

使用 S32K148 FlexNVM 内存

作者: 恩智浦半导体

1 介绍

本应用笔记介绍了如何使用S32K148器件中 FlexNVM 存储器。所有 S32K1xx 设备都有一个 FlexNVM 分区, 但是 FlexNVM 在S32K148器件中具有不同的实现方式。因此, 需要进行具体考虑。有关 FlexNVM 的更多详细信息, 请参阅参考手册。本应用笔记重点介绍该存储器在 S32K148 中的不同用法, 以及每种配置和软件设计需考虑的因素。

该应用笔记会介绍软件设计需要考虑的点、常见处理错误和实际代码示例。

2 概述

2.1 词汇表

本节提供了整个文档中使用的术语的简要定义, 以帮助读者更好地理解文档。

表1. 词汇表

术语	描述
P-Flash	程序闪存, 可以执行程序代码的一个或多个 NVM 读取分区。
FlexNVM	可以执行程序代码、存储数据或备份模拟 EEPROM 数据的非易失性闪存。该存储器位于 P-Flash 的单独分区中。
D-Flash	数据闪存, 用于用户数据、引导代码和附加代码存储的非易失性闪存。DFlash 可以从 FlexNVM 块中分区以存储数据或执行代码。
E-Flash	非易失性闪存作为 FlexNVM 的一部分, 由闪存控制器 FTFC 用于模拟 EEPROM (EEPROM 备份)。

表格在下一页继续.....

目录

1 介绍.....	1
2 概述.....	1
2.1 词汇表.....	1
2.2 FlexNVM描述.....	2
3 用例.....	3
3.1 未启用 EEPROM 或 CSEc.....	4
3.2 EEPROM 使能.....	5
3.3 使能 EEPROM 和 CSEc.....	6
4 限制.....	6
5 软件建议.....	7
6 应用实例.....	8
7 参考资料.....	9



表1. 词汇表 (续)

术语	描述
EEE	闪存模块 (FTFC) 通过有效地提供 RAM 缓冲区 (FlexRAM) 作为用户/EEPROM 驱动程序的接口的虚拟 EEPROM 来模拟 EEPROM 的特性功能。使用内置硬件系统将数据自动编写到 FlexNVM 块中。
FlexRAM	RAM 可用作 SRAM 或高耐用性仿真 EEPROM 存储器。
EERAM	RAM 作为 FlexRAM 的一部分, 用于 EEPROM 仿真。
Bank	读取分区。
Sector	最小的可擦除 Flash 区域。
Section	可定义大小的编程/读取块命令。
Interleaved block	在 S32K 上, 两个64位宽的闪存块可以连接成一个128位宽的交错存储块。

2.2 FlexNVM描述

所有 S32K1xx 系列设备都有一个称为 FlexNVM 的部分。此部分可用作 Pflash (程序闪存)、Dflash (数据闪存) 和模拟 EEPROM 备份 (E-Flash)。在大多数系列器件中, 此部分的长度为 64KB。但是, S32K148 设备具有 512KB 长度的 FlexNVM 加上 1.5MB 的 Pflash, 总共 2MB。我们需要对 FlexNVM 给予特殊考虑。它可以被读取、编程、擦除、模拟 EEPROM 更新或用于 CSEc 加密操作, 但一次只能使用其中一个功能。[下面一部分](#)展示了 FlexNVM 的不同用例。

