



Hi3516DV300 专业型 Smart IP Camera SoC

用户指南

文档版本 00B06

发布日期 2019-03-12

版权所有 © 深圳市海思半导体有限公司 2019。保留一切权利。

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

商标声明



HISILICON、海思和其他海思商标均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

深圳市海思半导体有限公司

地址：深圳市龙岗区坂田华为基地华为电气生产中心 邮编：518129

网址：<http://www.hisilicon.com>

客户服务电话：+86-755-28788858

客户服务传真：+86-755-28357515

客户服务邮箱：support@hisilicon.com



目 录

前 言.....	1



前言

概述

本文档介绍了 Hi3516DV300 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 Hi35xxVxxx 均为 Hi3516DV300。

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
Hi3516D	V300

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
 危险	以本标志开始的文本表示有高度潜在危险，如果不能避免，会导致人员死亡或严重伤害。



符号	说明
 警告	以本标志开始的文本表示有中度或低度潜在危险，如果不能避免，可能导致人员轻微或中等伤害。
 注意	以本标志开始的文本表示有潜在风险，如果忽视这些文本，可能导致设备或器件损坏、数据丢失、设备性能降低或不可预知的结果。
 窍门	以本标志开始的文本能帮助您解决某个问题或节省您的时间。
 说明	以本标志开始的文本是正文的附加信息，是对正文的强调和补充。

通用格式约定

格式	说明
宋体	正文采用宋体表示。
黑体	一级、二级、三级标题采用黑体。
楷体	警告、提示等内容一律用楷体，并且在内容前后增加线条与正文隔离。
“Terminal Display” 格式	“Terminal Display” 格式表示屏幕输出信息。此外，屏幕输出信息中夹杂的用户从终端输入的信息采用加粗字体表示。

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。



类型	说明	类型	说明
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
2019-03-12	00B06	第 11 章 音频接口 11.2.4 小节，AUDIO_CTRL_REG_1 涉及修改 第 13 章 安全模块 13.4.1.5 小节和 13.4.1.6 小节，涉及更新。 13.4.3 小节，寄存器 OTP_USER_FLAG_VALUE， OTP_USER_FLAG_INDEX，OTP_USER_LOCK_STA0 涉



修订日期	版本	修订说明
		及更新。
2019-01-15	00B05	第 2 章 硬件 2.5.1 小节涉及修改 2.5.3 小节，修改表 2-12~表 2-14 第 3 章 系统 添加 DIE_ID0 ~ DIE_ID5 寄存器 9.2.6 小节，添加 V0_HIPP_CSC_CTRL~ V0_HIPP_CSC_MAX_C 寄存器；G0_HIPP_CSC_CTRL~ G0_HIPP_CSC_MAX_C 寄存器 第 13 章 安全模块 13.4.1.6 小节涉及修改 13.4.3 小节，OTP_USER_REV_ADDR 涉及修改
2018-11-20	00B04	第 2 章 硬件 2.5.2 小节，表 2-11 涉及修改 第 3 章 系统 3.2.6 小节，PERI_CRG40、PERI_CRG61、PERI_CRG63 和 PERI_CRG64 涉及修改 3.10.2.3 小节，PWR_CTRL0 涉及修改 第 12 章 外围设备 12.7.1 小节涉及修改
2018-10-15	00B03	第 3 次临时版本发布 第 1 章 概述 1.3.4、1.3.13 小节涉及修改 第 3 章 系统 VPSS 频点涉及修改
2018-09-06	00B02	第 2 次临时版本发布。 新增视频解码相关内容 支持单 sensor 改为支持双 sensor （1、3、6、9、10 章节涉及修改） 第 2 章 硬件 2.5.3 小节，表 2-13 的 DVDD3318_FLASH 涉及修改 2.5.5 小节，表 2-15、表 2-16 涉及修改 2.6.11 小节，表 2-41 涉及更新。 第 9 章 视频接口 9.4.4 小节，涉及修改。 9.4.8 小节，寄存器 PLL_SET1、PLL_SET4 涉及修改。



修订日期	版本	修订说明
		第 12 章 外围 12.5.4 小节，MMC_FIFOTH 涉及更新。
2018-07-16	00B01	第 1 次临时版本发布。



目 录

1 产品概述.....	1-1
1.1 概述.....	1-1
1.2 应用场景.....	1-1
1.2.1 Hi3516DV300 HD IP 摄像机解决方案.....	1-1
1.3 架构.....	1-2
1.3.1 概述	1-2
1.3.2 处理器内核.....	1-3
1.3.3 视频编码	1-3
1.3.4 视频编解码处理性能.....	1-3
1.3.5 智能视频分析.....	1-4
1.3.6 视频与图形处理.....	1-4
1.3.7 ISP.....	1-4
1.3.8 音频编解码.....	1-4
1.3.9 安全	1-4
1.3.10 视频接口.....	1-5
1.3.11 音频接口.....	1-5
1.3.12 外围接口.....	1-5
1.3.13 外部存储器接口.....	1-6
1.3.14 启动	1-6
1.3.15 SDK	1-6
1.3.16 芯片物理规格.....	1-6
1.4 启动和升级模式.....	1-7
1.4.1 概述	1-7
1.4.2 启动模式和对应的信号锁存值对应关系	1-7
1.4.3 安全启动	1-8
1.5 地址空间映射.....	1-8



插图目录

图 1-1 Hi3516DV300 HD IP 摄像机解决方案.....	1-2
图 1-2 Hi3516DV300 芯片逻辑框图.....	1-3



表格目录

表 1-1 启动模式	1-7
表 1-2 地址空间映射表.....	1-8



1 产品概述

1.1 概述

Hi3516DV300 作为新一代行业专用 Smart HD IP 摄像机 SOC，集成新一代 ISP、业界最新的 H.265 视频压缩编码器，同时集成高性能 NNIE 引擎，使得 Hi3516DV300 在低码率、高画质、智能处理和分析、低功耗等方面引领行业水平。集成 POR、RTC、Audio Codec 以及待机唤醒电路，为客户极大的降低了 ebom 成本。且与海思 DVR/NVR 芯片相似的接口设计，能方便支撑客户产品开发和量产。

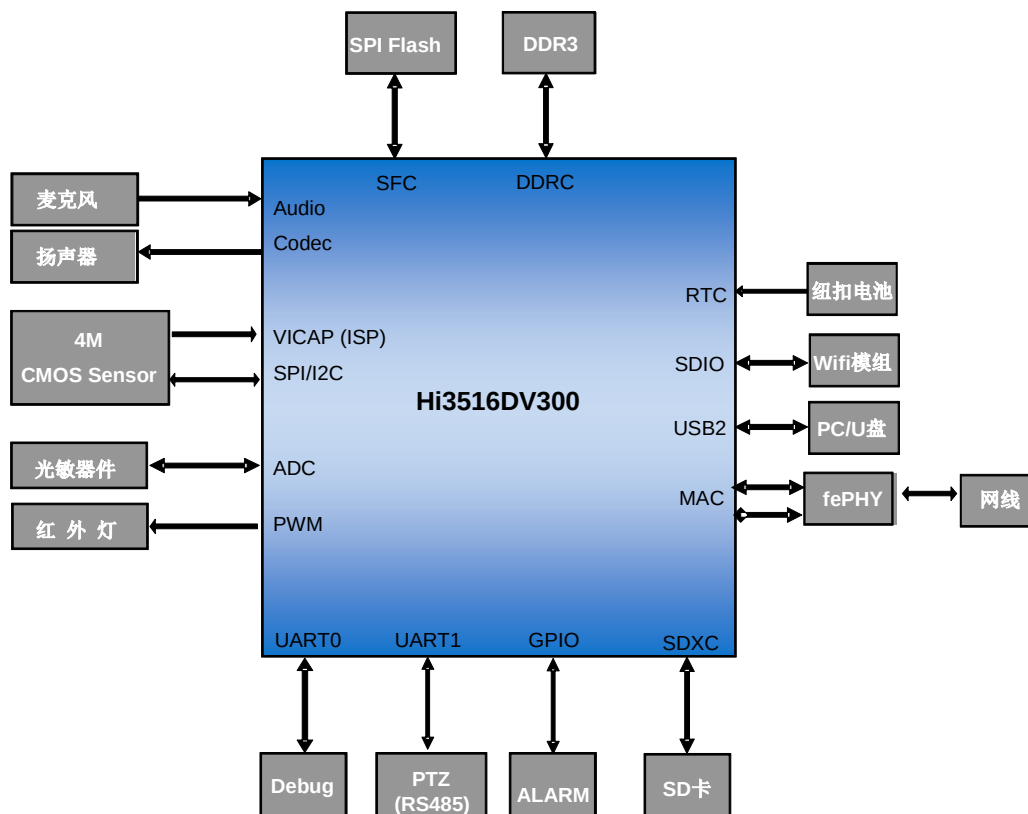
1.2 应用场景

1.2.1 Hi3516DV300 HD IP 摄像机解决方案

Hi3516DV300 HD IP 摄像机解决方案如[图 1-1](#) 所示。



图1-1 Hi3516DV300 HD IP 摄像机解决方案



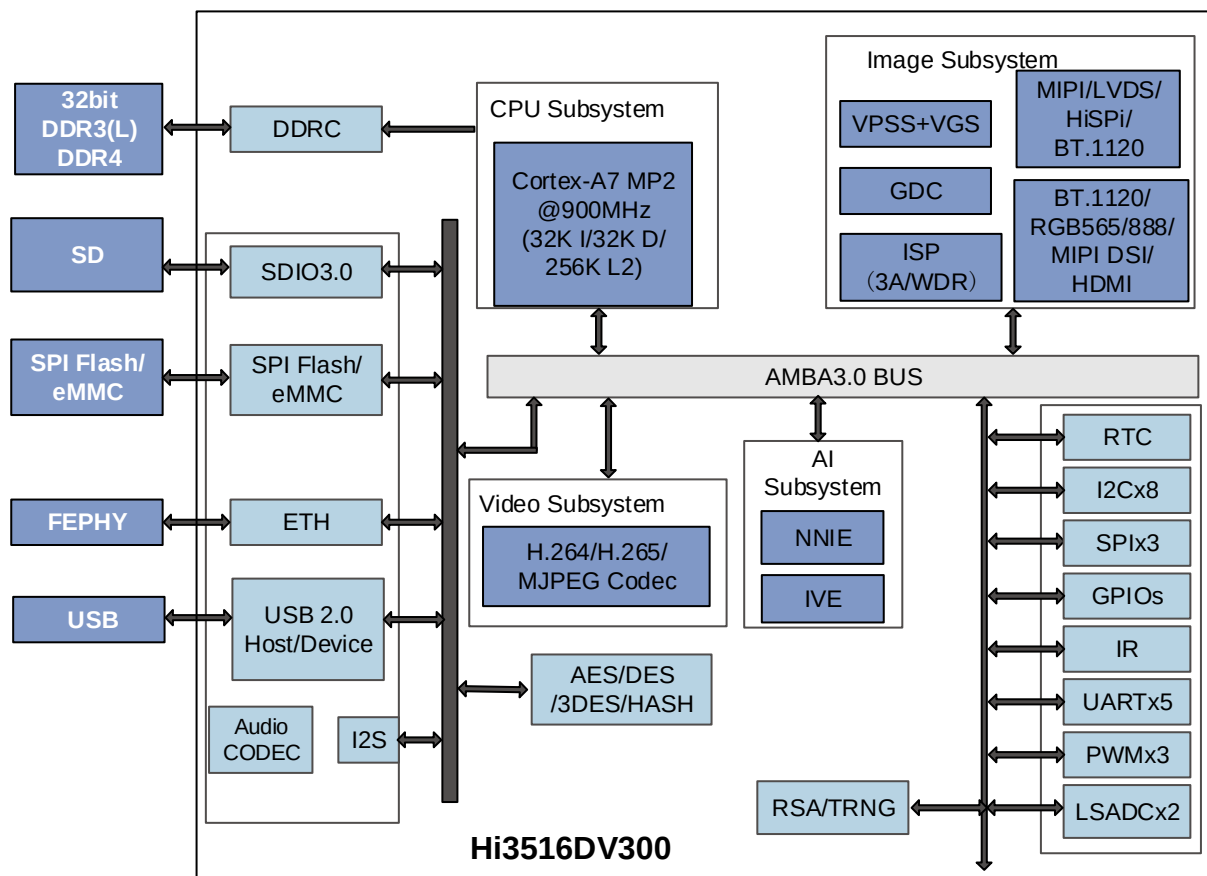
1.3 架构

1.3.1 概述

Hi3516DV300 芯片逻辑框图如图 1-2 所示。



图1-2 Hi3516DV300 芯片逻辑框图



1.3.2 处理器内核

- 双核 ARM Cortex-A7@ 900MHz, 32KB I-Cache, 32KB D-Cache, 256KB L2 Cache
- 支持 NEON 加速, 集成 FPU 处理单元

1.3.3 视频编解码

- H.264 BP/MP/HP
- H.265 Main Profile
- 支持 I/P 帧, 支持 Smart P 参考
- MJPEG/JPEG Baseline 编码

1.3.4 视频编解码处理性能

- H.264/H.265 编解码最大宽度 2688, 最大分辨率 2688x1944, 仅支持解码自身编码码流
- H.264/H.265 多码流实时编解码能力:
 - 2688x1536@30fps+720x480@30fps+360x240@30fps 编码



- 2688x1944@20fps+720x480@20fps+360x240@20fps 编码
- 1920x1080@30fps+720x480@30fps+1920x1080@30fps 解码
- JPEG 编码、解码能力：
 - 16M(4608x3456) @10fps
- 支持 CBR/VBR/FIXQP/AVBR/QPMAP 五种码率控制模式
- 输出码率最高 50Mbps
- 支持 8 个感兴趣区域 (ROI) 编码

1.3.5 智能视频分析

- 集成神经网络加速引擎，处理性能达 1.0Tops
- 集成智能计算加速引擎 (含跟踪、人脸校正)

1.3.6 视频与图形处理

- 3D 去噪、图像增强、动态对比度增强处理功能
- 视频、图形输出抗闪烁处理
- 视频、图形 1/15~16x 缩放功能
- 视频图形叠加
- 图像 90、180、270 度旋转
- 图像 Mirror、Flip 功能
- 8 个区域的编码前处理 OSD 叠加

1.3.7 ISP

- 3A (AE/AWB/AF)，支持第三方 3A 算法
- 固定模式噪声消除、坏点校正
- 镜头阴影校正、镜头畸变校正、紫边校正
- 方向自适应 demosaic
- gamma 校正、动态对比度增强、色彩管理和增强
- 区域自适应去雾
- 多级降噪 (BayerNR、3DNR)、细节增强及锐化增强
- Local Tone mapping
- Sensor built-in WDR 和 2F WDR(line-base/frame-base/DCG)
- 支持 6-DOF 防抖 (基于视频或陀螺仪信息)
- 提供 PC 端 ISP tuning tools

1.3.8 音频编解码

- 通过软件实现多协议语音编解码 (G.711、G.726、ADPCM)
- 支持音频 3A (AEC、ANR、AGC) 功能



1.3.9 安全

- 支持安全启动
- 硬件实现内存隔离
- 硬件实现 AES/DES/3DES/RSA 多种加解密算法
- 硬件实现 HASH(SHA1/SHA256/HMAC_SHA/HMAC_SHA256)
- 集成硬件随机数发生器
- 集成 8K bit OTP 存储空间

1.3.10 视频接口

- 输入
 - 支持两路输入
 - 第一路支持输入最大宽度 2688，最大分辨率 2688x1944
 - 第二路支持输入最大宽度 2048，最大分辨率 2048x1536
 - 支持 8/10/12/14 bit RGB Bayer DC 时序视频输入
 - 支持 BT.601、BT.656、BT.1120 视频输入接口
 - 支持 MIPI、LVDS/Sub-LVDS、HiSPi 接口
 - 支持与 SONY、ON、OmniVision、Panasonic 等主流高清 CMOS sensor 对接
 - 兼容多种 sensor 并行/差分接口电气特性
 - 提供可编程 sensor 时钟输出
- 输出
 - 支持 1 个 BT.656/BT.1120 视频输出接口
 - 支持 6/8bit 串行、16/18/24bit RGB 并行 LCD 输出
 - 支持 4Lane Mipi-DSI 接口输出
 - 支持 HDMI 1.4 输出（最大分辨率 1080p60）

1.3.11 音频接口

- 集成 Audio codec，支持 16bit 语音输入和输出
- 支持单声道 mic 差分输入，降低底噪
- 支持单端双声道输入
- 支持 I²S 接口，支持对接外部 Audio codec

1.3.12 外围接口

- 集成 POR
- 集成高精度 RTC
- 集成 2 通道 LSADC
- 若干 I²C、SPI、UART 接口
- 3 个 PWM 接口



- 2 个 SDIO3.0 接口，支持 3.3/1.8V 电平
其中一个支持 SD3.0 卡
- 1 个 USB 2.0 Host/Device 接口
- 支持 RMII 模式；支持 TSO 网络加速；支持 10/100Mbps 全双工或半双工模式，提供 PHY 时钟输出

1.3.13 外部存储器接口

- SDRAM 接口
 - 32bit DDR3(L)/DDR4，最大容量支持 16Gbit
 - 速率最高 1800Mbps
- SPI Nor Flash 接口
 - 支持 1、2、4 线模式
 - 最大容量支持 256MB
- SPI Nand Flash 接口
 - 最大 ECC 纠错能力 24bit/1KB
 - 最大容量支持 1GB
- eMMC4.5 接口
4bit 数据位宽

1.3.14 启动

可从 SPI Nor Flash、SPI Nand Flash 或 eMMC 启动

1.3.15 SDK

- 提供基于 Linux-4.9 SDK 包
- 提供 H.264 的高性能 PC 解码库
- 提供 H.265 的高性能 PC、Android、iOS 解码库

1.3.16 芯片物理规格

- 功耗
4M(2688 x 1536)P30, TBD 典型功耗
- 工作电压
 - 内核电压为 0.9V
 - IO 电压为 3.3V (+/-10%)
 - DDR3(L)/4 SDRAM 接口电压为 1.5(1.35)/1.2V
- 封装
14mm x 14mm, 367pin 0.65 管脚间距, TFBGA RoHS



1.4 启动和升级模式

1.4.1 概述

芯片中内置启动 ROM (BOOTROM)，芯片复位撤销后由 BOOTROM 开始执行启动引导程序。

启动介质选择

芯片中支持从以下两种存储介质启动：

- SPI Nand/Nor Flash 存储。
- eMMC 存储（只支持 eMMC 4bit 模式）。

启动介质的选择由 BOOT_SEL0 信号决定。

烧写（升级）模式

芯片还可以通过 SD 卡/USB/串口来对启动介质进行烧写(升级)。SD 卡、USB 升级模式由 UPDATE_MODE 信号决定，UART 烧写由 BOOT_SEL1 信号的值决定。



说明

当采用 SD 卡升级时，只能通过 SDIO0 接口上的 SD 卡升级。

1.4.2 启动模式和对应的信号锁存值对应关系

启动/升级模式由 BOOT_SEL1/BOOT_SEL0/UPDATE_MODE 信号来确定。



说明

- BOOT_SEL1 信号为 SENSOR0_RSTN 管脚的上电锁存值；
- BOOT_SEL0 信号为 MDCK 管脚的上电锁存值；
- UPDATE_MODE 信号为系统启动时 GPIO0_0 的状态，通常 GPIO0_0 可设计成按键，按下时状态为 0，表示升级模式；未按下时状态为 1，表示非升级模式；

通过系统控制寄存器 SYSSTAT[5:4]可以获取当前的 BOOT_SEL1/BOOT_SEL0 的状态，在芯片启动时通过查询 GPIO0_0 电平状态可以获取 UPDATE_MODE 的值。请参考“系统”章节的 SYSSTAT 寄存器中的描述。

启动模式和这些信号的关系如表 1-1 所示：

表1-1 启动模式

UPDATE_MODE	BOOT_SEL1	BOOT_SEL0	启动模式
1	0	0	从 SPI Nor/Nand Flash 启动。



1	0	1	从 eMMC Flash 启动。
1	1	0	串口烧写 SPI Nor/Nand Flash 镜像。
1	1	1	串口烧写 eMMC 镜像。
0	X	0	从 SDIO/USB 升级 SPI Nor/Nand Flash。
0	X	1	从 SDIO/USB 升级 eMMC Flash。

1.4.3 安全启动

Hi3516DV300 支持安全启动模式和非安全启动模式，安全启动模式和非安全启动模式由系统控制寄存器 PERISTAT [0] 的值来决定（此值来源于 OTP 中的安全启动标志位）。

- 在安全启动模式下，芯片将对 uBoot 进行验签。验签通过，则系统启动；验签不通过则系统不能启动。
- 非安全启动模式下，芯片不对 uBoot 进行验签。

1.5 地址空间映射

地址空间请参见表 1-2。

表1-2 地址空间映射表

起始地址	结束地址	功能	大小
0x0000_0000	0x03FF_FFFF	重映射时：BOOTROM 撤销重映射后：指向片内 RAM	64MB
0x0400_0000	0x0400_FFFF	BOOTROM 地址空间	64KB
0x0401_0000	0x0401_9AFFF	片内 RAM 地址空间	40KB
0x0401_B000	0x0FFF_FFFF	保留	-
0x1000_0000	0x1000_FFFF	FMC 寄存器	64KB
0x1001_0000	0x1001_FFFF	ETH 寄存器	64KB
0x1002_0000	0x1002_FFFF	SDIO1 寄存器	64KB
0x1003_0000	0x1003_FFFF	SCD 寄存器	64KB
0x1004_0000	0x1005_FFFF	保留	-
0x1006_0000	0x1006_FFFF	DMAC 寄存器	64KB
0x1007_0000	0x1007_FFFF	KLAD 寄存器	64KB
0x1008_0000	0x1008_FFFF	RSA0 寄存器	64KB



起始地址	结束地址	功能	大小
0x1009_0000	0x1009_FFFF	TRNG 寄存器	64KB
0x100A_0000	0x100A_FFFF	保留	-
0x100B_0000	0x100B_FFFF	OTP_USER 寄存器	64KB
0x100C_0000	0x100C_FFFF	SPACC 寄存器	64KB
0x100D_0000	0x100D_FFFF	RSA1 寄存器	64KB
0x100E_0000	0x100E_FFFF	USB CTRL 寄存器	64KB
0x100F_0000	0x100F_FFFF	SDIO0 寄存器	64KB
0x1010_0000	0x1010_FFFF	eMMC 寄存器	64KB
0x1011_0000	0x1011_FFFF	USB PHY 寄存器	64KB
0x1012_0000	0x10FE_FFFF	保留	-
0x10FF_0000	0x10FF_FFFF	IO 控制寄存器	64KB
0x1100_0000	0x1103_FFFF	VIPROC 寄存器	256KB
0x1104_0000	0x1104_FFFF	VPSS 寄存器	64KB
0x1105_0000	0x110F_FFFF	保留	-
0x1110_0000	0x1110_FFFF	NNIE 寄存器	64KB
0x1111_0000	0x1111_FFFF	GDC 寄存器	64KB
0x1112_0000	0x111E_FFFF	保留	-
0x111F_0000	0x111F_FFFF	IO 控制寄存器	64KB
0x1120_0000	0x1120_FFFF	GZIP 寄存器	64KB
0x1121_0000	0x1121_FFFF	TDE 寄存器	64KB
0x1122_0000	0x1122_FFFF	JPGE 寄存器	64KB
0x1123_0000	0x1123_FFFF	IVE 寄存器	64KB
0x1124_0000	0x1124_FFFF	VGS 寄存器	64KB
0x1125_0000	0x1125_FFFF	DDRT 寄存器	64KB
0x1126_0000	0x1126_FFFF	JPGD 寄存器	64KB
0x1127_0000	0x1127_FFFF	MIPI TX 寄存器	64KB
0x1128_0000	0x112E_FFFF	保留	-
0x112F_0000	0x112F_FFFF	IO 控制寄存器	64KB
0x1130_0000	0x1139_FFFF	VICAP 寄存器	640KB



起始地址	结束地址	功能	大小
0x113A_0000	0x113A_FFFF	MIPI RX 寄存器	64KB
0x113B_0000	0x113B_FFFF	AIAO 寄存器	64KB
0x113C_0000	0x113C_FFFF	AudioCodec 寄存器	64KB
0x113D_0000	0x113F_FFFF	保留	-
0x1140_0000	0x1140_FFFF	HDMI PHY 寄存器	64KB
0x1141_0000	0x1141_FFFF	保留	64KB
0x1142_0000	0x1143_FFFF	HDMI CTRL 寄存器	128KB
0x1144_0000	0x1147_FFFF	VDP 寄存器	256KB
0x1148_0000	0x114E_FFFF	保留	-
0x114F_0000	0x114F_FFFF	IO 控制寄存器	64KB
0x1150_0000	0x1150_FFFF	VEDU 寄存器	64KB
0x1151_FFFF	0x11FF_FFFF	保留	-
0x1200_0000	0x1200_0FFF	Timer0/Timer1 寄存器	4KB
0x1200_1000	0x1200_1FFF	Timer2/Timer3 寄存器	4KB
0x1200_2000	0x1200_2FFF	Timer4/Timer5 寄存器	4KB
0x1200_3000	0x1200_3FFF	Timer6/Timer7 寄存器	4KB
0x1201_0000	0x1201_FFFF	CRG 寄存器	64KB
0x1202_0000	0x1202_FFFF	系统控制寄存器	64KB
0x1203_0000	0x1203_FFFF	MISC 寄存器	64KB
0x1204_0000	0x1204_FFFF	SYSCNT 寄存器	64KB
0x1205_0000	0x1205_0FFF	WDG0 寄存器	4KB
0x1205_1000	0x1205_1FFF	WDG1 寄存器	4KB
0x1206_0000	0x1206_FFFF	DDRC 寄存器	64KB
0x1207_0000	0x1207_FFFF	PWM 寄存器	64KB
0x1208_0000	0x1208_FFFF	RTC 寄存器	64KB
0x1209_0000	0x1209_FFFF	PMC 寄存器	64KB
0x120A_0000	0x120A_0FFF	UART0 寄存器	4KB
0x120A_1000	0x120A_1FFF	UART1 寄存器	4KB
0x120A_2000	0x120A_2FFF	UART2 寄存器	4KB



起始地址	结束地址	功能	大小
0x120A_3000	0x120A_3FFF	UART3 寄存器	4KB
0x120A_4000	0x120A_4FFF	UART4 寄存器	4KB
0x120B_0000	0x120B_0FFF	I2C0 寄存器	4KB
0x120B_1000	0x120B_1FFF	I2C1 寄存器	4KB
0x120B_2000	0x120B_2FFF	I2C2 寄存器	4KB
0x120B_3000	0x120B_3FFF	I2C3 寄存器	4KB
0x120B_4000	0x120B_4FFF	I2C4 寄存器	4KB
0x120B_5000	0x120B_5FFF	I2C5 寄存器	4KB
0x120B_6000	0x120B_6FFF	I2C6 寄存器	4KB
0x120B_7000	0x120B_7FFF	I2C7 寄存器	4KB
0x120C_0000	0x120C_0FFF	SPI0 寄存器	4KB
0x120C_1000	0x120C_1FFF	SPI1 寄存器	4KB
0x120C_2000	0x120C_2FFF	SPI2 寄存器	4KB
0x120C_3000	0x120C_3FFF	3WIRE SPI 寄存器	4KB
0x120D_0000	0x120D_0FFF	GPIO0 寄存器	4KB
0x120D_1000	0x120D_1FFF	GPIO1 寄存器	4KB
0x120D_2000	0x120D_2FFF	GPIO2 寄存器	4KB
0x120D_3000	0x120D_3FFF	GPIO3 寄存器	4KB
0x120D_4000	0x120D_4FFF	GPIO4 寄存器	4KB
0x120D_5000	0x120D_5FFF	GPIO5 寄存器	4KB
0x120D_6000	0x120D_6FFF	GPIO6 寄存器	4KB
0x120D_7000	0x120D_7FFF	GPIO7 寄存器	4KB
0x120D_8000	0x120D_8FFF	GPIO8 寄存器	4KB
0x120D_9000	0x120D_9FFF	GPIO9 寄存器	4KB
0x120D_A000	0x120D_AFFF	GPIO10 寄存器	4KB
0x120D_B000	0x120D_BFFF	GPIO11 寄存器	4KB
0x120E_0000	0x120E_FFFF	LSADC 寄存器	64KB
0x120F_0000	0x120F_FFFF	IR 寄存器	64KB
0x1400_0000	0x14FF_FFFF	FMC 存储空间	16MB



起始地址	结束地址	功能	大小
0x8000_0000	0xFFFF_FFFF	DDR 存储空间	2GB



目 录

2 硬件特性	2-1
2.1 封装与管脚分布	2-1
2.1.1 封装	2-1
2.1.2 管脚分布	2-4
2.2 管脚信息描述	2-5
2.3 焊接工艺建议	2-5
2.3.1 无铅回流焊工艺参数要求	2-5
2.3.2 混合回流焊工艺参数要求	2-7
2.4 潮敏参数	2-8
2.4.1 海思产品防潮包装	2-8
2.4.2 存放与使用	2-9
2.4.3 重新烘烤	2-10
2.5 电性能参数	2-11
2.5.1 功耗参数	2-11
2.5.2 温度和热阻参数	2-12
2.5.3 工作条件	2-13
2.5.4 上下电顺序	2-16
2.5.5 DC/AC 电气参数	2-16
2.5.6 MIPI/LVDS Rx 电气参数	2-27
2.5.7 MIPI Tx 电气参数	2-29
2.5.8 SDIO 电气参数	2-31
2.5.9 AUDIO CODEC 电气参数	2-32
2.6 接口时序	2-33
2.6.1 DDR 接口时序	2-33
2.6.2 SPI FLASH 接口时序	2-36
2.6.3 Ethernet MAC 接口时序	2-39
2.6.4 VI 接口时序	2-42
2.6.5 VO 接口时序	2-42
2.6.6 AIAO 接口时序	2-44
2.6.7 I ² C 时序	2-45



2.6.8 SPI 接口时序.....2-47

2.6.9 MIPI Rx 接口时序 2-49

2.6.10 MIPI Tx 接口时序.....2-50

2.6.11 SDIO/MMC 接口时序.....2-51



插图目录

图 2-1 芯片封装顶视图.....	2-1
图 2-2 芯片封装底视图.....	2-2
图 2-3 芯片封装侧视图.....	2-2
图 2-4 Detail A 放大图	2-3
图 2-5 封装参数说明	2-4
图 2-6 无铅回流焊接工艺曲线.....	2-5
图 2-7 封装体测温示意图.....	2-7
图 2-8 干燥真空包装材料示意图.....	2-9
图 2-9 DDR3(L)/DDR4 命令和地址相对于 CK 的写操作时序图	2-33
图 2-10 DDR3(L)/DDR4 DQS 相对于 CK 的写操作时序图	2-34
图 2-11 DDR3(L)/DDR4 CK 相对于 DQS 的读操作时序图	2-34
图 2-12 DDR3(L)/DDR4 DQS 相对于 DQ 的读操作时序图	2-35
图 2-13 SFC 输入方向时序图-SDR 模式	2-37
图 2-14 SFC 输出方向时序图-SDR 模式	2-37
图 2-15 SFC 输入方向时序图-DDR 模式.....	2-38
图 2-16 SFC 输出方向时序图-DDR 模式.....	2-38
图 2-17 SFC CS 输出方向时序图（SDR 和 DDR 模式）	2-39
图 2-18 RMII 接口 100Mbit/s 接收时序	2-39
图 2-19 RMII 接口 100Mbit/s 发送时序	2-39
图 2-20 RMII 接口 10Mbit/s 接收时序	2-40
图 2-21 RMII 接口 10Mbit/s 发送时序	2-40
图 2-22 RMII 接口时序参数.....	2-40
图 2-23 MDIO 接口读时序.....	2-41
图 2-24 MDIO 接口写时序.....	2-41
图 2-25 MDIO 接口接收时序参数	2-41



图 2-26 VI 接口时序图	2-42
图 2-27 BT.656 接口时序.....	2-42
图 2-28 LCD 接口时序	2-43
图 2-29 BT.1120 接口时序	2-43
图 2-30 I ² S 接口接收时序图.....	2-44
图 2-31 I ² S 接口发送时序图.....	2-44
图 2-32 PCM 接口接收时序图	2-45
图 2-33 PCM 接口发送时序图	2-45
图 2-34 I ² C 传输时序图	2-46
图 2-35 SPICK 时序	2-47
图 2-36 SPI 主模式下接口时序（sph=1）	2-47
图 2-37 SPI 主模式下接口时序（sph=0）	2-48
图 2-38 MIPI Rx 接口时钟数据时序图.....	2-50
图 2-39 MIPI Tx 接口时钟数据时序图.....	2-51
图 2-40 单沿的 HS、DS 模式数据输入输出方向时序图.....	2-52
图 2-41 双沿 DDR50 模式的数据输入输出方向时序图	2-54
图 2-42 HS200 模式的数据命令输入输出方向时序图.....	2-55



表格目录

表 2-1 管脚数目统计表.....	2-4
表 2-2 无铅回流焊工艺参数.....	2-6
表 2-3 IPC/JEDEC 020D 中的无铅器件封装体耐温标准.....	2-6
表 2-4 混装回流焊工艺参数表.....	2-7
表 2-5 有铅器件封装体耐温标准.....	2-8
表 2-6 floor life 参照表	2-9
表 2-7 重新烘烤参考表.....	2-10
表 2-8 功耗参数	2-11
表 2-10 结温参数	2-13
表 2-11 热阻参数.....	2-13
表 2-12 使用 SVB 方案时，电源工作条件	2-13
表 2-13 常压电源工作条件.....	2-14
表 2-14 破坏性电压值	2-15
表 2-15 DC 电气参数表(DVDD33/DVDD3318_SENSOR/ DVDD3318_VI/ DVDD3318_UART1/ DVDD3318_SARADC/ DVDD3318_PC/ DVDD3318_FLASH =3.3V)	2-16
表 2-16 DC 电气参数表 (DVDD3318_SENSOR/ DVDD3318_VI/ DVDD3318_UART1/ DVDD3318_SARADC/ DVDD3318_PC/ DVDD3318_FLASH =1.8V， SDIO0_CARD_DETECT/SDIO0_CARD_POWER_EN 供电 1.8V)	2-21
表 2-17 DC 电气参数表 (VDDIO_DDR =1.5V，DDR3 模式)	2-25
表 2-18 AC 电气参数表 (VDDIO_DDR =1.5V，DDR3 模式)	2-25
表 2-19 DC 电气参数表 (VDDIO_DDR =1.35V，DDR3L 模式)	2-26
表 2-20 AC 电气参数表 (VDDIO_DDR =1.35V，DDR3L 模式)	2-26
表 2-21 AC 和 DC 电气参数表-数据线 (VDDIO_DDR =1.2V，DDR4 模式)	2-26
表 2-22 AC 和 DC 电气参数表-地址/命令线 (VDDIO_DDR =1.2V，DDR4 模式)	2-27
表 2-23 LVDS 差分 DC 电气参数表	2-28
表 2-24 MIPI DPHY HS (High Speed) DC 参数表	2-28



表 2-25 MIPI DPHY HS (High Speed) AC 参数表	2-29
表 2-26 MIPI DPHY LP (Low Power) DC 参数表	2-29
表 2-27 MIPI DPHY LP (Low Power) AC 参数表	2-29
表 2-28 MIPI Tx LP DC 参数表	2-29
表 2-29 MIPI Tx LP AC 参数表	2-30
表 2-30 MIPI Tx HS DC 参数表	2-30
表 2-31 MIPI Tx HS AC 参数表	2-31
表 2-32 SDIO 电性能参数表 3.3V(SDIO1 供电 3.3V 时).....	2-31
表 2-33 SDIO 电性能参数表 1.8V(SDIO0/1 供电 1.8V 时).....	2-32
表 2-34 总体指标表	2-32
表 2-35 DAC 主要指标表	2-32
表 2-36 ADC 主要指标表	2-32
表 2-37 MICBIAS 主要指标表	2-33
表 2-38 MICPGA 主要指标	2-33
表 2-39 DDR3 SDRAM 存储器参数表 (DDR3-1800)	2-35
表 2-40 DDR3 SDRAM 存储器参数表 (DDR3-1800)	2-35
表 2-41 DDR4 时钟参数表	2-36
表 2-42 DDR4 SDRAM 存储器参数表 (DDR4-1800)	2-36
表 2-43 SFC-SDR 模式输入方向时序参数表.....	2-37
表 2-44 SFC-SDR 输出方向时序参数表	2-37
表 2-45 SFC-DDR 输入方向时序参数表.....	2-38
表 2-46 SFC-DDR 输出方向时序参数表.....	2-38
表 2-47 SFC-SDR 和 SFC-DDR CS 输出方向时序参数表.....	2-39
表 2-48 RMII 接口时序参数说明.....	2-40
表 2-49 MDIO 接口时序参数	2-41
表 2-50 VI 接口时序参数表	2-42
表 2-51 BT.656 接口时序参数表.....	2-42
表 2-52 LCD 接口时序参数表.....	2-43
表 2-53 BT.1120 接口时序参数表	2-43
表 2-54 I ² S 接口时序参数表.....	2-44
表 2-55 PCM 接口时序参数表	2-45
表 2-56 I ² C 接口时序参数表	2-46



表 2-57 SPI 接口时序参数.....	2-48
表 2-58 MIPI Rx 时序参数表	2-50
表 2-59 MIPI Tx DATA-CLOCK 时序参数表.....	2-51
表 2-60 单沿 DS 模式 时序参数表.....	2-52
表 2-61 单沿 HS 模式 时序参数表.....	2-53
表 2-62 双沿 DDR50 模式时序参数表	2-54
表 2-63 HS200 模式输出时序参数表.....	2-55
表 2-64 HS200 模式输入时序参数表.....	2-55
表 2-65 SDIO/MMC 时序参数表	2-56



2 硬件特性

2.1 封装与管脚分布

2.1.1 封装

芯片采用 TFBGA 封装，封装尺寸为 14mmx14mm，管脚间距为 0.65mm。管脚总数为 367 个。

详细封装请参见图 2-1～图 2-4，封装尺寸参数请参见图 2-5。

图2-1 芯片封装顶视图

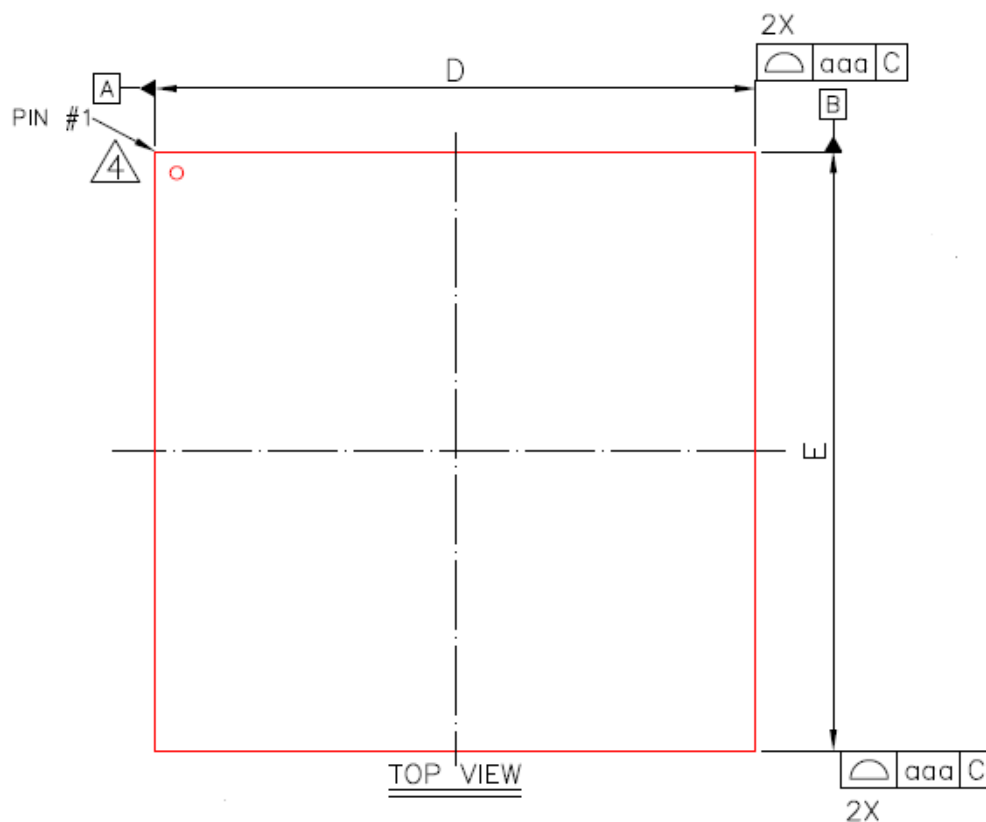


图2-2 芯片封装底视图

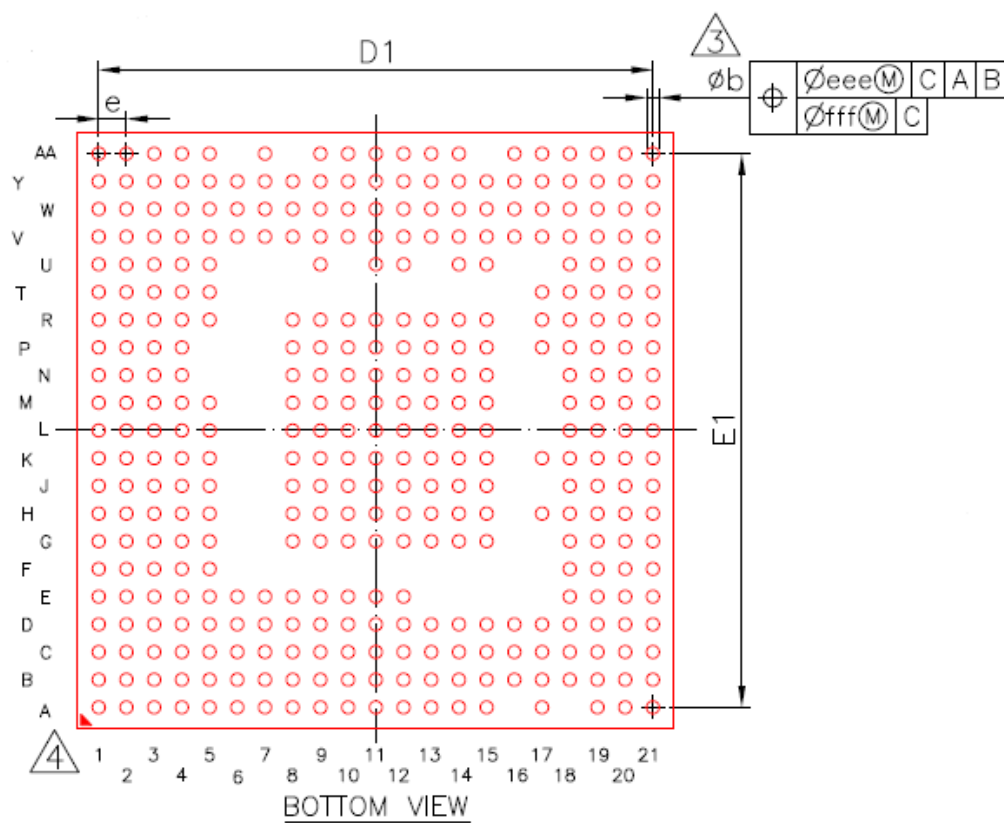


图2-3 芯片封装侧视图

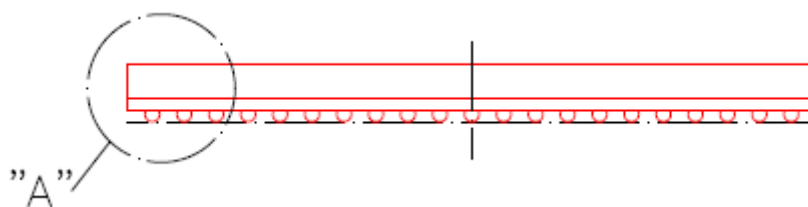




图2-4 Detail A 放大图

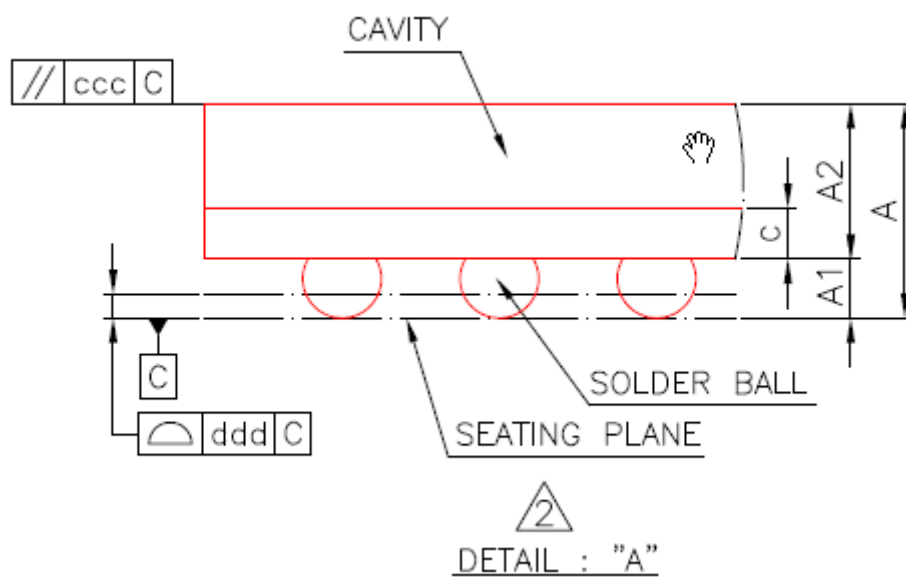




图2-5 封装参数说明

Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.10	1.17	1.24	0.043	0.046	0.049
A1	0.16	0.21	0.26	0.006	0.008	0.010
A2	0.91	0.96	1.01	0.036	0.038	0.040
c	0.22	0.26	0.30	0.009	0.010	0.012
D	13.90	14.00	14.10	0.547	0.551	0.555
E	13.90	14.00	14.10	0.547	0.551	0.555
D1	---	13.00	---	---	0.512	---
E1	---	13.00	---	---	0.512	---
e	---	0.65	---	---	0.026	---
b	0.25	0.30	0.35	0.010	0.012	0.014
aaa	0.15			0.006		
ccc	0.10			0.004		
ddd	0.08			0.003		
eee	0.15			0.006		
fff	0.08			0.003		
MD/ME	21 / 21					

NOTE :

1. CONTROLLING DIMENSION : MILLIMETER.
- ② PRIMARY DATUM C AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
- ③ DIMENSION b IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM C.
- ④ THE PATTERN OF PIN 1 FIDUCIAL IS FOR REFERENCE ONLY.
5. SPECIAL CHARACTERISTICS C CLASS: ccc, ddd
6. REFERENCE DOCUMENT : JEDEC PUBLICATION 95
DESIGN GUIDE 4.5
7. b IS SOLDER BALL DIAMETER = 0.30 mm (BEFORE REFLOW).
BGA PAD SOLDER MASK OPENING = 0.275 mm.
PKG BALL DIAMETER = 0.31 mm (AFTER REFLOW).

2.1.2 管脚分布

芯片的管脚有 367 个，管脚数目统计表如表 2-1 所示。

表2-1 管脚数目统计表

管脚类别	数量
I/O	206
数字电源	39
数字地	78



管脚类别	数量
其他/模拟电源	10
其他/模拟地	32
DDR 参考电源（VREF）	0
总计	367

2.2 管脚信息描述

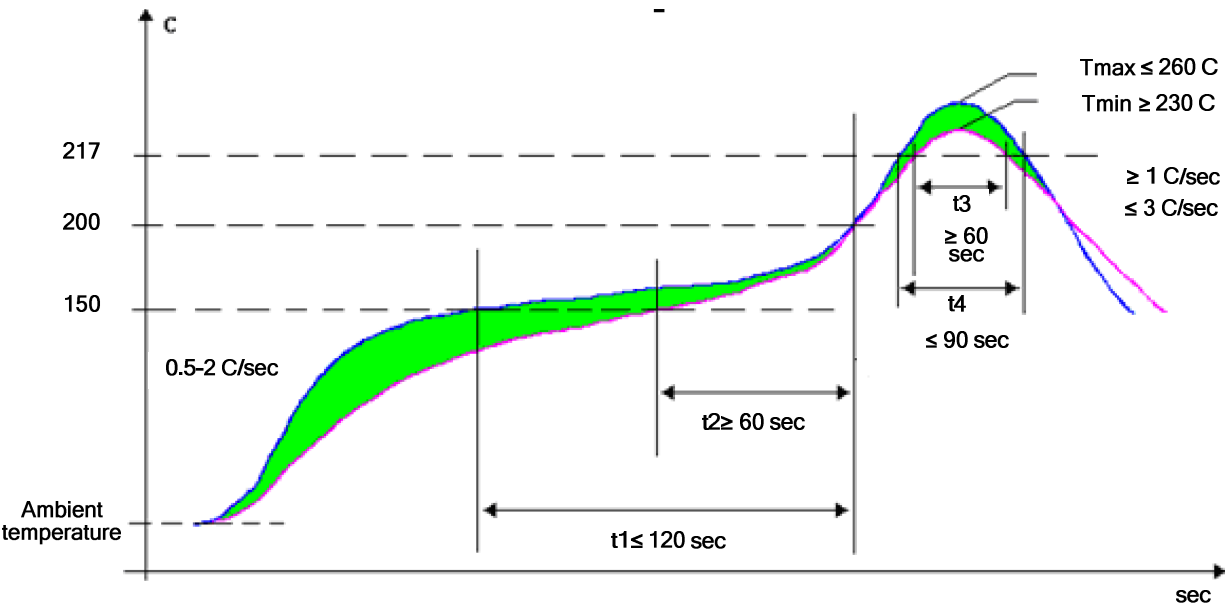
芯片管脚信息、管脚电源域信息、数字管脚默认状态描述请参考文档：《Hi35xxVxxx_PINOUT_CN》。

2.3 焊接工艺建议

2.3.1 无铅回流焊工艺参数要求

无铅回流焊接工艺曲线如图 2-6 所示。

图2-6 无铅回流焊接工艺曲线



无铅回流焊工艺参数如表 2-2 所示。



表2-2 无铅回流焊工艺参数

区域	时间	升温速率	峰值温度	降温速率
预热区（40~150℃）	60~150s	≤2.0℃/s	-	-
均温区（150~200℃）	60~120s	<1.0℃/s	-	-
回流区（>217℃）	60~90s	-	230-260℃	-
冷却区（Tmax~180℃）	-	-	-	1.0℃/s≤Slope≤4.0℃/s

说明

- 预热区：温度由 40℃~150℃，温度上升速率控制在 2℃/s 左右，该温区时间为 60~150s。
- 均温区：温度由 150℃~200℃，稳定缓慢升温，温度上升速率小于 1℃/s，且该区域时间控制在 60~120s（注意：该区域一定缓慢受热，否则易导致焊接不良）。
- 回流区：温度由 217℃~Tmax~217℃，整个区间时间控制在 60~90s。
- 冷却区：温度由 Tmax~180℃，温度下降速率最大不能超过 4℃/s。
- 温度从室温 25℃升温到 250℃时间不应该超过 6 分钟。
- 该回流焊曲线仅为推荐值，客户端需根据实际生产情况做相应调整。
- 回流时间以 60~90s 为目标，对于一些热容较大无法满足时间要求的单板可将回流时间放宽至 120s。封装体耐温标准参考 IPC/JEDEC J-STD-020D 标准，封装体测温方法参考 JEP 140 标准。

IPC/JEDEC J-STD-020D 标准，封装体测温方法按照 JEP 140 标准要求：IPC/JEDEC 020D 中的无铅器件封装体耐温标准如表 2-3 所示。

表2-3 IPC/JEDEC 020D 中的无铅器件封装体耐温标准

Package Thickness	Volume mm ³ <350	Volume mm ³ 350~2000	Volume mm ³ >2000
<1.6mm	260℃	260℃	260℃
1.6mm~2.5mm	260℃	250℃	245℃
>2.5mm	250℃	245℃	245℃

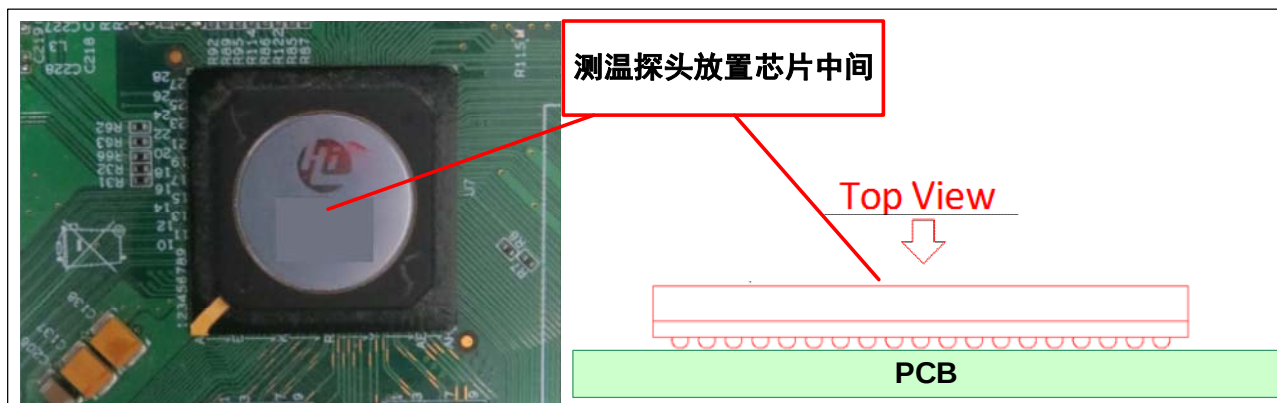
体积计算中不计入器件焊端（焊球，引脚）和外部散热片。

回流焊接工艺曲线测量方法：

JEP140 推荐：对于厚度较小的器件，测量封装体温度时，直接将热电偶贴放在器件表面，对于厚度较大的器件，在器件表面钻孔埋入热电偶进行测量。由于量化器件厚度的要求，推荐全部采用在封装体表面钻孔埋入热电偶的方式（特别薄器件，无法钻孔除外），如图 2-7 所示。



图2-7 封装体测温示意图



2.3.2 混合回流焊工艺参数要求

回流焊接过程中，如果出现器件混装现象，应首先保证无铅器件的正常焊接。具体要求如表 2-4 所示：

表2-4 混装回流焊工艺参数表

数值要求		有铅 BGA	无铅 BGA	其他器件
预热区（40~150℃）	时间	60-150Sec		
	升温斜率	<2.5℃/Sec		
均温区（150~183℃）	时间	30-90Sec		
	升温斜率	<1.0℃/Sec		
回流区（>183℃）	峰值温度	210~240℃	220~240℃	210~245℃
	时间	30~120s	60~120s	30~120s
冷却区（Tmax—150℃）	降温斜率	1.0℃/Sec≤Slope≤4.0℃/Sec		



注意

以上工艺参数要求均针对焊点温度。单板上焊点最热点和最冷点均需要满足以上规范要求。

曲线调制中，还需要满足单板上元器件的封装体耐温要求。封装体耐温标准按照 IPC/JEDEC J-STD-020D 标准，封装体测温方法按照 JEP 140 标准。

IPC/JEDEC J-STD-020D 标准，封装体测温方法按照 JEP 140 标准要求。



IPC/JEDEC 020D 中的有铅器件封装体耐温标准如表 2-5:

表2-5 有铅器件封装体耐温标准

Package Thickness	Temperature 1 (Package Volume < 350 mm ³ or 0.02 in. ³)	Temperature 1 (Package Volume ≥ 350 mm ³ or 0.02 in. ³)
< 2.5 mm (0.10 in.)	235°C (455°F)	220°C (428°F)
≥ 2.5 mm (0.10 in.)	220°C (428°F)	220°C (428°F)

体积计算中不计入器件焊端（焊球，引脚）和外部散热片。

JEP140 标准规定测量封装体温度方法同无铅工艺，请参考 2.3.1 无铅回流焊工艺参数要求详细说明

2.4 潮敏参数

本章节规定了 IC（潮敏产品）的使用原则，涉及到的术语解释如下：

- Floor life: 海思产品允许在车间保留的最长时间（环境<30℃/60% RH，在拆开防潮包装到 reflow 之前）
- Desiccant(干燥剂): 一种用于吸附潮气而保持干燥的材料
- Humidity Indicator Card (HIC): 湿度指示卡
- Moisture sensitivity level(MSL): 潮敏等级,本产品潮敏等级为 3 级。
- Moisture Barrier Bag (MBB): 防潮包装袋
- Solder Reflow: 回流焊
- Shelf Life: 防潮包装后正常存储时间

2.4.1 海思产品防潮包装

2.4.1.1 包装信息

干燥真空包装材料包含：

- 湿度指示卡（HIC）
- 防潮袋（MBB）
- 干燥剂



图2-8 干燥真空包装材料示意图



2.4.1.2 潮敏产品进料检验

在生产使用（SMT）之前，打开真空袋子后：

- 如果 HIC 的最大指示点已经变化（不是蓝色或土黄色），产品必须参照表 2-7 进行 rebake。
- 如果 HIC 中 10%RH dot 是蓝色或土黄色的，表示产品很干燥，可以仅仅更换防潮剂后真空封装。
- 如果 HIC 中 10%RH dot 不是蓝色或土黄色，5%RH dot 已经变红色或浅绿色，表示产品已经受潮，参考表 2-7 进行 rebake。

2.4.2 存放与使用

【存放环境】

建议产品真空包装存放，存放在<30°C/60% RH 下。

【shelf life】（防潮包装后正常存储时间）

存放环境<30°C/60% RH 下，真空包装存放，shelf life（存储期限）≤12 个月。

【floor life】

在环境条件<30°C/60%下，floor life 参照表 2-6 如下。

表2-6 floor life 参照表

MSL	Floor life(out of bag) at factory ambient≤30°C/60% RH or as stated
1	Unlimited at ≤30°C/85% RH
2	1 year



MSL	Floor life(out of bag) at factory ambient $\leq 30^{\circ}\text{C}/60\% \text{ RH}$ or as stated
2a	4 weeks
3	168 hours
4	72 hours
5	48 hours
5a	24 hours
6	Mandatory bake before use, must be reflowed within the time limit specified on the label

【潮敏产品的使用】

- 产品在 $\leq 30^{\circ}\text{C}/60\% \text{ RH}$ 下连续或累计暴露超过2个小时，建议进行 rebake 后再真空干燥包装。
- 产品在 $\leq 30^{\circ}\text{C}/60\% \text{ RH}$ 下暴露累计没有超过2个小时，可以不用 rebake，但要更换新的干燥剂，进行真空干燥包装。

本文没有提到的存储及使用原则，请直接参考 **JEDEC J-STD-033A**。

2.4.3 重新烘烤

【使用范围】

需要重新烘烤的 IC（潮敏产品）

【重新烘烤参考表】

表2-7 重新烘烤参考表

Body thickness	level	bake@ 125°C	bake@ $90^{\circ}\text{C} \leq 5\% \text{ RH}$	bake@ $40^{\circ}\text{C} \leq 5\% \text{ RH}$
$\leq 1.4\text{mm}$	2a	3 hours	11 hours	5 days
	3	7hours	23 hours	9 days
	4	7 hours	23 hours	9 days
	5	7 hours	24 hours	10 days
	5a	10 hours	24 hours	10 days
$\leq 2.0\text{mm}$	2a	16 hours	2 days	22 days
	3	17 hours	2 days	23 days
	4	20 hours	3 days	28 days
	5	25 hours	4 days	35 days
	5a	40 hours	6 days	56 days



Body thickness	level	bake@125℃	bake@90℃ $\leq$ 5% RH	bake@40℃ $\leq$ 5% RH
$\leq$ 4.5mm	2a	48 hours	7 days	67 days
	3	48 hours	8 days	67 days
	4	48 hours	10 days	67 days
	5	48 hours	10 days	67 days
	5a	48 hours	10 days	67 days



说明

- 表 2-7 显示的均是受潮后，必须的最小的烘烤时间。
- 重新烘烤优先选择低温烘烤。
- 详细情况请参考 JEDEC。

2.5 电性能参数

2.5.1 功耗参数

芯片的功耗参数如表 2-8 所示。



注意

- 功耗参数基于芯片典型工作场景。
- 单板电源设计必须参照《Hi35xxVxxx 硬件设计用户指南》。

表2-8 功耗参数

符号	参数	典型值	最大值	单位
VDD	内核电源	923	1564	mW
VDDIO_DDR	DDR IO 电源	312	352	mW
DVDD18	I/O 电源	18	26	mW
DVDD33	I/O 电源	144	145	mW



注意

典型场景:

1. 智能业务：全速跑 VGG16 NNIE 智能业务。
2. 媒体业务：
 - 模式：VI 在线 VPSS 离线；
 - VI: 2592*1536@30fps 输入 + 2 合一行模式 WDR + PIPE 开启行压,通道开启段压 +ISP 开启 DE, VI 送 VPSS;
 - VPSS: 3 个通道非压输出 (2592*1536@30+1024*576@30+720*480@30) + 开启 VPSS 的 NR+ 组叠加两个马赛克+ VPSS 的 0 通道旋转 90° , VPSS 送编码和 VO 预览;
 - VEDU: 3 路 H26x 编码 (H265 SmartP 2592*1536@30fps 8Mbps + H264 NormalP 1024*576@30fps 1Mbps + H264 NormalP 720*480@30fps 512Kbps) + 一路 Jpeg 编码 (2592*1536@2fps) +编码各通道上面叠加时间通道号;
 - 1 路大码流点播;
 - IVSMD: 720*480@15fps;
 - 音频: 1 路音频 32K 采样率 16bit 位宽 AAC 协议开 AGC、ANR 和 AEC。
 - 频点: CPU:900MHz/VICAP:396MHz/VIPROC:300MHz/VPSS:300MHz/VGS:475MHz/JPGE:396MHz/IVE:475MHz/VDP:300MHz/DDR:900MHz

2.5.2 温度和热阻参数

温度和热阻参数如表 2-9、表 2-10 和表 2-11 所示。



说明

- 热阻基于 JEDEC JESD51 系列标准给出，应用时的系统设计及环境可能与 JEDEC JESD51 系列标准不同，需要根据应用条件作出分析。
 - ✓ θ_{JA} 请参考 JESD 51-2 标准
 - ✓ θ_{JB} 请参考 JESD 51-8 标准
 - ✓ θ_{JC} 请参考：
 - (1) MIL-STD 883 1012.1
 - (2) SEMI G30-88
- 芯片结温和芯片功耗成正比关系，需要考虑将芯片结温控制在合理范围内，以匹配供电电源规格。

表2-9 工作环境参数

芯片	工作环境温度 (T_A)	
	最小值 (°C)	最大值 (°C)
Hi35xxVxxx	0	70



表2-10 结温参数

芯片	工作结温 (T_J)		破坏性结温 ($^{\circ}\text{C}$)
	最小值 ($^{\circ}\text{C}$)	最大值 ($^{\circ}\text{C}$)	
Hi35xxVxxx	0	105	125



注意

- 在任何条件下芯片的结温都不能大于表 2-10 中的“破坏性结温”。若芯片结温超过该极限值，可能导致芯片物理损坏。
- 在正常工作条件下芯片的结温必须在工作结温范围内且芯片工作的环境温度必须在工作环境温度范围内。

热阻参数如表 2-11 所示。

表2-11 热阻参数

参数	符号	典型值	单位
Junction-to-ambient thermal resistance	θ_{JA}	29.7	$^{\circ}\text{C}/\text{W}$
Junction-to-board thermal resistance	θ_{JB}	15.2	$^{\circ}\text{C}/\text{W}$
Junction-to-case thermal resistance	θ_{JC}	9.29	$^{\circ}\text{C}/\text{W}$

2.5.3 工作条件

工作条件如表 2-12 和表 2-13 所示。



注意

在使用 SVB 技术时，内核电源电压范围的前提条件是：客户单板必须使用 SVB 电路，且 SVB 电路的阻容参数必须完全拷贝《Hi35xxVxxx 硬件设计用户指南》中的相关 RC 参数。

表2-12 使用 SVB 方案时，电源工作条件

符号	描述	SVB 电压范围		单位
		设置值	容忍的误差范围	
VDD/AVDD09_PLL	内核、PLL 电源	0.83	± 0.03	V



符号	描述	SVB 电压范围		单位
		设置值	容忍的误差范围	
		0.86	±0.03	V
		0.89	±0.03	V
		0.92	±0.03	V
		0.95	±0.03	V
		0.98	±0.03	V
		0.99	±0.03	V

表2-13 常压电源工作条件

符号	描述	最小值	典型值	最大值	单位
AVDD09_PLL	PLL core 模拟电源	0.83	0.9	0.99	V
AVDD33_PLL	PLL IO 模拟电源	2.97	3.3	3.63	V
AVDD33_USB	USB I/O 电源	2.97	3.3	3.63	V
AVDD33_HDMITX	HDMI TX I/O 电源	2.97	3.3	3.63	V
AVDD09_HDMITX	HDMI TX core 电源	0.83	0.9	0.99	V
AVDD33_AC	ACODEC I/O 电源	2.97	3.3	3.63	V
AVDD_BAT	RTC 电池电源	1.6	3.0	3.6	V
DVDD33	I/O 电源	2.97	3.3	3.63	V
AVDD3318_MIPIRX	MIPI 模拟电源	1.71	1.8	1.89	V
	MIPI 复用为单端数字信号时，3.3V 电源	2.97	3.3	3.63	V
	MIPI 复用为单端数字信号时，1.8V 电源	1.62	1.8	1.98	V
DVDD3318_SENSOR	SENSOR、SPI0 I/O 电源	1.62	1.8	1.98	V
	SENSOR、SPI0 I/O 电源	2.97	3.3	3.63	V
DVDD3318_VI	VI/VO I/O 电源	1.62	1.8	1.98	V



符号	描述	最小值	典型值	最大值	单位
	VI/VO I/O 电源	2.97	3.3	3.63	V
DVDD3318_UART1	UART1 I/O 电源	1.62	1.8	1.98	V
	UART1 I/O 电源	2.97	3.3	3.63	V
DVDD3318_SARADC	LSADC I/O 电源	1.62	1.8	1.98	V
	LSADC I/O 电源	2.97	3.3	3.63	V
DVDD3318_PC	PC I/O 电源	1.62	1.8	1.98	V
	PC I/O 电源	2.97	3.3	3.63	V
DVDD3318_SDIO1	SDIO1 I/O 电源	1.71	1.8	1.89	V
	SDIO1 I/O 电源	3.135	3.3	3.465	V
AVDD3318_MIPITX	MIPI 模拟电源	1.71	1.8	1.89	V
	MIPI 复用为单端数字信号时 3.3V 电源	2.97	3.3	3.63	V
	MIPI 复用为单端数字信号时 1.8V 电源	1.62	1.8	1.98	V
DVDD3318_FLASH	EMMC/SFC IO 电源	1.7	1.8	1.95	V
	EMMC/SFC IO 电源	2.7	3.3	3.6	V
DVDD18_SDIO0	SDIO0 I/O 电源	1.71	1.8	1.89	V
AVDD33_DDR_PLL	DDR PLL 3.3V 模拟电源	3.20	3.3	3.40	V
VDDIO_DDR/VDDIO_DDR_CK	DDR3 I/O 电源	1.425	1.5	1.575	V
	DDR3L I/O 电源	1.283	1.35	1.45	V
	DDR4 I/O 电源	1.14	1.2	1.26	V

表2-14 破坏性电压值

参数描述	电源管脚名称	最小破坏性电压值 (V)	最大破坏性电压值 (V)
CPU 内核电源	VDD	-0.2	+1.17
Core 电源供电电压	VDD/AVDD09_PLL/AVDD09_HDMITX	-0.2	+1.17
DDR IO 电	VDDIO_DDR/VDDIO_DDR_	-0.2	+1.8



参数描述	电源管脚名称	最小破坏性电压值 (V)	最大破坏性电压值 (V)
源供电电压	CK		
DDR PLL 电源供电电压	AVDD33_DDR_PLL	-0.2	+3.96
1.8V IO 电源供电电压	DVDD18_SDIO0	-0.2	+1.98
3.3V IO 电源供电电压	DVDD33/ AVDD33_PLL/AVDD33_US B/AVDD33_HDMITX/AVD D33_AC	-0.2	+3.96
3.3V/1.8V IO 电源供电电压	AVDD3318_MIPIRX/ DVDD3318_SENSOR/ DVDD3318_UART1/ DVDD3318_SARADC/ DVDD3318_PC/ DVDD3318_SDIO1/ AVDD3318_MIPITX/ DVDD3318_FLASH	-0.2	+3.96/+1.98
RTC 电池供电电源	AVDD_BAT	-0.2	+3.96



注意

- 只要芯片供电电压超出破坏性电压值范围，就可能造成芯片的物理功能永久性损坏。
- 芯片供电电压必须在电源工作条件范围内，否则质量无法保证。

2.5.4 上下电顺序

请参考《Hi35xxVxxx 硬件设计用户指南》上下电时序章节。

2.5.5 DC/AC 电气参数

DC 电气参数如表 2-15 和表 2-16 所示。

表2-15 DC 电气参数表(DVDD33/DVDD3318_SENSOR/ DVDD3318_VI/
DVDD3318_UART1/ DVDD3318_SARADC/ DVDD3318_PC/ DVDD3318_FLASH =3.3V)

符号	参数	最小值	典型值	最大值	单位	说明
DVDD33	接口电压	2.97	3.3	3.6	V	-



符号	参数	最小值	典型值	最大值	单位	说明
V_{IH}	高电平输入电压	DVDD33*0.65	-	DVDD33+0.3	V	接口不兼容 5V 输入
V_{IL}	低电平输入电压	-0.3	-	DVDD33*0.35	V	-
I_L	输入漏电流	-	-	±10	μA	-
I_{OZ}	三态输出漏电流	-	-	±10	μA	-
V_{OH}	高电平输出电压	2.4	-	-	V	-
V_{OL}	低电平输出电压	-	-	0.4	V	-
I_{OH}	高电平输出电流	3.5	-	-	mA	IO2_档位 4
		7	-	-	mA	IO2_档位 3
		10.5	-	-	mA	IO2_档位 2
		14	-	-	mA	IO2_档位 1
I_{OL}	低电平输出电流	3	-	-	mA	IO2_档位 4
		6	-	-	mA	IO2_档位 3
		9	-	-	mA	IO2_档位 2
		11	-	-	mA	IO2_档位 1
I_{OH}	高电平输出电流	4	-	-	mA	IO3_档位 4
		13	-	-	mA	IO3_档位 3
		21	-	-	mA	IO3_档位 2
		25	-	-	mA	IO3_档位 1
I_{OL}	低电平输出电流	3	-	-	mA	IO3_档位 4
		10	-	-	mA	IO3_档位 3
		16	-	-	mA	IO3_档位 2
		18	-	-	mA	IO3_档位 1
I_{OH}	高电平输出电流	4	-	-	mA	IO4_档位 8
		7	-	-	mA	IO4_档位 7
		13	-	-	mA	IO4_档位 6
		15	-	-	mA	IO4_档位 5
		28	-	-	mA	IO4_档位 4
		28.8	-	-	mA	IO4_档位 3



