

## 使用赛普拉斯的 65 nm 异步 SRAM 降低单比特错误

作者： Nilesh Badodekar

相关项目： 无

相关器件系列： **CY621xxG30/CY7C10xxG30/  
CY621xxGE30/CY7C10xxGE30**

相关应用笔记： 无

本应用笔记介绍了赛普拉斯 65 nm 16 Mbit 异步 SRAM 纠错码 (ECC) 的特性。它说明了造成系统中单比特错误的主要原因，以及通常如何降低这些错误。本应用笔记提供了赛普拉斯 16 Mb 器件中所实现的 ECC 架构的概述，同时解释了这些器件中新信号的使用模型，用以在赛普拉斯 SRAM 中检测单比特错误及其纠正内容。

### 目录

1	简介 .....	1	4.3	受保护 ECC 存储器中的软错误累积效应.....	6
2	单粒子错误.....	2	4.4	32 位存储器架构的优势.....	7
2.1	SEU 的原因.....	2	5	通过 ASIC/FPGA/控制器连接至赛普拉斯 65 nm SRAM.....	8
3	减少 SEU .....	3	5.1	连接至 ERR 引脚 .....	8
3.1	改进加工技术和单元布局 .....	3	6	单比特错误纠正的操作示例 .....	10
3.2	改进芯片设计和架构 .....	3	7	总结 .....	12
3.3	系统级的降低 .....	4	文档修订记录.....	13	
4	赛普拉斯 65 nm 异步 SRAM 中软错误的降低情况 .....	5	全球销售和设计支持.....	14	
4.1	Hamming 代码 .....	5			
4.2	在赛普拉斯 SRAM 中纠正错误 .....	5			

## 1 简介

可靠性和数据的完整性是系统设计人员在设计存储设备时需要关注的最重要因素中的两个。现代系统要求不能因环境条件（如辐射）而破坏存储器中的数据。为提高可靠性，系统设计人员需要使用片外错误校验性能或冗余技术。使用这些技术会增大开销，如 PCB 面积或额外的处理时间。赛普拉斯的最新 SRAM 提供了一个带有片上纠错码 (ECC) 的单芯片解决方案，它能够缩小电路板的面积、降低成本以及设计的复杂性。与不带内置 ECC 的 SRAM (FIT 率超过 150 FIT/Mb) 相比，这些 SRAM 能够避免软失效<sup>1</sup> (FIT 率小于 0.1 FIT/Mb)。本应用笔记介绍了赛普拉斯 SRAM 的一个新特性，能帮助系统设计人员在 SRAM 中监控纠错码。

<sup>1</sup> 软错误仅在积累效应出现时才发生。欲了解更多信息，请点击[此处](#)。

## 2 单粒子错误

单粒子效应是指器件中逻辑状态或瞬态发生变化。这些变化由器件工作环境中的高能辐射粒子造成。高能粒子和半导体器件间相互作用可能会引起某一位翻转，即“软错误”，或造成永久性损坏，即“硬错误”。软错误是非破坏性的，因为通过复位或重写器件，可以恢复它的正常操作。这些错误与组件或制造故障无关。而硬错误则会破坏器件。常见的硬错误例子有：单粒子门锁以及由高电流和热失控引起的后续损坏。

由单辐射粒子引起的软错误被称为“单粒子错误 (SEU)”。SEU 可以是单比特错误 (SBU)，也可以是多单元错误 (MCU)。SBU 是指由单个高能粒子轰击而发生某个比特翻转的现象。MCU 是指由一个或多个辐射粒子移动而使存储器阵列中几个比特翻转的现象。

一个 MCU 事件是否会导致输出数据中的多比特错误 (MBU)，取决于器件架构。如果在一个输出数据字中发生了多比特翻转现象，那么将出现 MBU 事件。通常，需要降低 MBU 并避免发生 MBU，因为 MBU 会影响 Hamming 码错误的通用校正方案，仅在 SBU 中能够成功实现该校正方案。

由于采用了紧缩的加工技术 (250 nm 到 65 nm，甚至更小)，因此存储器单元的尺寸越来越小，内核电压也随之变小 (3.3 V 至 1.2 V 或更低)。这样会降低存储器单元的电容。因此，引起错误所需要的关键电荷 (单元保持数据所需的最小电荷) 会降低，使这些单元很容易发生 SEU。即使是一个  $\alpha$  粒子或由宇宙射线产生的中子，也能干扰存储单元。

SEU 是随机发生的，并且由于它们没有破坏器件，所以这类错误是非破坏性的。许多商业系统接受一定程度的软错误。但在重要的应用中，软错误不但可以破坏数据，而且还可以使整个系统发生故障，从而造成非常严重的后果。

### 2.1 SEU 的原因

引起 SEU 的主要原因包括：

- $\alpha$  粒子
- 高能中子
- 热中子

#### 2.1.1 $\alpha$ 粒子

通过模制化合物中石英填充物的微量铀-238 和钍-232 放射性衰变，生成  $\alpha$  粒子。这些杂质会释放  $\alpha$  粒子。因为  $\alpha$  粒子在物质中不能大距离移动，因此它只对附近的发射点 (大约 50  $\mu\text{m}$ ) 产生影响。因此，外部  $\alpha$  粒子源通常不会影响封装中的半导体器件。制造商通过使用超低能量的  $\alpha$  粒子发射复合转型来降低  $\alpha$  粒子引起的软错误。

