

## 前言

本编程手册介绍了 STM32F405xx、STM32F407xx、STM32F415xx 和 STM32F417xx 微控制器 Flash 的编程方法。为方便起见，本文档将上述产品统称为 STM32F40x 和 STM32F41x（除非另行声明）。

STM32F40x 和 STM32F41x 嵌入式 Flash 可采用在线编程或在应用中编程两种方式。

**在线编程 (ICP)** 方式适用于更新 Flash 的所有内容，更新时使用 JTAG、SWD 协议或自举程序将用户应用程序加载到微控制器。ICP 可实现快速而高效的设计迭代，并且避免了不必要的器件封装处理或插接。

与 ICP 方法相比，**在应用中编程 (IAP)** 可通过微控制器支持的任何通信接口（I/O、USB、CAN、UART、I<sup>2</sup>C 和 SPI 等）将编程数据下载到存储器。使用 IAP 方式时，可以在应用程序运行期间对 Flash 重新编程。但是，部分应用程序必须事先通过 ICP 方式编程到 Flash。

Flash 接口根据 AHB 协议实施指令访问和数据访问。它将实施可加快 CPU 代码执行速度的预取缓冲器，以及执行 Flash 操作（编程/擦除）所必需的逻辑。编程/擦除操作可在整个产品电压范围内执行。读写保护和选项字节的设置也通过 FLASH 接口来操作。

# 目录

<b>1</b>	<b>Flash 接口</b>	<b>6</b>
1.1	简介	6
1.2	主要特性	6
1.3	Flash	7
1.4	读接口	8
1.4.1	CPU 时钟频率与 Flash 读取时间之间的关系	8
1.4.2	自适应实时存储器加速器 (ART Accelerator™)	9
1.5	擦除和编程操作	11
1.5.1	Flash 控制寄存器解锁	11
1.5.2	编程/擦除并行位数	11
1.5.3	擦除	12
1.5.4	编程	12
1.5.5	中断	13
1.6	选项字节	14
1.6.1	关于用户选项字节的说明	14
1.6.2	用户选项字节编程	15
1.6.3	读保护 (RDP)	15
1.6.4	写保护	16
1.7	一次性可编程字节	17
1.8	Flash 接口寄存器	18
1.8.1	Flash 访问控制寄存器 (FLASH_ACR)	18
1.8.2	Flash 密钥寄存器 (FLASH_KEYR)	19
1.8.3	Flash 选项字节密钥寄存器 (FLASH_OPTKEYR)	19
1.8.4	Flash 状态寄存器 (FLASH_SR)	20
1.8.5	Flash 控制寄存器 (FLASH_CR)	21
1.8.6	Flash 选项控制寄存器 (FLASH_OPTCR)	22
1.8.7	Flash 接口寄存器映射	23
<b>2</b>	<b>版本历史</b>	<b>24</b>

表格索引

表 1. Flash 模块构成 ..... 7

表 2. CPU 时钟 (HCLK) 频率对应的等待周期数 (VOS = '1') ..... 8

表 3. 编程/擦除并行位数 ..... 12

表 4. 选项字节构成 ..... 14

表 5. 关于选项字节的说明 ..... 14

表 6. OTP 区域构成 ..... 17

表 7. Flash 寄存器映射与复位值 ..... 23

表 8. 文档版本历史 ..... 24

# 图片索引

图 1.        系统架构内的 Flash 接口连接 ..... 6

图 2.        32 位连续指令的执行 ..... 10

图 3.        RDP 级别 ..... 16



## 词汇表

本节简要介绍本文档中所用首字母缩略词和缩写词的定义：

- CPU 内核集成了两个调试端口：
  - JTAG 调试端口 (JTAG-DP) 提供基于联合测试工作组 (JTAG) 协议的 5 引脚标准接口。
  - SWD 调试端口 (SWD-DP) 提供基于串行线调试 (SWD) 协议的 2 引脚（时钟和数据）接口。

有关 JTAG 和 SWD 协议的信息，请参见《Cortex-M4F 技术参考手册》
- 字：32 位数据/指令。
- 半字：16 位数据/指令。
- 字节：8 位数据。
- 双字：64 位数据。
- IAP（在应用中编程）：IAP 是指可以在用户程序运行期间对微控制器的 Flash 进行重新编程。
- ICP（在线编程）：ICP 是指可以在器件安装于用户应用电路板上时使用 JTAG 协议、SWD 协议或自举程序对微控制器的 Flash 进行编程。
- I-Code：此总线用于将 CPU 内核的指令总线连接到 Flash 指令接口。通过此总线可执行预取操作。
- D-Code：此总线用于将 CPU 的 D-Code 总线（文字池数据加载和调试访问）连接到 Flash 数据接口。
- 选项字节：存储于 Flash 中的产品配置位。
- OBL：选项字节加载器。
- AHB：高级高性能总线。
- CPU：指 Cortex-M4F 内核。

# 1 Flash 接口

## 1.1 简介

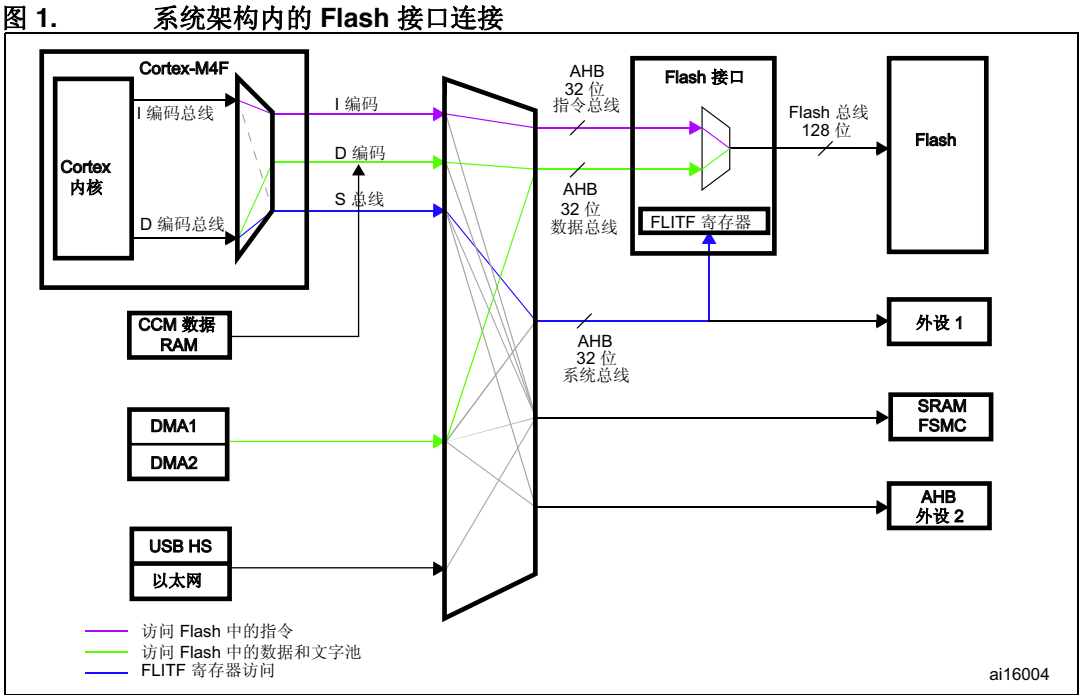
Flash 接口可管理 CPU 对 1 MB（64 Kb × 128 位）Flash 进行的 AHB I-Code 和 D-Code 访问。该接口可针对 Flash 执行擦除和编程操作，并实施读写保护机制。