符号	参数	最小值	典型值	最大值	单位	说明
		31	-	-	mA	IO4_档位 2
		32	-	-	mA	IO4_档位 1
I _{OL}	低电平输出电流	3	-	-	mA	IO4_档位 8
		6	-	-	mA	IO4_档位 7
		10	-	-	mA	IO4_档位 6
		12	-	-	mA	IO4_档位 5
		20	-	-	mA	IO4_档位 4
		20.8	-	-	mA	IO4_档位 3
		21.5	-	-	mA	IO4_档位 2
		22	-	-	mA	IO4_档位 1
I _{OH}	高电平输出电流	3.8	-	-	mA	IO6_档位 8
		7.6	-	-	mA	IO6_档位 7
		14	-	-	mA	IO6_档位 6
		18	-	-	mA	IO6_档位 5
		28	-	-	mA	IO6_档位 4
		32	-	-	mA	IO6_档位 3
		38	-	-	mA	IO6_档位 2
		42	-	-	mA	IO6_档位 1
I _{OL}	低电平输出电流	3.3	-	-	mA	IO6_档位 8
		6.6	-	-	mA	IO6_档位 7
		12	-	-	mA	IO6_档位 6
		15	-	-	mA	IO6_档位 5
		24	-	-	mA	IO6_档位 4
		28	-	-	mA	IO6_档位 3
		34	-	-	mA	IO6_档位 2
		36	-	-	mA	IO6_档位 1
I _{OH}	高电平输出电流	4	-	-	mA	IO16_档位 16
		8	-	-	mA	IO16_档位 15
		12	-	-	mA	IO16_档位 14



符号	参数	最小值	典型值	最大值	单位	说明
		18	-	-	mA	IO16_档位 13
		25	-	-	mA	IO16_档位 12
		32	-	-	mA	IO16_档位 11
		36	-	-	mA	IO16_档位 10
		40	-	-	mA	IO16_档位 9
		44	-	-	mA	IO16_档位 8
		48	-	-	mA	IO16_档位 7
		52	-	-	mA	IO16_档位 6
		56	-	-	mA	IO16_档位 5
		62	-	-	mA	IO16_档位 4
		66	-	-	mA	IO16_档位 3
		70	-	-	mA	IO16_档位 2
		74	-	-	mA	IO16_档位 1
I _{OL}	低电平输出电流	3.5	-	-	mA	IO16_档位 16
		7	-	-	mA	IO16_档位 15
		10.5	-	-	mA	IO16_档位 14
		14	-	-	mA	IO16_档位 13
		22	-	-	mA	IO16_档位 12
		26	-	-	mA	IO16_档位 11
		30	-	-	mA	IO16_档位 10
		34	-	-	mA	IO16_档位 9
		38	-	-	mA	IO16_档位 8
		42	-	-	mA	IO16_档位 7
		46	-	-	mA	IO16_档位 6
		49	-	-	mA	IO16_档位 5
		55	-	-	mA	IO16_档位 4
		59	-	-	mA	IO16_档位 3
		61	-	-	mA	IO16_档位 2
		64	-	-	mA	IO16_档位 1



符号	参数	最小值	典型值	最大值	单位	说明
R _{PU1}	内部上拉电阻	37	41	45	kΩ	-
R _{PD1}	内部下拉电阻	33	37	41	kΩ	-
R _{PU2}	内部上拉电阻	7	8	9	kΩ	-
R _{PD2}	内部下拉电阻	6	7	8	kΩ	-
R _{PU3}	内部上拉电阻	38	42	46	kΩ	-
R _{PU4}	内部上拉电阻	17	19	21	kΩ	-
R _{PD4}	内部下拉电阻	76	84	92	kΩ	-
R _{PD5}	内部下拉电阻	22	24	27	kΩ	-
R _{PU6}	内部上拉电阻	17	19	21	kΩ	-
R _{PD6}	内部下拉电阻	76	84	92	kΩ	-
R _{PU7}	内部上拉电阻	17	19	21	kΩ	-
R _{PD7}	内部下拉电阻	76	84	92	kΩ	-
R _{PU8}	内部上拉电阻	18	20	22	kΩ	-
R _{PD8}	内部下拉电阻	22	25	28	kΩ	-
R _{PU9}	内部上拉电阻	17	19	21	kΩ	-
R _{PD9}	内部下拉电阻	22	24	27	kΩ	-
R _{PU10}	内部上拉电阻	18	20	22	kΩ	-
R _{PD10}	内部下拉电阻	22	25	28	kΩ	-
R _{PU11}	内部上拉电阻	18	20	22	kΩ	-
R _{PD11}	内部下拉电阻	22	25	28	kΩ	-
R _{PU12}	内部上拉电阻	7	8	9	kΩ	-
R _{PD12}	内部下拉电阻	6	7	8	kΩ	-
R _{PU13}	内部上拉电阻	18	20	22	kΩ	-
R _{PD13}	内部下拉电阻	22	24	27	kΩ	-
R _{PU14}	内部上拉电阻	7	8	9	kΩ	-
R _{PD14}	内部下拉电阻	7	8	9	kΩ	-
R _{PU15}	内部上拉电阻	10	11	12	kΩ	-
R _{PD15}	内部下拉电阻	7	8	9	kΩ	-
R _{PU16}	内部上拉电阻	26	29	32	kΩ	-



符号	参数	最小值	典型值	最大值	单位	说明
R _{PD16}	内部下拉电阻	24	27	30	kΩ	-
R _{PU17}	内部上拉电阻	29	32	35	kΩ	-
R _{PD17}	内部下拉电阻	28	31	34	kΩ	-

表2-16 DC 电气参数表 (DVDD3318_SENSOR/ DVDD3318_VI/ DVDD3318_UART1/
DVDD3318_SARADC/ DVDD3318_PC/ DVDD3318_FLASH =1.8V,
SDIO0_CARD_DETECT/SDIO0_CARD_POWER_EN 供电 1.8V)

符号	参数	最小值	典型值	最大值	单位	说明
DVDD18	接口电压	1.7	1.8	1.95	V	-
V _{IH}	高电平输入电压	1.17	-	DVDD18+0.3	V	-
V _{IL}	低电平输入电压	-0.3	-	0.63	V	-
I _L	输入漏电流	-	-	±10	μA	-
I _{OZ}	三态输出漏电流	-	-	±10	μA	-
V _{OH}	高电平输出电压	1.35	-	-	V	-
V _{OL}	低电平输出电压	-	-	0.45	V	-
I _{OH}	高电平输出电流	1	-	-	mA	IO2_档位 4
		2	-	-	mA	IO2_档位 3
		3	-	-	mA	IO2_档位 2
		4	-	-	mA	IO2_档位 1
I _{OL}	低电平输出电流	1.5	-	-	mA	IO2_档位 4
		3	-	-	mA	IO2_档位 3
		4.5	-	-	mA	IO2_档位 2
		6	-	-	mA	IO2_档位 1
I _{OH}	高电平输出电流	1	-	-	mA	IO3_档位 4
		3	-	-	mA	IO3_档位 3
		6	-	-	mA	IO3_档位 2
		8	-	-	mA	IO3_档位 1
I _{OL}	低电平输出电流	1	-	-	mA	IO3_档位 4
		6	-	-	mA	IO3_档位 3



符号	参数	最小值	典型值	最大值	单位	说明
		11	-	-	mA	IO3_档位 2
		14	-	-	mA	IO3_档位 1
I _{OH}	高电平输出电流	1	-	-	mA	IO4_档位 8
		2	-	-	mA	IO4_档位 7
		3	-	-	mA	IO4_档位 6
		4	-	-	mA	IO4_档位 5
		9	-	-	mA	IO4_档位 4
		9.5	-	-	mA	IO4_档位 3
		10.5	-	-	mA	IO4_档位 2
		11	-	-	mA	IO4_档位 1
I _{OL}	低电平输出电流	1	-	-	mA	IO4_档位 8
		3	-	-	mA	IO4_档位 7
		6	-	-	mA	IO4_档位 6
		8	-	-	mA	IO4_档位 5
		16	-	-	mA	IO4_档位 4
		17	-	-	mA	IO4_档位 3
		18	-	-	mA	IO4_档位 2
		19	-	-	mA	IO4_档位 1
I _{OH}	高电平输出电流	1	-	-	mA	IO6_档位 8
		2	-	-	mA	IO6_档位 7
		4	-	-	mA	IO6_档位 6
		4.9	-	-	mA	IO6_档位 5
		7.1	-	-	mA	IO6_档位 4
		7.8	-	-	mA	IO6_档位 3
		9	-	-	mA	IO6_档位 2
		9.6	-	-	mA	IO6_档位 1
I _{OL}	低电平输出电流	1.6	-	-	mA	IO6_档位 8
		3.2	-	-	mA	IO6_档位 7
		6.4	-	-	mA	IO6_档位 6



符号	参数	最小值	典型值	最大值	单位	说明
		8	-	-	mA	IO6_档位 5
		12	-	-	mA	IO6_档位 4
		13	-	-	mA	IO6_档位 3
		15	-	-	mA	IO6_档位 2
		16	-	-	mA	IO6_档位 1
I _{OH}	高电平输出电流	1	-	-	mA	IO16_档位 16
		2	-	-	mA	IO16_档位 15
		3	-	-	mA	IO16_档位 14
		5	-	-	mA	IO16_档位 13
		7	-	-	mA	IO16_档位 12
		9	-	-	mA	IO16_档位 11
		10	-	-	mA	IO16_档位 10
		11	-	-	mA	IO16_档位 9
		13	-	-	mA	IO16_档位 8
		14	-	-	mA	IO16_档位 7
		15	-	-	mA	IO16_档位 6
		17	-	-	mA	IO16_档位 5
		19	-	-	mA	IO16_档位 4
		20	-	-	mA	IO16_档位 3
		21	-	-	mA	IO16_档位 2
		22	-	-	mA	IO16_档位 1
I _{OL}	低电平输出电流	1	-	-	mA	IO16_档位 16
		3	-	-	mA	IO16_档位 15
		5	-	-	mA	IO16_档位 14
		7	-	-	mA	IO16_档位 13
		11	-	-	mA	IO16_档位 12
		13	-	-	mA	IO16_档位 11
		15	-	-	mA	IO16_档位 10
		17	-	-	mA	IO16_档位 9



符号	参数	最小值	典型值	最大值	单位	说明
		19	-	-	mA	IO16_档位 8
		21	-	-	mA	IO16_档位 7
		23	-	-	mA	IO16_档位 6
		25	-	-	mA	IO16_档位 5
		29	-	-	mA	IO16_档位 4
		31	-	-	mA	IO16_档位 3
		33	-	-	mA	IO16_档位 2
		34	-	-	mA	IO16_档位 1
R _{PU1}	内部上拉电阻	53	59	65	kΩ	-
R _{PD1}	内部下拉电阻	46	51	56	kΩ	-
R _{PU2}	内部上拉电阻	7	8	9	kΩ	-
R _{PD2}	内部下拉电阻	6	7	8	kΩ	-
R _{PU3}	内部上拉电阻	54	60	66	kΩ	-
R _{PU4}	内部上拉电阻	39	43	47	kΩ	-
R _{PD4}	内部下拉电阻	212	236	260	kΩ	-
R _{PD5}	内部下拉电阻	46	51	56	kΩ	-
R _{PU6}	内部上拉电阻	39	43	47	kΩ	-
R _{PD6}	内部下拉电阻	212	236	260	kΩ	-
R _{PU8}	内部上拉电阻	40	44	48	kΩ	-
R _{PD8}	内部下拉电阻	50	56	62	kΩ	-
R _{PU9}	内部上拉电阻	39	43	47	kΩ	-
R _{PD9}	内部下拉电阻	46	51	56	kΩ	-
R _{PU11}	内部上拉电阻	38	42	46	kΩ	-
R _{PD11}	内部下拉电阻	50	56	62	kΩ	-
R _{PU12}	内部上拉电阻	7	8	9	kΩ	-
R _{PD12}	内部下拉电阻	6	7	8	kΩ	-
R _{PU13}	内部上拉电阻	40	44	48	kΩ	-
R _{PD13}	内部下拉电阻	50	56	62	kΩ	-
R _{PU14}	内部上拉电阻	8	9	10	kΩ	-



符号	参数	最小值	典型值	最大值	单位	说明
R _{PD14}	内部下拉电阻	8	9	10	kΩ	-
R _{PU15}	内部上拉电阻	14	16	18	kΩ	-
R _{PD15}	内部下拉电阻	8	9	10	kΩ	-
R _{PU16}	内部上拉电阻	29	32	35	kΩ	-
R _{PD16}	内部下拉电阻	27	30	33	kΩ	-
R _{PU17}	内部上拉电阻	32	36	40	kΩ	-
R _{PD17}	内部下拉电阻	77	86	95	kΩ	-

DDR3 模式下，DC 电气参数如表 2-17 所示。

表2-17 DC 电气参数表（VDDIO_DDR =1.5V，DDR3 模式）

符号	参数	最小值	典型值	最大值	单位	说明
VDDIO_D DR	接口电压	1.425	1.5	1.575	V	-
Vref	参考电压	0.49 x VDDIO_ _DDR	0.5 x VDDIO_ _DDR	0.51 x VDDIO_ _DDR	-	(0.49~0.51) x VDDIO_DDR
V _{IH(DC)}	高电平输入电压	Vref+0. 1	-	VDDIO_ _DDR	V	-
V _{IL(DC)}	低电平输入电压	0	-	Vref-0.1	V	-
V _{OH(DC)}	高电平输出电压	-	0.8 x VDDIO_ _DDR	-	V	驱动可配置
V _{OL(DC)}	低电平输出电压	0	0.2 x VDDIO_ _DDR	-	V	驱动可配置

DDR3 模式下，AC 电气参数如表 2-18 所示。

表2-18 AC 电气参数表（VDDIO_DDR =1.5V，DDR3 模式）

符号	参数	最小值	最大值	单位	说明
V _{IH(AC)}	高电平输入电压	Vref + 0.15	VDDIO_DDR+0.4	V	-
V _{IL(AC)}	低电平输入电压	-0.4	Vref-0.15	V	-



DDR3L 模式下，DC 电气参数如表 2-19 所示。

表2-19 DC 电气参数表（VDDIO_DDR =1.35V，DDR3L 模式）

符号	参数	最小值	典型值	最大值	单位	说明
VDDIO_D DR	接口电压	1.283	1.35	1.45	V	-
Vref	DDR 接口参考电 压	0.49 x VDDIO_DDR	0.5 x VDDIO_DDR	0.51 x VDDIO_DDR	V	
V _{IH(DC)}	高电平输入电压	Vref + 0.09	-	VDDIO_DDR	V	-
V _{IL(DC)}	低电平输入电压	0	-	Vref -0.09	V	-
V _{OH(DC)}	高电平输出电压	-	0.8x VDDIO_DDR	-	V	-
V _{OL(DC)}	低电平输出电压	-	0.2 x VDDIO_DDR	-	V	-

DDR3L 模式下，AC 电气参数如表 2-20 所示。

表2-20 AC 电气参数表（VDDIO_DDR =1.35V，DDR3L 模式）

符号	参数	最小值	最大值	单位	说明
V _{IH(AC)}	AC 高电平输入电压	Vref + 0.135	VDDIO_DDR + 0.4	V	-
V _{IL(AC)}	AC 低电平输入电压	-0.4	Vref - 0.135	V	-

DDR4 模式下，AC 和 DC 电气参数如表 2-21 和表 2-22 所示。

表2-21 AC 和 DC 电气参数表-数据线（VDDIO_DDR =1.2V，DDR4 模式）

符号	参数	最小值	典型值	最大值	单位	说明
VDDIO_D DR	接口电压	1.14	1.2	1.26	V	-
Vref	参考电压	0.4 * VDDIO_DD R	0.5 * VDDIO_DDR	0.9 * VDDIO_D DR	V	-
V _{OH(DC)}	高电平输出电压	-	1.1 * VDDIO_DDR	-	V	-
V _{OM(DC)}	中间电平输出电 压	-	0.8 * VDDIO_DDR	-	V	-
V _{OL(DC)}	低电平输出电压	-	0.5 * VDDIO_DDR	-	V	-



符号	参数	最小值	典型值	最大值	单位	说明
$V_{OH(AC)}$	高电平输出电压	-	$(0.7+0.15) * V_{DDIO\_DDR}$	-	V	-
$V_{OL(AC)}$	低电平输出电压	-	$(0.7-0.15) * V_{DDIO\_DDR}$	-	V	-
输出阻抗	-	34	-	240	Ω	-

表2-22 AC 和 DC 电气参数表-地址/命令线 ($V_{DDIO\_DDR} = 1.2V$, DDR4 模式)

符号	参数	最小值	典型值	最大值	单位	说明
V_{DDIO_DDR}	接口电压	1.14	1.2	1.26	V	-
V_{ref}	参考电压	$0.49 * V_{DDIO\_DDR}$	$0.5 * V_{DDIO\_DDR}$	$0.51 * V_{DDIO\_DDR}$	V	-
$V_{IH(DC75)}$	高电平输入电压	$V_{ref} + 0.075$	-	V_{DDIO_DDR}	V	-
$V_{IL(DC75)}$	低电平输入电压	0	-	$V_{ref} - 0.075$	V	-
$V_{IH(AC100)}$	高电平输入电压	$V_{ref} + 0.1$	-	$V_{DDIO\_DDR} + 0.012$	V	-
$V_{IL(AC100)}$	低电平输入电压	0.012	-	$V_{ref} - 0.1$	V	-
$V_{OH(DC)}$	高电平输出电压	-	$1.1 * V_{DDIO\_DDR}$	-	V	-
$V_{OM(DC)}$	中间电平输出电压	-	$0.8 * V_{DDIO\_DDR}$	-	V	-
$V_{OL(DC)}$	低电平输出电压	-	$0.5 * V_{DDIO\_DDR}$	-	V	-
$V_{OH(AC)}$	高电平输出电压	-	$0.85 * V_{DDIO\_DDR}$	-	V	-
$V_{OL(AC)}$	低电平输出电压	-	$0.55 * V_{DDIO\_DDR}$	-	V	-
输出阻抗	-	34	-	240	Ω	-

2.5.6 MIPI/LVDS Rx 电气参数

DPHY/HiSPi/Sub-LVDS/LVDS 差分 DC 电气参数如表 2-23 所示。



表2-23 LVDS 差分 DC 电气参数表

符号	参数		最小值	典型值	最大值	单位
VIDTH(SL)	Differential Input Threshold Voltage $ (V_P - V_M) $	Sub-LVDS	70	-	-	mV
VIDTH(HS)		HiSPi(SLVS)	70	-	-	
VIDTH(HiVCM)		HiSPi(HiVCM)	100	-	-	
VIDTH(DP)		D-PHY HS	70	-	-	
VIDTH(LV)		LVDS	100	-	-	
VIDTH(ML)		Mini-LVDS	100	-	-	
VCM(SL)	Common Mode Voltage Range $(V_P + V_M)/2$	Sub-LVDS	0.5	0.9	1.3	V
VCM(HS)		HiSPi(SLVS)	0.07	0.2	0.35	
VCM(HiVCM)		HiSPi(HiVCM)	0.66	0.90	1.17	
VCM(DP)		D-PHY HS	0.07	0.2	0.33	
VCM(LV)		LVDS	0.925	1.2	1.475	
VCM(ML)		Mini-LVDS	1.025	1.2	1.375	
VISVR (SL)	Single-ended Input Voltage Range VP, VM	Sub-LVDS	0.4	-	1.4	V
VCM(HS)		HiSPi	-0.165	-	0.575	
VCM(DP)		D-PHY HS	-0.04	-	0.46	
VCM(LV)		LVDS	0	-	1.8	
VCM(ML)		Mini-LVDS	0.825	-	1.575	
ZID(SL)	Internal Termination Resister Value	Sub-LVDS	80	100	125	Ω
ZID(HS)		HiSPi			125	
ZID(LV)		LVDS			125	
ZID(ML)		Mini-LVDS			125	
ZID(DP)		D-PHY HS			125	

MIPI DPHY 参数如表 2-24~表 2-27 所示。

表2-24 MIPI DPHY HS (High Speed) DC 参数表

符号	参数	最小值	典型值	最大值	单位
VTERM-EN	Single-ended threshold for HS termination enable	-	-	450	mV



表2-25 MIPI DPHY HS (High Speed) AC 参数表

符号	参数	最小值	典型值	最大值	单位
Δ VCMRX(HF)	Common-mode interface beyond 450MHz	-	-	100	mV
Δ VCMRX(LF)	Common-mode interface 50MHz-450MHz	-50	-	50	mV
CCM	Common-mode termination	-	-	60	pF

表2-26 MIPI DPHY LP (Low Power) DC 参数表

符号	参数	最小值	典型值	最大值	单位
VIHLP	Logic 1 input voltage	880	-	-	mV
VILLP	Logic 0 input voltage	-	-	550	
VHYST	Input hysteresis	25	-	-	

表2-27 MIPI DPHY LP (Low Power) AC 参数表

符号	参数	最小值	典型值	最大值	单位
e _{SPIKE}	Input pulse rejection	-	-	300	V • ps
T _{MIN-RX}	Minimun pulse rejection	20	-	-	ns
V _{INT}	Peak interference amplitude	-	-	200	mV
f _{INT}	Interference frequency	450	-	-	MHz

2.5.7 MIPI Tx 电气参数

下面各表给出了 MIPI Tx DPHY 的电气参数，各参数的具体含义请参考 MIPI 联盟的《Specification for D-PHY Version 1.2》。

MIPI Tx LP DC 参数如表 2-28 所示。

表2-28 MIPI Tx LP DC 参数表

符号	参数	最小值	典型值	最大值	单位
V _{OH}	Thevenin output high level	1.1	1.2	1.3	V
V _{OL}	Thevenin output low level	-50	-	50	mV



符号	参数	最小值	典型值	最大值	单位
Z_{OLP}	Output impedance of LP transmitter	110	-	-	Ω
V_{IH}	Logic 1 input voltage	880	-	-	mV
V_{IL}	Logic 0 input voltage	-	-	550	mV
V_{HYST}	Input hysteresis	25	-	-	mV

MIPI Tx LP AC 参数如表 2-29 所示。

表2-29 MIPI Tx LP AC 参数表

符号	参数	最小值	典型值	最大值	单位
T_{RLP}	15%-85% rise time, $C_L < 70\text{pF}$	-	-	25	ns
T_{FLP}	15%-85% fall time, $C_L < 70\text{pF}$	-	-	25	ns
T_{REOT}	30%-85% rise time and fall time, $C_L < 70\text{pF}$	-	-	35	ns
C_{LOAD}	Load capacitance	0	-	70	pF
$\delta V / \delta t_{SR}$	15% to 85%, $C_L < 70\text{pF}$	-	-	150	mV/ns
e_{SPIKE}	Input pulse rejection	-	-	300	V · ps
T_{MIN-RX}	Minimum pulse width response	20	-	-	ns
V_{INT}	Peak interference amplitude	-	-	200	mV
F_{INT}	Interference frequency	450	-	-	MHz

MIPI Tx HS DC 参数如表 2-30 所示。

表2-30 MIPI Tx HS DC 参数表

符号	参数	最小值	典型值	最大值	单位
V_{CMTX}	HS transmit static common mode voltage	150	200	250	mV
$ \Delta V_{CMTX}(1,0) $	Vcmtx mismatch when output is differential-1 or differential-0	-	-	5	mV
$ V_{OD} $	HS transmit differential voltage	140	200	270	mV



符号	参数	最小值	典型值	最大值	单位
$ \Delta V_{OD} $	Vod mismatch when output is differential-1 or differential-0	-	-	14	mV
V_{OHHS}	HS output high voltage	-	-	360	mV
ΔZ_{os}	single ended output impedance mismatch	-	-	10%	
Z_{os}	single ended output impedance	40	50	62.5	Ω

MIPI Tx HS AC 参数如表 2-31 所示。

表2-31 MIPI Tx HS AC 参数表

符号	参数	最小值	典型值	最大值	单位
$\Delta V_{CMTX(HF)}$	common-level variations above 450Mhz	-	-	20	mV
$\Delta V_{CMTX(LF)}$	common-level variations between 50-450Mhz	-	-	25	mV
t_R and t_F	20%-80% rise time and fall time when datarate $\leq$ 1Gbps	0.15	-	0.3UI	ns

2.5.8 SDIO 电气参数

SDIO 电性能参数表（3.3V）如表 2-32 所示。

表2-32 SDIO 电性能参数表 3.3V(SDIO1 供电 3.3V 时)

符号	参数	最小值	典型值	最大值	单位	说明
DVDD33	供电电压	3.135	3.3	3.465	V	-
VOH	输出高电平	$0.75 \cdot DVDD33$	-	-	V	
VOL	输出低电平	-	-	$0.125 \cdot DVDD33$	V	
VIH	输入高电平	$0.625 \cdot DVDD33$	-	$DVDD33 + 0.3$	V	
VIL	输入低电平	-0.3	-	$0.25 \cdot DVDD33$	V	
-	上电时间	-	-	0.018	V/us	0V 到 VDDIO



SDIO 电性能参数（1.8V）如表 2-33 所示。

表2-33 SDIO 电性能参数表 1.8V(SDIO0/1 供电 1.8V 时)

符号	参数	最小值	典型值	最大值	单位	说明
DVDD18	供电电压	1.71	1.8	1.89	V	-
VOH	输出高电平	1.4	-	-	V	
VOL	输出低电平	-	-	0.45	V	
VIH	输入高电平	1.27	-	1.98	V	
VIL	输入低电平	-0.3	-	0.58	V	
-	上电时间	-	-	0.018	V/us	0V 到 DVDD18

2.5.9 AUDIO CODEC 电气参数

Audio Code 电性能参数如表 2-34、表 2-35、表 2-36、表 2-37 和表 2-38 所示。

表2-34 总体指标表

参数	最小值	典型值	最大值	单位	说明
模拟电路电源 AVDD	3	3.3	3.6	V	相对于 AGND
VREF	-	AVDD/2	-	V	相对于 AGND

表2-35 DAC 主要指标表

参数	最小值	典型值	最大值	单位	说明
满幅输出幅度	-	0.875	-	Vrms	最大输出信号摆幅

表2-36 ADC 主要指标表

参数	最小值	典型值	最大值	单位	说明
最大输入幅度	-	1	-	Vrms	ADC 最大输入信号摆幅



表2-37 MICBIAS 主要指标表

参数	最小值	典型值	最大值	单位	说明
偏置电压	-	$2.1 \times AVDD / 3.3$	-	V	麦克风偏置电压
最大输出电流	-	-	3	mA	-

表2-38 MICPGA 主要指标

参数	最小值	典型值	最大值	单位	说明
输入电压范围	-	1	-	V _{rms}	最大输入信号摆幅
输入阻抗	-	30	36	k Ω	MICPGA 的输入阻抗

2.6 接口时序

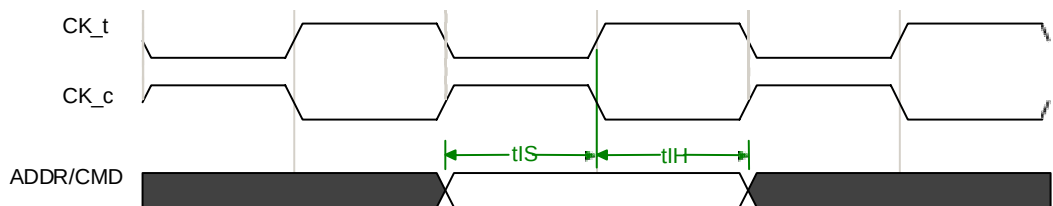
2.6.1 DDR 接口时序

2.6.1.1 写操作时序

命令和地址相对于 CK 的写操作时序

对于 DDR 命令和地址相对于 CK 的写操作时序， t_{IS} 为命令和地址到 CK 上升沿的建立时间， t_{IH} 为 CK 上升沿到命令和地址的保持时间。DDR3(L)/DDR4 命令和地址相对于 CK 的写操作时序如图 2-9 所示。

图2-9 DDR3(L)/DDR4 命令和地址相对于 CK 的写操作时序图

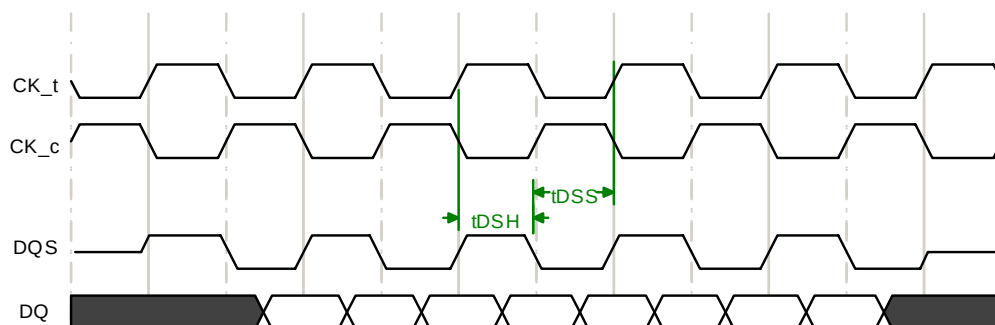


DQS 相对于 CK 的写操作时序

对于 DDR3(L)/DDR4 DQS 相对于 CK 的写操作时序， t_{DSH} 为 CK 上升沿到 DQS 下降沿的保持时间， t_{DSS} 为 DQS 下降沿到 CK 上升沿的建立时间。DQS 相对于 CK 的写操作时序如图 2-10 所示。



图2-10 DDR3(L)/DDR4 DQS 相对于 CK 的写操作时序图



2.6.1.2 读操作时序

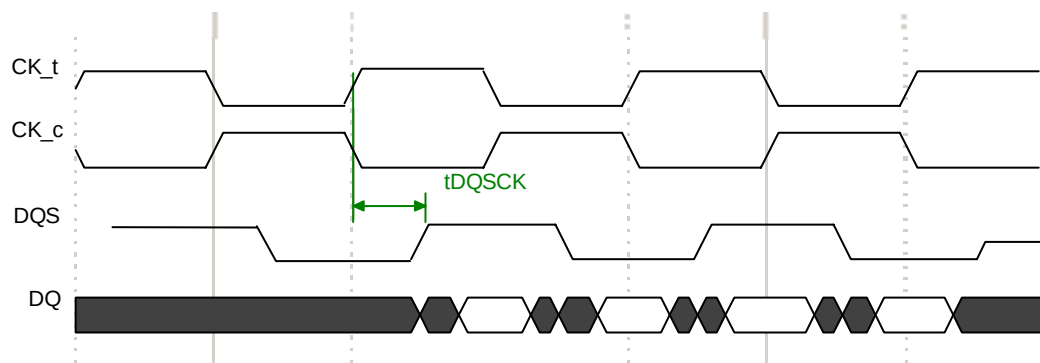
命令和地址相对于 CK 的读操作时序

“命令和地址相对于 CK 的读操作时序”与“命令和地址相对于 CK 的写操作时序”相同。

CK 相对于 DQS 的读操作时序

对于 DDR3(L)/DDR4，CK 相对于 DQS 的读操作时序， t_{DQSCK} 为有效 DQS 相对于 CK 的偏斜。DDR3(L)/DDR4 CK 相对于 DQS 的读操作时序如图 2-11 所示。

图2-11 DDR3(L)/DDR4 CK 相对于 DQS 的读操作时序图

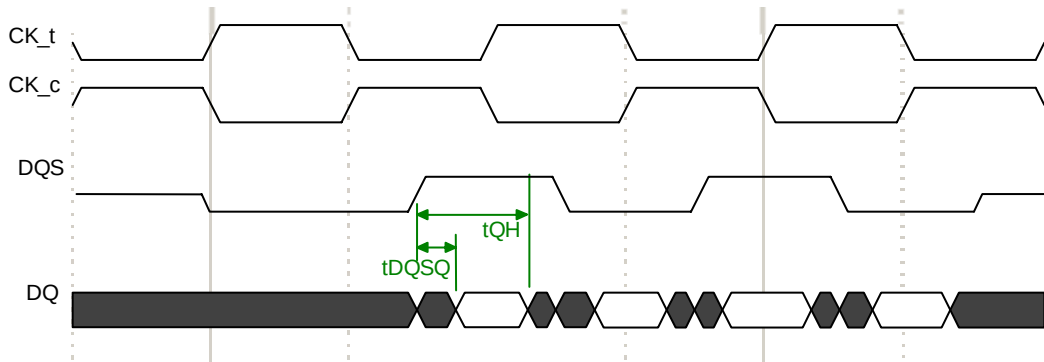


DQS 相对于 DQ 的读操作时序

对于 DDR3(L)/DDR4 DQS 相对于 DQ 的读操作时序， t_{QH} 是最早无效的 DQ 相对于 DQS 的抖动。DDR3(L)/DDR4 DQS 相对于 DQ 的读操作时序如图 2-12 所示。



图2-12 DDR3(L)/DDR4 DQS 相对于 DQ 的读操作时序图



2.6.1.3 时序参数

DDR 接口时序满足 JEDEC（JESD79-3E）标准协议，本文中描述的时序都是 DDR PHY 侧输出的时序。

DDR3-1800 SDRAM 存储器参数如表 2-39 所示。

表2-39 DDR3 SDRAM 存储器参数表（DDR3-1800）

参数	最小值	典型值	最大值	单位
存储器时钟频率	-	900	-	MHz
锁相环抖动 tJIT(cc)	-120	-	120	ps
锁相环抖动 tJIT(per)	-60	-	60	ps
锁相环占空比(avg)	47.000	-	53.000	%

DDR3-1800 SDRAM 存储器参数如表 2-40 所示。

表2-40 DDR3 SDRAM 存储器参数表（DDR3-1800）

参数	符号	最小值	最大值	单位
DQS 下降沿相对 DDR 时钟的建立时间	tDSS	0.18	-	tCK
DQS 下降沿相对 DDR 时钟的保持时间	tDSH	0.18	-	tCK
DQS 与 DQ 的偏斜	tDQSQ	-	85	ps
参考 Vih(ac) / Vil(ac)电平的地址和命令相对 DDR 时钟的建立时间	tIS(base)	TBD	-	ps
参考 Vih(dc) / Vil(dc)电平的地址和命令相对 DDR 时钟的保持时间	tIH(base)	TBD	-	-
DQS 输出时相对 DDR 时钟的偏斜	tDQCK	-195	195	ps



DDR4-1800 SDRAM 存储器参数如表 2-42 所示。

表2-41 DDR4 时钟参数表

参数	最小值	典型值	最大值	单位
存储器时钟频率	-	900	-	MHz
锁相环抖动 tJIT(cc)	-	-	111	ps
锁相环抖动 tJIT(per)	-	-	56	ps
锁相环占空比(avg)	48.000	-	52.000	%

表2-42 DDR4 SDRAM 存储器参数表（DDR4-1800）

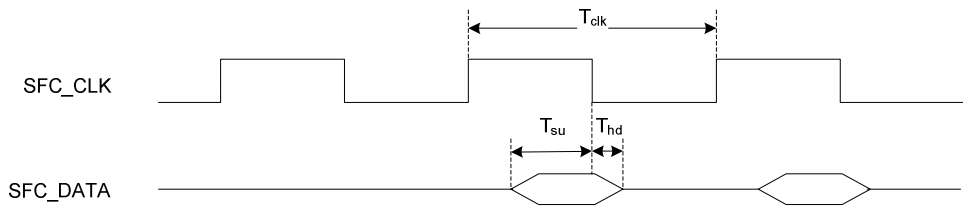
参数	符号	最小值	最大值	单位
DQS 下降沿相对 DDR 时钟的建立时间	tDSS	0.18	-	tCK
DQS 下降沿相对 DDR 时钟的保持时间	tDSH	0.18	-	tCK
DQS 与 DQ 的偏斜	tDQSQ	-	0.16	tCK
参考 Vih(ac) / Vil(ac)电平的地址和命令相对 DDR 时钟的建立时间	tIS(base)	100	-	ps
参考 Vref 电平的地址和命令相对 DDR 时钟的建立时间	tIS(Vref)	200	0	ps
参考 Vih(dc) / Vil(dc)电平的地址和命令相对 DDR 时钟的保持时间	tIH(base)	125	-	-
参考 vref 电平地址和命令相对 DDR 时钟的保持时间	tIH(Vref)	200	-	-
DQS 输出时相对 DDR 时钟的偏斜	tDQSCK	-195	195	ps

2.6.2 SPI FLASH 接口时序

SFC-SDR 模式输入方向时序如图 2-13 所示。



图2-13 SFC 输入方向时序图-SDR 模式



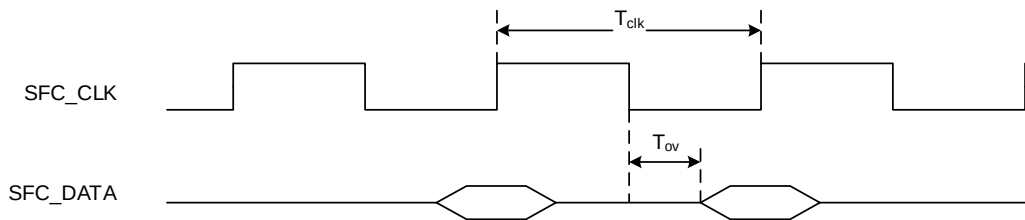
SFC-SDR 模式输入方向时序参数如表 2-43 所示。

表2-43 SFC-SDR 模式输入方向时序参数表

参数	符号	最小值	典型值	最大值	单位
SFC_CLK 时钟周期（SDR）	T_{clk}	10.0	-	83.2	ns
输入信号建立时间要求	T_{su}	2	-	-	ns
输入信号保持时间要求	T_{hd}	1.2	-	-	ns

SFC-SDR 输出方向时序如图 2-14 所示。

图2-14 SFC 输出方向时序图-SDR 模式



SFC-SDR 输出方向时序参数如表 2-44 所示。

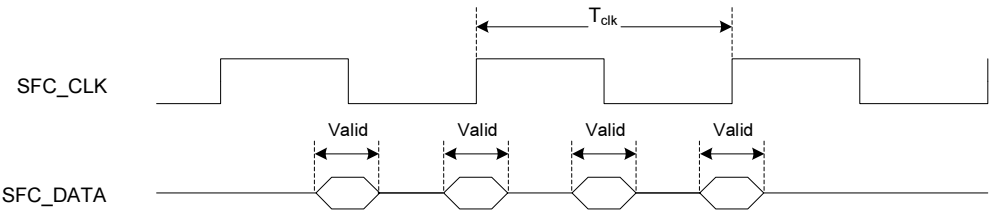
表2-44 SFC-SDR 输出方向时序参数表

参数	符号	最小值	典型值	最大值	单位
SFC_CLK 时钟周期（SDR）	T	10.0	-	83.2	ns
输出数据信号延时	T_{ov}	-3	-	2	ns

SFC-DDR 模式输入方向时序如图 2-15 所示。



图2-15 SFC 输入方向时序图-DDR 模式



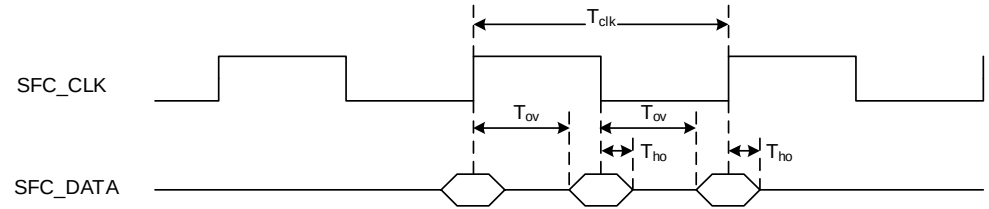
SFC-DDR 模式输入方向时序参数如表 2-45 所示。

表2-45 SFC-DDR 输入方向时序参数表

参数	符号	最小值	典型值	最大值	单位
SFC_CLK 时钟周期 (DDR)	T_{clk}	10.1	-	83.2	ns
输入信号有效时间要求	Valid	$T_{clk} * 30\%$	-	-	ns

SFC-DDR 输出方向时序如图 2-16 所示。

图2-16 SFC 输出方向时序图-DDR 模式



SFC-DDR 输出方向时序参数如表 2-46 所示。

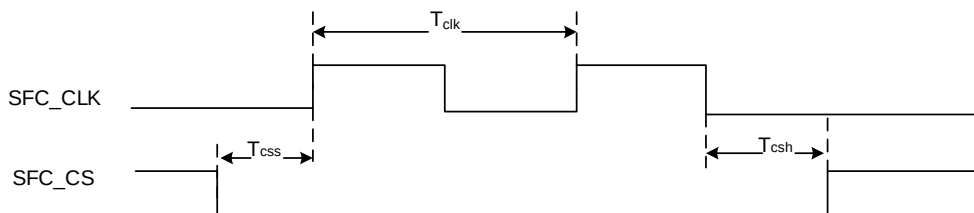
表2-46 SFC-DDR 输出方向时序参数表

参数	符号	最小值	典型值	最大值	单位
SFCCLK 时钟周期 (DDR)	T	10.1	-	83.2	ns
输出信号延时	T_{ov}	-	-	$T/2-1.25$	ns
输出信号保持时间	T_{ho}	1.25	-	-	ns

SFC-SDR 和 SFC-DDR CS 输出方向时序如图 2-17 所示。



图2-17 SFC CS 输出方向时序图（SDR 和 DDR 模式）



SFC-SDR 和 SFC-DDR CS 输出方向时序参数如表 2-47 所示。

表2-47 SFC-SDR 和 SFC-DDR CS 输出方向时序参数表

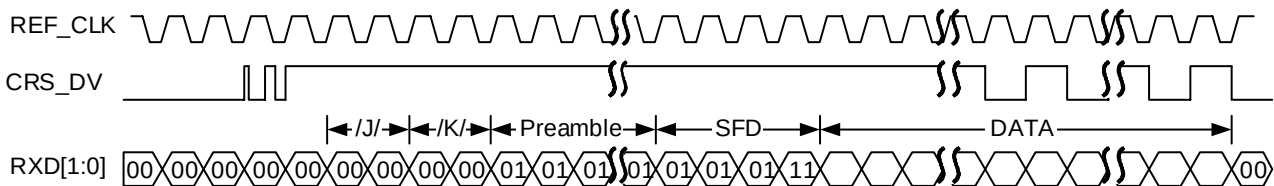
参数	符号	最小值	典型值	最大值	单位
SFC_CLK 时钟周期	T	10.1	-	83.2	ns
输出 CS 建立时间要求	T _{css}	10	-	-	ns
输出 CS 保持时间要求	T _{csh}	10	-	-	ns

2.6.3 Ethernet MAC 接口时序

2.6.3.1 RMII 接口时序

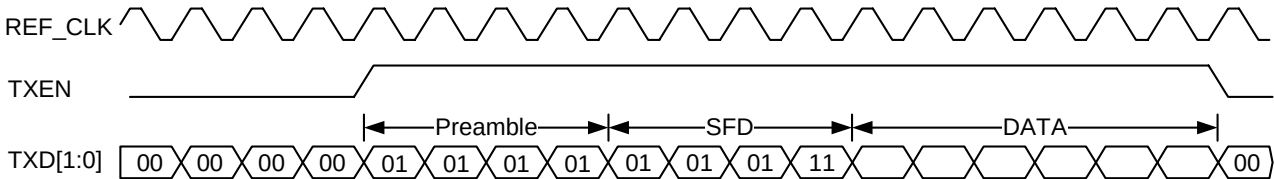
RMII 接口 100Mbit/s 接收时序如图 2-18 所示。

图2-18 RMII 接口 100Mbit/s 接收时序



RMII 接口 100Mbit/s 发送时序如图 2-19 所示。

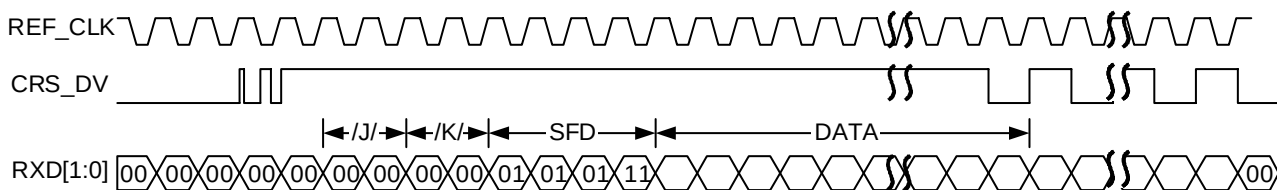
图2-19 RMII 接口 100Mbit/s 发送时序



RMII 接口 10Mbit/s 接收时序如图 2-20 所示。

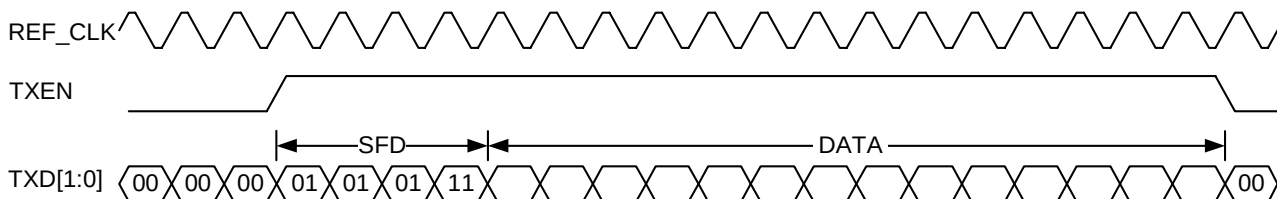


图2-20 RMII 接口 10Mbit/s 接收时序



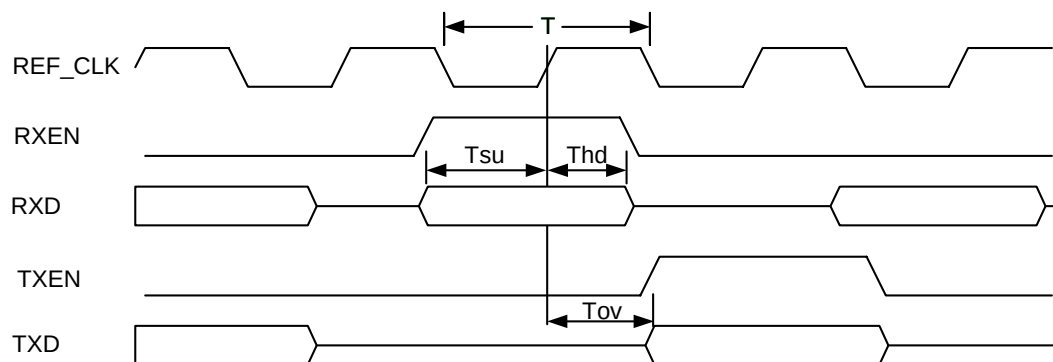
RMII 接口 10Mbit/s 发送时序如图 2-21 所示。

图2-21 RMII 接口 10Mbit/s 发送时序



RMII 接口时序参数如图 2-22 所示。

图2-22 RMII 接口时序参数



RMII 接口时序参数说明如表 2-48 所示。

表2-48 RMII 接口时序参数说明

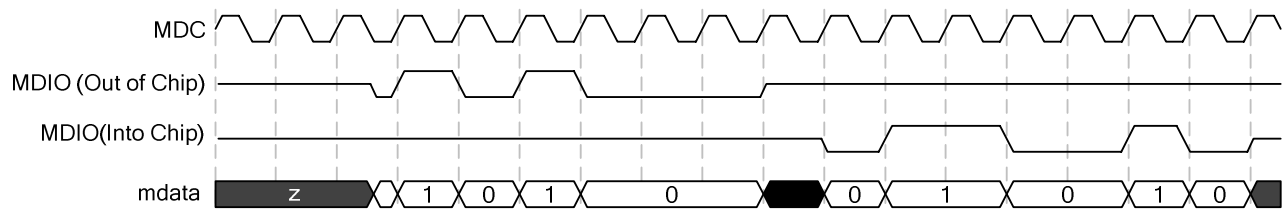
参数	符号	信号	最小值	最大值	单位
RMII 时钟周期	T	REF_CLK	20	20	ns
RMII 信号建立时间	T _{su} (RX)	CRS_DV/RXD[1:0]	4	-	ns
RMII 信号保持时间	Thd (RX)	CRS_DV/RXD[1:0]	2	-	ns
RMII 输出信号延时	T _{ov} (TX)	TXEN/TXD[1:0]	3	16	ns



2.6.3.2 MDIO 接口时序

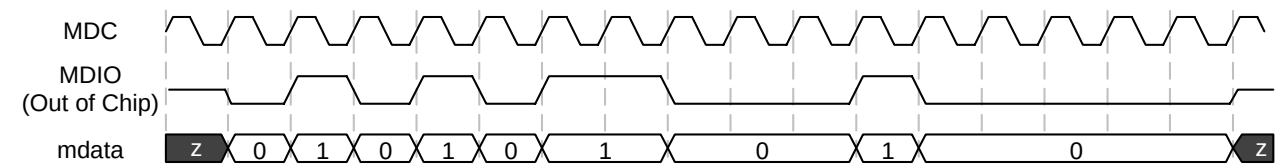
MDIO 接口读时序如图 2-23 所示。

图2-23 MDIO 接口读时序



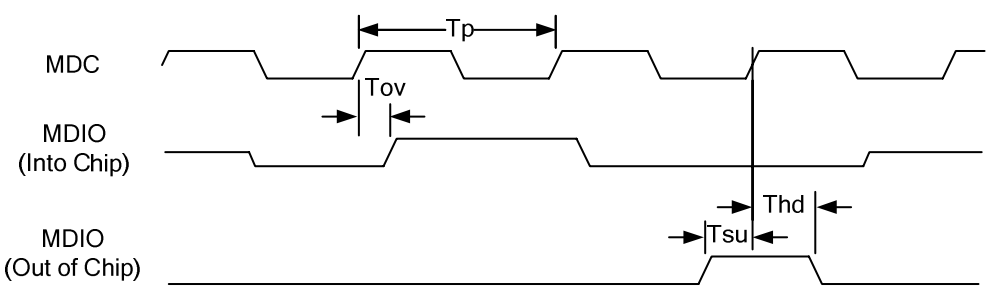
MDIO 接口写时序如图 2-24 所示。

图2-24 MDIO 接口写时序



MDIO 接口时序参数如图 2-25 所示。

图2-25 MDIO 接口接收时序参数



MDIO 接口时序参数说明如表 2-49 所示。

表2-49 MDIO 接口时序参数

参数	符号	信号	最小值	最大值	单位
MDIO 接收数据延迟时间	Tov	MDIO	0	375	ns
MDIO 时钟周期	Tp	MDCK	500	7400	ns
MDIO 发送数据建立时间	Tsu	MDIO	10	-	ns

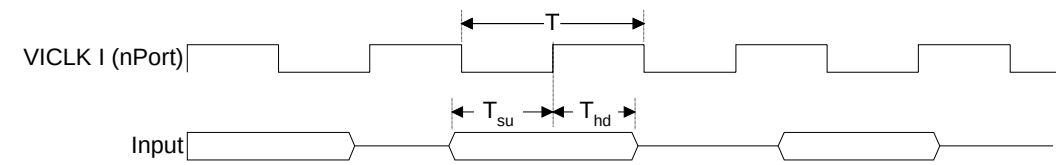


参数	符号	信号	最小值	最大值	单位
MDIO 发送数据保持时间	Thd	MDIO	10	-	ns

2.6.4 VI 接口时序

VI 接口 CMOS 模式时序如图 2-26 所示。

图2-26 VI 接口时序图



VI 接口时序参数如表 2-50 所示。

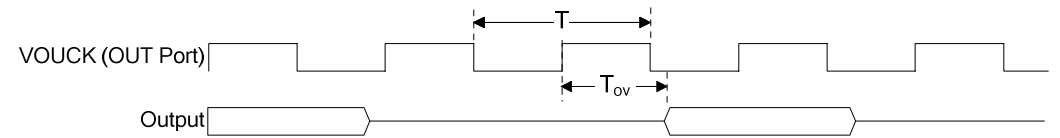
表2-50 VI 接口时序参数表

参数	符号	最小值	典型值	最大值	单位
VICLK 时钟周期	T	6.73	-	-	ns
输入信号建立时间要求	T _{su}	1.9	-	-	ns
输入信号保持时间要求	T _{hd}	0.8	-	-	ns

2.6.5 VO 接口时序

BT.656 接口时序如图 2-27 所示。

图2-27 BT.656 接口时序



BT.656 接口时序参数如表 2-51 所示。

表2-51 BT.656 接口时序参数表

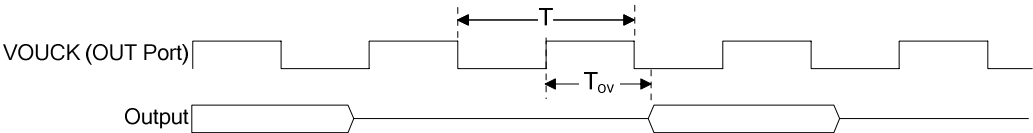
参数	符号	最小值	典型值	最大值	单位
VOUCLK 时钟周期	T	-	37	-	ns



参数	符号	最小值	典型值	最大值	单位
输出信号延时	T_{ov}	$T/2-1.5$	-	$T/2+1.5$	ns

LCD 接口时序如图 2-28 所示。

图2-28 LCD 接口时序



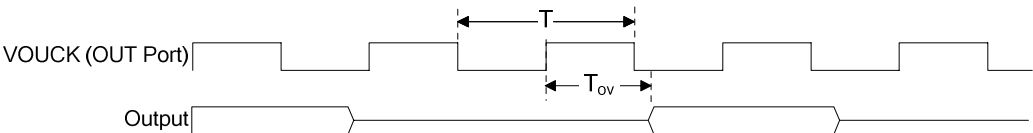
LCD 接口时序参数如表 2-52 所示。

表2-52 LCD 接口时序参数表

参数	符号	最小值	典型值	最大值	单位
VOUCLK 时钟周期	T	13.46	-	500	ns
输出信号延时	T_{ov}	$T/2-3$	-	$T/2+3$	ns

BT.1120 接口时序如图 2-29 所示。

图2-29 BT.1120 接口时序



BT.1120 接口时序参数如表 2-53 所示。

表2-53 BT.1120 接口时序参数表

参数	符号	最小值	典型值	最大值	单位
VOUCLK 时钟周期	T	-	6.75	13.48	ns
输出信号延时	T_{ov}	$T/2-1.5$	-	$T/2+1.5$	ns



2.6.6 AIAO 接口时序

2.6.6.1 I²S 接口时序

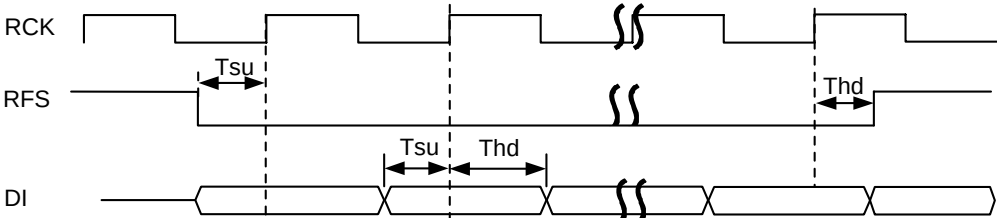


注意

外接 Audio codec 时，从模式需符合表 2-54 所示 I²S 时序标准及参数，若不符合，建议使用主模式。

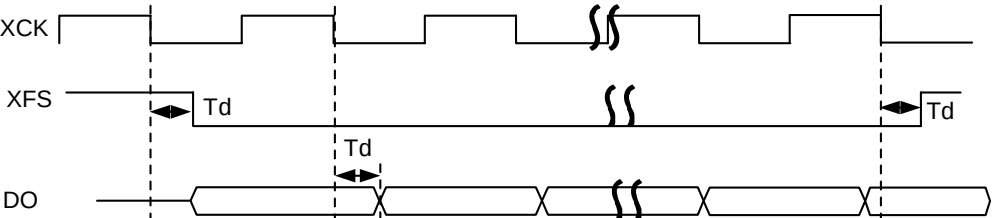
I²S 接口接收时序如图 2-30 所示。

图2-30 I²S 接口接收时序图



I²S 接口发送时序如图 2-31 所示。

图2-31 I²S 接口发送时序图



I²S 接口时序参数如表 2-54 所示。

表2-54 I²S 接口时序参数表

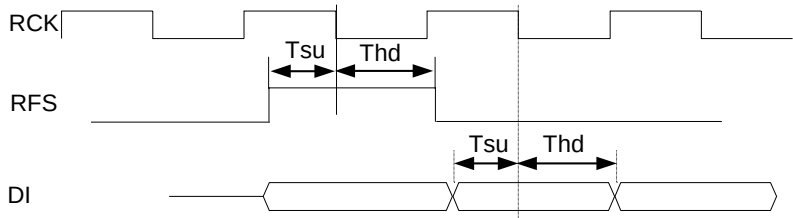
参数	符号	最小值	典型值	最大值	单位
输入信号建立时间	T_{su}	10	-	-	ns
输入信号保持时间	T_{hd}	10	-	-	ns
输出信号延时	T_d	0	-	8	ns



2.6.6.2 PCM 模式接口时序

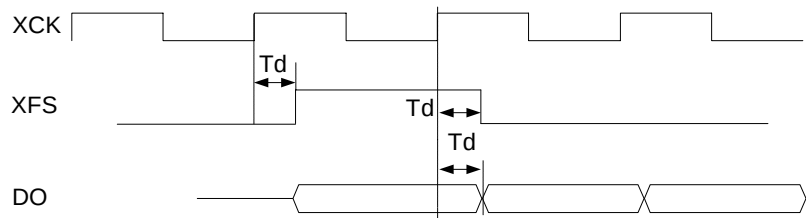
PCM 接口接收时序如图 2-32 所示。

图2-32 PCM 接口接收时序图



PCM 接口发送时序如图 2-33 所示。

图2-33 PCM 接口发送时序图



PCM 接口时序参数如表 2-55 所示。

表2-55 PCM 接口时序参数表

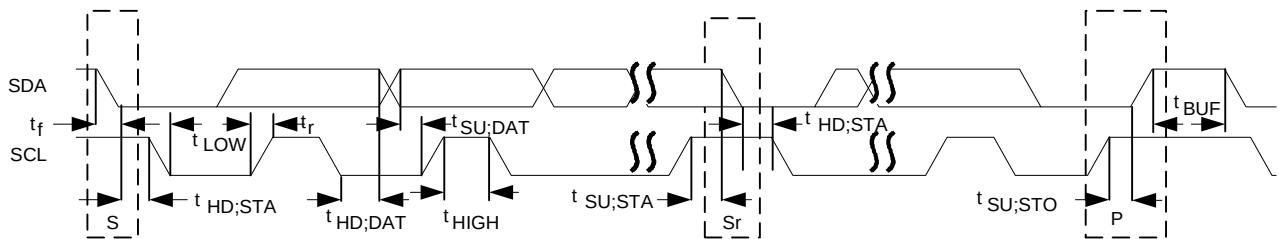
参数	符号	最小值	典型值	最大值	单位
输入信号建立时间	T_{su}	10	-	-	ns
输入信号保持时间	T_{hd}	10	-	-	ns
输出信号延时	T_d	0	-	8	ns

2.6.7 I²C 时序

I²C 传输时序如图 2-34 所示。



图2-34 I²C 传输时序图



I²C 接口时序参数如表 2-56 所示。

表2-56 I²C 接口时序参数表

参数	符号	标准模式		快速模式		单位
		最小值	最大值	最小值	最大值	
SCL 时钟频率	fSCL	0	100	0	400	kHz
启动保持时间	tHD;STA	4.0	-	0.6	-	μs
SCL 低电平周期	tLOW	4.7	-	1.3	-	μs
SCL 高电平周期	tHIGH	4.0	-	0.6	-	μs
启动建立时间	tSU;STA	4.7	-	0.6	-	μs
数据保持时间	tHD;DAT	0	3.45	0	0.9	μs
数据建立时间	tSU;DAT	250	-	100	-	ns
SDA、SCL 上升时间	tr	-	1000	20+0.1Cb	300	ns
SDA、SCL 下降时间	tf	-	300	20+0.1Cb	300	ns
结束建立时间	tSU;STO	4.0	-	0.6	-	μs
开始与结束之间的总线释放时间	tBUF	4.7	-	1.3	-	μs
总线负载	Cb	-	400	-	400	pF



2.6.8 SPI 接口时序

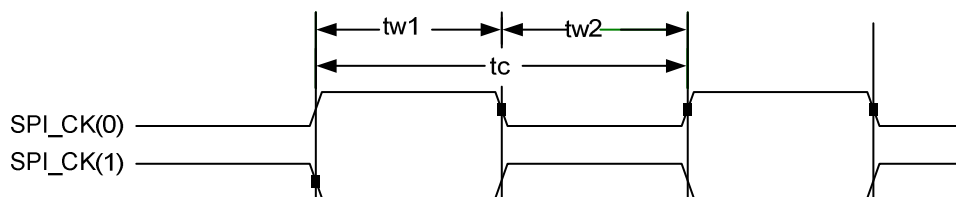
说明

图 2-35~图 2-37 中，以下缩略语或字母意义不变：

- MSB:Most Significant Bit
- LSB:Least Significant Bit
- SPI_CK(0):spo=0
- SPI_CK(1):spo=1

SPI 接口时钟时序如图 2-35 所示。

图2-35 SPICK 时序



SPI 主模式下接口时序分别如图 2-36 和图 2-37 所示。

图2-36 SPI 主模式下接口时序 (sph=1)

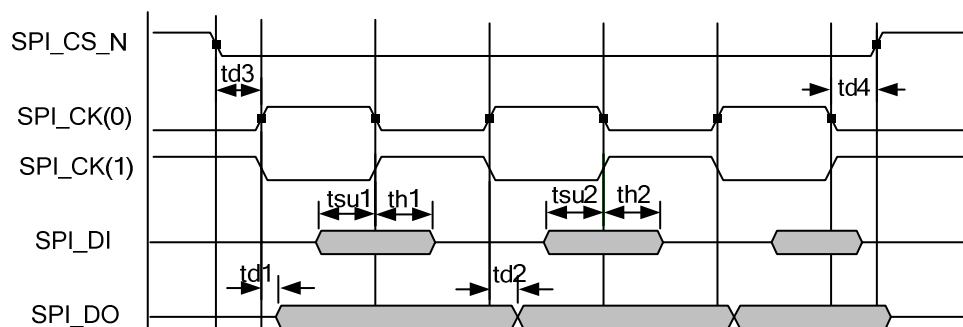
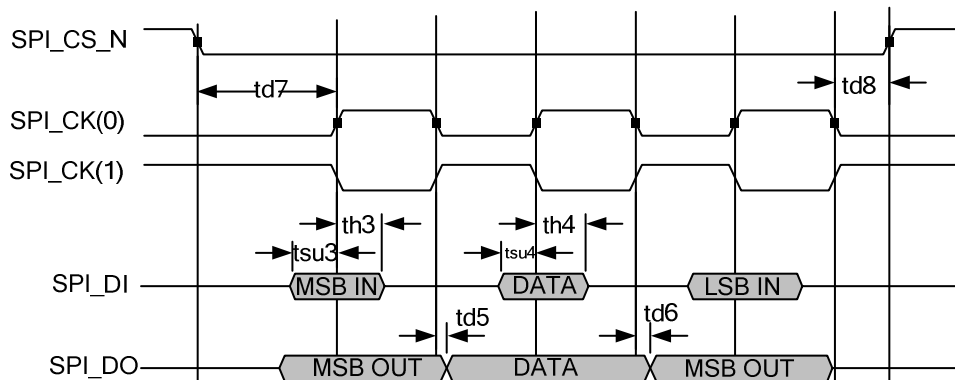




图2-37 SPI 主模式下接口时序 (sph=0)



SPI 接口时序参数如表 2-57 所示。

表2-57 SPI 接口时序参数

参数	符号	最小值	典型值	最大值	单位	备注
Cycle time, SPI_CK	tc	2000/Fss pclk	-	65024000/ Fsspclk	ns	Fsspclk 以 MHz 为单位
Pulse duration, SPI_CK high (All Master Modes)	tw1	-	1/2 tc	-	ns	
Pulse duration, SPI_CK low (All Master Modes)	tw2	-	1/2 tc	-	ns	
Setup time, SPI_DI (input) valid before SPI_CK (output) falling edge	tsu1	-	1/2 tc	-	ns	
Setup time, SPI_DI (in put) valid before SPICK (output) rising edge	tsu2	-	1/2 tc	-	ns	
Hold time, SPI_DI (input) valid after SPI_CK (output) falling edge	th1	-	1/2 tc	-	ns	
Hold time, SPI_DI (input) valid after SPI_CK (output) rising edge	th2	-	1/2 tc	-	ns	
Delay time, SPI_CK (output) rising edge to SPI_DO (output) transition	td1	0	-	-	ns	
Delay time, SPI_CK (output) falling edge to SPI_DO (output) transition	td2	0	-	-	ns	
Delay time, SPI_CS_N (output) falling edge to first SPI_CK (output) rising or falling edge	td3	-	tc	-	ns	



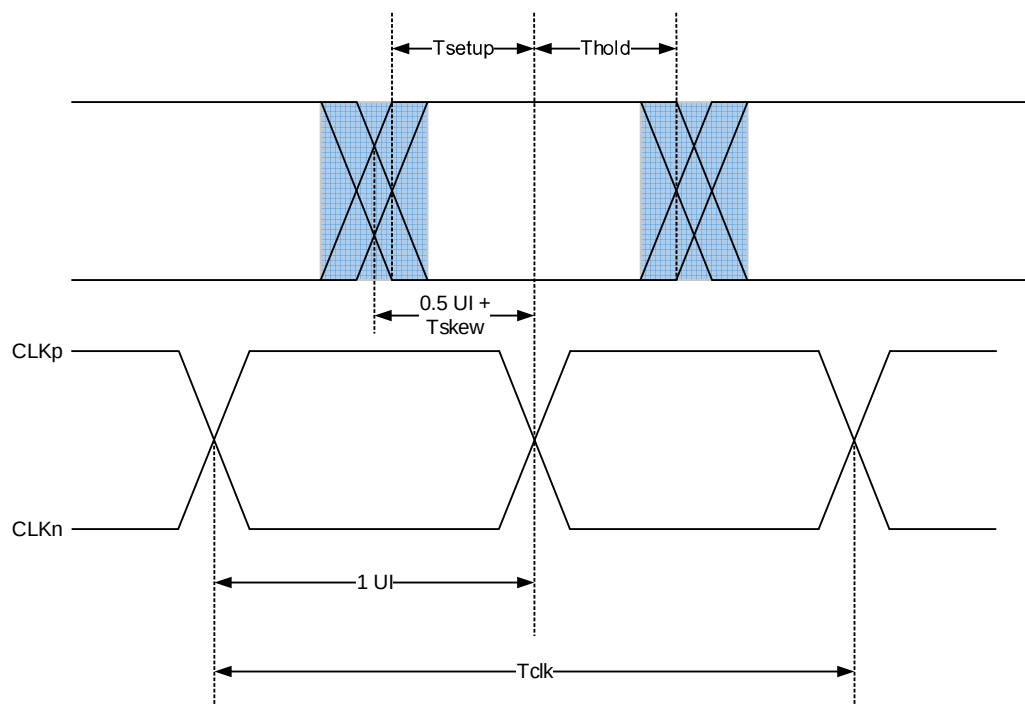
参数	符号	最小值	典型值	最大值	单位	备注
Delay time, SPI_CK (output) rising or falling edge to SPI_CS_N (output) rising edge	td4	-	1/2 tc	-	ns	
Setup time, SPI_DI (input) valid before SPI_CK (output) rising edge	tsu3	-	1/2 tc	-	ns	
Setup time, SPI_DI (input) valid before SPI_CK (output) falling edge	tsu4	-	1/2 tc	-	ns	
Hold time, SPI_DI (input) valid after SPI_CK (output) rising edge	th3	-	1/2 tc	-	ns	
Hold time, SPI_DI (input) valid after SPI_CK (output) falling edge	th4	-	1/2 tc	-	ns	
Delay time, SPI_CK (output) falling edge to SPI_DO (output) transition	td5	0	-	-	ns	
Delay time, SPI_CK (output) rising edge to SPI_DO (output) transition	td6	0	-	-	ns	
Delay time, SPI_CS_N (output) falling edge to first SPI_CK (output) rising or falling edge	td7	-	1/2 tc	-	ns	
Delay time, SPI_CK (output) rising or falling edge to SPI_CS_N (output) rising edge	td8	-	tc	-	ns	

2.6.9 MIPI Rx 接口时序

MIPI Rx 接口时钟时序如图 2-38 所示。



图2-38 MIPI Rx 接口时钟数据时序图



MIPI Rx 时序参数如表 2-58 所示。

表2-58 MIPI Rx 时序参数表

符号	参数	最小值	典型值	最大值	单位
FMAX	数据率	-	-	1.5G	bps
Tclk	差分时钟周期	2	-	-	ns
Tskew	Data to clock skew	-0.3*UI	-	0.3*UI	ns
Tsetup	差分时钟建立时间	0.15*UI	-	-	ns
Thold	差分时钟保持时间	0.15*UI	-	-	ns



说明

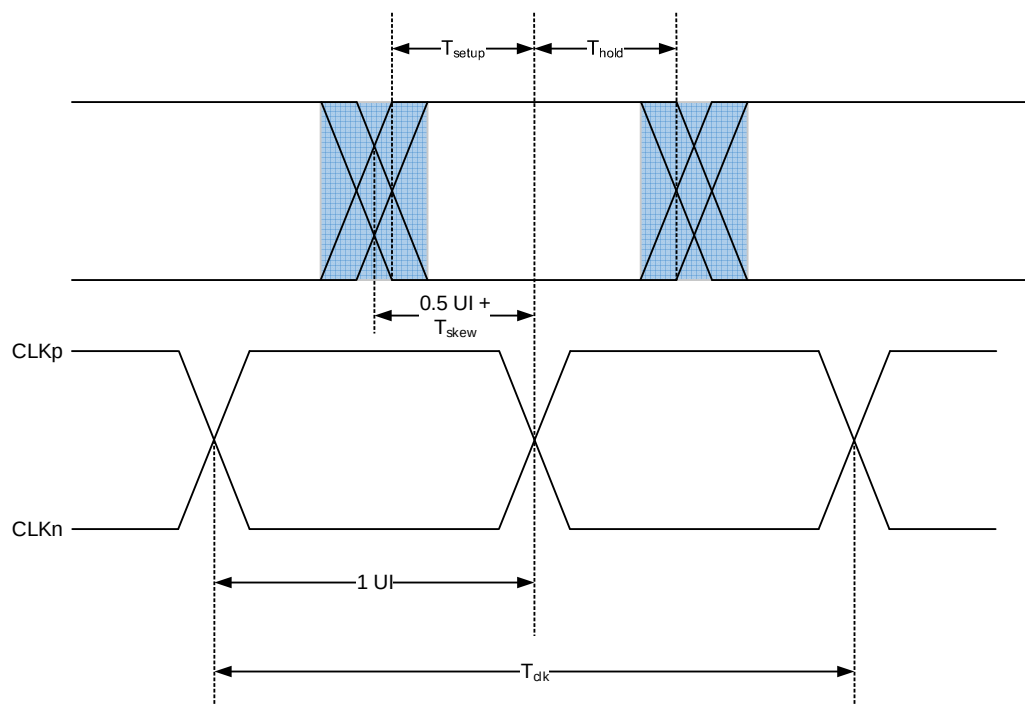
UI is equal to $T_{\text{PERIOD}}/2$.