#### 2.1.2 高能中子

高能宇宙粒子 (大部分为质子) 与地球高层大气发生反应。它们的碰撞 (由地球磁场引起) 产生高能量反冲粒子，如介子、 $\pi$  介子和中子。只有不带电荷的高能中子才能到达地球表面，与半导体器件相互作用。当这些中子到达地球表面时，其剩余的能量大概只有 1-500 MeV。

如果高能中子满足下面条件，它会引起软错误：高能中子必须通过碰撞硅核并与其发生电离才能产生电离粒子。这种碰撞可以生成  $\alpha$  粒子以及其他更重的离子，从而形成电子空穴对，其能量比模制化合物中典型  $\alpha$  粒子的能量更高。

中子会引起大麻烦，因为它们可以穿透大多数的人造建筑。例如，一个中子可以射穿 5 英尺厚的混凝土。通量率取决于地理位置，并且在位置增高的同时，它会随着大气屏蔽的降低而增大。常见的中子减速剂是水和土壤。因此，在隧道、山洞和潜艇中可以存在低能中子环境。

#### 2.1.3 热中子

热中子是位于宇宙簇射终端的低能中子。它们本来是高能中子，但它们在向地球表面移动过程中，由于多次碰撞而降低了本身能量。每次碰撞都会消耗一些能量。这样，在到达地球表面时，热中子的能量会非常低 (大概为 25 meV)。

一般情况下，热中子本身不会引起软错误。但是，10B (即在半导体的硼磷硅玻璃 (BPSG) 氧化膜中存在的一种硼的同位素) 将与热中子相互作用，并衰变为锂、氦核和一个  $\gamma$  射线，从而引起软错误。为了防止热中子产生的影响，半导体工业使用磷硅玻璃 (PSG) 氧化膜来替换 BPSG。赛普拉斯 65 nm 异步 SRAM 不包含 BPSG，因此它不受热中子的影响。

### 3 减少 SEU

在异步 SRAM 中减少软错误的主要方法包括：

- 改进加工技术和单元布局，以降低发生软错误的可能性
- 改进芯片设计和架构
- 系统级降低软错误

#### 3.1 改进加工技术和单元布局

当高能粒子与半导体衬底相互作用所产生的电荷超过 SRAM 单元中存储的关键电荷（QCRIT）时，将发生软错误。根据 Roche 以及其他人提出的模型（公式 1），QCRIT 是节点电容（CN）、工作电压（VDD）、最大漏极传导电流（IDP）以及单元翻转时间（TF）的函数。通过增加存储器单元的电容，可以增强 SRAM 器件的可靠性。

公式 1. 关键电荷（QCRIT）

$$Q_{CRIT} = C_N * V_{DD} + I_{DP} * T_F$$

然而，通过加工技术和单元布局等技术的更改来降低错误会引起负面影响。使用一个额外的电容或更大尺寸的单元来延长访问时间并提高待机电流（SRAM 被禁用时所消耗的电流），会对器件性能产生不利影响。

#### 3.2 改进芯片设计和架构

通过改进架构（如嵌入式 ECC 和比特交错）可以限制软错误对半导体器件的影响。赛普拉斯的 65 nm 异步 SRAM 实现了本节中所述的两种技术，用以降低软错误。

##### 3.2.1 嵌入式 ECC

在软件中实现的错误纠正和编码结构被固定编码到芯片架构内，以提示系统设计人员无需关注该纠正操作。

##### 3.2.2 比特交错

高能粒子与半导体原子碰撞时会对多个单元产生影响。在单个高能粒子影响同一个字中两个或多个比特时，将发生 MBU。采用比特交错技术可避免 SRAM 中发生 MBU 事件（由单粒子碰撞导致）。

比特交错技术通过安排多个 SRAM 单元使物理相邻位映射到不同的字寄存器。比特交错距离将映射到同一个字寄存器的两个连续位相分离。如果比特交错距离超过了一个多单元错误（MCU）大小，那么所产生的是多字节 SBU 而不是单字节 MBU。在比特交错存储器中，可以使用单比特错误纠正算法来检测和纠正所有错误。

图 1 和图 2 说明了发生 MCU 时比特交错技术带来的好处。如果存储器不具有比特交错特性（图 1），当发生 MCU 时，同一个字中的多个单元可能受到影响。当系统读取该字时，由于单比特 ECC 不能纠正该字，所以系统将读取并处理不正确的数据。

然而，如果存储器具有比特交错特性，则当发生 MCU 时，多个单元仍受影响，但它们被分离到多个字内。因此，每个字只有一个 SBU。当系统读取这些字时，单比特 ECC 可以分别纠正它们，因此系统能够处理所有字中的正确数据。