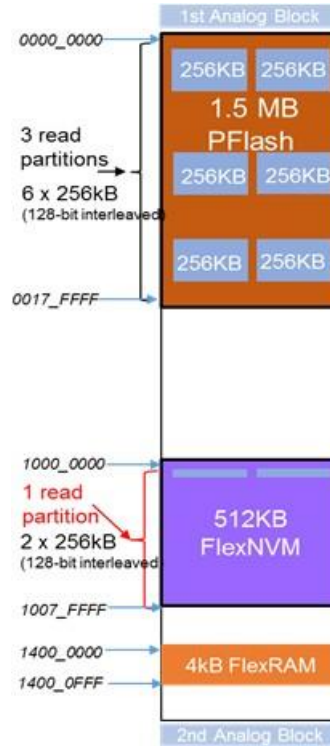


图1. S32K148 内存概览

注意

512KB FlexNVM 分区与 1.5MB PFlash 不相邻。这涉及链接器文件和编译属性的特殊注意事项，例如 FAR 属性。

3 用例

FlexNVM 部分可用于三种不同的配置：

- 未启用 EEPROM 或 CSEc。
- EEPROM 启用且 CSEc 禁用。
- EEPROM 和 CSEc 使能。

每种配置的主要特征以及 FlexNVM 部分的分区方式在以下各节中进行了描述。

3.1 未启用 EEPROM 或 CSEc

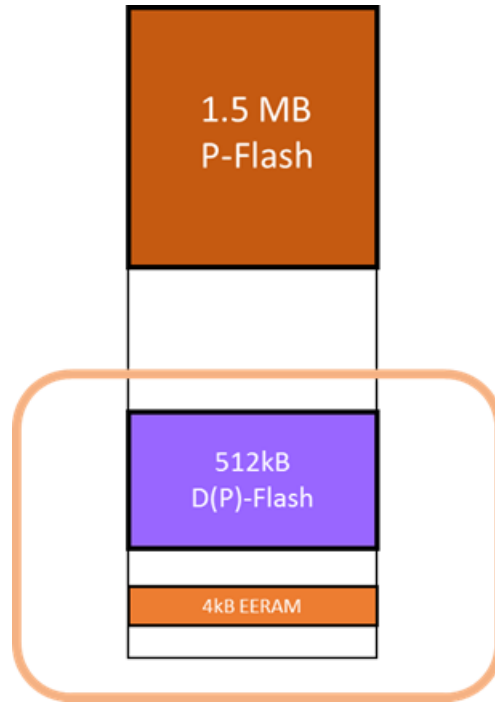


图2. FlexNVM 结构 (无 EEPROM 或 CSEc)

这种配置的主要特点是:

- 512kB FlexNVM 可用作 PFlash 或 DFlash。换言之，FlexNVM 可用于程序或数据存储。如果用作 PFlash，请记住 FlexNVM 部分不可缓存。因此，预计会出现一些性能下降。
- 4kB FlexRAM 可用作 SRAM，但它不像主 SRAM 那样具有 ECC，它以 Flash 闪存时钟的速度运行。
- 由于未启用 EEPROM 机制，FlexNVM 的耐用性将具有数据表中指定的 PFlash 存储器一样的特性。

何时使用此配置:

- 当数据闪存需要不经常更新数据并且需要超过 4KB（可用于 EEPROM 的最大大小）时，此配置非常有用。
- 如果需要超过 1.5MB 的 PFlash，此部分可用作 PFlash 和/或 DFlash，根据应用程序的需要进行划分。如果需要超过 2MB 的 PFlash，可以通过 QuadSPI 接口扩展外部存储器。

3.2 EEPROM 使能

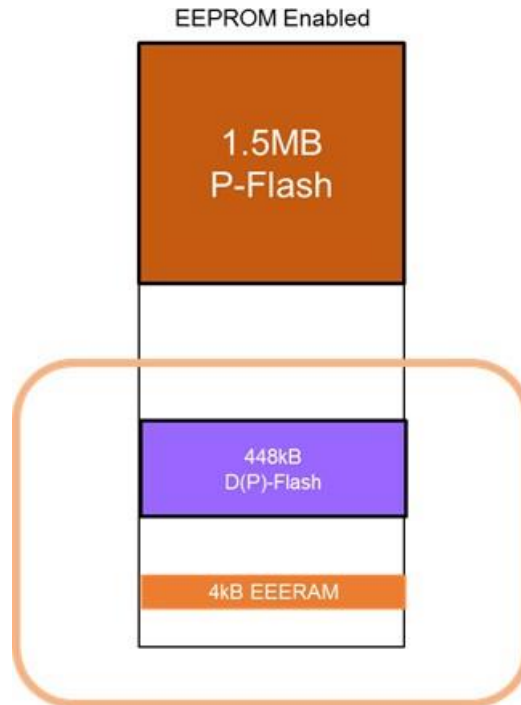


图3. FlexNVM 结构 (启用 EEPROM)

