

Rockchip RT-Thread Cache ECC

文件标识: RK-KF-YF-163

发布版本: V1.0.0

日期: 2021-05-28

文件密级: 绝密 秘密 内部资料 公开

免责声明

本文档按“现状”提供, 瑞芯微电子股份有限公司(“本公司”, 下同)不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因, 本文档将可能在未经任何通知的情况下, 不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标, 归本公司所有。

本文档可能提及的其他所有注册商标或商标, 由其各自所有者所有。

版权所有 © 2021 瑞芯微电子股份有限公司

超越合理使用范畴, 非经本公司书面许可, 任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部, 并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址: 福建省福州市铜盘路软件园A区18号

网址: www.rock-chips.com

客户服务电话: +86-4007-700-590

客户服务传真: +86-591-83951833

客户服务邮箱: fae@rock-chips.com

前言

概述

本文提供 RT-Thread 平台的 Cache ECC 测试方法。

产品版本

芯片名称	内核版本
RK356X	RT-Thread

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

版本号	作者	修改日期	修改说明
V1.0.0	谢修鑫	2021-05-28	初始版本

目录

Rockchip RT-Thread Cache ECC

1. 功能支持
2. 操作权限说明
3. 寄存器说明
4. 中断
 - 4.1 中断类型说明
 - 4.2 异常中断注册
5. 操作
 - 5.1 各个 Features 使能
 - 5.2 出错时信息获取
 - 5.3 错误注入测试
6. 测试 Demo

1. 功能支持

Cache ECC 操作需要在安全环境下进行，由于客户需要进行评估，所以这里提供的相关操作只是为了支持客户评估使用。

- Cache ECC 对应的 ARM 内部寄存器的读写操作
- Cache ECC 中断
- Cache 错误注入测试

2. 操作权限说明

AArch64 EL1 模式或 AArch32 SVC 等模式下没有权限访问 ECC 相关的寄存器；如果客户需要使用本文介绍的功能测试或开发 Cache ECC 相关功能，需求申请在 Trusted Firmware-A (TF-A) 中开放 AArch64 EL1 模式或 AArch32 SVC 等模式下访问 Cache ECC 相关寄存器的权限。开放权限的 TF-A 涉及安全问题，建议只是测试使用。

3. 寄存器说明

Cache ECC 寄存器涉及到两个模块，L1-L2 对应的 ECC 寄存器和 DSU-L3 对应的 ECC 寄存器。下面以 AArch64 的寄存器为例进行说明。

Cache ECC 的寄存器在 ARM 中只有一套，所以配置 L1-L2 还是 DSU-L3 的 ECC 寄存器时需要通过寄存器 ERRSELR_EL1 进行选择控制。以 Cache 出错信息寄存器 ERXSTATUS_EL1 为例：

- ERRSELR_EL1 为 0 时，ERXSTATUS_EL1 对应的是 L1-L2 的寄存器 ERR0STATUS
- ERRSELR_EL1 为 1 时，ERXSTATUS_EL1 对应的是 DSU-L3 的寄存器 ERR1STATUS

ECC 相关的寄存器说明如下：

- ERXCTLR_EL1：对应寄存器 ERR0CTLR 或 ERR1CTLR，控制 ECC 各个 Features 的使能。
- ERXFR_EL1：对应寄存器 ERR0FR 或 ERR1FR，显示 ECC 支持的 Features。
- ERXCFGCTLR_EL1：对应寄存器 ERR0CFGCTLR 或 ERR1CFGCTLR，控制注入 Cache 异常进行 ECC 测试。
- ERXCFGFR_EL1：对应寄存器 ERR0CFGFR 或 ERR1CFGFR，显示支持注入 Cache 异常的类型。
- ERXSTATUS_EL1：对应寄存器 ERR0STATUS 或 ERR1STATUS，显示 Cache 出错后出错信息，获取 Cache 出错原因。
- ERXMISC0_EL1：对应寄存器 ERR0MISC0 或 ERR1MISC0，是对 ERXSTATUS_EL1 信息的补充。

相关的寄存器说明参考 ARM 官方文档：

- [Arm DynamIQ Shared Unit Technical Reference Manual](#)
- [Arm Cortex-A55 Core Technical Reference Manual](#)

4. 中断

4.1 中断类型说明

检测到 Cache 出错后，会产生 fault 或 error 的 IRQ，下面是 A55 的文档说明：

If enabled in the ERXCTLR/ERXCTLR_EL1 register, all errors that are detected cause a fault handling interrupt. The fault handling interrupt is generated on the nFAULTIRQ[0] pin for L3 and snoop filter errors, or on the nFAULTIRQ[n+1] pin for core n L1 and L2 errors.

Uncorrectable errors in the L1 tag or dirty RAMs, or in the L2 tag RAMs, causes the nERRIRQ[n + 1] pin to be asserted for core n, if enabled.

Uncorrectable errors in the L3 tag RAMs or SCU snoop filter RAMs causes the nERRIRQ[0] pin to be asserted, if enabled.

