

# 使用 Kinetis 系列中的增强型 EEPROM 功能

作者: Melissa Hunter 和 Derrick Klotz

## 内容

## 1 简介

某些 Kinetis 系列处理器包含 FlexMemory 功能。FlexMemory 可让用户将某些片上闪存配置为增强型 EEPROM、附加闪存或此两者的组合。

本应用说明介绍了如何将 FlexMemory 用作增强型 EEPROM。另外,还介绍了增强型 EEPROM 的优势、存储器的配置以及耐受能力计算方法。

1	简介.....	1
2	增强型 EEPROM (EEE)功能.....	1
3	使用 EEE.....	11
4	EEE 性能.....	13
5	掉电检测.....	13

## 2 增强型 EEPROM (EEE)功能

FlexMemory 的增强型 EEPROM (EEE)提供了许多功能让您更换外部 EEPROM 及改进其性能。

FlexMemory EEE 的功能包括:

- 可配置,使设计人员能够根据所需的 EEE 耐受能力和所需的 EEE 与闪存总量做出权衡利弊的决策。
- 高耐受能力的 EEE 存储器的擦写周期数可达 1 亿次以上(取决于配置)
- 支持字节、字和长字存取
- 写入时间很短(写入预先擦除的位置时大约为 100  $\mu$ s,擦除和编程时为 1.5 ms)
- 在处理器接受的整个工作电压范围(1.71 V – 3.6 V)内都可以使用 EEE 功能

## 2.1 FlexMemory EEE 的工作原理

为了提供增强型 EEPROM 功能，FlexMemory 使用了 RAM 块(FlexRAM)、闪存块(FlexNVM)和 EEE 状态机。启用 EEE 功能后，FlexRAM 将变成 EEE 存储器。您可以在 FlexRAM 地址空间中存取所有 EEE 数据。存取 EEE 时，EEE 状态机将会跟踪数据，并将它作为数据记录进行备份，这些数据记录存储在用作 E-Flash 的某个 FlexNVM 部分中。如果使用较大的 E-Flash 块备份少量的 EEE 数据，则 FlexMemory EEE 器件可以提供极高的耐受能力。

EEE 状态机使用 32 位记录将数据从 EEE 备份到闪存中。记录的 16 个位用于数据，另外 16 个位是有关数据的地址和状态信息。可以根据需要写入和擦除数据记录。也就是说，如果从未存取过 EEPROM 中的某个位置，则该位置将不会有任何数据记录。这有助于减少需要备份的数据量，并可以提高存储器的耐受能力。

## 2.2 FlexMemory EEE 如何使用存储器

支持 FlexMemory 的设备具有两个独立的闪存块，即 P-Flash 块和 FlexNVM 块。P-Flash 块旨在用作编程闪存块，但实际上可以使用它来存储指令和数据。FlexNVM 块是可配置的闪存块，可用作附加闪存空间(D-Flash)、用于支持增强型 EEPROM 功能的备份存储器(E-Flash)，或此两者的组合。

### 注

未用作 EEE 备份存储器(E-Flash)的 FlexNVM 部分称为 D-Flash。通常，此闪存用于数据存储空间；但是，像 P-Flash 一样，D-Flash 实际上也可用于存储指令或数据。

FlexMemory 包括 FlexNVM 块、FlexRAM 和 EEE 状态机。要支持 EEE 功能，需要提供所有这 3 个块。

[图 1](#) 显示了配有 FlexMemory 的 256 KB 设备上的闪存块和 FlexMemory 组件。在本例中，设备未分区，因此，未将 FlexNVM 块指定为 D-Flash 或 E-Flash。

### 注

Kinetis 系列设备部件号仅指示了 P-Flash 块的大小，因此，部件号中未明确列出 FlexNVM 的大小。例如，MK60X256 设备具有 256 KB 的 P-Flash 和 256 KB 的 FlexNVM。