这种配置的主要特点是：

- 在此配置中，FlexNVM 的 **64KB** 用作 **E-Flash** (EEPROM 备份)，**448KB** 保留为 PFlash 或 DFlash。
- E-Flash 不是内存映射。在使能 EEPROM 后，无法以任何方式访问该存储器。
- 具有32位访问权限的 **4KB EERAM** 用于 **EEPROM** 模拟。

何时使用此配置：

- 当需要具有高耐久性 (**高达 100K 周期**) 的数据分区时，此配置非常有用。
- 非常适合频繁更新记录。例如：
 - 按键开/关更新

3.3 使能 EEPROM 和 CSEc

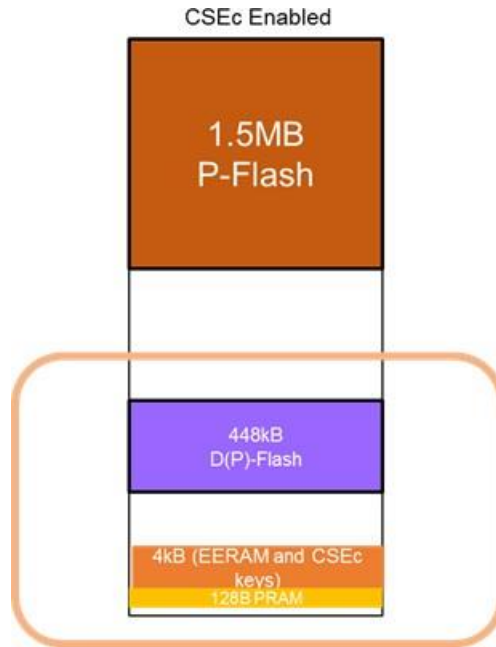


图4. FlexNVM 结构 (启用 CSEC)

这种配置的主要特点是：

- 在此配置中，FlexNVM 的 **64KB** 用作 **E-Flash (EEPROM 备份和密钥存储)**，**448KB** 保留为 PFlash 或 DFlash。
- E-Flash 不是映射存储器，即使在使能 EEPROM 后，无法以任何方式访问该存储器。
- 要使用CSEc 操作，**需要启用EEPROM**，因为E-Flash 也用于保存CSEc 数据。
 - 例如 密钥
- CSEc 使用多达 **512 字节**的密钥存储，EEPROM 使用剩余的 **3.5KB** EERAM

何时使用此配置：

- 当执行CSEc 操作时需要此配置。

4 限制

下图显示了在不同闪存扇区中同时操作的限制：

		Pflash			FlexNVM			FlexRAM			CSEc
		Read	Program Phase	Erase Flash Sector	Read	Program Phase	Erase Flash Sector	Read	EEPROM Write	RAM write	Any command
Program Flash	Read				Allowed	Allowed		Allowed		Allowed	
	Program Phase				Allowed			Allowed		Allowed	
	Erase Flash Sector				Allowed			Allowed		Allowed	
FlexNVM	Read		Allowed	Allowed							
	Program Phase	Allowed						Allowed		Allowed	
	Erase Flash Sector	Allowed						Allowed		Allowed	
FlexRAM	Read		Allowed	Allowed		Allowed	Allowed				
	EEPROM Write	Allowed									
	RAM Write		Allowed	Allowed		Allowed	Allowed				
CSEc	Any command	Allowed	Allowed	Allowed							

图5. Flash 扇区同时操作

从图 5 中需要注意，在 512KB FlexNVM 分区上只能执行一种 FlexNVM 操作。这些操作是：

- 读
- 编程
- 擦除
- EEPROM写入
- CSEc.

