AN13566

i.MX 8 系列上的 ECC

第 0 版 — 2022 年 4 月 1 日 应用笔记

目录

1 介绍

1.1 目的

纠错码（ECC）是若干 i.MX 8 应用处理器提供的一项功能，用于进一步增强存储在 DRAM 中的数据的完整性。本文档提供有关 ECC 功能的信息，以帮助读者开发启用现有 ECC 功能的应用程序。

ECC 功能仅在选定的设备上可用，例如：

- i.MX 8M Plus
- i.MX 8QuadXPlus / i.MX 8DualXPlus
- i.MX 8XLite（* 试验生产）

其他恩智浦处理器系列也支持 ECC，但本文档不做描述。大多数 DRAM 芯片包括“内部”片上纠错电路，本文档中也不做描述。

1.2 读者

本文档针对 i.MX 8 系列中的选定设备，面向了解以下内容的用户：

- SoC 上可用的 ECC 功能
- 启用 ECC 的具体要求和影响
- 如何开始开发一个利用 ECC 的应用程序

读者应该了解基本的内存架构概念和 DRAM 功能。

1.3 缩略语和释义

表 1. 缩略语和释义

缩略语	释义
DRAM	动态随机存取存储器
DDRC	DDR（DRAM）控制器
DFI	DDR PHY 接口
ECC	纠错码
CAM	内容可寻址存储器

表格在下一页继续...

表 1. 缩略语和释义（续）

缩略语	释义
EDAC	错误检测和纠正
FIT	一定时间故障次数
MTBF	平均故障间隔时间
RPA	寄存器编程辅助
RMW	读修改写
DM	数据掩码
HD	汉明间距
ISI	符号间干扰
SBR	ECC 洗涤器
SCU	系统控制器单元
SCFW	系统控制器单元固件
SER	软错误率
SECDED	单错校正双错检测
V2X	车联网

2 概述

2.1 数据损坏

在使用外部 DRAM 的现代 SoC 设计中，有几种机制会导致处理器接收到不正确的数据：

- 阿尔法/宇宙粒子/辐射
- 信号完整性/符号间干扰（ISI）/噪声
- 保留或耦合故障
- Row hammering

宇宙射线和其他外部事件可以通过改变电容器中的电荷水平而导致 DRAM 单元的损坏。为了解决这个问题，ECC 存储器在数据位旁边存储了额外的奇偶校验位以纠正这些错误。

DRAM 存储器可以通过依靠错误校正码提供更多的保护，防止软错误。这种纠错存储器（ECC）对于需要符合功能安全标准（ISO26262）的系统、高容错应用，以及辐射增加的航空/空间应用来说是理想的选择。

由于 DRAM 存储器采用了更小的技术节点，因此软错误率（SER）的概率会增加。SER 是指设备或系统遇到（或预测会遇到）软错误的比率。它通常表示为一定时间内的故障次数（FIT）或平均故障间隔时间（MTBF）。

需要在较高温度下运行的应用程序必须更频繁地刷新 DRAM，以确保 DRAM 单元中的电荷得以维持，从而防止数据损坏。

2.2 纠错

大多数 DRAM 芯片包括“内部”片上纠错电路，这使得具有非 ECC 存储器控制器的系统仍然可以获得 ECC 存储器的大部分好处。具有 ECC 功能的 DDR 内存控制器可以使用 CPU 和内存之间的“外部”电路检测和纠正错误。ECC 数据错误可以存储在附加存储器内或主 DRAM 存储器阵列内。

i.MX 8 DDR/DRAM 控制器（DDRC）支持错误检测和纠正（EDAC），配置为：单个“位”单错纠正/双错检测纠错码（SEC/DED ECC），用于 DRAM 数据宽度配置为 16 或 32 位的情况。

ECC 可以纠正“n”位错误（n≥1），并检测超过“n”位翻转的情况。为此，ECC 在每个数据字中加入冗余的 ECC 位，以“检查”其他位。数据位和 ECC 位的组合被称为一个码字。ECC 确保如果一个有效的码字中的任何位发生变化，它就不再是一个有效的码字。

- 任何码字之间的汉明间距（HD）至少为 3，这意味着它可以检测到最多 2 位的错误，并纠正一个可检测的错误。然而，区分一个字有 1 位的损坏和一个信息有 2 位的损坏是不可能的。为了解决这个问题，汉明码可以通过一个额外的奇偶校验位进行扩展。这样就有可能将汉明码的最小距离增加到 4，这使得解码器能够区分 1 位错误和 2 位错误。解码器可以检测并纠正单个错误，同时检测（但不纠正）双重错误。
- 单错校正/双错检测：最常见的错误纠正码（单错校正/双错检测汉明码）允许 1 位错误被纠正，（带有额外的奇偶校验位）2 位错误被检测。

3 纠错码

3.1 ECC 方案

有两种普遍使用的 ECC 方案（内联 ECC 和边带 ECC），它们在 i.MX 处理器上是相互排斥的（见图 1）。

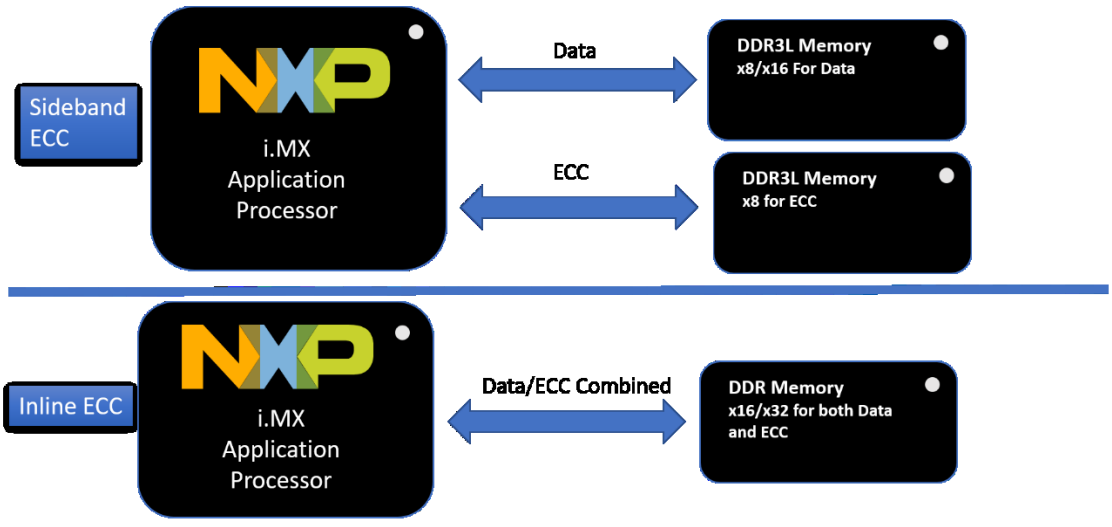

图 1. ECC 架构

内联 ECC 是边带 ECC 的一个替代方案，它在以下设备上得到支持：

- i.MX 8XLite – LPDDR4, DDR3L – 16-bit
- i.MX 8M Plus – LPDDR4, DDR4 – 32-bit

支持边带 ECC 的设备和内存配置如下：

- i.MX 8QuadXPlus and i.MX 8DualXPlus – DDR3L – 40-bit

注意

i.MX 8DualX 处理器不支持边带 ECC，因为它只支持 16 位 DDR 接口。由于 16 位 DDR 接口，17 x 17 mm 0.8 mm FCPBGA 封装选项不支持 ECC。