Flash 接口通过指令预取和缓存机制加速代码执行。

## 1.2 主要特性

- Flash 读操作
- Flash 编程/擦除操作
- 读/写保护
- I-Code 上的预取操作
- I-Code 上的 64 个缓存行（128 位）
- D-Code 上的 8 个缓存行（128 位）

图 1 所示为系统架构内的 Flash 接口连接。



## 1.3 Flash

Flash 具有以下主要特性：

- 高达 1 MB 容量
- 128 位宽数据读取
- 字节、半字、字和双字数据写入
- 扇区擦除与批量擦除
- 存储器构成

Flash 结构如下：

– 主存储器块，含 4 个 16 KB 扇区、1 个 64 KB 扇区 和 7 个 128 KB 扇区。

– 系统存储器，器件在系统存储器自举模式下从该存储器自举。

此区域为意法半导体预留，其中包含自举程序，用以通过以下接口之一对 Flash 进行重新编程：USART1、USART3、CAN2、USB OTG FS 设备模式（DFU：设备固件升级）。自举程序由 ST 在器件制造期间编写，用于防止误写/误擦除操作。

– 512 OTP（一次性可编程）字节，用于存储用户数据。

OTP 区域包含 16 个附加字节，用于锁定相应的 OTP 数据块。

– 选项字节：读写保护、BOR 级别、软件 / 硬件看门狗以及器件在待机或停机模式下的复位。

- 低功耗模式（有关详细信息，请参见参考手册的“电源控制 (PWR)”部分）

表 1. Flash 模块构成

块	名称	块基址	大小
主存储器	扇区 0	0x0800 0000 - 0x0800 3FFF	16 KB
	扇区 1	0x0800 4000 - 0x0800 7FFF	16 KB
	扇区 2	0x0800 8000 - 0x0800 BFFF	16 KB
	扇区 3	0x0800 C000 - 0x0800 FFFF	16 KB
	扇区 4	0x0801 0000 - 0x0801 FFFF	64 KB
	扇区 5	0x0802 0000 - 0x0803 FFFF	128 KB
	扇区 6	0x0804 0000 - 0x0805 FFFF	128 KB
	.	.	.
	扇区 11	0x080E 0000 - 0x080F FFFF	128 KB
系统存储器		0x1FFF 0000 - 0x1FFF 77FF	30 KB
OTP 区域		0x1FFF 7800 - 0x1FFF 7A0F	528 KB
选项字节		0x1FFF C000 - 0x1FFF C00F	16 KB

# 1.4 读接口

## 1.4.1 CPU 时钟频率与 Flash 读取时间之间的关系

为了准确读取 Flash 数据，必须根据 CPU 时钟 (HCLK) 频率和器件电源电压在 Flash 存取控制寄存器 (FLASH\_ACR) 中正确地编程等待周期数 (LATENCY)。表 2 所示为等待周期与 CPU 时钟频率之间的对应关系。

注：VOS = '0' 时， $f_{HCLK}$  最大值 = 144 MHz。

表 2. CPU 时钟 (HCLK) 频率对应的等待周期数 (VOS = '1')

等待周期 (WS) (LATENCY)	HCLK (MHz)			
	电压范围 2.7 V - 3.6 V	电压范围 2.4 V - 2.7 V	电压范围 2.1 V - 2.4 V	电压范围 1.8 V - 2.1 V <sup>(1)</sup>
0 WS (1 个 CPU 周期)	0 < HCLK ≤ 30	0 < HCLK ≤ 24	0 < HCLK ≤ 18	0 < HCLK ≤ 16
1 WS (2 个 CPU 周期)	30 < HCLK ≤ 60	24 < HCLK ≤ 48	18 < HCLK ≤ 36	16 < HCLK ≤ 32
2 WS (3 个 CPU 周期)	60 < HCLK ≤ 90	48 < HCLK ≤ 72	36 < HCLK ≤ 54	32 < HCLK ≤ 48
3 WS (4 个 CPU 周期)	90 < HCLK ≤ 120	72 < HCLK ≤ 96	54 < HCLK ≤ 72	48 < HCLK ≤ 64
4 WS (5 个 CPU 周期)	120 < HCLK ≤ 150	96 < HCLK ≤ 120	72 < HCLK ≤ 90	64 < HCLK ≤ 80
5 WS (6 个 CPU 周期)	150 < HCLK ≤ 168	120 < HCLK ≤ 144	90 < HCLK ≤ 108	80 < HCLK ≤ 96
6 WS (7 个 CPU 周期)		144 < HCLK ≤ 168	108 < HCLK ≤ 120	96 < HCLK ≤ 112
7 WS (8 个 CPU 周期)			120 < HCLK ≤ 138	112 < HCLK ≤ 128

1. 如果 PDR\_ON 设为 V<sub>SS</sub>，则该值可降至 1.7 V。

复位后，CPU 时钟频率为 16 MHz，FLASH\_ACR 寄存器中的等待周期 (WS) 为 0 个。

强烈建议通过以下软件序列来调整在访问 Flash 时所需的等待周期数和 CPU 频率。

### 需要提高 CPU 频率时的操作步骤

1. 将新的等待周期数编程到 FLASH\_ACR 寄存器中的 LATENCY 位。
2. 通过读取 FLASH\_ACR 寄存器，检查在访问 Flash 时是否采用了新的等待周期数。
3. 通过改写 RCC\_CFGR 寄存器中的 SW 位来修改 CPU 时钟源。
4. 如有需要，可通过改写 RCC\_CFGR 中的 HPRE 位来修改 CPU 时钟预分频器。
5. 通过读取 RCC\_CFGR 寄存器中相应的时钟源状态 (SWS 位) 和/或 AHB 预分频值 (HPRE 位)，检查是否采用了新的 CPU 时钟源和/或新的 CPU 时钟预分频值。