因此，我们需要考虑软件方面的问题。下一节将讨论这些注意事项。

5 软件建议

用户必须确保遵循这些建议以避免 FlexNVM 出现问题。FlexNVM 中的任何权限问题都会触发冲突，并且使正在执行的操作将无效。本节稍后将更详细地解释冲突处理。

用户需要遵循的建议是：

- 不得将任何使用 CSEc、EEPROM（只写）或闪存控制器命令的软件驱动程序放置在 FlexNVM 的 PFlash 中。例如：
 - 加密/解密 CSEc 命令。
 - 存储校准信息。
- 在 CSEc 或 EEPROM 写或编程/擦除操作期间必须读取的任何配置数据（常量参数）不得放置在 FlexNVM 的 Dflash 中。
- 任何与必须在 CSEc 或 EEPROM 写或编程/擦除操作期间提供服务的中断相关的 ISR 不得放置在 FlexNVM 的 Pflash 中。同样的限制适用于从 ISR 调用的函数。

如果满足以下条件，则可以消除大多数先前的约束：

- 在 FlexNVM 操作期间禁止所有中断。
- 使用 FlexNVM 的软件 API 正在同步工作（在请求的操作完成后返回）。

另一个建议是在启动任何涉及 CSEc、EEPROM（仅写入）或闪存控制器命令的操作之前轮询 CCIF 标志。以下例程是如何轮询 CCIF 的示例：

```
/* The following routines writes data into EEPROM */
voidEEPROM_write( uint32_t data){
    uint32_t *FlexRAMptr = ( uint32_t *) 0x14000000U; /* Pointer to FlexRAM
memory section */
    while ((FTFC-> FSTAT & FTFC_FSTAT_CCIF_MASK) == 0){} /* Wait for anyFlexNVM
operation to be finished */
    *FlexRAMptr = 0x00AA00BB; /* Writes into EEPROM */
}
```

上面的函数是向EEPROM写入存储器。可以看出，在写入该部分之前，它会轮询 CCIF 标志以确保没有先前的操作在运行。如果一个操作正在运行，它将一直等到操作完成（CCIF 标志设置为 1）。

如果用户跳过这些建议之一并导致同时访问，该事件将导致冲突（通过 FTFC 读取冲突错误标志 RDCOLERR 报告和中断）。这种冲突总是会触发硬故障错误。如果用户使能读取冲突错误标志中断，则在处理完硬故障后会触发中断。

以下例程是必须如何处理这些错误的示例：

```
voidHardFault_Handler(void){
    /* Handling of error must be
placed here.          * an error condition flag can
be set              * to trigger some recovery
routine */
}

voidRead_Collision_IRQHandler(void){
    FTFC-> FSTAT |= FTFC_FSTAT_RDCOLERR_MASK; /* clear interruptflag */
    while ((FTFC-> FSTAT & FTFC_FSTAT_CCIF_MASK) == 0){} /* wait for collision
condition to be done */
    /* any other routine that handles the issue */
}
```

最重要的是要记住，如果发生碰撞，程序总是先跳转到 Hardfault 硬件故障程序。然后，如果读冲突错误标志中断使能，程序将在硬故障程序之后处理读冲突中断服务程序。这些只是示例，用户可以使用这些示例作为自己应用程序开发的任何恢复/纠正机制。但是请记住，不建议发生冲突，因为它会使之前在 FlexNVM 中执行的操作无效。

6 应用实例

本节提供了一些可以从 FlexNVM 的 PFlash 部分存储或执行的软件块或驱动程序示例。但是，这将始终取决于用户的应用场景。只要不违反上述条件，用户可以使用此部分来满足其应用程序的任何需要。

- 每个不需要 EEPROM（写入）、CSEc 或 Flash 命令的外设配置/初始化代码。
- 引导加载程序代码（假设闪存命令是从 RAM 启动的）。
- 通用应用程序级代码。
- 中间件/堆栈。
- 后处理算法。
- 核心自检。
- 常量数据（如果需要在启动时复制到 RAM 或在 FLEXNVM 操作期间未使用）。

7 参考资料

- 恩智浦. [S32K1xx 数据表](#). 2017。
- 恩智浦. [S32K14x 系列参考手册](#). 2017。

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and μ Vision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ARM7, ARM9, ARM11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2017 NXP B.V.

Document Number: AN12003
Rev. 0, July 2017