2.6.10 MIPI Tx 接口时序

MIPI Tx 接口时钟时序如图 2-39 所示。



图2-39 MIPI Tx 接口时钟数据时序图



MIPI Tx DATA-CLOCK 时序参数如表 2-59 所示。

表2-59 MIPI Tx DATA-CLOCK 时序参数表

符号	参数	最小值	典型值	最大值	单位
Tskew	数据到时钟的 Skew	$-0.15 \cdot UI$	-	$0.15 \cdot UI$	ns
Tsetup	差分时钟建立时间、	$0.15 \cdot UI$	-	-	ns
Thold	差分时钟保持时间	$0.15 \cdot UI$	-	-	ns

2.6.11 SDIO/MMC 接口时序

单沿的数据输入输出方向时序如图 2-40 所示。



图2-40 单沿的 HS、DS 模式数据输入输出方向时序图

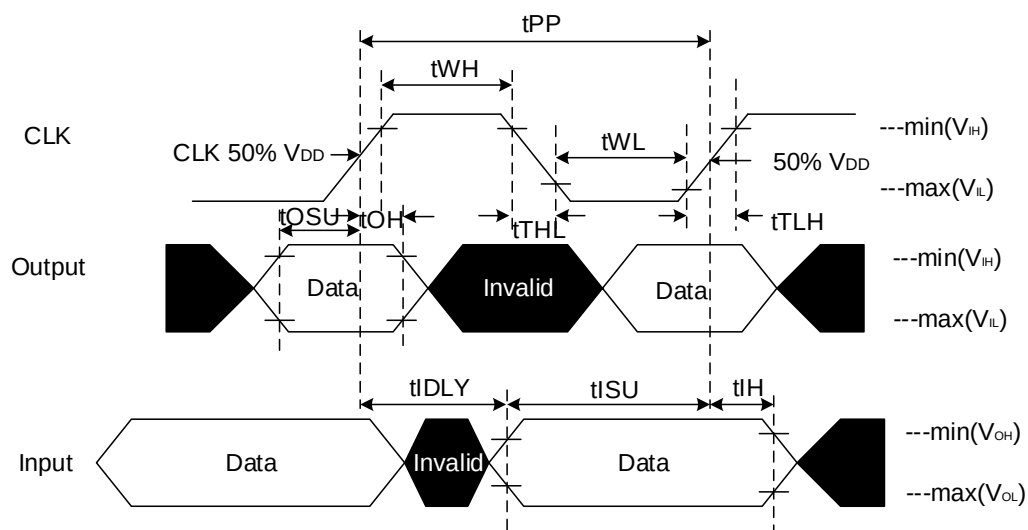


表2-60 单沿 DS 模式 时序参数表

符号	参数	最小值	典型值	最大值	单位	备注
Clock CLK						
f _{PP}	Clock frequency Data Transfer Mode (PP)	0	-	26	MHz	f _{pp} =1/t _{pp} CL ≤ 30 pF
t _{WH}	Clock high time	10	-	-		CL ≤ 30 pF
t _{WL}	Clock low time	10	-	-	ns	CL ≤ 30 pF
t _{TLH}	Clock rise time	-	-	10	ns	CL ≤ 30 pF
t _{THL}	Clock fall time	-	-	10	ns	CL ≤ 30 pF
Outputs DAT (referenced to CLK)						
t _{OSU}	Output set-up time	3	-	-	ns	CL ≤ 30 pF
t _{OH}	Output hold time	3	-	-	ns	CL ≤ 30 pF
Inputs DAT (referenced to CLK)						
t _{ISU}	Input set-up time	11.7	-	-	ns	CL ≤ 30 pF
t _{IH}	Input hold time	8.3	-	-	ns	CL ≤ 30 pF



表2-61 单沿 HS 模式 时序参数表

符号	参数	最小值	典型值	最大值	单位	备注
Clock CLK						
fPP	Clock frequency Data Transfer Mode (PP)	0	-	52	MHz	fpp=1/tpp CL ≤ 30 pF Tolerance: +100 KHz
tWH	Clock high time	6.5	-	-	ns	CL ≤ 30 pF
tWL	Clock low time	6.5	-	-	ns	CL ≤ 30 pF
tTLH	Clock rise time	-	-	3	ns	CL ≤ 30 pF
tTHL	Clock fall time	-	-	3	ns	CL ≤ 30 pF
Outputs DAT (referenced to CLK)						
tOSU	Output set-up time	3	-	-	ns	CL ≤ 30 pF
tOH	Output hold time	3	-	-	ns	CL ≤ 30 pF
Inputs CMD, DAT (referenced to CLK)						
tIDLY	Input delay time during data transfer	-	-	13.7	ns	CL ≤ 30 pF
tIH	Input hold time	2.5	-	-	ns	CL ≤ 30 pF

双沿的数据输入输出方向时序如图 2-41 所示。



图2-41 双沿 DDR50 模式的数据输入输出方向时序图

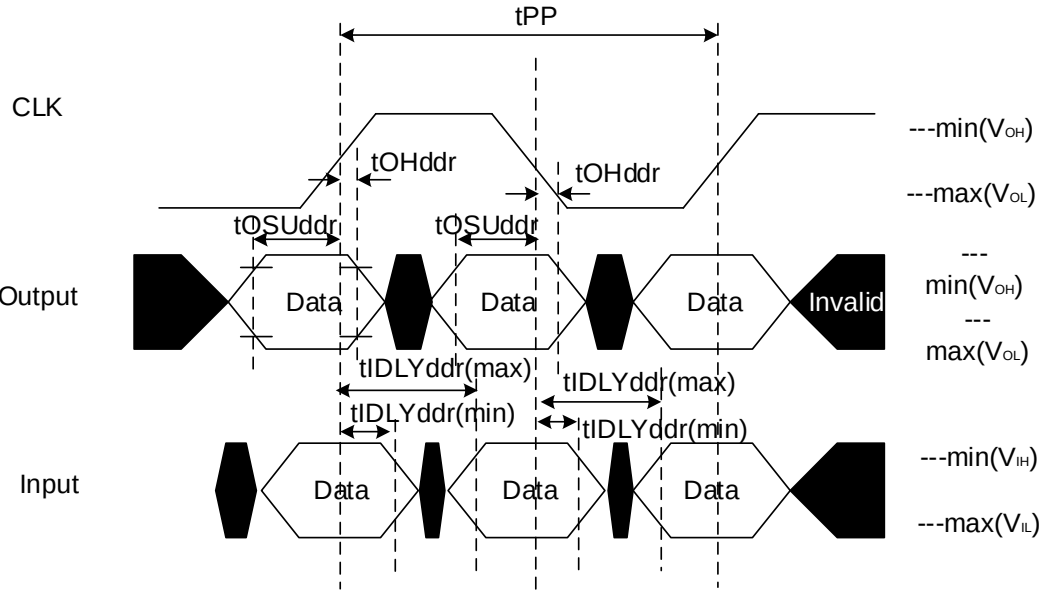


表2-62 双沿 DDR50 模式时序参数表

符号	参数	最小值	典型值	最大值	单位	备注
Input CLK						
fPP	Clock frequency Data Transfer Mode(PP)	0	-	52	MHz	fpp=1/tpp CL ≤ 30 pF Tolerance: +100 KHz
Output DAT (referenced to CLK-DDR mode)						
tOSUddr	Output set-up time	2.5	-	-	ns	CL ≤ 20 pF
tOHddr	Output hold time	2.5	-	-	ns	CL ≤ 20 pF
Input DAT (referenced to CLK-DDR mode)						
tIDLYddr	Input delay time during data transfer	1.5	-	7	ns	CL ≤ 20 pF



图2-42 HS200 模式的数据命令输入输出方向时序图

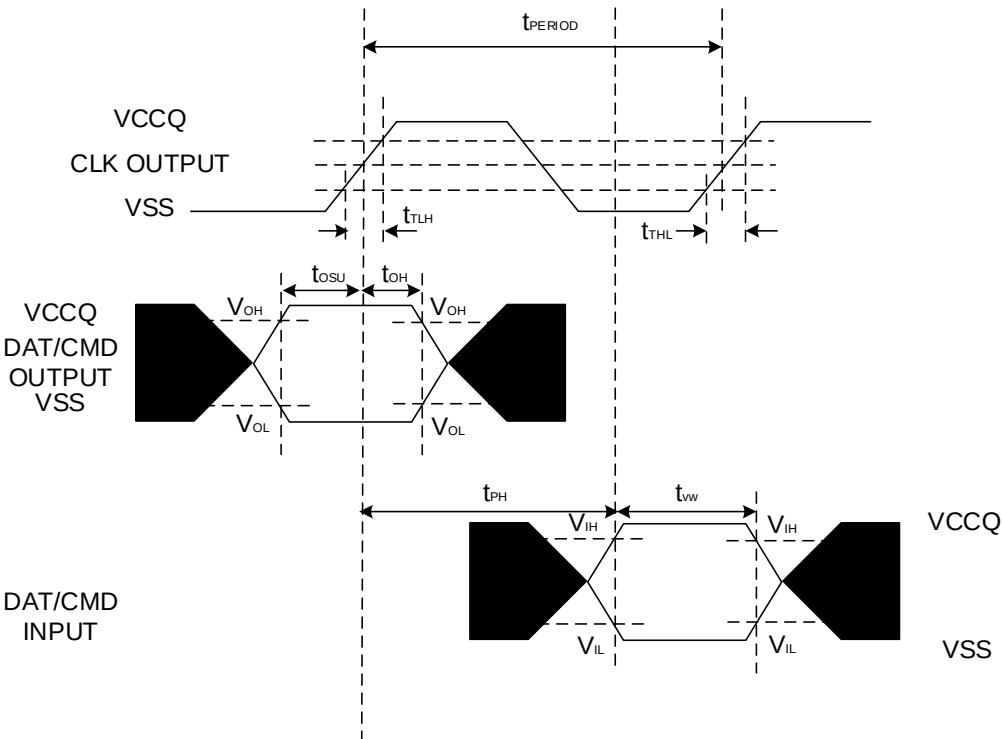


表2-63 HS200 模式输出时序参数表

符号	参数	最小值	典型值	最大值	单位	备注
t_{OSU}	Output set-up time	1.40	-	-	ns	$C_{DEVICE} \leq 6 \text{ pF}$
t_{OH}	Output hold time	0.8	-	-	ns	$C_{DEVICE} \leq 6 \text{ pF}$

表2-64 HS200 模式输入时序参数表

符号	参数	最小值	典型值	最大值	单位	备注
t_{PH}	器件输出 CMD/DAT 与接收 CLK 相位差	0	-	2	UI	UI 是单位标称时间，例如 200 MHz 时 UI=5ns。
t_{VW}	Input valid data window	0.575	-	-	UI	200 MHz 时 $t_{VW} = 2.88\text{ns}$

SDIO/MMC 时序参数表如表 2-65 所示。



表2-65 SDIO/MMC 时序参数表

速度模式	最大频率(MHz)/ 周期 (ns)		最小保 持时间	最小建 立时间	卡输出 延时的 最大值	卡时钟高电平 时间
MMC_HS200	100MHz	10 ns	0.8ns	1.4ns	-	(0.45~0.55) 时 钟周期
MMC DDR (CMD line)	50MHz	20ns	3.0ns	3.0ns	13.7ns	
MMC DDR (DAT line)	50MHz	20ns	2.5ns	2.5ns	7.0ns	
MMC_HS	50MHz	20ns	3.0ns	3.0ns	13.7ns	
SD_SDR104	100MHz	10 ns	0.8ns	1.4ns	10.0ns	
SD_SDR50	100MHz	10 ns	0.8ns	3.0ns	7.5ns	
SD_DDR50 (CMD line)	50MHz	20ns	0.8ns	6.0ns	13.7ns	
SD_DDR50 (DAT line)	50MHz	20ns	0.8ns	3.0ns	7ns	
SD_SDR25	50MHz	20ns	2.0ns	6.0ns	14ns	
SD_SDR12	25MHz	40ns	5.0ns	5.0ns	14ns	
SD_HS	50MHz	20ns	2.0ns	6.0ns	14ns	
SD_DS	25MHz	40ns	5.0ns	5.0ns	14ns	
Identification Mode	400KHz	2.5us	5.0ns	5.0ns	50ns	



目 录

3 系统	3-1
3.1 复位	3-1
3.1.1 概述	3-1
3.1.2 复位控制	3-1
3.1.3 复位配置	3-2
3.2 时钟	3-3
3.2.1 概述	3-3
3.2.2 功能框图	3-3
3.2.3 时钟资源分布	3-4
3.2.4 PLL 配置	3-7
3.2.5 CRG 寄存器概览	3-9
3.2.6 CRG 寄存器描述	3-11
3.3 处理器子系统	3-58
3.4 中断系统	3-59
3.5 系统控制器	3-60
3.5.1 功能描述	3-60
3.5.2 系统控制器寄存器概览	3-60
3.5.3 系统控制器寄存器描述	3-61
3.5.4 MISC 寄存器概览	3-68
3.5.5 MISC 寄存器描述	3-70
3.6 DMA 控制器	3-110
3.6.1 概述	3-110
3.6.2 特点	3-110
3.6.3 功能描述	3-110
3.6.4 工作方式	3-116
3.6.5 DMAC 寄存器概览	3-119
3.6.6 DMAC 寄存器描述	3-121
3.7 定时器	3-134
3.7.1 概述	3-134
3.7.2 特点	3-134



3.7.3 功能描述	3-134
3.7.4 工作方式	3-135
3.7.5 Timer 寄存器概览	3-136
3.7.6 Timer 寄存器描述	3-137
3.8 看门狗	3-140
3.8.1 概述	3-140
3.8.2 特点	3-140
3.8.3 功能描述	3-140
3.8.4 工作方式	3-142
3.8.5 WDG 寄存器概览	3-143
3.8.6 WDG 寄存器描述	3-143
3.9 实时时钟	3-146
3.9.1 概述	3-146
3.9.2 特点	3-146
3.9.3 功能描述	3-146
3.9.4 工作方式	3-146
3.9.5 RTC 寄存器概览	3-149
3.9.6 RTC 寄存器描述	3-150
3.9.7 RTC 内部寄存器描述	3-152
3.10 电源管理与低功耗模式	3-162
3.10.1 概述	3-162
3.10.2 PMC	3-163
3.10.3 时钟门控和时钟频率调整	3-174
3.10.4 模块级低功耗控制	3-174
3.10.5 AVS 功能说明	3-174



插图目录

图 3-1 复位信号控制图.....	3-1
图 3-2 时钟管理模块功能框图.....	3-4
图 3-3 时钟资源分布框图.....	3-5
图 3-4 链表在 DDR 内存放的格式示意图	3-115
图 3-5 WatchDog 应用框图.....	3-141
图 3-6 使用 PMC 时的电源网络粗略框图	3-164
图 3-7 不使用 PMC 功能时外围管脚的处理	3-164
图 3-8 PWR_BUTTON、PWR_STARTUP 第一次上电时序	3-165
图 3-9 PWR_BUTTON、PWR_STARTUP 上电时序	3-166
图 3-10 PWR_WAKEUP 上电时序（以上升沿触发为例）	3-166
图 3-11 RTC 定时中断上电时序	3-167
图 3-12 软件配置下电时序(以下电模式为 0 为例).....	3-168
图 3-13 异常下电时序(以下电顺序模式为 0 为例).....	3-169
图 3-14 复位状态示意图.....	3-170



表格目录

表 3-1 复位信号分类表.....	3-2
表 3-2 主要模块可选时钟列表.....	3-5
表 3-3 PLL 对应的配置寄存器.....	3-7
表 3-4 PLL 频率计算方法	3-8
表 3-1 CRG 寄存器概览（基址是 0x1201_0000）	3-9
表 3-2 中断源分配表	3-59
表 3-3 系统控制器寄存器概览（基址是 0x1202_0000）	3-61
表 3-4 MISC 寄存器概览（基址是 0x1203_0000）	3-68
表 3-5 DMAC 外设硬件请求线编号说明.....	3-111
表 3-6 DMAC 访问空间说明.....	3-112
表 3-7 各模块的寄存器偏移地址变量表.....	3-119
表 3-8 DMAC 寄存器概览.....	3-119
表 3-9 Timer 寄存器概览.....	3-136
表 3-10 WatchDog 寄存器概览.....	3-143
表 3-11 RTC 寄存器概览（基址是 0x1208_0000）	3-149
表 3-12 RTC 内部寄存器概览（基址是 0x00）	3-149
表 3-13 电源对接表	3-170
表 3-14 PMC 寄存器概览（基址是 0x1209_0000）	3-171



3 系统

3.1 复位

3.1.1 概述

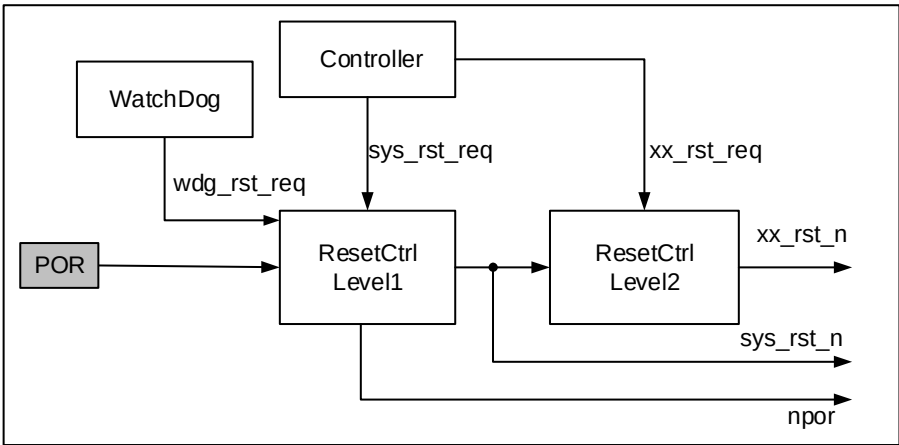
复位管理模块对整个芯片的复位、各功能模块的复位进行统一的管理，包括：

- 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

3.1.2 复位控制

复位信号控制如图 3-1 所示。

图3-1 复位信号控制图



POR	芯片内部上电复位（Power-On-Reset）模块。
wdg_rst_req	WatchDog 复位请求。
sys_rst_req	全局软复位请求信号，源自系统控制器。
xx_rst_req	子模块单独软复位请求信号，源自 CRG 控制寄存器。



xx_rst_n、sys_rst_n、npor 复位信号。

复位信号分类如表 3-1 所示。

表3-1 复位信号分类表

复位信号类型	产生方式	用途
全局硬复位 npor	外部复位和内部上电复位 POR 模块	对 SOC 系统进行全局复位。
全局软复位 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。
子模块复位 xx_rst_n	软件配置 CRG 控制寄存器 的子模块复位控制寄存器	SOC 系统各子模块的单独复位。

3.1.3 复位配置

3.1.3.1 上电复位

完成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 12 个 XIN 晶振时钟周期。
- 晶振时钟输入管脚 XIN 输入的时钟稳定。

3.1.3.2 系统复位

实现系统复位有以下途径：

- 上电复位。
- WatchDog 复位。
- 全局软复位，通过系统控制器控制。

3.1.3.3 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。



注意

- 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。

3.2 时钟

3.2.1 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：

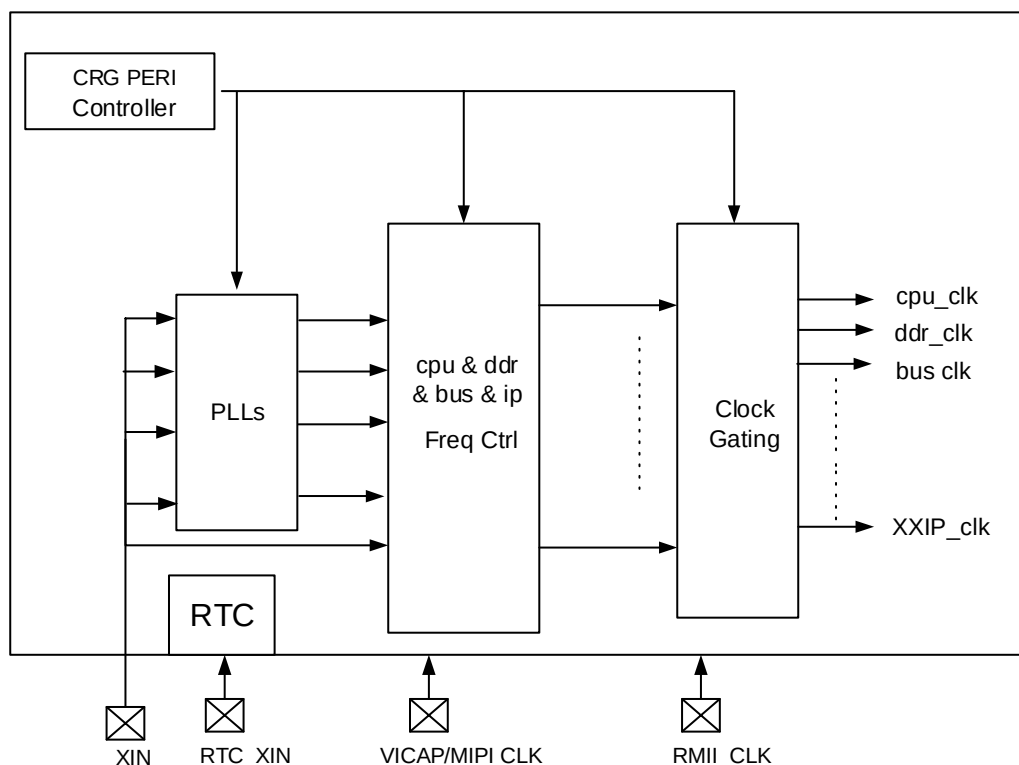
- 时钟输入的管理和控制
- 时钟分频和控制
- 生成各模块的工作时钟

3.2.2 功能框图

时钟管理模块功能框图如图 3-2 所示。



图3-2 时钟管理模块功能框图



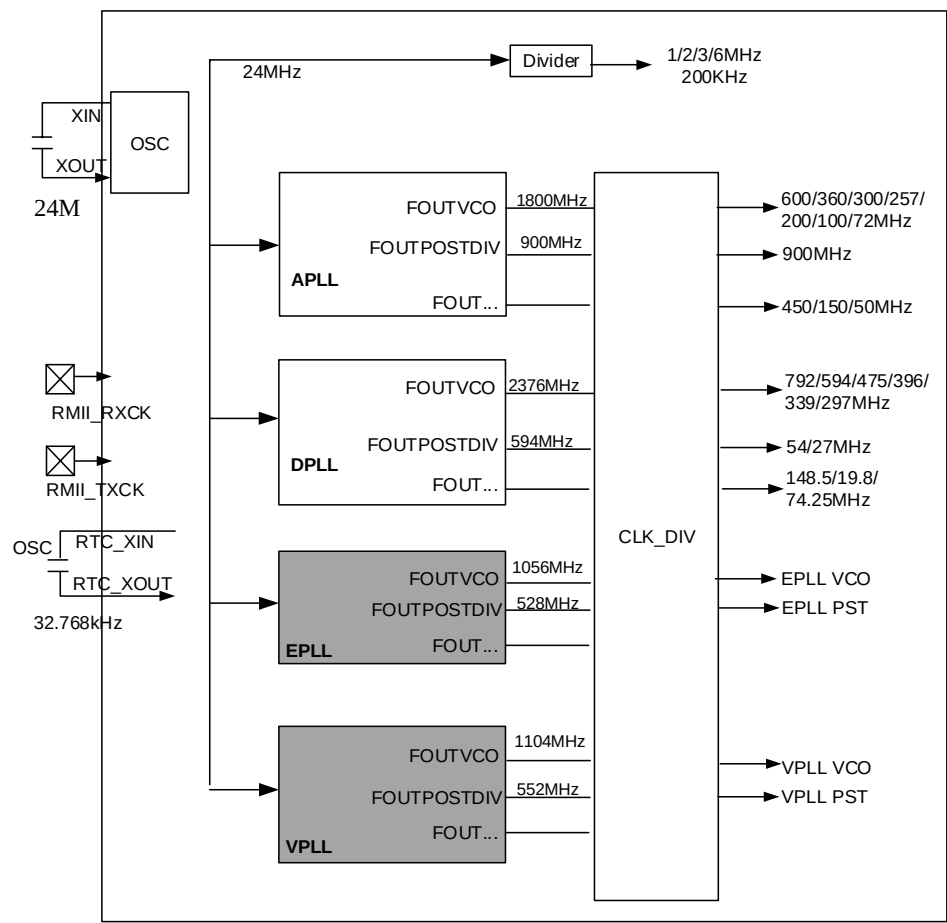
注：图中 XIN 为 PLL 输入时钟，固定连接 24MHz 晶体；RTC_XIN 为 RTC 输入时钟，固定连接 32.768KHz 晶体。

3.2.3 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，具体分布示意如图 3-3 所示。



图3-3 时钟资源分布框图



注：图中灰色的 PLL 用户可编程配置；其它 PLL 为固定配置。

主要模块可选时钟列表，如表 3-2 所示。（具体配置请参考各模块的时钟选择寄存器描述）

表3-2 主要模块可选时钟列表

模块	默认频点	所有频点
A7	24MHz	24MHz
		600MHz
		792MHz
		900MHz
DDR	24MHz	24MHz
		360MHz
		396MHz



模块	默认频点	所有频点
		450MHz
SYSAXI	24MHz	24MHz
		198MHz
		300MHz
SYSCFG	24MHz	24MHz
		148.5MHz
SYSAPB	24MHz	24MHz
		50MHz
VPSS	150MHz	150MHz
		300MHz
		396MHz
		475MHz
VIPROC	300MHz	150MHz
		300MHz
		396MHz
VICAP	300MHz	150MHz
		300MHz
		396MHz
VGS	300MHz	300MHz
		396MHz
		450MHz
		475MHz
IVE	475MHz	300MHz
		396MHz
		475MHz
GDC	300MHz	300MHz
		396MHz
		450MHz
		475MHz
FMC	24MHz	24MHz
		100MHz



模块	默认频点	所有频点
		150MHz
		163MHz
		200MHz
		257MHz
		300MHz
		396MHz
SDIO0	25MHz	25MHz
		50MHz
		100MHz
SDIO1	50MHz	25MHz
		50MHz
		100MHz
EMMC	50MHz	25MHz
		50MHz
		100MHz

3.2.4 PLL 配置

芯片内部使用了 4 个 PLL，每个 PLL 使用两组配置寄存器，对应关系如表 3-3 所示。

表3-3 PLL 对应的配置寄存器

PLL	配置寄存器 0	配置寄存器 1
APLL	PERI_CRG_PLL0	PERI_CRG_PLL1
DPLL	PERI_CRG_PLL2	PERI_CRG_PLL3
EPLL	PERI_CRG_PLL4	PERI_CRG_PLL5
VPLL	PERI_CRG_PLL6	PERI_CRG_PLL7

PLL 均采用管脚 XIN 输入的晶振时钟作为输入时钟，PLL 输出频率配置方法如表 3-4 所示。



表3-4 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	$FREF \times (fbdiv + \frac{frac}{2^{24}}) / refdiv$	PLL 工作频率，要求大于等于 800MHz，且小于等于 3.2GHz
FOUTPOSTDIV	$FOUTVCO / (pstdiv1 \times pstdiv2)$	$pstdiv1 \geq pstdiv2$
FOUT1PH0	$FOUTVCO / (pstdiv1 \times pstdiv2 \times 2)$	-
FOUT2	$FOUTVCO / (pstdiv1 \times pstdiv2 \times 4)$	-
FOUT3	$FOUTVCO / (pstdiv1 \times pstdiv2 \times 6)$	-
FOUT4	$FOUTVCO / (pstdiv1 \times pstdiv2 \times 8)$	-
备注： Fbdiv：倍频系数的整数部分； frac：倍频系数的小数部分； refdiv：参考时钟分频系数； pstdiv1：第一级输出分频系数； pstdiv2：第二级输出分频系数。 各 PLL 的配置系数请参见表 3-3 对应配置寄存器的相应 Bit 位。		

例如，需要配置 APLL FOUTVCO 输出 1200MHz，FOUTPOSTDIV 输出频率为 FOUTVCO 的一半，计算过程如下：

- 配置 FOUTPOSTDIV 输出频率为 FOUTVCO 的一半。根据 $FOUTPOSTDIV = FOUTVCO / (pstdiv1 \times pstdiv2)$ ，取 $pstdiv2 = 1$ ， $pstdiv1 = 2$
- 配置 FOUTVCO 输出 1200MHz，根据 $FREF \times (fbdiv + \frac{frac}{2^{24}}) / refdiv = 1200MHz$ ，其中 $FREF = 24MHz$ ，若取 $refdiv = 1$ ，计算出 $fbdiv = 50$ ， $frac = 0$

3.2.4.1 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. 等待 0.1ms 后观测 PLL LOCK 指示寄存器 [PERI_CRG_PLL122\[10:0\]](#)。
- 步骤 4. 如果 PLL LOCK 寄存器为 LOCK 状态，PLL 配置完成，否则回到步骤 3，直到 PLL LOCK 寄存器为 LOCK 状态。

---结束



3.2.5 CRG 寄存器概览

CRG 寄存器概览如表 3-1 所示。

表3-1 CRG 寄存器概览（基址是 0x1201_0000）

偏移地址	名称	描述	页码
0x0000	PERI_CRG_PLL0	APLL 配置寄存器 0	3-11
0x0004	PERI_CRG_PLL1	APLL 配置寄存器 1	3-11
0x0008	PERI_CRG_PLL2	DPLL 配置寄存器 0	3-12
0x000C	PERI_CRG_PLL3	DPLL 配置寄存器 1	3-13
0x0010	PERI_CRG_PLL4	EPLL 配置寄存器 0	3-13
0x0014	PERI_CRG_PLL5	EPLL 配置寄存器 1	3-14
0x0018	PERI_CRG_PLL6	VPLL 配置寄存器 0	3-15
0x001C	PERI_CRG_PLL7	VPLL 配置寄存器 1	3-15
0x0078	PERI_CRG30	CPU 频率模式及复位配置寄存器（安全模式下，非安全 CPU 不可写）	3-16
0x007C	PERI_CRG31	DDR 时钟及复位配置寄存器（安全模式下，非安全 CPU 不可写）	3-17
0x0080	PERI_CRG32	SOC 时钟选择寄存器	3-18
0x00A0	PERI_CRG40	媒体功能模块频率配置寄存器	3-19
0x00A4	PERI_CRG41	VEDU 时钟及软复位控制寄存器	3-20
0x00A8	PERI_CRG42	VPSS 时钟及软复位控制寄存器	3-20
0x00AC	PERI_CRG43	VGS 时钟及软复位控制寄存器	3-21
0x00B0	PERI_CRG44	JPGE 时钟及软复位控制寄存器	3-21
0x00B4	PERI_CRG45	GDC 时钟及软复位控制寄存器	3-22
0x00B8	PERI_CRG46	TDE 时钟及软复位控制寄存器	3-22
0x00C0	PERI_CRG48	IVE 时钟及软复位控制寄存器	3-23
0x00C4	PERI_CRG49	JPGD 时钟及软复位控制寄存器	3-23
0x00F0	PERI_CRG60	SENSOR 时钟及软复位控制寄存器	3-24
0x00F4	PERI_CRG61	VICAP 时钟及软复位控制寄存器	3-25
0x00F8	PERI_CRG62	MIPI RX 时钟及软复位控制寄存器	3-27
0x00FC	PERI_CRG63	VIPROC 时钟及软复位控制寄存器	3-28



偏移地址	名称	描述	页码
0x0100	PERI_CRG64	ISP 时钟及软复位控制寄存器	3-29
0x0104	PERI_CRG65	LCD 分频器配置寄存器	3-31
0x0108	PERI_CRG66	VDP 时钟及软复位控制寄存器	3-31
0x010C	PERI_CRG67	MIPI TX 时钟及软复位控制寄存器	3-34
0x0110	PERI_CRG68	HDMI TX 控制器时钟及软复位控制寄存器	3-34
0x0114	PERI_CRG69	HDMI TX PHY 时钟及软复位控制寄存器	3-36
0x0140	PERI_CRG80	USB2.0 相关的时钟及软复位控制寄存器	3-36
0x0144	PERI_CRG81	FMC 相关的时钟及软复位控制寄存器	3-37
0x0148	PERI_CRG82	EMMC 相关的时钟及软复位控制寄存器	3-38
0x014C	PERI_CRG83	EMMC DLL 相关的时钟及软复位控制寄存器	3-39
0x0150	PERI_CRG84	EMMC DLL 状态指示寄存器	3-40
0x0154	PERI_CRG85	SDIO1 相关的时钟及软复位控制寄存器	3-40
0x0158	PERI_CRG86	SDIO1 DLL 相关的时钟及软复位控制寄存器	3-41
0x015C	PERI_CRG87	SDIO1 DLL 状态指示寄存器	3-42
0x0160	PERI_CRG88	SDIO0 相关的时钟及软复位控制寄存器	3-43
0x0164	PERI_CRG89	SDIO0 DLL 相关的时钟及软复位控制寄存器	3-43
0x0168	PERI_CRG90	SDIO0 DLL 状态指示寄存器	3-45
0x016C	PERI_CRG91	ETH 接口相关的时钟及软复位控制寄存器	3-45
0x0190	PERI_CRG100	GZIP 时钟及软复位控制寄存器	3-46
0x0194	PERI_CRG101	DMAC 相关的时钟及软复位控制寄存器	3-47
0x0198	PERI_CRG102	DDRT 相关的时钟及软复位控制寄存器	3-47
0x019C	PERI_CRG103	AIAO 时钟复位控制寄存器	3-48
0x01A0	PERI_CRG104	SPACC/RSA/TRNG/KLAD 相关的时钟及软复位控制寄存器（安全模式下，非安全 CPU 不可写）	3-48
0x01A4	PERI_CRG105	AIAO MCLK2 时钟配置寄存器	3-49



偏移地址	名称	描述	页码
0x01A8	PERI_CRG106	AIAO MCLK1 时钟配置寄存器	3-50
0x01AC	PERI_CRG107	AIAO MCLK0 时钟配置寄存器	3-50
0x01B8	PERI_CRG110	I2C/UART 时钟及复位控制寄存器	3-50
0x01BC	PERI_CRG111	ADC/UART/SPI/OTP/IR/PWM 相关的时钟及软复位控制寄存器	3-53
0x01E0	PERI_CRG120	SOC 频率指示寄存器	3-56
0x01E4	PERI_CRG_PLL121	复位状态寄存器	3-57
0x01E8	PERI_CRG_PLL122	PLL LOCK 状态寄存器	3-58

3.2.6 CRG 寄存器描述

PERI_CRG_PLL0

PERI_CRG_PLL0 为 APLL 配置寄存器 0。

Offset Address: 0x0000 Total Reset Value: 0x1200_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30:28]	RW	apll_postdiv2	APLL 第二级输出分频系数。	0x1
[27]	-	reserved	保留。	0x0
[26:24]	RW	apll_postdiv1	APLL 第一级输出分频系数。	0x2
[23:0]	RW	apll_frac	APLL 倍频系数小数部分。	0x000000

PERI_CRG_PLL1

PERI_CRG_PLL1 为 APLL 配置寄存器 1。

Offset Address: 0x0004 Total Reset Value: 0x0100_104B

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x0
[26]	RW	apll_bypass	APLL 时钟分频旁路(bypass)控制。 0: 非旁路; 1: 旁路。	0x0



Bits	Access	Name	Description	Reset
[25]	RW	apll_dacpd	APLL 测试信号控制。 0: 正常工作状态; 1: power down 工作状态。	0x0
[24]	RW	apll_dsmpd	APLL 小数分频控制。 0: 小数模式; 1: 整数模式。	0x1
[23]	RW	apll_pd	APLL Power Down 控制。 0: 正常工作状态; 1: power down 工作状态。	0x0
[22]	-	reserved	保留。	0x0
[21]	RW	apll_postdivpd	APLL POSTDIV 输出 Power Down 控制。 0: 正常时钟输出; 1: 不输出时钟。 注意: VCO 时钟也会被 PowerDown。	0x0
[20]	RW	apll_fout4phasepd	APLL FOUT 输出 Power Down 控制。 0: 正常时钟输出; 1: 不输出时钟。	0x0
[19:18]	-	reserved	保留。	0x0
[17:12]	RW	apll_refdiv	APLL 参考时钟分频系数。	0x01
[11:0]	RW	apll_fbdiv	APLL 倍频系数整数部分。	0x04B

PERI_CRG_PLL2

PERI_CRG_PLL2 为 DPLL 配置寄存器 0。

Offset Address: 0x0008 Total Reset Value: 0x1400_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30:28]	RW	dppl_postdiv2	DPLL 第二级输出分频系数。	0x1
[27]	-	reserved	保留。	0x0
[26:24]	RW	dppl_postdiv1	DPLL 第一级输出分频系数。	0x4
[23:0]	RW	dppl_frac	DPLL 倍频系数小数部分。	0x000000



PERI_CRG_PLL3

PERI_CRG_PLL3 为 DPLL 配置寄存器 1。

Offset Address: 0x000C Total Reset Value: 0x0100_1063

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x0
[26]	RW	dpll_bypass	DPLL 时钟分频旁路(bypass)控制。 0: 非旁路; 1: 旁路。	0x0
[25]	RW	dpll_dacpd	DPLL 测试信号控制。 0: 正常工作状态; 1: power down 工作状态。	0x0
[24]	RW	dpll_dsmpd	DPLL 小数分频控制。 0: 小数模式; 1: 整数模式。	0x1
[23]	RW	dpll_pd	DPLL Power Down 控制。 0: 正常工作状态; 1: power down 工作状态。	0x0
[22]	-	reserved	保留。	0x0
[21]	RW	dpll_postdivpd	DPLL POSTDIV 输出 Power Down 控制。 0: 正常时钟输出; 1: 不输出时钟。 注意: VCO 时钟也会被 PowerDown。	0x0
[20]	RW	dpll_fout4phasepd	DPLL FOUT 输出 Power Down 控制。 0: 正常时钟输出; 1: 不输出时钟。	0x0
[19:18]	-	reserved	保留。	0x0
[17:12]	RW	dpll_refdiv	DPLL 参考时钟分频系数。	0x01
[11:0]	RW	dpll_fbdiv	DPLL 倍频系数整数部分。	0x063

PERI_CRG_PLL4

PERI_CRG_PLL4 为 EPLL 配置寄存器 0。



Offset Address: 0x0010 Total Reset Value: 0x1200_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30:28]	RW	epll_postdiv2	EPLL 第二级输出分频系数。	0x1
[27]	-	reserved	保留。	0x0
[26:24]	RW	epll_postdiv1	EPLL 第一级输出分频系数。	0x2
[23:0]	RW	epll_frac	EPLL 倍频系数小数部分。	0x000000

PERI_CRG_PLL5

PERI_CRG_PLL5 为 EPLL 配置寄存器 1。

Offset Address: 0x0014 Total Reset Value: 0x0100_102C

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x0
[26]	RW	epll_bypass	EPLL 时钟分频旁路(bypass)控制。 0: 非旁路; 1: 旁路。	0x0
[25]	RW	epll_dacpd	EPLL 测试信号控制。 0: 正常工作状态; 1: power down 工作状态。	0x0
[24]	RW	epll_dsmpd	EPLL 小数分频控制。 0: 小数模式; 1: 整数模式。	0x1
[23]	RW	epll_pd	EPLL Power Down 控制。 0: 正常工作状态; 1: power down 工作状态。	0x0
[22]	-	reserved	保留。	0x0
[21]	RW	epll_postdivpd	EPLL POSTDIV 输出 Power Down 控制。 0: 正常时钟输出; 1: 不输出时钟。 注意: VCO 时钟也会被 PowerDown。	0x0
[20]	RW	epll_fout4phasepd	EPLL FOUT 输出 Power Down 控制。 0: 正常时钟输出;	0x0



Bits	Access	Name	Description	Reset
			1: 不输出时钟。	
[19:18]	-	reserved	保留。	0x0
[17:12]	RW	epll_refdiv	EPLL 参考时钟分频系数。	0x01
[11:0]	RW	epll_fbdiv	EPLL 倍频系数整数部分。	0x02C

PERI_CRG_PLL6

PERI_CRG_PLL6 为 VPLL 配置寄存器 0。

Offset Address: 0x0018 Total Reset Value: 0x1200_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30:28]	RW	vpll_postdiv2	VPLL 第二级输出分频系数。	0x1
[27]	-	reserved	保留。	0x0
[26:24]	RW	vpll_postdiv1	VPLL 第一级输出分频系数。	0x2
[23:0]	RW	vpll_frac	VPLL 倍频系数小数部分。	0x000000

PERI_CRG_PLL7

PERI_CRG_PLL7 为 VPLL 配置寄存器 1。

Offset Address: 0x001C Total Reset Value: 0x0100_102E

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x0
[26]	RW	vpll_bypass	VPLL 时钟分频旁路(bypass)控制。 0: 非旁路; 1: 旁路。	0x0
[25]	RW	vpll_dacpd	VPLL 测试信号控制。 0: 正常工作状态; 1: power down 工作状态。	0x0
[24]	RW	vpll_dsmpd	VPLL 小数分频控制。 0: 小数模式; 1: 整数模式。	0x1



Bits	Access	Name	Description	Reset
[23]	RW	vpll_pd	VPLL Power Down 控制。 0: 正常工作状态; 1: power down 工作状态。	0x0
[22]	-	reserved	保留。	0x0
[21]	RW	vpll_postdivpd	VPLL POSTDIV 输出 Power Down 控制。 0: 正常时钟输出; 1: 不输出时钟。 注意: VCO 时钟也会被 PowerDown。	0x0
[20]	RW	vpll_fout4phasepd	VPLL FOUT 输出 Power Down 控制。 0: 正常时钟输出; 1: 不输出时钟。	0x0
[19:18]	-	reserved	保留。	0x0
[17:12]	RW	vpll_refdiv	VPLL 参考时钟分频系数。	0x01
[11:0]	RW	vpll_fbdiv	VPLL 倍频系数整数部分。	0x02E

PERI_CRG30

PERI_CRG30 为 CPU 频率模式及复位配置寄存器（安全模式下，非安全 CPU 不可写）。

Offset Address: 0x0078 Total Reset Value: 0x0000_0005

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留。	0x000000
[10]	RW	cs_srst_req	CS 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[9]	RW	socdbg_srst_req	SOCDBG 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[8]	RW	l2_srst_req	L2 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[7:5]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[4]	RW	arm_dbg1_srst_req	ARM DBG1 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[3]	RW	arm_dbg0_srst_req	ARM DBG0 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[2]	RW	arm_core1_srst_req	ARM CORE1 的软复位请求。 0: 撤消复位; 1: 复位。	0x1
[1]	RW	arm_core0_srst_req	ARM CORE0 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[0]	RW	pclkdbg_cken	A7 PCLKDBG 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x1

PERI_CRG31

PERI_CRG31 为 DDR 时钟及复位配置寄存器（安全模式下，非安全 CPU 不可写）。

Offset Address: 0x007C Total Reset Value: 0x0000_00F0

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RW	ddr_cken	DDR 门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[6]	RW	ddr_apb_cken	DDR APB 门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[5]	RW	ddr_hipack_cken	DDR HiPACK 门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[4]	RW	ddr_cfg_cken	DDR CFG 门控配置寄存器。 0: 关闭时钟;	0x1



Bits	Access	Name	Description	Reset
			1: 打开时钟。	
[3:2]	-	reserved	保留。	0x0
[1]	RW	ddr_apb_srst_req	DDR APB 软复位请求。 0: 撤消复位; 1: 复位。	0x0
[0]	RW	ddr_hipack_srst_req	DDR HiPACK 软复位请求。 0: 撤消复位; 1: 复位。	0x0

PERI_CRG32

PERI_CRG32 为 SOC 时钟选择寄存器。

Offset Address: 0x0080 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留。	0x000000
[10]	RW	sysapb_cksel	SYSAPB 时钟选择。 0: 晶振时钟; 1: 50MHz。	0x0
[9]	-	reserved	保留。	0x0
[8]	RW	syscfg_cksel	SYSCFG 时钟选择。 0: 晶振时钟; 1: 148.5MHz。	0x0
[7:6]	RW	sysaxi_cksel	SYSAXI 时钟选择。 00: 晶振时钟; 01: 200MHz; 10: 300MHz; 11: 保留。	0x0
[5:3]	RW	ddr_cksel	DDR 时钟选择。 000: 晶振时钟; 001: 360MHz; 010: 396MHz; 011: 450MHz; 其他: 保留。	0x0



Bits	Access	Name	Description	Reset
[2:0]	RW	a7_cksel	A7 时钟选择。 000: 晶振时钟; 001: 600MHz; 010: 792MHz; 011: 900MHz; 其他: 保留。	0x0

PERI_CRG40

PERI_CRG40 为媒体功能模块频率配置寄存器。

Offset Address: 0x00A0 Total Reset Value: 0x0079_2CF0

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:21]	RW	vpss_cksel	VPSS 时钟选择。 000: 475MHz; 001: 396MHz (需要先配置 PERI_CRG61[25] 为 1); 010: 300MHz; 011: 150MHz; 其他: 保留。	0x3
[20]	-	reserved	保留。	0x1
[19:18]	-	reserved	保留。	0x2
[17:15]	RW	ive_cksel	IVE 时钟选择。 000: 475MHz; 001: 396MHz; 010: 300MHz; 其他: 保留。	0x2
[14:13]	-	reserved	保留。	0x1
[12:10]	-	reserved	保留。	0x3
[9]	-	reserved	保留。	0x0
[8:6]	RW	vgs_cksel	VGS 时钟选择。 000: 475MHz; 001: 450MHz;	0x3



Bits	Access	Name	Description	Reset
			010: 396MHz; 011: 300MHz; 其他: 保留。	
[5:4]	RW	gdc_cksel	GDC 时钟选择。 00: 475MHz; 01: 450MHz; 10: 396MHz; 11: 300MHz; 其他: 保留。	0x3
[3:1]	-	reserved	保留。	0x0
[0]	RW	pll_50m_cksel	50MHz 时钟 PLL 源头选择。 0: 50MHz 来自 APLL 分频得到; 1: 50MHz 来自 EPLL 分频得到。	0x0

PERI_CRG41

PERI_CRG41 为 VEDU 时钟及软复位控制寄存器。

Offset Address: 0x00A4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x000000
[2]	RW	sed_cksel	SED 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[1]	RW	vedu_cken	VEDU 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[0]	RW	vedu_srst_req	VEDU 的软复位请求。 0: 撤消复位; 1: 复位。	0x0

PERI_CRG42

PERI_CRG42 为 VPSS 时钟及软复位控制寄存器。



Offset Address: 0x00A8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2]	RW	vpss_online_cksel	VPSS 在线模式时钟选择。 0: clk_vpss(即选择离线时钟); 1: clk_viproc; 在线配合 MIST_CTRL 那边的 viproc_vpss_online_mode 寄存器一起使用。	0x0
[1]	RW	vpss_cken	VPSS 时钟门控配置寄存器, 0: 关闭时钟; 1: 打开时钟。	0x0
[0]	RW	vpss_srst_req	VPSS 的软复位请求。 0: 撤消复位; 1: 复位。	0x0

PERI_CRG43

PERI_CRG43 为 VGS 时钟及软复位控制寄存器。

Offset Address: 0x00AC Total Reset Value: 0x0000_0002

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	vgs_cken	VGS 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[0]	RW	vgs_srst_req	VGS 的软复位请求。 0: 撤消复位; 1: 复位。	0x0

PERI_CRG44

PERI_CRG44 为 JPGE 时钟及软复位控制寄存器。

Offset Address: 0x00B0 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	jpgc_cken	JPGC 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[0]	RW	jpgc_srst_req	JPGC 的软复位请求。 0: 撤消复位; 1: 复位。	0x0

PERI_CRG45

PERI_CRG45 为 GDC 时钟及软复位控制寄存器。

Offset Address: 0x00B4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	gdc_cken	GDC 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[0]	RW	gdc_srst_req	GDC 的软复位请求。 0: 撤消复位; 1: 复位。	0x0

PERI_CRG46

PERI_CRG46 为 TDE 时钟及软复位控制寄存器。

Offset Address: 0x00B8 Total Reset Value: 0x0000_0000

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



[0]	RW	tde_srst_req	TDE 的软复位请求。 0: 撤消复位; 1: 复位。	0x0

PERI_CRG48

PERI_CRG48 为 IVE 时钟及软复位控制寄存器。

Offset Address: 0x00C0 Total Reset Value: 0x0000_0004

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2]	RW	ive_cts_cken	IVE 时钟树门控配置寄存器。(低功耗配置, 与 ive_cken 同时打开, IVE 才有时钟)。 0: 关闭时钟; 1: 打开时钟。	0x1
[1]	RW	ive_cken	IVE 时钟门控配置寄存器。(ive_cts_cken 同时打开, IVE 才有时钟) 0: 关闭时钟; 1: 打开时钟。	0x0
[0]	RW	ive_srst_req	IVE 的软复位请求。 0: 撤消复位; 1: 复位。	0x0

PERI_CRG49

PERI_CRG49 为 JPGD 时钟及软复位控制寄存器。

Offset Address: 0x00C4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	jpgd_cken	JPGD 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[0]	RW	jpgd_srst_req	JPGD 的软复位请求。 0: 撤消复位;	0x0



Bits	Access	Name	Description	Reset
			1: 复位。	

PERI_CRG60

PERI_CRG60 为 SENSOR 时钟及软复位控制寄存器。

Offset Address: 0x00F0 Total Reset Value: 0x0000_0041

Bits	Access	Name	Description	Reset
[31:12]	-	reserved	保留。	0x000000
[11:8]	RW	sensor1_cksel	SENSOR1 CLKOUT 时钟配置寄存器。 0000: 74.25MHz; 0001: 72MHz; 0010: 54MHz; 0011: 50MHz; 01XX: 24MHz; 1000: 37.125MHz; 1001: 36MHz; 1010: 27MHz; 1011: 25MHz; 11XX: 12MHz;	0x0
[7]	RW	sensor1_srst_req	SENSOR1 软复位请求。 0: 撤消复位; 1: 复位。	0x0
[6]	RW	sensor1_cken	SENSOR1 CLKOUT 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x1
[5:2]	RW	sensor0_cksel	SENSOR0 CLKOUT 时钟配置寄存器。 0000: 74.25MHz; 0001: 72MHz; 0010: 54MHz; 0011: 50MHz; 01XX: 24MHz; 1000: 37.125MHz; 1001: 36MHz; 1010: 27MHz;	0x0



Bits	Access	Name	Description	Reset
			1011: 25MHz; 11XX: 12MHz;	
[1]	RW	sensor0_srst_req	SENSOR0 软复位请求。 0: 撤消复位; 1: 复位。	0x0
[0]	RW	sensor0_cken	SENSOR0 CLKOUT 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x1

PERI_CRG61

PERI_CRG61 为 VICAP 时钟及软复位控制寄存器。

Offset Address: 0x00F4 Total Reset Value: 0x1F92_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x07
[25]	RW	Vicap_400m_cken	VICAP 396MHz 时钟门控（低功耗设置，必须打开此门控，VICAP、ISP、VPSS、VIPROC 才能选 396MHz）。 0: 时钟关闭; 1: 时钟打开。	0x1
[24]	-	reserved	保留。	0x1
[23:21]	RW	vicap_ppc_cksel	VICAP PPC 时钟选择。 011: 396MHz（需要先配置 PERI_CRG61 [25]为 1）; 100: 300MHz; 101: 150MHz; 其他: 保留。	0x4
[20:18]	RW	vicap_p1_cksel	VICAP Port1 时钟选择。 011: 396MHz（需要先配置 PERI_CRG61 [25]为 1）; 100: 300MHz; 101: 150MHz; 111: 外部管脚（CMOS）输入时钟; 其他: 保留。	0x4



Bits	Access	Name	Description	Reset
[17:15]	RW	vicap_p0_cksel	VICAP Port0 时钟选择。 011: 396MHz（需要先配置 PERI_CRG61 [25]为 1）； 100: 300MHz； 101: 150MHz； 其他：保留。	0x4
[14]	RW	vicap_cmos_pctrl	CMOS 输入的时钟相位选择 0: 时钟不取反； 1: 时钟取反。	0x0
[13]	RW	vicap_bus_srst_req	VICAP BUS 软复位请求。 0: 不复位； 1: 复位。	0x0
[12]	RW	vicap_ppc_srst_req	VICAP PPC 软复位请求。 0: 不复位； 1: 复位。	0x0
[11]	RW	vicap_p1_srst_req	VICAP Port1 软复位请求。 0: 不复位； 1: 复位。	0x0
[10]	RW	vicap_p0_srst_req	VICAP Port0 软复位请求。 0: 不复位； 1: 复位。	0x0
[9]	RW	vicap_bus_cken	VICAP BUS 时钟门控。 0: 时钟关闭； 1: 时钟打开。	0x0
[8]	RW	vicap_ppc_cken	VICAP PPC 时钟门控。 0: 时钟关闭； 1: 时钟打开。	0x0
[7]	RW	vicap_p1_cken	VICAP Port1 时钟门控。 0: 时钟关闭； 1: 时钟打开。	0x0
[6]	RW	vicap_p0_cken	VICAP Port0 时钟门控。 0: 时钟关闭； 1: 时钟打开。	0x0
[5]	RW	vicap_ppc_ch3_cken	VICAP PPC CH3 时钟门控（仅 RAW 格式	0x0



Bits	Access	Name	Description	Reset
			数据)。 0: 时钟关闭; 1: 时钟打开。	
[4]	RW	vicap_ppc_ch2_cken	VICAP PPC CH2 时钟门控 (仅 YUV 格式数据)。 0: 时钟关闭; 1: 时钟打开。	0x0
[3]	RW	vicap_ppc_ch1_cken	VICAP PPC CH1 时钟门控 (仅 RAW 格式数据)。 0: 时钟关闭; 1: 时钟打开。	0x0
[2]	RW	vicap_ppc_ch0_cken	VICAP PPC CH0 时钟门控 (仅 RAW 格式数据)。 0: 时钟关闭; 1: 时钟打开。	0x0
[1]	RW	vicap_ppc_pt1_cken	VICAP PPC PT1 时钟门控 (仅 CMOS 输入)。 0: 时钟关闭; 1: 时钟打开。	0x0
[0]	RW	vicap_ppc_pt0_cken	VICAP PPC PT0 时钟门控。 0: 时钟关闭; 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: 0x00F8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RW	mipi_hs_pctrl	MIPI RX PHY HS 接口时钟相位控制。默认正向。 0: 时钟不取反; 1: 时钟取反。	0x0
[6]	RW	mipi_bus_srst_req	MIPI RX BUS 逻辑软复位请求。	0x0



Bits	Access	Name	Description	Reset
			0: 不复位; 1: 复位。	
[5]	RW	mipi_pix1_core_srst_req	MIPI RX PIX1 的 CORE 逻辑软复位请求。 0: 不复位; 1: 复位。	0x0
[4]	RW	mipi_pix0_core_srst_req	MIPI RX PIX0 的 CORE 逻辑软复位请求。 0: 不复位; 1: 复位。	0x0
[3]	RW	mipi_bus_cken	MIPI RX 总线逻辑 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[2]	RW	mipi_pix1_cken	MIPI RX PIX1 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[1]	RW	mipi_pix0_cken	MIPI RX PIX0 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[0]	RW	cil_cken	MIPI RX CIL 逻辑 时钟门控。 0: 时钟关闭; 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: 0x00FC Total Reset Value: 0x0000_0006

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x00000000
[5]	RW	viprocsrst_req	VIPROC 软复位请求。 0: 不复位; 1: 复位。	0x0
[4]	RW	viprocsken	VIPROC 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0



Bits	Access	Name	Description	Reset
[3]	RW	viproc_online_cksel	VIPROC 在线模式选择。 0: 离线, clk_viproc; 1: 在线, clk_isp0; 配合 MISC_CTRL6 的 vicap_viproc_online_mode 寄存器一起使用。	0x0
[2]	-	reserved	保留。	0x1
[1:0]	RW	viproc_cksel	VIPROC 离线模式时钟选择。 01: 396MHz (需要先配置 PERI_CRG61 [25]为 1) ; 10: 300MHz; 11: 150MHz。 其他: 保留。	0x2

PERI_CRG64

PERI_CRG64 为 ISP 时钟及软复位控制寄存器。

Offset Address: 0x0100 Total Reset Value: 0x0000_0924

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23]	RW	isp3_cken	ISP3 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[22]	RW	isp2_cken	ISP2 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[21]	RW	isp1_cken	ISP1 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[20]	RW	isp0_cken	ISP0 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[19]	RW	isp3_core_srst_req	ISP3 core 软复位请求。 0: 不复位;	0x0



Bits	Access	Name	Description	Reset
			1: 复位。	
[18]	RW	isp2_core_srst_req	ISP2 core 软复位请求。 0: 不复位; 1: 复位。	0x0
[17]	RW	isp1_core_srst_req	ISP1 core 软复位请求。 0: 不复位; 1: 复位。	0x0
[16]	RW	isp0_core_srst_req	ISP0 core 软复位请求。 0: 不复位; 1: 复位。	0x0
[15]	RW	isp3_cfg_srst_req	ISP3 cfg 软复位请求。 0: 不复位; 1: 复位。	0x0
[14]	RW	isp2_cfg_srst_req	ISP2 cfg 软复位请求。 0: 不复位; 1: 复位。	0x0
[13]	RW	isp1_cfg_srst_req	ISP1 cfg 软复位请求。 0: 不复位; 1: 复位。	0x0
[12]	RW	isp0_cfg_srst_req	ISP0 cfg 软复位请求。 0: 不复位; 1: 复位。	0x0
[11:9]	RW	isp3_cksel	ISP3 时钟选择。 011: 396MHz (需要先配置 PERI_CRG61[25] 为 1) ; 100: 300MHz; 101: 150MHz; 其他: 保留。	0x4
[8:6]	RW	isp2_cksel	ISP2 时钟选择。 011: 396MHz (需要先配置 PERI_CRG61[25] 为 1) ; 100: 300MHz; 101: 150MHz; 其他: 保留。	0x4



Bits	Access	Name	Description	Reset
[5:3]	RW	isp1_cksel	ISP1 时钟选择。 011: 396MHz（需要先配置 PERI_CRG61 [25]为 1）； 100: 300MHz； 101: 150MHz； 其他：保留。	0x4
[2:0]	RW	isp0_cksel	ISP0 时钟选择。 011: 396MHz（需要先配置 PERI_CRG61 [25]为 1）； 100: 300MHz； 101: 150MHz； 其他：保留。	0x4

PERI_CRG65

PERI_CRG65 为 LCD 分频器配置寄存器。

Offset Address: 0x0104 Total Reset Value: 0x000F_FFFF

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27]	RW	lcd_cken	LCD 分频器时钟门控配置寄存器。 0: 关闭时钟； 1: 打开时钟。	0x0
[26:0]	RW	lcd_div	LCD 分频时钟，可配置。 假定目标频率 X(MHZ)，则 $\text{lcd\_mclk\_div} = (X/1188) * 2^{27}$ ；最大 75MHz。	0x0FFFFFFF

PERI_CRG66

PERI_CRG66 为 VDP 时钟及软复位控制寄存器。

Offset Address: 0x0108 Total Reset Value: 0x0030_4000

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29:25]	RW	vdp_hd_hdmi_clk_div	VDP HD 通道 hdmi 时钟分频配置。	0x00



Bits	Access	Name	Description	Reset
			N: N+1 分频。	
[24:21]	RW	vdp_hd_cksel	VDP HD 时钟源选择。 0000: VPLL PST 时钟; 0001: VPLL FOUT4 时钟; 0010: 297MHz; 0011: 74.25MHz; 0100: 108MHz; 0101: 27MHz; 0110: LCD 分频器时钟; 1010: 148.5MHz; 1011: 37.125MHz; 1100: 54MHz; 1101: 13.5MHz; 其他: 保留。	0x1
[20]	RW	vdp_out_pctrl	VDP 输出时钟(芯片输出时钟)相位控制。 默认取反 0: 时钟不取反; 1: 时钟取反。	0x1
[19]	RW	vdp_mipi_tx_cken	VDP_MIPI TX 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[18]	RW	vdp_hdmi_tx_cken	VDP_HDMI TX 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[17:15]	-	reserved	保留。	0x0
[14]	RW	vdp_hd_ppc_cksel	VDP HD 工作时钟选择。 0: 300MHz; 1: 150MHz。	0x1
[13:12]	RW	hd_div_mode	VDP HD 通道时钟分频比配置。 00: 不分频; 01: 2 分频; 10: 3 分频; 11: 4 分频。	0x0
[11]	RW	vdp_mipi_cken	VDP_MIPI TX 时钟门控配置寄存器。	0x0



Bits	Access	Name	Description	Reset
			0: 关闭时钟; 1: 打开时钟。	
[10]	RW	vdp_hdmi_cken	VDP_HDMI 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[9]	RW	vdp_bt_cken	VDP BT.1120/BT.656 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[8]	RW	vdp_bt_bp_cken	VDP bt_bp 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[7]	-	reserved	保留。	0x0
[6]	RW	vdp_hd_cken	VDP HD 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[5]	RW	vdp_hd_ppc_cken	VDP HD 工作时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[4]	-	reserved	保留。	0x0
[3]	RW	vdp_cfg_cken	VDP 配置时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[2]	RW	vdp_acken	VDP AXI 总线时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[1]	RW	vdp_pcken	VDP APB 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[0]	RW	vdp_srst_req	VDP 软复位请求。 0: 不复位; 1: 复位。	0x0



PERI_CRG67

PERI_CRG67 为 MIPI TX 时钟及软复位控制寄存器。

Offset Address: 0x010C Total Reset Value: 0x0000_0004

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RW	mipitxphy_clk_pctrl	MIPI TX PHY 接口时钟相位控制。 0: 时钟不取反; 1: 时钟取反。	0x0
[2]	RW	mipitx_ref_cksel	MIPI TX PHY REF 时钟选择。 0: 19.8MHz; 1: 27MHz。	0x1
[1]	RW	mipitx_srst_req	MIPI TX CTRL 软复位请求。 0: 不复位; 1: 复位。	0x0
[0]	RW	mipitx_cken	MIPI TX CTRL 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0

PERI_CRG68

PERI_CRG68 为 HDMI TX 控制器时钟及软复位控制寄存器。

Offset Address: 0x0110 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19]	RW	ssc_bypass_clk_sel	HDMI TX ssc bypass 时钟选择信号: 0: 选择 clk_ssc_bypass_occ 时钟; 1: 选择 vdp 送过来的时钟	0x0
[18:17]	-	reserved	保留。	0x0
[16]	RW	hdmitx_pxl_cken	HDMI TX PXL 时钟门控。 0: 时钟门控关闭; 1: 时钟门控打开。	0x0
[15:14]	-	reserved	保留。	0x0
[13:10]	RW	ssc_clk_div	HDMI TX SSC 时钟分频系数(1~16)。	0x0



Bits	Access	Name	Description	Reset
			n: (n+1)分频	
[9]	RW	hdmitx_ssc_srst_req	HDMI SSC 分频器软复位请求。 0: 不复位; 1: 复位。	0x0
[8]	RW	hdmitx_ctrl_cec_srst_req	HDMI TX CTRL CEC 软复位请求。 0: 不复位; 1: 复位。	0x0
[7]	RW	hdmitx_ctrl_srst_req	HDMI TX 软复位请求。 0: 不复位; 1: 复位。	0x0
[6]	RW	hdmitx_ctrl_bus_srst_req	HDMI TX 总线软复位请求。 0: 不复位; 1: 复位。	0x0
[5]	RW	hdmitx_ctrl_as_cken	HDMI TX AS 时钟门控。 0: 关断; 1: 打开。	0x0
[4]	RW	hdmitx_ctrl_os_cken	HDMI TX OS 60M 时钟门控。 0: 关断; 1: 打开。	0x0
[3]	RW	hdmitx_ctrl_cec_cken	HDMI TX CTRL CEC 时钟门控。 0: 关断; 1: 打开。	0x0
[2]	RW	hdmitx_ctrl_osc_24m_cken	HDMI TX OS 24M 时钟门控。 0: 关断; 1: 打开。	0x0
[1]	RW	ssc_bypass_cken	HDMI TX SSC BYPASS 时钟门控。 0: 关断; 1: 打开。	0x0
[0]	RW	ssc_in_cken	HDMI TX SSC 时钟门控。 0: 关断; 1: 打开。	0x0



PERI_CRG69

PERI_CRG69 为 HDMI TX PHY 时钟及软复位控制寄存器。

Offset Address: 0x0114 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:9]	-	reserved	保留。	0x000000
[8:6]	RW	tmds_clk_div	HDMI TX PHY TMDS 时钟分频系数 (1~8)。 n: (n+1)分频	0x0
[5]	RW	phy_tmds_srst_req	HDMI TX PHY TMDS 分频器软复位请求。 0: 不复位; 1: 复位。	0x0
[4]	RW	hdmitx_phy_srst_req	HDMI TX PHY 软复位请求。 0: 不复位; 1: 复位。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	phy_tmds_cken	HDMI TX PHY TMDS 时钟门控。 0: 关断; 1: 打开。	0x0

PERI_CRG80

PERI_CRG80 为 USB2.0 相关的时钟及软复位控制寄存器。

Offset Address: 0x0140 Total Reset Value: 0x0000_530F

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15]	RW	usb2_utmi_pctrl	USB2.0 UTMI 时钟相位控制。 0: 时钟不取反; 1: 时钟取反。	0x0
[14]	RW	usb2_phy_test_srst_req	USB2.0 PHY TEST 软复位请求。 0: 撤消复位; 1: 复位。	0x1
[13]	RW	usb2_utmi_cksel	USB2.0 UTMI 时钟源选择。 0: 选择 USB2.0 PHY 时钟;	0x0



Bits	Access	Name	Description	Reset
			1: 选择内部 60MHz。	
[12]	RW	usb2_utmi_cken	USB2.0 控制器 UTMI 时钟门控。 0: 关闭时钟; 1: 打开时钟。	0x1
[11:10]	-	reserved	保留。	0x0
[9]	RW	usb2_ref_cken	USB2.0 控制器 REF 时钟门控。 0: 关闭时钟; 1: 打开时钟。	0x1
[8]	RW	usb2_bus_cken	USB2.0 控制器总线时钟门控。 0: 关闭时钟; 1: 打开时钟。	0x1
[7:4]	-	reserved	保留。	0x0
[3]	RW	usb2_vcc_srst_req	USB2.0 控制器 VCC 软复位请求。 0: 不复位; 1: 复位。	0x1
[2]	RW	usb2_phy_cken	USB2.0 PHY 参考时钟门控 0: 关闭时钟; 1: 打开时钟。	0x1
[1]	RW	usb2_phy_port_treq	USB2.0 PHY TPOR 软复位请求。 0: 不复位; 1: 复位。	0x1
[0]	RW	usb2_phy_req	USB2.0 PHY POR 软复位请求。 0: 不复位; 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: 0x0144 Total Reset Value: 0x0000_0002

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x00000000
[4:2]	RW	fmc_cksel	FMC 时钟源选择。(SDR 模式下, 实际时钟为以下时钟的 2 分频; DDR 模式下, 为	0x0



Bits	Access	Name	Description	Reset
			以下时钟的 4 分频) 000: 24MHz; 001: 100MHz; 010: 150MHz; 011: 163MHz; 100: 200MHz; 101: 257MHz; 110: 300MHz(仅 DDR 模式可选); 111: 396MHz(仅 DDR 模式可选)。	
[1]	RW	fmc_cken	FMC 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[0]	RW	fmc_srst_req	FMC 的软复位请求。 0: 撤消复位; 1: 复位。	0x0

PERI_CRG82

PERI_CRG82 为 EMMC 相关的时钟及软复位控制寄存器。

Offset Address: 0x0148 Total Reset Value: 0x0000_000A

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3:2]	RW	emmc_cksel	EMMC 工作时钟选择。 01: 100MHz; 10: 50MHz; 11: 25MHz; 其他: 保留。	0x2
[1]	RW	emmc_cken	EMMC 时钟门控配置。 0: 关闭时钟; 1: 打开时钟。	0x1
[0]	RW	emmc_srst_req	EMMC 的软复位请求。 0: 撤消复位; 1: 复位。	0x0



PERI_CRG83

PERI_CRG83 为 EMMC DLL 相关的时钟及软复位控制寄存器。

Offset Address: 0x014C Total Reset Value: 0x0003_0001

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19]	RW	emmc_sap_dll_slave_en	EMMC_SAP_DLL 的从延迟线使能。 0: slave 停止工作; 1: slave 开始工作。	0x0
[18]	RW	emmc_sap_dll_stop	EMMC_SAP_DLL MASTER 计算时钟周期禁止信号。 0: 使能时钟检测; 1: 禁止时钟检测。	0x0
[17]	RW	emmc_sap_dll_bypass	EMMC_SAP_DLL SLAVE LINE bypass。 0: 正常模式; 1: slave line bypass。	0x1
[16]	RW	emmc_sap_dll_dllmode	EMMC_SAP_DLL 模式选择。 0: 正常模式; 1: 由 EMMC_SAP_DLL_dllssel 控制 SLAVE LINE。	0x1
[15:8]	RW	emmc_sap_dll_dllssel	EMMC_SAP_DLL SLAVE LINE delay 级数选择, EMMC_SAP_DLL_dllmode 为高时有效。	0x00
[7:4]	RW	emmc_sap_dll_tune	EMMC SAP DLL SLAVE tap 校准。 0000: 不校准; 0001: 增加 1 级 delay; 0010: 增加 2 级 delay; 0011: 增加 3 级 delay;; 0111: 增加 7 级 delay; 1000: 不校准; 1001: 减少 1 级 delay; 1010: 减少 2 级 delay; 1011: 减少 3 级 delay;; 1111: 减少 7 级 delay。	0x0



Bits	Access	Name	Description	Reset
[3:2]	-	reserved	保留。	0x0
[1]	RW	emmc_dll_srst_req	EMMC_SAP_DLL 软复位请求。 0: 撤消复位; 1: 复位。	0x0
[0]	RW	emmc_dll_cken	EMMC_SAP_DLL 时钟门控。 0: 关闭时钟; 1: 打开时钟。	0x1

PERI_CRG84

PERI_CRG84 为 EMMC DLL 状态指示寄存器。

Offset Address: 0x0150 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留。	0x000000
[10]	RO	emmc_sap_dll_overflow	EMMC_SAP_DLL MASTER overflow 状态。 0: EMMC_SAP_DLL MASTER 未溢出; 1: EMMC_SAP_DLL MASTER 溢出。	0x0
[9]	RO	emmc_sap_dll_ready	EMMC_SAP_DLL SLAVE ready 状态。 0: EMMC_SAP_DLL SLAVE 未就绪; 1: EMMC_SAP_DLL SLAVE 就绪。	0x0
[8]	RO	emmc_sap_dll_locked	EMMC_SAP_DLL MASTER lock 信号。 0: EMMC_SAP_DLL MASTER 未锁定; 1: EMMC_SAP_DLL MASTER 锁定。	0x0
[7:0]	RO	emmc_sap_dll_mdly_tap	EMMC_SAP_DLL MASTER LINE tap 值。	0x00

