Flash 使用指南

概述

片内 Flash 存储器为掉电非易失存储介质，用于存储用户程序、数据等内容。

适用范围	
SPC1125 系列	SPC1125，SPC1128
SPC1168 系列	SPC1155，SPC1156，SPC1158，SPC1168，SPD1148，SPD1178，SPD1188，SPD1163，SPM1173
SPC2168 系列	SPC2168，SPC2165，SPC2166，SPC1198
SPC1169 系列	SPC1169，SPD1179，SPD1176
SPC2188 系列	SPC1185，SPC2188

目录

1 SPC1125 系列 8
1.1 特性 8
1.2 功能描述 8
1.2.1 访问接口 8
1.2.2 写保护 10
1.2.3 配置字 10
1.3 功能实例 10
1.3.1 实例 1：设定时序参数 10
1.3.2 实例 2：XIP 读操作 11
1.3.3 实例 3：FLASH 控制器操作 11
2 SPC1168 系列 12
2.1 特性 12
2.2 功能描述 12
2.2.1 访问接口 12
2.2.2 写保护 14
2.2.3 配置字 14
2.3 功能实例 17
2.3.1 实例 1：设定时序参数 17
2.3.2 实例 2：XIP 读操作 17
2.3.3 实例 3：FLASH 控制器操作 17
3 SPC2168 系列 19
3.1 特性 19
3.2 功能描述 19
3.2.1 访问接口 19
3.2.2 写保护 20
3.2.3 配置字 21
3.3 功能实例 24
3.3.1 实例 1：设定时序参数 24
3.3.2 实例 2：XIP 读操作 24
3.3.3 实例 3：FLASH 控制器操作 24
4 SPC1169 系列 26
4.1 特性 26
4.2 功能描述 26

4.2.1 访问接口 26
4.2.2 写保护 27
4.2.3 配置字 28
4.3 功能实例 28
4.3.1 实例 1：设定时序参数 28
4.3.2 实例 2：XIP 读操作 28
4.3.3 实例 3：FLASH 控制器操作 29
5 SPC2188 系列 30
5.1 特性 30
5.2 功能描述 30
5.2.1 访问接口 30
5.2.2 配置字 31
5.3 功能实例 32
5.3.1 实例 1：设定时序参数 32
5.3.2 实例 2：XIP 读操作 32
5.3.3 实例 3：FLASH 控制器操作 32

图片列表

图 1-1：Flash 存储器访问接口 8
图 1-2：Flash 存储器逻辑结构 9
图 2-1：Flash 存储器访问接口 12
图 2-2：Flash 存储器逻辑结构 13
图 3-1：Flash 存储器访问接口 19
图 3-2：Flash 存储器逻辑结构 20
图 4-1：Flash 存储器访问接口 26
图 4-2：Flash 存储器逻辑结构 27
图 5-1：Flash 存储器访问接口 30
图 5-2：Flash 存储器逻辑结构 31

表格列表

表 1-1：Flash 存储器结构 9
表 1-2：配置字的构成及其描述 10
表 1-3：实例 3 代码路径 11
表 2-1：Flash 存储器结构 13
表 2-2：配置字的构成及其描述 14
表 2-3：实例 3 代码路径 18
表 3-1：Flash 存储器结构 20
表 3-2：配置字的构成及其描述 21
表 3-3：实例 3 代码路径 25
表 4-1：Flash 存储器结构 27
表 4-2：配置字的构成及其描述 28
表 4-3：实例 3 代码路径 29
表 5-1：Flash 存储器结构 31
表 5-2：配置字的构成及其描述 32
表 5-3：实例 3 代码路径 33

版本历史

版本	日期	作者	状态	变更
A/0	2023-04-20	CanChai	Outdated	1. 首次发布。
A/1	2023-06-09	CanChai	Outdated	1. 更新章节 1.5。
C/0	2024-08-13	LemengZhou	Outdated	1. 修改为全系列通用版本。
C/1	2024-09-19	CanChai	Released	1. 修改图 1-1、图 2-1。

术语或缩写

术语或缩写	描述
CPU	Central Processing Unit
XIP	Executed in place

1 SPC1125 系列

1.1 特性

片内 Flash 存储器用于存储用户程序、数据等内容。

- 容量可达 64KB/128KB
- 存储单元组成
  - 主存储单元：包含 128/256 个扇区（Sector），每个扇区由 128 个 Word 组成
  - NVR 存储单元：包含 2 个扇区，每个扇区由 128 个 Word 组成
- 最小编程单元为一个 Word（32-bit）
- 支持大小为 512 字节的区块擦除
- 支持至少 100000 次擦写（T_J = 85 °C）
- Flash 内数据至少保存 10 年（T_J = 85 °C）
- 支持区块保护模式

1.2 功能描述

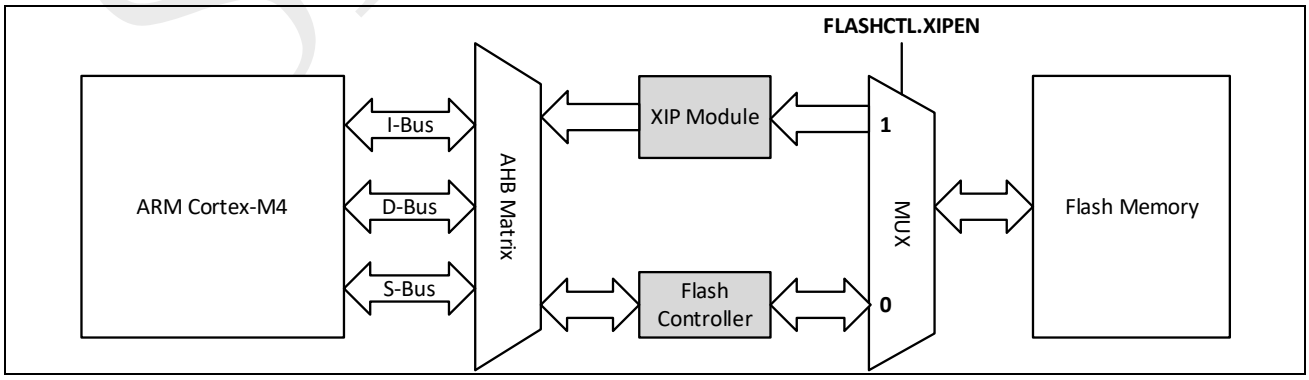
1.2.1 访问接口

FLASH 存储器的访问接口有两个：

- XIP 模块
- Flash 控制器

如图 1-1 所示，XIP 模块只能实现从 Flash 存储器中读取数据，用于 CPU 从 Flash 存储器中取代码和数据。Flash 控制器是接在 AHB 总线上的一个外设，可以实现 Flash 存储器的读、写以及擦除操作。但是，这两个接口不能同时工作，同一时刻只能有一个接口处于工作状态。Flash 存储器当前的访问接口由寄存器 FLASHCTL.XIPEN 进行控制。