表 2. 内联 ECC 和边带 ECC 的比较

	边带 ECC	内联 ECC
被支持的设备	i.MX 8QuadXPlus DDR3L 40-bit (32 + 8 ECC) i.MX 8DualXPlus DDR3L 40-bit (32 + 8 ECC)	i.MX 8XLite (16-bit) LPDDR4, DDR3L i.MX 8M Plus (32-bit) LPDDR4,DDR4
ECC 数据	存储在单独的 DRAM 中	为单个 DRAM 上的 ECC 数据保留 1/8 的密度
需要额外的 DRAM	是	否
数据与 ECC 的比率	8/1	8/1
汉明码	64/8 单错校正双错检测	64/8 单错校正双错检测
性能影响	是 （比内联 ECC 的影响更小）	是
数据掩码	可选择的	必需的

在内联 ECC 校正情况下，额外的内存周期被用来将 ECC 数据存储到现有的内存设备中。对于边带 ECC（用于 i.MX 8QuadXPlus/i.MX 8DualXPlus DDR3L），数据使用单独的引脚存储到额外的 DRAM 设备。在这两种情况下，存储比率是每 64 位数据有 8 位 ECC，使用 SECDED（单纠错双检错）汉明码。

注意

边带 ECC 支持与内联 ECC 支持相互互斥。支持内联 ECC 的 i.MX 8 设备不支持边带 ECC，反之亦然。

1.1 内联 ECC

内联 ECC 不需要为 ECC 提供额外的数据总线，所以实际的 DRAM 数据宽度等于“DRAM_DATA_WIDTH”。ECC 奇偶校验与数据一起存储，不需要使用专门的边带存储器设备。

注意

“DRAM_DATA_WIDTH”一词将被用来指在 DRAM 存储器中存储实际数据（而不是 ECC）的总线宽度。

当启用内联 ECC 时，必须编程将最高的 3 个列位映射到可能的最高地址映射位置。该控制器灵活的地址映射方案是受限的，因此最高的系统地址空间被保留给 ECC 奇偶校验，并作为单一的区域进行浪费。对于其余所有区域的正常数据，系统地址是线性连续的。有效数据不是 DRAM 的全部大小。

内联 ECC 的特性如下：

- ECC 奇偶校验（代码）与数据一起存储，不使用专门的边带存储设备。
- 支持 LPDDR4、DDR4 和 DDR3L 协议。
  - 支持的内存数据宽度为 16 和 32。
- 使用 64/8 单错校正双错检测汉明码：
  - 数据与 ECC 的比率为 8/1。
- 它要求启用数据掩码（DM）。
- 当写访问不能填满一个汉明码（64 位）时，需要使用 RMW 命令。
- 由于设备特性和设备拓扑结构（边带不实用），它适合 LPDDR4。

3.2.1 启用内联 ECC

与整个 ECC 功能一样，该功能是可选的，必须由用户专门启用。如果用户想使用内联 ECC 功能，他们必须通过恩智浦社区网页上提供的相关 i.MX 8 寄存器编程辅助（RPA）工具的“寄存器配置”选项卡（图 2）启用该功能。

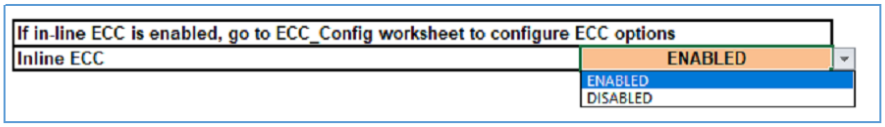

图 2. 启用内联 ECC

当 ECC 功能特性被启用后，用户不能动态地禁用它。如果必须修改任何 ECC 的配置设置，必须重新生成一个新的寄存器编程辅助脚本。

3.2.2 选择 ECC 区域

ECC 功能被启用后，进一步的 ECC 配置将在各自的 i.MX 8 寄存器编程辅助（RPA）工具的“ECC_Config”选项卡中进行。

内存的关键区域可以选择 ECC，其他区域可以取消选择 ECC。ECC 部分被映射到系统地址的前 1/8，剩余空间可以映射为 7 个可选区域，和 1 个“其他区域”，如果所选区域大小没有覆盖整个空间,。请参阅 ECC 应用考虑因素，以选择要保护的区域和其他特定应用程序的优化。

对于每个 bank，每行地址的下 7/8 是数据，上 1/8（按列地址区分）是 ECC 空间。数据存储在存储器的下 7/8 处，相关的 ECC（1/8 大小）存储在上 1/8 的相应部分。内存有一小块区域（1/8 的 1/8-1/64）被浪费了（内存的地址范围被分配到 ECC 空间）。

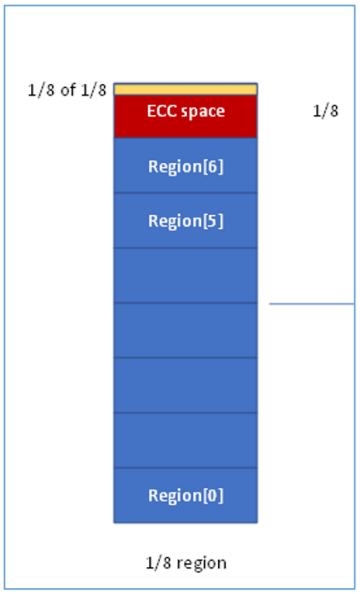

图 3. ECC 区域

每个区域可以是总内存映射的 1/8、1/16、1/32 或 1/64。没有被 7 个区域覆盖的剩余内存区域被称为“其他”区域，但 1/8 的粒度除外，因为这个粒度设置没有剩余的内存区域。用户可以选择在配置“其他”区域时是否有 ECC 保护。