### 需要降低 CPU 频率时的操作步骤

1. 通过改写 RCC\_CFGR 寄存器中的 SW 位来修改 CPU 时钟源。
2. 如有需要，可通过改写 RCC\_CFGR 中的 HPRE 位来修改 CPU 时钟预分频器。
3. 通过读取 RCC\_CFGR 寄存器中相应的时钟源状态 (SWS 位) 和/或 AHB 预分频值 (HPRE 位)，检查是否采用了新的 CPU 时钟源和/或新的 CPU 时钟预分频值。
4. 将新的等待周期数编程到 FLASH\_ACR 中的 LATENCY 位。
5. 通过读取 FLASH\_ACR 寄存器，检查在访问 Flash 时是否采用了新的等待周期数。



注: CPU 时钟配置或等待周期 (WS) 配置的更改不会立即生效。为了确保当前的 CPU 时钟频率即为所配置的频率, 用户可检查 AHB 预分频系数和时钟源状态值。为了确保所编程的 WS 数生效, 可读取 FLASH\_ACR 寄存器的内容来确认。

### 1.4.2 自适应实时存储器加速器 (ART Accelerator™)

专有的自适应实时 (ART) 存储器加速器面向 STM32 工业标准 ARM® Cortex™-M4F 处理器进行了优化。该加速器很好地体现了 ARM Cortex M4F 的固有性能优势, 克服了通常条件下, 高速的处理器在运行中需要经常等待 FLASH 读取的情况。

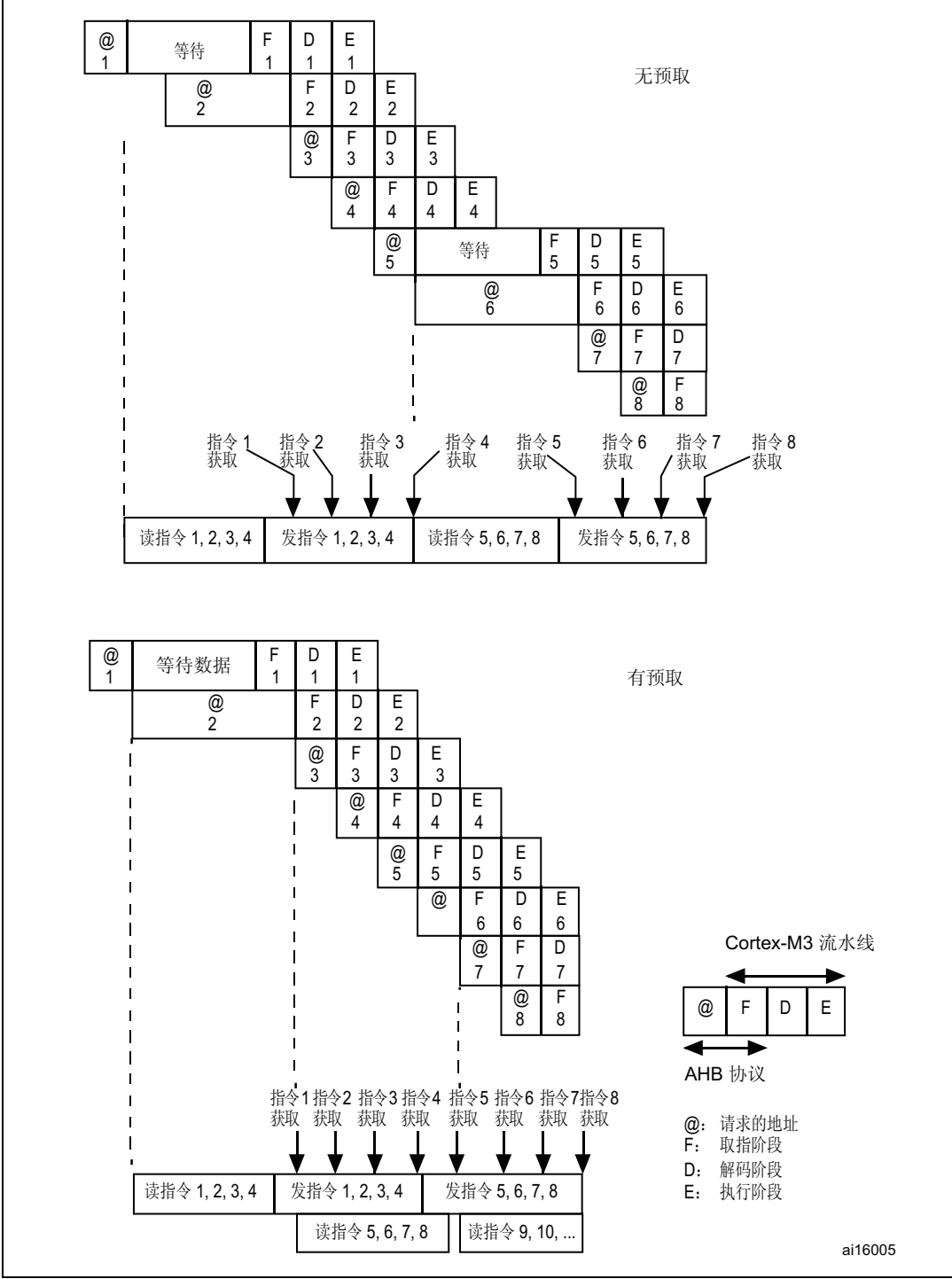
为了发挥处理器的全部性能, 该加速器将实施指令预取队列和分支缓存, 从而提高了 128 位 Flash 的程序执行速度。根据 CoreMark 基准测试, 凭借 ART 加速器所获得的性能相当于 Flash 在 CPU 频率高达 168 MHz 时以 0 个等待周期执行程序。

#### 指令预取

每个 Flash 读操作可读取 128 位, 可以是 4 行 32 位指令, 也可以是 8 行 16 位指令, 具体取决于启动的程序。因此对于连续代码, 执行前一个读指令行至少需要 4 个 CPU 周期。在 CPU 请求当前指令行时, 可使用 I-Code 总线的预取操作读取 Flash 中的下一个连续指令行。可将 FLASH\_ACR 寄存器中的 PRFTEN 位置 1, 来使能预取功能。当访问 Flash 至少需要一个等待周期时, 此功能非常有用。

图 2 所示为需要 3 WS 访问 Flash 时连续 32 位指令的执行过程, 图中分别介绍了使用和不使用预取操作两种情况。

图 2. 32 位连续指令的执行



处理非连续代码（有分支）时，指令可能并不存在于当前使用的或预取的指令行中。这种情况下（数据缺失），增加的周期数至少等于等待周期数。

## 指令缓存存储器

为了限制因指令跳转而产生的时间损耗，可将 64 行 128 位的指令保存到指令缓存存储器中。可将 **FLASH\_ACR** 寄存器中的指令缓存使能 (**ICEN**) 位置 1，来使能这一特性。每当出现数据缺失（即请求的数据未存在于当前使用的指令行、预取指令行或指令缓存存储器中）时，系统会将读取的行复制到指令缓存存储器中。如果 **CPU** 请求指令缓存内存中的某些数据，则无需插入任何延时即可立即获取。所有指令缓存存储器行存满数据后，可采用 **LRU**（最近最少使用）策略确定指令缓存存储器中待替换的指令行。此特性非常适用于包含循环的代码。