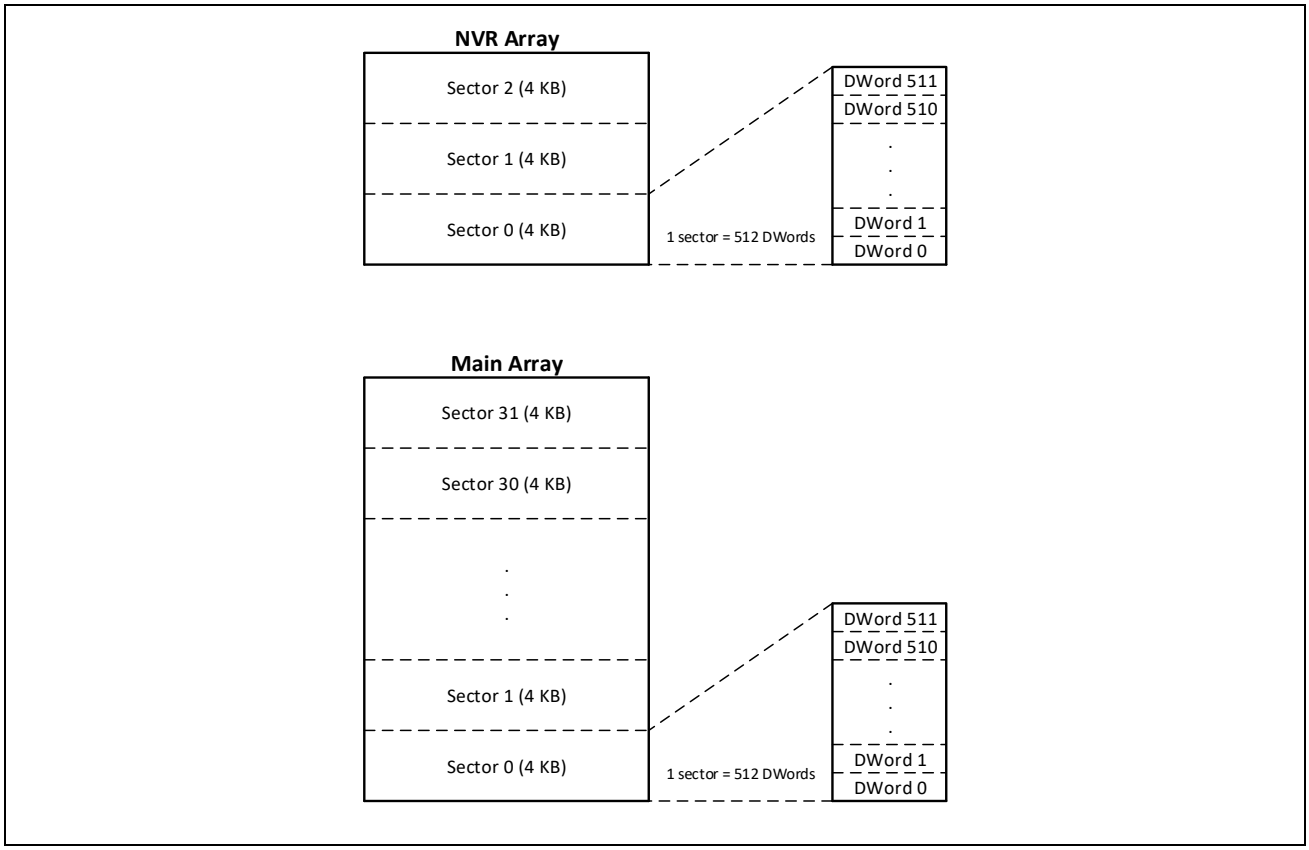
图 1-1：Flash 存储器访问接口



Flash 存储器的构成如图 1-2 和表 1-1 所示，包含两个部分：

- 主存储单元：这部分由 32 个扇区组成，每个扇区的大小为 4096 字节，共计 128KB；
- NVR 存储单元：这部分由 3 个扇区组成，每个扇区的大小为 4096 字节。

图 1-2：Flash 存储器逻辑结构



[1] 对于容量为 64KB 的 Flash 存储器，主存储单元只有 128 个扇区（Sector 0 ~ Sector127）。

表 1-1：Flash 存储器结构

存储区块	名称	起始地址	大小
主存储单元	Sector 0	0x1000 0000	4096 字节
	Sector 1	0x1000 1000	4096 字节
	Sector 2	0x1000 2000	4096 字节
	Sector 3	0x1000 3000	4096 字节
	…	…	…
	Sector 31	0x1001 F000	4096 字节
NVR 存储单元	Sector 0	0x1100 2000	4096 字节
	Sector 1	0x1100 3000	4096 字节
	Sector 2	0x1100 4000	4096 字节

1.2.2 写保护

Flash 主存储区可以设置保护，以防止意外写入。如果对设置了保护的扇区执行编程或擦除操作，则操作将被忽略。

写保护通过设置 FLASHWPx（x = 0，1，2，3）寄存器对应比特位激活。

重置寄存器对应比特位将关闭写保护。

1.2.3 配置字

配置字可以通过 XIP 或者 FLASH 控制器接口进行读取，可以通过 FLASH 控制器进行编程、擦写。新写入的配置字需要在芯片复位并重新加载后才能生效。

下面表 1-2 部分详细概述：

表 1-2：配置字的构成及其描述

地址	名字	描述
0x1100060C	CHIP_SECURITY0	锁住芯片调试接口配置字 0xFFFFFFFF：芯片调试接口将不会锁住 其它：芯片调试接口将会锁住
0x11000614	CHIP_SECURITY1	锁住芯片调试接口配置字 0xFFFFFFFF：芯片调试接口将不会锁住 其它：芯片调试接口将会锁住
0x11000700	WDT_ENABLE	Watchdog 开机使能字段 0xFFFFFFFF：关闭 Watchdog 当芯片启动时 其他值：使能 Watchdog 当芯片启动时

1.3 功能实例

1.3.1 实例 1：设定时序参数

Flash 正常工作时需要满足一定物理时序要求，用户可以调用 pHWLIB->FLASHC_SetTiming()函数对 Flash 时序进行设置。

每当用户计划更改 HCLK 频率时，应按如下步骤：

1. 在 HLK 频率更改之前调用 FLASHC_RelaxXIPTiming()，以宽松的时序访问 Flash。
2. 用户配置 HLK 频率
3. 在频率更改后，应调用 pHWLIB->FLASHC_SetTiming()函数以设置优化的 Flash 时序。

1.3.2 实例 2：XIP 读操作

通过 XIP 模块，可以实现对 Flash 存储器的读操作。用户可以直接对 Flash 进行字节、半字或者字形式的读取。下面是一个示例代码，从 Flash 中地址 0x10000100 处读取一个字节数据。

示例代码

```
uint8_t *pu8Data = (uint8_t *)0x10000100;
uint8_t u8Byte;

u8Byte = *pu8Data;
```

1.3.3 实例 3：FLASH 控制器操作

1.3.3.1 功能需求

对 FLASH 进行读取、编程、擦除操作。

1.3.3.2 功能实现

1. Flash 驱动代码固化存储于 ROM 空间。
2. 使用 SDK 提供的 FLASH 操作接口，对 FLASH 进行如下操作：
   a) 擦除一个扇区 FLASH
   b) 读取校验该扇区是否擦除成功
   c) 向该扇区写入数据
3. 以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 1-3：

表 1-3：实例 3 代码路径

MCU 产品型号	代码路径
SPC1125 系列	SDK 目录\0_Examples\ Flash_Operation

2 SPC1168 系列

2.1 特性

片内 Flash 存储器用于存储用户程序、数据等内容。

- 容量可达 64KB/128KB
- 存储单元组成
  - 主存储单元：包含 128/256 个扇区（Sector），每个扇区由 128 个 Word 组成
  - NVR 存储单元：包含 2 个扇区，每个扇区由 128 个 Word 组成
- 最小编程单元为一个 Word（32-bit）
- 支持大小为 512 字节的区块擦除
- 支持至少 100000 次擦写（T_J = 85 °C）
- Flash 内数据至少保存 10 年（T_J = 85 °C）
- 支持区块保护模式

2.2 功能描述

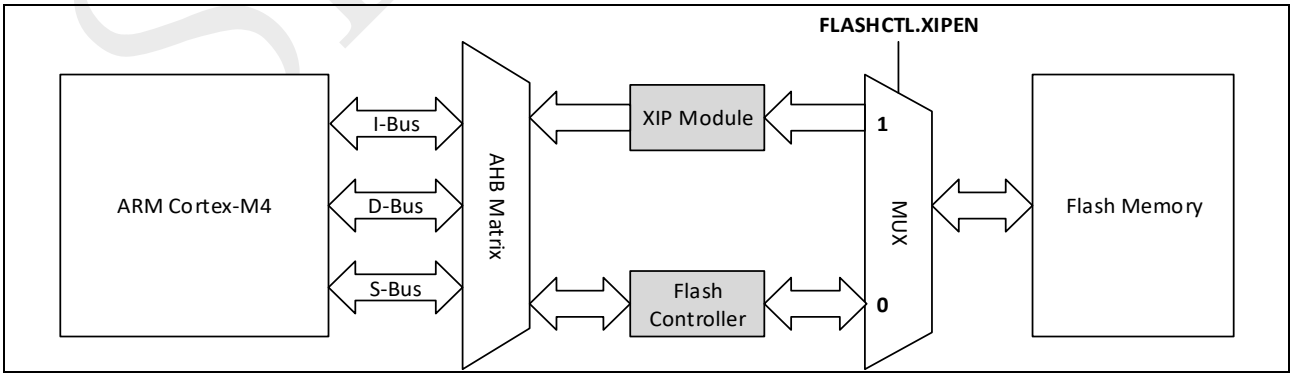
2.2.1 访问接口

FLASH 存储器的访问接口有两个：

- XIP 模块
- Flash 控制器

如图 2-1 所示，XIP 模块只能实现从 Flash 存储器中读取数据，用于 CPU 从 Flash 存储器中取代码和数据。Flash 控制器是接在 AHB 总线上的一个外设，可以实现 Flash 存储器的读、写以及擦除操作。但是，这两个接口不能同时工作，同一时刻只能有一个接口处于工作状态。Flash 存储器当前的访问接口由寄存器 FLASHCTL.XIPEN 进行控制。