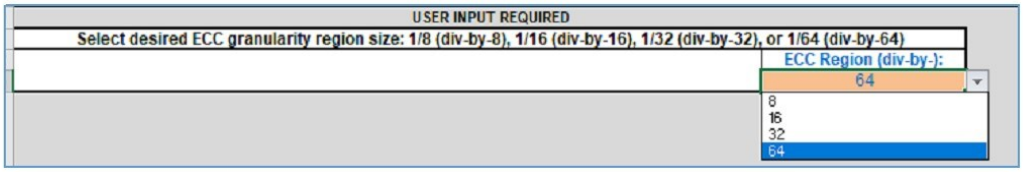

图 4. 选择 ECC 粒度

这 7 个区域（地址是基于从最低地址开始的区域数量）可以被配置为具有或不具有 ECC 保护。没有 ECC 保护，就不会遇到性能上的开销。没有内存空间被恢复，它只是没有被使用。在图 5 所示的例子中，区域 0 在 DRAM 地址范围的开始处，被配置为 ECC 保护。

Main Memory Region	Start Address of each Main Memory Region	Density of each Main Memory Region	User Input ECC Protection Configuration for each Main Memory Region
Other Region	0x087000000	784MB	UNPROTECTED
Region6	0x086000000	16MB	UNPROTECTED
Region5	0x085000000	16MB	UNPROTECTED
Region4	0x084000000	16MB	UNPROTECTED
Region3	0x083000000	16MB	UNPROTECTED
Region2	0x082000000	16MB	UNPROTECTED
Region1	0x081000000	16MB	UNPROTECTED
Region0	0x080000000	16MB	PROTECTED
Total DRAM density:	1024MB (8Gb)		

图 5. 选择 ECC 区域

本例内存（在 i.MX8 DualXLite EVK 上使用）对应的受保护 ECC 奇偶校验部分如下：

ECC Parity Region Section	Start Address of each ECC Parity Region Section	Density of each ECC Parity Region Section	ECC Parity Region Section memory attributes
ECC Parity Region 0 Section	0x0BFE00000	2MB	ACCESSIBLE
ECC Parity Region 1 Section	0x0BFC00000	2MB	ACCESSIBLE
ECC Parity Region 2 Section	0x0BFA00000	2MB	ACCESSIBLE
ECC Parity Region 3 Section	0x0BF800000	2MB	ACCESSIBLE
ECC Parity Region 4 Section	0x0BF600000	2MB	ACCESSIBLE
ECC Parity Region 5 Section	0x0BF400000	2MB	ACCESSIBLE
ECC Parity Region 6 Section	0x0BF200000	2MB	INACCESSIBLE
Other Region ECC Parity Region Section	0x0B9000000	98MB	ACCESSIBLE
Always user accessible	0x0B8000000	16MB	ACCESSIBLE

图 6. 相应的 ECC 奇偶校验部分

每个 ECC 奇偶校验区的内存都可以被用户访问，这取决于相应区域的 ECC 保护方案。如图 6 所示，不可访问的 ECC 奇偶校验部分 0 总是在 DRAM 内存图的最上面。这种配置的好处是只保留了最上面的内存，其他应用程序可以连续地访问其余的内存。

注意

用户软件必须确保没有应用程序访问被映射为不可访问的区域。试图访问不可访问的区域会导致数据中止。

3.2.3 区域锁

当启用 ECC 时，用户软件不能访问数据奇偶校验区（最多为 DRAM 总密度的 1/8）。通过编程“ECC_REGION_PARITY_LOCK”寄存器位，可以锁定 ECC 区域。当 ECC 被启用时，这个位默认为锁定位置。当设置时，它锁定了 ECC 区域（孔）的奇偶校验部分，这是内存的最高系统地址部分（ECC 奇偶性校验保护区域）。

废区也可以被锁定，以确保用户无法访问废区。“ECC_REGION_WASTE_LOCK”寄存器位锁定了没有被“ECC_REGION_PARITY_LOCK”锁定的 ECC 区域（孔）的剩余废区部分。

ECC 区域通常由控制器锁定和保护以避免访问。然而，如果有必要注入错误（测试机制），可以通过解锁然后，直接写入 ECC 区域来完成。

建议 LOCK 配置如下：

`ECC_REGION_PARITY_LOCK = 1` - 防止对 ECC 奇偶校验区的访问

`ECC_REGION_WASTE_LOCK = 0` - 允许访问 ECC 废区

注意

为了在启用 ECC 时，最大限度地提高内存使用率，用户可以通过解锁来访问废弃区域。然而，用户将负责他们的软件应用程序不访问保留（不可访问）的区域。默认情况下，RPA 启用了对废区的访问（这个废区默认是解锁的）。

3.2.4 非二进制密度

非二进制内存是一种密度为 6、12、24 或 48Gb 的内存。当使用这种密度时，必须在 DDRC 中编入一个特殊的地址映射寄存器。对于 ECC，当使用密度为 6、12、24 或 48Gb 的 DRAM 存储器时，需要特别考虑，用户必须选择单独的“ECC 配置”选项卡。

`ADDRMAP6.LPDDR4_6gb_12gb_24gb = 0` - 使用二进制对齐的工作表

`ADDRMAP6.LPDDR4_6gb_12gb_24gb = 1,2,3` - 使用非二进制对齐的工作表

注意

如果使用了不正确的工作表，RPA 会根据 `ADDRMAP6.LPDDR4_6gb_12gb_24gb` 设置发出警告。参见二进制/非二进制密度，以了解有关该主题的更多细节和建议。

3.2.5 操作流程

ECC 的生成和检查由控制器自动处理，生成单独的命令来存储和读取 ECC 值和数据。

对于“读”的操作，ECC 被读取并存储在内部。地址是由主读数据计算出来的。如果相关的 ECC 数据已经被加载，则不需要读取。读取数据被安排在读取 ECC 操作之后，以便在返回给请求者之前，使用已经存在的 ECC 数据对读取数据进行修正。

对于写操作，ECC 是在写到内存时计算的。写入完成后，写入 ECC 数据（在计算的地址）。对于需要对 ECC 进行不完全写入的情况，ECC 是通过屏蔽写入（必须启用），而不是对 ECC 数据进行读/修改/写入操作。如果需要对数据进行部分写入（如果只写入双字的一部分），则需要对数据进行读/修改/写入操作，以确保写入完整的双字，且计算出的 ECC 正确。

3.2.6 洗涤器

i.MX DDR 控制器（DDRC）提供了一个全面的解决方案，可以自动纠正 DRAM 中的 1 位错误。ECC 洗涤器（SBR）是一个模块，它初始化受保护的 DRAM 区域的 ECC 值，然后向 DDRC 发起周期性的后台读取命令。洗涤命令是对 1 个内存突发的读取（例如，BL8），它以最低的优先级周期性地发送给 DDRC。

