

# SOC 用户指南

文档版本 V3.0

发布日期 2025-12-01

# 前言

## 概述

本文档介绍了 XM7206V1 系列芯片的特性、逻辑结构，详细描述各个模块的功能、工作方式、相关寄存器定义，用图表的方式给出了接口时序关系和相关参数，并详细描述了芯片的管脚定义和用途以及芯片的性能参数和封装尺寸。



说明

本文中描述的芯片均为 XM7206V1 系列芯片。

## 产品版本

与本文档相对应的产品版本如下。

| 产品名称 | 产品版本 |
|------|------|
|      |      |

## 读者对象

本文档主要适用于以下工程师：

- 电子产品设计维护人员
- 电子产品元器件市场销售人员

## 约定符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

| 符号                                                                                          | 说明                                                                     |
|---------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
|  <b>危险</b> | 表示如不可避免则将会导致死亡或严重伤害的具有高等级风险的危害。                                        |
|  <b>警告</b> | 表示如不可避免则可能导致死亡或严重伤害的具有中等级风险的危害。                                        |
|  <b>注意</b> | 表示如不可避免则可能导致轻微或中度伤害的具有低等级风险的危害。                                        |
| <b>须知</b>                                                                                   | 用于传递设备或环境安全警示信息。如不可避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。<br>“须知”不涉及人身伤害。 |
|  <b>说明</b> | 对正文中重点信息的补充说明。<br>“说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。                        |

## 表格内容约定

| 内容 | 说明                 |
|----|--------------------|
| -  | 表格中的无内容单元。         |
| *  | 表格中的内容用户可根据需要进行配置。 |

## 寄存器访问类型约定

| 类型  | 说明                  | 类型  | 说明                     |
|-----|---------------------|-----|------------------------|
| RO  | 只读，不可写。             | W0C | 可读，写 0 清零，写 1 保持不变。    |
| WO  | 只写。                 | W1S | 可读，写 1 置 1，写 0 保持不变。   |
| RW  | 可读可写。               | W0S | 可读，写 0 置 1，写 1 保持不变。   |
| RC  | 读清零。                | OSW | 可读，写 1 后片内自清零，即产生一个脉冲。 |
| W1C | 可读，写 1 清零，写 0 保持不变。 |     |                        |

# 数值单位约定

数据容量、频率、数据速率等的表达方式说明如下。

| 类别             | 符号 | 对应的数值         |
|----------------|----|---------------|
| 数据容量（如 RAM 容量） | 1K | 1024          |
|                | 1M | 1,048,576     |
|                | 1G | 1,073,741,824 |
| 频率、数据速率等       | 1k | 1000          |
|                | 1M | 1,000,000     |
|                | 1G | 1,000,000,000 |

地址、数据的表达方式说明如下。

| 符号 | 举例                  | 说明                              |
|----|---------------------|---------------------------------|
| 0x | 0xFE04、0x18         | 用 16 进制表示的数据值、地址值。              |
| 0b | 0b000、0b00 00000000 | 表示 2 进制的数据值以及 2 进制序列（寄存器描述中除外）。 |

# 修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

| 修订日期       | 版本   | 修订说明                                                                                         |
|------------|------|----------------------------------------------------------------------------------------------|
| 2025-06-20 | V1.0 | DI 版本发布。                                                                                     |
| 2025-08-05 | V2.0 | 正式版本发布。                                                                                      |
| 2025-12-01 | V3.0 | 第 1 章 系统<br>1.10.2.4 小节 增加 GPIO 复用掉电锁存功能使用流程。<br>1.2.7 小节 增加 PERI_140/141/142/143/144 寄存器描述。 |



# 目 录

|                 |          |
|-----------------|----------|
| <b>1 系统</b>     | <b>1</b> |
| 1.1 复位          | 1        |
| 1.1.1 概述        | 1        |
| 1.1.2 复位控制      | 1        |
| 1.1.3 复位配置      | 3        |
| 1.2 时钟          | 4        |
| 1.2.1 概述        | 4        |
| 1.2.2 功能框图      | 4        |
| 1.2.3 时钟资源分布    | 5        |
| 1.2.4 PLL 配置    | 9        |
| 1.2.5 注意事项      | 10       |
| 1.2.6 CRG 寄存器概览 | 10       |
| 1.2.7 CRG 寄存器描述 | 15       |
| 1.3 中央处理器 CPU   | 76       |
| 1.4 微型处理器 MCU   | 76       |
| 1.4.1 概述        | 76       |
| 1.4.2 特点        | 76       |
| 1.4.3 功能描述      | 77       |
| 1.4.4 工作方式      | 86       |
| 1.4.5 寄存器概览     | 87       |
| 1.4.6 寄存器描述     | 91       |
| 1.5 中断系统        | 106      |
| 1.6 DMA 控制器     | 108      |
| 1.6.1 概述        | 108      |

|                           |            |
|---------------------------|------------|
| 1.6.2 特点.....             | 108        |
| 1.6.3 功能描述.....           | 108        |
| 1.6.4 工作方式.....           | 112        |
| 1.6.5 DMA 寄存器概述.....      | 114        |
| 1.6.6 DMA 寄存器描述.....      | 115        |
| 1.7 定时器.....              | 130        |
| 1.7.1 概述.....             | 130        |
| 1.7.2 特点.....             | 130        |
| 1.7.3 功能描述.....           | 130        |
| 1.7.4 TIMER 寄存器概览.....    | 131        |
| 1.7.5 TIMER 寄存器描述.....    | 132        |
| 1.8 看门狗.....              | 135        |
| 1.8.1 概述.....             | 135        |
| 1.8.2 特点.....             | 135        |
| 1.8.3 功能描述.....           | 136        |
| 1.8.4 工作方式.....           | 137        |
| 1.8.5 WDG 寄存器概览.....      | 138        |
| 1.8.6 WDG 寄存器描述.....      | 139        |
| 1.9 实时时钟.....             | 141        |
| 1.9.1 概述.....             | 141        |
| 1.9.2 特点.....             | 142        |
| 1.9.3 功能描述.....           | 142        |
| 1.9.4 工作方式.....           | 143        |
| 1.9.5 RTC 寄存器概览.....      | 145        |
| 1.9.6 RTC 寄存器描述.....      | 148        |
| 1.10 电源管理与低功耗模式.....      | 160        |
| 1.10.1 概述.....            | 160        |
| 1.10.2 PMC.....           | 160        |
| <b>2 外围设备.....</b>        | <b>182</b> |
| 2.1 I <sup>2</sup> C..... | 182        |
| 2.1.1 概述.....             | 182        |
| 2.1.2 功能描述.....           | 182        |

|                                    |     |
|------------------------------------|-----|
| 2.1.3 功能框图.....                    | 183 |
| 2.1.4 时序描述原理 .....                 | 183 |
| 2.1.5 工作方式.....                    | 192 |
| 2.1.6 I <sup>2</sup> C 寄存器概览.....  | 196 |
| 2.1.7 I <sup>2</sup> C 寄存器描述 ..... | 197 |
| 2.2 UART .....                     | 212 |
| 2.2.1 概述.....                      | 212 |
| 2.2.2 特点.....                      | 212 |
| 2.2.3 功能描述.....                    | 213 |
| 2.2.4 工作方式.....                    | 214 |
| 2.2.5 UART 寄存器概览 .....             | 217 |
| 2.2.6 UART 寄存器描述 .....             | 218 |
| 2.3 SPI.....                       | 231 |
| 2.3.1 概述.....                      | 231 |
| 2.3.2 特点.....                      | 232 |
| 2.3.3 功能描述.....                    | 232 |
| 2.3.4 三种外设总线时序 .....               | 233 |
| 2.3.5 工作方式.....                    | 240 |
| 2.3.6 寄存器概览.....                   | 244 |
| 2.3.7 寄存器描述 .....                  | 245 |
| 2.4 eMMC/SD/SDIO 控制器 .....         | 253 |
| 2.4.1 功能描述.....                    | 253 |
| 2.4.2 应用说明.....                    | 262 |
| 2.4.3 寄存器概览.....                   | 265 |
| 2.4.4 寄存器描述.....                   | 267 |
| 2.5 GPIO .....                     | 304 |
| 2.5.1 概述.....                      | 304 |
| 2.5.2 特点.....                      | 304 |
| 2.5.3 工作方式.....                    | 304 |
| 2.5.4 GPIO 寄存器概览.....              | 306 |
| 2.5.5 GPIO 寄存器描述 .....             | 307 |
| 2.6 USB DRD .....                  | 311 |

|                             |     |
|-----------------------------|-----|
| 2.6.1 概述.....               | 311 |
| 2.6.2 功能描述.....             | 312 |
| 2.6.3 工作方式.....             | 313 |
| 2.6.4 USB2.0 PHY 寄存器概览..... | 314 |
| 2.6.5 USB2.0 PHY 寄存器描述..... | 316 |
| 2.6.6 USB 控制器寄存器概览 .....    | 330 |
| 2.6.7 USB 控制器寄存器描述 .....    | 333 |
| 2.7 LSADC .....             | 368 |
| 2.7.1 概述.....               | 368 |
| 2.7.2 特点.....               | 369 |
| 2.7.3 工作方式.....             | 369 |
| 2.7.4 LSADC 寄存器概览.....      | 372 |
| 2.7.5 LSADC 寄存器描述.....      | 373 |
| 2.8 PWM.....                | 382 |
| 2.8.1 概述.....               | 382 |
| 2.8.2 特点.....               | 382 |
| 2.8.3 工作方式.....             | 382 |
| 2.8.4 PWM 寄存器概览.....        | 383 |
| 2.8.5 PWM 寄存器描述 .....       | 384 |

## 插图目录

|                                            |     |
|--------------------------------------------|-----|
| 图 1-1 复位信号控制图.....                         | 2   |
| 图 1-2 时钟管理模块功能框图.....                      | 4   |
| 图 1-3 时钟资源分布框图 .....                       | 5   |
| 图 1-4 MCU 应用快速 AE 示意图.....                 | 78  |
| 图 1-5 MCU 子系统架构图 .....                     | 78  |
| 图 1-6 MCU 访问地址区间分配图 .....                  | 81  |
| 图 1-7 WatchDog 应用框图.....                   | 136 |
| 图 1-8 使用 PMC 时的电源网络粗略框图.....               | 161 |
| 图 1-9 不使用 PMC 功能时外围管脚的处理 .....             | 162 |
| 图 1-10 PWR_BUTTON、PWR_STARTUP 第一次上电时序..... | 163 |
| 图 1-11 PWR_BUTTON、PWR_STARTUP 上电时序.....    | 164 |
| 图 1-12 PWR_WAKEUP 上电时序（以上升沿触发为例） .....     | 164 |
| 图 1-13 RTC 定时中断上电时序 .....                  | 165 |
| 图 1-14 软件配置下电时序.....                       | 166 |
| 图 1-15 异常下电重启时序.....                       | 167 |
| 图 1-16 复位状态示意图.....                        | 168 |
| 图 2-1 控制器的功能框图 .....                       | 183 |
| 图 2-2 7bit 寻址，写操作时序图 .....                 | 186 |
| 图 2-3 7bit 寻址，直接读时序图 .....                 | 187 |
| 图 2-4 10bit 寻址，写操作时序图 .....                | 188 |

|                                                      |     |
|------------------------------------------------------|-----|
| 图 2-5 10bit 寻址，组合读时序图 .....                          | 189 |
| 图 2-6 非标准时序图 .....                                   | 191 |
| 图 2-7 UART 的典型应用框图.....                              | 213 |
| 图 2-8 UART 帧格式 .....                                 | 213 |
| 图 2-9 SPI 接 Slave 时的应用 .....                         | 233 |
| 图 2-10 SPI 单帧帧格式 (SPO=0、SPH=0) .....                 | 233 |
| 图 2-11 SPI 连续帧帧格式 (SPO=0、SPH=0) .....                | 234 |
| 图 2-12 SPI 单帧帧格式 (SPO=0、SPH=1) .....                 | 234 |
| 图 2-13 SPI 连续帧帧格式 (SPO=0、SPH=1) .....                | 235 |
| 图 2-14 SPI 单帧帧格式 (SPO=1、SPH=0) .....                 | 235 |
| 图 2-15 SPI 连续帧帧格式 (SPO=1、SPH=0) .....                | 236 |
| 图 2-16 SPI 单帧帧格式 (SPO=1、SPH=1) .....                 | 236 |
| 图 2-17 SPI 连续帧帧格式 (SPO=1、SPH=1) .....                | 237 |
| 图 2-18 SPI 接口时序图.....                                | 238 |
| 图 2-19 TI 同步串行单帧帧格式 .....                            | 238 |
| 图 2-20 TI 同步串行连续帧帧格式 .....                           | 238 |
| 图 2-21 National Semiconductor Microwire 单帧帧格式.....   | 239 |
| 图 2-22 National Semiconductor Microwire 连续帧帧格式 ..... | 239 |
| 图 2-23 MMC 功能框图.....                                 | 255 |
| 图 2-24 MMC 典型应用图 .....                               | 256 |
| 图 2-25 MMC 指令格式.....                                 | 257 |
| 图 2-26 MMC 指令响应格式 .....                              | 257 |
| 图 2-27 MMC 非数据指令操作.....                              | 258 |
| 图 2-28 单块与多块读操作 .....                                | 259 |
| 图 2-29 单块与多块写操作 .....                                | 260 |
| 图 2-30 1bit 数据线传输模式下的块数据格式 .....                     | 260 |
| 图 2-31 4bit 数据线传输模式下的块数据格式 .....                     | 261 |

|                               |     |
|-------------------------------|-----|
| 图 2-32 USB 2.0 DRD 逻辑框图 ..... | 312 |
| 图 2-33 单次扫描处理流程 .....         | 370 |
| 图 2-34 连续扫描模式下通道轮询扫描示意图 ..... | 370 |
| 图 2-35 连续扫描处理流程 .....         | 371 |
| 图 2-36 16 次平均算法处理流程 .....     | 372 |

# 表格目录

表 1-1 复位信号分类表..... 2

表 1-2 主要模块可选时钟 ..... 6

表 1-3 PLL 频率计算方法..... 9

表 1-4 时钟寄存器概览..... 10

表 1-5 MCU 初始访问地址区间与属性..... 82

表 1-6 异常类型定义表..... 84

表 1-7 中断向量表..... 85

表 1-8 中断寄存器概述表..... 85

表 1-9 通用寄存器表单..... 87

表 1-10 浮点寄存器表单..... 88

表 1-11 中断寄存器表单..... 88

表 1-12 机器模式寄存器表单..... 88

表 1-13 用户模式寄存器表单..... 90

表 1-14 中断源分配表..... 106

表 1-15 DMAC 外设硬件请求线编号说明 ..... 109

表 1-16 DMAC 访问空间说明 ..... 109

表 1-17 DMA 寄存器概览..... 114

表 1-18 Timer 寄存器概览..... 131

表 1-19 WatchDog 寄存器概览..... 138

表 1-20 RTC 寄存器概览 ..... 145



|                                        |     |
|----------------------------------------|-----|
| 表 1-21 电源对接表.....                      | 169 |
| 表 1-22 PMC 寄存器概览.....                  | 169 |
| 表 2-1 时序命令 .....                       | 184 |
| 表 2-2 7bit 寻址，写操作对应的时序描述配置.....        | 186 |
| 表 2-3 7bit 寻址，直接读操作的时序描述配置.....        | 187 |
| 表 2-4 10bit 寻址，写操作对应的时序描述配置.....       | 188 |
| 表 2-5 10bit 寻址，组合读对应的时序描述配置.....       | 190 |
| 表 2-6 非标准时序对应时序描述配置.....               | 191 |
| 表 2-7 I <sup>2</sup> C 寄存器偏移地址变量表..... | 196 |
| 表 2-8 I <sup>2</sup> C 寄存器概览.....      | 196 |
| 表 2-9 UART 寄存器概览.....                  | 217 |
| 表 2-10 SPI 寄存器概览.....                  | 245 |
| 表 2-11 芯片中 3 个 MMC 控制器对应的功能信号和管脚名..... | 254 |
| 表 2-12 传输模式表.....                      | 261 |
| 表 2-13 传输模式表.....                      | 261 |
| 表 2-14 MMC 寄存器概览.....                  | 265 |
| 表 2-15 芯片 GPIO 寄存器对应的基地址 .....         | 306 |
| 表 2-16 GPIO 寄存器概览.....                 | 307 |
| 表 2-17 USB PHY 寄存器映射表 .....            | 314 |
| 表 2-18 USB 寄存器概览 .....                 | 330 |
| 表 2-19 LSADC 寄存器概览.....                | 372 |
| 表 2-20 PWMn 的寄存器基地址表 .....             | 383 |
| 表 2-21 PWM0 寄存器概览.....                 | 384 |

# 1 系统

---

## 1.1 复位

### 1.1.1 概述

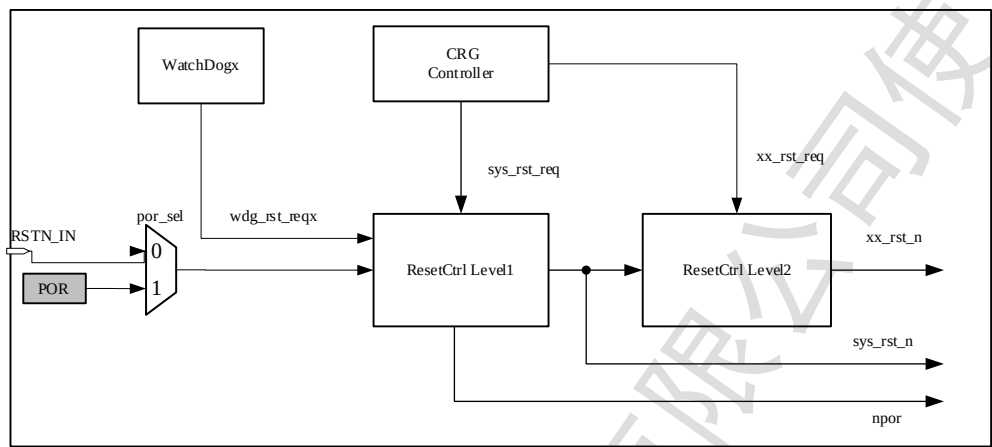
复位管理模块对整个芯片的复位、各功能模块的复位进行统一的管理，包括：

- 上电复位的管理和控制
- WatchDog 复位管理和控制
- 系统软复位、功能模块单独软复位控制
- 复位信号同步到各模块对应时钟域
- 生成芯片内部各功能模块的复位信号

### 1.1.2 复位控制

复位信号控制如图 1-1 所示。

图1-1 复位信号控制图



- POR

wdg\_rst\_req

sys\_rst\_req

xx\_rst\_req

xx\_rst\_n、sys\_rst\_n、

npor
- 芯片内部上电复位（Power-On-Reset）模块。

WatchDog 复位请求。

全局软复位请求信号，源自系统控制器。

子模块单独软复位请求信号，源自 CRG 控制寄存器。

复位信号。

全局硬复位

复位信号分类如表 1-1 所示。

表1-1 复位信号分类表

| 复位信号类型             | 产生方式               | 用途                                          |
|--------------------|--------------------|---------------------------------------------|
| 全局硬复位<br>npor      | 内部上电复位 POR 模块      | 对 SOC 系统进行全局复位。                             |
| 外部复位<br>RSTN_IN    | 外部复位               | 对 SOC 系统进行全局复位。                             |
| 全局软复位<br>sys_rst_n | 软件配置系统控制器的全局软复位寄存器 | 复位 SOC 系统中的所有模块，除了时钟复位电路，测试电路和部分不会被软复位的寄存器。 |

## 1.1.3 复位配置

### 1.1.3.1 上电复位

芯片分为内部 POR 复位和外部复位。内部复位和外部复位的切换由 PAD (POR\_SEL) 控制。

当使用外部复位时：por\_in\_n 必须至少维持 16 ms 时间低电平时间才可拉高。

完成内部上电复位过程必须同时满足以下条件：

- 内部 POR 模块产生一个低电平脉冲，且低电平维持时间大于 16ms。
- 晶振时钟输入管脚 XIN 输入的时钟稳定。

### 1.1.3.2 系统复位

实现系统复位有以下途径：

- 内部复位。
- 外部复位。
- WatchDog 复位。
- 全局软复位，通过系统控制器控制。

### 1.1.3.3 模块软复位

软复位控制通过配置相应的 CRG 控制器来实现，具体配置请参见每个模块的复位寄存器描述。

---

#### 须知

- 系统软复位请求发出后，电路必须等待至少 360 个系统时钟周期才完成复位撤消，在这 360 个系统时钟周期内不能再发系统软复位请求，否则系统状态混乱，可能无法完成复位操作。
  - 各模块单独软复位不会自动撤消，例如某模块的复位是配置 1 时，模块处于复位状态，必须再配置为 0，该模块复位才会撤消。
-

## 1.2 时钟

### 1.2.1 概述

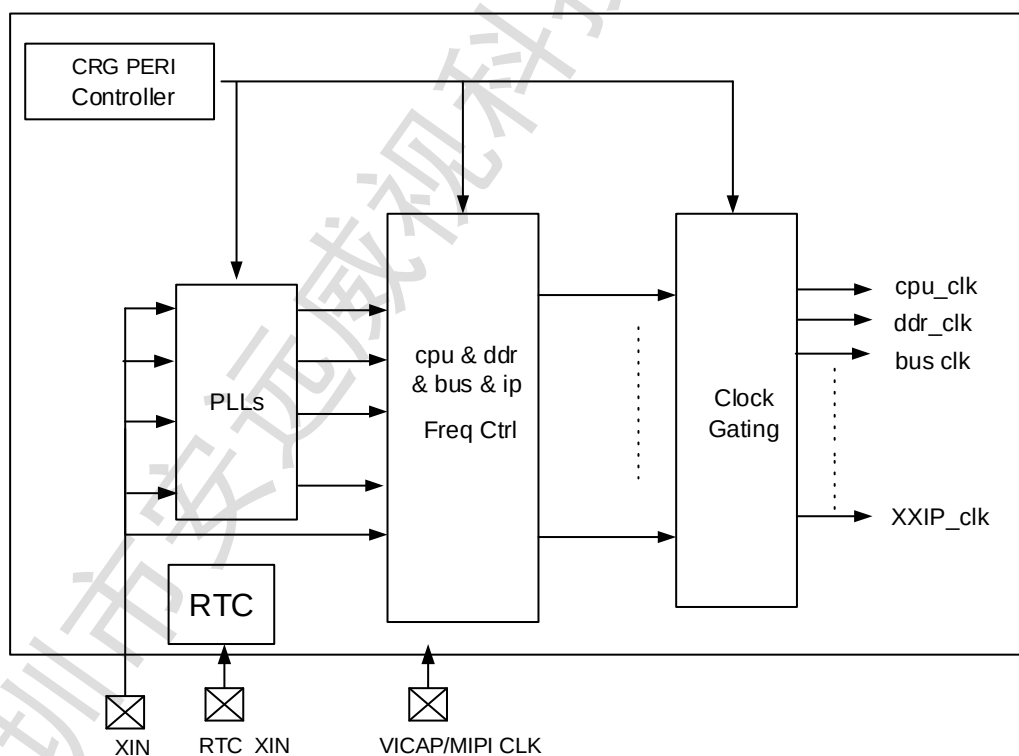
时钟管理模块对芯片时钟输入、时钟生成和控制进行统一的管理，包括：

- 时钟输入的管理和控制
- 时钟分频和控制
- 生成各模块的工作时钟

### 1.2.2 功能框图

时钟管理模块功能框图如图 1-2 所示。

图1-2 时钟管理模块功能框图

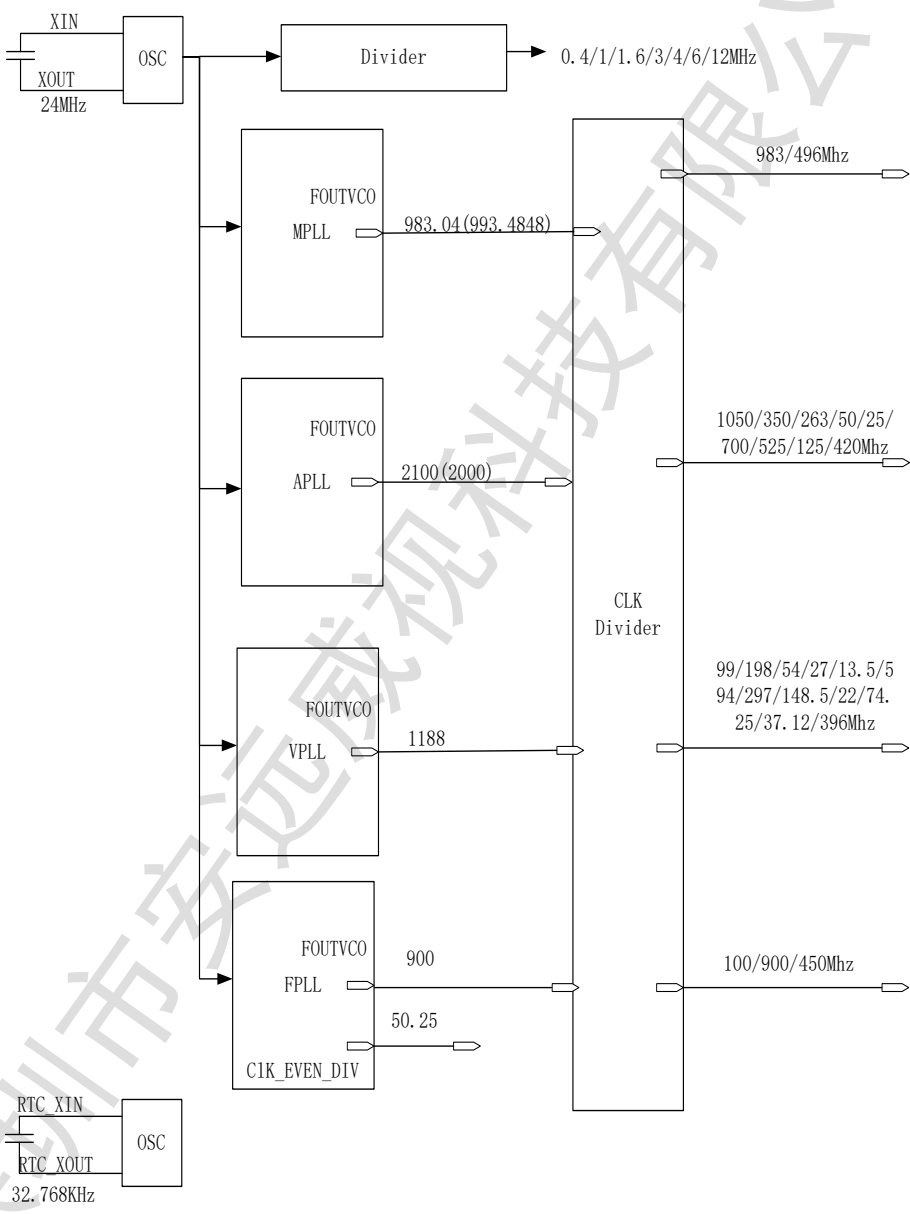


注：图中 XIN 为 PLL 输入时钟，固定连接 24MHz 晶体；RTC\_XIN 为 RTC 输入时钟，固定连接 32.768KHz 晶体。

### 1.2.3 时钟资源分布

时钟管理模块对源自芯片管脚的输入时钟和内部 PLL 进行配置、控制和管理，产生各模块所需的时钟资源，具体分布示意下图 1-3 所示。

图1-3 时钟资源分布框图



主要模块可选时钟列表，如表 1-2 所示，具体配置请参考各模块的时钟选择寄存器描述，其中 A7/SYSAXI/SYSCFG/SYSAPB/MDAAXI/FMC/MCU 的时钟选择电路使用了无毛刺切换逻辑。

表1-2 主要模块可选时钟

| 模块     | 时钟名称   | 默认频点   | 所有频点                                                                       |
|--------|--------|--------|----------------------------------------------------------------------------|
| CPU    | A7     | 24MHz  | 24M<br>1188MHz<br>993MHz<br>900MHz<br>700MHz<br>594MHz<br>396MHz<br>350MHz |
| 总线     | SYSAPB | 24MHz  | 24MHz<br>50MHz                                                             |
|        | SYSCFG | 24MHz  | 24MHz<br>99MHz                                                             |
|        | SYSAXI | 24MHz  | 24MHz<br>198MHz<br>150MHz<br>99MHz                                         |
|        | MDAAXI | 24MHz  | 24MHz<br>525MHz<br>496MHz<br>420MHz<br>350MHz                              |
| FMC    | FMC    | 24MHz  | 24MHz<br>99MHz<br>148.5MHz<br>198MHz<br>297MHz<br>396MHz                   |
| VIPROC | VIPROC | 300MHz | 300MHz<br>263MHz<br>198MHz<br>150MHz<br>100MHz                             |

| 模块     | 时钟名称     | 默认频点   | 所有频点                                                                |
|--------|----------|--------|---------------------------------------------------------------------|
| VICAP  | VICAP    | 496MHz | 496MHz<br>396MHz<br>350MHz<br>300MHz<br>263MHz<br>98MHz<br>148.5MHz |
| MIPIRX | MIPIRX   | 496MHz | 496MHz<br>396MHz<br>350MHz<br>300MHz<br>263MHz<br>198MHz            |
| NPU    | NPU      | 496MHz | 496MHz<br>450MHz<br>396MHz<br>350MHz<br>263MHz<br>525MHz            |
| VPSS   | CLK_VPSS | 396MHz | 396MHz<br>350MHz<br>300MHz<br>263MHz<br>200MHz<br>150MHz<br>100MHz  |
| VGS    | CLK_VGS  | 396MHz | 396MHz<br>350MHz<br>300MHz<br>263MHz<br>198MHz<br>150MHz<br>100MHz  |
| GZIP   | CLK_GZIP | 297MHz | 297MHz                                                              |
| VPU    | CLK_VPU  | 525MHz | 525MHz<br>396MHz<br>297MHz<br>263MHz                                |



| 模块        | 时钟名称          | 默认频点     | 所有频点                                                               |
|-----------|---------------|----------|--------------------------------------------------------------------|
| JPGE      | CLK_JPGE      | 496MHz   | 496MHz<br>420MHz<br>350MHz<br>300MHz<br>263MHz                     |
| USERSLVAE | CLK_USRSLV    | 496MHz   | 496MHz<br>396MHz                                                   |
| MCU       | CLK_MCU_CORE  | 24MHz    | 24MHz<br>396MHz<br>297MHz<br>198MHz                                |
|           | CLK_MCU_DBG   | 24MHz    | 24MHz<br>297MHz<br>198MHz<br>100MHz                                |
| AIAO      | clk_aiao_pll  | 993MHz   | 993MHz<br>1188MHz<br>24MHz<br>22MHz                                |
|           | clk_aiao_core | 99MHz    | 50MHz<br>99MHz                                                     |
| VOUT      | clk_vout_ppc  | 198MHz   | 198MHz<br>150MHz<br>99MHz<br>50MHz                                 |
|           | clk_vout_hd0  | 148.5MHz | 148.5MHz<br>74.25MHz<br>37.125MHz<br>27MHz<br>13.5MHz<br>LCD 分频器时钟 |

| 模块    | 时钟名称       | 默认频点   | 所有频点                                                                        |
|-------|------------|--------|-----------------------------------------------------------------------------|
| EMMC  | clk_emmc   | 1.6MHz | 0.4MHz<br>1.6MHz<br>99MHz<br>198MHz<br>396MHz<br>496MHz<br>594MHz<br>700MHz |
| SDIO0 | clk_sdio0  | 1.6MHz | 0.4MHz<br>1.6MHz<br>99MHz<br>198MHz                                         |
| SDIO1 | clk_sdio10 | 1.6MHz | 0.4MHz<br>1.6MHz<br>99MHz<br>198MHz                                         |
| IVE   | clk_ive    | 496MHz | 496MHz<br>420MHz<br>396MHz<br>297MHz                                        |

#### 1.2.4 PLL 配置

芯片内部使用了 4 个 PLL，每个 PLL 使用 6 组配置寄存器，详情参考寄存器表单。

PLL 均采用管脚 XIN 输入的晶振时钟作为参考时钟，PLL 输出频率配置方法如表 1-3 所示。

表1-3 PLL 频率计算方法

| PLL Pin | 计算方法描述                                                                                                         | 注意事项                   |
|---------|----------------------------------------------------------------------------------------------------------------|------------------------|
| FREF    | PLL 输入参考时钟                                                                                                     | 芯片要求固定输入 24MHz         |
| FOUTVCO | $FREF \times ((fbdiv(\text{整数分频}) + \text{frac}(\text{小数分频})) / (\text{refdiv}(\text{参考时钟分频}) \times 2^{11}))$ | PLL 工作频率要求小于等于 2400MHz |

配置 FOUTVCO 输出 1200MHz，根据  $FOUTVCO = FREF \times ((fbdiv(\text{整数分频}) + \frac{frac(\text{小数分频})}{2^{11}})) / (refdiv(\text{参考时钟分频})) = 1200\text{MHz}$ ，其中  $FREF=24\text{MHz}$ ，若  $refdiv=1$ ，计算出  $fbdiv=50$ ， $frac=0$ 。如果涉及这个例子中 1200 是 24 的整数倍，所以小数分频的参数为 0；如果  $FOUTVCO=900\text{MHz}$ ， $refdiv=1$ ，此时  $fbdiv=37$ ，小数分频部分=0.5，需要计算  $frac/(2^{11})=0.5$ ，得到  $frac=1024$ ，如果其它条件下结果不是整数  $frac$  应该取整。

### 1.2.5 注意事项

配置 PLL 需要按照以下步骤：

- 步骤 1 切换时钟，选择非 PLL 时钟，或者其他 PLL 时钟。
- 步骤 2 更改 PLL 配置。
- 步骤 3 配置 PU 为 1，即 `pu_xxppll` 和 `pu_xxppll_sfreq`。
- 步骤 4 复位 PLL，拉低 `xxppll_fbdiv_rstb` 至少大于 1us。
- 步骤 5 等待 30us 后，打开输出门控即可输出时钟。

----结束

### 1.2.6 CRG 寄存器概览

寄存器概览（基地址：0x1201\_0000）

表1-4 时钟寄存器概览

| 偏移地址   | 名称        | 描述            |
|--------|-----------|---------------|
| 0x0000 | PERI_CRG0 | APLL 配置寄存器 0。 |
| 0x0004 | PERI_CRG1 | APLL 配置寄存器 1。 |
| 0x0008 | PERI_CRG2 | APLL 配置寄存器 2。 |
| 0x000C | PERI_CRG3 | APLL 配置寄存器 3。 |
| 0x0010 | PERI_CRG4 | APLL 配置寄存器 4。 |
| 0x0014 | PERI_CRG5 | APLL 配置寄存器 5。 |
| 0x0018 | PERI_CRG6 | VPLL 配置寄存器 0。 |
| 0x001C | PERI_CRG7 | VPLL 配置寄存器 1。 |

| 偏移地址   | 名称         | 描述                  |
|--------|------------|---------------------|
| 0x0020 | PERI_CRG8  | VPLL 配置寄存器 2。       |
| 0x0024 | PERI_CRG9  | VPLL 配置寄存器 3。       |
| 0x0028 | PERI_CRG10 | VPLL 配置寄存器 4。       |
| 0x002C | PERI_CRG11 | VPLL 配置寄存器 5。       |
| 0x0030 | PERI_CRG12 | MPLL 配置寄存器 0。       |
| 0x0034 | PERI_CRG13 | MPLL 配置寄存器 1。       |
| 0x0038 | PERI_CRG14 | MPLL 配置寄存器 2。       |
| 0x003C | PERI_CRG15 | MPLL 配置寄存器 3。       |
| 0x0040 | PERI_CRG16 | MPLL 配置寄存器 4。       |
| 0x0044 | PERI_CRG17 | MPLL 配置寄存器 5。       |
| 0x0048 | PERI_CRG18 | FPLL 配置寄存器 0。       |
| 0x004C | PERI_CRG19 | FPLL 配置寄存器 1。       |
| 0x0050 | PERI_CRG20 | FPLL 配置寄存器 2。       |
| 0x0054 | PERI_CRG21 | FPLL 配置寄存器 3。       |
| 0x0058 | PERI_CRG22 | FPLL 配置寄存器 4。       |
| 0x005C | PERI_CRG23 | FPLL 配置寄存器 5。       |
| 0x0060 | PERI_CRG24 | PLL 备用寄存器。          |
| 0x0078 | PERI_CRG30 | CPU 频率模式及复位配置寄存器。   |
| 0x007C | PERI_CRG31 | DDR 时钟及复位配置寄存器。     |
| 0x0080 | PERI_CRG32 | SOC 时钟选择寄存器。        |
| 0x0084 | PERI_CRG33 | SOC 总线安全机制时钟及复位寄存器。 |
| 0x00A0 | PERI_CRG40 | 媒体功能模块频率配置寄存器。      |
| 0x00A4 | PERI_CRG41 | VPU 时钟及软复位控制寄存器。    |
| 0x00A8 | PERI_CRG42 | VPSS 时钟及软复位控制寄存器。   |

| 偏移地址   | 名称         | 描述                                                |
|--------|------------|---------------------------------------------------|
| 0x00AC | PERI_CRG43 | VGS 时钟及软复位控制寄存器。                                  |
| 0x00B0 | PERI_CRG44 | JPGE 时钟及软复位控制寄存器。                                 |
| 0x00B4 | PERI_CRG45 | NPU 时钟及软复位控制寄存器。                                  |
| 0x00BC | PERI_CRG47 | TDE 时钟及软复位控制寄存器。                                  |
| 0x00C0 | PERI_CRG48 | GDC 时钟及软复位控制寄存器。                                  |
| 0x00C8 | PERI_CRG50 | USRDEF SLV 时钟及软复位控制寄存器。                           |
| 0x00F0 | PERI_CRG60 | SENSOR 时钟及软复位控制寄存器。                               |
| 0x00F4 | PERI_CRG61 | VI 时钟及软复位控制寄存器。                                   |
| 0x00F8 | PERI_CRG62 | MIPI RX 时钟及软复位控制寄存器。                              |
| 0x00FC | PERI_CRG63 | VIPROC 时钟及软复位控制寄存器。                               |
| 0x0104 | PERI_CRG65 | LCD 分配器时钟控制寄存器。                                   |
| 0x0108 | PERI_CRG66 | VOU 接口相关<br>(LCD&BT&LVDS&MIPITX) 时钟及软复位<br>控制寄存器。 |
| 0x010C | PERI_CRG67 | VOU 时钟及软复位控制寄存器。                                  |
| 0x0134 | PERI_CRG77 | IVE 相关的时钟及软复位控制寄存器。                               |
| 0x0138 | PERI_CRG78 | SCD 相关的时钟及软复位控制寄存器。                               |
| 0x013C | PERI_CRG79 | USB2 相关的时钟及软复位控制寄存器。                              |
| 0x0140 | PERI_CRG80 | USB2 相关的时钟及软复位控制寄存器。                              |
| 0x0144 | PERI_CRG81 | FMC 相关的时钟及软复位控制寄存器。                               |
| 0x016C | PERI_CRG91 | FEPHY 钟及软复位控制寄存器。                                 |
| 0x0170 | PERI_CRG92 | FEPHY 钟及软复位控制寄存器。                                 |
| 0x0174 | PERI_CRG93 | MCU 相关时钟及软复位控制寄存器。                                |
| 0x0178 | PERI_CRG94 | 总线安全机制时钟和复位寄存器。                                   |

| 偏移地址   | 名称              | 描述                                        |
|--------|-----------------|-------------------------------------------|
| 0x017C | PERI_CRG95      | MCU 系统总线桥时钟和复位寄存器。                        |
| 0x0190 | PERI_CRG100     | GZIP 时钟及软复位控制寄存器。                         |
| 0x0194 | PERI_CRG101     | DMAC 相关的时钟及软复位控制寄存器。                      |
| 0x0198 | PERI_CRG102     | DDRTEST 相关的时钟及软复位控制寄存器。                   |
| 0x019C | PERI_CRG103     | AIAO/AUDIO CODEC/DOLPHIN CODEC 时钟复位控制寄存器。 |
| 0x01A0 | PERI_CRG104     | HASH/CIPHER/RSA/TRNG/KLAD 相关的时钟及软复位控制寄存器。 |
| 0x01A4 | PERI_CRG105     | GMAC DLL 控制寄存器。                           |
| 0x01A8 | PERI_CRG106     | GMAC DLL 状态寄存器。                           |
| 0x01B8 | PERI_CRG110     | PWM/I2C/UART/SSP/IR 时钟及复位控制寄存器。           |
| 0x01BC | PERI_CRG111     | CAN/PMC/RTC/ADC/OTP 相关的时钟及软复位控制寄存器。       |
| 0x01C4 | PERI_CRG_PLL113 | PLL 测试控制寄存器。（不对用户开放）                      |
| 0x01C8 | PERI_CRG_PLL114 | DWWD 时钟和复位控制寄存器。                          |
| 0x01E4 | PERI_CRG_PLL121 | CBB 复位状态寄存器。                              |
| 0x01E8 | PERI_CRG_PLL122 | PLL LOCK 状态寄存器。                           |
| 0x01EC | PERI_CRG123     | DEBUG 保留寄存器。                              |
| 0x01F0 | PERI_CRG_PLL124 | CPU 高速 MEM 调速方式选择。                        |
| 0x01F4 | PERI_CRG125     | eMMC 时钟复位及 SAM DLL 控制寄存器。                 |
| 0x01F8 | PERI_CRG126     | eMMC SAMB DLL 控制寄存器。                      |
| 0x01FC | PERI_CRG127     | eMMC DRV DLL 控制寄存器。                       |
| 0x0200 | PERI_CRG128     | eMMC DS DLL 控制寄存器。                        |

| 偏移地址   | 名称          | 描述                         |
|--------|-------------|----------------------------|
| 0x0204 | PERI_CRG129 | eMMC DS180 DLL 控制寄存器。      |
| 0x0208 | PERI_CRG130 | eMMC SAM DLL 状态寄存器。        |
| 0x0210 | PERI_CRG132 | eMMC DRV DLL 状态寄存器。        |
| 0x0214 | PERI_CRG133 | eMMC DS DLL 状态寄存器。         |
| 0x0218 | PERI_CRG134 | eMMC DS180 DLL 状态寄存器。      |
| 0x021C | PERI_CRG135 | SDIO0 SAMB DLL 控制寄存器。      |
| 0x0220 | PERI_CRG136 | SDIO0 DRV DLL 控制寄存器。       |
| 0x0224 | PERI_CRG137 | SDIO0 SAM DLL 状态寄存器。       |
| 0x0228 | PERI_CRG138 | SDIO0 DRV DLL 状态寄存器。       |
| 0x022C | PERI_CRG139 | SDIO0 时钟复位及 SAM DLL 控制寄存器。 |
| 0x0230 | PERI_CRG140 | SDIO1 SAMB DLL 控制寄存器。      |
| 0x0234 | PERI_CRG141 | SDIO1 DRV DLL 控制寄存器。       |
| 0x0238 | PERI_CRG142 | SDIO1 SAM DLL 状态寄存器。       |
| 0x023C | PERI_CRG143 | SDIO1 DRV DLL 状态寄存器。       |
| 0x0240 | PERI_CRG144 | SDIO1 时钟复位及 SAM DLL 控制寄存器。 |
| 0x0248 | PERI_CRG146 | eMMC 保留寄存器。                |
| 0x024C | PERI_CRG147 | SDIO0 保留寄存器。               |
| 0x0250 | PERI_CRG148 | SDIO1 保留寄存器。               |
| 0x0490 | PERI_CRG292 | 输入分辨率时钟频率限制寄存器。            |
| 0x0494 | PERI_CRG293 | mipi rx 时钟使能限制寄存器。         |
| 0x0498 | PERI_CRG294 | 处理器性能限制寄存器。                |
| 0x049C | PERI_CRG295 | NPU 性能限制寄存器。               |

## 1.2.7 CRG 寄存器描述

### PERI\_CRG0

peri\_crg0 为 APLL 配置寄存器 0。

Offset Address:0x0    Total Reset Value:0x0050690F

| Bits    | Access | Name                   | Description       | Reset |
|---------|--------|------------------------|-------------------|-------|
| [31:24] | RW     | reserved               | 保留。               | 0x0   |
| [23:22] | RW     | mipipll_vg13_sel       | 寄存器信号名为 apll_cfg0 | 0x1   |
| [21:20] | RW     | mipipll_vg11_sel       | 寄存器信号名为 apll_cfg0 | 0x1   |
| [19:18] | RW     | mipipll_refclk_sel     | 寄存器信号名为 apll_cfg0 | 0x0   |
| [17:14] | RW     | mipipll_refdiv_ratio   | 寄存器信号名为 apll_cfg0 | 0x1   |
| [13]    | RW     | mipipll_even_div_en    | 寄存器信号名为 apll_cfg0 | 0x1   |
| [12:6]  | RW     | mipipll_even_div_ratio | 寄存器信号名为 apll_cfg0 | 0x24  |
| [5]     | RW     | pu_mipipll             | 寄存器信号名为 apll_cfg0 | 0x0   |
| [4]     | RW     | pu_mipipll_sfreg       | 寄存器信号名为 apll_cfg0 | 0x0   |
| [3]     | RW     | pu_mipipll_cp          | 寄存器信号名为 apll_cfg0 | 0x1   |
| [2]     | RW     | pu_mipipll_fbdv        | 寄存器信号名为 apll_cfg0 | 0x1   |
| [1]     | RW     | mipipll_fbdv_rstb      | 寄存器信号名为 apll_cfg0 | 0x1   |
| [0]     | RW     | mipipll_sdm_rstb       | 寄存器信号名为 apll_cfg0 | 0x1   |

### PERI\_CRG1

peri\_crg1 为 APLL 配置寄存器 1。

Offset Address:0x4    Total Reset Value:0x0026B1A3

| Bits    | Access | Name                    | Description       | Reset |
|---------|--------|-------------------------|-------------------|-------|
| [31:23] | RW     | reserved                | 保留。               | 0x00  |
| [22:21] | RW     | mipipll_lock_win_sel    | 寄存器信号名为 apll_cfg1 | 0x1   |
| [20]    | RW     | mipipll_lock_clk_inv_en | 寄存器信号名为 apll_cfg1 | 0x0   |



|         |    |                        |                   |     |
|---------|----|------------------------|-------------------|-----|
| [19:18] | RW | mipipll_lock_clk_sel   | 寄存器信号名为 apll_cfg1 | 0x1 |
| [17]    | RW | mipipll_lock_det_en    | 寄存器信号名为 apll_cfg1 | 0x1 |
| [16:15] | RW | mipipll_sel_fb_clk     | 寄存器信号名为 apll_cfg1 | 0x1 |
| [14:13] | RW | mipipll_sel_sample_clk | 寄存器信号名为 apll_cfg1 | 0x1 |
| [12]    | RW | mipipll_fast_lock_en   | 寄存器信号名为 apll_cfg1 | 0x1 |
| [11]    | RW | mipipll_lf_test_en     | 寄存器信号名为 apll_cfg1 | 0x0 |
| [10:8]  | RW | mipipll_rz             | 寄存器信号名为 apll_cfg1 | 0x1 |
| [7:6]   | RW | mipipll_cz             | 寄存器信号名为 apll_cfg1 | 0x2 |
| [5:4]   | RW | mipipll_c3             | 寄存器信号名为 apll_cfg1 | 0x2 |
| [3]     | RW | mipipll_r4_short       | 寄存器信号名为 apll_cfg1 | 0x0 |
| [2:1]   | RW | mipipll_r4             | 寄存器信号名为 apll_cfg1 | 0x1 |
| [0]     | RW | mipipll_c4_en          | 寄存器信号名为 apll_cfg1 | 0x1 |

## PERI\_CRG2

peri\_crg2 为 APLL 配置寄存器 2。

Offset Address:0x8 Total Reset Value:0x000F1E17

| Bits    | Access | Name                  | Description       | Reset |
|---------|--------|-----------------------|-------------------|-------|
| [31:20] | RW     | reserved              | 保留。               | 0x0   |
| [19]    | RW     | mipipll_pfd_en        | 寄存器信号名为 apll_cfg2 | 0x1   |
| [18]    | RW     | mipipll_cp_ota_en     | 寄存器信号名为 apll_cfg2 | 0x1   |
| [17]    | RW     | mipipll_cp_opamp_en   | 寄存器信号名为 apll_cfg2 | 0x1   |
| [16]    | RW     | mipipll_cp_startup_en | 寄存器信号名为 apll_cfg2 | 0x1   |
| [15]    | RW     | mipipll_int_frac_sw   | 寄存器信号名为 apll_cfg2 | 0x0   |
| [14:13] | RW     | mipipll_icp_1u        | 寄存器信号名为 apll_cfg2 | 0x0   |
| [12:11] | RW     | mipipll_icp_5u        | 寄存器信号名为 apll_cfg2 | 0x3   |
| [10]    | RW     | mipipll_sel_cp_bias   | 寄存器信号名为 apll_cfg2 | 0x1   |

|       |    |                            |                   |     |
|-------|----|----------------------------|-------------------|-----|
| [9]   | RW | mipipll_vco_postdiv_clk_en | 寄存器信号名为 apll_cfg2 | 0x1 |
| [8:6] | RW | mipipll_vco_postdiv_sel    | 寄存器信号名为 apll_cfg2 | 0x0 |
| [5]   | RW | mipipll_vco_vdd_ctrl_extra | 寄存器信号名为 apll_cfg2 | 0x0 |
| [4:3] | RW | mipipll_vco_vdd_ctrl       | 寄存器信号名为 apll_cfg2 | 0x2 |
| [2:0] | RW | mipipll_vco_speed          | 寄存器信号名为 apll_cfg2 | 0x7 |

### PERI\_CRG3

peri\_crg3 为 APLL 配置寄存器 3。

Offset Address:0xc Total Reset Value:0x0015E001

| Bits    | Access | Name                  | Description       | Reset   |
|---------|--------|-----------------------|-------------------|---------|
| [31:23] | RW     | reserved              | 保留。               | 0x00000 |
| [22]    | RW     | mipipll_sdm_bypass    | 寄存器信号名为 apll_cfg3 | 0x0     |
| [21:3]  | RW     | mipipll_sdm_in        | 寄存器信号名为 apll_cfg3 | 0x2BC00 |
| [2:1]   | RW     | mipipll_sdm_dith_sel  | 寄存器信号名为 apll_cfg3 | 0x0     |
| [0]     | RW     | mipipll_sdm_order_sel | 寄存器信号名为 apll_cfg3 | 0x1     |

### PERI\_CRG4

peri\_crg4 为 APLL 配置寄存器 4。

Offset Address:0x10 Total Reset Value:0x00000100

| Bits   | Access | Name                    | Description       | Reset |
|--------|--------|-------------------------|-------------------|-------|
| [31:9] | RW     | reserved                | 保留。               | 0x0   |
| [8]    | RW     | mipipll_dtest_pull_down | 寄存器信号名为 apll_cfg4 | 0x1   |
| [7]    | RW     | mipipll_dten_pll_locked | 寄存器信号名为 apll_cfg4 | 0x0   |
| [6]    | RW     | mipipll_dten_pupll      | 寄存器信号名为 apll_cfg4 | 0x0   |
| [5]    | RW     | mipipll_dten_fsdm       | 寄存器信号名为 apll_cfg4 | 0x0   |
| [4]    | RW     | mipipll_dten_fref       | 寄存器信号名为 apll_cfg4 | 0x0   |
| [3]    | RW     | mipipll_dten_ckin       | 寄存器信号名为 apll_cfg4 | 0x0   |

|     |    |                      |                   |     |
|-----|----|----------------------|-------------------|-----|
| [2] | RW | mipipll_ten_sfreg    | 寄存器信号名为 apll_cfg4 | 0x0 |
| [1] | RW | mipipll_ten          | 寄存器信号名为 apll_cfg4 | 0x0 |
| [0] | RW | mipipll_dc_tp_out_en | 寄存器信号名为 apll_cfg4 | 0x0 |

## PERI\_CRG5

peri\_crg5 为 APLL 配置寄存器 5。

Offset Address:0x14 Total Reset Value:0x00001AC8

| Bits    | Access | Name                      | Description       | Reset |
|---------|--------|---------------------------|-------------------|-------|
| [31:16] | RW     | reserved                  | 保留。               | 0x00  |
| [15:13] | RW     | mipipll_tmux              | 寄存器信号名为 apll_cfg5 | 0x0   |
| [12]    | RW     | mipipll_ssc_start_gate_en | 寄存器信号名为 apll_cfg5 | 0x1   |
| [11:9]  | RW     | mipipll_ssc_gain          | 寄存器信号名为 apll_cfg5 | 0x5   |
| [8:1]   | RW     | mipipll_ssc_cnt           | 寄存器信号名为 apll_cfg5 | 0x64  |
| [0]     | RW     | mipipll_ssc_en            | 寄存器信号名为 apll_cfg5 | 0x0   |

## PERI\_CRG6

peri\_crg6 为 VPLL 配置寄存器 0。

Offset Address:0x18 Total Reset Value:0x0050690F

| Bits    | Access | Name                   | Description       | Reset |
|---------|--------|------------------------|-------------------|-------|
| [31:24] | RW     | reserved               | 保留。               | 0x0   |
| [23:22] | RW     | mipipll_vg13_sel       | 寄存器信号名为 vpll_cfg0 | 0x1   |
| [21:20] | RW     | mipipll_vg11_sel       | 寄存器信号名为 vpll_cfg0 | 0x1   |
| [19:18] | RW     | mipipll_refclk_sel     | 寄存器信号名为 vpll_cfg0 | 0x0   |
| [17:14] | RW     | mipipll_refdiv_ratio   | 寄存器信号名为 vpll_cfg0 | 0x1   |
| [13]    | RW     | mipipll_even_div_en    | 寄存器信号名为 vpll_cfg0 | 0x1   |
| [12:6]  | RW     | mipipll_even_div_ratio | 寄存器信号名为 vpll_cfg0 | 0x24  |
| [5]     | RW     | pu_mipipll             | 寄存器信号名为 vpll_cfg0 | 0x0   |

|     |    |                   |                   |     |
|-----|----|-------------------|-------------------|-----|
| [4] | RW | pu_mipipll_sfreg  | 寄存器信号名为 vpll_cfg0 | 0x0 |
| [3] | RW | pu_mipipll_cp     | 寄存器信号名为 vpll_cfg0 | 0x1 |
| [2] | RW | pu_mipipll_fbdv   | 寄存器信号名为 vpll_cfg0 | 0x1 |
| [1] | RW | mipipll_fbdv_rstb | 寄存器信号名为 vpll_cfg0 | 0x1 |
| [0] | RW | mipipll_sdm_rstb  | 寄存器信号名为 vpll_cfg0 | 0x1 |

## PERI\_CRG7

peri\_crg7 为 VPLL 配置寄存器 1。

Offset Address:0x1c Total Reset Value:0x0026B1A3

| Bits    | Access | Name                    | Description       | Reset |
|---------|--------|-------------------------|-------------------|-------|
| [31:23] | RW     | reserved                | 保留。               | 0x00  |
| [22:21] | RW     | mipipll_lock_win_sel    | 寄存器信号名为 vpll_cfg1 | 0x1   |
| [20]    | RW     | mipipll_lock_clk_inv_en | 寄存器信号名为 vpll_cfg1 | 0x0   |
| [19:18] | RW     | mipipll_lock_clk_sel    | 寄存器信号名为 vpll_cfg1 | 0x1   |
| [17]    | RW     | mipipll_lock_det_en     | 寄存器信号名为 vpll_cfg1 | 0x1   |
| [16:15] | RW     | mipipll_sel_fb_clk      | 寄存器信号名为 vpll_cfg1 | 0x1   |
| [14:13] | RW     | mipipll_sel_sample_clk  | 寄存器信号名为 vpll_cfg1 | 0x1   |
| [12]    | RW     | mipipll_fast_lock_en    | 寄存器信号名为 vpll_cfg1 | 0x1   |
| [11]    | RW     | mipipll_lf_test_en      | 寄存器信号名为 vpll_cfg1 | 0x0   |
| [10:8]  | RW     | mipipll_rz              | 寄存器信号名为 vpll_cfg1 | 0x1   |
| [7:6]   | RW     | mipipll_cz              | 寄存器信号名为 vpll_cfg1 | 0x2   |
| [5:4]   | RW     | mipipll_c3              | 寄存器信号名为 vpll_cfg1 | 0x2   |
| [3]     | RW     | mipipll_r4_short        | 寄存器信号名为 vpll_cfg1 | 0x0   |
| [2:1]   | RW     | mipipll_r4              | 寄存器信号名为 vpll_cfg1 | 0x1   |
| [0]     | RW     | mipipll_c4_en           | 寄存器信号名为 vpll_cfg1 | 0x1   |

## PERI\_CRG8

peri\_crg8 为 VPLL 配置寄存器 2。

Offset Address:0x20 Total Reset Value:0x000F1E13

| Bits    | Access | Name                       | Description       | Reset |
|---------|--------|----------------------------|-------------------|-------|
| [31:20] | RW     | reserved                   | 保留。               | 0x0   |
| [19]    | RW     | mipipll_pfd_en             | 寄存器信号名为 vpll_cfg2 | 0x1   |
| [18]    | RW     | mipipll_cp_ota_en          | 寄存器信号名为 vpll_cfg2 | 0x1   |
| [17]    | RW     | mipipll_cp_opamp_en        | 寄存器信号名为 vpll_cfg2 | 0x1   |
| [16]    | RW     | mipipll_cp_startup_en      | 寄存器信号名为 vpll_cfg2 | 0x1   |
| [15]    | RW     | mipipll_int_frac_sw        | 寄存器信号名为 vpll_cfg2 | 0x0   |
| [14:13] | RW     | mipipll_icp_1u             | 寄存器信号名为 vpll_cfg2 | 0x0   |
| [12:11] | RW     | mipipll_icp_5u             | 寄存器信号名为 vpll_cfg2 | 0x3   |
| [10]    | RW     | mipipll_sel_cp_bias        | 寄存器信号名为 vpll_cfg2 | 0x1   |
| [9]     | RW     | mipipll_vco_postdiv_clk_en | 寄存器信号名为 vpll_cfg2 | 0x1   |
| [8:6]   | RW     | mipipll_vco_postdiv_sel    | 寄存器信号名为 vpll_cfg2 | 0x0   |
| [5]     | RW     | mipipll_vco_vdd_ctrl_extra | 寄存器信号名为 vpll_cfg2 | 0x0   |
| [4:3]   | RW     | mipipll_vco_vdd_ctrl       | 寄存器信号名为 vpll_cfg2 | 0x2   |
| [2:0]   | RW     | mipipll_vco_speed          | 寄存器信号名为 vpll_cfg2 | 0x3   |

## PERI\_CRG9

peri\_crg9 为 VPLL 配置寄存器 3。

Offset Address:0x24 Total Reset Value:0x000C6001

| Bits    | Access | Name               | Description       | Reset   |
|---------|--------|--------------------|-------------------|---------|
| [31:23] | RW     | reserved           | 保留。               | 0x00000 |
| [22]    | RW     | mipipll_sdm_bypass | 寄存器信号名为 vpll_cfg3 | 0x0     |
| [21:3]  | RW     | mipipll_sdmin      | 寄存器信号名为 vpll_cfg3 | 0x18C00 |

|       |    |                       |                   |     |
|-------|----|-----------------------|-------------------|-----|
| [2:1] | RW | mipipll_sdm_dith_sel  | 寄存器信号名为 vp1l_cfg3 | 0x0 |
| [0]   | RW | mipipll_sdm_order_sel | 寄存器信号名为 vp1l_cfg3 | 0x1 |

## PERI\_CRG10

peri\_crg10 为 VPLL 配置寄存器 4。

Offset Address:0x28 Total Reset Value:0x00000100

| Bits   | Access | Name                    | Description       | Reset |
|--------|--------|-------------------------|-------------------|-------|
| [31:9] | RW     | reserved                | 保留。               | 0x0   |
| [8]    | RW     | mipipll_dtest_pull_down | 寄存器信号名为 vp1l_cfg4 | 0x1   |
| [7]    | RW     | mipipll_dten_pll_locked | 寄存器信号名为 vp1l_cfg4 | 0x0   |
| [6]    | RW     | mipipll_dten_pupll      | 寄存器信号名为 vp1l_cfg4 | 0x0   |
| [5]    | RW     | mipipll_dten_fsdm       | 寄存器信号名为 vp1l_cfg4 | 0x0   |
| [4]    | RW     | mipipll_dten_fref       | 寄存器信号名为 vp1l_cfg4 | 0x0   |
| [3]    | RW     | mipipll_dten_ckin       | 寄存器信号名为 vp1l_cfg4 | 0x0   |
| [2]    | RW     | mipipll_ten_sfreg       | 寄存器信号名为 vp1l_cfg4 | 0x0   |
| [1]    | RW     | mipipll_ten             | 寄存器信号名为 vp1l_cfg4 | 0x0   |
| [0]    | RW     | mipipll_dc_tp_out_en    | 寄存器信号名为 vp1l_cfg4 | 0x0   |

## PERI\_CRG11

peri\_crg11 为 VPLL 配置寄存器 5。

Offset Address:0x2c Total Reset Value:0x00001AC8

| Bits    | Access | Name                      | Description       | Reset |
|---------|--------|---------------------------|-------------------|-------|
| [31:13] | RW     | reserved                  | 保留。               | 0x00  |
| [15:13] | RW     | mipipll_tmux              | 寄存器信号名为 vp1l_cfg5 | 0x0   |
| [12]    | RW     | mipipll_ssc_start_gate_en | 寄存器信号名为 vp1l_cfg5 | 0x1   |
| [11:9]  | RW     | mipipll_ssc_gain          | 寄存器信号名为 vp1l_cfg5 | 0x5   |
| [8:1]   | RW     | mipipll_ssc_cnt           | 寄存器信号名为 vp1l_cfg5 | 0x64  |

|     |    |                |                    |     |
|-----|----|----------------|--------------------|-----|
| [0] | RW | mipipll_ssc_en | 寄存器信号名为 vppll_cfg5 | 0x0 |
|-----|----|----------------|--------------------|-----|

## PERI\_CRG12

peri\_crg12 为 MPLL 配置寄存器 0。

Offset Address:0x30 Total Reset Value:0x0051690F

| Bits    | Access | Name                   | Description        | Reset |
|---------|--------|------------------------|--------------------|-------|
| [31:24] | RW     | reserved               | 保留。                | 0x0   |
| [23:22] | RW     | mipipll_vg13_sel       | 寄存器信号名为 mppll_cfg0 | 0x1   |
| [21:20] | RW     | mipipll_vg11_sel       | 寄存器信号名为 mppll_cfg0 | 0x1   |
| [19:18] | RW     | mipipll_refclk_sel     | 寄存器信号名为 mppll_cfg0 | 0x0   |
| [17:14] | RW     | mipipll_refdiv_ratio   | 寄存器信号名为 mppll_cfg0 | 0x5   |
| [13]    | RW     | mipipll_even_div_en    | 寄存器信号名为 mppll_cfg0 | 0x1   |
| [12:6]  | RW     | mipipll_even_div_ratio | 寄存器信号名为 mppll_cfg0 | 0x24  |
| [5]     | RW     | pu_mipipll             | 寄存器信号名为 mppll_cfg0 | 0x0   |
| [4]     | RW     | pu_mipipll_sfreg       | 寄存器信号名为 mppll_cfg0 | 0x0   |
| [3]     | RW     | pu_mipipll_cp          | 寄存器信号名为 mppll_cfg0 | 0x1   |
| [2]     | RW     | pu_mipipll_fbdv        | 寄存器信号名为 mppll_cfg0 | 0x1   |
| [1]     | RW     | mipipll_fbdv_rstb      | 寄存器信号名为 mppll_cfg0 | 0x1   |
| [0]     | RW     | mipipll_sdm_rstb       | 寄存器信号名为 mppll_cfg0 | 0x1   |

## PERI\_CRG13

peri\_crg13 为 MPLL 配置寄存器 1。

Offset Address:0x34 Total Reset Value:0x0026B1A3

| Bits    | Access | Name                    | Description        | Reset |
|---------|--------|-------------------------|--------------------|-------|
| [31:23] | RW     | reserved                | 保留。                | 0x00  |
| [22:21] | RW     | mipipll_lock_win_sel    | 寄存器信号名为 mppll_cfg1 | 0x1   |
| [20]    | RW     | mipipll_lock_clk_inv_en | 寄存器信号名为 mppll_cfg1 | 0x0   |

|         |    |                        |                    |     |
|---------|----|------------------------|--------------------|-----|
| [19:18] | RW | mipipll_lock_clk_sel   | 寄存器信号名为 mppll_cfg1 | 0x1 |
| [17]    | RW | mipipll_lock_det_en    | 寄存器信号名为 mppll_cfg1 | 0x1 |
| [16:15] | RW | mipipll_sel_fb_clk     | 寄存器信号名为 mppll_cfg1 | 0x1 |
| [14:13] | RW | mipipll_sel_sample_clk | 寄存器信号名为 mppll_cfg1 | 0x1 |
| [12]    | RW | mipipll_fast_lock_en   | 寄存器信号名为 mppll_cfg1 | 0x1 |
| [11]    | RW | mipipll_lf_test_en     | 寄存器信号名为 mppll_cfg1 | 0x0 |
| [10:8]  | RW | mipipll_rz             | 寄存器信号名为 mppll_cfg1 | 0x1 |
| [7:6]   | RW | mipipll_cz             | 寄存器信号名为 mppll_cfg1 | 0x2 |
| [5:4]   | RW | mipipll_c3             | 寄存器信号名为 mppll_cfg1 | 0x2 |
| [3]     | RW | mipipll_r4_short       | 寄存器信号名为 mppll_cfg1 | 0x0 |
| [2:1]   | RW | mipipll_r4             | 寄存器信号名为 mppll_cfg1 | 0x1 |
| [0]     | RW | mipipll_c4_en          | 寄存器信号名为 mppll_cfg1 | 0x1 |

## PERI\_CRG14

peri\_crg14 为 MPLL 配置寄存器 2。

Offset Address:0x38 Total Reset Value:0x000F1E12

| Bits    | Access | Name                  | Description        | Reset |
|---------|--------|-----------------------|--------------------|-------|
| [31:20] | RW     | reserved              | 保留。                | 0x0   |
| [19]    | RW     | mipipll_pfd_en        | 寄存器信号名为 mppll_cfg2 | 0x1   |
| [18]    | RW     | mipipll_cp_ota_en     | 寄存器信号名为 mppll_cfg2 | 0x1   |
| [17]    | RW     | mipipll_cp_opamp_en   | 寄存器信号名为 mppll_cfg2 | 0x1   |
| [16]    | RW     | mipipll_cp_startup_en | 寄存器信号名为 mppll_cfg2 | 0x1   |
| [15]    | RW     | mipipll_int_frac_sw   | 寄存器信号名为 mppll_cfg2 | 0x0   |
| [14:13] | RW     | mipipll_icp_1u        | 寄存器信号名为 mppll_cfg2 | 0x0   |
| [12:11] | RW     | mipipll_icp_5u        | 寄存器信号名为 mppll_cfg2 | 0x3   |
| [10]    | RW     | mipipll_sel_cp_bias   | 寄存器信号名为 mppll_cfg2 | 0x1   |



|       |    |                            |                    |     |
|-------|----|----------------------------|--------------------|-----|
| [9]   | RW | mipipll_vco_postdiv_clk_en | 寄存器信号名为 mppll_cfg2 | 0x1 |
| [8:6] | RW | mipipll_vco_postdiv_sel    | 寄存器信号名为 mppll_cfg2 | 0x0 |
| [5]   | RW | mipipll_vco_vdd_ctrl_extra | 寄存器信号名为 mppll_cfg2 | 0x0 |
| [4:3] | RW | mipipll_vco_vdd_ctrl       | 寄存器信号名为 mppll_cfg2 | 0x2 |
| [2:0] | RW | mipipll_vco_speed          | 寄存器信号名为 mppll_cfg2 | 0x2 |

## PERI\_CRG15

peri\_crg15 为 MPLL 配置寄存器 3。

Offset Address:0x3c Total Reset Value:0x00333331

| Bits    | Access | Name                  | Description        | Reset   |
|---------|--------|-----------------------|--------------------|---------|
| [31:23] | RW     | reserved              | 保留。                | 0x00000 |
| [22]    | RW     | mipipll_sdm_bypass    | 寄存器信号名为 mppll_cfg3 | 0x0     |
| [21:3]  | RW     | mipipll_sdm_in        | 寄存器信号名为 mppll_cfg3 | 0x66666 |
| [2:1]   | RW     | mipipll_sdm_dith_sel  | 寄存器信号名为 mppll_cfg3 | 0x0     |
| [0]     | RW     | mipipll_sdm_order_sel | 寄存器信号名为 mppll_cfg3 | 0x1     |

## PERI\_CRG16

peri\_crg16 为 MPLL 配置寄存器 4。

Offset Address:0x40 Total Reset Value:0x00000100

| Bits   | Access | Name                    | Description        | Reset |
|--------|--------|-------------------------|--------------------|-------|
| [31:9] | RW     | reserved                | 保留。                | 0x0   |
| [8]    | RW     | mipipll_dtest_pull_down | 寄存器信号名为 mppll_cfg4 | 0x1   |
| [7]    | RW     | mipipll_dten_pll_locked | 寄存器信号名为 mppll_cfg4 | 0x0   |
| [6]    | RW     | mipipll_dten_pupll      | 寄存器信号名为 mppll_cfg4 | 0x0   |
| [5]    | RW     | mipipll_dten_fsdm       | 寄存器信号名为 mppll_cfg4 | 0x0   |
| [4]    | RW     | mipipll_dten_fref       | 寄存器信号名为 mppll_cfg4 | 0x0   |
| [3]    | RW     | mipipll_dten_ckin       | 寄存器信号名为 mppll_cfg4 | 0x0   |

|     |    |                      |                   |     |
|-----|----|----------------------|-------------------|-----|
| [2] | RW | mipipll_ten_sfreg    | 寄存器信号名为 mpll_cfg4 | 0x0 |
| [1] | RW | mipipll_ten          | 寄存器信号名为 mpll_cfg4 | 0x0 |
| [0] | RW | mipipll_dc_tp_out_en | 寄存器信号名为 mpll_cfg4 | 0x0 |

## PERI\_CRG17

peri\_crg17 为 MPLL 配置寄存器 5。

Offset Address:0x44 Total Reset Value:0x00001AC8

| Bits    | Access | Name                      | Description       | Reset |
|---------|--------|---------------------------|-------------------|-------|
| [31:13] | RW     | reserved                  | 保留。               | 0x00  |
| [15:13] | RW     | mipipll_tmux              | 寄存器信号名为 fpll_cfg5 | 0x0   |
| [12]    | RW     | mipipll_ssc_start_gate_en | 寄存器信号名为 fpll_cfg5 | 0x1   |
| [11:9]  | RW     | mipipll_ssc_gain          | 寄存器信号名为 fpll_cfg5 | 0x5   |
| [8:1]   | RW     | mipipll_ssc_cnt           | 寄存器信号名为 fpll_cfg5 | 0x64  |
| [0]     | RW     | mipipll_ssc_en            | 寄存器信号名为 fpll_cfg5 | 0x0   |

## PERI\_CRG18

peri\_crg18 为 FPLL 配置寄存器 0。

Offset Address:0x48 Total Reset Value:0x0050648F

| Bits    | Access | Name                   | Description       | Reset |
|---------|--------|------------------------|-------------------|-------|
| [31:24] | RW     | reserved               | 保留。               | 0x0   |
| [23:22] | RW     | mipipll_vg13_sel       | 寄存器信号名为 fpll_cfg0 | 0x1   |
| [21:20] | RW     | mipipll_vg11_sel       | 寄存器信号名为 fpll_cfg0 | 0x1   |
| [19:18] | RW     | mipipll_refclk_sel     | 寄存器信号名为 fpll_cfg0 | 0x0   |
| [17:14] | RW     | mipipll_refdiv_ratio   | 寄存器信号名为 fpll_cfg0 | 0x1   |
| [13]    | RW     | mipipll_even_div_en    | 寄存器信号名为 fpll_cfg0 | 0x1   |
| [12:6]  | RW     | mipipll_even_div_ratio | 寄存器信号名为 fpll_cfg0 | 0x12  |
| [5]     | RW     | pu_mipipll             | 寄存器信号名为 fpll_cfg0 | 0x0   |

|     |    |                   |                   |     |
|-----|----|-------------------|-------------------|-----|
| [4] | RW | pu_mipipll_sfreg  | 寄存器信号名为 fpll_cfg0 | 0x0 |
| [3] | RW | pu_mipipll_cp     | 寄存器信号名为 fpll_cfg0 | 0x1 |
| [2] | RW | pu_mipipll_fbdv   | 寄存器信号名为 fpll_cfg0 | 0x1 |
| [1] | RW | mipipll_fbdv_rstb | 寄存器信号名为 fpll_cfg0 | 0x1 |
| [0] | RW | mipipll_sdm_rstb  | 寄存器信号名为 fpll_cfg0 | 0x1 |

## PERI\_CRG19

peri\_crg19 为 FPLL 配置寄存器 1。

Offset Address:0x4c Total Reset Value:0x0026B1A3

| Bits    | Access | Name                    | Description       | Reset |
|---------|--------|-------------------------|-------------------|-------|
| [31:23] | RW     | reserved                | 保留。               | 0x00  |
| [22:21] | RW     | mipipll_lock_win_sel    | 寄存器信号名为 fpll_cfg1 | 0x1   |
| [20]    | RW     | mipipll_lock_clk_inv_en | 寄存器信号名为 fpll_cfg1 | 0x0   |
| [19:18] | RW     | mipipll_lock_clk_sel    | 寄存器信号名为 fpll_cfg1 | 0x1   |
| [17]    | RW     | mipipll_lock_det_en     | 寄存器信号名为 fpll_cfg1 | 0x1   |
| [16:15] | RW     | mipipll_sel_fb_clk      | 寄存器信号名为 fpll_cfg1 | 0x1   |
| [14:13] | RW     | mipipll_sel_sample_clk  | 寄存器信号名为 fpll_cfg1 | 0x1   |
| [12]    | RW     | mipipll_fast_lock_en    | 寄存器信号名为 fpll_cfg1 | 0x1   |
| [11]    | RW     | mipipll_lf_test_en      | 寄存器信号名为 fpll_cfg1 | 0x0   |
| [10:8]  | RW     | mipipll_rz              | 寄存器信号名为 fpll_cfg1 | 0x1   |
| [7:6]   | RW     | mipipll_cz              | 寄存器信号名为 fpll_cfg1 | 0x2   |
| [5:4]   | RW     | mipipll_c3              | 寄存器信号名为 fpll_cfg1 | 0x2   |
| [3]     | RW     | mipipll_r4_short        | 寄存器信号名为 fpll_cfg1 | 0x0   |
| [2:1]   | RW     | mipipll_r4              | 寄存器信号名为 fpll_cfg1 | 0x1   |
| [0]     | RW     | mipipll_c4_en           | 寄存器信号名为 fpll_cfg1 | 0x1   |

## PERI\_CRG20

peri\_crg20 为 FPLL 配置寄存器 2。

Offset Address:0x50 Total Reset Value:0x000F1E12

| Bits    | Access | Name                       | Description       | Reset |
|---------|--------|----------------------------|-------------------|-------|
| [31:20] | RW     | reserved                   | 保留。               | 0x0   |
| [19]    | RW     | mipipll_pfd_en             | 寄存器信号名为 fpll_cfg2 | 0x1   |
| [18]    | RW     | mipipll_cp_ota_en          | 寄存器信号名为 fpll_cfg2 | 0x1   |
| [17]    | RW     | mipipll_cp_opamp_en        | 寄存器信号名为 fpll_cfg2 | 0x1   |
| [16]    | RW     | mipipll_cp_startup_en      | 寄存器信号名为 fpll_cfg2 | 0x1   |
| [15]    | RW     | mipipll_int_frac_sw        | 寄存器信号名为 fpll_cfg2 | 0x0   |
| [14:13] | RW     | mipipll_icp_1u             | 寄存器信号名为 fpll_cfg2 | 0x0   |
| [12:11] | RW     | mipipll_icp_5u             | 寄存器信号名为 fpll_cfg2 | 0x3   |
| [10]    | RW     | mipipll_sel_cp_bias        | 寄存器信号名为 fpll_cfg2 | 0x1   |
| [9]     | RW     | mipipll_vco_postdiv_clk_en | 寄存器信号名为 fpll_cfg2 | 0x1   |
| [8:6]   | RW     | mipipll_vco_postdiv_sel    | 寄存器信号名为 fpll_cfg2 | 0x0   |
| [5]     | RW     | mipipll_vco_vdd_ctrl_extra | 寄存器信号名为 fpll_cfg2 | 0x0   |
| [4:3]   | RW     | mipipll_vco_vdd_ctrl       | 寄存器信号名为 fpll_cfg2 | 0x2   |
| [2:0]   | RW     | mipipll_vco_speed          | 寄存器信号名为 fpll_cfg2 | 0x2   |

## PERI\_CRG21

peri\_crg21 为 FPLL 配置寄存器 3。

Offset Address:0x54 Total Reset Value:0x00096001

| Bits    | Access | Name               | Description       | Reset   |
|---------|--------|--------------------|-------------------|---------|
| [31:23] | RW     | reserved           | 保留。               | 0x00000 |
| [22]    | RW     | mipipll_sdm_bypass | 寄存器信号名为 fpll_cfg3 | 0x0     |
| [21:3]  | RW     | mipipll_sadmin     | 寄存器信号名为 fpll_cfg3 | 0x12C00 |

|       |    |                       |                   |     |
|-------|----|-----------------------|-------------------|-----|
| [2:1] | RW | mipipll_sdm_dith_sel  | 寄存器信号名为 fpll_cfg3 | 0x0 |
| [0]   | RW | mipipll_sdm_order_sel | 寄存器信号名为 fpll_cfg3 | 0x1 |

## PERI\_CRG22

peri\_crg22 为 FPLL 配置寄存器 4。

Offset Address:0x58 Total Reset Value:0x00000100

| Bits   | Access | Name                    | Description       | Reset |
|--------|--------|-------------------------|-------------------|-------|
| [31:9] | RW     | reserved                | 保留。               | 0x0   |
| [8]    | RW     | mipipll_dtest_pull_down | 寄存器信号名为 fpll_cfg4 | 0x1   |
| [7]    | RW     | mipipll_dten_pll_locked | 寄存器信号名为 fpll_cfg4 | 0x0   |
| [6]    | RW     | mipipll_dten_pupll      | 寄存器信号名为 fpll_cfg4 | 0x0   |
| [5]    | RW     | mipipll_dten_fsdm       | 寄存器信号名为 fpll_cfg4 | 0x0   |
| [4]    | RW     | mipipll_dten_fref       | 寄存器信号名为 fpll_cfg4 | 0x0   |
| [3]    | RW     | mipipll_dten_ckin       | 寄存器信号名为 fpll_cfg4 | 0x0   |
| [2]    | RW     | mipipll_ten_sfreg       | 寄存器信号名为 fpll_cfg4 | 0x0   |
| [1]    | RW     | mipipll_ten             | 寄存器信号名为 fpll_cfg4 | 0x0   |
| [0]    | RW     | mipipll_dc_tp_out_en    | 寄存器信号名为 fpll_cfg4 | 0x0   |

## PERI\_CRG23

peri\_crg23 为 FPLL 配置寄存器 5。

Offset Address:0x5c Total Reset Value:0x00001AC8

| Bits    | Access | Name                      | Description       | Reset |
|---------|--------|---------------------------|-------------------|-------|
| [31:13] | RW     | reserved                  | 保留。               | 0x00  |
| [15:13] | RW     | mipipll_tmux              | 寄存器信号名为 fpll_cfg5 | 0x0   |
| [12]    | RW     | mipipll_ssc_start_gate_en | 寄存器信号名为 fpll_cfg5 | 0x1   |
| [11:9]  | RW     | mipipll_ssc_gain          | 寄存器信号名为 fpll_cfg5 | 0x5   |
| [8:1]   | RW     | mipipll_ssc_cnt           | 寄存器信号名为 fpll_cfg5 | 0x64  |

|     |    |                |                   |     |
|-----|----|----------------|-------------------|-----|
| [0] | RW | mipipll_ssc_en | 寄存器信号名为 fpll_cfg5 | 0x0 |
|-----|----|----------------|-------------------|-----|

## PERI\_CRG24

peri\_crg24 为 PLL 备用寄存器。

Offset Address:0x60 Total Reset Value:0x0000000

| Bits   | Access | Name           | Description                                                                       | Reset |
|--------|--------|----------------|-----------------------------------------------------------------------------------|-------|
| [31:6] | RW     | reserved       | 保留。                                                                               | 0x00  |
| [5]    | RW     | apll_vco_cken  | 0: vco 输出关闭<br>1: vco 输出打开                                                        | 0x0   |
| [4]    | RW     | vppl_vco_cken  | 0: vco 输出关闭<br>2: vco 输出打开                                                        | 0x0   |
| [3]    | RW     | mppl_vco_cken  | 0: vco 输出关闭<br>3: vco 输出打开                                                        | 0x0   |
| [2]    | RW     | fppl_vco_cken  | 0: vco 输出关闭<br>4: vco 输出打开                                                        | 0x0   |
| [1]    | RW     | clk_496m_chose | 0: 默认使用 mppl_vco 的 2 分频<br>1: 使用 fppl_vco 的 2 分频                                  | 0x0   |
| [0]    | RW     | rmii_ref_cksel | 0: apll_vco 2100m 时使用<br>1: appl_vco 2000m 时使用<br>注意: 此选项用于保证 mac 参考时钟<br>始终位 50m | 0x0   |

## PERI\_CRG30

peri\_crg30 为 CPU 频率模式及复位配置寄存器。

Offset Address:0x78 Total Reset Value:0x00100381

| Bits    | Access | Name        | Description                    | Reset    |
|---------|--------|-------------|--------------------------------|----------|
| [31:23] | RW     | reserved    | 保留。                            | 0x000000 |
| [22]    | RW     | l2_srst_req | l2 的复位请求<br>0: 撤消复位;<br>1: 复位。 | 0x0      |
| [21]    | RW     | cs_srst_req | CPU 内 apb 的复位请求                | 0x0      |

|         |    |                   |                                                                                                 |     |
|---------|----|-------------------|-------------------------------------------------------------------------------------------------|-----|
|         |    |                   | 0: 撤消复位;<br>1: 复位。                                                                              |     |
| [20:19] | RW | arm_core_srst_req | ARM core 的复位请求<br>[1] 0: core1 撤消复位;<br>1: core1 复位。<br>[0] 0: core0 撤消复位;<br>1: core0 复位。      | 0x2 |
| [18:17] | RW | arm_dbg_srst_req  | ARM debug 的复位请求<br>[1] 0: debug1 撤消复位;<br>1: debug1 复位。<br>[0] 0: debug0 撤消复位;<br>1: debug0 复位。 | 0x0 |
| [15]    | RW | soc_dbg_srst_req  | CPU 的 debug 软复位请求<br>0: 撤消复位;<br>1: 复位。                                                         | 0x0 |
| [7:6]   | RW | a7_por_srst       | A7 POR 的软复位请求<br>[x]: COREX<br>0: 撤消复位;<br>1: 复位。                                               | 0x2 |
| [5:1]   | RW | reserved          | 保留。                                                                                             | 0x0 |
| [0]     | RW | a7_pclkdbg_cken   | CPU 的 pclk debug 时钟门控<br>0: 时钟关闭;<br>1: 时钟打开。                                                   | 0x1 |

## PERI\_CRG31

peri\_crg31 为 DDR 时钟及复位配置寄存器。

Offset Address:0x7c Total Reset Value:0x00000010

| Bits    | Access | Name            | Description                         | Reset      |
|---------|--------|-----------------|-------------------------------------|------------|
| [31:12] | RW     | reserved        | 保留。                                 | 0x00000000 |
| [11]    | RW     | ddrphy_srst_req | DDR PHY 软复位请求<br>0: 撤消复位;<br>1: 复位。 | 0x0        |

|         |    |                  |                                                                                                                                                                                                                                             |     |
|---------|----|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| [10: 8] | RW | ddrpll_ref_cksel | DDRPHY PLL 参考时钟选择。<br>000: 24MHz;<br>001: 525MHz;<br>010: 496MHz; (不可展频)<br>011: 450Mhz; (不可展频)<br>100: 350MHz (333Mhz 需要将<br>apllvco 调为 2000MHz);<br>101: 263MHz;<br>110: 198MHz;<br>111: 99MHz;<br>DFI 时钟频率请参考 DDRPHY PLL 相<br>关文档自行按需配置。 | 0x0 |
| [7:5]   | RW | reserved         | 保留。                                                                                                                                                                                                                                         | 0x0 |
| [4]     | RW | ddr_bus_cken     | DDRC 总线域门控配置寄存器<br>0: 关闭时钟;<br>1: 打开时钟。                                                                                                                                                                                                     | 0x1 |
| [3:2]   | RW | reserved         | 保留。                                                                                                                                                                                                                                         | 0x0 |
| [1]     | RW | dfi_srst_req     | DDRC DFI 域软复位请求<br>0: 撤消复位;<br>1: 复位。                                                                                                                                                                                                       | 0x0 |
| [0]     | RW | ddr_bus_srst_req | DDRC 总线侧软复位请求 (复位范围包<br>括 ddr/ddrphy 的 apb, ddr 的 aix 总<br>线)<br>0: 撤消复位;<br>1: 复位。                                                                                                                                                         | 0x0 |

## PERI\_CRG32

peri\_crg32 为 SOC 时钟选择寄存器。

Offset Address:0x80 Total Reset Value:0x00000000

| Bits    | Access | Name         | Description               | Reset    |
|---------|--------|--------------|---------------------------|----------|
| [31:21] | RW     | reserved     | 保留。                       | 0x000000 |
| [20:18] | RW     | mdaaxi_cksel | MDAAXI 时钟选择。<br>000: 24M; | 0x0      |



|         |    |              |                                                                                                                                             |     |
|---------|----|--------------|---------------------------------------------------------------------------------------------------------------------------------------------|-----|
|         |    |              | 001: 525MHz;<br>010: 496MHz;<br>011: 420MHz;<br>100: 350 (333) MHz;<br>其它保留                                                                 |     |
| [17]    | RW | reserved     | 保留。                                                                                                                                         | 0x0 |
| [16]    | RW | sysapb_cksel | SYSAPB 时钟选择。<br>0: 24M;<br>1: 50MHz。                                                                                                        | 0x0 |
| [15]    | RW | reserved     | 保留。                                                                                                                                         | 0x0 |
| [14]    | RW | syscfg_cksel | SYSCFG 时钟选择。<br>0: 24M;<br>1: 100MHz;                                                                                                       | 0x0 |
| [13]    | RW | reserved     | 保留。                                                                                                                                         | 0x0 |
| [12:11] | RW | sysaxi_cksel | SYSAXI 时钟选择。<br>0: 24M;<br>1: 198MHz;<br>2: 150MHz;<br>3: 99MHz                                                                             | 0x0 |
| [10:3]  | RW | reserved     | 保留。                                                                                                                                         | 0x0 |
| [2:0]   | RW | a7_cksel     | A7 CORE 时钟选择。<br>000: 24M;<br>001: 1188MHz;<br>010: 993MHz;<br>011: 900MHz;<br>100: 700MHz;<br>101: 594MHz;<br>110: 396MHz;<br>111: 350MHz。 | 0x0 |

## PERI\_CRG40

peri\_crg40 为媒体功能模块频率配置寄存器。

Offset Address:0xa0 Total Reset Value:0x00000000

| Bits    | Access | Name         | Description                                                                                                                           | Reset |
|---------|--------|--------------|---------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:26] | RW     | reserved     | 保留。                                                                                                                                   | 0x000 |
| [12]    | RW     | usrslv_cksel | USERDEF SLV 时钟选择。<br>0: 496MHZ;<br>1: 396MHZ。                                                                                         | 0x0   |
| [11:9]  | RW     | vpss_cksel   | VPSS 时钟选择。<br>000: 396MHz;<br>001: 350Mhz;<br>010: 300MHz;<br>011: 263MHz;<br>100: 200MHZ;<br>101: 150MHz;<br>110: 100MHz;<br>其它: 保留。 | 0x0   |
| [8:6]   | RW     | vipro_cksel  | VIPROC 时钟选择。<br>000: 300MHz;<br>001: 263MHZ;<br>010: 198MHZ;<br>011: 150MHZ;<br>100: 100MHz;<br>其它: 保留。                               | 0x0   |

## PERI\_CRG41

peri\_crg41 为 VPU 时钟及软复位控制寄存器。

Offset Address:0xa4 Total Reset Value:0x00000000

| Bits   | Access | Name         | Description                            | Reset      |
|--------|--------|--------------|----------------------------------------|------------|
| [31:2] | RW     | reserved     | 保留。                                    | 0x00000000 |
| [1]    | RW     | vpu_cken     | VPU 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x0        |
| [0]    | RW     | vpu_srst_req | VPU 的软复位请求。<br>0: 撤消复位;<br>1: 复位。      | 0x0        |

## PERI\_CRG42

peri\_crg42 为 VPSS 时钟及软复位控制寄存器。  
Offset Address:0xa8    Total Reset Value:0x00000002

| Bits   | Access | Name          | Description                             | Reset      |
|--------|--------|---------------|-----------------------------------------|------------|
| [31:2] | RW     | reserved      | 保留。                                     | 0x00000000 |
| [1]    | RW     | vpss_cken     | VPSS 时钟门控配置寄存器,<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x1        |
| [0]    | RW     | vpss_srst_req | VPSS 的软复位请求。<br>0: 撤消复位;<br>1: 复位。      | 0x0        |

## PERI\_CRG43

peri\_crg43 为 VGS 时钟及软复位控制寄存器。  
Offset Address:0xac    Total Reset Value:0x00000000

| Bits   | Access | Name         | Description                            | Reset      |
|--------|--------|--------------|----------------------------------------|------------|
| [31:2] | RW     | reserved     | 保留。                                    | 0x00000000 |
| [1]    | RW     | vgs_cken     | VGS 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x0        |
| [0]    | RW     | vgs_srst_req | VGS 的软复位请求。<br>0: 撤消复位;<br>1: 复位。      | 0x0        |

## PERI\_CRG44

peri\_crg44 为 JPGE 时钟及软复位控制寄存器。  
Offset Address:0xb0    Total Reset Value:0x00000000

| Bits   | Access | Name      | Description                             | Reset      |
|--------|--------|-----------|-----------------------------------------|------------|
| [31:2] | RW     | reserved  | 保留。                                     | 0x00000000 |
| [1]    | RW     | jpge_cken | JPGE 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x0        |

|     |    |               |                                    |     |
|-----|----|---------------|------------------------------------|-----|
| [0] | RW | jpge_srst_req | JPGE 的软复位请求。<br>0: 撤消复位;<br>1: 复位。 | 0x0 |
|-----|----|---------------|------------------------------------|-----|