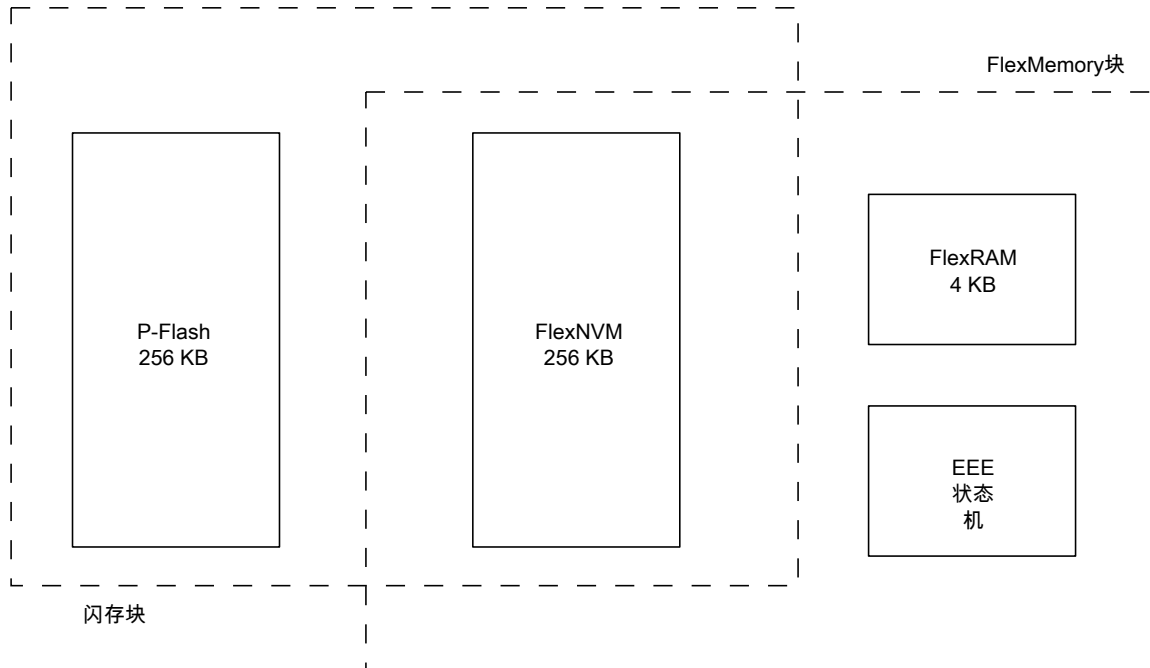


图 1. 装有 FlexMemory 的 256 KB 设备在分区之前的情况

## 2.2.1 禁用了 EEE 功能的 FlexMemory

图 2 显示了在支持 FlexMemory 的处理器上禁用 EEE 功能时存储块的工作方式。

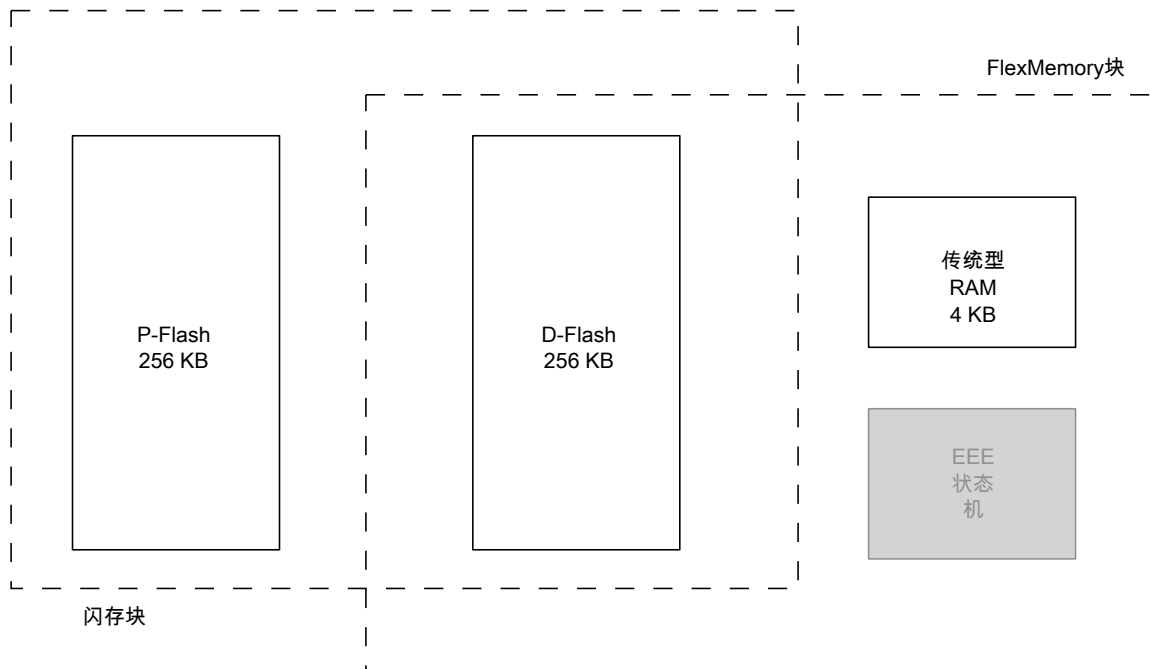


图 2. 装配禁用了 EEE 的 FlexMemory 的 256 KB 设备

P-Flash 存储器始终为 P-Flash。不管采用哪种 FlexMemory 配置，它的功能都是相同的。由于本例中未使用 EEE 功能，因此已将整个 FlexNVM 分配为 D-Flash 空间(无需 E-Flash)。FlexRAM 已变成 4 KB 传统 RAM。也就是说，可以将它用作附加的存储空间，但请注意，它是以闪存时钟速度而不是内核速度运行的(紧耦合存储器 RAM (TCM)以内核速度运行)。EEE 状态机在设备中存在，但处于非活动状态。

## 2.2.2 启用了 EEE 功能的 FlexMemory

图 3 显示了当已启用 EEE 功能并使用整个 FlexNVM 来备份 EEE 数据时存储块的工作方式。

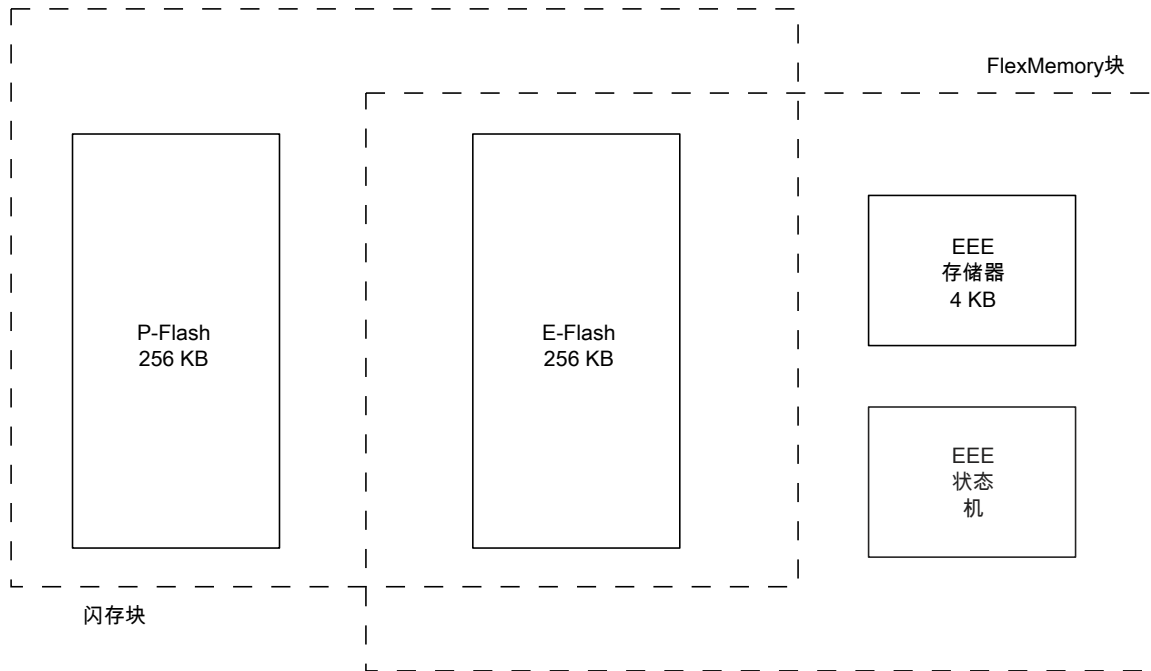


图 3. 装配启用了 EEE 的 FlexMemory 的 256 KB 设备(使用整个 FlexNVM 备份 EEE 数据)