注意

ECC 洗涤器（SBR）与 ECC 擦洗功能不同，后者只在边带 ECC 配置中被支持。

洗涤器的功能是通过纠正数据并写回内存来确保单个 ECC 错误不会累积。内联和边带 ECC 在机制上有所不同。

在内联 ECC 配置中，不支持控制器的 ECC 擦洗功能，它不能为每个可纠正的读错误进行擦洗。由于这个原因，读/修改/写（RMW）命令是由洗涤器本身对检测到的每一个 ECC 错误位发起的。

在内联 ECC 模式下，ECC 洗涤器（SBR）只在受保护的区域内生成地址。它自动跳过未受保护的区域和 ECC 区域。ECC 洗涤器（SBR）不向无效地址发送事务，而是在下一个周期跳过到下一个有效地址。在硬件控制的低功耗模式下，洗涤器继续自动运行，没有任何软件干预。当使用内联 ECC 模式时，RPA 确保洗涤器配置为覆盖所有受保护区域。

就我们的目的而言，当检测到 ECC 单个位错误时，所提供的数据被纠正并发送给请求者。然而，这个数据不会被写回内存。因为洗涤器不断地在整个保护空间内运行，读取数据和 ECC 并进行检查，它最终会遇到错误的数据字。当洗涤器检测到一个可纠正的错误时，将在没有有效数据的情况下安排一个 RMW 操作。它读取内存检查，纠正数据，并在数据纠正后执行写回内存的操作。这个操作周期性地运行，读取之间有一个可编程的时间，并且覆盖一个指定的地址范围。当“Scrub_Burst”被编程时，洗涤器会自动确保这些“背靠背”的事务，然后是一个很长的等待时间。它执行“n”个事务并等待“n”个间隔。这很有帮助，这样洗涤器就不会不断地中断系统流量。

SBR（洗涤器）编程

RPA 工具中预先配置了洗涤器范围、SBR（洗涤器）突发间隔和其他参数，以实现最佳操作。这些可编程寄存器的详细信息将包括在各自的 SoC 参考手册中。

SBRCTL - 洗涤器控制寄存器 - 用于编程洗涤间隔、洗涤突发计数，并在不同模式下启用洗涤功能。

SBRSTAT - 洗涤器状态寄存器 - 用于检查洗涤器命令的状态，报告繁忙和已完成启动的洗涤功能。

SBRWDATA0 - ECC 洗涤器为数据总线[31:0]的 ECC 初始化写数据模式。

SBRWDATA1 - ECC 洗涤器为数据总线[63:32]的 ECC 初始化写数据模式。

注意

额外的洗涤启动和范围功能仅用于调试目的，而正常操作时不需要。

洗涤机制在 DDR 初始化期间和定期进行，如下所述。

洗涤器在 DDR 初始化期间执行的洗涤

当 ECC 被启用时，洗涤器在 DDR 初始化期间执行洗涤，以便用初始化有效的数据和 ECC 保护区域。洗涤器只在 ECC 保护区域执行。这是基于特定 SoC 的 DDR RPA 工具的基础上自动配置的。

- 在 i.MX 8XLite 和 i.MX 8QuadXPlus/i.MX 8DualXPlus 的情况下，寄存器编程辅助（RPA）工具会根据哪些区域受到保护而自动填充洗涤器寄存器写入。
- 在 i.MX 8MPlus 的情况下，DDR 压力测试工具会生成一个“*.c”文件，该文件包含函数调用以洗涤受保护区域。

在这两种情况下，都不需要用户互动，因为这些机制是根据各自 RPA 工具中的 DRAM 参数和 ECC 保护区域配置而自动配置的。

洗涤器定期进行洗涤

当 ECC 被启用时，洗涤器会定期执行后台读取命令，但只在 ECC 保护的区域进行。当 ECC 选项被启用时，这个机制是默认启用的（在 RPA 工具中）。此外，其他字段，如洗涤间隔，也被预先配置为最佳操作。不需要进一步的用户配置。

3.2.7 报告 ECC 错误

DDR 控制器使用中断提供了 ECC 错误报告机制。有几个与内联 ECC 机制有关的错误被映射出来：

- `ECC_NCORRECT_INT`：检测到一个不可纠正的错误。
- `ECC_CORRECT_INT`：检测到一个可纠正的错误。
- `ECC_AP_ERR_INT`：检测到一个导致地址保护故障的不可纠正的错误。

这与之前的不可纠正的错误有区别，因为错误的数量更大（大于 *ECCCFG0.ecc_ap_err_threshold*），表明有数据不匹配。

ECC 中断被映射，可以通过 SoC 全局中断控制器（GIC）进行配置。

各个中断处理例程中的应用软件必须决定对所设置的 ECC 中断采取的具体行动。参见错误报告和行动。下面是 i.MX 8XLite 和 i.MX 8MPlus SoC 的中断映射示例。

Interrupt Name	Description	Interrupt Request (IRQ) Number
ECC_CORRECT_INT	A correctable ECC error has been detected	100
ECC_NCORRECT_INT	A non-correctable ECC error has been detected	101
ECC_AP_ERR_INT	A non-correctable error such that an address channel error is suspected	104

图 7. i.MX 8XLite ECC 中断映射

IRQ	Module	Logic	Interrupt Description
147	DDR	OR	DRAM Controller Error Interrupt for address protection fault.
147	DDR	OR	DRAM Controller Error Interrupt for correctable ECC error detected
147	DDR	OR	DRAM Controller Error Interrupt for uncorrectable ECC error detected

图 8. i.MX 8M Plus ECC 中断映射

注意

中断的分配因 SoC 而异，用户应参阅各自的参考手册以获得进一步信息。

错误映射：

64/8 SECDED（单错校正双错检测）汉明码可以检测/纠正 1 位错误，并在 64 位字（8 字节）内检测 2 位错误。以下情况下会产生内联 ECC 错误生成。

- 访问有 1 个错误位的 64 位的字 – 可纠正的错误
  - `ECCSTAT.ecc_corrected_err` 将被设置为 1
  - `ECCERRCNT.ecc_corr_err_cnt` 将显示检测到的、可纠正的 ECC 错误的数量
- 访问一个有 2 个错误位的 64 位字 – 不可纠正的错误和总线故障
  - `ECCSTAT.ecc_uncorrected_err` 将被设置为 1
  - `ECCERRCNT.ecc_uncorr_err_cnt` 将显示检测到的、不可纠正 ECC 错误的数量
- 访问一个有两个以上错误位的 64 位字。在这种情况下，错误的数量超过了 64/8 SECDED（单错校正双错检测）汉明码，其结果是不确定的。可能报告以下任何一种情况：可纠正的错误，不可纠正的错误，或没有错误。
- 访问一个有两个错误位的 64 位字，并且相应的突发有超过 `ECCCFG0.ecc_ap_err_threshold` 的不可纠正错误 – AP 错误、不可纠正的错误和总线故障。
  - `ECCAPSTAT.ecc_ap_err` = 1
  - `ECCSTAT.ecc_uncorrected_err` = 1