## 数据管理

在 **CPU** 流水线执行阶段，将通过 **D-Code** 总线获取 **Flash** 中的文字池。因此，直到提供请求的文字池后，**CPU** 流水线才会继续执行。为了限制因文字池而产生的时间损耗，通过 **AHB** 数据总线 **D-Code** 进行的访问优先于通过 **AHB** 指令总线 **I-Code** 进行的访问。

如果频繁使用某些文字池，可将 **FLASH\_ACR** 寄存器中的数据缓存使能 (**DCEN**) 位置 1，来使能数据缓存存储器。此特性的工作原理与指令缓存存储器类似，但保留的数据大小限制在 8 行（128 位）以内。

注： 用户配置扇区中的数据无法缓存。

## 1.5 擦除和编程操作

执行任何 **Flash** 编程操作（擦除或编程）时，**CPU** 时钟频率 (**HCLK**) 至少是 1 MHz。如果在 **Flash** 操作期间发生器件复位，则无法保证 **Flash** 中的内容不受影响。

在对 **Flash** 执行写/擦除操作期间，任何读取 **Flash** 的尝试都会导致总线阻塞。只有在完成编程操作后，才能正确处理读操作。这意味着，写/擦除操作进行期间不能从 **FLASH** 中执行代码或数据获取操作。

### 1.5.1 Flash 控制寄存器解锁

复位后，**Flash** 控制寄存器 (**FLASH\_CR**) 不允许执行写操作，以防 **Flash** 因电气干扰等原因出现意外操作。此寄存器的解锁顺序如下：

1. 在 **Flash** 密钥寄存器 (**FLASH\_KEYR**) 中写入 **KEY1 = 0x45670123**
2. 在 **Flash** 密钥寄存器 (**FLASH\_KEYR**) 中写入 **KEY2 = 0xCDEF89AB**

如果顺序出现任何错误，将返回总线错误并锁定 **FLASH\_CR** 寄存器，直到下一次复位。

将 **FLASH\_CR** 寄存器中的 **LOCK** 位置为 1 后，可通过软件再次锁定 **FLASH\_CR** 寄存器。

注： 当 **FLASH\_SR** 寄存器中的 **BSY** 位置为 1 后，将不能在写模式下访问 **FLASH\_CR** 寄存器。**BSY** 位置为 1 后，对该寄存器的任何写操作尝试都会导致 **AHB** 总线阻塞，直到 **BSY** 位清零。

### 1.5.2 编程/擦除并行位数

并行位数是指编程或擦除操作期间可在单步中编程为 0 的最大位数。最大编程/擦除并行位数受限于电源电压以及是否使用外部  $V_{PP}$  电源。执行编程/擦除操作前，必须在 FLASH\_CR 寄存器的并行位数位 PSIZE 中写入正确的值。表 3 提供了正确的 PSIZE 值。

表 3. 编程/擦除并行位数

	电压范围 2.7 - 3.6 V (使用外部 $V_{PP}$ )	电压范围 2.7 - 3.6 V	电压范围 2.4 - 2.7 V	电压范围 2.1 - 2.4 V	电压范围 1.8 V - 2.1 V <sup>(1)</sup>
最大并行位数	x64	x32	x16		x8
PSIZE(1:0)	11	10	01		00

1. 如果 STM32F20xx 器件上的 IRROFF 设为 VDD，当该器件在降低的温度范围内工作时，该值可降至 1.65 V。

**注：**如果在编程并行位数/电压范围设置不一致的情况下启动任何编程或擦除操作，可能会导致出现意外结果。即使后续的读操作指示逻辑值已有效写入存储器中，该值也无法保留。

要使用  $V_{PP}$ ，必须在  $V_{PP}$  引脚施加一个外部高电压电源（8 V 到 9 V 之间）。该外部电源必须在直流电耗超过 10 mA 时也能维持该电压范围。建议仅在工厂生产线上进行初始编程时使用  $V_{PP}$ 。 $V_{PP}$  电源的供电时间不得超过一小时，否则 Flash 可能会损坏。

### 1.5.3 擦除

Flash 擦除操作可针对扇区或整个 Flash（批量擦除）执行。执行批量擦除时，不会影响 OTP 扇区或配置扇区。

#### 扇区擦除

扇区擦除的具体步骤如下：

1. 检查 FLASH\_SR 寄存器中的 BSY 位，以确认当前未执行任何 Flash 操作
2. 在 FLASH\_CR 寄存器中将 SER 位置 1 并选择要擦除的扇区 (SNB)（主存储器块中的 12 个扇区之一）
3. 将 FLASH\_CR 寄存器中的 STRT 位置 1
4. 等待 BSY 位清零

#### 批量擦除

要执行批量擦除，建议采用以下步骤：

1. 检查 FLASH\_SR 寄存器中的 BSY 位，以确认当前未执行任何 Flash 操作
2. 将 FLASH\_CR 寄存器中的 MER 位置 1
3. 将 FLASH\_CR 寄存器中的 STRT 位置 1
4. 等待 BSY 位清零

## 1.5.4 编程

### 标准编程

Flash 编程顺序如下：

1. 检查 FLASH\_SR 中的 BSY 位，以确认当前未执行任何主要 Flash 操作。
2. 将 FLASH\_CR 寄存器中的 PG 位置 1。
3. 针对所需存储器地址（主存储器块或 OTP 区域内）执行数据写入操作：
  - 并行位数为 x8 时按字节访问
  - 并行位数为 x16 时按半字访问
  - 并行位数为 x32 时按字访问
  - 并行位数为 x64 时按双字访问
4. 等待 BSY 位清零。

### 编程错误

不允许针对 Flash 执行跨越 128 位行界限的数据编程操作。如果出现这种情况，写操作将不会执行，并且 FLASH\_SR 寄存器中的编程对齐错误标志位 (PGAERR) 将置 1。

写访问类型（字节、半字、字或双字）必须与所选并行位数类型（x8、x16、x32 或 x64）相符。否则，写操作将不会执行，并且 FLASH\_SR 寄存器中的编程并行位数错误标志位 (PGPERR) 将置 1。

如果未遵循标准的编程顺序（例如，在 PG 位未置 1 时尝试向 Flash 地址写入数据），则操作将中止并且 FLASH\_SR 寄存器中的编程顺序错误标志位 (PGSERR) 将置 1。