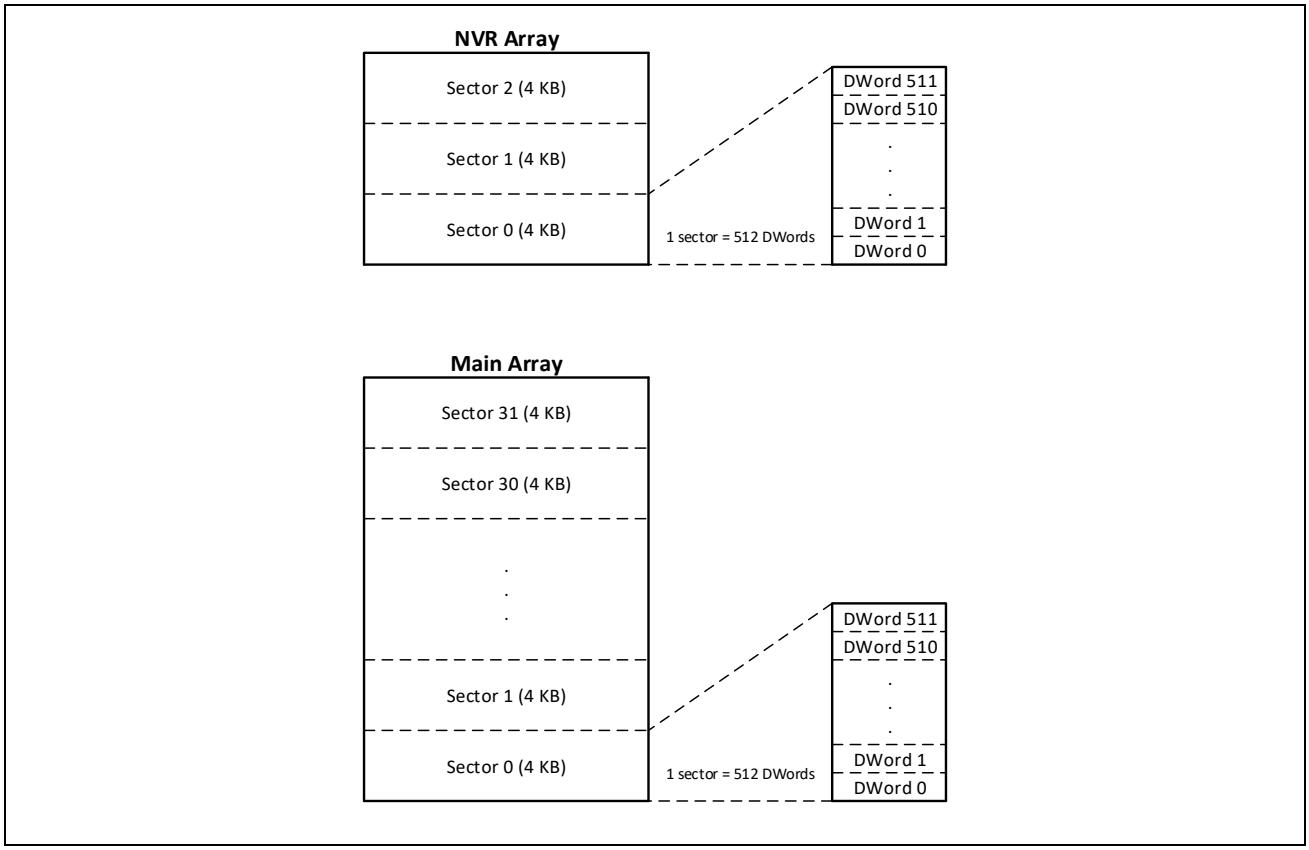
图 2-1：Flash 存储器访问接口



Flash 存储器的构成如图 2-2 和表 2-1 所示，包含两个部分：

- 主存储单元：这部分由 32 个扇区组成，每个扇区的大小为 4096 字节，共计 128KB；
- NVR 存储单元：这部分由 3 个扇区组成，每个扇区的大小为 4096 字节。

图 2-2：Flash 存储器逻辑结构



[1] 对于容量为 64KB 的 Flash 存储器，主存储单元只有 128 个扇区（Sector 0 ~ Sector127）。

表 2-1：Flash 存储器结构

存储区块	名称	起始地址	大小
主存储单元	Sector 0	0x1000 0000	4096 字节
	Sector 1	0x1000 1000	4096 字节
	Sector 2	0x1000 2000	4096 字节
	Sector 3	0x1000 3000	4096 字节
	…	…	…
	Sector 31	0x1001 F000	4096 字节
NVR 存储单元	Sector 0	0x1100 2000	4096 字节
	Sector 1	0x1100 3000	4096 字节
	Sector 2	0x1100 4000	4096 字节

2.2.2 写保护

Flash 主存储区可以设置保护，以防止意外写入。如果对设置了保护的扇区执行编程或擦除操作，则操作将被忽略。

写保护通过设置 FLASHWPx（x = 0，1，2，3）寄存器对应比特位激活。

重置寄存器对应比特位将关闭写保护。

2.2.3 配置字

配置字可以通过 XIP 或者 FLASH 控制器接口进行读取，可以通过 FLASH 控制器进行编程、擦写。新写入的配置字需要在芯片复位并重新加载后才能生效。

下面部分表 2-2 详细概述：

表 2-2：配置字的构成及其描述

地址	配置字名称	配置字描述
0x11000600	ZONE0_INFO_LOCK	锁定分区 0 的配置字(0x11000600 – 0x1100063F) 0xFFFFFFFF：分区 0 的配置字可以被编程 其他值：分区 0 的配置字不可以被编程
0x11000604	ZONE0_ENCRYPT	Flash 分区 0 代码的加密状态 0xDEC0DE：Flash 分区 0 的代码已被加密 其他值：Flash 分区 0 的代码是原始代码
0x11000608	ZONE0_DECRYPT	Flash 分区 0 代码的解密状态 0xAA621623：成功解密 Flash 分区 0 的代码 0x1E55051：未成功解密 Flash 分区 0 的代码 其他值：无效值
0x1100060C	ZONE0_FLASH_PROT	Flash 分区 0 保护字段 0xFFFFFFFF：关闭 Flash 分区 0 保护 其他值：使能 Flash 分区 0 保护
0x11000610	ZONE0_FLASH_ADDR	Flash 分区 0 起始地址
0x11000614	ZONE0_RAM_PROT	RAM 分区 0 保护字段 0xFFFFFFFF：关闭 RAM 分区 0 保护 其他值：使能 RAM 分区 0 保护
0x11000618	ZONE0_RAM_ADDR	RAM 分区 0 起始地址
0x1100061C~0x1100063F	Reserved for ZONE0	分区 0 保留配置字段
0x11000640	ZONE1_INFO_LOCK	锁定分区 1 的配置字(0x11000640 – 0x1100067F)