突发粒度：

ECC 引擎检查一个突发中的所有字。如果 DRC 客户端访问了一个正确的 64 位字，但在同一个突发中存在一个不正确的 64 位字，那么这个不正确的字将被报告错误，尽管这不是正在读取的字。

如果错误是不可纠正的，则不会报告总线故障，因为实际正在读取的数据没有被破坏。如果在一个突发中，发现超过 `ECCCFG0.ecc_ap_err_threshold` 的不可纠正的错误，但这些错误并不影响正在访问的 64 位字，将产生一个 AP 错误和一个不可纠正的错误，但不生成总线故障。

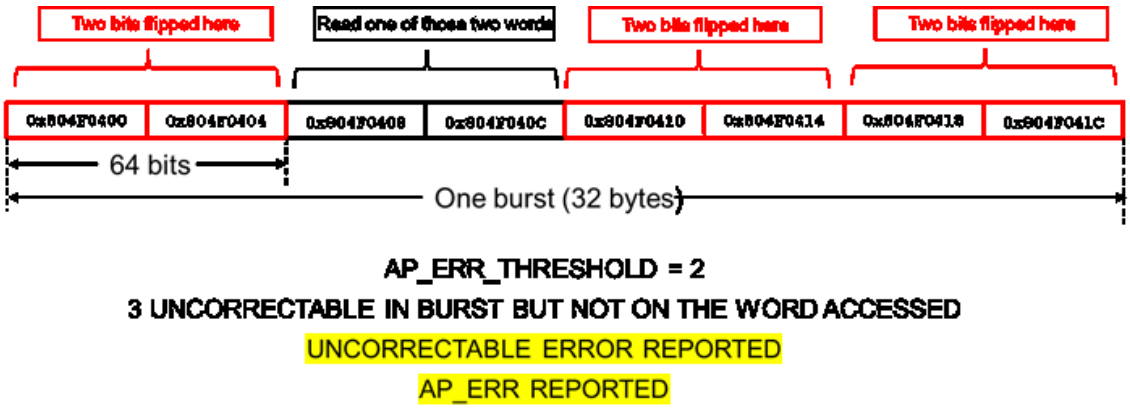

图 9. ECC AP 错误的映射示例

下面的章节提供了关于 ECC AP 错误的更多细节。

3.2.7.1 报告 ECC AP 错误

i.MX 8 DDR 控制器除了报告 ECC 可纠正和不可纠正的错误外，它还支持第三种错误报告机制（ECC AP 错误）。在这种情况下，如果在一个突发中，具有 1 位或 2 位错误的 64 位字的数量超过 `ECCCFG0.ecc_ap_err_threshold` 中编程的值，那么 `ECCAPSTAT.ecc_ap_err` 标志将被设置。为了确定如何配置 `ECCCFG0.ecc_ap_err_threshold`，用户必须首先确定一个突发中 64 位字的数量（1 个突发中 ECC 检查的总数量）。要确定这一点需要两个参数。

- DRAM 数据宽度：16 位或 32 位
- DRAM 突发长度（DRAM BL）
  - LPDDR4 使用的突发长度为 16
  - DDR3L 使用的突发长度为 8

确定一个突发中 ECC 检查总数的公式：

- 一个突发中 ECC 检查的总数 =（DRAM 数据宽度 x DRAM BL）/64。

下表显示了每个支持的内存和 DRAM 数据宽度的“一个突发中 ECC 检查的总数”：

表 3. 基于内存类型和 DRAM 数据总线宽度的一个突发中，ECC 检查的总数

内存类型和突发长度	16 位数据总线宽度	32 位数据总线宽度
LPDDR4 (BL16)	4	8
DDR3L (BL8)	2	4

为了检测 ECC AP 错误，建议将 `ECCCFG0.ecc_ap_err_threshold` 设置为一个突发内 ECC 检查的总数 – 1。下表显示了基于内存类型和 DRAM 数据宽度的 `ECCCFG0.ecc_ap_err_threshold` 的推荐设置。

表 4. 基于内存类型和 DRAM 数据总线宽度的 ECCFG0.ecc_ap_err_threshold 的推荐设置

内存类型和突发长度	16 位数据总线宽度	32 位数据总线宽度
LPDDR4 (BL16)	3	7
DDR3L (BL8)	1	3

注意

SoC 特定的 RPA 工具将 `ECCCFG0.ecc_ap_err_threshold` 字段预先配置为推荐值，因此不需要用户进一步互动。

3.2.7.2 ECC 错误报告示例

本节提供了各种 ECC 错误报告机制的说明性示例。

下面的例子是基于以下的系统条件：

- DRAM 类型：LPDDR4
- DRAM 数据宽度：16 位
- DRAM 突发长度（BL）：16
  - LPDDR4 突发长度 = 16
- 1 个突发中 ECC 检查的总数：4 个
  - （DRAM 数据宽度 x DRAM BL）/64 = (16 x 16)/64 = 4
- 建议 `ECCCFG0.ecc_ap_err_threshold` 设置：3
  - 在 1 个突发中的 ECC 检查总数：1