PERI_CRG85

PERI_CRG85 为 SDIO1 相关的时钟及软复位控制寄存器。

Offset Address: 0x0154 Total Reset Value: 0x0000_000A

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000



Bits	Access	Name	Description	Reset
[3:2]	RW	sdio1_cksel	SDIO1 工作时钟选择。 01: 100MHz; 10: 50MHz; 11: 25MHz; 其他: 保留。	0x2
[1]	RW	sdio1_cken	SDIO1 时钟门控配置。 0: 关闭时钟; 1: 打开时钟。	0x1
[0]	RW	sdio1_srst_req	SDIO1 的软复位请求。 0: 撤消复位; 1: 复位。	0x0

PERI_CRG86

PERI_CRG86 为 SDIO1 DLL 相关的时钟及软复位控制寄存器。

Offset Address: 0x0158 Total Reset Value: 0x0003_0001

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19]	RW	sdio1_sap_dll_slave_en	SDIO1_SAP_DLL 的从延迟线使能。 0: slave 停止工作; 1: slave 开始工作。	0x0
[18]	RW	sdio1_sap_dll_stop	SDIO1_SAP_DLL MASTER 计算时钟周期禁止信号。 0: 使能时钟检测; 1: 禁止时钟检测。	0x0
[17]	RW	sdio1_sap_dll_bypass	SDIO1_SAP_DLL SLAVE LINE bypass。 0: 正常模式; 1: slave line bypass。	0x1
[16]	RW	sdio1_sap_dll_dllmode	SDIO1_SAP_DLL 模式选择。 0: 正常模式; 1: 由 SDIO1_SAP_DLL_dllssel 控制 SLAVE LINE。	0x1
[15:8]	RW	sdio1_sap_dll_dllssel	SDIO1_SAP_DLL SLAVE LINE delay 级数选择, SDIO1_SAP_DLL_dllmode 为高时有	0x00



Bits	Access	Name	Description	Reset
			效。	
[7:4]	RW	sdio1_sap_dll_tune	SDIO1 SAP DLL SLAVE tap 校准。 0000: 不校准; 0001: 增加 1 级 delay; 0010: 增加 2 级 delay; 0011: 增加 3 级 delay;; 0111: 增加 7 级 delay; 1000: 不校准; 1001: 减少 1 级 delay; 1010: 减少 2 级 delay; 1011: 减少 3 级 delay;; 1111: 减少 7 级 delay。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	RW	sdio1_dll_srst_req	SDIO1_SAP_DLL 软复位请求。 0: 撤消复位; 1: 复位。	0x0
[0]	RW	sdio1_dll_cken	SDIO1_SAP_DLL 时钟门控。 0: 关闭时钟; 1: 打开时钟。	0x1

PERI_CRG87

PERI_CRG87 为 SDIO1 DLL 状态指示寄存器。

Offset Address: 0x015C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留。	0x000000
[10]	RO	sdio1_sap_dll_overflow	SDIO1_SAP_DLL MASTER overflow 状态。 0: SDIO1_SAP_DLL MASTER 未溢出; 1: SDIO1_SAP_DLL MASTER 溢出。	0x0



[9]	RO	sdio1_sap_dll_ready	SDIO1_SAP_DLL SLAVE ready 状态。 0: SDIO1_SAP_DLL SLAVE 未就绪; 1: SDIO1_SAP_DLL SLAVE 就绪。	0x0
[8]	RO	sdio1_sap_dll_locke d	SDIO1_SAP_DLL MASTER lock 信号。 0: SDIO1_SAP_DLL MASTER 未锁定; 1: SDIO1_SAP_DLL MASTER 锁定。	0x0
[7:0]	RO	sdio1_sap_dll_mdly_ tap	SDIO1_SAP_DLL MASTER LINE tap 值。	0x00

PERI_CRG88

PERI_CRG88 为 SDIO0 相关的时钟及软复位控制寄存器。

Offset Address: 0x0160 Total Reset Value: 0x0000_000E

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3:2]	RW	sdio0_cksel	SDIO0 工作时钟选择。 01: 100MHz; 10: 50MHz; 11: 25MHz; 其他: 保留。	0x3
[1]	RW	sdio0_cken	SDIO0 时钟门控配置。 0: 关闭时钟; 1: 打开时钟。	0x1
[0]	RW	sdio0_srst_req	SDIO0 的软复位请求。 0: 撤消复位; 1: 复位。	0x0

PERI_CRG89

PERI_CRG89 为 SDIO0 DLL 相关的时钟及软复位控制寄存器。

Offset Address: 0x0164 Total Reset Value: 0x0003_0001

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19]	RW	sdio0_sap_dll_slave_	SDIO0_SAP_DLL 的从延迟线使能。	0x0



Bits	Access	Name	Description	Reset
		en	0: slave 停止工作; 1: slave 开始工作。	
[18]	RW	sdio0_sap_dll_stop	SDIO0_SAP_DLL MASTER 计算时钟周期禁止信号。 0: 使能时钟检测; 1: 禁止时钟检测。	0x0
[17]	RW	sdio0_sap_dll_bypass	SDIO0_SAP_DLL SLAVE LINE bypass。 0: 正常模式; 1: slave line bypass。	0x1
[16]	RW	sdio0_sap_dll_dllmode	SDIO0_SAP_DLL 模式选择。 0: 正常模式; 1: 由 SDIO0_SAP_DLL_dllssel 控制 SLAVE LINE。	0x1
[15:8]	RW	sdio0_sap_dll_dllssel	SDIO0_SAP_DLL SLAVE LINE delay 级数选择, SDIO0_SAP_DLL_dllmode 为高时有效。	0x00
[7:4]	RW	sdio0_sap_dll_tune	SDIO0 SAP DLL SLAVE tap 校准。 0000: 不校准; 0001: 增加 1 级 delay; 0010: 增加 2 级 delay; 0011: 增加 3 级 delay;; 0111: 增加 7 级 delay; 1000: 不校准; 1001: 减少 1 级 delay; 1010: 减少 2 级 delay; 1011: 减少 3 级 delay;; 1111: 减少 7 级 delay。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	RW	sdio0_dll_srst_req	SDIO0_SAP_DLL 软复位请求。 0: 撤消复位; 1: 复位。	0x0
[0]	RW	sdio0_dll_cken	SDIO0_SAP_DLL 时钟门控。	0x1



Bits	Access	Name	Description	Reset
			0: 关闭时钟; 1: 打开时钟。	

PERI_CRG90

PERI_CRG90 为 SDIO0 DLL 状态指示寄存器。

Offset Address: 0x0168 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留。	0x000000
[10]	RO	sdio0_sap_dll_overflow	SDIO0_SAP_DLL MASTER overflow 状态。 0: SDIO0_SAP_DLL MASTER 未溢出; 1: SDIO0_SAP_DLL MASTER 溢出。	0x0
[9]	RO	sdio0_sap_dll_ready	SDIO0_SAP_DLL SLAVE ready 状态。 0: SDIO0_SAP_DLL SLAVE 未就绪; 1: SDIO0_SAP_DLL SLAVE 就绪。	0x0
[8]	RO	sdio0_sap_dll_locked	SDIO0_SAP_DLL MASTER lock 信号。 0: SDIO0_SAP_DLL MASTER 未锁定; 1: SDIO0_SAP_DLL MASTER 锁定。	0x0
[7:0]	RO	sdio0_sap_dll_mdly_tap	SDIO0_SAP_DLL MASTER LINE tap 值。	0x00

PERI_CRG91

PERI_CRG91 为 ETH 接口相关的时钟及软复位控制寄存器。

Offset Address: 0x016C Total Reset Value: 0x0000_0002

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RW	ethcore_cksel	ETH 时钟源选择 0: 100MHz; 1: 54MHz。	0x0
[6]	-	reserved	保留。	0x0
[5:4]	RW	ethphy_cksel	ETHPHY CLOCK OUT 时钟源选择	0x0



Bits	Access	Name	Description	Reset
			00: 25MHz; 01: 24MHz; 1X: 27MHz。	
[3]	RW	ext_fephy_srst_req	EPHY 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[2]	RW	eth_rmiick_sel	ETH RMII CLK 时钟源选择。 0: 选择内部 CRG 时钟; 1: 选择 PAD 输入时钟。	0x0
[1]	RW	eth_cken_pre	ETH 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[0]	RW	hrst_eth_s	ETH 的软复位请求。 0: 撤消复位; 1: 复位。	0x0

PERI_CRG100

PERI_CRG100 为 GZIP 时钟及软复位控制寄存器。

Offset Address: 0x0190 Total Reset Value: 0x0000_0007

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RW	gzip_srst_req	GZIP 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[2]	RW	gzip_pcken	GZIP APB 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[1]	RW	gzip_acken	GZIP AXI 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1



[0]	RW	gzip_cken	GZIP 时钟门控配置寄存器。 0: 关闭时钟; 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: 0x0194 Total Reset Value: 0x0000_0006

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2]	RW	dmac_cken	DMAC AXI 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[1]	RW	dmac_apb_cken	DMAC APB 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[0]	RW	dmac_srst_req	DMAC 的软复位请求。 0: 不复位; 1: 复位。	0x0

PERI_CRG102

PERI_CRG102 为 DDRT 相关的时钟及软复位控制寄存器。

Offset Address: 0x0198 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	ddrt_cken	DDRT 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[0]	RW	ddrt_srst_req	DDRT 的软复位请求。 0: 撤消复位; 1: 复位。	0x0



PERI_CRG103

PERI_CRG103 为 AIAO 时钟复位控制寄存器。

Offset Address: 0x019C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RW	aiao_pll_cken	AIAO PLL 时钟门控。 0: 关闭时钟; 1: 打开时钟。	0x0
[2]	RW	pop_srst_req	防爆破音 软复位请求。 0: 撤消复位; 1: 复位。	0x0
[1]	RW	aiao_cken	AIAO 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[0]	RW	aiao_srst_req	AIAO 软复位请求。 0: 撤消复位; 1: 复位。	0x0

PERI_CRG104

PERI_CRG104 为 SPACC/RSA/TRNG/KLAD 相关的时钟及软复位控制寄存器（安全模式下，非安全 CPU 不可写）。

Offset Address: 0x01A0 Total Reset Value: 0x0000_02AA

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x000000
[9]	RW	spacc_cken	SPACC 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[8]	RW	spacc_srst_req_ini	SPACC 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[7]	RW	rsa1_cken	RSA1 时钟门控配置寄存器。 0: 关闭时钟;	0x1



Bits	Access	Name	Description	Reset
			1: 打开时钟。	
[6]	RW	rsa1_srst_req	RSA1 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[5]	RW	rsa0_cken	RSA0 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[4]	RW	rsa0_srst_req	RSA0 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[3]	RW	trng_cken	TRNG 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[2]	RW	trng_srst_req	TRNG 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[1]	RW	klad_cken	KLAD 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[0]	RW	klad_srst_req	KLAD 的软复位请求。 0: 撤消复位; 1: 复位。	0x0

PERI_CRG105

PERI_CRG105 为 AIAO MCLK2 时钟配置寄存器。

Offset Address: 0x01A4 Total Reset Value: 0x001F_FFFF

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27]	RW	aiao_mclk2_cken	AIAO MCLK2 分频器时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0



[26:0]	RW	aiao_mclk2_div	AIAO MCLK2 分频时钟，可配置。 假定目标频率 X(MHZ)，则 $lcd\_mclk\_div = (X/1188) * 2^{27}$ ；最大 75MHz。	0x01FFFFFF

PERI_CRG106

PERI_CRG106 为'AIAO MCLK1 时钟配置寄存器。

Offset Address: 0x01A8 Total Reset Value: 0x001F_FFFF

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27]	RW	aiao_mclk1_cken	AIAO MCLK1 分频器时钟门控配置寄存器。 0: 关闭时钟； 1: 打开时钟。	0x0
[26:0]	RW	aiao_mclk1_div	AIAO MCLK1 分频时钟，可配置。 假定目标频率 X(MHZ)，则 $lcd\_mclk\_div = (X/1188) * 2^{27}$ ；最大 75MHz。	0x01FFFFFF

PERI_CRG107

PERI_CRG107 为'AIAO MCLK0 时钟配置寄存器。

Offset Address: 0x01AC Total Reset Value: 0x001F_FFFF

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27]	RW	aiao_mclk0_cken	AIAO MCLK0 分频器时钟门控配置寄存器。 0: 关闭时钟； 1: 打开时钟。	0x0
[26:0]	RW	aiao_mclk0_div	AIAO MCLK0 分频时钟，可配置。 假定目标频率 X(MHZ)，则 $lcd\_mclk\_div = (X/1188) * 2^{27}$ ；最大 75MHz。	0x01FFFFFF

PERI_CRG110

PERI_CRG110 为 I2C/UART 时钟及复位控制寄存器。



Offset Address: 0x01B8 Total Reset Value: 0x0007_F81F

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x00
[26]	RW	i2c7_srst_req	I2C7 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[25]	RW	i2c6_srst_req	I2C6 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[24]	RW	i2c5_srst_req	I2C5 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[23]	RW	i2c4_srst_req	I2C4 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[22]	RW	i2c3_srst_req	I2C3 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[21]	RW	i2c2_srst_req	I2C2 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[20]	RW	i2c1_srst_req	I2C1 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[19]	RW	i2c0_srst_req	I2C0 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[18]	RW	i2c7_cken	I2C7 时钟门控控制。 0: 关闭时钟; 1: 打开时钟。	0x1
[17]	RW	i2c6_cken	I2C6 时钟门控控制。 0: 关闭时钟; 1: 打开时钟。	0x1
[16]	RW	i2c5_cken	I2C5 时钟门控控制。	0x1



Bits	Access	Name	Description	Reset
			0: 关闭时钟; 1: 打开时钟。	
[15]	RW	i2c4_cken	I2C4 时钟门控控制。 0: 关闭时钟; 1: 打开时钟。	0x1
[14]	RW	i2c3_cken	I2C3 时钟门控控制。 0: 关闭时钟; 1: 打开时钟。	0x1
[13]	RW	i2c2_cken	I2C2 时钟门控控制。 0: 关闭时钟; 1: 打开时钟。	0x1
[12]	RW	i2c1_cken	I2C1 时钟门控控制。 0: 关闭时钟; 1: 打开时钟。	0x1
[11]	RW	i2c0_cken	I2C0 时钟门控控制。 0: 关闭时钟; 1: 打开时钟。	0x1
[10]	-	reserved	保留。	0x0
[9]	RW	uart4_srst_req	UART4 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[8]	RW	uart3_srst_req	UART3 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[7]	RW	uart2_srst_req	UART2 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[6]	RW	uart1_srst_req	UART1 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[5]	RW	uart0_srst_req	UART0 的软复位请求。 0: 撤消复位; 1: 复位。	0x0



Bits	Access	Name	Description	Reset
[4]	RW	uart4_cken	UART4 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[3]	RW	uart3_cken	UART3 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[2]	RW	uart2_cken	UART2 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[1]	RW	uart1_cken	UART1 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[0]	RW	uart0_cken	UART0 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1

PERI_CRG111

PERI_CRG111 为 ADC/UART/SPI/OTP/IR/PWM 相关的时钟及软复位控制寄存器。

Offset Address: 0x01BC Total Reset Value: 0x0180_7883

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25]	RW	wire3_spi_srst_req	3Wire SPI 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[24]	RW	wire3_spi_cken	3Wire SPI 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[23]	RW	adc_cken	ADC 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[22]	RW	uart4_cksel	UART4 时钟选择。 0: 24MHz;	0x0



Bits	Access	Name	Description	Reset
			1: 6MHz.	
[21]	RW	uart3_cksel	UART3 时钟选择。 0: 24MHz; 1: 6MHz.	0x0
[20]	RW	uart2_cksel	UART2 时钟选择。 0: 24MHz; 1: 6MHz.	0x0
[19]	RW	uart1_cksel	UART1 时钟选择。 0: 24MHz; 1: 6MHz.	0x0
[18]	RW	uart0_cksel	UART0 时钟选择。 0: 24MHz; 1: 6MHz.	0x0
[17]	RW	spi2_srst_req	SPI2 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[16]	RW	spi1_srst_req	SPI1 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[15]	RW	spi0_srst_req	SPI0 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[14]	RW	spi2_cken	SPI2 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[13]	RW	spi1_cken	SPI1 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[12]	RW	spi0_cken	SPI0 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[11]	RW	ir_cken	IR 时钟门控配置寄存器。 0: 关闭时钟;	0x1



Bits	Access	Name	Description	Reset
			1: 打开时钟。	
[10]	RW	ir_srst_req	IR 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[9:8]	RW	pwm_cksel	PWM 时钟选择。 00: 3MHz; 01: 50MHz; 1X: 24MHz。	0x0
[7]	RW	pwm_cken	PWM 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[6]	RW	pwm_srst_req	PWM 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[5]	-	reserved	保留。	0x0
[4]	RW	pmc_srst_req	PMC 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[3]	RW	rtc_srst_req	t_cap 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[2]	RW	lsadc_srst_req	LSADC 的软复位请求。 0: 撤消复位; 1: 复位。	0x0
[1]	RW	otp_cken	OTP 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x1
[0]	RW	test_clk_en	测试时钟使能寄存器。 0: 所有测试时钟关闭; 1: 所有测试时钟打开。	0x1



PERI_CRG120

PERI_CRG120 为 SOC 频率指示寄存器。

Offset Address: 0x01E0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:12]	RO	fmc_sc_seled	FMC 时钟切换完成指示信号。 000: 24MHz; 001: 100MHz; 010: 150MHz; 011: 163MHz; 100: 200MHz; 101: 257MHz; 110: 300MHz; 111: 396MHz。	0x0
[11]	-	reserved	保留。	0x0
[10]	RO	sysapb_sc_seled	SYSAPB 时钟切换完成指示信号。 0: 晶振时钟; 1: 50MHz。	0x0
[9]	-	reserved	保留。	0x0
[8]	RO	syscfg_sc_seled	SYSCFG 时钟切换完成指示信号。 0: 晶振时钟; 1: 148.5MHz。	0x0
[7:6]	RO	sysaxi_sc_seled	SYSAXI 时钟切换完成指示信号。 00: 晶振时钟; 01: 200MHz; 10: 300MHz; 11: 保留。	0x0
[5:3]	RO	ddr_sc_seled	DDR 时钟切换完成指示信号。 000: 晶振时钟; 001: 360MHz; 010: 396MHz; 011: 450MHz; 其他: 保留。	0x0
[2:0]	RO	a7_sc_seled	A7 时钟切换完成指示信号。	0x0



Bits	Access	Name	Description	Reset
			000: 晶振时钟; 001: 600MHz; 010: 792MHz; 011: 900MHz; 111: 保留。	

PERI_CRG_PLL121

PERI_CRG_PLL121 为复位状态寄存器。

Offset Address: 0x01E4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	保留。	0x0000000
[6]	RO	tde_softrst_state	TDE 复位状态。 0: 撤消成功; 1: 复位。	0x0
[5]	RO	vgs_softrst_state	VGS 复位状态。 0: 撤消成功; 1: 复位。	0x0
[4]	RO	viprocs_softrst_state	VIPROC 复位状态。 0: 撤消成功; 1: 复位。	0x0
[3]	RO	vpss_softrst_state	VPSS 复位状态。 0: 撤消成功; 1: 复位。	0x0
[2]	RO	gdc_softrst_state	GDC 复位状态。 0: 撤消成功; 1: 复位。	0x0
[1]	RO	gzip_rst_ok	GZIP 复位状态。 0: 撤消成功; 1: 复位。	0x0
[0]	RO	nnie_rst_ok	NNIE 复位状态。 0: 撤消成功; 1: 复位。	0x0



PERI_CRG_PLL122

PERI_CRG_PLL122 为 PLL LOCK 状态寄存器。

Offset Address: 0x01E8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RO	epll_lock	EPLL LOCK 状态。 0: Unlock; 1: Locked.	0x0
[2]	RO	vppll_lock	VPLL LOCK 状态。 0: Unlock; 1: Locked.	0x0
[1]	RO	dppll_lock	DPLL LOCK 状态。 0: Unlock; 1: Locked.	0x0
[0]	RO	apll_lock	APLL LOCK 状态。 0: Unlock; 1: Locked.	0x0

3.3 处理器子系统

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 900MHz。
- 集成 NEON（含 FPU 硬件浮点协处理器）。
- L1 Cache 包含 32KB Instruction Cache 和 32KB Data Cache。
- 内部集成 256KB 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。



3.4 中断系统

芯片支持 128 个中断源，对应的中断映射如表 3-2 所示。

表3-2 中断源分配表

中断位	中断源	中断位	中断源	中断位	中断源
0	SGI0	38	UART0	76	VIPROC
1	SGI1	39	UART1	77	JPGD
2	SGI2	40	UART2	78	A7_COMMTX[1]
3	SGI3	41	UART3	79	A7_COMMRX[1]
4	SGI4	42	UART4	80	A7_PMUIRQ[1]
5	SGI5	43	I2C0	81	保留
6	SGI6	44	I2C1	82	保留
7	SGI7	45	I2C2	83	SOFEWARE_INT
8	SGI8	46	I2C3	84	A7_COMMTX[0]
9	SGI9	47	I2C4	85	A7_COMMRX[0]
10	SGI10	48	GPIO0	86	A7_PMUIRQ[0]
11	SGI11	49	GPIO1	87	AIAO
12	SGI12	50	GPIO2	88	VICAP
13	SGI13	51	GPIO3	89	MIPI Rx
14	SGI14	52	GPIO4	90	VDP_INT0
15	SGI15	53	GPIO5	91	VDP_INT1
16	保留	54	GPIO6	92	HDMITX_AON
17	保留	55	GPIO7	93	HDMITX_PWD
18	保留	56	GPIO8	94	HDMITX_SEC
19	保留	57	GPIO9	95	MIPI Tx
20	保留	58	GPIO10	96	eMMC
21	保留	59	USB2.0	97	LSADC
22	保留	60	DMAC	98	DDR
23	保留	61	FMC	99	SCD
24	保留	62	SDIO0	100	SPI0
25	保留	63	SDIO1	101	SPI1



中断位	中断源	中断位	中断源	中断位	中断源
26	保留	64	ETH	102	SPI2
27	保留	65	DDR_SEC_INT	103	SPACC_SEC_INT
28	保留	66	GZIP	104	SPACC_NOR_INT
29	保留	67	TDE	105	保留
30	保留	68	JPGE	106	保留
31	保留	69	IVE	107	IR
32	WDG0	70	VGS	108	WDG1
33	Timer0/Timer1	71	DDRT	109	I2C5
34	Timer2/Timer3	72	VEDU	110	I2C6
35	Timer4/Timer5	73	NNIE	111	I2C7
36	Timer6/Timer7	74	GDC	112	GPIO11
37	RTC	75	VPSS	113~127	保留

3.5 系统控制器

3.5.1 功能描述

系统控制器通过寄存器实现对芯片的一些系统控制，包含系统软复位、地址重映射、模块工作模式、电源管理等控制寄存器，以及芯片 ID 等标识寄存器。

系统控制器支持对芯片全局软复位：当配置全局软复位寄存器 [SC_SYSRES](#) 后，芯片将被复位。

为防止软件对系统控制器的误操作对整个系统产生严重影响，系统控制器提供了一些关键配置寄存器的写保护功能。包括系统全局软复位控制寄存器：[SC_SYSRES](#)，对这关键寄存器进行写操作之前，必须配置寄存器 [SC_LOCKEN](#) 打开写权限。操作完成之后配置寄存器 [SC_LOCKEN](#) 关闭写权限，让这些关键寄存器不会被软件随意改写。

系统控制器提供了芯片标识寄存器 [CHIP_ID](#)，软件可以通过 [CHIP_ID](#) 来识别芯片类型。

寄存器分“系统控制寄存器”和“MISC 寄存器”两部分，详细信息请参考相关寄存器描述部分。

3.5.2 系统控制器寄存器概览

系统控制器寄存器概览如[表 3-3](#) 所示。



表3-3 系统控制器寄存器概览（基址是 0x1202_0000）

偏移地址	名称	描述	页码
0x0000	SC_CTRL	系统控制寄存器	3-61
0x0004	SC_SYSRES	系统软复位寄存器	3-63
0x001C	SOFTINT	软中断寄存器	3-63
0x0020	SOFTTYPE	软中断向量寄存器	3-63
0x0030	PERISTAT	外设状态寄存器	3-64
0x0044	SC_LOCKEN	关键系统控制寄存器的锁定寄存器	3-65
0x008C	SYSSTAT	系统状态寄存器	3-65
0x0400	DIE_ID0	DIE ID0 寄存器	3-66
0x0404	DIE_ID1	DIE ID1 寄存器	3-66
0x0408	DIE_ID2	DIE ID2 寄存器	3-67
0x040C	DIE_ID3	DIE ID3 寄存器	3-67
0x0410	DIE_ID4	DIE ID4 寄存器	3-67
0x0414	DIE_ID5	DIE ID5 寄存器	3-67
0x0EE0	CHIP_ID	芯片 ID 寄存器	3-67
0x0EEC	VENDOR_ID	VENDORID 寄存器	3-68

3.5.3 系统控制器寄存器描述

SC_CTRL

SC_CTRL 为系统控制寄存器。用于指定需要系统完成的操作。



注意

该寄存器可被寄存器 SC_LOCKEN 写保护，只有不使用写保护模式时，对这个寄存器的写操作才有效。

Offset Address: 0x0000 Total Reset Value: 0x0000_0200

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00



Bits	Access	Name	Description	Reset
[23]	RW	timeren7ov	Timer7 计数时钟选择。 0:使用 3MHz 时钟进行计数; 1:由总线时钟进行计数。	0x0
[22]	RW	timeren6ov	Timer6 计数时钟选择。 0:使用 3MHz 时钟进行计数; 1:由总线时钟进行计数。	0x0
[21]	RW	timeren5ov	Timer5 计数时钟选择。 0:使用 3MHz 时钟进行计数; 1:由总线时钟进行计数。	0x0
[20]	RW	timeren4ov	Timer4 计数时钟选择。 0:使用 3MHz 时钟进行计数; 1:由总线时钟进行计数。	0x0
[19]	RW	timeren3ov	Timer3 计数时钟选择。 0:使用 3MHz 时钟进行计数; 1:由总线时钟进行计数。	0x0
[18]	RW	timeren2ov	Timer2 计数时钟选择。 0:使用 3MHz 时钟进行计数; 1:由总线时钟进行计数。	0x0
[17]	RW	timeren1ov	Timer1 计数时钟选择。 0:使用 3MHz 时钟进行计数; 1:由总线时钟进行计数。	0x0
[16]	RW	timeren0ov	Timer0 计数时钟选择。 0:使用 3MHz 时钟进行计数; 1:由总线时钟进行计数。	0x0
[15:10]	-	reserved	保留。	0x00
[9]	RO	remapstat	地址重映射的状态。 0:非 Remap 状态; 1:Remap 状态。	0x1
[8]	RW	remapclear	地址重映射清除。 0:保持 Remap 状态。 1:清除 Remap 状态。 Clear Remap 前后地址映射关系请参见地址分配(1.5 小节)。	0x0



Bits	Access	Name	Description	Reset
[7:0]	-	reserved	保留。	0x00

SC_SYSRES

SC_SYSRES 为系统软复位寄存器。向该寄存器写入任何值都会使系统控制器向复位模块发出系统软复位请求，复位模块进行系统软复位。



注意

该寄存器可被寄存器 SC_LOCKEN 写保护，只有禁用写保护时，对这个寄存器的写操作才有效。

Offset Address: 0x0004 Total Reset Value: 0x0000_0002

Bits	Access	Name	Description	Reset
[31:0]	WO	softresreq	对该寄存器的任意写操作都会导致系统软复位。	0x00000002

SOFTINT

SOFTINT 为软中断寄存器。

Offset Address: 0x001C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留	0x00000000
[0]	RW	software_int	向此位写入 1 产生软件中断。	0x0

SOFTTYPE

SOFTTYPE 为软中断向量寄存器。

Offset Address: 0x0020 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	-	software_int_vector	软件中断向量。	0x00000000



PERISTAT

PERISTAT 为外设状态寄存器。

Offset Address: 0x0030 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:19]	-	reserved	保留。	0x0000
[18:17]	RO	usb2_update_delay_time	USB 升级操作延时选择。 00: 0ms; 01: 200ms; 10: 300ms; 11: 400ms。	0x0
[16:15]	RO	emmc_poweron_delay_time	eMMC 上电延时选择。 00: 0ms; 01: 20ms。	0x0
[14]	-	reserved	保留。	0x0
[13]	RO	emmc_pad_pu_disable	eMMC CMD/DATA IO 内部上拉电阻禁用标志。 0:内部上拉电阻使能; 1:内部上拉电阻禁用。	0x0
[12]	RO	sdio_pad_pu_disable	SDIO0 CMD/DATA IO 内部上拉电阻禁用标志。 0:内部上拉电阻使能; 1:内部上拉电阻禁用。	0x0
[11]	RO	boot_backup_enable	BOOT 备份功能使能。 0:关闭; 1:使能。	0x0
[10:8]	RO	emmc_boot_cksel	eMMC 启动频率选择。 000: 25M; 001: 400K; 010: 50M; 其他: 保留。	0x0
[7:6]	RO	usb_phy_para_sel	USB PHY 参数配置标志; 保留。	0x0
[5]	RO	update_from_usb_disable	USB 升级禁用。 0:使能 USB 升级; 1:禁用 USB 升级。	0x0



Bits	Access	Name	Description	Reset
[4]	RO	update_from_sdio_di sable	SDIO 升级禁用。 0:使能 SDIO 升级; 1:禁用 SDIO 升级。	0x0
[3]	RO	jtag_ca_en	JTAG 接入安全控制使能标志。 0:关闭; 1:使能。	0x0
[2]	RO	reserved	保留。	0x0
[1]	RO	ddrca_en	DDR 加扰使能标志。 0:DDR 不加扰; 1:DDR 加扰。	0x0
[0]	RO	sec_boot_mode	安全启动模式标志。 0:非安全启动模式; 1:安全启动模式。	0x0

SC_LOCKEN

SC_LOCKEN 为关键系统控制寄存器的锁定寄存器。

Offset Address: 0x0044 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	scper_lockl	关键系统控制寄存器的锁定寄存器。涉及寄存器包括 SC_CTRL 、 SC_SYSRES 。 向该寄存器写入 0x1ACC_E551，可打开所有寄存器的写权限，写入其他值则关闭写权限。 读该寄存器返回加锁的状态而不是写入该寄存器的值。 0x0000_0000：允许上述写访问(未加锁)。 0x0000_0001：禁止上述写访问(已加锁)。	0x00000000

SYSSTAT

SYSSTAT 为系统状态寄存器。

Offset Address: 0x008C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RO	sfc_addr_mode	当 sfc_device_mode=0 时表示 SPI NOR Flash 启动地址模式。 0: 3 字节模式。 1: 4 字节模式。 当 sfc_device_mode=1 时表示 SPI NAND Flash 启动模式: 0: 1 线启动模式; 1: 4 线启动模式。	0x0
[6]	RO	sfc_device_mode	SPI FLASH 器件类型。 0: SPI NOR Flash; 1: SPI NAND Flash。	0x0
[5:4]	RO	boot_sel	启动模式。 00: 从 SPI Nor/Nand Flash 启动; 01: 从 eMMC 启动; 10: FAST BOOT, 串口烧写 SPI Nor/Nand Flash; 11: FAST BOOT, 串口烧写 eMMC。 其他: 保留。	0x0
[3:0]	RO	reserved	保留。	0x0

DIE_ID0

DIE_ID0 为 DIE ID0 寄存器。

Offset Address: 0x0400 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	die_id0	DIE_ID[31:0]	0x00000000

DIE_ID1

DIE_ID1 为 DIE ID1 寄存器。

Offset Address: 0x0404 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RO	die_id1	DIE_ID[63:32]	0x00000000

DIE_ID2

DIE_ID2 为 DIE ID2 寄存器。

Offset Address: 0x0408 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	die_id2	DIE_ID[95:64]	0x00000000

DIE_ID3

DIE_ID3 为 DIE ID3 寄存器。

Offset Address: 0x040C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	die_id3	DIE_ID[127:96]	0x00000000

DIE_ID4

DIE_ID4 为 DIE ID4 寄存器。

Offset Address: 0x0410 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	die_id4	DIE_ID[159:128]	0x00000000

DIE_ID5

DIE_ID5 为 DIE ID5 寄存器。

Offset Address: 0x0414 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	die_id5	DIE_ID[191:160]	0x00000000



CHIP_ID

CHIP_ID 为芯片 ID 寄存器。

Offset Address: 0x0EE0 Total Reset Value: 0x3516D300

Bits	Access	Name	Description	Reset
[31:0]	RO	chip_id	CHIP_ID[31:0]	0x3516D300

VENDOR_ID

VENDOR_ID 为 VENDORID 寄存器。

Offset Address: 0x0EEC Total Reset Value: 0x0000_0035

Bits	Access	Name	Description	Reset
[31:0]	RO	vendor_id	VENDOR ID，固定值为 0x35。	0x00000035

3.5.4 MISC 寄存器概览

MISC 寄存器概览如表 3-4 所示。

表3-4 MISC 寄存器概览（基址是 0x1203_0000）

偏移地址	名称	描述	页码
0x0000	MISC_CTRL0	MISC 功能选择寄存器 0	3-70
0x0004	MISC_CTRL1	MISC 功能选择寄存器 1	3-72
0x000C	MISC_CTRL3	CPU 控制寄存器 1	3-72
0x0010	MISC_CTRL4	RAM Retention 控制寄存器 0	3-73
0x0018	MISC_CTRL6	VICAP/VDP 模式控制寄存器 1	3-74
0x0020	MISC_CTRL8	USB2.0 控制寄存器 0。	3-75
0x0028	MISC_CTRL10	DDR_CA 控制寄存器 0	3-76
0x002C	MISC_CTRL11	DDR_CA 控制寄存器 1	3-76
0x0030	MISC_CTRL12	DDR_CA 控制寄存器 2	3-76
0x0038	MISC_CTRL14	总线优先级控制寄存器 0	3-77
0x003C	MISC_CTRL15	总线优先级控制寄存器 1	3-77
0x0040	MISC_CTRL16	总线优先级控制寄存器 2	3-78



偏移地址	名称	描述	页码
0x0044	MISC_CTRL17	总线优先级控制寄存器 4	3-79
0x0048	MISC_CTRL18	总线优先级控制寄存器 5	3-79
0x004C	MISC_CTRL19	总线优先级控制寄存器 6	3-80
0x0050	MISC_CTRL20	总线优先级控制寄存器 7	3-80
0x0054	MISC_CTRL21	总线优先级控制寄存器 8	3-81
0x0058	MISC_CTRL22	总线优先级控制寄存器 9	3-81
0x005C	MISC_CTRL23	总线优先级控制寄存器 10	3-82
0x0060	MISC_CTRL24	总线优先级控制寄存器 11	3-82
0x0064	MISC_CTRL25	总线优先级控制寄存器 12	3-83
0x0068	MISC_CTRL26	总线优先级控制寄存器 13	3-83
0x006C	MISC_CTRL27	总线优先级控制寄存器 14	3-84
0x0070	MISC_CTRL28	总线优先级控制寄存器 15	3-84
0x0074	MISC_CTRL29	总线优先级控制寄存器 16	3-85
0x0078	MISC_CTRL30	总线优先级控制寄存器 17	3-85
0x007C	MISC_CTRL31	总线优先级控制寄存器 18	3-86
0x0080	MISC_CTRL32	MDDRC W QOS 寄存器 0	3-86
0x0084	MISC_CTRL33	MDDRC W QOS 寄存器 1	3-87
0x0088	MISC_CTRL34	MDDRC W QOS 寄存器 2	3-88
0x008C	MISC_CTRL35	MDDRC W QOS 寄存器 3	3-89
0x0090	MISC_CTRL36	MDDRC R QOS 寄存器 0	3-89
0x0094	MISC_CTRL37	MDDRC R QOS 寄存器 1	3-90
0x0098	MISC_CTRL38	MDDRC R QOS 寄存器 2	3-90
0x009C	MISC_CTRL39	MDDRC R QOS 寄存器 3	3-91
0x00B0	MISC_CTRL44	SVB 控制寄存器	3-91
0x00B4	MISC_CTRL45	Tsensor 控制寄存器	3-92
0x00B8	MISC_CTRL46	Tsensor 状态寄存器	3-93
0x00BC	MISC_CTRL47	Tsensor 温度记录值寄存器 0	3-93
0x00C0	MISC_CTRL48	Tsensor 温度记录值寄存器 1	3-94
0x00C4	MISC_CTRL49	Tsensor 温度记录值寄存器 2	3-94



偏移地址	名称	描述	页码
0x00C8	MISC_CTRL50	Tsensor 温度记录值寄存器 3	3-94
0x00CC	MISC_CTRL51	升级模式清除寄存器	3-95
0x00D0	MISC_CTRL52	HPM 控制寄存器 0	3-95
0x00D4	MISC_CTRL53	HPM 控制寄存器 1	3-95
0x00D8	MISC_CTRL54	HPM 状态寄存器 0	3-96
0x00DC	MISC_CTRL55	HPM 状态寄存器 1	3-96
0x00E0	MISC_CTRL56	SHPM 控制寄存器	3-97
0x0100	MISC_CTRL57	DDRC LOCK 寄存器	3-97
0x0104	MISC_CTRL58	安全访问控制寄存器 0	3-98
0x0108	MISC_CTRL59	安全访问控制寄存器 1	3-101
0x010C	MISC_CTRL60	安全访问控制寄存器 2	3-103
0x0110	MISC_CTRL61	安全访问控制寄存器 3	3-106
0x0114	MISC_CTRL62	安全访问控制寄存器 4	3-108
0x0118	MISC_CTRL63	安全访问控制寄存器 5	3-110

3.5.5 MISC 寄存器描述

MISC_CTRL0

MISC_CTRL0 为 MISC 功能选择寄存器 0。

Offset Address: 0x0000 Total Reset Value: 0x0000_2000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19:16]	RW	test_clk_sel	测试时钟 TEST_CLK 源选择: 0x0: pll_test_out[0]; 0x1: pll_test_out[1]; 0x2: pll_test_out[2]; 0x3: pll_test_out[3]; 0x4: clk_24m; 0x5: audiocodec_mclk; 0x6: usb_phy_clk;	0x0



Bits	Access	Name	Description	Reset
			0x7: mipitx_phy_lanebyte_clk; 0x8: hdmitx_phy_tmids_clk; 0x9: rtc_clk_out; 其他: 保留。	
[15]	-	reserved	保留。	0x0
[14]	RW	jtag_on_sdio	JTAG 信号复用以 SDIO0 接口。 0:JTAG 信号复用在 JTAG_*管脚上; 1:JTAG 信号复用在 SDIO0_*管脚上。	0x0
[13]	RW	vgs_bootram_mode	VGS 模块 RAM 复用为 BOOTRAM 使能。 当需要 BOOTRAM 时需使能, 当使用 VGS 功能时, 需将此位置为 0。 0:不使能, 1:使能。	0x1
[12]	RW	emmc_fix_bypass	eMMC 控制器卡不识别 BUG 修复逻辑使能控制。 0:使能修改逻辑; 1:不使能修改逻辑。	0x0
[11]	RW	sdio1_fix_bypass	SDIO1 控制器卡不识别 BUG 修复逻辑使能控制。 0:使能修改逻辑; 1:不使能修改逻辑。	0x0
[10]	RW	sdio1_pd_mux_bypass	SDIO1 控制器卡卡下电 BUG 修复逻辑使能控制。 0:使能修改逻辑; 1:不使能修改逻辑。	0x0
[9:7]	-	reserved	保留。	0x0
[6]	RW	sdio0_carddetect_pol	SDIO0 卡检测信号极性控制。 0:卡检测信号低有效。 1:卡检测信号高有效。	0x0
[5]	RW	spi2_cs_pctrl	SPI2 片选信号极性控制。 0:SPI2_CSN 低有效; 1:SPI2_CSN 高有效。	0x0
[4]	RW	spi1_cs1_pctrl	SPI1 片选信号 1 极性控制。 0:SPI1_CSN1 低有效; 1:SPI1_CSN1 高有效。	0x0



Bits	Access	Name	Description	Reset
[3]	RW	spi1_cs0_pctrl	SPI1 片选信号 0 极性控制。 0:SPI1_CSN0 低有效; 1:SPI1_CSN0 高有效。	0x0
[2]	RW	ssp1_cs_sel	SPI1 信号输出片选选择。 0:SPI1_CSN0; 1:SPI1_CSN1。	0x0
[1]	RW	spi_3wire_cs_pctrl	3WIRE SPI 片选信号极性控制。 0:SPI_3LINE_CSN 低有效; 1:SPI_3LINE_CSN 高有效。	0x0
[0]	RW	spi0_cs_pctrl	SPI0 片选信号极性控制。 0:SPI0_CSN 低有效; 1:SPI0_CSN 高有效。	0x0

MISC_CTRL1

MISC_CTRL1 为 MISC 功能选择寄存器 1。

Offset Address: 0x0004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2]	RW	sdio0_pswitch_ctrl_sel	SDIO0 PowerSwitch 控制选择。 0:由 SDIO0 控制器控制; 1:由 MISC 寄存器控制。	0x0
[1]	RW	sdio0_pswitch_sel	SDIO0 PowerSwitch 输出电压选择。 0:3.3V; 1:1.8V。	0x0
[0]	RW	sdio0_pswitch_en	SDIO0 PowerSwitch 使能。 0:SDIO0 PowerSwitch 不使能; 1:SDIO0 PowerSwitch 使能。 仅当 sdio0_pswitch_ctrl_sel 为 1 时, 此控制才有效。	0x0



MISC_CTRL3

MISC_CTRL3 为 CPU 控制寄存器 1。

Offset Address: 0x000C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	保留。	0x0000000
[6:5]	RO	a7_smpnamp	指示 A7 CORE0/CORE1 是处于 SMP(Symmetric MultiProcessing)模式还是 AMP(Asymmetric MultiProcessing)模式。 其中 a7_smpnamp[0]指示 CORE0 的状态, a7_smpnamp[1]指示 CORE1 的状态。 高表示处于 SMP 模式, 低表示处于 AMP 模式。	0x0
[4]	RO	a7_standbywfi_l2	A7 L2 WFI 低功耗状态指示。 0:非 WFI 低功耗状态; 1:处于 WFI 低功耗状态。	0x0
[3:2]	RO	a7_standbywfi	A7 CORE0/CORE1 WFI 低功耗状态指示。 其中 a7_standbywfi[0]指示 CORE0 的 WFI 状态, a7_standbywfi[1]指示 CORE1 的 WFI 状态; 高表示处于 CORE 处于 WFI 低功耗状态。	0x0
[1:0]	RO	a7_standbywfe	A7 CORE0/CORE1 WFE 低功耗状态指示。 其中 a7_standbywfe[0]指示 CORE0 的 WFE 状态, a7_standbywfe[1]指示 CORE1 的 WFE 状态; 高表示处于 CORE 处于 WFE 低功耗状态。	0x0

MISC_CTRL4

MISC_CTRL4 为 RAM Retention 控制寄存器 0。

Offset Address: 0x0010 Total Reset Value: 0x0000_000D

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RW	gzip_ram_clk_gate_en_bypass	GZIP 模块内 RAM 自动时钟门控控制。 0:自动时钟门控使能; 1:自动时钟门控关闭, GZIP 内 RAM 时钟常开。	0x1



Bits	Access	Name	Description	Reset
[2]	RW	vedu_ret1n_bypass	VEDU 模块自动 RAM Retention 功能控制。 0:VEDU RAM 自动 Retention 功能使能。 1:VEDU RAM 自动 Retention 功能关闭;	0x1
[1]	RW	bootrom_pgen	BOOTROM 供电控制。 0:BOOTROM 上电; 1:BOOTROM 下电, 此时 BOOTROM 不能工作。	0x0
[0]	RW	spacc_ret1n	SPACC 中 RAM Retention 控制。 0:SPACC RAM 处于 Retention 状态; 1:SPACC RAM 处于 Active 状态。 当 IR 控制器不使能时, 可以将此位设置为 0 以节省功耗。	0x1

MISC_CTRL6

MISC_CTRL6 为 VICAP/VDP 模式控制寄存器 1。

Offset Address: 0x0018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:12]	-	reserved	保留。	0x00000
[11]	RW	vo_pdata_reverse	VDP 并行输出 Bit 颠倒使能。 0: 不使能; 1: 使能。	0x0
[10:8]	RW	vo_pdata_mode	VDP 并行口输出模式选择。 000: LCD 并行 24bit 模式; 001: LCD 并行 18bit 模式; 010: LCD 并行 16bit 模式; 011: LCD 串行 8bit 模式; 100: LCD 串行 6bit 模式; 101: BT.1120 模式; 110: BT.656 模式; 其他: 保留。	0x0
[7]	RW	viproc_vpss_online_mode	VIPROC 到 VPSS 数据通路工作模式选择。	0x0



Bits	Access	Name	Description	Reset
			0: 离线模式; 1: 在线模式。	
[6]	RW	vicap_viproc_online_mode	VICAP 到 VIPROC 数据通路工作模式选择。 0: 离线模式; 1: 在线模式。	0x0
[5]	RW	vi_port1_input_sel	VICAP Port1 视频源选择。 0: MIPI 通道 1; 1: CMOS 输入;	0x0
[4]	-	reserved	保留。	0x0
[3:2]	RW	mipirx1_work_mode	MIPI RX 通道 1 模式选择。 0: MIPI 模式; 1: LVDS 模式。	0x0
[1:0]	RW	mipirx0_work_mode	MIPI RX 通道 0 模式选择。 0: MIPI 模式; 1: LVDS 模式。	0x0

MISC_CTRL8

USB2.0 控制寄存器 0。

Offset Address: 0x0020 Total Reset Value: 0x0004_00DF

Bits	Access	Name	Description	Reset
[31:19]	-	reserved	保留。	0x0000
[18:13]	RW	usb2_fladj_30mhz_reg	USB 端口 HS 时钟抖动控制。	0x20
[12:9]	RW	usb2_dbg_addr	USB Debug 地址。	0x0
[8]	RW	usb2_hub_port_perm_attach	USB 外接设备模式。 0: USB 外接的设备为可移除式设备; 1: USB 外接的设备为非可移除式设备。	0x0
[7]	RW	usb2_port_ovrcur_pos1	USB 过流保护极性控制。 0: 低电平有效; 1: 高电平有效。	0x1
[6]	RW	usb2_port_pwren_pos0	USB 电源使能极性控制。	0x1



Bits	Access	Name	Description	Reset
		1	0: 低电平有效; 1: 高电平有效。	
[5]	RW	usb2_port_ovrcur_ctl_en	USB 过流保护使能。 0: 不使能; 1: 使能。	0x0
[4]	RW	usb2_port_pwr_ctrl_en	USB 供电使能。 0: USB2.0 端口供电控制不使能; 1: USB2.0 端口供电使能由 USB2.0 控制器输出。	0x1
[3:0]	RW	usb2_bus_filter_bypass	USB 2.0 总线筛选旁路。必须设置为 4'b1111。	0xF

MISC_CTRL10

MISC_CTRL10 为 DDR_CA 控制寄存器 0。

Offset Address: 0x0028 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	ddrca_random_l	DDR CA 功能随机值 Bit31~0。	0x00000000

MISC_CTRL11

MISC_CTRL11 为 DDR_CA 控制寄存器 1。

Offset Address: 0x002C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	ddrca_random_h	DDR CA 功能随机值 Bit63~32。	0x00000000

MISC_CTRL12

MISC_CTRL12 为 DDR_CA 控制寄存器 2。

Offset Address: 0x0030 Total Reset Value: 0x0000_0005

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留	0x00000000



Bits	Access	Name	Description	Reset
[4]	RW	ddrca_update	DDR 加扰功能加载使能。 当此寄存器由 0 变为 1 时将 ddrca_en 和 ddrca_random 参数加载至 DDR 控制器。	0x0
[3:0]	RW	ddrca_en	DDR 加扰功能模式。 0x0: 打开 MDDRCA 功能; 0x1: 输出全 0 数据; 0x3: 输出全 1 数据; 0x5: 旁路掉 MDDRCA 功能; 0xD: 输出全 1 数据。 其他: 保留。	0x5

MISC_CTRL14

MISC_CTRL14 为总线优先级控制寄存器 0。

Offset Address: 0x0038 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	syscfg_timeout_en_m2	SYSCFG 总线 M2 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[30:16]	RW	syscfg_timeout_value_m2	SYSCFG 总线 M2 端口的 timeout 计数值。 计数值 =syscfg_timeout_value_m2×2。	0x0000
[15]	RW	syscfg_timeout_en_m1	SYSCFG 总线 M1 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[14:0]	RW	syscfg_timeout_value_m1	SYSCFG 总线 M1 端口的 timeout 计数值。 计数值 =syscfg_timeout_value_m1×2。	0x0000

MISC_CTRL15

MISC_CTRL15 为总线优先级控制寄存器 1。

Offset Address: 0x003C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30:28]	RW	syscfg_priority_s6	SYSCFG 总线 S6 端口总线访问优先级。 7 为最高优先级。	0x0
[27]	-	reserved	保留。	0x0
[26:24]	RW	syscfg_priority_s5	SYSCFG 总线 S5 端口总线访问优先级。 7 为最高优先级。	0x0
[23]	-	reserved	保留。	0x0
[22:20]	RW	syscfg_priority_s4	SYSCFG 总线 S4 端口总线访问优先级。 7 为最高优先级。	0x0
[19]	-	reserved	保留。	0x0
[18:16]	RW	syscfg_priority_s3	SYSCFG 总线 S3 端口总线访问优先级。 7 为最高优先级。	0x0
[15]	-	reserved	保留。	0x0
[14:12]	RW	syscfg_priority_s2	SYSCFG 总线 S2 端口总线访问优先级。 7 为最高优先级。	0x0
[11]	-	reserved	保留。	0x0
[10:8]	RW	syscfg_priority_s1	SYSCFG 总线 S1 端口总线访问优先级。 7 为最高优先级。	0x0
[7]	-	reserved	保留。	0x0
[6:4]	RW	syscfg_port1_pri_in	SYSCFG 总线 M2 端口优先级。 7 为最高优先级。	0x0
[3]	-	reserved	保留。	0x0
[2:0]	RW	syscfg_port0_pri_in	SYSCFG 总线 M1 端口优先级。 7 为最高优先级。	0x0

MISC_CTRL16

MISC_CTRL16 为总线优先级控制寄存器 2。

Offset Address: 0x0040 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	sysdata_timeout_en_	SYSDATA 总线 M2 端口的 timeout 计数使	0x0



		m2	能。 0:禁止; 1:使能。	
[30:16]	RW	sysdata_timeout_value_m2	SYSDATA 总线 M2 端口的 timeout 计数值。 计数值 = sysdata_timeout_value_m2 × 2。	0x0000
[15]	RW	sysdata_timeout_en_m1	SYSDATA 总线 M1 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[14:0]	RW	sysdata_timeout_value_m1	sysdata 总线 M1 端口的 timeout 计数值。 计数值 = sysdata_timeout_value_m1 × 2。	0x0000

MISC_CTRL17

MISC_CTRL17 为总线优先级控制寄存器 4。

Offset Address: 0x0044 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:12]	RW	sysdata_priority_s2	SYSDATA 总线 S2 端口总线访问优先级。 1 为最高优先级。	0x0
[11]	-	reserved	保留。	0x0
[10:8]	RW	sysdata_priority_s1	SYSDATA 总线 S1 端口总线访问优先级。 1 为最高优先级。	0x0
[7]	-	reserved	保留。	0x0
[6:4]	RW	sysdata_port1_pri_in	SYSDATA 总线 M2 端口优先级。 7 为最高优先级。	0x0
[3]	-	reserved	保留。	0x0
[2:0]	RW	sysdata_port0_pri_in	SYSDATA 总线 M1 端口优先级。 7 为最高优先级。	0x0

MISC_CTRL18

MISC_CTRL18 为总线优先级控制寄存器 5。



Offset Address: 0x0048 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	ahb1_timeout_en_m2	AHB1 总线 M2 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[30:16]	RW	ahb1_timeout_value_m2	AHB1 总线 M2 端口的 timeout 计数值。 计数值 = ahb1_timeout_value_m2 × 2。	0x0000
[15]	RW	ahb1_timeout_en_m1	AHB1 总线 M1 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[14:0]	RW	ahb1_timeout_value_m1	AHB1 总线 M1 端口的 timeout 计数值。 计数值 = ahb1_timeout_value_m1 × 2。	0x0000

MISC_CTRL19

MISC_CTRL19 为总线优先级控制寄存器 6。

Offset Address: 0x004C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	AHB1 总线 M2 端口的 timeout 计数使能。 0: 禁止; 1: 使能。	0x0000
[15]	RW	ahb1_timeout_en_m3	AHB1 总线 M3 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[14:0]	RW	ahb1_timeout_value_m3	AHB1 总线 M3 端口的 timeout 计数值。 计数值 = ahb1_timeout_value_m3 × 2。	0x0000

MISC_CTRL20

MISC_CTRL20 为总线优先级控制寄存器 7。

Offset Address: 0x0050 Total Reset Value: 0x0000_0234

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留。	0x000000



[10:8]	RW	ahb1_port2_pri_in	AHB1 总线 M3 端口优先级。 7 为最高优先级。	0x2
[7]	-	reserved	保留。	0x0
[6:4]	RW	ahb1_port1_pri_in	AHB1 总线 M2 端口优先级。 7 为最高优先级。	0x3
[3]	-	reserved	保留。	0x0
[2:0]	RW	ahb1_port0_pri_in	AHB1 总线 M1 端口优先级。 7 为最高优先级。	0x4

MISC_CTRL21

MISC_CTRL21 为总线优先级控制寄存器 8。

Offset Address: 0x0054 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	media0_timeout_en_m2	MEDIA0 总线 M2 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[30:16]	RW	media0_timeout_value_m2	MEDIA0 总线 M2 端口的 timeout 计数值。 计数值 = media0_timeout_value_m2 × 2。	0x0000
[15]	RW	media0_timeout_en_m1	MEDIA0 总线 M1 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[14:0]	RW	media0_timeout_value_m1	MEDIA0 总线 M1 端口的 timeout 计数值。 计数值 = media0_timeout_value_m1 × 2。	0x0000

MISC_CTRL22

MISC_CTRL22 为总线优先级控制寄存器 9。

Offset Address: 0x0058 Total Reset Value: 0x0000_0034

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	保留。	0x0000000



Bits	Access	Name	Description	Reset
[6:4]	RW	media0_port1_pri_in	MEDIA0 总线 M2 端口优先级。 7 为最高优先级。	0x3
[3]	-	reserved	保留。	0x0
[2:0]	RW	media0_port0_pri_in	MEDIA0 总线 M1 端口优先级。 7 为最高优先级。	0x4

MISC_CTRL23

MISC_CTRL23 为总线优先级控制寄存器 10。

Offset Address: 0x005C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	media1_timeout_en_m2	MEDIA1 总线 M2 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[30:16]	RW	media1_timeout_value_m2	MEDIA1 总线 M2 端口的 timeout 计数值。 计数值 = media1_timeout_value_m2 × 2。	0x0000
[15]	RW	media1_timeout_en_m1	MEDIA1 总线 M1 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[14:0]	RW	media1_timeout_value_m1	MEDIA1 总线 M1 端口的 timeout 计数值。 计数值 = media1_timeout_value_m1 × 2。	0x0000

MISC_CTRL24

MISC_CTRL24 为总线优先级控制寄存器 11。

Offset Address: 0x0060 Total Reset Value: 0x0000_0034

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	保留。	0x0000000
[6:4]	RW	media1_port1_pri_in	MEDIA1 总线 M2 端口优先级。 7 为最高优先级。	0x3



Bits	Access	Name	Description	Reset
[3]	-	reserved	保留。	0x0
[2:0]	RW	media1_port0_pri_in	MEDIA1 总线 M1 端口优先级。 7 为最高优先级。	0x4

MISC_CTRL25

MISC_CTRL25 为总线优先级控制寄存器 12。

Offset Address: 0x0064 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	media2_timeout_en_m2	MEDIA2 总线 M2 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[30:16]	RW	media2_timeout_value_m2	MEDIA2 总线 M2 端口的 timeout 计数值。 计数值 = media2_timeout_value_m2 × 2。	0x0000
[15]	RW	media2_timeout_en_m1	MEDIA2 总线 M1 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[14:0]	RW	media2_timeout_value_m1	MEDIA2 总线 M1 端口的 timeout 计数值。 计数值 = media2_timeout_value_m1 × 2。	0x0000

MISC_CTRL26

MISC_CTRL26 为总线优先级控制寄存器 13。

Offset Address: 0x0068 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	media2_timeout_en_m4	MEDIA2 总线 M4 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[30:16]	RW	media2_timeout_value_m4	MEDIA2 总线 M4 端口的 timeout 计数值。 计数值 = media2_timeout_value_m4 × 2。	0x0000



Bits	Access	Name	Description	Reset
[15]	RW	media2_timeout_en_m3	MEDIA2 总线 M3 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[14:0]	RW	media2_timeout_value_m3	MEDIA2 总线 M3 端口的 timeout 计数值。 计数值 = media2_timeout_value_m3 × 2。	0x0000

MISC_CTRL27

MISC_CTRL27 为总线优先级控制寄存器 14。

Offset Address: 0x006C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	media2_timeout_en_m6	MEDIA2 总线 M6 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[30:16]	RW	media2_timeout_value_m6	MEDIA2 总线 M6 端口的 timeout 计数值。 计数值 = media2_timeout_value_m6 × 2。	0x0000
[15]	RW	media2_timeout_en_m5	MEDIA2 总线 M5 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[14:0]	RW	media2_timeout_value_m5	MEDIA2 总线 M5 端口的 timeout 计数值。 计数值 = media2_timeout_value_m5 × 2。	0x0000

MISC_CTRL28

MISC_CTRL28 为总线优先级控制寄存器 15。

Offset Address: 0x0070 Total Reset Value: 0x0000_1234

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x00
[26:24]	RW	media2_port6_pri_in	MEDIA2 总线 M7 端口优先级。 7 为最高优先级。	0x0



Bits	Access	Name	Description	Reset
[23]	-	reserved	保留。	0x0
[22:20]	RW	media2_port5_pri_in	MEDIA2 总线 M6 端口优先级。 7 为最高优先级。	0x0
[19]	-	reserved	保留。	0x0
[18:16]	RW	media2_port4_pri_in	MEDIA2 总线 M5 端口优先级。 7 为最高优先级。	0x0
[15]	-	reserved	保留。	0x0
[14:12]	RW	media2_port3_pri_in	MEDIA2 总线 M4 端口优先级。 7 为最高优先级。	0x1
[11]	-	reserved	保留。	0x0
[10:8]	RW	media2_port2_pri_in	MEDIA2 总线 M3 端口优先级。 7 为最高优先级。	0x2
[7]	-	reserved	保留。	0x0
[6:4]	RW	media2_port1_pri_in	MEDIA2 总线 M2 端口优先级。 7 为最高优先级。	0x3
[3]	-	reserved	保留。	0x0
[2:0]	RW	media2_port0_pri_in	MEDIA2 总线 M1 端口优先级。 7 为最高优先级。	0x4

MISC_CTRL29

MISC_CTRL29 为总线优先级控制寄存器 16。

Offset Address: 0x0074 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	vivo_timeout_en_m2	VIVO 总线 M2 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[30:16]	RW	vivo_timeout_value_m2	VIVO 总线 M2 端口的 timeout 计数值。 计数值 = vivo_timeout_value_m2 × 2。	0x0000
[15]	RW	vivo_timeout_en_m1	VIVO 总线 M1 端口的 timeout 计数使能。 0:禁止;	0x0



Bits	Access	Name	Description	Reset
			1:使能。	
[14:0]	RW	vivo_timeout_value_m1	VIVO 总线 M1 端口的 timeout 计数值。 计数值 = vivo_timeout_value_m1 × 2。	0x0000

MISC_CTRL30

MISC_CTRL30 为总线优先级控制寄存器 17。

Offset Address: 0x0078 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	media2_timeout_en_m7	MEDIA2 总线 M7 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[30:16]	RW	media2_timeout_value_m7	MEDIA2 总线 M7 端口的 timeout 计数值。 计数值 = media2_timeout_value_m7 × 2。	0x0000
[15]	RW	vivo_timeout_en_m3	VIVO 总线 M3 端口的 timeout 计数使能。 0:禁止; 1:使能。	0x0
[14:0]	RW	vivo_timeout_value_m3	VIVO 总线 M3 端口的 timeout 计数值。 计数值 = vivo_timeout_value_m3 × 2。	0x0000

MISC_CTRL31

MISC_CTRL31 为总线优先级控制寄存器 18。

Offset Address: 0x007C Total Reset Value: 0x0000_0234

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留。	0x000000
[10:8]	RW	vivo_port2_pri_in	VIVO 总线 M3 端口优先级。 7 为最高优先级。	0x2
[7]	-	reserved	保留。	0x0
[6:4]	RW	vivo_port1_pri_in	VIVO 总线 M2 端口优先级。 7 为最高优先级。	0x3



Bits	Access	Name	Description	Reset
[3]	-	reserved	保留。	0x0
[2:0]	RW	vivo_port0_pri_in	VIVO 总线 M1 端口优先级。 7 为最高优先级。	0x4

MISC_CTRL32

MISC_CTRL32 为 MDDRC W QOS 寄存器 0。

Offset Address: 0x0080 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30:28]	RW	nnie_l_w_qos	NNIE L 写通道 QOS 分组。	0x0
[27]	-	reserved	保留。	0x0
[26:24]	RW	nnie_h_w_qos	NNIE H 写通道 QOS 分组。	0x0
[23]	-	reserved	保留。	0x0
[22:20]	RW	gdc_w_qos	GDC 写通道 QOS 分组。	0x0
[19]	-	reserved	保留。	0x0
[18:16]	RW	viproc_w_qos	VIPROC 写通道 QOS 分组。	0x0
[15]	-	reserved	保留。	0x0
[14:12]	RW	vpss_w_qos	VPSS 写通道 QOS 分组。	0x0
[11]	-	reserved	保留。	0x0
[10:8]	RW	aio_w_qos	AIAO 写通道 QOS 分组。	0x0
[7]	-	reserved	保留。	0x0
[6:4]	RW	vdv_w_qos	VDP 写通道 QOS 分组。	0x0
[3]	-	reserved	保留。	0x0
[2:0]	RW	vicap_w_qos	VICAP 写通道 QOS 分组。	0x0

MISC_CTRL33

MISC_CTRL33 为 MDDRC W QOS 寄存器 1。

Offset Address: 0x0084 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30:28]	RW	scd_w_qos	SCD 写通道 QOS 分组。	0x0
[27]	-	reserved	保留。	0x0
[26:24]	RW	spacc_w_qos	SPACC 写通道 QOS 分组。	0x0
[23]	-	reserved	保留。	0x0
[22:20]	RW	fmc_w_qos	FMC 写通道 QOS 分组。	0x0
[19]	-	reserved	保留。	0x0
[18:16]	RW	emmc_w_qos	eMMC 写通道 QOS 分组。	0x0
[15]	-	reserved	保留。	0x0
[14:12]	RW	sdio0_w_qos	SDIO0 写通道 QOS 分组。	0x0
[11]	-	reserved	保留。	0x0
[10:8]	RW	dmac_w_qos	DMAC 写通道 QOS 分组。	0x0
[7]	-	reserved	保留。	0x0
[6:4]	RW	usb_w_qos	USB 写通道 QOS 分组。	0x0
[3]	-	reserved	保留。	0x0
[2:0]	RW	eth_w_qos	ETH 写通道 QOS 分组。	0x0

MISC_CTRL34

MISC_CTRL34 为 MDDRC W QOS 寄存器 2。

Offset Address: 0x0088 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30:28]	RW	ddrt_w_qos	DDRT 写通道 QOS 分组。	0x0
[27]	-	reserved	保留。	0x0
[26:24]	RW	gzip_w_qos	GZIP 写通道 QOS 分组。	0x0
[23]	-	reserved	保留。	0x0
[22:20]	RW	tde_w_qos	TDE 写通道 QOS 分组。	0x0
[19]	-	reserved	保留。	0x0



[18:16]	RW	ive_w_qos	IVE 写通道 QOS 分组。	0x0
[15]	-	reserved	保留。	0x0
[14:12]	RW	jpge_w_qos	JPGE 写通道 QOS 分组。	0x0
[11]	-	reserved	保留。	0x0
[10:8]	RW	vgs_w_qos	VGS 写通道 QOS 分组。	0x0
[7]	-	reserved	保留。	0x0
[6:4]	RW	vedu_w_qos	VEDU 写通道 QOS 分组。	0x0
[3]	-	reserved	保留。	0x0
[2:0]	RW	cpu_w_qos	CPU 写通道 QOS 分组。	0x0

MISC_CTRL35

MISC_CTRL35 为 MDDRC W QOS 寄存器 3。

Offset Address: 0x008C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	保留	0x0000000
[6:4]	RW	sdio1_w_qos	SDIO1 写通道 QOS 分组。	0x0
[3]	-	reserved	保留	0x0
[2:0]	RW	jpgd_w_qos	JPGD 写通道 QOS 分组。	0x0

MISC_CTRL36

MISC_CTRL36 为 MDDRC R QOS 寄存器 0。

Offset Address: 0x0090 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30:28]	RW	nnie_l_r_qos	NNIE L 读通道 QOS 分组。	0x0
[27]	-	reserved	保留。	0x0
[26:24]	RW	nnie_h_r_qos	NNIE H 读通道 QOS 分组。	0x0
[23]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[22:20]	RW	gdc_r_qos	GDC 读通道 QOS 分组。	0x0
[19]	-	reserved	保留。	0x0
[18:16]	RW	viproc_r_qos	VIPROC 读通道 QOS 分组。	0x0
[15]	-	reserved	保留。	0x0
[14:12]	RW	vpss_r_qos	VPSS 读通道 QOS 分组。	0x0
[11]	-	reserved	保留。	0x0
[10:8]	RW	aio_r_qos	AIAO 读通道 QOS 分组。	0x0
[7]	-	reserved	保留。	0x0
[6:4]	RW	vdv_r_qos	VDP 读通道 QOS 分组。	0x0
[3]	-	reserved	保留。	0x0
[2:0]	RW	vicap_r_qos	VICAP 读通道 QOS 分组。	0x0

MISC_CTRL37

MISC_CTRL37 为 MDDRC R QOS 寄存器 1。

Offset Address: 0x0094 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30:28]	RW	scd_r_qos	SCD 读通道 QOS 分组。	0x0
[27]	-	reserved	保留。	0x0
[23]	-	reserved	保留。	0x0
[22:20]	RW	fmc_r_qos	FMC 读通道 QOS 分组。	0x0
[19]	-	reserved	保留。	0x0
[18:16]	RW	emmc_r_qos	eMMC 读通道 QOS 分组。	0x0
[15]	-	reserved	保留。	0x0
[14:12]	RW	sdio0_r_qos	SDIO0 读通道 QOS 分组。	0x0
[11]	-	reserved	保留。	0x0
[10:8]	RW	dmac_r_qos	DMAC 读通道 QOS 分组。	0x0
[7]	-	reserved	保留。	0x0



[6:4]	RW	usb_r_qos	USB 读通道 QOS 分组。	0x0
[3]	-	reserved	保留。	0x0
[2:0]	RW	eth_r_qos	ETH 读通道 QOS 分组。	0x0

MISC_CTRL38

MISC_CTRL38 为 MDDRC R QOS 寄存器 2。

Offset Address: 0x0098 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30:28]	RW	ddrt_r_qos	DDRT 读通道 QOS 分组。	0x0
[27]	-	reserved	保留。	0x0
[26:24]	RW	gzip_r_qos	GZIP 读通道 QOS 分组。	0x0
[23]	-	reserved	保留。	0x0
[22:20]	RW	tde_r_qos	TDE 读通道 QOS 分组。	0x0
[19]	-	reserved	保留。	0x0
[18:16]	RW	ive_r_qos	IVE 读通道 QOS 分组。	0x0
[15]	-	reserved	保留。	0x0
[14:12]	RW	jpge_r_qos	JPGE 读通道 QOS 分组。	0x0
[11]	-	reserved	保留。	0x0
[10:8]	RW	vgs_r_qos	VGS 读通道 QOS 分组。	0x0
[7]	-	reserved	保留。	0x0
[6:4]	RW	vedu_r_qos	VEDU 读通道 QOS 分组。	0x0
[3]	-	reserved	保留。	0x0
[2:0]	RW	cpu_r_qos	CPU 读通道 QOS 分组。	0x0

MISC_CTRL39

MISC_CTRL39 为 MDDRC R QOS 寄存器 3。

Offset Address: 0x009C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:7]	-	reserved	保留	0x00000000
[6:4]	RW	sdio1_r_qos	SDIO1 读通道 QOS 分组。	0x0
[3]	-	reserved	保留	0x0
[2:0]	RW	jpgd_r_qos	JPGD 读通道 QOS 分组。	0x0

MISC_CTRL44

MISC_CTRL44 为 SVB 控制寄存器。

Offset Address: 0x00B0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:16]	RW	svb_pwm_duty	SVB 输出的高电平周期数，实际高电平周期数为此数值加 1。	0x000
[15:14]	-	reserved	保留。	0x0
[13:4]	RW	svb_pwm_period	SVB 输出的循环周期数，实际周期数为此数值加 1。	0x000
[3]	-	reserved	保留。	0x0
[2]	WO	svb_pwm_load	SVB 输出参数加载控制,向此值写 1 使参数生效。	0x0
[1]	RW	svb_pwm_inv	SVB 输出正反相位控制。 0:正相; 1:反相。	0x0
[0]	RW	svb_pwm_enable	SVB 使能控制。 0:不使能; 1:使能。	0x0

MISC_CTRL45

MISC_CTRL45 为 Tsensor 控制寄存器。

Offset Address: 0x00B4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	tsensor_en	TSensor 使能控制。	0x0



Bits	Access	Name	Description	Reset
			0:关闭; 1:使能。	
[30]	RW	tsensor_monitor_en	TSensor 温度循环监测使能。 0:单次测量; 1:循环测量。	0x0
[29:28]	-	reserved	保留。	0x0
[27:20]	RW	tsensor_monitor_period	TSensor 温度循环监测周期, 计时基准为 2ms。	0x00
[19:10]	RW	tsensor_uplimit	温度上溢值。	0x000
[9:0]	RW	tsensor_lowlimit	温度下溢值。	0x000

MISC_CTRL46

MISC_CTRL46 为 Tsensor 状态寄存器。

Offset Address: 0x00B8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5]	RW	tsensor_vcalib1_en	TSensor 的 VCALIB1 使能信号: 0:不使能; 1:使能。	0x0
[4]	RW	tsensor_temp_calib	TSensor 校准算法选择位: 0:开启校准算法; 1:不开启校准算法。	0x0
[3:2]	RW	tsensor_temp_ct_sel	TSensor 转换一次温度码值时间选择: 00: 0.512ms; 01: 1.024ms; 10: 0.256ms; 11: 2.048ms;	0x0
[1]	RO	tsensor_low_warning	温度下溢告警, 高有效。	0x0
[0]	RO	tsensor_up_warning	温度上溢告警, 高有效。	0x0



MISC_CTRL47

MISC_CTRL47 为 Tsensor 温度记录值寄存器 0。

Offset Address: 0x00BC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:16]	RO	tsensor_result1	温度记录值 1。	0x000
[15:10]	-	reserved	保留。	0x00
[9:0]	RO	tsensor_result0	温度记录值 0。	0x000