### 编程与缓存

如果 Flash 写访问涉及数据缓存中的某些数据，Flash 写访问将修改 Flash 中的数据 and 缓存中的数据。

如果 Flash 中的擦除操作也涉及数据或指令缓存中的数据，则必须确保在代码执行期间访问这些数据之前将它们重写。如果无法安全执行这一操作，建议将 FLASH\_CR 寄存器中的 DCRST 和 ICRST 位置 1，以刷新缓存。

注：I/D 缓存只有在被禁止 (I/DCEN = 0) 的情况下才能刷新。

## 1.5.5 中断

如果将 FLASH\_CR 寄存器中的操作结束中断使能位 (EOPIE) 置 1，则在擦除或编程操作结束时，即 FLASH\_SR 寄存器中的繁忙位 (BSY) 清零（操作正确或非正确完成）时，将产生中断。此时，FLASH\_SR 寄存器中的操作结束 (EOP) 位置 1。

如果在请求编程或擦除操作期间出现错误，则 FLASH\_SR 寄存器中的以下错误标志位之一将置 1：

- PGAERR、PGPERR、PGSERR（编程错误标志）
- WRPERR（保护错误标志）

这种情况下，如果 FLASH\_SR 寄存器中的错误中断使能位 (ERRIE) 置 1，则将产生一个中断并且 FLASH\_SR 寄存器中的操作错误位 (OPERR) 置 1。

注：如果检测到多个连续错误（例如，在对 Flash 进行 DMA 传输期间），则直到连续写操作请求结束，这些错误标志才会清零。

# 1.6 选项字节

## 1.6.1 关于用户选项字节的说明

选项字节由最终用户根据具体的应用要求进行配置。[表 4](#) 介绍了这些字节在用户配置扇区内的构成。

**表 4. 选项字节构成**

地址	[63:16]	[15:0]
0x1FFF C000	保留	ROP 和用户选项字节 (RDP & USER)
0x1FFF C008	保留	写保护 nWRP

**表 5. 关于选项字节的说明**

选项字节（字，地址 0x1FFF C000）	
<b>RDP：读保护选项字节。</b> 读保护用于保护 Flash 中存储的软件代码。	
位 15:8	0xAA：级别 0，无保护 0xCC：级别 2，芯片保护（禁止调试和从 RAM 自举功能） 其它值：级别 1，存储器读保护（调试功能受限）
<b>USER：用户选项字节</b> 此字节用于配置以下功能： <ul style="list-style-type: none"> <li>– 选择看门狗：硬件或软件</li> <li>– 进入停机模式时产生复位事件</li> <li>– 进入待机模式时产生复位事件</li> </ul>	
位 7	<b>nRST_STDBY</b> 0：进入待机模式时产生复位 1：不产生复位
位 6	<b>nRST_STOP</b> 0：进入停机模式时产生复位 1：不产生复位
位 5	<b>WDG_SW</b> 0：硬件看门狗 1：软件看门狗
位 4	0x1：未使用
位 3:2	<b>BOR_LEV：BOR 复位级别</b> 这些位包含用来激活 / 释放复位的供电电压阈值。通过对这些位执行写操作，可将新的 BOR 级别值编程到 Flash。 00：BOR 级别 3 (VBOR3)，复位阈值电压为 2.70 V 到 3.60 V 01：BOR 级别 2 (VBOR2)，复位阈值电压为 2.40 V 到 2.70 V 10：BOR 级别 1 (VBOR1)，复位阈值电压为 2.10 V 到 2.40 V 11：BOR 关闭 (VBOR0)，复位阈值电压为 1.8 V 到 2.10 V
位 1:0	0x1：未使用

表 5. 关于选项字节的说明（续）

选项字节（字，地址 0x1FFF C008）	
位 15:12	0xF: 未使用
<b>nWRP</b> : Flash 写保护选项字节 扇区 0 到 11 可采用写保护。	
位 i (0 ≤ i ≤ 11)	<b>nWRPi</b> 0: 激活扇区 i 的写保护 1: 未激活扇区 i 的写保护

### 1.6.2 用户选项字节编程

要针对此扇区执行任何操作，Flash 选项控制寄存器 (FLASH\_OPTCR) 中的选项锁定位 (OPTLOCK) 必须清零。为了能够将该位清零，用户需要顺序执行以下步骤：

1. 在 Flash 选项字节密钥寄存器 (FLASH\_OPTKEYR) 中写入 OPTKEY1 = 0x0819 2A3B
2. 在 Flash 选项字节密钥寄存器 (FLASH\_OPTKEYR) 中写入 OPTKEY2 = 0x4C5D 6E7F

通过软件将 OPTLOCK 位置 1 后，可防止用户选项字节发生意外的擦除/编程操作。

#### 修改用户选项字节

要修改用户选项值，请顺序执行以下步骤：

1. 检查 FLASH\_SR 寄存器中的 BSY 位，以确认当前未执行任何 Flash 操作
2. 在 FLASH\_OPTCR 寄存器中写入所需的选项值
3. 将 FLASH\_OPTCR 寄存器中的选项启动位 (OPTSTRT) 置 1
4. 等待 BSY 位清零

注：按以下步骤可自动修改选项值：首先擦除用户配置扇区，然后以 FLASH\_OPTCR 寄存器中包含的值对所有选项字节进行编程。

### 1.6.3 读保护 (RDP)

可对 Flash 中的用户区域实施读保护，以防不受信任的代码读取其中的数据。读保护分三个级别，具体定义如下：

- 级别 0：无读保护。  
将 0xAA 写入读保护选项字节 (RDP) 时，读保护级别即设为 0。此时，在所有自举配置（Flash 用户自举、调试或从 RAM 自举）中，均可执行与 Flash 或备份 SRAM 相关的所有读/写操作（如果未设置写保护）。
- 级别 1：存储器读保护。  
这是擦除选项字节后的默认读保护级别。将任意值（分别用于设置级别 0 和级别 2 的 0xAA 和 0xCC 除外）写入 RDP 选项字节时，即激活读保护级别 1。设置读保护级别 1 后：
  - 在连接调试功能或从 RAM 进行自举时，将不执行任何 Flash 访问（读取、擦除和编程）。Flash 读请求将导致总线错误。而在使用 Flash 用户自举功能或在系统存储器自举模式下操作时，则可执行所有操作。
  - 激活级别 1 后，如果将保护选项字节 (RDP) 编程为级别 0，则将对 Flash 和备份 SRAM 执行批量擦除。因此，在取消读保护之前，用户代码区域会清零。批量擦除操作仅擦除用户代码区域。包括写保护在内的其它选项字节将保持与批量擦除操作前相同。OTP 区域不受批量擦除操作的影响，同样保持不变。