图 1. 不带比特交错特性的存储器

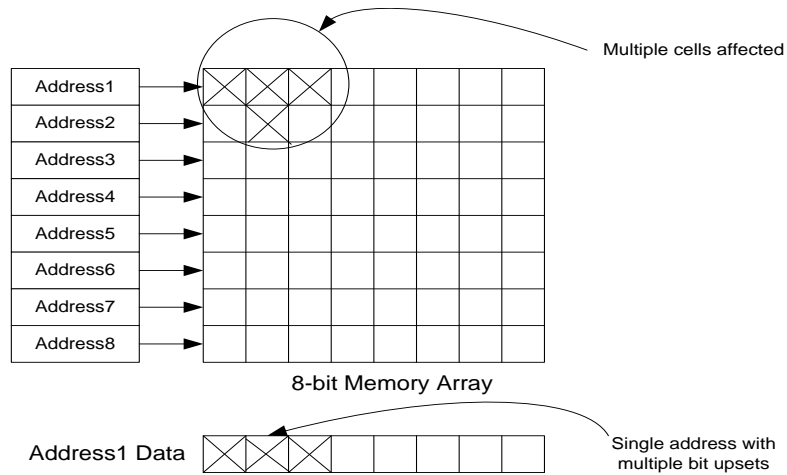
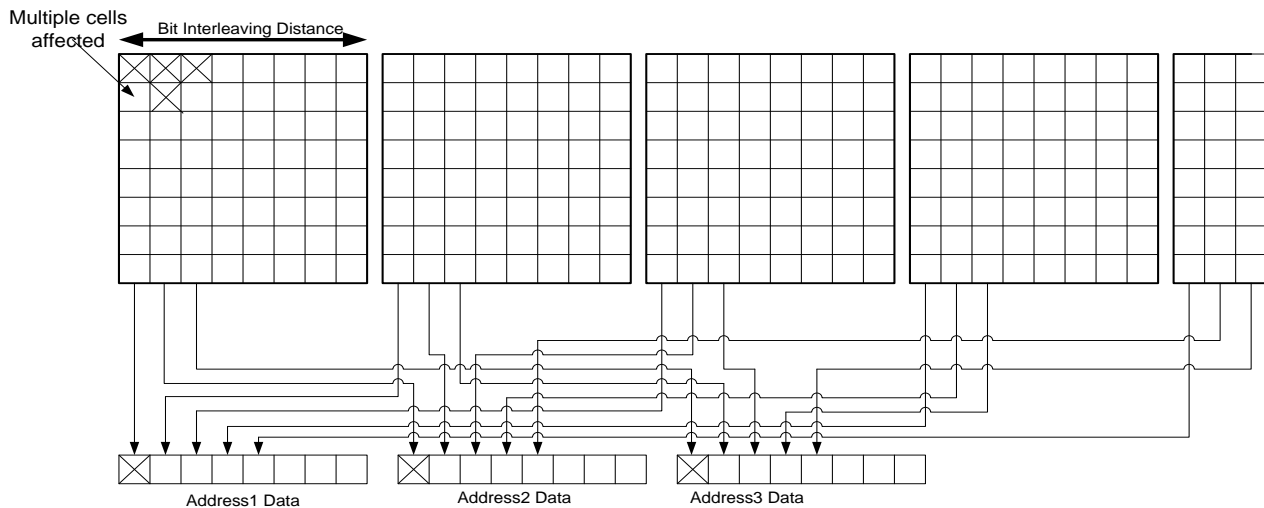


图 2. 带比特交错特性的存储器



### 3.3 系统级的降低

在系统级上，可通过在软件中实现冗余技术或 ECC 来降低软错误。

在三种模块冗余结构中，可同时将数据存入三个 SRAM 中。因此，每次进行读操作时，会同时读取三个 SRAM，并对大多数输出进行表决。如果其中一个 SRAM 受到了 SEU 的影响，则大多数表决结构可确保最终输出的数据是正确的，这是因为余下的两个 SRAM 不受 SEU 的影响。

如果在软件中，控制器或 ASIC 为所有的数据位生成奇偶校验位，并将其存储到独立的 SRAM 中，则系统设计人员可以在该软件中实现 ECC 结构。这样，每当执行读周期时，控制器都能访问数据位和校验奇偶位，从而在发生错误时纠正数据。

使用冗余 SRAM 或在软件中实现 ECC 结构会增加开销，如增大电路板面积和延长运行时间。此外，还会使设计变得更加复杂，因为要额外对电路板和软件进行设计。

## 4 赛普拉斯 65 nm 异步 SRAM 中软错误的降低情况

赛普拉斯 65 nm SRAM 使用了 Hamming 代码来降低单比特软错误。Hamming 代码是最简单、最可靠的知名 ECC 结构。本节介绍的是赛普拉斯 SRAM 中所使用的 Hamming 代码。