MISC_CTRL48

MISC_CTRL48 为 Tsensor 温度记录值寄存器 1。

Offset Address: 0x00C0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:16]	RO	tsensor_result3	温度记录值 3。	0x000
[15:10]	-	reserved	保留。	0x00
[9:0]	RO	tsensor_result2	温度记录值 2。	0x000

MISC_CTRL49

MISC_CTRL49 为 Tsensor 温度记录值寄存器 2。

Offset Address: 0x00C4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:16]	RO	tsensor_result5	温度记录值 5。	0x000
[15:10]	-	reserved	保留。	0x00
[9:0]	RO	tsensor_result4	温度记录值 4。	0x000



MISC_CTRL50

MISC_CTRL50 为 Tsensor 温度记录值寄存器 3。

Offset Address: 0x00C8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:16]	RO	tsensor_result7	温度记录值 7。	0x000
[15:10]	-	reserved	保留。	0x00
[9:0]	RO	tsensor_result6	温度记录值 6。	0x000

MISC_CTRL51

MISC_CTRL51 为升级模式清除寄存器。

Offset Address: 0x00CC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	WO	update_mode_clear	向此位写 1 清除升级模式，在升级完成时配置此寄存器。	0x0

MISC_CTRL52

MISC_CTRL52 为 HPM 控制寄存器 0。

Offset Address: 0x00D0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	cpu_hpm_srst_req	CPU HPM 复位请求，高有效。 0:撤消复位； 1:复位。	0x0
[30]	RW	cpu_hpm_monitor_en	CPU HPM 循环监控使能。 0:单次监控； 1:循环监控。	0x0
[29]	RW	cpu_hpm_en	CPU HPM 测量使能。 0:关闭 HPM 测量； 1:开始 HPM 测量。	0x0



Bits	Access	Name	Description	Reset
[28:16]	-	reserved	保留。	0x0000
[15:8]	RW	cpu_hpm_monitor_period	CPU HPM 循环监控周期，监控周期为： CPU_hpm_monitor_period*2.048ms。	0x00
[7:6]	-	reserved	保留。	0x0
[5:0]	RW	cpu_hpm_div	CPU HPM 参考时钟分频比配置，实际分频系数为此数值加 1。	0x00

MISC_CTRL53

MISC_CTRL53 为 HPM 控制寄存器 1。

Offset Address: 0x00D4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:22]	RW	cpu_hpm_offset	CPU HPM OFFSET 值。	0x0
[21:20]	RW	cpu_hpm_shift	CPU HPM SHIFT 值。	0x0
[19:10]	RW	cpu_hpm_lowlimit	CPU HPM 下限值。	0x000
[9:0]	RW	cpu_hpm_uplimit	CPU HPM 上限值。	0x000

MISC_CTRL54

MISC_CTRL54 为 HPM 状态寄存器 0。

Offset Address: 0x00D8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RO	cpu_hpm_up_warning	CPU HPM 上溢告警。高有效	0x0
[30]	RO	cpu_hpm_low_warning	CPU HPM 下溢告警。高有效。	0x0
[29]	RO	cpu_hpm_pc_valid	CPU HPM 输出有效指示。 0:无效; 1:有效。	0x0
[28:26]	-	reserved	保留。	0x0
[25:16]	RO	cpu_hpm_pc_record	CPU HPM 原始码型 1。	0x000



Bits	Access	Name	Description	Reset
		1		
[15:10]	-	reserved	保留。	0x00
[9:0]	RO	cpu_hpm_pc_record 0	CPU HPM 原始码型 0。	0x000

MISC_CTRL55

MISC_CTRL55 为 HPM 状态寄存器 1。

Offset Address: 0x00DC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:27]	RO	cpu_hpm_rcc	CPU HPM 输出有 RCC 码。	0x00
[26]	-	reserved	保留。	0x0
[25:16]	RO	cpu_hpm_pc_record 3	CPU HPM 原始码型 3。	0x000
[15:10]	-	reserved	保留。	0x00
[9:0]	RO	cpu_hpm_pc_record 2	CPU HPM 原始码型 2。	0x000

MISC_CTRL56

MISC_CTRL56 为 SHPM 控制寄存器。

Offset Address: 0x00E0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	viproc_shpm_stat	SHPM 状态。	0x0000
[15:0]	RW	viproc_shpm_cfg	SHPM 配置。	0x0000

MISC_CTRL57

MISC_CTRL57 为 DDRC LOCK 寄存器。

Offset Address: 0x0100 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000



Bits	Access	Name	Description	Reset
[5]	RW	ddr_phy_write_lock	DDR PHY 寄存器锁定控制。 0: DDR PHY 寄存器可以被改写; 1: DDR PHY 寄存器不能被改写。	0x0
[4]	RW	ddr_qosbuf_write_lock	DDR QOSBUF 寄存器锁定控制。 0: DDR QOSBUF 寄存器可以被改写; 1: DDR QOSBUF 寄存器不能被改写。	0x0
[3]	RW	ddr_addr_atrb_write_lock	DDR ADDR ATRB 寄存器锁定控制。 0: DDR ADDR ATRB 寄存器可以被改写; 1: DDR ADDR ATRB 寄存器不能被改写。	0x0
[2]	RW	ddr_dmc_write_lock	DDR DMC 寄存器锁定控制。 0: DDR DMC 寄存器可以被改写; 1: DDR DMC 寄存器不能被改写。	0x0
[1]	RW	ddr_axiif_write_lock	DDR AXIIF 寄存器锁定控制。 0: DDR AXIIF 寄存器可以被改写; 1: DDR AXIIF 寄存器不能被改写。	0x0
[0]	RW	ddr_secure_boot_lock	DDR SECURE 寄存器锁定控制。 0: DDR SECURE 寄存器可以被改写; 1: DDR SECURE 寄存器不能被改写。	0x0

MISC_CTRL58

MISC_CTRL58 为安全访问控制寄存器 0。

Offset Address: 0x0104 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	fmc_mem_slv_sec_acc_disable	FMC MEM 安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[30]	RW	fmc_reg_slv_sec_acc_disable	FMC 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[29]	RW	emmc_slv_sec_acc_disable	eMMC 寄存器安全访问开关。	0x0



Bits	Access	Name	Description	Reset
			0: 启用安全访问控制; 1: 关闭安全访问控制。	
[28]	RW	sdio0_slv_sec_acc_disable	SDIO0 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[27]	RW	usb2_slv_sec_acc_disable	USB2 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[26]	RW	bootram_slv_sec_acc_disable	BOOTROM/BOOTRAM 安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[25]	RW	scd_slv_sec_acc_disable	SCD 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[24]	RW	sdio1_slv_sec_acc_disable	SDIO0 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[23]	RW	eth_slv_sec_acc_disable	ETH 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[22]	RW	mipitx_slv_sec_acc_disable	MIPITX 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[21]	RW	jpgd_slv_sec_acc_disable	JPGD 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[20]	RW	ddrt_slv_sec_acc_disable	DDRT 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[19]	RW	gzip_slv_sec_acc_disable	GZIP 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[18]	RW	tde_slv_sec_acc_disable	TDE 寄存器安全访问开关。	0x0



Bits	Access	Name	Description	Reset
			0: 启用安全访问控制; 1: 关闭安全访问控制。	
[17]	RW	ive_slv_sec_acc_disable	IVE 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[16]	RW	jpgc_slv_sec_acc_disable	JPGC 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[15]	RW	vgs_slv_sec_acc_disable	VGS 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[14]	RW	powerctrl_slv_sec_acc_disable	PowerControl 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[13]	RW	rtc_slv_sec_acc_disable	RTC 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[12]	RW	ddrsubsys_slv_sec_acc_disable	MDDRC 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[11]	RW	crg_slv_sec_acc_disable	CRG 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[10]	RW	wdg1_slv_sec_acc_disable	WDG1 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[9]	RW	wdg0_slv_sec_acc_disable	WDG0 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[8]	RW	uart4_slv_sec_acc_disable	UART4 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[7]	RW	uart3_slv_sec_acc_disable	UART3 寄存器安全访问开关。	0x0



Bits	Access	Name	Description	Reset
			0: 启用安全访问控制; 1: 关闭安全访问控制。	
[6]	RW	uart2_slv_sec_acc_disable	UART2 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[5]	RW	uart1_slv_sec_acc_disable	UART1 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[4]	RW	uart0_slv_sec_acc_disable	UART0 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[3]	RW	timer67_slv_sec_acc_disable	Timer6/Timer7 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[2]	RW	timer45_slv_sec_acc_disable	Timer4/Timer5 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[1]	RW	timer23_slv_sec_acc_disable	Timer2/Timer3 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[0]	RW	timer01_slv_sec_acc_disable	Timer0/Timer1 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0

MISC_CTRL59

MISC_CTRL59 为安全访问控制寄存器 1。

Offset Address: 0x0108 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:18]	-	reserved	保留。	0x0000
[17]	RW	vedu_slv_sec_acc_disable	VEDU 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0



Bits	Access	Name	Description	Reset
[16]	RW	gdc_slv_sec_acc_disable	GDC 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[15]	RW	nnie_slv_sec_acc_disable	NNIE 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[14]	RW	vpss_slv_sec_acc_disable	VPSS 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[13]	RW	viproc_slv_sec_acc_disable	VIPROC 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[12]	RW	audiocodec_slv_sec_acc_disable	AudioCodec 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[11]	RW	aio_slv_sec_acc_disable	AIAO 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[10]	RW	hdmitx_slv_sec_acc_disable	HDMITX 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[9]	RW	vdp_slv_sec_acc_disable	VDP 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[8]	RW	mipirx_slv_sec_acc_disable	MIPIRX 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[7]	RW	vicap_slv_sec_acc_disable	VICAP 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[6]	RW	klad_slv_sec_acc_disable	KLAD 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0



Bits	Access	Name	Description	Reset
[5]	RW	dmac_slv_sec_acc_disable	DMAC 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[4]	RW	otp_slv_sec_acc_disable	OTP 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[3]	RW	trng_slv_sec_acc_disable	TRNG 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[2]	RW	rsa1_slv_sec_acc_disable	RSA1 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[1]	RW	rsa0_slv_sec_acc_disable	RSA0 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0
[0]	RW	spacc_slv_sec_acc_disable	SPACC 寄存器安全访问开关。 0: 启用安全访问控制; 1: 关闭安全访问控制。	0x0

MISC_CTRL60

MISC_CTRL60 为安全访问控制寄存器 2。

Offset Address: 0x010C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:23]	-	reserved	保留。	0x000
[22]	RW	mipitx_slv_acc_ctrl	MIPI TX 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 MIPI TX 寄存器。	0x0
[21]	RW	jpgd_slv_acc_ctrl	JPGD 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 JPGD 寄存器。	0x0



Bits	Access	Name	Description	Reset
[20]	RW	ddrt_slv_acc_ctrl	DDRT 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 DDRT 寄存器。	0x0
[19]	RW	gzip_slv_acc_ctrl	GZIP 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 GZIP 寄存器。	0x0
[18]	RW	tde_slv_acc_ctrl	TDE 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 TDE 寄存器。	0x0
[17]	RW	ive_slv_acc_ctrl	IVE 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 IVE 寄存器。	0x0
[16]	RW	jpgc_slv_acc_ctrl	JPGC 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 JPGC 寄存器。	0x0
[15]	RW	vgs_slv_acc_ctrl	VGS 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 VGS 寄存器。	0x0
[14]	RW	powerctrl_slv_acc_ctrl	PowerControl 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 PowerControl 寄存器。	0x0
[13]	RW	rtc_slv_acc_ctrl	RTC 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 RTC 寄存器。	0x0
[12]	RW	ddrsubsys_slv_acc_ctrl	MDDRC/DDRPHY 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写	0x0



Bits	Access	Name	Description	Reset
			MDDRC/DDRPHY 寄存器。	
[11]	RW	crg_slv_acc_ctrl	CRG 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 CRG 寄存器。	0x0
[10]	RW	wdg1_slv_acc_ctrl	WDG1 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 WDG1 寄存器。	0x0
[9]	RW	wdg0_slv_acc_ctrl	WDG0 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 WDG0 寄存器。	0x0
[8]	RW	uart4_slv_acc_ctrl	UART4 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 UART4 寄存器。	0x0
[7]	RW	uart3_slv_acc_ctrl	UART3 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 UART3 寄存器。	0x0
[6]	RW	uart2_slv_acc_ctrl	UART2 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 UART2 寄存器。	0x0
[5]	RW	uart1_slv_acc_ctrl	UART1 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 UART1 寄存器。	0x0
[4]	RW	uart0_slv_acc_ctrl	UART0 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 UART0 寄存器。	0x0
[3]	RW	timer67_slv_acc_ctrl	Timer6/Timer7 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写	0x0



Bits	Access	Name	Description	Reset
			Timer6/Timer7 寄存器。	
[2]	RW	timer45_slv_acc_ctrl	Timer4/Timer5 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 Timer4/Timer5 寄存器。	0x0
[1]	RW	timer23_slv_acc_ctrl	Timer2/Timer3 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 Timer2/Timer3 寄存器。	0x0
[0]	RW	timer01_slv_acc_ctrl	Timer0/Timer1 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 Timer0/Timer1 寄存器。	0x0

MISC_CTRL61

MISC_CTRL61 为安全访问控制寄存器 3。

Offset Address: 0x0110 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:18]	-	reserved	保留。	0x0000
[17]	RW	vedu_slv_acc_ctrl	VEDU 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 VEDU 寄存器。	0x0
[16]	RW	gdc_slv_acc_ctrl	GDC 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 GDC 寄存器。	0x0
[15]	RW	nnie_slv_acc_ctrl	NNIE 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 NNIE 寄存器。	0x0
[14]	RW	vpss_slv_acc_ctrl	VPSS 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 VPSS 寄存	0x0



Bits	Access	Name	Description	Reset
			器。	
[13]	RW	viproc_slv_acc_ctrl	VIPROC 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 VIPROC 寄存器。	0x0
[12]	RW	audiocodec_slv_acc_ctrl	AudioCodec 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 AudioCodec 寄存器。	0x0
[11]	RW	aio_slv_acc_ctrl	AIAO 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 AIAO 寄存器。	0x0
[10]	RW	hdmitx_slv_acc_ctrl	HDMITX 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 HDMITX 寄存器。	0x0
[9]	RW	vdp_slv_acc_ctrl	VDP 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 VDP 寄存器。	0x0
[8]	RW	mipirx_slv_acc_ctrl	MIPIRX 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 MIPIRX 寄存器。	0x0
[7]	RW	vicap_slv_acc_ctrl	VICAP 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 VICAP 寄存器。	0x0
[6]	RW	klad_slv_acc_ctrl	KLAD 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 KLAD 寄存器。	0x0
[5]	RW	dmac_slv_acc_ctrl	DMAC 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 DMAC 寄存	0x0



Bits	Access	Name	Description	Reset
			器。	
[4]	RW	otp_slv_acc_ctrl	OTP 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 OTP 寄存器。	0x0
[3]	RW	trng_slv_acc_ctrl	TRNG 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 TRNG 寄存器。	0x0
[2]	RW	rsa1_slv_acc_ctrl	RSA1 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 RSA1 寄存器。	0x0
[1]	RW	rsa0_slv_acc_ctrl	RSA0 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 RSA0 寄存器。	0x0
[0]	RW	spacc_slv_acc_ctrl	SPACC 寄存器安全访问控制模式。 0: 普通安全访问控制。 1: 强制非安全 CPU 不能改写 SPACC 寄存器。	0x0

MISC_CTRL62

MISC_CTRL62 为安全访问控制寄存器 4。

Offset Address: 0x0114 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:18]	-	reserved	保留。	0x0000
[17]	RW	fmc_mst_sec_acc	FMC Master 访问安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0
[16]	RW	emmc_mst_sec_acc	eMMC Master 访问安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0



Bits	Access	Name	Description	Reset
[15]	RW	sdio1_mst_sec_acc	SDIO1 Master 访问安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0
[14]	RW	scd_mst_sec_acc	SCD Master 访问安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0
[13]	RW	sdio0_mst_sec_acc	SDIO0 Master 访问安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0
[12]	RW	eth_mst_sec_acc	ETH Master 访问安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0
[11]	RW	aiao_mst_r_sec_acc	AIAO Master 读操作安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0
[10]	RW	aiao_mst_w_sec_acc	AIAO Master 写操作安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0
[9]	RW	jpgd_mst_r_sec_acc	JPGD Master 读操作安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0
[8]	RW	jpgd_mst_w_sec_acc	JPGD Master 写操作安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0
[7]	RW	usb2_mst_r_sec_acc	USB Master 读操作安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0
[6]	RW	usb2_mst_w_sec_acc	USB Master 写操作安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0
[5]	RW	gzip_mst_r_sec_acc	GZIP Master 读操作安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0



Bits	Access	Name	Description	Reset
[4]	RW	gzip_mst_w_sec_acc	GZIP Master 写操作安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0
[3]	RW	tde_mst_r_sec_acc	TDE Master 读操作安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0
[2]	RW	tde_mst_w_sec_acc	TDE Master 写操作安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0
[1]	RW	ive_mst_r_sec_acc	IVE Master 读操作安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0
[0]	RW	ive_mst_w_sec_acc	IVE Master 写操作安全属性设置。 0: 安全访问; 1: 非安全访问。	0x0

MISC_CTRL63

MISC_CTRL63 为安全访问控制寄存器 5。

Offset Address: 0x0118 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1:0]	RW	a7_cp15sdisable	禁止改写 CPU CP15 部分寄存器，高有效。	0x0

3.6 DMA 控制器

3.6.1 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。

3.6.2 特点

DMAC 有如下特点：

- 提供 8 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、外设到内存之间的传输。
- 支持 DMA 链表传输。
- 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。
- 各个通道均支持安全属性配置。
- 只支持小端模式。

3.6.3 功能描述

3.6.3.1 外设请求线

DMAC 的硬件请求和相应设备的对应关系如表 3-5 所示。

表3-5 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	I2C3 RX 通道 DMA 请求
7	I2C3 TX 通道 DMA 请求
8	I2C4 RX 通道 DMA 请求
9	I2C4 TX 通道 DMA 请求
10	I2C5 RX 通道 DMA 请求
11	I2C5 TX 通道 DMA 请求
12	I2C6 RX 通道 DMA 请求



外设硬件请求线编号	对应设备
13	I2C6 TX 通道 DMA 请求
14	I2C7 RX 通道 DMA 请求
15	I2C7 TX 通道 DMA 请求
16	UART0 RX 通道 DMA 请求
17	UART0 TX 通道 DMA 请求
18	UART1 RX 通道 DMA 请求
19	UART1 TX 通道 DMA 请求
20	UART2 RX 通道 DMA 请求
21	UART2 TX 通道 DMA 请求
22	UART3 RX 通道 DMA 请求
23	UART3 TX 通道 DMA 请求
24	UART4 RX 通道 DMA 请求
25	UART4 TX 通道 DMA 请求
26	SPI0 RX 通道 DMA 请求
27	SPI0 TX 通道 DMA 请求
28	SPI1 RX 通道 DMA 请求
29	SPI1 TX 通道 DMA 请求
30	SPI2 RX 通道 DMA 请求
31	SPI2 TX 通道 DMA 请求

3.6.3.2 访问空间

DMAC 可以访问的地址空间如表 3-6 所示。

表3-6 DMAC 访问空间说明

访问空间类型	描述
Memory	非安全属性的 DDR 空间
	安全属性的 DDR 空间
	共享属性的 DDR 空间
外设	UART0



访问空间类型	描述
	UART1
	UART2
	UART3
	UART4
	I2C0
	I2C1
	I2C2
	I2C3
	I2C4
	I2C5
	I2C6
	I2C7
	SPI0
	SPI1
	SPI2



注意

- 如果寄存器 C(n)_AXI_CONF[arprot]为 0x0，则表示该通道的读属性为安全，此时源地址空间只能为安全 DDR 区域或者共享 DDR 区域；
- 如果寄存器 C(n)_AXI_CONF[arprot]的 0x2，则表示该通道的读属性为非安全，此时源地址空间只能为非安全 DDR 区域或者共享 DDR 区域。
- 如果寄存器 C(n)_AXI_CONF[awprot]为 0x0，则表示该通道的写属性为安全，此时目的地址空间只能为安全 DDR 区域或者共享 DDR 区域；
- 如果寄存器 C(n)_AXI_CONF[awprot]为 0x2，则表示该通道的写属性为非安全，此时目的地址空间只能为非安全 DDR 区域或者共享 DDR 区域。
- 如果发生访问权限错误，DMAC 将会上报该通道的数据传输错误中断，清除中断后可以继续使用该通道。
- 通过访问外设时不区分读安全属性或者写完全属性。

3.6.3.3 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 65535 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。



注意

- 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。

3.6.3.4 链表传输

DMAC 支持硬件链表功能，即 DMA 会在传完基本传输后从 CPU 预置的链表空间读回下一节点的信息，并自动开始下一节点的传输，期间不需要 CPU 的干预。每个链表节点最多传输 65535Bytes。

三种基本传输均支持 DMA 链表功能。如需使能链表功能，必须配置寄存器 `C(n)_LLI[chain_en]` 为 0x2，且除链尾节点外的各节点链表信息中的 `C(n)_LLI[chain_en]` 也必须为 0x2。对于链表尾，应当配置该节点的 `C(n)_LLI` 寄存器的 `bit[chain_en]` 为 0x0，表示当该节点对应的数据块全部传输完毕后通道会自动关闭。

CPU 应首先配置链表地址（寄存器 `C(n)_LLI[lili]`），待上一节点完成后，DMA 将自动地从 DDR 空间载入下一个预置好的链表信息。并自动刷新链表信息到寄存器 `C(n)_SRC_ADDR`、`C(n)_DES_ADDR`、`C(n)_LLI`、`C(n)_CNT0`、`C(n)_CONFIG[31:1]` 中。

需要注意的是，如果通道的源地址空间为安全内存，那么链表信息也必须存放在安全内存当中，否则链表信息存放在非安全内存中。

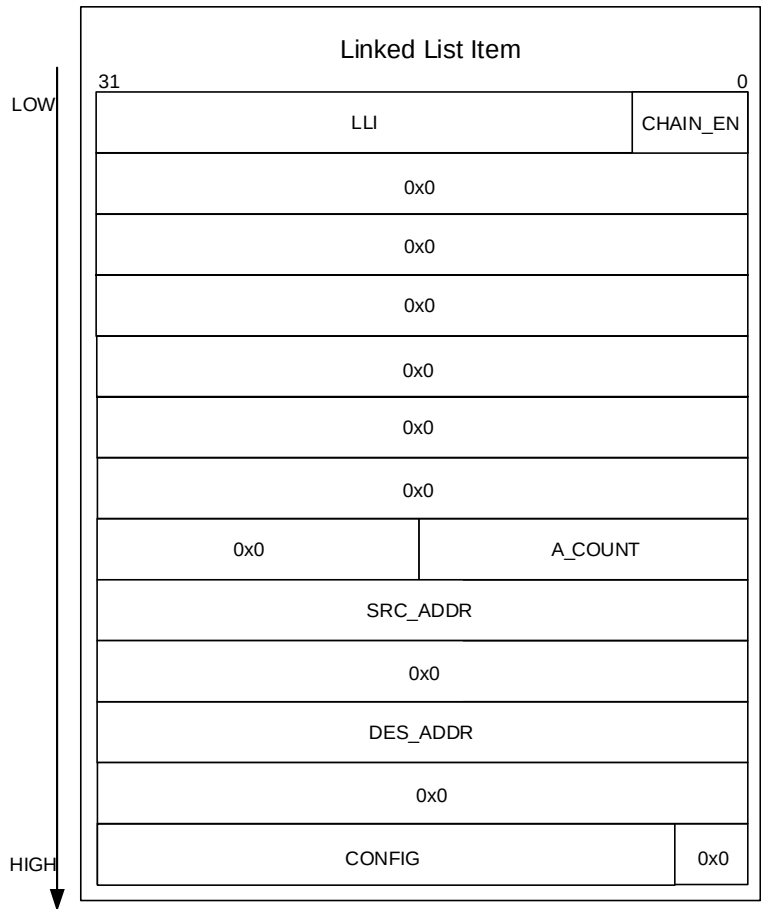


说明

- 如果通过链表刷新进来的配置参数导致配置错误，那么本次传输直接结束并产生配置错误中断。
- 如果发生读链表错误或者配置错误且上一个链表节点中断使能有效时，那么在产生错误中断的同时会产生上一个链表节点完成中断，否则只会产生错误中断。
- 链表数据结构中的 **C(n)_CONFIG** 寄存器的 **ch_en** 值不会被内部 DMAC 使用。

链表在 DDR 内存放的格式如图 3-4 所示。

图3-4 链表在 DDR 内存放的格式示意图



3.6.3.5 中断和状态

可以通过查询寄存器 **CH_STAT** 来确定对应的通道是否正在工作，在工作中的通道无法配置通道参数。

通道完成中断用来指示通道传输完成。

链表节点完成中断用来指示链表的节点传输完成 (需要配置 **C(n)_CONFIG[1]**使能)。链表的节点中断使能可以被节点的链表内容刷新进来，所以链表可以分别控制每个节点是否产生中断。需要注意的是，最后一个链表节点只产生传输完成中断，而不产生节点完成中断。

配置错误中断用来指示通道或链表发生非法配置。



读链表错误中断和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，其中读链表错误中断发生在读链表信息期间，数据传输错误中断发生在传输数据期间。

DMAC 提供各通道的当前信息查询。寄存器 [C\(n\)_CURR_CNT0](#) 用于查询剩余的数据量，寄存器 [C\(n\)_CURR_SRC_ADDR](#) 用于查询当前传输的源地址，寄存器 [C\(n\)_CURR_DES_ADDR](#) 用于查询当前传输的目的地址。

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。
- 如果链表的每个节点传输比较少的数据，且 CPU 响应节点中断和清除中断需要时间，可能会导致有些节点中断查询不到，而被一起清除。
- 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。

3.6.3.6 通道安全配置

可以通过配置寄存器 [C\(n\)_AXI_CONF](#) 的 arprot 域和 awprot 域来实现各个通道的安全属性，该域的配置遵循 AXI 总线协议，即寄存器 [C\(n\)_AXI_CONF](#) 的 bit6 和 bit18 反应通道 n 的读和写安全属性。只要 arprot 域或 awprot 域的安全标志位为 0x0，该通道就为安全通道，否则为非安全通道。

所有通道的安全属性均默认为安全通道。

说明

寄存器 [C\(n\)_AXI_CONF](#) 的 awprot 域和 arprot 域只被安全 CPU 配置。

3.6.4 工作方式

3.6.4.1 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 AXI 时钟门控，向 CRG 寄存器 [PERI_CRG101](#)[dmac_apb_cken]写 0x1，打开 DMAC 的 APB 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

3.6.4.2 设置访问方式

默认情况下，DMAC 的全部通道均为安全属性，下述章节“[3.6.4.3 初始化](#)”、“[3.6.4.4 基本传输](#)”、“[3.6.4.5 链表传输](#)”、“[3.6.4.6 中断处理](#)”的访问主体只能为安全处理器，访问的 DDR 内存属性可以是安全属性以及共享属性。



如果想要非安全处理器访问 DMAC，需要安全处理器首先执行以下步骤：

- 步骤 1. 依次向全部通道的 C(n)_AXI_CONF 寄存器的 arprot 域(bit[7:5])以及 awprot 域 (bit[19:17])配置 0b010，将通道的安全属性转换为非安全。

----结束

此时，DMAC 的全部通道均为非安全属性，下述章节“3.6.4.3 初始化”、“3.6.4.4 基本传输”、“3.6.4.5 链表传输”、“3.6.4.6 中断处理”的访问主体只能为非安全处理器，访问的 DDR 内存属性可以是非安全属性以及共享属性。

3.6.4.3 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. 根据场景需求，配置优先级控制寄存器 CH_PRI。
- 步骤 2. 向原始中断寄存器 INT_TC1_RAW、INT_TC2_RAW、INT_ERR1_RAW、INT_ERR2_RAW、INT_ERR3_RAW 的所有有效位写 0x1，清除全部中断状态。
- 步骤 3. 根据中断需求，配置中断屏蔽寄存器 INT_TC1_MASK、INT_TC2_MASK、INT_ERR1_MASK、INT_ERR2_MASK、INT_ERR3_MASK。
- 步骤 4. 依次向每个通道的各通道配置寄存器 C(n)_CONFIG[ch_en]写 0x0，关闭 DMAC 通道。

----结束

3.6.4.4 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. 读寄存器 CH_STAT，找出处于空闲的通道，并从中选择一个通道 n 用于配置。
- 步骤 2. 向寄存器 INT_TC1_RAW、INT_TC2_RAW、INT_ERR1_RAW、INT_ERR2_RAW 和 INT_ERR3_RAW 写 0x1<<n，清除选定通道的中断状态。
- 步骤 3. 写通道寄存器 C(n)_SRC_ADDR，设置源端的首物理地址。
- 步骤 4. 写通道寄存器 C(n)_DES_ADDR，设置目的端的首物理地址。
- 步骤 5. 写通道寄存器 C(n)_LLI[chain_en]为 0x0，表明本次传输为非链表传输。
- 步骤 6. 写通道寄存器 C(n)_CNT0[a_count]，设置搬运数据量，最大为 65535 Bytes。
- 步骤 7. 写通道寄存器 C(n)_CONFIG，设置地址类型、传输位宽、传输 Burst Length、传输类型以及外设号。
- 步骤 8. 写通道寄存器 C(n)_CONFIG[ch_en]为 0x1，使能通道，DMAC 开始传输。
- 步骤 9. 如果传输为内存和外设之间的传输，应该再启动对应外设，并打开其 DMA 功能。具体参考“外围设备”的 I2C、UART、SPI 章节。

----结束



说明

- 如果通道已经处于 BUSY 状态，软件向 `C(n)_CONFIG[ch_en]` 写 0x1 时，DMAC 不会对此次操作做出任何反应。
- 如果通道启动后产生了配置错误中断，则不会进行实际传输。
- 寄存器的配置只能在通道处于空闲状态才能有效的配入，如果通道正在工作，那么本次配置不会改变寄存器的值，同时 DMAC 不会通知软件本次操作失败。
- 原始中断寄存器 `INT_TC1_RAW`、`INT_TC2_RAW`、`INT_ERR1_RAW`、`INT_ERR2_RAW` 和 `INT_ERR3_RAW` 均为写 1 清 0 寄存器，写 0 和读寄存器均不会改变原始状态。

3.6.4.5 链表传输

- 步骤 1. 读寄存器 `INT_STAT`，找出处于空闲的通道，并从中选择一个通道 `n` 用于配置。
- 步骤 2. 向寄存器 `INT_TC1_RAW`、`INT_TC2_RAW`、`INT_ERR1_RAW`、`INT_ERR2_RAW` 和 `INT_ERR3_RAW` 写 `0x1 << n`，清除选定通道的中断状态。
- 步骤 3. 软件开辟链表空间，链表空间地址需要 512-bit 对齐。向链表空间按顺序填入链表内容，具体参考 3.6.3.4 “链表传输” 章节。
- 步骤 4. 写通道寄存器 `C(n)_SRC_ADDR`，设置链表首节点源端的首物理地址。
- 步骤 5. 写通道寄存器 `C(n)_DES_ADDR`，设置链表首节点目的端的首物理地址。
- 步骤 6. 写通道寄存器 `C(n)_LLI[lli]`，设置链表空间的首地址。
- 步骤 7. 写通道寄存器 `C(n)_LLI[chain_en]` 为 0x2，表示本节点不是链表尾。
- 步骤 8. 写通道寄存器 `C(n)_CNT0[a_count]`，设置本节点搬运的数据量，最大为 65535 Bytes。
- 步骤 9. 写通道寄存器 `C(n)_CONFIG`，设置本节点的地址类型、传输位宽、传输 Burst Length、传输类型、外设号以及链表中断使能。
- 步骤 10. 写通道寄存器 `C(n)_CONFIG[ch_en]` 为 0x1，使能通道，DMAC 开始传输。
- 步骤 11. 如果传输为内存和外设之间的传输，应再启动对应外设，并打开其 DMA 功能。具体参考“外围设备”的 I2C、UART、SPI 章节。

----结束

说明

具体参照“3.6.4.4 基本传输”小节的说明。

3.6.4.6 中断处理

中断程序的处理流程如下：

- 步骤 1. 读中断状态寄存器 `INT_STAT`，找出发出中断请求的通道。如果多个通道同时上报，优先服务优先级高的通道。
- 步骤 2. 读寄存器 `INT_TC1`，比较选定的位是否为 0x1，以确定对应通道发出的中断为传输完成中断。若是，则转到步骤 4 执行；否则转到步骤 3 继续执行。
- 步骤 3. 读寄存器 `INT_ERR1`、`INT_ERR2`、`INT_ERR3`，比较选定的位是否为 0x1，以确定对应通道发出的中断为错误中断。若是，则转到步骤 5 执行；否则转到步骤 6 继续执行。



步骤 4. 传输完成中断处理。可以分为以下几个子步骤：

- 写寄存器 [INT_TC1_RAW](#)，对选定的位写入 0x1，清除对应通道的中断状态。
- 取走或使用掉内存中的数据，有必要的话重新配置并启动该通道。
- 退出中断处理。

步骤 5. 错误中断处理。可以分为以下几个子步骤：

- 根据错误类型清除错误中断状态。如果是配置错误，则向寄存器 [INT_ERR1_RAW](#) 选定的位写入 0x1；如果是传输错误，则向寄存器 [INT_ERR2_RAW](#) 选定的位写入 0x1；如果是链表传输错误，则向寄存器 [INT_ERR3_RAW](#) 选定的位写入 0x1
- 给出错误信息，有必要的话重新配置并启动该通道。
- 退出中断处理。

步骤 6. 链表节点传输完成中断处理。可以分为以下几个子步骤：

- 写寄存器 [INT_TC2_RAW](#)，对选定的位写入 0x1，清除对应通道的中断状态。
- 取走或使用掉内存中的数据。
- 退出中断处理。

----结束

3.6.5 DMAC 寄存器概览

各模块的寄存器偏移地址中变量的取值范围和含义如[表 3-7](#) 所示。

表3-7 各模块的寄存器偏移地址变量表

变量名称	取值范围	描述
n	0~7	支持的通道个数。

DMAC 寄存器概览如[表 3-8](#) 所示。

- DMAC 基地址是：0x1006_0000

表3-8 DMAC 寄存器概览

偏移地址	名称	描述	页码
0x0000	INT_STAT	中断状态寄存器	3-121
0x0004	INT_TC1	通道传输完成中断状态寄存器	3-121
0x0008	INT_TC2	链表节点传输完成中断状态寄存器	3-121
0x000C	INT_ERR1	配置错误中断状态寄存器	3-122
0x0010	INT_ERR2	数据传输错误中断状态寄存器	3-122



偏移地址	名称	描述	页码
0x0014	INT_ERR3	读链表错误中断状态寄存器	3-122
0x0018	INT_TC1_MASK	通道传输完成中断屏蔽寄存器	3-123
0x001C	INT_TC2_MASK	链表节点传输完成中断屏蔽寄存器	3-123
0x0020	INT_ERR1_MASK	配置错误中断屏蔽寄存器	3-123
0x0024	INT_ERR2_MASK	数据传输错误中断屏蔽寄存器	3-124
0x0028	INT_ERR3_MASK	链表读取错误中断屏蔽寄存器	3-124
0x0404+ 0x20 × n	C(n)_CURR_CNT0	通道 n 的剩余数据量寄存器	3-124
0x0408+ 0x20 × n	C(n)_CURR_SRC_ADDR	通道 n 的源地址寄存器	3-125
0x0410+ 0x20 × n	C(n)_CURR_DES_ADDR	通道 n 的目的地址寄存器	3-125
0x0600	INT_TC1_RAW	原始通道传输完成中断状态寄存器	3-125
0x0608	INT_TC2_RAW	原始链表节点传输完成中断状态寄存器	3-126
0x0610	INT_ERR1_RAW	原始配置错误中断状态寄存器	3-126
0x0618	INT_ERR2_RAW	原始数据传输错误中断状态寄存器	3-127
0x0620	INT_ERR3_RAW	原始链表读取错误中断状态寄存器	3-127
0x0688	CH_PRI	优先级控制寄存器	3-127
0x0690	CH_STAT	DMA 状态寄存器	3-129
0x0800+ 0x40 × n	C(n)_LLI	通道 n 的链表地址低位配置寄存器	3-129
0x081C+ 0x40 × n	C(n)_CNT0	通道 n 的传输长度配置寄存器	3-130
0x0820+ 0x40 × n	C(n)_SRC_ADDR	通道 n 的源地址配置寄存器	3-130
0x0828+ 0x40 × n	C(n)_DES_ADDR	通道 n 的目的地址配置寄存器	3-130
0x0830+ 0x40 × n	C(n)_CONFIG	通道 n 的配置寄存器	3-130
0x0834+ 0x40 × n	C(n)_AXI_CONF	通道 n 的 AXI 特殊操作配置寄存器	3-133
0x0f00	DMAC_VERSION	版本寄存器	3-134



3.6.6 DMAC 寄存器描述

INT_STAT

INT_STAT 为中断状态寄存器。

Offset Address: 0x0000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RO	int_stat	DMAC 各通道经屏蔽后的中断状态，bit[7:0]分别对应通道 7~0。 0: 未产生中断； 1: 已产生中断，该中断请求可能来自该通道的错误中断或传输完成中断。	0x00

INT_TC1

INT_TC1 为通道传输完成中断状态寄存器。

Offset Address: 0x0004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RO	int_tc1	DMAC 各通道经过屏蔽后的通道传输完成中断状态，bit[7:0]分别对应通道 7~0。 0: 未产生中断； 1: 已产生中断。	0x00

INT_TC2

INT_TC2 为链表节点传输完成中断状态寄存器。

Offset Address: 0x0008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RO	int_tc2	DMAC 各通道经过屏蔽后的链表节点传输完成中断状态，bit[7:0]分别对应通道 7~0。	0x00



Bits	Access	Name	Description	Reset
			0: 未产生中断; 1: 已产生中断。	

INT_ERR1

INT_ERR1 为配置错误中断状态寄存器。

Offset Address: 0x000C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RO	int_err1	DMAC 各通道经过屏蔽后的配置错误中断状态, bit[7:0]分别对应通道 7~0。 0: 未产生中断; 1: 已产生中断。	0x00

INT_ERR2

INT_ERR2 为数据传输错误中断状态寄存器。

Offset Address: 0x0010 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RO	int_err2	DMAC 各通道经过屏蔽后的数据传输错误中断状态, bit[7:0]分别对应通道 7~0。 0: 未产生中断; 1: 已产生中断。	0x00

INT_ERR3

INT_ERR3 为读链表错误中断状态寄存器。

Offset Address: 0x0014 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RO	int_err3	DMAC 各通道经过屏蔽后的读链表错误中	0x00



Bits	Access	Name	Description	Reset
			断状态，bit[7:0]分别对应通道 7~0。 0：未产生中断； 1：已产生中断。	

INT_TC1_MASK

INT_TC1_MASK 为通道传输完成中断屏蔽寄存器。

Offset Address: 0x0018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	int_tc1_mask	DMAC 各通道传输完成中断屏蔽，bit[7:0]分别对应通道 7~0。 0：屏蔽中断； 1：不屏蔽中断。	0x00

INT_TC2_MASK

INT_TC2_MASK 为链表节点传输完成中断屏蔽寄存器。

Offset Address: 0x001C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	int_tc2_mask	DMAC 各通道链表节点传输完成中断屏蔽，bit[7:0]分别对应通道 7~0。 0：屏蔽中断； 1：不屏蔽中断。	0x00

INT_ERR1_MASK

INT_ERR1_MASK 为配置错误中断屏蔽寄存器。

Offset Address: 0x0020 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000



Bits	Access	Name	Description	Reset
[7:0]	RW	int_err1_mask	DMAC 各通道配置错误中断屏蔽，bit[7:0] 分别对应通道 7~0。 0：屏蔽中断； 1：不屏蔽中断。	0x00

INT_ERR2_MASK

INT_ERR2_MASK 为数据传输错误中断屏蔽寄存器。

Offset Address: 0x0024 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	int_err2_mask	DMAC 各数据传输错误中断屏蔽，bit[7:0] 分别对应通道 7~0。 0：屏蔽中断； 1：不屏蔽中断。	0x00

INT_ERR3_MASK

INT_ERR3_MASK 为链表读取错误中断屏蔽寄存器。

Offset Address: 0x0028 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	int_err3_mask	DMAC 各通道链表读取错误中断屏蔽，bit[7:0] 分别对应通道 7~0。 0：屏蔽中断； 1：不屏蔽中断。	0x00

C(n)_CURR_CNT0

C(n)_CURR_CNT0 为通道 n 的剩余数据量寄存器。

Offset Address: $0x0404 + 0x20 \times cn$ Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RO	curr_a_count	通道 n 当前节点剩余的 Byte 数。	0x0000

C(n)_CURR_SRC_ADDR

C(n)_CURR_SRC_ADDR 为通道 n 的源地址寄存器。该寄存器继承通道的安全属性。

Offset Address: $0x0408 + 0x20 \times cn$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	curr_src_addr	通道 n 当前正在传输的源地址。	0x00000000

C(n)_CURR_DES_ADDR

C(n)_CURR_DES_ADDR 为通道 n 的目的地址寄存器。该寄存器继承通道的安全属性。

Offset Address: $0x0410 + 0x20 \times n$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	curr_des_addr	通道 n 当前正在传输的目的地址。	0x00000000

INT_TC1_RAW

INT_TC1_RAW 为原始通道传输完成中断状态寄存器。该寄存器的各个位域继承对应通道的安全属性。

Offset Address: 0x0600 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RWC	int_tc1_raw	各通道屏蔽前的通道传输完成中断状态，bit[7:0]分别对应通道 7~0。 读： 0：未产生中断； 1：已产生中断。 写： 0：保持之前的值；	0x00



Bits	Access	Name	Description	Reset
			1: 清除通道传输完成中断。	

INT_TC2_RAW

INT_TC2_RAW 为原始链表节点传输完成中断状态寄存器。该寄存器的各个位域继承对应通道的安全属性。

Offset Address: 0x0608 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RWC	int_tc2_raw	各通道屏蔽前的链表节点传输完成中断状态，bit[7:0]分别对应通道 7~0。 读： 0: 未产生中断； 1: 已产生中断。 写： 0: 保持之前的值； 1: 清除节点完成中断。	0x00

INT_ERR1_RAW

INT_ERR1_RAW 为原始配置错误中断状态寄存器。该寄存器的各个位域继承对应通道的安全属性。

Offset Address: 0x0610 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RWC	int_err1_raw	各通道屏蔽前的配置错误中断状态，bit[7:0]分别对应通道 7~0。 读： 0: 未产生中断； 1: 已产生中断。 写： 0: 保持之前的值； 1: 清除配置错误中断。	0x00



INT_ERR2_RAW

INT_ERR2_RAW 为原始数据传输错误中断状态寄存器。该寄存器的各个位域继承对应通道的安全属性。

Offset Address: 0x0618 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RWC	int_err2_raw	各通道屏蔽前的数据传输错误中断状态， bit[7:0]分别对应通道 7~0。 读： 0：未产生中断； 1：已产生中断。 写： 0：保持之前的值； 1：清除数据传输错误中断。	0x00

INT_ERR3_RAW

INT_ERR3_RAW 为原始链表读取错误中断状态寄存器。该寄存器的各个位域继承对应通道的安全属性。

Offset Address: 0x0620 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RWC	int_err3_raw	各通道屏蔽前的链表读取错误中断状态， bit[7:0]分别对应通道 7~0。 读： 0：未产生中断； 1：已产生中断。 写： 0：保持之前的值； 1：清除链表读取错误中断。	0x00

CH_PRI

CH_PRI 为优先级控制寄存器。该寄存器只能被安全 CPU 访问。

Offset Address: 0x0688 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:14]	RW	ch7_pri	通道 7 优先级模式配置寄存器。 00: 0 优先级(最高); 01: 1 优先级; 10: 2 优先级; 11: 3 优先级(最低)。	0x0
[13:12]	RW	ch6_pri	通道 6 优先级模式配置寄存器。 00: 0 优先级(最高); 01: 1 优先级; 10: 2 优先级; 11: 3 优先级(最低)。	0x0
[11:10]	RW	ch5_pri	通道 5 优先级模式配置寄存器。 00: 0 优先级(最高); 01: 1 优先级; 10: 2 优先级; 11: 3 优先级(最低)。	0x0
[9:8]	RW	ch4_pri	通道 4 优先级模式配置寄存器。 00: 0 优先级(最高); 01: 1 优先级; 10: 2 优先级; 11: 3 优先级(最低)。	0x0
[7:6]	RW	ch3_pri	通道 3 优先级模式配置寄存器。 00: 0 优先级(最高); 01: 1 优先级; 10: 2 优先级; 11: 3 优先级(最低)。	0x0
[5:4]	RW	ch2_pri	通道 2 优先级模式配置寄存器。 00: 0 优先级(最高); 01: 1 优先级; 10: 2 优先级; 11: 3 优先级(最低)。	0x0
[3:2]	RW	ch1_pri	通道 1 优先级模式配置寄存器。 00: 0 优先级(最高); 01: 1 优先级;	0x0



Bits	Access	Name	Description	Reset
			10: 2 优先级; 11: 3 优先级(最低)。	
[1:0]	RW	ch0_pri	通道 0 优先级模式配置寄存器。 00: 0 优先级(最高); 01: 1 优先级; 10: 2 优先级; 11: 3 优先级(最低)。	0x0

CH_STAT

CH_STAT 为 DMA 状态寄存器。

Offset Address: 0x0690 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RO	ch_stat	DMAC 通道状态寄存器，bit[7:0]分别对应通道 7~0。 0: 对应的 DMAC 通道目前没有工作; 1: 对应的 DMAC 通道目前正在工作。	0x00

C(n)_LLI

C(n)_LLI 为通道 n 的链表地址配置寄存器。该寄存器继承通道的安全属性。

Offset Address: 0x0800 + 0x40 × n Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	RW	lli	通道 n 的下一个链表节点地址的[31:6]位，链表地址始终是 512-bit 对齐。	0x00000000
[5:2]	-	reserved	保留。	0x0
[1:0]	RW	chain_en	通道的链接使能。 00: 链表链接使能无效; 01: 保留; 10: 链表链接使能有效; 11: 保留。	0x0



C(n)_CNT0

C(n)_CNT0 为通道 n 的传输长度配置寄存器。该寄存器继承通道的安全属性。

Offset Address: $0x081C + 0x40 \times n$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	a_count	传输长度配置，即链表模式的单节点传输长度或非链表模式的传输总长度；有效长度为 1~65535，单位为 Byte；Account 配置为 0 会触发配置错误中断。	0x0000

C(n)_SRC_ADDR

C(n)_SRC_ADDR 为通道 n 的源地址配置寄存器。该寄存器继承通道的安全属性。

Offset Address: $0x0820 + 0x40 \times n$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	src_addr	通道 n 的源地址。	0x00000000

C(n)_DES_ADDR

C(n)_DES_ADDR 为通道 n 的目的地址配置寄存器。该寄存器继承通道的安全属性。

Offset Address: $0x0828 + 0x40 \times n$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	des_addr	通道 n 的目的地址。	0x00000000

C(n)_CONFIG

C(n)_CONFIG 为通道 n 的配置寄存器。该寄存器继承通道的安全属性。

Offset Address: $0x0830 + 0x40 \times n$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	si	源地址递增设置。 0: 源地址不递增； 1: 源地址每传一个数就递增一次。	0x0



Bits	Access	Name	Description	Reset
			源设备为外设时源地址不递增；源设备为存储器时源地址递增。	
[30]	RW	di	目的地址递增设置。 0：目的地址不递增； 1：目的地址每传一个数就递增一次。 目的设备为外设时目的地址不递增；目的设备为存储器时目的地址递增。	0x0
[29:28]	-	reserved	保留。	0x0
[27:24]	RW	sl	源传输 burst 长度。 0x0: burst length 为 1; 0x1: burst length 为 2; 0x2: burst length 为 3; 0x3: burst length 为 4; 0x4: burst length 为 5; 0x5: burst length 为 6; 0x6: burst length 为 7; 0x7: burst length 为 8; 0x8: burst length 为 9; 0x9: burst length 为 10; 0xA: burst length 为 11; 0xB: burst length 为 12; 0xC: burst length 为 13; 0xD: burst length 为 14; 0xE: burst length 为 15; 0xF: burst length 为 16。	0x0
[23:20]	RW	dl	目的传输 burst 长度。 0x0: burst length 为 1; 0x1: burst length 为 2; 0x2: burst length 为 3; 0x3: burst length 为 4; 0x4: burst length 为 5; 0x5: burst length 为 6; 0x6: burst length 为 7; 0x7: burst length 为 8; 0x8: burst length 为 9; 0x9: burst length 为 10;	0x0



Bits	Access	Name	Description	Reset
			0xA: burst length 为 11; 0xB: burst length 为 12; 0xC: burst length 为 13; 0xD: burst length 为 14; 0xE: burst length 为 15; 0xF: burst length 为 16。	
[19]	-	reserved	保留。	0x0
[18:16]	RW	sw	源数据宽度。 000: 8-bit; 001: 16-bit; 010: 32-bit; 011: 64-bit; 1xx: 保留。	0x0
[15]	-	reserved	保留。	0x0
[14:12]	RW	dw	目的数据宽度。 000: 8-bit; 001: 16-bit; 010: 32-bit; 011: 64-bit; 1xx: 保留。	0x0
[11:9]	-	reserved	保留。	0x0
[8:4]	RW	peri	外设请求号。 如果是存储器到存储器，这些位将被忽略。	0x00
[3:2]	RW	flow_ctrl	流控制和传输类型。 00: 存储器和存储器之间传输，DMAC 流控; 01: 存储器和外设之间传输，DMAC 流控; 1x: 保留。	0x0
[1]	RW	itc_en	链表节点传输完成中断使能。该位用于配置当前链表节点传输完成后是否触发节点传输完成中断。 0: 当前链表节点传输完成以后不触发链表节点传输完成中断; 1: 当前链表节点传输完成以后触发链表节	0x0



Bits	Access	Name	Description	Reset
			点传输完成中断，但是最后一个节点传输完成以后，链表节点传输完成中断不会被触发。 注意：无论该 bit 配置为 0 或 1，通道所有链表节点传输完成以后均会上报通道传输完成中断。	
[0]	RW	ch_en	当前通道使能控制。 当写该寄存器，控制当前通道的运行： 0：通道禁能； 1：通道使能。 当读该寄存器，显示当前通道的状态： 0：通道停止中； 1：通道运行中(链表传输使能时，通道一直使能)。	0x0

C(n)_AXI_CONF

C(n)_AXI_CONF 为通道 n 的 AXI 特殊操作配置寄存器。该寄存器继承通道的安全属性，awprot 位域和 arprot 位域只能被安全 CPU 访问。

Offset Address: $0x0834 + 0x40 \times n$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:20]	RW	awcache	配置 AXI 总线写命令 cache 的类型，遵循 AXI 总线协议。	0x0
[19:17]	RW	awprot	配置 AXI 总线写命令保护类型，遵循 AXI 总线协议。	0x0
[16:12]	-	reserved	保留。	0x00
[11:8]	RW	arcache	配置 AXI 总线读命令 cache 的类型，遵循 AXI 总线协议。	0x0
[7:5]	RW	arprot	配置 AXI 总线读命令保护类型，遵循 AXI 总线协议。	0x0
[4:0]	-	reserved	保留。	0x00



DMAC_VERSION

DMAC_VERSION 为版本寄存器。

Offset Address: 0x0f00 Total Reset Value: 0x0000_B310

Bits	Access	Name	Description	Reset
[31:0]	RO	dmac_version	控制器版本信息。	0x0000B310

3.7 定时器

3.7.1 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 8 个 Timer。

3.7.2 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。
- 计数时钟可配置：
主 SOC 子系统 Timer 的计数时钟可以选择为总线时钟（50MHz）或 3MHz 时钟。
- 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。

3.7.3 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。

3.7.4 工作方式

初始化

系统初始化时应对 Timer 进行初始化。初始化 TimerX（TimerX 中的“X”取值为 0~11）时应按以下步骤进行配置：

- 步骤 1. 写 `TIMERx_LOAD` 寄存器，为 Timer 载入计数初值。
- 步骤 2. 当需要 Timer 工作在周期计数模式下、且计数周期与载入 Timer 的计数初值不同时，写 `TIMERx_BGLOAD` 寄存器，设置 Timer 的计数周期。
- 步骤 3. 配置 `SC_CTRL` 寄存器，设置 Timer 的时钟使能信号的参考时钟。
- 步骤 4. 配置 `TIMERx_CONTROL` 寄存器，设置 Timer 的计数模式、计数器长度、预分频因子及中断屏蔽，同时启动 Timer 计数。

----结束

中断处理

Timer 主要用于定时产生中断，因此 Timer 的中断处理主要是激活等待定时中断的进程。操作步骤如下：

- 步骤 1. 配置 `TIMERx_INTCLR` 寄存器，清除 Timer 中断。
- 步骤 2. 激活等待该中断的进程，使其继续执行。
- 步骤 3. 当所有等待该中断的进程完成或再次执行到等待中断的休眠状态时，恢复中断现场，继续执行当前被中断的程序。

----结束

时钟选择

每个 Timer 均有 2 种计数时钟可选择。下面以 Timer0 为例，时钟选择配置流程如下：



选择总线时钟进行计数的步骤如下：

步骤 4. 配置系统控制器的 `SC_CTRL[timeren0ov]=1`。

步骤 5. 初始化 Timer，开始计数。

----结束

选择 3MHz 时钟进行计数的步骤如下：

步骤 6. 配置系统控制器的 `SC_CTRL[timeren0ov]=0`。

步骤 7. 初始化 Timer，开始计数。

----结束

3.7.5 Timer 寄存器概览

Timer 模块自有一组寄存器，每个 Timer 的寄存器除基址不相同外其他特性都相同。

- Timer0 基地址：0x1200_0000
- Timer1 基地址：0x1200_0020
- Timer2 基地址：0x1200_1000
- Timer3 基地址：0x1200_1020
- Timer4 基地址：0x1200_2000
- Timer5 基地址：0x1200_2020
- Timer6 基地址：0x1200_3000
- Timer7 基地址：0x1200_3020

说明

TIMERx 中的“x”取值为 0~7。其中 Timer0~Timer7 的寄存器相同，寄存器描述均以 Timer0 为例进行描述。

表3-9 Timer 寄存器概览

偏移地址	名称	描述	页码
0x000	TIMERx_LOAD	计数初值寄存器	3-137
0x004	TIMERx_VALUE	当前计数值寄存器	3-137
0x008	TIMERx_CONTROL	Timer 控制寄存器	3-137
0x00C	TIMERx_INTCLR	中断清除寄存器	3-138
0x010	TIMERx_RIS	原始中断寄存器	3-139
0x014	TIMERx_MIS	屏蔽后中断寄存器	3-139
0x018	TIMERx_BGLOAD	周期模式计数初值寄存器	3-139



3.7.6 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：

如果在被 TIMCLKENx 使能的 TIMCLK 的上升沿到来之前，向 TIMERx_BGLOAD 寄存器和 TIMERx_LOAD 寄存器都写入值，则在被 TIMCLKENx 使能的 TIMCLK 的下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 TIMERx_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

当定时器处于 16bit 模式时，32bit 的 TIMERx_VALUE 寄存器的高 16bit 并未被自动设为 0。若该定时器以前处于 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_CONTROL 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: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RW	timeren	定时器使能。 0: Timer 禁止; 1: Timer 使能。	0x0
[6]	RW	timermode	定时器的计数模式。 0: 自由运行模式; 1: 周期模式。	0x0
[5]	RW	intenable	<code>TIMERx_RIS</code> 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。	0x0
[4]	-	reserved	保留。	0x0
[3:2]	RW	timerpre	该字段用于设置 Timer 的预分频因子。 00: 不经过预分频, 时钟频率除以 1; 01: 4 级预分频, 将 Timer 时钟频率除以 16; 10: 8 级预分频, 将 Timer 时钟频率除以 256; 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 计数器操作模式。 0: 16bit 计数器; 1: 32bit 计数器。	0x0
[0]	RW	oneshot	选择计数模式为单次计数模式还是周期计数模式。 0: 周期计数模式或自由运行模式; 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的中断状态。



注意

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_RIS

TIMERx_RIS 为原始中断寄存器。

Offset Address: 0x010 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留。写入无效，读时返回 0。	0x00000000
[0]	RO	timer0ris	Timer0 的原始中断状态。 0: 未产生中断； 1: 已产生中断。	0x0

TIMERx_MIS

TIMERx_MIS 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。	0x00000000
[0]	RO	timer0mis	屏蔽后的 Timer0 的中断状态。 0: 中断无效； 1: 中断有效。	0x0

TIMERx_BGLOAD

TIMERx_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。

该寄存器提供了访问 [TIMERx_LOAD](#) 寄存器的另一种方法。不同之处在于写入值到 TIMERx_BGLOAD 寄存器中不会导致定时器立即从新写入值开始计数。

Offset Address: 0x018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	timer0bgload	Timer0 的计数初值。 注意：与 TIMERx_LOAD 寄存器有区别，具体请参见 TIMERx_LOAD 寄存器的描述。	0x00000000

3.8 看门狗

3.8.1 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 2 个 WatchDog 模块。

3.8.2 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。

3.8.3 功能描述

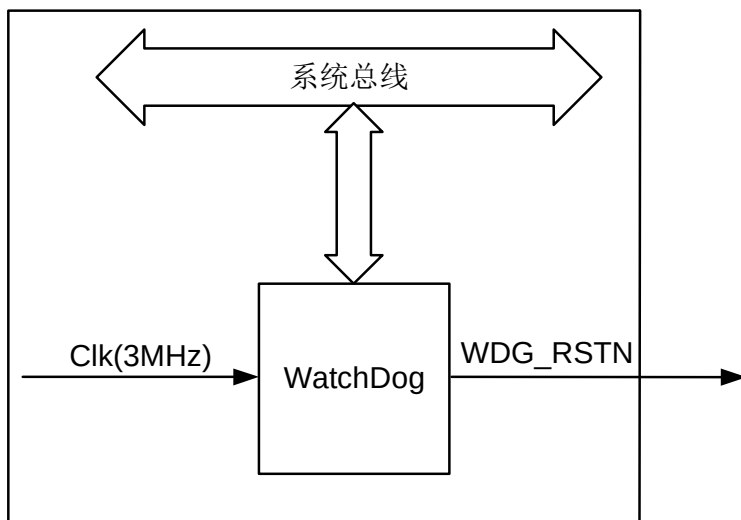
应用框图

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 3-5](#) 所示。



图3-5 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 **WDG_LOAD** 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 **WDG_LOAD** 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。

根据实际应用需要，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。

WatchDog 的计数时钟可以选择晶振时钟或者总线时钟，便于选择不同的计数时间范围。

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。



3.8.4 工作方式

计数时钟频率配置

WatchDog 计数时钟为 3MHz 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，各参数代表的意义分别是：

- 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~1400s。

系统初始化配置

系统上电复位后 WatchDog 计数器处于停止计数状态，在系统初始化过程中需要将 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。

3.8.5 WDG 寄存器概览

每个 WatchDog 模块各自有一组寄存器，每个 WatchDog 的寄存器除基址不相同外其他特性都相同。其中：

- WatchDog 0 基址：0x1205_0000。
- WatchDog 1 基址：0x1205_1000。

WatchDog 寄存器概览如表 3-10 所示。

表3-10 WatchDog 寄存器概览

偏移地址	名称	描述	页码
0x0000	WDG_LOAD	计数初值寄存器	3-143
0x0004	WDG_VALUE	计数器当前值寄存器	3-143
0x0008	WDG_CONTROL	控制寄存器	3-144
0x000C	WDG_INTCLR	中断清除寄存器	3-144
0x0010	WDG_RIS	原始中断寄存器	3-144
0x0014	WDG_MIS	屏蔽后中断寄存器	3-145
0x0C00	WDG_LOCK	LOCK 寄存器	3-145

3.8.6 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。用来读出 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	wdogvalue	WatchDog 计数器当前值。	0xFFFFFFFF

WDG_CONTROL

WDG_CONTROL 为控制寄存器。用来控制 WatchDog 的打开/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。 0: 禁止; 1: 使能。	0x0
[0]	RW	inten	WatchDog 中断信号输出使能。 0: 计数器停止计数, 计数值保持当前值不变, WatchDog 被关闭; 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	wdogris	WatchDog 原始中断状态，当计数器的计数值递减到 0 时，该位置 1。 0: 未产生中断； 1: 已产生中断。	0x0

WDG_MIS

WDG_MIS 为屏蔽后中断寄存器。用来反映屏蔽后的 WatchDog 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	wdogmis	WatchDog 屏蔽后的中断状态。 0: 未产生中断或者中断被屏蔽； 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	向该寄存器写入 0x1ACC_E551，可打开所有寄存器的写权限； 向该寄存器写入其他值则关闭写权限。 读该寄存器返回加锁的状态而不是写入该寄存器的值： 0x0000_0000: 允许写访问（未加锁）； 0x0000_0001: 禁止写访问（已加锁）。	0x00000000



3.9 实时时钟

3.9.1 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。

3.9.2 特点

RTC 具备以下特点：

- 内部具有 1 个 16bit 的天计数器，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。
- 计数时钟 100Hz
- 计数初值可配置
- 计数比较值可配置
- 支持超时中断产生
- 支持软复位
- 支持分频参数可配置
- 64bit 用户寄存器，供用户保存数据
- 电池低压检测

3.9.3 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。

3.9.4 工作方式

3.9.4.1 计数时钟频率

RTC 采用 100Hz 时钟进行计数，计数最大时间为：

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



T_{RTC} 表示 RTC 计数时间。

3.9.4.2 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。

软复位步骤如下：

步骤 1. 向 `RTC_POR_N` 写 0，对 RTC 软复位。

步骤 2. 等待 30ms。

----结束

3.9.4.3 RTC 初始化

RTC 在第一次上电时，系统需要将 RTC 初始化。RTC 的初始化过程如下：

步骤 1. 配置 `RTC_POR_N`，复位 RTC。

步骤 2. 等待 30ms。

步骤 3. 配置 `RTC_IMSC`，设置 RTC 中断屏蔽位。

步骤 4. 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 5. 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 6. 配置 `RTC_LORD` 为 1，让 RTC 计数初始值加载 RTC 计数器中。

步骤 7. 等待 10ms。

步骤 8. 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

3.9.4.4 RTC 时间配置

RTC 时间配置过程如下：

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 1，让 RTC 计数初始值加载 RTC 计数器中。

步骤 3. 等待 10ms。



----结束

3.9.4.5 RTC 时间读取

RTC 时间读取过程如下：

- 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. 等待 1ms。
- 步骤 4. 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

3.9.4.6 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_D_L`、`RTC_MR_D_H` 写入新的比较值。

----结束

3.9.4.7 RTC 寄存器的访问

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 4. 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 5. 读 **SPI_RW**，直到 **SPI_RW** [31]=0 为止。

步骤 6. 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 7. 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

3.9.5 RTC 寄存器概览

RTC 寄存器概览如表 3-11 所示。

表3-11 RTC 寄存器概览（基址是 0x1208_0000）

偏移地址	名称	描述	页码
0x0000	SPI_CLK_DIV	SPI 接口的时钟率值寄存器	3-150
0x0004	SPI_RW	SPI 接口读写寄存器	3-151

表3-12 RTC 内部寄存器概览（基址是 0x00）

偏移地址	名称	描述	页码
0x00	RTC_10MS_COUNT	RTC 10ms 计数值寄存器	3-152
0x01	RTC_S_COUNT	RTC 秒计数值寄存器	3-152
0x02	RTC_M_COUNT	RTC 分计数值寄存器	3-152
0x03	RTC_H_COUNT	RTC 时 计数值寄存器	3-152
0x04	RTC_D_COUNT_L	RTC 天计数值低 8 位寄存器	3-153
0x05	RTC_D_COUNT_H	RTC 天计数值高 8 位寄存器	3-153
0x06	RTC_MR_10MS	RTC 10ms 定时值寄存器	3-153
0x07	RTC_MR_S	RTC 秒定时值寄存器	3-154
0x08	RTC_MR_M	RTC 分定时值寄存器	3-154
0x09	RTC_MR_H	RTC 时定时值寄存器	3-154
0x0A	RTC_MR_D_L	RTC 天定时值的低 8 位寄存器	3-154
0x0B	RTC_MR_D_H	RTC 天定时值的高 8 位寄存器	3-155
0x0C	RTC_LR_10MS	RTC 10ms 设置值寄存器	3-155
0x0D	RTC_LR_S	RTC 秒设置值寄存器	3-155



偏移地址	名称	描述	页码
0x0E	RTC_LR_M	RTC 分设置值寄存器	3-156
0x0F	RTC_LR_H	RTC 时设置值寄存器	3-156
0x10	RTC_LR_D_L	RTC 天设置值的低 8 位寄存器	3-156
0x11	RTC_LR_D_H	RTC 天设置值的高 8 位寄存器	3-156
0x12	RTC_LORD	RTC 设置值使能加载寄存器	3-157
0x13	RTC_IMSC	RTC 中断使能寄存器	3-157
0x14	RTC_INT_CLR	RTC 中断清除寄存器	3-158
0x15	RTC_MSC_INT	RTC mask 中断状态寄存器	3-158
0x16	RTC_RAW_INT	RTC 原始中断状态寄存器	3-158
0x17	RTC_CLK	RTC 输出时钟选择寄存器	3-159
0x18	RTC_POR_N	RTC 复位控制寄存器	3-159
0x1A	RTC_UV_CTRL	RTC 内部低压检测控制寄存器	3-159
0x51	SDM_COEF_OUSID E_H	外部分频系数高 4 位寄存器	3-160
0x52	SDM_COEF_OUSID E_L	外部分频系数低 8 位寄存器	3-160
0x53	USER_REGISTER1	64bit 用户使用寄存器 1	3-161
0x54	USER_REGISTER2	64bit 用户使用寄存器 2	3-161
0x55	USER_REGISTER3	64bit 用户使用寄存器 3	3-161
0x56	USER_REGISTER4	64bit 用户使用寄存器 4	3-161
0x57	USER_REGISTER5	64bit 用户使用寄存器 5	3-161
0x58	USER_REGISTER6	64bit 用户使用寄存器 6	3-162
0x59	USER_REGISTER7	64bit 用户使用寄存器 7	3-162
0x5A	USER_REGISTER8	64bit 用户使用寄存器 8	3-162

3.9.6 RTC 寄存器描述

SPI_CLK_DIV

SPI_CLK_DIV 为 SPI 接口的时钟率值寄存器。

Offset Address: 0x0000 Total Reset Value: 0x0000_003B



Bits	Access	Name	Description	Reset
[31:8]	RO	reserved	保留。	0x000000
[7:0]	RW	spi_clk_div	SPI 接口的时钟分频系数。SPI 时钟不得超过 5MHz，建议配置为 5MHz。 取值范围为 1~255。spi_clk_div 的值用来计算 SPI 发送和接收的时钟频率，公式为 $FSPICLK = FAPBCLK / (2 \times (spi\_clk\_div + 1))$。其中 FAPBCLK 为总线的时钟频率。例如，总线时钟为 50MHz，期望 SPI 的时钟为 5MHz，则 spi_clk_div 应该配置为 $(50MHz/5MHz)/2 - 1 = 4$	0x3B

SPI_RW

SPI_RW 为 SPI 接口读写寄存器。

Offset Address: 0x0004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RO	spi_busy	SPI 接口读写忙状态指示。 0:空闲，可以发动新的 SPI 读写操作； 1:正在对 SPI 接口进行读写操作，不能发起新的 SPI 读写操作。	0x0
[30:25]	RO	reserved	保留。	0x00
[24]	W1_PULSE	spi_start	启动一次 SPI 读写操作。写 1 自动清零。 当 spi_busy 为 1 时写入无效。即在上一次读写操作没有完成之前不会启动新的 SPI 操作，若启动则硬件会忽略这一次启动请求。	0x0
[23]	RW	spi_rw	SPI 操作的类型。 0:写操作； 1:读操作。	0x0
[22:16]	RW	spi_add	SPI 操作的地址。 取值范围为 0~127。	0x00
[15:8]	RO	spi_rdata	SPI 接口读回的数据。	0x00
[7:0]	RW	spi_wdata	待写入 SPI 接口的数据。	0x00



3.9.7 RTC 内部寄存器描述

RTC_10MS_COUNT

RTC_10MS_COUNT 为 RTC 10ms 计数值寄存器。

Offset Address: 0x00 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: 0x01 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: 0x02 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。取之范围为 0~59。	0x00

RTC_H_COUNT

RTC_H_COUNT 为 RTC 时 计数值寄存器。



Offset Address: 0x03 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。 取之范围为 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: 0x04 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: 0x05 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: 0x06 Total Reset Value: 0x7F

Bits	Access	Name	Description	Reset
[7]	-	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: 0x07 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。 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: 0x08 Total Reset Value: 0x3F

Bits	Access	Name	Description	Reset
[7:6]	RO	reserved	保留	0x0
[5:0]	RW	rtc_mr_m	RTC 分定时值，表示定时的分时间值。 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: 0x09 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。 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: 0x0A 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 配合，表示定时的天时间值，取值范围为 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: 0x0B 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 配合，表示定时的天时间值，取值范围为 0~65535。	0xFF

RTC_LR_10MS

RTC_LR_10MS 为 RTC 10ms 设置值寄存器。

Offset Address: 0x0C 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。 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: 0x0D 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。 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: 0x0E 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。 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: 0x0F 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。 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: 0x10 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: 0x11 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~	0x00



			65535。	

RTC_LORD

RTC_LORD 为 RTC 设置值使能加载寄存器。

Offset Address: 0x12 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。 0:使能, RTC 计数值(0x00~0x05)只会在锁存成功后才更新。 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, 硬件会在锁存成功后自动清零。 注意:该寄存器在 rtc_lock_bypass 为 0 时才有效。	0x0
[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: 0x13 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。 0:不输出中断; 1:输出中断。	0x0
[1]	RW	rtc_imsc_uv	电池低压检测中断输出使能位。 0:不输出中断; 1:输出中断。	0x0
[0]	RW	rtc_imsc_time	RTC 定时中断输出使能位。 0:不输出中断;	0x0



			1:输出中断。	

RTC_INT_CLR

RTC_INT_CLR 为 RTC 中断清除寄存器。

Offset Address: 0x14 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: 0x15 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。 0:无中断; 1:有中断。	0x0
[0]	RO	mask_int_time	屏蔽后的 RTC 定时中断状态寄存器。 0:无中断; 1:有中断。	0x0

RTC_RAW_INT

RTC_RAW_INT 为 RTC 原始中断状态寄存器。

Offset Address: 0x16 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。 0:无中断; 1:有中断。	0x0



[0]	RO	raw_int_time	RTC 定时原始中断状态寄存器。 0:无中断; 1:有中断。	0x0

RTC_CLK

RTC_CLK 为 RTC 输出时钟选择寄存器。

Offset Address: 0x17 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。 00: 输出晶体的震荡时钟; 01: 输出校正后的 100Hz 时钟; 1X: 输出 1Hz 时钟。	0x0

RTC_POR_N

RTC_POR_N 为 RTC 复位控制寄存器。

Offset Address: 0x18 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。 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: 0x1A Total Reset Value: 0x00