例 1：检测到 ECC 可纠正错误（1 位错误）

设置以下 ECC 状态位：

- `ECCSTAT.ecc_corrected_err` = 1

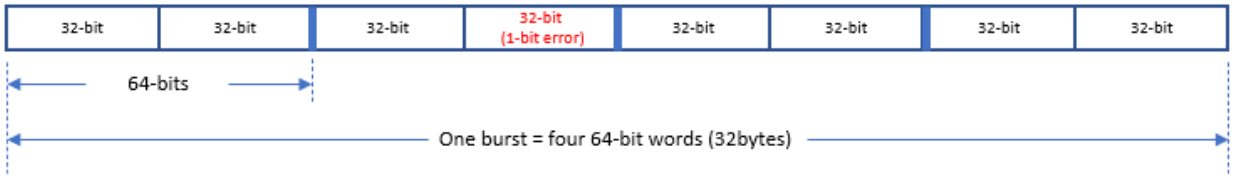

图 10. 检测到 ECC 可纠正错误（1 位错误）

例 2：检测到 ECC 不可纠正的错误（2 位错误）

设置以下 ECC 状态位：

- `ECCSTAT.ecc_uncorrected_err` = 1

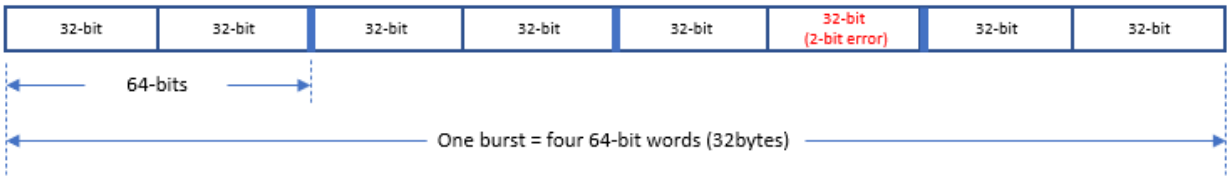

图 11. 检测到 ECC 不可纠正的错误（2 位错误）

例 3：检测到 ECC 可纠正和不可纠正的错误

设置以下 ECC 状态位：

- `ECCSTAT.ecc_corrected_err` = 1
- `ECCSTAT.ecc_uncorrected_err` = 1

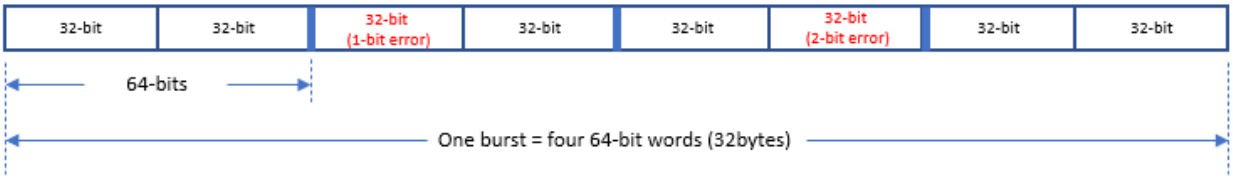

图 12. 检测到 ECC 可纠正和不可纠正的错误

例 4：检测到 ECC AP 错误

设置以下 ECC 状态位：

- `ECCSTAT.ecc_corrected_err` = 1
- `ECCSTAT.ecc_uncorrected_err` = 1
- `ECCAPSTAT.ecc_ap_err` = 1（超过 `ECCCFG0.ecc_ap_err_threshold` 设置：3）

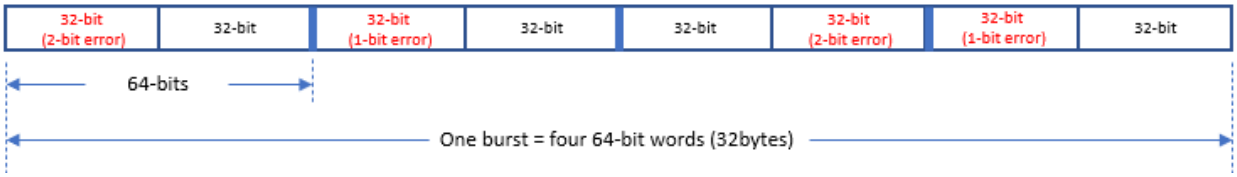

图 13. 检测到 ECC AP 错误

3.2.8 ECC 错误中断配置

上一节介绍了各种 ECC 错误以及它们产生中断的能力。本节介绍如何在 i.MX 8 DDR 控制器内启用和清除这些中断。本节不描述如何通过全局中断控制器（GIC）在 SoC 级别启用和处理中断。

用户可以在 ECC 清除寄存器（ECCCTL）中操作：

- 启用 ECC 错误报告中断
- 清除 ECC 错误和当前存储的 ECC 错误计数
- 强制发生 ECC 错误，以测试中断处理机制

下面的小节描述了如何在 ECCCTL 寄存器中启用各种 ECC 错误中断。对于下面的每一个小节，设置相应的位来启用所需的中断，并清除该位来禁用所需的中断：

- ecc_ap_err_intr_en (bit[10])：ecc_ap_err_intr 中断启用位
- ecc_uncorrected_err_intr_en (bit[9])：ecc_uncorrected_err_intr 中断启用位
- ecc_corrected_err_intr_en (bit[8])：ecc_corrected_err_intr 中断启用位

下面的小节描述了如何清除 ECCCTL 寄存器中的各种 ECC 中断和状态机制。对于下面的每一项，设置相应的位来清除所需的 ECC 状态：

- ecc_ap_err_intr_clr (bit[4])：ecc_ap_err 中断清除位
  - 设置后，ECCAPSTAT.ecc_ap_err 被清除
- ecc_uncorrected_err_clr (bit[1])：ecc_uncorrected_err 中断清除位
  - 设置后，ECCSTAT.ecc_uncorrected_err 被清除
- ecc_uncorr_err_cnt_clr (bit[3])：清除当前存储的未纠正的 ECC 错误计数
  - 设置后，ECCERRCNT.ecc_uncorr_err_cnt 被清除
- ecc_corrected_err_clr (bit[0])：ecc_corrected_err 中断清除位
  - 设置后，ECCSTAT.ecc_corrected_err 被清除
- ecc_corr_err_cnt_clr (bit[2])：清除当前存储的已纠正的 ECC 错误计数
  - 设置后，ECCERRCNT.ecc_corr_err_cnt 被清除

有一个可选的功能，允许用户强制中断以测试其系统软件的中断处理能力。对于以下每个小节，在 ECCCTL 寄存器中设置相应的位来强制执行所需的 ECC 错误中断：

- ecc_ap_err_intr_force (bit[18])
- ecc_uncorrected_err_intr_force (bit[17])
- ecc_corrected_err_intr_force (bit[16])

3.2.9 通过软件注入 ECC 错误

ECC 错误注入是一个有用的可选功能，用于系统级软件验证。与边带 ECC 不同，它没有专门的硬件支持。然而，通过“ECC_REGION_PARITY_LOCK”寄存器解锁 ECC 区域并覆盖 ECC 奇偶校验位，可以通过软件来注入错误。当从受保护的存储器区域读出相应的地址时，根据引入的错误类型，将会生成可纠正或不可纠正的 ECC 错误。关于这一功能的更多信息，可根据要求提供。

注意

DDR 控制器不支持 ECC 数据中毒。参考手册将被更新，以删除这一功能。

3.3 边带 ECC

边带 ECC 仅支持带 DDR3L 的 40 位（32+8）i.MX 8QuadXPlus/8DualXPlus。

当启用边带 ECC 时，一个额外的数据总线被用于 ECC。实际的 DRAM 数据宽度大于当前的“DRAM_DATA_WIDTH”。当启用时，它扩大了 DDR PHY 接口（DFI）的数据宽度，以适应额外的 ECC 字节。每 1 个 ECC 通道会增加 1 个 ECC 字节。