### 4.1 Hamming 代码

Hamming 码是一系列的线性 ECC。公式 2 总结了任意长度  $n$  的 Hamming 代码的基本公式。

公式 2. Hamming 代码公式

$$2^m \geq m + k + 1$$

其中：

$k$  = 数据位数量

$m$  = 奇偶校验位的数量

$n$  = 代码长度， $n = m + k$

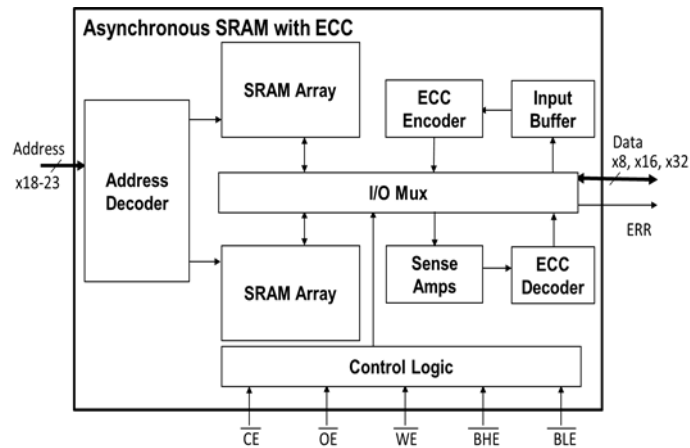
赛普拉斯 SRAM 的存储器阵列的组织方式是 32 位字。对于一个 32 位的数据字， $m$  的最小值为 6。这样，代码长度 ( $n$ ) 将为 38。因此，赛普拉斯使用 (32、38) Hamming 代码来检测并纠正单比特错误。

### 4.2 在赛普拉斯 SRAM 中纠正错误

赛普拉斯 65 nm 异步 SRAM 有一个硬件 ECC 模块，用于内联执行所有与 ECC 相关的功能。该编码过程无需用户干预，也不会影响器件访问时间性能。

16 比特交错方案提供了单比特错误的检测和纠正功能，以防止发生多比特错误。图 3 显示了赛普拉斯异步 65 nm SRAM 的框图。

图 3. 65 nm 异步 SRAM 的内部组织



每次进行写操作时，都会一同生成并保存 32 位字数据和 6 个奇偶校验位。通过 (32, 38) Hamming 架构，每次进行读操作时，赛普拉斯 65 nm SRAM 都能检测并纠正各单比特位。

订购代码带有 ‘E’ 字母（例如 CY7C1061GE30）的赛普拉斯 65 nm 异步 SRAM 封装中有一个额外的输出引脚，即 “ERR”。每个读周期中，系统设计人员会通过该 ERR 引脚监控 SRAM 中 ECC 逻辑的输出。在读周期中，如果没有发生单比特错误，则 ERR 引脚将被解除。但是，如果从 SRAM 读取的 32 位字中发生了单比特错误，则在当前的读周期内，ERR 引脚将被置于高电平状态。

即使存储器中存储的损坏数据的原始副本被破坏，ECC 逻辑仍可以纠正所读取的数据。进行读周期时，如果不执行任何单比特错误检测和纠正操作，则 ERR 引脚将被解除。经过一段时间后，如果后续 SEU 没有被应用程序覆盖掉，那么它们可以堆积在存储器阵列中。这样会使越来越多的软错误被“存入到”存储器阵列内。如果被存入的软错误数量达到关键阈值  $N_{TH}$ ，那么下一个被存入的软错误可能会引起在同一个数据字中发生多个 MBU，因为存储器阵列中的数据字已经受到一个 SBU 的影响。该效应称为“累积效应”，并在下一节中进行详细介绍。

赛普拉斯 ECC 架构不支持对 MBU 进行的检测和纠正。带有 ERR 引脚的器件会标记 MBU；但在这种情况下不能纠正数据，并且该数据可能包含多个 MBU。但达到累积软错误的关键阈值  $N_{TH}$  的大小，取决于软错误的生成率和应用模式。完全消除这种潜在问题的一个方法为：无论何时 ERR 引脚检测出并标记一个错误，都要将正确的数据写入到受影响的存储器单元内。这样可避免多个 SEU 累积并变成 MBU。如果不能马上重新写入到存储器内，那么下一节将向您介绍特殊应用指南。

### 4.3 受保护 ECC 存储器中的软错误累积效应

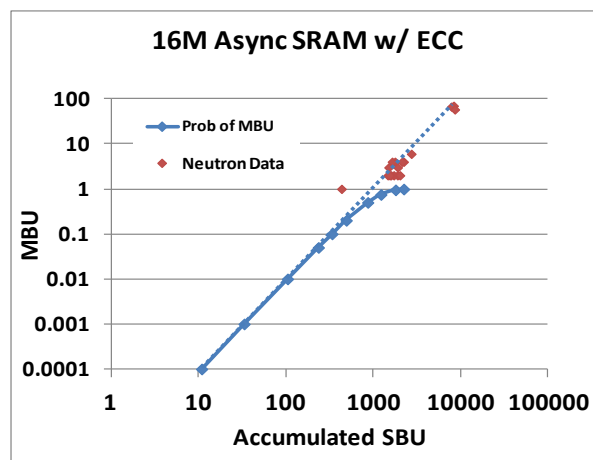
累积效应适用于各种受保护 ECC 存储器，如 DRAM、SRAM 和非易失性存储器。累积效应是否引起存储器数据级的 MBU 事件取决于嵌入式/系统 ECC 算法的执行情况。高位计数 ECC 结构（如 Reed-Solomon 和双位检错/纠错 Hamming 码方案）可以成功纠正由累积效应引起的潜在的双位。单位检错/纠错的经济方案需要深入分析，从而确定达到累积软错误的关键阈值  $N_{TH}$  的时间。