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。 0:关闭;	0x0



Bits	Access	Name	Description	Reset
			1:打开。	
[4]	RW	bat_uv_ctrl_sel	低压检测来源选择。 0:不经滤波处理; 1:经过滤波处理。	0x0
[3:2]	RO	reserved	保留。	0x0
[1:0]	RW	sample_time	低压检测周期。 00: 1 秒; 01: 1 分钟; 10: 10 分钟; 11: 30 分钟。	0x0

SDM_COEF_OUSIDE_H

SDM_COEF_OUSIDE_H 为外部分频系数高 4 位寄存器。

Offset Address: 0x51 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。	0x8

SDM_COEF_OUSIDE_L

SDM_COEF_OUSIDE_L 为外部分频系数低 8 位寄存器。

Offset Address: 0x52 Total Reset Value: 0x1B

Bits	Access	Name	Description	Reset
[7:0]	RW	sdm_coef_ouside_l	固定分频模式时的分频系数低 8 位。 注意：对分频系数寄存器(0x51、0x52)进行读/写时，应该先读/写高 4 位，再读/写低 8 位，并且需要高位到低位连续操作才能完成读/写操作。该寄存器的计算公式为 $(f-32700)*30.52$ ($32700 \leq f \leq 32799$ ，f 为外接晶体的振荡频率)。	0x1B



USER_REGISTER1

USER_REGISTER1 为 64bit 用户使用寄存器 1。

Offset Address: 0x53 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: 0x54 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: 0x55 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: 0x56 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: 0x57 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: 0x58 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: 0x59 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: 0x5A 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

3.10 电源管理与低功耗模式

3.10.1 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。

- 支持 AVS (Adaptive Voltage Scaling) 功能

3.10.2 PMC

3.10.2.1 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
- 下电顺序可配置
- 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 秒后触发下电时序



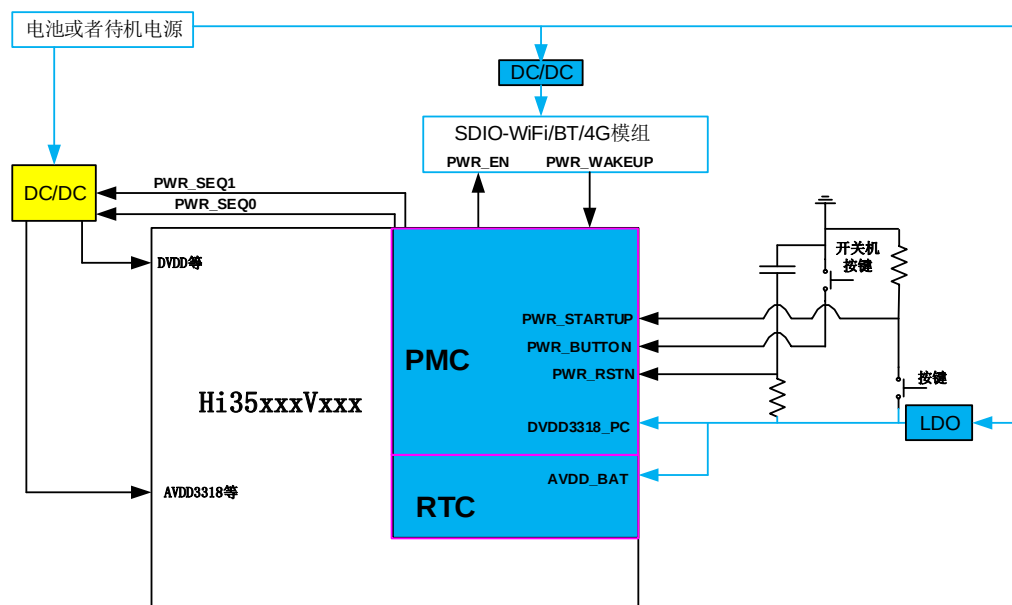
注意

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 3-6 所示。其中 PMC 为恒电区，PWR_SEQ0/1 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。

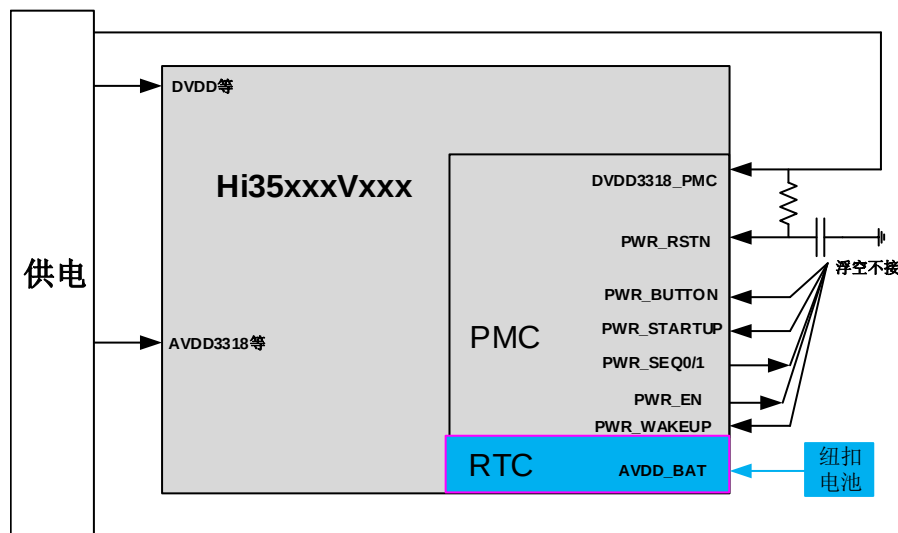


图3-6 使用 PMC 时的电源网络粗略框图



当不使用 PMC 功能时，相关管脚的处理如图 3-7 所示，其中 PWR_RSTN 接 RC 复位，其它管脚浮空处理。

图3-7 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

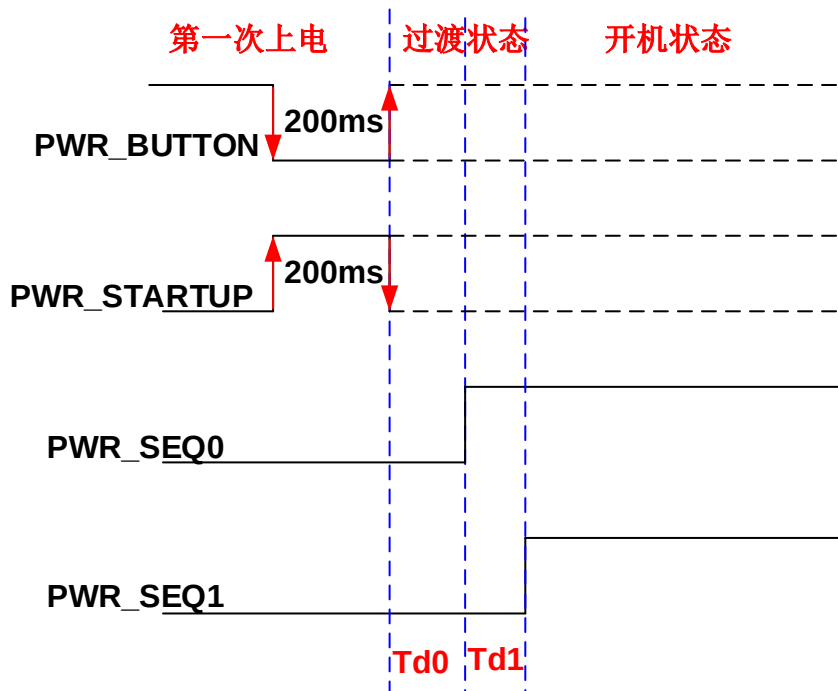
本章节中第一次上电状态、开机状态、关机状态、待机状态名词说明：



- 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_SEQ0/1 均为 0；
- 开机状态：整体芯片均上电，PWR_SEQ0/1 均为 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_SEQ0/1 均为 0，在此状态下可以由 PWR_BUTTON、PWR_STARTUP、PWR_WAKEUP、RTC 唤醒。

当在第一次上电时，PWR_BUTTON 保持 200ms 的低电平后或者 PWR_STARTUP 保持 200ms 的高电平后可触发上电时序。如图 3-8 所示。

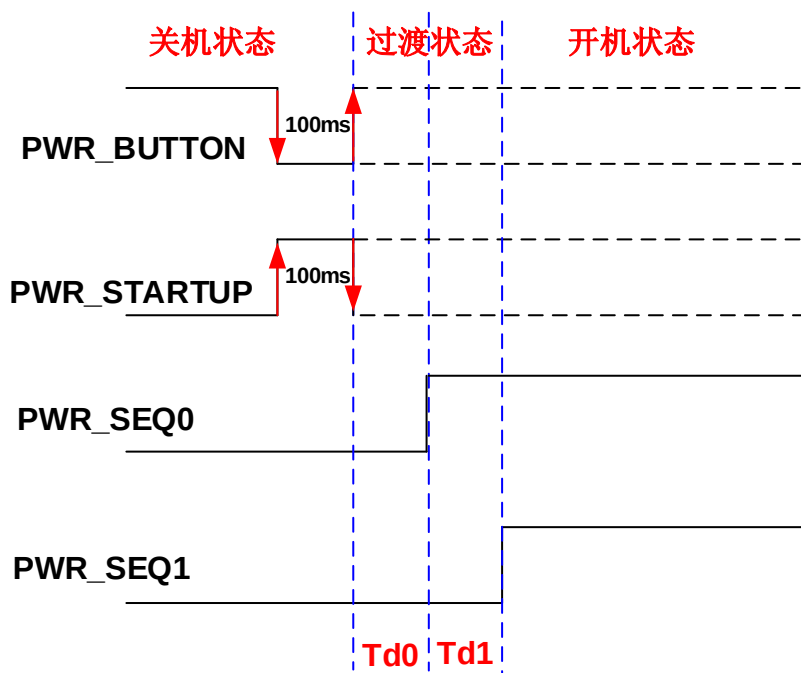
图3-8 PWR_BUTTON、PWR_STARTUP 第一次上电时序



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 3-9 所示（以 100ms 为例）。

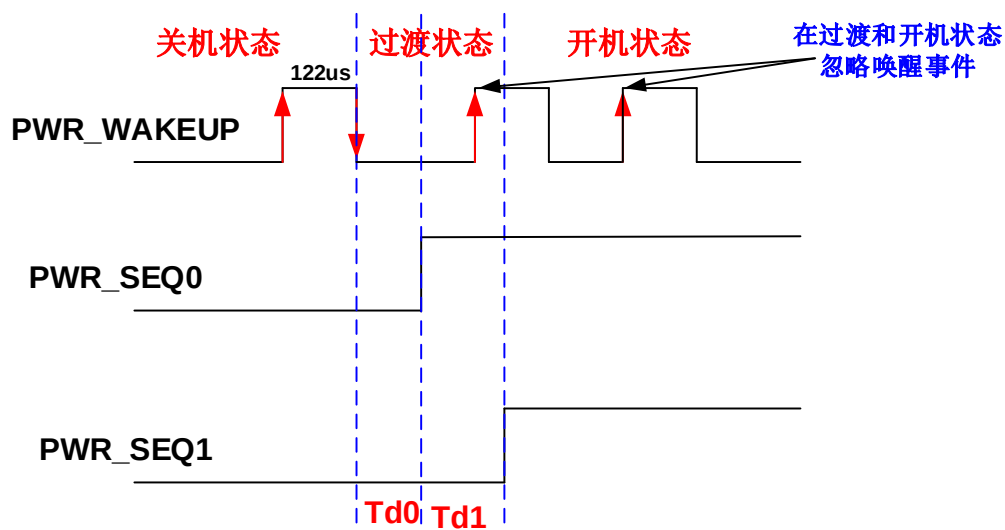


图3-9 PWR_BUTTON、PWR_STARTUP 上电时序



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，可以配置为上升沿、下降沿、高电平和低电平触发。如图 3-10 所示（以上升沿触发为例）。

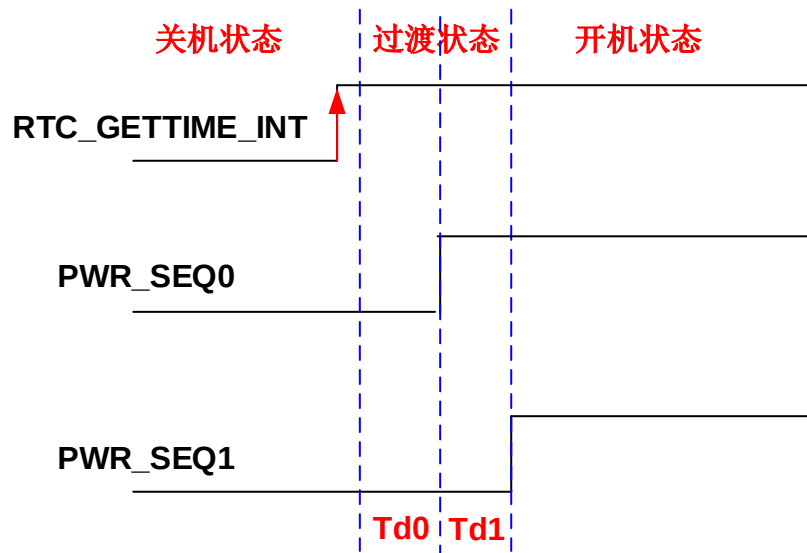
图3-10 PWR_WAKEUP 上电时序（以上升沿触发为例）





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 3-11 所示。

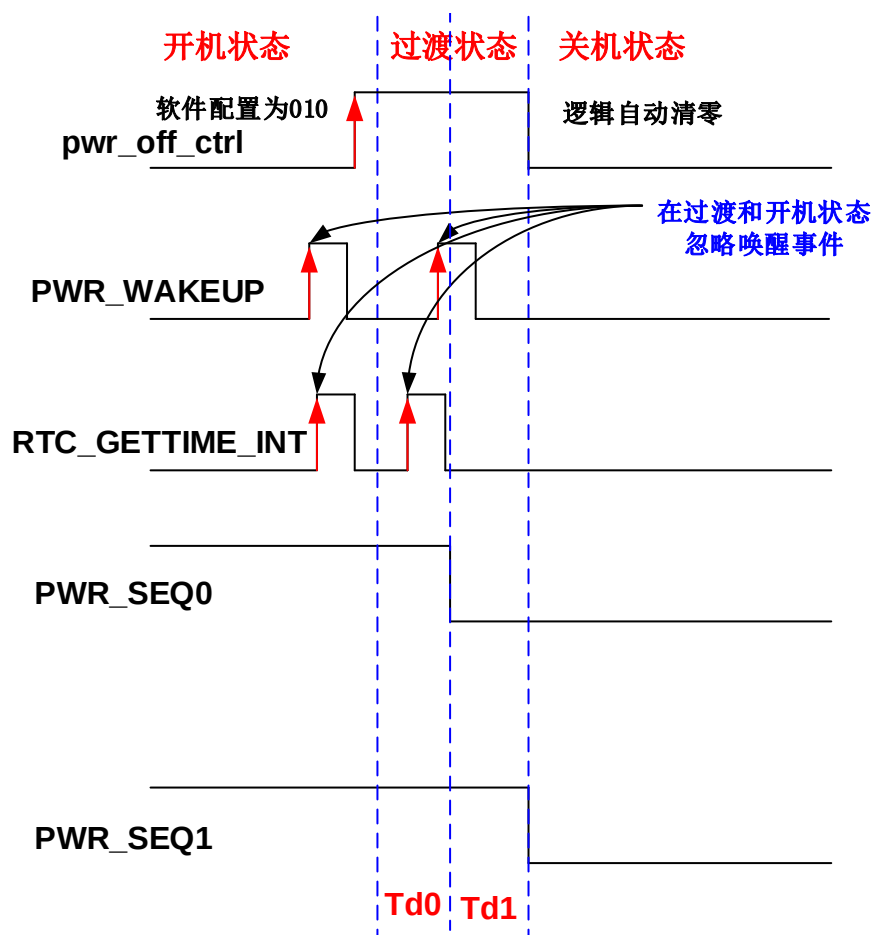
图3-11 RTC 定时中断上电时序



在开机状态下，通过配置寄存器 `PWR_CTRL0[pwr_off_ctrl]` 为 010 即可触发下电时序。如图 3-12 所示(以下电顺序模式为 0 为例)。



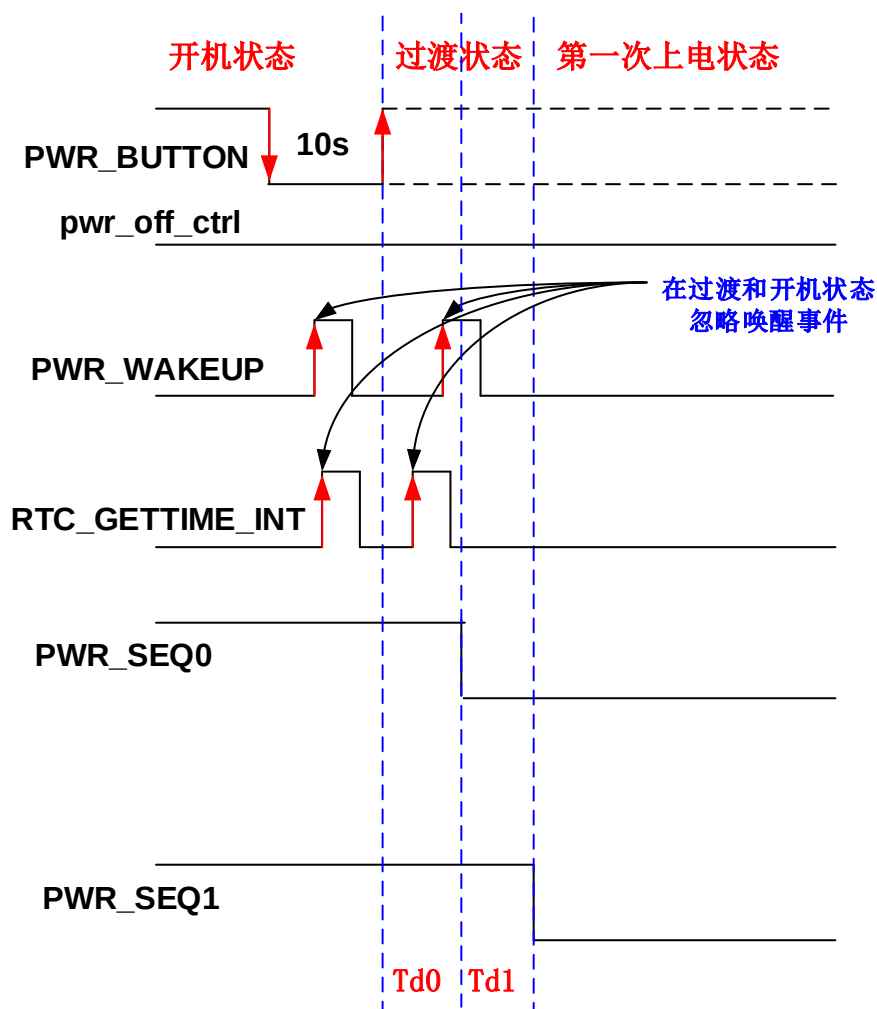
图3-12 软件配置下电时序(以下电模式为 0 为例)



在开机状态下，若软件出现异常，可以通过 PWR_BUTTON 保持低电平 10s 触发下电时序，如图 3-13 所示(以下电顺序模式为 0 为例)。



图3-13 异常下电时序(以下电顺序模式为 0 为例)



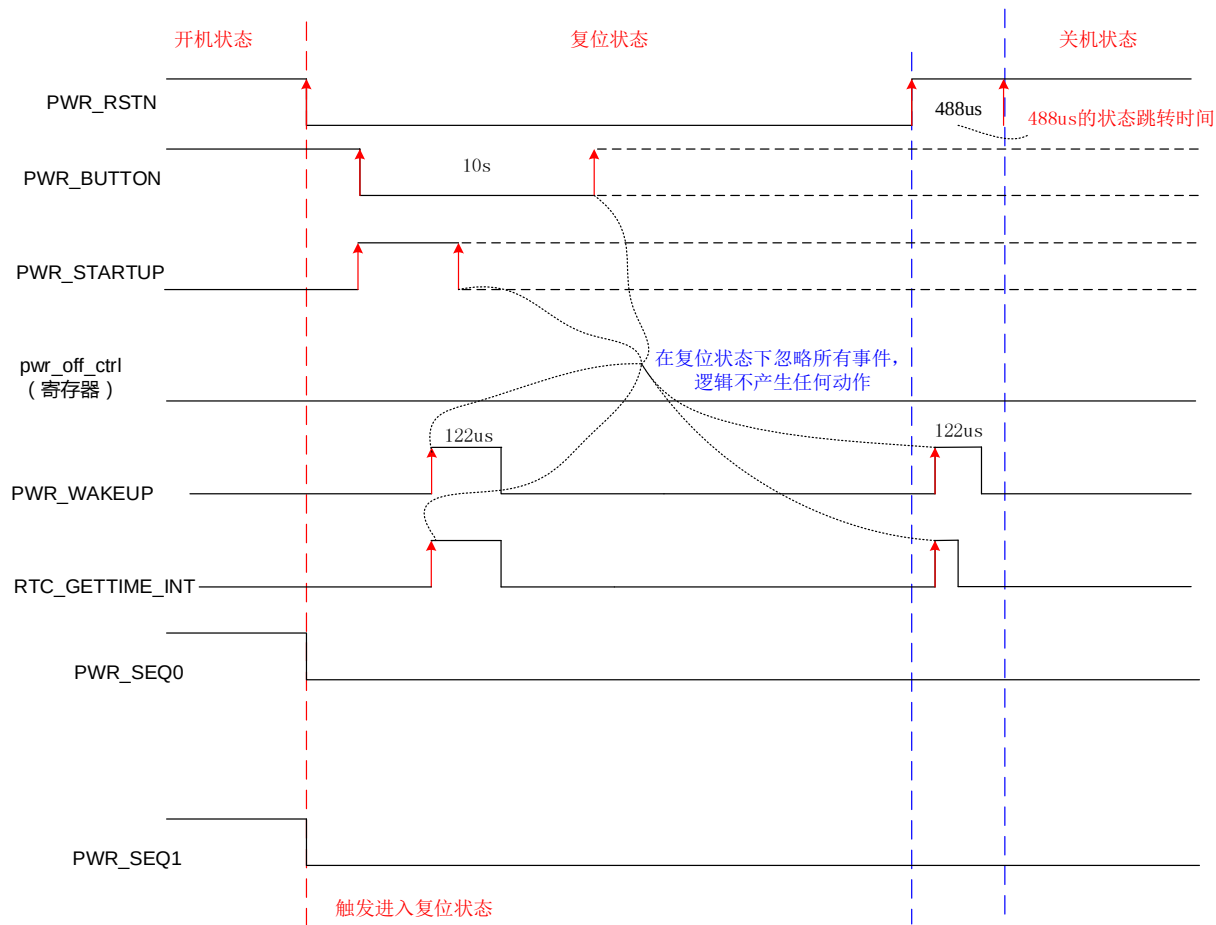
Td1 是可以编程的值，对应的寄存器为 PWR_CTRL1[pwr_td1]，取值范围为 1~255，计时单位为 ms，复位默认为 16。

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_SEQ0/1 输出为 0，如图 3-14 所示。



图3-14 复位状态示意图



注意

- 唤醒源 PWR_WAKEUP、PWR_STARTUP、PWR_BUTTON 只有在开机状态或者待机状态才会产生中断。
- 通过异常进入第一次上电状态时将复位 PWR_CTRL0/1/2/3/4/5/6 寄存器，其余寄存器不复位。
- PWR_STARTUP 除无长按 10s 异常关机、检测上升沿外，其余功能与 PWR_BUTTON 相同，即可以在关机下启动，在开机与待机下产生中断。

上下电时序的电源的分组，根据第二章“硬件”中的上下电时序要求，上电时先高压后低压，下电时先高压后低压(即下电顺序模式为 0)。相关电源管脚分组如表 3-13 所示。

表3-13 电源对接表

上下电时序分组	电源管脚
由 PWR_SEQ0 控制	AVDD33_AC



上下电时序分组	电源管脚
	AVDD33_DDR_PLL
	AVDD33_HDMITX
	AVDD33_PLL
	AVDD33_USB
	AVDD3318_MIPIRX
	AVDD3318_MIPITX
	DVDD18_SDIO0
	DVDD33
	DVDD3318_FLASH
	DVDD3318_SARADC
	DVDD3318_SDIO1
	DVDD3318_SENSOR
	DVDD3318_UART1
	DVDD3318_VI
	VDDIO_DDR
	VDDIO_DDR_CK
由 PWR_SEQ1 控制	VDD
	AVDD09_HDMITX
	AVDD09_PLL

3.10.2.2 PMC 寄存器概览

PMC 寄存器概览如表 3-14 所示。

表3-14 PMC 寄存器概览（基址是 0x1209_0000）

偏移地址	名称	描述	页码
0x000	PWR_CTRL0	PWR 控制寄存器 0	3-172
0x004	PWR_CTRL1	PWR 控制寄存器 1	3-173
0x008	PWR_STATUS	PWR 状态寄存器	3-173
0x00C	PWR_PD_SEL	PWR 下电时序控制寄存器	3-174



3.10.2.3 PMC 寄存器描述

PWR_CTRL0

PWR_CTRL0 为 PWR 控制寄存器 0。

Offset Address: 0x000 Total Reset Value: 0x0001_0160

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	保留。	0x0
[28]	RW	high_adj_startup_en	startup 管脚生效时间修改控制。 1: 固定 200ms; 0: 受 high_adj_startup_time 寄存器控制。	0x0
[27:20]	RW	high_adj_startup_time	startup 管脚生效时间修改值。单位为 100ms。生效值为配置值+1。	0x01
[19:17]	-	reserved	保留。	0x0
[16]	RW	high_adj_button_en	button 管脚生效时间修改控制。 1: 固定 200ms; 0: 受 high_adj_button_time 寄存器控制。	0x0
[15:8]	RW	high_adj_button_time	button 管脚生效时间修改值。单位为 100ms。生效值为配置值+1。	0x01
[7]	RW	gpio_at_pmc_enable	PMC 域上的 GPIO 管脚复用使能。 0: 不使能; 1: 使能。	0x0
[6]	RW	wakeup_mask	是否屏蔽 wakeup 唤醒。 0: 不屏蔽; 1: 屏蔽。	0x1
[5]	RW	rtc_int_mask	是否屏蔽 RTC 中断唤醒。 0: 不屏蔽; 1: 屏蔽。	0x1
[4:3]	RW	wakeup_act_mode	WAKEUP 有效模式选择。 00: 上升沿有效; 01: 下降沿有效; 10: 高电平有效; 11: 低电平有效。	0x0
[2]	RW	pwr_pen	pwr_pen0 输出控制。 0: 输出 0;	0x0



Bits	Access	Name	Description	Reset
			1: 输出 1。	
[1]	WO	start_powerdown	开始关机流程。	0x0
[0]	-	reserved	保留。	0x0

PWR_CTRL1

PWR_CTRL1 为 PWR 控制寄存器 1。

Offset Address: 0x004 Total Reset Value: 0x1010_1010

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

PWR_STATUS

PWR_STATUS 为 PWR 状态寄存器。

Offset Address: 0x008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:9]	-	reserved	保留。	0x000000
[8]	RO	wakeup_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。 bit0: WAKEUP; bit1: reserved; bit2: STARTUP; bit3: BUTTON; bit4: RTC。	0x00



[1]	-	reserved	保留。	0x0
[0]	-	reserved	保留。	0x0

PWR_PD_SEL

PWR_PD_SEL 为 PWR 下电时序控制寄存器。

Offset Address: 0x00C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	pd_seq_mode_sel	下电时序选择控制。 0: 模式 1, 下电时序为 SEQ0->SEQ1; 1: 模式 2, 下电时序为 SEQ1->SEQ0; 其他: 保留。	0x0

3.10.3 时钟门控和时钟频率调整

系统提供各个模块的时钟门控功能, 在模块空闲的时候, 可以关闭相应的时钟, 降低芯片功耗, 操作流程可以参见各个模块的时钟门控部分。

调整这些模块的工作频率, 也可以进一步降低系统的功耗。具体参见“[3.2.3 时钟资源分布](#)”中关于各模块的时钟源选择部分。

3.10.4 模块级低功耗控制

芯片中的 USB 2.0 Host/Device、PLL 等 PHY 模块都可以提供低功耗的工作模式, 可通过对应的 CRG 控制寄存器, 关闭对应模块的时钟, 不使用的 PLL 可通过对应的寄存器关闭, 以达到降低功耗的目的, 请参考“[3.2 时钟](#)”章节的寄存器描述。

3.10.5 AVS 功能说明

AVS 是根据芯片工艺、温度、电路时序情况, 实时动态调整电压来进一步降低功耗。系统可由 HPM 获取芯片的 PVT 信息, 并通过 SVB PWM 输出信号实现电压的自动调整。

AVS 提供芯片功耗管理的基础信息和控制, 主要有以下几项。

- SVB (Selective Voltage Binning) PWM
外围 PWM 调压电路可通过芯片的 SVB_PWM 输出, 调节芯片的核心供电电压以实现降低功耗。
- HPM 控制
通过 HPM 获取当前芯片的 PVT 综合信息, 根据这些信息产生芯片供电电压调整的决策参考。



- 芯片内部温度检测

芯片集成温度传感器，用于芯片内部温度的获取，为芯片的温度保护处理提供温度信息。

3.10.5.1 SVB 电压调节

SVB PWM 模块工作时钟为 24MHz，周期和占空比计数器为 10bits。通过配置 SVB PWM 相关的寄存器（MISC 寄存器 [MISC_CTRL44](#)），可实现指定周期和占空比的 PWM 输出。

配置 SVB PWM 输出的步骤为：

- 步骤 1. 根据所需的 PWM 输出频率(freq)和占空比(duty)，计算出对应的 PWM 循环周期数和高电平计数值。

循环周期数计算公式为： $\text{Period}_{\text{svb}} = (24000000 \div \text{freq}) - 1$

高电平周期数计算公式为： $\text{Duty}_{\text{svb}} = (24000000 \div \text{freq}) \times \text{duty} - 1$

- 步骤 2. 配置 PWM 循环周期数（[MISC_CTRL44](#)[13:4]）和高电平周期数寄存器（[MISC_CTRL44](#)[25:16]）。

- 步骤 3. 配置 [MISC_CTRL44](#)[2]为 1，加载配置的参数，使配置的参数生效。

- 步骤 4. 配置 [MISC_CTRL44](#)[0]为 1，使能 PWM 输出。

----结束

例如，芯片 CORE 电源调压控制由 SVB_PWM 信号控制，如果需要 SVB_PWM 输出频率为 200KHz，占空比为 75%，配置步骤如下：

- 步骤 5. 计算出所需的计算周期数和高电平计数值：

$\text{Period}_{\text{svb}} = (24000000 / 200000) - 1 = 119$ (16 进制值为 0x77)

$\text{Duty}_{\text{svb}} = (24000000 / 200000) * 0.75 - 1 = 89$ (16 进制值为 0x59)

- 步骤 6. 设置 SVB PWM 循环周期数为 0x77，高电平周期数为 0x59；

- 步骤 7. 设置 SVB PWM 加载控制。

- 步骤 8. 使能 SVB PWM。

----结束

3.10.5.2 芯片内部温度检测

芯片内部集成温度传感器(T-Sensor)，温度的检测范围为-40~125℃。使能温度传感器数据采集的流程如下：

T-sensor 相关的寄存器为 MISC 寄存器 [MISC_CTRL45~MISC_CTRL50](#)。

- 步骤 1. 设置 T-Sensor 采集模式 [MISC_CTRL45](#)[30]。



- 步骤 2. 如果设置为循环采集模式，需设置循环采集周期 `MISC_CTRL45[27:20]`；如果设置为单次采集模式，可略过此步骤。

循环采集周期计算公式为： $T = N \times 2 (ms)$

其中 $N = \text{MISC\_CTRL45}[27:20]$ 。

- 步骤 3. 使能 T-Sensor `MISC_CTRL45[31]`，开始温度采集。

- 步骤 4. 软件读取 T-Sensor 采集到的温度码（十六进制值）。

单次采集模式下，只有 `MISC_CTRL47[9:0]` 中记录的温度记录码 0 有效。

循环采集模式下，`MISC_CTRL47[31:0]~MISC_CTRL50[31:0]` 记录了最近八次温度记录码 0~7，其中最新的温度记录值为温度记录码 0。

- 步骤 5. 根据温度记录码计算出对应的温度值。

$$Temperature = (tsensor\_result - 136) / 793 \times 165 - 40 (^\circ C)$$

注：tsensor_result 是步骤 4 中获取的温度记录码的十进制值。

----结束



目 录

4 存储器接口	4-1
4.1 DDR 控制器	4-1
4.1.1 概述	4-1
4.1.2 特点	4-1
4.1.3 功能描述	4-1
4.1.4 工作方式	4-6
4.1.5 AXI 寄存器	4-7
4.1.6 TZASC 寄存器	4-24
4.1.7 QOSBUF 寄存器	4-36
4.1.8 DMC 寄存器	4-51
4.1.9 DDR PHY 寄存器	4-71
4.2 Flash Memory 控制器	4-77
4.2.1 概述	4-77
4.2.2 特点	4-77
4.2.3 功能描述	4-78
4.2.4 工作流程	4-85
4.2.5 数据结构(SPI NAND Flash)	4-90
4.2.6 ECC 模式选择说明	4-92
4.2.7 FMC 寄存器概览	4-93
4.2.8 FMC 寄存器描述	4-94



插图目录

图 4-1 DDRC 单通道模式下，通道与两片 16bit 位宽 DDR3(L)/DDR4 SDRAM 的互联示意图.....	4-2
图 4-2 Standard SPI 接口模式写操作时序图.....	4-78
图 4-3 Standard SPI 接口模式读操作时序图。.....	4-78
图 4-4 Dual-Output/Dual-Input SPI 接口时序图	4-79
图 4-5 Dual I/O SPI 模式 STR 时序图	4-79
图 4-6 Dual I/O SPI 模式 DTR 时序图.....	4-80
图 4-7 Quad-Output/Quad-Input SPI 模式时序图	4-80
图 4-8 Quad-I/O SPI 模式接口时序图.....	4-81
图 4-9 Quad-I/O SPI 模式 DTR 接口时序图.....	4-81
图 4-10 SPI 输出时序图.....	4-82
图 4-11 擦除操作流程图中.....	4-87
图 4-12 4bit ECC 模式 2K page_size 的数据结构	4-91
图 4-13 4bit ECC 模式 4K page_size 的数据结构	4-91
图 4-14 8bit ECC 模式 2K page_size 的数据结构	4-92
图 4-15 8bit ECC 模式 4K page_size 的数据结构	4-92
图 4-16 24bit ECC 模式 2K page-size 的数据结构.....	4-92
图 4-17 24bit ECC 模式 4K page_size 的数据结构	4-92



表格目录

表 4-1 DDR3 命令真值表	4-3
表 4-2 DDR4 命令真值表	4-3
表 4-3 AXI 寄存器偏移地址变量表	4-7
表 4-4 AXI 寄存器概览（基址是 0x1206_0000）	4-7
表 4-5 TZASC 寄存器偏移地址变量表	4-24
表 4-6 TZASC 寄存器概览（基址是 0x1206_1000）	4-24
表 4-7 QOSBUF 寄存器偏移地址变量表	4-36
表 4-8 QOSBUF 寄存器概览（基址是 0x1206_4000）	4-37
表 4-9 DMC 寄存器概览（DMC 基址是：0x1206_8000）	4-51
表 4-10 DDR PHY 寄存器概览（DDR PHY 基址是 0x1206_C000）	4-71
表 4-11 FLASH 地址分配表	4-82
表 4-12 非 ECC0 模式数据结构长度	4-90
表 4-13 FMC 寄存器概览（基址是 0x1000_0000）	4-93



4 存储器接口

4.1 DDR 控制器

4.1.1 概述

DDRC（DDR SDRAM Controller）实现对动态存储器 DDR3(L)/DDR4 SDRAM 的存取控制。

4.1.2 特点

DDRC 的功能特点：

- 支持最大存储空间为 2GB。
- 支持 DDR3(L)/DDR4 SDRAM 接口最高工作频率为 900MHz，即数据速率为 1.8Gbps。
- 支持 DDR3(L)/DDR4 SDRAM 的 Power Down 等低功耗模式。
- 支持单通道。

4.1.3 功能描述

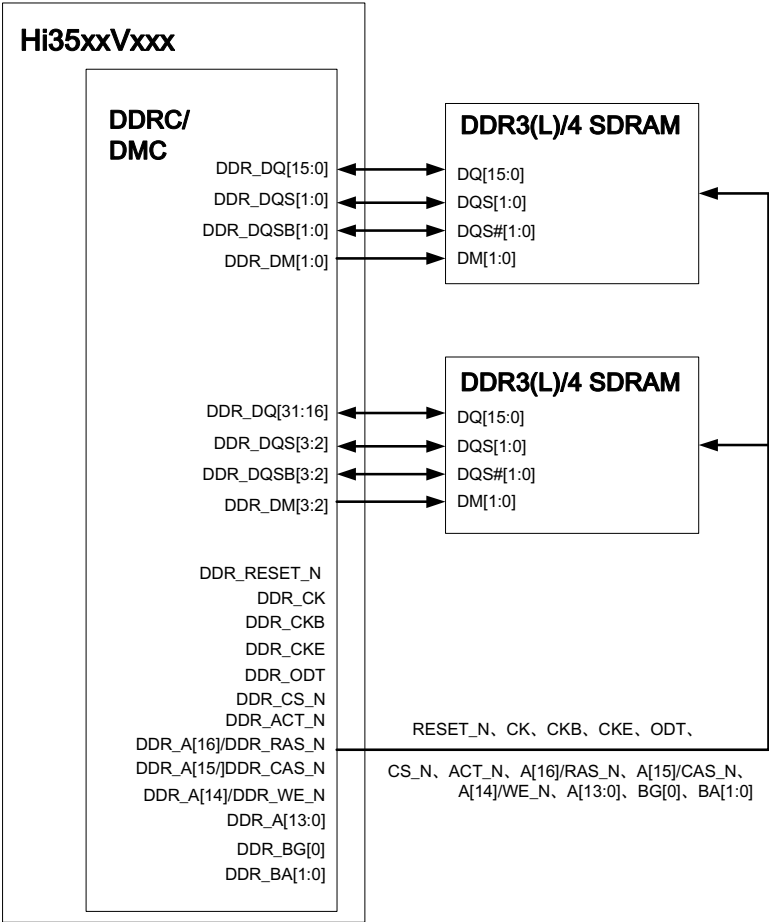
4.1.3.1 应用框图

DDRC 实现了 SoC 系统中的 CPU 等主设备对 DDR3(L)/DDR4 SDRAM 的数据访问。通过 CPU 配置 DDRC 的时序参数寄存器，可以支持符合 JEDEC 标准的 DDR3(L)/DDR4 SDRAM 器件。

对接 DDR3(L)/DDR4 时 DDRC 为单通道模式，通道支持 32bit 互联，连接示意图如图 4-1 所示。



图4-1 DDRC 单通道模式下，通道与两片 16bit 位宽 DDR3(L)/DDR4 SDRAM 的互联示意图



连接说明：
单通道模式下，32bit DDR3(L)/DDR4 SDRAM 由两片数据总线宽度为 16bit 的存储器件组成。
DDRC/DMC 对应的命令控制信号：DDR_CS_N、DDR_CK、DDR_CKB、DDR_ACT_N、DDR_CKE、DDR_RESET_N、DDR_A [16]/DDR_RAS_N、DDR_A [15]/DDR_CAS_N、DDR_A [14]/DDR_WE_N、DDR_A[13:0]、DDR_BG[0]、DDR_BA[1:0]，与 2 个 DDR3(L)/DDR4 SDRAM 颗粒的命令控制信号相连，DDRC 的命令控制总线是 1 驱 2 的连接模式。

4.1.3.2 功能原理

DDRC 接口时序满足 JEDEC 标准，通过发送 DDR3(L)/DDR4 SDRAM 的命令字，完成对 DDR3(L)/DDR4 SDRAM 的数据访问和状态控制。包括 DDR3(L)/DDR4 SDRAM 的读写访问、自动刷新、低功耗控制等功能。

命令真值表

DDRC 支持 DDR3(L)/DDR4 SDRAM 的读写和控制命令字。
DDR3 SDRAM 的命令真值表如表 4-1 所示。更加详细的信息请参阅 JEDEC 标准和器件手册。



表4-1 DDR3 命令真值表

FUNCTION	CKE	CSN	ACTN	RASN/ ADR16	CASN/ ADR15	WEN/ ADR14	ADDR			BA
							13:11	AP(10)	9:00	
DESELECT	H	H	X	X	X	X	X	X	X	X
ACTIVE	H	L	L	Row address			V	V	V	BA
READ	H	L	H	H	L	H	V	V	V	BA
WRITE	H	L	H	H	L	L	V	V	V	BA
PRECHARGE	H	L	H	L	H	L	V	L	V	BA
PRECHARGE ALL	H	L	H	L	H	L	V	H	V	V
AUTO REFRESH	H	L	H	L	L	H	V	V	V	V
SELF REFRESH ENTRY	H- >L	L	H	L	L	H	V	V	V	V
SELF REFRESH EXIT	L- >H	L	H	H	H	H	V	V	V	V
MODE REGISTER SET	H	L	H	L	L	L	V	V	V	V
ZQCL	H	L	H	H	H	L	V	H	V	V
ZQCS	H	L	H	H	H	L	V	L	V	V

H: 表示高电平; L: 表示低电平; V: 表示有效; X: 表示不关心。

DDR4 SDRAM 的命令真值表如表 4-2 所示。更加详细的信息请参阅 JEDEC 标准和器件手册。

表4-2 DDR4 命令真值表

FUNCTION	CKE	CSN	ACTN	RASN /ADR 16	CASN /ADR 15	WEN/ ADR1 4	ADDR			BG	BA
							13:11	AP (10)	9:0		
DESELECT	H	H	X	X	X	X	X	X	X	X	X
ACTIVE	H	L	L	Row address			V	V	V	BG	BA
READ	H	L	H	H	L	H	V	V	V	BG	BA
WRITE	H	L	H	H	L	L	V	V	V	BG	BA



FUNCTION	CKE	CSN	ACTN	RASN /ADR 16	CASN /ADR 15	WEN/ ADR1 4	ADDR			BG	BA
							13:11	AP (10)	9:0		
PRECHARGE	H	L	H	L	H	L	V	L	V	BG	BA
PRECHARGE ALL	H	L	H	L	H	L	V	H	V	V	V
AUTO REFRESH	H	L	H	L	L	H	V	V	V	V	V
SELF REFRESH ENTRY	H->L	L	H	L	L	H	V	V	V	V	V
SELF REFRESH EXIT	L->H	L	H	H	H	H	V	V	V	V	V
MODE REGISTER SET	H	L	H	L	L	L	V	V	V	V	V
ZQCL	H	L	H	H	H	L	V	H	V	V	V
ZQCS	H	L	H	H	H	L	V	L	V	V	V

自动刷新

当配置 [DDRC_CFG_TIMING2](#) [taref] 为非零值，DDRC 自动产生周期性 AUTO REFRESH 命令，完成对 DDR3(L)/DDR4 SDRAM 的刷新操作。常温下，DDR3(L)/DDR4 SDRAM 自动刷新操作的周期为 7.8us。实际值以器件手册为准。[DDRC_CFG_TIMING2](#) [taref] 的配置值（taref）与自动刷新周期（T）的关系为例：

$$T \geq \text{taref} * (16 * \text{DDR 接口时钟周期} * 2)$$

在配置了 [DDRC_CFG_TIMING2](#) [taref]，DDRC 内部的计数器会自动加载 taref 值进行减计数，当计数器值为 0 时，DDRC 发起一次自动刷新操作，同时重新加载 taref 值进行计数。

不同温度下 SDRAM 器件对刷新周期要求可能不同，实际配置值请参见 SDRAM 器件手册。

低功耗管理

DDRC 支持两种模式的低功耗管理：普通低功耗模式和自刷新低功耗模式。

当配置 [DDRC_CFG_PD](#) [pd_en]=1，使能 SDRAM 自动低功耗后，系统处于空闲状态时（DDRC 总线接口一定时间内无读写 DDR 访问），自动控制 DDR3(L)/DDR4 SDRAM 进入到普通低功耗模式。



当系统需要进入到待机模式时，可通过配置寄存器 `DDRC_CTRL_SREF` [`sref_req`]=1，控制 DDR3(L)/DDR4 SDRAM 进入到自刷新低功耗模式。该模式下可以将 DDR3(L)/DDR4 SDRAM 的功耗降至最低，同时保持 DDR3(L)/DDR4 SDRAM 中的数据，但是此时系统不能访问 DDR3(L)/DDR4 SDRAM。

仲裁机制

DDRC 主要通过优先级调度算法对各命令进行调度。另外，DDRC 内还实现了 timeout 调度辅助手段对命令的请求进行控制。

1. 优先级调度

优先级等级为：0~7。每个 AXI 端口的优先级配置是独立的。

2. 支持优先级自适应

每个端口支持优先级自适应功能，以帮助低优先级命令通过各级仲裁：每当优先级递增周期到达后，自动将命令优先级提升一级，直到将优先级提至次最高。

3. 同一端口的读写命令将被汇聚成一路，支持两种可选的仲裁规则：

- 读写命令先到先仲裁，同时到达时先读后写。
- 根据读写命令的优先级高低仲裁。
 - 不同优先级的命令：仲裁给优先级最高的命令。
 - 同优先级的命令：仲裁给最久未获得过仲裁的命令。

4. timeout 控制

配置 `QOSB_TIMEOUT_MODE`[`timeout_mode`]，选择命令的 timeout 属性映射模式。类似于优先级的映射，timeout 映射方法：在所配置映射模式下，查询出预设的 timeout 值中的一个值，并将该值添加给该命令作为 timeout 值（0 值代表禁止 timeout）。DDRC 内部，有 timeout 属性的命令，在等待时间到达，强制屏蔽其他不具有 timeout 属性命令向仲裁器发起仲裁请求，而自己立刻提起仲裁请求。

安全功能

DDRC 支持安全功能，安全功能使能时，对于越权的读写访问总线会返回 BUS ERROR，同时保护 DDR 数据不被改写/读出。

- 通过配置 `SEC_RGN_MAP`[`rgn`]，DDRC 可以同时维护最大 16 个安全地址区域，每个区域的安全属性都单独可配置。安全鉴权时，通过配置 `SEC_RGN_ATTRIB`，DDRC 支持读写鉴权分离，同时支持安全命令非安全命令鉴权分离。
- 同时 DDRC 支持基于 MID 的安全鉴权，即可以通过控制 `SEC_MID_WR/SEC_MID_RD` 信号来选择可以访问安全空间的 IP。

DDRC 支持安全功能寄存器读写保护，只有安全 CPU 才可以读写寄存器，非安全 CPU 无法对安全相关寄存器进行读写。

流量统计和命令 latency 统计功能

DDRC 支持流量统计功能：可以统计接口读写流量。可以对指定 ID 访问读/写流量单独统计。可以对总体读/写流量统计。支持 DDR 接口利用率统计，统计计数器支持连



续统计和单次统计，当为连续统计时计数器为非饱和计数器，计数器记到最大值后会卷绕，方便系统进行连续统计，所以需要系统在计数器未卷绕之前将计数器值读出。当为单次统计时，统计计数器在进行统计时间到达后停止统计，系统可以用此功能可以统计瞬时的流量以及 latency。

地址映射方式

DDRC 实现了将系统总线的访问地址转换为 DDR SDRAM 的访问地址。通过配置寄存器 `AXI_REGION_ATTRIB` 可以实现 `RRBC(Rank_row_bank_col)`、`RRBC(Row_rank_bank_col)`、`RRCBC(Rank_row_col_bank_col)`、`RRCBC(Row_rank_col_bank_col)` 四种映射方式。通过 `DDRC_CFG_RNKVOL [mem_row]` 和 `DDRC_CFG_RNKVOL [mem_col]` 寄存器对 SDRAM 的行地址和列地址位宽进行配置。DDRC 根据地址映射算法将系统总线地址转换为 DDR SDRAM 地址。

下面以 `RRCBC(Rank_row_col_bank_col)` 模式示例说明系统总线地址和 DDR SDRAM 地址的映射算法。假设系统总线地址为 `BUSADR[31:0]`，有效地址为 `BUSADR[m-1:0]`，DDR4 SDRAM 的地址为 `DDRADR[15:0]`。当 `DDRADR[15:0]` 作为行地址时，其有效地址为 `DDRROW[x-1:0]`，作为列地址时，其有效地址为 `DDRCOL[y-1:0]`，DDR 的 BANK 地址为 `DDRBA[z-1:0]`，DDRC 的存储数据总线宽度为 32bit，此时 `AXI_REGION_ATTRIB [ch_mode]=2'b01`（`AXI_REGION_ATTRIB [ch_mode]=2'b11` 时候暂不做介绍）。

`AXI_REGION_ATTRIB [bnk_mod]` 选择 bank 交织的粒度，假设配置为 `3'b011` 时（对接 32bit 位宽的器件），即按 256B 为粒度交织，则地址映射关系为：

$$\text{BUSADR}[m-1:0] = \{\text{DDRROW}[x-1:0], \text{DDRCOL}[y-1:6], \text{DDRBA}[z-1:0], \text{DDRCOL}[5:0], 2\{b0\}\}。$$

4.1.4 工作方式

4.1.4.1 软复位

DDRC 不能进行单独的复位操作。只有在全局软复位时，才能复位 DDRC。复位之后，需要按照初始化流程进行重新初始化 DDR3(L)/DDR4 SDRAM。

4.1.4.2 DDR 初始化配置流程

系统上电之后，必须先完成 DDR3(L)/DDR4 SDRAM 的初始化操作，系统才能访问 DDR SDRAM。在进行初始化之前需要注意以下几点：

- 对 DDR3(L)/DDR4 SDRAM 进行上电操作时，需要遵循 JEDEC 标准。
- 该初始化过程需要在系统进入 NORMAL 模式后进行。

整个 DDR 子系统的初始化步骤如下：

- 步骤 1. 配置 DDR 的地址区域映射，地址区域属性。
- 步骤 2. 配置 DMC 的工作模式，时钟频率，时序参数等。
- 步骤 3. 配置 DDRPHY 的工作模式，时序参数，DDRPHY IO 驱动和 ODT 阻抗，读写命令通路的 delay 参数等。
- 步骤 4. 启动 DDRPHY 进行初始化。



步骤 5. 等待 DDRPHY 初始化完成。

步骤 6. 配置 DMC 使能 auto refresh 命令发送功能。

----结束

完成以上步骤以后，DDR3(L)/DDR4 SDRAM 就可以正常工作。

说明：针对不同的 DDR 颗粒，所配置的寄存器的值可能会有部分不同，但配置过程需按上述步骤进行。

4.1.5 AXI 寄存器

4.1.5.1 AXI 寄存器概览

AXI 寄存器偏移地址中变量的取值范围和含义如表 4-3 所示。

表4-3 AXI 寄存器偏移地址变量表

变量名称	取值范围	描述
regions	0~3	地址区域个数。
ports	0~6	AXI 端口个数。
groups	0~3	端口 outstanding 统计分组组数。

AXI 寄存器概览如表 4-4 所示。

表4-4 AXI 寄存器概览（基址是 0x1206_0000）

偏移地址	名称	描述	页码
0x004	AXI_CFG_LOCK	寄存器锁定控制	4-8
0x008	AXI_CKG	模块的时钟门控	4-9
0x020	AXI_ACTION	AXI 接口工作模式寄存器	4-9
0x100+0x10 ×regions	AXI_REGION_MAP	地址区域映射	4-10
0x104+0x10 ×regions	AXI_REGION_ATTRIB	地址区域属性	4-11
0x200+0x10 ×ports	AXI_QOS_MAP	命令优先级映射方式	4-13
0x204+0x10 ×ports	AXI_QOS_WRPRIn	写命令优先级映射表	4-14
0x208+0x10	AXI_QOS_RDPRIn	读命令优先级映射表	4-16



偏移地址	名称	描述	页码
$\times \text{ports}$			
$0x20C + 0x10 \times \text{ports}$	AXI_QOS_ADPTn	优先级自适应周期映射表	4-17
$0x300 + 0x10 \times \text{ports}$	AXI_OSTD_PRTn	端口的命令 OSTD 限制	4-19
$0x304 + 0x10 \times \text{ports}$	AXI_OSTD_PRT_STn	端口的命令 OSTD 统计值	4-19
$0x400 + 0x10 \times \text{groups}$	AXI_OSTD_GROUPn	端口分组的命令 OSTD	4-19
$0x404 + 0x10 \times \text{groups}$	AXI_OSTD_PRI0	各端口分组内基于优先级的命令 OSTD	4-20
$0x408 + 0x10 \times \text{groups}$	AXI_OSTD_PRI1	各端口分组内基于优先级的命令 OSTD	4-21
$0x40C + 0x10 \times \text{groups}$	AXI_OSTD_GROUP_STn	端口分组的命令 OSTD 统计值	4-23
0x60C	AXI_STATUS	各个端口的工作状态	4-23
0x61C	AXI_INT_STATUS	中断状态	4-23

4.1.5.2 AXI 寄存器描述

AXI_CFG_LOCK

AXI_CFG_LOCK 为寄存器锁定控制。

Offset Address: 0x004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留。	0x00000000
[0]	RW	apb_cfg_lock	锁定 MDDRC 内所有的寄存器配置模块 (AXI_IF/SEC/DMC/PUB)。 0: 不锁定; 1: 锁定(除本寄存器外的所有 AXI_IF 配置寄存器皆无法访问, 可降低功耗)。	0x0



AXI_CKG

AXI_CKG 为模块的时钟门控。

Offset Address: 0x008 Total Reset Value: 0x000F_1FFF

Bits	Access	Name	Description	Reset
[31:20]	RO	reserved	保留。	0x000
[19:16]	RW	sta_ckg_dmc	各个 DMC 的静态时钟门控。 sta_ckg_dmc[n]=0: 关闭 DMCn 及对应 pub 的时钟; sta_ckg_dmc[n]=1: 打开 DMCn 及对应 pub 的时钟。 注意: 当前支持四通道, n = 0~3。	0xF
[15:13]	RO	reserved	保留。	0x0
[12]	RW	dyn_ckg_rdr	REORDER 模块的动态时钟门控。 0: 时钟常开; 1: 模块空闲时, 自动关闭时钟。	0x1
[11:0]	RW	dyn_ckg_axi	各个 AXI 端口的动态时钟门控。 dyn_ckg_axi[n]=0: AXI 端口 n 的时钟常开; dyn_ckg_axi[n]=1: AXI 端口 n 内部各模块空闲时, 自动关闭时钟。	0xFFF

AXI_ACTION

AXI_ACTION 为 AXI 接口工作模式寄存器。

Offset Address: 0x020 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:20]	RO	reserved	保留。	0x000
[19:8]	RW	wr_rcv_mode	防死锁模式使能。 wr_rcv_mode[n]=0: 端口 n 的防死锁模式关闭; wr_rcv_mode[n]=1: 端口 n 的防死锁模式使能。 注意: 1.防死锁模式使能与否取决于总线构架的需求; 2.防死锁模式使能后, 对于 AXI 端口的写	0x000



Bits	Access	Name	Description	Reset
			命令 outstanding 被限制为 1。	
[7:2]	RO	reserved	保留。	0x00
[1]	RW	rd_wrap_split_en	wrap 读命令拆分使能。 0: 不拆分, 由 DMC 完成 wrap 地址卷绕; 1: 拆分。	0x1
[0]	RW	exclusive_en	exclusive 命令使能。 0: 禁止; 1: 使能。	0x1

AXI_REGION_MAP

AXI_REGION_MAP 为地址区域映射。

Offset Address: $0x100 + 0x10 \times \text{regions}$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:13]	RO	reserved	保留。	0x00000
[12]	RW	rgn_en	使能当前地址区域。 0: 禁止; 1: 使能。	0x0
[11]	RO	reserved	保留。	0x0
[10:8]	RW	rgn_size	当前地址区域的大小。 000: 16MB; 001: 32MB; 此时 rgn_base_addr/ch_offset[0]必须配置为零; 010: 64MB; 此时 rgn_base_addr/ ch _offset[1:0]必须配置为零; 011: 128MB; 此时 rgn_base_addr/ ch _offset[2:0]必须配置为零; 100: 256MB; 此时 rgn_base_addr/ ch _offset[3:0]必须配置为零; 101: 512MB; 此时 rgn_base_addr/ ch _offset[4:0]必须配置为零; 110: 1GB; 此时 rgn_base_addr/ ch _offset[5:0]必须配置为零; 111: 2GB; 此时 rgn_base_addr/ ch _offset[6:0]必须配置为零。	0x0



Bits	Access	Name	Description	Reset
			注意： 1、以上配置以 32bits 地址为例，若地址位宽不同则配置含义不同。2、双通道交织时，rgn_size 不可配置为 0；四通道交织时，rgn_size 不可配置为 0 或 1。	
[7:0]	RW	rgn_base_addr	当前地址区域的基地址(高八位)。 注意： 由于只能设置地址的高八位，因此区域地址对齐的粒度会因地址位宽不同而不同。 例如： 对于 32bits 地址，地址区域的基地址低 24 位为零，区域地址按 16MB 对齐；	0x00

AXI_REGION_ATTRIB

AXI_REGION_ATTRIB 为地址区域属性。

Offset Address: $0x104 + 0x10 \times \text{regions}$ Total Reset Value: 0x0001_0004

Bits	Access	Name	Description	Reset
[31]	RO	reserved	保留。	0x0
[30:28]	RW	bnk_mod	当前地址区域的通道内 BANK 交织粒度(交织粒度随器件不同而不同)。 8Byte*2^n (8bit 器件); 16Byte*2^n (16bit 器件); 32Byte*2^n (32bit 器件); 64Byte*2^n (64bit 器件); 其中 n=0~7。 注意： bank 交织粒度必须大于等于地址对齐粒度(addr_aligned)。	0x0
[27:26]	RO	reserved	保留。	0x0
[25:24]	RW	rnk_mod	当前地址区域的通道内 RANK 交织模式。 00: 使用 DMC 内部配置; 01: DIMM 交织; 10: 双 RANK 地址交织; 11: 四 RANK 地址交织。	0x0



Bits	Access	Name	Description	Reset
[23:19]	RO	reserved	保留。	0x00
[18:16]	RW	addr_aligned	当前地址区域的地址边界对齐粒度。 000: 8Byte(只有当 DMC 数据位宽为 64bit 时, 才可选择此项); 001: 16Byte; 010: 32Byte; 011: 64Byte; 100: 128Byte; 101: 256Byte; 110: 512Byte; 111: 1KByte; 注意: 1.跨越此地址对齐边界的命令会被拆分; 2.地址对齐粒度必须小于等于 bank 交织粒度(bnk_mod); 3.地址交织时, 地址对齐粒度必须小于等于地址交织粒度(ch_intlv); 4.地址独立时, 地址对齐粒度必须小于等于 DDR 器件的 PAGE 大小。	0x1
[15:8]	RW	ch_offset	当前地址区域在通道内的偏移地址(高八位)。 注意: 此地址用于替换通道内地址的高八位, 以实现映射地址在通道内的偏移, 会因地址位宽不同而不同。例如: 对于 32bits 地址, 可替换通道映射后地址 [31:24];	0x00
[7:6]	RO	reserved	保留。	0x0
[5:4]	RW	ch_intlv	当前地址区域的地址交织粒度(单通道地址独立模式下, 此配置无效)。 00: 128Byte; 01: 256Byte; 10: 512Byte; 11: 1KByte; 注意: 1.命令以此粒度为边界, 被交替分发到多个通道; 2.通道交织时, 地址交织粒度必须大于等于	0x0



Bits	Access	Name	Description	Reset
			地址对齐粒度(addr_aligned)。	
[3:2]	RW	ch_mode	当前地址区域的通道映射模式。 00: 不映射; 01: 映射到单通道, 地址独立; 10: 映射到双通道, 地址交织; 注意: 试图访问未映射的地址区域的命令, 会被当做错误命令处理: 上报中断、记录命令信息、AXI 总线返回 SLVERR 响应。	0x1
[1:0]	RW	ch_start	当前地址区域的映射起始通道。 00: 通道 0; 01: 通道 1; 注意: 当映射模式为单通道时, 映射起始通道可为任意通道; 当映射模式为双通道时, 映射起始通道必须为通道 0(0/1 通道交织)或通道 2(2/3 通道地址交织);	0x0

AXI_QOS_MAP

AXI_QOS_MAP 为命令优先级映射方式。

Offset Address: 0x200+0x10 x ports Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:22]	RO	reserved	保留。	0x000
[21]	RW	qos_rever	QoS 反转使能(用于同一端口内读写命令的优先级仲裁)。 0: axqos=0 时, 为最低优先级; 1: axqos=0 时, 为最高优先级。 注意: 当 rw_arb_mode==1 时, 此寄存器有效。	0x0
[20]	RW	rw_arb_mode	同一端口内读写命令仲裁模式。 0: 基于命令读写类型: a.先到先仲裁; b.同时到达(或被同时被反压)时, 按照先读命令后写命令的顺序轮流仲裁;	0x0



Bits	Access	Name	Description	Reset
			1: 基于命令优先级: a.先到先仲裁; b.同时到达(或被同时被反压)时, 则按照命令优先级仲裁; c. 若优先级相同, 则按照 LRU 原则仲裁给上一次未获得仲裁的命令。	
[19:13]	RO	reserved	保留。	0x00
[12]	RW	pri_map_mode	读写命令优先级映射模式。 0: 采用读写命令的 3bit 随路信号映射索引; 1: 采用命令 ID 低 16 位中的任意 3bits 映射索引。 说明: 使用上述两种方法之一, 得到映射的索引, 再从优先级查找表(这里的查找表指的是 AXI_QOS_WRPRI 或者 AXI_QOS_RDPRI 寄存器内容)中映射出命令优先级	0x0
[11:0]	RW	id_map_idx	通过命令 ID 映射优先级时, 该寄存器配置命令 ID 的 3 个选择位: id_map_idx[11:8]: 选取命令 ID 低 16 位中的任意 1bit, 作为 idx[2]; id_map_idx[7:4]: 选取命令 ID 低 16 位中的任意 1bit, 作为 idx[1]; id_map_idx[3:0]: 选取命令 ID 低 16 位中的任意 1bit, 作为 idx[0]。 例如: id_map_idx 配置为 0x3A0, 则命令 ID 的 {ID[3],ID[10],ID[0]}拼合成索引 idx[2:0], 用于从优先级查找表(这里的查找表指的是 AXI_QOS_WRPRI 或者 AXI_QOS_RDPRI 寄存器内容)中映射出命令优先级。 注意: 此处的命令 ID 为命令的原始 ID。	0x000

AXI_QOS_WRPRI_n

AXI_QOS_WRPRI_n 为写命令优先级映射表。

Offset Address: 0x204 + 0x10 x ports Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31]	RO	reserved	保留。	0x0
[30:28]	RW	wr_pri7	写命令映射为 7 时的优先级配置。 0x0: 最高优先级; 0x7: 最低优先级。	0x0
[27]	RO	reserved	保留。	0x0
[26:24]	RW	wr_pri6	写命令映射为 6 时的优先级配置。 0x0: 最高优先级; 0x7: 最低优先级。	0x0
[23]	RO	reserved	保留。	0x0
[22:20]	RW	wr_pri5	写命令映射为 5 时的优先级配置。 0x0: 最高优先级; 0x7: 最低优先级。	0x0
[19]	RO	reserved	保留。	0x0
[18:16]	RW	wr_pri4	写命令映射为 4 时的优先级配置。 0x0: 最高优先级; 0x7: 最低优先级。	0x0
[15]	RO	reserved	保留。	0x0
[14:12]	RW	wr_pri3	写命令映射为 3 时的优先级配置。 0x0: 最高优先级; 0x7: 最低优先级。	0x0
[11]	RO	reserved	保留。	0x0
[10:8]	RW	wr_pri2	写命令映射为 2 时的优先级配置。 0x0: 最高优先级; 0x7: 最低优先级。	0x0
[7]	RO	reserved	保留。	0x0
[6:4]	RW	wr_pri1	写命令映射为 1 时的优先级配置。 0x0: 最高优先级;	0x0



Bits	Access	Name	Description	Reset
			 0x7: 最低优先级。	
[3]	RO	reserved	保留。	0x0
[2:0]	RW	wr_pri0	写命令映射为 0 时的优先级配置。 0x0: 最高优先级; 0x7: 最低优先级。	0x0

AXI_QOS_RDPRIn

AXI_QOS_RDPRIn 为读命令优先级映射表。

Offset Address: 0x208 + 0x10 x ports Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RO	reserved	保留。	0x0
[30:28]	RW	rd_pri7	读命令映射为 7 时的优先级配置。 0x0: 最高优先级; 0x7: 最低优先级。	0x0
[27]	RO	reserved	保留。	0x0
[26:24]	RW	rd_pri6	读命令映射为 6 时的优先级配置。 0x0: 最高优先级; 0x7: 最低优先级。	0x0
[23]	RO	reserved	保留。	0x0
[22:20]	RW	rd_pri5	读命令映射为 5 时的优先级配置。 0x0: 最高优先级; 0x7: 最低优先级。	0x0
[19]	RO	reserved	保留。	0x0
[18:16]	RW	rd_pri4	读命令映射为 4 时的优先级配置。 0x0: 最高优先级; 0x7: 最低优先级。	0x0



Bits	Access	Name	Description	Reset
[15]	RO	reserved	保留。	0x0
[14:12]	RW	rd_pri3	读命令映射为 3 时的优先级配置。 0x0: 最高优先级; 0x7: 最低优先级。	0x0
[11]	RO	reserved	保留。	0x0
[10:8]	RW	rd_pri2	读命令映射为 2 时的优先级配置。 0x0: 最高优先级; 0x7: 最低优先级。	0x0
[7]	RO	reserved	保留。	0x0
[6:4]	RW	rd_pri1	读命令映射为 1 时的优先级配置。 0x0: 最高优先级; 0x7: 最低优先级。	0x0
[3]	RO	reserved	保留。	0x0
[2:0]	RW	rd_pri0	读命令映射为 0 时的优先级配置。 0x0: 最高优先级; 0x7: 最低优先级。	0x0

AXI_QOS_ADPTn

AXI_QOS_ADPTn 为优先级自适应周期映射表。

Offset Address: 0x20C + 0x10 x ports Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:28]	RO	reserved	保留。	0x0
[27:24]	RW	wr_adpt_high	写命令优先级自适应周期配置(高档位)。 0x0: 禁止优先级自适应功能; 0x1~0xF: N*16 个时钟周期。 说明: 每当自适应周期到达后, 命令优先级自动提升一级, 优先级最高可以提升至次高优	0x0



Bits	Access	Name	Description	Reset
			优先级。	
[23]	RO	reserved	保留。	0x0
[22:20]	RW	wr_adpt_lvl	写命令优先级自适应周期配置水线。 0x0~0x1: 保留; 0x2~0x7: 命令优先级高于等于此水线时, 使用高档位配置, 否则使用低档位配置。 举例: 0x5: 优先级 2~5 的写命令使用 wr_adpt_high, 优先级 6~7 的写命令使用 wr_adpt_low。	0x0
[19:16]	RW	wr_adpt_low	写命令优先级自适应周期配置(低档位)。 0x0: 禁止优先级自适应功能; 0x1~0xF: N*16 个时钟周期。 说明: 每当自适应周期到达后, 命令优先级自动提升一级, 优先级最高可以提升至次高优先级。	0x0
[15:12]	RO	reserved	保留。	0x0
[11:8]	RW	rd_adpt_high	读命令优先级自适应周期配置(高档位)。 0x0: 禁止优先级自适应功能; 0x1~0xF: N*16 个时钟周期。 说明: 每当自适应周期到达后, 命令优先级自动提升一级, 优先级最高可以提升至次高优先级。	0x0
[7]	RO	reserved	保留。	0x0
[6:4]	RW	rd_adpt_lvl	读命令优先级自适应周期配置水线。 0x0~0x1: 保留; 0x2~0x7: 命令优先级高于等于此水线时, 使用高档位配置, 否则使用低档位配置。 举例: 0x5: 优先级 2~5 的读命令使用 rd_adpt_high, 优先级 6~7 的读命令使用 rd_adpt_low。	0x0
[3:0]	RW	rd_adpt_low	读命令优先级自适应周期配置(低档位)。	0x0



Bits	Access	Name	Description	Reset
			0x0: 禁止优先级自适应功能; 0x1~0xF: N*16 个时钟周期。 说明: 每当自适应周期到达后, 命令优先级自动提升一级, 优先级最高可以提升至次高优先级。	

AXI_OSTD_PRTn

AXI_OSTD_PRTn 为端口的命令 OSTD 限制。

Offset Address: 0x300+0x10 x ports Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:7]	RO	reserved	保留。	0x0000000
[6:0]	RW	prt_ostd_lvl	端口的命令 OSTD 水线。 0x00: 不限制本端口的命令 OSTD 个数; 0x01~0x7F: 当前端口允许的最大命令 OSTD 个数。 注意: 1. 此处的 OSTD 个数, 本质上是后级模块 QOSBUF 中的命令个数(下同); 2. 水线设置不能大于参数配置的 QOSBUF 的命令队列深度(下同)。	0x00

AXI_OSTD_PRT_STn

AXI_OSTD_PRT_STn 为端口的命令 OSTD 统计值。

Offset Address: 0x304+0x10 x ports Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:7]	RO	reserved	保留。	0x0000000
[6:0]	RO	prt_ostd_st	端口的命令 OSTD 统计值。	0x00

AXI_OSTD_GROUPn

AXI_OSTD_GROUPn 为端口分组的命令 OSTD。



Offset Address: 0x400+0x10 x groups Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:28]	RO	reserved	保留。	0x0
[27:16]	RW	group_ostd_sel	分组的端口选择： group_ostd_sel[n]=0: 端口 n 未被选定； group_ostd_sel[n]=1: 端口 n 被选定。 注意： 只有被选定的端口才参与分组的 OSTD 统计。	0x000
[15:7]	RO	reserved	保留。	0x000
[6:0]	RW	group_ostd_lvl	分组的命令 OSTD 水线。 0x00: 不限制本分组的命令 OSTD 个数； 0x01~0x7F: 当前分组允许的最大累积命令 OSTD 个数。 注意： 1. 分组命令 OSTD 是当前分组所选定端口的命令 OSTD 之和； 2. 超出水线后，分组内所有端口都会被阻塞。	0x00

AXI_OSTD_PRI0

AXI_OSTD_PRI0 为各端口分组内基于优先级的命令 OSTD。

Offset Address: 0x404+0x10 x groups Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RO	reserved	保留。	0x0
[30:24]	RW	pri3_ostd_lvl	优先级 3 的命令 OSTD 水线。 0x00: 不限制选定分组的累积命令 OSTD 个数； 0x01~0x7F: 选定分组的累积命令 OSTD 个数达到此水线时，只允许优先级高于 1 的命令通过。 注意： 1. 命令 OSTD 的统计和限制都是基于端口分组的； 2. 若使能优先级传递功能，则低优先级命令可能会因为优先级被提升而不再被阻塞。	0x00