地址	配置字名称	配置字描述
		0xFFFFFFFF：分区 1 的配置字可以被编程 其他值：分区 1 的配置字不可以被编程
0x11000644	ZONE1_ENCRYPT	Flash 分区 1 代码的加密状态 0xDEC0DE：Flash 分区 1 的代码已被加密 其他值：Flash 分区 1 的代码是原始代码
0x11000648	ZONE1_DECRYPT	Flash 分区 1 代码的解密状态 0xAA621623：成功解密 Flash 分区 1 的代码 0x1E55051：未成功解密 Flash 分区 1 的代码 其他值：无效值
0x1100064C	ZONE1_FLASH_PROT	Flash 分区 1 保护字段 0xFFFFFFFF：关闭 Flash 分区 1 保护 其他值：使能 Flash 分区 1 保护
0x11000650	ZONE1_FLASH_ADDR	Flash 分区 1 起始地址
0x11000654	ZONE1_RAM_PROT	RAM 分区 1 保护字段 0xFFFFFFFF：关闭 RAM 分区 1 保护 其他值：使能 RAM 分区 1 保护
0x11000658	ZONE1_RAM_ADDR	RAM 分区 1 起始地址
0x1100065C~0x1100067F	Reserved for ZONE1	分区 1 保留配置字段
0x11000680	ZONE2_INFO_LOCK	锁定分区 2 的配置字(0x11000680 – 0x110006BF) 0xFFFFFFFF：分区 2 的配置字可以被编程 其他值：分区 2 的配置字不可以被编程
0x11000684	ZONE2_ENCRYPT	Flash 分区 2 代码的加密状态 0xDEC0DE：Flash 分区 2 的代码已被加密 其他值：Flash 分区 2 的代码是原始代码
0x1100688	ZONE2_DECRYPT	Flash 分区 2 代码的解密状态 0xAA621623：成功解密 Flash 分区 2 的代码 0x1E55051：未成功解密 Flash 分区 2 的代码 其他值：无效值
0x1100068C	ZONE2_FLASH_PROT	Flash 分区 2 保护字段 0xFFFFFFFF：关闭 Flash 分区 2 保护 其他值：使能 Flash 分区 2 保护

地址	配置字名称	配置字描述
0x11000690	ZONE2_FLASH_ADDR	Flash 分区 2 起始地址
0x11000694	ZONE2_RAM_PROT	RAM 分区 2 保护字段 0xFFFFFFFF：关闭 RAM 分区 2 保护 其他值：使能 RAM 分区 2 保护
0x11000698	ZONE2_RAM_ADDR	RAM 分区 2 起始地址
0x1100069C~0x110006BF	Reserved for ZONE2	分区 2 保留配置字段
0x110006C0	ZONE3_INFO_LOCK	锁定分区 3 的配置字(0x110006C0 – 0x110006FF) 0xFFFFFFFF：分区 3 的配置字可以被编程 其他值：分区 3 的配置字不可以被编程
0x110006C4	ZONE3_ENCRYPT	Flash 分区 3 代码的加密状态 0xDEC0DE：Flash 分区 3 的代码已被加密 其他值：Flash 分区 3 的代码是原始代码
0x110006C8	ZONE3_DECRYPT	Flash 分区 3 代码的解密状态 0xAA621623：成功解密 Flash 分区 3 的代码 0x1E55051：未成功解密 Flash 分区 3 的代码 其他值：无效值
0x110006CC	ZONE3_FLASH_PROT	Flash 分区 3 保护字段 0xFFFFFFFF：关闭 Flash 分区 3 保护 其他值：使能 Flash 分区 3 保护
0x110006D0	ZONE3_FLASH_ADDR	Flash 分区 3 起始地址
0x110006D4	ZONE3_RAM_PROT	RAM 分区 3 保护字段 0xFFFFFFFF：关闭 RAM 分区 3 保护 其他值：使能 RAM 分区 3 保护
0x110006D8	ZONE3_RAM_ADDR	RAM 分区 3 起始地址
0x110006DC ~ 0x110006FF	Reserved for ZONE3	分区 3 保留配置字段
0x11000700	WDT_ENABLE	Watchdog 开机使能字段 0xFFFFFFFF：关闭 Watchdog 当芯片启动时 其他值：使能 Watchdog 当芯片启动时
0x11000704 ~ 0x110007FF	Reserved	未使用

2.3 功能实例

2.3.1 实例 1：设定时序参数

Flash 正常工作时需要满足一定的物理时序要求，用户可以调用 SetTiming 函数对 Flash 时序进行设置。

每当用户计划更改 HCLK 频率时，应按如下步骤：

1. 如果系统的工作频率要升高，用户需要先调用 FLASH_SetTiming 设置时序参数，再将系统工作频率升高。
2. 如果系统的工作频率要降低，用户需要先将系统工作频率降低，再调用 FLASH_SetTiming 设置时序参数。

2.3.2 实例 2：XIP 读操作

通过 XIP 模块，可以实现对 Flash 存储器的读操作。用户可以直接对 Flash 进行字节、半字或者字形式的读取。下面是一个示例代码，从 Flash 中地址 0x10000100 处读取一个字节数据。

示例代码

```
uint8_t *pu8Data = (uint8_t *)0x10000100;
uint8_t u8Byte;

u8Byte = *pu8Data;
```

2.3.3 实例 3：FLASH 控制器操作

2.3.3.1 功能需求

对 FLASH 进行读取、编程、擦除操作。

2.3.3.2 功能实现

1. 用户的代码一般是在 Flash 存储器中运行的，此时 XIP 模块是使能的。此时，通过 Flash 控制器对 Flash 存储器进行操作，涉及到 Flash 访问接口切换的问题，需要特别注意。为了能够使 Flash 访问接口切换时，程序能够正常工作，需要将 Flash 控制器的操作代码放在 SRAM 中运行。因此，通过 Flash 控制器对 Flash 存储器进行操作的过程如下：
   a) 程序从 Flash 中跳转到 SRAM 中，关闭 XIP 模块，使能 Flash 控制器；
   b) 通过 Flash 控制器执行相应的 Flash 操作；
   c) 关闭 Flash 控制器，使能 XIP 模块，程序重新跳转到 Flash 执行。
2. SDK 中提供了 Flash 驱动函数。用户在实际使用中只需要将 Flash 驱动函数编译到 SRAM 中运行即可，这一步可以通过 Keil 软件提供的 Scatter 文件实现。下面是通过 Flash 控制器。
3. 使用 SDK 提供的 FLASH 操作接口，对 FLASH 进行如下操作：
   d) 擦除一个扇区 FLASH
   e) 读取校验该扇区是否擦除成功

f) 向该扇区写入数据

4. 以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 2-3：

表 2-3：实例 3 代码路径

MCU 产品型号	代码路径
SPC1155，SPC1156，SPC1158，SPC1168，SPD1148，SPD1178，SPD1188	SDK 目录\0_Examples\ Flash_Operation

3 SPC2168 系列

3.1 特性

片内 Flash 存储器用于存储用户程序、数据等内容。

- 容量可达 64KB/128KB
- 存储单元组成
  - 主存储单元：包含 128/256 个扇区（Sector），每个扇区由 128 个 Word 组成
  - NVR 存储单元：包含 2 个扇区，每个扇区由 128 个 Word 组成
- 最小编程单元为一个 Word（32-bit）
- 支持大小为 512 字节的区块擦除
- 支持至少 100000 次擦写（T_J = 85 °C）
- Flash 内数据至少保存 10 年（T_J = 85 °C）
- 支持区块保护模式

3.2 功能描述

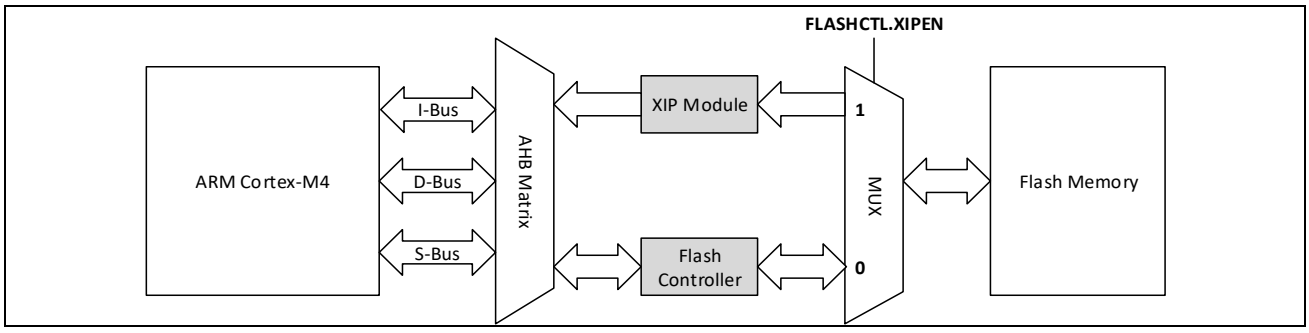
3.2.1 访问接口

Flash 存储器的访问接口有两个：

- XIP 模块
- FLASH 控制器

如图 3-1 所示，XIP 模块只能实现从 Flash 存储器中读取数据，用于 CPU 从 Flash 存储器中取代码和数据。FLASH 控制器是接在 AHB 总线上的一个外设，可以实现 Flash 存储器的读、写以及擦除操作。但是，这两个接口不能同时工作，同一时刻只能有一个接口处于工作状态。Flash 存储器当前的访问接口由寄存器 FLASHCTL.XIPEN 进行控制。