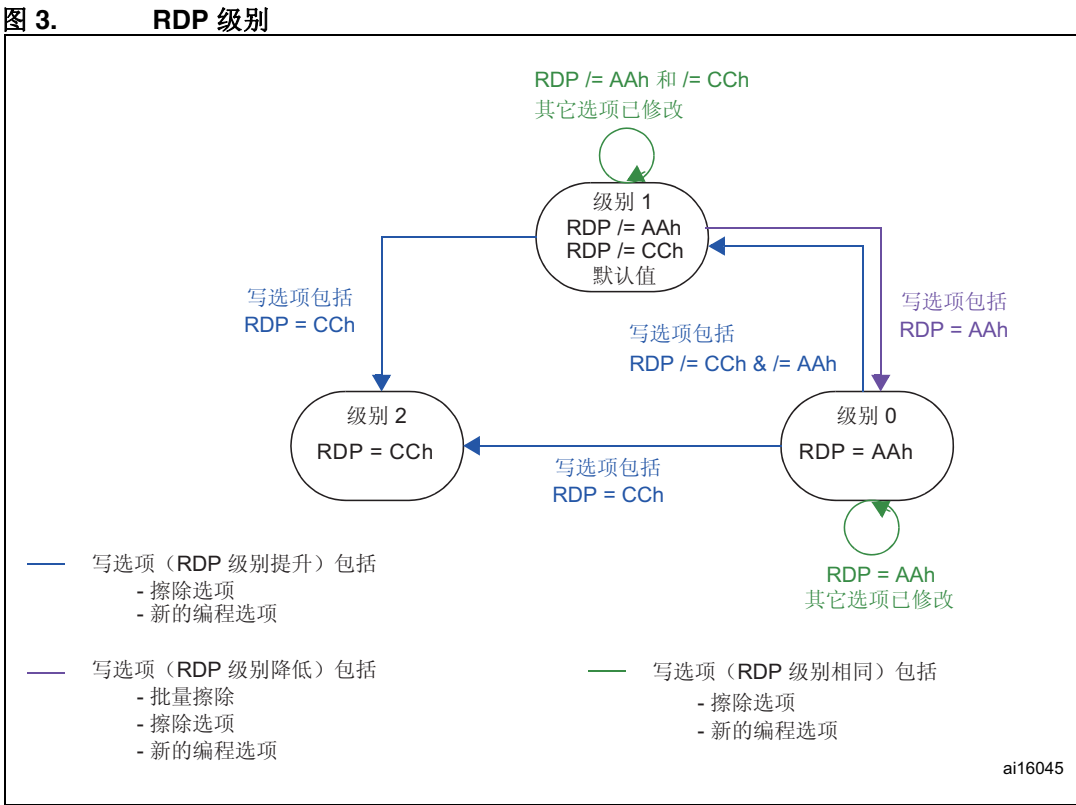


只有在已激活级别 1 并请求级别 0 时，才会执行批量擦除。当提高保护级别 (0->1, 1->2, 0->2) 时，不会执行批量擦除。

- 级别 2：禁止调试/芯片读保护  
将 0xCC 写入 RDP 选项字节时，即激活读保护级别 2。此时，由级别 1 提供的所有保护均生效，从 SRAM 或系统存储器自举时将禁止系统存储器和所有调试功能（CPU JTAG 调试和单线调试），并且用户选项不能再更改。  
存储器读保护级别 2 是不可撤消的操作。激活级别 2 后，保护级别不能再降回级别 0 或级别 1。

注：激活级别 2 后，将永久性禁止 JTAG 端口（相当于 JTAG 熔丝）。这样，将无法执行边界扫描。意法半导体无法对设为保护级别 2 的缺陷部件进行分析。

图 3 所示为 RDP 级别切换的过程。



### 1.6.4 写保护

Flash 中的用户扇区（0 到 11）具备写保护功能，可防止因程序计数器 (PC) 跑飞而发生意外的写操作。当扇区 i 中的非写保护位 (nWRPi, 0 ≤ i ≤ 11) 为低电平时，无法对扇区 i 执行擦除或编程操作。因此，如果某个扇区处于写保护状态，则无法执行批量擦除。

如果尝试对 Flash 中处于写保护状态的区域执行擦除/编程操作（由写保护位保护的扇区、锁定的 OTP 区域或永远不能执行写操作的 Flash 区域，例如 ICP），则 FLASH\_SR 寄存器中的写保护错误标志位 (WRPERR) 将置 1。

注：选择存储器读保护级别（RDP 级别 = 1）后，如果已连接 CPU 调试功能（JTAG 调试或单线调试）或者正在从 RAM 执行自举代码，则即使 nWRPi = 1，也无法对 Flash 扇区 i 执行编程或擦除操作。



# 1.7 一次性可编程字节

表 6 所示为 OTP 区域中一次性可编程 (OTP) 部分的构成。

表 6. OTP 区域构成

块	[128:96]	[95:64]	[63:32]	[31:0]	地址字节 0
0	OTP0	OTP0	OTP0	OTP0	0x1FFF 7800
	OTP0	OTP0	OTP0	OTP0	0x1FFF 7810
1	OTP1	OTP1	OTP1	OTP1	0x1FFF 7820
	OTP1	OTP1	OTP1	OTP1	0x1FFF 7830
.	.				.
.	.				.
.	.				.
15	OTP15	OTP15	OTP15	OTP15	0x1FFF 79E0
	OTP15	OTP15	OTP15	OTP15	0x1FFF 79F0
锁定块	LOCKB15 ... LOCKB12	LOCKB11 ... LOCKB8	LOCKB7 ... LOCKB4	LOCKB3 ... LOCKB0	0x1FFF 7A00

OTP 区域划分为 16 个 32 字节的 OTP 数据块和 1 个 16 字节的 OTP 锁定块。OTP 数据块和锁定块均无法擦除。锁定块中包含 16 字节的 LOCKBi (0 ≤ i ≤ 15)，用于锁定相应的 OTP 数据块（块 0 到 15）。每个 OTP 数据块均可编程，除非相应的 OTP 锁定字节编程为值 0x00。锁定字节的值只能是 0x00 和 0xFF，否则这些 OTP 字节无法正确使用。

# 1.8 Flash 接口寄存器

## 1.8.1 Flash 访问控制寄存器 (FLASH\_ACR)

Flash access control register

Flash 访问控制寄存器用于使能/禁止加速功能，并且可根据 CPU 频率控制 Flash 访问时间。

地址偏移：0x00

复位值：0x0000 0000

访问：无等待周期，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DCRST	ICRST	DCEN	ICEN	PRFTEN	Reserved					LATENCY		
			rw	w	rw	rw	rw						rw	rw	rw

位 31:11 保留，必须保持清零。

位 12 **DCRST**：数据缓存复位 (Data cache reset)