Bits	Access	Name	Description	Reset
[23]	RO	reserved	保留。	0x0
[22:16]	RW	pri2_ostd_lvl	优先级 2 的命令 OSTD 水线。 0x00: 不限制选定分组的累积命令 OSTD 个数; 0x01~0x7F: 选定分组的累积命令 OSTD 个数达到此水线时, 只允许优先级高于 1 的命令通过。 注意: 1.命令 OSTD 的统计和限制都是基于端口分组的; 2.若使能优先级传递功能, 则低优先级命令可能会因为优先级被提升而不再被阻塞。	0x00
[15]	RO	reserved	保留。	0x0
[14:8]	RW	pri1_ostd_lvl	优先级 1 的命令 OSTD 水线。 0x00: 不限制选定分组的累积命令 OSTD 个数; 0x01~0x7F: 选定分组的累积命令 OSTD 个数达到此水线时, 只允许优先级高于 1 的命令通过。 注意: 1.命令 OSTD 的统计和限制都是基于端口分组的; 2.若使能优先级传递功能, 则低优先级命令可能会因为优先级被提升而不再被阻塞。	0x00
[7:0]	RO	reserved	保留。	0x00

AXI_OSTD_PRI1

AXI_OSTD_PRI1 为各端口分组内基于优先级的命令 OSTD。

Offset Address: 0x408+0x10 x groups Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RO	reserved	保留。	0x0
[30:24]	RW	pri7_ostd_lvl	优先级 7 的命令 OSTD 水线: 0x00: 不限制选定分组的累积命令 OSTD 个数; 0x01~0x7F: 选定分组的累积命令 OSTD	0x00



Bits	Access	Name	Description	Reset
			个数达到此水线时，只允许优先级高于 1 的命令通过。 注意： 1.命令 OSTD 的统计和限制都是基于端口分组的； 2.若使能优先级传递功能，则低优先级命令可能会因为优先级被提升而不再被阻塞。	
[23]	RO	reserved	保留。	0x0
[22:16]	RW	pri6_ostd_lvl	优先级 6 的命令 OSTD 水线： 0x00：不限制选定分组的累积命令 OSTD 个数； 0x01~0x7F：选定分组的累积命令 OSTD 个数达到此水线时，只允许优先级高于 1 的命令通过。 注意： 1.命令 OSTD 的统计和限制都是基于端口分组的； 2.若使能优先级传递功能，则低优先级命令可能会因为优先级被提升而不再被阻塞。	0x00
[15]	RO	reserved	保留。	0x0
[14:8]	RW	pri5_ostd_lvl	优先级 5 的命令 OSTD 水线： 0x00：不限制选定分组的累积命令 OSTD 个数； 0x01~0x7F：选定分组的累积命令 OSTD 个数达到此水线时，只允许优先级高于 1 的命令通过。 注意： 1.命令 OSTD 的统计和限制都是基于端口分组的； 2.若使能优先级传递功能，则低优先级命令可能会因为优先级被提升而不再被阻塞。	0x00
[7]	RO	reserved	保留。	0x0
[6:0]	RW	pri4_ostd_lvl	优先级 4 的命令 OSTD 水线： 0x00：不限制选定分组的累积命令 OSTD 个数； 0x01~0x7F：选定分组的累积命令 OSTD 个数达到此水线时，只允许优先级高于 1 的命令通过。	0x00



Bits	Access	Name	Description	Reset
			注意： 1.命令 OSTD 的统计和限制都是基于端口分组的； 2.若使能优先级传递功能，则低优先级命令可能会因为优先级被提升而不再被阻塞。	

AXI_OSTD_GROUP_STn

AXI_OSTD_GROUP_STn 为端口分组的命令 OSTD 统计值。

Offset Address: 0x40C + 0x10 x groups Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:7]	RO	reserved	保留。	0x0000000
[6:0]	RO	group_ostd_st	端口分组的命令 OSTD 统计值。	0x00

AXI_STATUS

AXI_STATUS 为各个端口的工作状态。

Offset Address: 0x60C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:12]	RO	reserved	保留。	0x00000
[11:0]	RO	axi_if_busy	各个 AXI 端口的工作状态。 axi_prt_busy[n]=0: AXI 端口 n 空闲； axi_prt_busy[n]=1: AXI 端口 n 正在处理命令或数据。	0x000

AXI_INT_STATUS

AXI_INT_STATUS 为中断状态。

Offset Address: 0x61C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:12]	RO	reserved	保留。	0x00000
[11:0]	RO	int_ports	中断源指示。	0x000



Bits	Access	Name	Description	Reset
			interrupt_ports[n]=1: 中断来自于端口 n。 注意: 当命令未通过安全权限检查且中断使能时, 相应端口会产生中断。	

4.1.6 TZASC 寄存器

4.1.6.1 TZASC 寄存器概览

TZASC 寄存器偏移地址中变量的取值范围和含义如表 4-5 所示。

表4-5 TZASC 寄存器偏移地址变量表

变量名称	取值范围	描述
SEC_Regions	0~15	安全地址区域个数。

TZASC 寄存器概览如表 4-6 所示。

表4-6 TZASC 寄存器概览（基址是 0x1206_1000）

偏移地址	名称	描述	页码
0x004	SEC_BYPASS	安全功能使能	4-25
0x010	SEC_LOCKDOWN_SELECT	锁定安全属性相关配置寄存器	4-26
0x018	SEC_INT_MSK_W MID	写 MID 的中断屏蔽	4-28
0x01C	SEC_INT_MSK_RM ID	读 MID 的中断屏蔽	4-28
0x020	SEC_INT_EN	中断使能	4-29
0x024	SEC_INT_STATUS	中断状态	4-29
0x028	SEC_INT_CLEAR	中断清除	4-30
0x080	SEC_FAIL_CMD_I NF_0	记录首个越权命令的信息 0	4-30
0x084	SEC_FAIL_CMD_I NF_1	记录首个越权命令的信息 1	4-30
0x088	SEC_FAIL_CMD_I NF_2	记录首个越权命令的信息 2	4-31



偏移地址	名称	描述	页码
0x100+0x10* sec_regions	SEC_RGN_MAP	地址区域映射	4-31
0x104+0x10* sec_regions	SEC_RGN_ATTRIB	各地址区域的权限属性设置	4-31
0x108+0x10* sec_regions	SEC_MID_WR	写命令匹配 MasterID 及其掩码	4-32
0x10C+0x10* sec_regions	SEC_MID_RD	读命令匹配 MasterID 及其掩码。	4-33
0x200+0x10* sec_regions	SEC_RGN_MAP_EX T	地址区域映射	4-33
0x204+0x10* sec_regions	SEC_MID_WR_EX T	读命令匹配 MasterID 及其掩码。 (高 32 个 MID)	4-33
0x208+0x10* sec_regions	SEC_MID_RD_EXT	读命令匹配 MasterID 及其掩码。	4-34
0x21C	SEC_MASTER_TY PE1_EXT	master 的写命令鉴权类型 1 (高 32 个 MID)	4-34
0x310	SEC_MASTER_TY PE1	master 的写命令鉴权类型 1	4-34
0x330	SEC_INT_MSK_W MID	写 MID 的中断屏蔽 (高 32 个 MID)	4-28
0x334	SEC_INT_MSK_RM ID	读 MID 的中断屏蔽 (高 32 个 MID)	4-28
0x3C0	SEC_ITCRG	集成测试控制寄存器。	4-35
0x3C4	SEC_ITIP	集成测试输入寄存器	4-36
0x3C8	SEC_ITOP	集成测试输出寄存器	4-36

4.1.6.2 TZASC 寄存器描述

SEC_BYPASS

SEC_BYPASS 为安全功能使能。

Offset Address: 0x004 Total Reset Value: 0x0000_0005

Bits	Access	Name	Description	Reset
[31:8]	RO	reserved	保留。	0x0000000
[7: 0]	RW	bypass	安全模块 bypass。	0x5



Bits	Access	Name	Description	Reset
			8'b10100101: 安全模块 bypass; 其他值: 安全模模块使能。 被锁定条件: sucure_boot_lock 信号被拉高过, 且 ld_bypass 配置为 1。	

SEC_LOCKDOWN_SELECT

SEC_LOCKDOWN_SELECT 为锁定安全属性相关配置寄存器。

Offset Address: 0x008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RO	reserved	保留。	0x00
[23:16]	RW	ld_rgns_sel	选择需将相关配置寄存器锁定为只读的 regions 范围。 ld_rgns_sel=N: 从最高位起选定连续的 N+1 个 rgn。 举例: ld_rgns_sel=5: rgn15~rgn10 的相关寄存器被选定 (假设 no_of_rgns=16)。 注意: 1.若选定的 rgn 个数大于实际 rgn 总数 no_of_rgns, 则选定全部 rgn; 2.若某 rgn 被选定, 则此 rgn 对应的如下相关配置寄存器皆被选定: (1) SEC_rgn_MAP; (2) SEC_rgn_ATTRIB; (3) SEC_MID_WR; (4) SEC_MID_RD; 被锁定条件: sucure_boot_lock 信号被拉高过, 且 ld_rgn_register 配置为 1。	0x00
[15:13]	RO	reserved	保留。	0x0
[12]	RW	ld_sec_master_type0	锁定 SEC_MASTER_TYPE0 寄存器为只读。 0: 不锁定; 1: 锁定。 被锁定条件: sucure_boot_lock 信号被拉高过。	0x0



Bits	Access	Name	Description	Reset
[11]	RW	ld_sec_share_rgn_start_addr	锁定 SEC_MASTER_TYPE1 寄存器为只读。 0: 不锁定; 1: 锁定。 被锁定条件: sucure_boot_lock 信号被拉高过。	0x0
[10]	RW	ld_sec_master_type0	锁定 SEC_SHARE_RGN_START_ADDR0~3 寄存器为只读。 0: 不锁定; 1: 锁定。 被锁定条件: sucure_boot_lock 信号被拉高过。	0x0
[9]	RW	ld_sec_share_rgn_end_addr	锁定 SEC_SHARE_RGN_END_ADDR0~3 寄存器为只读。 0: 不锁定; 1: 锁定。 被锁定条件: sucure_boot_lock 信号被拉高过。	0x0
[8]	RW	ld_sec_share_rgn_en	锁定 SEC_SHARE_RGN_EN0~3 寄存器为只读。 0: 不锁定; 1: 锁定。 被锁定条件: sucure_boot_lock 信号被拉高过。	0x0
[7:5]	RO	reserved	保留。	0x0
[4]	RW	ld_bypass	锁定 SEC_BYPASS 寄存器为只读。 0: 不锁定; 1: 锁定。 被锁定条件: sucure_boot_lock 信号被拉高过。	0x0
[3]	RW	ld_mid_wr	锁定选定 region 的 SEC_MID_WR 寄存器为只读。 0: 不锁定; 1: 锁定。 被锁定条件: sucure_boot_lock 信号被拉高过。	0x0



Bits	Access	Name	Description	Reset
[2]	RW	ld_mid_rd	锁定选定 region 的 SEC_MID_RD 寄存器为只读。 0: 不锁定; 1: 锁定。 被锁定条件: sucure_boot_lock 信号被拉高过。	0x0
[1]	RW	ld_rgn_attrib	锁定选定 region 的 SEC_REGION_ATTRIB 寄存器为只读。 0: 不锁定; 1: 锁定。 被锁定条件: sucure_boot_lock 信号被拉高过。	0x0
[0]	RW	ld_rgn_map	锁定选定 region 的 SEC_REGION_MAP 寄存器为只读。 0: 不锁定; 1: 锁定。 被锁定条件: sucure_boot_lock 信号被拉高过。	0x0

SEC_INT_MSK_WMID

SEC_INT_MSK_WMID 为写 MID 的中断屏蔽寄存器。

Offset Address: 0x018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31: 0]	RW	int_msk_wr_mid	根据 mid 来 mask 掉对应 master 的写中断上报和信息记录; Bit[0]: 1 表示屏蔽掉写 mid 为 0 的中断; 依此类推	0x00000000

SEC_INT_MSK_RMID

SEC_INT_MSK_RMID 为读 MID 的中断屏蔽寄存器。

Offset Address: 0x01C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31: 0]	RW	int_msk_rd_mid	根据 mid 来 mask 掉对应 master 的读中断	0x00000000



Bits	Access	Name	Description	Reset
			上报和信息记录； Bit[0]: 1 表示屏蔽掉读 mid 为 0 的中断； 依此类推	

SEC_INT_EN

SEC_INT_EN 为中断使能寄存器。

Offset Address: 0x020 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留。	0x0000000
[0]	RW	int_en	中断使能。 0: 禁止； 1: 使能。 注意：只是禁止中断输出，但相关寄存器 还是会正常记录中断状态及越权命令信 息。	0x1

SEC_INT_STATUS

SEC_INT_STATUS 为中断状态寄存器。

Offset Address: 0x024 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	RO	reserved	保留。	0x00000
[1]	RO	overrun	多次中断指示。 0: 中断未产生或只产生了一次； 1: 中断已产生多次。	0x0
[0]	RO	status	中断状态指示。 0: 中断未产生； 1: 中断已产生。 注意：从 AXI_IF 模块的 AXI_INT_STATUS 寄存器中，可识别出中 断来自于哪个 AXI 端口的 SEC 模块。	0x0



SEC_INT_CLEAR

SEC_INT_CLEAR 为中断清除寄存器。

Offset Address: 0x028 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31: 0]	WO	clear	写入任意值，可清零中断状态寄存器 int_status	0x00000000

SEC_FAIL_CMD_INF_0

SEC_FAIL_CMD_INF_0 为首个越权命令的信息 0。

Offset Address: 0x080 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31: 0]	RO	address_low	中断状态清除前的首个越权命令的低 32 位 地址	0x00000000

SEC_FAIL_CMD_INF_1

SEC_FAIL_CMD_INF_1 为首个越权命令的信息 1。

Offset Address: 0x084 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:21]	RO	reserved	保留。	0x000
[20]	RO	access_type	中断状态清除前首个越权命令的读写模式 信息。 0: read 访问; 1: write 访问。	0x0
[19:18]	RO	reserved	保留。	0x0
[17]	RO	status	中断状态清除前首个越权命令的安全模式 信息。 0: secure 访问; 1: non-secure 访问。	0x0
[16]	RO	status	中断状态清除前首个越权命令的特权模式 信息。 0: unprivileged 访问; 1: privileged 访问。	0x0



Bits	Access	Name	Description	Reset
[15:8]	RO	reserved	保留。	0x00
[7:0]	RO	status	中断状态清除前的首个越权命令的高位地址（如果存在高位地址）	0x00

SEC_FAIL_CMD_INF_2

SEC_FAIL_CMD_INF_2 为首个越权命令的信息 2。

Offset Address: 0x088 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RO	Mid	中断状态清除前的首个越权命令的命令 MasterID	0x00
[23:0]	RO	Id	中断状态清除前的首个越权命令的命令 ID	0x000000

SEC_RGN_MAP

SEC_RGN_MAP 为地址区域映射寄存器。

Offset Address: 0x100+0x10*sec_regions Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	rgn_en	各个区域的使能控制。 0: 不使能; 1: 使能; 注意: region0 不可配置, 默认使能。	0x00
[30:24]	RO	reserved	保留。	0x00
[23:0]	RW	rgn_base_addr	各个区域的起始地址设置（高位部分）。 设置地址的[31:16]位。 注意: rgn0 不可配置, 默认覆盖整个地址空间。	0x00

SEC_RGN_ATTRIB

SEC_RGN_ATTRIB 为各地址区域的权限属性设置寄存器。

Offset Address: 0x104+0x10*sec_regions Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:10]	RO	reserved	保留。	0x000000
[9]	RW	Mid_inv	MasterID 匹配权限反转。 0: MID 匹配成功的命令才可获得访问权限; 1: MID 匹配失败的命令才可获得访问权限。	0x0
[8]	RW	Mid_en	MasterID 匹配功能使能。 0: 禁止; 1: 使能。	0x0
[7:5]	RO	reserved	保留。	0x000000
[4]	RW	security_inv	安全权限反转使能。 0: 禁止 (若某 region 允许非安全的读/写, 则也必定允许安全的读/写) 1: 使能 (非安全读/写和安全读/写权限之间无依赖关系, 需各自设定)	0x0
[3:0]	RW	sp	当前区域的安全权限属性。 sp[3]: 安全读属性; sp[2]: 安全写属性; sp[1]: 非安全读属性; sp[0]: 非安全写属性; sp[n]=0: 禁止访问; sp[n]=1: 允许访问。	0x0

SEC_MID_WR

SEC_MID_WR 为写命令匹配 MasterID 及其掩码。

Offset Address: 0x108+0x10*sec_regions Total Reset Value: 0xffff_ffff

Bits	Access	Name	Description	Reset
[31: 0]	RW	mid_sel_wr	MID 的比特匹配选择 (写命令)。 mid_sel_wr[n]=0: 写命令 MID 的低六位为 n 时, 此命令为 MID 匹配失败; mid_sel_wr[n]=1: 写命令 MID 的低六位为 n 时, 此命令为 MID 匹配成功。	0xffffffff



SEC_MID_RD

SEC_MID_RD 为读命令匹配 MasterID 及其掩码。

Offset Address: 0x10C+0x10*sec_regions Total Reset Value: 0xffff_ffff

Bits	Access	Name	Description	Reset
[31:0]	RW	mid_sel_rd	MID 的比特匹配选择（读命令）。 mid_sel_rd[n]=0: 读命令 MID 的低六位为 n 时，此命令为 MID 匹配失败； mid_sel_rd[n]=1: 读命令 MID 的低六位为 n 时，此命令为 MID 匹配成功。	0xffffffff

SEC_RGN_MAP_EXT

SEC_RGN_MAP_EXT 为读命令匹配 MasterID 及其掩码。（高 32 个 MID）。

Offset Address: 0x204+0x10*sec_regions Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RO	reserved	保留。	0x00
[23:0]	RW	rgn_rgn_size	当前区域的大小： n*64KB。 注意： 1. rgn0 不可配置，默认覆盖整个地址空间； 2. 为每个区域配置的区域大小，累加上其区域起始地址后，不允许超出总的地址空间大小。	0x000000

SEC_MID_WR_EXT

SEC_MID_WR_EXT 为写命令匹配 MasterID 及其掩码。（高 32 个 MID）。

Offset Address: 0x204+0x10*sec_regions Total Reset Value: 0xffff_ffff

Bits	Access	Name	Description	Reset
[31: 0]	RW	mid_sel_wr	MID 的比特匹配选择（写命令）。 mid_sel_wr[n]=0: 写命令 MID 的低六位为 n 时，此命令为 MID 匹配失败； mid_sel_wr[n]=1: 写命令 MID 的低六位为 n 时，此命令为 MID 匹配成功。	0xffffffff



SEC_MID_RD_EXT

SEC_MID_RD_EXT 为读命令匹配 MasterID 及其掩码。（高 32 个 MID）。

Offset Address: 0x208+0x10*sec_regions Total Reset Value: 0xffff_ffff

Bits	Access	Name	Description	Reset
[31: 0]	RW	mid_sel_rd	MID 的比特匹配选择（读命令）。 mid_sel_rd[n]=0: 读命令 MID 的低六位为 n 时，此命令为 MID 匹配失败； mid_sel_rd[n]=1: 读命令 MID 的低六位为 n 时，此命令为 MID 匹配成功。	0xffffffff

SEC_MASTER_TYPE1_EXT

SEC_MASTER_TYPE1_EXT 为 master 的写命令鉴权类型 1（高 32 个 MID）。

Offset Address: 0x21C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31: 0]	RW	master_type1_ext	master_type[n]。 0: mid 为 n 的安全写访问到非共享区域中的某个地址区域时候，该地址区域是允许安全写,则允许访问； 1: mid 为 n 的安全写访问到非共享区域中的某个地址区域时候，该地址区域必须是允许安全写且不允许非安全读的属性，则允许访问。	0x00000000

SEC_MASTER_TYPE1

SEC_MASTER_TYPE1 为 master 的写命令鉴权类型 1。

Offset Address: 0x310 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31: 0]	RW	master_type1	master_type[n]。 0: mid 为 n 的安全写访问到非共享区域中的某个地址区域时候，该地址区域是允许安全写,则允许访问； 1: mid 为 n 的安全写访问到非共享区域中的某个地址区域时候，该地址区域必须是	0x00000000



Bits	Access	Name	Description	Reset
			允许安全写且不允许非安全读的属性，则允许访问。	

SEC_INT_MSK_WMID_EXT

SEC_INT_MSK_WMID_EXT 为写 MID 的中断屏蔽（高 32 个 MID）。

Offset Address: 0x330 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31: 0]	RW	int_msk_wr_mid_ext	根据 mid 来 mask 掉对应 master 的写中断上报和信息记录； bit[0]: 1 表示屏蔽掉写 mid 为 0 的中断；依此类推。	0x00000000

SEC_INT_MSK_RMID_EXT

SEC_INT_MSK_RMID_EXT 为读 MID 的中断屏蔽（高 32 个 MID）。

Offset Address: 0x334 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31: 0]	RW	int_msk_rd_mid_ext	根据 mid 来 mask 掉对应 master 的写中断上报和信息记录； bit[0]: 1 表示屏蔽掉写 mid 为 0 的中断；依此类推	0x00000000

SEC_ITCRG

SEC_ITCRG 为集成测试控制寄存器。

Offset Address: 0x3C0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留。	0x00000000
[0]	RW	int_test_en	中断测试使能。 0: 禁止； 1: 使能。 注意：使能此寄存器后，模块的中断输出，由 SEC_ITOP 寄存器配置决定。	0x0



SEC_ITIP

SEC_ITIP 为集成测试输入寄存器。

Offset Address: 0x3C4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留。	0x00000000
[0]	RO	itip_secure_boot_lock	寄存器保护的锁定状态指示。 0: 未锁定; 1: 已锁定。	0x0

SEC_ITOP

SEC_ITOP 为集成测试输出寄存器。

Offset Address: 0x3C8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留。	0x00000000
[0]	RO	itop_int	中断状态设置。 0: 低电平; 1: 高电平。 注意: 只有当中断测试使能时, 此寄存器设置有效。此时忽略模块内部产生的正常中断。	0x0

4.1.7 QOSBUF 寄存器

4.1.7.1 QOSBUF 寄存器概览

QOSBUF 寄存器偏移地址中变量的取值范围和含义如表 4-7 所示。

表4-7 QOSBUF 寄存器偏移地址变量表

变量名称	取值范围	描述
chans	0~3	通道个数
fgps	0~7	流量统计的组数
fids	0~6	流量统计匹配 id 的个数



QOSBUF 寄存器概览如表 4-8 所示。

表4-8 QOSBUF 寄存器概览（基址是 0x1206_4000）

偏移地址	名称	描述	页码
0x004	QOSB_ADPT_CTRL	配置 QosBuf 的 adpt 功能	4-38
0x088	QOSB_BUF_BYP	QosBuf 的 bypass 功能控制寄存器	4-39
0x09C	QOSB_WRTOUT0	QosBuf 的写命令 timeout 配置寄存器 0	4-39
0x0A0	QOSB_WRTOUT1	QosBuf 的写命令 timeout 配置寄存器 1	4-40
0x0A4	QOSB_WRTOUT2	QosBuf 的写命令 timeout 配置寄存器 2	4-41
0x0A8	QOSB_WRTOUT3	QosBuf 的写命令 timeout 配置寄存器 3	4-41
0x0AC	QOSB_WRTOUT4	QosBuf 的写命令 timeout 配置寄存器 4	4-42
0x0B0	QOSB_WRTOUT5	QosBuf 的写命令 timeout 配置寄存器 5	4-42
0x0BC	QOSB_WRTOUT_M AP	QosBuf 的写命令 timeout 映射控制寄存器	4-42
0x0D0	QOSB_RDTOUT_M AP	QosBuf 的读命令 timeout 映射控制寄存器	4-43
0x108	QOSB_CKG_CFG	QosBuf 的时钟控制寄存器	4-44
0x10C+ 0x4× chans	QOSB_DMC_LVL	QosBuf 的进入 DMC 的命令的水线控制寄存器	4-44
0x124	QOSB_CMD_SUM	QosBuf 命令计数的累积寄存器	4-44
0x180	QOSB_CMD_CNT	QOSBUF 中当前有多少命令的统计寄存器	4-45
0x190+ 0x4× chans	QOSB_RNK_CNT	QOSBUF 中当前每个 rank 中有多少命令的统计寄存器	4-45
0x1A0+ 0x4× chans	QOSB_BNK_CNT0	QOSBUF 中当前每个 Bank 有多少命令的统计寄存器 0	4-45
0x1B0+ 0x4× chans	QOSB_BNK_CNT1	QOSBUF 中当前每个 Bank 有多少命令的统计寄存器 1	4-46
0x1C0+ 0x4× chans	QOSB_BNK_CNT2	QOSBUF 中当前每个 Bank 有多少命令的统计寄存器 2	4-46



偏移地址	名称	描述	页码
0x1D0+ 0x4× chans	QOSB_BNK_CNT3	QOSBUF 中当前每个 Bank 有多少命令的统计寄存器 3	4-46
0x1E0	QOSB_OSTD_CNT	QOSBUF 中当前每个通道有多少 Outstanding 命令的统计寄存器	4-47
0x1E4	QOSB_WR_CMD_SUM	QosBuf 写命令计数的累积寄存器	4-47
0x1E8	QOSB_RD_CMD_SUM	QosBuf 读命令计数的累积寄存器	4-47
0x1F0	QOSB_TIMEOUT_MODE	QOSBUF 中 timeout 模式选择寄存器	4-48
0x300	QOSB_RDTOUT0	QosBuf 的读命令 timeout 配置寄存器 0	4-48
0x304	QOSB_RDTOUT1	QosBuf 的读命令 timeout 配置寄存器 1	4-49
0x308	QOSB_RDTOUT2	QosBuf 的读命令 timeout 配置寄存器 2	4-49
0x30C	QOSB_RDTOUT3	QosBuf 的读命令 timeout 配置寄存器 3	4-49
0x400	QOSB_RDTOUT4	QosBuf 的读命令 timeout 配置寄存器 4	4-50
0x404	QOSB_RDTOUT5	QosBuf 的读命令 timeout 配置寄存器 5	4-50

4.1.7.2 QOSBUF 寄存器描述

QOSB_ADPT_CTRL

QOSB_ADPT_CTRL 为配置 QosBuf 的 adpt 功能。

Offset Address: 0x004 Total Reset Value: 0x0000_0FF0

Bits	Access	Name	Description	Reset
[31:20]	RO	reserved	保留。	0x000
[19:4]	RW	adpt_share_cnt	Adapt 功能的低位共享计数器。当该计数器递减为 0 时，每个 cmd 对应的高位计数器减 1。	0x00FF
[3]	RO	reserved	保留。	0x0
[2]	RW	adpt_pri_lmt	Adapt 功能使能情况下，自适应最高优先级控制。 0: 在 Adpt 使能情况下，命令优先级可以自适应到次高优先级； 1: 在 Adpt 使能情况下，命令优先级可以	0x0



Bits	Access	Name	Description	Reset
			自适应到最高优先级。	
[1]	RW	wr_adpt_en	写命令 Adapt 功能使能控制。 0: 写命令 adapt 功能只有受到读命令 push 写命令才能起作用; 1: 写命令 Adapt 功能不需要读命令 push 写命令起作用。	0x0
[0]	RW	adpt_en	Adapt 功能使能控制。 0: Adapt 功能关闭; 1: Adapt 功能打开。	0x0

QOSB_BUF_BYP

QOSB_BUF_BYP 为 QosBuf 的 bypass 功能控制寄存器。

Offset Address: 0x088 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	RO	reserved	保留。	0x00000000
[1:0]	RW	qos_buf_byp	Qos buf 的 bypass 控制。 X0: 表示强制非 bypass, 即配成 00 或 10 时, 所有命令都要进入 qosbuf; 01: 表示根据 dmc 中 queue 的状态, 自动控制是否 bypass; 11: 表示强制 bypass, 所有命令都不进入 qosbuf (注意: 这些模式在访问开始之前就需要配置, 而不能在访问过程中进行动态配置)	0x0

QOSB_WRTOUT0

QOSB_WRTOUT0 为 QosBuf 的写命令 timeout 配置寄存器 0。

Offset Address: 0x09C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	RO	reserved	保留。	0x0
[29:20]	RW	wr_tout2	写命令 timeout 配置(档位 2)。 0x0: 禁止 timeout 功能;	0x000



Bits	Access	Name	Description	Reset
			0x1~0xFF: n*4 个时钟周期; 注意: timeout 配置为 10bits, 但实际计数为 12bits, 即低两位固定为零。	
[19:10]	RW	wr_tout1	写命令 timeout 配置(档位 1)。 0x0: 禁止 timeout 功能; 0x1~0xFF: n*4 个时钟周期; 注意: timeout 配置为 10bits, 但实际计数为 12bits, 即低两位固定为零。	0x000
[9:0]	RW	wr_tout0	写命令 timeout 配置(档位 0)。 0x0: 禁止 timeout 功能; 0x1~0xFF: n*4 个时钟周期; 注意: timeout 配置为 10bits, 但实际计数为 12bits, 即低两位固定为零。	0x000

QOSB_WRTOUT1

QOSB_WRTOUT1 为 QosBuf 的写命令 timeout 配置寄存器 1。

Offset Address: 0x0A0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	RO	reserved	保留。	0x0
[29:20]	RW	wr_tout5	写命令 timeout 配置(档位 5)。 0x0: 禁止 timeout 功能; 0x1~0xFF: n*4 个时钟周期; 注意: timeout 配置为 10bits, 但实际计数为 12bits, 即低两位固定为零。	0x000
[19:10]	RW	wr_tout4	写命令 timeout 配置(档位 4)。 0x0: 禁止 timeout 功能; 0x1~0xFF: n*4 个时钟周期; 注意: timeout 配置为 10bits, 但实际计数为 12bits, 即低两位固定为零。	0x000



Bits	Access	Name	Description	Reset
[9:0]	RW	wr_tout3	写命令 timeout 配置(档位 3)。 0x0: 禁止 timeout 功能; 0x1~0xFF: n*4 个时钟周期; 注意: timeout 配置为 10bits, 但实际计数为 12bits, 即低两位固定为零。	0x000

QOSB_WRTOUT2

QOSB_WRTOUT2 为 QosBuf 的写命令 timeout 配置寄存器 2。

Offset Address: 0x0A4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	RO	reserved	保留。	0x0
[29:20]	RW	wr_tout8	写命令 timeout 配置(档位 8)。 配置方式同 wr_tout0。	0x000
[19:10]	RW	wr_tout7	写命令 timeout 配置(档位 7)。 配置方式同 wr_tout0。	0x000
[9:0]	RW	wr_tout6	写命令 timeout 配置(档位 6)。 配置方式同 wr_tout0。	0x000

QOSB_WRTOUT3

QOSB_WRTOUT3 为 QosBuf 的写命令 timeout 配置寄存器 3。

Offset Address: 0x0A8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	RO	reserved	保留。	0x0
[29:20]	RW	wr_tout11	写命令 timeout 配置(档位 11)。 配置方式同 wr_tout0。	0x000
[19:10]	RW	wr_tout10	写命令 timeout 配置(档位 10)。 配置方式同 wr_tout0。	0x000
[9:0]	RW	wr_tout9	写命令 timeout 配置(档位 9)。 配置方式同 wr_tout0。	0x000



QOSB_WRTOUT4

QOSB_WRTOUT4 为 QosBuf 的写命令 timeout 配置寄存器 4。

Offset Address: 0x0AC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	RO	reserved	保留。	0x0
[29:20]	RW	wr_tout14	写命令 timeout 配置(档位 14)。 配置方式同 wr_tout0。	0x000
[19:10]	RW	wr_tout13	写命令 timeout 配置(档位 13)。 配置方式同 wr_tout0。	0x000
[9:0]	RW	wr_tout12	写命令 timeout 配置(档位 12)。 配置方式同 wr_tout0。	0x000

QOSB_WRTOUT5

QOSB_WRTOUT5 为 QosBuf 的写命令 timeout 配置寄存器 5。

Offset Address: 0x0B0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	RO	reserved	保留。	0x0
[9:0]	RW	wr_tout15	写命令 timeout 配置(档位 15)。 配置方式同 wr_tout0。	0x000

QOSB_WRTOUT_MAP

QOSB_WRTOUT_MAP 为 QosBuf 的写命令 timeout 映射控制寄存器。

Offset Address: 0x0BC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:29]	RO	reserved	保留。	0x0
[28:24]	RW	wrtout_map3	timeout 一共分为 16 个档位，其映射方法是 ID 中选出 4bit，并根据这 4bit 的值选择对应的档位。wrtout_map3 表示要选择的 bit3 的在整个 ID 中的位置。	0x00



Bits	Access	Name	Description	Reset
[23:21]	RO	reserved	保留。	0x0
[20:16]	RW	wrtout_map2	timeout 一共分为 16 个档位，其映射方法是 ID 中选出 4bit，并根据这 4bit 的值选择对应的档位。wrtout_map2 表示要选择的 bit2 的在整个 ID 中的位置。	0x00
[15:13]	RO	reserved	保留。	0x0
[12:8]	RW	wrtout_map1	timeout 一共分为 16 个档位，其映射方法是 ID 中选出 4bit，并根据这 4bit 的值选择对应的档位。wrtout_map1 表示要选择的 bit1 的在整个 ID 中的位置。	0x00
[7:5]	RO	reserved	保留。	0x0
[4:0]	RW	wrtout_map0	timeout 一共分为 16 个档位，其映射方法是 ID 中选出 4bit，并根据这 4bit 的值选择对应的档位。wrtout_map0 表示要选择的 bit0 的在整个 ID 中的位置。	0x00

QOSB_RDTOUT_MAP

QOSB_RDTOUT_MAP 为 QosBuf 的读命令 timeout 映射控制寄存器。

Offset Address: 0x0D0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:29]	RO	reserved	保留。	0x0
[28:24]	RW	rdtout_map3	timeout 一共分为 16 个档位，其映射方法是 ID 中选出 4bit，并根据这 4bit 的值选择对应的档位。rdtout_map3 表示要选择的 bit3 的在整个 ID 中的位置。	0x00
[23:21]	RO	reserved	保留。	0x0
[20:16]	RW	rdtout_map2	timeout 一共分为 16 个档位，其映射方法是 ID 中选出 4bit，并根据这 4bit 的值选择对应的档位。rdtout_map2 表示要选择的 bit2 的在整个 ID 中的位置。	0x00
[15:13]	RO	reserved	保留。	0x0
[12:8]	RW	rdtout_map1	timeout 一共分为 16 个档位，其映射方法是 ID 中选出 4bit，并根据这 4bit 的值选择对应的档位。rdtout_map1 表示要选择的 bit1 的在整个 ID 中的位置。	0x00



Bits	Access	Name	Description	Reset
[7:5]	RO	reserved	保留。	0x0
[4:0]	RW	rdtout_map0	timeout 一共分为 16 个档位，其映射方法是 ID 中选出 4bit，并根据这 4bit 的值选择对应的档位。rdtout_map0 表示要选择的 bit0 的在整个 ID 中的位置。	0x00

QOSB_CKG_CFG

QOSB_CKG_CFG 为 QosBuf 的时钟控制寄存器。

Offset Address: 0x108 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留。	0x00000000
[0]	RW	dyn_ck_gate	Qosbuf 模块的动态时钟门控。 0: 时钟常开; 1: 模块空闲时, 时钟自动门控。	0x1

QOSB_DMC_LVL

QOSB_DMC_LVL 为 QosBuf 的进入 DMC 的命令的线控制寄存器。

Offset Address: 0x10C + 0x4 x chans Total Reset Value: 0x0000_000F

Bits	Access	Name	Description	Reset
[31:5]	RO	reserved	保留。	0x00000000
[4:0]	RW	dmc_cmd_full_lvl	QOS BUF 进入 dmc 的命令的个数，当达到这个水线，表示 dmc 的 queue 已经满。 注意：由于流水的问题，当配置为 N 时，实际通过的命令个数为 N+1。另外，该寄存器不能在访问过程中进行动态配置。	0x0F

QOSB_CMD_SUM

QOSB_CMD_SUM 为 QosBuf 命令计数的累积寄存器。

Offset Address: 0x124 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RO	qos_cmd_sum	当前 QosBuf 暂存命令的值按周期累积，溢出卷绕。	0x00000000

QOSB_CMD_CNT

QOSB_CMD_CNT 为 QOSBUF 中当前有多少命令的统计寄存器。

Offset Address: 0x180 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	RO	reserved	保留。	0x000000
[7:0]	INT	qos_cmd_cnt	QosBuf 当前命令的个数，包括流水线上的命令。	0x00

QOSB_RNK_CNT

QOSB_RNK_CNT 为 QOSBUF 中当前每个 rank 中有多少命令的统计寄存器。

Offset Address: 0x190 + 0x4 x chans Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	RO	reserved	保留。	0x000000
[7:0]	RO	rnk0_cmd_cnt	QosBuf 中当前通道属于 rank0 的命令个数。	0x00

QOSB_BNK_CNT0

QOSB_BNK_CNT0 为 QOSBUF 中当前每个 Bank 有多少命令的统计寄存器 0。

Offset Address: 0x1A0 + 0x4 x chans Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RO	bnk3_cmd_cnt	统计 DMC 中当前通道属于 bank3 的命令个数。	0x00
[23:16]	RO	bnk2_cmd_cnt	统计 DMC 中当前通道属于 bank2 的命令个数。	0x00
[15:8]	RO	bnk1_cmd_cnt	统计 DMC 中当前通道属于 bank1 的命令个数。	0x00



Bits	Access	Name	Description	Reset
[7:0]	RO	bnk0_cmd_cnt	统计 DMC 中当前通道属于 bank0 的命令个数。	0x00

QOSB_BNK_CNT1

QOSB_BNK_CNT1 为 QOSBUF 中当前每个 Bank 有多少命令的统计寄存器 1。

Offset Address: $0x1B0 + 0x4 \times \text{chans}$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RO	bnk7_cmd_cnt	统计 DMC 中当前通道属于 bank7 的命令个数。	0x00
[23:16]	RO	bnk6_cmd_cnt	统计 DMC 中当前通道属于 bank6 的命令个数。	0x00
[15:8]	RO	bnk5_cmd_cnt	统计 DMC 中当前通道属于 bank5 的命令个数。	0x00
[7:0]	RO	bnk4_cmd_cnt	统计 DMC 中当前通道属于 bank4 的命令个数。	0x00

QOSB_BNK_CNT2

QOSB_BNK_CNT2 为 QOSBUF 中当前每个 Bank 有多少命令的统计寄存器 2。

Offset Address: $0x1C0 + 0x4 \times \text{chans}$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RO	bnk11_cmd_cnt	统计 DMC 中当前通道属于 bank11 的命令个数。	0x00
[23:16]	RO	bnk10_cmd_cnt	统计 DMC 中当前通道属于 bank10 的命令个数。	0x00
[15:8]	RO	bnk9_cmd_cnt	统计 DMC 中当前通道属于 bank9 的命令个数。	0x00
[7:0]	RO	bnk8_cmd_cnt	统计 DMC 中当前通道属于 bank8 的命令个数。	0x00

QOSB_BNK_CNT3

QOSB_BNK_CNT3 为 QOSBUF 中当前每个 Bank 有多少命令的统计寄存器 3。

Offset Address: $0x1D0 + 0x4 \times \text{chans}$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RO	bnk15_cmd_cnt	统计 DMC 中当前通道属于 bank15 的命令个数。	0x00
[23:16]	RO	bnk14_cmd_cnt	统计 DMC 中当前通道属于 bank14 的命令个数。	0x00



Bits	Access	Name	Description	Reset
[15:8]	RO	bnk13_cmd_cnt	统计 DMC 中当前通道属于 bank13 的命令个数。	0x00
[7:0]	RO	bnk12_cmd_cnt	统计 DMC 中当前通道属于 bank12 的命令个数。	0x00

QOSB_OSTD_CNT

QOSB_OSTD_CNT 为 QOSBUF 中当前每个通道有多少 Outstanding 命令的统计寄存器。

Offset Address: 0x1E0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	保留。	0x0000
[15:8]	RO	ch1_cmd_ostd	QosBuf 中当前通道属于 channel1 的 Outstanding 命令个数。	0x00
[7:0]	RO	ch0_cmd_ostd	QosBuf 中当前通道属于 channel0 的 Outstanding 命令个数。	0x00

QOSB_WR_CMD_SUM

QOSB_WR_CMD_SUM 为 QosBuf 写命令计数的累积寄存器。

Offset Address: 0x1E4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	qos_wr_cmd_sum	当前 QosBuf 暂存写命令的值按周期累积，溢出卷绕。	0x00000000

QOSB_RD_CMD_SUM

QOSB_RD_CMD_SUM 为 QosBuf 读命令计数的累积寄存器。

Offset Address: 0x1E8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	qos_rd_cmd_sum	当前 QosBuf 暂存读命令的值按周期累积，溢出卷绕。	0x00000000



QOSB_TIMEOUT_MODE

QOSB_TIMEOUT_MODE 为 QOSBUF 中 timeout 模式选择寄存器。

Offset Address: 0x1F0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留。	0x00000000
[0]	RW	timeout_mode	读写命令的 Timeout 映射模式选择寄存器。 0: 表示通过 ID 来映射, 共 16 级(每一级定义在 QOSB_WRTOUT0~QOSB_WRTOUT3 或者 QOSB_RDTOUT0~QOSB_RDTOUT3); 1: 表示通过优先级来映射, 共 8 级。其中映射表与 ID 的映射表的低 8 级相同。	0x0

QOSB_RDTOUT0

QOSB_RDTOUT0 为 QosBuf 的读命令 timeout 配置寄存器 0。

Offset Address: 0x300 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	RO	reserved	保留。	0x0
[29:20]	RW	rd_tout2	读命令 timeout 配置(档位 2)。 0x0: 禁止 timeout 功能; 0x1~0xFF: n*4 个时钟周期; 注意: timeout 配置为 10bits, 但实际计数为 12bits, 即低两位固定为零。	0x000
[19:10]	RW	rd_tout1	读命令 timeout 配置(档位 1)。 0x0: 禁止 timeout 功能; 0x1~0xFF: n*4 个时钟周期; 注意: timeout 配置为 10bits, 但实际计数为 12bits, 即低两位固定为零。	0x000
[9:0]	RW	rd_tout0	读命令 timeout 配置(档位 0)。 0x0: 禁止 timeout 功能; 0x1~0xFF: n*4 个时钟周期; 注意:	0x000



Bits	Access	Name	Description	Reset
			timeout 配置为 10bits，但实际计数为 12bits，即低两位固定为零。	

QOSB_RDTOUT1

QOSB_RDTOUT1 为 QosBuf 的读命令 timeout 配置寄存器 1。

Offset Address: 0x304 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	RO	reserved	保留。	0x0
[29:20]	RW	rd_tout5	读命令 timeout 配置(档位 5)。 配置方式同 rd_tout0。	0x000
[19:10]	RW	rd_tout4	读命令 timeout 配置(档位 4)。 配置方式同 rd_tout0。	0x000
[9:0]	RW	rd_tout3	读命令 timeout 配置(档位 3)。 配置方式同 rd_tout0。	0x000

QOSB_RDTOUT2

QOSB_RDTOUT2 为 QosBuf 的读命令 timeout 配置寄存器 2。

Offset Address: 0x308 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	RO	reserved	保留。	0x0
[29:20]	RW	rd_tout8	读命令 timeout 配置(档位 8)。 配置方式同 rd_tout0。	0x000
[19:10]	RW	rd_tout7	读命令 timeout 配置(档位 7)。 配置方式同 rd_tout0。	0x000
[9:0]	RW	rd_tout6	读命令 timeout 配置(档位 6)。 配置方式同 rd_tout0。	0x000

QOSB_RDTOUT3

QOSB_RDTOUT3 为 QosBuf 的读命令 timeout 配置寄存器 3。



Offset Address: 0x30C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	RO	reserved	保留。	0x0
[29:20]	RW	rd_tout11	读命令 timeout 配置(档位 11)。 配置方式同 rd_tout0。	0x000
[19:10]	RW	rd_tout10	读命令 timeout 配置(档位 10)。 配置方式同 rd_tout0。	0x000
[9:0]	RW	rd_tout9	读命令 timeout 配置(档位 9)。 配置方式同 rd_tout0。	0x000

QOSB_RDTOUT4

QOSB_RDTOUT3 为 QosBuf 的读命令 timeout 配置寄存器 4。

Offset Address: 0x310 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	RO	reserved	保留。	0x0
[29:20]	RW	rd_tout14	读命令 timeout 配置(档位 14)。 配置方式同 rd_tout0。	0x000
[19:10]	RW	rd_tout13	读命令 timeout 配置(档位 13)。 配置方式同 rd_tout0。	0x000
[9:0]	RW	rd_tout12	读命令 timeout 配置(档位 12)。 配置方式同 rd_tout0。	0x000

QOSB_RDTOUT5

QOSB_RDTOUT5 为 QosBuf 的读命令 timeout 配置寄存器 5。

Offset Address: 0x314 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	RO	reserved	保留。	0x0
[9:0]	RW	rd_tout15	读命令 timeout 配置(档位 15)。 配置方式同 rd_tout0。	0x000



4.1.8 DMC 寄存器

4.1.8.1 DMC 寄存器概览

DMC 寄存器概览如表 4-9 所示。

表4-9 DMC 寄存器概览（DMC 基地址：0x1206_8000）

偏移地址	名称	描述	页码
0x000	DDRC_CTRL_SREF	DDR 自刷新控制寄存器	4-52
0x020	DDRC_CRG_SREF	DDR 自刷新配置寄存器	4-52
0x028	DDRC_CFG_PD	DDR 低功耗 PowerDown 状态配置寄存器	4-53
0x02C	DDRC_CFG_AREF	DDRC 自动刷新模式配置寄存器	4-54
0x040	DDRC_CFG_WORKMODE	DDRC 工作模式配置寄存器	4-56
0x044	DDRC_CFG_WORKMODE2	DDRC 工作模式配置寄存器	4-57
0x050	DDRC_DDRMODE	DDR 工作模式配置寄存器	4-57
0x058	DDRC_CFG_SCRAMB	DDR 数据加扰配置寄存器	4-61
0x060	DDRC_CFG_RNKVOLn	DDRC 控制的每个 RANK 的 DDR 容量配置寄存器	4-62
0x0A0	DDRC_CFG_ODT	ODT 特性配置寄存器	4-63
0x100	DDRC_CFG_TIMING0	DDRC 的时序参数寄存器 0	4-63
0x104	DDRC_CFG_TIMING1	DDRC 的时序参数寄存器 1	4-64
0x108	DDRC_CFG_TIMING2	DDRC 的时序参数寄存器 2	4-65
0x10C	DDRC_CFG_TIMING3	DDRC 的时序参数寄存器 3	4-66
0x110	DDRC_CFG_TIMING4	DDRC 的时序参数寄存器 4	4-67
0x114	DDRC_CFG_TIMING5	DDRC 的时序参数寄存器 5	4-67
0x118	DDRC_CFG_TIMING6	DDRC 的时序参数寄存器 6	4-68
0x11C	DDRC_CFG_TIMING7	DDRC 的时序参数寄存器 7	4-69



偏移地址	名称	描述	页码
0x140	DDRC_CFG_BLDATA	DDRC 写数据预接收写数据配置寄存器	4-70
0x144	DDRC_CFG_DMCLVL	DDR 控制器命令队列深度流水线配置寄存器	4-70
0x200	DDRC_CFG_DDRPHY	DDR IO 配置寄存器	4-70

4.1.8.2 DMC 寄存器描述

DDRC_CTRL_SREF

DDRC_CTRL_SREF 为 DDR 自刷新控制寄存器。

Offset Address: 0x000 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:2]	RO	reserved	保留。	0x00000000
[1]	RW	sref_done	DDR PHY 自刷新操作完成。 0: 正常工作状态; 1: 配置从 0 到 1 的跳变表示 DDR PHY 完成所有自刷新退出后必须的操作, DMC 可以接受新的请求。	0x0
[0]	RW	sref_req	SDRAM 自刷新请求。 0: 退出自刷新状态; 1: 进入自刷新状态。	0x1

DDRC_CFG_SREF

DDRC_CFG_SREF 为 DDR 自刷新配置寄存器。

Offset Address: 0x020 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	asref_en	SDRAM 自动进入 SELF REFRESH 使能。 1: 使能; 0: 禁止。 注意, 每个 RANK 对应一个控制位, 当 DDRC_CFG_PD[PD_EN]有效, 且 RANK 对应的 asref_en 有效时, 这个 RANK 才能	0x0000



Bits	Access	Name	Description	Reset
			自动进入 SELF REFRESH。 当前版本不支持该功能。	
[15:12]	RW	sref_arefnum	DDR4 SDRAM 在 DFS 流程中，退出自刷新后发出的自动刷新操作次数。 0x0: 不发送自动刷新操作。 0x1~0xF: n 次。	0x0
[11:9]	RO	reserved	保留。	0x0
[8]	RW	clk_switch	DDRC 低功耗时钟切换控制。DDRC 进入低功耗状态(DDR Self Refresh)状态是否反压 AXI 接口命令。 0: 反压接口命令，同时将原有命令执行完成； 1: 反压接口命令，时钟切换完成后，继续执行原有命令。	0x0
[7:5]	RO	reserved	保留。	0x0
[4]	RW	sref_odis	自刷新模式，DDR 命令和数据 IO 的输出关断配置。 0: 不关断管脚输出； 1: 关断管脚输出。 注意：此配置为静态配置。建议当配置 DDR 进入 self_refresh 后，可配置为 1，关断 DDR 命令和数据 IO 的输出。在配置 DDR 退出 self_refresh 前，配置为 0，打开 DDR 命令和数据 IO 的输出。	0x0
[3:2]	RO	reserved	保留。	0x0
[1]	RW	asref_fast_wakeup	AUTO SELFREFRESH 快速唤醒使能。 0: 禁止； 1: 使能。	0x0
[0]	RW	sref_cc	自刷新模式，SDRAM 时钟控制。 0: 不关断 SDRAM 时钟； 1: 关断 SDRAM 时钟。	0x0

DDRC_CFG_PD

DDRC_CFG_PD 为 DDR 低功耗 PowerDown 状态配置寄存器。

Offset Address: 0x028 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:20]	RO	asref_prd	SDRAM 低功耗(Self-refresh)周期配置。当 DDRC 在连续的 asref_prd 个周期内没有接受到命令，则控制 SDRAM 进入到自刷新状态，当新命令到来，则控制 SDRAM 退出自刷新状态。 0x0~0xFFF: n*16 个时钟周期。 注意：该参数仅在 asref_en 为 1 时有效。	0x000
[19:16]	RW	t_clk_cke	CLK 与 CKE 关系设置。 0x0~0x7: DDRPHY 关闭 DDR 颗粒时钟相对 CKE 的延时，跟具体的 DDRPHY 相关。 当前版本在使能时钟关闭情况下，配置为 0 即可。	0x0
[15:4]	RW	pd_prd	SDRAM 低功耗(Power Down)周期配置。当 DDRC 在连续的 pd_prd 个周期内没有接受到命令，则控制 SDRAM 进入到低功耗状态，当新命令到来，则控制 SDRAM 退出低功耗状态。 0x00: 不进入 power down 状态； 0x01~0xFFF: n 个时钟周期。 注意：该参数仅在 pd_en 为 1 时有效。	0x000
[3]	RO	reserved	保留	0x0
[2]	RO	reserved	保留	0x0
[1]	RW	pd_ac	低功耗模式(PowerDown)，SDRAM 地址命令动态关断控制。 1: 关断管脚输出； 0: 不关断管脚输出。 注意：在 pd_en 使能下有效，控制管脚不包括 CKE、ODT、CSN 和 RESET_N。	0x0
[0]	RW	pd_en	SDRAM 自动低功耗使能。 1: 使能； 0: 禁止。	0x0

DDRC_CFG_AREF

DDRC_CFG_AREF 为 DDRC 自动刷新模式配置寄存器。

Offset Address: 0x02C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	保留。	0x0000
[15:8]	RW	aref_alarm_num	配置当无法发若干个 auto refresh 命令后，强制发出 self refresh 命令。 0x0~0xFF: 缺少 n+1 个 auto refresh。(注意，但 n=255 时，由于计数器为 8 位，则进位后高位丢失，相当于 1) 当为优化模式时，取 aref_opt=1，则当 postponed 的命令个数大于等于 15 时就会报警。	0x00
[7:5]	-	reserved	保留。	0x0
[4]	RW	aref_alarm_en	启动 AREF 功能中丢 auto refresh 的报警使能： 1: 使能； 0: 不使能。	0x0
[3]	RW	aref_dual_rank	AREF 模块发送 REF 命令的模式选择。 0: 每次只向一个 rank 发送 REF 命令 1: 每次会向两个 rank 发送 REF 命令。	0x0
[2]	RW	aref_opt	auto refresh 优化使能。 1: 使能自动刷新优化； 0: 禁止自动刷新优化。 使能该功能时，aref_mode 建议配置为 00。 DMC 会在每次 auto refresh 时动态检测 DDR 是否空闲，如果空闲，则发送 AREF 命令，否则将等待，直到 AREF 周期累积到 9 个 AREF 周期时，强行终止 DDR 访问，插入 AREF 命令。	0x0
[1:0]	RW	aref_mode	非优化自动刷新模式选择。 11: 每 9 个 tREFI 周期执行 8 次自动刷新操作； 10: 每 5 个 tREFI 周期执行 4 次自动刷新操作 01: 每 4 个 tREFI 周期执行 3 次自动刷新操作 00: 每 1 个 tREFI 周期执行 1 次自动刷新操作。 优化自动刷新模式。 11: postponed 命令水线为 8；	0x0



Bits	Access	Name	Description	Reset
			10: postponed 命令水线为 4; 01: postponed 命令水线为 2; 00: postponed 命令水线为 1;	

DDRC_CFG_WORKMODE

DDRC_CFG_WORKMODE 为 DDRC 工作模式配置寄存器。

Offset Address: 0x040 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:15]	RO	reserved	保留	0x00000
[14]	RW	wr_intlv_en	DMC BURST 写间插使能。 0: 禁止; 1: 使能。 在 DDR4 模式下, 配置为 1 可以获得最高性能, 其他模式建议配置为 0。	0x0
[13]	RW	hdr_mode	DFI 接口模式选择。 0: SDR 模式; 1: HDR 模式。 当前版本只支持 HDR 模式。	0x0
[12]	RW	read_mode	控制器读模式选择。 0: 随路读模式; 1: 延迟读模式。 随路读模式, 是指控制器根据 PHY 送出的数据有效信号完成数据采样。 延迟读模式, 是指控制器内部延迟等待完成对 PHY 送出数据的采样。	0x0
[11:10]	RW	addr_mode	DDR 地址线翻转模式选择。 00: IDLE 时命令地址全部归 1; 01: IDLE 时保持前一个命令的状态; 10: IDLE 时与前一次状态取反; 11: 保留。	0x0
[9]	RW	intlv_en	DMC BURST 读间插使能。 0: 禁止; 1: 使能。 在 DDR4 模式下, 配置为 1 可以获得最高	0x0



Bits	Access	Name	Description	Reset
			性能，其他模式建议配置为 0。	
[8:5]	RO	reserved	保留。	0x0
[4]	RW	apre_en	Auto Precharge 功能使能： 0：禁止； 1：使能。	0x0
[3]	RW	func_clkon	功能模块时钟开关。 0：内部自动控制时钟； 1：强制打开时钟。	0x0
[2]	RW	data_clkon	数据通道时钟开关。 0：内部自动控制时钟； 1：强制打开时钟。	0x0
[1]	RW	cmd_clkon	命令通道时钟开关。 0：内部自动控制时钟； 1：强制打开时钟。	0x0
[0]	RW	clk_ratio	控制器工作模式。 1：DDRC 与 PHY 频率比为 1：2； 0：DDRC 与 PHY 频率比为 1：1。 注意：当前版本需固定为 1。	0x0

DDRC_CFG_WORKMODE2

DDRC_CFG_WORKMODE2 为 DDRC 工作模式配置寄存器。

Offset Address: 0x044 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	保留。	0x0000
[15:0]	RW	rank_disable	RANK 禁止开关。 0：正常工作模式； 1：禁止某个 RANK 的 AUTO REFRESH/POWER DOWN/SELFREFRESH 功能。	0x0000

DDRC_CFG_DDRMODE

DDRC_CFG_DDRMODE 为 DDRC 工作模式配置寄存器。



Offset Address: 0x050 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30:28]	RW	bank_mode	BANK 交织模式。 存储总线位宽为 8bit 时： 000: 8Byte 交织； 001: 16Byte 交织； ... 111: 1kByte 交织。 存储总线位宽为 16bit 时： 000: 16Byte 交织； 001: 32Byte 交织； ... 111: 2kByte 交织。 存储总线位宽为 32bit 时： 000: 32Byte 交织； 001: 64Byte 交织； ... 111: 4kByte 交织。 存储总线位宽为 64bit 时： 000: 64Byte 交织； 001: 128Byte 交织； ... 111: 8kByte 交织。 注意： 当采用寄存器 AXI_REGION_ATTRIB [bnk_mod]来映射 ddr 的 bank 地址时，这个域的配置将不会生效。	0x0
[27:26]	-	reserved	保留。	0x0
[25:24]	RW	bank_xor	BANK 地址与 ROW 地址加扰使能。 00: 禁止； 01: BANK 与偏移地址 0 加扰； 10: BANK 与偏移地址 1 加扰； 11: BANK 与偏移地址 2 加扰；	0x0
[23]	RW	asref_zqc_en	ASREF 退出时的 SDRAM ZQ 使能。 0: 禁止；	0x0



Bits	Access	Name	Description	Reset
			1: 使能。	
[22]	RW	sref_zqc_en	SREF 退出时的 SDRAM ZQ 使能。 0: 禁止; 1: 使能。 注意: 仅对 DDR4 SDRAM 有效, 默认配置为 0。另外注意, 该配置位不允许在退出自刷新过程中动态变化。	0x0
[21:20]	RW	rank	控制器 RANK 配置。 00: 1 个 RANK; 01: 2 个 RANK; 10: 3 个 RANK; 11: 4 个 RANK。 注意: 当外接 DDR 器件大于 4 个 RANK 时, 此控制位同时用于 0-3,4-7,8-11,12-16 四组 RANK 的控制	0x0
[19:18]	RW	rank_mode	RANK 交织模式。 00: 单 RANK 模式; 01: DIMM 交织模式; 10: 双 RANK 交织; 11: 四 RANK 交织。 注意: 多 RANK 交织只有在 RANK 的配置完全相同时才能使用。否则将会出现不可预知的错误。	0x0
[17]	RW	odt_on	输出给 SDRAM 的 ODT 信号为固定值。 0: 由 DDRC 自动控制; 1: 固定输出为 rank0 的 wodt 配置。	0x0
[16]	RW	zqc_en	Aref 模块 ZQ 命令使能。 0: 禁止; 1: 使能。	0x0
[15]	-	reserved	保留。	0x0
[14]	RW	scramb_en	数据加扰使能(使用地址作为种子产生扰码, 然后与数据进行异或, 以减少同步翻转)。 0: 禁止; 1: 使能。	0x0
[13]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[12]	-	reserved	保留。	0x0
[11]	RO	reserved	保留。必须保持 1' b0。	0x0
[10]	RW	rd_2t_pre	DDR4 读方向 2T PREAMBLE 使能。 0: 禁止; 1: 使能。	0x0
[9]	RW	wr_2t_pre	DDR4 写方向 2T PREAMBLE 使能。 0: 禁止; 1: 使能。	0x0
[8]	RW	brstlen	控制器 Burst Length 配置。 当 brstlen2 配置为 0 时, 1: BL8; 0: BL4。 当 brstlen2 配置为 1 时, 1: BL32; 0: BL16。 DDR4 只能配置为 Burst8 模式。 注意: 这里的 burst len 与 dmc 与 phy 的 clk ratio 有关系。其配置值应该是 BLx/n(其中 x 表示 4/8/16/32 这几种 burst 类型, n 表示 dmc 与 phy 的时钟关系为 1:n)。以 clk ratio=1: 2 为例, 如果要配置 ddr bl32,只需要配置 brstlen=0;brstlen2=1 即可。	0x0
[7]	RW	brstlen2	控制器 Burst Length2 配置。 1: 扩展模式; 0: 基本模式。 与 brstlen 共同作用。	0x0
[6]	-	reserved	保留。	0x0
[5:4]	RW	mem_width	存储数据总线位宽。 00: 8bit; 01: 16bit; 10: 32bit; 11: 64bit。	0x0
[3:0]	RW	dram_type	外部存储器类型。 0110: DDR3; 0111: DDR4;	0x0



Bits	Access	Name	Description	Reset
			其他：保留。	

DDRC_CFG_SCRAMB

DDRC_CFG_SCRAMB 为 DDR 数据加扰配置寄存器。

Offset Address: 0x058 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:19]	-	reserved	保留。	0x0000
[18]	RW	dbi_low_act	DBI 有效电平。 0: 高有效; 1: 低有效。	0x0
[17]	RW	rd_dbi_en	读方向 DBI 使能。 0: 禁止; 1: 使能。	0x0
[16]	RW	wr_dbi_en	写方向 DBI 使能。 0: 禁止; 1: 使能。	0x0
[15:5]	-	reserved	保留。	0x000
[4]	RW	scramb_seed_type	用于加扰的地址模式。 0: 用片选和 BA 加扰; 1: 用片选、BA、和低位列地址加扰。	0x0
[3]	RO	reserved	保留。	0x0
[2:0]	RW	scramb_seed_sort	用于加扰的地址排列方式。 000: CS_BA_COL 排列; 001: CS_COL_BA 排列; 010: COL_CS_BA 排列; 011: COL_BA_CS 排列; 100: BA_CS_COL 排列; 101: BA_COL_CS 排列; 其他：保留。	0x0



DDRC_CFG_RNKVOLn

DDRC_CFG_RNKVOLn 为 DDRC 控制的每个 RANK 的 DDR 容量配置寄存器。

Offset Address: 0x060 Total Reset Value: 0x0000_0022

Bits	Access	Name	Description	Reset
[31:17]	RO	reserved	保留。	0x0000
[16]	RW	mem_x4	外接器件为 4bit 器件拼接模式。 0: 外部由 8/16/32bit 器件拼接组成; 1: 外部由 4bit 器件拼接组成。	0x0
[15:14]	RO	reserved	保留。	0x0
[13:12]	RW	mem_map	SDRAM 地址译码模式。 00: {Rank,Row,Ba,Col,DW}=AXI_Address; 01: {Rank,Ba,Row,Col,DW}=AXI_Address; 10: {Rank,Row,Ba,Col,cs,Col,DW}=AXI_Address; 11: {Rank,Ba,Row,Col,cs,Col,DW}=AXI_Address。 该参数, 需要在 DDRC_CONFIG1[dual_ch] 有效时, 才能配置为'b10 或者'b11。 当有多个 rank 时, 各个 rank 必须使用相同的配置。 当前版本只支持 00 配置。	0x0
[11:10]	RW	mem_bankgroup	单片 SDRAM Bank Group 数。 00: 1 个 Bank Group; 01: 2 个 Bank Group; 10: 4 个 Bank Group; 11: 保留。 注意: 只有在 DDR4 模式下有效。	0x0
[9:8]	RW	mem_bank	单片 SDRAM Bank 数。 00: 4 Bank; 01: 8 Bank; 10: 16 Bank; 11: 保留。	0x0
[7]	RO	reserved	保留。	0x0
[6:4]	RW	mem_row	单片 SDRAM 行地址位宽配置。	0x2



Bits	Access	Name	Description	Reset
			000: 11 bit; 001: 12 bit; 010: 13 bit; 011: 14 bit; 100: 15 bit; 101: 16 bit; 其它: 保留。	
[3]	RO	reserved	保留。	0x0
[2:0]	RW	mem_col	单片 SDRAM 列地址位宽配置。 000: 8 bit; 001: 9 bit; 010: 10 bit; 011: 11 bit; 100: 12 bit。 其它: 保留。	0x2

DDRC_CFG_ODT

DDRC_CFG_ODT 为 DDR 的 ODT 特性配置寄存器。

Offset Address: 0x0A0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	rodt	当前 RANK 发送读命令时, 其他 RANK 的 ODT 配置。 1: 使能读 ODT; 0: 禁止读 ODT。	0x0000
[15:0]	RW	wodt	当前 RANK 发送写命令时, 其他 RANK 的 ODT 配置。 1: 使能写 ODT; 0: 禁止写 ODT。	0x0000

DDRC_CFG_TIMING0

DDRC_CFG_TIMING0 为 DDRC 的时序参数寄存器 0。

Offset Address: 0x100 Total Reset Value: 0xFFFF_FF3F



Bits	Access	Name	Description	Reset
[31:28]	RW	tmrdr	模式寄存器加载(LMR)命令的等待周期(在DDR4 时配置为 tMRD)。 0x0~0x1: 1 个时钟周期; 0x2~0xF: n 个时钟周期。	0xF
[27:24]	RW	trrd	打开 BANK A 到打开 BANK B 的(ACT bank a to ACT bank b)等待周期。 0x0~0x1: 1 个时钟周期; 0x2~0xF: n 个时钟周期。	0xF
[23:19]	RW	trp_pb	Per Bank 关闭(PRE period)命令的等待周期。 0x0~0x1: 1 个时钟周期; 0x2~0xF: n 个时钟周期。	0x1F
[18:14]	RW	trcd	同 BANK 激活到读或写(ACT to READ or WRITE)命令的等待周期。 0x0~0x3: 3 时钟周期; 0x4~0xF: n 时钟周期。	0x1F
[13:8]	RW	trc	同 BANK 的激活命令到激活命令(active a bank to active a bank)的等待周期。 0x00~0x01: 1 个时钟周期; 0x02~0x3F: n 个时钟周期。	0x3F
[7:6]	RO	reserved	保留。	0x0
[5:0]	RW	tras	同 BANK 的激活命令到关闭命令(ACT to PRE)的等待周期。本产品支持最大配置为 0xF。 0x00~0x01: 1 个时钟周期; 0x02~0x0F: n 个时钟周期。	0x3F

DDRC_CFG_TIMING1

DDRC_CFG_TIMING1 为 DDRC 的时序参数寄存器 1。

Offset Address: 0x104 Total Reset Value: 0xFF22_15FF

Bits	Access	Name	Description	Reset
[31:24]	RW	tsre	退出自刷新(Self-Refresh)到读命令的等待周期。 0x0: 保留;	0xFF



Bits	Access	Name	Description	Reset
			0x01~0xFF: n×4 个时钟周期。 DDR4 SDRAM 时, 该值配置 tXSDLL 和 tXS 的最大值。	
[23:20]	RW	trtw	最后一个读数据到第一个写数据延迟。 0x0~0x1: 1 个时钟周期; 0x2~0xF: n+1 个时钟周期。 注意: trtw 参数主要考虑单板、封装、IO 的延时。	0x2
[19]	-	reserved	保留。	0x0
[18:14]	-	reserved	保留。	0x0
[13:8]	RW	twl	写命令到写数据的等待周期。 0x0~0x1: 1 个时钟周期; 0x2~0xF: n 个时钟周期。 如: 0x3: 3 个时钟周期。	0x04
[7:6]	RO	reserved	保留。	0x0
[5:0]	RW	tcl	DDR 的读命令到读数据的延迟 (CAS Latency)。 0x0~0x1: 1 个时钟周期; 0x2~0xF: n 个时钟周期。	0x1FF

DDRC_CFG_TIMING2

DDRC_CFG_TIMING2 为 DDRC 的时序参数寄存器 2。

Offset Address: 0x108 Total Reset Value: 0xF303_F000

Bits	Access	Name	Description	Reset
[31:28]	RW	tcke	低功耗状态维持的最短时间。 0x0: 保留; 0x1~0xF: n 个时钟周期。 该值需要配置 tCKESR, tCKSRE, tCKSRX, tCKE, tCPDED 几个值中的最大值。	0xF
[27:24]	RW	twtr	写操作最后一个写数据到读命令(write to read)的等待周期。 0x0~0x1: 1 个时钟周期;	0x3



Bits	Access	Name	Description	Reset
			0x2~0xf: n 个时钟周期。 如 0x3: 3 个时钟周期。	
[23:18]	-	reserved	保留。	0x3F
[17:12]	RW	tfaw	连续 4 个激活命令周期。 0x00~0x3F: n 个时钟周期; 如: 0x14: 20 个时钟周期。	0x3F
[11]	-	reserved	保留。	0x0
[10:0]	RW	taoref	自动刷新周期。 0x000: 自动刷新禁止; 0x001~0x7FF: SDRAM 刷新周期时间为 16×n 时钟周期。 如 0x008: 128 个时钟周期(16×8)。 配置间隔时间为 tREFI = 7800/16/tclk。Tclk 为 DDR 的运行周期的 2 倍。	0x000

DDRC_CFG_TIMING3

DDRC_CFG_TIMING3 为 DDRC 的时序参数寄存器 3。

Offset Address: 0x10C Total Reset Value: 0xFFFF_E0F2

Bits	Access	Name	Description	Reset
[31:22]	RW	tzq_prd	ZQCS 命令周期。 0x000: ZQCS 命令禁止; 0x001~0x3FF: n×128 个 AREF 周期。 ZQCS 命令周期时间为 n×128 个 taref 时钟周期。	0x3FF
[21:13]	RW	tzqinit	ZQ 初始化延迟周期。 0x0~0x1ff: nx4 个时钟周期。 该值配置 tZQINIT, tDLLK 的最大值。	0x1FF
[12:8]	RW	taond	ODT(On-die termination)打开和关闭周期。 在 DDR4 模式下, 该值配置为 tWL-2(tWL 为 DDRC_CFG_TIMING1[twl])。	0x00
[7:4]	RW	txard	退出 DDR 低功耗状态等待周期。 0x0~0xF: n 个时钟周期, n 代表十进制数; 如: 0x7: 为 7 个时钟周期。	0xF



Bits	Access	Name	Description	Reset
			取 {tXP,tXARD,tXARDS,tXS} 中的最大值。	
[3:0]	RW	trtp	读命令到关闭命令的等待延迟。 0x0~0x2: 2 个时钟周期; 0x3~0xF: n 个时钟周期。 Trtp 的计算公式是 $AL+BL/2+Max(trtp,2)-2$ 。	0x2

DDRC_CFG_TIMING4

DDRC_CFG_TIMING4 为 DDRC 的时序参数寄存器 4。

Offset Address: 0x110 Total Reset Value: 0x01FF_2018

Bits	Access	Name	Description	Reset
[31:25]	-	reserved	保留。	0x00
[24:20]	RW	tmod	MRS 命令到 ODT 和 ZQCL 有效的延时参数。 0x0~0x1: 1 个时钟周期; 02~0x1F: n 个时钟周期。	0x1F
[19:0]	-	reserved	保留。	0xF2018

DDRC_CFG_TIMING5

DDRC_CFG_TIMING5 为 DDRC 的时序参数寄存器 5。

Offset Address: 0x114 Total Reset Value: 0x1113_FF1F

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x1113
[15:8]	RW	tzqcs	ZQCS 校准延迟周期。 0x0~0xff: n*2 个时钟周期。 由于 dmc 的设计, 该值需要约束为大于等于 10	0xFF
[7:5]	-	reserved	保留。	0x0
[4:0]	RW	twr	写恢复(write recovery)等待周期。 0x0~0x1: 1 个时钟周期; 0x2~0x1f: n 个时钟周期。 注意: 在有 DFS 的需求下, tWR 参数需要按照 DFS 中芯片可能使用最高频率配置。并且不能随 DDR 的频率变化修改 tWR 的配置。	0x1F