已修改的生日统计<sup>2</sup>用于计算在存储器阵列中累积的给定单比特错误发生无法纠正的双位错误的概率。图 4 显示的是发生 MBU 事件的概率，包括了使用赛普拉斯 16 Mb 异步 SRAM 器件的  $\alpha$  粒子和重离子的测量结果（表 1）对概率斜线进行的实验验证。

根据图 4 和表 1 中显示的实验数据，累积得到的软错误关键阈值确定为  $N_{TH} = 1,000$  个错误。如果累积在存储器中的软错误超过了  $N_{TH}$ ，那么将发生 MBU 事件。在最坏情况下，用于积累软错误的应用模式是数据保持模式和仅读访问模式。在这两种模式下，不能对存储器进行写操作，在存储器进行操作过程中，软错误将被积累起来。在最坏情况下，积累速率为 838 FIT/Mb（数据保留模式下不带 ECC 保护故障率）。但如果在 NYC 位置，存储器阵列中积累 1,000 个软错误将需要长达 3,417 年。

表 2 对已给 SBU 错误数量提供了 MBU 的发生概率。通过该数据，可以预测在预期发生某些 SBU 的系统中发生 MBU 的可能性。例如，如果系统中发生 864 个单比特错误，将有 50% 的可能性会发生一个 ECC 无法纠正的双比特错误事件（一个 MBU）。

图 4. 无法纠正 MBU 与被积累的 SBU 事件



<sup>2</sup> Tausch, H.J., “简单生日统计与 Hamming EDAC”, *IEEE Transactions on Nuclear Science*, 第 56(2)卷, 2009 年, 474 - 478 页。



表 1. 积累效应<sup>3</sup>

	Unit	16M Async SRAM w/ECC Accumulation Effect
Neutron SER Rate @ NYC w/o ECC	FIT/Mb	838
Alpha SER Rate for ULA w/o ECC	FIT/Mb	1250
Density	Mb	16
Failures in 10 <sup>9</sup> device hrs	#err	33408
Time to reach 1000 errors	yrs	3417

表 2. 积累 SBU 的 MBU 发生概率

	MBU 发生概率									
	0.0001	0.001	0.01	0.05	0.1	0.2	0.5	0.75	0.95	0.99
积累的单比特错误	10.88975332	33.32841314	104.5366	235.5294	337.3452	490.7113	864.48	1222.3522	1796.649	2227.467

因此，可以得出结论，对于商业/工业应用，即使存储器在其整个产品使用寿命周期中没有进行更新，积累效应也不会有问题。

#### 4.4 32 位存储器架构的优势

系统设计人员可以根据接口要求选择一个带有 16 位或 8 位数据总线的 SRAM 器件。但(32, 38) Hamming 架构始终使用内部尺寸为 32 位的字进行操作。这对系统设计人员非常有用，因为如果某个字（宽度为 32 位）存在 SBU，那么当读取四个 8 位数据中的某一个，或读取两个 16 位数据中的某一个时，将检测并纠正该错误。图 5 显示的是大小为 1 Mb 的 16 位 SRAM（16 Mb SRAM）的情况。

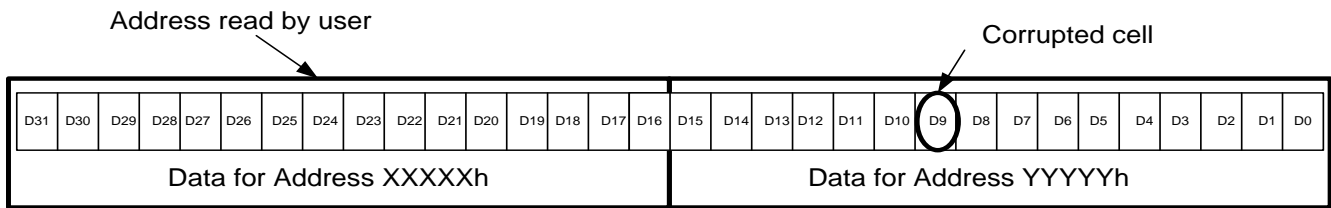
在宽度为 16 位的 SRAM 中，内部 32 位数据字由两个 16 位数据字组成。为了简单，假设这两个地址位置（XXXXXh 和 YYYYYh）组成一个 32 位的内部字。

如果地址 YYYYYh 中发生单比特错误，并且正在访问的是地址 XXXXXh 中的数据，那么，ECC 架构将内部读取并纠正该 32 位数据，但输出的只有与地址 XXXXXh 相应的 16 位数据。如果地址 XXXXXh 中的数据通过一个外部控制器进行更新，那么在该写入周期内，内部 ECC 逻辑将读取现存的数据、重建 32 位正确数据并将其存入到中间缓冲区内。在这个中间缓冲区中，地址 XXXXXh 中的新数据将被覆盖，此外，ECC 逻辑会生成新的奇偶校验位。