中断说明：

- FAULTIRQ[0]的信号对应的是 DSU-L3 的 CACHE。
- nFAULTIRQ[n+1]对应的是各个 CPU 的 L1-L2 的异常，nFAULTIRQ[1]即对应的是 CPU0。
- nERRIRQ[0]的信号对应的是 DSU-L3 的 CACHE。
- nERRIRQ[n+1]对应的是各个 CPU 的 L1-L2 的异常，nERRIRQ[1]即对应的是 CPU0。

RK356X 对应的 Cache ECC 中断：

```
NFAULT0_IRQn      = 272,      /*!< DSU L3 CACHE ECC FAULT Interrupt */
NFAULT1_IRQn      = 273,      /*!< CPU0 L1-L2 CACHE ECC FAULT Interrupt */
NFAULT2_IRQn      = 274,      /*!< CPU1 L1-L2 CACHE ECC FAULT Interrupt */
NFAULT3_IRQn      = 275,      /*!< CPU2 L1-L2 CACHE ECC FAULT Interrupt */
NFAULT4_IRQn      = 276,      /*!< CPU3 L1-L2 CACHE ECC FAULT Interrupt */
NERR0_IRQn        = 277,      /*!< DSU L3 CACHE ECC ERROR Interrupt */
NERR1_IRQn        = 278,      /*!< CPU0 L1-L2 CACHE ECC ERROR Interrupt */
NERR2_IRQn        = 279,      /*!< CPU1 L1-L2 CACHE ECC ERROR Interrupt */
NERR3_IRQn        = 280,      /*!< CPU2 L1-L2 CACHE ECC ERROR Interrupt */
NERR4_IRQn        = 281,      /*!< CPU3 L1-L2 CACHE ECC ERROR Interrupt */
```

异常出错后可以在 nFAULTIRQ 的中断回调函数中通过 ERXSTATUS_EL1 & ERXMISC0_EL1 获取 ECC 出错信息。

4.2 异常中断注册

RT-Thread 中通过下面函数注册所有的 CACHE ECC 中断：

```
int rk_cache_ecc_init(uint32_t err1_irq_cpu)
```

- rk_cache_ecc_init()的参数 err1_irq_cpu 用于指定 DSU-L3 的 FAULTIRQ[0]和 nERRIRQ[0]中断关联到那个 cpu 上面。
- 每个 CPU 的 L1-L2 对应的 nFAULTIRQ[n+1] 和 nERRIRQ[n+1]中断关联到自己的 CPU。

RT-Thread 对应的中断函数中会打印 Cache 出错时寄存器 ERXSTATUS_EL1、ERXMISC0_EL1 值，可以通过这些寄存器的值判断出错原因。

5. 操作

5.1 各个 Features 使能

通过下面函数控制 ECC 各个 Features 的使能；Trusted Firmware-A 中会进行配置，所以 RT-thread 中不需要配置。如果客户有特殊要求，可以通过这个函数进行更改。

```
HAL_Status HAL_CACHE_ECC_SetErxctlr(uint32_t err0ctlr, uint32_t err1ctlr)
```

- 参数 err0ctlr: 控制 L1-L2 ECC Features 的使能。
- 参数 err1ctlr: 控制 DSU-L3 ECC Features 的使能。

5.2 出错时信息获取

通过下面两个函数获取 ERXSTATUS_EL1、ERXMISC0_EL1 的信息。

```
uint32_t HAL_CACHE_ECC_GetErxstatus(eCACHE_ECC_RecodeID errSel)  
uint32_t HAL_CACHE_ECC_GetErxmisc0(eCACHE_ECC_RecodeID errSel)
```

参数 errSel: 用于选择获取的是 L1-L2 还是 DSU-L3 对应的寄存器。

5.3 错误注入测试

由于外部无法模拟 Cache 出错，无法进行 Cache ECC 测试；ARM 提供 Cache 错误注入功能，可以通过该功能进行 Cache ECC 功能的验证。正是由于 Cache 错误注入功能的存在，所以该文档提供的操作只能作为客户验证 Cache ECC 功能使用。发布版本的 Trusted Firmware-A 不会开放 Cache ECC 的操作权限。

通过下面函数向 Cache 中注入错误

```
HAL_Status HAL_CACHE_ECC_Inject(eCACHE_ECC_RecodeID errSel, eCACHE_ECC_InjectFault  
injectFault)
```

- 参数 errSel: 选择向 L1-L2 还是 DSU-L3 注入错误。
- 参数 injectFault: 注入错误的类型。

6. 测试 Demo

RT-Thread 操作 Cache ECC 时，需要先申请开放 Cache ECC 相关寄存器操作权限的 Trusted Firmware-A, 对应的 Trusted Firmware-A 启动时会打印如下信息：

```
INFO:    Cache ECC registers are write-accessible from EL1 Non-secure.
```

通过在 RT-Thread 中加入下面代码进行 Cache ECC 测试：

```
rk_cache_ecc_init(0);  
HAL_CACHE_ECC_Inject(CACHE_ECC_ERR0, CACHE_ECC_INJECT_UC);
```

- 函数 `rk_cache_ecc_init()`: 注册 Cache 出错时通知 IRQ 的 handler。
- 函数 `HAL_CACHE_ECC_Inject()`: 向 Cache 注入错误。