启用 EEE 功能后,可以使用许多的配置选项。图 3 显示了将整个 FlexNVM 用作 E-Flash 存储器的示例。FlexRAM 已变成 EEE 存储空间(最高 4 KB)。任何读取和写入 EEE 数据的操作都将使用此 4 KB 存储空间,因为 E-Flash 不可直接存取。EEE 状态机将自动管理对 EEE 存储空间所做的全部写入操作,并根据需要在 E-Flash 中生成闪存编程和擦除操作。

### 2.2.2.1 E-Flash 和 EEE 存储器配置详细信息

图 3 显示了存储块的总体结构,同时还显示了可配置的存储块的相关功能。本部分详细介绍了如何实际使用 E-Flash 和 FlexRAM 块创建 EEE 子系统,并提供了一些不同配置的示例。

#### 2.2.2.1.1 EEE 配置参数

FlexMemory 允许使用许多不同的存储器配置选项。EEE 数据量以及用于备份这些 EEE 数据的 E-Flash 存储量都可编程。这样,您便可以在存储器大小与 EEE 耐受能力之间作出取舍。

可以使用 3 个可编程的选项来定义系统使用的确切存储量。这些参数为:

1. EEE 大小 - 所需的 EEE 数据总大小。EEE 总大小可以是 32 字节到 4 KB,可以使用 2 的幂次方(32、64、128...)为增量编程。

2. EEE 分割 - EEE 状态机将 EEE 视为两个独立的子系统(子系统 A 和子系统 B)。始终使用相同的 E-Flash 存储量备份每个子系统, 但 EEE 存储器的大小并不一定相同。可以使用分配给子系统 B 的余数将子系统 A 编程为 EEE 总大小的 1/8、1/4 或 1/2。如果将 EEE 分割编程为 1/2, 则在两个子系统之间平均分割 EEE 存储量, 因此, 将使用相同的 E-Flash 量来备份相同的 EEE 数据量。这样, 便可以建立两个具有相同耐受能力的子系统。如果将 EEE 分割编程为 1/8 或 1/4, 则子系统 A 使用的 E-Flash 量与子系统 B 相同, 但是, 前者现在备份的 EEE 数据更少。EEE 数据量与备份空间之间的这种比例变化意味着子系统 A 的耐受能力高于子系统 B。使用 EEE 分割参数可以将数据划分到高耐受能力和中等耐受能力区域, 而标准闪存空间(P-Flash 或 D-Flash)将用作标准耐受能力区域。
3. FlexNVM 分区 - 此参数定义要将多少 FlexNVM 用作标准闪存(D-Flash), 以及要将多少 FlexNVM 用作 EEE 备份存储器(E-Flash)。如果使用 EEE, 则必须至少分配 32 KB 的 FlexNVM 用作 E-Flash。为了在 EEE 中尽可能地获得最高耐受能力, 可以将整个 FlexNVM 用作 E-Flash。

### 2.2.2.1.2 计算 EEE 耐受能力

在许多应用中, EEE 数据的耐受能力要求使得用户必须对 FlexMemory 的配置方式做出合理的决策。通常, 用户需要了解各种参数对耐受能力会造成怎样的影响, 以便决定使用哪种 EEE 配置。

可以使用以下公式计算 EEE 的估计最低耐受能力:

$$Endurance = \frac{EFlash - (2 \times SPLIT \times EEESIZE)}{SPLIT \times EEESIZE} \times RecordEfficiency \times BaselineEndurance \quad Eqn. 2-1$$

其中:

- E-flash - 为子系统的 EEE 备份存储器分配的 FlexNVM 量
- SPLIT - 子系统的分割系数, 决定子系统中 EEE 的字节数
- EEESIZE - EEE 总大小, 两个子系统中使用的 FlexRAM 字节数
- Record Efficiency - 对于 32 位和 16 位写入, 该值为 0.5; 对于 8 位写入, 该值为 0.25
- Baseline Endurance - 闪存预期的正常耐受能力(10K 个周期)

此公式用于计算每个 EEE 数据位置中允许的, 且保证能够执行的最小写入次数, 其假设条件为使用整个 EEE 大小, 均衡写入各数据位置, 并且不写入混合的数据大小。如果写入某些 EEE 位置的频率较低, 最小写入次数的计算公式允许对其他位置进行更多的写入。

可以使用该公式来大致了解系统可能需要的 E-Flash 量。但是, 这只是一个简化的公式, 旨在用作一般指导。飞思卡尔还提供了 EEE 耐受能力计算器应用程序。该计算器应用程序考虑了 EEE 数据更新方式的细节, 而不是假设使用相同的存取大小均衡更新所有数据值。该计算器还使用更复杂、更精确的工具来为耐受能力建模。对于提出了非常具体的 EEE 耐受能力要求的应用, 我们强烈建议使用该计算器工具。可以从[飞思卡尔网站](#)下载该计算器, 在关键字搜索框中键入“C90TFS\_FLEXMEM\_CALC”即可。

### 2.2.2.1.3 EEE 存储器配置示例 1 - 最大耐受能力

本示例介绍可为 EEE 数据尽可能地提供最大耐受能力的 EEE 配置。要尽可能地获得最大耐受能力, 您需要配置尽可能多的 E-Flash, 用于备份尽可能少的 EEE 数据。

要最大程度地利用 256 KB Kinetis 设备的 E-Flash 量, 请将整个 256 KB 的 FlexNVM 用作 E-Flash 存储器。一半的 E-Flash (前 128 KB)用于存储 EEE 子系统 A 的备份数据记录, 另一半 E-Flash (后 128 KB)用于存储 EEE 子系统 B 的备份数据记录。