得益于这种内部清理，与地址 YYYYYh 相应的数据将被自动纠正。因此，对于 16 位 SRAM，用户能在所有写入周期内在多个 16 位位置上纠正数据；对于 8 位 SRAM，在所有写入周期内，只有 3 个 16 位位置得到更新。对于带 ERR 引脚的器件，由于 SRAM 检测并纠正了 32 位字中存在的错误，所以即使正在访问地址 XXXXXh，仍会激活 ERR 引脚。系统设计人员需要识别 ERR 激活状况，并回写先前从地址 XXXXXh 读取的同一数据。在回写过程中，内部 ECC 逻辑将生成新的奇偶校验位，并纠正整个 32 位数据字。地址 YYYYYh 也会随之得到纠正，从而保证数据的完整性。

<sup>3</sup> ULA — 超低阿尔法级测试

图 5. 从 32 位内部字读取 16 位数据字



## 5 通过 ASIC/FPGA/控制器连接至赛普拉斯 65 nm SRAM

系统设计人员可以使用该 ERR 引脚来监控 SRAM 中数据的完整性。由于仅在 SRAM 检测到并纠正单比特错误时才会激活 ERR 信号，因此建议弱下拉该信号，从而避免在进行写入操作期间或 SRAM 被禁用时发生中间电压电平。在写周期或芯片被禁用的周期内，ERR 引脚处于高阻态。

### 5.1 连接至 ERR 引脚

在系统中，对不带 ERR 引脚的 SRAM 的连接方式与对传统 SRAM 的连接方式相同。对于带 ERR 的 SRAM，系统设计人员应该在系统中正确地连接 ERR 引脚。如果系统中没有使用 ERR 引脚，则可以将该引脚置于未连接状态。本节介绍的是在使用 SRAM 的三种基本情况下如何连接至 ERR 引脚。

- 带有单一 SRAM 的系统
- 宽度扩展
- 深度扩展

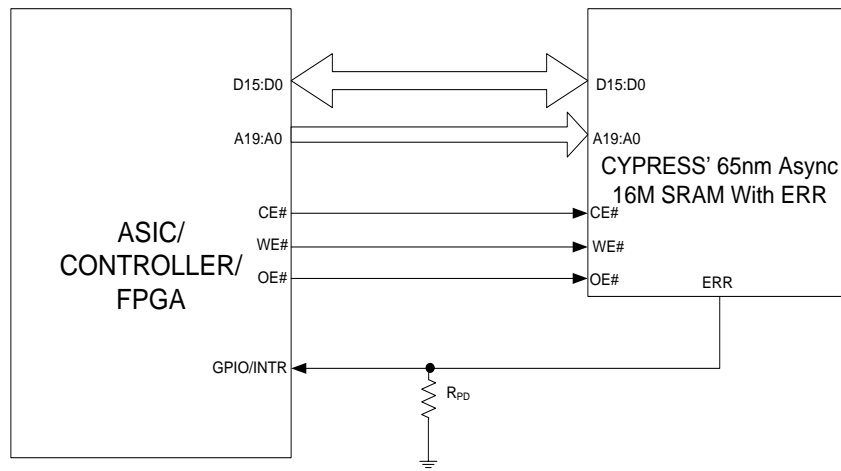
#### 5.1.1 带有单一 SRAM 的系统

在只有一个 SRAM 器件连接至板上控制器/ASIC/FPGA 的系统中<sup>4</sup>，ERR 引脚可以连接至 ASIC 中的某一个 GPIO/中断引脚。在每个读周期中，控制器都会监控 ERR 引脚的输出。当 ERR 引脚处于高电平状态时，ASIC 要启动一个软件子程序来重新写入正确的数据（从 SRAM 输出）。图 6 显示的是这种情况的架构。

<sup>4</sup> 可以互换使用“控制器”、“ASIC”和“FPGA”等术语。



图 6. 连接至赛普拉斯 SRAM

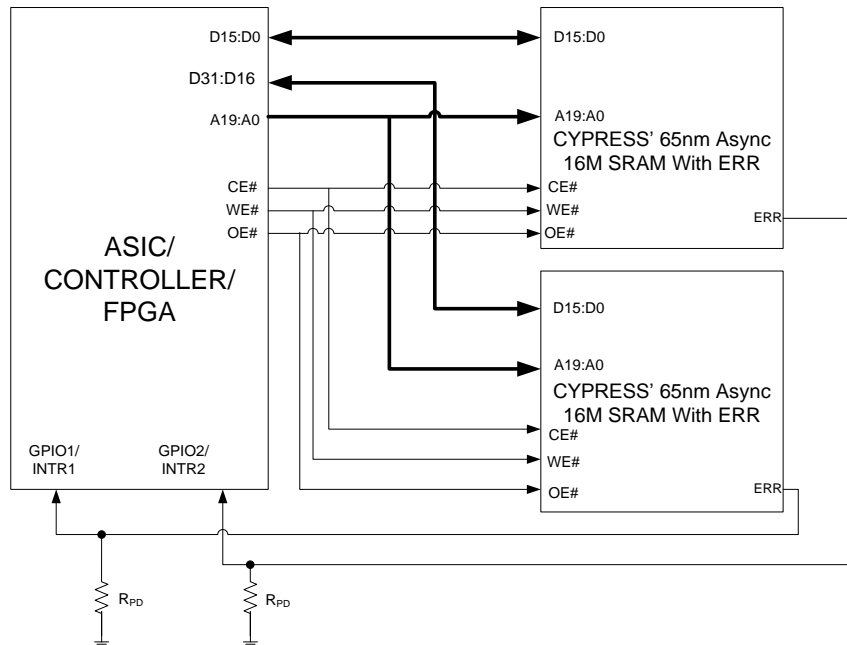


### 5.1.2 宽度扩展

新一代控制器具有一个 32 位的数据接口。在这种应用中，系统设计人员可以通过使用两个带有 ERR 引脚的赛普拉斯 65 nm SRAM 来扩展数据总线的宽度。系统软件需要分别监控两个 ERR 引脚。系统设计人员可以将两个 x16 赛普拉斯 SRAM 连接至其控制器的 32 位总线上。