图 3-1：Flash 存储器访问接口



Flash 存储器的构成如图 3-2 和表 3-1 所示，包含两个部分：

- 主存储单元：这部分由 1024 个扇区组成，每个扇区的大小为 512 字节，共计 512KB；
- NVR 存储单元：这部分由 2 个扇区组成，每个扇区的大小为 512 字节。这两个扇区中，一

个扇区可以被配置为 OTP Flash 使用，通过在该扇区开始处写入 0x4C4F434B 实现；另一个扇区被用作配置字（Configuration Words），用于配置分区保护功能、WDT 启动使能以及 Flash Cache 使能等。需要特别说明的是，擦除 Configuration Words 扇区，同时会擦除整个主存储单元。

图 3-2：Flash 存储器逻辑结构

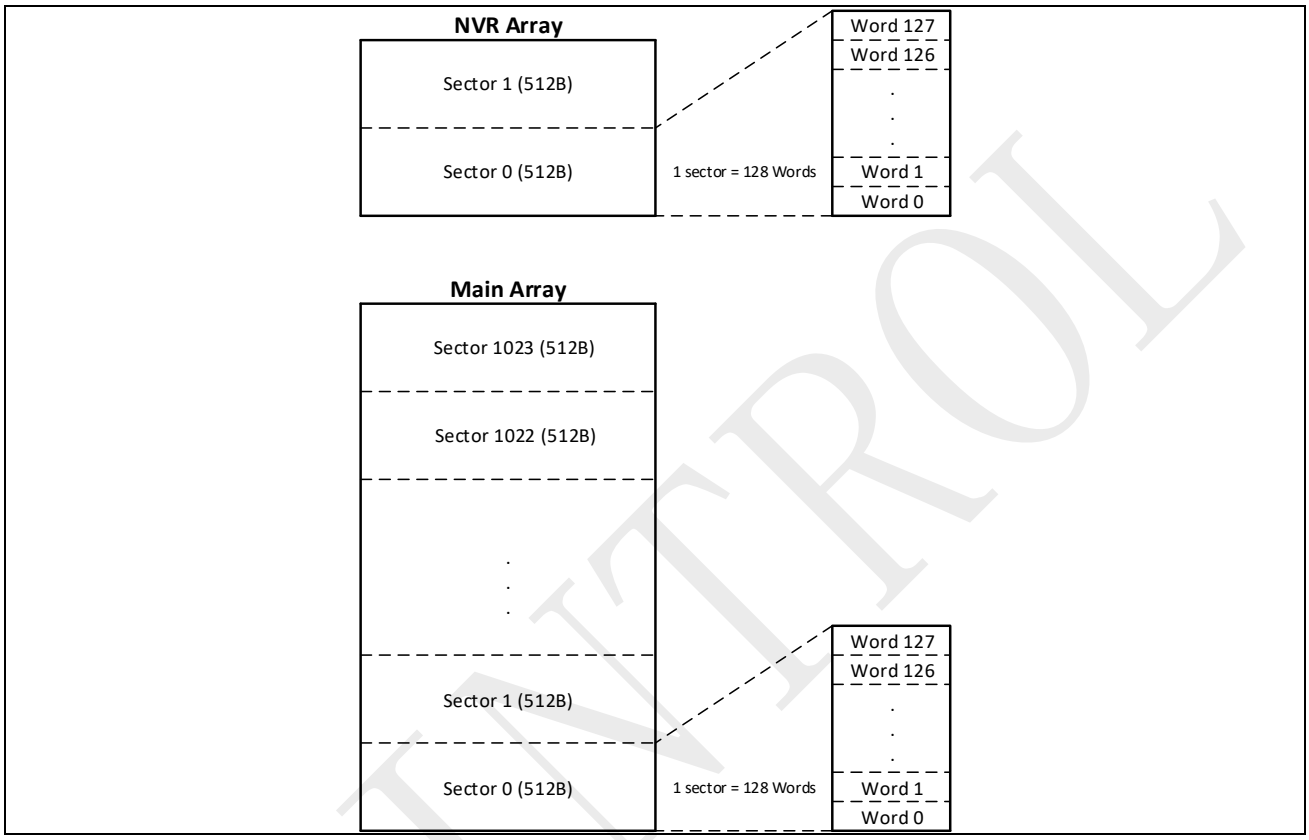


表 3-1：Flash 存储器结构

存储区块	名称	起始地址	大小
主存储单元	Sector 0	0x1000 0000	512 字节
	Sector 1	0x1000 0200	512 字节
	Sector 2	0x1000 0400	512 字节
	Sector 3	0x1000 0600	512 字节
	…	…	…
	Sector 1023	0x1007 FE00	512 字节
NVR 存储单元	OTP	0x1100 0400	512 字节
	Configuration Words	0x1100 0600	512 字节

3.2.2 写保护

Flash 主存储区可以设置保护，以防止意外写入。如果对设置了保护的扇区执行编程或擦除操作，则操作将被忽略。

写保护通过设置 FLASHWPx（x = 0，1，2，3）寄存器对应比特位激活。

重置寄存器对应比特位将关闭写保护。

3.2.3 配置字

配置字可以通过 XIP 或者 FLASH 控制器接口进行读取，可以通过 FLASH 控制器进行编程、擦写。新写入的配置字需要在芯片复位并重新加载后才能生效。

下面部分表 3-2 详细概述：

表 3-2：配置字的构成及其描述

地址	配置字名称	配置字描述
0x11000600	ZONE0_INFO_LOCK	锁定分区 0 的配置字（0x11000600 – 0x1100063F） 0xFFFFFFFF：分区 0 的配置字可以被编程 其他值：分区 0 的配置字不可以被编程
0x11000604	ZONE0_ENCRYPT	Flash 分区 0 代码的加密状态 0xDEC0DE：Flash 分区 0 的代码已被加密 其他值：Flash 分区 0 的代码是原始代码
0x11000608	ZONE0_DECRYPT	Flash 分区 0 代码的解密状态 0xAA621623：成功解密 Flash 分区 0 的代码 0x1E55051：未成功解密 Flash 分区 0 的代码 其他值：无效值
0x1100060C	ZONE0_FLASH_PROT	Flash 分区 0 保护字段 0xFFFFFFFF：关闭 Flash 分区 0 保护 其他值：使能 Flash 分区 0 保护
0x11000610	ZONE0_FLASH_ADDR	Flash 分区 0 起始地址
0x11000614	ZONE0_RAM_PROT	RAM 分区 0 保护字段 0xFFFFFFFF：关闭 RAM 分区 0 保护 其他值：使能 RAM 分区 0 保护
0x11000618	ZONE0_RAM_ADDR	RAM 分区 0 起始地址
0x1100061C ~ 0x1100063F	Reserved for ZONE0	分区 0 保留配置字段
0x11000640	ZONE1_INFO_LOCK	锁定分区 1 的配置字（0x11000640 – 0x1100067F） 0xFFFFFFFF：分区 1 的配置字可以被编程 其他值：分区 1 的配置字不可以被编程
0x11000644	ZONE1_ENCRYPT	Flash 分区 1 代码的加密状态 0xDEC0DE：Flash 分区 1 的代码已被加密 其他值：Flash 分区 1 的代码是原始代码