注意

边带 ECC 与内联 ECC 是相互排斥的。

将边带 ECC 添加到基于 LPDDR4 的系统中是很困难的，因为 JEDEC 标准要求 16 位 LPDDR4 内存设备。在这种配置下，边带 ECC 配置中有超过一半的内存未使用。ECC 字节不使用最上面设备的全部宽度。

将边带 ECC 添加到 DDR3L 并不那么困难，因为 8 位内存设备很容易获得。在边带 ECC 配置中，如果用“读取”命令检测到 1 位 ECC 错误，DDR 控制器可以发出 RMW 命令。

如果 `ECCCFG0.ecc_mode` 为“100”，1 位单错校正双错检测 ECC 被启用。在这种模式下，DDR 控制器会执行以下功能：

- 在写入时，在每个 ECC 通道上计算 ECC，得到的 ECC 代码与 ECC 通道中的数据一起作为一个额外的字节写入。这个额外的 ECC 字节总是被写入 DRAM 的最上面的字节中。
- 读取时，从 DRAM 中读取 ECC 通道（包括 ECC 字节）。然后对其进行解码。根据 ECC 通道中的数据，进行检查以验证 ECC 字节是否符合预期。如果它是正确的，数据就会被正常地发送到 SoC。

3.3.1 ECC 擦洗

ECC 擦洗功能仅支持边带 ECC 配置（i.MX 8QuadXPlus 和 i.MX 8DualXPlus - DDR3L - 40（32+8）位）。可以通过将 `ECCCFG0.dis_scrub` 设置为 0 来启用它。同样的寄存器位可以用来禁用该功能。当这个功能被启用时，DDR 控制器会在检测到单个位错误时，会安排 ECC 擦洗操作。对导致单个位错误的位置，擦洗是作为一个新的 RMW 操作来执行的。

当检测到一个单个位错误时，通过导致单个位错误的地址和写与读内容可寻址内存（CAM）中分配保留的条目，安排一个 RMW 操作。与常规的 RMW 请求一样，只有在擦洗 RMW 的“读”部分被安排好、读数据被返回、被 ECC 解码器纠正，并写入写数据缓冲区的适当位置后，擦洗 RMW 的 "写" 部分才会被启用。

在任何时候，只允许有一个未完成的 ECC 擦洗操作。当擦洗操作待处理时，由读解码器引擎检测到的一个新的单个位错误不会导致 ECC 擦洗操作的启动。控制器不能同时处理一个以上的擦洗操作。ECC 擦洗 RMW 操作是由控制器发起的。没有为擦洗 RMW 操作发送任何读响应（因为这不是由 SoC 核发起的）。

注意

在内联 ECC 配置（i.MX 8MPlus）中，ECC 擦洗功能被禁用，并且不会对每个可纠正的读错误自动擦洗。（`ECCCFG0.dis_scrub` 应被设置为 1）。

4 ECC 应用考虑因素

虽然 ECC 保护似乎是提高数据完整性的双赢方案，但在使用 ECC 功能时，有几个重要的考虑因素。本节简要介绍一些重要的系统级设计考虑因素，这些考虑因素可能会产生不同的影响，具体取决于最终用户应用程序和产品目标。

4.1 要保护什么？

需要完整性保护的代码和数据是 ECC 保护的主要候选对象。正如下面几节所描述的，ECC 保护大段的 DRAM 会对启动时间和性能产生影响。强烈建议用户优化要保护的数据，并将其限制为尽可能小的内存占用。应使用功能安全评估来确定哪些代码应通过 ECC 来保护。对于这样的应用程序，确保整个数据路径得到保护也很重要。

ECC 只能保证数据的完整性，不应用作保护代码或实现数据真实性的机制。ECC 可以用来确保代码中没有数据损坏，可用于安全应用程序，如 Arm Trusted Firmware 或其他加密的数据。ECC 不应该用于防范其他侧信道攻击。ECC 和/或增加刷新率可能会使一些侧信道技术更难使用。

4.2 错误报告和行动

DDR 控制器使用中断提供 ECC 报告机制。当发现 ECC 错误时，可以启用几个中断来指示。终端用户可以开发与这些错误相对应的操作。所需的行动取决于 ECC 错误的类型以及 ECC 功能在终端应用程序中的使用方式。

4.3 性能

ECC 块被映射，以便数据的 ECC 总是被映射到同一个页面（bank 和行），以提高 DDR 效率（因为每个数据访问可能有一个单独的 ECC 访问）。ECC 访问通常与它们相关的 bank 访问一起被优化。如果你有多个连续数据的访问，ECC 操作会被合并。

即使有这些优化，使用内联 ECC 也会对性能产生高达 25%的影响，这将取决于不同的访问类型和使用的 DDR 类型。它取决于单次读取与突发读取、部分写入与突发写入、首次读取与从缓冲区读取等等。对于写操作，我们大约有 90%的效率。对于读取，有 4-8 个 DDR 周期的延迟影响，数据传输效率下降到 90%。

注意

对于没有启用 ECC 的 DDR 的任何部分，没有相关的内存性能影响。性能开销只与 ECC 区域有关。受保护的区域应该被最小化。

用户应该预期到性能会下降，评估这是否可以接受，并根据其具体应用程序确定性能。边带 ECC 也会出现类似的性能下降。

4.3.1 功率

启用 ECC 可能会因为纠错电路而导致额外的功耗，并且在相同的工作负载下会有更多的能量消耗。通过限制需要保护的 ECC 区域，可以将功率开销降到最低。

启用洗涤器功能后，低功耗的进入和退出时间也可能增加。用户必须有一些控制措施来确定洗涤器在低功耗模式下的操作。默认配置使用自动硬件控制的低功耗功能。

4.4 启动时间延迟

当启用 ECC 时，必须启用 ECC 洗涤器。它执行强制性的 ECC 初始化，从而增加了启动时间。有启动时间延迟要求的客户应该记住，在启用这个功能时，会产生额外的延迟。

对整个内存映射进行 ECC 保护可能会显著增加初始化时间。恩智浦建议尽可能地限制 ECC 保护的区域，以减少启用 ECC 时初始化 DDR 的时间。

示例：

- 洗涤器时间是基于恩智浦在 1200 MHz 下运行的 LPDDR4 电路板。
- 基于恩智浦 LPDDR4 EVK 和 8Gb（1GB）配置，tRFC 为 280 ns。