在这种情况下，在读周期内，如果来自两个 SRAM 中的某一个的 ERR 信号被激活，那么，该 SRAM 器件中的数据将被刷新并被重新写入，以保持数据的完整性。图 7 显示的是在这种情况下连接方式。在宽度扩展模式下连接两个 1M x16 (16 Mb) SRAM，以创建大小为 1M x32 (32 Mb) 的存储器空间。

图 7. 宽度扩展和 ERR

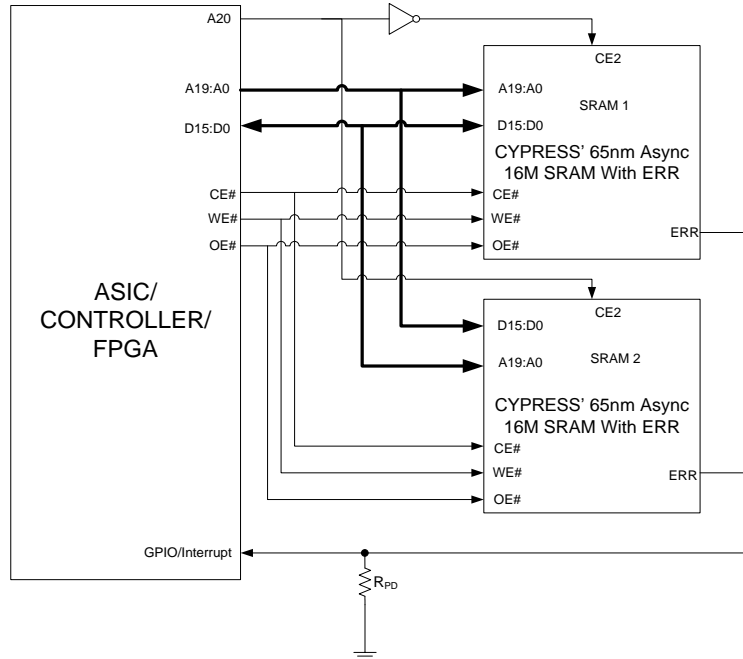


### 5.1.3 深度扩展

在某些内存密集型应用中，SRAM 中可用空间可能满足不了应用存储器的要求。这时，系统设计人员可以使用赛普拉斯 65 nm SRAM 实现深度扩展。

序列较高的地址信号（在该示例中为 A20）用于选择 SRAM，余下的地址信号、I/O 信号、控制信号以及来自两个 SRAM 的 ERR 信号按照图 8 显示的情况进行连接。在深度扩展模式下连接两个 1M x 16（16 Mb）的 SRAM，以创建 2M x 16（32 Mb）的存储器空间。根据 A20 信号，一个 SRAM 器件被使能，另一个 SRAM 器件被禁用。被禁用的 SRAM 器件将其 ERR 引脚置于高阻态，以便使被使能的 SRAM 正确驱动组合的 ERR 信号。

图 8. 深度扩展和 ERR



## 6 单比特错误纠正的操作示例

本节从系统设计人员角度介绍了 ECC 逻辑和 ERR 引脚的功能。本节假设系统设计人员已经使用上述方法中的一种，连接好了赛普拉斯 65 nm 16 Mb SRAM。表 3 中的示例显示了 SRAM 中四个存储器位置所存储的数据。