## PERI\_CRG45

peri\_crg45 为 NPU 时钟及软复位控制寄存器。

Offset Address:0xb4 Total Reset Value:0x00000000

| Bits   | Access | Name         | Description                     | Reset  |
|--------|--------|--------------|---------------------------------|--------|
| [31:2] | RW     | reserved     | 保留。                             | 0x0000 |
| [1]    | RW     | npu_cken     | NPU 时钟门控。<br>0: 关断;<br>1: 打开。   | 0x0    |
| [0]    | RW     | npu_srst_req | NPU 软复位请求。<br>0: 不复位;<br>1: 复位。 | 0x0    |

## PERI\_CRG50

peri\_crg50 为 USRDEF SLV 时钟及软复位控制寄存器。

Offset Address:0xc8 Total Reset Value:0x00000000

| Bits   | Access | Name            | Description                                   | Reset      |
|--------|--------|-----------------|-----------------------------------------------|------------|
| [31:2] | RW     | reserved        | 保留。                                           | 0x00000000 |
| [1]    | RW     | usrslv_cken     | USRDEF SLV 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x0        |
| [0]    | RW     | usrslv_srst_req | USRDEF SLV 的软复位。<br>0: 撤消复位;<br>1: 复位。        | 0x0        |

## PERI\_CRG60

peri\_crg60 为 SENSOR 时钟及软复位控制寄存器。

Offset Address:0xf0 Total Reset Value:0x00000000

| Bits | Access | Name | Description | Reset |
|------|--------|------|-------------|-------|
|------|--------|------|-------------|-------|

|         |    |                  |                                                                                                                                                                          |            |
|---------|----|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| [31:30] | RW | reserved         | 保留。                                                                                                                                                                      | 0x00000000 |
| [29:26] | RW | sensor3_cksel    | SENSOR3 CLKOUT 时钟配置寄存器。<br>0x0: 74.25MHz<br>0x2: 54MHz<br>0x3: 50MHZ<br>0x4: 37.125Mhz<br>0x6: 27Mhz<br>0x7: 25MHZ<br>0x8: 24Mhz<br>0x9: 20Mhz<br>0xb: 12Mhz<br>其它: 保留   | 0x0        |
| [25]    | RW | sensor3_srst_req | SENSOR3 软复位请求。<br>0: 撤消复位;<br>1: 复位。                                                                                                                                     | 0x0        |
| [24]    | RW | sensor3_cken     | SENSOR3 CLKOUT 时钟门控。<br>0: 时钟关闭;<br>1: 时钟打开。                                                                                                                             | 0x0        |
| [23:22] | RW | reserved         | 保留。                                                                                                                                                                      | 0x0        |
| [21:18] | RW | sensor2_cksel    | SENSOR2 CLKOUT 时钟配置寄存器。<br>0x0: 74.25MHz;<br>0x2: 54MHz;<br>0x3: 50MHZ<br>0x4: 37.125Mhz<br>0x6: 27Mhz<br>0x7: 25MHZ<br>0x8: 24Mhz<br>0x9: 20Mhz<br>0xb: 12Mhz<br>其它: 保留 | 0x0        |
| [17]    | RW | sensor2_srst_req | SENSOR2 软复位请求。<br>0: 撤消复位;<br>1: 复位。                                                                                                                                     | 0x0        |

|         |    |                  |                                                                                                                                                                          |     |
|---------|----|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| [16]    | RW | sensor2_cken     | SENSOR2 CLKOUT 时钟门控。<br>0: 时钟关闭;<br>1: 时钟打开。                                                                                                                             | 0x0 |
| [15:14] | RW | reserved         | 保留。                                                                                                                                                                      | 0x0 |
| [13:10] | RW | sensor1_cksel    | SENSOR1 CLKOUT 时钟配置寄存器。<br>0x0: 74.25MHz<br>0x2: 54MHz<br>0x3: 50MHZ<br>0x4: 37.125Mhz<br>0x6: 27Mhz<br>0x7: 25MHZ<br>0x8: 24Mhz<br>0x9: 20Mhz<br>0xb: 12Mhz<br>其它: 保留   | 0x0 |
| [9]     | RW | sensor1_srst_req | SENSOR1 软复位请求。<br>0: 撤消复位;<br>1: 复位。                                                                                                                                     | 0x0 |
| [8]     | RW | sensor1_cken     | SENSOR1 CLKOUT 时钟门控。<br>0: 时钟关闭;<br>1: 时钟打开。                                                                                                                             | 0x0 |
| [7:6]   | RW | reserved         | 保留。                                                                                                                                                                      | 0x0 |
| [5:2]   | RW | sensor0_cksel    | SENSOR0 CLKOUT 时钟配置寄存器。<br>0x0: 74.25MHz;<br>0x2: 54MHz;<br>0x3: 50MHZ<br>0x4: 37.125Mhz<br>0x6: 27Mhz<br>0x7: 25MHZ<br>0x8: 24Mhz<br>0x9: 20Mhz<br>0xb: 12Mhz<br>其它: 保留 | 0x0 |

|     |    |                  |                                              |     |
|-----|----|------------------|----------------------------------------------|-----|
| [1] | RW | sensor0_srst_req | SENSOR0 软复位请求。<br>0: 撤消复位;<br>1: 复位。         | 0x0 |
| [0] | RW | sensor0_cken     | SENSOR0 CLKOUT 时钟门控。<br>0: 时钟关闭;<br>1: 时钟打开。 | 0x0 |

## PERI\_CRG61

peri\_crg61 为 VI 时钟及软复位控制寄存器。

Offset Address:0xf4 Total Reset Value:0x00000000

| Bits    | Access | Name               | Description                                       | Reset |
|---------|--------|--------------------|---------------------------------------------------|-------|
| [31:29] | RW     | reserved           | 保留。                                               | 0x000 |
| [28]    | RW     | vi_dvpin_1_pctrl   | VICAP dvp_1 管脚输入时钟相位选择。<br>0: 时钟不取反;<br>1: 时钟取反   | 0x0   |
| [27]    | RW     | vi_dvpin_0_pctrl   | VICAP dvp_0 管脚输入时钟相位选择。<br>0: 时钟不取反;<br>1: 时钟取反   | 0x0   |
| [26]    | RW     | vi_bt656in_2_pctrl | VICAP bt656_2 管脚输入时钟相位选择。<br>0: 时钟不取反;<br>1: 时钟取反 | 0x0   |
| [25]    | RW     | vi_bt656in_1_pctrl | VICAP bt656_1 管脚输入时钟相位选择。<br>0: 时钟不取反;<br>1: 时钟取反 | 0x0   |
| [24]    | RW     | vi_bt656in_0_pctrl | VICAP bt656_0 管脚输入时钟相位选择。<br>0: 时钟不取反;<br>1: 时钟取反 | 0x0   |
| [23:21] | RW     | reserved           | 保留。                                               | 0x0   |
| [20]    | RW     | vi_cmos_cken       | VI CMOS in 时钟门控。<br>0: 时钟关闭;<br>1: 时钟打开。          | 0x0   |

|         |    |                    |                                      |     |
|---------|----|--------------------|--------------------------------------|-----|
| [19:18] | RW | reserved           | 保留。                                  | 0x0 |
| [17]    | RW | vicap_all_srst_req | VICAP 总软复位请求。<br>0: 不复位;<br>1: 复位。   | 0x0 |
| [16]    | RW | vi_fe7_srst_req    | VI FE7 软复位请求。<br>0: 不复位;<br>1: 复位。   | 0x0 |
| [15]    | RW | vi_fe6_srst_req    | VI FE6 软复位请求。<br>0: 不复位;<br>1: 复位。   | 0x0 |
| [14]    | RW | vi_fe5_srst_req    | VI FE5 软复位请求。<br>0: 不复位;<br>1: 复位。   | 0x0 |
| [13]    | RW | vi_fe4_srst_req    | VI FE4 软复位请求。<br>0: 不复位;<br>1: 复位。   | 0x0 |
| [12]    | RW | vi_fe3_srst_req    | VI FE3 软复位请求。<br>0: 不复位;<br>1: 复位。   | 0x0 |
| [11]    | RW | vi_fe2_srst_req    | VI FE2 软复位请求。<br>0: 不复位;<br>1: 复位。   | 0x0 |
| [10]    | RW | vi_fe1_srst_req    | VI FE1 软复位请求。<br>0: 不复位;<br>1: 复位。   | 0x0 |
| [9]     | RW | vi_fe0_srst_req    | VI FE0 软复位请求。<br>0: 不复位;<br>1: 复位。   | 0x0 |
| [8]     | RW | vi_p4_srst_req     | VI Port4 软复位请求。<br>0: 不复位;<br>1: 复位。 | 0x0 |



|       |    |                |                                      |     |
|-------|----|----------------|--------------------------------------|-----|
| [7]   | RW | vi_p3_srst_req | VI Port3 软复位请求。<br>0: 不复位;<br>1: 复位。 | 0x0 |
| [6]   | RW | vi_p2_srst_req | VI Port2 软复位请求。<br>0: 不复位;<br>1: 复位。 | 0x0 |
| [5]   | RW | vi_p1_srst_req | VI Port1 软复位请求。<br>0: 不复位;<br>1: 复位。 | 0x0 |
| [4]   | RW | vi_p0_srst_req | VI Port0 软复位请求。<br>0: 不复位;<br>1: 复位。 | 0x0 |
| [3:1] | RW | reserved       | 保留。                                  | 0x0 |
| [0]   | RW | vicap_cken     | VICAP 时钟门控。<br>0: 时钟关闭;<br>1: 时钟打开。  | 0x0 |

## PERI\_CRG62

peri\_crg62 为 MIPI RX 时钟及软复位控制寄存器。  
Offset Address:0xf8 Total Reset Value:0x00000000

| Bits    | Access | Name                | Description                                        | Reset    |
|---------|--------|---------------------|----------------------------------------------------|----------|
| [31:11] | RW     | reserved            | 保留。                                                | 0x000000 |
| [9]     | RW     | mipi_phy0_cken      | MIPI PHY0 时钟门控。<br>0: 时钟关闭;<br>1: 时钟打开。            | 0x0      |
| [8]     | RW     | mipi_hs_pctrl       | MIPIPHY HS 接口时钟相位控制。默认正向。<br>0: 时钟不取反;<br>1: 时钟取反。 | 0x0      |
| [7:4]   | RW     | reserved            | 保留。                                                | 0x0      |
| [3]     | RW     | mipi_rx0_4line_cken | MIPI RX0 4line 模式时钟门控。                             | 0x0      |

|     |    |                       |                                                                        |     |
|-----|----|-----------------------|------------------------------------------------------------------------|-----|
|     |    |                       | 0: 时钟关闭;<br>1: 时钟打开。                                                   |     |
| [2] | RW | mipi_deskew_ck<br>sel | 4lane mipirx rxbyteclkhs_mipi 选择<br>0: clk_hs0_phy0<br>1: clk_hs0_phy1 | 0x0 |
| [1] | RW | mipi_rx1_cken         | MIPI RX1 时钟门控。<br>0: 时钟关闭;<br>1: 时钟打开。                                 | 0x0 |
| [0] | RW | mipi_rx0_cken         | MIPI RX0 时钟门控。<br>0: 时钟关闭;<br>1: 时钟打开。                                 | 0x0 |

## PERI\_CRG63

peri\_crg63 为 VIPROC 时钟及软复位控制寄存器。  
Offset Address:0xfc Total Reset Value:0x00000000

| Bits   | Access | Name            | Description                          | Reset     |
|--------|--------|-----------------|--------------------------------------|-----------|
| [31:5] | RW     | reserved        | 保留。                                  | 0x0000000 |
| [4]    | RW     | viproc_srst_req | VIPROC 软复位请求。<br>0: 不复位;<br>1: 复位。   | 0x0       |
| [3:1]  | RW     | reserved        | 保留。                                  | 0x0       |
| [0]    | RW     | viproc_cken     | VIPROC 时钟门控。<br>0: 时钟关闭;<br>1: 时钟打开。 | 0x0       |

## PERI\_CRG65

peri\_crg65 为 LCD 分配器时钟控制寄存器。  
Offset Address:0x104 Total Reset Value:0x000FFFFF

| Bits    | Access | Name         | Description                   | Reset |
|---------|--------|--------------|-------------------------------|-------|
| [31:29] | RW     | reserved     | 保留。                           | 0x0   |
| [28]    | RW     | lcddiv_cksel | LCD 分频器时钟源头选择。<br>0: 1188MHZ; | 0x0   |

|        |    |             |                                                                             |            |
|--------|----|-------------|-----------------------------------------------------------------------------|------------|
|        |    |             | 1: APLL VCO 时钟。                                                             |            |
| [27]   | RW | lcddiv_cken | LCD 分频器分频使能。                                                                | 0x0        |
| [26:0] | RW | lcddiv_cfg  | LCD 分频器分频配置。假定目标频率 X (MHz), 则 $lcd\_div = (X/分频源头时钟 (默认 1188M)) * 2^{27}$ 。 | 0x00FFFFFF |

## PERI\_CRG77

peri\_crg77 为 IVE 相关的时钟及软复位控制寄存器。

Offset Address:0x134 Total Reset Value:0x00000004

| Bits   | Access | Name         | Description                                                      | Reset   |
|--------|--------|--------------|------------------------------------------------------------------|---------|
| [31:4] | RW     | reserved     | 保留。                                                              | 0x00000 |
| [3:2]  | RW     | ive_cksel    | IVE 时钟选择:<br>0: 496MHz;<br>1: 420MHz;<br>2: 396MHz;<br>3: 297MHz | 0x1     |
| [1]    | RW     | ive_srst_req | IVE 的软复位请求。<br>0: 撤消复位;<br>1: 复位。                                | 0x0     |
| [0]    | RW     | ive_cken     | IVE 的软时钟门控。<br>0: 关闭时钟;<br>1: 打开时钟。                              | 0x0     |

## PERI\_CRG80

peri\_crg80 为 USB2 相关的时钟及软复位控制寄存器。

Offset Address:0x140 Total Reset Value:0x0000130F

| Bits    | Access | Name               | Description                                              | Reset   |
|---------|--------|--------------------|----------------------------------------------------------|---------|
| [31:14] | RW     | reserved           | 保留。                                                      | 0x00000 |
| [13]    | RW     | usb2_freeclk_cksel | USB2 free 时钟选择。<br>0: 选择 UTMI 时钟;<br>1: 选择 free 60M 时钟;。 | 0x0     |
| [12]    | RW     | usb2_utmi_cken     | USB2 控制器 UTMI 时钟门控。                                      | 0x1     |

|     |    |                    |                                            |     |
|-----|----|--------------------|--------------------------------------------|-----|
|     |    |                    | 0: 关闭时钟;<br>1: 打开时钟。                       |     |
| [9] | RW | usb2_ref_cken      | USB2 控制器 REF 时钟门控。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x1 |
| [8] | RW | usb2_bus_cken      | USB2 控制器总线时钟门控。<br>0: 关闭时钟;<br>1: 打开时钟。    | 0x1 |
| [7] | RW | reserved           | 保留。                                        | 0x0 |
| [3] | RW | usb2_vcc_srst_req  | USB2 控制器 VCC 软复位请求。<br>0: 不复位;<br>1: 复位。   | 0x1 |
| [2] | RW | usb2_phy_xtal_cken | USB2PHY 晶振参考时钟门控<br>0: 关闭时钟;<br>1: 打开时钟。   | 0x1 |
| [1] | RW | usb2_phy_por_srst  | USB2 PHY POR 软复位请求。<br>0: 不复位;<br>1: 复位。   | 0x1 |
| [0] | RW | usb2_phy_cfg_req   | USB2 PHY CFG 软复位请求。<br>0: 不复位;<br>1: 复位。   | 0x1 |

## PERI\_CRG81

peri\_crg81 为 FMC 相关的时钟及软复位控制寄存器。  
Offset Address:0x144 Total Reset Value:0x00000002

| Bits   | Access | Name      | Description                                                                                                       | Reset     |
|--------|--------|-----------|-------------------------------------------------------------------------------------------------------------------|-----------|
| [31:5] | RW     | reserved  | 保留。                                                                                                               | 0x0000000 |
| [4:2]  | RW     | fmc_cksel | FMC 时钟源选择。(SDR 模式下,<br>PHY_CLK_OUT 为原时钟的 DIV2; DDR<br>模式下, 为原时钟的 DIV4)<br>000: 24MHz;<br>001: 396MHz; (预留频点, 不使用) | 0x0       |

|     |    |              |                                                                                       |     |
|-----|----|--------------|---------------------------------------------------------------------------------------|-----|
|     |    |              | 010: 297MHz; (仅 DDR 模式可选)<br>011: 198MHz;<br>100: 148.5MHz;<br>101: 99MHz;<br>其他: 保留。 |     |
| [1] | RW | fmc_cken     | FMC 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。                                                | 0x1 |
| [0] | RW | fmc_srst_req | FMC 的软复位请求。<br>0: 撤消复位;<br>1: 复位。                                                     | 0x0 |

## PERI\_CRG91

peri\_crg91 为 FEPHY 钟及软复位控制寄存器。  
Offset Address:0x16c Total Reset Value:0x00000001

| Bits   | Access | Name             | Description                            | Reset    |
|--------|--------|------------------|----------------------------------------|----------|
| [31:3] | RW     | reserved         | 保留。                                    | 0x000000 |
| [2]    | RW     | fephy_apbbrg_req | FEPHY APB 桥复位请求<br>0: 撤销复位;<br>1: 复位。  | 0x0      |
| [1]    | RW     | fephy_srst_req   | FEPHY 软复位请求。<br>0: 撤销复位;<br>1: 复位。     | 0x0      |
| [0]    | RW     | fephy_cken       | ETH 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x1      |

## PERI\_CRG92

peri\_crg92 为 FEPHY 钟及软复位控制寄存器。  
Offset Address:0x170 Total Reset Value:0x00000001

| Bits   | Access | Name     | Description | Reset    |
|--------|--------|----------|-------------|----------|
| [31:6] | RW     | reserved | 保留。         | 0x000000 |

|     |    |              |                                                     |     |
|-----|----|--------------|-----------------------------------------------------|-----|
| [5] | RW | rmii_mode    | RMII 模式控制<br>0: INT (SOC 给时钟);<br>1: EXT (对接芯片给时钟)。 | 0x0 |
| [3] | RW | reserved     | 保留。                                                 | 0x0 |
| [1] | RW | mac_srst_req | GMAC 软复位请求。<br>0: 撤销复位;<br>1: 复位。                   | 0x0 |
| [0] | RW | mac_cken     | GMAC 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。             | 0x1 |

### PERI\_CRG93

peri\_crg93 为 MCU 相关时钟及软复位控制寄存器。  
Offset Address:0x174 Total Reset Value:0x00000700

| Bits    | Access | Name           | Description                                                    | Reset    |
|---------|--------|----------------|----------------------------------------------------------------|----------|
| [31:11] | RW     | reserved       | 保留。                                                            | 0x000000 |
| [10]    | RW     | mcu_bus_srst   | MCU BUS 软复位请求。<br>0: 撤销复位;<br>1: 复位。                           | 0x1      |
| [9]     | RW     | mcu_dbg_srst   | MCU DBG 软复位请求。<br>0: 撤销复位;<br>1: 复位。                           | 0x1      |
| [8]     | RW     | mcu_core_srst  | MCU CORE 软复位请求。<br>0: 撤销复位;<br>1: 复位。                          | 0x1      |
| [7:6]   | RW     | mcu_dbg_cksel  | MCU DBG 时钟源选择<br>0: 24M<br>1: 297MHz<br>2: 198Mhz<br>3: 100MHz | 0x0      |
| [5]     | RW     | reserved       | 保留                                                             | 0x0      |
| [4:3]   | RW     | mcu_core_cksel | MCU CORE 时钟源选择                                                 | 0x0      |

|     |    |               |                                               |     |
|-----|----|---------------|-----------------------------------------------|-----|
|     |    |               | 0: 24m<br>1: 396MHz<br>2: 297Mhz<br>3: 198MHz |     |
| [2] | RW | mcu_bus_cken  | BUS 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。        | 0x0 |
| [1] | RW | mcu_dbg_cken  | DBG 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。        | 0x0 |
| [0] | RW | mcu_core_cken | CORE 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。       | 0x0 |

## PERI\_CRG95

peri\_crg95 为 MCU 系统总线桥时钟和复位寄存器  
Offset Address:0x17c Total Reset Value:0x00000001

| Bits   | Access | Name         | Description                           | Reset    |
|--------|--------|--------------|---------------------------------------|----------|
| [31:2] | RW     | reserved     | 保留。                                   | 0x000000 |
| [1]    | RW     | mcu_brg_srst | MCU 总线桥软复位请求。<br>0: 撤销复位;<br>1: 复位。   | 0x0      |
| [0]    | RW     | mcu_brg_cken | MCU 总线桥时钟使能寄存器<br>0: 关闭时钟;<br>1: 打开时钟 | 0x1      |

## PERI\_CRG100

peri\_crg100 为 GZIP 时钟及软复位控制寄存器。  
Offset Address:0x190 Total Reset Value:0x00000001

| Bits   | Access | Name          | Description  | Reset      |
|--------|--------|---------------|--------------|------------|
| [31:2] | RW     | reserved      | 保留。          | 0x00000000 |
| [1]    | RW     | gzip_srst_req | GZIP 的软复位请求。 | 0x0        |

|     |    |           |                                         |     |
|-----|----|-----------|-----------------------------------------|-----|
|     |    |           | 0: 撤消复位;<br>1: 复位。                      |     |
| [0] | RW | gzip_cken | GZIP 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x1 |

## PERI\_CRG101

peri\_crg101 为 DMAC 相关的时钟及软复位控制寄存器。

Offset Address:0x194 Total Reset Value:0x00000000

| Bits   | Access | Name          | Description                                 | Reset      |
|--------|--------|---------------|---------------------------------------------|------------|
| [31:3] | RW     | reserved      | 保留。                                         | 0x00000000 |
| [2]    | RW     | dmac_cken     | DMAC 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。     | 0x0        |
| [1]    | RW     | dmac_apb_cken | DMAC APB 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x0        |
| [0]    | RW     | dmac_srst_req | DMAC 的软复位请求。<br>0: 不复位;<br>1: 复位。           | 0x0        |

## PERI\_CRG102

peri\_crg102 为 DDRTEST 相关的时钟及软复位控制寄存器。

Offset Address:0x198 Total Reset Value:0x00000000

| Bits   | Access | Name             | Description                                | Reset      |
|--------|--------|------------------|--------------------------------------------|------------|
| [31:2] | RW     | reserved         | 保留。                                        | 0x00000000 |
| [1]    | RW     | ddrtest_cken     | DDRTEST 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x0        |
| [0]    | RW     | ddrtest_srst_req | DDRTEST 的软复位请求。<br>0: 撤消复位;<br>1: 复位。      | 0x0        |



## PERI\_CRG103

peri\_crg103 为 AIAO/AUDIO CODEC 时钟复位控制寄存器。

Offset Address:0x19c Total Reset Value:0x00000000

| Bits    | Access | Name           | Description                                                                     | Reset      |
|---------|--------|----------------|---------------------------------------------------------------------------------|------------|
| [31:11] | RW     | reserved       | 保留。                                                                             | 0x00000000 |
| [10]    | RW     | codec_adc_cken | CODEC ADC 部分时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。                                  | 0x0        |
| [9]     | RW     | codec_dac_cken | CODEC DAC 部分时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。                                  | 0x0        |
| [7:6]   | RW     | aiao_pll_cksel | AIAO PLL 时钟源头选择。<br>0x0: 993MHz;<br>0x1: 1188MHz;<br>0x2: 24MHz;<br>0x3: 22MHz。 | 0x0        |
| [5]     | RW     | aiao_cksel     | AIAO 时钟选择。<br>0: 50MHz;<br>1: 100MHz。                                           | 0x0        |
| [4]     | RW     | codec_cken     | AUDIO CODEC 时钟门控请求。<br>0: 关闭时钟;<br>1: 打开时钟。                                     | 0x0        |
| [3]     | RW     | codec_rst_req  | AUDIO CODEC 复位请求。<br>0: 撤消复位;<br>1: 复位。                                         | 0x0        |
| [2]     | RW     | aiao_pll_cken  | AIAO PLL 时钟门控。<br>0: 关闭时钟;<br>1: 打开时钟。                                          | 0x0        |
| [1]     | RW     | aiao_cken      | AIAO 时钟门控配置寄存器。控制 core, apb, axi 时钟<br>0: 关闭时钟;<br>1: 打开时钟。                     | 0x0        |

|     |    |               |                                   |     |
|-----|----|---------------|-----------------------------------|-----|
| [0] | RW | aiao_srst_req | AIAO 软复位请求。<br>0: 撤消复位;<br>1: 复位。 | 0x0 |
|-----|----|---------------|-----------------------------------|-----|

## PERI\_CRG104

peri\_crg104 为 HASH/CIPHER/RSA/TRNG/KLAD 相关的时钟及软复位控制寄存器。  
Offset Address:0x1a0 Total Reset Value:0x0000002A

| Bits    | Access | Name                   | Description                              | Reset    |
|---------|--------|------------------------|------------------------------------------|----------|
| [31:10] | RW     | reserved               | 保留。                                      | 0x000000 |
| [9]     | RW     | spacc_cken             | SPACC 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x0      |
| [8]     | RW     | spacc_srst_req_i<br>ni | SPACC 的软复位请求。<br>0: 撤消复位;<br>1: 复位。      | 0x0      |
| [7:6]   | RW     | reserved               | 保留。                                      | 0x0      |
| [5]     | RW     | rsa0_cken              | RSA0 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。  | 0x1      |
| [4]     | RW     | rsa0_srst_req          | RSA0 的软复位请求。<br>0: 撤消复位;<br>1: 复位。       | 0x0      |
| [3]     | RW     | trng_cken              | TRNG 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。  | 0x1      |
| [2]     | RW     | trng_srst_req          | TRNG 的软复位请求。<br>0: 撤消复位;<br>1: 复位。       | 0x0      |
| [1]     | RW     | klad_cken              | KLAD 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。  | 0x1      |

|     |    |               |                                    |     |
|-----|----|---------------|------------------------------------|-----|
| [0] | RW | klad_srst_req | KLAD 的软复位请求。<br>0: 撤消复位;<br>1: 复位。 | 0x0 |
|-----|----|---------------|------------------------------------|-----|

## PERI\_CRG110

peri\_crg110 为 PWM/I2C/UART/时钟及复位控制寄存器。  
Offset Address:0x1b8 Total Reset Value:0x00000001

| Bits    | Access | Name          | Description                          | Reset |
|---------|--------|---------------|--------------------------------------|-------|
| [31:28] | RW     | reserved      | 保留。                                  | 0x000 |
| [27]    | RW     | i2c3_srst_req | I2C3 的软复位请求。<br>0: 撤消复位;<br>1: 复位。   | 0x0   |
| [26]    | RW     | i2c2_srst_req | I2C2 的软复位请求。<br>0: 撤消复位;<br>1: 复位。   | 0x0   |
| [25]    | RW     | i2c1_srst_req | I2C1 的软复位请求。<br>0: 撤消复位;<br>1: 复位。   | 0x0   |
| [24]    | RW     | i2c0_srst_req | I2C0 的软复位请求。<br>0: 撤消复位;<br>1: 复位。   | 0x0   |
| [19]    | RW     | i2c3_cken     | I2C3 时钟门控控制。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x0   |
| [18]    | RW     | i2c2_cken     | I2C2 时钟门控控制。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x0   |
| [17]    | RW     | i2c1_cken     | I2C1 时钟门控控制。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x0   |
| [16]    | RW     | i2c0_cken     | I2C0 时钟门控控制。<br>0: 关闭时钟;             | 0x0   |

|         |    |                |                                          |     |
|---------|----|----------------|------------------------------------------|-----|
|         |    |                | 1: 打开时钟。                                 |     |
| [15:14] | RW | reserved       | 保留。                                      | 0x0 |
| [11]    | RW | uart3_srst_req | UART3 的软复位请求。<br>0: 撤消复位;<br>1: 复位。      | 0x0 |
| [10]    | RW | uart2_srst_req | UART2 的软复位请求。<br>0: 撤消复位;<br>1: 复位。      | 0x0 |
| [9]     | RW | uart1_srst_req | UART1 的软复位请求。<br>0: 撤消复位;<br>1: 复位。      | 0x0 |
| [8]     | RW | uart0_srst_req | UART0 的软复位请求。<br>0: 撤消复位;<br>1: 复位。      | 0x0 |
| [7:6]   | RW | reserved       | 保留。                                      | 0x0 |
| [3]     | RW | uart3_cken     | UART3 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x0 |
| [2]     | RW | uart2_cken     | UART2 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x0 |
| [1]     | RW | uart1_cken     | UART1 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x0 |
| [0]     | RW | uart0_cken     | UART0 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。 | 0x1 |

## PERI\_CRG111

peri\_crg111 为 SSP/UART/ADC/EFUSE/PWM 相关的时钟及软复位控制寄存器。  
Offset Address:0x1bc Total Reset Value:0x00000003

| Bits    | Access | Name          | Description                                                     | Reset |
|---------|--------|---------------|-----------------------------------------------------------------|-------|
| [31:28] | RW     | reserved      | 保留。                                                             | 0x000 |
| [27:26] | RW     | uart3_cksel   | UART3 时钟选择。<br>00: 24MHz;<br>01: 3MHz;<br>10: 74.25MHz;<br>其它保留 | 0x0   |
| [25:24] | RW     | uart2_cksel   | UART2 时钟选择。<br>00: 24MHz;<br>01: 3MHz;<br>10: 74.25MHz;<br>其它保留 | 0x0   |
| [23:22] | RW     | uart1_cksel   | UART1 时钟选择。<br>00: 24MHz;<br>01: 3MHz;<br>10: 74.25MHz;<br>其它保留 | 0x0   |
| [21:20] | RW     | uart0_cksel   | UART0 时钟选择。<br>00: 24MHz;<br>01: 3MHz;<br>10: 74.25MHz;<br>其它保留 | 0x0   |
| [18]    | RW     | ssp2_srst_req | SSP2 的软复位请求。<br>0: 撤消复位;<br>1: 复位。                              | 0x0   |
| [17]    | RW     | ssp1_srst_req | SSP1 的软复位请求。<br>0: 撤消复位;<br>1: 复位。                              | 0x0   |
| [16]    | RW     | ssp0_srst_req | SSP0 的软复位请求。<br>0: 撤消复位;<br>1: 复位。                              | 0x0   |
| [15]    | RW     | ssp_cksel     | SSP 时钟选择。<br>0: 594MHz;                                         | 0x0   |

|       |    |                 |                                                    |     |
|-------|----|-----------------|----------------------------------------------------|-----|
|       |    |                 | 1: 396MHz;                                         |     |
| [14]  | RW | ssp2_cken       | SSP2 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。            | 0x0 |
| [13]  | RW | ssp1_cken       | SSP1 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。            | 0x0 |
| [12]  | RW | ssp0_cken       | SSP0 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。            | 0x0 |
| [9:8] | RW | pwm_cksel       | PWM 时钟选择。<br>00: 3MHz;<br>01: 50MHz;<br>1X: 24MHz。 | 0x0 |
| [7]   | RW | pwm_cken        | PWM 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。             | 0x0 |
| [6]   | RW | pwm_srst_req    | PWM 的软复位请求。<br>0: 撤消复位;<br>1: 复位。                  | 0x0 |
| [5]   | RW | reserved        | 保留。                                                | 0x0 |
| [4]   | RW | pmc_srst_req    | PMC 的软复位请求。<br>0: 撤消复位;<br>1: 复位。                  | 0x0 |
| [3]   | RW | ls_adc_cken     | LS ADC 时钟门控配置寄存器。<br>0: 关闭时钟;<br>1: 打开时钟。          | 0x0 |
| [2]   | RW | ls_adc_srst_req | LS ADC 的软复位请求。<br>0: 撤消复位;<br>1: 复位。               | 0x0 |
| [1]   | RW | efuse_cken      | efuse 时钟门控配置寄存器。<br>0: 关闭时钟;                       | 0x1 |

|     |    |             |                                            |     |
|-----|----|-------------|--------------------------------------------|-----|
|     |    |             | 1: 打开时钟。                                   |     |
| [0] | RW | test_clk_en | 测试时钟使能寄存器。<br>0: 所有测试时钟关闭;<br>1: 所有测试时钟打开。 | 0x1 |

## PERI\_CRG\_PLL121

peri\_crg\_pll121 为 CBB 复位状态寄存器。

Offset Address:0x1e4 Total Reset Value:0x00000000

| Bits   | Access | Name              | Description                                                                                                                                                                                                                                                                                                                                                                                                     | Reset      |
|--------|--------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| [32]   | RO     | reserved          | 保留。                                                                                                                                                                                                                                                                                                                                                                                                             | 0x00000000 |
| [31:0] | RO     | cbb_softrst_state | CBB 复位状态。<br>0: 撤消成功<br>1: 复位。<br>其他: 保留<br>[24]: jpeg<br>[23]: spacc<br>[22]: gzip<br>[21]: vicap all<br>[20]: vicap fe7<br>[19]: vicap fe6<br>[18]: vicap fe5<br>[17]: vicap fe4<br>[16]: vicap fe3<br>[15]: vicap fe2<br>[14]: vicap fe1<br>[13]: vicap fe0<br>[12]: vicap pt4<br>[11]: vicap pt3<br>[10]: vicap pt2<br>[9]: vicap pt1<br>[8]: vicap pt0<br>[7]: viproc<br>[6]: npu<br>[5]: ive<br>[4]: vpss | 0x0        |

|  |  |  |                                               |  |
|--|--|--|-----------------------------------------------|--|
|  |  |  | [3]: tde<br>[2]: vgs<br>[1]: vpu<br>[0]: vout |  |
|--|--|--|-----------------------------------------------|--|

## PERI\_CRG123

peri\_crg123 为 DEBUG 保留寄存器。

Offset Address:0x1ec Total Reset Value:0x00000000

| Bits    | Access | Name       | Description | Reset  |
|---------|--------|------------|-------------|--------|
| [31:16] | RW     | reserved   | 保留。         | 0x0000 |
| [15:0]  | RW     | eco_reserv | 保留。         | 0x0000 |

## PERI\_CRG\_PLL124

peri\_crg\_pll124 为 CPU 高速 MEM 调速方式选择。

Offset Address:0x1f0 Total Reset Value:0x00000001

| Bits   | Access | Name              | Description                                               | Reset      |
|--------|--------|-------------------|-----------------------------------------------------------|------------|
| [31:1] | RW     | reserved          | 保留。                                                       | 0x00000000 |
| [0]    | RW     | mem_adjust_bypass | CPU 高速 Memory 调速配置方式。<br>0: 非 bypass 方式;<br>1: bypass 方式。 | 0x1        |

## PERI\_CRG125

peri\_crg125 为 eMMC 时钟复位及 SAM DLL 控制寄存器。

Offset Address:0x1f4 Total Reset Value:0x11010000

| Bits    | Access | Name                 | Description                              | Reset |
|---------|--------|----------------------|------------------------------------------|-------|
| [31:30] | RW     | reserved             | 保留。                                      | 0x0   |
| [29]    | RW     | emmc_drv_dll_rst_req | eMMC DRV DLL 软复位请求。<br>0: 不复位;<br>1: 复位。 | 0x0   |
| [28]    | RW     | emmc_cken            | eMMC 时钟门控配置。<br>0: 关闭;                   | 0x1   |



|         |    |                   |                                                                                                                                                                                                                                                |     |
|---------|----|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
|         |    |                   | 1: 打开。                                                                                                                                                                                                                                         |     |
| [27]    | RW | emmc_srst_req     | eMMC 的软复位请求。<br>0: 不复位;<br>1: 复位。                                                                                                                                                                                                              | 0x0 |
| [26:24] | RW | emmc_cksel        | eMMC 时钟选择。4X 时钟, 内部还会 4 分频得到最高 150MHz。<br>000: 400KHz;<br>001: 1600KHz;<br>010: 100MHz;<br>011: 200MHz;<br>100: 396MHz;<br>101: 496MHz;<br>110: 594MHz;<br>111: 700MHz。                                                                        | 0x1 |
| [23]    | RW | emmc_tx_srst_req  | eMMC TX 方向软复位请求。<br>0: 不复位;<br>1: 复位。                                                                                                                                                                                                          | 0x0 |
| [22]    | RW | emmc_rx_srst_req  | eMMC RX 方向软复位请求。<br>0: 不复位;<br>1: 复位。                                                                                                                                                                                                          | 0x0 |
| [21]    | RW | reserved          | 保留。                                                                                                                                                                                                                                            | 0x0 |
| [20:17] | RW | emmc_sam_dll_tune | eMMC SAM DLL 输入时钟相位控制:<br>0x0: 不校准;<br>0x1: 增加 1 级 delay;<br>0x2: 增加 2 级 delay;<br>0x3: 增加 3 级 delay;<br>...<br>0x7: 增加 7 级 delay;<br>0x8: 不校准;<br>0x9: 减少 1 级 delay;<br>0xA: 减少 2 级 delay;<br>0xB: 减少 3 级 delay;<br>...<br>0xF: 减少 7 级 delay; | 0x0 |

|         |    |                       |                                                                            |      |
|---------|----|-----------------------|----------------------------------------------------------------------------|------|
| [16]    | RW | emmc_sam_dll_slave_en | eMMC SAM DLL slave_en 信号。<br>0: slave 停止工作;<br>1: slave 开始工作。              | 0x1  |
| [15:14] | RW | reserved              | 保留。                                                                        | 0x0  |
| [13]    | RW | emmc_sam_dll_mode     | eMMC SAM DLL mode 信号。<br>0: 正常模式;<br>1: 由 emmc_sam_dll_ssel 控制 SLAVE LINE。 | 0x0  |
| [12:8]  | RW | reserved              | 保留。                                                                        | 0x00 |
| [7:0]   | RW | emmc_sam_dll_ssel     | eMMC SAM DLL ssel 信号。DLL SLAVE LINE delay 初始级数选择。                          | 0x00 |

## PERI\_CRG126

peri\_crg126 为 eMMC SAMB DLL 控制寄存器。  
Offset Address:0x1f8 Total Reset Value:0x00000008

| Bits   | Access | Name                   | Description                                                                                                                         | Reset      |
|--------|--------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------|------------|
| [31:5] | RW     | reserved               | 保留。                                                                                                                                 | 0x00000000 |
| [4:0]  | RW     | emmc_sample_b_cclk_sel | 边沿检测功能时，做 B 时钟调试用，时钟相位配置。<br>0x00: 0C;<br>0x01: 11.25C;<br>0x02: 22.5C;<br>...<br>0x1E: 337.5C;<br>0x1F: 348.75C。<br>默认值 90 度即 5'd8 | 0x08       |

## PERI\_CRG127

peri\_crg127 为 eMMC DRV DLL 控制寄存器。  
Offset Address:0x1fc Total Reset Value:0x10014000

| Bits    | Access | Name     | Description | Reset |
|---------|--------|----------|-------------|-------|
| [31:29] | RW     | reserved | 保留。         | 0x0   |

|         |    |                        |                                                                                                                                                                                                                                                |      |
|---------|----|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| [28:24] | RW | emmc_drv_clk_phase_sel | eMMC DRV 时钟相位配置。默认 180C<br>0x00: 0C;<br>0x01: 11.25C;<br>0x02: 22.5C;<br>0x3: 33.75<br>...<br>0x1E: 337.5C;<br>0x1F: 348.75C。                                                                                                                  | 0x10 |
| [23:21] | RW | reserved               | 保留。                                                                                                                                                                                                                                            | 0x0  |
| [20:17] | RW | emmc_drv_dll_tune      | eMMC DRV DLL 输入时钟相位控制:<br>0x0: 不校准;<br>0x1: 增加 1 级 delay;<br>0x2: 增加 2 级 delay;<br>0x3: 增加 3 级 delay;<br>...<br>0x7: 增加 7 级 delay;<br>0x8: 不校准;<br>0x9: 减少 1 级 delay;<br>0xA: 减少 2 级 delay;<br>0xB: 减少 3 级 delay;<br>...<br>0xF: 减少 7 级 delay; | 0x0  |
| [16]    | RW | emmc_drv_dll_slave_en  | eMMC drv DLL slave_en 信号。<br>0: slave 停止工作;<br>1: slave 开始工作。                                                                                                                                                                                  | 0x1  |
| [15]    | RW | emmc_drv_dll_bypass    | eMMC DRV DLL bypass 信号。<br>0: 正常模式, 输出时钟相对输入时钟移相, 具体移相度数由 SDIO 控制器相关寄存器配置°;<br>1: slave line bypass, 输出时钟不移相。                                                                                                                                  | 0x0  |
| [14]    | RW | emmc_drv_dll_stop      | eMMC DRV DLL stop 信号。<br>0: 使能时钟检测;<br>1: 禁止时钟检测。                                                                                                                                                                                              | 0x1  |
| [13]    | RW | emmc_drv_dll_mode      | eMMC DRV DLL mode 信号。<br>0: 正常模式;                                                                                                                                                                                                              | 0x0  |

|        |    |                   |                                                   |      |
|--------|----|-------------------|---------------------------------------------------|------|
|        |    |                   | 1: 由 emmc_sam_dll_ssel 控制 SLAVE LINE。             |      |
| [12:8] | RW | reserved          | 保留。                                               | 0x00 |
| [7:0]  | RW | emmc_drv_dll_ssel | eMMC DRV DLL ssel 信号。DLL SLAVE LINE delay 初始级数选择。 | 0x00 |

## PERI\_CRG128

peri\_crg128 为 eMMC DS DLL 控制寄存器。

Offset Address:0x200 Total Reset Value:0x00010000

| Bits    | Access | Name                 | Description                                                                                                                                                                                                                                   | Reset |
|---------|--------|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:21] | RW     | reserved             | 保留。                                                                                                                                                                                                                                           | 0x000 |
| [20:17] | RW     | emmc_ds_dll_tune     | eMMC ds DLL 输入时钟相位控制:<br>0x0: 不校准;<br>0x1: 增加 1 级 delay;<br>0x2: 增加 2 级 delay;<br>0x3: 增加 3 级 delay;<br>...<br>0x7: 增加 7 级 delay;<br>0x8: 不校准;<br>0x9: 减少 1 级 delay;<br>0xA: 减少 2 级 delay;<br>0xB: 减少 3 级 delay;<br>...<br>0xF: 减少 7 级 delay; | 0x0   |
| [16]    | RW     | emmc_ds_dll_slave_en | eMMC ds DLL slave_en 信号。<br>0: slave 停止工作;<br>1: slave 开始工作。                                                                                                                                                                                  | 0x1   |
| [15]    | RW     | emmc_ds_dll_bypass   | eMMC ds DLL bypass 信号。<br>0: 正常模式, 输出时钟相对输入时钟移相, 具体移相度数由 SDIO 控制器相关寄存器配置°;<br>1: slave line bypass, 输出时钟不移相。                                                                                                                                  | 0x0   |
| [14]    | RW     | reserved             | 保留。                                                                                                                                                                                                                                           | 0x0   |

|        |    |                  |                                                                          |      |
|--------|----|------------------|--------------------------------------------------------------------------|------|
| [13]   | RW | emmc_ds_dll_mode | eMMC ds DLL mode 信号。<br>0: 正常模式;<br>1: 由 emmc_ds_dll_ssel 控制 SLAVE LINE。 | 0x0  |
| [12:8] | RW | reserved         | 保留。                                                                      | 0x00 |
| [7:0]  | RW | emmc_ds_dll_ssel | eMMC ds DLL ssel 信号。DLL SLAVE LINE delay 初始级数选择。                         | 0x00 |

## PERI\_CRG129

peri\_crg129 为 eMMC DS180 DLL 控制寄存器。  
Offset Address:0x204 Total Reset Value:0x08010000

| Bits    | Access | Name                      | Description                                                                                                                                                                                                                 | Reset |
|---------|--------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:29] | RW     | reserved                  | 保留。                                                                                                                                                                                                                         | 0x0   |
| [28:24] | RW     | emmc_ds_180_clk_phase_sel | eMMC ds_180 时钟相位配置。默认 90C<br>0x00: 0C;<br>0x01: 11.25C;<br>0x02: 22.5C;<br>...<br>0x1E: 337.5C;<br>0x1F: 348.75C。                                                                                                           | 0x08  |
| [23:21] | RW     | reserved                  | 保留。                                                                                                                                                                                                                         | 0x0   |
| [20:17] | RW     | emmc_ds_180_dll_tune      | eMMC ds_180 DLL 输入时钟相位控制:<br>0x0: 不校准;<br>0x1: 增加 1 级 delay;<br>0x2: 增加 2 级 delay;<br>0x3: 增加 3 级 delay;<br>...<br>0x7: 增加 7 级 delay;<br>0x8: 不校准;<br>0x9: 减少 1 级 delay;<br>0xA: 减少 2 级 delay;<br>0xB: 减少 3 级 delay;<br>... | 0x0   |

|        |    |                          |                                                                                                                  |      |
|--------|----|--------------------------|------------------------------------------------------------------------------------------------------------------|------|
|        |    |                          | 0xF: 减少 7 级 delay;                                                                                               |      |
| [16]   | RW | emmc_ds_180_dll_slave_en | eMMC ds_180 DLL slave_en 信号。<br>0: slave 停止工作;<br>1: slave 开始工作。                                                 | 0x1  |
| [15]   | RW | emmc_ds_180_dll_bypass   | eMMC ds_180 DLL bypass 信号。<br>0: 正常模式, 输出时钟相对输入时钟移相, 具体移相度数由 SDIO 控制器相关寄存器配置°;<br>1: slave line bypass, 输出时钟不移相。 | 0x0  |
| [14]   | RW | reserved                 | 保留。                                                                                                              | 0x0  |
| [13]   | RW | emmc_ds_180_dll_mode     | eMMC ds_180 DLL mode 信号。<br>0: 正常模式;<br>1: 由 emmc_ds_180_dll_ssel 控制 SLAVE LINE。                                 | 0x0  |
| [12:8] | RW | reserved                 | 保留。                                                                                                              | 0x00 |
| [7:0]  | RW | emmc_ds_180_dll_ssel     | eMMC ds_180 DLL ssel 信号。DLL SLAVE LINE delay 初始级数选择。                                                             | 0x00 |

## PERI\_CRG130

peri\_crg130 为 eMMC SAM DLL 状态寄存器。

Offset Address:0x208 Total Reset Value:0x00000000

| Bits   | Access | Name               | Description                                                              | Reset      |
|--------|--------|--------------------|--------------------------------------------------------------------------|------------|
| [31:1] | RO     | reserved           | 保留。                                                                      | 0x00000000 |
| [0]    | RO     | emmc_sam_dll_ready | eMMC SAM DLL ready 信号。<br>0: DLL SLAVE not ready;<br>1: DLL SLAVE ready。 | 0x0        |

## PERI\_CRG132

peri\_crg132 为 eMMC DRV DLL 状态寄存器。

Offset Address:0x210 Total Reset Value:0x00000000

| Bits | Access | Name | Description | Reset |
|------|--------|------|-------------|-------|
|------|--------|------|-------------|-------|

|         |    |                       |                                                                                   |        |
|---------|----|-----------------------|-----------------------------------------------------------------------------------|--------|
| [31:16] | RO | reserved              | 保留。                                                                               | 0x0000 |
| [15]    | RO | emmc_drv_dll_lock     | eMMC drv DLL lock 信号。<br>0: DLL MASTER unlock;<br>1: DLL MASTER locked.           | 0x0    |
| [14]    | RO | emmc_drv_dll_ready    | eMMC drv DLL ready 信号。<br>0: DLL SLAVE not ready;<br>1: DLL SLAVE ready.          | 0x0    |
| [13]    | RO | emmc_drv_dll_overflow | eMMC drv DLL overflow 信号。<br>0: DLL MASTER unoverflow;<br>1: DLL MASTER overflow. | 0x0    |
| [12:8]  | RW | reserved              | 保留。                                                                               | 0x00   |
| [7:0]   | RO | emmc_drv_dll_mdly_tap | eMMC drv DLL mdly_tap 信号。<br>DLL MASTER LINE tap 值。                               | 0x00   |

### PERI\_CRG133

peri\_crg133 为 eMMC DS DLL 状态寄存器。

Offset Address:0x214 Total Reset Value:0x00000000

| Bits   | Access | Name              | Description                                                             | Reset      |
|--------|--------|-------------------|-------------------------------------------------------------------------|------------|
| [31:1] | RO     | reserved          | 保留。                                                                     | 0x00000000 |
| [0]    | RO     | emmc_ds_dll_ready | eMMC ds DLL ready 信号。<br>0: DLL SLAVE not ready;<br>1: DLL SLAVE ready. | 0x0        |

### PERI\_CRG134

peri\_crg134 为 eMMC DS180 DLL 状态寄存器。

Offset Address:0x218 Total Reset Value:0x00000000

| Bits   | Access | Name                         | Description | Reset      |
|--------|--------|------------------------------|-------------|------------|
| [31:5] | RO     | reserved                     | 保留。         | 0x00000000 |
| [4:1]  | RO     | reserve_for_eco_emmc_crg_out | 保留。         | 0x0        |

|     |    |                       |                                                                             |     |
|-----|----|-----------------------|-----------------------------------------------------------------------------|-----|
| [0] | RO | emmc_ds_180_dll_ready | eMMC ds_180 DLL ready 信号。<br>0: DLL SLAVE not ready;<br>1: DLL SLAVE ready. | 0x0 |
|-----|----|-----------------------|-----------------------------------------------------------------------------|-----|

## PERI\_CRG135

peri\_crg135 为 SDIO0 SAMB DLL 控制寄存器。

Offset Address:0x21c Total Reset Value:0x08000000

| Bits    | Access | Name                   | Description                                                                                                                         | Reset    |
|---------|--------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------|----------|
| [31:29] | RW     | reserved               | 保留。                                                                                                                                 | 0x0      |
| [28:24] | RW     | sdio0_sample_b_clk_sel | 边沿检测功能时，做 B 时钟调试用，时钟相位配置。<br>0x00: 0C;<br>0x01: 11.25C;<br>0x02: 22.5C;<br>...<br>0x1E: 337.5C;<br>0x1F: 348.75C。<br>默认值 90 度即 5'd8 | 0x08     |
| [23:0]  | RW     | reserved               | 保留                                                                                                                                  | 0x000000 |

## PERI\_CRG136

peri\_crg136 为 SDIO0 DRV DLL 控制寄存器。

Offset Address:0x220 Total Reset Value:0x10014000

| Bits    | Access | Name                    | Description                                                                                                      | Reset |
|---------|--------|-------------------------|------------------------------------------------------------------------------------------------------------------|-------|
| [31:29] | RW     | reserved                | 保留。                                                                                                              | 0x0   |
| [28:24] | RW     | sdio0_drv_clk_phase_sel | SDIO0 DRV 时钟相位配置。默认 180C<br>0x00: 0C;<br>0x01: 11.25C;<br>0x02: 22.5C;<br>...<br>0x1E: 337.5C;<br>0x1F: 348.75C。 | 0x10  |



|         |    |                        |                                                                                                                                                                                                                                       |      |
|---------|----|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| [23:21] | RW | reserved               | 保留。                                                                                                                                                                                                                                   | 0x0  |
| [20:17] | RW | sdio0_drv_dll_tune     | SDIO0 SAM DLL 输入时钟相位控制：<br>0x0：不校准；<br>0x1：增加 1 级 delay；<br>0x2：增加 2 级 delay；<br>0x3：增加 3 级 delay；<br>...<br>0x7：增加 7 级 delay；<br>0x8：不校准；<br>0x9：减少 1 级 delay；<br>0xA：减少 2 级 delay；<br>0xB：减少 3 级 delay；<br>...<br>0xF：减少 7 级 delay； | 0x0  |
| [16]    | RW | sdio0_drv_dll_slave_en | SDIO0 drv DLL slave_en 信号。<br>0：slave 停止工作；<br>1：slave 开始工作。                                                                                                                                                                          | 0x1  |
| [15]    | RW | sdio0_drv_dll_bypass   | SDIO0 SAM DLL bypass 信号。<br>0：正常模式，输出时钟相对输入时钟移相，具体移相度数由 SDIO 控制器相关寄存器配置°；<br>1：slave line bypass，输出时钟不移相。                                                                                                                             | 0x0  |
| [14]    | RW | sdio0_drv_dll_stop     | SDIO0 SAM DLL stop 信号。<br>0：使能时钟检测；<br>1：禁止时钟检测。                                                                                                                                                                                      | 0x1  |
| [13]    | RW | sdio0_drv_dll_mode     | SDIO0 SAM DLL mode 信号。<br>0：正常模式；<br>1：由 SDIO_drv_dll_ssel 控制 SLAVE LINE。                                                                                                                                                             | 0x0  |
| [12:8]  | RW | reserved               | 保留                                                                                                                                                                                                                                    | 0x00 |
| [7:0]   | RW | sdio0_drv_dll_ssel     | SDIO0 SAM DLL ssel 信号。DLL SLAVE LINE delay 初始级数选择。                                                                                                                                                                                    | 0x00 |

## PERI\_CRG137

peri\_crg137 为 SDIO0 SAM DLL 状态寄存器。

Offset Address:0x224 Total Reset Value:0x00000000

| Bits   | Access | Name                | Description                                                               | Reset      |
|--------|--------|---------------------|---------------------------------------------------------------------------|------------|
| [31:1] | RO     | reserved            | 保留。                                                                       | 0x00000000 |
| [0]    | RO     | sdio0_sam_dll_ready | SDIO0 SAM DLL ready 信号。<br>0: DLL SLAVE not ready;<br>1: DLL SLAVE ready. | 0x0        |

## PERI\_CRG138

peri\_crg138 为 SDIO0 DRV DLL 状态寄存器。

Offset Address:0x228 Total Reset Value:0x00000000

| Bits    | Access | Name                          | Description                                                                        | Reset |
|---------|--------|-------------------------------|------------------------------------------------------------------------------------|-------|
| [31:20] | RO     | reserved                      | 保留。                                                                                | 0x000 |
| [19:16] | RO     | reserve_for_eco_sdio0_crg_out | 保留。                                                                                | 0x0   |
| [15]    | RO     | sdio0_drv_dll_lock            | SDIO0 drv DLL lock 信号。<br>0: DLL MASTER unlock;<br>1: DLL MASTER locked.           | 0x0   |
| [14]    | RO     | sdio0_drv_dll_ready           | SDIO0 drv DLL ready 信号。<br>0: DLL SLAVE not ready;<br>1: DLL SLAVE ready.          | 0x0   |
| [13]    | RO     | sdio0_drv_dll_overflow        | SDIO0 drv DLL overflow 信号。<br>0: DLL MASTER unoverflow;<br>1: DLL MASTER overflow. | 0x0   |
| [12:8]  | RW     | reserved                      | 保留。                                                                                | 0x00  |
| [7:0]   | RO     | sdio0_drv_dll_mdly_tap        | SDIO0 drv DLL mdly_tap 信号。<br>DLL MASTER LINE tap 值。                               | 0x00  |

## PERI\_CRG139

peri\_crg139 为 SDIO0 时钟复位及 SAM DLL 控制寄存器。

Offset Address:0x22c Total Reset Value:0x01010000

| Bits    | Access | Name                       | Description                                                                                                                                                            | Reset |
|---------|--------|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:30] | RW     | reserved                   | 保留。                                                                                                                                                                    | 0x0   |
| [29]    | RW     | sdio0_drv_dll_sr<br>st_req | SDIO0 DRV DLL 软复位请求。<br>0: 不复位;<br>1: 复位。                                                                                                                              | 0x0   |
| [28]    | RW     | sdio0_cken                 | SDIO0 时钟门控配置。<br>0: 关闭;<br>1: 打开。                                                                                                                                      | 0x0   |
| [27]    | RW     | sdio0_srst_req             | SDIO0 的软复位请求。<br>0: 不复位;<br>1: 复位。                                                                                                                                     | 0x0   |
| [26:24] | RW     | sdio0_cksel                | SDIO0 时钟选择。4X 时钟, 内部还会 4 分<br>频得到最高 50MHz。<br>000: 400KHz;<br>001: 1600KHz;<br>010: 100MHz;<br>011: 200MHz;                                                            | 0x1   |
| [23]    | RW     | sdio0_tx_srst_re<br>q      | sdio0 TX 方向软复位请求。<br>0: 不复位;<br>1: 复位。                                                                                                                                 | 0x0   |
| [22]    | RW     | sdio0_rx_srst_re<br>q      | sdio0 RX 方向软复位请求。<br>0: 不复位;<br>1: 复位。                                                                                                                                 | 0x0   |
| [21]    | RW     | reserved                   | 保留。                                                                                                                                                                    | 0x0   |
| [20:17] | RW     | sdio0_sam_dll_t<br>une     | SDIO0 SAM DLL 输入时钟相位控制:<br>0x0: 不校准;<br>0x1: 增加 1 级 delay;<br>0x2: 增加 2 级 delay;<br>0x3: 增加 3 级 delay;<br>...<br>0x7: 增加 7 级 delay;<br>0x8: 不校准;<br>0x9: 减少 1 级 delay; | 0x0   |

|         |    |                        |                                                                              |      |
|---------|----|------------------------|------------------------------------------------------------------------------|------|
|         |    |                        | 0xA: 减少 2 级 delay;<br>0xB: 减少 3 级 delay;<br>...<br>0xF: 减少 7 级 delay;        |      |
| [16]    | RW | sdio0_sam_dll_slave_en | SDIO0 samb DLL slave_en 信号。<br>0: slave 停止工作;<br>1: slave 开始工作。              | 0x1  |
| [15:14] | RW | reserved               | 保留。                                                                          | 0x0  |
| [13]    | RW | sdio0_sam_dll_mode     | SDIO0 SAM DLL mode 信号。<br>0: 正常模式;<br>1: 由 SDIO0_sam_dll_ssel 控制 SLAVE LINE。 | 0x0  |
| [12:8]  | RW | reserved               | 保留。                                                                          | 0x00 |
| [7:0]   | RW | sdio0_sam_dll_ssel     | SDIO0 SAM DLL ssel 信号。DLL SLAVE LINE delay 初始级数选择。                           | 0x00 |

## PERI\_CRG140

peri\_crg140 为 SDIO1 SAMB DLL 控制寄存器。  
Offset Address:0x230 Total Reset Value:0x00000000

| Bits    | Access | Name                         | Description                                                                                                                          | Reset |
|---------|--------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:29] | RW     | reserved                     | 保留。                                                                                                                                  | 0x0   |
| [28:24] | RW     | sdio1_sample_b_clk_sel       | 边沿检测功能时，做 B 时钟调试用，时钟相位配置。<br>0x00: 0C;<br>0x01: 11.25C;<br>0x02: 22.5C;<br>...<br>0x1E: 337.5C;<br>0x1F: 348.75C。<br>默认值 90 度即 5'd8。 | 0x08  |
| [23:0]  | RW     | Reserved_for_ecc_emmc_crg_in | 保留。                                                                                                                                  | 0x0   |

## PERI\_CRG141

peri\_crg141 为 SDIO1 DRV DLL 控制寄存器。

Offset Address:0x234 Total Reset Value:0x10014000

| Bits    | Access | Name                    | Description                                                                                                                                                                                                                                     | Reset |
|---------|--------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:29] | RW     | reserved                | 保留。                                                                                                                                                                                                                                             | 0x0   |
| [28:24] | RW     | sdio1_drv_clk_phase_sel | SDIO1 DRV 时钟相位配置。默认 180C<br>0x00: 0C;<br>0x01: 11.25C;<br>0x02: 22.5C;<br>...<br>0x1E: 337.5C;<br>0x1F: 348.75C。                                                                                                                                | 0x10  |
| [23:21] | RW     | reserved                | 保留。                                                                                                                                                                                                                                             | 0x0   |
| [20:17] | RW     | sdio1_drv_dll_tune      | SDIO1 SAM DLL 输入时钟相位控制:<br>0x0: 不校准;<br>0x1: 增加 1 级 delay;<br>0x2: 增加 2 级 delay;<br>0x3: 增加 3 级 delay;<br>...<br>0x7: 增加 7 级 delay;<br>0x8: 不校准;<br>0x9: 减少 1 级 delay;<br>0xA: 减少 2 级 delay;<br>0xB: 减少 3 级 delay;<br>...<br>0xF: 减少 7 级 delay; | 0x0   |
| [16]    | RW     | sdio1_drv_dll_slave_en  | SDIO1 drv DLL slave_en 信号。<br>0: slave 停止工作;<br>1: slave 开始工作。                                                                                                                                                                                  | 0x1   |
| [15]    | RW     | sdio1_drv_dll_bypass    | SDIO1 SAM DLL bypass 信号。<br>0: 正常模式, 输出时钟相对输入时钟移相, 具体移相度数由 SDIO 控制器相关寄存器配置°;                                                                                                                                                                    | 0x0   |

|        |    |                    |                                                                              |      |
|--------|----|--------------------|------------------------------------------------------------------------------|------|
|        |    |                    | 1: slave line bypass, 输出时钟不移相。                                               |      |
| [14]   | RW | sdio1_drv_dll_stop | SDIO1 SAM DLL stop 信号。<br>0: 使能时钟检测;<br>1: 禁止时钟检测。                           | 0x1  |
| [13]   | RW | sdio1_drv_dll_mode | SDIO1 SAM DLL mode 信号。<br>0: 正常模式;<br>1: 由 SDIO1_drv_dll_ssel 控制 SLAVE LINE。 | 0x0  |
| [12:8] | RW | reserved           | 保留                                                                           | 0x00 |
| [7:0]  | RW | sdio1_drv_dll_ssel | SDIO1 SAM DLL ssel 信号。DLL SLAVE LINE delay 初始级数选择。                           | 0x00 |

## PERI\_CRG142

peri\_crg142 为 SDIO1 SAM DLL 状态寄存器。

Offset Address:0x238 Total Reset Value:0x00000000

| Bits   | Access | Name                | Description                                                               | Reset      |
|--------|--------|---------------------|---------------------------------------------------------------------------|------------|
| [31:1] | RO     | reserved            | 保留。                                                                       | 0x00000000 |
| [0]    | RO     | sdio1_sam_dll_ready | SDIO1 SAM DLL ready 信号。<br>0: DLL SLAVE not ready;<br>1: DLL SLAVE ready。 | 0x0        |

## PERI\_CRG143

peri\_crg143 为 SDIO1 DRV DLL 状态寄存器。

Offset Address:0x23C Total Reset Value:0x00000000

| Bits    | Access | Name                          | Description                                                              | Reset |
|---------|--------|-------------------------------|--------------------------------------------------------------------------|-------|
| [31:20] | RO     | reserved                      | 保留。                                                                      | 0x000 |
| [19:16] | RO     | reserve_for_eco_sdio1_crg_out | 保留。                                                                      | 0x0   |
| [15]    | RO     | sdio1_drv_dll_lock            | SDIO1 drv DLL lock 信号。<br>0: DLL MASTER unlock;<br>1: DLL MASTER locked。 | 0x0   |

|        |    |                        |                                                                                    |      |
|--------|----|------------------------|------------------------------------------------------------------------------------|------|
| [14]   | RO | sdio1_drv_dll_ready    | SDIO1 drv DLL ready 信号。<br>0: DLL SLAVE not ready;<br>1: DLL SLAVE ready。          | 0x0  |
| [13]   | RO | sdio1_drv_dll_overflow | SDIO0 drv DLL overflow 信号。<br>0: DLL MASTER unoverflow;<br>1: DLL MASTER overflow。 | 0x0  |
| [12:8] | RW | reserved               | 保留。                                                                                | 0x00 |
| [7:0]  | RO | sdio1_drv_dll_mdly_tap | SDIO0 drv DLL mdly_tap 信号。<br>DLL MASTER LINE tap 值。                               | 0x00 |

## PERI\_CRG144

peri\_crg144 为 SDIO1 时钟复位及 SAM DLL 控制寄存器。  
Offset Address:0x240 Total Reset Value:0x01010000

| Bits    | Access | Name                  | Description                                                                                            | Reset |
|---------|--------|-----------------------|--------------------------------------------------------------------------------------------------------|-------|
| [31:30] | RW     | reserved              | 保留。                                                                                                    | 0x0   |
| [29]    | RW     | sdio1_drv_dll_rst_req | SDIO1 DRV DLL 软复位请求。<br>0: 不复位;<br>1: 复位。                                                              | 0x0   |
| [28]    | RW     | sdio1_cken            | SDIO1 时钟门控配置。<br>0: 关闭;<br>1: 打开。                                                                      | 0x0   |
| [27]    | RW     | sdio1_srst_req        | SDIO1 的软复位请求。<br>0: 不复位;<br>1: 复位。                                                                     | 0x0   |
| [26:24] | RW     | sdio1_cksel           | SDIO1 时钟选择。4X 时钟，内部还会 4 分频得到最高 50MHz。<br>000: 400KHz;<br>001: 1600KHz;<br>010: 100MHz;<br>011: 200MHz; | 0x1   |
| [23]    | RW     | sdio1_tx_srst_req     | sdio1 TX 方向软复位请求。<br>0: 不复位;                                                                           | 0x0   |

|         |    |                        |                                                                                                                                                                                                                                                 |      |
|---------|----|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
|         |    |                        | 1: 复位。                                                                                                                                                                                                                                          |      |
| [22]    | RW | sdio1_rx_srst_req      | sdio1 RX 方向软复位请求。<br>0: 不复位;<br>1: 复位。                                                                                                                                                                                                          | 0x0  |
| [21]    | RW | reserved               | 保留。                                                                                                                                                                                                                                             | 0x0  |
| [20:17] | RW | sdio1_sam_dll_tune     | SDIO1 SAM DLL 输入时钟相位控制:<br>0x0: 不校准;<br>0x1: 增加 1 级 delay;<br>0x2: 增加 2 级 delay;<br>0x3: 增加 3 级 delay;<br>...<br>0x7: 增加 7 级 delay;<br>0x8: 不校准;<br>0x9: 减少 1 级 delay;<br>0xA: 减少 2 级 delay;<br>0xB: 减少 3 级 delay;<br>...<br>0xF: 减少 7 级 delay; | 0x0  |
| [16]    | RW | sdio1_sam_dll_slave_en | SDIO1 samb DLL slave_en 信号。<br>0: slave 停止工作;<br>1: slave 开始工作。                                                                                                                                                                                 | 0x1  |
| [15:14] | RW | reserved               | 保留。                                                                                                                                                                                                                                             | 0x0  |
| [13]    | RW | sdio1_sam_dll_mode     | SDIO1 SAM DLL mode 信号。<br>0: 正常模式;<br>1: 由 SDIO1_sam_dll_ssel 控制 SLAVE LINE。                                                                                                                                                                    | 0x0  |
| [12:8]  | RW | reserved               | 保留。                                                                                                                                                                                                                                             | 0x00 |
| [7:0]   | RW | sdio1_sam_dll_ssel     | SDIO1 SAM DLL ssel 信号。DLL SLAVE LINE delay 初始级数选择。                                                                                                                                                                                              | 0x00 |

## PERI\_CRG146

peri\_crg146 为 eMMC 保留寄存器。

Offset Address:0x248 Total Reset Value:0x00000000



| Bits   | Access | Name                            | Description | Reset      |
|--------|--------|---------------------------------|-------------|------------|
| [31:4] | RW     | reserved                        | 保留。         | 0x00000000 |
| [3:0]  | RW     | reserve_for_eco_e<br>mmc_crg_in | 保留。         | 0x0        |

## PERI\_CRG147

peri\_crg147 为 SDIO0 保留寄存器。

Offset Address:0x24c Total Reset Value:0x00000000

| Bits   | Access | Name                             | Description | Reset      |
|--------|--------|----------------------------------|-------------|------------|
| [31:4] | RW     | reserved                         | 保留。         | 0x00000000 |
| [3:0]  | RW     | reserve_for_eco_s<br>dio0_crg_in | 保留。         | 0x0        |

## PERI\_CRG148

peri\_crg148 为 SDIO1 保留寄存器。

Offset Address:0x250 Total Reset Value:0x00000000

| Bits   | Access | Name                             | Description | Reset      |
|--------|--------|----------------------------------|-------------|------------|
| [31:4] | RW     | reserved                         | 保留。         | 0x00000000 |
| [3:0]  | RW     | reserve_for_eco_s<br>dio1_crg_in | 保留。         | 0x0        |

## PERI\_CRG292

peri\_crg292 为输入分辨率时钟频率限制寄存器。

Offset Address:0x490 Total Reset Value:0x00000000

| Bits    | Access | Name         | Description                                                                                | Reset      |
|---------|--------|--------------|--------------------------------------------------------------------------------------------|------------|
| [31:22] | W1S    | reserved     | 保留。                                                                                        | 0x00000000 |
| [21:20] | W1S    | vpu_max_freq | vpu 最高频率限制寄存器<br>0x0: 不限制, 最高 525<br>0x1: 限制最高频率 396<br>0x2: 限制最高频率 297<br>0x3: 限制最高频率 263 | 0x0        |
| [19:17] | W1S    | vgs_max_freq | vgs 最高频率限制寄存器<br>0x0: 不限制, 最高 396                                                          | 0x0        |

|         |     |                 |                                                                                                                                                                      |     |
|---------|-----|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
|         |     |                 | 0x1: 限制最高频率 350<br>0x2: 限制最高频率 297<br>0x3: 限制最高频率 263<br>0x4: 限制最高频率 198<br>0x5: 限制最高频率 150<br>0x6: 限制最高频率 99<br>其它: 限制最高频率 99                                       |     |
| [16:15] | RW  | reserved        | 保留。                                                                                                                                                                  | 0x0 |
| [14:12] | W1S | vpss_max_freq   | vpss 最高频率限制寄存器<br>0x0: 不限制, 最高 396<br>0x1: 限制最高频率 350<br>0x2: 限制最高频率 297<br>0x3: 限制最高频率 263<br>0x4: 限制最高频率 198<br>0x5: 限制最高频率 150<br>0x6: 限制最高频率 99<br>其它: 限制最高频率 99 | 0x0 |
| [11:9]  | W1S | viproc_max_freq | vproc 最高频率限制寄存器<br>0x0: 不限制, 最高 297<br>0x1: 限制最高频率 263<br>0x2: 限制最高频率 198<br>0x3: 限制最高频率 150<br>0x4: 限制最高频率 99<br>其它: 限制最高频率 99                                      | 0x0 |
| [8:6]   | W1S | vicap_max_freq  | vicap 最高频率限制寄存器<br>0x0: 不限制, 最高 496<br>0x1: 限制最高频率 396<br>0x2: 限制最高频率 350<br>0x3: 限制最高频率 297<br>0x4: 限制最高频率 263<br>0x5: 限制最高频率 198<br>0x6: 限制最高频率                    | 0x0 |

|       |     |               |                                                                                                                                                         |     |
|-------|-----|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
|       |     |               | 148.5<br>其它：限制最高频率<br>148.5                                                                                                                             |     |
| [5:3] | RWC | reserved      | 保留                                                                                                                                                      | 0x0 |
| [2:0] | W1S | mipi_max_freq | mipi rx ctrl 最高频率限制<br>寄存器<br>0x0：不限制，最高 496<br>0x1：限制最高频率 396<br>0x2：限制最高频率 350<br>0x3：限制最高频率 297<br>0x4：限制最高频率 263<br>0x5：限制最高频率 198<br>其它：限制最高频率 198 | 0x0 |

## PERI\_CRG293

peri\_crg293 为 mipi rx 时钟使能限制寄存器

Offset Address:0x494 Total Reset Value:0x00000007

| Bits   | Access | Name                      | Description                                          | Reset     |
|--------|--------|---------------------------|------------------------------------------------------|-----------|
| [31:3] | W0C    | reserved                  | 保留。                                                  | 0x0000000 |
| [2]    | W0C    | mipi_rx0_4line_cken_limit | mipi rx0 4line ctrl 时钟使能限制寄存器<br>0x0：限制使用<br>0x1：不限制 | 0x1       |
| [1]    | W0C    | mipi_rx1_cken_limit       | mipi rx2 ctrl 时钟使能限制寄存器<br>0x0：限制使用<br>0x1：不限制       | 0x1       |
| [0]    | W0C    | mipi_rx0_cken_limit       | mipi rx0 ctrl 时钟使能限制寄存器<br>0x0：限制使用<br>0x1：不限制       | 0x1       |

## PERI\_CRG294

peri\_crg294 为处理器性能限制寄存器

Offset Address:0x498 Total Reset Value:0x00010040

| Bits    | Access | Name               | Description                                                                                                                                                               | Reset      |
|---------|--------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| [31:16] | RW     | reserved           | 保留。                                                                                                                                                                       | 0x00000000 |
| [17]    | W1S    | mcu_rst_limit      | mcu 复位限制寄存器<br>0x0: 不限制<br>0x1: mcu core, bus, dgb 被复位                                                                                                                    | 0x0        |
| [16]    | W0C    | mcu_cken_limit     | mcu 时钟使能限制寄存器<br>0x0: 限制使用 bus, core, dbg 时钟<br>0x1: 不限制                                                                                                                  | 0x1        |
| [15:8]  | RW     | reserved           | 保留。                                                                                                                                                                       | 0x0        |
| [7]     | W1S    | a7_core1_rst_limit | a7 core1 复位限制寄存器<br>0x0: 不限制<br>0x1: a7 core1 被复位                                                                                                                         | 0x0        |
| [5:3]   | W1S    | a7_max_freq        | a7 core 最高频率限制寄存器<br>0x0:不限制, 最高 1188<br>0x1: 限制最高频率 993<br>0x2: 限制最高频率 900<br>0x3: 限制最高频率 700<br>0x4: 限制最高频率 594<br>0x5: 限制最高频率 396<br>0x6: 限制最高频率 350<br>其他: 限制最高频率 350 | 0x0        |
| [2:0]   | RW     | reserved           | 保留。                                                                                                                                                                       | 0x0        |

## PERI\_CRG295

peri\_crg295 为 NPU 性能限制寄存器

Offset Address:0x49c Total Reset Value:0x00000000

| Bits | Access | Name | Description | Reset |
|------|--------|------|-------------|-------|
|------|--------|------|-------------|-------|

|        |     |              |                                                                                                                                                    |           |
|--------|-----|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| [31:3] | RW  | reserved     | 保留。                                                                                                                                                | 0x0000000 |
| [2:0]  | W1S | npu_max_freq | npu 最高频率限制寄存器<br>0x0: 不限制, 最高 525<br>0x1: 限制最高频率 496<br>0x2: 限制最高频率 450<br>0x3: 限制最高频率 396<br>0x4: 限制最高频率 350<br>0x5: 限制最高频率 263<br>其他: 限制最高频率 263 | 0x0       |

## 1.3 中央处理器 CPU

该芯片采用 ARM Cortex-A7 双核处理器，具有以下特点：

- 处理器工作频率 1GHz。
- 集成 NEON（含 FPU 硬件浮点协处理器）。
- L1 Cache 包含：32KB 指令 Cache 和 32KB 数据 Cache。
- 内部集成 128KB L2 Cache
- 支持 MMU（Memory Management Unit）。
- 处理器内部集成中断控制器 GIC（Generic Interrupt Controller），支持 128 个中断源的处理，其中外部中断 96 个。

支持 JTAG 调试接口。

## 1.4 微型处理器 MCU

### 1.4.1 概述

MCU（Micro Controller Unit）用于支持 AE/AWB 快速启动。

### 1.4.2 特点

MCU 的功能特点：

- 32 位 RISC-V 处理器，支持 RISC-V RV32IMAC 32/16 位混编指令集。
- 整型 5 级/浮点 7 级，单发射，顺序执行流水线。
- 主频最高可达 400MHz。
- 仅支持小端访问模式。
- 4KB 组相联 I-Cache 和 4KB 组相联 D-Cache。
- 支持 64 个外部中断源，支持矢量和非矢量中断，支持中断嵌套。
- 内存保护单元 PMP，支持自定义配置 8 个可选区域的访问权限。
- 支持可配置的 MCU 访问内存属性定义。
- 支持 5 线通用 JTAG 调试。

### 1.4.3 功能描述

#### 1.4.3.1 应用框图

MCU 实现了 SoC 系统中的 AE/AWB 快速启动应用，过程中需要与 CPU 交互。

生命周期：

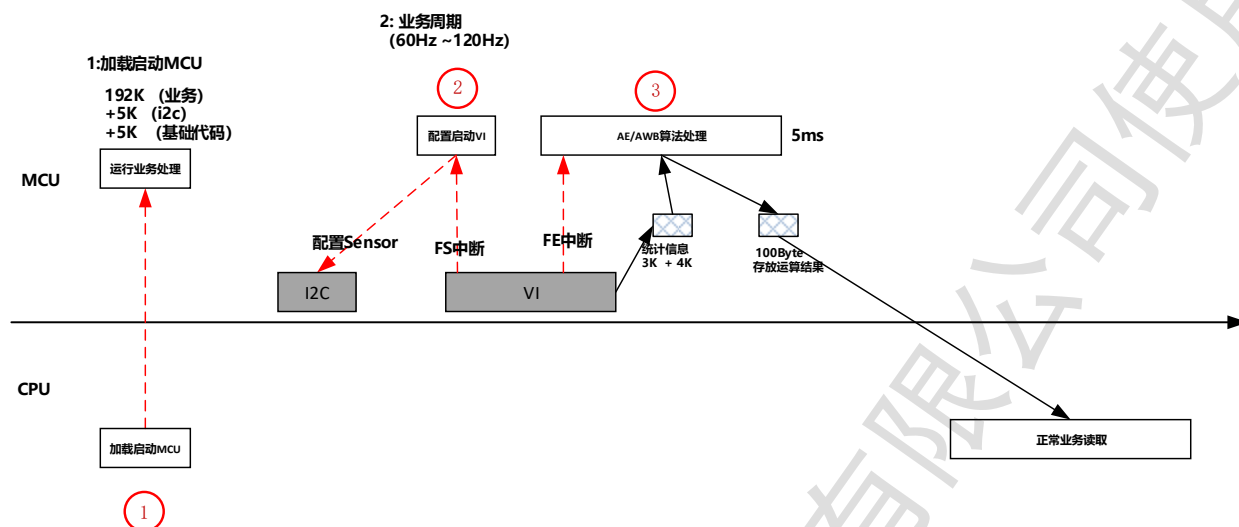
1：快启：在业务启动前可以退出

限制条件：

快启处理过程中，内核有可能未到加载驱动阶段，所以该阶段不能考虑与 CPU 侧有交互。

应用示意图如图 1-4 所示。

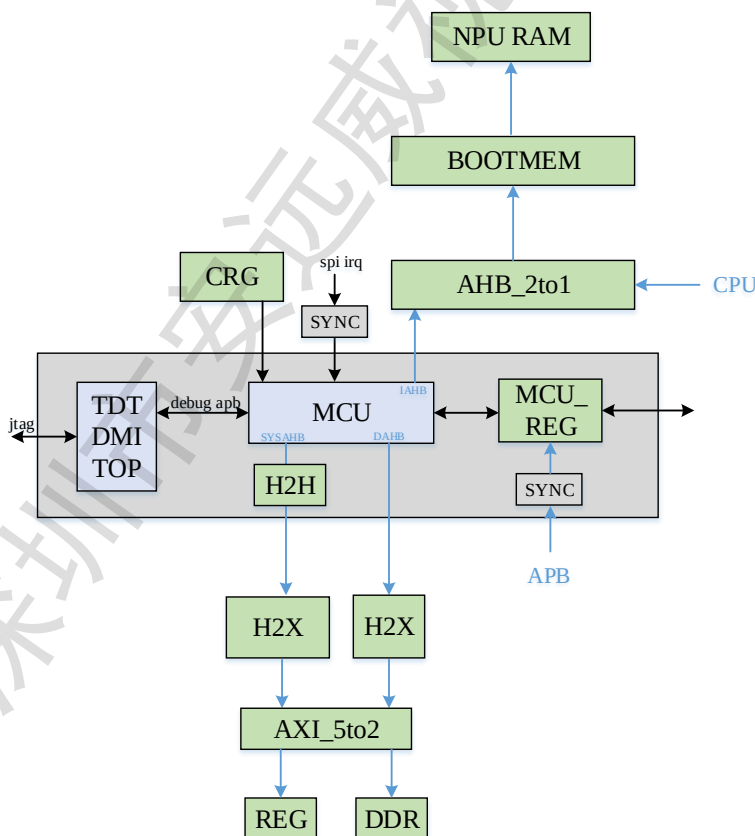
图1-4 MCU 应用快速 AE 示意图



### 1.4.3.2 功能原理

MCU 通过对 BOOTMEM 访问、与 CPU 中断和 message 交互、对 DDR 访问、对系统模块寄存器访问，实现目标应用场景。

图1-5 MCU 子系统架构图



#### 1) CPU 和 MCU 共享 NPU RAM:

NPU 内部给 ext\_sram 共分配 256KB。

可配置 NPU 内部寄存器，切换 NPU 共享 memory 模式：给 CPU 和 MCU 使用、不给 CPU 和 MCU 使用。

#### 2) MCU-DDR 通路

DDR 空间 0x4000\_0000~0x7fff\_ffff 走 DAHB 接口，支持 single 传输和 burst 传输。

DDR 空间 0x8000\_0000~0xdfff\_ffff/0xf000\_0000~0xffff\_ffff 走 SYSAHB 接口，只支持 single 传输。

MCU 访问 DDR 除了做数据交互外，也可以做 MCU 程序运行使用。

#### 3) SYSAHB 用于访问系统寄存器，除了 IAHB 和 DAHB 分配的地址以及 MCU 内部中断寄存器，都通过 SYSAHB 访问，仅支持 single 访问。

#### 4) MCU 的一些状态寄存器，和 CPU 的中断和 mailbox 交互，需要 CPU 给的控制信号，做成一组 MCU 寄存器，CPU 和 MCU 都可访问。

## 流水线结构

MCU 采用 5 级流水线结构：取指、译码、执行、内存访问、写回。

取指阶段，访问指令 Cache 或者指令总线，获取指令，同时访问 BTB，发起 0 延时跳转。

译码阶段，访问动态分支预测器和返回栈，发起分支的预测跳转，同时进行指令译码，读取寄存器堆，处理数据相关性和数据前馈。

执行阶段，完成单周期整型计算指令和多周期乘除法指令的执行、存储/加载指令地址计算和跳转指令处理。其中，整型计算包括普通的算术指令和逻辑指令。

内存访问阶段，利用执行阶段产生的存储/载入指令的目标地址访问数据 Cache 或者数据总线。

写回阶段，将指令执行结果写回寄存器堆。

## 指令集

MCU 实现的 RV32IMAFIC 指令集，包括标准的整型指令集 (RV32I)，RV 乘除法指



令集 (RV32M), RV 原子指令集 (RV32A), RV 单精度浮点指令集 (32F), RV 压缩指令 (RVC)。

指令类型如下:

- 加减法指令
- 逻辑操作指令
- 移位指令
- 比较指令
- 数据传输指令
- 分支跳转指令
- 内存存取指令
- 控制寄存器操作指令
- 低功耗指令
- 异常返回指令
- 特殊功能指令加减法指令

## 低功耗管理

MCU 支持深睡眠和浅睡眠两种睡眠模式, 模式状态在 MCU 的系统寄存器中。默认应用对两种睡眠模式的低功耗处理相同, 用户可根据 MCU 的个性化应用实现不同的低功耗策略。

WFI 指令用于使处理器从正常工作模式转入低功耗模式, 降低功耗水平。在低功耗模式下, MCU 的内部门控时钟管理单元会将绝大多数的寄存器时钟关闭, 而跟处理器唤醒功能相关的逻辑部分的时钟不会被关闭。WFI 指令只有 MCU 处于机器模式下可以被正常执行, 用户模式下执行该指令会触发非法指令异常。

低功耗唤醒支持如下请求类型:

- 调试请求;
- NMI 请求;
- 中断请求;
- 外部事件。

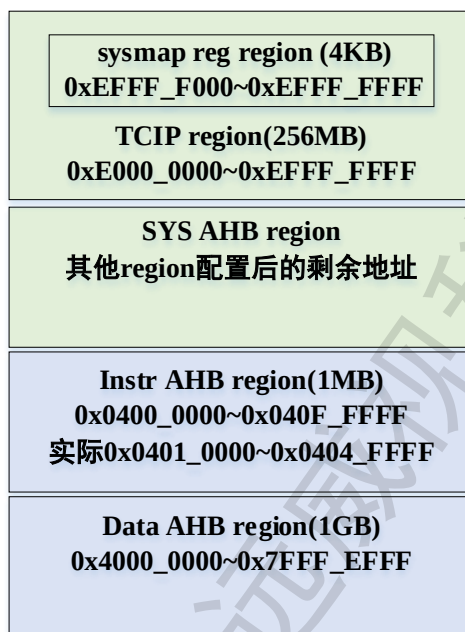
MEXSTATUS 寄存器的 WFE 域来指示唤醒 MCU 的请求类型，当该域值为 1 时，上述所有请求都可以唤醒处于低功耗模式的 MCU，当该域值为 0 时，除了外部事件，其他请求均可以唤醒 MCU。

在 NMI 请求唤醒 MCU 后，MCU 会去响应 NMI 请求进而处理该请求。

在调试请求唤醒 MCU 后，MCU 会进入调试模式。

## 地址映射

图1-6 MCU 访问地址区间分配图



MCU 内置 MATRIX 将不同地址的访问引到不同的通路 TCIP, IAHB, DAHB 和 SYSAHB。

TCIP 是 MCU 内部寄存器（中断、SYSMAP 相关），对应地址映射 0xe000\_0000~0xefff\_fff。

IAHB 访问存放 MCU 应用程序的 NPU 共享 RAM，以及和 CPU 的部分数据交互区，对应地址映射 0x0401\_0000~0x0404\_fff。

DAHB 访问 DDDR 的 1GB 空间，对应地址映射 0x4000\_0000~0x7fff\_fff。

SYSAHB 访问系统地址映射的其他空间，对应地址映射 0x0~0x03ff\_fff, 0x1000\_0000~0x3fff\_fff, 0x8000\_0000~0xdfff\_fff, 0xf000\_0000~0xffff\_fff。

## 内存属性

支持两种内存类型，分别是存储（Memory 类型）和外设（Device 类型）。

Device 类型内存需要配置成 Strong Order 属性，即 MCU 对该片地址空间的访问顺序跟指令序列顺序完全一致，在配置为 Strong Order 后，就不可高缓，对该地址空间的访问不可缓存进指令或者数据 cache。

Memory 类型内存需要配置成 Weak Order 属性。Memory 类型内存支持配置为是否可高缓（cacheable 或者 non-cacheable），对于 cacheable 的内存地址空间，MCU 从该区域访问得到的数据可以缓存在内部的指令或者数据 cache 中。

支持对内存配置为是否可暂存的内存（bufferable 或者 non-bufferable）。而对于 bufferable 的内存地址空间，MCU 在对该地址空间进行写操作时，总线写响应可由总线通路的任一节点返回，而不必等待访问的 slave 上写操作完成后再返回。反之，对于 non-bufferable 的内存地址空间，MCU 对该地址空间的写操作的响应必须在访问的 slave 上写操作完成后由该 slave 返回。

MCU 对 SOC 地址空间的初始划分和属性分配如下表。

表1-5 MCU 初始访问地址区间与属性

| 初始区间                      | 初始属性                                          |
|---------------------------|-----------------------------------------------|
| 0x0 ~ 0x04ff_ffff         | Weak-order / Cacheable / Bufferable           |
| 0x0500_0000 ~ 0x0fff_ffff | Weak -order / Non-cacheable / Non_bufferable  |
| 0x1000_0000 ~ 0x13ff_ffff | Strong-order / Non-cacheable / Non_bufferable |
| 0x1400_0000 ~ 0x14ff_ffff | Weak -order / Non-cacheable / Bufferable      |
| 0x1500_0000 ~ 0x3fff_ffff | Weak -order / Non-cacheable / Non_bufferable  |
| 0x4000_0000 ~ 0x7fff_ffff | Weak -order / Cacheable / Non_bufferable      |
| 0x8000_0000 ~ 0xffff_ffff | Strong-order / Non-cacheable / Non_bufferable |
| 0xf000_0000 ~ 0xffff_ffff | Strong-order / Non-cacheable / Non_bufferable |

MCU 对 SOC 地址空间的划分和属性支持再配置。全空间内存属性的配置支持对 8 个内存地址空间的属性设定，地址区间由 TCIP 寄存器 SYSMAP\_ADDR<sub>i</sub> (i = 0~7) 和 SYSMAP\_FLAG<sub>i</sub> (i = 0~7) 定义。

SYSMAP\_FLAG 用于配置对应内存地址空间的内存属性：strong order/weak order, cacheable/non-cacheable, bufferable/non-bufferable。

SYSMAP\_ADDR(i-1) <= 第 i 个地址空间地址 < SYSMAP\_ADDR<sub>i</sub>。第 0 个地址空间下限是 0x0，内存地址不在配置的 8 个地址区间的地址属性默认为 cacheable 和 bufferable。每个地址空间大小的配置粒度是 4KB。

对于 CLINT 和 CLIC 所在的 TCIP 地址空间，0xE000\_0000 至 0xEFFF\_FFFF，需要设置为 Strong Order 以及 non-cacheable, nonbufferable 属性。

## 物理内存保护单元

PMP 主要保护两类系统资源：存储器和外围设备。PMP 主要负责对存储器和外围设备访问的合法性进行检查，判定当前工作模式下 MCU 是否具备对内存地址的读/写/执行访问权限。

PMP 单元支持 8 个表项可配置，在机器模式下 MCU 可以对区域的访问权限进行设置。每个表项通过编号 0-7 来标识和索引。

PMP 单元的主要特征有：

- 支持 8 个 PMP 表项；
- 每个表项地址划分最小粒度为 128 字节；
- 支持 OFF、TOR (Top of Range)、NAPOT (Naturally Aligned Power-of-Two Region) 三种地址匹配模式，不支持 NA4 (Naturally Aligned 4-Byte Region) 匹配模式；
- 支持可读、可写、可执行三种权限的配置；
- 支持表项 Lock 功能。

## 异常处理

在 MCAUSE 寄存器的 exception\_code 字段观测 MCU 异常状态如下表：

表1-6 异常类型定义表

| 异常类型        | 异常向量号 | 异常优先级              |
|-------------|-------|--------------------|
| 调试断点异常      | 3     | 最高<br>>>>>>><br>最低 |
| 取指令访问错误异常   | 1     |                    |
| 非法指令异常      | 2     |                    |
| 用户模式环境调用异常  | 8     |                    |
| 机器模式环境调用异常  | 11    |                    |
| 存储指令非对齐访问异常 | 6     |                    |
| 加载指令非对齐访问异常 | 4     |                    |
| 存储指令访问错误异常  | 7     |                    |
| 加载指令访问错误异常  | 5     |                    |

异常按以下步骤被处理，所有步骤在一个处理器时钟周期完成：

1. 处理器保存 PC 到异常保留程序计数器（MEPC）中。
2. 将 MCAUSE 中的异常向量号 Exception Code 域更新为当前发生的异常向量号，标识异常类别。
3. 将 MSTATUS 中的中断使能位 MIE 保存到 MPIE 中。
4. 将 MSTATUS 中的中断使能位 MIE 位清零，禁止响应中断。
5. 将 MXSTATUS 中的 PM 域保存到 MSTATUS 寄存器的 MPP 域，并将 PM 设置为机器模式，即异常响应后 MCU 进入机器模式。
6. 将产生异常事件的原因更新到 MTVAL 寄存器。
7. 处理器根据向量基址寄存器 MTVEC 中的基址得到异常服务程序入口地址进行跳转执行。