0：数据缓存不复位

1：数据缓存复位

只有在禁止数据缓存时才能在该位中写入值。

位 11 **ICRST**：指令缓存复位 (Instruction cache reset)

0：指令缓存不复位

1：指令缓存复位

只有在禁止指令缓存时才能在该位中写入值。

位 10 **DCEN**：数据缓存使能 (Data cache enable)

0：禁止数据缓存

1：使能数据缓存

位 9 **ICEN**：指令缓存使能 (Instruction cache enable)

0：禁止指令缓存

1：使能指令缓存

位 8 **PRFTEN**：预取使能 (Prefetch enable)

0：禁止预取

1：使能预取

位 7:3 保留，必须保持清零。

位 2:0 **LATENCY**：延迟 (Latency)

这些位表示 CPU 时钟周期与 Flash 访问时间之比。

000：零等待周期

001：一个等待周期

010：两个等待周期

011：三个等待周期

100：四个等待周期

101：五个等待周期

110：六个等待周期

111：七个等待周期

## 1.8.2 Flash 密钥寄存器 (FLASH\_KEYR)

Flash key register

借助 Flash 密钥寄存器，可允许访问 Flash 控制寄存器，进而允许执行编程和擦除操作。

地址偏移：0x04

复位值：0x0000 0000

访问：无等待周期，按字访问

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

位 31:0 **FKEYR**: FPEC 密钥 (FPEC key)

要将 FLASH\_CR 寄存器解锁并允许对其执行编程/擦除操作，必须顺序编程以下值：

- a) KEY1 = 0x45670123
- b) KEY2 = 0xCDEF89AB

## 1.8.3 Flash 选项字节密钥寄存器 (FLASH\_OPTKEYR)

Flash option key register

借助 Flash 选项字节密钥寄存器，可允许在用户配置扇区中执行编程和擦除操作。

地址偏移：0x08

复位值：0x0000 0000

访问：无等待周期，按字访问

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

位 31:0 **OPTKEYR**: 选项字节密钥 (Option byte key)

要将 FLASH\_OPTCR 寄存器解锁并允许对其编程，必须顺序编程以下值：

- a) OPTKEY1 = 0x08192A3B
- b) OPTKEY2 = 0x4C5D6E7F

### 1.8.4 Flash 状态寄存器 (FLASH\_SR)

Flash status register

Flash 状态寄存器提供正在执行的编程和擦除操作的相关信息。

地址偏移：0x0C

复位值：0x0000 0000

访问：无等待周期，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															BSY
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								PGSERR	PGPERR	PGAERR	WRPERR	Reserved		OPERR	EOP
								rc_w1	rc_w1	rc_w1	rc_w1			rc_w1	rc_w1

位 31:17 保留，必须保持清零。

位 16 **BSY**：繁忙 (Busy)

该位指示某个 Flash 操作正在进行。该位在 Flash 操作开始时置 1，在操作结束或出现错误时清零。

- 0：当前未执行任何 Flash 操作
- 1：正在执行 Flash 操作

位 15:8 保留，必须保持清零。

位 7 **PGSERR**：编程顺序错误 (Programming sequence error)

如果代码在控制寄存器未正确配置的情况下对 Flash 执行写访问，将由硬件为该位置 1。  
写入 1 即可将该位清零。

位 6 **PGPERR**：编程并行位数错误 (Programming parallelism error)

如果在编程期间数据访问类型（字节、半字、字和双字）与配置的并行位数 PSIZE (x8, x16, x32, x64) 不符，将由硬件为该位置 1。  
写入 1 即可将该位清零。

位 5 **PGAERR**：编程对齐错误 (Programming alignment error)

如果要编程的数据不能包含在同一个 128 位 Flash 行中，将由硬件为该位置 1。  
写入 1 即可将该位清零。

位 4 **WRPERR**：写保护错误 (Write protection error)

如果要擦除/编程的地址属于 Flash 中处于写保护状态的区域，将由硬件为该位置 1。  
写入 1 即可将该位清零。

位 3:2 保留，必须保持清零。

位 1 **OPERR**：操作错误 (Operation error)

如果检测到 Flash 操作（编程/擦除）请求，但由于存在并行位数错误、对齐错误、顺序错误或写保护错误而无法运行，将由硬件对该位置 1。只有在使能错误中断 (ERRIE = 1) 后，该位才会置 1。

位 0 **EOP**：操作结束 (End of operation)

当成功完成一个或多个 Flash 操作（编程/擦除）时，由硬件将该位置 1。只有在使能操作结束中断 (EOPIE = 1) 后，该位才会置 1。  
写入 1 即可将该位清零。

# 1.8.5 Flash 控制寄存器 (FLASH\_CR)

Flash control register

Flash 控制寄存器用于配置和启动 Flash 操作。

地址偏移: 0x10

复位值: 0x8000 0000

访问: 当前未执行任何 Flash 操作时无等待周期, 按字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	Reserved					ERRIE	EOPIE	Reserved							STRT
rs						rw	rw								rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						PSIZE[1:0]		Reserved	SNB[3:0]				MER	SER	PG
						rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31 **LOCK**: 锁定 (Lock)

该位只能写入 1。该位置 1 时, 表示 FLASH\_CR 寄存器已锁定。当检测到解锁序列时, 由硬件将该位清零。  
如果解锁操作失败, 该位仍保持置 1, 直到下一次复位。

位 31:26 保留, 必须保持清零。

位 25 **ERRIE**: 错误中断使能 (Error interrupt enable)

当 FLASH\_SR 寄存器中的 OPERR 位置 1 后, 可通过该位使能中断产生功能。  
0: 禁止产生错误中断  
1: 使能产生错误中断

位 24 **EOPIE**: 操作结束中断使能 (End of operation interrupt enable)

当 FLASH\_SR 寄存器中的 EOP 位置 1 后, 可通过该位使能中断产生功能。  
0: 禁止产生中断  
1: 使能产生中断

位 23:17 保留, 必须保持清零。

位 16 **STRT**: 启动 (Start)

该位置 1 后可触发擦除操作。该位只能通过软件置 1, 并在 BSY 位清零后随之清零。

位 15:10 保留, 必须保持清零。

位 9:8 **PSIZE**: 编程大小 (Program size)

这些位用于选择编程并行位数。  
00 x8 编程  
01 x16 编程  
10 x32 编程  
11 x64 编程

位 7 保留, 必须保持清零。

位 6:3 **SNB**: 扇区编号 (Sector number)

这些位用于选择要擦除的扇区。  
0000 扇区 0  
0001 扇区 1  
...  
1011 扇区 11  
不允许使用其它值

- 位 2 **MER**: 批量擦除 (Mass Erase)  
针对所有用户扇区激活擦除操作。
- 位 1 **SER**: 扇区擦除 (Sector Erase)  
激活扇区擦除。
- 位 0 **PG**: 编程 (Programming)  
激活 Flash 编程。