要最小程度地利用 EEE 数据量, 请将 EEE 大小配置为最小可能值, 即 32 字节。然后, 可以将 EEE 分割选项设置为 1/8。这意味着 EEE 子系统 A 具有 4 个数据字节(单个 EEE 子系统的的核心可能大小)。子系统 B 为 28 字节。

可以使用[计算 EEE 耐受能力](#)部分中所述的耐受能力计算公式 2-1 来计算两个 EEE 子系统的预期耐受能力。

对于子系统 A, 假设进行 16 位和 32 位存取, 则预期的耐受能力为:

$$Endurance_{SubsystemA} = \frac{(128 \times 1024) - (2 \times 1/8 \times 3)}{1/8 \times 32} \times 0.5 \times 10000 = 163830000 \approx 164Mcycles \quad Eqn. 2-2$$

对于子系统 B，假设进行 16 位和 32 位存取，则预期的耐受能力为：

$$Endurance_{SubsystemB} = \frac{(128 \times 1024) - (2 \times 7/8 \times 3)}{7/8 \times 32} \times 0.5 \times 10000 = 23395714 \approx 23Mcycles \quad Eqn. 2-3$$

图 4 显示了如何为此配置使用 FlexNVM 和 FlexRAM 块。

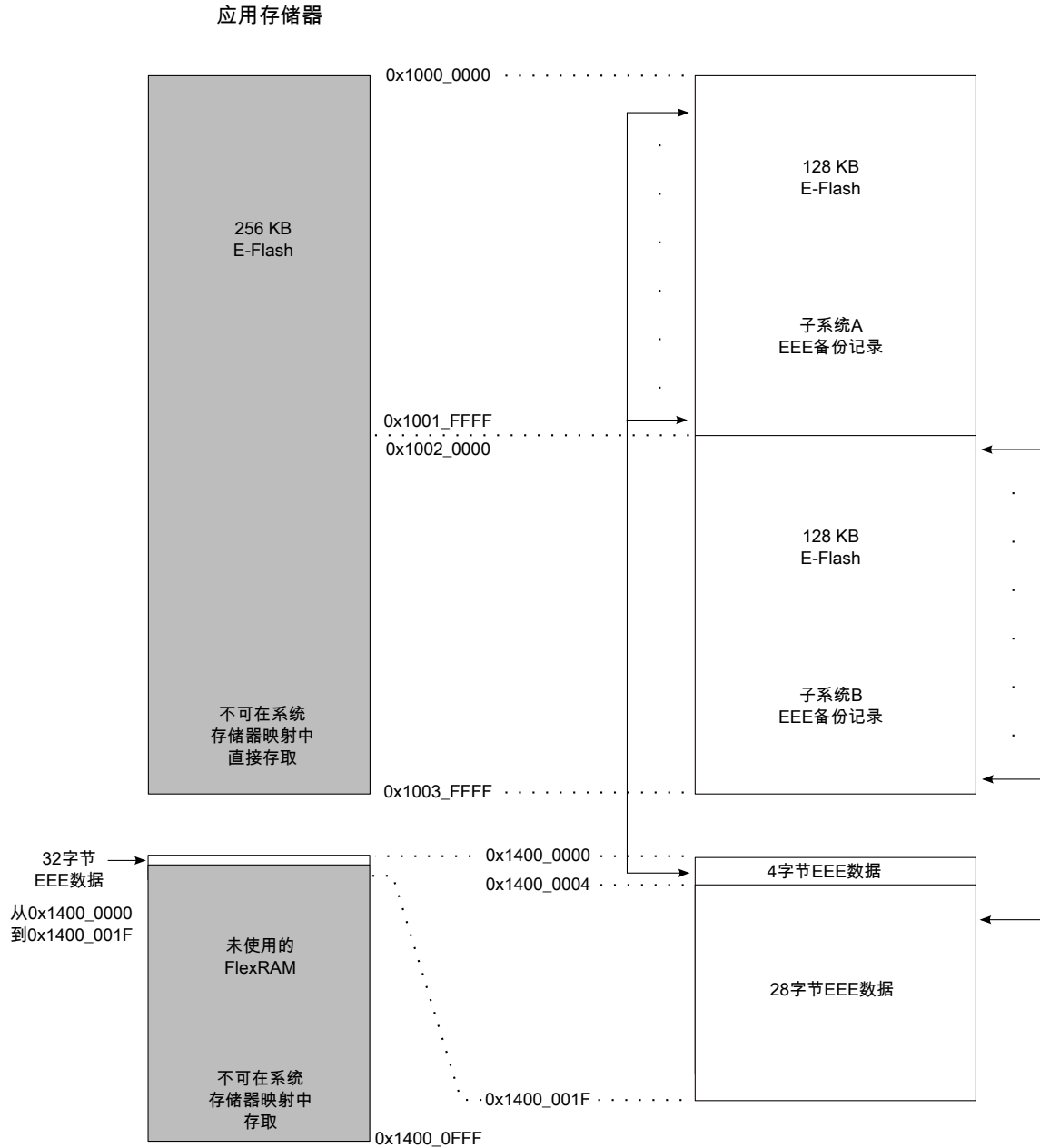


图 4. E-Flash = 256 KB、子系统 A = 4 字节、子系统 B = 28 字节的 EEE 配置

### 2.2.2.1.4 EEE 存储器配置示例 2 - 最大 EEE 数据空间

本示例介绍可以产生最大 EEE 数据空间量，并针对该数据量尽可能地提供最佳耐受能力的 EEE 配置。最大可能的 EEE 数据大小为 4 KB。如果将整个 FlexNVM 配置为 E-Flash (256 KB)，则您可以在整个 4 KB EEE 数据集方面尽可能地获得最佳的耐受能力。在本示例中，EEE 分割配置为 1/2，因此，整个 4 KB 的 EEE 具有相同的耐受能力。

可以使用 [计算 EEE 耐受能力](#) 部分中所述的耐受能力计算公式 2-1 来计算预期耐受能力。由于子系统 A 和子系统 B 的大小相同，因此，两个子系统的耐受能力相同。

