

F-RAM™ 作为代码和数据存储器应用中的单芯片解决方案

作者: Harsha Medu

相关项目: 无

相关器件系列: 无

软件版本: 无

AN101 讨论了代码和数据对存储器的要求, 以及以便能在单芯片 F-RAM 器件中将它们结合起来的设计中的注意事项。

概述

通常将采用存储器技术的应用分为可执行代码和数据任务。可执行代码要求使用非易失性存储器, 并在所有条件下都能够保留存储器中的代码。数据任务则要求能够对存储器进行快速、简易且无限次数的读/写访问。会根据应用程序来决定执行数据任务的存储器是易失性的还是非易失性的。通常, 可执行代码的存储器采用基于 ROM 的技术, 而数据任务的存储器则采用基于 RAM 的技术。作为赛普拉斯中的一款非易失性 RAM 产品, F-RAM 提供了独特的优点 — 它能够将在可执行代码和数据组合在一个单存储器中。

本应用笔记对代码和数据存储器在两种情况下的要求进行了比较: 分别单独放置代码和数据; 采用将代码和数据组合存储在单个系统存储器内的解决方案。本应用笔记还推荐了一个设计解决方案, 用以防止在组合代码-数据存储器应用中意外覆盖掉 F-RAM 中的代码部分。

代码存储对存储器的要求

- 非易失性
- 相应的容量
- 读取访问时间
- 防止意外写入的能力
- 在某部分或某段代码中的现场可编程能力
- 编程的同时执行读访问的能力

代码存储的基本要求是: 应该是非易失性存储器, 并在所有条件下都能够保持它的状态。代码存储器是只读的, 并且在运行过程中不会更新存储在该存储器中的内容。

保持代码所需的存储器空间由应用程序设置。通常, 额外存储器空间中的 20%-30% 提供给将来系统修改使用。

代码执行时间是由存储器的读访问时间决定的; 访问时间越短, 对控制器开销的影响越小。

代码存储器是只读的。应防止发生任何意外的写入操作, 因为该操作会引起应用故障。不过在几种情况下, 代码存储器将被覆盖。周期性更新设计以增加新特性或纠正硬件错误便是一种情况。现场重新编程代码存储器能力是一项主要的优势和功能。通常, 重新编程只会影响某部分或某段代码, 并不能影响整个代码。执行重新编程期间, 写访问时间是决定重新编程器件所需时间的一个因素。

代码存储器包含实现擦除/编程功能的代码, 该代码负责实现预先计划的现场可编程能力。在现场编程过程中, 要确保能够对存储器的字节或寄存器的某些部分同时执行读/写访问, 以避免将擦除/编程代码复制到其他存储器 (RAM) 内并从这些存储空间执行代码所引起的不必要麻烦。

NAND 闪存是代码存储器的常用解决方案, 但在某些应用程序 (它的代码规模比较小, 并且应用程序需要读/写访问时间 (数据记录、能量采集或使用寿命长的电池等操作) 对称) 中, 更应该选择 F-RAM。

数据存储对存储器的要求

- 快速写访问
- 允许大量的写入次数
- 简单的写协议 (最好没有协议)
- 按字节寻址的写入

- 非易失性（在某些应用程序中）
- 满足易失性和非易失性要求的能力

在许多方面，数据存储对存储器的要求与代码存储的要求完全相反。数据存储是一个更为多样化的任务，要求灵活性和易于写访问等特性。通常，数据存储应用要求存储器能够进行快速的读/写访问，根据数据或应用的类型，它可以是非易失性或易失性存储器。

由于不同应用的需求，数据存储往往要进行多次写入操作。一些是周期性的写入，一些则是基于事件的写入操作。数据存储所需的存储器要有快速写访问能力，从而进行快速存储数据。因此，数据存储应能够进行无限次数的写入和快速访问。由于需要进行大量的数据传输，因此它还要具备简单且容易访问的能力。