地址	配置字名称	配置字描述
0x11000648	ZONE1_DECRYPT	Flash 分区 1 代码的解密状态 0xAA621623：成功解密 Flash 分区 1 的代码 0x1E55051：未成功解密 Flash 分区 1 的代码 其他值：无效值
0x1100064C	ZONE1_FLASH_PROT	Flash 分区 1 保护字段 0xFFFFFFFF：关闭 Flash 分区 1 保护 其他值：使能 Flash 分区 1 保护
0x11000650	ZONE1_FLASH_ADDR	Flash 分区 1 起始地址
0x11000654	ZONE1_RAM_PROT	RAM 分区 1 保护字段 0xFFFFFFFF：关闭 RAM 分区 1 保护 其他值：使能 RAM 分区 1 保护
0x11000658	ZONE1_RAM_ADDR	RAM 分区 1 起始地址
0x1100065C ~ 0x1100067F	Reserved for ZONE1	分区 1 保留配置字段
0x11000680	ZONE2_INFO_LOCK	锁定分区 2 的配置字(0x11000680 – 0x110006BF) 0xFFFFFFFF：分区 2 的配置字可以被编程 其他值：分区 2 的配置字不可以被编程
0x11000684	ZONE2_ENCRYPT	Flash 分区 2 代码的加密状态 0xDEC0DE：Flash 分区 2 的代码已被加密 其他值：Flash 分区 2 的代码是原始代码
0x1100688	ZONE2_DECRYPT	Flash 分区 2 代码的解密状态 0xAA621623：成功解密 Flash 分区 2 的代码 0x1E55051：未成功解密 Flash 分区 2 的代码 其他值：无效值
0x1100068C	ZONE2_FLASH_PROT	Flash 分区 2 保护字段 0xFFFFFFFF：关闭 Flash 分区 2 保护 其他值：使能 Flash 分区 2 保护
0x11000690	ZONE2_FLASH_ADDR	Flash 分区 2 起始地址
0x11000694	ZONE2_RAM_PROT	RAM 分区 2 保护字段 0xFFFFFFFF：关闭 RAM 分区 2 保护 其他值：使能 RAM 分区 2 保护

地址	配置字名称	配置字描述
0x11000698	ZONE2_RAM_ADDR	RAM 分区 2 起始地址
0x1100069C ~ 0x110006BF	Reserved for ZONE2	分区 2 保留配置字段
0x110006C0	ZONE3_INFO_LOCK	锁定分区 3 的配置字（0x110006C0 – 0x110006FF） 0xFFFFFFFF：分区 3 的配置字可以被编程 其他值：分区 3 的配置字不可以被编程
0x110006C4	ZONE3_ENCRYPT	Flash 分区 3 代码的加密状态 0xDEC0DE：Flash 分区 3 的代码已被加密 其他值：Flash 分区 3 的代码是原始代码
0x110006C8	ZONE3_DECRYPT	Flash 分区 3 代码的解密状态 0xAA621623：成功解密 Flash 分区 3 的代码 0x1E55051：未成功解密 Flash 分区 3 的代码 其他值：无效值
0x110006CC	ZONE3_FLASH_PROT	Flash 分区 3 保护字段 0xFFFFFFFF：关闭 Flash 分区 3 保护 其他值：使能 Flash 分区 3 保护
0x110006D0	ZONE3_FLASH_ADDR	Flash 分区 3 起始地址
0x110006D4	ZONE3_RAM_PROT	RAM 分区 3 保护字段 0xFFFFFFFF：关闭 RAM 分区 3 保护 其他值：使能 RAM 分区 3 保护
0x110006D8	ZONE3_RAM_ADDR	RAM 分区 3 起始地址
0x110006DC ~ 0x110006FF	Reserved for ZONE3	分区 3 保留配置字段
0x11000700	WDT_ENABLE	Watchdog 开机使能字段 0xFFFFFFFF：关闭 Watchdog 当芯片启动时 其他值：使能 Watchdog 当芯片启动时
0x11000704 ~ 0x110007FF	Reserved	未使用

3.3 功能实例

3.3.1 实例 1：设定时序参数

Flash 正常工作时需要满足一定的物理时序要求，用户可以调用 SetTiming 函数对 Flash 时序进行设置。

每当用户计划更改 HCLK 频率时，应按如下步骤：

1. 如果系统的工作频率要升高，用户需要先调用 FLASH_SetTiming 设置时序参数，再将系统工作频率升高。
2. 如果系统的工作频率要降低，用户需要先将系统工作频率降低，再调用 FLASH_SetTiming 设置时序参数。

3.3.2 实例 2：XIP 读操作

通过 XIP 模块，可以实现对 Flash 存储器的读操作。用户可以直接对 Flash 进行字节、半字或者字形式的读取。下面是一个示例代码，从 Flash 中地址 0x10000100 处读取一个字节数据。

示例代码

```
uint8_t *pu8Data = (uint8_t *)0x10000100;
uint8_t u8Byte;

u8Byte = *pu8Data;
```

3.3.3 实例 3：FLASH 控制器操作

3.3.3.1 功能需求

对 FLASH 进行读取、编程、擦除操作。

3.3.3.2 功能实现

1. 用户的代码一般是在 Flash 存储器中运行的，此时 XIP 模块是使能的。此时，通过 Flash 控制器对 Flash 存储器进行操作，涉及到 Flash 访问接口切换的问题，需要特别注意。为了能够使 Flash 访问接口切换时，程序能够正常工作，需要将 Flash 控制器的操作代码放在 SRAM 中运行。因此，通过 Flash 控制器对 Flash 存储器进行操作的过程如下：
   a) 程序从 Flash 中跳转到 SRAM 中，关闭 XIP 模块，使能 Flash 控制器；
   b) 通过 Flash 控制器执行相应的 Flash 操作；
   c) 关闭 Flash 控制器，使能 XIP 模块，程序重新跳转到 Flash 执行。
2. Flash 驱动函数中都帮用户实现了。用户在实际使用中只需要将 Flash 驱动函数编译到 SRAM 中运行即可，这一步可以通过 Keil 软件提供的 Scatter 文件实现。下面是通过 Flash 控制器。
3. 使用 SDK 提供的 FLASH 操作接口，对 FLASH 进行如下操作：

a) 擦除一个扇区 FLASH

b) 读取校验该扇区是否擦除成功

c) 向该扇区写入数据

4. 以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 3-3：

表 3-3：实例 3 代码路径

MCU 产品型号	代码路径
SPC2168，SPC2165，SPC2166，SPC1198	SDK 目录\0_Examples\ Flash_Operation

4 SPC1169 系列

4.1 特性

片内 Flash 存储器用于存储用户程序、数据等内容。

- 高达 128KB 存储空间
- 存储空间组成：
  - 主存储空间：共 32 扇区，每个扇区大小为 512*64 bits
  - NVR 空间：共 3 扇区，每个扇区 512*64 bits
- 最小编程单元为一个 Dword（64-bit）
- 单个扇区可擦除次数为：至少 100000 次@ T_J=85 ℃
- 数据保存时间期限：超过 20 年@ T_J=85 ℃
- ECC 保护
- 符合 AEC-Q100 一级标准

4.2 功能描述

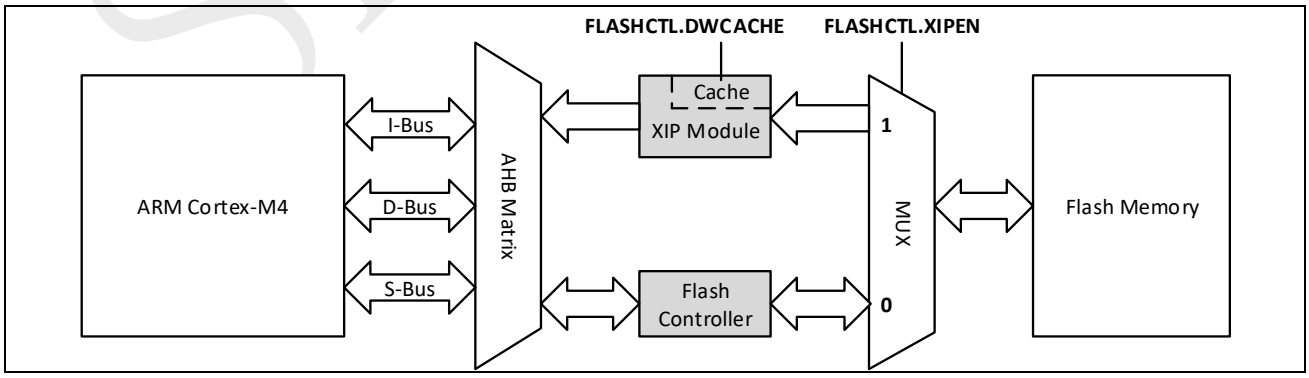
4.2.1 访问接口

FLASH 存储器的访问接口有两个：

- XIP 模块
- Flash 控制器，

如图 4-1 所示。XIP 模块只能实现从 Flash 存储器中读取数据，用于 CPU 从 Flash 存储器中取代码和数据；Flash 控制器是接在 AHB 总线上的一个外设，可以实现 Flash 存储器的读、写以及擦除操作。但是，这两个接口不能同时工作，同一时刻只能有一个接口处于工作状态。Flash 存储器当前的访问接口由寄存器 FLASHCTL.XIPEN 进行控制。

