总共多达64KB的Flash存储器由这个寄存器控制。其它的FMPPEn寄存器 (如果有) 提供对其它64KB块的保护。这个寄存器在上电复位序列加载。对所有执行存储块来说, **FMPREN** 和 **FMPPEn** 寄存器在出厂时都被设置为1。这样可以实现一种开放的访问和编程策略。可以通过写特定的寄存器位来更改这个寄存器的位。但是, 这个寄存器是R/W0, 用户只能将保护位从1变为0 (不可以从0变为1)。这种改变不是永久的, 直到寄存器被提交 (保存), 此时位的改变是永久性的。如果一个位从1变为0且没有提交, 那么它可以通过执行一段上电复位序列来恢复。显示的复位值只适用于上电复位, 任何其它类型的复位都不会影响这个寄存器。一旦提交, 恢复该寄存器出厂默认值的唯一方法是执行“恢复锁死的设备”序列, 详情见JTAG一章。如果Flash存储器空间小于192KB, 读取该寄存器通常为0, 但是软件不能依赖这些位作为0。更多信息见 "Flash 存储器保护" 一节。

Flash 存储器保护编程使能 3 (FMPPE3)

基址 0x400F.E000
 偏移量 0x40C
 类型 R/W, 复位 0xFFFF.FFFF



位/域	名称	类型	复位	描述
31:0	编程_使能	R/W	0xFFFF.FFFF	Flash 编程使能

将2KB的Flash块配置为只执行。这些策略可以如表“Flash保护策略组合”所示那样组合。

值	描述
0xFFFF.FFFF	位[31:0]中每一个位使能一个2KB的Flash块, Flash存储器的范围从193到256KB。

北京锐鑫同创公司相关信息

技术支持

如果您对文档有所疑问，您可以在办公时间（星期一至星期五上午 8:30~11:50；下午 1:30~5:30）拨打技术支持电话或 E-mail 联系。

北京锐鑫同创是 TI 第三方合作伙伴，专注于 TI Stellaris M3 产品的市场推广、方案设计和技术服务，同时提供开发板、仿真器、编程器等开发工具，公司以“把握市场脉搏，专注技术创新，提供诚信服务，实现共赢发展！”为核心价值理念，为客户提供实时、高效的技术和服务。

电话：010-82418301

传真：010-82418302

Email: support@realsense.com.cn

网站: www.realsense.com.cn

技术论坛: www.hellom3.cn