### 1.8.6 Flash 选项控制寄存器 (FLASH\_OPTCR)

Flash option control register

FLASH\_OPTCR 寄存器用于修改用户选项字节。

地址偏移: 0x14

复位值: 0x0FFF AAED。释放复位时, 将 Flash 中的值加载到这些选项位。

访问: 当前未执行任何 Flash 操作时无等待周期, 按字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				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_STDBY	nRST_STOP	WDG_SW	Reserved	BOR_LEV		OPTSTRT	OPTLOCK
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rs	rs

- 位 31:28 保留, 必须保持清零。
- 位 27:16 **nWRP**: 无写保护 (Not write protect)  
这些位包含复位后写保护选项字节的值。通过对这些位执行写操作, 可将新的写保护值编程到 Flash。  
0: 已激活写保护  
1: 未激活写保护
- 位 15:8 **RDP**: 读保护 (Read protect)  
这些位包含复位后读保护选项字节的值。通过对这些位执行写操作, 可将新的读保护值编程到 Flash。  
0xAA: 级别 0, 未激活读保护  
0xCC: 级别 2, 激活芯片读保护  
其它值: 级别 1, 激活存储器读保护
- 位 7:5 **USER**: 用户选项字节 (User option bytes)  
这些位包含复位后用户选项字节的值。通过对这些位执行写操作, 可将新的用户选项字节值编程到 Flash。  
位 7: nRST\_STDBY  
位 6: nRST\_STOP  
位 5: WDG\_SW  
注: 当 WDG 模式从硬件切换到软件或从软件切换到硬件时, 需要执行系统复位才能更改生效。
- 位 4 保留, 必须保持清零。

### 位 3:2 **BOR\_LEV**: BOR 复位级别 (BOR reset Level)

这些位包含用来激活/释放复位的供电电压阈值。可通过对这些位执行写操作，来编程新的 BOR 级别。BOR 默认为关闭。当电源电压 ( $V_{DD}$ ) 降至所选 BOR 级别以下时，将产生器件复位。

00: BOR 级别 3 (VBOR3)，复位阈值电压范围为 2.70 V 到 3.60 V

01: BOR 级别 2 (VBOR2)，复位阈值电压范围为 2.40 V 到 2.70 V

10: BOR 级别 1 (VBOR1)，复位阈值电压范围为 2.10 V 到 2.40 V

11: BOR 关闭 (VBOR0)，复位阈值电压范围为 1.80 V 到 2.10 V

注：有关 BOR 特性的完整详情，请参见器件数据手册中的“电气特性”部分。

### 位 1 **OPTSTRT**: 启动选项 (Option start)

该位置 1 后可触发用户选项操作。该位只能通过软件置 1，并在 BSY 位清零后随之清零。

### 位 0 **OPTLOCK**: 锁定选项 (Option lock)

该位只能写入 1。该位置 1 时，表示 FLASH\_OPTCR 寄存器已锁定。当检测到解锁序列时，由硬件将该位清零。

如果解锁操作失败，该位仍保持置 1，直到下一次复位。

## 1.8.7 Flash 接口寄存器映射

表 7. Flash 寄存器映射与复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	FLASH_ACR	Reserved																				DCRST	ICRST	DCEN	ICEN	PRFTEN	Reserved					LATENCY		
	Reset value																					0	0	0	0	0						0	0	0
0x04	FLASH_KEYR	KEY[31:16]																KEY[15:0]																
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x08	FLASH_OPTKEYR	OPTKEYR[31:16]																OPTKEYR[15:0]																
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x0C	FLASH_SR	Reserved																BSY	Reserved					PGSERR	PGPERR	PGAERR	WRPERR	Reserved		OPERR	EOP			
	Reset value																	0						0	0	0	0			0	0			
0x10	FLASH_CR	LOCK	Reserved						EOPIE		Reserved						STRT	Reserved				PSIZE[1:0]		Reserved	SNB[3:0]			MER	SER	PG				
	Reset value	1							0								0					0		0	0	0			0	0				
0x14	FLASH_OPTCR	Reserved				nWRP[11:0]												RDP[7:0]					nRST_STDBY	nRST_STOP	WDG_SW	Reserved	BOR_LEV		OPTSTRT	OPTLOCK				
	Reset value					1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1	1	1	1	0	1	

## 2 版本历史

表 8. 文档版本历史

日期	版本	变更
2011 年 09 月 15 日	1	初始版本。





**请仔细阅读：**

中文翻译仅为方便阅读之目的。该翻译也许不是对本文档最新版本的翻译，如有任何不同，以最新版本的英文原版文档为准。

本文档中信息的提供仅与 ST 产品有关。意法半导体公司及其子公司（“ST”）保留随时对本文档及本文所述产品与服务进行变更、更正、修改或改进的权利，恕不另行通知。

所有 ST 产品均根据 ST 的销售条款出售。

买方自行负责对本文所述 ST 产品和服务的选择和使用，ST 概不承担与选择或使用本文所述 ST 产品和服务相关的任何责任。

无论之前是否有任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为 ST 授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在 ST 的销售条款中另有说明，否则，ST 对 ST 产品的使用和 / 或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

意法半导体的产品不得应用于武器。此外，意法半导体产品也不是为下列用途而设计并不得应用于下列用途：（A）对安全性有特别要求的应用，例如，生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）汽车应用或汽车环境，且 / 或（D）航天应用或航天环境。如果意法半导体产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向意法半导体发出了书面通知，采购商仍将独自承担因此而导致的任何风险，意法半导体的产品规格明确指定的汽车、汽车安全或医疗工业领域专用产品除外。根据相关政府主管部门的规定，ESCC、QML 或 JAN 正式认证产品适用于航天应用。

经销的 ST 产品如有不同于本文档中提出的声明和 / 或技术特点的规定，将立即导致 ST 针对本文所述 ST 产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大 ST 的任何责任。

ST 和 ST 徽标是 ST 在各个国家或地区的商标或注册商标。

本文档中的信息取代之前提供的所有信息。

ST 徽标是意法半导体公司的注册商标。其他所有名称是其各自所有者的财产。

© 2013 STMicroelectronics 保留所有权利

意法半导体集团公司

澳大利亚 - 比利时 - 巴西 - 加拿大 - 中国 - 捷克共和国 - 芬兰 - 法国 - 德国 - 中国香港 - 印度 - 以色列 - 意大利 - 日本 - 马来西亚 - 马耳他 - 摩洛哥 - 菲律宾 - 新加坡 - 西班牙 - 瑞典 - 瑞士 - 英国 - 美国

[www.st.com](http://www.st.com)