图 4-1：Flash 存储器访问接口



Flash 存储器的构成如图 4-2 和表 4-1 所示，包含两个部分：

- 主存储单元：这部分由 32 个扇区组成，每个扇区的大小为 4096 字节，共计 128KB；

– NVR 存储单元：这部分由 3 个扇区组成，每个扇区的大小为 4096 字节。

图 4-2：Flash 存储器逻辑结构

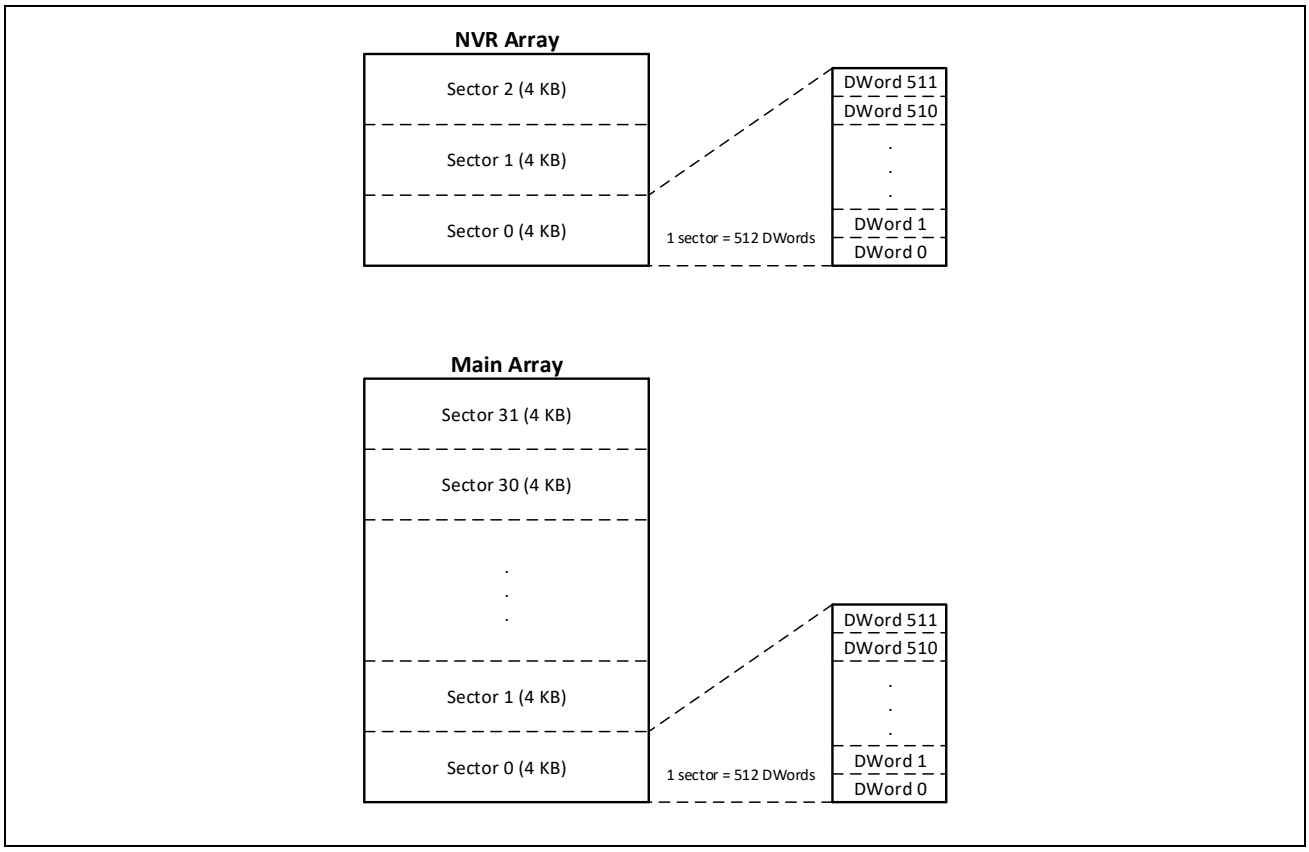


表 4-1：Flash 存储器结构

存储区块	名称	起始地址	大小
主存储单元	Sector 0	0x1000 0000	4096 字节
	Sector 1	0x1000 1000	4096 字节
	Sector 2	0x1000 2000	4096 字节
	Sector 3	0x1000 3000	4096 字节
	…	…	…
	Sector 31	0x1001 F000	4096 字节
NVR 存储单元	Sector 0	0x1100 2000	4096 字节
	Sector 1	0x1100 3000	4096 字节
	Sector 2	0x1100 4000	4096 字节

4.2.2 写保护

Flash 主存储区可以设置保护，以防止意外写入。如果对设置了保护的扇区执行编程或擦除操作，则操作将被忽略。

写保护通过设置 FLASHWP 寄存器对应比特位激活。

重置寄存器对应比特位将关闭写保护。

4.2.3 配置字

配置字可以通过 XIP 或者 FLASH 控制器接口进行读取，可以通过 FLASH 控制器进行编程、擦写。新写入的配置字需要在芯片复位并重新加载后才能生效。

下面部分表 4-2 详细概述：

表 4-2：配置字的构成及其描述

地址	名字	描述
0x1001FFF8	CHIP_SECURITY	锁住芯片调试接口配置字 0xFFFFFFFF：芯片调试接口将不会锁住 其它：芯片调试接口将会锁住
0x1001FFFC	WDT_ENABLE	看门狗使能配置字 0xFFFFFFFF：芯片启动时关闭看门狗 其它：芯片启动时使能看门狗

4.3 功能实例

4.3.1 实例 1：设定时序参数

Flash 正常工作时需要满足一定物理时序要求，用户可以调用 pHWLIB->FLASHC_SetTiming()函数对 Flash 时序进行设置。

每当用户计划更改 HCLK 频率时，应按如下步骤：

1. 在 HLK 频率更改之前调用 FLASHC_RelaxXIPTiming()，以宽松的时序访问 Flash。
2. 用户配置 HLK 频率
3. 在频率更改后，应调用 pHWLIB->FLASHC_SetTiming()函数以设置优化的 Flash 时序。

4.3.2 实例 2：XIP 读操作

通过 XIP 模块，可以实现对 Flash 存储器的读操作。用户可以直接对 Flash 进行字节、半字或者字形式的读取。下面是一个示例代码，从 Flash 中地址 0x10000100 处读取一个字节数据。

示例代码

```
uint8_t *pu8Data = (uint8_t *)0x10000100;
uint8_t u8Byte;

u8Byte = *pu8Data;
```

4.3.3 实例 3：FLASH 控制器操作

4.3.3.1 功能需求

对 FLASH 进行读取、编程、擦除操作。

4.3.3.2 功能实现

1. Flash 驱动代码固化存储于 ROM 空间。
2. 使用 SDK 提供的 FLASH 操作接口，对 FLASH 进行如下操作：
   a) 擦除一个扇区 FLASH
   b) 读取校验该扇区是否擦除成功
   c) 向该扇区写入数据
3. 以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 4-3：

表 4-3：实例 3 代码路径

MCU 产品型号	代码路径
SPC1169，SPD1179，SPD1176，SPD1163，SPM1173	SDK 目录\0_Examples\ Flash_Operation

5 SPC2188 系列

5.1 特性

片内 Flash 存储器用于存储用户程序、数据等内容。

- 高达 2MB 的存储空间
- 支持通过 QSPI1 以单线、双线输出、双线 I/O、四线输出、四线 I/O 模式访问；
- 支持串行接口时钟 SCLK 最高频率为 80MHz
- 2MB 主分区支持 ECC
  - 使能 ECC 时有效容量减半
- 分为 8192 个页面，每页大小为 256 字节（使能 ECC 时为 128 字节）；
- 最小编程单元为一个 Byte（8-bit）
- 每次可以按照以如下大小为单位擦除：
  - 4KB 扇区（使能 ECC 时为 2KB）：即 16 个页面
  - 64KB 块（使能 ECC 时为 32KB）：即 256 个页面
  - 整片：即所有空间
- 提供 1 个 256 字节的 OTP
  - 不支持 ECC