表 5. 启动时间的延迟

ECC 粒度	1 个区域的大小	1 个区域的洗涤器时间（µs）	每 MB 计算的 1 个区域的洗涤器时间（µs）
1/8	128	34601	270.32
1/16	64	17301	270.33
1/32	32	8652	270.38
1/64	16	4327	270.44

注意

在上面的例子中，洗涤器速率大约是每 270µs/MB。对于一个 1/64 区域的 ECC 粒度，所有区域都是 ECC 保护的，并且其他区域也是保护的（受保护的 DDR 是 896 MB），总时间是 242 ms。

因为洗涤器时间取决于内存的 tRFC 值，更高的内存密度（有更大的 tRFC 值）有更长的初始化时间。

4.5 内存

下面介绍在启用 ECC 时，必须评估的几个内存考虑因素。

4.5.1 二进制/非二进制密度

如 ECC 中所述，用非二进制密度的 DDR（如 3GB 或 6GB）启用 ECC 需要额外考虑。非二进制密度要求保留多个部分，并以不连续的内存部分分割 DDR 内存空间。这就要求软件应用程序绕过这些不可访问的区域（孔）。要确保你的特定应用程序不访问这些区域。否则，你会遇到数据中止的情况。

内存利用率和不可访问区域增加，迫使更高比例的 DRAM 内存不能被利用。由于这种复杂性和内存利用率的降低，强烈建议使用二进制内存密度（如仅 1GB、2GB 和 4GB）来使用内联 ECC 功能。

4.5.2 密度

当启用内联 ECC 时，至少预留 1/8 的总可用 DRAM 密度用于奇偶校验数据。预留的内存量根据 ECC 保护区域的不同而不同。已经提供了尽量减少保留内存区域的选项。在本文档中将进一步讨论关于减少保留区域的其他影响。对于边带 ECC，需要一个单独的 DRAM 存储器，存储器的密度也必须作相应调整。

4.5.3 连续的内存映射

正如 ECC 中所描述的，ECC 保护的内存部分的配置可以影响到应用软件无法访问的 DRAM 内存地址范围或区域。让我们考虑一个例子，用户想要 ECC 保护一个 16 MB 的单一区域。RPA 被配置为保护区域 6，从地址 0x8600 0000 开始。

Main Memory Region	Start Address of each Main Memory Region	Density of each Main Memory Region	User Input ECC Protection Configuration for each Main Memory Region
Other Region	0x087000000	784MB	UNPROTECTED
Region6	0x086000000	16MB	PROTECTED

图 14. RPA 区域

不可访问的奇偶校验区将是一个 2 MB 的内存区，从地址 0xBF20 0000 开始。

Main Memory Region	Start Address of each Main Memory Region	Density of each Main Memory Region	User Input ECC Protection Configuration for each Main Memory Region
Other Region	0x087000000	784MB	UNPROTECTED
Region6	0x086000000	16MB	PROTECTED

图 15. 不可访问的奇偶校验区域

不可访问的 ECC 奇偶校验区 6 部分位于内存映射的中间，为应用程序创建了一个不连续的不可访问的内存。

注意

对于连续的内存空间，我们建议将 DRAM 内存地址范围末端的相应的奇偶校验部分从区域 0 中保护出来并保留。

在图 16 中，只有区域 0 是 ECC 保护的，它只保留了 ECC 奇偶校验区域 0，剩下的邻近区域是可用的。

ECC Parity Region Section	Start Address of each ECC Parity Region Section	Density of each ECC Parity Region Section	ECC Parity Region Section memory attributes
ECC Parity Region 0 Section	0x0BFE00000	2MB	INACCESSIBLE
ECC Parity Region 1 Section	0x0BFC00000	2MB	ACCESSIBLE
ECC Parity Region 2 Section	0x0BFA00000	2MB	ACCESSIBLE

图 16. 配置示例

4.5.4 U-Boot 配置

用户必须仔细构造他们的软件和 U-Boot 配置，以确保应用程序不会访问保留的不可访问区域。如果没有实现这一点，就会发生数据中止。

配置示例如下：

- 定义 DRAM 密度
- 保留 ECC 部分

```
memory@80000000
    { device_type = "memory";
    - reg = <0x00000000 0x80000000 0 0x40000000>;
    + reg = <0x00000000 0x80000000 0 0x20000000>;
    /* DRAM space - 1, size : 1 GB DRAM */
 };
    @@ -104,6 +104,13 @@
    no-map;
    reg = <0 0x90400000 0 0x1C00000>;
    };
+ /* top 128 MB reserved for ECC */
+ /*
+ ecc_reserved: ecc@0xb8000000 {
    + no-map;
    + reg = <0 0xb8000000 0 0x08000000>;
+ };
+ */
/*global autoconfigured region for contiguous allocations*/
linux,cma {
    compatible = "shared-dma-pool";
```

4.5.5 复位矢量

i.MX 8 对在 DDR 中执行的起始地址施加了限制。启动代码（通常是 UBOOT 或 ATF）必须位于 DDR 的起始位置。因此，用户不能从 DDR 基址的这个起始位置放置他们的应用程序代码，以进行 ECC 保护。相反，可以使用 UBOOT 和 ATF 代码之后的第一个空闲区域。

4.5.6 物料清单成本

根据设备和 ECC 方案的不同，可能会产生额外的物料清单（BOM）成本。

- 对于边带 ECC，需要一个单独的 DRAM 设备。在系统和制造成本中，必须考虑额外的电路板空间和布局。
- 对于内联 ECC，可能需要一个密度（容量）更高的 DRAM，以考虑到不可访问的保留区域。这可能需要更高的 DRAM 密度（与基于非 ECC 的应用程序相比），增加系统的物料清单（BOM）成本。

恩智浦可以帮助用户确定适合其特定应用程序的最佳存储器，以降低系统物料清单（BOM）成本。

5 参考资料

- i.MX 8M 系列 DDR 工具发布：https://community.nxp.com/docs/DOC-340179
- i.MX 8/8X 系列 DDR 工具发布：https://community.nxp.com/docs/DOC-346060
- i.MX 8 系列参考手册可在以下网站获得：www.nxp.com

6 修订历史

表 6. 修订历史

版本号	日期	实质性变更
0	2022 年 4 月 1 日	初版发布

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Limited warranty and liability — 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.

Right to make changes - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX,EMBRACE, GREENCHIP, HITAG, 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, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. 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. M, M Mobileye and other Mobileye trademarks or logos appearing herein are trademarks of Mobileye Vision Technologies Ltd. in the United States, the EU and/or other jurisdictions.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 01 April 2022
Document identifier: AN13566