对于子系统 A 或 B，假设进行 16 位和 32 位存取，则预期的耐受能力为：

$$Endurance_{SubsystemA/B} = \frac{(128 \times 1024) - (2 \times 1/2 \times 4 \times 102)}{1/2 \times 4 \times 1024} \times 0.5 \times 10000 = 310000 = 310kcycles \quad Eqn. 2-4$$

图 5 显示了如何为此配置使用 FlexNVM 和 FlexRAM 块。

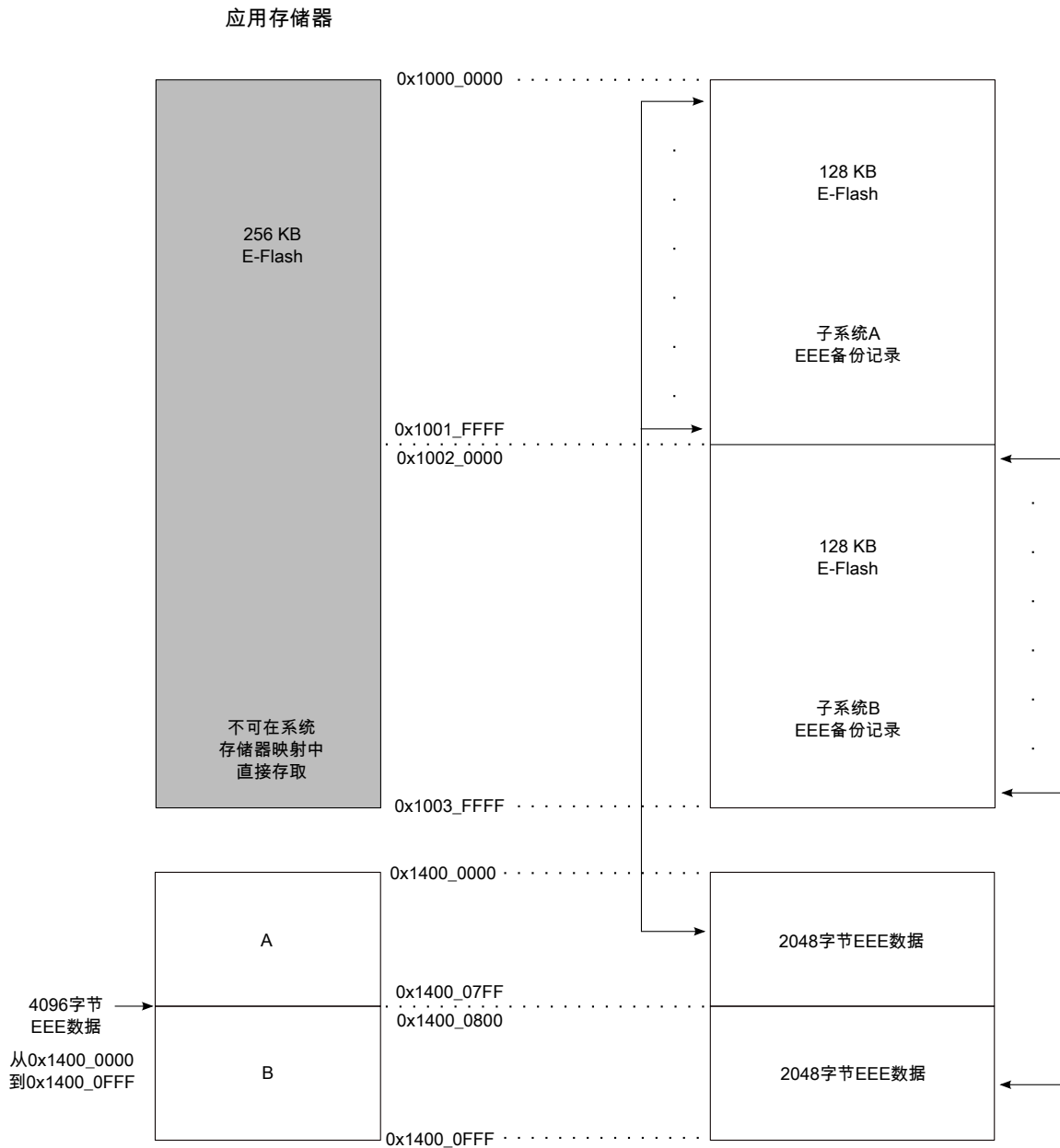


图 5. E-Flash = 256 KB、子系统 A = 2 KB、子系统 B = 2 KB 的 EEE 配置

### 2.2.2.1.5 EEE 存储器配置示例 3 - D-Flash 和 E-Flash 的混合形式

本示例将配置要用作 D-Flash 和 E-Flash 混合形式的 FlexNVM。在本示例中，FlexNVM 已对半分割，从而提供 128 KB 的 D-Flash 存储器和 128 KB 的 E-Flash。EEE 数据大小配置为 2 KB，并同样在子系统 A 和子系统 B 之间对半分割。

如果将本示例与示例 2 相比较，您会发现，E-Flash 和 EEE 大小都已减半。由于 E-Flash 与 EEE 大小的比相同，因此，预期的耐受能力也相同。可以使用 [计算 EEE 耐受能力](#) 部分中所述的耐受能力计算公式 2-1 来计算两个 EEE 子系统的预期耐受能力。可以发现，计算结果与 [EEE 存储器配置示例 2 - 最大 EEE 数据空间](#) 部分中的结果相同。

对于子系统 A，或者在假设进行 16 位和 32 位存取的情况下，预期的耐受能力为：



$$Endurance_{Subsystem A/B} = \frac{(64 \times 1024) - (2 \times 1/2 \times 2 \times 102)}{1/2 \times 2 \times 1024} \times 0.5 \times 10000 = 310000 = 310kcycles \quad Eqn. 2-5$$