5.2 功能描述

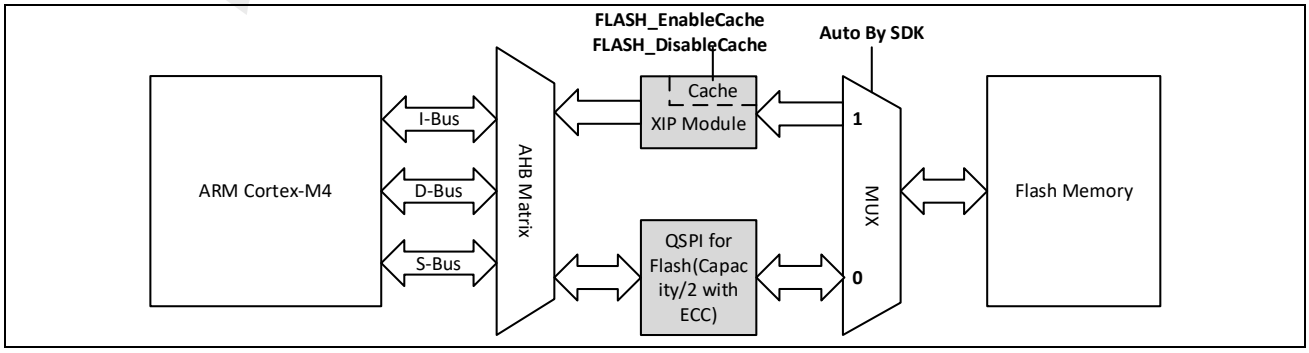
5.2.1 访问接口

FLASH 存储器的访问接口有两个：

- XIP 模块
- QSPI Flash 控制器

如图 5-1 所示。XIP 模块只能实现从 Flash 存储器中读取数据，用于 CPU 从 Flash 存储器中取代码和数据；Flash 控制器是接在 AHB 总线上的一个外设，可以实现 QSPI Flash 存储器的读、写以及擦除操作。但是，这两个接口不能同时工作，同一时刻只能有一个接口处于工作状态。芯片中固化的函数库提供了关于 FLASH 的各种操作的函数接口，且函数中已实现了 FLASH 存储器访问接口转换动作。

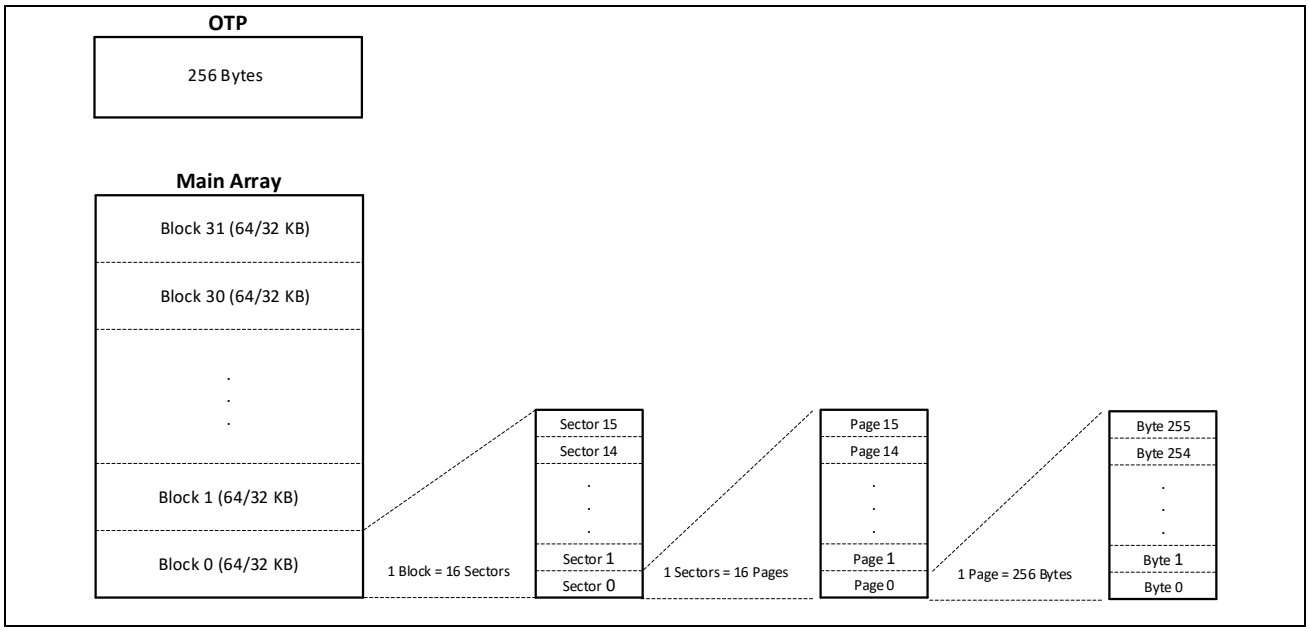
图 5-1：Flash 存储器访问接口



Flash 存储器的构成如图 5-2 和表 5-1 所示，包含两个部分：

- 主存储单元：由 8192 个页面组成，每页大小为 256 字节。
- OTP 存储单元：由 256 字节组成。

图 5-2：Flash 存储器逻辑结构



[1] 使能 ECC 后，存储器容量减半。

表 5-1：Flash 存储器结构

存储区块	名称	起始地址	大小
主存储单元	Page 0	0x1000 0100	256 字节
	Page 1	0x1000 0200	256 字节
	Page 2	0x1000 0300	256 字节
	Page 3	0x1000 0400	256 字节
	…	…	…
	Page 8191	0x101F FF00	256 字节
NVR 存储单元	OTP	0x1100 0000	256 字节

[1] 使能 ECC 后，存储器容量减半。

5.2.2 配置字

配置字可以通过 XIP 或者 FLASH 控制器接口进行读取，可以通过 FLASH 控制器进行编程、擦写。新写入的配置字需要在芯片复位并重新加载后才能生效。

下面部分表 5-2 详细概述：

表 5-2：配置字的构成及其描述

地址	名字	描述
0x100FFFF8	CHIP_SECURITY	锁住芯片调试接口配置字 0xFFFFFFFF：芯片调试接口将不会锁住 其它：芯片调试接口将会锁住
0x100FFFFC	WDT_ENABLE	看门狗使能配置字 0xFFFFFFFF：芯片启动时关闭看门狗 其它：芯片启动时使能看门狗

5.3 功能实例

5.3.1 实例 1：设定时序参数

Flash 正常工作时需要满足一定的物理时序要求，用户可以调用 pHWLIB->FLASHC_SetTiming()函数对 Flash 时序进行设置。

每当用户计划更改 HCLK 频率时，应按如下步骤：

1. 在 HLK 频率更改之前调用 FLASHC_RelaxXIPTiming()，以宽松的时序访问 Flash。
2. 用户配置 HLK 频率
3. 在频率更改后，应调用 pHWLIB->FLASHC_SetTiming()函数以设置优化的 Flash 时序。

5.3.2 实例 2：XIP 读操作

通过 XIP 模块，可以实现对 Flash 存储器的读操作。用户可以直接对 Flash 进行字节、半字或者字形式的读取。下面是一个示例代码，从 Flash 中地址 0x10000100 处读取一个字节数据。

示例代码

```
uint8_t *pu8Data = (uint8_t *)0x10000100;
uint8_t u8Byte;

u8Byte = *pu8Data;
```

5.3.3 实例 3：FLASH 控制器操作

5.3.3.1 功能需求

对 FLASH 进行读取、编程、擦除操作。

5.3.3.2 功能实现

1. Flash 驱动代码固化存储于 ROM 空间。

2. 使用 SDK 提供的 FLASH 操作接口，对 FLASH 进行如下操作：
   a) 擦除一个扇区 FLASH
   b) 读取校验该扇区是否擦除成功
   c) 向该扇区写入数据
3. 以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 5-3：

表 5-3：实例 3 代码路径

MCU 产品型号	代码路径
SPC2188，SPC1185	SDK 目录\0_Examples\ Flash_Operation