异常服务程序的返回和现场恢复需要通过执行 MRET 指令实现：

1. 将 PC 恢复成 MEPC 寄存器的值，精确异常模式下可以保证 MCU 从异常服务程序返回后可以从触发异常的地方重新执行指令。这就要求上节的异常服务程序中将触发该异常的事件进行修复，避免再次触发异常。

2. MSTATUS.MIE 被恢复成 MSTATUS.MPIE 的值，MPIE 被设置为 1。
3. MXSTATUS.PM 域被恢复成 MSTATUS.MPP 的值，MPP 被设置为 2'b00。

## 中断处理

MCU 支持机器模式软件中断和 64 个外部中断，

表1-7 中断向量表

| 向量号   | 中断类型      |
|-------|-----------|
| 0-2   | 未实现/保留    |
| 3     | 机器模式软件中断  |
| 4-15  | 未实现/保留    |
| 16-79 | 中断控制器外接中断 |

内置中断控制器，寄存器在 TCIP 地址空间中。寄存器定义如下表：

表1-8 中断寄存器概述表

| 寄存器名    | 描述                                                                       |
|---------|--------------------------------------------------------------------------|
| 中断阈值寄存器 | 处于等待状态的中断请求的优先级必须高于寄存器定义的阈值才能向处理器发起中断。                                   |
| 中断等待寄存器 | 寄存器最低位被置起表示对应的中断源有中断等待被处理。（每根外部中断对应一个寄存器）                                |
| 中断使能寄存器 | 寄存器最低位被置起表示对应的中断源被使能。<br>（每根外部中断对应一个寄存器）                                 |
| 中断属性寄存器 | 用于配置不同的中断源的属性，包括了可响应中断的 CPU 特权态、中断的触发模式以及中断是否为硬件矢量模式。<br>（每根外部中断对应一个寄存器） |
| 中断控制寄存器 | 用于表示每一个中断源参与仲裁的优先级。<br>（每根外部中断对应一个寄存器）                                   |

中断流程：

1. 中断请求：系统模块产生中断（电平或脉冲），信号传给 MCU 的中断输入。
2. 中断响应：进入中断等待状态，对应的 pending 寄存器拉起。电平中断源仍保持；同一脉冲中断源再次触发时，保持第一次未处理等待中的中断。

3. 中断仲裁：根据设置的优先级或默认优先级，仲裁决定当前等待状态中的中断，哪个进行下一个中断处理。
4. 中断处理：进入对应的中断堆栈入口地址（非矢量为 MTVE 寄存器值，矢量中断为 MTVT 寄存器值+中断号偏移），执行中断处理程序。
5. 中断返回：中断处理完成后，对于电平中断，直接在中断源头清除中断请求；对于脉冲中断，清除 pending 状态（矢量中断自动清除出 pending 状态）。同时 pc 指针弹出中断堆栈，此时可能发生中断咬尾（弹出堆栈之间查询其他 pending，如果存在优先级高于 MCAUSE 的中断，直接跳转到新中断的入口地址）。

## 调试

采用标准 5 线 JTAG 调试接口，兼容 RISC-V debug V0.13.2 协议标准。

调试单元的主要特征有：

- 支持同步调试和异步调试；
- 支持软断点，可配置多个硬断点；
- 支持指令单步执行或多步执行；
- 检查和设置 MCU 寄存器的值；
- 支持 MCU 运行时通过调试端口直接访问内存。

### 1.4.4 工作方式

#### 1.4.4.1 软复位位于时钟门控

MCU 在默认上电后不工作，软复位不释放，时钟门控关断。

当需要执行 MCU 应用时，CPU 或 DMA 提前将对应的 MCU 应用程序装载到 MCU 的指令存储基地址。装载完毕后打开 MCU 时钟门控，并释放 MCU 软复位。MCU 开始工作。

#### 1.4.4.2 MCU 配置流程

MCU 步骤如下：

步骤 1 启动时，将 MCU firmware 从 flash 中装载到 NPU 共享 RAM 区域或者 DDR 区域。

步骤 2 CPU 配置 MCU 启动地址寄存器 MCU\_RST\_ADDR 为 NPU 共享 RAM 区域或者 DDR 区域。

- 步骤 3 如果 MCU 启动地址为 NPU 共享 RAM 区域，CPU 检查 NPU 共享 RAM 状态，如果未开放，配置 NPU 寄存器 DMA\_CONV\_PRIORITY\_NUM 开放给 CPU 和 MCU。
- 步骤 4 CPU 配置 CRG 寄存器 PERI\_CRG93 打开 MCU 的时钟门控和软复位，启动 MCU。
- 步骤 5 MCU 初始化通用寄存器数值。
- 步骤 6 MCU 配置中断, Cache, PMP, SYSMAP。

----结束

完成以上步骤以后，MCU 就可以正常工作。

### 1.4.5 寄存器概览

MCU 在用户模式下的 32 个 32 位通用寄存器。

表1-9 通用寄存器表单

- 仅 MCU 可访问。
- 通过整形运算指令访问。

| 名称      | 描述         |
|---------|------------|
| x0      | 零值         |
| x1      | 返回地址       |
| x2      | 堆栈指针       |
| x3      | 全局指针       |
| x4      | 线程指针       |
| x5      | 临时/备用链接寄存器 |
| x6~x7   | 临时寄存器      |
| x8      | 保留寄存器/帧指针  |
| x9      | 保留寄存器      |
| x10~x11 | 函数参数/返回值   |
| x12~x17 | 函数参数       |
| x18~x27 | 保留寄存器      |
| x28~x31 | 临时寄存器      |

MCU 在用户模式下的 32 个 32 位浮点寄存器。



表1-10 浮点寄存器表单

- 仅 MCU 可访问。
- 通过浮点运算指令访问。

| 名称      | 描述          |
|---------|-------------|
| f0~f7   | 浮点临时寄存器     |
| f8~f9   | 浮点保留寄存器     |
| f10~f11 | 浮点参数/返回值寄存器 |
| f12~f17 | 浮点参数寄存器     |
| f18~f27 | 浮点保留寄存器     |
| f28~f31 | 浮点临时寄存器     |

表1-11 中断寄存器表单

- 仅 MCU 可访问，对于 MCU 基地址是 0XE000\_0000。

| 偏移地址           | 名称             | 描述                                    |
|----------------|----------------|---------------------------------------|
| 0x0            | MSIP           | 机器模式软件中断配置寄存器：<br>高位硬件固定为 0，bit[0]有效。 |
| 0x80_0000      | CLICCFG        | 中断控制器配置寄存器。                           |
| 0x80_0008      | MINTTHRESH     | 中断阈值寄存器。                              |
| 0x080_1000+4*i | CLICINTIP[i]   | 中断源 i 等待寄存器。                          |
| 0x080_1001+4*i | CLICINTIE[i]   | 中断源 i 使能寄存器。                          |
| 0x080_1002+4*i | CLICINTATTR[i] | 中断源 i 属性寄存器。                          |
| 0x080_1003+4*i | CLICINTCTL[i]  | 中断源 i 控制寄存器。                          |

表1-12 机器模式寄存器表单

- 仅 MCU 可访问。
- 通过 CP 寄存器读写指令访问。

| 名称           | 描述           |
|--------------|--------------|
| 机器模式异常设置寄存器组 |              |
| MSTATUS      | 机器模式处理器状态寄存器 |

| 名称           | 描述                   |
|--------------|----------------------|
| MISA         | 机器模式处理器指令集信息寄存器      |
| MIE          | 机器模式中断使能控制寄存器        |
| MTVEC        | 机器模式异常向量基址寄存器        |
| MTVT         | 机器模式矢量中断基址寄存器        |
| 机器模式异常处理寄存器组 |                      |
| MSCRATCH     | 机器模式数据备份寄存器          |
| MSCRATCHCSW  | 机器模式多模式数据备份寄存器       |
| MSCRATCHCSWL | 机器模式中断数据备份寄存器        |
| MCLICBASE    | 机器模式中断控制器基址寄存器       |
| MEPC         | 机器模式异常程序计数器          |
| MEPC         | 机器模式异常程序计数器          |
| MCAUSE       | 机器模式异常向量寄存器          |
| MNXTI        | 机器模式等待中断向量地址和中断使能寄存器 |
| MINTSTATUS   | 机器模式中断状态寄存器          |
| MTVAL        | 机器模式异常原因寄存器          |
| MIP          | 机器模式中断等待寄存器          |
| MCYCLE       | 机器模式周期计数器            |
| MINSTRET     | 机器模式退休指令计数器          |
| 机器模式内存保护寄存器组 |                      |
| PMPCFG0      | 机器模式物理内存保护配置寄存器 0    |
| PMPCFG1      | 机器模式物理内存保护配置寄存器 1    |
| PMPADDR0     | 机器模式物理内存地址寄存器 0      |
| PMPADDR1     | 机器模式物理内存地址寄存器 1      |

| 名称         | 描述              |
|------------|-----------------|
| PMPADDR2   | 机器模式物理内存地址寄存器 2 |
| PMPADDR3   | 机器模式物理内存地址寄存器 3 |
| PMPADDR4   | 机器模式物理内存地址寄存器 4 |
| PMPADDR5   | 机器模式物理内存地址寄存器 5 |
| PMPADDR6   | 机器模式物理内存地址寄存器 6 |
| PMPADDR7   | 机器模式物理内存地址寄存器 7 |
| 机器模式扩展寄存器组 |                 |
| MXSTATUS   | 扩展状态寄存器         |
| MHCR       | 硬件配置寄存器         |
| MHINT      | 隐式操作寄存器         |
| MEXSTATUS  | 扩展异常状态寄存器       |
| MRADDR     | 复位地址指示寄存器       |
| MNMICAUSE  | NMI 状态寄存器       |
| MNMICAUSE  | NMI 状态寄存器       |
| MNMIPC     | NMI 异常程序计数器     |

表1-13 用户模式寄存器表单

- 仅 MCU 可访问。
- 通过 CP 寄存器读写指令访问。

| 名称     | 描述            |
|--------|---------------|
| FFLAGS | 浮点异常累积状态寄存器   |
| FRM    | 浮点动态舍入模式寄存器   |
| FCSR   | 浮点控制状态寄存器     |
| FXCR   | 用户模式浮点扩展控制寄存器 |

## 1.4.6 寄存器描述

### 1.4.6.1 中断寄存器

#### MSIP

机器模式软件中断寄存器。

Offset Address: 0x0

| Bits | Access | Name | Description                                                                          | Reset |
|------|--------|------|--------------------------------------------------------------------------------------|-------|
| [0]  | RW     | MSIP | 机器模式软件中断的中断状态<br>当 MSIP 位置 1, 当前存在有效的机器模式软件中断请求。<br>当 MSIP 位置 0, 当前不存在有效的机器模式软件中断请求。 | 0x0   |

#### CLICCFG

中断控制器配置寄存器。

Offset Address: 0x800000

| Bits  | Access | Name   | Description       | Reset |
|-------|--------|--------|-------------------|-------|
| [6:5] | RW     | NMBITS | 在机器模式响应中断。值恒为 0。  | 0x0   |
| [4:1] | RW     | NLBITS | 中断优先级的位数。值恒为 3。   | 0x3   |
| [0]   | RW     | NVBITS | 支持硬件矢量模式中断。值恒为 1。 | 0x1   |

#### MINTTHRESH

中断阈值寄存器。

Offset Address: 0x800008

| Bits    | Access | Name | Description                                           | Reset |
|---------|--------|------|-------------------------------------------------------|-------|
| [31:24] | RW     | MTH  | 机器模式中断的阈值。处于等待状态的中断请求的优先级必须高于寄存器定义的阈值 MTH 才能向处理器发起中断。 | 0x0   |

## CLICINTIP

中断等待寄存器。

Offset Address: 0x801000+4\*i (i=0~127)

| Bits | Access | Name | Description                                                                                                                                                                                                                                            | Reset |
|------|--------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [0]  | RO/RW  | IP   | 中断源是否有中断等待响应。<br><br>电平中断模式下，CLICINTIP 为只读寄存器。外部中断源为高则 IP 为 1，外部中断源为低则 IP 为 0。<br><br>边缘中断模式下，CLICINTIP 为可读可写寄存器，且带有自动清除 IP 位的功能。当中断配置为硬件矢量模式时，响应中断的同时会自动清除该中断的 IP 位；对于非硬件矢量模式的中断，通过执行一条 CSR 访问指令同时产生对 MNXTI 寄存器有效的读写操作来获取中断等待状态，同时该操作可以清除在等待响应的对应的中断。 | 0x0   |

## CLICINTIE

中断等待寄存器。

Offset Address: 0x801001+4\*i (i=0~127)

| Bits | Access | Name | Description              | Reset |
|------|--------|------|--------------------------|-------|
| [0]  | RW     | IE   | 1：对应中断被使能；<br>0：对应中断未使能。 | 0x0   |

## CLICINTATTR

中断属性寄存器。

Offset Address: 0x801002+4\*i (i=0~127)

| Bits  | Access | Name | Description       | Reset |
|-------|--------|------|-------------------|-------|
| [7:6] | RW     | MODE | 机器模式中断，固定为 2'b11。 | 0x3   |
| [2:1] | RW     | TRIG | 用于区分脉冲中断和电平中断，    | 0x0   |

|     |    |     |                                                                                                     |     |
|-----|----|-----|-----------------------------------------------------------------------------------------------------|-----|
|     |    |     | 当 trig[0]为 0 时，代表为电平中断。<br>当 trig[0]为 1 且 trig[1]为 0 时，代表上升沿中断；当 trig[0]为 1 且 trig[1]为 1 时，代表下降沿中断。 |     |
| [0] | RW | SHV | 中断是否为硬件矢量中断。                                                                                        | 0x0 |

## CLICINTCTL

中断控制寄存器。

Offset Address: 0x801003+4\*i (i=0~127)

| Bits  | Access | Name    | Description                                           | Reset |
|-------|--------|---------|-------------------------------------------------------|-------|
| [7:5] | RW     | INT_CTL | 对应中断的优先级。<br>可配置：31, 63, 95, 127, 159, 191, 223, 255。 | 0x0   |

### 1.4.6.2 机器模式寄存器

## MSTATUS

机器模式处理器状态寄存器。

| Bits    | Access | Name | Description                                                                                                                                                                   |
|---------|--------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [17]    | RW     | MPRV | 当 MPRV 为 0 时，访存地址的转换和保护判断没有特殊要求；<br>当 MPRV 为 1 时，加载和存储指令（load/store）要根据 MPP 位中存储的特权模式信息进行地址转换和保护判断。                                                                           |
| [14:13] | RW     | FS   | 用于指示浮点控制寄存器以及浮点通用寄存器的状态。<br>2 'b00：无法访问浮点控制寄存器和浮点通用寄存器，访问这类寄存器会触发非法指令异常；<br>2 ' b01：浮点控制寄存器和浮点通用寄存器处于初始化状态；<br>2 ' b10：浮点控制寄存器和浮点通用寄存器是干净的；<br>2' b11：部分浮点控制寄存器或者浮点通用寄存器被写脏过； |

|         |    |      |                                                                                                                           |
|---------|----|------|---------------------------------------------------------------------------------------------------------------------------|
| [12:11] | RW | MPP  | <p>用于保存处理器进入异常服务程序前的特权状态。</p> <p>当 MPP 为 2' b00 时，表示处理器进入异常服务程序前处于用户模式；</p> <p>当 MPP 为 2' b11 时，表示处理器进入异常服务程序前处于机器模式；</p> |
| [7]     | RW | MPiE | <p>用于保存处理器进入异常服务程序前 MIE 位的值。</p> <p>复位值为零，在处理器退出异常服务程序时被置 1。</p>                                                          |
| [3]     | RW | MIE  | <p>当 MIE 为 0 时，中断无效；</p> <p>当 MIE 为 1 时，中断有效。</p> <p>在处理器响应异常时被清零；在处理器退出异常时被置为 MPiE 的值。</p>                               |

## MISA

机器模式处理器指令集信息寄存器。

| Bits   | Access | Name | Description                                                                         |
|--------|--------|------|-------------------------------------------------------------------------------------|
| [31:0] | RO     | MISA | <p>机器模式处理器指令集寄存器（MISA）存储了处理器所支持的指令集架构信息。MCU 支持的指令集架构为 RV32IMAC，对应值为 0x40901105。</p> |

## MIE

机器模式中断使能控制寄存器。

| Bits | Access | Name | Description                                                                                                                                                                         |
|------|--------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [11] | RW     | MEIE | <p>当 MEIE 为 0 时，外部中断无效，处理器在低功耗模式下无法被外部中断唤醒；</p> <p>当 MEIE 为 1 时，外部中断有效，处理器在低功耗模式下可以被外部中断唤醒；</p> <p>当处理器配置 CLIC 模式时，且处理器 MTVEC 寄存器配置为 CLIC 模式下（MTVEC.mode[1] 为 1' b1），该寄存器将会置 0。</p> |
| [3]  | RW     | MSIE | <p>机器模式软件中断使能位。</p> <p>0：机器模式软件中断未使能；</p>                                                                                                                                           |

|  |  |  |                |
|--|--|--|----------------|
|  |  |  | 1: 机器模式软件中断使能。 |
|--|--|--|----------------|

## MTVEC

机器模式异常向量基址寄存器。

| Bits   | Access | Name | Description                                                                                                                                                                                                      |
|--------|--------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [31:2] | RW     | BASE | 向量基址位指示了异常服务程序入口地址的高 30 位，将此基址低位拼接 2' b00 即可得到异常服务程序入口地址。<br>复位值为零。                                                                                                                                              |
| [1:0]  | RO     | MODE | CPU 使用 MTVEC[31:6]<<6 作为异常的服务程序入口地址并跳转执行，硬件矢量中断模式下，CPU 首先使用 MTVT + 4* 中断 ID 为地址，取出中断服务程序入口地址，并跳转到该入口地址执行中断服务程序，非硬件矢量中断模式下 CPU 使用 MTVEC[31:6]<<6 作为中断服务程序入口地址并跳转执行。MTVT 为矢量中断基址寄存器，在 CLIC 配置下存在。<br>固定为 3，软件不可设置。 |

## MTVT

机器模式矢量中断基址寄存器。

| Bits   | Access | Name | Description                                                                                  |
|--------|--------|------|----------------------------------------------------------------------------------------------|
| [31:6] | RW     | BASE | 向量基址位指示了矢量中断向量表基址，处理器通过计算该基址加上每个中断的地址偏移量 (MTVT+4* 中断 ID) 得到矢量中断向量表中每个中断服务程序的入口地址并跳转执行。复位值为零。 |

## MSCRATCH

机器模式数据备份寄存器。

| Bits   | Access | Name     | Description                                  |
|--------|--------|----------|----------------------------------------------|
| [31:0] | RW     | MSCRATCH | 用于处理器在异常服务程序中备份临时数据。一般用来存储机器模式本地上下文空间的入口指针值。 |



## MSCRATCHCSW

机器模式多模式数据备份寄存器。

| Bits   | Access | Name        | Description                                                                           |
|--------|--------|-------------|---------------------------------------------------------------------------------------|
| [31:0] | RW     | MSCRATCHCSW | 用于加速多特权模式下的中断处理。一般用法为，在不同的特权模式转换时，支持 MSCRATCH 在满足进入中断之前的特权模式不为机器模式时，与栈指针交换数值，否则值保持不变。 |

## MSCRATCHCSWL

机器模式中断数据备份寄存器。

| Bits   | Access | Name         | Description                                      |
|--------|--------|--------------|--------------------------------------------------|
| [31:0] | RW     | MSCRATCHCSWL | 用于加速处理器前后两个状态不同时都为中断处理的情况。可用于 MSCRATCH 与栈指针交换数值。 |

## MCLICBASE

机器模式中断控制器基址寄存器。

| Bits   | Access | Name      | Description                                         |
|--------|--------|-----------|-----------------------------------------------------|
| [31:0] | RO     | MCLICBASE | 用于向软件指示 CLIC 内存映射寄存器的地址。<br>固定为 0xe00800000，软件不可改写。 |

## MEPC

机器模式异常程序寄存器。

| Bits   | Access | Name | Description                       |
|--------|--------|------|-----------------------------------|
| [31:0] | RW     | MEPC | 用于存储程序从异常服务程序退出时要返回的程序计数器值（PC 值）。 |

## MCAUSE

机器模式异常向量寄存器。

| Bits | Access | Name      | Description                       |
|------|--------|-----------|-----------------------------------|
| [31] | RW     | Interrupt | 当 Interrupt 位为 0 时，表示触发异常的来源不是中断， |

|         |    |                |                                                                                                                      |
|---------|----|----------------|----------------------------------------------------------------------------------------------------------------------|
|         |    |                | Exception Code 按照普通异常规则解析；<br>当 Interrupt 位为 1 时，表示触发异常的来源是中断，Exception Code 按照中断规则解析。<br>该位复位值为零。                   |
| [30]    | RW | MINHV          | 用于指示处理器是否正在取矢量中断入口地址，在处理器响应矢量中断时，该位会被置高。成功获取矢量中断服务程序入口地址后清 0。<br>该位复位值为零。                                            |
| [29:28] | RW | MPP            | MSTATUS.MPP[1:0]的镜像。                                                                                                 |
| [27]    | RW | MPIE           | MSTATUS.MPIE 的镜像。                                                                                                    |
| [23:16] | RW | MPIL           | 保存处理器进入中断服务程序前的中断优先级，即将 MINTSTATUS.MIL 位。执行 MRET 指令从中断返回时，处理器将 MPIL 位拷贝至 MINTSTATUS 寄存器中的 MIL 位。<br>处理器响应异常时，该位保持不变。 |
| [11:0]  | RW | Exception_Code | 在处理器进入异常时，异常向量号域会被更新为异常来源的向量号。<br>该位复位值为零。                                                                           |

## MNXTI

机器模式等待中断向量地址和中断使能寄存器。

| Bits   | Access | Name  | Description                                                                        |
|--------|--------|-------|------------------------------------------------------------------------------------|
| [31:0] | RW     | MNXTI | 使用该寄存器加速中断响应。在机器模式下，当前处于等待状态中断的优先级高于保留中断优先级 (MCAUSE.mpil) 时，软件通过读该寄存器可以获取下一中断入口地址。 |

## MINTSTATUS

机器模式中断状态寄存器。

| Bits    | Access | Name | Description                                                                                                                                          |
|---------|--------|------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| [31:24] | RO     | MIL  | <p>处理器当前处理的中断的优先级。</p> <p>当处理器响应中断时，该域被更新为当前被响应中断的优先级。当处理器执行 MRET 指令从中断服务程序返回时，处理器将 MCAUSE.mpil 拷贝至该位。</p> <p>当处理器响应异常时，该位不发生改变。</p> <p>该位复位值为零。</p> |

## MTVAL

机器模式异常原因寄存器。

| Bits   | Access | Name  | Description                       |
|--------|--------|-------|-----------------------------------|
| [31:0] | RW     | MTVAL | 用于保存触发异常事件的具体原因，比如访问错误异常的错误访问地址等。 |

## MIP

机器模式中断等待寄存器。

| Bits   | Access | Name | Description                                     |
|--------|--------|------|-------------------------------------------------|
| [31:0] | RO     | MIP  | 用于保存处理器的中断等待状态，该寄存器仅在非 CLIC 式下有意义。寄存器值为零，软件不可写。 |

## PMPCFG0

物理内存保护设置寄存器 0。

| Bits    | Access | Name            | Description                                                                          |
|---------|--------|-----------------|--------------------------------------------------------------------------------------|
| [31:24] | RW     | pmp region3     | 与 pmp region0 分布相同。                                                                  |
| [23:16] | RW     | pmp region2     | 与 pmp region0 分布相同。                                                                  |
| [15:8]  | RW     | pmp region1     | 与 pmp region0 分布相同。                                                                  |
| [7]     | RW     | pmp region0 - L | 当 L 为 0 时，该表项的物理内存保护设置寄存器和地址寄存器都可以修改。机器模式下的访问权限不受 R/W/X 位的限制，用户模式下的访问权限受 R/W/X 位的限制； |

|       |    |                 |                                                                                                                                                                                                                                                                                         |
|-------|----|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |    |                 | <p>当 L 为 1 时，该表项所有内容，包括 L 位，在 CPU 复位前都不可以修改。且机器模式下的访问和用户模式下的访问都将受到 R/W/X 位的限制；当该表项的物理内存保护设置寄存器配置为 TOR 地址匹配模式时，其前一表项的物理内存保护地址寄存器也无法修改。</p> <p>复位值为零。</p>                                                                                                                                 |
| [4:3] | RW | pmp region0 - A | <p>当 A 为 00 时，表项无效；</p> <p>当 A 为 01 时，TOR (Top Of Range) 模式，使用相邻表项的地址作为匹配区间；</p> <p>当 A 为 10 时，NA4 (Naturally Aligned 4-byte) 模式，使用 4 字节大小作为匹配区间；由于 MCU 地址划分最小粒度为 128 字节，不支持 NA4 匹配模式；</p> <p>当 A 为 11 时，NAPOT (Naturally Aligned Power Of Two) 模式，使用 2 的幂次方大小作为匹配空间；</p> <p>复位值为零。</p> |
| [2]   | RW | pmp region0 - X | <p>当 X 为 0 时，该区域为不可执行；</p> <p>当 X 为 1 时，该区域为可执行；</p> <p>复位值为零。</p>                                                                                                                                                                                                                      |
| [1]   | RW | pmp region0 - W | <p>当 W 为 0 时，该区域为不可写；</p> <p>当 W 为 1 时，该区域为可写；</p> <p>复位值为零。</p>                                                                                                                                                                                                                        |
| [0]   | RW | pmp region0 - R | <p>当 R 为 0 时，该区域为不可读；</p> <p>当 R 为 1 时，该区域为可读；</p> <p>复位值为零。</p>                                                                                                                                                                                                                        |

## PMPCFG1

物理内存保护设置寄存器 1。

| Bits    | Access | Name        | Description         |
|---------|--------|-------------|---------------------|
| [31:24] | RW     | pmp region7 | 与 pmp region0 分布相同。 |
| [23:16] | RW     | pmp region6 | 与 pmp region0 分布相同。 |

|        |    |             |                     |
|--------|----|-------------|---------------------|
| [15:8] | RW | pmp region5 | 与 pmp region0 分布相同。 |
| [7:0]  | RW | pmp region4 | 与 pmp region0 分布相同。 |

## PMPADDR0~7

物理内存保护地址寄存器。

| Bits   | Access | Name    | Description         |
|--------|--------|---------|---------------------|
| [31:0] | RW     | PMPADDR | 用于配置物理内存表项的地址和区域大小。 |

## MCYCLE

机器模式周期计数寄存器。

| Bits   | Access | Name   | Description                                                                        |
|--------|--------|--------|------------------------------------------------------------------------------------|
| [31:0] | RO     | MCYCLE | 用于存储处理器已经执行的周期数，当处理器处于执行状态（即非低功耗状态）下，MCYCLE 寄存器就会在每个处理器执行周期自增计数，每个周期加 1。<br>复位值为零。 |

## MINSTRET

机器模式退休指令计数寄存器。

| Bits   | Access | Name     | Description                                                         |
|--------|--------|----------|---------------------------------------------------------------------|
| [31:0] | RO     | MINSTRET | 用于存储处理器已经退休的指令数，MINSTRET 寄存器会在每条指令退休时自增计数，每退休一条指令该寄存器加 1。<br>复位值为零。 |

## MXSTATUS

机器模式扩展状态寄存器。

| Bits    | Access | Name | Description                                   |
|---------|--------|------|-----------------------------------------------|
| [31:30] | RO     | PM   | 当前处理器的所处的特权模式，2' b11 时处理器为机器模式，2' b00 时为用户模式。 |

|      |    |           |                                                                                                                             |
|------|----|-----------|-----------------------------------------------------------------------------------------------------------------------------|
|      |    |           | 该域复位值为 2' b11                                                                                                               |
| [22] | RW | THEADISAE | <p>当 THEADISAE 为 0 时，执行所有 T-Head 自定义扩展指令产生非法指令异常；</p> <p>当 THEADISAE 为 1 时，处理器可以正常执行 T-Head 自定义扩展指令集。</p> <p>复位值为 1' b1</p> |
| [15] | RW | MM        | <p>1' b1: 当内存访问出现地址非对齐时，不会上报非对齐访问异常，硬件会对非对齐访问请求进行拆分处理；</p> <p>1' b0: 当内存访问出现地址非对齐时，上报非对齐访问异常；</p> <p>复位值为 1' b1</p>         |
| [13] | RW | PMDM      | <p>1' b0: 机器模式下事件监测计数器可以正常计数</p> <p>1' b1: 机器模式下事件监测计数器禁止计数</p> <p>复位值为 1' b0</p>                                           |
| [10] | RW | PMDU      | <p>1' b0: 用户模式下事件监测计数器可以正常计数</p> <p>1' b1: 用户模式下事件监测计数器禁止计数</p> <p>复位值为 1' b0</p>                                           |

## MHCR

机器模式硬件配置寄存器。

| Bits | Access | Name | Description                                                           |
|------|--------|------|-----------------------------------------------------------------------|
| [12] | RW     | BTB  | <p>当 BTB 值为 0 时，分支目标预测关闭；当 BTB 值为 1 时，分支目标预测开启。</p> <p>复位值为 1' b0</p> |
| [5]  | RW     | BPE  | <p>当 BPE 为 0 时，预测跳转关闭；当 BPE 为 1 时，预测跳转开启。</p> <p>复位值为 1' b0</p>       |
| [4]  | RW     | RS   | <p>当 RS 为 0 时，返回栈关闭；当 RS 为 1 时，返回栈开启。</p> <p>复位值为 1' b0</p>           |

|     |    |    |                                                                                                                                                                                                                                                         |
|-----|----|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [3] | RW | WA | 表示处理器执行 Store 指令且数据 cache 缺失时，是否在数据 cache 中分配缺失的缓存行。当 WA 为 0 时，数据 cache 为 write non-allocate 模式，即在数据 cache 缺失时不会将缺失的数据缓存行从内存加载到数据 cache 内；当 WA 为 1 时，数据 cache 为 write allocate 模式，即在 Store 指令导致的数据 cache 缺失时会将该缺失的数据缓存行从内存中加载到数据 cache 中。<br>复位值为 1' b0 |
| [2] | RW | WB | 当 WB 为 0 时，数据 cache 为写直模式；当 WB 为 1 时，数据 cache 为写回模式。<br>复位值为 1' b0                                                                                                                                                                                      |
| [1] | RW | DE | 当 DE 为 0 时，数据 cache 关闭；当 DE 为 1 时，数据 cache 开启。<br>复位值为 1' b0                                                                                                                                                                                            |
| [0] | RW | IE | 当 IE 为 0 时，指令 cache 关闭；当 IE 为 1 时，指令 cache 开启。<br>复位值为 1' b0                                                                                                                                                                                            |

## MHINT

机器模式隐式操作寄存器。

| Bits | Access | Name           | Description                                                             |
|------|--------|----------------|-------------------------------------------------------------------------|
| [21] | RW     | ICACHE_FIFO_EN | 此 bit 为 1 时，使用 FIFO 路替换策略实现 ICACHE 的路替换。                                |
| [20] | RW     | AEE            | 当 AEE 为 1 时，处理器处于精确异常工作模式。<br>当 AEE 为 0 时，处理器处于非精确异常工作模式。<br>复位值为 1' b0 |
| [10] | RW     | IWPE           | 当 IWPE 为 0 时，ICACHE 路预测关闭；<br>当 IWPE 为 1 时，ICACHE 路预测开启；                |

## MEXSTATUS

机器模式扩展异常状态寄存器。

| Bits  | Access | Name     | Description                                                                                                                |
|-------|--------|----------|----------------------------------------------------------------------------------------------------------------------------|
| [17]  | RW     | SPSWAPEN | 该位为 0 时，不使能自动中断栈切换机制；该位为 1 时，使能自动中断栈切换机制，当响应第 0 层中断时，自动将 mscratchcswl 寄存器的值写入 sp 寄存器。第 0 层中断判断条件：当前 MIL 为 0。<br>复位值为 1' b0 |
| [16]  | RW     | SPUSHEN  | 该位为 0 时，不使能硬件自动压栈机制；该位为 1 时，使能硬件自动压栈机制，响应中断时，硬件投机执行一条 IPUSH 指令。<br>复位值为 1' b0                                              |
| [8]   | RW     | BUSERR   | 当 BUSERR 为 1 时，表示处理器上一次发生的异常为 BUS ERR 异常。<br>复位值为 1' b0                                                                    |
| [7]   | RO     | NMI_STS  | 当 NMI_STS 为 1 时，表示处理器进入 NMI 处理状态。<br>复位值为 1' b0                                                                            |
| [6]   | RO     | LOCKUP   | 当 LOCKUP 为 1 时，表示处理器进入锁定状态。<br>复位值为 1' b0                                                                                  |
| [5]   | RO     | EXPT_VLD | 当 EXPT_VLD 为 1 时，表示处理器进入异常处理状态。<br>复位值为 1' b0                                                                              |
| [4]   | RW     | WFEEN    | 当 WFEEN 为 1 时，执行 WFI 指令实际为 WFE 功能，event 可以唤醒，中断不需要考虑优先级即可唤醒；当 WFEEN 为 0 时，WFI 指令仅能由中断进行唤醒，而且需要考虑中断优先级。<br>复位值为 1' b1       |
| [3:2] | RW     | LPMD     | 当 LPMD 为 2' b00 时，下次使用 WFI 指令进入深睡眠；当 LPMD 为 2' b01 时，下次使用 WFI 指令进入浅睡眠。<br>复位值为 2' b00                                      |
| [1:0] | RW     | SOFT_RST | 向 SOFT_RST 写 2' b00 时，表示不需要进行软复位；向 SOFT_RST 写 2' b01 时，表示需要复位核；                                                            |



|  |  |  |                                                                                      |
|--|--|--|--------------------------------------------------------------------------------------|
|  |  |  | 向 SOFT_RST 写 2' b10 时, 表示需要复位整个系统;<br>向 SOFT_RST 写 2' b11 为 reserved。<br>复位值为 2' b00 |
|--|--|--|--------------------------------------------------------------------------------------|

## MRADDR

机器模式复位地址指示寄存器。

| Bits   | Access | Name   | Description       |
|--------|--------|--------|-------------------|
| [31:0] | RO     | MRADDR | 用于指示 CPU 的复位启动地址。 |

## MNMICAUSE

机器模式 NMI 状态寄存器。

| Bits    | Access | Name       | Description                            |
|---------|--------|------------|----------------------------------------|
| [31]    | RO     | NMI_INTR   | NMI 响应时 MCAUSE 内 INTR 寄存器的值。           |
| [29:28] | RO     | NMI_MPP    | NMI 响应时 MSTATUS 内 MPP 寄存器的值。           |
| [27]    | RO     | NMI_MPIE   | NMI 响应时 MSTATUS 内 MPIE 寄存器的值。          |
| [11:0]  | RO     | NMI_VECTOR | NMI 响应时 MCAUSE 内 exception code 寄存器的值。 |

## MNMIPC

机器模式 NMI 异常程序计数寄存器。

| Bits   | Access | Name   | Description  |
|--------|--------|--------|--------------|
| [31:0] | RO     | MNMIPC | NMI 异常程序计数值。 |

### 1.4.6.3 机器模式寄存器

## FFLAGS

浮点异常累积状态寄存器。

| Bits   | Access | Name   | Description                |
|--------|--------|--------|----------------------------|
| [31:0] | RO     | FFLAGS | 浮点控制状态寄存器 (FCSR) 的异常累积域映射。 |

## FRM

浮点动态舍入模式寄存器。

| Bits   | Access | Name | Description              |
|--------|--------|------|--------------------------|
| [31:0] | RW     | FRM  | 浮点控制状态寄存器（FCSR）的舍入模式域映射。 |

## FCSR

浮点控制状态寄存器。

| Bits  | Access | Name | Description                                                                                                                                   |
|-------|--------|------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| [7:5] | RW     | RM   | 当 RM=0 时，RNE 舍入模式，向最近偶数舍入。<br>当 RM=1 时，RTZ 舍入模式，向 0 舍入。<br>当 RM=2 时，RDN 舍入模式，向负无穷舍入。<br>当 RM=3 时，RUP 舍入模式，向正无穷舍入。<br>当 RM=4 时，RMM 舍入模式，向最近舍入。 |
| [4]   | RW     | NV   | 当 NV=0 时，没有产生无效操作数异常。<br>当 NV=1 时，产生无效操作数异常。                                                                                                  |
| [3]   | RW     | DZ   | 当 DZ=0 时，没有产生除 0 异常。<br>当 DZ=1 时，产生除 0 异常。                                                                                                    |
| [2]   | RW     | OF   | 当 OF=0 时，没有产生上溢异常。<br>当 OF=1 时，产生上溢异常。                                                                                                        |
| [1]   | RW     | UF   | 当 UF=0 时，没有产生下溢异常。<br>当 UF=1 时，产生下溢异常。                                                                                                        |
| [0]   | RW     | NX   | 当 NX=0 时，没有产生非精确异常。<br>当 NX=1 时，产生非精确异常。                                                                                                      |

## FXCR

浮点扩展控制寄存器。

| Bits    | Access | Name | Description   |
|---------|--------|------|---------------|
| [26:24] | RW     | RM   | FCSR 对应字段的映射。 |

|     |    |       |                                                                                                       |
|-----|----|-------|-------------------------------------------------------------------------------------------------------|
| [5] | RW | DQNaN | 当 DQNaN 为 0 时，计算输出的 QNaN 值为 RISC-V 规定的固定值，即 0x7FC0_0000。<br>当 DQNaN 为 1 时，计算输出的 QNaN 值根 IEEE754 标准一致。 |
| [5] | RW | FE    | 当有任何一个浮点异常发生时，该位将被置为 1。                                                                               |
| [4] | RW | NV    | FCSR 对应字段的映射。                                                                                         |
| [3] | RW | DZ    | FCSR 对应字段的映射。                                                                                         |
| [2] | RW | OF    | FCSR 对应字段的映射。                                                                                         |
| [1] | RW | UF    | FCSR 对应字段的映射。                                                                                         |
| [0] | RW | NX    | FCSR 对应字段的映射。                                                                                         |

## 1.5 中断系统

芯片支持 128 个中断源，对应的中断映射如表 1-14 所示。

表1-14 中断源分配表

| 中断位         | 中断源      | 中断位       | 中断源   | 中断位        | 中断源      |
|-------------|----------|-----------|-------|------------|----------|
| <b>0~31</b> | CPU 内部中断 | <b>64</b> | EMMC  | <b>97</b>  | MCU      |
| <b>32</b>   | RTC      | <b>65</b> | SDIO0 | <b>98</b>  | MCU_LOCK |
| <b>33</b>   | SOFTWARE | <b>66</b> | SDIO1 | <b>99</b>  | IVE      |
| <b>34</b>   | WDG0     | <b>67</b> | SPACC | <b>100</b> | 保留       |
| <b>35</b>   | WDG1     | <b>68</b> | FMC   | <b>101</b> | 保留       |
| <b>36</b>   | WDG2     | <b>69</b> | ETH   | <b>102</b> | 保留       |
| <b>37</b>   | LSADC    | <b>70</b> | USB2  | <b>103</b> | 保留       |
| <b>38</b>   | TIMER0   | <b>71</b> | FEPHY | <b>104</b> | 保留       |
| <b>39</b>   | TIMER1   | <b>72</b> | JPEG  | <b>105</b> | 保留       |

| 中断位 | 中断源    | 中断位 | 中断源             | 中断位 | 中断源 |
|-----|--------|-----|-----------------|-----|-----|
| 40  | TIMER2 | 73  | VPSS            | 106 | 保留  |
| 41  | TIMER3 | 74  | DDRC            | 107 | 保留  |
| 42  | TIMER4 | 75  | DDRT            | 108 | 保留  |
| 43  | UART0  | 76  | VGS             | 109 | 保留  |
| 44  | UART1  | 77  | VPU             | 110 | 保留  |
| 45  | UART2  | 78  | NPU             | 111 | 保留  |
| 46  | UART3  | 79  | AIAO            | 112 | 保留  |
| 47  | I2C0   | 80  | VIPROC          | 113 | 保留  |
| 48  | I2C1   | 81  | MIPIRX0_4L      | 114 | 保留  |
| 49  | I2C2   | 82  | MIPIRX0_2L      | 115 | 保留  |
| 50  | I2C3   | 83  | VICAP           | 116 | 保留  |
| 51  | SPI0   | 84  | VOUT            | 117 | 保留  |
| 52  | SPI1   | 85  | GZIP            | 118 | 保留  |
| 53  | SPI2   | 86  | A7_COMMRX<br>0  | 119 | 保留  |
| 54  | GPIO0  | 87  | A7_COMMRX<br>1  | 120 | 保留  |
| 55  | GPIO1  | 88  | A7_COMMTX<br>0  | 121 | 保留  |
| 56  | GPIO2  | 89  | A7_COMMTX<br>1  | 122 | 保留  |
| 57  | GPIO3  | 90  | A7_PMUIRQ0      | 123 | 保留  |
| 58  | GPIO4  | 91  | A7_PMUIRQ1      | 124 | 保留  |
| 59  | GPIO5  | 92  | 保留              | 125 | 保留  |
| 60  | GPIO6  | 93  | 保留              | 126 | 保留  |
| 61  | GPIO7  | 94  | AUDIO_COD<br>EC | 127 | 保留  |

| 中断位 | 中断源   | 中断位 | 中断源 | 中断位 | 中断源 |
|-----|-------|-----|-----|-----|-----|
| 62  | GPIO8 | 95  | 保留  |     |     |
| 63  | DMAC  | 96  | 保留  |     |     |

## 1.6 DMA 控制器

### 1.6.1 概述

DMA(Direct Memory Access)是一种高速的数据传输操作，允许不通过 CPU 在外部设备和存储器之间直接读写数据。DMAC (DMA Controller) 直接在存储器和外设、外设和外设、存储器和存储器之间进行数据传输，避免 CPU 干涉并减少了 CPU 中断处理开销。

芯片只支持一个 DMAC。

### 1.6.2 特点

DMAC 有如下特点：

- 提供 4 个 DMA 通道，每个通道可配置用于一种单向传输。
- 支持内存到内存、内存到外设、外设到内存、bootram 到内存之间的传输。
- 支持通道优先级可配。
- 支持软件配置 DMA 请求。
- 支持 DMA BURST 长度软件可配。
- 支持 8bit、16bit、32bit、64bit 数据位宽方式传输。
- 只支持小端模式。

### 1.6.3 功能描述

#### 1.6.3.1 外设请求线

DMAC 的硬件请求和相应设备的对应关系如表 1-15 所示。

表1-15 DMAC 外设硬件请求线编号说明

| 外设硬件请求线编号 | 对应设备               |
|-----------|--------------------|
| 0         | I2C0 RX 通道 DMA 请求  |
| 1         | I2C0 TX 通道 DMA 请求  |
| 2         | I2C1 RX 通道 DMA 请求  |
| 3         | I2C1 TX 通道 DMA 请求  |
| 4         | I2C2 RX 通道 DMA 请求  |
| 5         | I2C2 TX 通道 DMA 请求  |
| 6         | SPI2 RX 通道 DMA 请求  |
| 7         | SPI2 TX 通道 DMA 请求  |
| 8         | UART1 RX 通道 DMA 请求 |
| 9         | UART1 TX 通道 DMA 请求 |
| 10        | UART2 RX 通道 DMA 请求 |
| 11        | UART2 TX 通道 DMA 请求 |
| 12        | SPI0 RX 通道 DMA 请求  |
| 13        | SPI0 TX 通道 DMA 请求  |
| 14        | SPI1 RX 通道 DMA 请求  |
| 15        | SPI1 TX 通道 DMA 请求  |

### 1.6.3.2 访问空间

DMAC 可以访问的地址空间如表 1-16 所示。

表1-16 DMAC 访问空间说明

| 访问空间类型 | 描述      |
|--------|---------|
| Memory | DDR 空间  |
|        | BOOTRAM |

| 访问空间类型 | 描述       |
|--------|----------|
|        | MCU_IMEM |
| 外设     | UART1    |
|        | UART2    |
|        | I2C0     |
|        | I2C1     |
|        | I2C2     |
|        | SPI0     |
|        | SPI1     |
|        | SPI2     |

### 1.6.3.3 基本传输

DMAC 支持支持三种基本传输，即内存之间的传输、内存到外设的传输以及外设到内存的传输。一次基本传输最大可搬运 $(2^{22}-1)*8$  Bytes 的数据。

- 内存之间的传输是源地址和目的地址均为内存物理地址的传输。
- 内存到外设的传输是源地址为内存物理地址而目的地址为外设的数据 TX FIFO 寄存器地址的传输，且目的地址必须配置为在传输过程中固定不变。
- 外设到内存的传输是源地址为外设的数据 RX FIFO 寄存器地址而目的地址为内存物理地址的传输，且源地址必须配置为在传输过程中固定不变。
- 如果源地址配置为不递增，源地址需要和源端的传输位宽对齐，否则会报配置错误中断；如果源端为外设时，源端传输位宽的大小应该与外设 RX FIFO 位宽相等。
- 如果目的地址配置为不递增，目的地址需要和目的端的传输位宽对齐，否则会报配置错误中断；如果目的端为外设时，目的端传输位宽的大小应该与外设 TX FIFO 位宽相等。

## 须知

- 对于内存到外设以及外设到内存的传输，当一次传输的总数据量大于一次 Burst 传输时，DMA 会优先响应 Burst 请求，而忽略 Single 请求。
- 当 DMAC 执行内存到外设的传输时，由于外设的 TX FIFO 水线及 DMA 目的端的 Burst 数据量设置不合理，可能导致 DMA 一次 BURST 传输数目大于 TX FIFO 剩余的空间，从而造成数据丢失。
- 当 DMAC 执行外设到内存的传输时，由于外设的 RX FIFO 水线及 DMA 源端的 Burst 数据量设置不合理，可能存在 RX FIFO 剩余的数据量大于 DMA 源端一次 BURST 数据量，但却小于 RX 水线的情况，此时外设不会产生 RX BURST 请求，而 DMA 只响应 RX BURST 请求，从而造成 DMA 挂死。

### 1.6.3.4 中断和状态

通道完成中断用来指示通道传输完成。

终止错误中断有 2 种发生情况：

- 1) 对 CES[3:0]任意 bit 写 0；
- 2) Axi bus 读操作的时候接收错误响应；

看门狗中断用来指示 dma 传输过程超时；

数据传输错误中断表示 dma 读传输或写传输错误；

数据传输错误中断用来指示总线的响应错误，如访问非法空间、访问权限出错等，其中，数据传输错误中断发生在传输数据期间。

#### 说明

- 在传输开始前应该清除所有中断，保证工作时中断可以正常产生。
- 通道正常传输完成或者产生错误中断，通道使能自动清除。
- 通道为外设传输时，应在外设请求无效时查询，查询到值为 DMAC 实际剩余未搬移的数据量和下一次外设请求时搬移的地址。
- 通道为内存之间的传输时，DMAC 内部一直在搬移，因此，查询到的值与 DMAC 实际搬移的值有一定差异，因此，内存之间传输时查询该寄存器意义不大。



## 1.6.4 工作方式

### 1.6.4.1 时钟和复位

DMAC 在使用前需要打开时钟并执行复位操作：

步骤 1 向 CRG 寄存器 PERI\_CRG101 [dmac\_cken]写 0x1，打开 DMAC 的时钟门控。

步骤 2 向 CRG 寄存器 PERI\_CRG101 [dmac\_srst\_req]写 0x1，对 DMAC 进行复位。

步骤 3 向 CRG 寄存器 PERI\_CRG101 [dmac\_srst\_req]写 0x0，撤离对 DMAC 模块的复位。

----结束

### 1.6.4.2 初始化

在撤销复位以及设置访问方式后，DMAC 应该进行初始化。初始化步骤如下：

步骤 1 根据场景配置 DMAC 优先级寄存器 CNCFG，[ChPri]写 1 为配置高优先级；

步骤 2 根据中断需求配置中断屏蔽寄存器 CNCFG，[AbtIntMsk]通道中止中断屏蔽，  
[ErrIntMsk]通道错误中断屏蔽，[TCIntMsk]传输结束中断屏蔽。

----结束

### 1.6.4.3 基本传输

DMAC 初始化完成之后，需要配置并启动通道，才可以使用 DMAC 进行数据传输。具体步骤如下：

步骤 1 写通道寄存器 DMAC\_SRC\_ADDR (chn\_index)，设置源端首物理地址；

步骤 2 写通道寄存器 DMAC\_DEST\_ADDR (chn\_index)，设置目的端首物理地址；

步骤 3 写通道寄存器 DMAC\_CNTSIZE (chn\_index)，设置传输次数，传输大小为源端读取的数据的 size 的大小，最大支持传输次数为 222-1；

步骤 4 写通道寄存器 DMAC\_CONFIG (chn\_index)：

- 1) 设置 srcTcnt 用于配置 axi burst\_len，注意 mem2mem 传输的时候配置这个值无效，dma 根据传输数据量自动设置成最大；
- 2) 设置 srcwidth/dstwidth 用于配置源/目的数据传输 axi size 值；
- 3) 设置 srcCtrl/dstCtrl 用于配置源/目的端传输 burst\_type；
- 4) 设置 ChEN 用于配置 dma 通道使能；

- 5) 设置 Wsync 用于配置 DMA 传输完成中断判断条件，写 1 为等待 DMA 所有命令写入 AXI 总线标记传输完成，写 0 为所有 DMA 命令推入命令队列即标记传输完成。

步骤 5 写通道寄存器 CNCFG (chn\_index):

- 1) 设置 UnalignMode, 用于配置成对齐模式 (0x1) 和非对齐模式 (0x0);
- 2) 设置 ChPri, 用于配置成通道为高优先级还是低优先级;
- 3) 设置 ChGntWin, 用于配置优先级窗口大小。当处理 axi 命令长度大于窗口值时, 重新进行优先级分配;
- 4) 设置 DstHEN/DstRS, 用于配置目的端握手使能 (外设为目的端需要配置这个为 1) 及使用哪个通道进行传输;
- 5) 设置 SrcHEN/SrcRS, 用于配置源端握手使能 (外设为源端需要配置这个为 1) 及使用哪个通道进行传输;

步骤 6 等待 DMA 传输完成中断 DMAC\_TCINT, 清除对应中断寄存器 DMAC\_TCINT\_CLR (该寄存器为写 1 清寄存器);

----结束

#### 1.6.4.4 中断处理

中断程序的处理流程如下:

步骤 1 读 dma 总中断状态寄存器 INT, 找出发出中断请求的通道;

步骤 2 读 TCINT 和 EAINT, 确认是异常中断错误上报还是传输完成中断上报;

步骤 3 传输完成中断上报处理:

- 1) 写寄存器 TCINTC, 对选定的对应位写入 0x1, 清楚对应通道的中断状态;
- 2) 取走对应内存/外设中的数据, 有必要的话重新配置该通道并启动该通道;
- 3) 退出中断处理;

步骤 4 错误中断处理:

- 1) 根据错误类型使用 EAINTC 寄存器对应位写 0x1 清除对应错误类型;
- 2) 给出错误信息, 有必要的话重新配置并启动该通道;
- 3) 退出中断处理;

----结束

## 1.6.5 DMA 寄存器概述

DMA 基地址: 0x100B0000

表1-17 DMA 寄存器概览

| 偏移地址 | 名称        | 描述                |
|------|-----------|-------------------|
| 0x00 | INT       | dma 总中断           |
| 0x04 | TCINT     | dma 传输完成中断        |
| 0x08 | TCINTC    | dma 传输完成中断清除      |
| 0x0c | EAINC     | dma 异常中断          |
| 0x10 | EAINC     | dma 异常中断清除        |
| 0x14 | TCS       | dma 通道传输完成状态      |
| 0x18 | EAS       | dma 异常中断状态        |
| 0x1c | CES       | dma 通道使能          |
| 0x20 | SPI       | 同步外设接口            |
| 0x24 | LDMBA     | 本地描述符 mem 基地址     |
| 0x28 | WDT       | 看门狗计数器            |
| 0x2c | GER       | 全局事件寄存器           |
| 0x30 | PSE       | PSlvErr 使能        |
| 0x34 | REV       | 版本号               |
| 0x38 | FEA       | 硬件特性              |
| 0x54 | FEA2      | 硬件特性 2            |
| 0x3c | LMFFS (n) | 本地描述符 mem 空闲 flag |
| 0x48 | LMFFS (n) | 本地描述符 mem 空闲 flag |
| 0x4c | EDNC      | 大小端配置             |
| 0x50 | CVWOM     | 约束只写模式写入值         |

| 偏移地址         | 名称               | 描述                            |
|--------------|------------------|-------------------------------|
| 0x100+0x20*N | CnCTL (n)        | DMA 通道 N 控制寄存器                |
| 0x104+0x20*N | CnCFG            | DMA 通道 N 配置寄存器                |
| 0x108+0x20*N | CnSRCA (n)       | DMA 通道 N 源地址寄存器               |
| 0x10c+0x20*N | CnDSTA (n)       | DMA 通道 N 目的地址寄存器              |
| 0x110+0x20*N | CnLNK (n)        | DMA 通道 N LLP 寄存器              |
| 0x114+0x20*N | CnTSIZE(n)       | DMA 通道传输大小寄存器                 |
| 0x114+0x20*N | CnTSIZE(n)       | DMA 通道传输大小寄存器 (当设置为 2D 模式时使用) |
| 0x118+0x20*N | CnSSDA (n)       | DMA 通道 N 源/目的步长寄存器 (2D 模式时使用) |
| 0x200+0x20*N | CnSADAEXP (n)    | DMA 通道 N 源地址和目的地址拓展位寄存器       |
| 0x204+0x20*N | CnLLPUSEREXP (n) | DMA 通道 N LLP 和用户自定义信号拓展位寄存器   |

## 1.6.6 DMA 寄存器描述

### INT

int 为 dma 总中断。

Offset Address:0x0 Total Reset Value:0x0

| Bits   | Access | Name     | Description  | Reset |
|--------|--------|----------|--------------|-------|
| [31:8] | RO     | reserved | 保留           | 0x00  |
| [7:0]  | RO     | intr     | 每个通道的 dma 中断 | 0x00  |

### TCINT

tcint 为 dma 传输完成中断。

Offset Address:0x4 Total Reset Value:0x0

| Bits   | Access | Name     | Description    | Reset |
|--------|--------|----------|----------------|-------|
| [31:8] | RO     | reserved | 保留             | 0x00  |
| [7:0]  | RO     | TCintr   | 每个通道的 dma 完成中断 | 0x00  |

## TCINTC

tcintc 为 dma 传输完成中断清除。

Offset Address:0x8 Total Reset Value:0x0

| Bits   | Access | Name      | Description     | Reset |
|--------|--------|-----------|-----------------|-------|
| [31:8] | RO     | reserved  | 保留              | 0x00  |
| [7:0]  | RWC    | TCintrClr | 每个通道 dma 完成中断清除 | 0x00  |

## EAINT

eaint 为 dma 异常中断。

Offset Address:0xc Total Reset Value:0x0

| Bits    | Access | Name     | Description | Reset |
|---------|--------|----------|-------------|-------|
| [31:24] | RO     | reserved | 保留          | 0x00  |
| [23:16] | RO     | AbtIntr  | DMA 通道终止中断  | 0x00  |
| [15:8]  | RO     | WdtIntr  | DMA 通道超时中断  | 0x00  |
| [7:0]   | RO     | ErrInt   | DMA 通道错误中断  | 0x00  |

## EAINTC

eaintc 为 dma 异常中断清除。

Offset Address:0x10 Total Reset Value:0x0

| Bits    | Access | Name       | Description | Reset |
|---------|--------|------------|-------------|-------|
| [31:24] | RO     | reserved   | 保留          | 0x00  |
| [23:16] | RWC    | AbtIntrClr | DMA 通道终止中   | 0x00  |

|        |     |            |                     |      |
|--------|-----|------------|---------------------|------|
|        |     |            | 断清除，写 1 清除          |      |
| [15:8] | RWC | WdtIntrClr | DMA 通道超时中断清除，写 1 清除 | 0x00 |
| [7:0]  | RWC | ErrIntClr  | DMA 通道错误中断清除，写 1 清除 | 0x00 |

## TCS

tcs 为 dma 通道传输完成状态。

Offset Address:0x14 Total Reset Value:0x0

| Bits   | Access | Name     | Description              | Reset |
|--------|--------|----------|--------------------------|-------|
| [31:8] | RO     | reserved | 保留                       | 0x00  |
| [7:0]  | RO     | TCStatus | 每个通道 dma 完成源中断状态，不看 mask | 0x00  |

## EAS

eas 为 dma 异常中断状态。

Offset Address:0x18 Total Reset Value:0x0

| Bits    | Access | Name      | Description         | Reset |
|---------|--------|-----------|---------------------|-------|
| [31:24] | RO     | reserved  | 保留                  | 0x00  |
| [23:16] | RO     | AbtStatus | DMA 通道终止源中断，不看 mask | 0x00  |
| [15:8]  | RO     | WdtStatus | DMA 通道超时源中断，不看 mask | 0x00  |
| [7:0]   | RO     | ErrStatus | DMA 通道错误源中断，不看 mask | 0x00  |

## CES

ces 为 dma 通道使能。

Offset Address:0x1c Total Reset Value:0x0

| Bits   | Access | Name     | Description                                                                    | Reset |
|--------|--------|----------|--------------------------------------------------------------------------------|-------|
| [31:8] | RO     | reserved | 保留                                                                             | 0x00  |
| [7:0]  | RW     | ChEnable | DMA 通道使能状态，这个寄存器可以读出通道是否使能。写 0 会导致通道进入终止状态，写 1 没有效果，通道使能配置由 control reg 寄存器配置。 | 0x00  |

## SPI

spi 为同步外设接口。

Offset Address:0x20 Total Reset Value:0x0

| Bits    | Access | Name       | Description                      | Reset |
|---------|--------|------------|----------------------------------|-------|
| [31:16] | RO     | reserved   | 保留                               | 0x00  |
| [15:0]  | RW     | SyncDmaReq | 外设接口同步使能寄存器。写 1 同步外设请求到 dma 时钟域。 | 0x00  |

## LDMBA

ldmba 为本地描述符 mem 基地址。

Offset Address:0x24 Total Reset Value:0x0

| Bits     | Access | Name     | Description                               | Reset |
|----------|--------|----------|-------------------------------------------|-------|
| [31:20]  | RW     | LDMBase  | 本地描述符 mem 基地址                             | 0x00  |
| [19:M+1] | RO     | reserved | 保留                                        | 0x00  |
| [M:0]    | RW     | LDMBExp  | 本地描述符基地址拓展位 (M 通过 FTDMAC030_AXI_AW_N 来计算) | 0x00  |

## WDT

wdt 为看门狗计数器。

Offset Address:0x28      Total Reset Value:0x0

| Bits    | Access | Name     | Description | Reset |
|---------|--------|----------|-------------|-------|
| [31:16] | RO     | reserved | 保留          | 0x00  |
| [15:0]  | RW     | WDTimer  | 看门狗计数值      | 0x00  |

## GER

ger 为全局事件寄存器。

Offset Address:0x2c      Total Reset Value:0x0

| Bits    | Access | Name        | Description              | Reset |
|---------|--------|-------------|--------------------------|-------|
| [31:24] | RO     | reserved    | 保留                       | 0x00  |
| [23:16] | RWC    | GlbEventClr | 对应 bit 为 0 清除全局事件, 写 1 清 | 0x00  |
| [15:8]  | RWC    | GlbEventSet | 对应 bit 为 1 置位全局事件, 写 1 清 | 0x00  |
| [7:0]   | RO     | GlbEvent    | 全局事件                     | 0x00  |

## PSE

pse 为 PSIVErr 使能。

Offset Address:0x30      Total Reset Value:0x0

| Bits   | Access | Name      | Description             | Reset |
|--------|--------|-----------|-------------------------|-------|
| [31:1] | RO     | reserved  | 保留                      | 0x00  |
| [0]    | RW     | PSIVErrEn | apb slave 总线访问非法地址时报错使能 | 0x00  |

## REV

rev 为版本号。

Offset Address:0x34      Total Reset Value:0x0



| Bits   | Access | Name   | Description   | Reset |
|--------|--------|--------|---------------|-------|
| [31:0] | RO     | RevNum | FTDMAC030 版本号 | 0x00  |

## FEA

fea 为硬件特性。

Offset Address:0x38 Total Reset Value:0x0

| Bits    | Access | Name     | Description                                                   | Reset |
|---------|--------|----------|---------------------------------------------------------------|-------|
| [31:30] | RO     | reserved | 保留                                                            | 0x00  |
| [29:28] | RO     | CmdDepth | 读写命令队列深度 0: Depth =2,1: Depth=4,2: Depth=8,3: Depth=16        | HwCfg |
| [27:26] | RO     | reserved | 保留                                                            | 0x00  |
| [25:24] | RO     | LdmDepth | 本地描述符深度 0: Depth =32,1: Depth=64,2: Depth=128                 | HwCfg |
| [23:21] | RO     | reserved | 保留                                                            | 0x00  |
| [20]    | RO     | LdmOn    | 本地描述符 mem 配置 0: 不支持配置 1: 支持配置                                 | HwCfg |
| [19:16] | RO     | PriNum   | 外设接口数量 0:1 对 (dma_req/dma_ack) ... 15:16 对                    | HwCfg |
| [15:13] | RO     | reserved | 保留                                                            | 0x00  |
| [12]    | RO     | PriOn    | 外设接口配置 0: 不支持 外设接口 1: 支持外设接口                                  | HwCfg |
| [11]    | RO     | reserved | 保留                                                            | 0x00  |
| [10:8]  | RO     | DFDepth  | 设置数据 fifo 的深度 0: Depth =8,1: Depth=16,2: Depth=32,3: Depth=64 | HwCfg |

|       |    |             |                                                              |       |
|-------|----|-------------|--------------------------------------------------------------|-------|
| [7:6] | RO | SLVDWidth   | axi slave 端 data bus 位宽<br>00: :32bit 01:64bit<br>10:128bit  | HwCfg |
| [5:4] | RO | Dwidth      | axi master 端 data bus 位宽<br>00: :32bit 01:64bit<br>10:128bit | HwCfg |
| [3]   | RO | UnalignMode | 非对齐模式配置 0: 不支持非对齐 1: 支持非对齐                                   | HwCfg |
| [2:0] | RO | ChNum       | dma 通道数量配置 0: 支持单个通道 ... 7:支持 8 个通道                          | HwCfg |

## FEA2

fea2 为硬件特性 2。

Offset Address:0x54 Total Reset Value:0x0

| Bits    | Access | Name        | Description        | Reset |
|---------|--------|-------------|--------------------|-------|
| [31:20] | RO     | reserved    | 保留                 | 0x00  |
| [19:16] | RO     | ARUserWidth | 用户自定义读地址通道数量       | HwCfg |
| [15:12] | RO     | reserved    | 保留                 | 0x00  |
| [11:8]  | RO     | AWUserWidth | 用户自定义写地址通道数量       | HwCfg |
| [7:6]   | RO     | reserved    | 保留                 | 0x00  |
| [5:0]   | RO     | AddrWidth   | axi 地址总线位宽 (32~48) | HwCfg |

## LMFFS (N)

lmffs (n) 为本地描述符 mem 空闲 flag。

Offset Address:0x3c~0x48 Total Reset Value:0x0

| Bits   | Access | Name       | Description | Reset      |
|--------|--------|------------|-------------|------------|
| [31:0] | RWC    | FreeFlagS0 | 本地描述符 mem   | 0xffff_fff |

|  |  |  |                              |  |
|--|--|--|------------------------------|--|
|  |  |  | 空闲 flag 0: 被占用 1: 空闲 , 写 1 清 |  |
|--|--|--|------------------------------|--|

## EDNC

ednc 为大小端配置。

Offset Address:0x4c Total Reset Value:0x0

| Bits    | Access | Name            | Description                              | Reset |
|---------|--------|-----------------|------------------------------------------|-------|
| [31:17] | RO     | reserved        | 保留                                       | 0x00  |
| [16]    | RW     | LMEndianConvert | axi slave 端大小端转换 1: BE-8 to Le 转换 0: 不转换 | 0x00  |
| [15:8]  | RO     | reserved        | 保留                                       | 0x00  |
| [7:0]   | RW     | EndianConvert   | 通道 0~通道 7 的大小端转换 1: BE-8                 | 0x00  |

## CVWOM

cvwom 为约束只写模式写入值。

Offset Address:0x50 Total Reset Value:0x0

| Bits   | Access | Name    | Description                                  | Reset |
|--------|--------|---------|----------------------------------------------|-------|
| [31:0] | RW     | WOValue | 当 WO 模式使能时候, axi master 将会写入部分 mem 区域约束范围内的值 | 0x00  |

## CNCTL (N)

cnctl (n) 为 DMA 通道 N 控制寄存器。

Offset Address:0x100+0x20\*N Total Reset Value:0x0

| Bits    | Access | Name    | Description | Reset |
|---------|--------|---------|-------------|-------|
| [31:29] | RO     | SrcTcnt | DMA 握手操作源   | 0x00  |

|         |    |          |                                                                                                                                                                                            |      |
|---------|----|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
|         |    |          | 端传输计数<br>000: 1beat<br>001: 2beat<br>010: 4beat<br>011: 8beat<br>100: 16beat<br>101: 32beat<br>110: 64beat<br>111: 128beat                                                                 |      |
| [28]    | RW | TCMsk    | 完成计数 mask<br>1: 屏蔽完成计数<br>状态更新 0: 不屏蔽<br>完成计数状态更新                                                                                                                                          | 0x00 |
| [27:25] | RW | SrcWidth | 源端数据位宽。<br>000: 8bit<br>001: 16bit<br>010: 32bit<br>011: 64bit<br>100: 128bit<br>注意：当配置为非<br>对齐模式时，源端<br>数据位宽应该设置<br>成数据总线的位<br>宽。当配置成 WO<br>模式时，源端数据<br>位宽应该和目的端<br>数据位宽一致，并<br>且小于 32bit | 0x00 |
| [24:22] | RW | DstWidth | 目的端数据位宽。<br>000: 8bit<br>001: 16bit                                                                                                                                                        | 0x00 |

|         |    |         |                                                                                                                        |      |
|---------|----|---------|------------------------------------------------------------------------------------------------------------------------|------|
|         |    |         | 010: 32bit<br>011: 64bit<br>100: 128bit<br>注意：当配置为非对齐模式时，目的端数据位宽应该设置成数据总线的位宽。当配置成 WO 模式时，目的端数据位宽应该和源端数据位宽一致，并且小于 32bit |      |
| [21:20] | RW | SrcCtrl | 源端 burst type<br>00: Incr 01: fixed 注意在非对齐模式和 WO 模式下只能配置成 00                                                           | 0x00 |
| [19:18] | RW | DstCtrl | 目的端 burst type<br>00: Incr 01: fixed 注意在非对齐模式和 WO 模式下只能配置成 00                                                          | 0x00 |
| [17]    | RW | WDTEn   | 看门狗计数器使能<br>0: 不使能 1: 使能 注意,一旦通道被终止, 看门狗计数器使能也将无效                                                                      | 0x00 |
| [16]    | RW | ChEn    | 通道使能 0: 不使能 1: 使能 注意: 这个 bit 将自动关闭一旦请求结束                                                                               | 0x00 |

|        |    |          |                                                                                               |      |
|--------|----|----------|-----------------------------------------------------------------------------------------------|------|
| [15]   | RW | ExpEn    | 拓展控制使能位<br>0: 不使能 1: 使能                                                                       | 0x00 |
| [14]   | RW | 2DEn     | 2D 传输使能 0:<br>不使能 1: 使能                                                                       | 0x00 |
| [13]   | RW | WEventEn | 等待事件使能 0:<br>不使能 1: 使能<br>(使能后一旦当前的全局事件寄存器<br>值等于等待事件寄存器的值, dma<br>将会开始传输)                    | 0x00 |
| [12]   | RW | SEventEn | 全局事件置位使能<br>0: 不使能 1: 使能 (DMA 设置全<br>局事件寄存器)                                                  | 0x00 |
| [11:9] | RW | ChSEvent | 通道设置事件 在<br>dma 描述符都被<br>执行后会通过这个<br>寄存器设置一个规<br>定的事件。注意这<br>个设置只能在最后<br>一个队列最后一个<br>描述符的时候设置 | 0x00 |
| [8]    | RW | Wsync    | 等待最后的写完成<br>0: 所有的 DMA<br>命令推入命令队列<br>标记传输完成 1:<br>等待 DMA 所有的<br>命令写入 AXI 总线<br>标记传输完成        | 0x00 |
| [7:0]  | RW | ChWEvent | 通道等待事件 等<br>待规定的时间在<br>dma 开始传输之                                                              | 0x00 |

|  |  |  |                |  |
|--|--|--|----------------|--|
|  |  |  | 前，一个通道可以等待多个事件 |  |
|--|--|--|----------------|--|

## CNCFG

cncfg 为 DMA 通道 N 配置寄存器。

Offset Address:0x104+0x20\*N Total Reset Value:0x0

| Bits    | Access | Name        | Description                                                                             | Reset |
|---------|--------|-------------|-----------------------------------------------------------------------------------------|-------|
| [31]    | RW     | UnalignMode | 非对齐模式使能 0: 不使能<br>1: 使能 当非对齐模式使能的时候，源端和目的端的数据应该设置成 AXI 数据位宽，数据打包和解包将不会被执行。              | 0x00  |
| [30]    | RW     | WOMode      | WO 模式 0: 不使能 1: 使能 WO 模式使能时，源端数据位宽应该等于或大于 32bit，目的端数据位宽应该等于或大于 32bit，源端数据位宽和目的端数据位宽应该一样 | 0x00  |
| [29]    | RO     | reserved    | 保留                                                                                      | 0x00  |
| [28]    | RW     | ChPri       | 通道优先级配置 1: 高优先级 0: 低优先级                                                                 | 0x00  |
| [27:20] | RW     | ChGntWin    | 通道优先级窗口配置                                                                               | 0x00  |
| [19:16] | RW     | LLPCnt      | 链表指针递增计数                                                                                | 0x00  |
| [15:14] | RO     | reserved    | 保留                                                                                      | 0x00  |
| [13]    | RW     | DstHEn      | 目的端握手使能 1: 使能 (目的端使用外设接口和 dma 握手) 0: 不使能                                                | 0x00  |
| [12:9]  | RW     | DstRS       | 目的端握手通道选择 0: 使用 dma_req[0]/dma_ack[0] ...                                               | 0x00  |

|       |    |           |                                                                                 |      |
|-------|----|-----------|---------------------------------------------------------------------------------|------|
|       |    |           | 15:使用<br>dma_req[15]/dma_ack[15]                                                |      |
| [8]   | RO | reserved  | 保留                                                                              | 0x00 |
| [7]   | RW | SrcHEn    | 源端握手使能 1: 使能 (源端使用外设接口和 dma 握手) 0: 不使能                                          | 0x00 |
| [6:3] | RW | SrcRS     | 目的端握手通道选择 0:使用<br>dma_req[0]/dma_ack[0] ...<br>15:使用<br>dma_req[15]/dma_ack[15] | 0x00 |
| [2]   | RW | AbtIntMsk | 通道终止中断屏蔽 0: 关闭<br>屏蔽 1: 使能屏蔽                                                    | 0x00 |
| [1]   | RW | ErrIntMsk | 通道错误中断屏蔽 0: 关闭<br>屏蔽 1: 使能屏蔽                                                    | 0x00 |
| [0]   | RW | TCIntMsk  | 通道完成中断屏蔽 0: 关闭<br>屏蔽 1: 使能屏蔽                                                    | 0x00 |

## CNSRCA (N)

cnsrca (n) 为 DMA 通道 N 源地址寄存器。

Offset Address:0x108+0x20\*N Total Reset Value:0x0

| Bits   | Access | Name    | Description | Reset |
|--------|--------|---------|-------------|-------|
| [31:0] | RW     | SrcAddr | 源端地址寄存器     | 0x00  |

## CNDSTA (N)

cndsta (n) 为 DMA 通道 N 目的地址寄存器。

Offset Address:0x10c+0x20\*N Total Reset Value:0x0

| Bits   | Access | Name    | Description | Reset |
|--------|--------|---------|-------------|-------|
| [31:0] | RO     | DstAddr | 目的端地址寄存器    | 0x00  |



## CNLNK (N)

cnlnk (n) 为 DMA 通道 N LLP 寄存器。

Offset Address:0x110+0x20\*N Total Reset Value:0x0

| Bits   | Access | Name | Description | Reset |
|--------|--------|------|-------------|-------|
| [31:0] | RW     | LLP  | 链表指针        | 0x00  |

## CNTSIZE(N)

cntsize(n)为 DMA 通道传输大小寄存器。

Offset Address:0x114+0x20\*N Total Reset Value:0x0

| Bits    | Access | Name     | Description                     | Reset |
|---------|--------|----------|---------------------------------|-------|
| [31:22] | RO     | reserved | 保留                              | 0x00  |
| [21:0]  | RW     | Tcnt     | 传输大小 注意当非对齐模式使能的时候，传输大小固定位 byte | 0x00  |

## CNTSIZE(N)

cntsize(n)为 DMA 通道传输大小寄存器（当设置为 2D 模式时使用）。

Offset Address:0x114+0x20\*N Total Reset Value:0x0

| Bits    | Access | Name  | Description | Reset |
|---------|--------|-------|-------------|-------|
| [31:16] | RW     | YTCnt | Y 分量传输大小    | 0x00  |
| [15:0]  | RW     | XTCnt | X 分量传输大小    | 0x00  |

## CNSSDA (N)

cnssda (n) 为 DMA 通道 N 源/目的步长寄存器（2D 模式时使用）。

Offset Address:0x118+0x20\*N Total Reset Value:0x0

| Bits    | Access | Name      | Description                                                                              | Reset |
|---------|--------|-----------|------------------------------------------------------------------------------------------|-------|
| [31:16] | RW     | DstStride | 目的端步长<br>0x0000:0 ...<br>0x7fff:32767<br>0x8000:-32768<br>0x8001:-32767 ...<br>0xffff:-1 | 0x00  |

|        |    |           |                                                                                         |      |
|--------|----|-----------|-----------------------------------------------------------------------------------------|------|
| [15:0] | RW | SrcStride | 源端步长<br>0x0000:0 ...<br>0x7fff:32767<br>0x8000:-32768<br>0x8001:-32767 ...<br>0xffff:-1 | 0x00 |
|--------|----|-----------|-----------------------------------------------------------------------------------------|------|

## CNSADAEXP (N)

cnsadaexp (n) 为 DMA 通道 N 源地址和目的地址拓展位寄存器。

Offset Address:0x200+0x20\*N Total Reset Value:0x0

| Bits    | Access | Name       | Description                                           | Reset |
|---------|--------|------------|-------------------------------------------------------|-------|
| [31:16] | RW     | DstAddrExp | 目的地址拓展位 拓展目的地址[N-1:32] N 取决于硬件宏<br>FTDMAC030_AXI_AW_N | 0x00  |
| [15:0]  | RW     | SrcAddrExp | 源地址拓展位 拓展源地址[N-1:32] N 取决于硬件宏<br>FTDMAC030_AXI_AW_N   | 0x00  |

## CNLLPUSEREXP (N)

cnllpuserexp (n) 为 DMA 通道 N LLP 和用户自定义信号拓展位寄存器。

Offset Address:0x204+0x20\*N Total Reset Value:0x0

| Bits    | Access | Name      | Description                                                                         | Reset |
|---------|--------|-----------|-------------------------------------------------------------------------------------|-------|
| [31:24] | RW     | ARUserDef | 读地址通道用户自定义信号 这个域将会覆盖 AXI 读地址通道中 ARUser 的值通过硬件宏定义<br>FTDMAC030_AXI_ARUSER_N,N 不能等于 0 | 0x00  |
| [23:16] | RW     | AWUserDef | 写地址通道用户自定义信号 这个域将会覆盖 AXI 写地址通道中 AWUser 的值通过硬件宏定义<br>FTDMAC030_AXI_AWUSER_N,N 不能等于 0 | 0x00  |
| [15:0]  | RW     | LLPExp    | LLP 指针拓展位, 额外的 LLP 的拓展位 [N-1:32],通过硬件宏定义                                            | 0x00  |

|  |  |  |                    |  |
|--|--|--|--------------------|--|
|  |  |  | FTDMAC030_AXI_AW_N |  |
|--|--|--|--------------------|--|

## 1.7 定时器

### 1.7.1 概述

Timer 模块主要实现定时、计数功能，可以供操作系统用作系统时钟，也可以供应用程序用作定时和计数。系统提供 5 个 Timer。

### 1.7.2 特点

Timer 具有以下特点：

- 带可编程 8 位预分频器的 32bit/16bit 减法定时器/计数器。
- 支持 3 种计数模式：自由运行模式、周期模式和单次计数模式。
- 有 2 种载入计数初值的方法，分别通过 `TIMERx_LOAD` 和 `TIMERx_BGLOAD` 寄存器实现。
- 当前的计数值可随时读取。
- 当计数值减到 0 时会产生一个中断。

### 1.7.3 功能描述

Timer 基于一个 32bit/16bit（可配置）减法计数器。计数器的值在每个计数时钟的上升沿减 1。当计数值递减到零，Timer 将产生一个中断。

Timer 有以下 3 种计数模式：

- 自由运行模式  
定时器持续计数，当计数值减到 0 时又自动回转到其最大值，并继续计数。当计数长度为 32bit 时，最大值为 `0xFFFF_FFFF`。当计数长度为 16bit 时，最大值为 `0xFFFF`。在自由模式下，也可以载入计数值，并立即从载入值递减计数，但计到 0 时回转到其最大值。
- 周期模式  
定时器持续计数，当计数值减到 0 时从 `TIMERx_BGLOAD` 寄存器中再次载入初值并继续计数。
- 单次计数模式

向定时器中载入计数初值。当定时器的计数值减到 0 时就停止计数，直到重新被载入新值且定时器处于使能状态，才再次开始计数。

每个 Timer 具有一个预分频计数器 (prescaler)，可将其工作时钟在 Timer 内部再次进行 1 分频、16 分频或 256 分频。进一步提高计数时钟频率的选择灵活性。

对定时器载入计数初值的方法如下：

- 通过写 `TIMERx_LOAD` 寄存器可对定时器载入计数初值。当定时器处于工作状态时，如果向 `TIMERx_LOAD` 寄存器写入值，会导致定时器立刻从新值开始重新计数。适用于所有计数模式。
- 通过写 `TIMERx_BGLOAD` 寄存器可以设定周期计数模式的计数周期。写该寄存器不会立刻影响定时器的当前计数，定时器会继续计数直到计数值减到 0。然后载入 `TIMERx_BGLOAD` 寄存器中的新值开始计数。

#### 1.7.4 TIMER 寄存器概览

TIMER0 基地址：0x1200\_0000

TIMER1 基地址：0x1200\_0100

TIMER2 基地址：0x1200\_0200

TIMER3 基地址：0x1200\_0300

TIMER4 基地址：0x1200\_0400

表1-18 Timer 寄存器概览

| 基地址   | 名称             | 描述          |
|-------|----------------|-------------|
| 0x000 | TIMERx_LOAD    | 计数初值寄存器     |
| 0x004 | TIMERx_VALUE   | 当前计数值寄存器    |
| 0x008 | TIMERx_CONTROL | Timer 控制寄存器 |
| 0x00C | TIMERx_INTCLR  | 中断清除寄存器     |
| 0x010 | TIMERx_RIS     | 原始中断寄存器     |
| 0x014 | TIMERx_MIS     | 屏蔽后中断寄存器    |
| 0x018 | TIMERx_BGLOAD  | 周期模式计数初值寄存器 |

## 1.7.5 TIMER 寄存器描述

### TIMERx\_LOAD

TIMERx\_LOAD 为计数初值寄存器。用来配置定时器的计数初值。

 说明

向 TIMERx\_LOAD 寄存器写入的最小有效值为 1。

当向 TIMERx\_LOAD 写 0 时，Dual-Timer 将会立刻产生 1 个中断。

TIMERx\_LOAD 和 TIMERx\_BGLOAD 的区别：

寄存器和 TIMERx\_LOAD 寄存器都写入值，则在被 TIMCLKENx 使能的 TIMCLK 的一个上升沿当前计数值首先更新为 TIMERx\_LOAD 的写入值。由于向 TIMERx\_LOAD 寄存器写入值时，TIMERx\_BGLOAD 的值也会被覆盖，所以读 TIMERx\_BGLOAD，返回的值为 TIMERx\_LOAD 与 TTIMERx\_BGLOAD 中最晚被写入的寄存器的值。当定时器处于周期模式且计数值减到 0 时，将从 TIMERx\_BGLOAD 寄存器中再次载入初值并继续计数。

Offset Address: 0x000 Total Reset Value: 0x0000\_0000

| Bits   | Access | Name        | Description  | Reset      |
|--------|--------|-------------|--------------|------------|
| [31:0] | RW     | timer0_load | Timer0 的计数初值 | 0x00000000 |

### TIMERx\_VALUE

TIMERx\_VALUE 为当前计数值寄存器。用于给出正在递减的计数器的当前值。

当向 TIMERx\_LOAD 寄存器的写操作发生后，TIMERx\_VALUE 在 PCLK 时钟域立刻反映出计数器的新载入值，不用等到下一个被 TIMCLKENx 使能的 TIMCLK 时钟沿到来。

 说明

向 TIMERx\_LOAD 寄存器写入的最小有效值为 1。

该定时器以前处于 32bit 模式，并且自从进入 16bit 模式后 TIMERx\_LOAD 从未被写过，则 TIMERx\_VALUE 寄存器的高 16bit 可能具有非零值。

Offset Address: 0x004 Total Reset Value: 0xFFFF\_FFFF

| Bits   | Access | Name         | Description       | Reset      |
|--------|--------|--------------|-------------------|------------|
| [31:0] | RO     | timer0_value | 正在递减的 Timer0 的当前值 | 0xFFFFFFFF |

## TIMERx\_CONTROL

TIMERx\_VALUE 为 TIMER 控制寄存器。

 说明

当选择用周期模式进行计数时，需要将 TIMERx\_CONTROL[timermode]置 1、TIMERx\_CONTROL[oneshot]置 0。

Offset Address: 0x008 Total Reset Value: 0xFFFF\_FFFF

| Bits   | Access | Name      | Description                                                                                                                                                           | Reset    |
|--------|--------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| [31:8] | -      | reserved  | 保留                                                                                                                                                                    | 0x000000 |
| [7]    | RW     | timeren   | 定时器使能<br>0: Timer 禁止;<br>1: Timer 使能。                                                                                                                                 | 0x0      |
| [6]    | RW     | timermode | 定时器的计数模式<br>0: 自由运行模式;<br>1: 周期模式                                                                                                                                     | 0x0      |
| [5]    | RW     | intenable | TIMERx_RIS 中断屏蔽<br>0: 屏蔽该中断;<br>1: 不屏蔽该中断。                                                                                                                            | 0x0      |
| [4]    | -      | reserved  | 保留                                                                                                                                                                    | 0x       |
| [3:2]  | RW     | timepre   | 该字段用于设置 Timer 的预分频因子<br>00: 不经过预分频, 时钟频率除以 1;<br>01: 4 级预分频, 将 Timer 时钟频率除以 16;<br>10: 8 级预分频, 将 Timer 时钟频率除以 256;<br>11: 未定义, 若设为该值, 相当于 8 级预分频, 将 Timer 时钟频率除以 256。 | 0x0      |
| [1]    | RW     | timersize | 选择 16bit/32bit 计数器操作模                                                                                                                                                 | 0x0      |

|     |    |         |                                                           |     |
|-----|----|---------|-----------------------------------------------------------|-----|
|     |    |         | 式<br>0: 16bit 计数器;<br>1: 32bit 计数器。                       |     |
| [0] | RW | oneshot | 选择计数模式为单次计数模式还是周期计数模式。<br>0: 周期计数模式或自由运行模式;<br>1: 单次计数模式。 | 0x0 |

## TIMERx\_INTCLR

TIMERx\_INTCLR 为中断清除寄存器。对该寄存器的任何写操作都会清除相应计数器。

 说明

本寄存器是只写寄存器，写进去任意值，都会引起 Timer 清中断，内部并不记忆写入的值，无复位值。

Offset Address: 0x00C Total Reset Value: 0x0000\_0000

| Bits   | Access | Name          | Description           | Reset      |
|--------|--------|---------------|-----------------------|------------|
| [31:0] | WO     | timer0_intclr | 写该寄存器，清除 Timer0 的中断输出 | 0x00000000 |

## TIMERx\_INTRAWA

TIMERx\_INTRAWA 为原始中断寄存器。

Offset Address: 0x010 Total Reset Value: 0x0000\_0000

| Bits   | Access | Name           | Description                              | Reset    |
|--------|--------|----------------|------------------------------------------|----------|
| [31:1] | -      | reserved       | 保留                                       | 0x000000 |
| [0]    | RO     | timer0_intrawa | Timer0 的原始中断状态<br>0: 未产生中断;<br>1: 已产生中断。 | 0x0      |

## TIMERx\_INTA

TIMERx\_INTA 为屏蔽后中断寄存器。

Offset Address: 0x014 Total Reset Value: 0x0000\_0000

| Bits   | Access | Name        | Description                                | Reset    |
|--------|--------|-------------|--------------------------------------------|----------|
| [31:1] | -      | reserved    | 保留                                         | 0x000000 |
| [0]    | RO     | timer0_inta | 屏蔽后的 Timer0 的中断状态。<br>0: 中断无效;<br>1: 中断有效。 | 0x0      |

## TIMERx\_BGLOAD

IMERx\_BGLOAD 为周期模式计数初值寄存器。

TIMERx\_BGLOAD 寄存器中包含了定时器的计数初值。该寄存器用于在周期模式下，当定时器的计数值递减到 0 时重新载入计数初值。

Offset Address: 0x018 Total Reset Value: 0x0000\_0000

| Bits   | Access | Name          | Description  | Reset      |
|--------|--------|---------------|--------------|------------|
| [31:0] | RW     | timer0_bgload | Timer0 的计数初值 | 0x00000000 |

## 1.8 看门狗

### 1.8.1 概述

看门狗 WatchDog 用于系统异常情况下，一定时间内发出复位信号，以复位整个系统。系统提供 3 个 WatchDog 模块。

### 1.8.2 特点

WatchDog 具备以下特点：

- 内部具有一个 32bit 减法计数器。
- 支持超时时间间隔（即计数初值）可配置。
- 支持寄存器锁定，防止寄存器被误改。
- 支持超时中断产生。
- 支持复位信号产生。



- 支持调试模式。

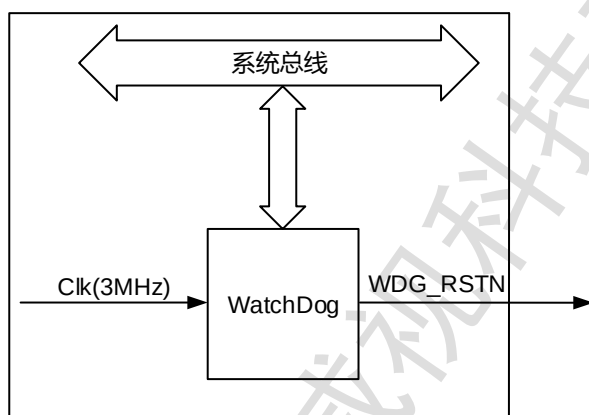
### 1.8.3 功能描述

#### 应用框图

系统通过系统总线给 WatchDog 配置寄存器参数值。WatchDog 定时发出中断请求给系统，并在系统没有响应中断的情况下（如：死机），发出 WDG\_RSTN 复位信号，使系统复位，达到监控系统运行的目的。

WatchDog 应用框图如图 1-7 所示。

图1-7 WatchDog 应用框图



#### 功能原理

WatchDog 的运行基于 1 个 32bit 减法计数器，计数初值由寄存器载入。在 WatchDog 时钟使能情况下，计数值在每个计数时钟的上升沿减 1。当计数值递减到 0，WatchDog 将产生一个中断。然后在下一个计数时钟上升沿，计数器又从寄存器中重新载入计数初值，开始递减计数。

如果计数器的计数值第二次计数递减到 0 时，CPU 还没有清除 WatchDog 中断，则 WatchDog 将发出复位信号 WDG\_RSTN，计数器停止计数。

根据实际需要，可通过配置 WDG\_CONTROL 使能或者禁止 WatchDog 产生中断和复位信号：

- 当禁止产生中断时，计数器将停止计数。
- 当重新开启中断时，WatchDog 将从 WDG\_LOAD 的设定值开始计数，而不是从计数器上次停止时的计数值开始计数。在中断到来之前，可以重新载入初值。
- 通过配置 WDG\_LOCK 寄存器，可以禁止对 WatchDog 内部寄存器进行写操作：

- 向 WDG\_LOCK 写入 0x1ACC\_E551，可以打开所有 WatchDog 寄存器的写权限。
- 向 WDG\_LOCK 寄存器写入其他任何值，可以关闭所有 WatchDog 寄存器 (WDG\_LOCK 寄存器除外) 的写权限。

该特性保护 WatchDog 的寄存器不被软件错误修改，从而使得在异常情况下，WatchDog 不至于被软件错误地中止操作。

在调试模式下，WatchDog 自动关闭，以防止干扰正常的调试操作。

## 1.8.4 工作方式

### 计数时钟频率配置

WatchDog 计数时钟为 24MHz 时钟。

WatchDog 计数时间为  $T_{WDG}$ ：

$$T_{WDG} = Value_{WDG\_LOAD} \times \left( \frac{1}{f_{clk}} \right)$$

 说明

其中上式中，各参数代表的意思分别是：

- $T_{WDG}$  表示 WatchDog 计数时间；
- $Value_{WDG\_LOAD}$  表示 WatchDog 计数初值；
- $f_{clk}$  表示 WatchDog 计数时钟频率。

WatchDog 的计数时间范围值为 0s ~ 178s。

### 系统初始化配置

WatchDog 初始化并启动其运行。WatchDog 的初始化过程如下：

步骤 1 写寄存器 WDG\_LOAD，设定计数初值。

步骤 2 写寄存器 WDG\_CONTROL，打开中断屏蔽并启动 WatchDog 计数。

步骤 3 写寄存器 WDG\_LOCK，给 WatchDog 上锁，防止软件错误修改 WatchDog 的配置。

----结束

## 中断处理过程

收到 WatchDog 发出的中断后，应及时清除其中断状态，并使其载入计数初值重新开始计数。WatchDog 中断处理的过程如下：

- 步骤 1 向 WDG\_LOCK 写 0x1ACC\_E551，为 WatchDog 开锁。
- 步骤 2 写寄存器 WDG\_INTCLR，清除 WatchDog 的中断状态，同时也使 WatchDog 自动载入计数初值重新开始计数。
- 步骤 3 向寄存器 WDG\_LOCK 写入 0x1ACC\_E551 以外的任何值，给 WatchDog 上锁。