图 6 显示了如何为此配置使用 FlexNVM 和 FlexRAM 块。

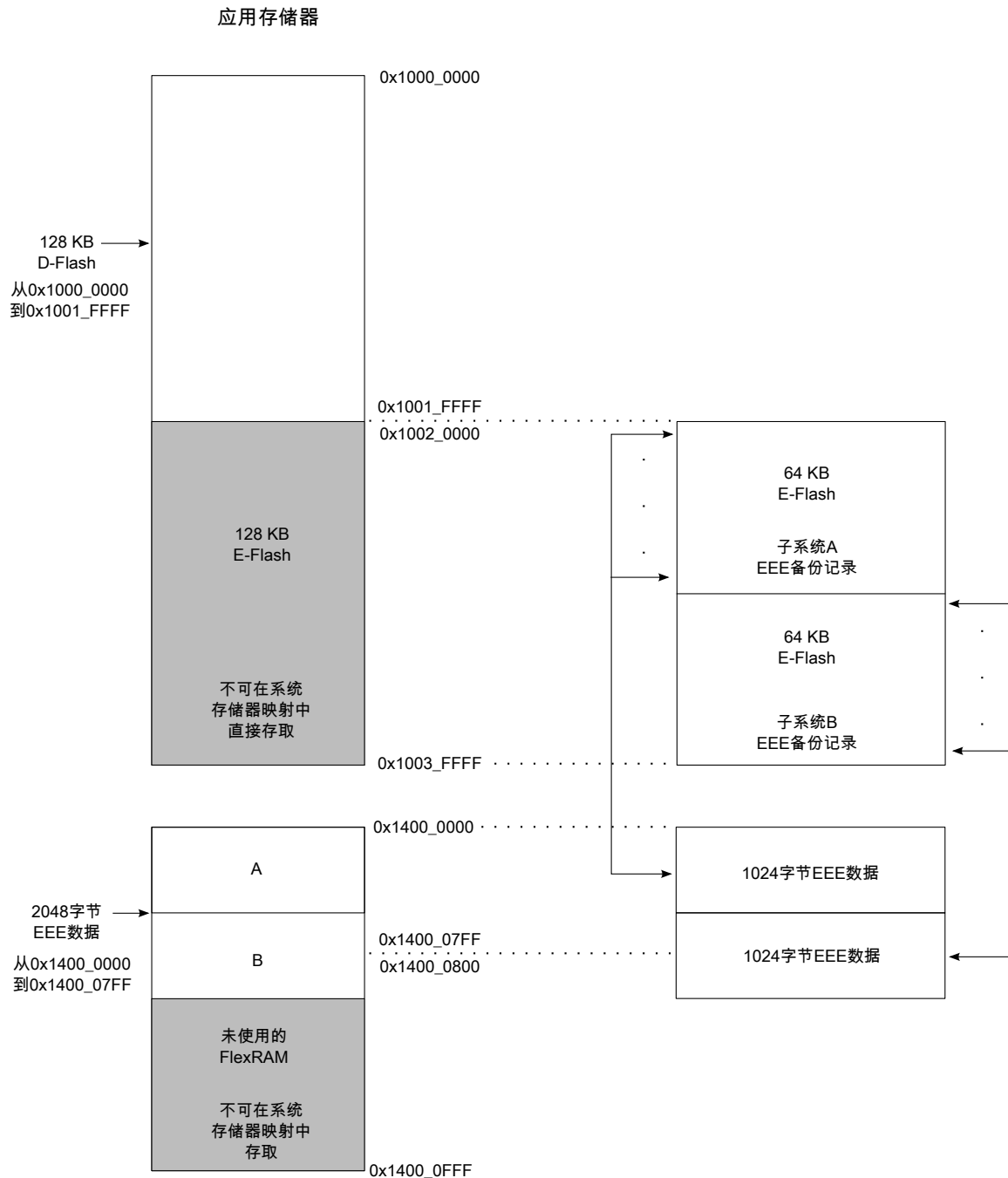


图 6. E-Flash = 128 KB、子系统 A = 1 KB、子系统 B = 1 KB 的 EEE 配置

### 2.2.2.1.6 EEE 存储器配置示例 4 - 1/4 EEE 分割选项

本示例将配置要用作 D-Flash 和 E-Flash 混合形式的 FlexNVM。在本例中，192 KB 的 FlexNVM 用作 D-Flash，64 KB 用作 E-Flash。EEE 数据大小配置为 2 KB，但这次的 EEE 分割选项配置为 1/4。因此，子系统 A 为 512 字节，子系统 B 为 1536 字节。

可以使用 [计算 EEE 耐受能力](#) 部分中所述的耐受能力计算公式 2-1 来计算两个 EEE 子系统的预期耐受能力。

对于子系统 A，假设进行 16 位和 32 位存取，则预期的耐受能力为：

$$Endurance_{SubsystemA} = \frac{(32 \times 1024) - (2 \times 1/4 \times 2 \times 102)}{1/4 \times 2 \times 1024} \times 0.5 \times 10000 = 310000 = 310kcycles \quad Eqn. 2-6$$

对于子系统 B，假设进行 16 位和 32 位存取，则预期的耐受能力为：

$$Endurance_{SubsystemB} = \frac{(32 \times 1024) - (2 \times 3/4 \times 2 \times 102)}{3/4 \times 2 \times 1024} \times 0.5 \times 10000 = 96666 \approx 97kcycles \quad Eqn. 2-7$$