表 3. 存储器中的初始数据

地址位置	存储器中的数据
0000h	0000h
0001h	FFFFh
0002h	0002h
0003h	FFFFh

由于两个存储单元发生了 SBU，地址位置 0001h 和 0002h 中的数据被破坏。表 4 显示的是被更改的数据。

表 4. 在存储器中发生 SBU 的数据

地址位置	存储器中的数据
00000h	0000h
00001h	FFFEh
00002h	0003h
00003h	FFFFh

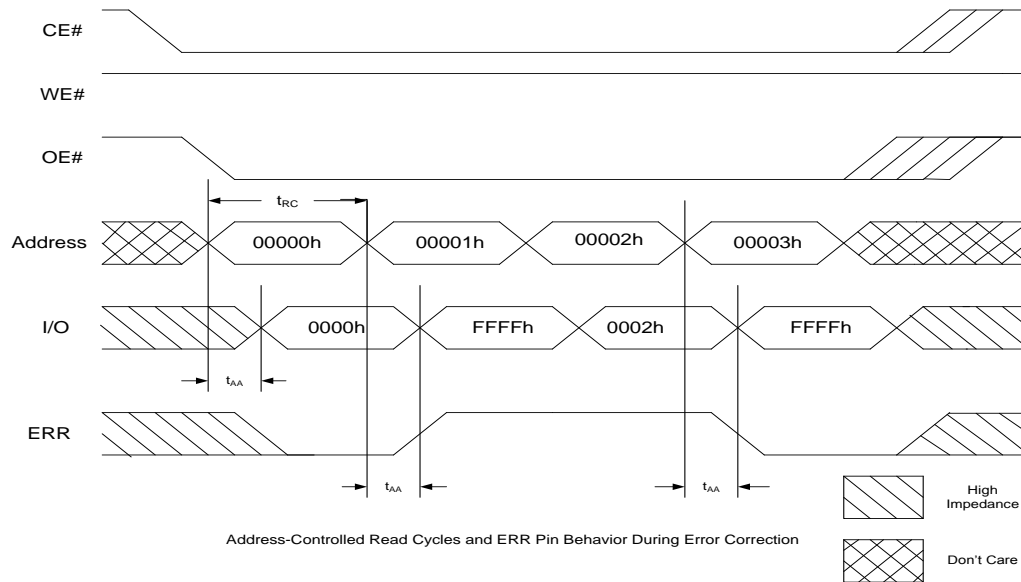
如果对这四个地址进行读访问，则 ECC 逻辑将纠正被损坏的数据字。这样，读取到的是正确的数据。对于带有 ERR 引脚的器件，在数据字被纠正的读周期内，该引脚将被激活。在不发生任何单比特错误的读周期内，SRAM 将取消激活 ERR 引脚。表 5 显示的是 ERR 引脚最终的输出及状态。

表 5. 读周期

地址位置	存储器中的数据	读周期期间 I/O 上的数据	ERR 引脚状态
00000h	0000h	0000h	0
00001h	FFFEh	FFFFh	1
00002h	0003h	0002h	1
00003h	FFFFh	FFFFh	0

所有的单比特错误都被检测和纠正，并且带 ERR 引脚选项的器件能标识所有的纠正操作。在应用软件中，可以通过 ERR 引脚的状态启动某个子程序，并将正确的数据回写到受影响的地址位置内。图 9 总结了整个读取操作，并高亮显示了关键的时序参数。有关 ERR 引脚的时序参数，请参考器件数据手册。

图 9. 错误纠正过程中由地址控制的读周期以及 ERR 引脚的状态



## 7 总结

在现代系统中，数据完整性是一个非常重要的特性。SEU 是对 SRAM 中存储的数据的威胁。在系统软件中或在系统级上实现的错误检测和纠正都会对设计（如 PCB 面积和时序性能）产生不利影响。通过赛普拉斯 65 nm 异步 SRAM，系统设计人员能够减轻 SRAM 对单比特错误纠正功能的负担。可选的 ERR 特性允许用户监控 SRAM 数据及其单比特错误，并通过刷新被破坏的数据来保持数据的完整性。

---

### 关于作者

姓名: Nilesh Badodekar.  
职务: 应用工程师  
背景: Nilesh Badodekar 从 I.I.T. Kharagpur, India 获取视觉信息加工和嵌入式系统的硕士学位。

## 文档修订记录

文档标题: AN88889 – 使用赛普拉斯的 65 nm 异步 SRAM 降低单比特错误

文档编号: 001-92385

版本	ECN	原始变更	提交日期	变更说明
**	4371858	NILE	05/06/2014	本文档版本号为 Rev**, 译自英文版 001-88889 Rev**。
*A	4471508	YLIU	08/12/2014	本文档版本号为 Rev*A, 译自英文版 001-88889 Rev*A。
*B	4769235	YLIU	06/03/2015	本文档版本号为 Rev*B, 译自英文版 001-88889 Rev*B。
*C	5138395	YLIU	02/16/2016	本文档版本号为 Rev*C, 译自英文版 001-88889 Rev*C。

## 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、工厂代表和经销商组成的全球性网络。要找到离您最近的办事处，请存取赛普拉斯所在地。

### 产品

汽车级	<a href="http://cypress.com/go/automotive">cypress.com/go/automotive</a>
时钟与缓冲器	<a href="http://cypress.com/go/clocks">cypress.com/go/clocks</a>
接口	<a href="http://cypress.com/go/interface">cypress.com/go/interface</a>
照明与电源控制	<a href="http://cypress.com/go/powerpsoc">cypress.com/go/powerpsoc</a>
存储器	<a href="http://cypress.com/go/memory">cypress.com/go/memory</a>
PSoC	<a href="http://cypress.com/go/psoc">cypress.com/go/psoc</a>
触摸感应	<a href="http://cypress.com/go/touch">cypress.com/go/touch</a>
USB 控制器	<a href="http://cypress.com/go/usb">cypress.com/go/usb</a>
无线/射频	<a href="http://cypress.com/go/wireless">cypress.com/go/wireless</a>

### PSoC®解决方案

[psoc.cypress.com/solutions](http://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](http://cypress.com/go/support)

PSoC 是赛普拉斯半导体公司的注册商标。此处引用的所有其他商标或注册商标归其各自所有者所有。



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

©赛普拉斯半导体公司，2014-2016。此处所包含的信息可随时更改，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出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。

产品使用可能受限于赛普拉斯软件许可协议。