---结束

## 关闭 WatchDog

向寄存器 WDG\_CONTROL[inten]控制位写入 0 或 1 控制 WatchDog 的状态：

- 0：关闭 WatchDog；
- 1：打开 WatchDog。

## 1.8.5 WDG 寄存器概览

WatcgDog0 基地址：0x1203\_0000

WatcgDog1 基地址：0x1203\_0020

WatcgDog2 基地址：0x1203\_0040

WatchDog 寄存器概览如表 1-19 所示。

表1-19 WatchDog 寄存器概览

| 偏移地址   | 名称          | 描述        |
|--------|-------------|-----------|
| 0x0000 | WDG_LOAD    | 计数初值寄存器   |
| 0x0004 | WDG_VALUE   | 计数器当前值寄存器 |
| 0x0008 | WDG_CONTROL | 控制寄存器     |
| 0x000C | WDG_INTCLR  | 中断清除寄存器   |
| 0x0010 | WDG_RIS     | 原始中断寄存器   |
| 0x0014 | WDG_MIS     | 屏蔽后中断寄存器  |

|        |          |          |
|--------|----------|----------|
| 0x0C00 | WDG_LOCK | LOCK 寄存器 |
|--------|----------|----------|

## 1.8.6 WDG 寄存器描述

### WDG\_LOAD

WDG\_LOAD 为计数初值寄存器。

用来配置 WatchDog 内部计数器的计数初值。

Offset Address: 0x0000 Total Reset Value: 0xFFFF\_FFFF

| Bits   | Access | Name     | Description | Reset      |
|--------|--------|----------|-------------|------------|
| [31:0] | RW     | wdg_load | 计数初值        | 0xFFFFFFFF |

### WDG\_VALUE

WDG\_value 为计数器当前值寄存器。

读出 WatchDog 内部计数器的当前计数值。

Offset Address: 0x0004 Total Reset Value: 0xFFFF\_FFFF

| Bits   | Access | Name      | Description     | Reset      |
|--------|--------|-----------|-----------------|------------|
| [31:0] | RO     | wdg_value | WatchDog 计数器当前值 | 0xFFFFFFFF |

### WDG\_CONTROL

WDG\_CONTROL 为控制寄存器。

来控制 WatchDog 打开/关闭、中断和复位功能。

Offset Address: 0x0008 Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description                            | Reset      |
|--------|--------|----------|----------------------------------------|------------|
| [31:2] | -      | reserved | 保留                                     | 0x00000000 |
| [1]    | RW     | resen    | WatchDog 复位信号输出使能。<br>0: 禁止;<br>1: 使能。 | 0x0        |

|     |    |       |                                                                                             |     |
|-----|----|-------|---------------------------------------------------------------------------------------------|-----|
| [0] | RW | inten | WatchDog 中断信号输出使能<br>0: 计数器停止计数, 计数值保持当前值不变, WatchDog 被关闭;<br>1: 既启动计数器又使能中断, WatchDog 被启动。 | 0x0 |
|-----|----|-------|---------------------------------------------------------------------------------------------|-----|

## WDG\_INTCLR

WDG\_INTCLR 为中断清除寄存器。

用来清除 WatchDog 中断, 使 WatchDog 重新载入初值进行计数。

本寄存器是只写寄存器, 写进去任意值, 都会引起 WatchDog 清中断, 内部并不记忆写入的值, 无复位值。

Offset Address: 0x000C Total Reset Value: 0x0000\_0000

| Bits   | Access | Name       | Description                                                       | Reset       |
|--------|--------|------------|-------------------------------------------------------------------|-------------|
| [31:0] | WO     | wdg_intclr | 对该寄存器写入任意值均可清除 WatchDog 的中断, 并使 WatchDog 从寄存器 WDG_LOAD 中重新载入初值计数。 | 0x0000_0000 |

## WDG\_RIS

WDG\_RIS 为原始中断寄存器。用来反映 WatchDog 原始中断状态。

Offset Address: 0x0010 Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description                                                    | Reset      |
|--------|--------|----------|----------------------------------------------------------------|------------|
| [31:1] | -      | reserved | 保留                                                             | 0x00000000 |
| [0]    | RO     | wdg_ris  | WatchDog 原始中断状态, 当计数器的值递减到 0 是, 该位置为 1<br>0: 未产生中断<br>1: 已产生中断 | 0x0        |

## WDG\_MIS

WDG\_RIS 为屏蔽后中断寄存器。

用来反映 WatchDog 屏蔽后的中断状态。

Offset Address: 0x0014 Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description                                       | Reset      |
|--------|--------|----------|---------------------------------------------------|------------|
| [31:1] | -      | reserved | 保留                                                | 0x00000000 |
| [0]    | RO     | wdg_ris  | WatchDog 屏蔽后的中断状态。<br>0: 未产生中断或者中断被屏蔽<br>1: 已产生中断 | 0x0        |

## WDG\_LOCK

WDG\_LOCK 为 LOCK 寄存器。用来控制 WatchDog 寄存器的读写权限。

Offset Address: 0x0C00 Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description                                                                                                                                  | Reset       |
|--------|--------|----------|----------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| [31:0] | RW     | wdg_lock | 有寄存器的写 0x1ACC_E551, 可打开所有寄存器的写权限;<br>向该寄存器写入其他值则关闭写权限。<br>读该寄存器返回加锁的状态而不是写入该寄存器的值:<br>0x0000_0000: 允许写访问 (未加锁);<br>0x0000_0001: 禁止写访问 (已加锁)。 | 0x0000_0000 |

## 1.9 实时时钟

### 1.9.1 概述

实时时钟 RTC (Real Time Clock) 用于实现时间显示和定时报警功能。

## 1.9.2 特点

RTC 具有以下特点:

- RTC 具有 1 个 16bit 的天加法计数器, 5bit 的小时计数器, 6bit 的分计数器, 6bit 的秒计数器和 7bit 的 10ms 计数器;
- RTC 具有 1 个分频模块, 分频参数可配置;
- RTC 计数器时钟为 100Hz;
- RTC 计数器初值可配置;
- RTC 计数器比较值可配置;
- 支持超时中断产生与低压中断;
- 支持软复位;
- 64bit 用户寄存器, 供用户保存数据;
- 支持电池低压检测。

## 1.9.3 功能描述

RTC 的运行基于 1 个共同 40bit 的 (天, 时, 分, 秒, 10ms) 加法计数器, 计数初值由寄存器 RTC\_LR\_10MS、RTC\_LR\_S、RTC\_LR\_M、RTC\_LR\_H、RTC\_LR\_D\_L、RTC\_LR\_D\_H 载入。在当计数值递加到寄存器与 RTC\_MR\_10MS、RTC\_MR\_S、RTC\_MR\_M、RTC\_MR\_H、RTC\_MR\_D\_L、RTC\_MR\_D\_H 寄存器值相等时, RTC 将产生一个中断, 然后在下一个计数时钟上升沿, 计数器继续递加计数。根据实际应用需要, 可通过配置 RTC\_IMSC 使能或者禁止 RTC 产生中断信号。此时, 存在以下两种情况:

当禁止产生中断时, RTC 计数器继续递加计数, 将不会对外产生中断, 在 RTC\_MSC\_INT 中显示屏蔽后中断的状态, 在 RTC\_RAW\_INT 中显示原始中断状态。

当重新开启中断时, RTC 计数器仍然继续递加计数, 当计数值递加到与 RTC\_MR\_10MS、RTC\_MR\_S、RTC\_MR\_M、RTC\_MR\_H、RTC\_MR\_D\_L、RTC\_MR\_D\_H 寄存器值相等时, RTC 将产生一个中断。

RTC 的计数时钟采用的是 100Hz 时钟, 同时提供 16bit 的天计数, 便于通过天计数值转换为具体的年、月、日。

## 1.9.4 工作方式

### 计数时钟频率

RTC 采用 100Hz 时钟进行计数，计数的最大时间为：

$$T_{\text{RTC}} = 2^{16} = 65536 \text{ 天}$$

### 软复位

通过配置 RTC 复位寄存器 RTC\_POR\_N，可以实现对 RTC 的单独软复位。软复位后各个 RTC 配置寄存器的值均恢复为默认值，因此软复位后需要重新对这些寄存器进行初始化配置。

软复位操作步骤如下：

步骤 1 向复位寄存器 RTC\_POR\_N 写 0 进行复位。

步骤 2 等待 30ms 完成复位。

----结束

### RTC 初始化

RTC 初始化操作步骤如下：

步骤 1 配置 SDM\_COEF\_OUSIDE\_L，初始 RTC 的分频系数，默认配置 0x20(默认晶振频率 32768Hz)，其他晶振频率根据寄存器表单公式进行配置。

步骤 2 配置 RTC\_IMSC，设置 RTC 中断屏蔽位。

步骤 3 配置 RTC\_MR\_10MS、RTC\_MR\_S、RTC\_MR\_M、RTC\_MR\_H、RTC\_MR\_D\_L、RTC\_MR\_D\_H，设置 RTC 比较值。

步骤 4 配置 RTC\_LR\_10MS、RTC\_LR\_S、RTC\_LR\_M、RTC\_LR\_H、RTC\_LR\_D\_L、RTC\_LR\_D\_H，设置 RTC 计数初始值。

步骤 5 配置 RTC\_LORD 为 1，让 RTC 计数初始值加载 RTC 计数器中。

步骤 6 等待 10ms。

步骤 7 RTC 按照 100Hz 的计数时钟频率，从 RTC\_LR\_10MS、RTC\_LR\_S、RTC\_LR\_M、RTC\_LR\_H、RTC\_LR\_D\_L、RTC\_LR\_D\_H 中的值开始计数，当计数到



RTC\_MR\_10MS、RTC\_MR\_S、RTC\_MR\_M、RTC\_MR\_H、RTC\_MR\_D\_L、RTC\_MR\_D\_H 中的值时，将根据 RTC\_IMSC 的设置，决定是否产生中断。

----结束

## RTC 时间配置

RTC 时间配置操作步骤如下：

- 步骤 1 配置 RTC\_LR\_10MS、RTC\_LR\_S、RTC\_LR\_M、RTC\_LR\_H、RTC\_LR\_D\_L、RTC\_LR\_D\_H，设置 RTC 计数初始值。
- 步骤 2 配置 RTC\_LORD 为 0x1，让 RTC 计数初始值加载 RTC 计数器中。
- 步骤 3 回读 RTC\_LORD 的值，若不为 0x0，重复步骤 2；若为 0x0，向下执行。

----结束

## RTC 时间读取

RTC 时间读取操作步骤如下：

- 步骤 1 配置 RTC\_LORD 为 0x2。
- 步骤 2 回读 RTC\_LORD 的值，若不为 0x0，重复步骤 2；若为 0x0，向下执行步骤 3。
- 步骤 3 从 RTC\_10MS\_COUNT、RTC\_S\_COUNT、RTC\_M\_COUNT、RTC\_H\_COUNT、RTC\_D\_COUNT\_L、RTC\_D\_COUNT\_H 寄存器读取时间值。

----结束

## 中断处理

系统收到 RTC 发出的中断后，表示定时时间到，用户可以执行相应的自定义操作，RTC 计数器仍然保持递加计数。RTC 中断处理的过程如下：

- 步骤 1 配置 RTC\_INT\_CLR 为 1，清除 RTC 的中断状态。
- 步骤 2 如果需要继续设置定时时间，则向寄存器 RTC\_MR\_10MS、RTC\_MR\_S、RTC\_MR\_M、RTC\_MR\_M、RTC\_MR\_D\_L、RTC\_MR\_D\_H 写入新的比较值。

----结束

## RTC 寄存器访问

RTC 的寄存器是在 RTC 模块内部的，并不在总线上。总线上的 RTC 寄存器只是提供访问 RTC 内部寄存器的通路。

写 RTC 内部寄存器的步骤如下：

- 步骤 1 配置 SPI\_CLK\_DIV；假如总线时钟为 50MHz，期望 SPI 的时钟为 5MHz，则 SPI\_CLK\_DIV[spi\_clk\_div]应该配置为  $(50\text{MHz}/5\text{MHz}) / 2 - 1 = 4 = 0x04$ 。（如果已经配置，并且不想更改 SPI 的时钟频率则可以省略这一步骤）。
- 步骤 2 读 SPI\_RW，直到 SPI\_RW [31]=0 为止。
- 步骤 3 配置 SPI\_RW；假如要对 RTC\_MR\_10MS 写入 0x10，该寄存器在 RTC 内部的地址为 0x06，所以 SPI\_RW 应该配置为 0x01060010（spi\_start=1, spi\_rw=0, spi\_add=0x06, spi\_wdata=0x10）。

----结束

读 RTC 内部寄存器的步骤如下：

- 步骤 1 配置 SPI\_CLK\_DIV；假如总线时钟为 50MHz，期望 SPI 的时钟为 5MHz，则 spi\_clk\_div 应该配置为  $(50\text{MHz}/5\text{MHz}) / 2 - 1 = 4 = 0x04$ 。（如果已经配置过了，并且不想更改 spi 的时钟频率则可以省略这一步骤）。
- 步骤 2 读 SPI\_RW，直到 SPI\_RW [31]=0 为止。
- 步骤 3 配置 SPI\_RW；假如要对读 RTC\_MR\_10MS，该寄存器在 RTC 内部的地址为 0x06，所以 SPI\_RW 应该配置为 0x01860000（spi\_start=1, spi\_rw=1, spi\_add=0x06）。
- 步骤 4 读 SPI\_RW，直到 SPI\_RW [31]=0 为止。则 SPI\_RW [15:8]为 RTC\_MR\_10MS 的读回值。

----结束

### 1.9.5 RTC 寄存器概览

RTC 基地址：120E\_0000

表1-20 RTC 寄存器概览

| 偏移地址 | 名称             | 描述              |
|------|----------------|-----------------|
| 0x01 | RTC_10MS_COUNT | RTC 10ms 计数值寄存器 |

|      |               |                   |
|------|---------------|-------------------|
| 0x02 | RTC_S_COUNT   | RTC 秒计数值寄存器       |
| 0x03 | RTC_M_COUNT   | RTC 分计数值寄存器       |
| 0x04 | RTC_H_COUNT   | RTC 时计数值寄存器       |
| 0x05 | RTC_D_COUNT_L | RTC 天计数值低 8 位寄存器  |
| 0x06 | RTC_D_COUNT_H | RTC 天计数值高 8 位寄存器  |
| 0x07 | RTC_MR_10MS   | RTC 10ms 定时值寄存器   |
| 0x08 | RTC_MR_S      | RTC 秒定时值寄存器       |
| 0x09 | RTC_MR_M      | RTC 分定时值寄存器       |
| 0x0A | RTC_MR_H      | RTC 时定时值寄存器       |
| 0x0B | RTC_MR_D_L    | RTC 天定时值的低 8 位寄存器 |
| 0x0C | RTC_MR_D_H    | RTC 天定时值的高 8 位寄存器 |
| 0x0D | RTC_LR_10MS   | RTC 10ms 设置值寄存器   |
| 0x0E | RTC_LR_S      | RTC 秒设置值寄存器       |
| 0x0F | RTC_LR_M      | RTC 分设置值寄存器       |
| 0x10 | RTC_LR_H      | RTC 时设置值寄存器       |
| 0x11 | RTC_LR_D_L    | RTC 天设置值的低 8 位寄存器 |
| 0x12 | RTC_LR_D_H    | RTC 天设置值的高 8 位寄存器 |
| 0x13 | RTC_LORD      | RTC 设置值使能加载寄存器    |
| 0x14 | RTC_IMSC      | RTC 中断使能寄存器       |
| 0x15 | RTC_INT_CLR   | RTC 中断清除寄存器       |
| 0x16 | RTC_MSC_INT   | RTC mask 中断状态寄存器  |
| 0x17 | RTC_RAW_INT   | RTC 原始中断状态寄存器     |
| 0x18 | RTC_CLK       | RTC 输出时钟选择寄存器     |
| 0x19 | RTC_POR_N     | RTC 复位控制寄存器       |
| 0x1A | RTC_UV_CTRL   | RTC 内部低压检测控制寄存器   |

|      |                   |                                                                                                     |
|------|-------------------|-----------------------------------------------------------------------------------------------------|
| 0x1B | RTC_ANA_CTRL      | "RTC 振荡电流控制寄存器（受 RTC 锁定寄存器锁定，只有当 {RTC_LOCK3, RTC_LOCK2, RTC_LOCK1, RTC_LOCK0}=0x5A5AABCD 时才可以被修改。）" |
| 0x1C | RTCLDO_SEL        | 新增：输出到模拟电路的 RTC_AVDD09 电压控制                                                                         |
| 0x1D | RG_BG_TRIM18      | 新增：输出到模拟电路的 bg 电压控制                                                                                 |
| 0x50 | SDM_COEF_OUSIDE_H | 外部分频系数高 4 位寄存器                                                                                      |
| 0x51 | SDM_COEF_OUSIDE_L | 外部分频系数低 8 位寄存器                                                                                      |
| 0x52 | USER_REGISTER1    | 64bit 用户使用寄存器 1                                                                                     |
| 0x53 | USER_REGISTER2    | 64bit 用户使用寄存器 2                                                                                     |
| 0x54 | USER_REGISTER3    | 64bit 用户使用寄存器 3                                                                                     |
| 0x55 | USER_REGISTER4    | 64bit 用户使用寄存器 4                                                                                     |
| 0x56 | USER_REGISTER5    | 64bit 用户使用寄存器 5                                                                                     |
| 0x57 | USER_REGISTER6    | 64bit 用户使用寄存器 6                                                                                     |
| 0x58 | USER_REGISTER7    | 64bit 用户使用寄存器 7                                                                                     |
| 0x59 | USER_REGISTER8    | 64bit 用户使用寄存器 8                                                                                     |
| 0x64 | RTC_LOCK0         | 32bit RTC 锁定寄存器 0                                                                                   |
| 0x65 | RTC_LOCK1         | 32bit RTC 锁定寄存器 1                                                                                   |
| 0x66 | RTC_LOCK2         | 32bit RTC 锁定寄存器 2                                                                                   |
| 0x67 | RTC_LOCK3         | 32bit RTC 锁定寄存器 3                                                                                   |

## 1.9.6 RTC 寄存器描述

### RTC\_10MS\_COUNT

rtc\_10ms\_count 为 RTC 10ms 计数值寄存器。

Offset Address:0x5    Total Reset Value:0x00

| Bits  | Access | Name           | Description                                    | Reset |
|-------|--------|----------------|------------------------------------------------|-------|
| [7]   | RO     | reserved       | 保留                                             | 0x0   |
| [6:0] | RO     | rtc_10ms_count | RTC 10ms 计数器值，表示当前计时的 10ms 时间个数。取值范围值为 0 ~ 99。 | 0x00  |

### RTC\_S\_COUNT

rtc\_s\_count 为 RTC 秒计数值寄存器。

Offset Address:0x6    Total Reset Value:0x00

| Bits  | Access | Name        | Description                          | Reset |
|-------|--------|-------------|--------------------------------------|-------|
| [7:6] | RO     | reserved    | 保留。                                  | 0x0   |
| [5:0] | RO     | rtc_s_count | RTC 秒计数器值，表示当前计时的秒时间个数。取值范围为 0 ~ 59。 | 0x00  |

### RTC\_M\_COUNT

rtc\_m\_count 为 RTC 分计数值寄存器。

Offset Address:0x7    Total Reset Value:0x00

| Bits  | Access | Name        | Description                          | Reset |
|-------|--------|-------------|--------------------------------------|-------|
| [7:6] | RO     | reserved    | 保留。                                  | 0x0   |
| [5:0] | RO     | rtc_m_count | RTC 分计数器值，表示当前计时的分时间个数。取值范围为 0 ~ 59。 | 0x00  |

### RTC\_H\_COUNT

rtc\_h\_count 为 RTC 时计数值寄存器。

Offset Address:0x8    Total Reset Value:0x00

| Bits  | Access | Name     | Description | Reset |
|-------|--------|----------|-------------|-------|
| [7:5] | RO     | reserved | 保留          | 0x0   |

|       |    |             |                                       |      |
|-------|----|-------------|---------------------------------------|------|
| [4:0] | RO | rtc_h_count | RTC 时计数器值，表示当前计时的小时时间个数。取值范围为 0 ~ 23。 | 0x00 |
|-------|----|-------------|---------------------------------------|------|

## RTC\_D\_COUNT\_L

rtc\_d\_count\_l 为 RTC 天计数值低 8 位寄存器。

Offset Address:0x9    Total Reset Value:0x00

| Bits  | Access | Name          | Description                                                      | Reset |
|-------|--------|---------------|------------------------------------------------------------------|-------|
| [7:0] | RO     | rtc_d_count_l | RTC 天计数器值的低 8 位，与 RTC_D_COUNT_H 配合，表示当前计时的天时间个数，取值范围为 0 ~ 65535。 | 0x00  |

## RTC\_D\_COUNT\_H

rtc\_d\_count\_h 为 RTC 天计数值高 8 位寄存器。

Offset Address:0xa    Total Reset Value:0x00

| Bits  | Access | Name          | Description                                                      | Reset |
|-------|--------|---------------|------------------------------------------------------------------|-------|
| [7:0] | RO     | rtc_d_count_h | RTC 天计数器值的高 8 位，与 RTC_D_COUNT_L 配合，表示当前计时的天时间个数，取值范围为 0 ~ 65535。 | 0x00  |

## RTC\_MR\_10MS

rtc\_mr\_10ms 为 RTC 10ms 定时值寄存器。

Offset Address:0xb    Total Reset Value:0x7F

| Bits  | Access | Name        | Description                                | Reset |
|-------|--------|-------------|--------------------------------------------|-------|
| [7]   | RO     | reserved    | 保留。                                        | 0x0   |
| [6:0] | RW     | rtc_mr_10ms | RTC 10ms 定时值，表示定时的 10ms 时间值。取值范围值为 0 ~ 99。 | 0x7F  |

## RTC\_MR\_S

rtc\_mr\_s 为 RTC 秒定时值寄存器。

Offset Address:0xc    Total Reset Value:0x3F

| Bits  | Access | Name     | Description | Reset |
|-------|--------|----------|-------------|-------|
| [7:6] | RO     | reserved | 保留。         | 0x0   |

|       |    |          |                                       |      |
|-------|----|----------|---------------------------------------|------|
| [5:0] | RW | rtc_mr_s | RTC 秒定时值，表示定时的秒时间值。<br>取值范围值为 0 ~ 59。 | 0x3F |
|-------|----|----------|---------------------------------------|------|

## RTC\_MR\_M

rtc\_mr\_m 为 RTC 分定时值寄存器。

Offset Address:0xd    Total Reset Value:0x3F

| Bits  | Access | Name     | Description                            | Reset |
|-------|--------|----------|----------------------------------------|-------|
| [7:6] | RO     | reserved | 保留                                     | 0x0   |
| [5:0] | RW     | rtc_mr_m | RTC 时定时值，表示定时的分钟时间值。<br>取值范围值为 0 ~ 59。 | 0x3F  |

## RTC\_MR\_H

rtc\_mr\_h 为 RTC 时定时值寄存器。

Offset Address:0xe    Total Reset Value:0x1F

| Bits  | Access | Name     | Description                            | Reset |
|-------|--------|----------|----------------------------------------|-------|
| [7:5] | RO     | reserved | 保留                                     | 0x0   |
| [4:0] | RW     | rtc_mr_h | RTC 时定时值，表示定时的小时时间值。<br>取值范围值为 0 ~ 23。 | 0x1F  |

## RTC\_MR\_D\_L

rtc\_mr\_d\_l 为 RTC 天定时值的低 8 位寄存器。

Offset Address:0xf    Total Reset Value:0xFF

| Bits  | Access | Name       | Description                                                    | Reset |
|-------|--------|------------|----------------------------------------------------------------|-------|
| [7:0] | RW     | rtc_mr_d_l | RTC 天定时值低 8 位，与 RTC_MR_D_H<br>配合，表示定时的天时间值，取值范围为<br>0 ~ 65535。 | 0xFF  |

## RTC\_MR\_D\_H

rtc\_mr\_d\_h 为 RTC 天定时值的高 8 位寄存器。

Offset Address:0x10    Total Reset Value:0xFF

| Bits  | Access | Name       | Description                | Reset |
|-------|--------|------------|----------------------------|-------|
| [7:0] | RW     | rtc_mr_d_h | RTC 天定时值高 8 位，与 RTC_MR_D_L | 0xFF  |

|  |  |  |                               |  |
|--|--|--|-------------------------------|--|
|  |  |  | 配合，表示定时的天时间值，取值范围为 0 ~ 65535。 |  |
|--|--|--|-------------------------------|--|

## RTC\_LR\_10MS

rtc\_lr\_10ms 为 RTC 10ms 设置值寄存器。

Offset Address:0x11 Total Reset Value:0x00

| Bits  | Access | Name        | Description                                   | Reset |
|-------|--------|-------------|-----------------------------------------------|-------|
| [7]   | RO     | reserved    | 保留。                                           | 0x0   |
| [6:0] | RW     | rtc_lr_10ms | RTC 10ms 设置值，表示设置的 10ms 时间值。<br>取值范围为 0 ~ 99。 | 0x00  |

## RTC\_LR\_S

rtc\_lr\_s 为 RTC 秒设置值寄存器。

Offset Address:0x12 Total Reset Value:0x00

| Bits  | Access | Name     | Description                          | Reset |
|-------|--------|----------|--------------------------------------|-------|
| [7:6] | RO     | reserved | 保留                                   | 0x0   |
| [5:0] | RW     | rtc_lr_s | RTC 秒设置值，表示设置的秒时间值。<br>取值范围为 0 ~ 59。 | 0x00  |

## RTC\_LR\_M

rtc\_lr\_m 为 RTC 分设置值寄存器。

Offset Address:0x13 Total Reset Value:0x00

| Bits  | Access | Name     | Description                          | Reset |
|-------|--------|----------|--------------------------------------|-------|
| [7:6] | RO     | reserved | 保留。                                  | 0x0   |
| [5:0] | RW     | rtc_lr_m | RTC 分设置值，表示设置的分时间值。<br>取值范围为 0 ~ 59。 | 0x00  |

## RTC\_LR\_H

rtc\_lr\_h 为 RTC 时设置值寄存器。

Offset Address:0x14 Total Reset Value:0x00



| Bits  | Access | Name     | Description                           | Reset |
|-------|--------|----------|---------------------------------------|-------|
| [7:5] | RO     | reserved | 保留。                                   | 0x0   |
| [4:0] | RW     | rtc_lr_h | RTC 时设置值，表示设置的小时时间值。<br>取值范围为 0 ~ 23。 | 0x00  |

## RTC\_LR\_D\_L

rtc\_lr\_d\_l 为 RTC 天设置值的低 8 位寄存器。

Offset Address:0x15 Total Reset Value:0x00

| Bits  | Access | Name       | Description                                            | Reset |
|-------|--------|------------|--------------------------------------------------------|-------|
| [7:0] | RW     | rtc_lr_d_l | RTC 天设置值低 8 位，与 RTC_LR_D_H 配合表示设置的天时间，取值范围为 0 ~ 65535。 | 0x00  |

## RTC\_LR\_D\_H

rtc\_lr\_d\_h 为 RTC 天设置值的高 8 位寄存器。

Offset Address:0x16 Total Reset Value:0x00

| Bits  | Access | Name       | Description                                            | Reset |
|-------|--------|------------|--------------------------------------------------------|-------|
| [7:0] | RW     | rtc_lr_d_h | RTC 天设置值高 8 位，与 RTC_LR_D_L 配合表示设置的天时间，取值范围为 0 ~ 65535。 | 0x00  |

## RTC\_LORD

rtc\_lord 为 RTC 设置值使能加载寄存器。

Offset Address:0x17 Total Reset Value:0x00

| Bits  | Access | Name            | Description                                                                           | Reset |
|-------|--------|-----------------|---------------------------------------------------------------------------------------|-------|
| [7:3] | RO     | reserved        | 保留                                                                                    | 0x00  |
| [2]   | RW     | rtc_lock_bypass | RTC 时间锁存使能信号。<br>0：使能，RTC 计数值(0x00~0x05)只会在锁存成功后才更新。<br>1：不使能，RTC 计数值(0x00~0x05)实时更新。 | 0x0   |
| [1]   | RW     | rtc_lock        | RTC 时间锁存信号。软件写入 1 后，硬件                                                                | 0x0   |

|     |    |          |                                                       |     |
|-----|----|----------|-------------------------------------------------------|-----|
|     |    |          | 会在锁存成功后自动清零。<br>注意:该寄存器在 rtc_lock_bypass 为 0 时才有效。    |     |
| [0] | RW | rtc_load | 把 RTC 的时间配置值写入 RTC 累加器中的使能信号。软件写入 1 后, 硬件会在加载成功后自动清零。 | 0x0 |

## RTC\_IMSC

rtc\_imsc 为 RTC 中断使能寄存器。

Offset Address:0x18 Total Reset Value:0x00

| Bits  | Access | Name          | Description                             | Reset |
|-------|--------|---------------|-----------------------------------------|-------|
| [7:3] | RO     | reserved      | 保留。                                     | 0x00  |
| [2]   | RW     | rtc_imsc      | RTC 总中断输出使能位。<br>0: 不输出中断;<br>1: 输出中断。  | 0x0   |
| [1]   | RW     | rtc_imsc_uv   | 电池低压检测中断输出使能位。<br>0: 不输出中断;<br>1: 输出中断。 | 0x0   |
| [0]   | RW     | rtc_imsc_time | RTC 定时中断输出使能位。<br>0: 不输出中断;<br>1: 输出中断。 | 0x0   |

## RTC\_INT\_CLR

rtc\_int\_clr 为 RTC 中断清除寄存器。

Offset Address:0x19 Total Reset Value:0x00

| Bits  | Access | Name        | Description                         | Reset |
|-------|--------|-------------|-------------------------------------|-------|
| [7:1] | RO     | reserved    | 保留。                                 | 0x00  |
| [0]   | WO     | rtc_int_clr | RTC 中断清除寄存器, 软件写入任意值都可以清除中断, 回读无意思。 | 0x0   |

## RTC\_MSC\_INT

rtc\_msc\_int 为 RTC mask 中断状态寄存器。

Offset Address:0x1a Total Reset Value:0x00

| Bits  | Access | Name          | Description                             | Reset |
|-------|--------|---------------|-----------------------------------------|-------|
| [7:2] | RO     | reserved      | 保留。                                     | 0x00  |
| [1]   | RO     | mask_int_uv   | 屏蔽后的电池低压检测中断状态寄存器。<br>0：无中断；<br>1：有中断。  | 0x0   |
| [0]   | RO     | mask_int_time | 屏蔽后的 RTC 定时中断状态寄存器。<br>0：无中断；<br>1：有中断。 | 0x0   |

## RTC\_RAW\_INT

rtc\_raw\_int 为 RTC 原始中断状态寄存器。

Offset Address:0x1b Total Reset Value:0x00

| Bits  | Access | Name         | Description                          | Reset |
|-------|--------|--------------|--------------------------------------|-------|
| [7:2] | RO     | reserved     | 保留。                                  | 0x00  |
| [1]   | RO     | raw_int_uv   | 电池低压检测原始中断状态寄存器。<br>0：无中断；<br>1：有中断。 | 0x0   |
| [0]   | RO     | raw_int_time | RTC 定时原始中断状态寄存器。<br>0：无中断；<br>1：有中断。 | 0x0   |

## RTC\_CLK

rtc\_clk 为 RTC 输出时钟选择寄存器。

Offset Address:0x1c Total Reset Value:0x00

| Bits  | Access | Name        | Description                                                             | Reset |
|-------|--------|-------------|-------------------------------------------------------------------------|-------|
| [7:2] | RO     | reserved    | 保留。                                                                     | 0x00  |
| [1:0] | RW     | clk_out_sel | RTC 输出的测试时钟选择。<br>00：输出晶体的震荡时钟；<br>01：输出校正后的 100Hz 时钟；<br>1X：输出 1Hz 时钟。 | 0x0   |

## RTC\_POR\_N

rtc\_por\_n 为 RTC 复位控制寄存器。

Offset Address:0x1d Total Reset Value:0x01

| Bits  | Access | Name      | Description                      | Reset |
|-------|--------|-----------|----------------------------------|-------|
| [7:1] | RO     | reserved  | 保留。                              | 0x00  |
| [0]   | RW     | rtc_por_n | RTC 模块的复位信号。复位成功后自动置1。<br>0: 复位。 | 0x1   |

## RTC\_UV\_CTRL

rtc\_uv\_ctrl 为 RTC 内部低压检测控制寄存器。

Offset Address:0x1e Total Reset Value:0x04

| Bits  | Access | Name            | Description                                                     | Reset |
|-------|--------|-----------------|-----------------------------------------------------------------|-------|
| [7:6] | RO     | reserved        | 保留。                                                             | 0x0   |
| [5]   | RW     | bat_uv_ctrl_en  | 低压检测使能控制。<br>0: 关闭;<br>1: 打开。                                   | 0x0   |
| [4]   | RW     | bat_uv_ctrl_sel | 低压检测来源选择。<br>0: 不经过滤波处理;<br>1: 经过滤波处理。                          | 0x0   |
| [3:2] | RW     | bat_uvp_sel     | 电池欠压检测阈值电压设置。<br>00: 1.6V;<br>01: 1.8V;<br>10: 2V;<br>11: 2.2V。 | 0x1   |
| [1:0] | RW     | sample_time     | 低压检测周期。<br>00: 1 秒;<br>01: 1 分钟;<br>10: 10 分钟;<br>11: 30 分钟。    | 0x0   |

## RTC\_ANA\_CTRL

rtc\_ana\_ctrl 为 RTC 振荡电流控制寄存器 (受 RTC 锁定寄存器锁定, 只有当 {RTC\_LOCK3, RTC\_LOCK2, RTC\_LOCK1, RTC\_LOCK0}=0x5A5AABCD 时才可以被修改。)

Offset Address:0x1f    Total Reset Value:0x00

| Bits  | Access | Name         | Description                                                                                                                                    | Reset |
|-------|--------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [7:0] | RW     | rtc_ana_ctrl | RTC 振荡电流控制。<br>000: 0.7uA;<br>001: 0.8uA;<br>010: 0.9uA;<br>011: 1.0uA;<br>100: 0.3uA;<br>101: 0.4uA;<br>110: 0.5uA;<br>111: 0.6uA;<br>其他: 保留。 | 0x0   |

## RTCLDO\_SEL

rtcldo\_sel 为新增: 输出到模拟电路的 RTC\_AVDD09 电压控制。

Offset Address:0x20    Total Reset Value:0x00

| Bits  | Access | Name       | Description                                                                                                                                        | Reset |
|-------|--------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [7:3] | RO     | reserved   | 保留                                                                                                                                                 | 0x0   |
| [2:0] | RW     | rtcldo_sel | Voltage trim of RTC_AVDD09<br>000: 900mV;<br>001: 920mV;<br>010: 940mV;<br>011: 960mV;<br>100: 820mV;<br>101: 840mV;<br>110: 860mV;<br>111: 880mV; | 0x0   |

## RG\_BG\_TRIM18

rg\_bg\_trim18 为新增: 输出到模拟电路的 bg 电压控制。

Offset Address:0x21    Total Reset Value:0x00

| Bits  | Access | Name         | Description                                                                                                                                                                                                                                                        | Reset |
|-------|--------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [7:4] | RO     | reserved     | 保留                                                                                                                                                                                                                                                                 | 0x0   |
| [3:0] | RW     | rg_bg_trim18 | Voltage trim of bg<br>0000: 801mV<br>0001: 797mV<br>0010: 793mV<br>0011: 789mV<br>0100: 785mV<br>0101: 780mV<br>0110: 776mV<br>0111: 772mV<br>1000: 805mV<br>1001: 809mV<br>1010: 813mV<br>1011: 817mV<br>1100: 822mV<br>1101: 826mV<br>1110: 830mV<br>1111: 834mV | 0x0   |

## SDM\_COEF\_OUSIDE\_H

sdm\_coef\_ouside\_h 为外部分频系数高 4 位寄存器。

Offset Address:0x54 Total Reset Value:0x08

| Bits  | Access | Name              | Description                     | Reset |
|-------|--------|-------------------|---------------------------------|-------|
| [7:4] | RO     | reserved          | 保留。                             | 0x0   |
| [3:0] | RW     | sdm_coef_ouside_h | 固定分频模式时的分频系数高 4 位。<br>目前无作用，可读写 | 0x8   |

## SDM\_COEF\_OUSIDE\_L

sdm\_coef\_ouside\_l 为外部分频系数低 8 位寄存器。

Offset Address:0x55 Total Reset Value:0x44

| Bits  | Access | Name              | Description                                        | Reset |
|-------|--------|-------------------|----------------------------------------------------|-------|
| [7:0] | RW     | sdm_coef_ouside_l | 晶振的实际频率参数 b: 配置范围为<br>(0~99), 该参数计算公式为 $b=100-(f-$ | 0x1B  |

|  |  |  |                   |  |
|--|--|--|-------------------|--|
|  |  |  | 32700 ) (f 为晶振频率) |  |
|--|--|--|-------------------|--|

## USER\_REGISTER1

user\_register1 为 64bit 用户使用寄存器 1。

Offset Address:0x56 Total Reset Value:0x00

| Bits  | Access | Name           | Description                   | Reset |
|-------|--------|----------------|-------------------------------|-------|
| [7:0] | RW     | user_register1 | 64bit 用户使用寄存器 1, 对应 bit[7:0]。 | 0x00  |

## USER\_REGISTER2

user\_register2 为 64bit 用户使用寄存器 2。

Offset Address:0x57 Total Reset Value:0x00

| Bits  | Access | Name           | Description                    | Reset |
|-------|--------|----------------|--------------------------------|-------|
| [7:0] | RW     | user_register2 | 64bit 用户使用寄存器 2, 对应 bit[15:8]。 | 0x00  |

## USER\_REGISTER3

user\_register3 为 64bit 用户使用寄存器 3。

Offset Address:0x58 Total Reset Value:0x00

| Bits  | Access | Name           | Description                     | Reset |
|-------|--------|----------------|---------------------------------|-------|
| [7:0] | RW     | user_register3 | 64bit 用户使用寄存器 3, 对应 bit[23:16]。 | 0x00  |

## USER\_REGISTER4

user\_register4 为 64bit 用户使用寄存器 4。

Offset Address:0x59 Total Reset Value:0x00

| Bits  | Access | Name           | Description                     | Reset |
|-------|--------|----------------|---------------------------------|-------|
| [7:0] | RW     | user_register4 | 64bit 用户使用寄存器 4, 对应 bit[31:24]。 | 0x00  |

## USER\_REGISTER5

user\_register5 为 64bit 用户使用寄存器 5。

Offset Address:0x5a Total Reset Value:0x00

| Bits  | Access | Name           | Description                     | Reset |
|-------|--------|----------------|---------------------------------|-------|
| [7:0] | RW     | user_register5 | 64bit 用户使用寄存器 5, 对应 bit[39:32]。 | 0x00  |

## USER\_REGISTER6

user\_register6 为 64bit 用户使用寄存器 6。

Offset Address:0x5b Total Reset Value:0x00

| Bits  | Access | Name           | Description                     | Reset |
|-------|--------|----------------|---------------------------------|-------|
| [7:0] | RW     | user_register6 | 64bit 用户使用寄存器 6, 对应 bit[47:40]。 | 0x00  |

## USER\_REGISTER7

user\_register7 为 64bit 用户使用寄存器 7。

Offset Address:0x5c Total Reset Value:0x00

| Bits  | Access | Name           | Description                     | Reset |
|-------|--------|----------------|---------------------------------|-------|
| [7:0] | RW     | user_register7 | 64bit 用户使用寄存器 7, 对应 bit[55:48]。 | 0x00  |

## USER\_REGISTER8

user\_register8 为 64bit 用户使用寄存器 8。

Offset Address:0x5d Total Reset Value:0x00

| Bits  | Access | Name           | Description                     | Reset |
|-------|--------|----------------|---------------------------------|-------|
| [7:0] | RW     | user_register8 | 64bit 用户使用寄存器 8, 对应 bit[63:56]。 | 0x00  |

## RTC\_LOCK0

rtc\_lock0 为 32bit RTC 锁定寄存器 0。

Offset Address:0x68 Total Reset Value:0x00

| Bits  | Access | Name      | Description                     | Reset |
|-------|--------|-----------|---------------------------------|-------|
| [7:0] | RW     | rtc_lock0 | 32bit RTC 锁定寄存器 0, 对应 bit[7:0]。 | 0x00  |

## RTC\_LOCK1

rtc\_lock1 为 32bit RTC 锁定寄存器 1。

Offset Address:0x69 Total Reset Value:0x00

| Bits  | Access | Name      | Description                      | Reset |
|-------|--------|-----------|----------------------------------|-------|
| [7:0] | RW     | rtc_lock1 | 32bit RTC 锁定寄存器 1, 对应 bit[15:8]。 | 0x00  |

## RTC\_LOCK2

rtc\_lock2 为 32bit RTC 锁定寄存器 2。

Offset Address:0x6a Total Reset Value:0x00



| Bits  | Access | Name      | Description                       | Reset |
|-------|--------|-----------|-----------------------------------|-------|
| [7:0] | RW     | rtc_lock2 | 32bit RTC 锁定寄存器 2, 对应 bit[23:16]。 | 0x00  |

## RTC\_LOCK3

rtc\_lock3 为 32bit RTC 锁定寄存器 3。

Offset Address:0x6b    Total Reset Value:0x00

| Bits  | Access | Name      | Description                       | Reset |
|-------|--------|-----------|-----------------------------------|-------|
| [7:0] | RW     | rtc_lock3 | 32bit RTC 锁定寄存器 3, 对应 bit[31:24]。 | 0x00  |

## 1.10 电源管理与低功耗模式

### 1.10.1 概述

- PMC (Power Management Controller) 通过对非恒电区的电源使能进行控制, 实现芯片的开关机功能。
- 低功耗模式用来有效的减少芯片的功耗, 芯片提供多种低功耗的控制来动态降低芯片的功耗。
  - 时钟门控和时钟频率调整  
提供时钟关断功能, 可以关闭没有必要的时钟, 减少芯片的功耗。系统工作的时钟频率可以进行调整, 在满足功能的情况下可以调节时钟频率, 降低芯片功耗。
  - 模块级低功耗控制  
提供模块级的低功耗控制, 在某模块不工作的情况下, 关断该模块或使模块处于低功耗状态, 以减少芯片的功耗。

### 1.10.2 PMC

#### 1.10.2.1 功能描述

PMC 通过对非恒电区的电源使能进行控制, 实现芯片的上下电时序控制和开关机功能。

PMC 具备以下特点:

- 唤醒源极性可配置

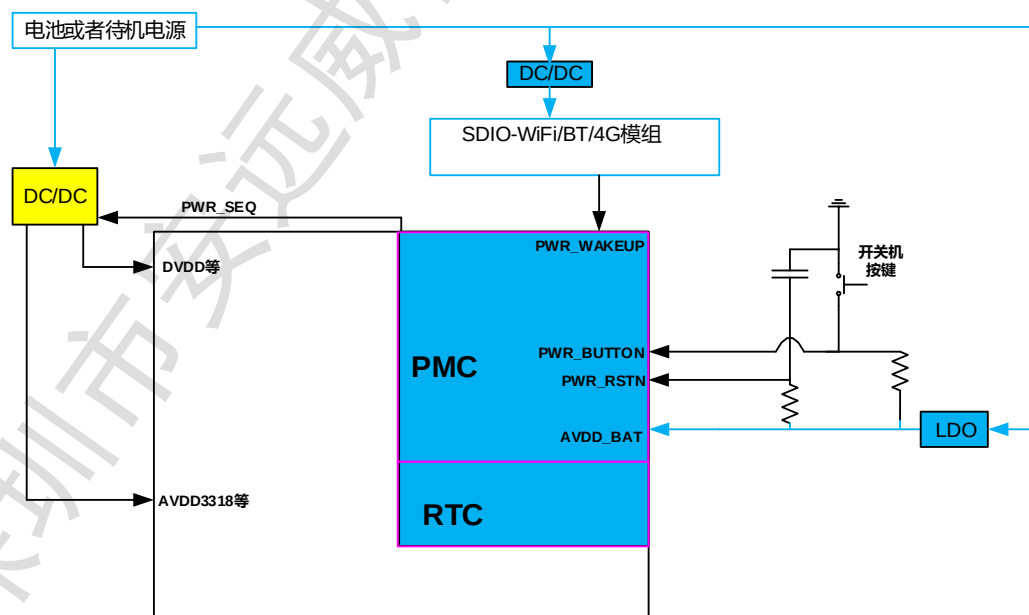
- 上下电时序时间间隔可配置
- 支持 RTC 定时开机
- 支持第一次开机后上电时间可配置
- 支持异常时，按 PWR\_BUTTON 按键 10.2s 秒触发重启时序
- 支持 PMC 区域部分管脚可以配置为外部 GPIO 功能，并且下电后锁存 GPIO 管脚配置的值
- 支持 NPU MEM Retention 功能开关可配置（且只允许配置一次即锁定）

### 须知

PMC 的工作时钟为 RTC 提供的 32768Hz 时钟，所以当需要使用 PMC 功能时，RTC 必须处于工作模式(即 32768Hz 的 RTC 晶体必须接上，同时 AVDD\_BAT 管脚必须供电)。

使用 PMC 模块进行芯片的上下电时序控制时，电源网络粗略框图如图 1-8 所示。其中 PMC 为恒电区，PWR\_SEQ 控制的芯片其它电源为可下电区，当处于关机状态下，芯片除了 PMC 模块，其它区域都下电，这时整个芯片处于最小的功耗状态，待机电流为 uA 级别。

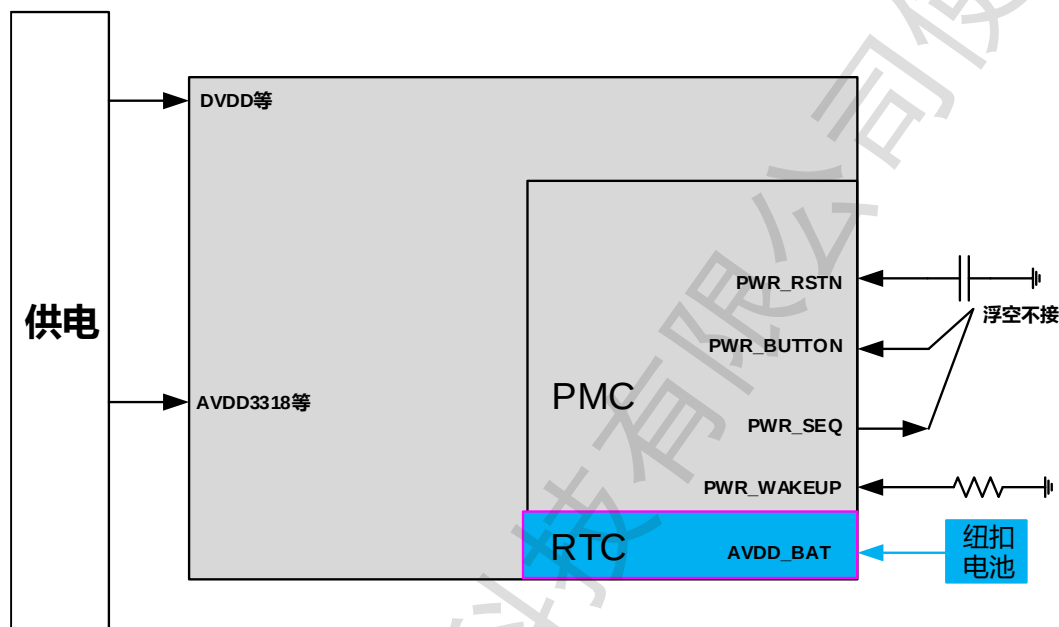
图1-8 使用 PMC 时的电源网络粗略框图



当不使用 PMC 功能时，相关管脚的处理如图 1-9 所示，其中 PWR\_RSTN 接电容到地，PWR\_WAKEUP 管脚下拉到地，其它管脚浮空处理。

说明：PWR\_WAKEUP 有两个 PWR\_WAKEUP0, PWR\_WAKEUP1。

图1-9 不使用 PMC 功能时外围管脚的处理



当 PMC 处于非复位状态时 (PWR\_RSTN 为高)，逻辑才会正常运转。PMC 由外部唤醒管脚(PWR\_WAKEUP、PWR\_STARTUP 和 PWR\_BUTTON)或者 RTC 定时中断触发。

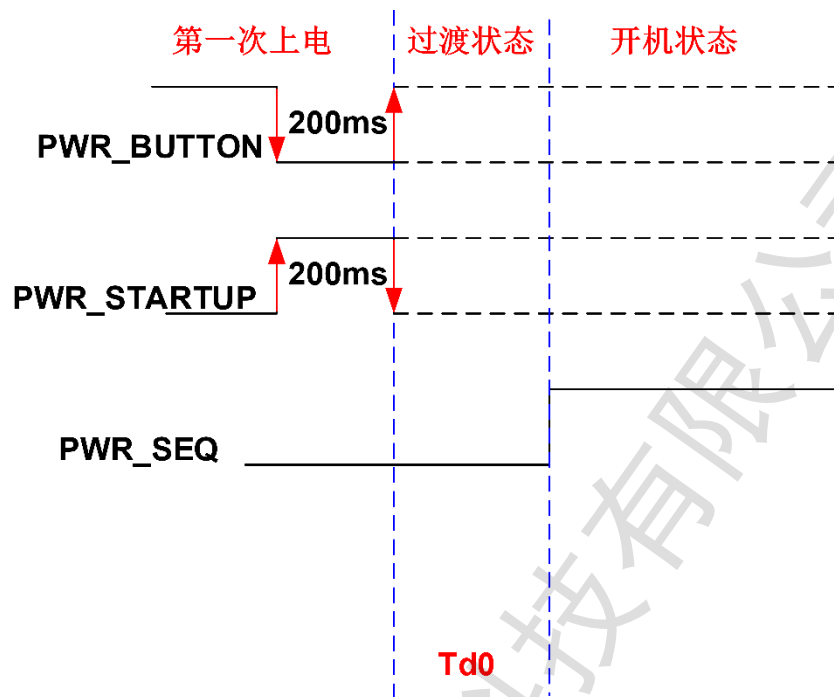
#### 说明

本章节中名词说明：

- 第一次上电状态：是指第一次安装电池，除 PMC 有电外芯片其余模块均未上电，在该状态下只能通过 PWR\_BUTTON 或 PWR\_STARTUP 开机，PWR\_SEQ 为 0；
- 开机状态：整体芯片均上电，PWR\_SEQ 为 1；
- 关机状态：正常的关机，除 PMC 有电外芯片其余模块均未上电，PWR\_SEQ 为 0，在此状态下可以由 PWR\_BUTTON、PWR\_WAKEUP、RTC 唤醒；

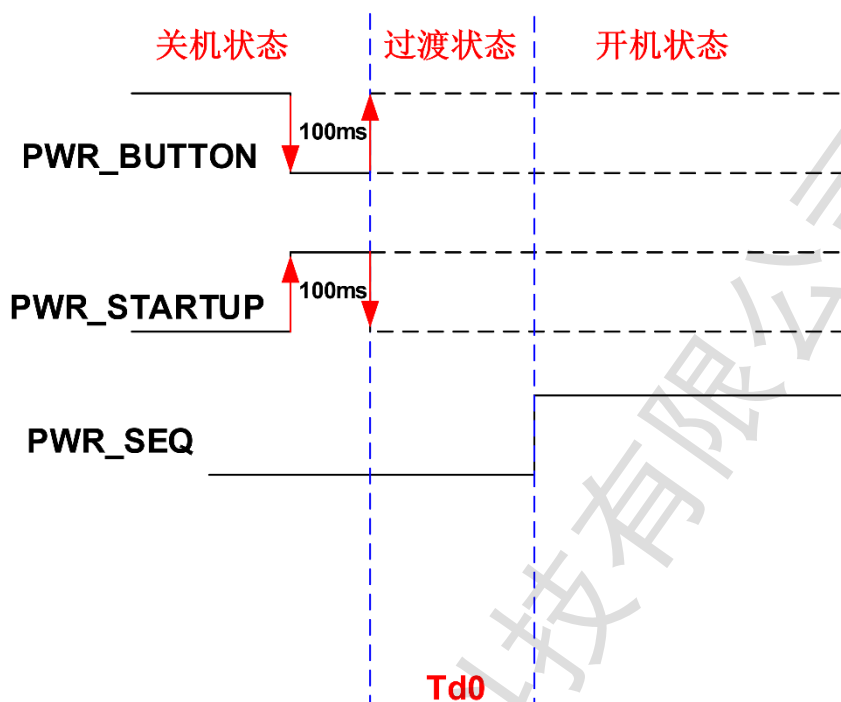
当在第一次上电时，PWR\_BUTTON 保持 200ms 的低电平后可触发上电时序。如图 1-10 所示。

图1-10 PWR\_BUTTON、PWR\_STARTUP 第一次上电时序



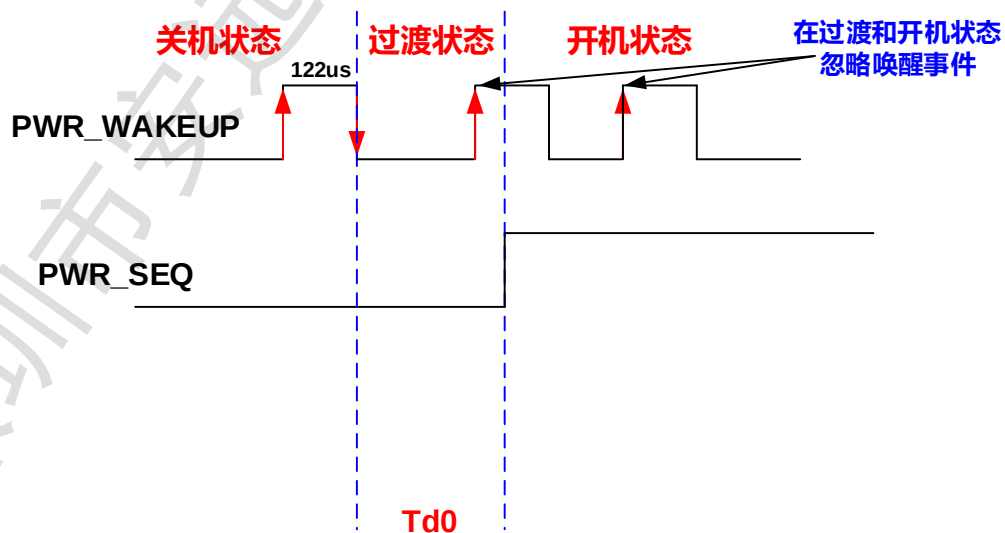
第一次上电开机后，当在关机状态时 PWR\_BUTTON、PWR\_STARTUP 的上电时间可配置，单位为 100ms，覆盖 0~3s；PWR\_BUTTON、PWR\_STARTUP 的上电时序如图 1-11 所示（以 100ms 为例）。

图1-11 PWR\_BUTTON、PWR\_STARTUP 上电时序



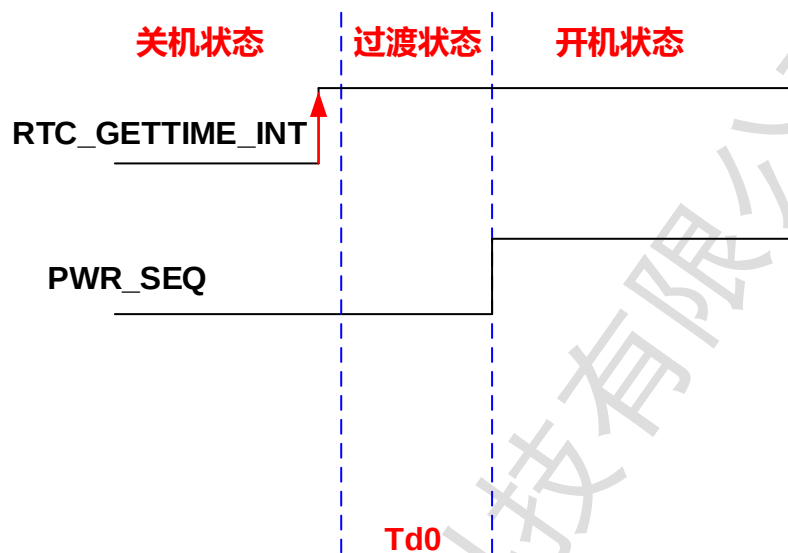
PWR\_WAKEUP 默认是上升沿触发上电时序，可以在开机状态下进行重新配置，可配置为上升沿、下降沿、高电平和低电平触发。如图 1-12 所示（以上升沿触发为例）。

图1-12 PWR\_WAKEUP 上电时序（以上升沿触发为例）



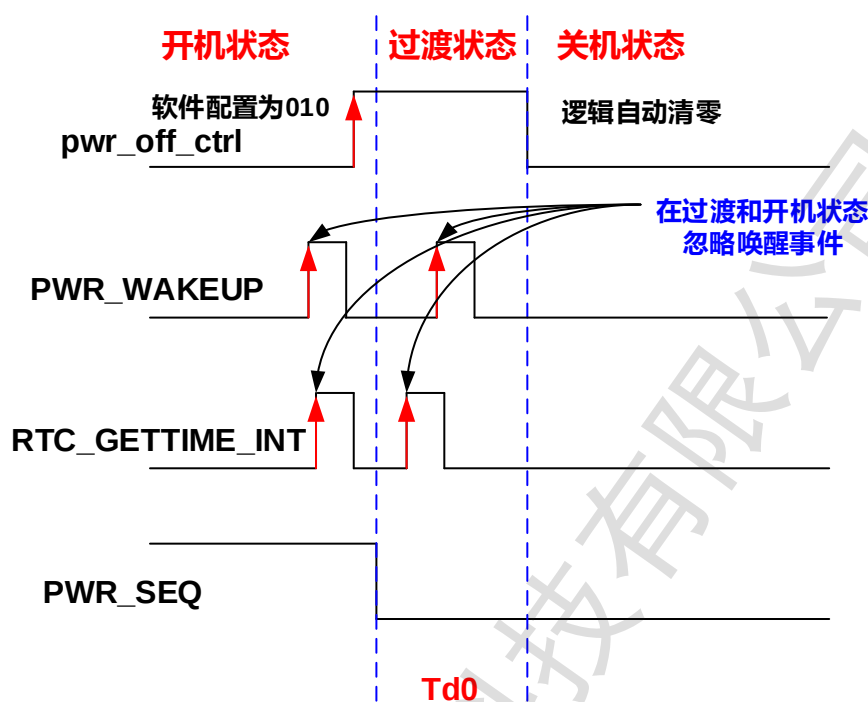
在关机状态下，RTC 定时中断产生且寄存器 `rtc_int_mask` 配置为 0 时，触发上电时序。如图 1-13 所示。

图1-13 RTC 定时中断上电时序



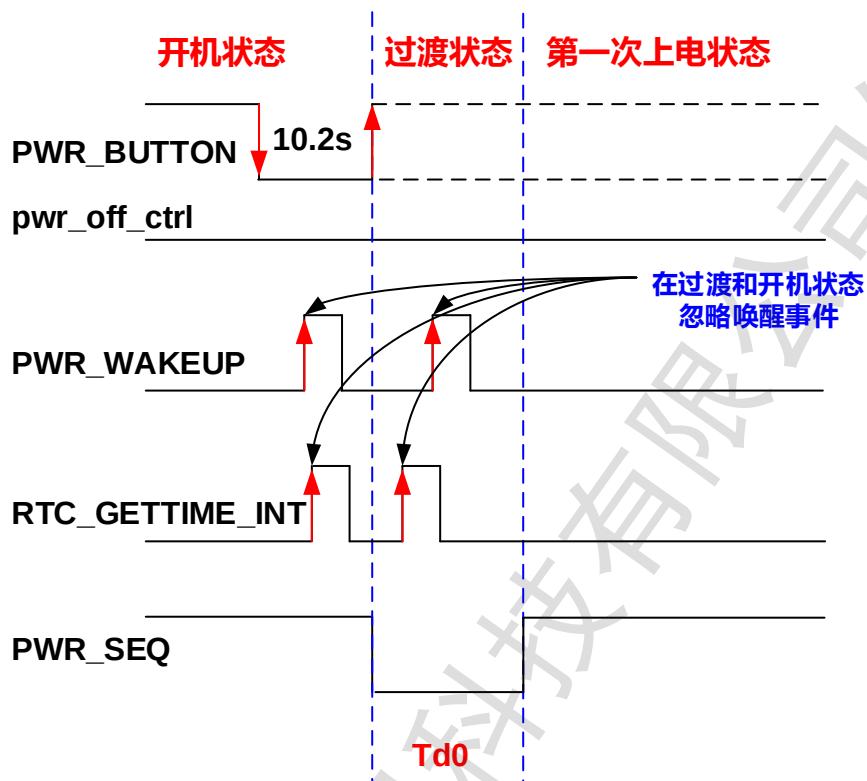
在开机状态下，通过配置寄存器[start\_powerdown]为 1 即可触发下电时序。如图 1-14 所示。

图1-14 软件配置下电时序



在开机状态下，若软件出现异常，可以通过 PWR\_BUTTON 保持低电平 10.2s 触发下电重启时序，如图 1-15 所示。

图1-15 异常下电重启时序



Td0 是可以编程的值，对应的寄存器为 PWR\_CTRL1[pwr\_td0]，取值范围为 1~255，计时单位为 ms，复位默认为 16。

PMC 对输入信号(PWR\_WAKEUP、PWR\_STARTUP 和 PWR\_BUTTON)进行滤波处理，宽度低于 122us 的电平将会滤除掉。

PWR\_RSTN 会复位整个 PMC 模块，复位生效后进入复位状态，在复位状态下忽略所有事件，同时 PWR\_SEQ 输出为 0，如图 1-16 所示。

NPU MEM Retention 功能开关

配置 close\_npu\_mem\_retention（偏移地址 0x58）值为 0 为打开 RETNETION，默认为打开。

配置 close\_npu\_mem\_retention（偏移地址 0x58）值为 1 为关闭 RETENTION 功能，并且关闭后无法打开。

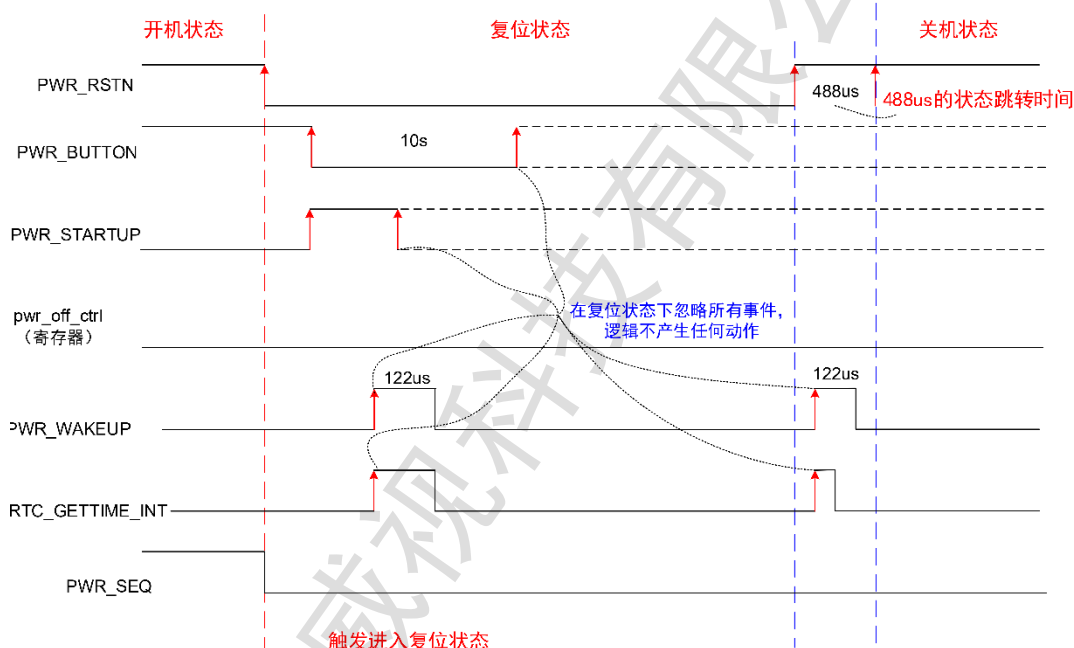
PMC 区域部分管脚可以配置为外部 GPIO 功能。



pwr\_wakeup1, pwr\_startup, pwr\_uart2\_rxd 可以配置为 GPIO 功能，配置为使能 GPIO 功能后，下电后其值被锁定。若要解除锁定 GPIO 功能，上电后重新配置不使能 GPIO 功能。

配置方法:配置 npu\_ack\_pmc\_lock 寄存器的相应 bit，配置 1 为使能 GPIO 功能，配置 0 为不使能 GPIO 功能。

图1-16 复位状态示意图



## 须知

- 唤醒源 PWR\_WAKEUP、PWR\_STARTUP 和 PWR\_BUTTON 只有在开机状态或者待机状态才会产生中断。
- 通过异常重启进入第一次上电状态时将复位全部寄存器。
- PWR\_STARTUP 除无长按 10.2s 强制关机重启、开机或待机状态下检测上升沿开机外，其余功能与 PWR\_BUTTON 相同，即 PWR\_STARTUP 可以在关机下启动，在开机与待机下产生中断。

相关电源管脚分组如表 1-21 所示。

表1-21 电源对接表

| 上下电时序分组      | 电源管脚             |
|--------------|------------------|
| 由 PWR_SEQ 控制 | AVDD_PLL         |
|              | AVDD33_PLL       |
|              | DVDD33           |
|              | VDD              |
|              | DVDD3318_LCD     |
|              | DVDD3318_OSC_LCD |
|              | DVDD3318_SENSOR  |
|              | AVDD3318_MIPIRX  |
|              | AVDD33_AC_U2     |
|              | DVDD3318_FLASH   |
|              | AVDD33_DDR_PLL   |
|              | VDDIO_DDR        |

### 1.10.2.2 PMC 寄存器概览

PMC 基地址：120F\_0000

表1-22 PMC 寄存器概览

| 偏移地址   | 名称               | 描述                   |
|--------|------------------|----------------------|
| 0x004  | PWR_STATUS       | PWR 状态寄存器。           |
| 0x008  | PWR_PD_SEL       | PWR 下电时序控制寄存器。       |
| 0x0010 | PWR_CTRL0        | PWR 控制寄存器 0。         |
| 0x014  | PWR_CTRL1        | PWR 控制寄存器 1。         |
| 0x018  | PWR_BUTTON_CRTL  | PWR_BUTTON 管脚控制寄存器。  |
| 0x020  | PWR_STARTUP_CRTL | PWR_STARTUP 管脚控制寄存器。 |
| 0x024  | PWR_RSTN_CRTL    | PWR_RSTN 管脚控制寄存器。    |
| 0x028  | PWR_SEQ_CRTL     | PWR_SEQ 管脚控制寄存器。     |
| 0x030  | PWR_WAKEUP0_CRTL | PWR_WAKEUP0 管脚控制寄存   |

|       |                  |                           |
|-------|------------------|---------------------------|
|       |                  | 器。                        |
| 0x038 | PWR_WAKEUP1_CRTL | PWR_WAKEUP1 管脚控制寄存器。      |
| 0x048 | pwr_user_reg0    | PWR 用户寄存器 0               |
| 0x04C | pwr_user_reg1    | PWR 用户寄存器 1               |
| 0x050 | pwr_user_reg2    | PWR 用户寄存器 2               |
| 0x054 | mem_retention    | NPU MEM RETENTION 开关控制寄存器 |
| 0x058 | pwr_user_reg4    | PWR 用户寄存器 4               |
| 0x060 | npa_ack_pmc_lock | PWR 锁存信号使能                |
| 0x064 | npa_ps_ack       | NPU MEM 上电状态寄存器           |
| 0x068 | l1_cache_size    | L1 dcache/ccache size     |

### 1.10.2.3 PMC 寄存器描述

#### PWR\_STATUS

pwr\_status 为 PWR 状态寄存器。

Offset Address:0x4 Total Reset Value:0x00000000

| Bits    | Access | Name          | Description                                                                 | Reset    |
|---------|--------|---------------|-----------------------------------------------------------------------------|----------|
| [31:11] | RO     | reserved      | 保留。                                                                         | 0x000000 |
| [10]    | RO     | wakeup1_in    | WAKEUP1 管脚输入值。                                                              | 0x0      |
| [9]     | RO     | startup_in    | STRATUP 管脚输入值。                                                              | 0x0      |
| [8]     | RO     | wakeup0_in    | WAKEUP 管脚输入值。                                                               | 0x0      |
| [7]     | RO     | button_in     | BUTTON 管脚输入值。                                                               | 0x0      |
| [6:2]   | RO     | wakeup_source | 唤醒源。<br>bit0: WAKEUP;<br>bit1: reserved;<br>bit2: STARTUP;<br>bit3: BUTTON; | 0x00     |

|     |    |                       |                                  |     |
|-----|----|-----------------------|----------------------------------|-----|
|     |    |                       | bit4: RTC。                       |     |
| [1] | RO | poweron_from_standby  | 从待机进入开机状态的指示。<br>1: 是;<br>0: 不是。 | 0x0 |
| [0] | RO | poweron_from_poweroff | 从关机进入开机状态的指示。<br>1: 是;<br>0: 不是。 | 0x0 |

## PWR\_PD\_SEL

pwr\_pd\_sel 为 PWR 下电时序控制寄存器。

Offset Address:0x8 Total Reset Value:0x00000010

| Bits    | Access | Name              | Description                                                                                                                                      | Reset    |
|---------|--------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| [31:10] | RW     | reserved          | 保留。                                                                                                                                              | 0x000000 |
| [9:5]   | RW     | pldo_cal          | LDO 电压调整<br>00000: 0.9V;<br>00001: 0.91V;<br>...<br>01111: 1.06V;<br>10000: 0.73V;<br>10001: 0.74V;<br>...<br>11111: 0.89V                       | 0x0      |
| [4]     | RW     | ddr_ioctl_rt_en_n | DDR RETENSITON 使能。<br>0: 使能;<br>1: 不使能。                                                                                                          | 0x1      |
| [3]     | RW     | reserved          | 保留。                                                                                                                                              | 0x0      |
| [2:0]   | RW     | pd_seq_mode_sel   | 下电时序选择控制。<br>0: 模式 1, 下电时序为<br>SEQ0->SEQ1->SEQ2;<br>1: 模式 2, 下电时序为<br>SEQ1->SEQ0->SEQ2;<br>2: 模式 3, 下电时序为<br>SEQ2->SEQ0->SEQ1;<br>3: 模式 4, 下电时序为 | 0x0      |

|  |  |  |                                                                                                                                         |  |
|--|--|--|-----------------------------------------------------------------------------------------------------------------------------------------|--|
|  |  |  | SEQ2->SEQ1->SEQ0;<br>4: 模式 5, 下电时序为<br>SEQ1->SEQ2->SEQ0;<br>5: 模式 6, 下电时序为<br>SEQ0->SEQ2->SEQ1;<br>其它: 模式 1, 下电时序为<br>SEQ0->SEQ1->SEQ2。 |  |
|--|--|--|-----------------------------------------------------------------------------------------------------------------------------------------|--|

## PWR\_CTRL0

pwr\_ctrl0 为 PWR 控制寄存器 0。

Offset Address:0x10 Total Reset Value:0x00100160

| Bits    | Access | Name                  | Description                                                                                    | Reset     |
|---------|--------|-----------------------|------------------------------------------------------------------------------------------------|-----------|
| [31:29] | -      | reserved              | 保留。                                                                                            | 0x0000000 |
| [28]    | RW     | high_adj_button0_en   | 0: 使能<br>1: 不使能                                                                                | 0x00      |
| [27:20] | RW     | high_adj_button0_time | time 配置参数                                                                                      | 0x1       |
| [19:17] | RW     | reserved              | 保留。                                                                                            | 0x00      |
| [16]    | RW     | high_adj_startup_en   | 0: 使能<br>1: 不使能                                                                                | 0x00      |
| [15:8]  | RW     | high_adj_startup_time | time 配置参数                                                                                      | 0x1       |
| [6]     | RW     | wakeup_mask           | 是否屏蔽 wakeup 唤醒。<br>0: 不屏蔽;<br>1: 屏蔽。                                                           | 0x1       |
| [5]     | RW     | rtc_int_mask          | 是否屏蔽 RTC 中断唤醒。<br>0: 不屏蔽;<br>1: 屏蔽。                                                            | 0x1       |
| [4:3]   | RW     | wakeup_act_mode       | WAKEUP 有效模式选择。<br>00: 上升沿有效;<br>01: 下降沿有效;<br>10: 高电平有效;<br>11: 低电平有效。<br>若两个 wakeup 唤醒源都使用, 只 | 0x0       |

|     |    |                 |              |           |
|-----|----|-----------------|--------------|-----------|
|     |    |                 | 能选择上升沿/高电平有效 |           |
| [2] | RW | reserved        | 保留。          | 0x0000000 |
| [1] | WO | start_powerdown | 开始关机流程。      | 0x0       |
| [0] | WO | start_standby   | 开始待机流程。      | 0x0       |

## PWR\_CTRL1

pwr\_ctrl1 为 PWR 控制寄存器 1。

Offset Address:0x14 Total Reset Value:0x10051010

| Bits    | Access | Name    | Description                           | Reset |
|---------|--------|---------|---------------------------------------|-------|
| [31:24] | RW     | pwr_td0 | 电源时序参数 0 配置。单位为 ms。<br>0 无效，实际生效值为 1。 | 0x10  |
| [23:16] | RW     | pwr_td1 | 电源时序参数 1 配置。单位为 ms。<br>0 无效，实际生效值为 1。 | 0x5   |
| [15:8]  | RW     | pwr_td2 | 电源时序参数 2 配置。单位为 ms。<br>0 无效，实际生效值为 1。 | 0x10  |
| [7:0]   | RW     | pwr_td3 | 电源时序参数 3 配置。单位为 ms。<br>0 无效，实际生效值为 1。 | 0x10  |

## PWR\_BUTTON\_CTRL

pwr\_button\_ctrl 为 PWR\_BUTTON 管脚控制寄存器。

Offset Address:0x18 Total Reset Value:0x00000004

| Bits   | Access | Name                   | Description                                                         | Reset     |
|--------|--------|------------------------|---------------------------------------------------------------------|-----------|
| [31:9] | RW     | reserved               | 保留。                                                                 | 0x0000000 |
| [8]    | RW     | pad_pwr_button_ioshare | pwr_button 管脚复用选择。<br>0: PWR_BUTTON 功能;<br>1: 其它管脚功能 (PMC 外部逻辑控制) 。 | 0x0       |
| [5]    | RW     | pad_pwr_button_st0     | 施密特输入使能                                                             | 0x0       |
| [4]    | RW     | pad_pwr_button_sr      | 管脚电平转换速率控制<br>0: 电平转换速率慢<br>1: 电平转换速率快                              | 0x0       |

|       |    |                   |                                                     |     |
|-------|----|-------------------|-----------------------------------------------------|-----|
| [3]   | RW | pad_pwr_button_pd | 下拉                                                  | 0x0 |
| [2]   | RW | pad_pwr_button_pu | 上拉                                                  | 0x1 |
| [1:0] | RW | pad_pwr_button_ds | 管脚驱动能力：<br>00：档位 4<br>01：档位 3<br>10：档位 2<br>11：档位 1 | 0x0 |

## PWR\_STARTUP\_CTRL

pwr\_startup\_ctrl 为 PWR\_STARTUP 管脚控制寄存器。

Offset Address:0x20 Total Reset Value:0x00000008

| Bits   | Access | Name                    | Description                                                       | Reset      |
|--------|--------|-------------------------|-------------------------------------------------------------------|------------|
| [31:9] | RW     | reserved                | 保留。                                                               | 0x00000000 |
| [8]    | RW     | pad_pwr_startup_ioshare | pwr_startup 管脚复用选择。<br>0：PWR_STARTUP 功能；<br>1：其它管脚功能（PMC 外部逻辑控制）。 | 0x0        |
| [5]    | RW     | pad_pwr_startup_st0     | 施密特输入使能                                                           | 0x0        |
| [4]    | RW     | pad_pwr_startup_sr      | 管脚电平转换速率控制<br>0：电平转换速率慢<br>1：电平转换速率快                              | 0x0        |
| [3]    | RW     | pad_pwr_startup_pd      | 下拉                                                                | 0x1        |
| [2]    | RW     | pad_pwr_startup_pu      | 上拉                                                                | 0x0        |
| [1:0]  | RW     | pad_pwr_startup_ds      | 管脚驱动能力：<br>00：档位 4<br>01：档位 3<br>10：档位 2<br>11：档位 1               | 0x0        |

## PWR\_RSTN\_CTRL

pwr\_rstn\_ctrl 为 PWR\_RSTN 管脚控制寄存器。

Offset Address:0x24 Total Reset Value:0x00000004

| Bits   | Access | Name             | Description                                             | Reset      |
|--------|--------|------------------|---------------------------------------------------------|------------|
| [31:6] | RW     | reserved         | 保留。                                                     | 0x00000000 |
| [5]    | RW     | pad_pwr_rstn_st0 | 施密特输入使能                                                 | 0x0        |
| [4]    | RW     | pad_pwr_rstn_sr  | 管脚电平转换速率控制<br>0: 电平转换速率慢<br>1: 电平转换速率快                  | 0x0        |
| [3]    | RO     | pad_pwr_rstn_pd  | 下拉(接死为 0)                                               | 0x0        |
| [2]    | RO     | pad_pwr_rstn_pu  | 上拉(接死为 1)                                               | 0x1        |
| [1:0]  | RW     | pad_pwr_rstn_ds  | 管脚驱动能力:<br>00: 档位 4<br>01: 档位 3<br>10: 档位 2<br>11: 档位 1 | 0x0        |

## PWR\_SEQ\_CRTL

pwr\_seq\_crtl 为 PWR\_SEQ 管脚控制寄存器。

Offset Address:0x28 Total Reset Value:0x00000000

| Bits   | Access | Name                | Description                                                     | Reset      |
|--------|--------|---------------------|-----------------------------------------------------------------|------------|
| [31:9] | RW     | reserved            | 保留。                                                             | 0x00000000 |
| [8]    | RW     | pad_pwr_seq_ioshare | pwr_seq0 管脚复用选择。<br>0: PWR_SEQ0 功能;<br>1: 其它管脚功能 (PMC 外部逻辑控制) 。 | 0x0        |
| [5]    | RW     | pad_pwr_seq_st0     | 施密特输入使能                                                         | 0x0        |
| [4]    | RW     | pad_pwr_seq_sr      | 管脚电平转换速率控制<br>0: 电平转换速率慢<br>1: 电平转换速率快                          | 0x0        |
| [3]    | RW     | pad_pwr_req_pd      | 下拉                                                              | 0x0        |
| [2]    | RW     | pad_pwr_req_pu      | 上拉                                                              | 0x0        |
| [1:0]  | RW     | pad_pwr_req_ds      | 管脚驱动能力:                                                         | 0x0        |



|  |  |  |                                              |  |
|--|--|--|----------------------------------------------|--|
|  |  |  | 00: 档位 4<br>01: 档位 3<br>10: 档位 2<br>11: 档位 1 |  |
|--|--|--|----------------------------------------------|--|

## PWR\_WAKEUP0\_CRTL

pwr\_wakeup0\_crtl 为 PWR\_WAKEUP0 管脚控制寄存器。  
Offset Address:0x30 Total Reset Value:0x00000400

| Bits    | Access | Name                    | Description                                                               | Reset      |
|---------|--------|-------------------------|---------------------------------------------------------------------------|------------|
| [31:12] | RW     | reserved                | 保留。                                                                       | 0x00000000 |
| [11]    | RW     | pad_pwr_wakeup0_out     | pwr 的 wakeup 作为输出去唤醒板级其他器件时所用的电平值<br>0: 低电平唤醒<br>1: 高电平唤醒                 | 0x0        |
| [10]    | RW     | pad_pwr_wakeup0_oen     | pwr 管脚复用为 PWE_WAKEUP 功能时, 可调整输入输出方向<br>0: wakeup 向外部输出<br>1: wakeup 从外部输入 | 0x1        |
| [9]     | RW     | reserved                | 保留。                                                                       | 0x0        |
| [8]     | RW     | pad_pwr_wakeup0_ioshare | pwr_wakeup0 管脚复用选择。<br>0: PWR_WAKEUP 功能;<br>1: 其它管脚功能 (PMC 外部逻辑控制) 。      | 0x0        |
| [5]     | RW     | pad_pwr_wakeup0_st0     | 施密特输入使能                                                                   | 0x0        |
| [4]     | RW     | pad_pwr_wakeup0_sr      | 管脚电平转换速率控制<br>0: 电平转换速率慢<br>1: 电平转换速率快                                    | 0x0        |
| [3]     | RW     | pad_pwr_wakeup0_pd      | 下拉, 若只使用一个 wakeup, 需要将未使用的 wakeup 配置管脚为下拉                                 | 0x0        |
| [2]     | RW     | pad_pwr_wakeup0_pu      | 上拉                                                                        | 0x0        |

|       |    |                    |                                                     |     |
|-------|----|--------------------|-----------------------------------------------------|-----|
| [1:0] | RW | pad_pwr_wakeup0_ds | 管脚驱动能力：<br>00：档位 4<br>01：档位 3<br>10：档位 2<br>11：档位 1 | 0x0 |
|-------|----|--------------------|-----------------------------------------------------|-----|

## PWR\_WAKEUP1\_CRTL

pwr\_wakeup1\_crtl 为 PWR\_WAKEUP1 管脚控制寄存器。

Offset Address:0x38 Total Reset Value:0x00000400

| Bits    | Access | Name                    | Description                                                            | Reset      |
|---------|--------|-------------------------|------------------------------------------------------------------------|------------|
| [31:12] | RW     | reserved                | 保留。                                                                    | 0x00000000 |
| [11]    | RW     | pad_pwr_wakeup1_out     | pwr 的 wakeup 作为输出去唤醒板级其他器件时所用的电平值<br>0：低电平唤醒<br>1：高电平唤醒                | 0x0        |
| [10]    | RW     | pad_pwr_wakeup1_oen     | pwr 管脚复用为 PWE_WAKEUP 功能时，可调整输入输出方向<br>0：wakeup 向外部输出<br>1：wakeup 从外部输入 | 0x1        |
| [9]     | RW     | reserved                | 保留。                                                                    | 0x0        |
| [8]     | RW     | pad_pwr_wakeup1_ioshare | pwr_wakeup 管脚复用选择。<br>0：PWR_WAKEUP 功能；<br>1：其它管脚功能（PMC 外部逻辑控制）。        | 0x0        |
| [5]     | RW     | pad_pwr_wakeup1_st0     | 施密特输入使能                                                                | 0x0        |
| [4]     | RW     | pad_pwr_wakeup1_sr      | 管脚电平转换速率控制<br>0：电平转换速率慢<br>1：电平转换速率快                                   | 0x0        |
| [3]     | RW     | pad_pwr_wakeup1_pd      | 下拉，若只使用一个 wakeup，需要将未使用的 wakeup 配置管脚为下拉                                | 0x0        |

|       |    |                    |                                                         |     |
|-------|----|--------------------|---------------------------------------------------------|-----|
| [2]   | RW | pad_pwr_wakeup1_pu | 上拉                                                      | 0x0 |
| [1:0] | RW | pad_pwr_wakeup1_ds | 管脚驱动能力：<br>00: 档位 4<br>01: 档位 3<br>10: 档位 2<br>11: 档位 1 | 0x0 |

## PWR\_USER\_REG0

pwr\_user\_reg0 为 PWR 用户寄存器 0。

Offset Address:0x48 Total Reset Value:0x00000000

| Bits   | Access | Name          | Description | Reset      |
|--------|--------|---------------|-------------|------------|
| [31:0] | RW     | pwr_user_reg0 | 用户定义        | 0x00000000 |

## PWR\_USER\_REG1

pwr\_user\_reg1 为 PWR 用户寄存器 1。

Offset Address:0x4c Total Reset Value:0x00000000

| Bits   | Access | Name          | Description | Reset      |
|--------|--------|---------------|-------------|------------|
| [31:0] | RW     | pwr_user_reg1 | 用户定义        | 0x00000000 |

## PWR\_USER\_REG2

pwr\_user\_reg2 为 PWR 用户寄存器 2。

Offset Address:0x50 Total Reset Value:0x00000000

| Bits   | Access | Name          | Description | Reset      |
|--------|--------|---------------|-------------|------------|
| [31:0] | RW     | pwr_user_reg2 | 用户定义        | 0x00000000 |

## PWR\_USER\_REG3

pwr\_user\_reg3 为 PWR 用户寄存器 3。

Offset Address:0x54 Total Reset Value:0x00000000

| Bits   | Access | Name          | Description | Reset      |
|--------|--------|---------------|-------------|------------|
| [31:0] | RW     | pwr_user_reg3 | 用户定义        | 0x00000000 |

## MEM\_RETENTION

mem\_retention 为 NPU MEM RETENTION 开关控制寄存器。

Offset Address:0x58 Total Reset Value:0x00000000

| Bits   | Access | Name                    | Description                                                | Reset      |
|--------|--------|-------------------------|------------------------------------------------------------|------------|
| [31:1] | RW     | reserved                | 保留                                                         | 0x00000000 |
| [0]    | RW     | close_npu_mem_retention | NPU MEM RETENTION 功能控制<br>(关闭后不能再开启)<br><br>0: 开启<br>1: 关闭 | 0x0        |

## NPU\_ACK\_PMC\_LOCK

npu\_ack\_pmc\_lock 为 PWR 锁存信号使能。

Offset Address:0x60 Total Reset Value:0x00000000

| Bits    | Access | Name                  | Description                                                                                                                       | Reset      |
|---------|--------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------|------------|
| [31:14] | RW     | reserved              | 保留。                                                                                                                               | 0x00000000 |
| [13:3]  | RW     | npu_ack_cnt           | NPU mem retention 功能, 收到 ACK 信号延迟计数 (包括 npu_pmc_ps_ack 和 npu_core_ps_ack) 。<br><br>表示收到 ACK 信后后多久才跳转状态机, 下电另一个 POWER GATE 单位 (ms) | 0x0000     |
| [2]     | RW     | pwr_wakeup1_lock_en   | PWR_WAKEUP1 管脚 gpio 功能锁存使能<br><br>1: 使能<br>0: 不使能                                                                                 | 0x0        |
| [1]     | RW     | pwr_startup_lock_en   | PWR_STARTUP 管脚 gpio 功能锁存使能<br><br>1: 使能<br>0: 不使能                                                                                 | 0x0        |
| [0]     | RW     | pwr_uart2_rxd_lock_en | PWR_UART2_RXD 管脚 gpio 功能                                                                                                          | 0x0        |

|  |  |  |                         |  |
|--|--|--|-------------------------|--|
|  |  |  | 锁存使能<br>1: 使能<br>0: 不使能 |  |
|--|--|--|-------------------------|--|

## NPU\_PS\_ACK

npu\_ps\_ack 为 NPU MEM 上电状态寄存器

Offset Address:0x64 Total Reset Value:0x00000000

| Bits   | Access | Name            | Description                               | Reset      |
|--------|--------|-----------------|-------------------------------------------|------------|
| [31:2] | RW     | reserved        | 保留。                                       | 0x00000000 |
| [1]    | RO     | npu_pmc_ps_ack  | NPU MEM 的 PMC 供电应答信号<br>0: 未供电<br>1: 已供电  | 0x0        |
| [0]    | RO     | npu_core_ps_ack | NPU MEM 的 CORE 供电应答信号<br>0: 未供电<br>1: 已供电 | 0x0        |

## L1\_CACHE\_SIZE

l1\_cache\_size 为 L1 dcache/ccache size。

Offset Address:0x68 Total Reset Value:0x0000001B

| Bits   | Access | Name                | Description                                                                        | Reset      |
|--------|--------|---------------------|------------------------------------------------------------------------------------|------------|
| [31:3] | RW     | reserved            | 保留。                                                                                | 0x00000000 |
| [5:3]  | RW     | l1_icache_size[5:3] | 3'b001: L1 icache size : 16KB<br>3'b011: L1 icache size : 32KB<br>others: reserved | 0x3        |
| [2:0]  | RW     | l1_dcache_size[2:0] | 3'b001: L1 dcache : 16KB<br>3'b011: L1 dcache : 32KB<br>others: reserved           | 0x3        |

#### 1.10.2.4 GPIO 管脚复用掉电锁存功能使用流程

PWR\_WAKEUP1/PWR\_STARUP/PWR\_UART2\_RXD 支持掉电锁存功能。

使用流程，以 PWR\_WAKEUP1 为例：

- 步骤 1 将需要使用该功能的管脚选为 GPIO 复用，配置 0x120f0400 的 pad\_pwr\_wakeup1\_iosshare=1;
- 步骤 2 打开对应管脚锁存功能，配置 0x120f0060 的 pwr\_wakeup1\_lock\_en=1;
- 步骤 3 配置 PMC 关机/待机，将掉电区下电;
- 步骤 4 锁存功能会在 PMC 状态机从 ON 状态跳转后，自动报错 GPIO 输出的状态不变（注意：前文配置管脚锁存使能打开后并不会骂死锁存管脚，是下电的状态跳转才会触发锁存）；
- 步骤 5 PMC 重新上电后管脚依然保存锁存;
- 步骤 6 关闭对应管脚锁存使能，配置 0x120f0060 的 pwr\_wakeup1\_lock\_en=0;
- 步骤 7 管脚接触锁存状态，重新动态输出 GPIO 的值。

----结束

# 2 外围设备

## 2.1 I<sup>2</sup>C

### 2.1.1 概述

I<sup>2</sup>C(Inter-Integrated Circuit)控制器为 Master 接口，完成 CPU 对 I<sup>2</sup>C 总线上连接的从设备的读写访问。