[图 7](#) 显示了如何为此配置使用 FlexNVM 和 FlexRAM 块。

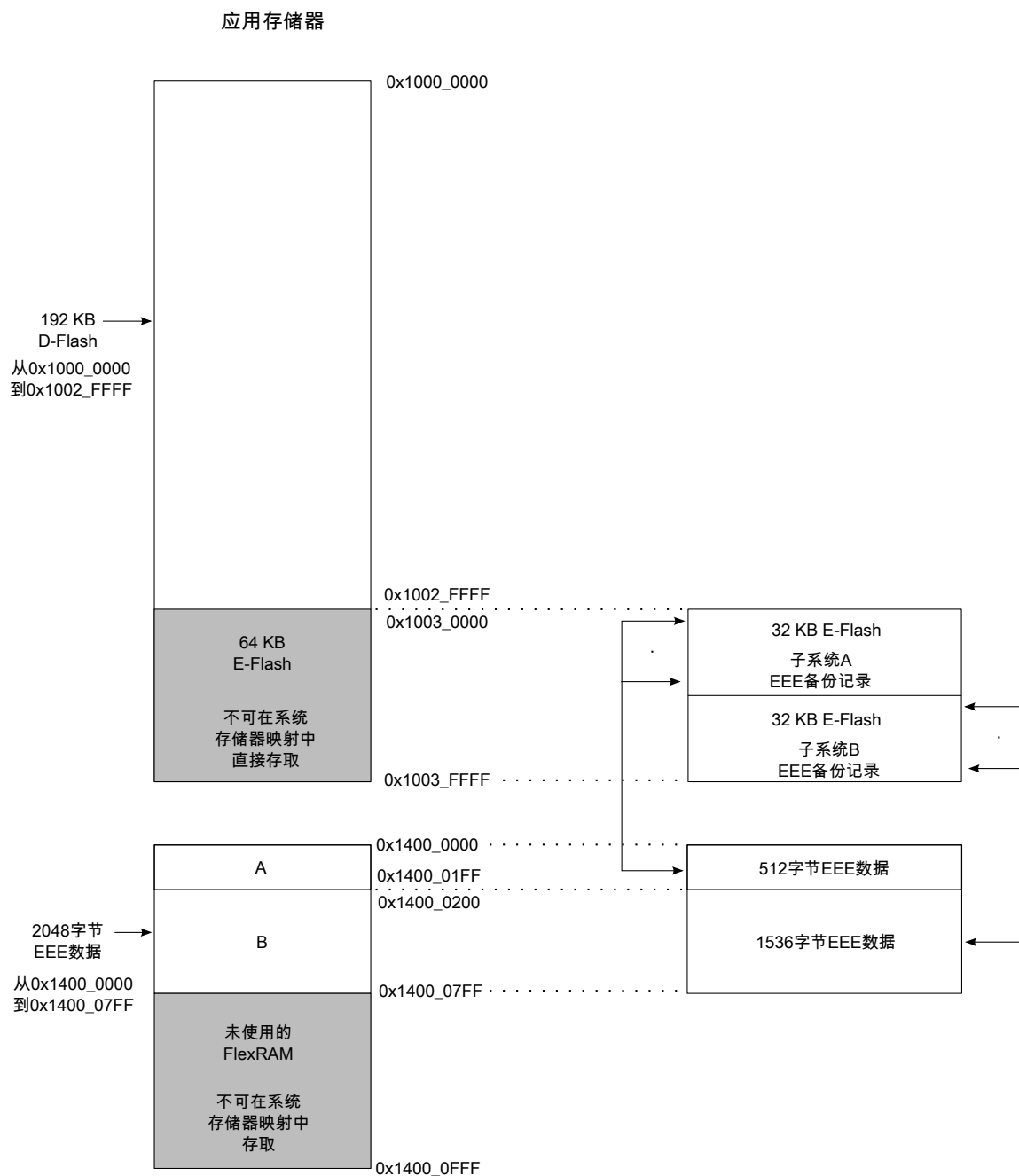


图 7. E-Flash = 64 KB、子系统 A = 512 字节、子系统 B = 1536 字节的 EEE 配置

### 3 使用 EEE

本部分介绍 FlexMemory 分区、EEE 启动以及如何读取和写入 EEE。

## 3.1 FlexMemory 分区

要使用 EEE 功能, 必须将存储器分区。分区过程将会告诉状态机要使用多少 EEE 存储量, 以及要使用多少 FlexNVM 闪存量来备份 EEE。

闪存具有一个用于配置 EEE 的特殊编程分区命令。使用该编程分区命令可以对 [EEE 配置参数](#) 部分中所述的 3 个 EEE 配置参数(EEE 大小、EEE 分割和 FlexNVM 分区)进行编程。这 3 个参数将编程到闪存块自身的特殊位置。由于这是一个非易失性存储位置, 因此, 在设备的整个生存期内, 只需执行分区一次。在启动编程分区命令之前, FlexNVM 和 D-Flash IFR 必须处于已擦除状态。在做出厂编程时, 建议首先将新设备分区。

### 注

- 分区只能执行一次。如果针对不同的配置反复为闪存分区, 则记录的数据将会丢失, 并且不保证您能获得预期的耐受能力。
- 如果整体擦除设备, 分区信息、EEE 数据和 EEE 位置信息将会丢失。在使用了 EEE 的设备上启用安全性时, 强烈建议使用后门密钥。通过该后门密钥, 您可以临时禁用安全性, 且无需执行整体擦除。如果使用整体擦除, 则不再能保证 EEE 的耐受能力。

## 3.2 EEE 启动

复位后, 将自动加载在分区过程中写入的 EEE 配置选项。如果已启用 EEE, 则状态机将在系统引导期间从 E-Flash 中加载包含 EEE 数据的 FlexRAM。将数据从 E-Flash 复制到 FlexRAM 中所需的时间根据 EEE 的配置大小以及需要分析的备份 E-Flash 量的不同而异。在完成 EEE 数据加载之前, 将清除 FTFL\_FCNFG[EEERDY]标志, 因此, 在尝试存取 FlexRAM 中的 EEE 数据之前, 软件必须等待设置 EEERDY 标志。如果需要中断驱动的选项而不是软件轮询, 则可以使用 CCIF 中断, 而不是轮询 EEERDY。

### 注

有关 EEE 启动及初始化代码的更多参考信息, 请参阅 [www.freescale.com](http://www.freescale.com) 上的 *Kinetis* 快速参考用户指南中的第 8 章“使用 FlexMemory”。

## 3.3 读取和写入 EEE

可通过存取 FlexRAM 地址空间来读取和写入 EEE 数据。EEE 空间的分配从 FlexRAM 的起始位置开始。可寻址空间为 FlexRAM 基地址(0x1400\_0000), 最大为编程的 EEE 大小。启用 EEE 功能后, 不能存取 FlexRAM 中未用作 EEE 的任何空间。例如, 如果 EEE 大小配置为总计 32 字节, 则允许对介于 0x1400\_0000 和 0x1400\_001F 之间的任何地址进行读取或写入存取, 但是, 对介于 0x1400\_0020 和 0x17FF\_FFFF 之间的地址进行这种存取会生成总线错误。