按字节写入存储器是数据存储的另一重要方面。因为数据可能包含某个变量要多次更新，选择按字节写入的访问比较合适（不影响其他内容）。

有些应用程序要求数据具有非易失性，以便在掉电-上电时仍能保存应用程序的设置情况。用户可以灵活地修改设置并将其保存，无需执行任何复杂的操作。

在易失性应用程序中，数据存储器件通常采用了 RAM 技术。而在非易失性的应用程序中，数据存储则成为主流存储器技术的一个挑战。基于 ROM 技术的器件（例如，闪存）不是最佳的数据存储器，因为不能对这些存储器进行灵活写入，并且这些存储器比较适合静态配置。EEPROM 便是非易失性数据存储器的另一种选择。不过，它的写入时间很长（与 F-RAM 的纳秒相比，EEPROM 的写入时间以毫秒为单位），在写入寿命内，它允许写入的次数比较少。

F-RAM 针对非易失性数据存储进行了优化。以读取速度进行 F-RAM 写入。F-RAM 是一款非易失性存储器，它提供了较多的访问次数（ $>10^{14}$ ）。对该存储器进行写入操作不需要任何特别的算法或协议，它是按字节寻址的。因此，F-RAM 提供了数据存储最灵活的解决方案。

单芯片代码和数据要求

- 带有快速读/写访问能力的非易失性存储器
- 大量的写入次数
- 防止意外写入存储器模块的能力

- 更改代码和数据存储器大小的灵活性

将代码和数据组合在一个单个器件中，这样只需要一个存储器仍能够给代码和数据提供常常会互相排斥的服务。

代码存储器要求非易失性存储和支持偶尔写入的能力。升级代码时，不需要进行多次写入操作（周期）。通常，无需关注写入时间。在某些方面，存储器写入代码越难越好，因为意外写入会引起严重的破坏。不过，写访问时间越短，运行代码则越快。

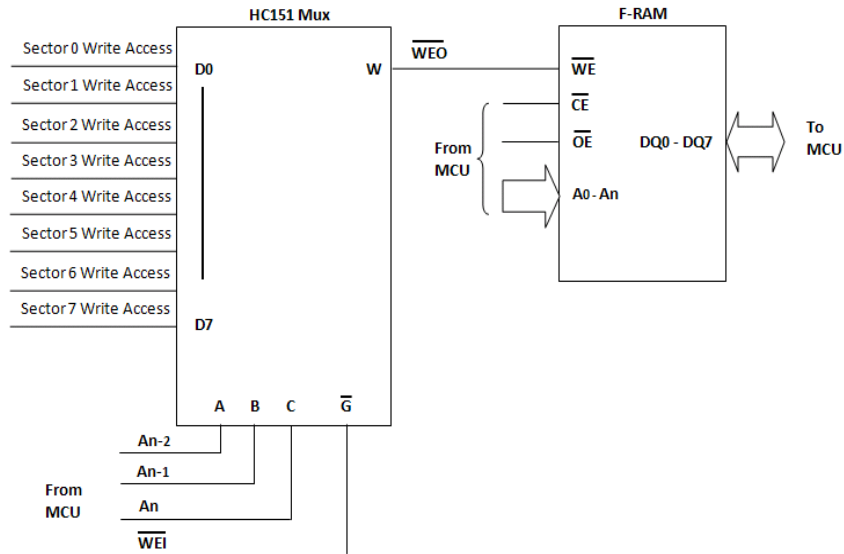
通常，数据存储需要结合使用易失性和非易失性存储器，或至少是能够无限制写入的非易失性存储器。它需要相当的读/写次数，而且读/写访问时间也是一个重要的因素。它还要求能够进行大量的写入次数（最好是无限次的）。代码存储器很难同时满足上述要求。一个单芯片解决方案应该足够灵活才能满足这些条件。

在某些应用中（如数据记录器），测量和收集数据所需的代码大小可能会小于存储数据的存储器大小。如果只有一个单独的代码存储器，那么可能无法充分利用该存储器。在这种情况下，单芯片解决方案便是更好的选择。该解决方案要求足够灵活，以满足每个应用程序对代码和数据存储器不同的要求。

当前的 F-RAM 是针对数据存储的应用进行优化的。F-RAM 具有更好的写性能使之比闪存或 EEPROM 更具优越性，F-RAM 的非易失性特点则使之比 RAM 更好。因为它是非易失性的，所以 F-RAM 可作为代码存储器使用。在单芯片应用程序中使用 F-RAM 的主要限制是大小受限。目前，赛普拉斯的并行和串行的 F-RAM 的最大容量均为 4 Mbit。在需要代码尺寸更小和数据尺寸更大的应用程序（特别是电池供电或能量采集的应用程序）中，可选择 F-RAM 作为代码和数据的单芯片解决方案。