芯片共有 3 个 I<sup>2</sup>C 控制器。

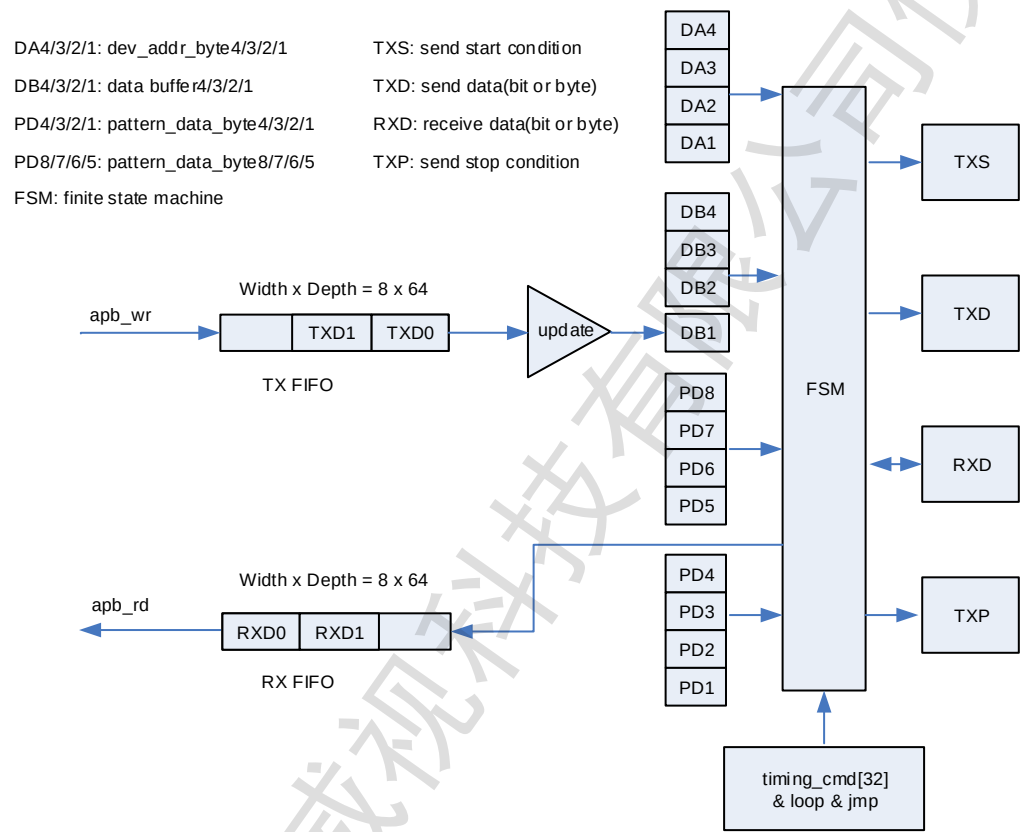
### 2.1.2 功能描述

I<sup>2</sup>C 控制器具有以下功能特点：

- 芯片的 I<sup>2</sup>C 是 Master 接口，支持标准时序和非标准时序。
- 支持多主设备时的总线仲裁。
- 支持 Clock synchronization 和 Bit and Byte waiting。
- 支持标准地址（7bit）和扩展地址（10bit）。
- 传输速率支持标准模式（100kbit/s）和快速模式（400kbit/s）。
- 支持 General Call 和 Start Byte 功能。
- 不支持 CBUS 器件。
- 支持 DMA 操作。
- 支持 64 x 8bit 的 TX FIFO 和 64 x 8bit 的 RX FIFO。

### 2.1.3 功能框图

图2-1 控制器的功能框图



### 2.1.4 时序描述原理

时序描述包括时序格式描述和时序数据准备两部分。时序格式描述通过配置 I2C\_TIMING\_CMD、I2C\_LOOP1、I2C\_DST1、I2C\_LOOP2、I2C\_DST2、I2C\_LOOP3、I2C\_DST3 实现，既能配置出标准时序也能配置出非标准时序。

- 时序数据准备通过配置 I2C\_DEV\_ADDR、I2C\_DATA\_BUF、I2C\_PATTERN\_DATA1、I2C\_PATTERN\_DATA2、I2C\_TX\_FIFO 实现，配置的数值都是需要发送到从设备的。
- 在时序格式描述时 I2C\_TIMING\_CMD 配置的是时序命令信息，I2C\_LOOP1、I2C\_DST1、I2C\_LOOP2、I2C\_DST2、I2C\_LOOP3、I2C\_DST3 配置的是跳转控制信息。I2C\_TIMING\_CMD 可以配置的时序命令信息如表 2-1 所示。



表2-1 时序命令

| 时序命令编号 | 时序命令代号 | 时序命令含义                                                                                     |
|--------|--------|--------------------------------------------------------------------------------------------|
| 0x00   | EXIT   | 结束，用于控制器退出                                                                                 |
| 0x01   | S      | 总线 START                                                                                   |
| 0x02   | SDA4   | 发送 dev_addr_byte4                                                                          |
| 0x03   | SDA3   | 发送 dev_addr_byte3                                                                          |
| 0x04   | SDA2   | 发送 dev_addr_byte2                                                                          |
| 0x05   | SDA1   | 发送 dev_addr_byte1                                                                          |
| 0x06   | SDB4   | 发送 data_buf_byte4                                                                          |
| 0x07   | SDB3   | 发送 data_buf_byte3                                                                          |
| 0x08   | SDB2   | 发送 data_buf_byte2                                                                          |
| 0x09   | SDB1   | 发送 data_buf_byte1                                                                          |
| 0x0A   | SPD8   | 发送 pattern_data_byte8                                                                      |
| 0x0B   | SPD7   | 发送 pattern_data_byte7                                                                      |
| 0x0C   | SPD6   | 发送 pattern_data_byte6                                                                      |
| 0x0D   | SPD5   | 发送 pattern_data_byte5                                                                      |
| 0x0E   | SPD4   | 发送 pattern_data_byte4                                                                      |
| 0x0F   | SPD3   | 发送 pattern_data_byte3                                                                      |
| 0x10   | SPD2   | 发送 pattern_data_byte2                                                                      |
| 0x11   | SPD1   | 发送 pattern_data_byte1                                                                      |
| 0x12   | RD     | 接收 1 字节数据<br><b>注意：执行 RD 命令时，若 RX FIFO 满了，则控制器会等待直到 RX FIFO 有空位；等待期间控制器不会改变 I2C 总线的状态。</b> |
| 0x13   | RACK   | 接收低电平应答                                                                                    |
| 0x14   | RNACK  | 接收高电平非应答                                                                                   |

| 时序命令编号 | 时序命令代号 | 时序命令含义                                                                                                                       |
|--------|--------|------------------------------------------------------------------------------------------------------------------------------|
| 0x15   | RNC    | 接收应答，高电平低电平都无所谓                                                                                                              |
| 0x16   | SACK   | 发送低电平应答                                                                                                                      |
| 0x17   | SNACK  | 发送高电平非应答                                                                                                                     |
| 0x18   | JMPN1  | 有限次数跳转，目的由 DST1 寄存器指出，次数由 LOOP1 寄存器指出                                                                                        |
| 0x19   | JMPN2  | 有限次数跳转，目的由 DST2 寄存器指出，次数由 LOOP2 寄存器指出                                                                                        |
| 0x1A   | JMPN3  | 有限次数跳转，目的由 DST3 寄存器指出，次数由 LOOP3 寄存器指出                                                                                        |
| 0x1B   | UNDEF  | 未定义                                                                                                                          |
| 0x1C   | UNDEF  | 未定义                                                                                                                          |
| 0x1D   | UDB1   | 从 TX FIFO 更新数据到 data_buf_byte1<br><b>注意：执行 UDB1 命令时，若 TX FIFO 为空，则控制器会等待直到 TX FIFO 有数据；等待期间控制器不会改变 I<sup>2</sup>C 总线的状态。</b> |
| 0x1E   | SR     | 总线 repeated START                                                                                                            |
| 0x1F   | P      | 总线 STOP                                                                                                                      |

标准时序----7bit 寻址，写操作

图2-2 7bit 寻址，写操作时序图

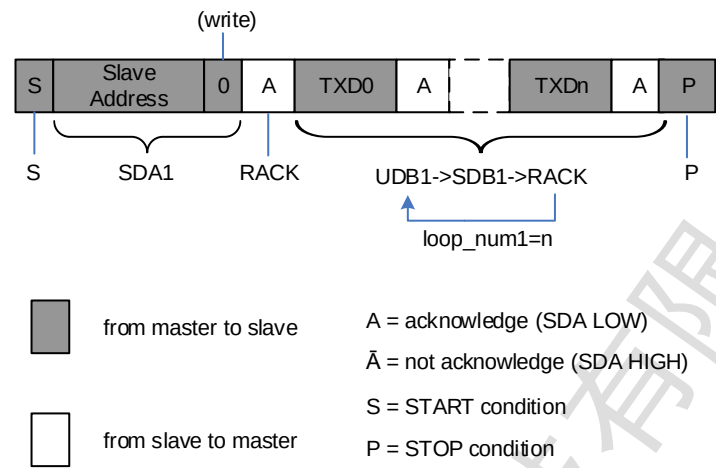


图 2-2 对应的时序描述配置如表 2-2 所示。

表2-2 7bit 寻址，写操作对应的时序描述配置

| 类型     | 寄存器               | 配置                           |
|--------|-------------------|------------------------------|
| 时序格式描述 | I2C_TIMING_CMD[0] | S                            |
|        | I2C_TIMING_CMD[1] | SDA1                         |
|        | I2C_TIMING_CMD[2] | RACK                         |
|        | I2C_TIMING_CMD[3] | UDB1                         |
|        | I2C_TIMING_CMD[4] | SDB1                         |
|        | I2C_TIMING_CMD[5] | RACK                         |
|        | I2C_TIMING_CMD[6] | JMPN1                        |
|        | I2C_TIMING_CMD[7] | P                            |
|        | I2C_TIMING_CMD[8] | EXIT                         |
|        | I2C_LOOP1         | n                            |
|        | I2C_DST1          | 3                            |
|        | I2C_DEV_ADDR      | (Slave Address+ '0' ) 共 8bit |
| 时序数据准备 | I2C_TX_FIFO       | TXD0、TXD1、...TXDn 依次写入。      |

标准时序----7bit 寻址，直接读操作

图2-3 7bit 寻址，直接读时序图

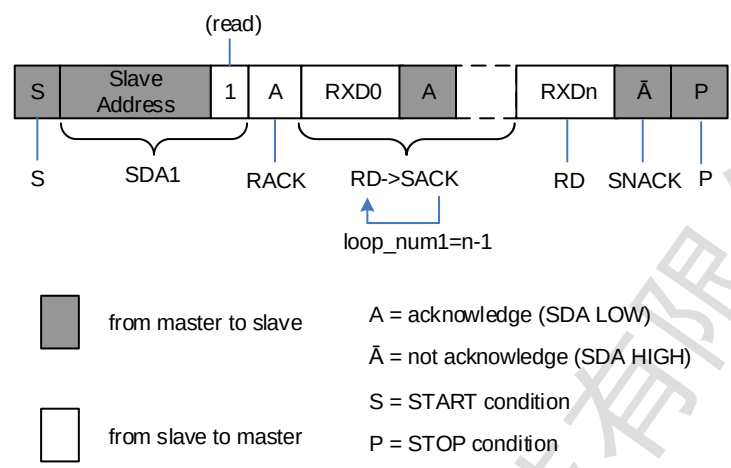


图 2-3 对应的时序描述配置如表 2-3 所示。

表2-3 7bit 寻址，直接读操作的时序描述配置

| 类型     | 寄存器               | 配置                           |
|--------|-------------------|------------------------------|
| 时序格式描述 | I2C_TIMING_CMD[0] | S                            |
|        | I2C_TIMING_CMD[1] | SDA1                         |
|        | I2C_TIMING_CMD[2] | RACK                         |
|        | I2C_TIMING_CMD[3] | RD                           |
|        | I2C_TIMING_CMD[4] | SACK                         |
|        | I2C_TIMING_CMD[5] | JMPN1                        |
|        | I2C_TIMING_CMD[6] | RD                           |
|        | I2C_TIMING_CMD[7] | SNACK                        |
|        | I2C_TIMING_CMD[8] | P                            |
|        | I2C_TIMING_CMD[9] | EXIT                         |
|        | I2C_LOOP1         | n-1                          |
|        | I2C_DST1          | 3                            |
| 时序数据准备 | I2C_DEV_ADDR      | (Slave Address+ '1' ) 共 8bit |

说明  
读回的数据 RXD0、RXD1、... RXDn 依次填入 I2C\_RX\_FIFO。

标准时序----10bit 寻址，写操作

图2-4 10bit 寻址，写操作时序图

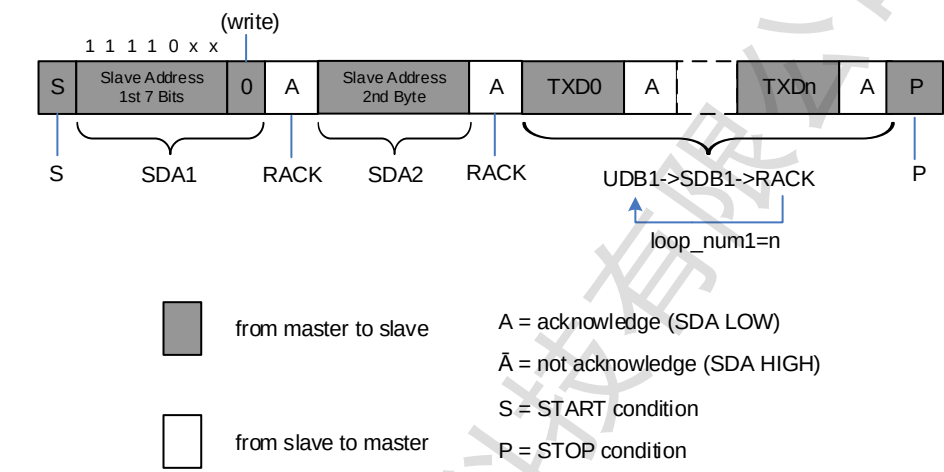


图 2-4 对应的时序描述配置如表 2-4 所示。

表2-4 10bit 寻址，写操作对应的时序描述配置

| 类型     | 寄存器                | 配置    |
|--------|--------------------|-------|
| 时序格式描述 | I2C_TIMING_CMD[0]  | S     |
|        | I2C_TIMING_CMD[1]  | SDA1  |
|        | I2C_TIMING_CMD[2]  | RACK  |
|        | I2C_TIMING_CMD[3]  | SDA2  |
|        | I2C_TIMING_CMD[4]  | RACK  |
|        | I2C_TIMING_CMD[5]  | UDB1  |
|        | I2C_TIMING_CMD[6]  | WDB1  |
|        | I2C_TIMING_CMD[7]  | RACK  |
|        | I2C_TIMING_CMD[8]  | JMPN1 |
|        | I2C_TIMING_CMD[9]  | P     |
|        | I2C_TIMING_CMD[10] | EXIT  |
|        | I2C_LOOP1          | n     |

| 类型     | 寄存器          | 配置                                                                                                        |
|--------|--------------|-----------------------------------------------------------------------------------------------------------|
|        | I2C_DST1     | 5                                                                                                         |
| 时序数据准备 | I2C_DEV_ADDR | (Slave Address 1st 7Bits + '0' ) 共 8bit 写入 dev_addr_byte1;<br>(Slave Address 2nd Byte) 写入 dev_addr_byte2。 |
|        | I2C_TX_FIFO  | TXD0、TXD1、...TXDn 依次写入。                                                                                   |

标准时序----10bit 寻址， 组合读操作

图2-5 10bit 寻址， 组合读时序图

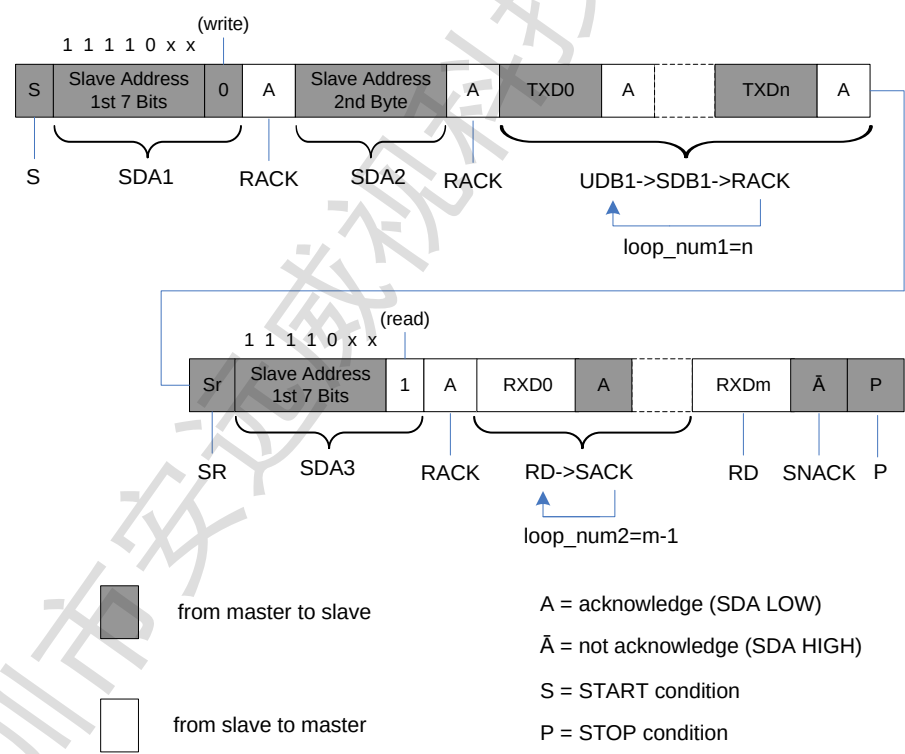


图 2-5 对应的时序描述配置如表 2-5 所示。

表2-5 10bit 寻址，组合读对应的时序描述配置

| 类型     | 寄存器                | 配置                                                                                                                                                                       |
|--------|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 时序格式描述 | I2C_TIMING_CMD[0]  | S                                                                                                                                                                        |
|        | I2C_TIMING_CMD[1]  | SDA1                                                                                                                                                                     |
|        | I2C_TIMING_CMD[2]  | RACK                                                                                                                                                                     |
|        | I2C_TIMING_CMD[3]  | SDA2                                                                                                                                                                     |
|        | I2C_TIMING_CMD[4]  | RACK                                                                                                                                                                     |
|        | I2C_TIMING_CMD[5]  | UDB1                                                                                                                                                                     |
|        | I2C_TIMING_CMD[6]  | WDB1                                                                                                                                                                     |
|        | I2C_TIMING_CMD[7]  | RACK                                                                                                                                                                     |
|        | I2C_TIMING_CMD[8]  | JMPN1                                                                                                                                                                    |
|        | I2C_TIMING_CMD[9]  | SR                                                                                                                                                                       |
|        | I2C_TIMING_CMD[10] | SDA3                                                                                                                                                                     |
|        | I2C_TIMING_CMD[11] | RACK                                                                                                                                                                     |
|        | I2C_TIMING_CMD[12] | RD                                                                                                                                                                       |
|        | I2C_TIMING_CMD[13] | SACK                                                                                                                                                                     |
|        | I2C_TIMING_CMD[14] | JMPN2                                                                                                                                                                    |
|        | I2C_TIMING_CMD[15] | RD                                                                                                                                                                       |
|        | I2C_TIMING_CMD[16] | SNACK                                                                                                                                                                    |
|        | I2C_TIMING_CMD[17] | P                                                                                                                                                                        |
|        | I2C_TIMING_CMD[18] | EXIT                                                                                                                                                                     |
|        | I2C_LOOP1          | n                                                                                                                                                                        |
|        | I2C_DST1           | 5                                                                                                                                                                        |
|        | I2C_LOOP2          | m-1                                                                                                                                                                      |
|        | I2C_DST2           | 12                                                                                                                                                                       |
| 时序数据准备 | I2C_DEV_ADDR       | (Slave Address 1st 7Bits + '0' ) 共 8bit 写入 dev_addr_byte1;<br><br>(Slave Address 2nd Byte) 写入 dev_addr_byte2; (Slave Address 1st 7Bits + '1' ) 共 8bit 写入 dev_addr_byte3。 |
|        | I2C_TX_FIFO        | TXD0、TXD1、...TXDn 依次写入。                                                                                                                                                  |

说明  
读回的数据 RXD0、RXD1、... RXDm 依次填入 I2C\_RX\_FIFO

非标准时序举例

非标准时序为非 I2C 标准协议，图 2-6 为一个非标准时序，此时序连续向从设备发送 64Byte 数据才接收一个应答。

图2-6 非标准时序图

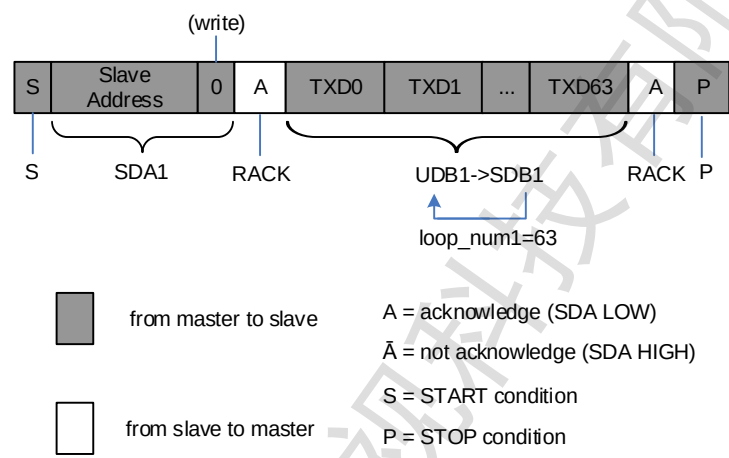


图 2-6 对应的时序描述配置如表 2-6 所示。

表2-6 非标准时序对应时序描述配置

| 类型     | 寄存器               | 配置    |
|--------|-------------------|-------|
| 时序格式描述 | I2C_TIMING_CMD[0] | S     |
|        | I2C_TIMING_CMD[1] | SDA1  |
|        | I2C_TIMING_CMD[2] | RACK  |
|        | I2C_TIMING_CMD[3] | UDB1  |
|        | I2C_TIMING_CMD[4] | SDB1  |
|        | I2C_TIMING_CMD[5] | JMPN1 |
|        | I2C_TIMING_CMD[6] | RACK  |
|        | I2C_TIMING_CMD[7] | P     |
|        | I2C_TIMING_CMD[8] | EXIT  |
|        | I2C_LOOP1         | 63    |



| 类型     | 寄存器          | 配置                           |
|--------|--------------|------------------------------|
|        | I2C_DST1     | 3                            |
| 时序数据准备 | I2C_DEV_ADDR | (Slave Address+ '0' ) 共 8bit |
|        | I2C_TX_FIFO  | TXD0、TXD1、...TXD63 依次写入。     |

## 2.1.5 工作方式

### 2.1.5.1 非 DMA 方式下的数据传输(中断方式写操作)

请遵循以下步骤：

步骤 1 配置 I2C\_HCNT 和 I2C\_LCNT，使控制器工作在正确的模式下。

步骤 2 配置 I2C\_TX\_WATERMARK 为 0x20。

步骤 3 配置 I2C\_GLB，使能控制器。

步骤 4 根据实际时序，参考 2.1.4 时序描述原理完成时序配置。

步骤 5 配置 I2C\_CTRL1 为 0x1，启动控制器。

步骤 6 配置 I2C\_INTR\_EN 为 0x049。

步骤 7 在中断服务程序里，查询 I2C\_INTR\_STAT，若 arb\_lost 或 ack\_bit\_unmatch 为 1，则认为是异常，否则，若 tx\_it\_watermark 为 1 中断，此时可往 TX FIFO 里填入多达 32 个待写数据。

如果还有待写数据，则配置 I2C\_INTR\_RAW 为 0x00，清除触发本次中断处理的 tx\_it\_watermark 中断，然后等待下一次中断服务程序被触发。

如果已经没有待写数据，则配置 I2C\_INTR\_EN 为 0x0011，然后等待 all\_cmd\_done 中断，收到 all\_cmd\_done 中断时，表示所有数据都已经写到从设备端。

步骤 8 配置 I2C\_INTR\_EN 为 0x0，I2C\_INTR\_RAW 为 0x007f，清除中断并且禁止中断上报，为不同工作方式之间的切换做好清理工作。

----结束

### 2.1.5.2 非 DMA 方式下的数据传输(查询方式写操作)

请遵循以下步骤：

步骤 1 配置 I2C\_HCNT 和 I2C\_LCNT，使控制器工作在正确的模式下。

步骤 2 配置 I2C\_GLB，使能控制器。

步骤 3 根据实际时序，参考 2.1.4 时序描述原理完成时序配置。

步骤 4 配置 I2C\_CTRL1 为 0x1，启动控制器。

步骤 5 不断查询 I2C\_FIFO\_STAT [tx\_fifo\_not\_full]，查到为 1 后，将 1Byte 待写数据写入 TX FIFO。当所有待写数据都写入 TX FIFO 后，执行下一步骤。

步骤 6 不断查询 I2C\_INTR\_RAW，查到 arb\_lost\_raw 或者 ack\_bit\_unmatch\_raw 为 1 则认为是异常，查到 all\_cmd\_done\_raw 为 1 则表示所有时序已经完成。

步骤 7 配置 I2C\_INTR\_EN 为 0x0，I2C\_INTR\_RAW 为 0x007f，清除中断并且禁止中断上报，为不同工作方式之间的切换做好清理工作。

----结束

### 2.1.5.3 非 DMA 方式下的数据传输(中断方式读操作)

请遵循以下步骤：

步骤 1 配置 I2C\_HCNT 和 I2C\_LCNT，使控制器工作在正确的模式下。

步骤 2 配置 I2C\_RX\_WATERMARK 为 0x20。

步骤 3 配置 I2C\_GLB，使能控制器。

步骤 4 根据实际时序，参考 2.1.4 时序描述原理完成时序配置。

步骤 5 配置 I2C\_CTRL1 为 0x1，启动控制器。

步骤 6 配置 I2C\_INTR\_EN 为 0x0035。

步骤 7 在中断服务程序里，查询 I2C\_INTR\_STAT，当 arb\_lost 或 ack\_bit\_unmatch 为 1 则认为异常，如无异常，all\_cmd\_done 为 1 说明所有数据都已经读回。如无异常同时 all\_cmd\_done 为 0 说明 rx\_gt\_watermark 中断，此时可以从 RX FIFO 里读出 32 个已读回的数据。读走 32 个数据后配置 I2C\_INTR\_RAW 为 0x20，清除触发本次中断处理的 tx\_lt\_watermark 中断，然后等待下一次中断服务程序被触发。

步骤 8 配置 I2C\_INTR\_EN 为 0x0，I2C\_INTR\_RAW 为 0x007f，清除中断并且禁止中断上报，为不同工作方式之间的切换做好清理工作。

----结束

#### 2.1.5.4 非 DMA 方式下的数据传输(查询方式读操作)

请遵循以下步骤：

- 步骤 1 配置 I2C\_HCNT 和 I2C\_LCNT，使控制器工作在正确的模式下。
- 步骤 2 配置 I2C\_GLB，使能控制器。
- 步骤 3 根据实际时序，参考 2.1.4 时序描述原理完成时序配置。
- 步骤 4 配置 I2C\_CTRL1 为 0x1，启动控制器。
- 步骤 5 不断查询 I2C\_FIFO\_STAT [rx\_fifo\_not\_empty]，查到为 1 后，从 RX FIFO 中读出 1Byte 数据。当所有数据都已读回，执行下一步骤。
- 步骤 6 不断查询 I2C\_INTR\_RAW，查到 arb\_lost\_raw 或者 ack\_bit\_unmatch\_raw 为 1 则认为异常，查到 all\_cmd\_done\_raw 为 1 则表示所有时序已经完成。
- 步骤 7 配置 I2C\_INTR\_EN 为 0x0，I2C\_INTR\_RAW 为 0x007f，清除中断并且禁止中断上报，为不同工作方式之间的切换做好清理工作。

----结束

#### 2.1.5.5 DMA 方式下的数据传输(写操作)

请遵循以下步骤：

- 步骤 1 获取一个 DMAC 通道。
- 步骤 2 完成 DMAC 通道的设置并启动。
- 步骤 3 配置 I2C\_HCNT 和 I2C\_LCNT，使控制器工作在正确的模式下。
- 步骤 4 配置 I2C\_TX\_WATERMARK 为 0x10（该值仅为示例，具体配置值应根据实际使用决定）。
- 步骤 5 配置 I2C\_GLB，使能控制器。
- 步骤 6 根据实际时序，参考 2.1.4 时序描述原理完成时序配置。
- 步骤 7 配置 I2C\_CTRL1 为 0x5，启动控制器。
- 步骤 8 等待 DMAC 完成中断。
- 步骤 9 不断查询 I2C\_INTR\_RAW，查到 arb\_lost\_raw 或者 ack\_bit\_unmatch\_raw 为 1 则认为异常，查到 all\_cmd\_done\_raw 为 1 则表示所有时序已经完成。

步骤 10 配置 I2C\_INTR\_EN 为 0x0, I2C\_INTR\_RAW 为 0x007f, 清除中断并且禁止中断上报, 为不同工作方式之间的切换做好清理工作。

----结束

#### 2.1.5.6 DMA 方式下的数据传输(读操作)

请遵循以下步骤:

步骤 1 获取一个 DMAC 通道。

步骤 2 完成 DMAC 通道的设置并启动。

步骤 3 配置 I2C\_HCNT 和 I2C\_LCNT, 使控制器工作在正确的模式下。

步骤 4 配置 I2C\_RX\_WATERMARK 为 0x10 (该值仅为示例, 具体配置值应根据实际使用决定)。

步骤 5 配置 I2C\_GLB, 使能控制器。

步骤 6 根据实际时序, 参考 2.1.4 时序描述原理完成时序配置。

步骤 7 配置 I2C\_CTRL1 为 0x7, 启动控制器。

步骤 8 等待 DMAC 完成中断。

步骤 9 不断查询 I2C\_INTR\_RAW, 查到 arb\_lost\_raw 或者 ack\_bit\_unmatch\_raw 为 1 则认为是异常, 查到 all\_cmd\_done\_raw 为 1 则表示所有时序已经完成。

步骤 10 配置 I2C\_INTR\_EN 为 0x0, I2C\_INTR\_RAW 为 0x007f, 清除中断并且禁止中断上报, 为不同工作方式之间的切换做好清理工作。

----结束

#### 2.1.5.7 异常处理流程

请遵循以下步骤:

步骤 1 配置 I2C\_GLB [i2c\_enable] 为 0, 禁止控制器。

步骤 2 配置 I2C\_INTR\_EN 为 0x0, I2C\_INTR\_RAW 为 0x007f, 清除中断并且禁止中断上报, 为不同工作方式之间的切换做好清理工作。

----结束

### 2.1.6 I<sup>2</sup>C 寄存器概览

I<sup>2</sup>C 基地址如下：

- I<sup>2</sup>C0 寄存器基地址为 0x1206\_0000。
- I<sup>2</sup>C1 寄存器基地址为 0x1206\_1000。
- I<sup>2</sup>C2 寄存器基地址为 0x1206\_2000。
- I<sup>2</sup>C3 寄存器基地址为 0x1206\_3000。

I<sup>2</sup>C 寄存器偏移地址中变量的取值范围和含义如表 2-7 所示。

表2-7 I<sup>2</sup>C 寄存器偏移地址变量表

| 变量名称 | 取值范围   | 描述                     |
|------|--------|------------------------|
| n    | 0 ~ 31 | I2C 时序命令，最多支持 32 个时序命令 |

表2-8 I<sup>2</sup>C 寄存器概览

| 偏移地址   | 名称               | 描述                             |
|--------|------------------|--------------------------------|
| 0x0000 | I2C_GLB          | I2C 全局配置寄存器                    |
| 0x0004 | I2C_CTRL1        | I <sup>2</sup> C 控制寄存器 1       |
| 0x0008 | I2C_CTRL2        | I <sup>2</sup> C 控制寄存器 2       |
| 0x000c | I2C_FIFO_STAT    | I <sup>2</sup> C FIFO 状态寄存器    |
| 0x0010 | I2C_TX_FIFO      | I <sup>2</sup> C TX FIFO 数据寄存器 |
| 0x0014 | I2C_RX_FIFO      | I <sup>2</sup> C RX FIFO 数据寄存器 |
| 0x0018 | I2C_TX_WATERMARK | I <sup>2</sup> C TX 水线寄存器      |
| 0x001c | I2C_RX_WATERMARK | I <sup>2</sup> C RX FIFO 水线寄存器 |
| 0x0020 | I2C_HCNT         | I <sup>2</sup> C 高电平时长寄存器      |
| 0x0024 | I2C_LCNT         | I <sup>2</sup> C 低电平时长寄存器      |
| 0x0028 | I2C_DEV_ADDR     | I <sup>2</sup> C 器件地址寄存器       |
| 0x002c | I2C_DATA_BUF     | I <sup>2</sup> C 数据 buf 寄存器    |

| 偏移地址              | 名称                | 描述                                |
|-------------------|-------------------|-----------------------------------|
| 0x0030            | I2C_INTR_RAW      | I <sup>2</sup> C 原始中断寄存器          |
| 0x0034            | I2C_INTR_EN       | I <sup>2</sup> C 中断使能寄存器          |
| 0x0038            | I2C_INTR_STAT     | I <sup>2</sup> C 中断状态寄存器          |
| 0x0040 + n<br>x 4 | I2C_TIMING_CMD    | I <sup>2</sup> C 时序命令寄存器          |
| 0x00C0            | I2C_LOOP1         | I <sup>2</sup> C 循环次数 1 寄存器       |
| 0x00C4            | I2C_DST1          | I <sup>2</sup> C 跳转目的 1 寄存器       |
| 0x00C8            | I2C_LOOP2         | I <sup>2</sup> C 循环次数 2 寄存器       |
| 0x00CC            | I2C_DST2          | I <sup>2</sup> C 跳转目的 2 寄存器       |
| 0x00D0            | I2C_LOOP3         | I <sup>2</sup> C 循环次数 3 寄存器       |
| 0x00D4            | I2C_DST3          | I <sup>2</sup> C 跳转目的 3 寄存器       |
| 0x00D8            | I2C_BUS_FREE_THR  | I <sup>2</sup> C 总线空闲阈值寄存器        |
| 0x00DC            | I2C_FSM_STAT      | I <sup>2</sup> C 状态机状态寄存器         |
| 0x00E0            | I2C_PATTERN_DATA1 | I <sup>2</sup> C PATTERN 数据 1 寄存器 |
| 0x00E4            | I2C_PATTERN_DATA2 | I <sup>2</sup> C PATTERN 数据 2 寄存器 |

## 2.1.7 I<sup>2</sup>C 寄存器描述

### I2C\_GLB

I2C\_GLB 为 I<sup>2</sup>C 全局配置寄存器。

Offset Address: 0x0000    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name              | Description       | Reset  |
|---------|--------|-------------------|-------------------|--------|
| [31:24] | RO     | reserved          | 保留。               | 0x00   |
| [23:20] | RW     | dly_sda_duration  | SDA dly time 时长   | 0x0    |
| [19:4]  | RW     | sda_hold_duration | SDA hold time 时长。 | 0x0000 |

| Bits  | Access | Name       | Description                                                                                                                            | Reset |
|-------|--------|------------|----------------------------------------------------------------------------------------------------------------------------------------|-------|
|       |        |            | 参考时钟 PCLK 为 50MHz。配置为 0xa，不允许配置其他值。<br><b>注意：只能控制从主机发往从设备的数据的 hold time 时长。</b>                                                        |       |
| [3:2] | RO     | reserved   | 保留。                                                                                                                                    | 0x00  |
| [1]   | RW     | lit_end    | 接收数据大小端模式。<br>0: LSB first<br>1: MSB first                                                                                             | 0x0   |
| [0]   | RW     | i2c_enable | I <sup>2</sup> C 使能控制。<br>0: 禁止；此时会清空 TX FIFO 和 RX FIFO。<br>1: 使能。<br><b>注意：若出现时序异常，请先禁止控制器，然后再使能，以防 TX FIFO 和 RX FIFO 有上次操作的残留数据。</b> | 0x0   |

## I2C\_CTRL1

I2C\_CTRL1 为 I<sup>2</sup>C 控制寄存器 1。

Offset Address: 0x0004    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name          | Description                                                           | Reset    |
|--------|--------|---------------|-----------------------------------------------------------------------|----------|
| [31:3] | RO     | reserved      | 保留。                                                                   | 0x000000 |
| [2:1]  | RW     | dma_operation | DMA 操作控制。<br>00: 非 DMA 方式;<br>01: 保留;<br>10: DMA 写方式;<br>11: DMA 读方式。 | 0x0      |
| [0]    | RW     | start         | 启动控制。写 1 会启动控制器内部的状态机去执行时序命令序列，执行完毕或者异常后逻辑会反向把此位清 0。                  | 0x0      |

|  |  |  |                   |  |
|--|--|--|-------------------|--|
|  |  |  | 0: 不启动;<br>1: 启动。 |  |
|--|--|--|-------------------|--|

## I2C\_CTRL2

I2C\_CTRL2 为 I<sup>2</sup>C 控制寄存器 2。

Offset Address: 0x0008    Total Reset Value: 0x0000\_007E

| Bits   | Access | Name          | Description                                                     | Reset |
|--------|--------|---------------|-----------------------------------------------------------------|-------|
| [31:7] | RO     | reserved      | 保留。                                                             | 0x0   |
| [6]    | RO     | i2c_sda_oen   | 监测内部 I2C 的 SCL 输出电平。<br>0: SCL 输出低电平;<br>1: SCL 输出高电平。          | 0x1   |
| [5]    | RO     | i2c_sda_in    | 监测外部 I2C 总线 SDA 的电平。<br>0: 总线上 SDA 为低电平;<br>1: 总线上 SDA 为高电平。    | 0x1   |
| [4]    | RO     | i2c_scl_oen   | 监测内部 I2C 的 SDA 输出电平。<br>0: SDA 输出低电平;<br>1: SDA 输出高电平。          | 0x1   |
| [3]    | RO     | i2c_scl_in    | 监测外部 I2C 总线 SCL 的电平。<br>0: 总线上 SCL 为低电平;<br>1: 总线上 SCL 为高电平。    | 0x1   |
| [2]    | RW     | force_sda_oen | 在 gpio_mode 有效时控制 SDA 管脚电平。<br>0: SDA 管脚为低电平;<br>1: SDA 管脚为高电平。 | 0x1   |
| [1]    | RW     | force_scl_oen | 在 gpio_mode 有效时控制 SCL 管脚电平。<br>0: SCL 管脚为低电平;<br>1: SCL 管脚为高电平。 | 0x1   |



| Bits | Access | Name      | Description                                                                                     | Reset |
|------|--------|-----------|-------------------------------------------------------------------------------------------------|-------|
| [0]  | RW     | gpio_mode | 将 I2C 的 SCL 和 SDA 管脚配置成 gpio 模式，管脚电平由 force_scl_oe_n 和 force_sda_oe_n 决定。<br><br>0：禁止；<br>1：使能。 | 0x0   |

## I2C\_FIFO\_STAT

I2C\_FIFO\_STAT 为 I<sup>2</sup>C FIFO 状态寄存器。

Offset Address: 0x000c    Total Reset Value: 0x0000\_000A

| Bits    | Access | Name              | Description                    | Reset |
|---------|--------|-------------------|--------------------------------|-------|
| [31:19] | RO     | reserved          | 保留                             | 0x000 |
| [18:12] | RO     | tx_fifo_vld_num   | TX FIFO 里的有效数目。                | 0x00  |
| [11]    | RO     | reserved          | 保留。                            | 0x0   |
| [10:4]  | RO     | rx_fifo_vld_num   | RX FIFO 里的有效数目。                | 0x00  |
| [3]     | RO     | tx_fifo_not_full  | TX FIFO 非满指示。<br>0：满；<br>1：非满。 | 0x1   |
| [2]     | RO     | tx_fifo_not_empty | TX FIFO 非空指示。<br>0：空；<br>1：非空。 | 0x0   |
| [1]     | RO     | rx_fifo_not_full  | RX FIFO 非满指示。<br>0：满；<br>1：非满。 | 0x1   |
| [0]     | RO     | rx_fifo_not_empty | RX FIFO 非空指示。<br>0：空；<br>1：非空。 | 0x0   |

## I2C\_TX\_FIFO

I2C\_TX\_FIFO 为 I<sup>2</sup>C TX FIFO 数据寄存器。

Offset Address: 0x0010    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description                    | Reset    |
|--------|--------|----------|--------------------------------|----------|
| [31:8] | RO     | reserved | 保留。                            | 0x000000 |
| [7:0]  | WO     | tx_fifo  | TX FIFO 入口。<br>写入的数据是待发送出去的数据。 | 0x00     |

## I2C\_RX\_FIFO

I2C\_RX\_FIFO 为 I<sup>2</sup>C RX FIFO 数据寄存器。

Offset Address: 0x0014    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description                         | Reset    |
|--------|--------|----------|-------------------------------------|----------|
| [31:8] | RO     | reserved | 保留。                                 | 0x000000 |
| [7:0]  | RO     | rx_fifo  | RX FIFO 出口。<br>读出的数据是 I2C 总线上收到的数据。 | 0x00     |

## I2C\_TX\_WATERMARK

I2C\_TX\_WATERMARK 为 I<sup>2</sup>C TX FIFO 水线寄存器。

Offset Address: 0x0018    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name         | Description | Reset     |
|--------|--------|--------------|-------------|-----------|
| [31:6] | RO     | reserved     | 保留。         | 0x0000000 |
| [5:0]  | RW     | tx_watermark | TX FIFO 水线。 | 0x00      |

## I2C\_RX\_WATERMARK

I2C\_RX\_WATERMARK 为 I<sup>2</sup>C RX FIFO 水线寄存器。

Offset Address: 0x001C    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name         | Description                                                                     | Reset     |
|--------|--------|--------------|---------------------------------------------------------------------------------|-----------|
| [31:6] | RO     | reserved     | 保留。                                                                             | 0x0000000 |
| [5:0]  | RW     | rx_watermark | RX FIFO 水线。当 RX FIFO 里的数据数目比 rx_watermark 大时,<br>rx_gt_watermark_raw 原始中断会被置 1。 | 0x00      |

## I2C\_HCNT

I2C\_HCNT 为 I2C 高电平时长寄存器。

Offset Address: 0x00204 Total Reset Value: 0x0000\_0000

| Bits    | Access | Name              | Description                                                                                                                                                                                                                                                                                                                                                 | Reset  |
|---------|--------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:16] | RO     | reserved          | 保留                                                                                                                                                                                                                                                                                                                                                          | 0x0000 |
| [15:0]  | RW     | i2c_high_duration | SCL 高电平时长。<br>参考时钟为 50MHz。<br>建议在标准模式下为 SCL 周期的 1/2，快速模式下为 SCL 周期的 0.36。<br>标准模式下：<br>$i2c\_high\_duration = (F_{PCLK}/F_{SCL}) \times 0.5$ ;<br>快速模式下：<br>$i2c\_high\_duration = (F_{PCLK}/F_{SCL}) \times 0.36$ 。<br>( $F_{SCL}$ 为 SCL 总线的频率)<br>以参考时钟为 50MHz 且工作在快速模式下 ( $F_{SCL}=400KHz$ )为例, $i2c\_high\_duration = (50MHz / 400KHz) \times 0.36=45$ 。 | 0x0000 |

## I2C\_LCNT

I2C\_LCNT 为 I2C 低电平时长寄存器。

Offset Address: 0x0024 Total Reset Value: 0x0000\_0000

| Bits    | Access | Name     | Description | Reset  |
|---------|--------|----------|-------------|--------|
| [31:16] | RO     | reserved | 保留。         | 0x0000 |

| Bits   | Access | Name             | Description                                                                                                                                                                                                                                                                                                                                                                                                 | Reset  |
|--------|--------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [15:0] | RW     | i2c_low_duration | <p>SCL 低电平时长。</p> <p>参考时钟为 50MHz。</p> <p>建议在标准模式下为 SCL 周期的 1/2，快速模式下为 SCL 周期的 0.64。</p> <p>标准模式下：</p> $i2c\_low\_duration = (F_{PCLK}/F_{SCL}) \times 0.5;$ <p>快速模式下：</p> $i2c\_low\_duration = (F_{PCLK}/F_{SCL}) \times 0.64.$ <p>(<math>F_{SCL}</math> 为 SCL 总线的频率)</p> <p>以参考时钟为 50MHz 且工作在快速模式下 (<math>F_{SCL}=400KHz</math>)为例，<math>i2c\_low\_duration = (50MHz / 400KHz) \times 0.64=80</math>。</p> | 0x0000 |

## I2C\_DEV\_ADDR

I2C\_DEV\_ADDR 为 I<sup>2</sup>C 器件地址寄存器。

Offset Address: 0x0028    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name           | Description | Reset |
|---------|--------|----------------|-------------|-------|
| [31:24] | RW     | dev_addr_byte4 | 器件地址字节 4。   | 0x00  |
| [23:16] | RW     | dev_addr_byte3 | 器件地址字节 3。   | 0x00  |
| [15:8]  | RW     | dev_addr_byte2 | 器件地址字节 2。   | 0x00  |
| [7:0]   | RW     | dev_addr_byte1 | 器件地址字节 1。   | 0x00  |

## I2C\_DATA\_BUF

I2C\_DATA\_BUF 为 I<sup>2</sup>C 数据 buf 寄存器。

Offset Address: 0x002C    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name           | Description  | Reset |
|---------|--------|----------------|--------------|-------|
| [31:24] | RW     | data_buf_byte4 | 数据 buf 字节 4。 | 0x00  |

|         |    |                |              |      |
|---------|----|----------------|--------------|------|
| [23:16] | RW | data_buf_byte3 | 数据 buf 字节 3。 | 0x00 |
| [15:8]  | RW | data_buf_byte2 | 数据 buf 字节 2。 | 0x00 |
| [7:0]   | RW | data_buf_byte1 | 数据 buf 字节 1。 | 0x00 |

## I2C\_INTR\_RAW

I2C\_INTR\_RAW 为 I<sup>2</sup>C 原始中断寄存器。

Offset Address: 0x0030    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name                | Description                                          | Reset   |
|--------|--------|---------------------|------------------------------------------------------|---------|
| [31:6] | RO     | reserved            | 保留。                                                  | 0x00000 |
| [5]    | RWC    | rx_gt_watermark_raw | 接收数据时, RX FIFO 里的数据数目高于<br>水线。<br>0: 无中断;<br>1: 有中断。 | 0x0     |
| [4]    | RWC    | all_cmd_done_raw    | 时序命令序列执行到 EXIT 命令时产生此中<br>断。<br>0: 无中断;<br>1: 有中断。   | 0x0     |
| [3]    | RWC    | arb_lost_raw        | 仲裁丢失。<br>0: 无中断;<br>1: 有中断。                          | 0x0     |
| [2]    | RWC    | stop_det_raw        | 检测到 STOP。<br>0: 无中断;<br>1: 有中断。                      | 0x0     |
| [1]    | RWC    | start_det_raw       | 检测到 START。<br>0: 无中断;<br>1: 有中断。                     | 0x0     |
| [0]    | RWC    | ack_bit_unmatch_raw | 应答位不符合预期。                                            | 0x0     |

| Bits | Access | Name | Description        | Reset |
|------|--------|------|--------------------|-------|
|      |        |      | 0: 无中断;<br>1: 有中断。 |       |

## I2C\_INTR\_EN

I2C\_INTR\_EN 为 I<sup>2</sup>C 中断使能寄存器。

Offset Address: 0x0034    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name               | Description                                | Reset   |
|--------|--------|--------------------|--------------------------------------------|---------|
| [31:6] | RO     | reserved           | 保留。                                        | 0x00000 |
| [5]    | RW     | rx_gt_watermark_en | rx_gt_watermark 中断使能。<br>0: 不使能;<br>1: 使能。 | 0x0     |
| [4]    | RW     | all_cmd_done_en    | all_cmd_done 中断使能。<br>0: 不使能;<br>1: 使能。    | 0x0     |
| [3]    | RW     | arb_lost_en        | arb_lost 中断使能。<br>0: 不使能;<br>1: 使能。        | 0x0     |
| [2]    | RW     | stop_det_en        | STOP 中断使能。<br>0: 不使能;<br>1: 使能。            | 0x0     |
| [1]    | RW     | start_det_en       | start_det 中断使能。<br>0: 不使能;<br>1: 使能。       | 0x0     |
| [0]    | RW     | ack_bit_unmatch_en | ack_bit_unmatch 中断使能。<br>0: 不使能;<br>1: 使能。 | 0x0     |

## I2C\_INTR\_STAT

I2C\_INTR\_STAT 为 I<sup>2</sup>C 中断状态寄存器。

Offset Address: 0x0038    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name            | Description                                          | Reset   |
|--------|--------|-----------------|------------------------------------------------------|---------|
| [31:7] | RO     | reserved        | 保留。                                                  | 0x00000 |
| [5]    | RO     | rx_gt_watermark | 接收数据时, RX FIFO 里的数据数目高于<br>水线。<br>0: 无中断;<br>1: 有中断。 | 0x0     |
| [4]    | RO     | all_cmd_done    | 所有时序命令正常完成。<br>0: 无中断;<br>1: 有中断。                    | 0x0     |
| [3]    | RO     | arb_lost        | 仲裁丢失。<br>0: 无中断;<br>1: 有中断。                          | 0x0     |
| [2]    | RO     | stop_det        | 检测到 STOP。<br>0: 无中断;<br>1: 有中断。                      | 0x0     |
| [1]    | RO     | start_det       | 检测到 START。<br>0: 无中断;<br>1: 有中断。                     | 0x0     |
| [0]    | RO     | ack_bit_unmatch | 应答位不符合预期。<br>0: 无中断;<br>1: 有中断。                      | 0x0     |

## I2C\_TIMING\_CMD

I2C\_TIMING\_CMD 为 I<sup>2</sup>C 时序命令寄存器。

Offset Address: 0x0040+n x 4    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset     |
|--------|--------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| [31:5] | RO     | reserved   | 保留。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0x0000000 |
| [4:0]  | RW     | timing_cmd | <p>时序命令。n = 0, 1, 2, ... 31.</p> <p>0x00: EXIT(结束, 用于逻辑退出);</p> <p>0x01: S (总线 START);</p> <p>0x02: WDA4 (发送 dev_addr_byte4);</p> <p>0x03: WDA3 (发送 dev_addr_byte3);</p> <p>0x04: WDA2 (发送 dev_addr_byte2);</p> <p>0x05: WDA1 (发送 dev_addr_byte1);</p> <p>0x06: WDB4 (发送 data_buf_byte4);</p> <p>0x07: WDB3 (发送 data_buf_byte3);</p> <p>0x08: WDB2 (发送 data_buf_byte2);</p> <p>0x09: WDB1 (发送 data_buf_byte1);</p> <p>0x0A: WPD8 (发送 pattern_data_byte8);</p> <p>0x0B: WPD7 (发送 pattern_data_byte7);</p> <p>0x0C: WPD6 (发送 pattern_data_byte6);</p> <p>0x0D: WPD5 (发送 pattern_data_byte5);</p> <p>0x0E: WPD4 (发送 pattern_data_byte4);</p> <p>0x0F: WPD3 (发送 pattern_data_byte3);</p> <p>0x10: WPD2 (发送 pattern_data_byte2);</p> <p>0x11: WPD1 (发送 pattern_data_byte1);</p> <p>0x12: RD (接收 1 字节数据);</p> <p>0x13: RACK (接收低电平应答);</p> | 0x00      |



|  |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |  |
|--|--|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
|  |  |  | <p>0x14: RNACK (接收高电平非应答);</p> <p>0x15: RNC (接收应答, 高低电平都无所谓);</p> <p>0x16: SACK (发送低电平应答);</p> <p>0x17: SNACK (发送高电平非应答);</p> <p>0x18: JMPN1(有限次数跳转, 目的由 DST1 寄存器指出,次数由 LOOP1 寄存器指出);</p> <p>0x19: JMPN2(有限次数跳转, 目的由 DST2 寄存器指出,次数由 LOOP2 寄存器指出);</p> <p>0x1A: JMPN3(有限次数跳转, 目的由 DST3 寄存器指出,次数由 LOOP3 寄存器指出);</p> <p>0x1B: UNDEF (未定义);</p> <p>0x1C: UNDEF (未定义);</p> <p>0x1D: UDB1 (从 TX FIFO 更新数据到 data_buf_byte1);</p> <p>0x1E: SR (总线 repeated START);</p> <p>0x1F: P(总线 STOP)。</p> <p><b>注意:</b></p> <p><b>执行 UDB1 命令时, 若 TX FIFO 为空, 则控制器会等待直到 TX FIFO 有数据; 等待期间控制器不会改变 I2C 总线的状态。</b></p> <p><b>执行 RD 命令时, 若 RX FIFO 满了, 则控制器会等待直到 RX FIFO 有空位; 等待期间控制器不会改变 I2C 总线的状态。</b></p> |  |
|--|--|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

## I2C\_LOOP1

I2C\_LOOP1 为 I<sup>2</sup>C 循环次数 1 寄存器。

Offset Address: 0x00C0    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name      | Description | Reset      |
|--------|--------|-----------|-------------|------------|
| [31:0] | RW     | loop_num1 | 指定循环次数。     | 0x00000000 |

## I2C\_DST1

I2C\_DST1 为 I<sup>2</sup>C 跳转目的 1 寄存器。

Offset Address: 0x00C4    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name            | Description     | Reset      |
|--------|--------|-----------------|-----------------|------------|
| [31:5] | RW     | reserved        | 保留。             | 0x00000000 |
| [4:0]  | RW     | dst_timing_cmd1 | 指定跳转到哪个时序命令寄存器。 | 0x00       |

## I2C\_LOOP2

I2C\_LOOP2 为 I<sup>2</sup>C 循环次数 2 寄存器。

Offset Address: 0x00C8    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name      | Description | Reset      |
|--------|--------|-----------|-------------|------------|
| [31:0] | RW     | loop_num2 | 指定循环次数。     | 0x00000000 |

## I2C\_DST2

I2C\_DST2 为 I<sup>2</sup>C 跳转目的 2 寄存器。

Offset Address: 0x00CC    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name            | Description     | Reset      |
|--------|--------|-----------------|-----------------|------------|
| [31:5] | RW     | reserved        | 保留。             | 0x00000000 |
| [4:0]  | RW     | dst_timing_cmd2 | 指定跳转到哪个时序命令寄存器。 | 0x00       |

## I2C\_LOOP3

I2C\_LOOP3 为 I<sup>2</sup>C 循环次数 3 寄存器。

Offset Address: 0x00D0    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name      | Description | Reset      |
|--------|--------|-----------|-------------|------------|
| [31:0] | RW     | loop_num3 | 指定循环次数。     | 0x00000000 |

## I2C\_DST3

I2C\_DST3 为 I<sup>2</sup>C 跳转目的 3 寄存器。

Offset Address: 0x00D4    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name            | Description     | Reset      |
|--------|--------|-----------------|-----------------|------------|
| [31:5] | RW     | reserved        | 保留。             | 0x00000000 |
| [4:0]  | RW     | dst_timing_cmd3 | 指定跳转到哪个时序命令寄存器。 | 0x00       |

## I2C\_BUS\_FREE\_THR

I2C\_BUS\_FREE\_THR 为 I<sup>2</sup>C 总线空闲阈值寄存器。

Offset Address: 0x00D8    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name         | Description | Reset |
|--------|--------|--------------|-------------|-------|
| [31:0] | RW     | bus_free_thr | I2C 总线空闲阈值。 | 0x00  |

## I2C\_FSM\_STAT

I2C\_FSM\_STAT 为 I<sup>2</sup>C 状态机状态寄存器。

Offset Address: 0x00DC    Total Reset Value: 0x1000\_0000

| Bits    | Access | Name         | Description  | Reset |
|---------|--------|--------------|--------------|-------|
| [31]    | RO     | reserved     | 保留           | 0x01  |
| [30:28] | RO     | l2c_bus_idle | I2c_bus 空闲状态 | 0x00  |
| [27:25] | RO     | reserved     | 保留           | 0x00  |
| [24:20] | RO     | cur_cmd      | 当前命令         | 0x00  |
| [19:16] | RO     | fsm_state    | 状态机状态        | 0x00  |
| [15]    | RO     | reserved     | 保留           | 0x00  |

| Bits    | Access | Name      | Description | Reset |
|---------|--------|-----------|-------------|-------|
| [14:12] | RO     | txp_state | txp 状态      | 0x00  |
| [11]    | RO     | reserved  | 保留          | 0x00  |
| [10:8]  | RO     | rxn_state | rxn 状态      | 0x00  |
| [7]     | RO     | reserved  | 保留          | 0x00  |
| [6:4]   | RO     | txd_state | txp 状态      | 0x00  |
| [3]     | RO     | reserved  | 保留          | 0x00  |
| [2:0]   | RO     | txs_state | txs 状态      | 0x00  |

## I2C\_PATTERN\_DATA1

I2C\_PATTERN\_DATA1 为 I<sup>2</sup>C PATTERN 数据 1 寄存器。

Offset Address: 0x00E0    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name               | Description     | Reset |
|---------|--------|--------------------|-----------------|-------|
| [31:24] | RW     | pattern_data_byte4 | PATTERN 数据字节 4。 | 0x00  |
| [23:16] | RW     | pattern_data_byte3 | PATTERN 数据字节 3。 | 0x00  |
| [15:8]  | RW     | pattern_data_byte2 | PATTERN 数据字节 2。 | 0x00  |
| [7:0]   | RW     | pattern_data_byte1 | PATTERN 数据字节 1。 | 0x00  |

## I2C\_PATTERN\_DATA2

I2C\_PATTERN\_DATA2 为 I<sup>2</sup>C PATTERN 数据 2 寄存器。

Offset Address: 0x00E4    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name               | Description     | Reset |
|---------|--------|--------------------|-----------------|-------|
| [31:24] | RW     | pattern_data_byte8 | PATTERN 数据字节 8。 | 0x00  |
| [23:16] | RW     | pattern_data_byte7 | PATTERN 数据字节 7。 | 0x00  |
| [15:8]  | RW     | pattern_data_byte6 | PATTERN 数据字节 6。 | 0x00  |

|       |    |                    |                 |      |
|-------|----|--------------------|-----------------|------|
| [7:0] | RW | pattern_data_byte5 | PATTERN 数据字节 5。 | 0x00 |
|-------|----|--------------------|-----------------|------|

## 2.2 UART

### 2.2.1 概述

通用异步收发器 UART (Universal Asynchronous Receiver Transmitter) 是一个异步串行的通信接口，主要功能是将来自外围设备的数据进行串并转换之后传入内部总线，以及将数据进行并串转换之后输出到外部设备。UART 的主要功能是和外部芯片的 UART 进行对接，从而实现两芯片间的通信。

芯片供 4 个 UART 控制器：

- UART0/2/3：2 线 UART，主要用于调试。
- UART1：4 线 UART。

#### 须知

当使用 UART 功能时，需要在 UART 初始化前对 RXD 管脚配置为上拉使能。

### 2.2.2 特点

UART 模块有以下特点：

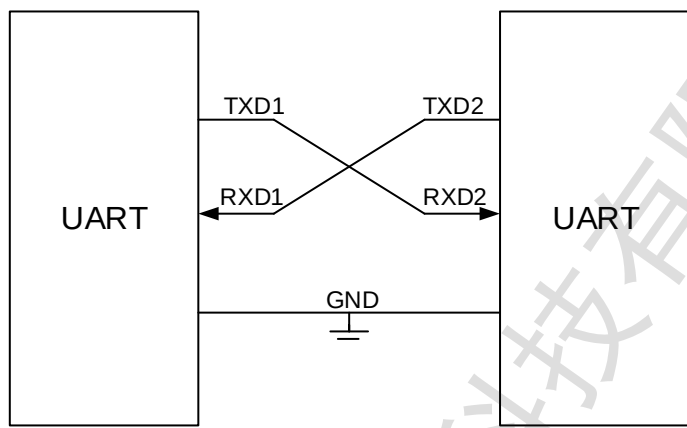
- 支持 32 x 8bit 的发送 FIFO 和 32 x 12bit 的接收 FIFO。
- 支持数据位和停止位的位宽可编程。数据位可通过编程设定为 5/6/7/8 比特；停止位可通过编程设定为 1bit 或 2bit。
- 支持奇、偶校验方式或者无校验。
- 支持传输速率可编程。
- 支持接收 FIFO 中断、发送 FIFO 中断、接收超时中断、错误中断。
- 支持初始中断状态查询和屏蔽后中断状态查询。
- 支持通过编程禁止 UART 模块或者 UART 发送/接收功能以降低功耗。
- 支持关断 UART 时钟以节省功耗。
- 支持 DMA 操作。
- 最大波特率：4Mbps，最小波特率：110bps

### 2.2.3 功能描述

#### 应用框图

UART 的典型应用框图如图 2-7 所示。

图2-7 UART 的典型应用框图

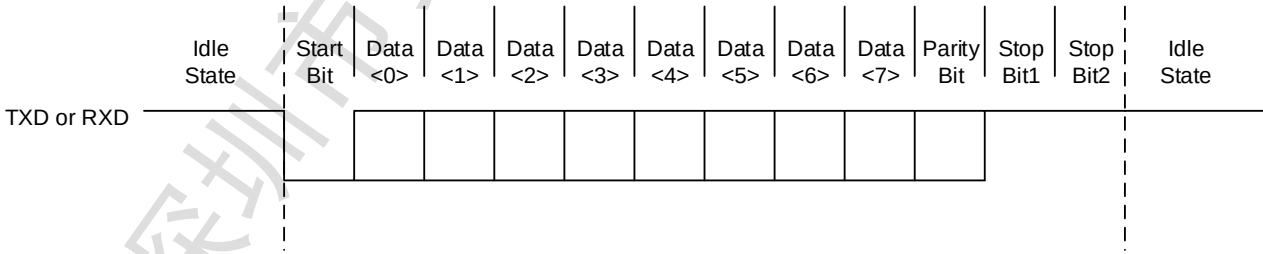


UART 是一种异步双向串行总线，它提供了一种简单有效的数据传输方式，只需要两根数据线互相对接。

#### 功能原理

UART 的一次帧传输主要包括起始信号、数据、校验位和结束信号，如图 2-8 所示。数据帧从某一 UART 的 TXD 端输出，从另一个 UART 的 RXD 端输入。

图2-8 UART 帧格式



起始信号、数据、校验位和结束信号的含义如下：

- 起始信号 (start bit)

一个数据帧开始的标志，UART 协议规定 TXD 信号出现一个低电平就表示一个数据帧的开始。在 UART 不传输数据时，应该保持高电平。

- 数据信号 (data bit)

数据位宽可以根据不同的应用要求进行调整，可以配置成 5bit/6bit/7bit/8bit 数据位宽。

- 校验位 (parity bit)

校验位是 1 比特纠错信号，UART 的校验位有奇校验、偶校验和固定校验位，同时支持校验位的使能和禁止，详细描述请见 UART\_LCR\_H 寄存器。

- 结束信号 (stop bit)

结束信号即数据帧的停止位，支持 1 比特和 2 比特停止位两种配置。数据帧的结束信号就是把 TXD 拉成高电平。

## 2.2.4 工作方式

### 2.2.4.1 波特率配置

通过配置寄存器 UART\_IBRD 和 UART\_FBRD 可以设置 UART 工作的波特率，波特率计算公式为：

当前波特率=UART 参考时钟频率（默认为 24MHz） / （16 x 分频系数）

分频系数有整数和小数两部分组成，分别对应寄存器 UART\_IBRD 和 UART\_FBRD。

例如：UART 参考时钟频率为 24MHz，如果配置 UART\_IBRD 为 0x1E，UART\_FBRD 为 0x00，按照波特率计算公式，则当前的波特率为  $24 / (16 \times 30) = 0.005\text{Mbit/s}$ 。

UART 波特率配置的典型值为：9,600bit/s、14,400bit/s、19,200bit/s、38,400bit/s、57,600bit/s、76,800bit/s、115,200bit/s、230,400bit/s、460,800bit/s。

分频系数值的计算以及分频系数寄存器的配置举例如下：

如果要求波特率为 230,400bit/s，并且 UART 参考时钟频率为 24MHz，那么分频系数为  $(24 \times 10^6) / (16 \times 230400) = 6.5104$ ，因此 IBRD（整数部分）为 6，FBRD（小数部分）为 0.5104。

计算 6bit UART\_FBRD 寄存器中的数值：根据  $m = \text{integer}(\text{FBRD} \times 2^n + 0.5)$  ( $n = \text{UART\_FBRD}$  寄存器的宽度)，计算出  $m = \text{integer}(0.5104 \times 2^6 + 0.5) = 33$ ，在 UART\_IBRD 寄存器中配置 0x0006，UART\_FBRD 寄存器中配置 0x21。

当分频系数小数部分配置成 33 时，波特率除数的实际数值为  $6+33/2^6=6.5156$ ，产生的波特率为  $(24 \times 10^6) / (16 \times 6.5156) = 230216.7107$ ，误差率为  $(230216.7107 - 230400) / 230400 \times 100 = -0.07956\%$ 。

使用 6bitUART\_FBRD 寄存器最大的误差率为  $1/2^6 \times 100 = 1.56\%$ ，当  $m=1$  时会出现，误差率累计超过 64 个时钟周期。

#### 2.2.4.2 软复位



说明

复位寄存器请参考 CRG 寄存器描述。

复位后各配置寄存器的值均为默认值，因此复位后需要重新对这些寄存器进行初始化配置。

#### 2.2.4.3 中断或查询方式下的数据传输

##### 初始化

初始化步骤如下：

步骤 1 向 UART\_CR bit[0]写 0，使 UART 处于禁止状态。

步骤 2 写相应的配置值到 UART\_IBRD、UART\_FBRD 寄存器，配置传输速率。

步骤 3 配置 UART\_CR、UART\_LCR\_H，设定相应的 UART 工作模式。

步骤 4 配置 UART\_IFLS 设定相应的发送及接收 FIFO 阈值。

步骤 5 如果驱动程序采用中断方式则需设定 UART\_IMSC，使能相应中断信号；采用查询方式时应禁止产生相应中断信号。

步骤 6 向 UART\_CR bit[0]写 1，使能 UART，完成初始化配置。

----结束

##### 数据发送

数据发送步骤如下：

步骤 1 将发送数据写入 UART\_DR，启动数据发送。

步骤 2 查询方式下，进行连续数据发送时通过读取 UART\_FR bit[5]检测 TX\_FIFO 状态，根据 TX\_FIFO 的状态决定是否向 TX\_FIFO 中发送数据；中断方式下，则根据相应中断状态位检测；决定是否向 TX\_FIFO 中发送数据。



步骤 3 通过检测 UART\_FR bit[7]是否为 1，判断 UART 是否完成全部数据发送。

----结束

## 数据接收

数据接收的处理方式如下：

- 查询方式下，进行数据接收时通过读取 UART\_FR [rxfe]检测 RX\_FIFO 状态，根据 RX\_FIFO 的状态决定是否读取 RX\_FIFO 中的数据。
- 中断方式下，则根据相应中断状态位检测决定是否读取 RX\_FIFO 中的数据。

### 2.2.4.4 DMA 方式下的数据传输

## 初始化

初始化步骤如下：

步骤 1 向 UART\_CR[uarten]写 0，使 UART 处于禁止状态。

步骤 2 写相应的配置值到 UART\_IBRD、UART\_FBRD 寄存器，配置传输速率。

步骤 3 配置 UART\_CR、UART\_LCR\_H，设定相应的 UART 工作模式。

步骤 4 配置 UART\_IFLS 设定相应的发送及接收 FIFO 阈值。

步骤 5 如果驱动程序采用中断方式则需设定 UART\_IMSC，使能相应中断信号；采用查询方式时应禁止产生相应中断信号。

步骤 6 向 UART\_CR [uarten]写 1，使能 UART，完成初始化配置。

----结束

## 数据发送

数据发送（以 DMA 模式为例）步骤如下：

步骤 1 配置 DMA 数据通道，包括数据传输源和目的地址、数据传输个数、传输类型等参数。具体配置时请参见“直接存储器存取控制器”的相关描述。

步骤 2 配置 UART\_DMACR 为 0x2，使能 UART 的 DMA 发送功能。

步骤 3 通过 DMA 中断上报，判断数据是否发送完成，如果完成则关闭 UART 的 DMA 发送功能。

----结束

数据接收

数据接收（以 DMA 模式为例）步骤如下：

- 步骤 1 配置 DMA 数据通道，包括数据传输源和目的地址、数据接收区地址、数据传输个数、传输类型等参数。
- 步骤 2 配置 UART\_DMCCR 为 0x1，使能 UART 的 DMA 接收功能。
- 步骤 3 通过 DMA 状态查询，判断数据是否接收完成，如果完成则关闭 UART 的 DMA 接收功能。

----结束

2.2.5 UART 寄存器概览

地址空间内：

- UART0 寄存器基地址为：0x1204\_0000。
- UART1 寄存器基地址为：0x1204\_1000。
- UART2 寄存器基地址为：0x1204\_2000。
- UART3 寄存器基地址为：0x1204\_3000。

UART 寄存器概览如表 2-9 所示。

表2-9 UART 寄存器概览

| 偏移地址  | 名称         | 描述              |
|-------|------------|-----------------|
| 0x000 | UART_DR    | 数据寄存器           |
| 0x004 | UART_RSR   | 接收状态寄存器/错误清除寄存器 |
| 0x018 | UART_FR    | 标志寄存器           |
| 0x024 | UART_IBRD  | 整数波特率寄存器        |
| 0x028 | UART_FBRD  | 小数波特率寄存器        |
| 0x02C | UART_LCR_H | 线控寄存器           |
| 0x030 | UART_CR    | 控制寄存器           |

| 偏移地址  | 名称         | 描述              |
|-------|------------|-----------------|
| 0x034 | UART_IFLS  | 中断 FIFO 阈值选择寄存器 |
| 0x038 | UART_IMSC  | 中断屏蔽寄存器         |
| 0x03C | UART_RIS   | 原始中断状态寄存器       |
| 0x040 | UART_MIS   | 屏蔽后中断状态寄存器      |
| 0x044 | UART_ICR   | 中断清除寄存器         |
| 0x048 | UART_DMACR | DMA 控制寄存器       |

## 2.2.6 UART 寄存器描述

### UART\_DR

UART\_DR 为 UART 数据寄存器，存放接收数据和发送数据，同时可以从该寄存器中读出接收状态。

Offset Address: 0x000    Total Reset Value: 0x0000

| Bits    | Access | Name     | Description                                                                                            | Reset |
|---------|--------|----------|--------------------------------------------------------------------------------------------------------|-------|
| [15:12] | RO     | reserved | 保留。                                                                                                    | 0x0   |
| [11]    | RO     | oe       | 溢出错误。<br>0: 无溢出错误;<br>1: 有溢出错误, 接收 FIFO 满后接收到了数据。                                                      | 0x0   |
| [10]    | RO     | be       | Break 错误。<br>0: 无 break 错误;<br>1: 有 break 错误, 即接收数据的输入保持低的时间比一个全字传输(包括 start、data、parity、stop bit)还要长。 | 0x0   |
| [9]     | RO     | pe       | 校验错误。<br>0: 无校验错误;<br>1: 有校验错误。                                                                        | 0x0   |

| Bits  | Access | Name | Description                          | Reset |
|-------|--------|------|--------------------------------------|-------|
| [8]   | RO     | fe   | 帧错误。<br>0: 无帧错误;<br>1: 有帧错误(错误的停止位)。 | 0x0   |
| [7:0] | RW     | data | 接收数据和发送数据。                           | 0x00  |

## UART\_RSR

UART\_RSR 为接收状态寄存器/错误清除寄存器。

- 寄存器读时作为接收状态寄存器。
- 寄存器写时作为错误清除寄存器。

接收状态也可以从 UART\_DR 中读出。从 UART\_DR 中读出的 break、frame、parity 的状态信息要比从 UART\_RSR 读出的信息优先级高 (即 UART\_DR 中的状态变化比 UART\_RSR 更快)。

对 UART\_RSR 寄存器的任何写操作都会对 UART\_RSR 寄存器进行复位。

Offset Address: 0x004    Total Reset Value: 0x00

| Bits  | Access | Name     | Description                                                                                                      | Reset |
|-------|--------|----------|------------------------------------------------------------------------------------------------------------------|-------|
| [7:4] | RO     | reserved | 保留。                                                                                                              | 0x0   |
| [3]   | RW     | oe       | 溢出错误。<br>0: 无溢出错误;<br>1: 溢出错误。<br>当 FIFO 满时, FIFO 中的内容保持有效, 因为不会有下一个数据写到 FIFO 中, 只是移位寄存器会溢出。CPU 必须立刻读数据以腾空 FIFO。 | 0x0   |
| [2]   | RW     | be       | Break 错误。<br>0: 无 break 错误;<br>1: break 错误。<br>Break 的条件: 接收数据的输入保持低的时间比一个全字传输(定义了 start、data、                   | 0x0   |

|     |    |    |                                                                      |     |
|-----|----|----|----------------------------------------------------------------------|-----|
|     |    |    | parity、stop bit)还要长。                                                 |     |
| [1] | RW | pe | 校验错误。<br>0: 无校验错误;<br>1: 接收数据的校验错误。<br>FIFO 模式下, 该错误与 FIFO 顶部的数据相关联。 | 0x0 |
| [0] | RW | fe | 帧错误。<br>0: 无帧错误;<br>1: 接收到的数据的停止位错误(有效的停止位为 1)。                      | 0x0 |

## UART\_FR

UART\_FR 为 UART 标志寄存器。

Offset Address: 0x018    Total Reset Value: 0x0096

| Bits   | Access | Name     | Description                                                                                                                         | Reset |
|--------|--------|----------|-------------------------------------------------------------------------------------------------------------------------------------|-------|
| [15:8] | RO     | reserved | 保留。                                                                                                                                 | 0x00  |
| [7]    | RO     | txfe     | 该位的含义由 UART_LCR_H[fen]的状态决定。<br>如果 UART_LCR_H[fen]为 0, 则当发送 holding register 空时该位置 1;<br>如果 UART_LCR_H[fen]为 1, 则当发送 FIFO 为空时该位置 1。 | 0x1   |
| [6]    | RO     | rxff     | 该位的含义由 UART_LCR_H[fen]的状态决定。<br>如果 UART_LCR_H[fen]为 0, 则当接收 holding register 满时该位置 1;<br>如果 UART_LCR_H[fen]为 1, 则当接收 FIFO 为满时该位置 1。 | 0x0   |
| [5]    | RO     | txff     | 该位的含义由 UART_LCR_H[fen]的状态决定。                                                                                                        | 0x0   |

| Bits  | Access | Name     | Description                                                                                                                                | Reset |
|-------|--------|----------|--------------------------------------------------------------------------------------------------------------------------------------------|-------|
|       |        |          | 如果 UART_LCR_H[fen]为 0, 则当发送 holding register 满时该位置 1;<br>如果 UART_LCR_H[fen]为 1, 当发送 FIFO 为满时该位置 1。                                         |       |
| [4]   | RO     | rxfe     | 该位的含义由 UART_LCR_H[fen]的状态决定。<br>如果 UART_LCR_H[fen]为 0, 则当接收 holding register 空时该 bit 置 1;<br>如果 UART_LCR_H[fen]为 1, 则当接收 FIFO 为空时该位就置 1。   | 0x1   |
| [3]   | RO     | busy     | UART 忙闲状态位。<br>0: UART 空闲或者完成发送数据;<br>1: UART 正忙于发送数据。<br>该位一旦置位, 该状态一直保持到整个字节(包括所有的停止位)完全从移位寄存器中发送出去。<br>一旦发送 FIFO 非空该位就置位, 不管 UART 使能与否。 | 0x0   |
| [2:0] | RO     | reserved | 保留。                                                                                                                                        | 0x6   |

## UART\_IBRD

UART\_IBRD 为整数波特率寄存器。

Offset Address: 0x024 Total Reset Value: 0x0000

| Bits   | Access | Name        | Description        | Reset  |
|--------|--------|-------------|--------------------|--------|
| [15:0] | RW     | baud divint | 整数波特率分频值。复位时全部清 0。 | 0x0000 |

## UART\_FBRD

UART\_FBRD 为小数波特率寄存器。

### 须知

- 整数波特率寄存器和小数波特率寄存器的值必须等到当前数据发送和接收完毕才能更新。
- 最小的分频值为 1，最大的分频值为 65535 ( $2^{16} - 1$ )。即 UART\_IBRD=0 是无效的，而此时 UART\_FBRD 将被忽略。同样，如果 UART\_IBRD=65535 (0xFFFF)，UART\_FBRD 就只能是 0，如果比 0 大，则会导致发送和接收的失败。

Offset Address: 0x028    Total Reset Value: 0x00

| Bits  | Access | Name         | Description        | Reset |
|-------|--------|--------------|--------------------|-------|
| [7:6] | RO     | reserved     | 保留。                | 0x0   |
| [5:0] | RW     | baud divfrac | 小数波特率分频值。复位时全部清 0。 | 0x00  |

## UART\_LCR\_H

UART\_LCR\_H 为线控寄存器，UART\_LCR\_H、UART\_IBRD、UART\_FBRD 组成一个 30bit 宽的寄存器。如果更新 UART\_IBRD 和 UART\_FBRD 的内容，必须同时更新 UART\_LCR\_H。

Offset Address: 0x02C    Total Reset Value: 0x0000

| Bits   | Access | Name     | Description                                                                                                                                                                                                                                                                       | Reset |
|--------|--------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [15:8] | RO     | reserved | 保留。                                                                                                                                                                                                                                                                               | 0x00  |
| [7]    | RW     | sps      | stick parity 选择。 <ul style="list-style-type: none"><li>• 当本寄存器的 bit[1]、bit[2]、bit[7]被置位时，校验位就会作为 0 发送和检测；</li><li>• 当本寄存器的 bit[1]、bit[7]被置位，bit[2]为 0 时，校验位就会作为 1 发送和检测。</li><li>• 当本寄存器的 bit[7]为 0 时，stick parity 禁止。</li></ul> 因为按照现在的说法，其功能应该是检验被禁止，而该 bit 只与 stick parity 相关。 | 0x0   |

| Bits  | Access | Name | Description                                                                                                 | Reset |
|-------|--------|------|-------------------------------------------------------------------------------------------------------------|-------|
| [6:5] | RW     | wlen | 指示发送和接收一个帧里数据比特的数目。<br>00: 5bit;<br>01: 6bit;<br>10: 7bit;<br>11: 8bit。                                     | 0x0   |
| [4]   | RW     | fen  | 发送和接收 FIFO 使能控制。<br>0: 不使能发送和接收 FIFO;<br>1: 发送和接收 FIFO 使能。                                                  | 0x0   |
| [3]   | RW     | stp2 | 发送帧尾 2bit 停止位判断。<br>0: 发送的帧尾没有 2bit 停止位;<br>1: 发送的帧尾有 2bit 停止位。<br>接收逻辑在接收时不检查 2bit 的停止位。                   | 0x0   |
| [2]   | RW     | eps  | 发送和接收过程中的奇偶校验选择。<br>0: 在发送和接收过程中生成奇校验或检查奇校验;<br>1: 在发送和接收过程中生成偶校验或检查偶校验。<br>当 UART_LCR_H[pen]为 0 时, 该位不起作用。 | 0x0   |
| [1]   | RW     | pen  | 校验选择位。<br>0: 不作校验;<br>1: 发送方向产生校验, 接收方向作校验检查。                                                               | 0x0   |
| [0]   | RW     | brk  | 发送 break。<br>0: 无效;<br>1: 在完成当前数据的发送后, UTXD 连续输出低电平。<br>注意:                                                 | 0x0   |



| Bits | Access | Name | Description                                             | Reset |
|------|--------|------|---------------------------------------------------------|-------|
|      |        |      | 要正确的执行 break 命令，软件将该位置 1 的时间必须超过 2 个完整帧；在正常使用中，该位必须清 0。 |       |

## UART\_CR

UART\_CR 为 UART 控制寄存器。

配置 UART\_CR 遵循以下步骤：

步骤 1 向 UART\_CR[uarthen]写 0，禁止 UART。

步骤 2 等待当前数据发送或接收结束。

步骤 3 将 UART\_LCR\_H[fen]清 0。

步骤 4 配置 UART\_CR。

步骤 5 向 UART\_CR[uarthen]写 1，使能 UART。

----结束

Offset Address: 0x030 Total Reset Value: 0x0300

| Bits    | Access | Name     | Description                                                              | Reset |
|---------|--------|----------|--------------------------------------------------------------------------|-------|
| [15]    | RW     | ctsen    | CTS 硬件流控使能。<br>0：不使能 CTS 硬件流控；<br>1：使能 CTS 硬件流控，只有当 nUARTCTS 信号有效时才发送数据。 | 0x0   |
| [14]    | RW     | rtsen    | RTS 硬件流控使能。<br>0：不使能 RTS 硬件流控；<br>1：使能 RTS 硬件流控，只有当接收 FIFO 有空间时才请求接收数据。  | 0x0   |
| [13:12] | RO     | reserved | 保留。                                                                      | 0x0   |
| [11]    | RW     | rts      | 请求发送。<br>该 bit 为 UART modem 状态输出信号 nUARTRTS 的取反。                         | 0x0   |

| Bits  | Access | Name     | Description                                                                                    | Reset |
|-------|--------|----------|------------------------------------------------------------------------------------------------|-------|
|       |        |          | 0: 输出信号不变;<br>1: 即该 bit 配置为 1, 则输出信号为 0。                                                       |       |
| [10]  | RW     | dtr      | 数据发送准备。<br>该 bit 为 UART modem 状态输出信号 nUARTDTR 的取反。<br>0: 输出信号不变;<br>1: 即该 bit 配置为 1, 则输出信号为 0。 | 0x0   |
| [9]   | RW     | rx       | UART 接收使能。<br>0: 禁止;<br>1: 使能。<br>在接收的过程中如果 UART 被禁止, 则当前数据的接收就会在正常停止之前结束。                     | 0x1   |
| [8]   | RW     | tx       | UART 发送使能。<br>0: 禁止;<br>1: 使能。<br>在发送的过程中如果 UART 被禁止, 则当前数据的发送就会在正常停止之前结束。                     | 0x1   |
| [7]   | RW     | lbe      | 环回使能。<br>0: 禁止;<br>1: UARTTXD 输出环回到 UARTRXD。                                                   | 0x0   |
| [6:1] | RO     | reserved | 保留。                                                                                            | 0x00  |
| [0]   | RW     | uarten   | UART 使能。<br>0: 禁止;<br>1: 使能。<br>如果在发送和接收过程中将 UART 禁止, 则会在正常停止之前结束当前数据的传送。                      | 0x0   |

## UART\_IFLS

UART\_IFLS 为中断 FIFO 阈值选择寄存器，用于设置 FIFO 的中断（UART\_TXINTR 或 UART\_RXINTR）触发线。

Offset Address: 0x034    Total Reset Value: 0x0012

| Bits   | Access | Name      | Description                                                                                                                                                                                                   | Reset |
|--------|--------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [15:6] | RO     | reserved  | 保留。                                                                                                                                                                                                           | 0x000 |
| [5:3]  | RW     | rxiflssel | 接收中断 FIFO 的阈值选择，接收中断的触发点如下。<br>000: 接收 FIFO $\geq$ 1/8full;<br>001: 接收 FIFO $\geq$ 1/4full;<br>010: 接收 FIFO $\geq$ 1/2full;<br>011: 接收 FIFO $\geq$ 3/4full;<br>100: 接收 FIFO $\geq$ 7/8full;<br>101 ~ 111: 保留。 | 0x2   |
| [2:0]  | RW     | txiflssel | 发送中断 FIFO 的阈值选择，发送中断的触发点如下。<br>000: 发送 FIFO $\leq$ 1/8full;<br>001: 发送 FIFO $\leq$ 1/4full;<br>010: 发送 FIFO $\leq$ 1/2full;<br>011: 发送 FIFO $\leq$ 3/4full;<br>100: 发送 FIFO $\leq$ 7/8full;<br>101 ~ 111: 保留。 | 0x2   |

## UART\_IMSC

UART\_IMSC 为中断屏蔽寄存器，用于屏蔽中断。

Offset Address: 0x038    Total Reset Value: 0x0000

| Bits    | Access | Name     | Description  | Reset |
|---------|--------|----------|--------------|-------|
| [15:11] | RO     | reserved | 保留。          | 0x00  |
| [10]    | RW     | oeim     | 溢出错误中断的屏蔽状态。 | 0x0   |

| Bits  | Access | Name     | Description                                 | Reset |
|-------|--------|----------|---------------------------------------------|-------|
|       |        |          | 0: 屏蔽该中断;<br>1: 不屏蔽该中断。                     |       |
| [9]   | RW     | beim     | break 错误中断的屏蔽状态。<br>0: 屏蔽该中断;<br>1: 不屏蔽该中断。 | 0x0   |
| [8]   | RW     | peim     | 校验中断的屏蔽状态。<br>0: 屏蔽该中断;<br>1: 不屏蔽该中断。       | 0x0   |
| [7]   | RW     | feim     | 帧错误中断的屏蔽状态。<br>0: 屏蔽该中断;<br>1: 不屏蔽该中断。      | 0x0   |
| [6]   | RW     | rtim     | 接收超时中断的屏蔽状态。<br>0: 屏蔽该中断;<br>1: 不屏蔽该中断。     | 0x0   |
| [5]   | RW     | txim     | 发送中断的屏蔽状态。<br>0: 屏蔽该中断;<br>1: 不屏蔽该中断。       | 0x0   |
| [4]   | RW     | rxim     | 接收中断的屏蔽状态。<br>0: 屏蔽该中断;<br>1: 不屏蔽该中断。       | 0x0   |
| [3:0] | RO     | reserved | 保留。                                         | 0x0   |

## UART\_RIS

UART\_RIS 为原始中断状态寄存器，其内容不受中断屏蔽寄存器的影响。

Offset Address: 0x03C    Total Reset Value: 0x0002

| Bits    | Access | Name     | Description | Reset |
|---------|--------|----------|-------------|-------|
| [15:11] | RO     | reserved | 保留。         | 0x00  |

| Bits  | Access | Name     | Description                                 | Reset |
|-------|--------|----------|---------------------------------------------|-------|
| [10]  | RO     | oeris    | 原始的溢出错误中断状态。<br>0: 未产生中断;<br>1: 已产生中断。      | 0x0   |
| [9]   | RO     | beris    | 原始的 break 错误中断状态。<br>0: 未产生中断;<br>1: 已产生中断。 | 0x0   |
| [8]   | RO     | peris    | 原始的校验中断状态。<br>0: 未产生中断;<br>1: 已产生中断。        | 0x0   |
| [7]   | RO     | feris    | 原始的错误中断状态。<br>0: 未产生中断;<br>1: 已产生中断。        | 0x0   |
| [6]   | RO     | rtris    | 原始的接收超时中断状态。<br>0: 未产生中断;<br>1: 已产生中断。      | 0x0   |
| [5]   | RO     | txris    | 原始的发送中断状态。<br>0: 未产生中断;<br>1: 已产生中断。        | 0x0   |
| [4]   | RO     | rxris    | 原始的接收中断状态。<br>0: 未产生中断;<br>1: 已产生中断。        | 0x0   |
| [3:0] | RO     | reserved | 保留。                                         | 0x2   |

## UART\_MIS

UART\_MIS 为屏蔽后中断状态寄存器，其内容为原始中断状态和中断屏蔽进行“与”操作后的结果。

Offset Address: 0x040    Total Reset Value: 0x0000

| Bits    | Access | Name     | Description                                  | Reset |
|---------|--------|----------|----------------------------------------------|-------|
| [15:11] | RO     | reserved | 保留。                                          | 0x00  |
| [10]    | RO     | oemis    | 屏蔽后的溢出错误中断状态。<br>0: 未产生中断;<br>1: 已产生中断。      | 0x0   |
| [9]     | RO     | bemis    | 屏蔽后的 break 错误中断状态。<br>0: 未产生中断;<br>1: 已产生中断。 | 0x0   |
| [8]     | RO     | pemis    | 屏蔽后的校验中断状态。<br>0: 未产生中断;<br>1: 已产生中断。        | 0x0   |
| [7]     | RO     | femis    | 屏蔽后的错误中断状态。<br>0: 未产生中断;<br>1: 已产生中断。        | 0x0   |
| [6]     | RO     | rtmis    | 屏蔽后的接收超时中断状态。<br>0: 未产生中断;<br>1: 已产生中断。      | 0x0   |
| [5]     | RO     | txmis    | 屏蔽后的发送中断状态。<br>0: 未产生中断;<br>1: 已产生中断。        | 0x0   |
| [4]     | RO     | rxmis    | 屏蔽后的接收中断状态。<br>0: 未产生中断;<br>1: 已产生中断。        | 0x0   |
| [3:0]   | RO     | reserved | 保留。                                          | 0x0   |

## UART\_ICR

UART\_ICR 为中断清除寄存器，写 1 时相应的中断被清除，写 0 则不起作用。

Offset Address: 0x044    Total Reset Value: 0x0000

| Bits    | Access | Name     | Description                          | Reset |
|---------|--------|----------|--------------------------------------|-------|
| [15:11] | RO     | reserved | 保留。                                  | 0x00  |
| [10]    | WO     | oeic     | 清除溢出错误中断。<br>0: 无效;<br>1: 清除中断。      | 0x0   |
| [9]     | WO     | beic     | 清除 break 错误中断。<br>0: 无效;<br>1: 清除中断。 | 0x0   |
| [8]     | WO     | peic     | 清除校验中断。<br>0: 无效;<br>1: 清除中断。        | 0x0   |
| [7]     | WO     | feic     | 清除错误中断。<br>0: 无效;<br>1: 清除中断。        | 0x0   |
| [6]     | WO     | rtic     | 清除接收超时中断。<br>0: 无效;<br>1: 清除中断。      | 0x0   |
| [5]     | WO     | txic     | 清除发送中断。<br>0: 无效;<br>1: 清除中断。        | 0x0   |
| [4]     | WO     | rxic     | 清除接收中断。<br>0: 无效;<br>1: 清除中断。        | 0x0   |
| [3:0]   | RO     | reserved | 保留。                                  | 0x0   |

## UART\_DMCCR

UART\_DMCCR 为 DMA 控制寄存器，用于配置发送 FIFO 和接收 FIFO 的 DMA 使能。

Offset Address: 0x048    Total Reset Value: 0x0000

| Bits   | Access | Name     | Description                                                                                                                                                                                                          | Reset  |
|--------|--------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [15:3] | RO     | reserved | 保留。                                                                                                                                                                                                                  | 0x0000 |
| [2]    | RW     | dmaonerr | UART 错误中断(UARTEINTR)出现时的接收通道 DMA 使能控制。<br><br>0: 当 UART 错误中断(UARTEINTR)有效时, 接收通道 DMA 的请求输出 (UARTRXDMASREQ 或 UARRTXDMABREQ)有效;<br><br>1: 当 UART 错误中断(UARTEINTR)有效时, 接收通道 DMA 的请求输出 (UARTRXDMASREQ 或 UARRTXDMABREQ)无效。 | 0x0    |
| [1]    | RW     | txdmae   | 发送 FIFO 的 DMA 使能控制。<br><br>0: 禁止;<br>1: 使能。                                                                                                                                                                          | 0x0    |
| [0]    | RW     | rxdmae   | 接收 FIFO 的 DMA 使能控制。<br><br>0: 禁止;<br>1: 使能。                                                                                                                                                                          | 0x0    |

## 2.3 SPI

### 2.3.1 概述

SPI(serial peripheral interface)控制器实现数据的串并、并串转换, 可以作为主机或者从机与外部设备进行同步串行通信。支持 MOTOROLA 的 SPI、TI 串行同步、MicroWire 三种外设接口协议。