由于 EEE 数据是通过 RAM 存取的, 因此, 可以读取和写入任意字节、字或长字大小的数据。尽管支持任意存取大小, 但用于备份 EEE 数据的记录使用大小以字为单位的数据字段。也就是说, 允许进行字节写入, 但是, 这种操作会降低 E-Flash 的使用效率。正因如此, [计算 EEE 耐受能力](#) 部分中的耐受能力计算公式 2-1 针对 8 位存取使用了不同于 16 位或 32 位存取的效率因子。

### 3.3.1 EEE 写入

写入 EEE 空间会启动一项 EEE 操作, 以便在 E-Flash 存储器中存储数据。由于这是一项闪存编程操作, 因此, 在写入 EEE 空间之前, 软件必须测试 CCIF 位, 以确定是否正在进行任何其他闪存操作。由于不允许在同一个闪存块中执行多个并发的写入操作和写入时读取操作, 因此, 在完成 EEE 写入之前, 不允许存取 EEE 或 D-Flash 空间。

**注**

写入 EEE 空间会启动一项 EEE 操作，以便在 E-Flash 存储器中存储数据。由于这是一项闪存编程操作，因此，在写入 EEE 空间之前，软件必须测试 CCIF 位，以确定是否正在进行任何其他闪存操作。由于不允许在同一个闪存块中执行多个并发的写入操作和写入时读取操作，因此，在完成 EEE 写入之前，不允许存取 EEE 或 D-Flash 空间。

### 3.3.2 EEE 读取

读取 EEE 时，数据将由 FlexRAM 提供，因此，不会触发任何闪存操作。但是，当 EEE 写入正在进行时，不允许进行 EEE 读取。软件必须在执行读取操作之前测试 EEERDY 位，或者在执行写入存取之后等待设置 EEERDY，然后软件才能继续运行。在许多情况下，最有效的做法是让软件在写入之前和之后测试 EEERDY (或 CCIF)，并在写入之后、设置 EEERDY 之前阻止其他 EEE 操作。这样，便需要使用一个特殊的函数来进行 EEE 写入，但是，EEE 读取不需要任何特殊软件。采用这种方法的另一个优势在于，如果您要进行多次 EEE 读取存取，并且每两次存取之间没有任何 EEE 写入周期，则不需要提供额外的延迟或标志检查。

对于 EEE 读取，必须考虑到复位后首次存取 EEE 这种特殊情况。复位后首次读取 EEE 时，可能需要测试 EEERDY 位，以确保状态机完成了将数据从 E-Flash 载入 FlexRAM 的初始操作。如果系统启动时间较长，则可以确保首次读取 EEE 之前初始数据加载有时间得以完成，此时，可能不需要在首次读取之前测试 EEERDY 标志。但是，更安全的做法是在对 EEE 进行首次读取存取之前显式测试 EEERDY 位。

## 4 EEE 性能

FlexMemory EEE 器件除了具有灵活性和较高的耐受能力以外，还比典型的 EEPROM 速度更快。

通常，传统的外部 EEPROM 需要花费大约 5 ms 来配合最大编程速度。相比之下，在表现最差的情况下，EEE 在 1.5 ms 内就能完成擦写。

还可以通过将 0xFF 写入 EEE 数据位置来预先擦除 EEE。预先擦除存储位置有助于缩短编程时间，因为这可以确保无需执行一个擦除周期。对预先擦除的数据位置进行编程的典型时间为大约 100  $\mu$ s。在时间关键的应用场合中，此功能可让用户快速进行数据记录。

典型用例之一就是构建一个系统，以便在检测到即将断电时，将故障数据或操作信息存储在其中。需要保存的数据量乘以最大写入时间得到的结果，决定了要在该系统中提供多大的去耦能力，以保持足够大的最低运算能力使数据得到存储。预先擦除数据位置后 EEE 编程时间的大幅缩短意味着在此情况下所需的去耦能力更低，并且在断电前存储的数据更多。

## 5 掉电检测

EEE 状态机包含的逻辑可以检测是否有任何 EEE 数据未完全编程。

在文档中，此功能称为掉电检测，但是，该名称并不完全准确。在任何情况下，只要检测到 EEE 数据由于在写入过程中发生掉电或任何形式的复位而未完全编程，这种情况就会得到相同的处理。如果 EEE 写入正在进行时发生复位，则可能会损坏数据。EEE 状态机将会测试容易受损的 EEE 数据记录，以检测出可能未完全编程的值。如果检测到不完整的记录，状态机会将该数据标记为受损，并在下一次执行 EEE 写入期间，将该记录替换为 EEE 地址关联的上一条有效数据记录。这可以确保当 EEE 写入出于任何原因中断时，您仍可以获取已正确写入到 EEE 中的最后一个值。根据发生复位时写入操作的执行程度，此值可能是上一个值或者新值，但无论如何，您不会收到一个损坏的值。

**How to Reach Us:**

**Home Page:**  
[freescale.com](http://freescale.com)

**Web Support:**  
[freescale.com/support](http://freescale.com/support)

本文档中的信息仅供系统和软件实施方使用 Freescale 产品。本文并未明示或者暗示授予利用本文档信息进行设计或者加工集成电路的版权许可。Freescale 保留对此处任何产品进行更改的权利，恕不另行通知。

Freescale 对其产品在任何特定用途方面的适用性不做任何担保、表示或保证，也不承担因为应用程序或者使用产品或电路所产生的任何责任，明确拒绝承担包括但不限于后果性的或附带性的损害在内的所有责任。

Freescale 的数据表和/或规格中所提供的“典型”参数在不同应用中可能并且确实不同，实际性能会随时间而有所变化。所有运行参数，包括“经典值”在内，必须经由客户的技术专家对每个客户的应用程序进行验证。

Freescale 未转让与其专利权及其他权利相关的许可。Freescale 销售产品时遵循以下网址中包含的标准销售条款和条件：[freescale.com/SalesTermsandConditions](http://freescale.com/SalesTermsandConditions)。

Freescale, the Freescale logo, and Kinetis, are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners.

© 2011 Freescale Semiconductor, Inc.

© 2011 飞思卡尔半导体有限公司

Document Number AN4282  
Revision 0, 03/2011