当 F-RAM 用于代码存储时，必须保证系统不会意外对存储器的一个区域（正在用于执行代码）进行写操作。通过创建一个简单的写保护电路，F-RAM 便能够很容易解决该问题。可通过该逻辑来防止无意对代码区域（或数据区域，如果需要）执行写操作。图 1 显示的是一个示例电路。它为 F-RAM 或任何其他 RAM 解决方案提供了可编程模块写保护性能。如果固定连接扇区写访问输入，会创建一个固定的模块写保护原理图；如果将这些输入连接到逻辑或微控制器，则会提供动态更改写保护性能的方法。

图 1. 扇区写保护电路



写保护原理图的操作

图 1 中的电路采用了普通的 CMOS 复用器（如 HC151）来创建基于地址的写使能（可直接连接到 F-RAM 器件）。HC151 的真值表显示在表 1 内。

表 1. HC151 真值表

| 选择输入 | | | 选通: \overline{G} [WEI] | \overline{W} [WEO] |
|-----------|-------------|-------------|-----------------------------|-------------------------|
| C [An] | B [An-1] | A [An-2] | | |
| X | X | X | H | H |
| L | L | L | L | $\overline{D0}$ |
| L | L | H | L | $\overline{D1}$ |
| L | H | L | L | $\overline{D2}$ |
| L | H | H | L | $\overline{D3}$ |
| H | L | L | L | $\overline{D4}$ |
| H | L | H | L | $\overline{D5}$ |
| H | H | L | L | $\overline{D6}$ |
| H | H | H | L | $\overline{D7}$ |

将扇区的写访问值设置为 1 可允许向该扇区执行写入操作。将扇区的写访问值设置为 0 可阻止向该扇区执行写入操作。

微控制器的写输入信号将被标注为 \overline{WEI} ，它连接着 HC151 复用器的 \overline{G} 选通。HC151 复用器的写输出信号（连接到 F-RAM）被标注为 \overline{WEO} 。如果 \overline{WEI} 信号为高电平，则 HC151 的 \overline{WEO} 输出会保持为高电平状态。如果 \overline{WEI} 变为低电平，那么当前地址和相应的写访问输入则决定了连接到 F-RAM 的 \overline{WEO} 能否变为低电平，如该章节后面所介绍的。

假如三个最高有效存储器地址行（An、An-1、An-2）都连接着 HC151，那么存储器空间将被分为八个大小相等的扇区。对于每个扇区，HC151 的 Dn 输入决定了该扇区是否被写保护。输入的地址会根据三个最高有效地址行选择八个扇区中的一个。当相应的 Dn 输入为高电平时，如果 \overline{WEI} 为低电平，那么 HC151 的 \overline{WEO} 将变为低电平。因此，可以对这些扇区进行写入操作。如果选定的 Dn 为低电平，那么对于该扇区中的所有地址， \overline{WEO} 输出均保持为高电平状态，无论 \overline{WEI} 输入的状态如何。

该简单的电路允许系统能够向任何 RAM 存储器（包括 F-RAM）添加可编程模块写保护功能。一个选项是：根据代码和数据的位置为写访问设置固定数值。这样做可防止意外执行写入操作，但消除了实现代码的现场升级（无需更改电路板）的可能性。另外一个选项与上述选项不同，它使用了跳线连接。这样允许进行修改，但仍需要用户的干预。一项更加灵活的选择是：将写访问的设置连接到微控制器上（可以通过其他逻辑进行）。如果这些输入的上电复位状态被置为低电平，那么在默认条件下用于对整个器件进行写保护。在软件控制下，系统可以更改这些设置，以便向特定扇区提供有限/无限写的访问功能或向所有扇区提供动态打开/关闭访问功能。该原理图允许对重要的数据区域和代码进行写保护。

串行 F-RAM 中的写保护

还可以将串行 F-RAM 作为代码和数据存储器。大部分 SPI F-RAM 都有模块保护特性。使用该特性，代码存储器被写保护。它可以对前四分之一、前半部分或整个存储器进行保护。在状态寄存器中定义了模块保护位（BP1 和 BP0）。表 2 提供模块保护的详细信息。