## 2.3.2 特点

### 须知

- 芯片总共有 3 组 SPI 接口。
- 工作参考时钟为 600M，SPI 做从机时输入的 SPI\_CLK 最大支持 50MHz。

SPI 的功能特点有：

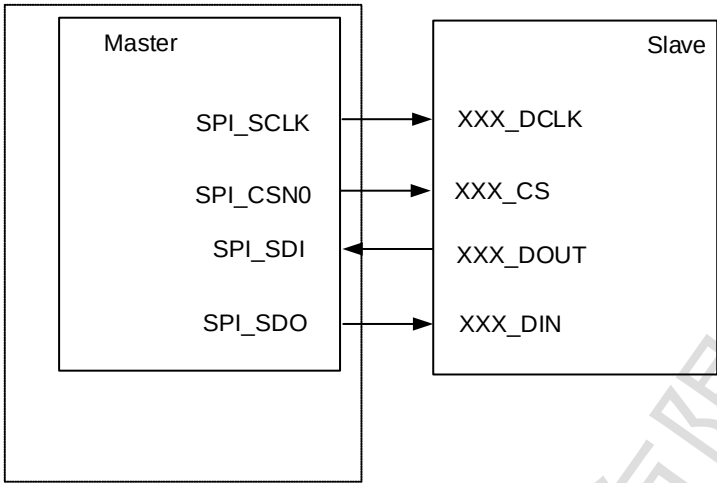
- 接口时钟频率可编程。
- 收/发分开的宽度 16bit、深度为 32 的 FIFO。
- 串行数据帧长度可编程：4bit ~ 16bit。
- 内部提供环回测试模式。
- 支持 DMA 操作。
- 支持 SPI、MicroWire、TI 同步串行三种接口，支持单帧和连续帧格式。
- 支持 SPI 全双工工作模式，时钟极性、相位可配置。
- 支持 MicroWire 半双工工作模式。
- 支持 TI 同步串行接口全双工工作模式。
- 支持 slave 模式，对接 SPI sensor，最高频率 50MHz。

## 2.3.3 功能描述

### 典型应用

SPI 接 Slave 时的应用框图如图 2-9 所示。

图2-9 SPI 接 Slave 时的应用



2.3.4 三种外设总线时序

图 2-10 ~ 图 2-17 中的缩略语含义为：

- MSB: Most Significant Bit
- LSB: Least Significant Bit
- Q: Q is an undefined signal

SPI 接口



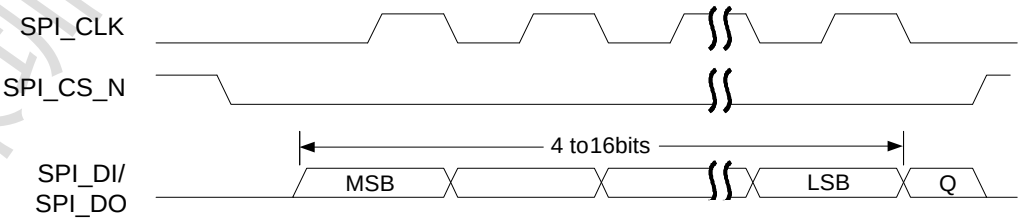
说明

SPO 表示 SPICLKOUT 极性，SPH 表示 SPICLKOUT 相位。它们是寄存器 SPICR0 bit[7:6]。

(1) SPO=0、SPH=0

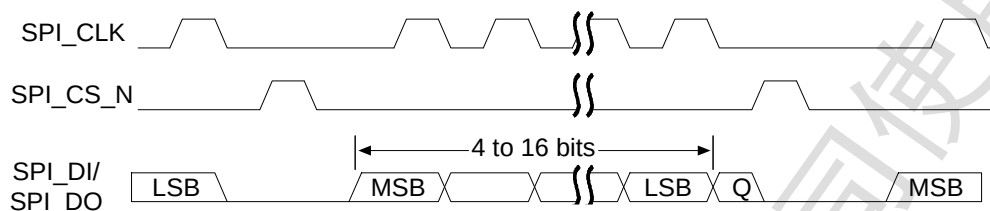
SPI 单帧帧格式如图 2-10 所示。

图2-10 SPI 单帧帧格式 (SPO=0、SPH=0)



SPI 连续帧帧格式如图 2-11 所示。

图2-11 SPI 连续帧格式 (SPO=0、SPH=0)



在该模式下，当 SPI 处于空闲状态时：

- SPI\_CLK 信号设置为低
- SPI\_CS\_N 信号设置为高
- 发送数据线 SPI\_DO 强制为低

当 SPI 处于使能状态，而且发送 FIFO 内存在有效数据时，设置 SPI\_CS\_N 信号为低，表示开始传输数据。来自 Slave 的数据立刻发送到 Master 的接收数据线 SPI\_DI。半个 SPI\_CLK 时钟周期之后，有效的 Master 数据传输到 SPI\_DO。此时 Master 和 Slave 数据都已经有效，SPI\_CLK 管脚在接下来的半个 SPI\_CLK 时钟周期之后变为高电平。数据在 SPI\_CLK 时钟的上升沿被捕获，在时钟的下降沿被传送。

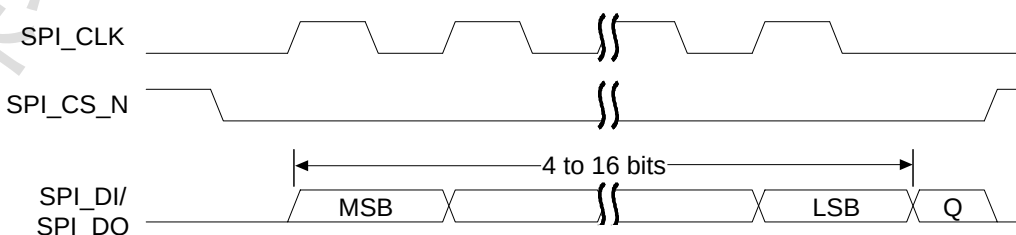
如果传输单个 word，当捕捉到最后 1bit 数据时，SPI\_CS\_N 在接下来的 1 个 SPI\_CLK 时钟周期之后恢复为高电平。

如果是连续的传输，SPI\_CS\_N 信号在每个 word 传输之间必须将 SPI\_CLK 时钟拉高一个时钟周期。这是因为 SPH 为 0 时，Slave 选择管脚会固定其内部串行设备寄存器的数据，使它不会变化。因此在连续传输时，主设备必须在每个 word 传输之间将 SPI\_CS\_N 信号拉高。连续传输结束时，SPI\_CS\_N 在捕捉到最后 1bit 之后的 1 个 SPI\_CLK 时钟周期之后恢复为高电平。

## (2) SPO=0、SPH=1

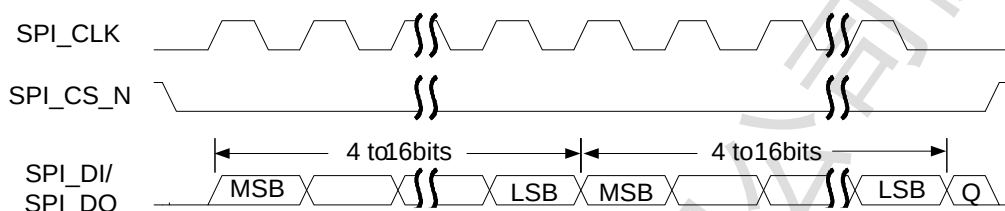
SPI 单帧格式如图 2-12 所示。

图2-12 SPI 单帧格式 (SPO=0、SPH=1)



SPI 连续帧格式如图 2-13 所示。

图2-13 SPI 连续帧格式 (SPO=0、SPH=1)



在该模式下，当 SPI 处于空闲状态时：

- SPI\_CLK 信号设置为低
- SPI\_CS\_N 设置为高
- 发送数据线 SPI\_DO 强制为低

当 SPI 为使能状态，而且发送 FIFO 内有有效数据时，设置 SPI\_CS\_N 信号为低表示开始传输数据。半个 SPI\_CLK 时钟周期之后，Master 和 Slave 的有效数据分别在各自的传输线上有效。同时，SPI\_CLK 从第一个上升沿开始有效。数据在 SPI\_CLK 时钟的下降沿被捕获，在时钟的上升沿被传送。

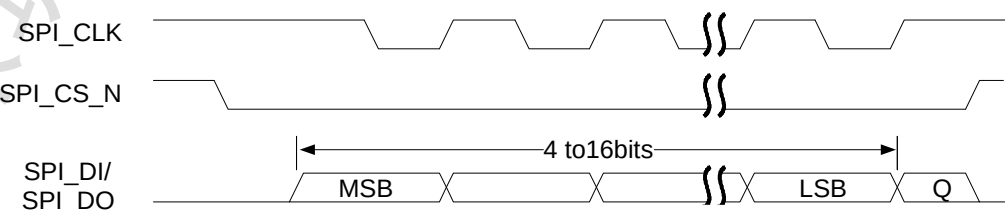
如果传输单个 word，当捕捉到最后 1bit 数据时，SPI\_CS\_N 在接下来的 1 个 SPI\_CLK 时钟之后恢复为高电平。

当连续传输时，在传输数据 word 之间 SPI\_CS\_N 保持为低。连续传输结束时，SP\_CS\_N 在最后 1bit 捕获之后的 1 个 SPI\_CLK 时钟之后恢复为高电平。

### (3) SPO=1、SPH=0

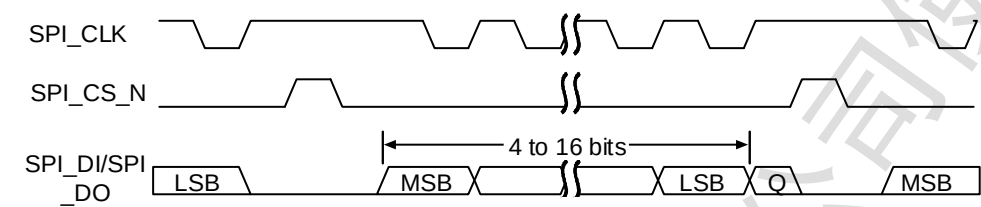
SPI 单帧格式如图 2-14 所示。

图2-14 SPI 单帧格式 (SPO=1、SPH=0)



SPI 连续帧格式如图 2-15 所示。

图2-15 SPI 连续帧格式 (SPO=1、SPH=0)



在该配置下，当 SPI 处于空闲状态时：

- SPI\_CLK 信号设置为高
- SPI\_CS\_N 信号设置为高
- 发送数据线 SPI\_DO 强制为低

当 SPI 为使能状态，而且发送 FIFO 内有有效数据时，设置 SPI\_CS\_N 信号为低表示开始传输数据。此时 Slave 的数据立刻发送到 Master 的接收数据线 SPI\_DI。半个 SPI\_CLK 周期之后，Master 的有效数据传送到 SPI\_DO。再过半个 SPI\_CLK 时钟周期之后，SPI\_CLK Master 管脚设置为低。这表示数据在 SPI\_CLK 时钟的下降沿被捕获，在 SPI\_CLK 时钟的上升沿被传送。

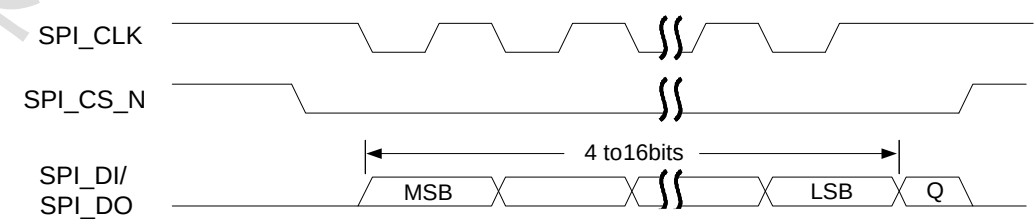
如果传输单个 word，当捕捉到最后 1bit 数据时，SPI\_CS\_N 在接下来的 1 个 SPI\_CLK 时钟之后恢复为高电平。

如果是连续的传输，SPI\_CS\_N 信号在每个 word 传输之间必须拉高。这是因为当 SPH 为 0 时，Salve 选择管脚固定其内部串行设备寄存器的数据，使它不会变化。SPI\_CS\_N 在捕捉到最后 1bit 数据之后的 1 个 SPI\_CLK 时钟周期之后恢复为高电平。

#### (4) SPO=1、SPH=1

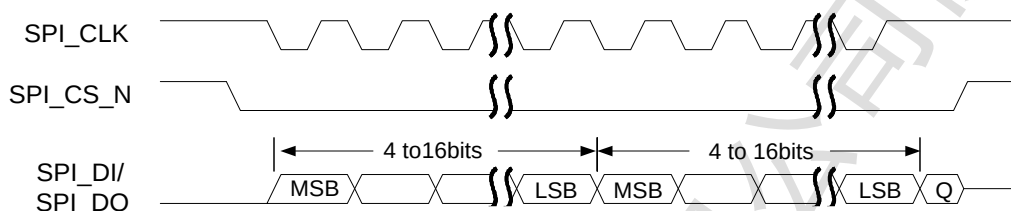
SPI 单帧格式如图 2-16 所示。

图2-16 SPI 单帧格式 (SPO=1、SPH=1)



SPI 连续帧格式如图 2-17 所示。

图2-17 SPI 连续帧格式 (SPO=1、SPH=1)



在该模式下，当 SPI 处于空闲状态时：

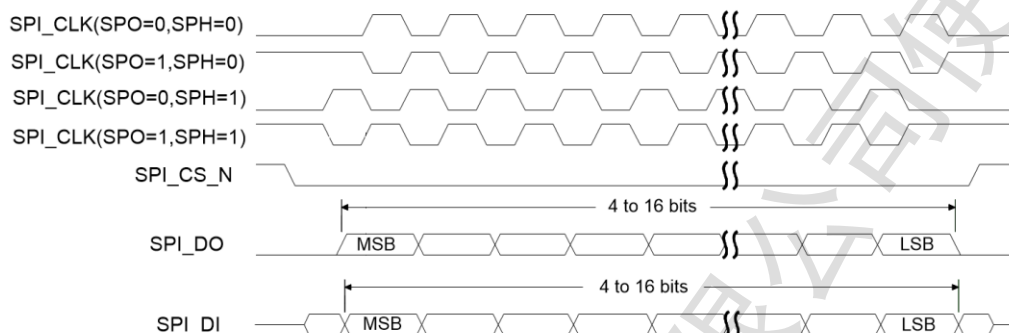
- SPI\_CLK 信号设置为高
- SPI\_CS\_N 信号设置为高
- 发送数据线 SPI\_DO 强制为低

当 SPI 为使能状态，而且发送 FIFO 内有有效数据时，设置 SPI\_CS\_N Master 信号为低表示开始传输数据。半个 SPI\_CLK 时钟周期后，Master 和 Slave 数据在各自的传输线上有效。同时，时钟 SPI\_CLK 从 1 个下降沿开始有效。数据在 SPI\_CLK 时钟的上升沿被捕获，在时钟的下降沿被传送。当传输单个 word 时，SPI\_CS\_N 在传输的最后 1bit 捕获之后的 1 个 SPI\_CLK 时钟周期之后恢复为高电平。

如果是连续传输，SPI\_CS\_N 信号始终保持为低。SPI\_CS\_N 在捕获到最后 1bit 之后的 1 个 SPI\_CLK 时钟周期之后恢复到高状态。对于连续传输来说，SPI\_CS\_N 在传输过程中一直保持为低，结束方式与单个传输方式相同。

##### (5) 接口时序

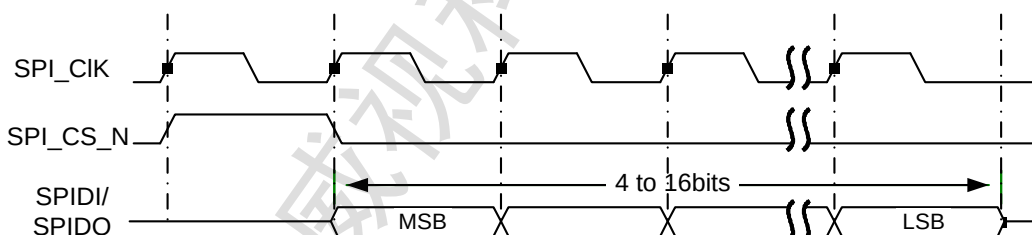
图2-18 SPI 接口时序图



## TI 同步串行接口

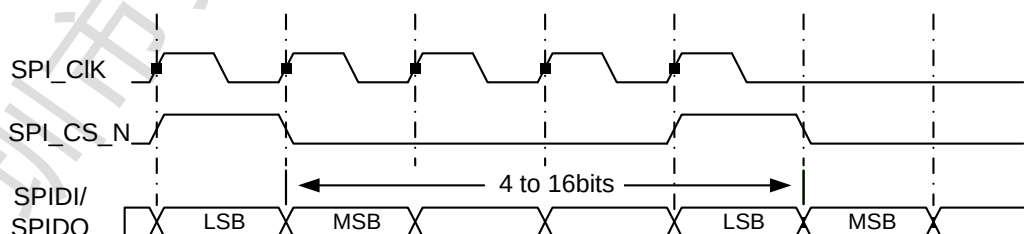
TI 同步串行单帧格式如图 2-19 所示。

图2-19 TI 同步串行单帧格式



TI 同步串行连续帧格式如图 2-20 所示。

图2-20 TI 同步串行连续帧格式



在该模式下，当 SPI 处于空闲状态时：

- SPI\_CK 为低电平。
- SPI\_CS\_N 为低电平。

- 传输数据线 SPIDO 保持为高阻。

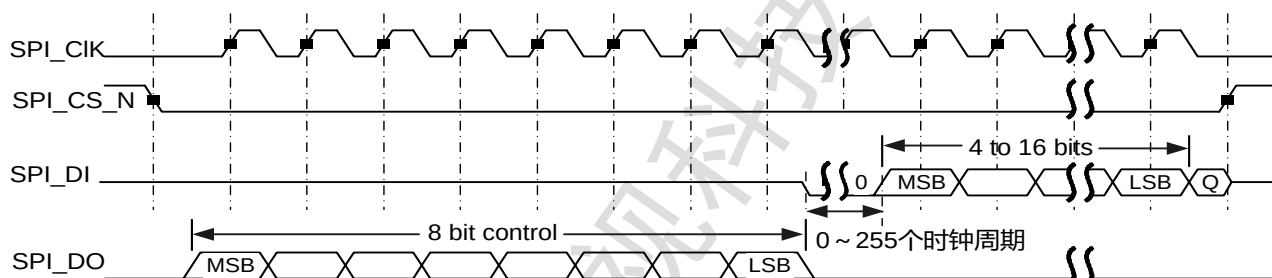
一旦发送 FIFO 有数据，SPICSN 就会产生一个 SPICK 时钟周期的高电平脉冲，将被发送的数据就会从发送 FIFO 传送到发送逻辑串行移位寄存器。在 SPICK 时钟的下一个上升沿，4bit ~ 16bit 数据帧的 MSB 就会从 SPIDO 移位输出。同样，从外部串行 slave 设备接收数据的 MSB 会从 SPIDI 管脚移位输入。

SPI 和片外串行设备在 SPICK 时钟的下降沿将数据存入串行移位寄存器。接收串行寄存器在接收到 LSB 之后的第一个 SPICK 时钟上升沿将数据送给接收 FIFO。

## National Semiconductor Microwire 接口

National Semiconductor Microwire 单帧格式如图 2-21 所示。

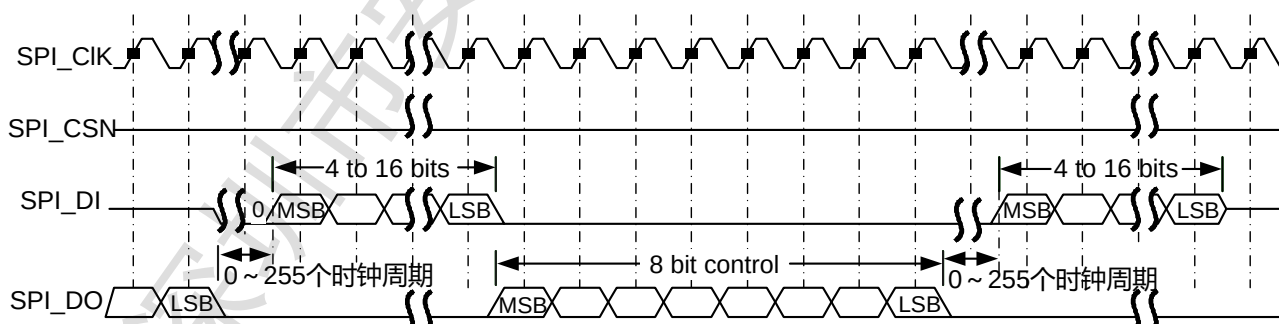
图2-21 National Semiconductor Microwire 单帧格式



在 SPIDO LSB 结束和 SPIDI MSB 开始之间可以延迟 0 ~ 255 个时钟周期。

National Semiconductor Microwire 连续帧格式如图 2-22 所示。

图2-22 National Semiconductor Microwire 连续帧格式



在 SPI\_DO LSB 结束和 SPI\_DI MSB 开始之间可以延迟 0 ~ 255 个时钟周期。

Microwire 的格式与 SPI 的格式非常相近，使用 master-slave 信息的传输技术，只不过 SPI 是全双工通信，而 Microwire 半双工通信。在 SPI 向外部芯片发送串行数据的



时候，都要先加 8bit 控制字。在这个过程中，SPI 没有接收到任何数据。传输完毕之后，片外芯片对接收到的数据进行解码，在与 8bit 控制信息间隔一个时钟周期之后，slave 开始响应所需求的数据。返回的数据长度为 4bit ~ 16bit，使得整个帧的长度为 13bit ~ 25bit。

在该模式下，当 SPI 处于空闲状态时：

- SPI\_CLK 信号设置为低电平。
- SPI\_CS\_N 设置为高电平。
- 发送数据线 SPI\_DO 强制为低电平。

向发送 FIFO 内部写进一个控制字节开始一次传送。SPI\_CS\_N 的下降沿引发数据的传输，发送 FIFO 的数据被发送到串行移位寄存器，8bit 控制帧的 MSB 被发送到发送管脚 SPIDO。在帧的传送过程中，SPI\_CS\_N 保持为低。SPI\_DI 在这个传送过程中保持为高阻。

片外的串行从设备在 SPI\_CLK 时钟的每一个上升沿将数据锁存到串行移位寄存器中。当从设备锁存完最后 1 个 bit 的数据之后，在接下来的 1 个时钟周期的等待时间里，对接收到的数据开始解码，然后从设备反馈给 SPI 所要求的数据。每 1 个 bit 都是在 SPICK 时钟的下降沿写到 SPI\_DI 的。对单个数据传送来说，在帧的结尾，SPI\_CS\_N 在最后 1 个 bit 写到接收串行寄存器之后的 1 个时钟周期后拉高，这样就使接收到的数据传送到接收 FIFO。

对于连续的传送来说，数据传送的开始和结束都和单个数据的传送方式相同。在这个传送过程中，信号 SPI\_CS\_N 时一直保持为低的，传送的数据也是连续的。下一帧的控制字直接和上一帧的 LSB 相邻。当帧的 LSB 锁存到 SPI 之后，接收到的每一个数值都是在 SPI\_CLK 时钟的下降沿取自接收移位寄存器。

## 2.3.5 工作方式

### 工作模式

SPI 的工作模式分为中断或查询方式下的数据传输和 DMA 方式下的数据传输。

### 时钟与复位

输出 SPI 时钟频率计算方式如下：

$$F_{sspcclkout} = F_{sspcclk} / (CPSDVSRx (1+SCR))$$

F<sub>sspcclk</sub>: SPI 的工作参考时钟 600M。

CPSDVS、SCR 请查询相应寄存器。

支持独立软复位：由寄存器 PERI\_CRG111 的 bit[16:15]控制。相应位写 “0”，SPI 退出软复位；相应位写 “1”，SPI 进入软复位。上电缺省值为 0。

支持时钟门控：由寄存器 PERI\_CRG111 的 bit[13:12]控制。相应位写 “0”，关闭时钟；相应位写 “1”，打开时钟。上电缺省值为 1。

## 中断处理

SPI 有 5 个中断，其中前 4 个是独立中断源、可屏蔽、高电平有效。

- SPIRXINTR  
接收 FIFO 中断请求。当接收 FIFO 中有 4 个或更多的有效数据时，该中断置位。
- SPITXINTR  
发送 FIFO 中断请求。当发送 FIFO 中有 4 个或更多的有效数据时，该中断置位。
- SPIRORINTR  
接收 overrun 中断请求。当 FIFO 已满，且又有新的数据需要写入 FIFO 时，会引起 FIFO overrun，该中断置位。此时数据被写入接收移位寄存器，而不是 FIFO。
- SPIRTINTR  
接收 time out 中断请求。当接收 FIFO 非空，且 SPI 处于 idle 态超过一个固定的 32bit 周期，该中断置位。  
此时表明接收 FIFO 中仍有数据需要传输。如果接收 FIFO 被读空或者当有新的数据被接收到 SPIRXD 中，该中断解除置位。也可以通过写寄存器 SPIICR[RTIC]清除该中断。
- SPIINTR  
组合中断，为以上 4 个中断经过“或”运算后的结果。如果上述 4 个独立中断中任意一个置位且使能，该中断置位。

SPI 的中断 SPIINTR 请参见中断处理章节。

## 初始化

初始化步骤如下：

- 步骤 1 向寄存器 SPICR1 [sse]写 “0”，禁止 SPI。
- 步骤 2 写寄存器 SPICR0，配置帧格式及传输数据位宽等参数。
- 步骤 3 配置寄存器 SPICPSR，设定时钟分频因子。

步骤 4 中断方式下，设置寄存器 SPIIMSC，使能相应中断信号；查询、DMA 方式下，应禁止产生相应中断信号。

步骤 5 中断或 DMA 方式，设置 SPITXFIFOCR 和 SPIRXFIFOCR。

步骤 6 DMA 方式下，配置寄存器 SPIDMACR，使能 SPI 的 DMA 功能。

----结束

## 查询方式下的数据传输

具体步骤如下：

步骤 1 向寄存器 SPICR1 [sse]写 “1” ，使能 SPI。

步骤 2 将需发送的数据连续写到寄存器 SPIDR。

步骤 3 轮询寄存器 SPISR，直到[BSY]=0（表示总线不忙）、[TFE]=1（表示发送 FIFO 已空）、[RNE]=1（表示接收 FIFO 非空）。

步骤 4 读出数据，需保证读空接收 FIFO（可通过查询 SPISR [RNE]得到）。

---

### 须知

SPI/Microwire 的全双工特性，每发一个数据就接收一个数据，即使只需发送数据，也需要清空接收 FIFO。

---

步骤 5 向寄存器 SPICR1 [sse]写 “0” ，禁止 SPI。

----结束

## 中断方式下的数据传输

具体步骤如下：

步骤 1 向寄存器 SPICR1 [sse]写 “1” ，使能 SPI。

步骤 2 将需发送的数据连续写到寄存器 SPIDR。

步骤 3 等待中断 SPIRXINTR，读出数据。循环直到读出所有数据。

## 须知

SPI/Microwire 的全双工特性，每发一个数据就接收一个数据，即使只需发送数据，也需要清空接收 FIFO。

步骤 4 向寄存器 SPICR1 [sse]写 “0” ，禁止 SPI。

----结束

## DMA 方式下的数据传输

具体步骤如下：

步骤 1 获取一个 DMAC 通道。

步骤 2 向寄存器 SPICR1 [sse]写 “1” ，使能 SPI。

步骤 3 发送数据。

1. 配置该 DMAC 通道的配置寄存器和控制寄存器中的相关参数。
2. 启动 DMAC，响应 SPI 发送 FIFO 的 DMA 请求进行数据传输。
3. 通过 DMA 中断上报，判断数据是否发送完成，如果完成则关闭 SPI 的 DMA 功能。

步骤 4 接收数据

1. 配置该 DMAC 通道的配置寄存器和控制寄存器中的相关参数。
2. 启动 DMAC，响应 SPI 接收 FIFO 的 DMA 请求进行数据传输。
3. 通过 DMA 中断上报，判断数据是否接收完成，如果完成则关闭 SPI 的 DMA 功能。

步骤 5 向寄存器 SPICR1 [sse]写 “0” ，禁止 SPI。

----结束

## SPI 从机下的数据接收

举例，对接 sensor 时，SPI2 接口频率 50MHz，做从机接收数据 640\*320 图像帧数据时的，具体步骤如下：

步骤 1 配置 SPI2 时钟门控与 DMA 门控。

1. 0x120101bc + 0x4000 #ssp2\_cken。

2. 0x12010194 + 0x4 #dma\_cken。

步骤 2 配置 IO 复用。

1. 0x100C\_0038 + 0x1004 #spi2\_sclk
2. 0x100C\_003c + 0x1004 #spi2\_sdi

步骤 3 配置 SPI2 从机片选信号选择，将片选信号拉低

0x12028000 + 0x40 #s\_spi2\_cs\_pctrl

步骤 4 配置 SPI2 从机。

1. 0x12072010 + 0x0c #clk div
2. 0x12072000 + 0x87 #data width: 8bit, SPH=1, SPO=0
3. 0x12072004 + 0x04 #MSB->LSB; 从机模式
4. 0x12072014 + 0x0c #中断屏蔽
5. 0x12072004 + 0x6 #使能 spi
6. 0x1207202c + 0x3 #接收 fifo 水线
7. 0x12072024 + 0x1 #DMA 接收 fifo 使能寄存器使能

步骤 5 配置 DMA。

1. 0x100b0108 + 0x12072008 #dma 源地址 spi
2. 0x100b010c + 0x40000000 #dma 目标地址 ddr
3. 0x100b0114 + 0x00032000 #dma 搬运次数('d640 \* 'd320)
4. 0x100b0104 + 0x102000b0 #dma 中断源
5. 0x100b0100 + 0x80218100 #dma 搬运数据位宽

步骤 6 等待 DMA 搬运数据结束。

0x100b0004 + 0x00000001 # 读 edma 寄存器直到中断拉高，传输完成

步骤 7 清除 DMA 中断。

0x100b0008 + 0x0000000f #清除中断

----结束

## 2.3.6 寄存器概览

SPI 寄存器概览如表 2-10 所示。

- SPI0 基地址是 0x1207\_0000

- SPI1 基地址是 0x1207\_1000
- SPI2 基地址是 0x1207\_2000

表2-10 SPI 寄存器概览

| 偏移地址  | 名称          | 描述            |
|-------|-------------|---------------|
| 0x000 | SPICR0      | 控制寄存器 0       |
| 0x004 | SPICR1      | 控制寄存器 1       |
| 0x008 | SPIDR       | 数据寄存器         |
| 0x00C | SPISR       | 状态寄存器         |
| 0x010 | SPICPSR     | 时钟分频寄存器       |
| 0x014 | SPIIMSC     | 中断屏蔽寄存器       |
| 0x018 | SPIRIS      | 原始中断状态寄存器     |
| 0x01C | SPIMIS      | 屏蔽后中断状态寄存器    |
| 0x020 | SPIICR      | 中断清除寄存器       |
| 0x024 | SPIDMACR    | DMA 控制寄存器     |
| 0x028 | SPITXFIFOCR | 发送 FIFO 控制寄存器 |
| 0x02C | SPIRXFIFOCR | 接收 FIFO 控制寄存器 |

2.3.7 寄存器描述

SPICR0

SPICR0 为控制寄存器 0。

Offset Address: 0x000    Total Reset Value: 0x0000

| Bits   | Access | Name | Description                                                                      | Reset |
|--------|--------|------|----------------------------------------------------------------------------------|-------|
| [15:8] | RW     | SCR  | 串行时钟率，取值范围 0 ~ 255。SCR 的值用来产生 SPI 发送和接收的比特率，公式为 $F_{SSPCLK}/(CPSDVSR*(1+SCR))$ 。 | 0x00  |

|       |    |     |                                                                                                                                                                                                                              |     |
|-------|----|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
|       |    |     | CPSDVSR 是一个 2 ~ 254 之间的偶数，由寄存器 SPICPSR 配置。                                                                                                                                                                                   |     |
| [7]   | RW | SPH | SPICLKOUT 相位，具体含义请参见“2.3.4 三种外设总线时序”的 SPI 帧格式。                                                                                                                                                                               | 0x0 |
| [6]   | RW | SPO | SPICLKOUT 极性，具体含义请参见“2.3.4 三种外设总线时序”的 SPI 帧格式。                                                                                                                                                                               | 0x0 |
| [5:4] | RW | FRF | 帧格式选择。<br>00: Motorola SPI 帧格式;<br>01: TI 同步串行帧格式;<br>10: National Microwire 帧格式;<br>11: 保留。                                                                                                                                 | 0x0 |
| [3:0] | RW | DSS | 设置数据位宽。<br>0011: 4bit;<br>0100: 5bit;<br>0101: 6bit;<br>0110: 7bit;<br>0111: 8bit;<br>1000: 9bit;<br>1001: 10bit;<br>1010: 11bit;<br>1011: 12bit;<br>1100: 13bit;<br>1101: 14bit;<br>1110: 15bit;<br>1111: 16bit;<br>其他: 保留。 | 0x0 |

## SPICR1

SPICR1 为控制寄存器 1。

Offset Address: 0x004    Total Reset Value: 0x7F00

| Bits   | Access | Name          | Description                                                                            | Reset |
|--------|--------|---------------|----------------------------------------------------------------------------------------|-------|
| [15]   | RW     | WaitEn        | 等待使能，当 SPICR0 寄存器的 FRF 配置为 National Microwire 帧格式时有效。<br>0：不使能；<br>1：使能。               | 0x0   |
| [14:8] | RW     | WaitVal       | National Microwire 帧格式时，写和读之间的等待拍数。当 WaitEn 为 1 并且帧格式为 National Microwire 时有效。         | 0x7F  |
| [7]    | RO     | reserved      | 保留。                                                                                    | 0x0   |
| [6]    | RW     | mode_altasens | 0：片选信号由芯片逻辑根据所选时序自动产生<br>1：当采用 Motorola SPI 帧格式时，片选 CS 信号由 SPI 使能信号控制，使能后，片选拉低，否则片选拉高。 | 0x0   |
| [5]    | RO     | reserved      | 保留。                                                                                    | 0x0   |
| [4]    | RW     | BitEnd        | 数据正反序控制位。<br>0：数据顺序为 MSB 到 LSB；<br>1：数据顺序为 LSB 到 MSB。                                  | 0x0   |
| [3]    | RO     | reserved      | 保留。                                                                                    | 0x0   |
| [2]    | RW     | MS            | 设置 Master 或者 Slave 模式，此位只能在 SPI 被禁止时改变。<br>0：Master 模式（默认）；<br>1：从机模式。                 | 0x0   |
| [1]    | RW     | SSE           | 设置 SPI 使能。<br>0：不使能；<br>1：使能。                                                          | 0x0   |



| Bits | Access | Name | Description                                                       | Reset |
|------|--------|------|-------------------------------------------------------------------|-------|
| [0]  | RW     | LBM  | 设置环回模式。<br>0: 正常的串行接口操作使能;<br>1: 发送串行移位寄存器的输出在内部连接到接收串行移位寄存器的输入上。 | 0x0   |

## SPIDR

SPIDR 为数据寄存器。

Offset Address: 0x008    Total Reset Value: 0x0000

| Bits   | Access | Name | Description                                                                                       | Reset  |
|--------|--------|------|---------------------------------------------------------------------------------------------------|--------|
| [15:0] | RW     | DATA | 发送/接收 FIFO。<br>读: 接收 FIFO;<br>写: 发送 FIFO。<br>如果数据比特数少于 16 则必须右对齐。发送逻辑将忽略高位未使用的比特位, 接收逻辑则自动将数据右对齐。 | 0x0000 |

## SPISR

SPISR 为状态寄存器。

Offset Address: 0x00C    Total Reset Value: 0x0003

| Bits   | Access | Name     | Description                       | Reset |
|--------|--------|----------|-----------------------------------|-------|
| [15:5] | RO     | reserved | 保留。                               | 0x000 |
| [4]    | RW     | BSY      | SPI 忙标记。<br>0: 空闲;<br>1: 忙。       | 0x0   |
| [3]    | RW     | RFF      | 接收 FIFO 是否已满。<br>0: 未滿;<br>1: 已滿。 | 0x0   |

| Bits | Access | Name | Description                       | Reset |
|------|--------|------|-----------------------------------|-------|
| [2]  | RW     | RNE  | 接收 FIFO 是否未空。<br>0: 已空;<br>1: 未空。 | 0x0   |
| [1]  | RW     | TNF  | 发送 FIFO 是否未空。<br>0: 已空;<br>1: 未空。 | 0x1   |
| [0]  | RW     | TFE  | 发送 FIFO 是否已空。<br>0: 未空;<br>1: 已空。 | 0x1   |

## SPIPCPSR

SPIPCPSR 为时钟分频寄存器。

Offset Address: 0x010 Total Reset Value: 0x0000

| Bits   | Access | Name     | Description                                             | Reset |
|--------|--------|----------|---------------------------------------------------------|-------|
| [15:8] | RO     | reserved | 保留。                                                     | 0x00  |
| [7:0]  | RW     | CPDVDVSR | 时钟分频因子。此值必须是 2 ~ 254 之间的偶数，取决于输入时钟 SPICLK 的频率。最低位读作“0”。 | 0x00  |

## SPIIMSC

SPIIMSC 为中断屏蔽寄存器。

Offset Address: 0x014 Total Reset Value: 0x0000

| Bits   | Access | Name     | Description                        | Reset |
|--------|--------|----------|------------------------------------|-------|
| [15:4] | RO     | reserved | 保留。                                | 0x000 |
| [3]    | RW     | TXIM     | 发送 FIFO 中断屏蔽。<br>0: 半空或更少情况下中断被屏蔽; | 0x0   |

| Bits | Access | Name  | Description                                                                                       | Reset |
|------|--------|-------|---------------------------------------------------------------------------------------------------|-------|
|      |        |       | 1: 半空或更少情况下中断未被屏蔽。                                                                                |       |
| [2]  | RW     | RXIM  | 接收 FIFO 中断屏蔽。<br>0: 半空或更少情况下中断被屏蔽;<br>1: 半空或更少情况下中断未被屏蔽。                                          | 0x0   |
| [1]  | RW     | RTIM  | 接收超时中断。<br>0: 接收超时中断屏蔽;<br>1: 接收超时中断不屏蔽。                                                          | 0x0   |
| [0]  | RW     | RORIM | 接收溢出中断屏蔽。<br>0: 接收 FIFO 溢出中断屏蔽;<br>1: 接收 FIFO 溢出中断不屏蔽。<br>值为“1”时使能硬件流控功能, 即接收 FIFO 满后 SPI 停止发送数据。 | 0x0   |

## SPIRIS

SPIRIS 为原始中断状态寄存器。值“0”表示无中断, 值“1”表示有中断。

Offset Address: 0x018    Total Reset Value: 0x0008

| Bits   | Access | Name     | Description        | Reset |
|--------|--------|----------|--------------------|-------|
| [15:4] | RO     | reserved | 保留。                | 0x000 |
| [3]    | RO     | TXRIS    | 发送 FIFO 中断的原始中断状态。 | 0x1   |
| [2]    | RO     | RXRIS    | 接收 FIFO 中断的原始中断状态。 | 0x0   |
| [1]    | RO     | RTRIS    | 接收超时中断的原始中断状态。     | 0x0   |
| [0]    | RO     | RORRIS   | 接收溢出中断的原始中断状态。     | 0x0   |

## SPIMIS

SPIMIS 为屏蔽后中断状态寄存器。值“0”表示无中断, 值“1”表示有中断。

Offset Address: 0x01C    Total Reset Value: 0x0000

| Bits   | Access | Name     | Description       | Reset |
|--------|--------|----------|-------------------|-------|
| [15:4] | RO     | reserved | 保留。               | 0x000 |
| [3]    | RO     | TXMIS    | 发送 FIFO 中断屏蔽后的状态。 | 0x0   |
| [2]    | RO     | RXMIS    | 接收 FIFO 中断屏蔽后的状态。 | 0x0   |
| [1]    | RO     | RTMIS    | 接收超时中断屏蔽后的状态。     | 0x0   |
| [0]    | RO     | RORMIS   | 接收溢出中断屏蔽后的状态。     | 0x0   |

## SPIICR

SPIICR 为中断清除寄存器。写“1”清除中断，写“0”无影响。

Offset Address: 0x020    Total Reset Value: 0x0000

| Bits   | Access | Name     | Description | Reset  |
|--------|--------|----------|-------------|--------|
| [15:2] | RO     | reserved | 保留。         | 0x0000 |
| [1]    | RO     | RTIC     | 清除接收超时中断。   | 0x0    |
| [0]    | RO     | RORIC    | 清除接收溢出中断。   | 0x0    |

## SPIDMACR

SPIDMACR 为 DMA 控制寄存器。

Offset Address: 0x024    Total Reset Value: 0x0000

| Bits   | Access | Name     | Description                            | Reset  |
|--------|--------|----------|----------------------------------------|--------|
| [15:2] | RO     | reserved | 保留位。                                   | 0x0000 |
| [1]    | WO     | TXDMAE   | DMA 发送 FIFO 使能寄存器。<br>0: 禁止;<br>1: 使能。 | 0x0    |
| [0]    | WO     | RXDMAE   | DMA 接收 FIFO 使能寄存器。<br>0: 禁止;<br>1: 使能。 | 0x0    |

## SPITXFIFO CR

SPITXFIFO CR 为发送 FIFO 控制寄存器。

Offset Address: 0x028    Total Reset Value: 0x0009

| Bits   | Access | Name        | Description                                                                                                                                                          | Reset |
|--------|--------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [15:6] | RO     | reserved    | 保留。                                                                                                                                                                  | 0x000 |
| [5:3]  | RW     | TXINTSize   | 配置发送 FIFO 请求中断的水线。即，发送 FIFO 中数据数目小于或等于 TXINTSize 所配置的字数时，TXRIS 有效。<br><br>000: 1;<br>001: 4;<br>010: 8;<br>011: 16;<br>100: 32;                                      | 0x1   |
| [2:0]  | RW     | DMATXBRSIZE | 配置发送 FIFO 请求 DMA 进行 burst 传输的水线。即，发送 FIFO 中数据数目小于或等于 (256-DMATXBRSIZE) 所配置的字数时，DMATXBREQ 有效，此处字长是 16 位。<br><br>000: 1;<br>001: 4;<br>010: 8;<br>011: 16;<br>100: 32; | 0x1   |

## SPIRXFIFO CR

SPIRXFIFO CR 为接收 FIFO 控制寄存器。

Offset Address: 0x02C    Total Reset Value: 0x0009

| Bits   | Access | Name       | Description                                                                                                                                        | Reset |
|--------|--------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [15:6] | RO     | reserved   | 保留。                                                                                                                                                | 0x000 |
| [5:3]  | RW     | RXINTSize  | 配置接收 FIFO 请求中断的水线。即，接收 FIFO 中数据数目大于或等于 (256-RXINTSize) 所配置的字数时，RXRIS 有效，此处字长是 16 位。<br><br>000: 1;<br>001: 4;<br>010: 8;<br>011: 16;<br>100: 32;   | 0x1   |
| [2:0]  | RW     | DMARXBSize | 配置接收 FIFO 请求 DMA 进行 burst 传输的水线。即，接收 FIFO 中数据数目大于或等于 DMARXBSize 所配置的字数时，DMARXBREQ 有效。<br><br>000: 1;<br>001: 4;<br>010: 8;<br>011: 16;<br>100: 32; | 0x1   |

## 2.4 eMMC/SD/SDIO 控制器

### 2.4.1 功能描述

#### 功能框图

eMMC/SD/SDIO 控制器 (以下简称 MMC) 用于处理对 SD 卡、eMMC 器件的读写等操作，并通过 SDIO(secure digital input/output) 协议实现对扩展外设 (如蓝牙、WiFi 等) 的支持。芯片提供 3 个 MMC 控制器。

芯片中 3 个 MMC 控制器对应的功能信号和管脚名，如下表 2-11 所示。

表2-11 芯片中 3 个 MMC 控制器对应的功能信号和管脚名

| MMC 控制器 | 功能信号                                | 管脚名                 |
|---------|-------------------------------------|---------------------|
| eMMC    | SFC_CLK                             | EMMC_CLK            |
|         | SFC_MOST_IO0                        | EMMC_CMD            |
|         | SFC_MISO_IO1                        | EMMC_DATA0          |
|         | SFC_CSN                             | EMMC_DATA1          |
|         | SFC_HOLD_IO3                        | EMMC_DATA2          |
|         | SFC_WP_IO2                          | EMMC_DATA3          |
|         | SDIO0_CARD_POWER_EN                 | EMMC_RST_N          |
|         | SDIO0_CDATA1                        | EMMC_DATA4          |
|         | SDIO0_CDATA0                        | EMMC_DATA5          |
|         | SDIO0_CMD                           | EMMC_DATA6          |
|         | SDIO0_CDATA3                        | EMMC_DATA7          |
| SDIO0   | SDIO0_CLK                           | SDIO0_CCLK_OUT      |
|         | SDIO0_CMD                           | SDIO0_CCMD          |
|         | SDIO0_DATA0                         | SDIO0_CDATA0        |
|         | SDIO0_CDATA1                        | SDIO0_CDATA1        |
|         | SDIO0_CDATA2                        | SDIO0_CDATA2        |
|         | SDIO0_CDATA3                        | SDIO0_CDATA3        |
|         | SDIO0_CARD_POWER_EN/LSAD C_CH2/PWM9 | SDIO0_CARD_POWER_EN |
| SDIO1   | SDIO1_CCLK_OUT/UART1_RTSN           | SDIO1_CCLK_OUT      |
|         | SDIO1_CCMD/UART1_CTSN               | SDIO1_CCMD          |
|         | SDIO1_CDATA0/UART1_RXD              | SDIO1_CDATA0        |
|         | SDIO1_CDATA1/UART1_TXD              | SDIO1_CDATA1        |
|         | SDIO1_CDATA2/JTAG_TMS               | SDIO1_CDATA2        |
|         | SDIO1_CDATA3/LCD_VSYNC              | SDIO1_CDATA3        |

- eMMC/SDIO0 支持符合以下协议的设备：  
MultiMediaCard (eMMC-version 4.5)

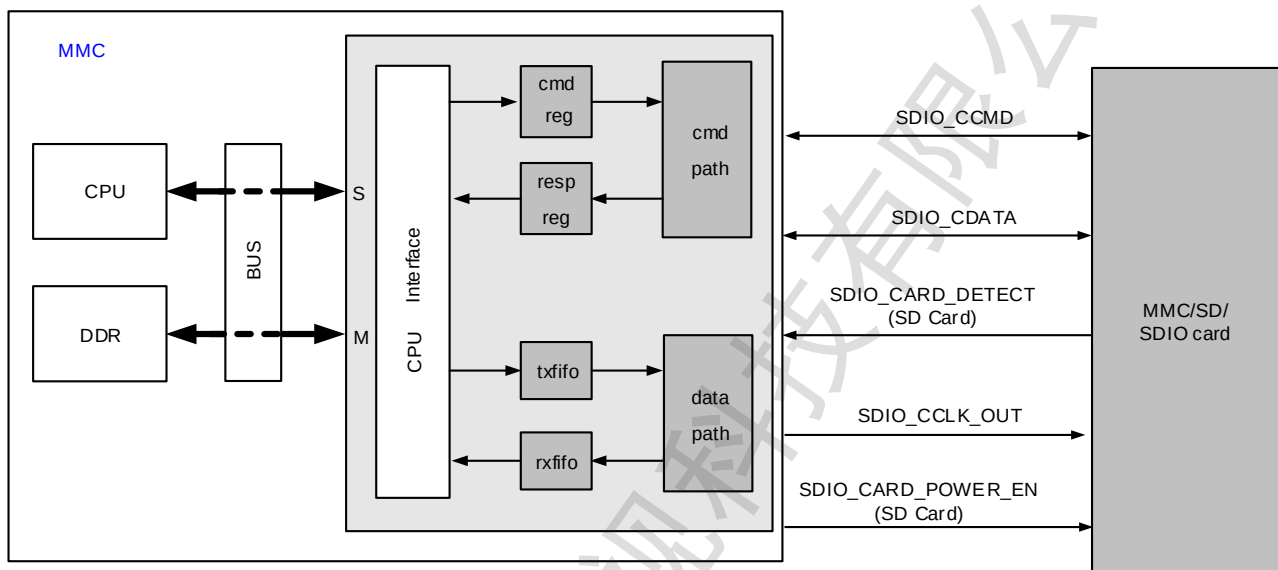
Secure Digital Memory (SD mem-version 2.0)

- SDIO1 支持符合以下协议的设备：

Secure Digital I/O (SDIO version2.0)

MMC 的功能框图如图 2-23 所示。

图2-23 MMC 功能框图



注：S：slave 接口；M：master 接口。

MMC 通过内部总线与系统连接，由以下单元构成：

- 命令通道  
完成指令的发送与响应的接收。
- 数据通道  
配合命令通道完成数据读写操作。
- 接口时钟控制单元  
控制接口时钟的关闭与开启,以及接口方向时钟相位调节。

MMC 的功能特点有：

- 最大可支持 2TB 容量的 SD 卡和 eMMC 器件。
- 支持 non-DMA、SDMA、ADMA2、ADMA3 的数据传输。
- 支持命令与数据的 CRC 生成与校验。

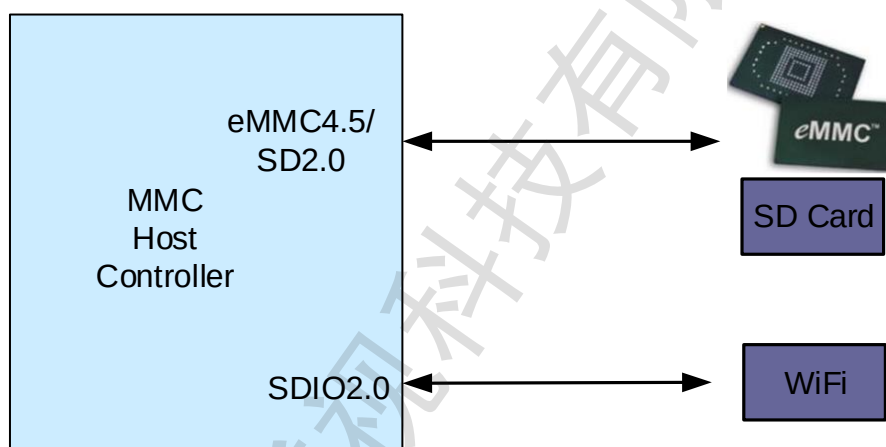


- 数据位宽支持 1bit、4bit，可根据对接器件选择。
- 支持 1byte ~ 65535byte 的块数据读写操作。
- SD 支持 Default Speed(DS)/High Speed(HS)模式。
- eMMC 支持 DS/HS/HS200 模式，支持 boot 功能。

## 典型应用

MMC 的典型应用如图 2-24 所示。

图2-24 MMC 典型应用图



## 指令与响应

MMC 与卡设备之间的所有交互操作均通过指令完成，包括卡初始化、寄存器读写、状态查询、数据传输等。

MMC 指令为 48bit 的串行数据，由起始位、传输位、指令序号、指令参数、CRC 校验位和终止位组成。卡收到指令后，会根据指令类型返回 48bit 或 136bit 的响应。

图2-25 MMC 指令格式

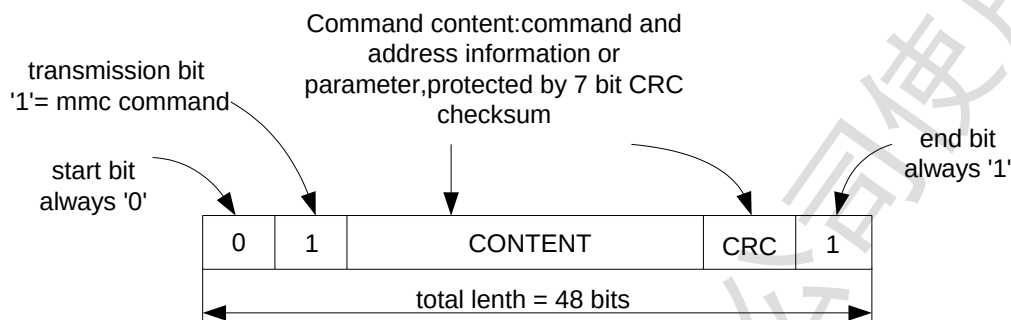
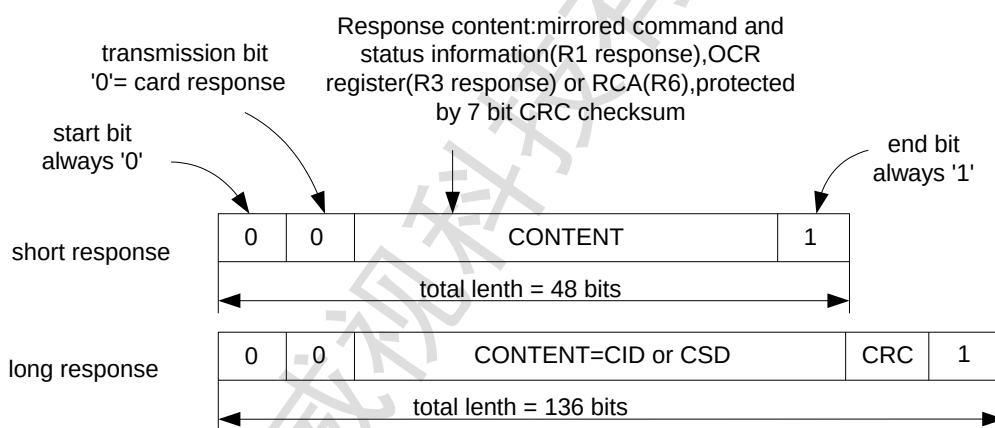


图2-26 MMC 指令响应格式



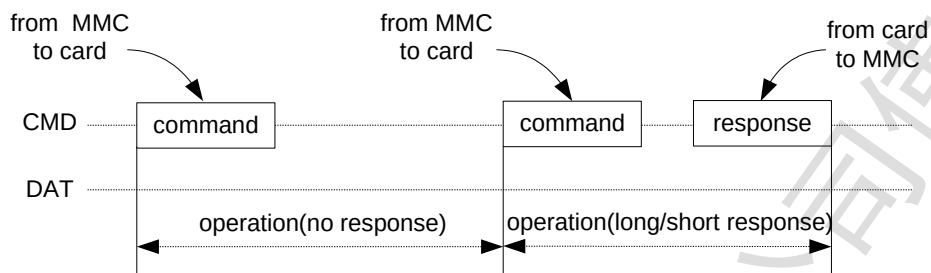
根据是否有数据传输，指令分为以下 2 种：

- **非数据传输指令**  
基于指令信号线 CMD，MMC 与卡采用串行方式进行指令发送与响应接收。
- **数据传输指令**  
除指令线上的交互外，还有数据线 DAT0 ~ DAT3 上的数据传输。

#### (1) 非数据传输指令

MMC 与卡设备之间的非数据传输指令操作如图 2-27 所示。

图2-27 MMC 非数据指令操作



## (2) 数据传输指令

MMC 卡支持以下数据传输指令：

- 流数据读写指令

仅 MMC 卡支持，只使用 1 根数据线（即 DAT0）进行数据传输，无 CRC 校验。

- 单块数据读写指令

一次传输完成一个块大小的数据，不需要使用停止命令结束一次数据传输。

- 多块数据读写指令

- predefined block count 方式

在多块读写指令前，发送块数量指令指定待传输的数据量。

- open ended 方式

发送读写指令后，在数据传输末尾，需使用停止指令来结束一次数据传输。

SDIO 设备的多块读写指令，不同于上述 2 种方式，在发送读写指令时，指令参数中包含待传输的数据量。

根据响应的类型，指令分为以下 3 种：

- 无响应指令

如卡复位指令。

- 短响应指令

数据传输指令、卡状态查询等均属于这类指令。

- 长响应指令

仅用于读取卡的寄存器 CID 和 CSD 信息。

# 数据传输

单块读写和多块读写为较常用的数据传输方式。通常 SD 卡、eMMC 器件数据传输的一个块大小为 512byte，而 SDIO 设备可根据应用自定义。

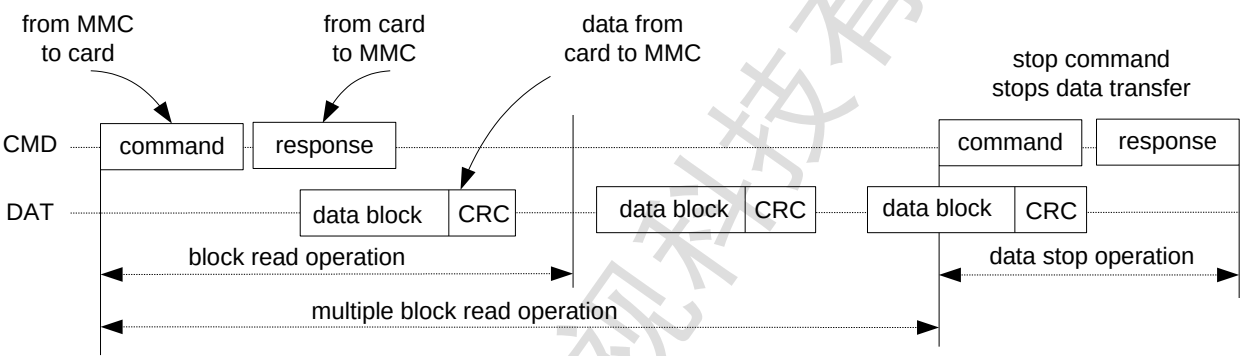
 说明

以块读写指令方式进行数据传输时，传输数据总量必须为块大小的整数倍。

数据传输指令均为短响应指令，并伴随着数据线上的数据传输。指令、响应及数据线上的时序配合关系如图 2-28 和图 2-29 所示。

(1) 单块与多块读操作

图2-28 单块与多块读操作

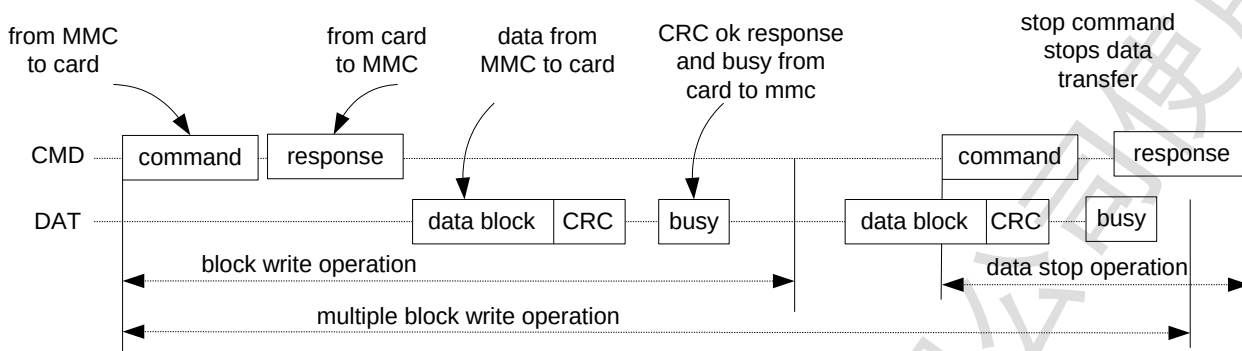


MMC 向卡发送单块或多块读指令。在接收响应的过程中，接收以块为单位的数据，其中每块数据中均包含有 CRC 校验位，以保证数据传输的完整性。

单块读指令操作时，MMC 在接收一块数据后完成一次数据传输；多块读指令操作时，MMC 在接收多块数据后，需发送一条停止指令结束本次数据传输（仅 open ended 多块读指令）。

(2) 单块与多块写操作

图2-29 单块与多块写操作



MMC 往卡发送单块或多块写指令。在接收到响应后，往卡发送以块为单位的数据，其中每块数据中均包含有 CRC 校验位，卡会对每块数据进行 CRC 校验，并反馈 CRC 状态以确认数据传输的正确性。

单块写指令操作时，MMC 在发送一块数据后完成一次数据传输；多块写指令操作时，MMC 在发送多块数据后，需发送一条停止指令完成本次数据传输（仅 open ended 多块读指令）。写操作结束后，卡可能会因为编程 Flash 而处于繁忙状态，MMC 需查询 DAT0 状态，以确认卡脱离繁忙状态后才能对卡进行下一步操作。

### (3) 数据传输格式

块方式读写中，MMC 与卡之间可采用 1bit 或 4bit 数据线方式进行数据传输。在发送数据传输指令之前，应分别设置 MMC 与卡的数据传输位宽模式（1bit 或 4bit），使它们保持一致。MMC 的数据位宽通过寄存器 HOST\_CTRL1\_R 设置，卡的数据位宽则通过发送相应的指令进行设置。

1bit 和 4bit 模式下的数据传输格式如图 2-30 和图 2-31 所示。

图2-30 1bit 数据线传输模式下的块数据格式

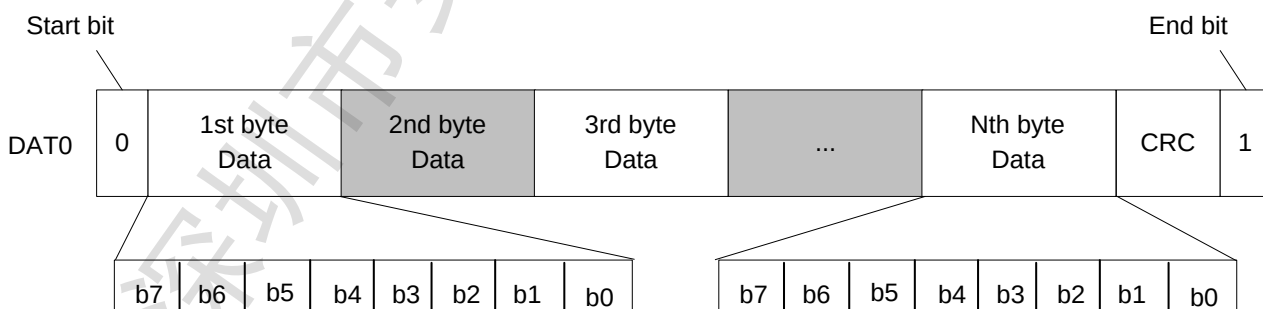
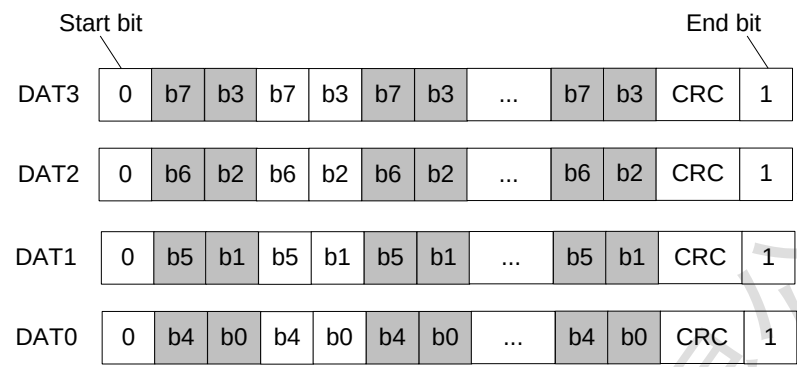


图2-31 4bit 数据线传输模式下的块数据格式



SD2.0 支持的速度模式

控制器支持 SD2.0, 在 SD-mode 模式下各传输模式如表 2-12 所示。

表2-12 传输模式表

| 模式 | 支持的输入时钟（MHz） | 卡侧时钟  | 最大数据位宽 | 电压   |
|----|--------------|-------|--------|------|
| HS | 50MHz        | 50MHz | 4bit   | 3.3V |
| DS | 25MHz        | 25MHz | 4bit   | 3.3V |

eMMC 支持的速度模式及电压

控制器在 eMMC-mode 模式下支的各传输模式如表 2-13 所示。

表2-13 传输模式表

| 模式     | 支持的输入时钟（MHz） | 卡侧时钟   | 最大数据位宽 | 电压        |
|--------|--------------|--------|--------|-----------|
| DS     | 26MHZ        | 26MHZ  | 8bit   | 1.8V/3.3V |
| HS_SDR | 52MHZ        | 52MHZ  | 8bit   | 1.8V/3.3V |
| HS_DDR | 52MHZ        | 52MHZ  | 8bit   | 1.8V/3.3V |
| HS200  | 100MHZ       | 100MHZ | 8bit   | 1.8V      |

## 2.4.2 应用说明



说明

时钟复位寄存器请参考“CRG 寄存器描述”。

### 软复位

在数据传输出现异常而导致 MMC 无法回到空闲状态时：

- CRG 寄存器 PERI\_CRG125 bit[27]写 “1” 对 eMMC 模块进行软复位；
- CRG 寄存器 PERI\_CRG139 bit[27]写 “1” 对 SDIO0 模块进行软复位；
- CRG 寄存器 PERI\_CRG144 bit[27]写 “1” 对 SDIO1 模块进行软复位；

**建议在使用 MMC 之前和热插拔卡后，软复位 MMC。**

### 接口时钟配置

遵从不同协议版本的 MMC 卡，以及 MMC 卡处于不同的状态时，均使用不同的时钟频率。

- eMMC 通过 CRG 寄存器 PERI\_CRG125 bit[26:24]配置；
- SDIO0 通过 CRG 寄存器 PERI\_CRG139 bit[26:24]配置；
- SDIO1 通过 CRG 寄存器 PERI\_CRG144 bit[26:24]配置

在改变 MMC 卡的时钟频率之前，必须保证没有数据或指令正在传输。为了避免输出到 MMC 卡的时钟产生毛刺，在改变 MMC 卡的时钟频率时应该遵照以下步骤：

步骤 1 关闭接口时钟。

将寄存器 CLK\_CTRL\_R bit[2]置 “0”。

步骤 2 设置接口时钟频率。

根据不同的 MMC 配置对应的 CRG 寄存器，选择接口时钟频率。

步骤 3 重新使能接口时钟。

将寄存器 CLK\_CTRL\_R bit[2]置 “1”。

----结束

### MMC 初始化

与卡进行命令和数据的交互前，需要初始化控制器，步骤如下：

- 步骤 1 清中断。将寄存器 NORMAL\_INT\_STAT\_R bit[15:0] 和 ERROR\_INT\_STAT\_R bit[15:0]所有位置 “1” ，清除原始中断状态位。
- 步骤 2 关闭中断信号使能。将寄存器 NORMAL\_INT\_SIGNAL\_EN\_R bit[15:0] 和 ERROR\_INT\_SIGNAL\_EN\_R bit[15:0]所有位置 “0” 。
- 步骤 3 设置超时计数。配置寄存器 TOUT\_CTRL\_R bit[3:0]。
- 步骤 4 如果为 eMMC，将寄存器 EMMC\_CTRL\_R bit[0] 置 “1” ，否则置 “0” 。
- 步骤 5 打开时钟。配置寄存器 CLK\_CTRL\_R bit[3]、[2]和[0] 为 “1” 。
- 步骤 6 开启 power\_on。将寄存器 PWR\_CTRL\_R bit[0] 置 “1” 。
- 步骤 7 打开中断状态使能。配置寄存器 NORMAL\_INT\_STAT\_EN\_R bit[15:0]和 ERROR\_INT\_STAT\_R bit[15:0]。
- 步骤 8 打开中断信号使能。配置寄存器 NORMAL\_INT\_SIGNAL\_EN\_R bit[15:0]和 ERROR\_INT\_SIGNAL\_EN\_R bit[15:0]。

----结束

完成以上步骤后，配置接口时钟以后，就可以往卡发送指令了。

## 非数据传输指令

发送非数据传输指令的步骤如下：

- 步骤 1 在寄存器 ARGUMENT\_R 中设置相应的指令内容。
- 步骤 2 在寄存器 XFER\_MODE\_R 和 CMD\_R 中设置相应指令参数。
- 步骤 3 等待指令被 MMC 执行。
- 步骤 4 检查寄存器 NORMAL\_INT\_STAT\_R bit[0]是否产生指令完成中断。
- 步骤 5 检查寄存器 ERROR\_INT\_STAT\_R bit[15:0]是否有异常响应中断，必要时可读取响应值。

----结束

## 单块或多块读数据

读取单块或多块数据的步骤如下：

- 步骤 1 清除中断寄存器 NORMAL\_INT\_STAT\_R 和 ERROR\_INT\_STAT\_R 中的中断。



步骤 2 向寄存器 BLOCKSIZE\_R 和 BLOCKCOUNT\_R bit[11:0]写入块的大小。

步骤 3 向寄存器 SDMASA\_R 写入块的个数。

步骤 4 向寄存器 ARGUMENT\_R 写入读取数据的起始地址。

步骤 5 配置寄存器 XFER\_MODE\_R 和 CMD\_R, 设置数据传输参数。

对于 SD 卡、eMMC 器件, 分别使用指令 CMD17/CMD18 进行单块/多块读操作; 对于 SDIO 卡, 使用指令 CMD53 进行单块/多块读操作。

一旦寄存器 CMD\_R 被写入, MMC 就执行指令; 当指令被送到总线上以后, 产生 cmd\_done 中断。

步骤 6 Non\_DMA 方式读取数据。检查寄存器 NORMAL\_INT\_STAT\_R bit[5], 如果为 1, 则从寄存器 BUF\_DATA\_R 读取 FIFO 中的数据, 以便 MMC 接收后面的数据; 同时检查数据错误中断, 即寄存器 ERROR\_INT\_STAT\_R bit[4]、bit[5]、bit[6]。此时, 程序可以发送一个停止指令中止数据的传输。

步骤 7 当寄存器 NORMAL\_INT\_STAT\_R bit[1]为 1 时, 数据传输完成, 从寄存器 BUF\_DATA\_R 中读取残留在 FIFO 中的数据。

----结束

## 单块与多块写数据

写入单块或多块数据的步骤如下:

步骤 1 清除中断寄存器 NORMAL\_INT\_STAT\_R 和 ERROR\_INT\_STAT\_R 中的中断。

步骤 2 向寄存器 BLOCKSIZE\_R 和 BLOCKCOUNT\_R bit[11:0]写入块的大小。

步骤 3 向寄存器 SDMASA\_R 写入块的个数。

步骤 4 向寄存器 ARGUMENT\_R 写入要写数据的起始地址。

步骤 5 配置寄存器 XFER\_MODE\_R 和 CMD\_R, 设置数据传输参数。

对于 SD 卡、eMMC 器件, 分别使用指令 CMD24/CMD25 进行单块/多块写操作; 对于 SDIO 卡, 使用指令 CMD53 进行单块/多块写操作。

步骤 6 Non\_DMA 方式写数据。将数据写入 FIFO, 即写寄存器 BUF\_DATA\_R。

步骤 7 检查寄存器 NORMAL\_INT\_STAT\_R bit[4], 如果为 1, 则向寄存器 BUF\_DATA\_R 写入要填充到 FIFO 中的数据, 以便 MMC 发送后面的数据; 同时检查数据错误中断, 即

寄存器 ERROR\_INT\_STAT\_R bit[4]、bit[5]、bit[6]。此时，程序可以发送一个停止指令中止数据的传输。

步骤 8 当寄存器 NORMAL\_INT\_STAT\_R bit[1]为 1 时，数据传输完成。

----结束

### 2.4.3 寄存器概览

MMC 寄存器概览如表 2-14 所示。

MMC 寄存器概览 (eMMC/SDIO0 基地址是 0x1001\_0000/0x1004\_0000; SDIO1 基地址是 0x1002\_0000)

表2-14 MMC 寄存器概览

| 偏移地址   | 名称           | 描述              |
|--------|--------------|-----------------|
| 0x0000 | SDMASA_R     | SDMA 系统地址寄存器    |
| 0x0004 | BLOCKSIZE_R  | BLOCK SIZE 寄存器  |
| 0x0006 | BLOCKCOUNT_R | BLOCK CNT 寄存器   |
| 0x0008 | ARGUMENT_R   | SD/eMMC 命令内容寄存器 |
| 0x000c | XFER_MODE_R  | Transfer 模式寄存器  |
| 0x000e | CMD_R        | 命令寄存器           |
| 0x0010 | RESP01_R     | RESP01 寄存器      |
| 0x0014 | RESP23_R     | RESP23 寄存器      |
| 0x0018 | RESP45_R     | RESP45 寄存器      |
| 0x001c | RESP67_R     | RESP67 寄存器      |
| 0x0020 | BUF_DATA_R   | Buffer 数据寄存器    |
| 0x0024 | PSTATE_R     | 当前状态寄存器         |
| 0x0028 | HOST_CTRL1_R | HOST 控制 1 寄存器   |
| 0x0029 | PWR_CTRL_R   | 上电控制寄存器         |

| 偏移地址   | 名称                     | 描述                                |
|--------|------------------------|-----------------------------------|
| 0x002a | WUP_CTRL_R             | Wakeup 控制寄存器                      |
| 0x002c | CLK_CTRL_R             | 时钟控制寄存器                           |
| 0x002e | TOUT_CTRL_R            | 超时控制寄存器                           |
| 0x002f | SW_RST_R               | 软复位控制寄存器                          |
| 0x0030 | NORMAL_INT_STAT_R      | 普通中断状态寄存器                         |
| 0x0032 | ERROR_INT_STAT_R       | 错误中断状态寄存器                         |
| 0x0034 | NORMAL_INT_STAT_EN_R   | 普通中断状态使能寄存器                       |
| 0x0036 | ERROR_INT_STAT_EN_R    | 错误中断状态使能寄存器                       |
| 0x0038 | NORMAL_INT_SIGNAL_EN_R | 普通中断信号使能寄存器                       |
| 0x003a | ERROR_INT_SIGNAL_EN_R  | 错误中断信号使能寄存器                       |
| 0x003c | AUTO_CMD_STAT_R        | Auto 命令状态寄存器                      |
| 0x003e | HOST_CTRL2_R           | HOST 控制 2 寄存器                     |
| 0x0040 | CAPABILITIES1_R        | 能力 1 寄存器                          |
| 0x0044 | CAPABILITIES2_R        | 能力 2 寄存器                          |
| 0x0048 | CURR_CAPBILITIES1_R    | 当前能力 1 寄存器                        |
| 0x0054 | ADMA_ERR_STAT_R        | ADMA 错误状态寄存器                      |
| 0x0058 | ADMA_SA_LOW_R          | ADMA 系统地址低 32bit 寄存器              |
| 0x005c | ADMA_SA_HIGH_R         | ADMA 系统地址高 32bit 寄存器              |
| 0x0078 | ADMA_ID_LOW_R          | ADMA Intergrated 描述子地址低 32bit 寄存器 |
| 0x007c | ADMA_ID_HIGH_R         | ADMA Intergrated 描述子地址高 32bit 寄存器 |
| 0x0500 | MSHC_VER_ID_R          | 控制器版本 ID 寄存器                      |

| 偏移地址   | 名称              | 描述                |
|--------|-----------------|-------------------|
| 0x0504 | MSHC_VER_TYPE_R | 控制器版本类型寄存器        |
| 0x0510 | MBIU_CTRL_R     | MBIU 控制寄存器        |
| 0x052c | EMMC_CTRL_R     | eMMC 控制寄存器        |
| 0x052e | BOOT_CTRL_R     | BOOT 控制寄存器        |
| 0x0534 | GP_OUT_R        | eMMC 复位和 GPIO 寄存器 |
| 0x0540 | AT_CTRL_R       | Tuning 控制寄存器      |
| 0x0544 | AT_STAT_R       | Tuning 状态寄存器      |
| 0x054c | MULTI_CYC_EN    | 多时钟延迟控制寄存器        |
| 0x0550 | MEM_CONFIG      | Mem 调速控制寄存器       |

## 2.4.4 寄存器描述

### SDMASA\_R

SDMASA\_R 为 SDMA 系统地址寄存器。

Offset Address: 0x0000    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name            | Description                                                                                             | Reset      |
|--------|--------|-----------------|---------------------------------------------------------------------------------------------------------|------------|
| [31:0] | RW     | blockcnt_sdmasa | Host Version Enable=0, 表示 SDMA 系统地址。<br><br>Host Version Enable=1, 表示 Block Count。<br><br>1 表示一个 block。 | 0x00000000 |

### BLOCKSIZE\_R

BLOCKSIZE\_R 为 BLOCK SIZE 寄存器。

Offset Address: 0x0004    Total Reset Value: 0x0000

| Bits    | Access | Name            | Description                                                                   | Reset |
|---------|--------|-----------------|-------------------------------------------------------------------------------|-------|
| [15]    | -      | reserved        | 保留。                                                                           | 0x0   |
| [14:12] | RW     | sdma_buf_bdary  | SDMA 数据块边界值。<br>000: 4K bytes;<br>001: 8K bytes;<br>.....<br>111: 512K bytes。 | 0x0   |
| [11:0]  | RW     | xfer_block_size | 传输 block size。<br>0x800: 2048bytes。                                           | 0x000 |

## BLOCKCOUNT\_R

BLOCKCOUNT\_R 为 BLOCK COUNT 寄存器。

Offset Address: 0x0006    Total Reset Value: 0x0000

| Bits   | Access | Name      | Description                                                  | Reset  |
|--------|--------|-----------|--------------------------------------------------------------|--------|
| [15:0] | RW     | block_cnt | Block 个数。<br>0x1: 1 block;<br>.....<br>0xFFFF: 65535 blocks。 | 0x0000 |

## ARGUMENT\_R

ARGUMENT\_R 为 SD/eMMC 命令内容寄存器。

Offset Address: 0x0008    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description  | Reset      |
|--------|--------|----------|--------------|------------|
| [31:0] | RW     | argument | 32bit 命令内容值。 | 0x00000000 |

## XFER\_MODE\_R

XFER\_MODE\_R 为 Transfer 模式命令寄存器。

Offset Address: 0x000c    Total Reset Value: 0x0000

| Bits   | Access | Name                | Description                                                                                                                        | Reset |
|--------|--------|---------------------|------------------------------------------------------------------------------------------------------------------------------------|-------|
| [15:9] | -      | reserved            | 保留。                                                                                                                                | 0x00  |
| [8]    | RW     | resp_int_disable    | response 中断关闭。<br>0: 使能;<br>1: 关闭。                                                                                                 | 0x0   |
| [7]    | RW     | resp_err_chk_enable | response 错误检查使能。<br>0: 不使能;<br>1: 使能。                                                                                              | 0x0   |
| [6]    | RW     | resp_type           | response R1/R5 类型。<br>0: R1;<br>1: R5。                                                                                             | 0x0   |
| [5]    | RW     | multi_blk_sel       | multiple/single block 选择。<br>0: single block;<br>1: multiple block。                                                                | 0x0   |
| [4]    | RW     | data_xfer_dir       | Data Transfer 方向。<br>0: 控制器到卡;<br>1: 卡到控制器。                                                                                        | 0x0   |
| [3:2]  | RW     | auto_cmd_enable     | Auto Command Enable。<br>00: 关闭;<br>01: Auto CMD12 Enable;<br>10: Auto CMD23 Enable;<br>11: Auto CMD Auto Select。<br>(SDIO 中固定为 00) | 0x0   |
| [1]    | RW     | block_count_enable  | Block count 使能。<br>0: 关闭;                                                                                                          | 0x0   |

| Bits | Access | Name       | Description                                      | Reset |
|------|--------|------------|--------------------------------------------------|-------|
|      |        |            | 1: 使能。                                           |       |
| [0]  | RW     | dma_enable | DMA 使能。<br>0: 无数据传输或者非 DMA 数据传输;<br>1: DMA 数据传输。 | 0x0   |

## CMD\_R

CMD\_R 为 Transfer 模式命令寄存器。

Offset Address: 0x000e

Total Reset Value: 0x0000

| Bits    | Access | Name               | Description                                                                   | Reset |
|---------|--------|--------------------|-------------------------------------------------------------------------------|-------|
| [15:14] | -      | reserved           | 保留。                                                                           | 0x0   |
| [13:8]  | RW     | cmd_index          | 命令编号。                                                                         | 0x00  |
| [7:6]   | RW     | cmd_type           | 命令类型。<br>00: Normal 命令;<br>01: Suspend 命令;<br>10: Resume 命令;<br>11: Abort 命令。 | 0x0   |
| [5]     | RW     | data_present_sel   | 当前是否为数据传输。<br>0: 无数据传输;<br>1: 有数据传输。                                          | 0x0   |
| [4]     | RW     | cmd_idx_chk_enable | 命令编号检查使能。<br>0: 关闭;<br>1: 使能。                                                 | 0x0   |
| [3]     | RW     | cmd_crc_chk_enable | 命令 CRC 检查使能。<br>0: 关闭;<br>1: 使能。                                              | 0x0   |
| [2]     | RW     | sub_cmd_flag       | 子命令标识。                                                                        | 0x0   |

| Bits  | Access | Name             | Description                                                                                                             | Reset |
|-------|--------|------------------|-------------------------------------------------------------------------------------------------------------------------|-------|
|       |        |                  | 0: Main;<br>1: Sub Command.                                                                                             |       |
| [1:0] | RW     | resp_type_select | response 类型。<br>00: 无 response;<br>01: response Length 136;<br>10: response length 48;<br>11: response length 48 check. | 0x0   |

## RESP01\_R

RESP01\_R 为 RESP01 寄存器。

Offset Address: 0x0010    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name   | Description                                      | Reset      |
|--------|--------|--------|--------------------------------------------------|------------|
| [31:0] | RO     | resp01 | RESP01 寄存器。存放控制器发送命令的 response bit[39:8]bits 内容。 | 0x00000000 |

## RESP23\_R

RESP23\_R 为 RESP23 寄存器。

Offset Address: 0x0014    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name   | Description                                  | Reset      |
|--------|--------|--------|----------------------------------------------|------------|
| [31:0] | RO     | resp23 | RESP23 寄存器。存放控制器发送命令的 response bit[71:40]内容。 | 0x00000000 |

## RESP45\_R

RESP45\_R 为 RESP45 寄存器。

Offset Address: 0x0018    Total Reset Value: 0x0000\_0000



| Bits   | Access | Name   | Description                                   | Reset      |
|--------|--------|--------|-----------------------------------------------|------------|
| [31:0] | RO     | resp45 | RESP45 寄存器。存放控制器发送命令的 response bit[103:72]内容。 | 0x00000000 |

## RESP67\_R

RESP67\_R 为 RESP67 寄存器。

Offset Address: 0x001c    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name   | Description                                    | Reset      |
|--------|--------|--------|------------------------------------------------|------------|
| [31:0] | RO     | resp67 | RESP67 寄存器。存放控制器发送命令的 response bit[135:104]内容。 | 0x00000000 |

## BUF\_DATA\_R

BUF\_DATA\_R 为 Buffer 数据寄存器。

Offset Address: 0x0020    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description                                                           | Reset      |
|--------|--------|----------|-----------------------------------------------------------------------|------------|
| [31:0] | RW     | buf_data | 控制器中存放接收发送数据的 Buffer，在 Slave 方式下通过不断写这个寄存器来写数据到卡中，通过不断读这个寄存器来获取卡中的数据。 | 0x00000000 |

## PSTATE\_R

PSTATE\_R 为当前状态寄存器。

Offset Address: 0x0024    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name          | Description             | Reset |
|---------|--------|---------------|-------------------------|-------|
| [31:28] | -      | reserved      | 保留。                     | 0x0   |
| [27]    | RO     | cmd_issue_err | 命令发送出错状态。<br>0: 发送命令无错; | 0x0   |

| Bits    | Access | Name                | Description                                                                 | Reset |
|---------|--------|---------------------|-----------------------------------------------------------------------------|-------|
|         |        |                     | 1: 命令不能被发送出去。                                                               |       |
| [26:25] | -      | reserved            | 保留。                                                                         | 0x0   |
| [24]    | RO     | cmd_line_lvl        | Cmd 管脚状态。<br>0: cmd 管脚为低电平;<br>1: cmd 管脚为高电平。                               | 0x0   |
| [23:20] | RO     | sd_dat_in[3:0]      | Data[3:0]管脚状态, 每个 bit 含义。<br>0: Data 管脚为低电平;<br>1: Data 管脚为高电平。             | 0x0   |
| [19]    | -      | reserved            | 保留。                                                                         | 0x0   |
| [18]    | RO     | Card_detect_pin_lvl | Card_detect_n 管脚状态。<br>0: Card_detect_n 管脚为低电平;<br>1: Card_detect_n 管脚为高电平。 | 0x0   |
| [17:12] | -      | reserved            | 保留。                                                                         | 0x0   |
| [11]    | RO     | buf_rd_enable       | Buffer 读使能。<br>0: 不使能;<br>1: 使能。                                            | 0x0   |
| [10]    | RO     | buf_wr_enable       | Buffer 写使能。<br>0: 不使能;<br>1: 使能。                                            | 0x0   |
| [9]     | RO     | Rd_xfer_active      | 读 transfer 有效。<br>0: 空闲状态;<br>1: 正在读数据。                                     | 0x0   |
| [8]     | RO     | Wr_xfer_active      | 写 transfer 有效。<br>0: 空闲状态;<br>1: 正在写数据。                                     | 0x0   |
| [7:4]   | RO     | sd_dat_in[7:4]      | Data[7:4]管脚状态, 每个 bit 含义。                                                   | 0x0   |

| Bits | Access | Name            | Description                                              | Reset |
|------|--------|-----------------|----------------------------------------------------------|-------|
|      |        |                 | 0: Data 管脚为低电平;<br>1: Data 管脚为高电平。                       |       |
| [3]  | -      | reserved        | 保留。                                                      | 0x0   |
| [2]  | RO     | Dat_line_active | 数据线有效。<br>0: 空闲状态;<br>1: 正在数据传输。                         | 0x0   |
| [1]  | RO     | cmd_inhibit_dat | 带数据的命令有效。<br>0: 空闲状态, 可以发送带数据的命令;<br>1: 数据线正在传输或者读操作有效中。 | 0x0   |
| [0]  | RO     | cmd_inhibit     | 数据线有效。<br>0: 空闲状态, 控制器可以发送命令;<br>1: 忙状态, 控制器不可以发送命令。     | 0x0   |

## HOST\_CTRL1\_R

HOST\_CTRL1\_R 为 HOST 控制 1 寄存器。

Offset Address: 0x0028      Total Reset Value: 0x00

| Bits  | Access | Name                | Description                                                    | Reset |
|-------|--------|---------------------|----------------------------------------------------------------|-------|
| [7]   | RW     | card_detect_sig_lv  | 卡检测信号选择。<br>0: 卡检测信号 card_detect_n 选择 (常规使用);<br>1: 卡检测测试电平选择。 | 0x0   |
| [6]   | RW     | card_detect_test_lv | 卡检测测试电平。<br>0: 无卡;<br>1: 有卡插入。                                 | 0x0   |
| [5]   | -      | reserved            | 保留。                                                            | 0x0   |
| [4:3] | RW     | dma_sel             | DMA 选择。                                                        | 0x0   |

| Bits | Access | Name           | Description                                                                                                                                                                                                                             | Reset |
|------|--------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
|      |        |                | 当 HOST_CTRL2_R 寄存器中 Host Version 4 Enable 为 1 时：<br>00: SDMA;<br>01: 保留;<br>10: ADMA2;<br>11: ADMA2 或者 ADMA3。<br>当 HOST_CTRL2_R 寄存器中 Host Version 4 Enable 为 0 时：<br>00: SDMA;<br>01: 保留;<br>10: 32bit 地址 ADMA2;<br>11: 64bit 地址 ADMA2。 |       |
| [2]  | RW     | high_speed_en  | 高速使能。<br>0: Normal Speed;<br>1: High Speed。                                                                                                                                                                                             | 0x0   |
| [1]  | RW     | dat_xfer_width | 传输数据位宽。<br>0: 1bit;<br>1: 4bit。                                                                                                                                                                                                         | 0x0   |
| [0]  | -      | reserved       | 保留。                                                                                                                                                                                                                                     | 0x0   |

## PWR\_CTRL\_R

PWR\_CTRL\_R 为 HOST 基础配置控制寄存器。

Offset Address: 0x0029    Total Reset Value: 0x00

| Bits  | Access | Name            | Description                     | Reset |
|-------|--------|-----------------|---------------------------------|-------|
| [7:1] | -      | reserved        | 保留。                             | 0x00  |
| [0]   | RW     | sd_bus_pwr_vdd1 | VDD1 Power 使能。<br>0: Power off; | 0x0   |

| Bits | Access | Name | Description  | Reset |
|------|--------|------|--------------|-------|
|      |        |      | 1: Power on. |       |

## WUP\_CTRL\_R

WUP\_CTRL\_R 为 Wakeup 控制寄存器。

Offset Address: 0x002b    Total Reset Value: 0x20

| Bits  | Access | Name                 | Description                                                             | Reset |
|-------|--------|----------------------|-------------------------------------------------------------------------|-------|
| [7:6] | -      | reserved             | 保留。                                                                     | 0x0   |
| [5]   | RW     | sd_vdd1_on_polarity  | sd_vdd1_on 极性控制。<br>0: sd_vdd1_on 极性为正;<br>1: sd_vdd1_on 极性为负。          | 0x1   |
| [4]   | RW     | Write_prot_polarity  | Write_protect 极性控制。<br>0: write_protect 极性为正;<br>1: write_protect 极性为负。 | 0x0   |
| [3]   | RW     | card_detect_polarity | Card_detect 极性控制。<br>0: card_detect 极性为正;<br>1: card_detect 极性为负。       | 0x0   |
| [2:0] | -      | reserved             | 保留。                                                                     | 0x0   |

## CLK\_CTRL\_R

CLK\_CTRL\_R 为时钟控制寄存器。

Offset Address: 0x002c    Total Reset Value: 0x0000

| Bits   | Access | Name       | Description                          | Reset |
|--------|--------|------------|--------------------------------------|-------|
| [15:4] | -      | reserved   | 保留。                                  | 0x00  |
| [3]    | RW     | pll_enable | PLL 使能开关。<br>0: PLL 处于 low power 模式; | 0x0   |

| Bits | Access | Name                | Description                       | Reset |
|------|--------|---------------------|-----------------------------------|-------|
|      |        |                     | 1: PLL 使能。                        |       |
| [2]  | RW     | sd_clk_en           | SD/eMMC 时钟使能。<br>0: 关闭;<br>1: 打开。 | 0x0   |
| [1]  | RO     | internal_clk_stable | 内部时钟状态。<br>0: 不稳定;<br>1: 稳定。      | 0x0   |
| [0]  | RW     | internal_clk_en     | 内部时钟使能。<br>0: 不使能;<br>1: 使能。      | 0x1   |

## TOUT\_CTRL\_R

TOUT\_CTRL\_R 为超时控制寄存器。

Offset Address: 0x002e    Total Reset Value: 0x00

| Bits  | Access | Name     | Description                                                                         | Reset |
|-------|--------|----------|-------------------------------------------------------------------------------------|-------|
| [7:4] | -      | reserved | 保留。                                                                                 | 0x0   |
| [3:0] | RW     | tout_cnt | 数据超时计数值。<br>0x0: $TMCLK \times 2^{13}$ ;<br>0xe: $TMCLK \times 2^{27}$ ;<br>其他: 保留。 | 0x0   |

