

Rockchip Developer Guide HAL DDR ECC

文件标识：RK-KF-YF-169

发布版本：V1.0.0

日期：2021-03-29

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，由其各自拥有者所有。

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

本文主要描述了HAL 裸系统下DDR ECC的原理和使用方法。

产品版本

| 芯片名称 | 内核版本 |
|--------|------|
| RK356X | HAL |

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 | 何智欢 | 2021-03-29 | 初始版本 |

目录

Rockchip Developer Guide HAL DDR ECC

1. 名词解释
2. 简介
3. 开启DDR ECC
4. HAL里获取DDR ECC信息
 - 4.1 配置
 - 4.2 代码和API
 - 4.3 使用范例

1. 名词解释

| 本文档的简写 | 本文档内释义 |
|--------------|--|
| ECC | Error Correcting Code |
| SEC ECC | Single Bit Single Error Correction Code |
| DED ECC | Double Error Detection Error Correction Code |
| DDR | Double Data Rate SDRAM |
| CE | Correctable Error，指单bit可检测可纠正性错误 |
| UE | Uncorrectable Error，指双bit可检测不可纠正性错误 |
| cs | chip select |
| Row | 特指 DDR row 地址 |
| Chip ID | 特指 DDR chip id，未激活功能，请忽略 |
| BankGroup | 特指 DDR4 Bank Group 地址，其他DDR类型请忽略 |
| Bank | 特指 DDR bank 地址 |
| Col | 特指 DDR column 地址 |
| Bit position | 特指 CE 纠正的bit位 |

2. 简介

ECC 指 Error Correcting Code ，而DDR ECC 是对DDR数据进行错误检查和纠正的。RK3568只支持 SEC/DED ECC。目前仅支持SideBand ECC，即在DDR数据通道旁增加一个专门存放ECC数据的DDR通道。32bit位宽的DDR至少需要7bit位宽的ECC，16bit位宽的DDR至少需要6bit位宽的ECC，8bit位宽的DDR至少需要5bit位宽的ECC。PCB设计请参考带DDR ECC的设计，如 RK_EVB6_RK3568_DDR3P416_ECCP216_DD6_V10。

3. 开启DDR ECC

对于满足SideBand ECC要求的DDR通道，loader会识别出这种设计，并自动使能DDR ECC。

4. HAL里获取DDR ECC信息

DDR ECC 具体的错误检查与纠正行为是硬件算法完成，软件可以获取相关信息。

4.1 配置

在相应工程的 hal_conf.h 下使能DDR ECC模块。如rk3568，在project/rk3568/src/hal_conf.h添加如下代码：

```
#define HAL_DDR_ECC_MODULE_ENABLED
```

4.2 代码和API

- lib/hal/src/hal_ddr_ecc.c
- lib/hal/inc/hal_ddr_ecc.h

```
/* 初始化DDR ECC相关信息 */
HAL_Status HAL_DDR_ECC_Init(struct DDR_ECC_CNT *p);

/* 获取DDR ECC累计的统计信息，包括单bit可纠正错误的数量和双bit可检测不可纠正错误的数量 */
HAL_Status HAL_DDR_ECC_GetInfo(struct DDR_ECC_CNT *p);
```

4.3 使用范例

上层软件可以用两种方式获取DDR ECC信息：软件轮询和硬件中断。

- 软件轮询方式

示例如下：

```
struct DDR_ECC_CNT eccInfo;

void HAL_DDR_ECC_TEST_POLL(void)
{
    uint32_t cpuID;

    cpuID = HAL_CPU_TOPOLOGY_GetCurrentCpuId();
    if (cpuID == 0) {
        /* 使用一个cpu、线程或其他方式，初始化并轮询DDR ECC状态 */
        HAL_DDR_ECC_Init(&eccInfo);
        while (1) {
            /* 初始化DDR ECC信息之后，轮询获取DDR ECC信息 */
            HAL_DDR_ECC_GetInfo(&eccInfo); /* 累计的CE和UE数量存放在结构体eccInfo中 */
            HAL_DelayMs(50);                /* 轮询间隔时间，可用其他让cpu空闲的API */
        }
    }
}
```

- 硬件中断方式

示例如下：

```
struct DDR_ECC_CNT eccInfo;

void HAL_DDR_ECC_IRQHandler(uint32_t irq)
{
    HAL_DDR_ECC_GetInfo(&eccInfo);
}

void HAL_DDR_ECC_TEST_INT(void)
{
    uint32_t cpuID;

    cpuID = HAL_CPU_TOPOLOGY_GetCurrentCpuId();
    if (cpuID == 0) { /* 使用一个cpu、线程或其他方式，初始化DDR ECC相关 */
        HAL_DDR_ECC_Init(&eccInfo);
        HAL_GIC_SetHandler(DDR_ECC_CE_IRQn, HAL_DDR_ECC_IRQHandler); /* 挂载CE中断服务子程序 */
        HAL_GIC_SetHandler(DDR_ECC_UE_IRQn, HAL_DDR_ECC_IRQHandler); /* 挂载UE中断服务子程序 */
        HAL_GIC_Enable(DDR_ECC_CE_IRQn); /* 使能CE中断服务 */
        HAL_GIC_Enable(DDR_ECC_UE_IRQn); /* 使能UE中断服务 */
    }
}
```

- 若检测到ECC出错，则打印获取的ECC信息。

```
# 检测到CE（可纠正错误）2个
[HAL WARNING] DDR ECC error: CE, 2 errors, the last is in DDR cs 0, Row 0xa0,
ChipID 0x0, BankGroup 0x0, Bank 0x5, Col 0x318, Bit position 0x10000000

# 检测到UE（不可纠正错误）1个
[HAL ERROR] DDR ECC error: UE, 1 errors, the last is in DDR cs 0, Row 0xa0,
ChipID 0x0, bankGroup 0x0, Bank 0x5, Col 0x354
```