表 2. 模块存储器写保护

| BP1 | BP0 | 受保护的区域 |
|-----|-----|----------|
| 0 | 0 | 无 |
| 0 | 1 | 前四分之一存储器 |
| 1 | 0 | 前半部分存储器 |
| 1 | 1 | 整个存储器 |

串行 F-RAM 还定义了硬件写保护引脚，用于防止无意写入的额外安全工作。

总结

AN101 介绍了代码、数据和两者对存储器的要求。它还说明了采用目前几种解决方案（如闪存和 EEPROM 与 F-RAM 相比）各自的优点和缺点。它还提供了将代码和数据组合在一个 F-RAM 器件中的设计注意事项。由于 F-RAM 具有的简单系统设计采用了写保护电路和快速存储器访问，对于代码和数据，它明显是更好的单芯片解决方案。

文档修订记录

文档标题: F-RAM™作为代码和数据存储器应用中的单芯片解决方案 — AN101

文档编号: 001-96570

| 修订版本 | ECN | 变更者 | 提交日期 | 变更说明 |
|------|---------|------|------------|---------------------------------------|
| ** | 4710811 | YLIU | 04/02/2015 | 本文档版本号为 Rev**, 译自英文版 001-87060 Rev*A。 |

全球销售和设计支持

赛普拉斯公司拥有一个由办事处、解决方案中心、厂商代表和经销商组成的全球性网络。如果想要查找离您最近的办事处，请访问赛普拉斯所在地。

产品

| | |
|---------|--|
| 汽车级产品 | cypress.com/go/automotive |
| 时钟与缓冲器 | cypress.com/go/clocks |
| 接口 | cypress.com/go/interface |
| 照明和电源控制 | cypress.com/go/powerpsoc cypress.com/go/plc |
| 存储器 | cypress.com/go/memory |
| PSoC | cypress.com/go/psoc |
| 触摸感应 | cypress.com/go/touch |
| USB 控制器 | cypress.com/go/usb |
| 无线/射频 | cypress.com/go/wireless |

PSoC®解决方案

psoc.cypress.com/solutions
PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

赛普拉斯开发者社区

[社区](#) | [论坛](#) | [博客](#) | [视频](#) | [培训](#)

技术支持

cypress.com/go/support

此处引用的所有商标或注册商标归其各自所有者所有。

| | | |
|---|--|---|
|  | 赛普拉斯半导体 198 Champion Court San Jose, CA 95134-1709 | 电话 : 408-943-2600 传真 : 408-943-4730 网址 : www.cypress.com |
|---|--|---|

©赛普拉斯半导体公司，2013-2015。此处，所包含的信息可能会随时更改，恕不另行通知。除赛普拉斯产品内嵌的电路外，赛普拉斯半导体公司不对任何其他电路的使用承担任何责任。也不根据专利或其他权利以明示或暗示的方式授予任何许可。除非与赛普拉斯签订明确的书面协议，否则赛普拉斯不保证产品能够用于或适用于医疗、生命支持、救生、关键控制或安全应用领域。此外，对于可能发生运转异常和故障并对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统中，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

该源代码（软件和/或固件）均归赛普拉斯半导体公司（赛普拉斯）所有，并受全球专利法规（美国和美国以外的专利法规）、美国版权法以及国际条约规定的保护和约束。赛普拉斯据此向获许可者授予适用于个人的、非独占性、不可转让的许可，用以复制、使用、修改、创建赛普拉斯源代码的派生作品、编译赛普拉斯源代码和派生作品，并且其目的只能是创建自定义软件和/或固件，以支持获许可者仅将其获得的产品依照适用协议规定的方式与赛普拉斯集成电路配合使用。除上述指定的用途外，未经赛普拉斯明确的书面许可，不得对此类源代码进行任何复制、修改、转换、编译或演示。

免责声明：赛普拉斯不针对此材料提供任何类型的明示或暗示保证，包括（但不限于）针对特定用途的适销性和适用性的暗示保证。赛普拉斯保留在不做通知的情况下对此处所述材料进行更改的权利。赛普拉斯不对此处所述之任何产品或电路的应用或使用承担任何责任。对于可能发生运转异常和故障，并对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统中，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

产品使用可能受适用于赛普拉斯软件许可证的限制。