## SW\_RST\_R

SW\_RST\_R 为软复位控制寄存器。

Offset Address: 0x002f    Total Reset Value: 0x00

| Bits  | Access | Name     | Description | Reset |
|-------|--------|----------|-------------|-------|
| [7:3] | -      | reserved | 保留。         | 0x00  |

| Bits | Access | Name       | Description                    | Reset |
|------|--------|------------|--------------------------------|-------|
| [2]  | RW     | sw_rst_dat | 软复位数据线请求。<br>0: 不复位;<br>1: 复位。 | 0x0   |
| [1]  | RW     | sw_rst_cmd | 软复位命令线请求。<br>0: 不复位;<br>1: 复位。 | 0x0   |
| [0]  | RW     | sw_rst_all | 软复位控制器请求。<br>0: 不复位;<br>1: 复位。 | 0x0   |

## NORMAL\_INT\_STAT\_R

NORMAL\_INT\_STAT\_R 为普通中断状态寄存器。

Offset Address: 0x0030    Total Reset Value: 0x0000

| Bits | Access | Name          | Description                            | Reset |
|------|--------|---------------|----------------------------------------|-------|
| [15] | RO     | err_interrupt | 汇总错误中断状态。<br>0: 无中断;<br>1: 有中断。        | 0x0   |
| [14] | -      | reserved      | 保留。                                    | 0x0   |
| [13] | RO     | fx_event      | TX 事件中断状态。<br>0: 无中断;<br>1: 有中断。       | 0x0   |
| [12] | RO     | re_tune_event | Retuning 事件中断状态。<br>0: 无中断;<br>1: 有中断。 | 0x0   |
| [11] | RO     | int_c         | INT_C 中断状态。<br>0: 无中断;                 | 0x0   |

| Bits | Access | Name           | Description                                | Reset |
|------|--------|----------------|--------------------------------------------|-------|
|      |        |                | 1: 有中断。                                    |       |
| [10] | RO     | int_b          | INT_B 中断状态。<br>0: 无中断;<br>1: 有中断。          | 0x0   |
| [9]  | RO     | int_a          | INT_A 中断状态。<br>0: 无中断;<br>1: 有中断。          | 0x0   |
| [8]  | RO     | card_interrupt | 卡中断中断状态。<br>0: 无中断;<br>1: 有中断。             | 0x0   |
| [7]  | WC     | card_removal   | 卡移除中断状态。<br>0: 无中断;<br>1: 有中断。             | 0x0   |
| [6]  | WC     | card_insertion | 卡插入中断状态。<br>0: 无中断;<br>1: 有中断。             | 0x0   |
| [5]  | WC     | buf_rd_ready   | Buffer 读 ready 中断状态。<br>0: 无中断;<br>1: 有中断。 | 0x0   |
| [4]  | WC     | buf_wr_ready   | Buffer 写 ready 中断状态。<br>0: 无中断;<br>1: 有中断。 | 0x0   |
| [3]  | WC     | dma_interrupt  | DMA 中断状态。<br>0: 无中断;<br>1: 有中断。            | 0x0   |
| [2]  | WC     | bgap_event     | 因停止请求发生 Block Gap 事件中断状态。                  | 0x0   |



| Bits | Access | Name         | Description                                | Reset |
|------|--------|--------------|--------------------------------------------|-------|
|      |        |              | 0: 无中断;<br>1: 有中断。                         |       |
| [1]  | WC     | xfer_comlete | 读/写 transfer 完成中断状态。<br>0: 无中断;<br>1: 有中断。 | 0x0   |
| [0]  | WC     | cmd_complete | 命令完成中断状态。<br>0: 无中断;<br>1: 有中断。            | 0x0   |

## ERROR\_INT\_STAT\_R

ERROR\_INT\_STAT\_R 为错误中断状态寄存器。

Offset Address: 0x0032    Total Reset Value: 0x0000

| Bits | Access | Name         | Description                                              | Reset |
|------|--------|--------------|----------------------------------------------------------|-------|
| [15] | -      | vendor_err3  | 保留。                                                      | 0x0   |
| [14] | -      | vendor_err2  | 保留。                                                      | 0x0   |
| [13] | -      | vendor_err1  | 保留。                                                      | 0x0   |
| [12] | WC     | boot_ack_err | Boot Ack 错误中断, 仅用于 eMMC。<br>0: 无中断;<br>1: 有中断。           | 0x0   |
| [11] | WC     | resp_err     | response 错误中断, 仅用于 SD 模式。<br>0: 无中断;<br>1: 有中断。          | 0x0   |
| [10] | -      | reserved     | 保留。                                                      | 0x0   |
| [9]  | WC     | adma_err     | ADMA 错误中断, 系统总线错误 response, 或者 ADMA2/3 描述子无效。<br>0: 无中断; | 0x0   |

| Bits | Access | Name             | Description                                 | Reset |
|------|--------|------------------|---------------------------------------------|-------|
|      |        |                  | 1: 有中断。                                     |       |
| [8]  | WC     | auto_cmd_err     | Auto CMD 错误中断状态。<br>0: 无中断;<br>1: 有中断。      | 0x0   |
| [7]  | WC     | cur_lmt_err      | Current limit 错误中断状态。<br>0: 无中断;<br>1: 有中断。 | 0x0   |
| [6]  | WC     | data_end_bit_err | 数据 endbit 错误中断状态。<br>0: 无中断;<br>1: 有中断。     | 0x0   |
| [5]  | WC     | data_crc_err     | 数据 CRC 错误中断状态。<br>0: 无中断;<br>1: 有中断。        | 0x0   |
| [4]  | WC     | data_tout_err    | 数据超时错误中断状态。<br>0: 无中断;<br>1: 有中断。           | 0x0   |
| [3]  | WC     | cmd_idx_err      | 命令 Index 错误中断状态。<br>0: 无中断;<br>1: 有中断。      | 0x0   |
| [2]  | WC     | cmd_end_bit_err  | 命令 endbit 错误中断状态。<br>0: 无中断;<br>1: 有中断。     | 0x0   |
| [1]  | WC     | cmd_crc_err      | 命令 CRC 错误中断状态。<br>0: 无中断;<br>1: 有中断。        | 0x0   |
| [0]  | WC     | cmd_tout_err     | 命令超时错误中断状态。<br>0: 无中断;                      | 0x0   |

| Bits | Access | Name | Description | Reset |
|------|--------|------|-------------|-------|
|      |        |      | 1: 有中断。     |       |

## NORMAL\_INT\_STAT\_EN\_R

NORMAL\_INT\_STAT\_EN\_R 为普通中断状态使能寄存器。

Offset Address: 0x0034

Total Reset Value: 0x0000

| Bits | Access | Name                   | Description                             | Reset |
|------|--------|------------------------|-----------------------------------------|-------|
| [15] | -      | reserved               | 保留。                                     | 0x0   |
| [14] | RW     | cqe_event_stat_en      | CMD 事件中断状态使能。<br>0: 不使能;<br>1: 使能。      | 0x0   |
| [13] | RW     | fx_event_stat_en       | TX 事件中断状态使能。<br>0: 不使能;<br>1: 使能。       | 0x0   |
| [12] | RW     | re_tune_event_stat_en  | Retuning 事件中断状态使能。<br>0: 不使能;<br>1: 使能。 | 0x0   |
| [11] | RW     | int_c_stat_en          | INT_C 中断状态使能。<br>0: 不使能;<br>1: 使能。      | 0x0   |
| [10] | RW     | int_b_stat_en          | INT_B 中断状态使能。<br>0: 不使能;<br>1: 使能。      | 0x0   |
| [9]  | RW     | int_a_stat_en          | INT_A 中断状态使能。<br>0: 不使能;<br>1: 使能。      | 0x0   |
| [8]  | RW     | card_interrupt_stat_en | 卡中断状态使能。                                | 0x0   |

| Bits | Access | Name                   | Description                                    | Reset |
|------|--------|------------------------|------------------------------------------------|-------|
|      |        |                        | 0: 不使能;<br>1: 使能。                              |       |
| [7]  | RW     | card_removal_stat_en   | 卡移除中断使能。<br>0: 不使能;<br>1: 使能。                  | 0x0   |
| [6]  | RW     | card_insertion_stat_en | 卡插入中断使能。<br>0: 不使能;<br>1: 使能。                  | 0x0   |
| [5]  | RW     | buf_rd_ready_stat_en   | Buffer 读 ready 中断使能。<br>0: 不使能;<br>1: 使能。      | 0x0   |
| [4]  | RW     | buf_wr_ready_stat_en   | Buffer 写 ready 中断使能。<br>0: 不使能;<br>1: 使能。      | 0x0   |
| [3]  | RW     | dma_interrupt_stat_en  | DMA 中断使能。<br>0: 不使能;<br>1: 使能。                 | 0x0   |
| [2]  | RW     | bgap_event_stat_en     | 因停止请求发生 Block Gap 事件中断使能。<br>0: 不使能;<br>1: 使能。 | 0x0   |
| [1]  | RW     | xfer_complete_stat_en  | 读/写 transfer 完成中断状态使能。<br>0: 不使能;<br>1: 使能。    | 0x0   |
| [0]  | RW     | cmd_complete_stat_en   | 命令完成中断状态使能。<br>0: 不使能;<br>1: 使能。               | 0x0   |

## ERROR\_INT\_STAT\_EN\_R

ERROR\_INT\_STAT\_EN\_R 为错误中断状态使能寄存器。

Offset Address: 0x0036    Total Reset Value: 0x0000

| Bits | Access | Name                 | Description                                                              | Reset |
|------|--------|----------------------|--------------------------------------------------------------------------|-------|
| [15] | RW     | vendor_err_stat_en3  | 保留。                                                                      | 0x0   |
| [14] | RW     | vendor_err_stat_en2  | 保留。                                                                      | 0x0   |
| [13] | RW     | vendor_err_stat_en1  | 保留。                                                                      | 0x0   |
| [12] | RW     | boot_ack_err_stat_en | Boot Ack 错误中断状态使能，仅用于 eMMC。<br>0: 不使能;<br>1: 使能。                         | 0x0   |
| [11] | RW     | resp_err_stat_en     | response 错误中断状态使能，仅用于 SD 模式。<br>0: 不使能;<br>1: 使能。                        | 0x0   |
| [10] | -      | reserved             | 保留。                                                                      | 0x0   |
| [9]  | RW     | adma_err_stat_en     | ADMA 错误中断状态使能。<br>系统总线错误 response，或者 ADMA2/3 描述子无效。<br>0: 不使能;<br>1: 使能。 | 0x0   |
| [8]  | RW     | auto_cmd_err_stat_en | Auto CMD 错误中断状态使能。<br>0: 不使能;<br>1: 使能。                                  | 0x0   |
| [7]  | RW     | cur_lmt_err_stat_en  | Current limit 错误中断状态使能。<br>0: 不使能;                                       | 0x0   |

| Bits | Access | Name                     | Description                              | Reset |
|------|--------|--------------------------|------------------------------------------|-------|
|      |        |                          | 1: 使能。                                   |       |
| [6]  | RW     | data_end_bit_err_stat_en | 数据 endbit 错误中断状态使能。<br>0: 不使能;<br>1: 使能。 | 0x0   |
| [5]  | RW     | data_crc_err_stat_en     | 数据 CRC 错误中断状态使能。<br>0: 不使能;<br>1: 使能。    | 0x0   |
| [4]  | RW     | data_tout_err_stat_en    | 数据超时错误中断状态使能。<br>0: 不使能;<br>1: 使能。       | 0x0   |
| [3]  | RW     | cmd_idx_err_stat_en      | 命令 Index 错误中断状态使能。<br>0: 不使能;<br>1: 使能。  | 0x0   |
| [2]  | RW     | cmd_end_bit_err_stat_en  | 命令 endbit 错误中断状态使能。<br>0: 不使能;<br>1: 使能。 | 0x0   |
| [1]  | RW     | cmd_crc_err_stat_en      | 命令 CRC 错误中断状态使能。<br>0: 不使能;<br>1: 使能。    | 0x0   |
| [0]  | RW     | cmd_tout_err_stat_en     | 命令超时错误中断状态使能。<br>0: 不使能;<br>1: 使能。       | 0x0   |

## NORMAL\_INT\_SIGNAL\_EN\_R

NORMAL\_INT\_SIGNAL\_EN\_R 为普通中断信号使能寄存器。

Offset Address: 0x0038    Total Reset Value: 0x0000

| Bits | Access | Name                     | Description                             | Reset |
|------|--------|--------------------------|-----------------------------------------|-------|
| [15] | -      | reserved                 | 保留                                      | 0x0   |
| [14] | RW     | cqe_event_signal_en      | CMD 事件中断信号使能。<br>0: 不使能;<br>1: 使能。      | 0x0   |
| [13] | RW     | fx_event_signal_en       | TX 事件中断信号使能。<br>0: 不使能;<br>1: 使能。       | 0x0   |
| [12] | RW     | re_tune_event_signal_en  | Retuning 事件中断信号使能。<br>0: 不使能;<br>1: 使能。 | 0x0   |
| [11] | RW     | int_c_signal_en          | INT_C 中断信号使能。<br>0: 不使能;<br>1: 使能。      | 0x0   |
| [10] | RW     | int_b_signal_en          | INT_B 中断信号使能。<br>0: 不使能;<br>1: 使能。      | 0x0   |
| [9]  | RW     | int_a_signal_en          | INT_A 中断信号使能。<br>0: 不使能;<br>1: 使能。      | 0x0   |
| [8]  | RW     | card_interrupt_signal_en | 卡中断信号使能。<br>0: 不使能;<br>1: 使能。           | 0x0   |
| [7]  | RW     | card_removal_signal_en   | 卡移除中断信号使能。<br>0: 不使能;<br>1: 使能。         | 0x0   |
| [6]  | RW     | card_insertion_signal_en | 卡插入中断信号使能。<br>0: 不使能;                   | 0x0   |

| Bits | Access | Name                    | Description                                      | Reset |
|------|--------|-------------------------|--------------------------------------------------|-------|
|      |        |                         | 1: 使能。                                           |       |
| [5]  | RW     | buf_rd_ready_signal_en  | Buffer 读 ready 信号使能。<br>0: 不使能;<br>1: 使能。        | 0x0   |
| [4]  | RW     | buf_wr_ready_signal_en  | Buffer 写 ready 信号使能。<br>0: 不使能;<br>1: 使能。        | 0x0   |
| [3]  | RW     | dma_interrupt_signal_en | DMA 中断信号使能。<br>0: 不使能;<br>1: 使能。                 | 0x0   |
| [2]  | RW     | bgap_event_signal_en    | 因停止请求发生 Block Gap 事件中断信号使能。<br>0: 不使能;<br>1: 使能。 | 0x0   |
| [1]  | RW     | xfer_complete_signal_en | 读/写 transfer 完成中断信号使能。<br>0: 不使能;<br>1: 使能。      | 0x0   |
| [0]  | RW     | cmd_complete_signal_en  | 命令完成中断信号使能。<br>0: 不使能;<br>1: 使能。                 | 0x0   |

## ERROR\_INT\_SIGNAL\_EN\_R

ERROR\_INT\_SIGNAL\_EN\_R 为错误中断信号使能寄存器。

Offset Address: 0x003a    Total Reset Value: 0x0000

| Bits | Access | Name                  | Description | Reset |
|------|--------|-----------------------|-------------|-------|
| [15] | RW     | vendor_err_signal_en3 | 保留。         | 0x0   |



| Bits | Access | Name                       | Description                                                        | Reset |
|------|--------|----------------------------|--------------------------------------------------------------------|-------|
| [14] | RW     | vendor_err_signal_en2      | 保留。                                                                | 0x0   |
| [13] | RW     | vendor_err_signal_en1      | 保留。                                                                | 0x0   |
| [12] | RW     | boot_ack_err_signal_en     | Boot Ack 错误中断信号使能，仅用于 eMMC。<br>0：不使能；<br>1：使能。                     | 0x0   |
| [11] | RW     | resp_err_signal_en         | response 错误中断信号使能，仅用于 SD 模式。<br>0：不使能；<br>1：使能。                    | 0x0   |
| [10] | -      | reserved                   | 保留。                                                                | 0x0   |
| [9]  | RW     | adma_err_signal_en         | ADMA 错误中断信号使能，系统总线错误 response，或者 ADMA2/3 描述子无效。<br>0：不使能；<br>1：使能。 | 0x0   |
| [8]  | RW     | auto_cmd_err_signal_en     | Auto CMD 错误中断信号使能。<br>0：不使能；<br>1：使能。                              | 0x0   |
| [7]  | RW     | cur_lmt_err_signal_en      | Current limit 错误中断信号使能。<br>0：不使能；<br>1：使能。                         | 0x0   |
| [6]  | RW     | data_end_bit_err_signal_en | 数据 endbit 错误中断信号使能。<br>0：不使能；<br>1：使能。                             | 0x0   |
| [5]  | RW     | data_crc_err_signal_en     | 数据 CRC 错误中断信号使能。<br>0：不使能；<br>1：使能。                                | 0x0   |

| Bits | Access | Name                      | Description                              | Reset |
|------|--------|---------------------------|------------------------------------------|-------|
| [4]  | RW     | data_tout_err_signal_en   | 数据超时错误中断信号使能。<br>0: 不使能;<br>1: 使能。       | 0x0   |
| [3]  | RW     | cmd_idx_err_signal_en     | 命令 Index 错误中断信号使能。<br>0: 不使能;<br>1: 使能。  | 0x0   |
| [2]  | RW     | cmd_end_bit_err_signal_en | 命令 endbit 错误中断信号使能。<br>0: 不使能;<br>1: 使能。 | 0x0   |
| [1]  | RW     | cmd_crc_err_signal_en     | 命令 CRC 错误中断信号使能。<br>0: 不使能;<br>1: 使能。    | 0x0   |
| [0]  | RW     | cmd_tout_err_signal_en    | 命令超时错误中断信号使能。<br>0: 不使能;<br>1: 使能。       | 0x0   |

## AUTO\_CMD\_STAT\_R

AUTO\_CMD\_STAT\_R 为 Auto 命令状态寄存器。

Offset Address: 0x003c    Total Reset Value: 0x0000

| Bits   | Access | Name                      | Description                                                | Reset |
|--------|--------|---------------------------|------------------------------------------------------------|-------|
| [15:8] | -      | reserved                  | 保留。                                                        | 0x00  |
| [7]    | RO     | cmd_not_issued_auto_cmd12 | 有 Auto CMD12 错误时发送后续命令状态。<br>0: 无错误;<br>1: 还没有通过数据线发送 CMD。 | 0x0   |
| [6]    | -      | reserved                  | 保留。                                                        | 0x0   |

| Bits | Access | Name                | Description                              | Reset |
|------|--------|---------------------|------------------------------------------|-------|
| [5]  | RO     | auto_cmd_resp_err   | Auto CMD response 状态。<br>0: 无错;<br>1: 错。 | 0x0   |
| [4]  | RO     | auto_cmd_idx_err    | Auto CMD Index 状态。<br>0: 无错;<br>1: 错。    | 0x0   |
| [3]  | RO     | auto_cmd_endbit_err | Auto CMD Endbit 状态。<br>0: 无错;<br>1: 错。   | 0x0   |
| [2]  | RO     | auto_cmd_crc_err    | Auto CMD CRC 状态。<br>0: 无错;<br>1: 错。      | 0x0   |
| [1]  | RO     | auto_cmd_tout_err   | Auto CMD 超时状态。<br>0: 无错;<br>1: 错。        | 0x0   |
| [0]  | RO     | auto_cmd12_not_exec | Auto CMD12 执行状态。<br>0: 执行;<br>1: 没有执行。   | 0x0   |

## HOST\_CTRL2\_R

HOST\_CTRL2\_R 为控制器 2 寄存器。

Offset Address: 0x003e    Total Reset Value: 0x0000

| Bits | Access | Name              | Description                                       | Reset |
|------|--------|-------------------|---------------------------------------------------|-------|
| [15] | RW     | preset_val_enable | 预设值自动选择使能。<br>0: 不使能, SDCLK 和驱动能力由驱动控制;<br>1: 使能。 | 0x0   |

| Bits  | Access | Name             | Description                                                                 | Reset |
|-------|--------|------------------|-----------------------------------------------------------------------------|-------|
| [14]  | RW     | async_int_enable | 异步中断使能。<br>0: 关闭;<br>1: 打开。                                                 | 0x0   |
| [13]  | RW     | addressing       | 总线 64bit 地址使能。<br>0: 不使能, 使用 32bit 地址;<br>1: 使能, 使用 64bit 地址。               | 0x0   |
| [12]  | RW     | host_ver4_enable | 控制器版本 4 使能。<br>0: 3.0 版本;<br>1: 4.0 版本。                                     | 0x0   |
| [11]  | RW     | cmd23_enable     | CMD23 使能。<br>0: Auto CMD23 关闭;<br>1: Auto CMD23 使能。                         | 0x0   |
| [10]  | RW     | adma2_len_mode   | ADMA2 长度。<br>0: 16bit 数据长度;<br>1: 26bit 数据长度。                               | 0x0   |
| [9:8] | -      | reserved         | 保留。                                                                         | 0x0   |
| [7]   | RW     | sample_clk_sel   | sample 时钟选择。<br>0: 选择 fixed 时钟采数据;<br>1: 选 tuned 时钟采数据。                     | 0x0   |
| [6]   | RW     | exec_tuning      | 执行 tuning,完成后自动清 0。<br>0: 没做 tuning 或者 tuning 完成;<br>1: 执行 tuning。          | 0x0   |
| [5:4] | RW     | drv_strength_sel | 驱动能力选择。<br>00: 选择 TYPEB;<br>01: 选择 TYPEA;<br>10: 选择 TYPEC;<br>11: 选择 TYPED。 | 0x0   |

| Bits  | Access | Name         | Description                                                                | Reset |
|-------|--------|--------------|----------------------------------------------------------------------------|-------|
| [3]   | -      | reserved     | 保留。                                                                        | 0x0   |
| [2:0] | RW     | uhs_mode_sel | eMMC 模式：<br>000: Legacy;<br>001: High Speed SDR;<br>011: HS200;<br>其他: 保留。 | 0x0   |

## CAPABILITIES1\_R

CAPABILITIES1\_R 为能力 1 寄存器。

Offset Address: 0x0040    Total Reset Value: 0x3F6E\_0181

| Bits    | Access | Name              | Description                                                                                      | Reset |
|---------|--------|-------------------|--------------------------------------------------------------------------------------------------|-------|
| [31:30] | RO     | slot_type_r       | Slot 类型。<br>00: 可移除卡 Slot;<br>01: Embedded Slot;<br>10: 共享总线 Slot(SD 模式下使用);<br>11: UHS2(暂时不支持)。 | 0x0   |
| [29]    | RO     | async_int_support | 同步中断是否支持。<br>0: 不支持;<br>1: 支持。                                                                   | 0x1   |
| [28]    | RO     | sys_addr_64_v3    | 64bit 系统地址用于 V3 版本。<br>0: 不支持;<br>1: 支持。                                                         | 0x1   |
| [27]    | RO     | sys_addr_64_v4    | 64bit 系统地址用于 V4 版本。<br>0: 不支持;<br>1: 支持。                                                         | 0x1   |
| [26]    | RO     | volt_18           | 电压支持 1.8V。                                                                                       | 0x1   |

| Bits    | Access | Name               | Description                                     | Reset |
|---------|--------|--------------------|-------------------------------------------------|-------|
|         |        |                    | 0: 不支持;<br>1: 支持。                               |       |
| [25]    | RO     | volt_30            | 电压支持 3.0V。<br>0: 不支持;<br>1: 支持。                 | 0x1   |
| [24]    | RO     | volt_33            | 电压支持 3.3V。<br>0: 不支持;<br>1: 支持。                 | 0x1   |
| [23]    | RO     | sus_res_support    | 暂停、恢复支持。<br>0: 不支持;<br>1: 支持。                   | 0x0   |
| [22]    | RO     | sdma_support       | SDMA 支持。<br>0: 不支持;<br>1: 支持。                   | 0x1   |
| [21]    | RO     | high_speed_support | 是否支持 High Speed。<br>0: 不支持;<br>1: 支持。           | 0x1   |
| [20]    | -      | reserved           | 保留。                                             | 0x0   |
| [19]    | RO     | adma2_support      | 是否支持 ADMA2。<br>0: 不支持;<br>1: 支持。                | 0x1   |
| [18]    | RO     | embedded_8_bit     | 是否支持 Embedded 器件 8bit。<br>0: 不支持;<br>1: 支持。     | 0x1   |
| [17:16] | RO     | max_blk_len        | 最大 block 长度。<br>0x0: 512Byte;<br>0x1: 1024Byte; | 0x2   |

| Bits   | Access | Name          | Description                                                                | Reset |
|--------|--------|---------------|----------------------------------------------------------------------------|-------|
|        |        |               | 0x2: 2048Byte;<br>0x3: 保留。                                                 |       |
| [15:8] | RO     | base_clk_freq | SD 时钟基础频率。<br>0x0: 1MHz;<br>0x3f: 63MHz;<br>0x40~0xff: 不支持。                | 0x01  |
| [7]    | RO     | tout_clk_unit | 超时时钟单位。<br>0: KHz;<br>1: MHz。                                              | 0x1   |
| [6]    | -      | reserved      | 保留。                                                                        | 0x0   |
| [5:0]  | RO     | tout_clk_freq | 超时频率。<br>0x1: 1KHz/1MHz;<br>0x2: 2KHz/2MHz;<br>.....<br>0x3f: 63KHz/63MHz。 | 0x01  |

## CAPABILITIES2\_R

CAPABILITIES2\_R 为能力 2 寄存器。

Offset Address: 0x0044    Total Reset Value: 0x0800\_2077

| Bits    | Access | Name             | Description                          | Reset |
|---------|--------|------------------|--------------------------------------|-------|
| [31:29] | -      | reserved         | 保留。                                  | 0x0   |
| [28]    | RO     | vdd2_18v_support | 是否支持 1.8V VDDR。<br>0: 不支持;<br>1: 支持。 | 0x0   |
| [27]    | RO     | adma3_support    | 是否支持 ADMA3。<br>0: 不支持;               | 0x1   |

| Bits    | Access | Name             | Description                                                                                                                   | Reset |
|---------|--------|------------------|-------------------------------------------------------------------------------------------------------------------------------|-------|
|         |        |                  | 1: 支持。                                                                                                                        |       |
| [26:16] | -      | reserved         | 保留。                                                                                                                           | 0x000 |
| [15:14] | RO     | re_tuning_modes  | retuning 模式。<br>00: MODE1,Timer;<br>01: MODE2,Timer 和 ReTuning 请求;<br>10: MODE3,Auto retuning Timer 和 Retuning 请求;<br>11: 保留。 | 0x0   |
| [13]    | RO     | use_tuning_ser50 | SDR50 使用 Tuning。<br>0: 不使用;<br>1: 使用。                                                                                         | 0x1   |
| [12]    | -      | reserved         | 保留。                                                                                                                           | 0x0   |
| [11:8]  | RO     | retune_cnt       | Retuning 计数。<br>0x1: 1 秒;<br>0x3: 4 秒;<br>其他: 保留。                                                                             | 0x0   |
| [7]     | -      | reserved         | 保留。                                                                                                                           | 0x0   |
| [6]     | RO     | drv_typed        | 是否支持 TYPED。<br>0: 不支持;<br>1: 支持。                                                                                              | 0x1   |
| [5]     | RO     | drv_typec        | 是否支持 TYPEC。<br>0: 不支持;<br>1: 支持。                                                                                              | 0x1   |
| [4]     | RO     | drv_typea        | 是否支持 TYPEA。<br>0: 不支持;<br>1: 支持。                                                                                              | 0x1   |
| [3]     | RO     | uhs2_support     | 是否支持 UHS2。                                                                                                                    | 0x0   |



| Bits | Access | Name           | Description                       | Reset |
|------|--------|----------------|-----------------------------------|-------|
|      |        |                | 0: 不支持;<br>1: 支持。                 |       |
| [2]  | RO     | ddr50_support  | 是否支持 DDR50。<br>0: 不支持;<br>1: 支持。  | 0x1   |
| [1]  | RO     | sdr104_support | 是否支持 SDR104。<br>0: 不支持;<br>1: 支持。 | 0x1   |
| [0]  | RO     | sdr50_support  | 是否支持 SDR50。<br>0: 不支持;<br>1: 支持。  | 0x1   |

## CURR\_CAPBILITIES1\_R

CURR\_CAPBILITIES1\_R 为当前能力 1 寄存器。

Offset Address: 0x0048    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name        | Description | Reset |
|---------|--------|-------------|-------------|-------|
| [31:24] | -      | reserved    | 保留。         | 0x00  |
| [23:16] | RO     | max_cur_18v | 1.8V 下电流值。  | 0x00  |
| [15:8]  | RO     | max_cur_30v | 3.0V 下电流值。  | 0x00  |
| [7:0]   | RO     | mac_cur_33v | 3.3V 下电流值。  | 0x00  |

## ADMA\_ERR\_STAT\_R

ADMA\_ERR\_STAT\_R 为 ADMA 错误状态寄存器。

Offset Address: 0x0054    Total Reset Value: 0x00

| Bits  | Access | Name            | Description                                                                | Reset |
|-------|--------|-----------------|----------------------------------------------------------------------------|-------|
| [7:3] | -      | reserved        | 保留。                                                                        | 0x00  |
| [2]   | RO     | adma_len_err    | ADMA 长度不匹配错误状态。<br>0: 无错;<br>1: 有错。                                        | 0x0   |
| [1:0] | RO     | adma_err_states | ADMA 错误状态。<br>00: ST_STOP;<br>01: ST_FDS;<br>10: ST_UNUSED;<br>11: ST_TFR。 | 0x0   |

## ADMA\_SA\_LOW\_R

ADMA\_SA\_LOW\_R 为 ADMA 系统地址低 32bit 寄存器。

Offset Address: 0x0058    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name        | Description       | Reset      |
|--------|--------|-------------|-------------------|------------|
| [31:0] | RW     | adma_sa_low | ADMA 系统地址低 32bit。 | 0x00000000 |

## ADMA\_SA\_HIGH\_R

ADMA\_SA\_HIGH\_R 为 ADMA 系统地址高 32bit 寄存器。

Offset Address: 0x005c    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name         | Description       | Reset      |
|--------|--------|--------------|-------------------|------------|
| [31:0] | RW     | adma_sa_high | ADMA 系统地址高 32bit。 | 0x00000000 |

## ADMA\_ID\_LOW\_R

ADMA\_ID\_LOW\_R 为 ADMA Intergrated 描述子地址低 32bit 寄存器。

Offset Address: 0x0078    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name        | Description                    | Reset      |
|--------|--------|-------------|--------------------------------|------------|
| [31:0] | RW     | adma_id_low | ADMA Intergrated 描述子地址低 32bit。 | 0x00000000 |

## ADMA\_ID\_HIGH\_R

ADMA\_ID\_HIGH\_R 为 ADMA Intergrated 描述子地址高 32bit 寄存器。

Offset Address: 0x007c Total Reset Value: 0x0000\_0000

| Bits   | Access | Name         | Description                    | Reset      |
|--------|--------|--------------|--------------------------------|------------|
| [31:0] | RW     | adma_id_high | ADMA Intergrated 描述子地址高 32bit。 | 0x00000000 |

## MSHC\_VER\_ID\_R

MSHC\_VER\_ID\_R 为控制器版本 ID 寄存器。

Offset Address: 0x0500 Total Reset Value: 0x3137\_302A

| Bits   | Access | Name        | Description   | Reset      |
|--------|--------|-------------|---------------|------------|
| [31:0] | RO     | mshc_ver_id | 控制器版本 ID 寄存器。 | 0x3137302A |

## MSHC\_VER\_TYPE\_R

MSHC\_VER\_TYPE\_R 为控制器版本类型寄存器。

Offset Address: 0x0504 Total Reset Value: 0x6761\_2A2A

| Bits   | Access | Name        | Description | Reset      |
|--------|--------|-------------|-------------|------------|
| [31:0] | RO     | mshc_ver_id | 控制器版本类型寄存器。 | 0x67612A2A |

## MBIU\_CTRL\_R

MBIU\_CTRL\_R 为 MBIU 控制寄存器。

Offset Address: 0x0510 Total Reset Value: 0x0000\_000F

| Bits   | Access | Name             | Description                                                                       | Reset     |
|--------|--------|------------------|-----------------------------------------------------------------------------------|-----------|
| [31:4] | -      | reserved         | 保留。                                                                               | 0x0000000 |
| [3]    | RW     | gm_enburst16     | 16burst 使能。<br>0: 关闭 16burst;<br>1; 使能 16burst;                                   | 0x1       |
| [2]    | RW     | gm_enburst8      | 8burst 使能。<br>0: 关闭 8burst;<br>1; 打开 8burst;                                      | 0x1       |
| [1]    | RW     | gm_enburst4      | 4burst 使能。<br>0: 关闭 4burst;<br>1; 打开 4burst;                                      | 0x1       |
| [0]    | RW     | gm_enburst_undef | burst 配置生效使能。<br>0: 按下面三种配置 gm_enburst4/8/16 产生固定 burst;<br>1: 按实际数据长度产生对应 burst; | 0x1       |

## EMMC\_CTRL\_R

EMMC\_CTRL\_R 为 eMMC 控制寄存器。

Offset Address: 0x052c Total Reset Value: 0x0000

| Bits    | Access | Name                 | Description                                            | Reset |
|---------|--------|----------------------|--------------------------------------------------------|-------|
| [15:10] | -      | reserved             | 保留                                                     | 0x00  |
| [9]     | RW     | cqe_algo_sel         | 任务排列算法选择。<br>0: 高优先级的任务先执行, 同等优先级的按先来先执行;<br>1: 先来先执行。 | 0x0   |
| [8:2]   | -      | reserved             | 保留。                                                    | 0x00  |
| [1]     | RW     | disable_data_crc_chk | 关闭数据 CRC 检查。                                           | 0x00  |

| Bits | Access | Name         | Description                           | Reset |
|------|--------|--------------|---------------------------------------|-------|
|      |        |              | 0: 数据 CRC 检查使能;<br>1: 数据 CRC 检查关闭。    |       |
| [0]  | RW     | card_is_emmc | 连接的卡类型。<br>0: 非 emmc 卡;<br>1: emmc 卡。 | 0x0   |

## BOOT\_CTRL\_R

BOOT\_CTRL\_R 为 BOOT 控制寄存器。

Offset Address: 0x052e    Total Reset Value: 0x0000

| Bits    | Access | Name            | Description                                                                                                                        | Reset |
|---------|--------|-----------------|------------------------------------------------------------------------------------------------------------------------------------|-------|
| [15:12] | RW     | boot_tout_cnt   | boot ack 超时计数。<br>0x0: $TMCLK \times 2^{13}$ ;<br>0x1: $TMCLK \times 2^{14}$ ;<br>.....<br>0xE: $TMCLK \times 2^{27}$ ;<br>其他: 保留。 | 0x00  |
| [11:9]  | -      | reserved        | 保留。                                                                                                                                | 0x0   |
| [8]     | RW     | boot_ack_enable | boot ack 使能。<br>0: 关闭;<br>1: 使能。                                                                                                   | 0x0   |
| [7]     | WO     | validate_boot   | 强制 boot 有效使能。<br>0: 无影响;<br>1: boot 使能有效。                                                                                          | 0x0   |
| [6:1]   | -      | reserved        | 保留。                                                                                                                                | 0x00  |
| [0]     | RW     | man_boot_en     | 强制 boot 使能, 要和 validate_boot 一起配置, boot 传输完成后, 这个 bit 会自清                                                                          | 0x0   |

| Bits | Access | Name | Description              | Reset |
|------|--------|------|--------------------------|-------|
|      |        |      | 0: 无影响;<br>1: boot 使能有效。 |       |

## GP\_OUT\_R

GP\_OUT\_R 为 eMMC 复位和 GPIO 寄存器。

Offset Address: 0x0534    Total Reset Value: 0x0000\_0001

| Bits   | Access | Name       | Description                      | Reset    |
|--------|--------|------------|----------------------------------|----------|
| [31:1] | -      | reserved   | 保留。                              | 0x000000 |
| [0]    | RW     | emmc_rst_n | eMMC 卡复位信号。<br>0: 复位;<br>1: 不复位。 | 0x1      |

## AT\_CTRL\_R

AT\_CTRL\_R 为 tuning 控制寄存器。

Offset Address: 0x0540    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name         | Description                        | Reset    |
|--------|--------|--------------|------------------------------------|----------|
| [31:5] | -      | reserved     | 保留。                                | 0x000000 |
| [4]    | RW     | sw_tuning_en | 软件配置 tuning 使能。<br>0: 不能使<br>1: 使能 | 0x00     |
| [3:0]  | -      | reserved     | 保留。                                | 0x0      |

## AT\_STAT\_R

AT\_STAT\_R 为 tuning 状态寄存器。

Offset Address: 0x0544    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name           | Description | Reset    |
|--------|--------|----------------|-------------|----------|
| [31:8] | -      | reserved       | 保留。         | 0x000000 |
| [7:0]  | RW     | sw_ph_sel_code | 软件配置相位值。    | 0x0      |

## MULTI\_CYC\_EN

MULTI\_CYC\_EN 为多时钟延迟控制寄存器。

Offset Address: 0x054c Total Reset Value: 0x0004\_0227

| Bits    | Access | Name                    | Description                                                                                                                  | Reset    |
|---------|--------|-------------------------|------------------------------------------------------------------------------------------------------------------------------|----------|
| [31:19] | -      | reserved                | 保留。                                                                                                                          | 0x000000 |
| [18]    | RW     | wr_autocmd12_dly        | 控制控制器 tx 的内部信号<br>gen_auto_cmd12_c 是否 delay 1T                                                                               | 0x1      |
| [17]    | RW     | sd4_data_stop_clk_nodly | 在信号 cmd12_in_proj==0 情况下将<br>stop_sdclk 信号提前 1T;<br><br>以下两个信号除外, 仍然使用原本的<br>stop_sdclk 作为条件, 不受影响<br>(blk_gap,stop_sdclk_d) | 0x0      |
| [16]    | RW     | cmd_out_clkneg          | 在 ddr_mode==1 情况下控制是否将<br>ccmd_out,ccmd_out_en 沿 cclk_tx 的负沿<br>输出                                                           | 0x0      |
| [15]    | RW     | pd_mux_bypass           | 连接到 PHY 的控制信号 sdio_dig_phy_ctrl<br>bit14                                                                                     | 0x0      |
| [14:12] | RW     | edge_detect_ctrl[5:3]   | 连接到 PHY 的控制信号 sdio_dig_phy_ctrl<br>bit12~10                                                                                  | 0x0      |
| [11]    | RO     | tuning_find_edge        | 连接到 PHY 的状态信号<br>sdio_dig_phy_stat                                                                                           | 0x0      |
| [10:8]  | RW     | edge_detect_ctrl[2:0]   | 连接到 PHY 的控制信号 sdio_dig_phy_ctrl<br>bit9~7                                                                                    | 0x2      |
| [7]     | -      | reserved                | 保留。                                                                                                                          | 0x0      |
| [6:4]   | RW     | multi_cycle_en          | 数据输出 delay 控制<br><br>bit0: 有 response 的 ccmd_out_en_int 的<br>下降沿 delay 1T, 1 表示 delay                                        | 0x2      |

|       |    |                 |                                                                                                                                                                                                                                                                                                                                                                                                                                      |     |
|-------|----|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
|       |    |                 | bit1: cdat_out_en 的上升沿 delay 1T, 1 表示不 delay, 0 表示要 delay<br>bit2: cdat_out_en 的下降沿 delay 1T, 1 表示 delay                                                                                                                                                                                                                                                                                                                             |     |
| [3:0] | RW | digphy_dly_ctrl | 连接到 PHY 的控制信号 sdio_dig_phy_ctrl<br>bi3~0 (1 表示 delay)<br>bit0: 控制 PHY 内部的 cmd_out 和 cmd_out_en 是否 delay 1T<br>bit1: 控制 PHY 内部的数据_out 和数据_out_en 是否 delay 1T<br>bit2: 控制 PHY 内部的 cmd in 是否 delay 1T<br>bit3: 控制 PHY 内部的数据 in 是否 delay 1T<br><br>附加修改: 控制器 rx 的内部信号 sample_gen_auto_cmd12 需要提前 digphy_dly_ctrl[3]+digphy_dly_ctrl[0] 个周期, 以适应 PHY 的 delay 带来的影响。<br>(CMD12: STOP_TRANSMISSION, forces the Device to stop transmission) | 0x7 |

## MEM\_CONFIG

MEM\_CONFIG 为 mem 调速控制寄存器。

Offset Address: 0x0550    Total Reset Value: 0x0000\_0007

| Bits  | Access | Name              | Description        | Reset    |
|-------|--------|-------------------|--------------------|----------|
| [7]   | RW     | peri_ras_stov     | RAS STOV MEM 调速配置  | 0x0      |
| [6]   | RW     | peri_ras_emas     | RAS EMAS MEM 调速配置  | 0x000000 |
| [5:4] | RW     | peri_ras_emaw     | RAS EMAW MEM 调速配置  | 0x0      |
| [3:1] | RW     | peri_ras_ema      | RAS EMA MEM 调速配置   | 0x3      |
| [0]   | RW     | peri_ram_ck_gt_en | 恒电区 MEM 不操作时自动门控使能 | 0x1      |



|  |  |  |                  |  |
|--|--|--|------------------|--|
|  |  |  | 0: 不使能;<br>1: 使能 |  |
|--|--|--|------------------|--|

## 2.5 GPIO

### 2.5.1 概述

每个 GPIO(General-purpose input/output)管脚可以配置为输入或者输出。这些管脚用于生成特定应用的输出信号或采集特定应用的输入信号。作为输入管脚时，GPIO 可作为中断源；作为输出管脚时，每个 GPIO 都可以独立地清 0 或置 1。

GPIO 可以根据电平或跳变值产生可屏蔽的中断。GPIOINTR (General Purpose Input Output Interrupt) 信号给中断控制器一个指示，表示有中断发生。

#### 须知

对于默认是输出信号的管脚上复用的 GPIO，请注意对接芯片和器件的管脚必须是输入。

### 2.5.2 特点

- 每个 GPIO 管脚均可配置为输入、输出。
- 作为输入管脚时，可作为中断源。
- 作为输出管脚时，每个 GPIO 管脚都可以独立地清 0 或置 1。
- 系统提供 9 组 GPIO 控制器。

### 2.5.3 工作方式

#### 接口复位

在芯片上电复位或系统复位时，GPIO 模块同时被复位，GPIO 管脚在复位之后处于输入状态。

## 通用输入输出

每个管脚可以配置为输入或者输出，具体步骤如下：

- 步骤 1 参考“管脚复用控制寄存器”配置管脚的相应位，使能需要使用的 GPIO 管脚功能。
- 步骤 2 配置寄存器 GPIO\_DIR，选择 GPIO 是作为输入还是输出。
- 步骤 3 当配置成输入管脚时，读取 GPIO\_DATA 寄存器可查看输入信号值；当配置成输出管脚时，通过向 GPIO\_DATA 寄存器写入输出值可控制 GPIO 管脚输出电平。

----结束

### 须知

当 GPIO 用作输出时，应不使能 GPIO 的中断功能，否则当输出信号符合中断产生条件时，会产生 GPIO 中断。

## 中断操作

GPIO 的中断通过 GPIO\_IS 等 7 个寄存器进行控制。通过这些寄存器可以选择中断源、极性以及边沿特性。GPIO 对应的中断号请参见“3.3 中断系统”。

当有多个中断同时发生时，将会统一汇集成一个中断进行上报，GPIO 的中断映射关系请参见“中断系统”。

GPIO\_IS、GPIO\_IBE、GPIO\_IEV 三个寄存器共同决定了中断源的特性和中断触发类别。

通过 GPIO\_RIS 和 GPIO\_MIS 分别读取中断的原始状态和屏蔽后的状态。通过 GPIO\_IE 可以控制中断的最终上报情况。此外还提供了单独的 GPIO\_IC 用于对中断状态进行清除控制。

每个 GPIO 管脚都可以配置成中断模式，配置步骤如下：

- 步骤 1 配置 GPIO\_IS，选择边沿触发或电平触发。
- 步骤 2 配置 GPIO\_IEV，选择下降沿/上升沿触发和高电平/低电平触发。
- 步骤 3 如果选择边沿触发，需配置 GPIO\_IBE，选择单沿或双沿触发方式。
- 步骤 4 向寄存器 GPIO\_IC 写 0xFF，清中断。
- 步骤 5 配置 GPIO\_IE 为 1，使能中断。

----结束

须知

初始化过程中 GPIO 管脚上数据需保持稳定，以免产生假中断。

GPIO 的中断设置由 7 个寄存器控制。当有一个或多个 GPIO 管脚产生中断，一个组合中断会输送到中断控制器。边沿触发和电平触发有以下不同：

- 边沿触发的中断：软件必需清除该中断以使能更深的中断。
- 电平触发的中断：外部中断源应该保持该电平直到处理器识别到该中断。

编程顺序

- 步骤 1 配置 pinshare，使 PAD 复用成 GPIO 口，具体参考 IO 复用表单；
- 步骤 2 配置 GPIO 口朝向寄存器，寄存器偏移地址为 0x400 寄存器（GPIO\_DIR）低 8 位对应 GPIO 口的 8 个管脚，配置成 1 时为输出，配置成 0 时为输入；
- 步骤 3 读取或写入 GPIO 接口值，寄存器偏移地址为 0x3fc(GPIO\_DATA)，寄存器低 8 位对应 GPIO 口 8 个管脚。

2.5.4 GPIO 寄存器概览

芯片的 GPIO 基地址如表 2-15 所示。

表2-15 芯片 GPIO 寄存器对应的基地址

| GPIO 控制器 | 基地址         |
|----------|-------------|
| GPIO8    | 0x120B_8000 |
| GPIO7    | 0x120B_7000 |
| GPIO6    | 0x120B_6000 |
| GPIO5    | 0x120B_5000 |
| GPIO4    | 0x120B_4000 |
| GPIO3    | 0x120B_3000 |
| GPIO2    | 0x120B_2000 |
| GPIO1    | 0x120B_1000 |
| GPIO0    | 0x120B_0000 |



说明

GPIO<sub>n</sub> 对应的寄存器地址为：GPIO<sub>n</sub> 基地址+该寄存器偏移地址。

表 2-16 是单组 GPIO 内部寄存器的偏移地址以及定义，GPIO<sub>0</sub> ~ GPIO<sub>n</sub> 具有相同的寄存器组。

表2-16 GPIO 寄存器概览

| 偏移地址          | 名称        | 描述             |
|---------------|-----------|----------------|
| 0x000 ~ 0x3FC | GPIO_DATA | GPIO 数据寄存器     |
| 0x400         | GPIO_DIR  | GPIO 方向控制寄存器   |
| 0x404         | GPIO_IS   | GPIO 中断触发寄存器   |
| 0x408         | GPIO_IBE  | GPIO 双沿触发中断寄存器 |
| 0x40C         | GPIO_IEV  | GPIO 触发中断条件寄存器 |
| 0x410         | GPIO_IE   | GPIO 中断屏蔽寄存器   |
| 0x414         | GPIO_RIS  | GPIO 原始中断状态寄存器 |
| 0x418         | GPIO_MIS  | GPIO 屏蔽状态中断寄存器 |
| 0x41C         | GPIO_IC   | GPIO 中断清除寄存器   |

## 2.5.5 GPIO 寄存器描述

### GPIO\_DATA

GPIO\_DATA 为 GPIO 数据寄存器。用来对输入或输出数据进行缓存。

当配置 GPIO\_DIR 中对应位为输出时，写入 GPIO\_DATA 寄存器的值将会输出到相应的管脚（注意需要配置正确的管脚复用）；如果配置为输入时，将会读取相应输入管脚的值。

## 须知

当 GPIO\_DIR 相应的比特配置为输入时，有效读取的结果将返回管脚的值；当配置为输出的时候，有效读取的结果将返回写入的值。

GPIO\_DATA 寄存器利用 PADDR[9:2]实现了读写寄存器比特的屏蔽操作。该寄存器对应 256 个地址空间。PADDR[9:2]分别对应 GPIO\_DATA[7:0]，当相应的 bit 为高时，则可以对相应的位进行读写操作；反之，若对应 bit 为低则不能进行操作。例如：

- 若地址为 0x3FC (0b11\_1111\_1100)，则对 GPIO\_DATA[7:0]这 8bit 操作全部有效。
- 若地址为 0x200 (0b10\_0000\_0000)，则仅对 GPIO\_DATA[7]的操作有效。

Offset Address: 0x000 ~ 0x3FC    Total Reset Value: 0x00

| Bits  | Access | Name      | Description                                                                  | Reset |
|-------|--------|-----------|------------------------------------------------------------------------------|-------|
| [7:0] | RW     | gpio_data | 当 GPIO 配置为输入模式时，为 GPIO 输入数据；当 GPIO 配置为输出模式时，为输出数据。各比特均可独立控制。与 GPIO_DIR 配合使用。 | 0x00  |

## GPIO\_DIR

GPIO\_DIR 为 GPIO 方向控制寄存器。用来配置 GPIO 管脚方向。

Offset Address: 400    Total Reset Value: 0x00

| Bits  | Access | Name     | Description                                                          | Reset |
|-------|--------|----------|----------------------------------------------------------------------|-------|
| [7:0] | RW     | gpio_dir | GPIO 方向控制寄存器。bit[7:0]分别对应 GPIO_DATA[7:0]，各比特可独立控制。<br>0：输入；<br>1：输出。 | 0x00  |

## GPIO\_IS

GPIO\_IS 为 GPIO 中断触发寄存器。用来配置 GPIO 管脚触发电平方式。

Offset Address: 404    Total Reset Value: 0x00

| Bits  | Access | Name    | Description                                                                       | Reset |
|-------|--------|---------|-----------------------------------------------------------------------------------|-------|
| [7:0] | RW     | gpio_is | GPIO 中断触发控制寄存器, bit[7:0]分别对应 GPIO_DATA[7:0], 各比特独立控制。<br>0: 边沿触发中断;<br>1: 电平触发中断。 | 0x00  |

## GPIO\_IBE

GPIO\_IBE 为 GPIO 双沿触发中断寄存器。用来配置 GPIO 管脚沿触发方式。

Offset Address: 408 Total Reset Value: 0x00

| Bits  | Access | Name     | Description                                                                                                      | Reset |
|-------|--------|----------|------------------------------------------------------------------------------------------------------------------|-------|
| [7:0] | RW     | gpio_ibe | GPIO 中断沿触发控制寄存器, bit[7:0]分别对应 GPIO_DATA[7:0], 各比特独立控制。<br>0: 单边沿触发中断, 具体是上升沿还是下降沿触发由 GPIO_IIEV 控制;<br>1: 双边触发中断。 | 0x00  |

## GPIO\_IIEV

GPIO\_IIEV 为 GPIO 触发中断条件寄存器。用来配置 GPIO 管脚触发中断条件。

Offset Address: 40C Total Reset Value: 0x00

| Bits  | Access | Name      | Description                                                                                 | Reset |
|-------|--------|-----------|---------------------------------------------------------------------------------------------|-------|
| [7:0] | RW     | gpio_iiev | GPIO 触发中断条件寄存器, bit[7:0]分别对应 GPIO_DATA[7:0], 各比特独立控制。<br>0: 下降沿或低电平触发中断;<br>1: 上升沿或高电平触发中断。 | 0x00  |

## GPIO\_IIIE

GPIO\_IIIE 为 GPIO 中断屏蔽寄存器。用来屏蔽 GPIO 管脚中断。

Offset Address: 410 Total Reset Value: 0x00

| Bits  | Access | Name    | Description                                                                  | Reset |
|-------|--------|---------|------------------------------------------------------------------------------|-------|
| [7:0] | RW     | gpio_ie | GPIO 中断屏蔽寄存器, bit[7:0]分别对应 GPIO_DATA[7:0], 各比特独立控制。<br>0: 屏蔽中断;<br>1: 不屏蔽中断。 | 0x00  |

## GPIO\_RIS

GPIO\_RIS 为 GPIO 原始中断状态寄存器。用来查询 GPIO 管脚原始中断状态。

Offset Address: 414 Total Reset Value: 0x00

| Bits  | Access | Name     | Description                                                                                            | Reset |
|-------|--------|----------|--------------------------------------------------------------------------------------------------------|-------|
| [7:0] | RO     | gpio_ris | GPIO 原始中断寄存器, bit[7:0]分别对应 GPIO_DATA[7:0], 指示未屏蔽的中断状态。该状态不受 GPIO_IE 寄存器屏蔽控制。<br>0: 未发生中断;<br>1: 已发生中断。 | 0x00  |

## GPIO\_MIS

GPIO\_MIS 为 GPIO 屏蔽状态中断寄存器。用来查询 GPIO 管脚屏蔽后的中断状态。

Offset Address: 418 Total Reset Value: 0x00

| Bits  | Access | Name     | Description                                                                                            | Reset |
|-------|--------|----------|--------------------------------------------------------------------------------------------------------|-------|
| [7:0] | RO     | gpio_mis | GPIO 屏蔽后中断寄存器, bit[7:0]分别对应 GPIO_DATA [7:0], 指示经屏蔽后的中断状态。该状态受 GPIO_IE 寄存器屏蔽控制。<br>0: 中断无效;<br>1: 中断有效。 | 0x00  |

## GPIO\_IC

GPIO\_IC 为 GPIO 中断清除寄存器。用来清除 GPIO 管脚产生的中断，同时清除 GPIO\_RIS 寄存器和 GPIO\_MIS 寄存器。

Offset Address: 41C    Total Reset Value: 0x00

| Bits  | Access | Name    | Description                                                              | Reset |
|-------|--------|---------|--------------------------------------------------------------------------|-------|
| [7:0] | WC     | gpio_ic | GPIO 中断清除寄存器，bit[7:0]分别对应 GPIO_DATA [7:0]，各比特可独立控制。<br>0：无影响；<br>1：清除中断。 | 0x00  |

## 2.6 USB DRD

### 2.6.1 概述

芯片支持 1 个 USB2.0 DRD (Dule Role Device)接口，支持工作在 USB2.0 Host 模式或 USB2.0 Device 模式，不支持动态切换。

- USB2.0 Host 支持 USB2.0 协议规定的 480Mbps/12Mbps/1.5Mbps 三种传输速率；支持控制传输、批量传输、同步传输、中断传输四种基本的传输方式；内部集成一个 Root Hub，最多同时支持 64 个设备。
- USB2.0 Device 支持 480Mbps/12Mbps 两种传输速率，支持控制传输、批量传输、同步传输、中断传输四种基本的传输方式，支持控制端点在内的 7 个端点号。

USB DRD 主要完成：

- 协议的控制和处理
- 数据的解析和打包
- 传输信号的编码和解码
- 为驱动程序提供中断向量接口

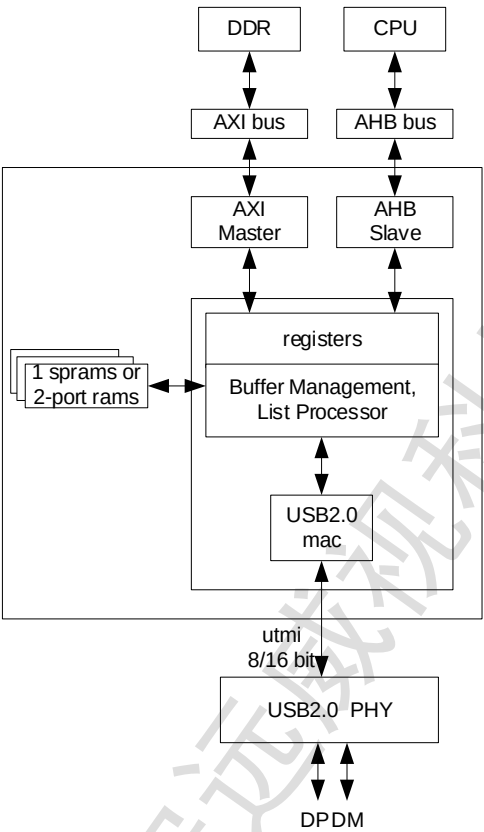


## 2.6.2 功能描述

### 逻辑框图

USB 2.0 DRD 逻辑框图如图 2-32 所示。

图2-32 USB 2.0 DRD 逻辑框图



UTMI: USB2.0 Transceiver Macrocell Interface

### 功能特点

USB2.0 DRD 具有以下功能特点：

- 支持 USB2.0 协议，向下兼容 USB 1.1 协议。
- 支持工作在 Host 模式或 Device 模式，不支持动态切换。
- Host 模式支持支持 Low、FULL、High speed 模式、480Mbps、12Mbps、1.5Mbps 三种传输速率。
- Devcie 模式支持 480Mbps、12Mbps 两种传输速率，支持控制端点在内的 7 个端点号。

- 支持 USB 2.0 Suspend、Resume、Remote wakeup 低功耗特性。
- 支持控制传输、批量传输、同步传输、中断传输四种基本的传输方式。
- 支持 DMA 数据存取。
- 通过外扩 USB Hub，最多可连接 64 个设备。

## 工作原理

USB2.0 工作于 host 模式时，支持以下 4 种标准的传输方式：

- Control Transfer（控制传输）

主要用于 USB Host 与 USB Device 端点 0 之间的传输。控制传输是双向传输，数据量通常较小，可以传输 8byte、16byte、32byte 或 64byte 的数据，取决于设备和传输速率。
- Bulk Transfer（批量传输）

主要用于没有带宽和间隔时间要求的设备的数据传输，这类设备适合于传输非常慢和大量的被延迟的数据，可以等到所有其它类型的数据传输完成之后再传输。它的特点是以错误检测和重传的方式保证 USB Host 与 USB Device 之间无差错的数据传输。
- Isochronous Transfer（同步传输）

主要用于时间严格并具有较强的容错性的流数据传输，或者用于数据传输速率恒定的即时应用中。同步传输提供了确定的带宽和间隔时间。
- Interrupt Transfer（中断传输）

主要用于少量、分散、不可预测的数据的传输。中断传输方式下，定时查询设备是否有中断数据要发送。设备的端点模式器的结构决定了它的查询频率为 1ms ~ 255ms。典型的中断方式传输是单向的，并且对于 USB Host 来说只有输入的方式。

### 2.6.3 工作方式

#### USB2.0 DRD 时钟复位

在初始化控制器之前，对时钟复位寄存器做相应的配置。

步骤如下：

步骤 1 向 PERI\_CRG80[3]写 0，撤离 USB2.0 控制器的 VCC 软复位；

- 步骤 2 向系统控制寄存器 MISC\_CTRL11[0]写 1，打开 USB2.0 PHY 内部的 PLL 使能。同时根据当前 USB2.0 PHY 工作模式选择 utmi 接口 data\_bus 是 16bit 还是 8bit，16bit 模式下向 MISC\_CTRL11[1]写 1，8bit 则写 0。
- 步骤 3 向 PERI\_CRG80[1]写入 0x0，撤离 USB2.0 PHY 的数字复位；向 PERI\_CRG80[0]写入 0x0，撤离 USB2.0 PHY 的配置接口复位。

----结束

Host/Device 工作模式切换

切换操作如下：

- 向 PERI\_USB2\_GCTL [prtcapdir] 写入 2'b01，DRD 控制器将工作于 Host 模式。
- 向 PERI\_USB2\_GCTL [prtcapdir] 写入 2'b10，DRD 控制器将工作于 Device 模式。

须知

Host/Device 工作模式切换只支持静态切换，不支持动态切换。

2.6.4 USB2.0 PHY 寄存器概览

USB2 PHY 寄存器映射表如表 2-17 所示。

表2-17 USB PHY 寄存器映射表

(基地址是 0x11FF\_0000)

| 偏移地址   | 名称       | 描述             |
|--------|----------|----------------|
| 0x0000 | FC_REG00 | USB2 PHY 寄存器 0 |
| 0x0004 | FC_REG01 | USB2 PHY 寄存器 1 |
| 0x0008 | FC_REG02 | USB2 PHY 寄存器 2 |
| 0x0010 | FC_REG03 | USB2 PHY 寄存器 3 |
| 0x0014 | FC_REG04 | USB2 PHY 寄存器 4 |

| 偏移地址        | 名称          | 描述                 |
|-------------|-------------|--------------------|
| 0x0018      | FC_REG05    | USB2 PHY 寄存器 5     |
| 0x001c      | FC_REG06    | USB2 PHY 寄存器 6     |
| 0x0020      | FC_REG07    | USB2 PHY 寄存器 7     |
| 0x0024      | FC_REG08    | USB2 PHY 寄存器 8     |
| 0x0028      | FC_REG09    | USB2 PHY 寄存器 9     |
| 0x002c      | FC_REG10    | USB2 PHY 寄存器 10    |
| 0x0030      | FC_REG11    | USB2 PHY 寄存器 11    |
| 0x0034      | FC_REG12    | USB2 PHY 寄存器 12    |
| 0x0038      | FC_REG13    | USB2 PHY 寄存器 13    |
| 0x003c      | FC_REG14    | USB2 PHY 寄存器 14    |
| 0x0040      | FC_REG15    | USB2 PHY 寄存器 15    |
| 0x0044      | FC_REG16    | USB2 PHY 寄存器 16    |
| 0x0048      | FC_REG17    | USB2 PHY 寄存器 17    |
| 0x004c~0x60 | FC_REG18~23 | USB2 PHY 寄存器 18~23 |
| 0x0064      | FC_REG24    | USB2 PHY 寄存器 24    |
| 0x0068      | FC_REG25    | USB2 PHY 寄存器 25    |
| 0x006c      | FC_REG26    | USB2 PHY 寄存器 26    |
| 0x0070      | FC_REG27    | USB2 PHY 寄存器 27    |
| 0x0074      | FC_REG28    | USB2 PHY 寄存器 28    |
| 0x0078      | FC_REG29    | USB2 PHY 寄存器 29    |
| 0x007c      | FC_REG30    | USB2 PHY 寄存器 30    |
| 0x0080      | FC_REG31    | USB2 PHY 寄存器 31    |

## 2.6.5 USB2.0 PHY 寄存器描述

### FC\_REG00

Offset Address: 0x0000 Total Reset Value: 0x08

| Bits   | Access | Name            | Description                                                                                                    | Reset |
|--------|--------|-----------------|----------------------------------------------------------------------------------------------------------------|-------|
| [31:8] | -      | Reserved        | 保留。                                                                                                            | 0x0   |
| [7]    | RW     | PHY_MODE_INTERR | PHY 内部模式                                                                                                       | 0x0   |
| [6]    | RW     | PHY_MODE_SEL    | PHY 模式选择                                                                                                       | 0x0   |
| [5]    | RW     | CLK_SEL         | CLK_60_30 输出源选择                                                                                                | 0x0   |
| [4]    | RW     | COUNTER_SEL     | SIE 输入采样使能                                                                                                     | 0x0   |
| [3]    | RW     | RXGAP_FIX_EN    | RXGAP 固定使能                                                                                                     | 0x1   |
| [2:0]  | RW     | CLK_MODE        | 时钟门控。CLK_MODE[2]表示 TX 的时钟门控。1: 正常功耗模式, 0: 低功耗模式。<br>CLK_MODE[1:0]表示 RX 的时钟门控。1x: 正常功耗模式, 00: 低功耗模式。01: 超低功耗模式。 | 0x0   |

### FC\_REG01

Offset Address: 0x0004 Total Reset Value: 0x32

| Bits   | Access | Name              | Description                                                                       | Reset |
|--------|--------|-------------------|-----------------------------------------------------------------------------------|-------|
| [31:8] | -      | Reserved          | 保留。                                                                               | 0x0   |
| [7]    | RW     | SHIFT_COUNT16_SEL | 移位计数选择                                                                            | 0x0   |
| [6]    | RW     | INV_CLK_UTMI      | CLK_60_30 时钟反转。1: 反转, 0: 不反转                                                      | 0x0   |
| [5]    | RW     | LS_PAR_EN_BYPASS  | LS_PAR_EN 连接控制 bypass 信号。1: LS_PAR_EN 通过 r 寄存器设置。0: LS_PAR_EN 通过 OTG_SUSPENDM 控制。 | 0x1   |
| [4]    | RW     | LS_KPALV_EN       | LS 模式保持有效使能                                                                       | 0x1   |

| Bits | Access | Name          | Description | Reset |
|------|--------|---------------|-------------|-------|
| [3]  | RW     | PRE_HPHY_LSIE | 关闭前导使能      | 0x0   |
| [2]  | RW     | DET_FSEDP_EN  | FS EOP 检测使能 | 0x0   |
| [1]  | RW     | LS_PAR_EN     | LS 模式下并行使能  | 0x1   |

## FC\_REG02

Offset Address: 0x0008    Total Reset Value: 0xc3

| Bits   | Access | Name              | Description | Reset |
|--------|--------|-------------------|-------------|-------|
| [31:8] | -      | Reserved          | 保留。         | 0x0   |
| [7]    | RW     | Direct_dcp        | 保留。         | 0x1   |
| [6]    | RW     | Aca_dock_n        | 保留。         | 0x1   |
| [5]    | RW     | Dpt_connect_force | 保留。         | 0x0   |
| [4]    | RW     | Dpt_charger_force | 保留。         | 0x0   |
| [3]    | RW     | Upt_connect_force | 保留。         | 0x0   |
| [2]    | RW     | Upt_charger_force | 保留。         | 0x0   |
| [1:0]  | RW     | Charge_en         | 保留。         | 0x3   |

## FC\_REG03

Offset Address: 0x000c    Total Reset Value: 0x07

| Bits   | Access | Name              | Description | Reset |
|--------|--------|-------------------|-------------|-------|
| [31:8] | -      | Reserved          | 保留。         | 0x0   |
| [7]    | RW     | Rx_byte_count_sel | 数据输出相位选择。   | 0x0   |
| [6]    | RW     | DM_PULLDOWN_sel   | 保留。         | 0x0   |
| [5]    | RW     | DP_PULLDOWN_sel   | 保留。         | 0x0   |
| [4]    | RW     | Upt_idle_j        | 保留。         | 0x0   |

| Bits | Access | Name             | Description | Reset |
|------|--------|------------------|-------------|-------|
| [3]  | RW     | Charge_en_sel    | 保留。         | 0x0   |
| [2]  | RW     | Cdp_det_dp_first | 保留。         | 0x1   |
| [1]  | RW     | Dcd_timeout_en   | 保留。         | 0x1   |
| [0]  | RW     | lic_byte_done    | 保留。         | 0x1   |

## FC\_REG04

Offset Address: 0x0010    Total Reset Value: 0x19

| Bits   | Access | Name                 | Description                                                                                                             | Reset |
|--------|--------|----------------------|-------------------------------------------------------------------------------------------------------------------------|-------|
| [31:8] | -      | Reserved             | 保留。                                                                                                                     | 0x0   |
| [7]    | RW     | Vref_pd18_sel        | 内部 VREF_pd18 选择信号。<br>1: PLL_PD 通过 PLL_EN SUPENDM 控制。<br>0: PLL_PD 通过 PLL_EN 控制。                                        | 0x0   |
| [6]    | RW     | Pll_pd_sel           | 内部 PLL_PD 选择信号。1: PLL_PD 通过 PLL_EN SUPENDM 控制。<br>0: PLL_PD 通过 PLL_EN 控制。                                               | 0x0   |
| [5:4]  | RW     | Clk_480_mux          | CLK_480 输出时间选择。10: clk_480 在 pll lock 后有效。<br>01: clk_480 在 pll lock 后延迟一段 delay 值后有效。<br>00: clk_480 在 pll lock 过程中有效。 | 0x1   |
| [3:1]  | RW     | Hs_tx2rx_dly_cnt_sel | 总线反向延迟计数器选择。                                                                                                            | 0x4   |
| [0]    | RW     | Cdp_charge_valid_sel | 保留                                                                                                                      | 0x1   |

## FC\_REG05

Offset Address: 0x0014    Total Reset Value: 0x00

| Bits   | Access | Name                 | Description                                                                                                             | Reset |
|--------|--------|----------------------|-------------------------------------------------------------------------------------------------------------------------|-------|
| [31:8] | -      | Reserved             | 保留。                                                                                                                     | 0x0   |
| [7]    | RW     | Vref_pd18_sel        | 内部 VREF_pd18 选择信号。<br>1: PLL_PD 通过 PLL_EN SUPENDM 控制。<br>0: PLL_PD 通过 PLL_EN 控制。                                        | 0x0   |
| [6]    | RW     | Pll_pd_sel           | 内部 PLL_PD 选择信号。1: PLL_PD 通过 PLL_EN SUPENDM 控制。<br>0: PLL_PD 通过 PLL_EN 控制。                                               | 0x0   |
| [5:4]  | RW     | Clk_480_mux          | CLK_480 输出时间选择。10: clk_480 在 pll lock 后有效。<br>01: clk_480 在 pll lock 后延迟一段 delay 值后有效。<br>00: clk_480 在 pll lock 过程中有效。 | 0x1   |
| [3:1]  | RW     | Hs_tx2rx_dly_cnt_sel | 总线反向延迟计数器选择。                                                                                                            | 0x4   |
| [0]    | RW     | Cdp_charge_valid_sel | 保留                                                                                                                      | 0x1   |

## FC\_REG06

Offset Address: 0x0018 Total Reset Value: 0x00

| Bits   | Access | Name     | Description | Reset |
|--------|--------|----------|-------------|-------|
| [31:8] | -      | Reserved | 保留。         | 0x0   |
| [7:0]  | -      | Reserved | 保留。         | 0x0   |

## FC\_REG07

Offset Address: 0x0018 Total Reset Value: 0x00

| Bits   | Access | Name     | Description   | Reset |
|--------|--------|----------|---------------|-------|
| [31:8] | -      | Reserved | 保留。           | 0x0   |
| [7:0]  | RW     | TEST_SEL | Phy DTO 输出选择。 | 0x0   |



## FC\_REG08

Offset Address: 0x0020    Total Reset Value: 0x00

| Bits   | Access | Name                | Description                                                                                                                                                                           | Reset |
|--------|--------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:8] | -      | Reserved            | 保留。                                                                                                                                                                                   | 0x0   |
| [7:6]  | RW     | Cfg_pll_adj         | Kvco 校准信号。01:1000Mhz/v。<br>00:780Mhz/v                                                                                                                                                | 0x0   |
| [5:3]  | RW     | Cfg_ivref_adj       | Bit[2]: bandgap 电压测试使能。<br>1: 使能。<br>0: 不使能。<br><br>Bit[1]: bandgap ibias 选择。<br>1: alone-ibias<br>0: self_ibias<br><br>Bit[0]: bypass bandgap 信号。<br>1: bandgap 常开。<br>0: normal 模式。 | 0x0   |
| [2]    | RW     | Cfg_bypas_swcal     | TX swing 校准功能 bypass 信号。<br>1: bypass 模式。<br>0: normal 模式。                                                                                                                            | 0x0   |
| [1]    | RW     | Cfg_bypas_rcal_comp | RREF 比较器 bypass 信号。<br>1: bypass 模式。<br>0: normal 模式。                                                                                                                                 | 0x0   |
| [0]    | RW     | Cfg_byps_rcal       | 校准功能 bypass 信号。<br>1: bypass 模式, 校准功能关闭。<br>0: normal 模式。                                                                                                                             | 0x0   |

## FC\_REG09

Offset Address: 0x0024    Total Reset Value: 0x00

| Bits   | Access | Name               | Description                                                                                                                                                                       | Reset |
|--------|--------|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:8] | -      | Reserved           | 保留。                                                                                                                                                                               | 0x0   |
| [7:5]  | RW     | Cfg_swcal_voff     | HS TX swing code。                                                                                                                                                                 | 0x0   |
| [4]    | RW     | Cfg_swcal_sel_voff | 校准 code 选择信号。<br>1: 选择 cfg_swcal_off[3:0]。<br>0: 选择校准 code。                                                                                                                       | 0x0   |
| [3:0]  | RW     | Cfg_pll_adj[5:2]   | Cfg_pll_adj [5:4]: LPF 校准寄存器。<br>11: 10kohm。<br>10: 15kohm。<br>01: 20kohm。<br>00: 25kohm。<br>Cfg_pll_adj [3:2]: charge pump 电流校准信号。11: 20uA。<br>10: 15uA。<br>01: 10uA。<br>00: 5uA | 0x0   |

## FC\_REG10

Offset Address: 0x0028    Total Reset Value: 0x0a

| Bits   | Access | Name           | Description                                                                  | Reset |
|--------|--------|----------------|------------------------------------------------------------------------------|-------|
| [31:8] | -      | Reserved       | 保留。                                                                          | 0x0   |
| [7]    | RW     | Cfg_sq_vref[0] | 噪声消除检测电压阈值 Cfg_sq_vref [3:0]:<br>1111: 152mv。<br>1000: 124mv。<br>0000: 92mv。 | 0x0   |

| Bits  | Access | Name                  | Description                                                      | Reset |
|-------|--------|-----------------------|------------------------------------------------------------------|-------|
| [6]   | RW     | Cfg_byps_swcal_select | TX swing 校准功能 bypass 选择信号。<br>1: cfg_byps_swcal。<br>0: SUSPENDM。 | 0x0   |
| [5:4] | RW     | Cfg_tx_hs_sr[1:0]     | HS TX slew rate 校准选择寄存器。                                         | 0x0   |
| [3]   | RW     | Reserved              | 保留                                                               | 0x1   |
| [2]   | RW     | Cfg_hs_rcv_sel_phase  | HS 接收器采样时钟相位选择。                                                  | 0x0   |
| [1]   | RW     | Reserved              | 保留                                                               | 0x1   |
| [0]   | RW     | Cfg_swcal_voff[3]     | HS TX swing code 寄存器。                                            | 0x0   |

## FC\_REG11

Offset Address: 0x002c    Total Reset Value: 0x44

| Bits   | Access | Name               | Description                                                                             | Reset |
|--------|--------|--------------------|-----------------------------------------------------------------------------------------|-------|
| [31:8] | -      | Reserved           | 保留。                                                                                     | 0x0   |
| [7]    | RW     | Cfg_tx_clk_polar   | HS TX 时钟相位极性选择。<br>1: 极性模式。<br>0: 正常模式。                                                 | 0x0   |
| [6:3]  | RW     | Cfg_host_vref[3:0] | Host 断开连接检测阈值电压。<br>Cfg_host_vref[3:0]:<br>1111: 640mv。<br>1000: 570mv。<br>0000: 490mv。 | 0x8   |
| [2:0]  | RW     | Cfg_sq_vref[3:1]   | 噪声消除检测电压阈值 Cfg_sq_vref<br>[3:0]:<br>1111: 152mv。<br>1000: 124mv。<br>0000: 92mv。         | 0x4   |

## FC\_REG12

Offset Address: 0x0030 Total Reset Value: 0xe1

| Bits   | Access | Name               | Description                                                                           | Reset |
|--------|--------|--------------------|---------------------------------------------------------------------------------------|-------|
| [31:8] | -      | Reserved           | 保留。                                                                                   | 0x0   |
| [7:5]  | RW     | Cfg_rcal_voff[2:0] | 端接电阻 code。                                                                            | 0x7   |
| [4]    | RW     | Cfg_rcal_sel_voff  | 校准 code 选择信号。                                                                         | 0x0   |
| [3:2]  | RW     | Cfg_rcal_rb_dly    | 复位数字部分 Delay time 调整信号。<br>11: 343.2us。<br>10: 171.6us。<br>01: 42.5us。<br>00: 85.8us。 | 0x0   |
| [1]    | RW     | Cfg_byps_dis_con   | RREF 断开检测 bypass 功能。<br>1: bypass 模式, RREF 断开检测功能不使能。<br>0: 正常模式。                     | 0x0   |
| [0]    | RW     | Reserved           | 保留                                                                                    | 0x1   |

## FC\_REG13

Offset Address: 0x0034 Total Reset Value: 0x80

| Bits   | Access | Name                 | Description                                                                     | Reset |
|--------|--------|----------------------|---------------------------------------------------------------------------------|-------|
| [31:8] | -      | Reserved             | 保留。                                                                             | 0x0   |
| [7]    | RW     | Cfg_real_delay[0]    | En_rcal 的 delay time 选择。<br>11: 700us。<br>10: 340us。<br>01: 170us。<br>00: 85us。 | 0x1   |
| [6]    | RW     | Cfg_bypas_vdd_detect | Bypass VDDA 检测。<br>1: bypass。                                                   | 0x0   |

| Bits  | Access | Name                 | Description                                                                                           | Reset |
|-------|--------|----------------------|-------------------------------------------------------------------------------------------------------|-------|
|       |        |                      | 0: normal。                                                                                            |       |
| [5:4] | RW     | Cfg_hs_test_sel[1:0] | Cp 测试选择信号。<br>11: 480Mbps 时钟数据测试。<br>10: HS low level 测试。<br>01: HS high level 测试。<br>00: 45ohm 端接测试。 | 0x0   |
| [3]   | RW     | Cfg_test_mode        | CP 测试模式使能信号。<br>1: 使能。<br>0: 不使能。                                                                     | 0x0   |
| [2:1] | RW     | Cfg_rref[1:0]        | 参考电阻选择。<br>11: 2Kohm。<br>10: 1kohm。<br>01: 390ohm。<br>00: 200ohm,                                     | 0x0   |
| [0]   | RW     | Cfg_rcal_voff[3]     | 端接电阻 code。                                                                                            | 0x0   |

## FC\_REG14

Offset Address: 0x0038 Total Reset Value: 0xb0

| Bits   | Access | Name            | Description                                                          | Reset |
|--------|--------|-----------------|----------------------------------------------------------------------|-------|
| [31:8] | -      | Reserved        | 保留。                                                                  | 0x0   |
| [7]    | RW     | Reserved        | 保留。                                                                  | 0x1   |
| [6:4]  | RW     | Cfg_charge_vref | Vdpsrc 和 Vdmsrc 电压校准信号。<br>111: 720mv。<br>011: 600mv。<br>000: 510mv。 | 0x3   |
| [3]    | RW     | Cfg_byps_link   | Idp_src link 控制 bypass 信号。<br>1: idp_src 和 idm_src 下电。               | 0x0   |

| Bits | Access | Name              | Description                                                                     | Reset |
|------|--------|-------------------|---------------------------------------------------------------------------------|-------|
|      |        |                   | 0: vdp_src 和 idm_src 一起控制。                                                      |       |
| [2]  | RW     | Cfg_byps_lgc      | 逻辑检测 dp/dm bypass 信号。<br>1: bypass 模式。<br>0: normal 模式。                         | 0x0   |
| [1]  | RW     | Cfg_byps_charge   | Charge bypass 信号。<br>1: bypass 模式。<br>0: normal 模式。                             | 0x0   |
| [0]  | RW     | Cfg_rcal_delay[1] | En_rcal 的 delay time 选择。<br>11: 700us。<br>10: 340us。<br>01: 170us。<br>00: 85us。 | 0x0   |

## FC\_REG15

Offset Address: 0x0040 Total Reset Value: 0x05

| Bits   | Access | Name               | Description                                                                                      | Reset |
|--------|--------|--------------------|--------------------------------------------------------------------------------------------------|-------|
| [31:8] | -      | Reserved           | 保留。                                                                                              | 0x0   |
| [7:5]  | RW     | Cfg_hs_pw_sel[2:0] | 减少输出 swing 的超级省电选择信号。<br>100: 500mv。<br>011: 100mv。<br>010: 200mv。<br>001: 300mv。<br>000: 400mv。 | 0x0   |
| [4]    | RW     | Cfg_hsr_x_test     | HS RX 功能测试使能。<br>1: 使能。<br>0: 不使能。                                                               | 0x0   |

| Bits  | Access | Name        | Description                                                      | Reset |
|-------|--------|-------------|------------------------------------------------------------------|-------|
| [3:2] | RW     | Cfg_idp_src | Idp_src 校准信号。<br>11: 20uA。<br>10: 15uA。<br>01: 10uA。<br>00: 5uA。 | 0x1   |
| [1:0] | RW     | Reserved    | 保留。                                                              | 0x1   |

## FC\_REG16

Offset Address: 0x0044      Total Reset Value: 0x00

| Bits   | Access | Name                      | Description                                                                                   | Reset |
|--------|--------|---------------------------|-----------------------------------------------------------------------------------------------|-------|
| [31:8] | -      | Reserved                  | 保留。                                                                                           | 0x0   |
| [7]    | RW     | Cfg_adj_crossover_fs[0]   | FS 眼图串扰电压校准。<br>11: 1.04v。<br>10: 1.38v。<br>01: 2.367v。<br>00: 1.72v。                         | 0x0   |
| [6:4]  | RW     | Cfg_vref_adj[2:0]         | Swing 校准信号。110:440mv。<br>101:500mv。100:480mv。<br>011:380mv。010:360mv。<br>001:420mv。000:400mv。 | 0x0   |
| [3:2]  | RW     | Cfg_adj_crossover_ls[1:0] | LS 眼图串扰电压校准。<br>11: 1.91v。<br>10: 2.06v。<br>01: 1.22v。<br>00: 1.65v。                          | 0x0   |
| [1:0]  | RW     | Cfg_vbus_adj[1:0]         | VBUS_VALID 电压阈值校准。<br>11: 4.45v。<br>10: 4.56v。                                                | 0x0   |

| Bits | Access | Name | Description             | Reset |
|------|--------|------|-------------------------|-------|
|      |        |      | 01: 4.76v。<br>00: 4.66v |       |

## FC\_REG17

Offset Address: 0x0048      Total Reset Value: 0x00

| Bits   | Access | Name                    | Description                                                                 | Reset |
|--------|--------|-------------------------|-----------------------------------------------------------------------------|-------|
| [31:8] | -      | Reserved                | 保留。                                                                         | 0x0   |
| [7:3]  | RW     | Reserved                | 保留。                                                                         | 0x0   |
| [2:1]  | RW     | Tx_ibias_adj[1:0]       | HS TX pre 驱动 bias 电流校准。<br>11: 55uA。<br>10: 45uA。<br>01: 35uA。<br>00: 25uA。 | 0x0   |
| [0]    | RW     | Cfg_adj_crossover_fs[1] | FS 眼图串扰电压校准。<br>11: 1.04v。<br>10: 1.38v。<br>01: 2.367v。<br>00: 1.72v。       | 0x0   |

## FC\_REG18~23

Offset Address: 0x004c~0x0060      Total Reset Value: 0x00

| Bits   | Access | Name     | Description | Reset |
|--------|--------|----------|-------------|-------|
| [31:0] | -      | Reserved | 保留。         | 0x0   |

## FC\_REG24

Offset Address: 0x0064      Total Reset Value: 0x28



| Bits   | Access | Name            | Description                                                                              | Reset |
|--------|--------|-----------------|------------------------------------------------------------------------------------------|-------|
| [31:7] | -      | Reserved        | 保留。                                                                                      | 0x0   |
| [6]    | RW     | Short_5v        | 无                                                                                        | 0x0   |
| [5]    | RW     | Short_5v_enable | 无                                                                                        | 0x1   |
| [4]    | RW     | Uni_bidi_i      | 无                                                                                        | 0x0   |
| [3]    | RW     | Usable_en       | 比较开始延迟时间选择                                                                               | 0x1   |
| [2]    | RW     | Clk480_sel      | Clk_480 时钟输出选择。<br>1: 在 pll lock 后 clk_480 有效。<br>0: 在 pll_lock 后延迟一段时间<br>clk_480 输出有效。 | 0x0   |
| [1:0]  | RW     | Cnt_num         | 3ms 计数器选择。<br>11: 2.72ms。<br>10: 1.36ms。<br>01: 682us。<br>00: 392us。                     | 0x0   |

## FC\_REG25

Offset Address: 0x0068      Total Reset Value: 0xf0

| Bits   | Access | Name     | Description | Reset |
|--------|--------|----------|-------------|-------|
| [31:0] | -      | Reserved | 保留。         | 0xf0  |

## FC\_REG26

Offset Address: 0x006c      Total Reset Value: 0xf0

| Bits   | Access | Name     | Description | Reset |
|--------|--------|----------|-------------|-------|
| [31:0] | -      | Reserved | 保留。         | 0xf0  |

## FC\_REG27

Offset Address: 0x0070      Total Reset Value: 8'hxx

| Bits   | Access | Name           | Description            | Reset |
|--------|--------|----------------|------------------------|-------|
| [31:8] | -      | Reserved       | 保留。                    | 0x0   |
| [7:5]  | RO     | Swcal_out[2:0] | HS TX swing 输出校准       | Xx    |
| [4]    | RO     | Rcal_trim_done | 端接校准结束标志。1: 完成。0: 未完成。 | Xx    |
| [3:0]  | RO     | Rcal_out       | 端接校准输出。                | Xx    |

## FC\_REG28

Offset Address: 0x0074      Total Reset Value: 8'hxx

| Bits   | Access | Name            | Description                         | Reset |
|--------|--------|-----------------|-------------------------------------|-------|
| [31:3] | -      | Reserved        | 保留。                                 | 0x0   |
| [2]    | RO     | Hsrx_test_o     | HS RX 功能测试输出。1: DP>DM。<br>0: DP<DM。 | Xx    |
| [1]    | RO     | Swcal_trim_done | HS TX swing 校准完成输出标志。1: 完成。0: 未完成。  | Xx    |
| [0]    | RO     | Swcal_out[3]    | HS TX swing 输出校准                    | Xx    |

## FC\_REG29

Offset Address: 0x0078      Total Reset Value: 8'hxx

| Bits   | Access | Name     | Description | Reset |
|--------|--------|----------|-------------|-------|
| [31:8] | -      | Reserved | 保留。         | 0x0   |
| [7:0]  | RO     | Reserved | 保留。         | Xx    |

## FC\_REG30

Offset Address: 0x007C      Total Reset Value: 8'hxx

| Bits   | Access | Name     | Description | Reset |
|--------|--------|----------|-------------|-------|
| [31:8] | -      | Reserved | 保留。         | 0x0   |
| [7:0]  | RO     | Reserved | 保留。         | Xx    |

## FC\_REG31

Offset Address: 0x0080      Total Reset Value: 8'hxx

| Bits   | Access | Name     | Description | Reset |
|--------|--------|----------|-------------|-------|
| [31:8] | -      | Reserved | 保留。         | 0x0   |
| [7:0]  | RO     | Reserved | 保留。         | Xx    |

## 2.6.6 USB 控制器寄存器概览

USB 寄存器概览如表 2-18 所示。

USB2.0 基址: 0x1003\_0000

表2-18 USB 寄存器概览

| 偏移地址   | 名称                  | 描述               |
|--------|---------------------|------------------|
| 0xC100 | PERI_USB2_GSBUSCFG0 | 全局 SOC 总线配置寄存器 0 |
| 0xC104 | PERI_USB2_GSBUSCFG1 | 全局 SOC 总线配置寄存器 1 |
| 0xC108 | PERI_USB2_GTXTHRCFG | 全局发送门限控制寄存器      |
| 0xC10C | PERI_USB2_GRXTHRCFG | 全局接收门限控制寄存器      |
| 0xC110 | PERI_USB2_GCTL      | 全局 core 控制寄存器    |
| 0xC118 | PERI_USB2_GSTS      | 全局状态寄存器          |
| 0xC11C | PERI_USB2_GUCTL1    | 全局用户控制寄存器 1      |
| 0xC124 | PERI_USB2_GGPIO     | 全局 GPIO 寄存器      |
| 0xC128 | PERI_USB2_GUID      | 全局用户 ID 寄存器      |
| 0xC12C | PERI_USB2_GUCTL     | 全局用户控制寄存器        |

| 偏移地址   | 名称                        | 描述                           |
|--------|---------------------------|------------------------------|
| 0xC130 | PERI_USB2_GBUSERRADDR_HI  | 全局总线错误地址高 32 位寄存器            |
| 0xC134 | PERI_USB2_GBUSERRADDR_LO  | 全局总线错误地址低 32 位寄存器            |
| 0xC138 | PERI_USB2_GPRTBIMAP_HI    | 超速端口-总线映射高 32 位寄存器           |
| 0xC13C | PERI_USB2_GPRTBIMAP_LO    | 超速端口-总线映射低 32 位寄存器           |
| 0xC180 | PERI_USB2_GPRTBIMAP_HS_HI | 高速端口-总线映射高 32 位寄存器           |
| 0xC184 | PERI_USB2_GPRTBIMAP_HS_LO | 高速端口-总线映射低 32 位寄存器           |
| 0xC188 | PERI_USB2_GPRTBIMAP_FS_HI | 全速端口-总线映射高 32 位寄存器           |
| 0xC18C | PERI_USB2_GPRTBIMAP_FS    | 全速端口-总线映射低 32 位寄存器           |
| 0xC200 | PERI_USB2_GUSB2PHYCFG     | 全局 USB2.0 PHY 配置寄存器          |
| 0xC304 | PERI_USB2_GTXFIFOSIZN     | 全局发送 FIFO 大小寄存器              |
| 0xC384 | PERI_USB2_GRXFIFOSIZN     | 全局接收 FIFO 大小寄存器              |
| 0xC410 | PERI_USB2_GEVNTADR_HI     | 全局 event buffer 的地址高 32 位寄存器 |
| 0xC414 | PERI_USB2_GEVNTADR_LO     | 全局 event buffer 的地址低 32 位寄存器 |
| 0xC418 | PERI_USB2_GEVNTSIZN       | 全局 event buffer 的大小寄存器       |
| 0xC41C | PERI_USB2_GEVNTCOUNTN     | 全局 event buffer 的计数寄存器       |

| 偏移地址   | 名称                      | 描述                            |
|--------|-------------------------|-------------------------------|
| 0xC610 | PERI_USB2_GTXFIFOPRIDEV | 外设的全局 TX FIFO DMA 优先寄存器       |
| 0xC618 | PERI_USB2_GTXFIFOPRIHST | host 的全局 TX FIFO DMA 优先寄存器    |
| 0xC61C | PERI_USB2_GRXFIFOPRIHST | host 的全局 RX FIFO DMA 优先寄存器    |
| 0xC620 | PERI_USB2_GFIFOPRIDBC   | host 的全局 Debug 性能时 DMA 优先寄存器  |
| 0xC624 | PERI_USB2_GDMAHLRATIO   | host 的全局 FIFO DMA 高、低优先权比例寄存器 |
| 0xC630 | PERI_USB2_GFLADJ        | GFLADJ 为全局帧长度调整寄存器            |
| 0xC700 | PERI_USB2_DCFG          | 外设配置寄存器                       |
| 0xC704 | PERI_USB2_DCTL          | 外设控制寄存器                       |
| 0xC708 | PERI_USB2_DEVTEN        | 外设事件使能寄存器                     |
| 0xC70C | PERI_USB2_DSTS          | 外设状态寄存器                       |
| 0xC710 | PERI_USB2_DGCMDPAR      | 外设类命令参数寄存器                    |
| 0xC714 | PERI_USB2_DGCMD         | 外设类命令寄存器                      |
| 0xC718 | PERI_USB2_DALEPENA      | 外设 USB 端点使能寄存器                |
| 0xC810 | PERI_USB2_DEPCMDPAR2N   | 外设端点命令参数寄存器 2                 |
| 0xC814 | PERI_USB2_DEPCMDPAR1N   | 外设端点命令参数寄存器 1                 |
| 0xC818 | PERI_USB2_DEPCMDPAR0N   | 外设端点命令参数寄存器 0                 |
| 0xC81C | PERI_USB2_DEPCMDN       | 外设物理端点命令寄存器                   |

## 2.6.7 USB 控制器寄存器描述

### PERI\_USB2\_GSBUSCFG0

PERI\_USB2\_GSBUSCFG0 为全局 SOC 总线配置寄存器 0。

Offset Address: 0xC100    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name           | Description                                                    | Reset |
|---------|--------|----------------|----------------------------------------------------------------|-------|
| [31:28] | RW     | datrdreqinfo   | AHB-prot/AXI-cache/OCP-ReqInfo 读数据的请求。                         | 0x0   |
| [27:24] | RW     | desrdreqinfo   | AHB-prot/AXI-cache/OCP-ReqInfo 读链表请求。                          | 0x0   |
| [23:20] | RW     | datwrreqinfo   | AHB-prot/AXI-cache/OCP-ReqInfo 写数据请求。                          | 0x0   |
| [19:16] | RW     | deswrreqinfo   | AHB-prot/AXI-cache/OCP-ReqInfo 写链表请求。                          | 0x0   |
| [15:12] | RW     | reserved       | 保留。                                                            | 0x0   |
| [11]    | RW     | datbigend      | 数据存取大小端选择。<br>0: 小端;<br>1: 大端。                                 | 0x0   |
| [10]    | RW     | descbigend     | 链表存取大小端选择。<br>0: 小端;<br>1: 大端。                                 | 0x0   |
| [9:8]   | -      | reserved       | 保留。                                                            | 0x0   |
| [7]     | RW     | incr256brstena | AHB master INCR 进行 128 beat burst 传输使能信号。<br>0: 不使能;<br>1: 使能。 | 0x0   |
| [6]     | RW     | incr128brstena | AHB master INCR 进行 128 beat burst 传输使能信号。<br>0: 不使能;           | 0x0   |

| Bits | Access | Name          | Description                                                   | Reset |
|------|--------|---------------|---------------------------------------------------------------|-------|
|      |        |               | 1: 使能。                                                        |       |
| [5]  | RW     | incr64brstena | AHB master INCR 进行 64 beat burst 传输使能信号。<br>0: 不使能;<br>1: 使能。 | 0x0   |
| [4]  | RW     | incr32brstena | AHB master INCR 进行 32 beat burst 传输使能信号。<br>0: 不使能;<br>1: 使能。 | 0x0   |
| [3]  | RW     | incr16brstena | AHB master INCR 进行 16 beat burst 传输使能信号。<br>0: 不使能;<br>1: 使能。 | 0x0   |
| [2]  | RW     | incr8brstena  | AHB master INCR 进行 8 beat burst 传输使能信号。<br>0: 不使能;<br>1: 使能。  | 0x0   |
| [1]  | RW     | incr4brstena  | AHB master INCR 进行 4 beat burst 传输使能信号。<br>0: 不使能;<br>1: 使能。  | 0x0   |
| [0]  | RW     | incrbrstena   | AHB master INCR 进行 1 beat burst 传输使能信号。<br>0: 不使能;<br>1: 使能。  | 0x0   |

## PERI\_USB2\_GSBUSCFG1

PERI\_USB2\_GSBUSCFG1 为全局 SOC 总线配置寄存器 1。

Offset Address: 0xC104 Total Reset Value: 0x0000\_0000

| Bits    | Access | Name           | Description                                                                                                     | Reset   |
|---------|--------|----------------|-----------------------------------------------------------------------------------------------------------------|---------|
| [31:13] | -      | reserved       | 保留。                                                                                                             | 0x00000 |
| [12]    | RW     | en1kpage       | 1K Byte 边界选择。<br>0: 4K Byte 边界;<br>1: 1K Byte 边界。                                                               | 0x0     |
| [11:8]  | RW     | pipetranslimit | AXI master outstanding 请求数目。<br>0x0: 1 个请求;<br>0x1: 2 个请求;<br>0x2: 3 个请求;<br>0x3: 4 个请求;<br>...<br>0xF: 16 个请求。 | 0x0     |
| [7:0]   | -      | reserved       | 保留。                                                                                                             | 0x00    |

## PERI\_USB2\_GTXTHRCFG

PERI\_USB2\_GTXTHRCFG 为全局发送门限控制寄存器。

Offset Address: 0xC108 Total Reset Value: 0x0000\_0000

| Bits    | Access | Name           | Description                                                                                                               | Reset |
|---------|--------|----------------|---------------------------------------------------------------------------------------------------------------------------|-------|
| [31:30] | -      | reserved       | 保留。                                                                                                                       | 0x0   |
| [29]    | RW     | usbtxpktcntsel | USB TXFIFO 门限选择, 仅在 SuperSpeed 时有效。<br>0: USB 只在全部的包被读取到既定的 TXFIFO 后, 才开始进行传输;<br>1: USB 只在设定的包被读取到既定的 TXFIFO 后, 才开始进行传输。 | 0x0   |
| [28]    | -      | reserved       | 保留。                                                                                                                       | 0x0   |
| [27:24] | RW     | usbtxpktcnt    | TXFIFO 门限值设置, 有效值在 1-15 以                                                                                                 | 0x0   |



| Bits    | Access | Name              | Description                                                                                        | Reset  |
|---------|--------|-------------------|----------------------------------------------------------------------------------------------------|--------|
|         |        |                   | 内。                                                                                                 |        |
| [23:16] | RW     | usbmaxtxburstsize | 发送 burst 的最大值, 仅在 host 模式下 SuperSpeed 的 bulk, Isochronous 和 Interrupt 传输的 Out 端点时有效, 有效值在 1-16 之间。 | 0x00   |
| [15:0]  | -      | reserved          | 保留。                                                                                                | 0x0000 |

## PERI\_USB2\_GRXTHRCFG

PERI\_USB2\_GRXTHRCFG 为全局接收门限控制寄存器。

Offset Address: 0xC10C Total Reset Value: 0x0000\_0000

| Bits    | Access | Name              | Description                                                                                                               | Reset   |
|---------|--------|-------------------|---------------------------------------------------------------------------------------------------------------------------|---------|
| [31:30] | -      | reserved          | 保留。                                                                                                                       | 0x0     |
| [29]    | RW     | usbtxpktcntsel    | USB RXFIFO 门限选择, 仅在 SuperSpeed 时有效。<br>0: USB 只在全部的包被读取到既定的 RXFIFO 后, 才开始进行传输;<br>1: USB 只在设定的包被读取到既定的 RXFIFO 后, 才开始进行传输。 | 0x0     |
| [28]    | -      | reserved          | 保留。                                                                                                                       | 0x0     |
| [27:24] | RW     | usbtxpktcnt       | RXFIFO 门限值设置, 有效值在 1-15 以内。                                                                                               | 0x0     |
| [23:19] | RO     | usbmaxtxburstsize | 接收 burst 的最大值, 仅在 host 模式下 SuperSpeed 的 bulk, Isochronous 和 Interrupt 传输的 IN 端点时有效, 有效值在 1-16 之间。                         | 0x00    |
| [18:0]  | -      | reserved          | 保留。                                                                                                                       | 0x00000 |

## PERI\_USB2\_GCTL

PERI\_USB2\_GCTL 为全局 core 控制寄存器。

Offset Address: 0xC110    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name             | Description                                                                                                                                                                    | Reset  |
|---------|--------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:19] | RW     | pwrdsnscle       | Suspend_clk 设置, $GCTL[31:19] * 16K = Suspend\_clk$<br>说明: $32kHz < Suspend\_clk < 125MHz$ 。                                                                                    | 0x0000 |
| [18]    | RW     | masterfiltbypass | 滤波功能选择。<br>0: 当 DWC_USB2_EN_BUS_FILTERS 值为 1 时, 选择滤波功能;<br>1: 无论当 DWC_USB2_EN_BUS_FILTERS 值为多少, 关闭滤波功能。                                                                        | 0x0    |
| [17]    | RW     | bypssetaddr      | Device 模式下 SetAddress 命令选择。<br>0: Host 正常发送 SetAddress 命令给 Device;<br>1: Host 不发送 SetAddress 命令给 Device, Device 读取 DCFG[DevAddress] 比特位的值作为 Address 的值。<br>说明: 只在仿真时设置此 bit 位。 | 0x0    |
| [16]    | RW     | u2rstecn         | 超时连接选择。<br>0: 当超速连接失败时, 设备处于 HS 模式;<br>1: 当超速连接失败时, Devcie 将花费大于 3 个 cycle 的时间等待连接。<br>说明: 此比特位仅在 Device 模式下有效。                                                                | 0x0    |
| [15:14] | RW     | frmsclown        | SOF/USOF/ITP 时间间隔选择。对于 SS/HS 模式:<br>00: 125us;<br>01: 62.5us;<br>10: 31.25us;                                                                                                  | 0x0    |

| Bits    | Access | Name          | Description                                                                                                                                                                                                 | Reset |
|---------|--------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
|         |        |               | 11: 15.625us;<br>其他: 保留。<br>对于 FS 模式, 将上述值*8 即可。<br>当选择 xHCI Debug 模式时, 可配置仿真时 bulk in 和 bulkout 传输的 MaxPacketSize,<br>00: 1024 bytes;<br>01: 512 bytes;<br>10: 256 bytes;<br>11: 128bytes;<br>其他: 保留。      |       |
| [13:12] | RW     | prtcapdir     | 端口配置类型。<br>00: 保留;<br>01: Host 配置;<br>10: Device 配置;<br>11: 保留。                                                                                                                                             | 0x0   |
| [11]    | RW     | coresoftreset | Core 软复位选择。<br>0: 不进行软复位;<br>1: 对 core 软复位。<br>说明: 当对 core 进行软复位操作时, 将清除中断及初下列寄存器以外的所有中断:<br>-GCTL<br>-GUCTL<br>-GSTS<br>-GSNPSID<br>-GGPIO<br>-GUID<br>-GUSB2PHYCFGn<br>-DCFG<br>-DCTL<br>-DEVTEN<br>-DSTS | 0x0   |

| Bits  | Access | Name           | Description                                                                                                                                                                                                                            | Reset |
|-------|--------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [10]  | RW     | sofitpsync     | <p>0: UTMI/ULPI PHY 的第一个端口都将处于非挂起状态, 不管是否有其他的 SS 端口不处于 Rx.Detect, SS.Disable 和 U3 状态;</p> <p>1: UTMI/ULPI PHY 的第一个端口都将处于非挂起状态, 不管是否有其他的非 SS 端口不处于非挂起状态。</p> <p>说明: 此 bit 只在控制器工作于 Host 模式时有效。</p>                                      | 0x0   |
| [9]   | RW     | u1u2timerscale | <p>U1/U2 timer scaledown 选择。</p> <p>0: 不关闭;</p> <p>1: 如果 PERI_USB2_GCTL[5:4] (ScaleDown) = X1, 则关闭 U1/U2 的反应时间的 scaledown。</p>                                                                                                         | 0x0   |
| [8]   | RW     | debugattach    | <p>Debug Attach 信号</p> <p>当此位被置 1 时</p> <ul style="list-style-type: none"> <li>• 当 DCTL 寄存器里的 Ru/Stop 位被置位后, SS 控制器将直接进入 Polling link 状态而不需要检测远程设备的连接</li> <li>• Link LFPS polling 的超时时间有限</li> <li>• TS1 的 Polling 超时时间有限。</li> </ul> | 0x0   |
| [7:6] | RW     | ramcksel       | <p>RAM Clock 选择。</p> <p>00: bus clock;</p> <p>01: pipe clock;</p> <p>10: pipe/2 clock;</p> <p>11: reserved。</p> <p>说明: 当处于 host 模式时, 硬件将置此 2 位为 00, 即将 ram_clk 接 bus_clk, 因为当 SS port 处于 P3 状态时, pipe_clk 会被关闭, USB2.0 port 将不会工作。</p> | 0x0   |

| Bits  | Access | Name         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|-------|--------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [5:4] | RW     | scaledown    | <p>scale-down timing 选择。</p> <p>HS/FS/LS 模式下：</p> <p>00：关闭掉所有的 scale-down timing，使用实际的 timing 进行仿真；</p> <p>01：启用除以下功能外地所有的 scale-down timing：</p> <ul style="list-style-type: none"> <li>-speed enumeration</li> <li>-HNP/SRP</li> <li>-Host 模式的 suspend 和 resume；</li> </ul> <p>10：仅仅开启 device 模式时 suspend 和 resume 功能时的 scale-down timing；</p> <p>11：打开所有的 scale-down timing。</p> <p>SS 模式下</p> <p>HS/FS/LS 模式下</p> <p>00：关闭掉所有的 scale-down timing，使用实际的 timing 进行仿真；</p> <p>01：开启 SS 的 scale-down timing，包括：</p> <ul style="list-style-type: none"> <li>-减少 TxEq training sequences 到 8</li> <li>-减少 LFPS polling burst time 到 100ns</li> <li>-减少 LFPS warm reset receive 到 30us</li> </ul> <p>10：不发送 TxEq training sequences；</p> <p>11：打开所有的 scale-down timing。</p> | 0x0   |
| [3]   | RW     | dissscrumble | <p>关闭加扰功能。</p> <p>1：关闭；</p> <p>0：不关闭。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0x0   |
| [2]   | RW     | u2exit_lfps  | <p>U2 状态退出信号。</p> <p>0：link 将把 248ns 的 LFPS 信号当做有效的 U2 退出状态信号；</p> <p>1：link 层在检测到有效的 U2 退出信号前</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |

| Bits | Access | Name             | Description                                                                                                    | Reset |
|------|--------|------------------|----------------------------------------------------------------------------------------------------------------|-------|
|      |        |                  | 等待 8us。                                                                                                        |       |
| [1]  | RW     | gblhibernationen | 休眠使能。<br>0：关闭全局休眠功能，pmu 接收 D0->D3 或者 D3->D0 的状态切换，core 内部的状态将不会保存或者恢复；<br>1：打开全局休眠功能。                          | 0x0   |
| [0]  | RW     | dsblclkgtnng     | 内部 Clock Gating 选择。<br>0：选择内部 clock gating；<br>1：当 core 处于 lpm 模式时，关闭掉内部 clock gating。<br>说明：上电复位后可将此 bit 置为 1 | 0x0   |

## PERI\_USB2\_GSTS

PERI\_USB2\_GSTS 为全局状态寄存器。

Offset Address: 0xC118 Total Reset Value: 0x0000\_0000

| Bits    | Access | Name     | Description                                                                             | Reset |
|---------|--------|----------|-----------------------------------------------------------------------------------------|-------|
| [31:20] | RO     | cbelt    | 指示在 Host 模式下，所有接收到的 device BELT value 的最小值，以及 Set Latency Tolerance Value 命令设置的 BELT 值。 | 0x000 |
| [19:10] | -      | reserved | 保留。                                                                                     | 0x000 |
| [9]     | RO     | bc_ip    | 指示 BCEVT 寄存器中有一个与 BC 有关的中断等待处理。<br>0：无指示；<br>1：有指示。                                     | 0x0   |
| [8]     | RO     | adp_ip   | 指示 ADPEVT 寄存器中有一个与 ADP 有关的中断等待处理。<br>0：无指示；<br>1：有指示。                                   | 0x0   |

| Bits  | Access | Name          | Description                                                              | Reset |
|-------|--------|---------------|--------------------------------------------------------------------------|-------|
| [7]   | RO     | host_ip       | 指示 Host event queue 中有一个与 xHCI 有关的中断等待处理。<br>0: 无指示;<br>1: 有指示。          | 0x0   |
| [6]   | RO     | device_ip     | 指示 Device event queue 中有一个与 xHCI 有关的中断等待处理。<br>0: 无指示;<br>1: 有指示。        | 0x0   |
| [5]   | RO     | csrtimeout    | 指示软件访问寄存器的时间超出了 DWC_USB2_CSR_ACCESS_TIMEOUT 定义的时间。<br>0: 无指示;<br>1: 有指示。 | 0x0   |
| [4]   | RO     | buserraddrvld | 指示 GBUSERRADDR 寄存器是否有效并指出发生错误的首地址。<br>0: 无指示;<br>1: 有指示。                 | 0x0   |
| [3:2] | -      | reserved      | 保留。                                                                      | 0x0   |
| [1:0] | RO     | curmod        | 当前的工作模式:<br>00: Device 模式;<br>01: Host 模式;<br>其他: 保留。                    | 0x0   |

## PERI\_USB2\_GUCTL1

PERI\_USB2\_GUCTL1 为全局用户控制寄存器 1。

Offset Address: 0xC11C Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description | Reset      |
|--------|--------|----------|-------------|------------|
| [31:2] | -      | reserved | 保留。         | 0x00000000 |

| Bits | Access | Name              | Description                                                                                   | Reset |
|------|--------|-------------------|-----------------------------------------------------------------------------------------------|-------|
| [1]  | RW     | ovrld_l1_susp_com | 如果此位写 1, utmi_l1_suspend_com_n 信号将会被 utmi_sleep_n 替代。此位一般会在当 PHY 进入 L1 sleep 模式下停止提供时钟的情况下置起。 | 0x0   |
| [0]  | RW     | loa_filter_en     | 检测端口的关闭状态, 当此位置 1 时, 在端口关闭之前, controller 将发送三个连续的 cycle 检查端口的状态。                              | 0x0   |

## PERI\_USB2\_GGPIO

PERI\_USB2\_GGPIO 为全局 GPIO 寄存器。

Offset Address: 0xC124    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name | Description      | Reset  |
|---------|--------|------|------------------|--------|
| [31:16] | RW     | gpo  | 驱动 gp_o[15:0]的值。 | 0x0000 |
| [15:0]  | RO     | gpi  | 读取 gp_i[15:0]的值。 | 0x0000 |

## PERI\_USB2\_GUID

PERI\_USB2\_GUID 为全局用户 ID 寄存器。

Offset Address: 0xC128    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description                 | Reset      |
|--------|--------|----------|-----------------------------|------------|
| [31:0] | RO     | reserved | 用户信息, 包括:<br>系统版本;<br>硬件配置。 | 0x00000000 |

## PERI\_USB2\_GUCTL

PERI\_USB2\_GUCTL 为全局用户控制寄存器。

Offset Address: 0xC12C    Total Reset Value: 0x0000\_0000



| Bits    | Access | Name                    | Description                                                                                                                                                    | Reset |
|---------|--------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:22] | RW     | refclkper               | 参考时钟 ref_clk 用 ns 表示<br>例如: ref_clk=125MHz,则此处为<br>1/125MHz=8ns。                                                                                               | 0x000 |
| [21]    | RW     | noextrdl_               | SOF 包和第一个包之间的额外延时选择。<br>0: Host 在 SOF 包之后等待 2ms 再发送<br>第一个包;<br>1: Host 在 SOF 之后不延时直接发送第一<br>一个包。                                                              | 0x0   |
| [20:18] | -      | reserved                | 保留。                                                                                                                                                            | 0x0   |
| [17]    | RW     | sprscrtltransen         | 分散控制传输使能。<br>0: 禁止;<br>1: 使能。<br>一些 Device 对控制传输的响应很慢, 在 1<br>帧/微帧内进行多次传输的话会导致 Device<br>行为紊乱。当此位被置 1 时, host 控制器<br>将把一个控制传输分散在不同的帧或者微<br>帧中。                 | 0x0   |
| [16]    | RW     | resbwhseps              | 保留 85%的带宽给高速周期性端点, 只在<br>HOST 模式或者 DRD 模式下的 Host 操作<br>模式下有效。                                                                                                  | 0x0   |
| [15]    | RW     | cmdevaddr               | Device 地址的模式。<br>1: 根据每个设备分配地址命令递增;<br>0: 设备地址等于 Slot ID。                                                                                                      | 0x0   |
| [14]    | RW     | usbhstlnautoretrye<br>n | Host 输入传输自动重传使能。<br>0: 自动重传功能关闭;<br>如果 host 的输入传输发生错误, host 变自<br>动回复给 device 一个终结的 ACK(Retry =<br>1 and NumP = 0);<br>1: 自动重传功能开启。<br>当自动重传功能开启后, 如果 host 的输入 | 0x0   |

| Bits   | Access | Name         | Description                                                                                                                     | Reset |
|--------|--------|--------------|---------------------------------------------------------------------------------------------------------------------------------|-------|
|        |        |              | 传输发生错误, host 变自动回复给 device 一个不是终结的 ACK(Retry = 1 and NumP != 0)。                                                                |       |
| [13]   | RW     | enoverlapchk | LFPS 叠加信号检测使能。<br>0: 不检测 LFPS 叠加信号;<br>1: 检测 LFPS 叠加信号避免毛刺影响。                                                                   | 0x0   |
| [12]   | -      | reserved     | 保留。                                                                                                                             | 0x0   |
| [11]   | RW     | csr          | 在全速 BULKOUT 传输之间插入额外的延时。<br>0: 不插入;<br>1: 插入。                                                                                   | 0x0   |
| [10:9] | RW     | dtct         | Device 响应 Host 的 Timeout 粗略时间, 若此位置 0, 则 timeout 时间由 DTFT 定义, 若此位为非 0, 则:<br>00: 0us;<br>01: 500us;<br>10: 1.5ms;<br>11: 6.5ms。 | 0x0   |
| [8:0]  | RW     | dtft         | Device 响应 Host 的 Timeout 精确时间, DTCT 为 0 的时候有效,<br>$T=DTFT*256*8\text{ us}$ 。                                                    | 0x000 |

## PERI\_USB2\_GBUSERRADDR\_HI

PERI\_USB2\_GBUSERRADDR\_HI 为全局总线错误地址高 32 位寄存器。

Offset Address: 0xC130 Total Reset Value: 0x0000\_0000

| Bits   | Access | Name      | Description                                       | Reset      |
|--------|--------|-----------|---------------------------------------------------|------------|
| [31:0] | RO     | busaddrhi | 发生错误的高 32 位地址。<br>说明: 只在 GSTS.BusErrAddrVld 为 1 时 | 0x00000000 |

|  |  |  |                                     |  |
|--|--|--|-------------------------------------|--|
|  |  |  | 有效<br>只在复位时清 0，只支持 AHB 和 AXI 的总线配置。 |  |
|--|--|--|-------------------------------------|--|

## PERI\_USB2\_GBUSERRADDR\_LO

PERI\_USB2\_GBUSERRADDR\_LO 为全局总线错误地址低 32 位寄存器。

Offset Address: 0xC134 Total Reset Value: 0x0000\_0000

| Bits   | Access | Name      | Description                                                                          | Reset      |
|--------|--------|-----------|--------------------------------------------------------------------------------------|------------|
| [31:0] | RO     | busaddrlo | 发生错误的低 32 位地址。<br>说明：只在 GSTS.BusErrAddrVld 为 1 时有效。<br>只在复位时清 0，只支持 AHB 和 AXI 的总线配置。 | 0x00000000 |

## PERI\_USB2\_GPRTBIMAP\_HI

PERI\_USB2\_GPRTBIMAP\_HI 为超速端口-总线映射高 32 位寄存器。

Offset Address: 0xC138 Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description | Reset      |
|--------|--------|----------|-------------|------------|
| [31:0] | -      | reserved | 保留          | 0x00000000 |

## PERI\_USB2\_GPRTBIMAP\_LO

PERI\_USB2\_GPRTBIMAP\_LO 为超速端口-总线映射低 32 位寄存器。

Offset Address: 0xC13C Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description              | Reset      |
|--------|--------|----------|--------------------------|------------|
| [31:4] | -      | reserved | 保留。                      | 0x00000000 |
| [3:0]  | RW     | binumn   | 指示当前连接的 superspeed 的总线号。 | 0x0        |

## PERI\_USB2\_GPRTBIMAP\_HS\_HI

PERI\_USB2\_GPRTBIMAP\_HS\_HI 为高速端口-总线映射高 32 位寄存器。

Offset Address: 0xC180    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description | Reset      |
|--------|--------|----------|-------------|------------|
| [31:0] | -      | reserved | 保留。         | 0x00000000 |

## PERI\_USB2\_GPRTBIMAP\_HS\_LO

PERI\_USB2\_GPRTBIMAP\_HS\_LO 为高速端口-总线映射低 32 位寄存器。

Offset Address: 0xC184    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description             | Reset      |
|--------|--------|----------|-------------------------|------------|
| [31:4] | -      | reserved | 保留。                     | 0x00000000 |
| [3:0]  | RW     | binumn   | 指示当前连接的 highspeed 的总线号。 | 0x0        |

## PERI\_USB2\_GPRTBIMAP\_FS\_HI

PERI\_USB2\_GPRTBIMAP\_FS\_HI 为全速端口-总线映射高 32 位寄存器。

Offset Address: 0xC188    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description | Reset      |
|--------|--------|----------|-------------|------------|
| [31:0] | -      | reserved | 保留。         | 0x00000000 |

## PERI\_USB2\_GPRTBIMAP\_FS

PERI\_USB2\_GPRTBIMAP\_FS 为全速端口-总线映射低 32 位寄存器。

Offset Address: 0xC18C    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description             | Reset      |
|--------|--------|----------|-------------------------|------------|
| [31:4] | -      | reserved | 保留。                     | 0x00000000 |
| [3:0]  | RW     | binumn   | 指示当前连接的 fullspeed 的总线号。 | 0x0        |

## PERI\_USB2\_GUSB2PHYCFGN

PERI\_USB2\_GUSB2PHYCFGN 为全局 USB2.0 PHY 配置寄存器。

Offset Address: 0xC200    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name                    | Description                                                                                                                                                                                                                    | Reset |
|---------|--------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31]    | RW     | phy_soft_rst            | 触发 usb2phy_reset 信号对 UTMI PHY 进行软复位。此处不对 ULPI PHY 进行复位，因为 ULPIPHY 是通过 FunctionControl.Reset 寄存器进行复位的，当 core 复位的时候，core 自动配置如下寄存器进行复位：vcc_reset_n, 小 xHCI_USBCMD[hcrst], PERI_USB2_DCTL[SoftReset]or PERI_USB2_GCTL[SoftReset]。 | 0x0   |
| [30]    | RW     | u2_freeclk_exists       | USB2 PHY 是否提供 free_clk。<br>0: 不提供 free clock;<br>1: 提供 free clock。                                                                                                                                                             | 0x0   |
| [29:19] | -      | reserved                | 保留。                                                                                                                                                                                                                            | 0x000 |
| [18]    | RW     | ulpi_ext_vbus_indicator | ULPI 接口外部 Vbus 指示。<br>0: PHY 使用内部 Vbus 作为比较电压;<br>1: PHY 使用外部 Vbus 作为比较电压。                                                                                                                                                     | 0x0   |
| [17]    | RW     | ulpi_ext_vbus_drv       | ULPI 外 Vbus 驱动。<br>0: PHY 用内部 VBUS 电压驱动;<br>1: PHY 用外部 VBUS 电压驱动。                                                                                                                                                              | 0x0   |
| [16]    | -      | reserved                | 保留。                                                                                                                                                                                                                            | 0x0   |
| [15]    | RW     | ulpi_auto_res           | ULPI 自动唤醒。<br>0: PHY 不启用自动唤醒功能;<br>1: PHY 启用自动唤醒功能。                                                                                                                                                                            | 0x0   |
| [14]    | -      | reserved                | 保留。                                                                                                                                                                                                                            | 0x0   |
| [13:10] | RW     | usbtrdtim               | USB2 周转时间(Turnround Time), 指 MAC 请求 Packet FIFO Controller (PFC)                                                                                                                                                               | 0x0   |

| Bits | Access | Name          | Description                                                                                                                             | Reset |
|------|--------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------|-------|
|      |        |               | 从 DFIFO (SPRAM)取回数据的响应时间。<br>当 16-bit UTMI+ 时: 0x5<br>当 8-bit UTMI+/ULPI 接口时: 0x9。                                                      |       |
| [9]  | RW     | xcvrdly       | 收发延时选择,当此位置 1 时, 在 Transceiver Select 被置为 00(高速)和 TxValid 被拉为 0 之间加上 2.5us 的延时, 用于发送 chirp-K 握手信号。                                      | 0x0   |
| [8]  | RW     | enblslpm      | utmi_sleep_n 和 utmi_l1_suspend_n 信号使能。<br>0: utmi_sleep_n 和 utmi_l1_suspend_n 信号不接 PHY;<br>1: utmi_sleep_n 和 utmi_l1_suspend_n 信号接 PHY。 | 0x0   |
| [7]  | RW     | physel        | PHY 的接口类型选择。<br>0: USB2.0 高速 UTMI+或者 ULPI PHY;<br>1: USB1.1 全速串行接口;<br>当作为只写时, 此 bit 为 1。                                               | 0x0   |
| [6]  | RW     | susphy        | USB2.0 HS/FS/LS PHY 挂起选择。<br>0: 不挂起;<br>1: 挂起。<br>说明: DRD 模式时, core 初始化完毕后再将此 bit 置 1。                                                  | 0x0   |
| [5]  | RW     | fsintf        | 全速 PHY 串行接口类型选择。<br>0: 6-pin 单向全速串行传输接口;<br>1: 3-pin 双向全速串行传输接口。<br>当作为只读时, 返回值为 0。                                                     | 0x0   |
| [4]  | RW     | ulpi_utmi_sel | 高速 PHY 接口类型选择。<br>0: UTMI+ ;<br>1: ULPI。                                                                                                | 0x0   |

| Bits  | Access | Name    | Description                                                                                                                                                                                                                                                                                | Reset |
|-------|--------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [3]   | RW     | phyif   | UTMI 接口数据位宽选择。<br>0: 8bits;<br>1: 16bits。                                                                                                                                                                                                                                                  | 0x0   |
| [2:0] | RW     | toutcal | HS/FS Timeout 校准。<br>每个 PHY clock 加上相应的 bit time<br>High-speed 模式:<br>One 30-MHz PHY clock = 16 bit times<br>One 60-MHz PHY clock = 8 bit times<br>Full-speed 模式:<br>One 30-MHz PHY clock = 0.4 bit times<br>One 60-MHz PHY clock = 0.2 bit times<br>One 48-MHz PHY clock = 0.25 bit times | 0x0   |

## PERI\_USB2\_GTXFIFOSIZN

PERI\_USB2\_GTXFIFOSIZN 为全局发送 FIFO 大小寄存器。

Offset Address: 0xC304 Total Reset Value: 0x0000\_0000

| Bits    | Access | Name        | Description                                                                    | Reset  |
|---------|--------|-------------|--------------------------------------------------------------------------------|--------|
| [31:16] | RW     | txfstaddr_n | Transmit FIFO 在 RAM 中的起始地址。                                                    | 0x0000 |
| [15:0]  | RW     | txfdep_n    | Transmit FIFO 深度。<br>最小: 32 MDWIDTH-bit words;<br>最大: 32768 MDWIDTH-bit words。 | 0x0000 |

## PERI\_USB2\_GRXFIFOSIZN

PERI\_USB2\_GRXFIFOSIZN 为全局接收 FIFO 大小寄存器。

Offset Address: 0xC384 Total Reset Value: 0x0000\_0000

| Bits    | Access | Name        | Description             | Reset  |
|---------|--------|-------------|-------------------------|--------|
| [31:16] | RW     | rxfstaddr_n | Receive FIFO 在 RAM 中的起始 | 0x0000 |

|        |    |           |                                                                             |        |
|--------|----|-----------|-----------------------------------------------------------------------------|--------|
|        |    |           | 地址。                                                                         |        |
| [15:0] | RW | rxfddep_n | Receive FIFO 深度。<br>最小：32 MDWIDTH-bit words;<br>最大：32768 MDWIDTH-bit words。 | 0x0000 |

## PERI\_USB2\_GEVNTADRN\_HI

PERI\_USB2\_GEVNTADRN\_HI 为全局 event buffer 的地址高 32 位寄存器。

Offset Address: 0xC410    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name      | Description            | Reset      |
|--------|--------|-----------|------------------------|------------|
| [31:0] | RWSC   | evntadrhi | Event Buffer 高 32 位地址。 | 0x00000000 |

## PERI\_USB2\_GEVNTADRN\_LO

PERI\_USB2\_GEVNTADRN\_LO 为全局 event buffer 的地址低 32 位寄存器。

Offset Address: 0xC414    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name      | Description            | Reset      |
|--------|--------|-----------|------------------------|------------|
| [31:0] | RWSC   | evntadrlo | Event Buffer 低 32 位地址。 | 0x00000000 |

## PERI\_USB2\_GEVNTSIZN

PERI\_USB2\_GEVNTSIZN 为全局 event buffer 的大小寄存器。

Offset Address: 0xC418    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name        | Description            | Reset  |
|---------|--------|-------------|------------------------|--------|
| [31]    | RW     | evntintmask | Event 中断 Mask。         | 0x0    |
| [30:16] | -      | reserved    | 保留。                    | 0x0000 |
| [15:0]  | RW     | evntsiz     | Event Buffer 大小(byte)。 | 0x0000 |



## PERI\_USB2\_GEVNTCOUNTN

PERI\_USB2\_GEVNTCOUNTN 为全局 event buffer 的计数寄存器。

Offset Address: 0xC41C    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name      | Description                                                         | Reset  |
|---------|--------|-----------|---------------------------------------------------------------------|--------|
| [31:16] | -      | reserved  | 保留。                                                                 | 0x0000 |
| [15:0]  | RWSC   | evntcount | 当读此寄存器的时候,返回 Event Buffer 中有效的 event 数; 当写此寄存器的时候,硬件将自减写入的 count 数。 | 0x0000 |

## PERI\_USB2\_GTXFIFOPRIDEV

PERI\_USB2\_GTXFIFOPRIDEV 为外设的全局 TX FIFO DMA 优先寄存器。

Offset Address: 0xC610    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name                   | Description                               | Reset      |
|--------|--------|------------------------|-------------------------------------------|------------|
| [31:1] | -      | reserved               | 保留。                                       | 0x00000000 |
| [0]    | RW     | device_txfifo_priority | Device TXFIFO 优先级。<br>0: low;<br>1: high。 | 0x0        |

## PERI\_USB2\_GTXFIFOPRIHST

PERI\_USB2\_GTXFIFOPRIHST 为 Host 的全局 TX FIFO DMA 优先寄存器。

Offset Address: 0xC618    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name                 | Description                             | Reset      |
|--------|--------|----------------------|-----------------------------------------|------------|
| [31:1] | -      | reserved             | 保留。                                     | 0x00000000 |
| [0]    | RW     | host_txfifo_priority | Host TXFIFO 优先级。<br>0: low;<br>1: high。 | 0x0        |

## PERI\_USB2\_GRXFIFOPRIHST

PERI\_USB2\_GRXFIFOPRIHST 为 Host 的全局 RX FIFO DMA 优先寄存器。

Offset Address: 0xC61C    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name                 | Description                                 | Reset      |
|--------|--------|----------------------|---------------------------------------------|------------|
| [31:1] | -      | reserved             | 保留。                                         | 0x00000000 |
| [0]    | RW     | host_rxfifo_priority | Host RXFIFO 优先级。<br><br>0: low;<br>1: high. | 0x0        |

## PERI\_USB2\_GFIFOPRIDBC

PERI\_USB2\_GFIFOPRIDBC 为 host 的全局 Debug 性能时 DMA 优先寄存器。

Offset Address: 0xC620    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name                  | Description                                                              | Reset      |
|--------|--------|-----------------------|--------------------------------------------------------------------------|------------|
| [31:2] | -      | reserved              | 保留。                                                                      | 0x00000000 |
| [1:0]  | RW     | host_dbc_dma_priority | Host DbC DMA 优先级。<br><br>00: Low;<br>01: Normal;<br>10: High;<br>其他: 保留。 | 0x0        |

## PERI\_USB2\_GDMAHLRATIO

PERI\_USB2\_GDMAHLRATIO 为 host 的全局 FIFO DMA 高、低优先权比例寄存器。

Offset Address: 0xC624    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name                               | Description                 | Reset   |
|---------|--------|------------------------------------|-----------------------------|---------|
| [31:13] | -      | reserved                           | 保留。                         | 0x00000 |
| [12:8]  | RW     | hst_rxfifo_dma_hilo_priority_ratio | Host RXFIFO DMA 高-低 优先级 比例。 | 0x00    |
| [7:5]   | -      | reserved                           | 保留。                         | 0x0     |

| Bits  | Access | Name                               | Description                 | Reset |
|-------|--------|------------------------------------|-----------------------------|-------|
| [4:0] | RW     | hst_txfifo_dma_hilo_priority_ratio | Host TXFIFO DMA 高-低 优先级 比例。 | 0x00  |

## PERI\_USB2\_GFLADJ

PERI\_USB2\_GFLADJ 为 GFLADJ 为全局帧长度调整寄存器。

Offset Address: 0xC630 Total Reset Value: 0x0000\_0000

| Bits    | Access | Name                           | Description                                                                                                                                     | Reset  |
|---------|--------|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31]    | RW     | gfladj_refclk_240mhz_decr_pls1 | GFLADJ_REFCLK_240MHZ_DECR/ref_frequency 的精度调整。<br>0: 余数小于 0.5;<br>1: 余数大于等于 0.5。                                                                | 0x0    |
| [30:24] | RW     | gfladj_refclk_240mhz_decr      | 以 240/ref_clk_frequency 计算得出的计数器值:<br>$GFLADJ\_REFCLK\_240MHZ\_DECR = 240/ref\_clk\_frequency$ 。                                                | 0x00   |
| [23]    | RW     | gfladj_refclk_lpm_sel          | SOF/ITP 计数时钟选择。<br>若此位被置 1, 则 SOF/ITP 已 ref_clk 为时钟计数。                                                                                          | 0x0    |
| [22]    | RO     | reserved                       | 保留                                                                                                                                              | 0x0    |
| [21:8]  | RW     | gfladj_refclk_fladj            | 当 bit[23]为 1 时, SOF/ITP 校准值<br>$FLADJ\_REF\_CLK\_FLADJ = ((125000/ref\_clk\_period\_integer) - (125000/ref\_clk\_period)) * ref\_clk\_period$ 。 | 0x0000 |
| [7]     | RW     | gfladj_30mhz_reg_sel           | SOF/ITP 校准选择信号。<br>0: 控制器用 fladj_30mhz_reg 的值来对 SOF/ITP 进行校准;<br>1: 控制器用 GFLADJ.GFLADJ_30MHZ 的值来对 SOF/ITP 进行校准。                                 | 0x0    |
| [6]     | RO     | reserved                       | 保留。                                                                                                                                             | 0x0    |

| Bits  | Access | Name         | Description                                                          | Reset |
|-------|--------|--------------|----------------------------------------------------------------------|-------|
| [5:0] | RW     | gfladj_30mhz | 当第 bit[7] 为 1 且 SOF/ITP 以 UTMI/ULPI 为时钟计数时，控制器用 此域的值来对 SOF/ITP 进行校准。 | 0x00  |

## PERI\_USB2\_DCFG

PERI\_USB2\_DCFG 为外设配置寄存器。

Offset Address: 0xC700 Total Reset Value: 0x0008\_0004

| Bits    | Access | Name                      | Description                                                          | Reset |
|---------|--------|---------------------------|----------------------------------------------------------------------|-------|
| [31:24] | -      | reserved                  | 保留。                                                                  | 0x00  |
| [23]    | RW     | ignore_stream_pp          | 忽略 Packet Pending 比特位。<br>0: 不忽略;<br>1: 忽略。                          | 0x0   |
| [22]    | RW     | lpm_capable               | LPM 功能选择。<br>0: LPM 功能开启;<br>1: LPM 功能关闭。                            | 0x0   |
| [21:17] | RW     | number_of_receive_buffers | 接收 Buffer 号。<br>指示 ACK TP 中返回的接收 buffer 号。                           | 0x04  |
| [16:12] | RW     | interrupt_number          | 中断号指示。<br>指示设备产生的非端点中断的中断号。                                          | 0x00  |
| [11:10] | -      | reserved                  | 保留。                                                                  | 0x0   |
| [9:3]   | RW     | device_address            | 设备地址。<br>每当 Device 接收到 SetAddress 命令，对此处配置 device 的地址。USB 复位后，此处清 0。 | 0x00  |
| [2:0]   | RW     | device_speed              | Device 传输速率选择。<br>000: 高速;<br>001: 全速;                               | 0x4   |

| Bits | Access | Name | Description         | Reset |
|------|--------|------|---------------------|-------|
|      |        |      | 100: 超速;<br>其他: 保留。 |       |

## PERI\_USB2\_DCTL

PERI\_USB2\_DCTL 为外设控制寄存器。

Offset Address: 0xC704    Total Reset Value: 0x0000\_0000

| Bits | Access | Name            | Description                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|------|--------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31] | RW     | run_stop        | <p>设备开启/停止选择。</p> <p>软件写 1 开启 device 的操作；如果要停止 device 操作，先停止一切的正在进行的传输，在往此 bit 写 0。</p> <p>Run/stop 比特的必须在如下的情况下操作：</p> <p>1.上电复位后，软件必须给此位写 1 以初始化 device 控制器，否则 device controller 将不会知会 host 已经连接上；</p> <p>2.如果软件要在软断开或者检测到断开事件后重新启动连接，则需要往此比特写 1 之前将 DCTL[8: 5]设置成 5；</p> <p>3.当 USB 处于低功耗状态并且 USB 是双电源域配置，软件对此比特写 0 意味着将关闭 core 电源域。在软件重启 core 电源域和 device 控制器后，将此比特置 1。</p> | 0x0   |
| [30] | RWSC   | soft_core_reset | <p>core 软复位。</p> <p>将对所有时钟域的中断和寄存器做如下操作：</p> <p>1: 清除所有中断和复位出一下寄存器以外的所有寄存器；</p> <ul style="list-style-type: none"> <li>- GCTL</li> <li>- GUCTL</li> <li>- GSTS</li> </ul>                                                                                                                                                                                                       | 0x0   |

| Bits    | Access | Name           | Description                                                                                                                                                                                                                                                                                                                                                                                                                               | Reset |
|---------|--------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
|         |        |                | <ul style="list-style-type: none"> <li>- GSNPSID</li> <li>- GGPI0</li> <li>- GUID</li> <li>- GUSB2PHYCFGn registers</li> <li>- GUSB2PIPECTLn registers</li> <li>- DCFG</li> <li>- DCTL</li> <li>- DEVTEN</li> <li>- DSTS</li> </ul> <p>2: 所有状态机将回归 IDLE 状态, 所有的 FIFO 将清 0;</p> <p>3: SoC 总线上的所有传输将立刻终止。</p>                                                                                                                               |       |
| [29]    | -      | reserved       | 保留。                                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |
| [28:24] | RW     | hird_threshold | <p>HIRD 门限值</p> <p>core 将在满足以下的条件下产生 utmi_l1_suspend_n 和 utmi_sleep_n 信号。</p> <p>1.当满足下面的条件时, core 在 L1 状态产生 utmi_l1_suspend_n , 将 PHY 切换到深度 Low-Power:</p> <ul style="list-style-type: none"> <li>-HIRD 的值大于或者等于 DCTL.HIRD_Thres[3:0]</li> <li>-HIRD_Thres[4]的值为 1</li> </ul> <p>2.当满足以下条件时, core 在 L1 状态将产生 utmi_sleep_n:</p> <ul style="list-style-type: none"> <li>-HIRD 的值小于 HIRD_Thres[3:0]</li> <li>-HIRD_Thres[4]的值为 0</li> </ul> | 0x00  |
| [23]    | RW     | appl1res       | <p>LPM 响应配置。</p> <p>当 PERI_USB2_DCFG[lpm_capable]为 0: core 没有响应;</p> <p>当 PERI_USB2_DCFG[lpm_capable]为 1:</p> <p>0: 当满足下面的条件时, LPM 收发成功</p>                                                                                                                                                                                                                                                                                               | 0x0   |

| Bits    | Access | Name                     | Description                                                                                                                                                                                              | Reset |
|---------|--------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
|         |        |                          | 后将返回 ACK<br>-没有校验错误<br>-接收到有效的 bLinkState = 0001B (L1)<br>-发送 FIFO 中没有存留数据并且 OUT 端点没有处于 flow controlled state (NYET 除外)<br>1: 当满足下面的条件时, LPM 收发成功后将返回 ACK<br>-没有校验错误<br>-接收到有效的 bLinkState = 0001B (L1)。 |       |
| [22:20] | -      | reserved                 | 保留。                                                                                                                                                                                                      | 0x0   |
| [19]    | RW     | keepconnect              | 保持连接状态。<br>如果此位被置 1 时, 当 runStop =0 时, 所有涉及模型将得到保存或被恢复。同时, 当 link 进入 U3 或者 L2 时, 将使能 Hibernation Request Event。                                                                                          | 0x0   |
| [18]    | RW     | l1_hibernation_en        | 当 keepconnect 为 1, 并且此位为 1, 如果 L1 被使能 并且 HIRD 的值大于 PERI_USB2_DCTL[hird_threshold]中配置的阈值, device core 将产生一个 Hibernation Request Event。                                                                    | 0x0   |
| [17]    | RW     | controller_restore_state | 控制器恢复状态。<br>当软件对此比特写 1 时, 控制器立即将 DSTS.RSS 置 1, 当控制器完成恢复以后, 将 PERI_USB2_DSTS[rss]置 0。<br>说明: 当读此比特的时候, 返回 0。                                                                                              | 0x0   |
| [16]    | RW     | controller_save_state    | 控制器保存状态。<br>当软件对此比特写 1 时, 控制器立即将 DSTS.SSS 置 1, 当控制器保存以后, 将 PERI_USB2_DSTS[sss]置 0。                                                                                                                       | 0x0   |

| Bits    | Access | Name               | Description                                                                                                                        | Reset |
|---------|--------|--------------------|------------------------------------------------------------------------------------------------------------------------------------|-------|
|         |        |                    | 说明：当读此比特的時候，返回 0。                                                                                                                  |       |
| [15:13] | -      | reserved           | 保留。                                                                                                                                | 0x0   |
| [12]    | RW     | initiate_u2_enable | 初始化 U2 使能寄存器。<br>0：不初始化 U2；<br>1：初始化 U2。                                                                                           | 0x0   |
| [11]    | RW     | accept_u2_enable   | U2 状态使能。<br>0：core 拒绝进入 U1 状态除非；<br>Force_LinkPM_Accep 为 1<br>1：如果用户端没有任何设备在等待，则<br>core 可以接受允许进入 U2 状态。<br>说明：USB 复位后，硬件自动清除此比特位。 | 0x0   |
| [10]    | RW     | initiate_u1_enable | 初始化 U1 使能寄存器。<br>0：不初始化 U1；<br>1：初始化 U1。                                                                                           | 0x0   |
| [9]     | RW     | accept_u1_enable   | U1 状态使能。<br>0：core 拒绝进入 U1 状态除非；<br>Force_LinkPM_Accep 为 1；<br>1：如果用户端没有任何设备在等待，则<br>core 可以接受允许进入 U1 状态<br>说明：USB 复位后，硬件自动清除此比特位。 | 0x0   |
| [8:5]   | WO     | ulstchngraq        | USB/Link 状态切换请求。<br>软件对配置此处发起相应的状态切换请求<br>SS 模式：<br>0x0：No Action；<br>0x4：SS.Disabled；<br>0x5：Rx.Detect；<br>0x6：SS.Inactive；       | 0x0   |



| Bits  | Access | Name         | Description                                                                                                                                                      | Reset |
|-------|--------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
|       |        |              | 0x8: Recovery;<br>0xA: Compliance;<br>其他: 保留。<br>HS/FS/LS 模式:<br>0x8: Remote wakeup request;<br>其他: 保留。                                                          |       |
| [4:1] | RW     | test_control | 测试控制。<br>0x0: Test mode 关闭;<br>0x1: Test_J mode;<br>0x2: Test_K mode;<br>0x3: Test_SE0_NAK mode;<br>0x4: Test_Packet mode;<br>0x5: Test_Force_Enable;<br>其他: 保留。 | 0x0   |
| [0]   | -      | reserved     | 保留。                                                                                                                                                              | 0x0   |

## PERI\_USB2\_DEVTEN

PERI\_USB2\_DEVTEN 为外设事件使能寄存器。

Offset Address: 0xC708    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name                       | Description                    | Reset    |
|---------|--------|----------------------------|--------------------------------|----------|
| [31:10] | -      | reserved                   | 保留。                            | 0x000000 |
| [9]     | RW     | erratic_error_event_enable | 不规则错误事件使能。<br>0: 禁止;<br>1: 使能。 | 0x0      |
| [8]     | -      | reserved                   | 保留。                            | 0x0      |
| [7]     | RW     | sof_en                     | SOF 包使能。<br>0: 禁止;             | 0x0      |

| Bits | Access | Name                   | Description                            | Reset |
|------|--------|------------------------|----------------------------------------|-------|
|      |        |                        | 1: 使能。                                 |       |
| [6]  | RW     | u3l2l1_susp_en         | U3/L2-L1 挂起事件使能。<br>0: 禁止;<br>1: 使能。   | 0x0   |
| [5]  | RW     | hibernation_req_evt_en | 休眠请求使能。<br>0: 禁止;<br>1: 使能。            | 0x0   |
| [4]  | RW     | wkup_evt_en            | 恢复/远程唤醒检测事件使能。<br>0: 禁止;<br>1: 使能。     | 0x0   |
| [3]  | RW     | ulst_cng_en            | USB/LINK 状态切换事件使能。<br>0: 禁止;<br>1: 使能。 | 0x0   |
| [2]  | RW     | connect_done_en        | 连接完成使能。<br>0: 禁止;<br>1: 使能。            | 0x0   |
| [1]  | RW     | usbrst_en              | USB 复位使能。<br>0: 禁止;<br>1: 使能。          | 0x0   |
| [0]  | RW     | disconn_evt_en         | 断开检测使能。<br>0: 禁止;<br>1: 使能。            | 0x0   |

## PERI\_USB2\_DSTS

PERI\_USB2\_DSTS 为外设状态寄存器。

Offset Address: 0xC70C    Total Reset Value: 0x0012\_0004

| Bits    | Access | Name       | Description                                                                                                                               | Reset |
|---------|--------|------------|-------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:30] | -      | reserved   | 保留。                                                                                                                                       | 0x0   |
| [29]    | RO     | dcnrd      | Device 控制器未准备好。<br>指示控制器从休眠状态退出后，正在进行状态转换。                                                                                                | 0x0   |
| [28:26] | -      | reserved   | 保留。                                                                                                                                       | 0x0   |
| [25]    | RO     | rss        | 恢复状态。<br>当控制器完成恢复的操作后，将此位置 0。                                                                                                             | 0x0   |
| [24]    | RO     | sss        | 保存状态。<br>当控制器完成保存的操作后，将此位置 0。                                                                                                             | 0x0   |
| [23]    | RO     | core_idle  | core 空闲状态。<br>指示 core 完成了所有 RxFIFO 的数据向系统 memory 的传输，完成所有链表的输出，所有事件计数器归 0。                                                                | 0x0   |
| [22]    | RO     | devctrlhlt | Device 控制器挂起。<br>当 PERI_USB2_DCTL[run_stop]为 1 的时候，此位置 0；<br>当软件向 PERI_USB2_DCTL[run_stop]写 0 后，并且 core 处于空闲状态以及更下一层完成断开的操作，core 将对此位写 1。 | 0x0   |
| [21:18] | RO     | usb_lnk_st | USB/Link 状态。<br>SS 模式：<br>0x0: U0；<br>0x1: U1；<br>0x2: U2；<br>0x3: U3；<br>0x4: SS_DIS；<br>0x5: RX_DET；                                    | 0x4   |

| Bits   | Access | Name        | Description                                                                                                                                                                                                                                                                                                                                            | Reset  |
|--------|--------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
|        |        |             | 0x6: SS_INACT;<br>0x7: POLL;<br>0x8: RECOV;<br>0x9: HRESET;<br>0xa: CMPPLY;<br>0xb: LPBK;<br>0xf: Resume/Reset;<br>其他: 保留。<br>HS/FS/LS 模式:<br>0x0: On 状态;<br>0x2: Sleep (L1) 状态;<br>0x3: Suspend (L2) 状态;<br>0x4: Disconnected 状态 (默认状态);<br>0x5: Early Suspend 状态(仅当休眠功能关闭是有效);<br>0xe: Reset (仅当休眠功能开启时有效);<br>0xf: Resume (仅当休眠功能开启时有效);<br>其他: 保留。 |        |
| [17]   | -      | reserved    | 保留。                                                                                                                                                                                                                                                                                                                                                    | 0x1    |
| [16:3] | RO     | soffn       | SOF 帧/微帧数目:<br>HS 模式<br>[16:6]指示帧数目<br>[5:3]指示微帧数目<br>全速模式<br>[16:14]不用<br>[13:3]指帧数目                                                                                                                                                                                                                                                                  | 0x0000 |
| [2:0]  | RO     | connect_spd | 连接速度指示。<br>000: 高速(PHY clock 工作在 30 或者                                                                                                                                                                                                                                                                                                                 | 0x4    |

| Bits | Access | Name | Description                                                                                                                                           | Reset |
|------|--------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
|      |        |      | 60MHz);<br>001: 全速(PHY clock 工作在 30 或者 60MHz);<br>010: 低速 (PHY clock 工作在 6MHz);<br>010: 低速 (PHY clock 工作在 48MHz)100: 超速(PHY clock 工作在 125 或者 250MHz)。 |       |

## PERI\_USB2\_DGCMDPAR

PERI\_USB2\_DGCMDPAR 为外设类命令参数寄存器。

Offset Address: 0xC710 Total Reset Value: 0x0000\_0000

| Bits   | Access | Name          | Description   | Reset      |
|--------|--------|---------------|---------------|------------|
| [31:0] | RW     | parameter31_0 | 配置外设类命令参数寄存器。 | 0x00000000 |

## PERI\_USB2\_DGCMD

PERI\_USB2\_DGCMD 为外设类命令寄存器。

Offset Address: 0xC714 Total Reset Value: 0x0000\_0000

| Bits    | Access | Name       | Description                                              | Reset  |
|---------|--------|------------|----------------------------------------------------------|--------|
| [31:16] | -      | reserved   | 保留。                                                      | 0x0000 |
| [15]    | RO     | cmd_status | 命令状态。<br>0: Device 控制器处理命令成功;<br>1: Device 控制器处理命令时遇到错误。 | 0x0    |
| [14:11] | -      | reserved   | 保留。                                                      | 0x0    |
| [10]    | RWSC   | cmdact     | 软件设置此位为 1, 使能 Device 控制器执行产生的命令<br>控制器执行完命令后将此位置 0。      | 0x0    |
| [9]     | -      | reserved   | 保留。                                                      | 0x0    |

| Bits  | Access | Name     | Description                                                                                                                                                 | Reset |
|-------|--------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [8]   | WO     | cmdloc   | 命令完成中断。<br>说明：PERI_USB2_DCTL[run_stop]为 0 时，此位为 1。                                                                                                          | 0x0   |
| [7:4] | -      | reserved | 保留。                                                                                                                                                         | 0x0   |
| [3:0] | RW     | cmdtyp   | 命令类型。<br>0x0：保留；<br>0x1：设置端点的配置；<br>0x2：设置端点传输源配置；<br>0x3：获取端点状态；<br>0x4：设置挂起；<br>0x5：清除挂起；<br>0x6：开始传输；<br>0x7：跟新传输；<br>0x8：结束传输；<br>0x9：开始新的配置；<br>其他：保留。 | 0x0   |

## PERI\_USB2\_DALEPENA

PERI\_USB2\_DALEPENA 为外设 USB 端点使能寄存器。

Offset Address: 0xC718    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name     | Description                                                                                                                                   | Reset      |
|--------|--------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------|------------|
| [31:0] | RW     | usbactep | 指示了在当前的配置和接口下，USB 的端口是否有效。对于 USB 输入端点 0-15 和输出端点 0-15，所有 32 个端点每个通过 1bit 来控制，偶数位控制输出端点，奇数位控制输入端点。例：<br>Bit[0]：USB 输出端点 0<br>Bit[1]：USB 输入端点 0 | 0x00000000 |

|  |  |  |                                                                                                                                                                                                                        |  |
|--|--|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
|  |  |  | <p>Bit[2]: USB 输出端点 1</p> <p>Bit[3]: USB 输入端点 1</p> <p>对该寄存器操作时必须设置 bit0 和 bit1, 这样 USBReset 后才会使能物理端点中的控制端点。</p> <p>当探测到 USB reset 时, 应用软件会清除了输出端点 0/输入端点 0 其他所有端点对应的位。当收到设置配置和设置接口要求时, 应用程序相应的设置端点寄存器以及本寄存器的这些位。</p> |  |
|--|--|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

## PERI\_USB2\_DEPCMDPAR2N

PERI\_USB2\_DEPCMDPAR2N 为外设端点命令参数寄存器 2。

Offset Address: 0xC810    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name       | Description                          | Reset      |
|--------|--------|------------|--------------------------------------|------------|
| [31:0] | RW     | parameter2 | 外设的物理端点命令参数寄存器 2, 该寄存器必须在执行命令之前进行配置。 | 0x00000000 |

## PERI\_USB2\_DEPCMDPAR1N

PERI\_USB2\_DEPCMDPAR1N 为外设端点命令参数寄存器 1。

Offset Address: 0xC814    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name       | Description                          | Reset      |
|--------|--------|------------|--------------------------------------|------------|
| [31:0] | RW     | parameter1 | 外设的物理端点命令参数寄存器 1, 该寄存器必须在执行命令之前进行配置。 | 0x00000000 |

## PERI\_USB2\_DEPCMDPAR0N

PERI\_USB2\_DEPCMDPAR0N 为外设端点命令参数寄存器 0。

Offset Address: 0xC818    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name       | Description                                                                   | Reset      |
|--------|--------|------------|-------------------------------------------------------------------------------|------------|
| [31:0] | RW     | parameter0 | 外设的物理端点命令参数寄存器 0，该寄存器必须在命令之前或者伴随着进行配置。因为命令仅仅需要 32bit 的参数，所以该寄存器必须随着命令寄存器进行配置。 | 0x00000000 |

## PERI\_USB2\_DEPCMDN

PERI\_USB2\_DEPCMDN 为外设物理端点命令寄存器。

Offset Address: 0xC81C    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name               | Description                                                                                                                                                                                                                                                                 | Reset  |
|---------|--------|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:16] | RW     | command_parameters | <p>这些 bit 表示命令的参数，往寄存器写相关值其意义如下：</p> <p>对于开始传输命令：</p> <p>[31:16]：表示为本次传输声明的 USB StreamID。</p> <p>对于实时端点的开始传输命令：</p> <p>[31:16]：表示首个 TRB 应用时的微帧(帧)的数目。</p> <p>对于更新传输、结束传输和开始新的配置命令：</p> <p>[22:16]：传输资源索引(XferRscIdx)。硬件为传输申明传输资源索引，其返回后来响应开始传输命令；应用软件为开始新的配置命令声明传输资源索引。</p> | 0x0000 |
| [15:12] | RW     | cmd_status         | 命令完成状态：这些位保存了已完成命令的额外信息。这些信息与端点命令完成事件的 bit[15:12]完全相同。                                                                                                                                                                                                                      | 0x0    |
| [11]    | RW     | hipri_forcerm      | <p>HighPriority：仅在开始传输命令才有效。</p> <p>ForceRM：仅在结束传输命令才有效。</p>                                                                                                                                                                                                                | 0x0    |
| [10]    | RW     | cmd_act            | 软件设置该 bit 为 1 来使能外设端点控制器执行类命令。                                                                                                                                                                                                                                              | 0x0    |



|       |    |          |                                                                                                                                                                                                                                                                   |     |
|-------|----|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
|       |    |          | 当本寄存器的 CmdStatus 有效，同时端点已经准备好来接收另一个命令时，外设控制器将会设置该 bit 为 0。但是这并不能表明先前发送的命令都已实现。                                                                                                                                                                                    |     |
| [9]   | -  | reserved | 保留。                                                                                                                                                                                                                                                               | 0x0 |
| [8]   | RW | cmdloc   | 命令完成中断：当该 bit 设置为 1 时，表明执行命令后，外设控制器产生了一个通用的端点命令完成事件。<br><br>说明：当 PERI_USB2_DCTL[run_stop]=0 时，该 bit 不能设置为 1。                                                                                                                                                      | 0x0 |
| [7:4] | -  | reserved | 保留。                                                                                                                                                                                                                                                               | 0x0 |
| [3:0] | RW | cmdtyp   | 定义了驱动要求 core 去执行的命令类型。<br>0x0：保留；<br>0x1：设置端点配置(64 或 96bit 参数)；<br>0x2：设置端点传输资源配置(32bit 参数)；<br>0x3：得到端点状态(不需要参数)；<br>0x4：设置 Stall(不需要参数)；<br>0x5：清除 Stall(不需要参数)；<br>0x6：开始传输(64bit 参数)；<br>0x7：更新传输(不需要参数)；<br>0x8：结束传输(不需要参数)；<br>0x9：开始新的配置(不需要参数)；<br>其他：保留。 | 0x0 |

## 2.7 LSADC

### 2.7.1 概述

LSADC(Low Speed ADC)实现对外部模拟信号转换成一定比例的数字值，从而实现对模拟信号的测量，可应用于电量检测，按键检测等。芯片中的 LSADC 是由一个 12bit

的逐次逼近数模转换电路（SAR ADC）实现。可实现 3 通道分时复用，独立输入，采样率为 200ksps；电源电压 3.3V，参考电压为电源电压 3.3V。

## 2.7.2 特点

LSADC 具有以下特点：

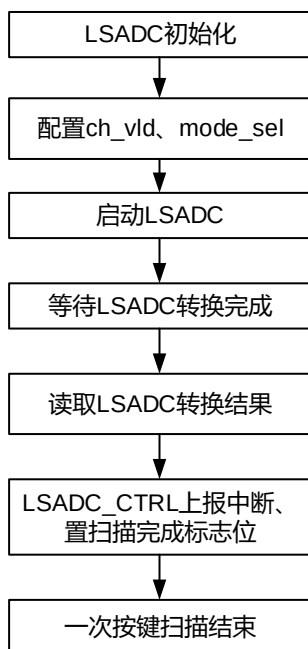
- 参考电源电压 3.3V；
- 扫描频率不能高于 200ksps；
- 12bit 采样精度，3 个独立通道；
- 支持单次扫描和连续扫描模式；
- 支持 16 次平均算法模式和 32 次平均算法模式；
- 扫描完成自动上报中断；
- 支持 lsadc 外部片选信号配置调节，此信号至少维持 2 个时钟周期。

## 2.7.3 工作方式

### 单次扫描处理流程

在单次读取模式（LSADC\_CTRL0[model\_sel]=0），CPU 配置扫描通道号（仅且只能配置一个通道）、扫描模式、键值映射表信息，启动 LSADC 完成一次通道扫描。通道扫描完成后，通过中断通知系统扫描完成，CPU 可以获取转换结果。

图2-33 单次扫描处理流程



## 连续扫描处理流程

在连续读取模式 ([mode\_sel]=1)，CPU 根据应用场景设置连续扫描的时间间隔 Tscan、毛刺宽度 (Tglitch)、有效通道号 (ch\_vld)，启动 LSADC。LSADC 在一个时间间隔 Tscan 内完成一个有效通道（配置通道是否有效指示位为有效）的扫描。在下一个扫描时刻到来时，启动对下一个有效通道的扫描。待完成对所有有效通道的扫描后，启动下一轮对有效通道的扫描。

以使能通道 0、1 为例，连续扫描模式下通道轮询扫描示意图如图 2-34 所示。

图2-34 连续扫描模式下通道轮询扫描示意图

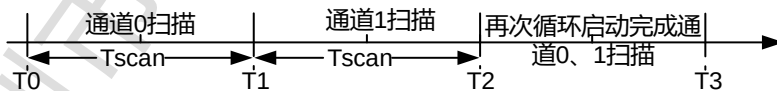
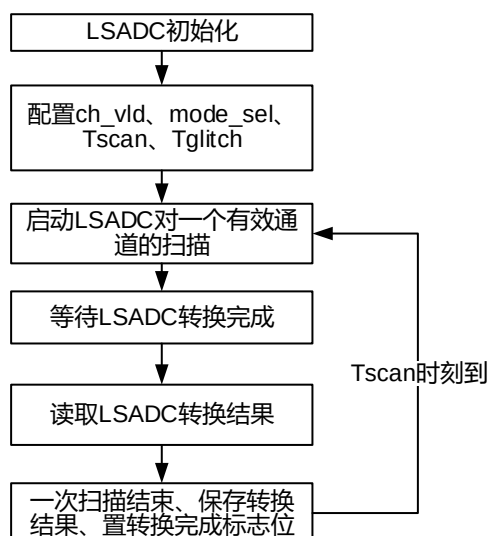


图2-35 连续扫描处理流程



## 滤毛刺流程

滤毛刺电路采用多数判决算法。在滤毛刺窗口 Tglitch 中，如果出现多数次的 ADC 采样值 value，且 value 不为空按键时 ADC 采样值，则认为 value 为一次有效的按键值，否则认为是一个毛刺信号。

- 在单次读取模式，不进行滤毛刺操作。
- 在连续扫描模式下，使能滤毛刺功能，需要设置合适的滤毛刺时间窗口 Tglitch。请参考 LSADC\_CTRL1 寄存器的描述。

## 采样精度设置

通过 LSADC\_CTRL9 [11:0]可以设置采样精度，可以根据应用需要设置对应的采样精度。

- 当采样精度设置成 12bit 时，采样结果 12bit 全部有效。
- 当采样精度设置小于 12bit 时，对应的采样结果高位有效，例如：当采样精度设置成 10bit 时，采样结果的高 10bit 才有效。

## 平均算法设置

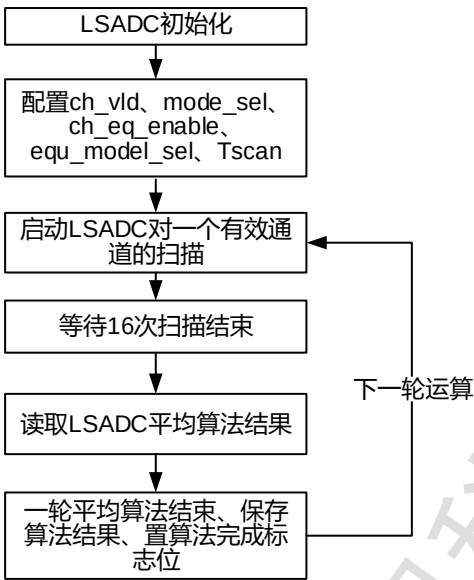
通过 LSADC\_CTRL0[12]可以选择通道采样的平均算法模式，可以根据应用需要设置对应的平均算法模式。寄存器的描述。

- 单次读取模式，不支持采样平均算法操作。

- 连续扫描模式，可选择 16 次或 32 次平均算法。

以使能 16 次平均算法模式为例，连续扫描模式下通道平均算法示意图如图 2-36 所示。

图2-36 16 次平均算法处理流程



2.7.4 LSADC 寄存器概览

LSADC 寄存器概览如表 2-19 所示。

LSADC 基地址：0x120A\_0000

表2-19 LSADC 寄存器概览

| 偏移地址   | 名称          | 描述              |
|--------|-------------|-----------------|
| 0x0000 | LSADC_CTRL0 | LSADC 配置寄存器     |
| 0x0004 | LSADC_CTRL1 | 滤毛刺配置寄存器        |
| 0x0008 | LSADC_CTRL2 | 扫描间隔配置寄存器       |
| 0x000C | LSADC_CTRL3 | LSADC 片选信号长度寄存器 |
| 0x0010 | LSADC_CTRL4 | 中断使能寄存器         |
| 0x0014 | LSADC_CTRL5 | 中断状态寄存器         |

| 偏移地址   | 名称           | 描述                   |
|--------|--------------|----------------------|
| 0x0018 | LSADC_CTRL6  | 中断清除寄存器              |
| 0x001C | LSADC_CTRL7  | Start 配置寄存器          |
| 0x0020 | LSADC_CTRL8  | Stop 配置寄存器           |
| 0x0024 | LSADC_CTRL9  | 转换结果精度寄存器            |
| 0x0028 | LSADC_CTRL10 | LSADC_ZERO 寄存器       |
| 0x002C | LSADC_CTRL11 | LSADC 通道 0 数据寄存器     |
| 0x0030 | LSADC_CTRL12 | LSADC 通道 1 数据寄存器     |
| 0x0034 | LSADC_CTRL13 | LSADC 通道 2 数据寄存器     |
| 0x003C | LSADC_CTRL15 | LSADC 配置寄存器          |
| 0x0050 | LSADC_CTRL20 | LSADC 通道 0 平均算法数据寄存器 |
| 0x0054 | LSADC_CTRL21 | LSADC 通道 1 平均算法数据寄存器 |
| 0x0058 | LSADC_CTRL22 | LSADC 通道 2 平均算法数据寄存器 |

## 2.7.5 LSADC 寄存器描述

### LSADC\_CTRL0

LSADC\_CTRL0 为 LSADC 配置寄存器。

Offset Address: 0x0000    Total Reset Value: 0x0000\_80F7

| Bits    | Access | Name             | Description                                                     | Reset |
|---------|--------|------------------|-----------------------------------------------------------------|-------|
| [31:24] | -      | reserved         | 保留。                                                             | 0x00  |
| [23:20] | RW     | lsadc_data_delta | LSADC 转换结果误差范围(在连续扫描模式下使用，使用确定两次转换结果间的误差范围，在误差范围内，则认定两次转换结果相同)。 | 0x0   |
| [19:18] | -      | reserved         | 保留。                                                             | 0x0   |
| [17]    | RW     | deglitch_bypass  | 滤毛刺功能 bypass。(此功能在连续扫描模                                         | 0x0   |

| Bits  | Access | Name          | Description                                         | Reset |
|-------|--------|---------------|-----------------------------------------------------|-------|
|       |        |               | 式下使用)<br>0: 使能滤毛刺功能;<br>1: bypass.                  |       |
| [16]  | -      | reserved      | 保留。                                                 | 0x0   |
| [15]  | RW     | lsadc_reset   | 设置 LSADC 是否进入复位状态。<br>0: 进入复位状态;<br>1: 退出复位状态。      | 0x1   |
| [14]  | -      | reserved      | 保留。                                                 | 0x0   |
| [13]  | RW     | model_sel     | LSADC 扫描模式选择。<br>0: 单次扫描模式;<br>1: 连续扫描模式。           | 0x0   |
| [12]  | RW     | equ_model_sel | LSADC 平均算法模式选择。<br>0: 16 次平均算法模式;<br>1: 32 次平均算法模式。 | 0x0   |
| [11]  |        | reserved      | 保留。                                                 | 0x0   |
| [10]  | RW     | ch_2_vld      | LSADC 通道 2 是否有效。<br>0: 无效;<br>1: 有效。                | 0x0   |
| [9]   | RW     | ch_1_vld      | LSADC 通道 1 是否有效。<br>0: 无效;<br>1: 有效。                | 0x0   |
| [8]   | RW     | ch_0_vld      | LSADC 通道 0 是否有效。<br>0: 无效;<br>1: 有效。                | 0x0   |
| [7:4] | -      | reserved      | 保留。                                                 | 0xF   |
| [3]   |        | reserved      | 保留。                                                 | 0x0   |

| Bits | Access | Name           | Description                                | Reset |
|------|--------|----------------|--------------------------------------------|-------|
| [2]  | RW     | ch_2_eq_enable | LSADC 通道 0 采样平均算法是否有效。<br>0: 无效;<br>1: 有效。 | 0x1   |
| [1]  | RW     | ch_1_eq_enable | LSADC 通道 1 采样平均算法是否有效。<br>0: 无效;<br>1: 有效。 | 0x1   |
| [0]  | RW     | ch_0_eq_enable | LSADC 通道 0 采样平均算法是否有效。<br>0: 无效;<br>1: 有效。 | 0x1   |

## LSADC\_CTRL1

LSADC\_CTRL1 为滤毛刺配置寄存器。

Offset Address: 0x0004    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name          | Description                                                                                | Reset      |
|--------|--------|---------------|--------------------------------------------------------------------------------------------|------------|
| [31:0] | RW     | glitch_sample | 滤毛刺时间窗口(当 LSADC 的转换结果在此时间窗口内保持不变时, 则认为此转换结果为有效值, 否则认为是毛刺, 此窗口值一般设为 ms 级, 在连续扫描模式下不能配置为 0)。 | 0x00000000 |

## LSADC\_CTRL2

LSADC\_CTRL2 为扫描间隔配置寄存器。

Offset Address: 0x0008    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name      | Description                                                                                              | Reset      |
|--------|--------|-----------|----------------------------------------------------------------------------------------------------------|------------|
| [31:0] | RW     | time_scan | 在连续扫描模式下, 两个扫描通道之间连续扫描的时间间隔。时间间隔为<br>$T_{scan} = time\_scan * 0.25\mu s$ (连续扫描时间间隔应该大于 LSADC 的转换时间, 即配置值 | 0x00000000 |



|  |  |  |           |  |
|--|--|--|-----------|--|
|  |  |  | 不能少于 20)。 |  |
|--|--|--|-----------|--|

## LSADC\_CTRL3

LSADC\_CTRL3 为片选信号长度寄存器。

Offset Address: 0x000C    Total Reset Value: 0x0000\_0503

| Bits   | Access | Name   | Description       | Reset       |
|--------|--------|--------|-------------------|-------------|
| [15:8] | RW     | pd_td1 | power down 计数长度   | 0x0000_0500 |
| [7:0]  | RW     | pd_td0 | 配置 st 为外部片选信号使能长度 | 0x0000_0003 |

## LSADC\_CTRL4

LSADC\_CTRL4 为中断使能寄存器。

Offset Address: 0x0010    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name       | Description                      | Reset      |
|--------|--------|------------|----------------------------------|------------|
| [31:1] | -      | reserved   | 保留。                              | 0x00000000 |
| [0]    | RW     | int_enable | 扫描值有效中断使能位。<br>0: 不使能;<br>1: 使能。 | 0x0        |

## LSADC\_CTRL5

LSADC\_CTRL5 为中断状态寄存器。

Offset Address: 0x0014    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name            | Description                | Reset      |
|--------|--------|-----------------|----------------------------|------------|
| [31:5] | -      | reserved        | 保留。<br>写无效, 读为 0。          | 0x00000000 |
| [4]    | RO     | lsadc_auto_busy | 在自动扫描模式下, LSADC busy 状态指示。 | 0x0        |

| Bits | Access | Name         | Description                            | Reset |
|------|--------|--------------|----------------------------------------|-------|
|      |        |              | 0: IDLE;<br>1: BUSY。                   |       |
| [3]  |        | reserved     | 保留。                                    | 0x0   |
| [2]  | RO     | int_flag_in2 | 通道 2 扫描值有效中断标志位。<br>0: 无中断;<br>1: 有中断。 | 0x0   |
| [1]  | RO     | int_flag_in1 | 通道 1 扫描值有效中断标志位。<br>0: 无中断;<br>1: 有中断。 | 0x0   |
| [0]  | RO     | int_flag_in0 | 通道 0 扫描值有效中断标志位。<br>0: 无中断;<br>1: 有中断。 | 0x0   |

## LSADC\_CTRL6

LSADC\_CTRL6 为中断清除寄存器。

Offset Address: 0x0018    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name             | Description                          | Reset      |
|--------|--------|------------------|--------------------------------------|------------|
| [31:4] | -      | reserved         | 保留。                                  | 0x00000000 |
| [3]    |        | reserved         | 保留。                                  | 0x0        |
| [2]    | WO     | clr_int_flag_in2 | 通道 2 中断清除寄存器。<br>0: 不清除;<br>1: 清除中断。 | 0x0        |
| [1]    | WO     | clr_int_flag_in1 | 通道 1 中断清除寄存器。<br>0: 不清除;<br>1: 清除中断。 | 0x0        |
| [0]    | WO     | clr_int_flag_in0 | 通道 0 中断清除寄存器。                        | 0x0        |

|  |  |  |                     |  |
|--|--|--|---------------------|--|
|  |  |  | 0: 不清除;<br>1: 清除中断。 |  |
|--|--|--|---------------------|--|

## LSADC\_CTRL7

LSADC\_CTRL7 为 Start 配置寄存器。

Offset Address: 0x001C    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name  | Description                       | Reset      |
|--------|--------|-------|-----------------------------------|------------|
| [31:0] | WO     | start | LSADC 启动信号(向此寄存器写任意值都可以启动 LSADC)。 | 0x00000000 |

## LSADC\_CTRL8

LSADC\_CTRL8 为 Stop 配置寄存器。

Offset Address: 0x0020    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name | Description                                                             | Reset      |
|--------|--------|------|-------------------------------------------------------------------------|------------|
| [31:0] | WO     | stop | 停止自动扫描(在自动扫描模式下, 向此寄存器写任意值都可以停止 LSADC 的自动扫描功能, 需再写使能 start 才可以重新启动自动扫描) | 0x00000000 |

## LSADC\_CTRL9

LSADC\_CTRL9 为转换结果精度寄存器。

Offset Address: 0x0024    Total Reset Value: 0x0000\_0FFF

| Bits    | Access | Name             | Description                                                                            | Reset    |
|---------|--------|------------------|----------------------------------------------------------------------------------------|----------|
| [31:12] | -      | reserved         | 保留。                                                                                    | 0x000000 |
| [11:0]  | RW     | lsadc_active_bit | LSADC 转换结果精度。<br>12'b111111111111 表示 12 比特精度;<br>12'b111111111110 表示 11 比特精度;<br>..... | 0xFFF    |

| Bits | Access | Name | Description                | Reset |
|------|--------|------|----------------------------|-------|
|      |        |      | 12'b100000000000 表示 1 比特精度 |       |

## LSADC\_CTRL10

LSADC\_CTRL10 为 LSADC\_ZERO 寄存器。

Offset Address: 0x0028    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name       | Description       | Reset    |
|---------|--------|------------|-------------------|----------|
| [31:12] | -      | reserved   | 保留。               | 0x000000 |
| [11:0]  | RW     | lsadc_zero | 键盘无键按下时 LSADC 的值。 | 0x000    |

## LSADC\_CTRL11

LSADC\_CTRL11 为 LSADC 数据保持寄存器 0。

Offset Address: 0x002C    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name           | Description     | Reset    |
|---------|--------|----------------|-----------------|----------|
| [31:12] | -      | reserved       | 保留。             | 0x000000 |
| [11:0]  | RO     | lsadc_data_in0 | LSADC 通道 0 扫描值。 | 0x000    |

## LSADC\_CTRL12

LSADC\_CTRL12 为 LSADC 通道 1 数据寄存器。

Offset Address: 0x0030    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name           | Description     | Reset    |
|---------|--------|----------------|-----------------|----------|
| [31:12] | -      | reserved       | 保留。             | 0x000000 |
| [11:0]  | RO     | lsadc_data_in1 | LSADC 通道 1 扫描值。 | 0x000    |

## LSADC\_CTRL13

LSADC\_CTRL13 为 LSADC 通道 2 数据寄存器。

Offset Address: 0x0034    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name           | Description     | Reset    |
|---------|--------|----------------|-----------------|----------|
| [31:12] | -      | reserved       | 保留。             | 0x000000 |
| [11:0]  | RO     | lsadc_data_in2 | LSADC 通道 2 扫描值。 | 0x000    |

## LSADC\_CTRL15

LSADC\_CTRL15 为 LSADC 配置寄存器。

Offset Address: 0x003C    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name                | Description                                | Reset    |
|---------|--------|---------------------|--------------------------------------------|----------|
| [31]    | -      | power_downm_act_bit | 断电有效信号                                     | 0x000000 |
| [30:10] | RW     | reserved            | 保留。                                        | 0x000000 |
| [9]     | RW     | rg_auxadc_en        | adc en signal, ADC 使能信号<br>0: 不使能<br>1: 使能 | 0x000    |
| [8:4]   | RW     | rg_spl_num          | Sample time adjust signal<br>采样时间配置        | 0x000    |
| [3:0]   | RW     | rg_auxadc_bias_cal  | ADC BIAS select signal<br>电流控制信号, 默认为 0    | 0x000    |

## LSADC\_CTRL20

LSADC\_CTRL20 为 LSADC 通道 0 平均算法数据寄存器。

Offset Address: 0x0050    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name                   | Description                                  | Reset  |
|---------|--------|------------------------|----------------------------------------------|--------|
| [31:18] | -      | reserved               | 保留。                                          | 0x0000 |
| [17]    | RO     | lsadc_equ_vaule_vald_0 | LSADC 通道 0 采样平均算法结果有效指示。<br>0: 无效;<br>1: 有效。 | 0x0    |

|         |    |                   |                      |       |
|---------|----|-------------------|----------------------|-------|
| [16:12] | -  | reserved          | 保留。                  | 0x0   |
| [11:0]  | RO | lsadc_equ_vaule_0 | LSADC 通道 0 采样平均算法结果。 | 0x000 |

## LSADC\_CTRL21

LSADC\_CTRL21 为 LSADC 通道 1 平均算法数据。

Offset Address: 0x0054    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name                        | Description                                  | Reset  |
|---------|--------|-----------------------------|----------------------------------------------|--------|
| [31:18] | -      | reserved                    | 保留。                                          | 0x0000 |
| [17]    | RO     | lsadc_equ_vaule_v<br>aild_1 | LSADC 通道 1 采样平均算法结果有效指示。<br>0: 无效;<br>1: 有效。 | 0x0    |
| [16:12] | -      | reserved                    | 保留。                                          | 0x0    |
| [11:0]  | RO     | lsadc_equ_vaule_1           | LSADC 通道 1 采样平均算法结果。                         | 0x000  |

## LSADC\_CTRL22

LSADC\_CTRL21 为 LSADC 通道 2 平均算法数据。

Offset Address: 0x0054    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name                        | Description                                  | Reset  |
|---------|--------|-----------------------------|----------------------------------------------|--------|
| [31:18] | -      | reserved                    | 保留。                                          | 0x0000 |
| [17]    | RO     | lsadc_equ_vaule_v<br>aild_2 | LSADC 通道 2 采样平均算法结果有效指示。<br>0: 无效;<br>1: 有效。 | 0x0    |
| [16:12] | -      | reserved                    | 保留。                                          | 0x0    |
| [11:0]  | RO     | lsadc_equ_vaule_2           | LSADC 通道 2 采样平均算法结果。                         | 0x000  |

## 2.8 PWM

### 2.8.1 概述

芯片的 PWM(pulse-width modulation)可以提供脉宽调制信号输出。

### 2.8.2 特点

对于每组 PWM 输出：

- 24MHz 工作时钟源。
- 内部有 26bit 计数器，输出周期可配置。支持最高 12MHz (24MHz/2) 输出，最低约 0.36Hz (24MHz/ 67108863) 输出。
- 内部有 26bit 计数器，输出高电平拍数可配置。
- 内部 10bit 计数器，输出脉冲个数 (最大 1023 个) 可配置。可以工作在固定个数输出模式和一直输出模式：
  - 当 pwmn\_keep(pwmn 表示第 n 路 pwm)配置为 0 时，pwmn 输出固定数目的方波，该数目由 pwmn\_num (pwmn 表示第 n 路 pwm) 寄存器控制。
  - 当 pwmn\_keep(pwmn 表示第 n 路 pwm)配置为 1 时，pwmn 一直输出方波。

### 2.8.3 工作方式

PWM 内部工作于 24MHz，PWM (以 PWM0 为例) 输出配置流程如下：

步骤 1 选择合适的时钟源，通过计算得到需要的周期数和高电平拍数。

步骤 2 将对应数据写入 PWM0\_CFG\_CYCLE、PWM0\_CFG\_HLINT、PWM0\_CFG\_PULSE 寄存器。

步骤 3 对 PWM0\_CTRL bit[0]写入 1，使能 PWM 输出。

-----结束

例如：需要输出 1 个频率为 3kHz，高电平占 72.5% (即占空比)，脉冲个数为 10 的波形。

周期数配置为  $24\text{MHz} / 3\text{kHz} = 8000$ ，四舍五入后为 8000，十六进制为 0x0001F40。高电平数配置为  $8000 \times 72.5\%$ （占空比） $\approx 5800$ ，四舍五入后为 5800，十六进制为 0x00016A8。

按如下步骤进行寄存器操作，即可输出所需要的波形：

- 步骤 1 向写 PERI\_CRG111[9:8]写 0x2，选择 PWM 的时钟源为 24MHz，打开 PWM 时钟。
- 步骤 2 读取 PWM0\_STAT\_PULSE bit[20]，等待 bit[20]为 0（表示 PWM 空闲，可以输出方波）
- 步骤 3 向 PWM0\_CFG\_CYCLE 写入 0x0000\_1F40。
- 步骤 4 向 PWM0\_CFG\_HLINT 写入 0x0000\_16A8。
- 步骤 5 向 PWM0\_CFG\_PULSE 写入 0x0000\_000A。
- 步骤 6 向 PWM0\_CTRL 写入 0x4。（以下步骤可以省略，只是为了验证正在输出的方波是否按配置输出）
- 步骤 7 读取 PWM0\_STAT\_PULSE bit[20]，等待 bit20 位为 1（表示 PWM 正在输出方波）
- 步骤 8 读取 PWM0\_STAT\_CYCLE 和 0x0000\_1F40 进行校验。
- 步骤 9 读取 PWM0\_STAT\_HLINT 和 0x0000\_16A8 进行校验。
- 步骤 10 读取 PWM0\_STAT\_PULSE bit[19:10]和 0x0A 进行校验(当 PWM0\_STAT\_PULSE bit[20]为 1 时表示 PWM 正在输出方波，当该位为 0 时表示已经输出完设定的方波数目)。

----结束

## 2.8.4 PWM 寄存器概览

各路 PWM 寄存器的基地址如表 2-20 所示。

表2-20 PWMn 的寄存器基地址表

| PWM 路数 | 基地址         |
|--------|-------------|
| PWM0   | 0x1208_0000 |
| PWM1   | 0x1208_0100 |
| PWM2   | 0x1208_0200 |
| PWM3   | 0x1208_0300 |



| PWM 路数 | 基地址         |
|--------|-------------|
| PWM4   | 0x1208_0400 |
| PWM5   | 0x1208_0500 |
| PWM6   | 0x1208_0600 |
| PWM7   | 0x1208_0700 |
| PWM8   | 0x1208_0800 |
| PWM9   | 0x1208_0900 |
| PWM10  | 0x1208_0a00 |

以 PWM0 寄存器的描述为例，寄存器概览如表 2-21 所示。

表2-21 PWM0 寄存器概览

| 偏移地址   | 名称              | 描述               |
|--------|-----------------|------------------|
| 0x0000 | PWM0_CTRL       | PWM0 的控制寄存器      |
| 0x0010 | PWM0_CFG_PULSE  | PWM0 的方波数配置寄存器   |
| 0x0014 | PWM0_STAT_PULSE | PWM0 的方波数状态寄存器   |
| 0x0020 | PWM0_CFG_HLINT  | PWM0 的高电平拍数配置寄存器 |
| 0x0024 | PWM0_STAT_PULSE | PWM0 的高电平拍数状态寄存器 |
| 0x0030 | PWM0_CFG_CYCLE  | PWM0 的周期数配置寄存器   |
| 0x0034 | PWM0_STAT_CYCLE | PWM0 的技术周期状态寄存器  |
| 0x0000 | PWM0_CTRL       | PWM0 的控制寄存器      |

### 2.8.5 PWM 寄存器描述

#### PWM0\_CTRL

PWM0\_CTRL 为 PWM0 的控制寄存器。

Offset Address: 0x0000    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name        | Description                                        | Reset      |
|--------|--------|-------------|----------------------------------------------------|------------|
| [31:3] | -      | reserved    | 保留。                                                | 0x00000000 |
| [2]    | RW     | pwm0_enable | PWM 使能控制。<br>0: PWM0 模块关闭;<br>1: PWM0 模块使能。        | 0x0        |
| [1]    | RW     | pwm0_inv    | PWM 输出正反相控制。<br>0: PWM0 正常输出方波;<br>1: PWM0 反相输出方波。 | 0x0        |
| [0]    | RW     | pwm0_keep   | PWM 输出模式。<br>0: PWM0 输出固定数目的方波;<br>1: PWM0 一直输出方波。 | 0x0        |

## PWM0\_CFG\_PULSE

PWM0\_CFG\_PULSE 为 PWM0 的方波数配置寄存器。

Offset Address: 0x0010    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name     | Description   | Reset    |
|---------|--------|----------|---------------|----------|
| [31:10] | -      | reserved | 保留。           | 0x000000 |
| [9:0]   | RW     | pwm0_num | PWM0 输出方波的数目。 | 0x000    |

## PWM0\_STAT\_PULSE

PWM0\_STAT\_PULSE 为 PWM0 的方波数状态寄存器。

Offset Address: 0x0014    Total Reset Value: 0x0000\_0000

| Bits    | Access | Name         | Description                            | Reset |
|---------|--------|--------------|----------------------------------------|-------|
| [31:22] | -      | reserved     | 保留。                                    | 0x000 |
| [21]    | RO     | pwm0_keep_st | PWM0 的内部模块采用的输出方波的模式。<br>0: 输出固定数目的方波; | 0x0   |

|         |    |                |                                                           |       |
|---------|----|----------------|-----------------------------------------------------------|-------|
|         |    |                | 1: 一直输出方波。                                                |       |
| [20]    | RO | pwm0_busy      | PWM0 模块的工作状态。<br>0: 空闲, 输出完成;<br>1: 正在输出方波。               | 0x0   |
| [19:10] | RO | pwm0_period_st | PWM0 的内部模块采用的输出方波数目。                                      | 0x000 |
| [9:0]   | RO | pwm0_cnt_st    | PWM0 模块还需要输出的方波数目。只有当 pwm0_busy==1,pwm0_keep_st==0 时才有意义。 | 0x000 |

## PWM0\_CFG\_HLINT

PWM0\_CFG\_HLINT 为 PWM0 的高电平拍数配置寄存器。

Offset Address: 0x0020    Total Reset Value: 0x0000\_00C7

| Bits   | Access | Name      | Description                                                 | Reset      |
|--------|--------|-----------|-------------------------------------------------------------|------------|
| [31:0] | RW     | pwm0_duty | PWM0 的高电平拍数, 如果大于等于周期数, 则输出一直为高电平。<br><b>注意: 要求配置值必须≥1。</b> | 0x000_00C7 |

## PWM0\_STAT\_HLINT

PWM0\_STAT\_HLINT 为 PWM0 的高电平拍数状态寄存器。

Offset Address: 0x0024    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name         | Description         | Reset    |
|--------|--------|--------------|---------------------|----------|
| [31:0] | RO     | pwm0_duty_st | PWM0 的内部模块采用的高电平拍数。 | 0x000000 |

## PWM0\_CFG\_CYCLE

PWM0\_CFG\_CYCLE 为 PWM0 的周期数配置寄存器。

Offset Address: 0x0030    Total Reset Value: 0x0000\_018F

| Bits   | Access | Name        | Description                                                                                                      | Reset      |
|--------|--------|-------------|------------------------------------------------------------------------------------------------------------------|------------|
| [31:0] | RW     | pwm0_period | PWM0 的周期数。<br> 说明<br>要求配置值必须≥2。 | 0x000_018F |

## PWM0\_STAT\_CYCLE

PWM0\_STAT\_CYCLE 为 PWM0 的计数周期状态寄存器。

Offset Address: 0x0034    Total Reset Value: 0x0000\_0000

| Bits   | Access | Name           | Description         | Reset    |
|--------|--------|----------------|---------------------|----------|
| [31:0] | RO     | pwm0_period_st | PWM0 的内部模块采用的计数周期数。 | 0x000000 |