DDRC_CFG_TIMING6

DDRC_CFG_TIMING6 为 DDRC 的时序参数寄存器 6。

Offset Address: 0x118 Total Reset Value: 0x0000_00FF

Bits	Access	Name	Description	Reset
[31:28]	RW	tccd_nsw	在使用 2T preamble 或者 2T postamble 的情况下，两个非连续的写操作之间的最小间隔。 0x0~0x1: 1 个时钟周期; 0x2~0xf: n 个时钟周期。	0x0
[27:24]	RW	tccd_nsr	在使用 2T preamble 或者 2T postamble 的情况下，两个非连续的读操作之间的最小间隔。 0x0~0x1: 1 个时钟周期; 0x2~0xf: n 个时钟周期。	0x0
[23:20]	RW	trrd_l	Bank Group 之间的 trrd 等待周期。 0x0~0x1: 1 个时钟周期 0x2~0xf: n 个时钟周期。	0x0
[19:16]	RW	twtr_l	Bank Group 之间的 twtr 等待周期。 0x0~0x1: 1 个时钟周期 0x2~0xf: n 个时钟周期。	0x0
[15:12]	RW	tccd_l	Bank Group 之间的 tccd 等待周期。 0x0~0x1: 1 个时钟周期 0x2~0xf: n 个时钟周期。	0x0
[11:8]	RO	reserved	保留。	0x0
[7:4]	RW	tcksrx	DDR4 的 tCKSRX 参数。在退出自刷新之前时钟需要提前的有效拍数。 0x0~0x1: 1 个时钟周期; 0x2~0xF: n 个时钟周期。	0xF
[3:0]	RW	tcksre	DDR4 的 tCKSRE 参数。在进入自刷新后时钟需要保持的拍数。 0x0~0x1: 1 个时钟周期; 0x2~0xF: n 个时钟周期。	0xF



DDRC_CFG_TIMING7

DDRC_CFG_TIMING7 为 DDRC 的时序参数寄存器 7。

Offset Address: 0x11c Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27:24]	-	reserved	保留。	0x0
[23:20]	RW	dimm_trtr	DIMM 间读到读切换基于 rnk_trtr 额外增加的延时，用于匹配 DIMM 间的最大 skew。	0x0
[19:16]	RW	dimm_twtw	DIMM 间写到写切换基于 rnk_twtw 额外增加的延时，用于匹配 DIMM 间的最大 skew。	0x0
[15:12]	RW	rnk_trtr	RANK 间读到读的延时。	0x0
[11:8]	RW	rnk_twtw	RANK 间写到写的延时。	0x0
[7:4]	RW	rnk_trtw	RANK 间读到写的延时。	0x0
[3:0]	RW	rnk_twttr	RANK 间写到读的延时。	0x0

DDRC_CFG_TIMING8

DDRC_CFG_TIMING8 为 DDRC 的时序参数寄存器 8。

Offset Address: 0x120 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27:19]	RW	trfc_ab	自动刷新命令周期和自动刷新到激活（AREF period or AREF to ACT）命令的等待周期。该寄存器选择配置 max{trfc,tzqcs} 的大值。 0x00：保留； 0x01~0x1FF：n 个时钟周期。	0x0
[18:10]	RW	trfc_pb	per bank 的自刷新周期时间参数。	0x0
[9:5]	-	reserved	保留。	0x0
[4:0]	RW	trp_ab	all bank 的 precharge 命令的时间参数。	0x0



DDRC_CFG_BLDATA

DDRC_CFG_BLDATA 为 DDRC 写数据预接收写数据配置寄存器。

Offset Address: 0x148 Total Reset Value: 0x0000_0002

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3:0]	RW	bl_data	当前模式下，每个 DDR 命令对应的 DMC 数据拍数。 0x0~0xF: n 拍。	0x2

DDRC_CFG_DMCLVL

DDRC_CFG_DMCLVL 为 DDR 控制器命令队列深度水位配置寄存器。

Offset Address: 0x14C Total Reset Value: 0x0000_0008

Bits	Access	Name	Description	Reset
[31:13]	RO	reserved	保留。	0x00000
[12:8]	RW	mbist_que_level	MTEST 或者 MCLR 功能时，DMC 启动的命令 BUFFER 个数。 0x0: 1 个。 0x1~0x10: 1~16 个，最大值跟代码配置有关。	0x00
[7:5]	RO	reserved	保留。	0x0
[4:0]	RW	que_level	DMC 中，命令寄存器 FIFO 的深度。 0x1~0x10: n 个命令深度。 其它: 保留。	0x16

DDRC_CFG_DDRPHY

DDRC_CFG_DDRPHY 为 DDR IO 配置寄存器。

Offset Address: 0x200 Total Reset Value: 0x001F_1000

Bits	Access	Name	Description	Reset
[31:21]	RO	reserved	保留。	0x000
[20:16]	RW	wr_busy_dly	PHY 内部用作数据打拍的门控信号。 必须配置大于 WL+2。	0x1F



			对功耗不敏感的场景，可配置为 0x1F。	
[15:13]	RO	reserved	保留。	0x0
[12]	RW	phy_upden	控制器响应 DDRPHY 发起 dfi_phyupd_en。 0: 禁止； 1: 使能。	0x1
[11:8]	RW	trdlat	DDRPHY 固有延迟。 0x0~0xF: n+1 个周期。	0x0
[7:0]	RO	reserved	保留。	0x00

4.1.9 DDR PHY 寄存器

4.1.9.1 DDR PHY 寄存器概览

DDR PHY 寄存器概览如表 4-10 所示。

表4-10 DDR PHY 寄存器概览（DDR PHY 基址是 0x1206_C000）

偏移地址	名称	描述	页码
0x030	DRAMTIMER0	DDR PHY 的时序参数寄存器 0。	4-72
0x034	DRAMTIMER1	DDR PHY 的时序参数寄存器 1。	4-72
0x038	DRAMTIMER2	DDR PHY 的时序参数寄存器 2。	4-73
0x03C	DRAMTIMER3	DDR PHY 的时序参数寄存器 3。	4-73
0x040	DRAMTIMER4	DDR PHY 的时序参数寄存器 4。	4-73
0x0BC	DRAMTIMER5	DDR PHY 的时序参数寄存器 5。	4-74
0x0064	MODEREG01	SDRAM 模式寄存器。	4-75
0x0068	MODEREG23	SDRAM 模式寄存器。	4-75
0x00E0	MODEREG45	SDRAM 模式寄存器。	4-75
0x00E4	MODEREG67	SDRAM 模式寄存器。	4-75
0x0070	MISC	DDR PHY 多功能控制寄存器。	4-76



4.1.9.2 DDR PHY 寄存器描述

DRAMTIMER0

DRAMTIMER0 为 DDR PHY 的时序参数寄存器 0。

Offset Address: 0x030

Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	RW	t_rc	同 BANK 的激活命令到激活命令(active a bank to active a bank)的等待周期。单位: 4xnCK	0x00
[25:22]	RW	t_rrd	打开 BANK A 到打开 BANK B 的(ACT bank a to ACT bank b)等待周期。单位: 4xnCK	0x0
[21:16]	RW	t_ras	同 BANK 的激活命令到关闭命令(ACT to PRE)的等待周期。单位: 4xnCK	0x00
[15:12]	RW	t_rcd	同 BANK 激活到读或写(ACT to READ or WRITE)命令的等待周期。单位: 8xnCK	0x0
[11:8]	RW	t_rp	关闭(PRE period)命令的等待周期。单位: 8xnCK	0x0
[7:4]	RW	t_wtr	写操作最后一个写数据到读命令(write to read)的等待周期。单位: 4xnCK	0x0
[3:0]	RW	t_rtp	读命令到关闭命令的等待延迟。单位: 16xnCK	0x0

DRAMTIMER1

DRAMTIMER1 为 DDR PHY 的时序参数寄存器 1。

Offset Address: 0x034

Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	RO	reserved	保留。	0x0
[29:25]	RW	t_rtw	最后一个读数据到第一个写数据延迟。单位: 1xnCK	0x00
[24:16]	RW	t_rfc	自动刷新命令周期和自动刷新到激活(AREF period or AREF to ACT)命令的等待周期。单位: 2xnCK	0x000
[15]	RO	reserved	保留。	0x0
[14:9]	RW	t_faw	连续 4 个激活命令周期。单位: 4xnCK	0x00
[8:4]	RW	t_mod	MRS 命令到 ODT 和 ZQCL 有效的延时参数。	0x00
[3:0]	RW	t_mrd	模式寄存器加载(LMR)命令的等待周期。单位: 4nCK	0x0



DRAMTIMER2

DRAMTIMER2 为 DDR PHY 的时序参数寄存器 2。

Offset Address: 0x038 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	t_ccd	读命令到读命令，或者写命令到写命令的等待周期。 0 = BL/2 1 = BL/2 + 1 单位：1xnCK	0x0
[30]	RO	reserved	保留。	0x0
[29:20]	RW	t_dllk	DLL 锁定时间。单位：1xnCK	0x000
[19:16]	RW	t_cke	CKE 维持有效的最短时间。单位：4xnCK	0x0
[15]	RO	reserved	保留。	0x0
[14:10]	RW	t_xp	Power Down 退出延迟。配置值需大于等于 2。 单位：2xnCK	0x00
[9:0]	RW	t_xs	Self Refresh 退出延迟。单位：1xnCK	0x00

DRAMTIMER3

DRAMTIMER3 为 DDR PHY 的时序参数寄存器 3。

Offset Address: 0x03C Total Reset Value: 0x0210_0200

Bits	Access	Name	Description	Reset
[31:28]	RW	t_wr	写恢复(write recovery)等待周期。单位：4xnCK	0x0
[27:24]	RW	tdshtrain	VREF Training 时序参数。 单位：1xdfi_clk	0x2
[23:10]	RW	t_init5	LPDDR 器件的初始化延迟参数。 单位：1xnCK	0x0400
[9:0]	RW	t_zcal	ZQ 校准到有效命令的等待周期。 单位：2xnCK	0x200

DRAMTIMER4

DRAMTIMER4 为 DDR PHY 的时序参数寄存器 4。

Offset Address: 0x040 Total Reset Value: 0x5034_0000



Bits	Access	Name	Description	Reset
[31:28]	RW	t_ccd_ddr4	DDR4 同 bank group 内读命令到读命令和写命令到写命令的延迟周期。对应的 SDRAM 时序参数是 t_ccd_l。 单位：1xnCK	0x5
[27:24]	RW	t_odton	ODT 打开的等待周期。 单位：1xnCK	0x0
[23]	RO	reserved	保留。	0x0
[22:20]	RW	t_vref	VREF(DQ) 更新的等待周期。 单位：1xdci_clk	0x3
[19:16]	RW	t_ccd_s	DDR4 不同 bank group 内读命令到读命令和写命令到写命令的延迟周期。 单位：1xnCK	0x4
[15:12]	RW	t_odt	ODT 有效的时钟周期。 单位：1xnCK	0x0
[11:7]	RW	t_wlo	Write Leveling 输出结果的等待周期。 单位：4xnCK	0x00
[6]	RW	t_wr_ext	t_wr 时序参数配置寄存器的扩展位。	0x0
[5:0]	RW	t_wlmrd	进入 Write Leveling 模式到第一个 DQS/DQS#上升沿的等待周期。 单位：1nCK	0x00

DRAMTIMER5

DRAMTIMER5 为 DDR PHY 的时序参数寄存器 5。

Offset Address: 0x0BC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:12]	RO	reserved	保留。	0x00000
[11:8]	RW	t_odton_h	t_odton 时序参数的高 4bit 扩展位。	0x0
[7:0]	RW	t_wl	写延迟周期。 DDR4 时配置 CWL+AL; 单位：1xnCK	0x00



MODEREG01

MODEREG01 配置 SDRAM 模式寄存器。

Offset Address: 0x0064

Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	mr1	模式寄存器 1。	0x0000
[15:0]	RW	mr0	模式寄存器 0。	0x0000

MODEREG23

MODEREG23 配置 SDRAM 模式寄存器。

Offset Address: 0x0068

Total Reset Value: 0xFF0A_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	mr3	模式寄存器 3。	0xFF0A
[15:0]	RW	mr2	模式寄存器 2。	0x0000

MODEREG45

MODEREG45 配置 SDRAM 模式寄存器。

Offset Address: 0x00E0

Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	mr5	模式寄存器 5。	0x0000
[15:0]	RW	mr4	模式寄存器 4。	0x0000

MODEREG67

MODEREG67 配置 SDRAM 模式寄存器。

Offset Address: 0x00E4

Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	mr7	模式寄存器 7。	0x0000
[15:0]	RW	mr6	模式寄存器 6。	0x0000



MISC

MISC 为多功能控制寄存器。

Offset Address: 0x0070

Total Reset Value: 0x0814_0000

Bits	Access	Name	Description	Reset
[31:30]	RO	reserved	保留。	0x0
[29:27]	RW	swapwl_en	Write Leveling 的 swap 选择。 000: 无 Swap; 001: Swap 1; 010: Swap 2; 011: Swap 3; 100: Swap 4。	0x1
[26:25]	RW	swapdm_en	DM 总线的 swap 选择。 00: 无 Swap; 01: Swap 1。	0x0
[24]	RO	reserved	保留。	0x0
[23]	RW	reserved	保留。	0x0
[22:20]	RW	swap_en	DQ 总线的 swap 选择。 000: 无 Swap; 001: Swap 1; 010: Swap 2; 011: Swap 3; 100: Swap 4; 101: Swap 5。	0x1
[19]	RW	cfg_dlyupd	当配置为 1 时, 将寄存器配置的延迟参数立即作用到 DDRPHY。	0x0
[18]	RW	addr_delay	地址/命令总线增加 1T 延迟。	0x1
[17]	RW	addr_toggle	当配置为 1, 每个时钟周期都切换地址总线。	0x0
[16]	RW	scramb_en	数据加扰开关。 1: 打开; 0: 关闭。	0x0
[15:8]	RW	cfg_rl	DDRPHY 读延迟参数。 DDR4: CL+AL-4	0x00
[7:0]	RW	cfg_wl	DDRPHY 写延迟参数。	0x00



Bits	Access	Name	Description	Reset
			DDR4: CWL+AL-1	

4.2 Flash Memory 控制器

4.2.1 概述

FMC（Flash Memory Controller）提供存储控制器接口连接片外 SPI NAND Flash 或者 SPI NOR Flash，从而完成数据的存取。

4.2.2 特点

FMC 的主要特点如下：

- 提供一个 4KB+256Byte 片内缓存，提高读取速度。
- 支持外接 1 片选（外接器件 SPI NAND Flash 或者 SPI NOR Flash）。
- 支持 SPI NOR Flash、SPI NAND Flash 两种类型的器件。
- 支持 Standard SPI、Dual-Output/Dual-Input SPI、Quad-Output/Quad-Input SPI、Dual I/O SPI、Quad I/O SPI 五种 SPI 接口类型。
- 支持 SPI NOR DTR /STR 模式的读数据。
- 支持 SPI NAND DTR /STR 模式的读数据。
- 支持多种规格的 SPI NAND Flash 器件。
 - 支持页大小 2K、4K 的器件。
 - 支持块大小 64 Pages/Block、128 Pages/Block 的器件。
- 支持 SPI NOR Flash、SPI NAND Flash 的 CS0 的 BOOT 功能；
 - 支持 1MB 的 Boot 空间。
 - 支持 Boot 模式下，控制器先自动发送器件的复位命令再进行读数据操作（仅在 SPI NAND Flash 模式下发送复位命令）。
 - 支持自动跳坏块功能（SPI NAND Flash），最大可以跳过连续 4 个坏块。
 - 支持自适应 Boot 功能（SPI NAND Flash），控制器自动找到正确的 Page-size、ECC 类型、Block-size 配置。
 - 对 SPI NAND Flash 支持 1 线和 4 线 Boot；对 SPI NOR Flash 仅支持 1 线 Boot。
 - 对 SPI NOR Flash 支持 3Byte 地址器件和 4Byte 地址器件的 Boot。
- 支持 SPI NOR Flash、SPI NAND Flash 的 DMA 读写功能；
 - SPI NAND Flash 支持内置 DMA 模式读写操作。写操作都是以整个 page 为单位进行；读操作则支持读整个 page、只读控制信息（只读 OOB）。
 - 对 SPI NOR Flash 支持 DMA 读写操作，读写长度可配置。
- 支持寄存器操作方式手动配置命令组合，从而完成各种命令。（仅支持在 ECC0 模式下的手动配置操作）。



- 支持 ECC 功能（仅对 SPI NAND Flash）。
- 支持 7 种中断：操作完成、编程操作失败、ECC 校验可纠错、ECC 校验告警、ECC 校验不可纠错、总线操作错误、DMA 传输错误。
- 支持低功耗模式，自动时钟门控。

4.2.3 功能描述

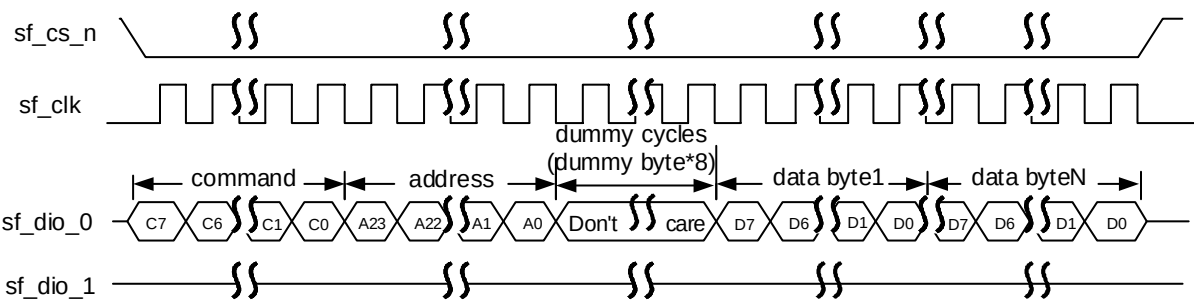
4.2.3.1 接口描述

FMC 可以支持五种 SPI 接口类型，五种类型分别为 Standard SPI、Dual-Output/Dual-Input SPI、Quad-Output/Quad-Input SPI、Dual I/O SPI、Quad I/O SPI 类型，Standard SPI、Dual I/O SPI、Quad I/O SPI 类型支持 STR、DTR 模式。

Standard SPI 模式

Standard SPI 接口模式具有 1bit 数据输入线和 1bit 数据输出线。图 4-2 为 Standard SPI 接口模式写操作时序图。

图4-2 Standard SPI 接口模式写操作时序图

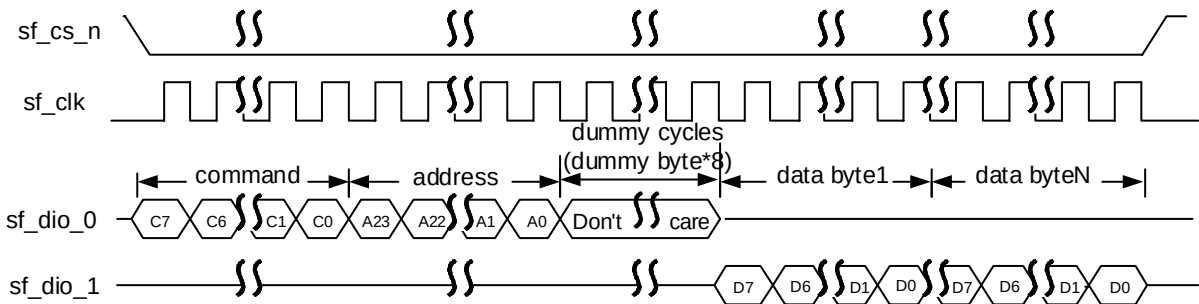


时序说明：

- command/address/dummy cycles 以单 bit 串行方式在 sf_dio_0 线上输出；
- Data 以单 bit 串行方式在 sf_dio_0 线上输出。

图 4-3 为 Standard SPI 接口模式读操作时序图。

图4-3 Standard SPI 接口模式读操作时序图。





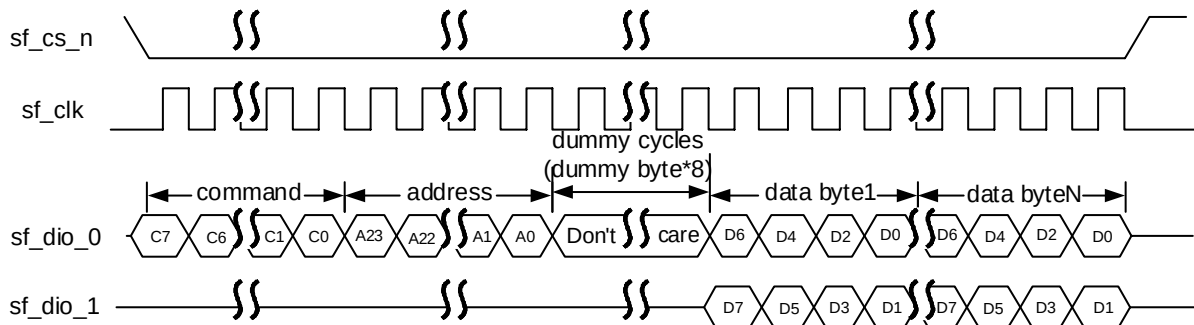
时序说明：

- command/address/dummy cycles 以单 bit 串行方式在 sf_dio_0 线上输出。
- Data 以单 bit 串行方式在 sf_dio_1 线上输入。

Dual-Output /Dual-Input SPI 模式

Dual-Output /Dual-Input SPI 接口时序图 4-4 所示。

图4-4 Dual-Output/Dual-Input SPI 接口时序图



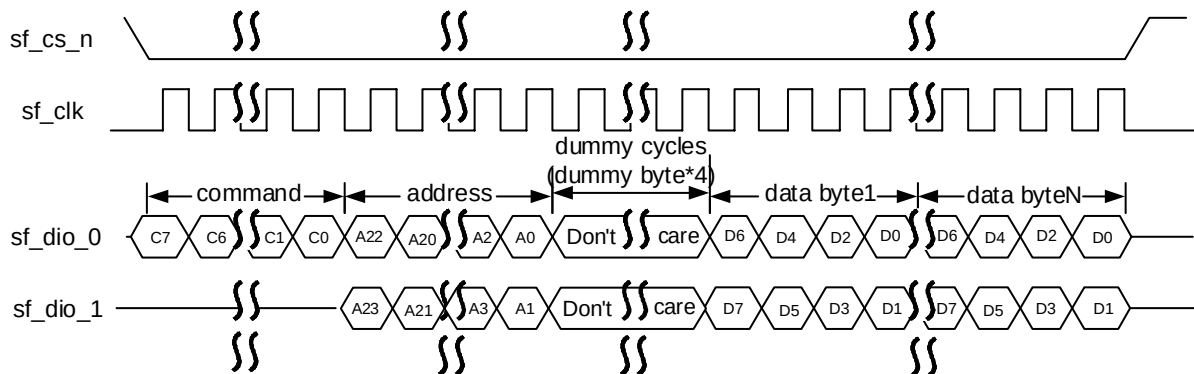
时序说明：

- command/address/dummy cycles 以单 Bit 串行方式在 sf_dio_0 线上输出。
- Data 以 2 Bits 方式在 sf_dio_0/sf_dio_1 线上输出（写）或输入（读）。

Dual I/O SPI 模式

Dual I/O 时序图 4-5 和 4-6 所示。

图4-5 Dual I/O SPI 模式 STR 时序图



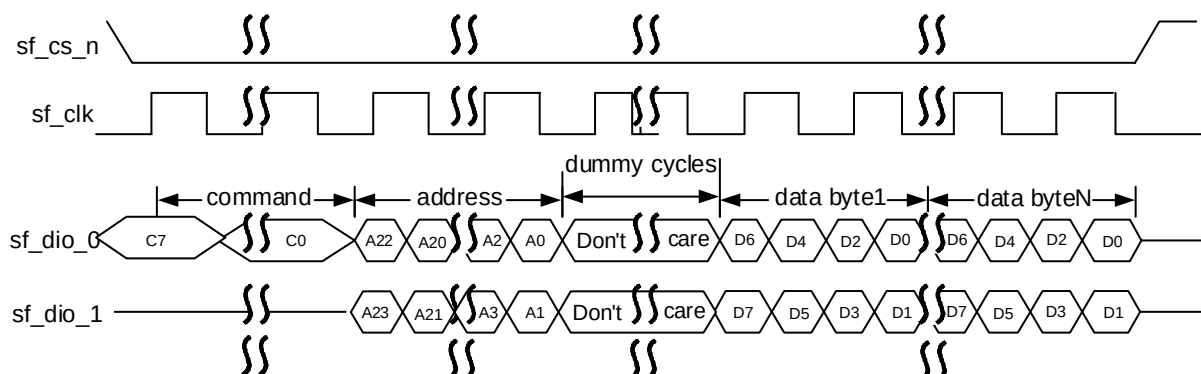
时序说明：

- command 以单 Bit 串行方式在 sf_dio_0 线上输出。
- address/dummy cycles 以 2 Bits 方式在 sf_dio_0/sf_dio_1 线上输出。



- Data 以 2 Bits 方式在 sf_dio_0/sf_dio_1 线上输出（写）或输入（读）。

图4-6 Dual I/O SPI 模式 DTR 时序图



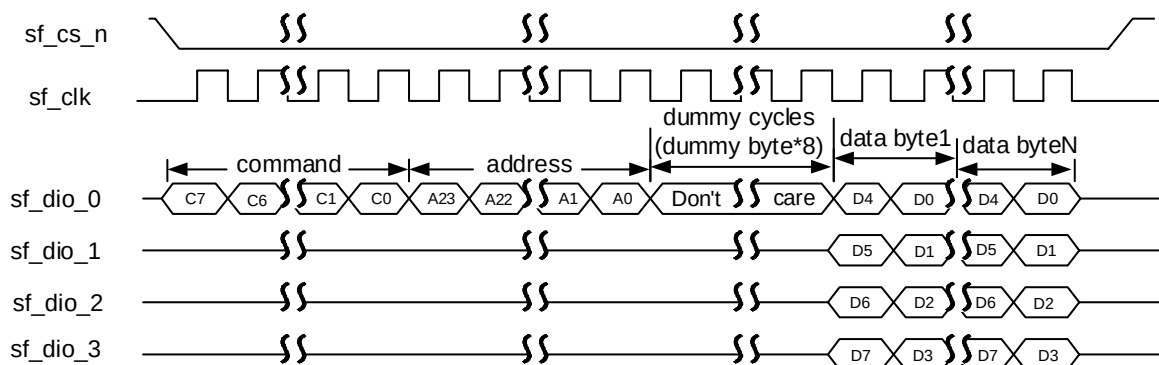
时序说明：

- command 以单 Bit 串行方式在 sf_dio_0 线上输出。
- address/dummy cycles 以 2 Bits 双沿有效方式在 sf_dio_0/sf_dio_1 线上输出。
- Data 以 2 Bits 双沿有效方式在 sf_dio_0/sf_dio_1 线上输入（读）。
- DTR 模式只支持读。

Quad-Output /Quad-Input SPI

Quad-Output / Quad-Input SPI 时序图为图 4-7 所示。

图4-7 Quad-Output/Quad-Input SPI 模式时序图



时序说明：

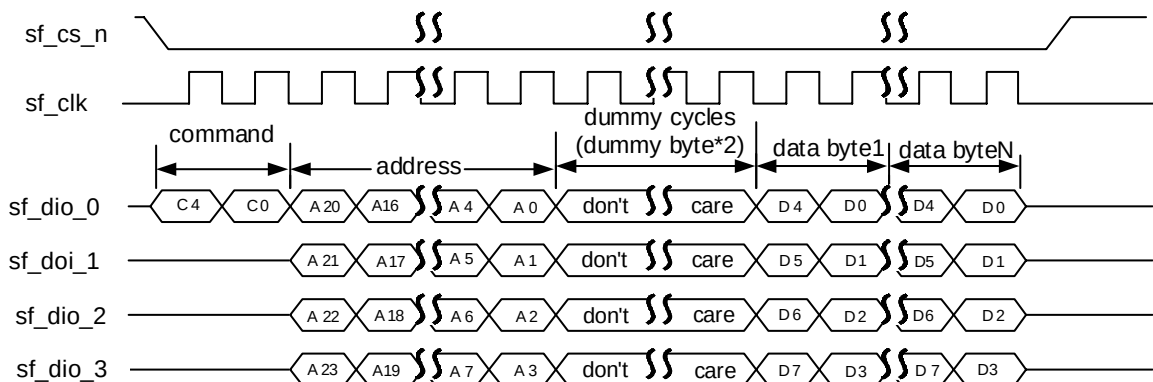
- command/address/dummy cycles 以单 Bit 串行方式在 sf_dio_0 线上输出。
- Data 以 4 Bits 方式在 sf_dio_0/sf_dio_1/sf_dio_2/sf_dio_3 线上输出（写）或输入（读）。



Quad I/O SPI 模式

Quad I/O SPI 模式时序图如图 4-8 和图 4-9 所示。

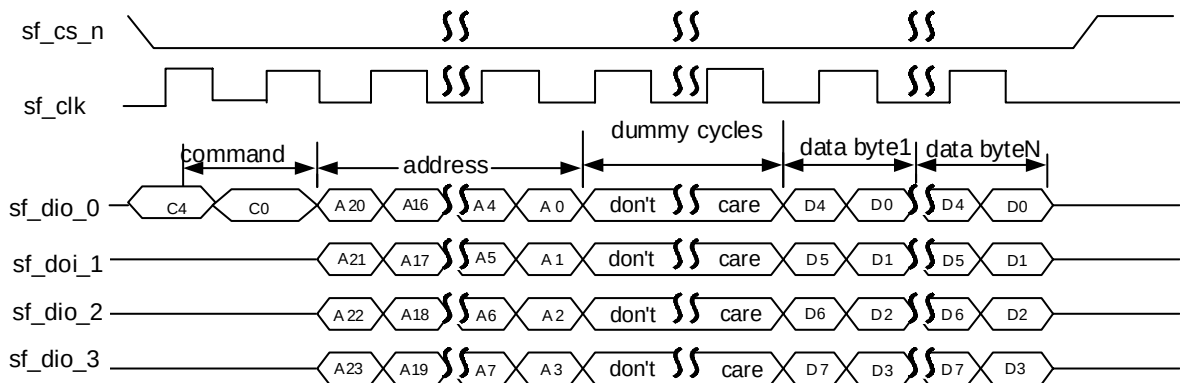
图4-8 Quad-I/O SPI 模式接口时序图



时序说明：

- command 以单 Bit 串行方式在 sf_dio_0 线上输出。
- address/dummy cycles 以 4 Bits 方式在 sf_dio_0/sf_dio_1/sf_dio_2/sf_dio_3 线上输出。
- Data 以 4 Bits 方式在 sf_dio_0/sf_dio_1/sf_dio_2/sf_dio_3 线上输出（写）或输入（读）。

图4-9 Quad-I/O SPI 模式 DTR 接口时序图



时序说明：

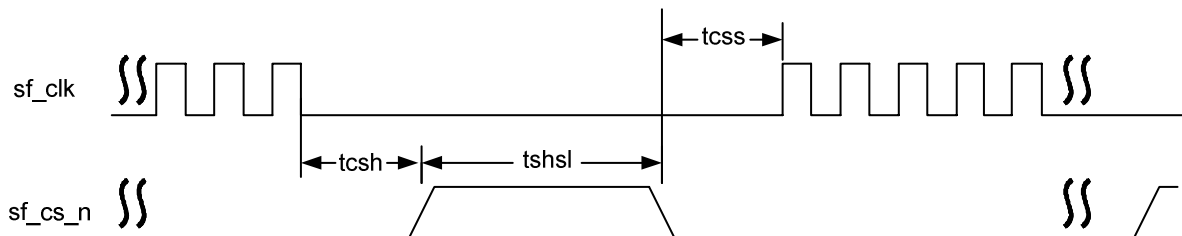
- command 以单 Bit 单沿数据串行方式在 sf_dio_0 线上输出。
- address/dummy cycles 以 4 Bits 双沿方式在 sf_dio_0/sf_dio_1/sf_dio_2/sf_dio_3 线上输出。
- Data 以 4 Bits 双沿采数方式在 sf_dio_0/sf_dio_1/sf_dio_2/sf_dio_3 线上输入。



4.2.3.2 SPI 接口时序说明

SPI 接口时序以及参数说明如图 4-10 所示。

图4-10 SPI 输出时序图



注：时序配置对应 TIMING_SPI_CFG 寄存器。

tcsh: sf_cs_n hold time。

tcss: sf_cs_n setup time。

tshsl: sf_cs_n deselect time。

4.2.3.3 SPI NAND FLASH 地址说明

对于 SPI NAND Flash 器件地址，如表 4-11。其中 1st Byte 和 2nd Byte 为列地址，3rd Byte，4th Byte 和 5th Byte 为行地址。

表4-11 FLASH 地址分配表

Byte 数	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
1st Byte	A0	A1	A2	A3	A4	A5	A6	A7
2nd Byte	A8	A9	A10	A11	A12*	A13*	0	0
3rd Byte	A12	A13	A14	A15	A16	A17	A18	A19
4th Byte	A20	A21	A22	A23	A24	A25	A26	A27
5th Byte*	A28*	A29*	0	0	0	0	0	0

说明

- A0-A11 位对 2KB page_size 配置为有效列地址，A0-A12 对 4KB page_size 配置为有效列地址。
- A12 位（2KB page_size），A13 位（4KB page_size），对于 Micron 器件为 plane 地址，但对于其他厂家没有 plane 地址的概念。
- A28 和 A29 位，需要根据具体器件看是否需要，如果没有的话，输出为 0。

在下发 SPI NAND Flash 的读写操作时，操作时根据具体的操作下发行列地址。

- 写操作在 LOAD 操作时配置列地址，在 PROGRAM 操作时配置行地址；
- 读操作在 PAGE READ TO CACHE 操作时配置行地址，在 READ 操作时配置列地址；



内置 DMA 操作，地址下发由控制器完成，软件需要根据操作地址配置 **FMC_ADDRL** 和 **FMC_ADDRH**。其中 **FMC_ADDRL** 配置值为 1st Byte-4th Byte，**FMC_ADDRH** 配置值为 5th Byte。

说明

- 对于 2KB page_size 配置，A12 位当成 plane 地址，不支持 A12 有其他含义，影响读写结果的器件。
- 对于 4KB page_size 配置，A13 位当成 plane 地址，不支持 A13 有其他含义，影响读写结果的器件。

4.2.3.4 Boot 功能

FMC 默认处于 Boot 模式，可支持直接从 FLASH 中读取数据进行启动。CPU 可以直接读取 0x00_0000~0x0F_FFFF 地址范围的数据，整个空间大小为 1MB。

SPI NOR Flash 的 Boot 功能

由于 SPI NOR Flash 的地址空间是连续的，因此 1MB 的 Boot 数据直接映射到 SPI NOR Flash 的地址空间 0x00_0000~0x0F_FFFF。

开始 boot 之前，当 CPU 第一次读取 boot 空间数据时。FMC 会通过 Read Status 命令获取器件的 WIP，若 WIP 为 1，则 FMC 会继续获取，直到 WIP 返回值为 0。FMC 才开始向 Flash 发送读操作。

SPI NAND Flash 的 Boot 功能

对于 SPI NAND Flash，由于其地址空间不连续且存在坏块的可能，因此 1M 的 Boot 数据并不能直接映射到 Flash 中，并且 Boot 还需要 page_size、block_size 信息进行地址译码。

FMC 支持自适应 Boot 功能，能够根据 Block0 的数据自动适配器件的 ecc_type、page_size、block_size 信息。控制器要求物理 Block0 必须为好块，其他块为坏块则可以自动跳过。

FMC 在 Boot 时，遇到坏块之后能够自动跳过坏块去寻找好块并读取 Boot 信息。每次跳坏块过程最大可以连续跳过 4 个坏块，若在一次跳坏块的过程中连续出现 5 个物理坏块，则 Boot 失败。整个 Boot 过程可以出现若干次跳坏块过程，当遇到好块时，此次跳坏块过程结束，若再次遇到坏块，则开始一次新的跳坏块过程。

FMC 逻辑对坏块的判定条件有三个：

- 当前物理块的第一个页和最后一个页坏块标记位标记位 0xff。
- 当前物理块的第一个页和最后一个页空块标记位标记位 0x00。
- 当前物理块的第一个页和最后一个页的 OOB 所在纠错单元 ECC 可纠错。

只有以上三个条件同时满足，FMC 逻辑才能判定这个块为好块，否则判定为坏块。

4.2.3.5 寄存器方式操作

软件配置操作相关寄存器，如操作命令、地址等，最后配置 FMC_OP 寄存器下发操作，控制器根据软件配置值，下发操作给器件。如果还需要向器件传输数据，那会使用内部的 buffer 来传输数据。



read id, set feature, erase 等操作都是通过此种方式下发操作。

寄存器操作方式可以组合出所有的器件操作，可以单独下发指令、地址、传输数据等。

4.2.3.6 内置 DMA 操作方式

FMC 支持使用内置 DMA 模式进行读写操作以提高访问速度。通过此种方式，FMC 可以通过总线直接访问 DDR。

- DMA 写操作：SPI NOR Flash 支持从 DDR 的任意地址搬运任意长度的数据写到器件的任意地址，对于 SPI NAND Flash，只支持按 page 的写操作。
- DMA 读操作：和写操作相同，SPI NOR Flash 支持从器件的任意地址搬运任意长度的数据写到 DDR 的任意地址。对于 SPI NAND Flash 支持读整个 page，只读 OOB 操作。
- 只读 OOB 操作：当软件只需要软件管理信息（如坏块标记，空块标记等）时，不需要读取整个 page 的数据出来，只需要读取控制信息部分，使用只读 OOB 方式就能完成此目的。

4.2.3.7 ECC 校验功能

对于 SPI NAND Flash，FMC 支持 ECC 校验纠错功能。支持的最大纠错能力有四种分别为：8bit /1KB、16bit /1KB 和 24bit /1KB。以 8bit ECC 为例，表示的含义为 1KB¹数据中最大能纠正 8bit 错误。

说明

1: 1KB 纠错单元包括 ECC 保护数据、OOB 信息以及 ECC 码。纠错算法按纠错单元进行计算。可以认为 DATA+OOB 为计算 ECC 的数据，对于 2KB page_size，每个纠错单元数据大小为 (DATA+OOB) /2，对于 4KB page_size，每个纠错单元数据大小为 (DATA+OOB) /4。因此纠错单元实际大小大于 1KB。

- OOB 为软件管理信息部分，具体见数据结构描述。
- DATA 为真实数据，2KB page_size 为 2048Byte，4KB page_size 为 4096Byte。

当超过最大纠错能力时，可以上报不可纠错中断。FMC 提供告警中断，当一次纠错过程错误位数等于或大于配置的错误阈值（FMC_ERR_THD[fmc_err_thd]），则会上报错误告警，提示软件此次操作错误位数已经达到预设值。若有一个或多个纠错单元出现错误 bit，并且所有纠错单元的错误 bit 数均未达到错误阈值和不可纠错值，则可纠错中断标志 FMC_INT[err_val_int]会变成 1。

当读取一个 page 数据时，若所有纠错单元中存在某个纠错单元出现不可纠错，则上报不可纠错中断状态；否则如果存在错误告警纠错单元，则上报纠错告警状态；如果有纠错，但纠错位数没有达到错误阈值，则可以上报可纠错中断状态。

工作在 SPI NOR Flash 模式下，不使能 ECC 校验功能。当配置 FMC_CFG[ecc_type]为零，对于 SPI NAND Flash 工作在无 ECC 校验功能模式，控制器搬运数据不做数据结构处理，直接进行 flash 数据的搬运。

4.2.3.8 TIMEOUT 功能

FMC 为 DMA 写操作提供 TIMEOUT 机制，控制器在等待 DMA 写操作结束期间，会不断的发送 GET FEATURE 操作来查询写操作是否结束；对于 SPI NOR Flash，控制器则会在等待器件发送 RDSR 操作来查询操作是否结束。FMC 等待的时间若超过寄存器



[FMC_TIMEOUT_WR](#)[timeout_wr]的配置值，则结束 DMA 写操作，并上报 [FMC_INT](#)[op_fail]中断。

4.2.4 工作流程

4.2.4.1 初始化流程

初始化流程如下：

- 步骤 1. （如果需要调整 Timing 参数）根据器件配置寄存器 [TIMING_SPI_CFG](#)（SPI NAND Flash 或者 SPI NOR Flash）。
- 步骤 2. 根据对接的 FLASH 器件手册，配置 FMC 配置寄存器（[FMC_CFG](#)）。主要配置有器件接口类型、ecc_type、page_size 等。
- 步骤 3. 对于 SPI NOR Flash，如果器件是 4Byte 地址器件，而默认是 3Byte 地址模式，则参考 [4.2.4.4 SPI NOR Flash 地址模式切换流程](#)完成 4Byte 地址模式切换。
- 步骤 4. 初始化完成，可根据具体的操作配置寄存器下发操作。

----结束

4.2.4.2 FMC_OP 操作流程（寄存器操作方式）

对器件的寄存器操作（如读 ID 或者配置器件寄存器）使用配置 [FMC_OP](#) 下发操作的方式完成。操作步骤如下：

- 步骤 1. 配置 [FMC_CFG](#) 的 ecc_type 为 0。
- 步骤 2. 如果是写寄存器操作（如配置 FLASH 的配置寄存器），先从 Buffer 访问起始地址开始写入期望的操作数据。
- 步骤 3. 根据器件操作需要，配置操作命令 [FMC_CMD](#)，器件操作地址 [FMC_ADDRL](#)，读写数据个数 [FMC_DATA_NUM](#) 等操作类寄存器。
- 步骤 4. 根据下发的器件操作，匹配 [FMC_OP](#) 配置 [FMC_OP_CFG](#) 寄存器。
- 步骤 5. 配置 [FMC_OP](#) 寄存器下发操作，具体配置值参考 [FMC_OP](#) 寄存器描述。
- 步骤 6. 如果使用查询方式，查询 [FMC_OP](#)[reg_op_start]位为 1 表示操作完成；如果使用中断方式，检测到 [FMC_INT](#)[op_done_init]，表示操作完成。
- 步骤 7. 对于读寄存器操作（如读 ID 操作），在步骤 6 完成后读出的寄存器存放在 buffer 中，软件可以从中读出器件寄存器值。

----结束

4.2.4.3 读器件状态寄存器操作

如果想要读取器件的状态寄存器，提供一种简易操作方法。具体如下：

- 步骤 1. 配置 [FMC_OP](#) 寄存器下发操作（配置 [FMC_OP](#) [read_status_en]和 [FMC_OP](#) [reg_op_start]为 1）。



步骤 2. 读出的状态寄存器存放在 **FMC_FLASH_INFO** 寄存器中。

----结束

4.2.4.4 SPI NOR Flash 地址模式切换流程

对于 SPI NOR Flash 器件，支持 3Byte 与 4Byte 两种 Flash 地址模式，可通过将管脚上下拉来选择初始默认地址模式，也可以在芯片启动起来之后通过配置寄存器动态切换地址模式。

如果默认是 3Byte 模式启动，但器件是 4Byte 地址模式器件，在芯片启动起来之后切换 Flash 地址模式的步骤如下：

步骤 1. 无 Flash 操作或保证之前 Flash 操作完成。

步骤 2. 根据器件要求，用寄存器操作方式配置器件的相关寄存器发特定命令配置 Flash 进入 4B 模式。

步骤 3. 配置 FMC 的 **FMC_CFG** [spi_nor_addr_mode] 式为 4B 模式，完成 3Byte 模式到 4Byte 模式切换。

----结束



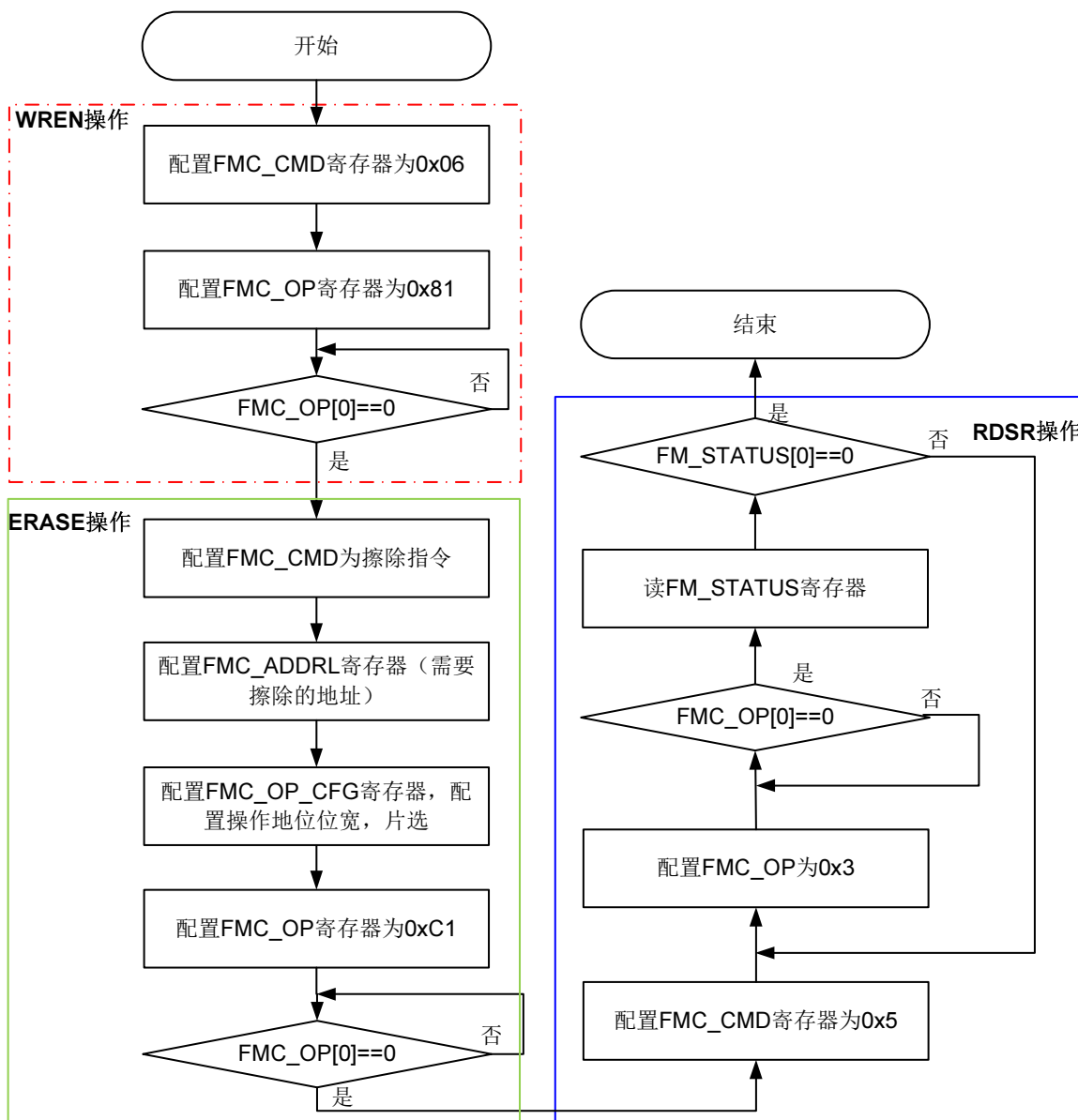
说明

具体 SPI NOR Flash 地址模式切换的命令，请查找相关器件手册。

4.2.4.5 擦除操作流程（SPI NAND Flash 和 SPI NOR Flash）

对于 flash 操作，在编程操作前，都必须进行擦除，在擦除操作前还必须完成 WREN 操作。擦除操作参考流程（查询方式）为图 4-11。

图4-11 擦除操作流程图



说明

- 上图中为查询方式，如果使用中断方式，查询 `FMC_OP[reg_op_start]` 操作变成中断处理方式，检测到 `FMC_INT[op_done_init]`，表示操作完成。
- ERASE 操作过程，配置的擦除指令及操作地址需要根据具体器件手册进行配置。

4.2.4.6 内置 DMA 读操作流程（FMC_OP_CTRL 读操作）

使用内置 DMA 方式进行读数据的操作流程如下：

- 步骤 1. 配置寄存器 `FMC_CFG[op_mode]` 为 1，确保控制器处于 NORMAL 模式。
- 步骤 2. 配置 `FMC_ADDRL` 和 `FMC_ADDRH`，配置器件操作地址。对于 SPI NOR Flash 只需要配置 `FMC_ADDRL`。



注意

对于 SPI NOR Flash, DMA 操作时地址周期数由 `FMC_CFG[spi_nor_addr_mode]` 决定: 支持 3Byte 地址模式和 4Byte 地址模式选择。对于 SPI NAND Flash, 控制器默认为 5Byte 地址模式, 且不可配置。

步骤 3. 配置 `FMC_DMA_SADDR_D0` 和 `FMC_DMA_SADDR_OOB` 寄存器, 配置在 DDR 中存放数据的起始地址。

- 对于 SPI NOR FLASH 不需要配置 `FMC_DMA_SADDR_OOB` 寄存器;
- 对于 SPI Nand Flash 的只读 OOB 操作, 只需配置 `FMC_DMA_SADDR_OOB` 寄存器。



注意

- 对于 SPI NAND Flash, 配置 DDR 地址时, 要求 4Byte 地址对齐。
- 对于 SPI NAND Flash 的 ECC0 模式, `FMC_DMA_LEN` 需要 4Byte 对齐。
- `FMC_DMA_SADDR_*` 表示目的 ddr 地址的低 32bit。 `FMC_DMA_SADDRH_*` 表示目的 ddr 地址的高 32bit。

步骤 4. 配置 `FMC_DMA_LEN` 寄存器。在 SPI NOR Flash 模式下, 需要配置 `FMC_DMA_LEN` 为读取数据的长度; 在 SPI NAND Flash 的 ECC0 模式下, 则需要配置 spare 区数据长度; 其他操作中则不需要配置此寄存器。

步骤 5. 配置 `FMC_OP_CFG` 寄存器, 根据下发的读操作要求进行配置。

- 对于 SPI NAND Flash 和 SPI NOR Flash, 要根据器件读操作时序中 dummy 周期数以及 SPI 接口类型, 配置 `FMC_OP_CFG[dummy_num]` 和 `FMC_OP_CFG[mem_if_type]`。
- 配置 `FMC_OP_CFG[fm_cs]`, 选择操作片选。

步骤 6. 配置 `FMC_OP_CTRL` 寄存器, 配置 `FMC_OP_CTRL[dma_op_ready]` 为 1, 下发 FMC 读 FLASH 操作。

- 配置 `FMC_OP_CTRL[rw_op]` 为零, 表示 DMA 读操作。
- 对于 SPI NAND Flash 的只读 OOB 操作, 需配置 `FMC_OP_CTRL[rd_op_sel]`。
- 对于 SPI NAND Flash 和 SPI NOR Flash, 需要配置 `FMC_OP_CTRL[rd_opcode]`, 根据器件 READ 操作指令进行配置。

步骤 7. 使用查询方式, 检测到 `FMC_OP_CTRL[0]` 变低, 表示读操作已经完成, 数据已经写入 DDR 中; 使用中断方式, 检测到 `FMC_INT[op_done_int]` 变高, 表示读操作已经完成, 数据已经写入 DDR 中。

----结束



4.2.4.7 内置 DMA 写操作流程（FMC_OP_CTRL 写操作）

使用内置 DMA 方式进行写 FLASH 的操作，流程如下：

- 步骤 1. 配置寄存器 **FMC_CFG** [op_mode] 为 1，确保控制器处于 NORMAL 模式。
- 步骤 2. 配置 **FMC_ADDRL** 和 **FMC_ADDRH**，配置器件操作地址。对于 SPI NOR FLASH 只需要配置 **FMC_ADDRL**。



注意

对于 SPI NOR Flash，DMA 操作时地址周期数由 **FMC_CFG**[spi_nor_addr_mode] 决定：支持 3Byte 地址模式和 4Byte 地址模式选择。对于 SPI NAND Flash，控制器默认为 5Byte 地址模式，且不可配置。

- 步骤 3. 配置 **FMC_DMA_SADDR_D0** 和 **FMC_DMA_SADDR_OOB** 寄存器，配置从 DDR 中搬运数据的起始地址。SPI NOR Flash 不需要配置 **FMC_DMA_SADDR_OOB** 寄存器。



注意

- 对于 SPI NAND Flash，配置 DDR 地址时，要求 4Byte 地址对齐。
- 对于 SPI NAND Flash 的 ECC0 模式，**FMC_DMA_LEN** 需要 4Byte 对齐。
- **FMC_DMA_SADDR_*** 表示目的 ddr 地址的低 32bit。**FMC_DMA_SADDRH_*** 表示目的 ddr 地址的高 32bit。

- 步骤 4. 配置 **FMC_DMA_LEN** 寄存器。在 SPI NOR Flash 模式下，需要配置 **FMC_DMA_LEN** 为读取数据的长度；在 SPI NAND Flash 的 ECC0 模式下，则需要配置 spare 区数据长度；其他操作中则不需要配置此寄存器。
- 步骤 5. 配置 **FMC_OP_CFG** 寄存器，根据下发的写操作要求配置
 - 对于 SPI NAND Flash 和 SPI NOR Flash，根据所需写操作 SPI 接口选择，配置 **FMC_OP_CFG** [mem_if_type]。
 - **FMC_OP_CFG** [fm_cs]，选择操作片选。
- 步骤 6. 配置 **FMC_OP_CTRL** 寄存器下发操作。
 - 配置 rw_op 信息为写（配置为 1），表示 DMA 写操作。
 - 对于 SPI NAND Flash 和 SPI NOR Flash，需要配置 **FMC_OP_CTRL** [wr_opcode]，根据器件 PROGRAM 操作指令进行配置。
- 步骤 7. 使用查询方式，检测到 **FMC_OP_CTRL** [0] 变低，表示一次 DMA 写操作已经完成，数据已经写入 FLASH 中；使用中断方式，检测到 **FMC_INT** [op_done_int] 变高，表示数据已经写入 FLASH 中。

----结束



4.2.4.8 其它注意事项

其它注意事项如下：

- 对于 SPI NAND Flash 器件，在使用之前或者异常复位后必须先进行器件的 Reset 操作。
- 在 `FMC_OP_CTRL[dma_op_ready]` 或者 `FMC_OP[reg_op_start]` 为 1 过程中（控制器操作进行中），不进行相关操作寄存器的配置，否则可能导致操作不正常。
- 软件配置需要保证在正确合理的原则下进行，否则乱配置寄存器可能会导致 FMC 挂死。

4.2.5 数据结构(SPI NAND Flash)

FMC 的数据结构分成两种，一种是用户原始数据，另一种是在非 ECC0 模式下，经过 FMC 转化，存放在 SPI NAND Flash 上的数据结构。在非 ECC0 模式下，原始用户数据和经过 FMC 转化的数据都具有固定的数据结构，各个数据成员的长度和位置均严格固定。而在 ECC0 模式下，用户数据原封不动的写入/读出，FMC 不对用户数据做任何改变。

对于 ECC0 模式，FMC 对数据进行透传处理：写操作时，控制器将 Buffer 中的数据不作处理直接写入 Flash；读操作时，控制器把从 Flash 中读取的数据不作处理直接写入到 Buffer。

在非 ECC0 模式下，对 Flash 进行写操作时，FMC 对用户数据进行两次处理：生成 ECC 码和重组。控制器在内部 Buffer 进行 ECC 编码，并生成 ECC 码，然后将 Buffer 中的数据进行重组之后再写入 Flash。对 Flash 进行读操作时，FMC 同样要进行两次处理：重组 Flash 数据和 ECC 纠错。FMC 把从 Flash 中读到的数据重组后写入内部 Buffer，然后在 Buffer 中对数据进行 ECC 纠错。

在非 ECC0 模式下，Buffer 中的数据和 Flash 中的数据被划分成不同的结构。不同的配置下，各数据段的长度参考表 4-12。

表4-12 非 ECC0 模式数据结构长度

ECC ^a (bit)	ecc_len ^b (Byte)	page_size(Byte)	oob_len ^c (Byte)	sec_len ^d (Byte)
4/512B (8/1K)	14	2048	30+2	1040
		4096		1032
8/512B (16/1K)	28	2048	6+2	1028
		4096	14+2	
24/1K	42	2048	30+2	1040
		4096		1032

a: 4bit/512B 等价于 8bit/1K，8bit/512B 等价于 16bit/1K。

b: ecc_len 表示每个纠错单元生成的 ECC 码的长度。

c: oob_len 表示上层软件可见的冗余区长度，由 CTRL+BB 组成。

d: sec_len 表示每个纠错单元的数据区长度， $sec\_len = 1024 + (oob\_len * 1024) / page\_size$



e: ECC 的每个纠错单元为 1K, 此处 1K 并不是严格意义上的 1024Byte, 而是指长度处于 1K 这一数量级。例如表中纠错单元长度 sec_len 从 1026-1040 不等, 统称为 1K。

下面结合写操作的过程说明各个数据段的构成。

FMC 将数据写入到 Flash 中时, 会按照 DATA+ECC 的格式交替连续存放, 但 OOB 区数据需要进行特殊处理:

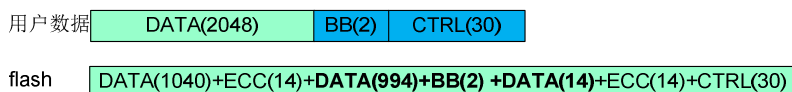
- BB 标识需要放在 Flash 页中 page_size 的地方, 即对于 2K、4K 的配置, BB 数据将被放置到 2048、4096 Byte 开始的两个字节。这样会打断原有的 DATA 或 ECC 数据。
- CTRL 数据需要放置在有效数据的末尾, 这样最后一个数据段在 Flash 中的存放形式为: DATA(不含 BB 和 CTRL)+ECC+CTRL。
- EB 作为标记空块的标记位, 固定放置在 CTRL 最后 2Byte。

4.2.5.2 4bit ECC 模式(纠错 8bit/1KB)

2KB page_size

对 2KB page_size 的配置, 软件可用的冗余区大小为 32Byte。数据在 Buffer 和 Flash 中的结构如图 4-12 所示。

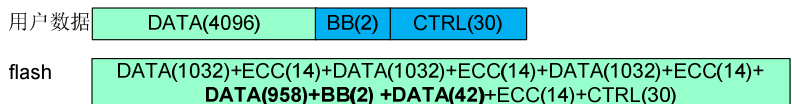
图4-12 4bit ECC 模式 2K page_size 的数据结构



4KB page_size

对 4KB page_size 的配置, 软件可用的冗余区大小为 32Byte。数据在 Buffer 和 Flash 中的结构如图 4-13 所示。

图4-13 4bit ECC 模式 4K page_size 的数据结构



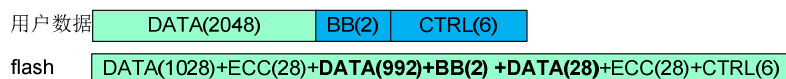
4.2.5.3 8bit ECC 模式(纠错 16bit/1KB)

2KB page_size

对 2KB page_size 的配置, 软件可用的冗余区大小为 8Byte。数据在 Buffer 和 Flash 中的结构如图 4-14 所示。



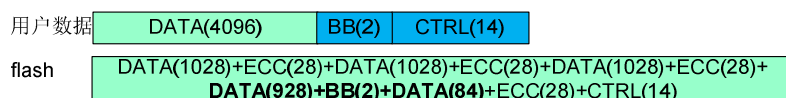
图4-14 8bit ECC 模式 2K page_size 的数据结构



4KB page_size

对 4KB page_size 的配置，软件可用的冗余区大小为 16Byte。数据在 Buffer 和 Flash 中的结构如图 4-15 所示。

图4-15 8bit ECC 模式 4K page_size 的数据结构

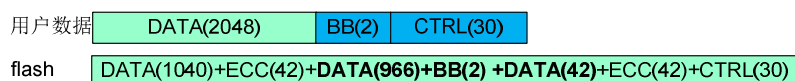


4.2.5.4 24bit ECC 模式(纠错 24bit/1KB)

2KB page-size

对 2KB page-size 的配置，软件可用的冗余区大小为 32Byte。数据在 Buffer 和 Flash 中的结构如图 4-16 所示。

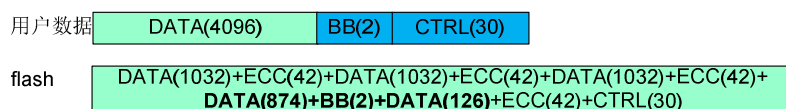
图4-16 24bit ECC 模式 2K page-size 的数据结构



4KB page_size

对 4KB page_size 的配置，软件可用的冗余区大小为 32Byte。数据在 Buffer 和 Flash 中的结构如图 4-17 所示。

图4-17 24bit ECC 模式 4K page_size 的数据结构



4.2.6 ECC 模式选择说明

因为控制器使用的“ECC 纠错 IP”在处理数据时是“基于 1KB 的数据块”进行处理的，所以描述控制器的 ECC 纠错性能时是这么写的：8bit/1KByte、24bit/1KByte 等。为了可靠地使用器件，器件 datasheet 往往会给出建议的 ECC 纠错性能，如



1bit/512Byte、4bit/512Byte、8bit/512Byte 等。8bit/1KByte 与 4bit/512Byte 描述的纠错性能其实是一样的。使用控制器时，ECC 模式选择需遵循以下 2 条准则：

- 控制器的 ECC 纠错性能 $\geq$ 器件建议的 ECC 纠错性能
比如说，如果器件建议 1bit/512Byte 或者 4bit/512Byte，那么控制器可以选择 8bit/1KByte 纠错性能的 ECC 模式。
- 器件的页大小 $\geq$ 控制器 ECC 模式所需的存储大小
比如说，如果器件页大小是 2K+64Byte，那么控制器不能选择 24bit/1KByte 纠错性能的 ECC 模式，因为这种 ECC 模式所需的存储大小为 2K+116Byte，ECC 模式所需的存储大小请参考“4.2.5 数据存储结构”这一章节。

4.2.7 FMC 寄存器概览

FMC 寄存器概览如表 4-13 所示。

表4-13 FMC 寄存器概览（基址是 0x1000_0000）

偏移地址	名称	描述	页码
0x0000	FMC_CFG	器件配置寄存器	4-94
0x0004	GLOBAL_CFG	全局配置寄存器	4-95
0x0008	TIMING_SPI_CFG	SPI Timing 配置寄存器	4-97
0x0018	FMC_INT	中断状态寄存器	4-97
0x001C	FMC_INT_EN	中断使能寄存器	4-98
0x0020	FMC_INT_CLR	中断清除寄存器	4-99
0x0024	FMC_CMD	命令字配置寄存器	4-100
0x0028	FMC_ADDRH	器件地址配置寄存器高 byte	4-100
0x002C	FMC_ADDRL	器件地址配置寄存器低 4byte	4-100
0x0030	FMC_OP_CFG	操作配置寄存器	4-101
0x0034	SPI_OP_ADDR	操作地址配置寄存器	4-102
0x0038	FMC_DATA_NUM	数据长度寄存器	4-102
0x003C	FMC_OP	操作寄存器	4-102
0x0040	FMC_DMA_LEN	DMA 操作长度寄存器	4-103
0x0048	FMC_DMA_AHB_CTRL	DMA AHB 总线操作控制寄存器	4-104
0x004C	FMC_DMA_SADDR_D0	DMA 操作 DDR 起始地址寄存器 0	4-104
0x005C	FMC_DMA_SADDR_OOB	DMA 操作 DDR 存放 OOB 信息起始地址寄存器	4-104



偏移地址	名称	描述	页码
0x0068	FMC_OP_CTRL	操作控制寄存器	4-105
0x006C	FMC_TIMEOUT_W R	写操作 TIMEOUT 时间寄存器	4-105
0x0070	FMC_OP_PARA	OP 操作参数选择寄存器	4-106
0x0074	FMC_BOOT_SET	boot 设定寄存器	4-106
0x0078	FMC_LP_CTRL	低功耗控制寄存器	4-107
0x00A8	FMC_ERR_THD	ECC 告警阈值寄存器	4-107
0x00AC	FMC_FLASH_INFO	器件状态寄存器值	4-107
0x00BC	FMC_VERSION	版本寄存器	4-108
0x00C0	FMC_ERR_NUM0_ BUF0	spinand flash 和 nandflash 第一次 buffer 纠错信息 0 统计寄存器	4-108
0x00D0	FMC_ERR_ALARM_ ADDRH	ECC 告警 flash 地址高 byte 寄存器	4-108
0x00D4	FMC_ERR_ALARM_ ADDRL	ECC 告警 flash 地址低 byte 寄存器	4-109
0x00D8	FMC_ECC_INVALID_ ADDRH	ECC 不可纠错地址高 byte 寄存器	4-109
0x00DC	FMC_ECC_INVALID_ ADDRL	ECC 不可纠错地址低 4byte 寄存器	4-109

4.2.8 FMC 寄存器描述

FMC_CFG

FMC_CFG 为器件配置寄存器。

Offset Address: 0x0000 Total Reset Value: 0x0000_1820

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12:11]	RW	spi_nand_plane_en	Plane bit 功能使能（发送地址时，附带发送 Plane 地址位信息，SPI NAND 模式下有效） 11：使能； 其他：不使能。	0x3
[10]	RW	spi_nor_addr_mode	SPI 地址模式（只对 SPI NOR 有效） 0：3Bytes address mode (default);	0x0



Bits	Access	Name	Description	Reset
			1: 4Bytes address mode。 复位值由管脚决定。	
[9:8]	RW	block_size	SPI NAND Flash 的块大小。 00: 64 page; 01: 128 page; 其他: 保留。	0x0
[7:5]	RW	ecc_type	配置控制器的 ECC 类型。 000: 无 ECC; 001: 8bit ECC; 010: 16bit ECC; 011: 24bit ECC; 其他: 保留。	0x1
[4:3]	RW	page_size	SPI NAND Flash 的页大小。 00: 2KB page_size; 01: 4KB page_size; 其他:保留。	0x0
[2:1]	RW	flash_sel	Flash 类型选择。 00: SPI NOR FLASH; 01: SPI NAND FLASH; 其他: 保留。 复位值由管脚决定。	0x0
[0]	RW	op_mode	FMC 操作模式选择。 0: Boot 模式; 1: Normal 模式。	0x0

GLOBAL_CFG

GLOBAL_CFG 为全局配置寄存器。

Offset Address: 0x0004 Total Reset Value: 0x0000_00C4

Bits	Access	Name	Description	Reset
[31:25]	-	reserved	保留。	0x0000
[24]	RW	ddr_mode_byp	spl nand dtr 功能 bypass, 为了保证之前所有功能的正确性。 1: 使能 bypass 功能。	0x0



Bits	Access	Name	Description	Reset
			0: 不使能 bypass 功能。 注: 该位域只针对 spi nand dtr。	
[23:16]	-	reserved	保留。	0x0000
[15:12]	RW	sample_point	DDR 模式时根据不同器件不同延时参数选择采样点。采样时钟频率 4 倍于接口读写时钟。 0x0 延时最小, 0xB 延时最大。 0x0: 采样点在有效沿后 2 个采样时钟周期。 0x1: 采样点在有效沿后 2.5 个采样时钟周期。 0x2: 采样点在有效沿后 3 个采样时钟周期。 0x3: 采样点在有效沿后 3.5 个采样时钟周期。 0x4: 采样点在有效沿后 4 个采样时钟周期。 0x5: 采样点在有效沿后 4.5 个采样时钟周期。 0x6: 采样点在有效沿后 5 个采样时钟周期。 0x7: 采样点在有效沿后 5.5 个采样时钟周期。 0x8: 采样点在有效沿后 6 个采样时钟周期。 0x9: 采样点在有效沿后 6.5 个采样时钟周期。 0xA: 采样点在有效沿后 7 个采样时钟周期。 0xB: 采样点在有效沿后 7.5 个采样时钟周期。 其他: 保留。	0x0
[11]	RW	ddr_mode	DDR 模式使能控制。 0: 正常 SDR 模式; 1: DDR 模式。	0x0
[10:7]	-	reserved	保留。	0x1
[6]	RW	wp_en	写保护使能(WP 管脚)。当使能时, 芯片对 WP 管脚输出为 0。 0: 不使能; 1: 使能。	0x1
[5:3]	RW	rd_delay	SPI 读出数据延迟周期个数。(SDR 模式下) 000: 读数据延迟 0 个时钟周期; (default) 001: 读数据延迟 0.5 个器件工作时钟周期; 010: 读数据延迟 1 个器件工作时钟周期; 011: 读数据延迟 1.5 个器件工作时钟周期; 100: 读数据延迟 2 个器件工作时钟周期; 101: 读数据延迟 2.5 个器件工作时钟周期; 110: 读数据延迟 3 个器件工作时钟周期;	0x0



Bits	Access	Name	Description	Reset
			111: 读数据延迟 3.5 个器件工作时钟周期。	
[2]	RO	reserved	保留。	0x1
[1:0]	RO	reserved	保留。	0x0

TIMING_SPI_CFG

TIMING_SPI_CFG 为 SPI Timing 配置寄存器。

Offset Address: 0x0008 Total Reset Value: 0x0000_006F

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:4]	RW	tcss	CS setup time。 000~111: n+1 个接口时钟周期。n 为 0、1、2、……、7。	0x6
[3:0]	RW	tshsl	设置片选的 Deselect time，相当于两次 Flash 操作之间的时间间隔。 0000~1111: n+1 个接口时钟周期。n 为 0、1、2、……、15。	0xF

FMC_INT

FMC_INT 为中断状态寄存器。

Offset Address: 0x0018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RO	ahb_op_int	FMC 对 Flash 进行数据操作期间，cpu 读写内部 buffer。 0: 无中断; 1: 有中断。	0x0
[6]	-	reserved	保留。	0x0
[5]	RO	dma_err_int	DMA 传输总线出现错误中断。 0: 无中断; 1: 有中断。	0x0



Bits	Access	Name	Description	Reset
[4]	RO	err_alarm_int	ECC 校验告警中断。当错误的 bit 达到设定的阈值，产生中断。 0: 无中断; 1: 有中断。	0x0
[3]	RO	err_inval_int	ECC 校验不可纠正的错误。 8bit 纠错模式下，校验 1024byte 数据中出现 8bit 以上错误，产生中断。 16bit 纠错模式下，校验 1024byte 数据中出现 16bit 以上错误，产生中断。 24bit 纠错模式下，校验 1024byte 数据中出现 24bit 以上错误，产生中断。 0: 无中断; 1: 有中断。	0x0
[2]	RO	err_val_int	ECC 校验可纠正的错误。 8bit 纠错模式下，校验 1024byte 数据中出现 1—8bit 错误，产生中断。 16bit 纠错模式下，校验 1024byte 数据中出现 1—16bit 错误，产生中断。 24bit 纠错模式下，校验 1024byte 数据中出现 1—24bit 错误，产生中断。 0: 无中断; 1: 有中断。	0x0
[1]	RO	op_fail_int	编程操作失败，上报中断。 0: 无中断; 1: 有中断。	0x0
[0]	RO	op_done_int	controller 本次操作结束中断。 写 operation 寄存器后，该标志自动清零。 0: 无中断; 1: 有中断。	0x0

FMC_INT_EN

FMC_INT_EN 为中断使能寄存器。

Offset Address: 0x001C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RW	ahb_op_int_en	FMC 读写 Flash 数据期间，cpu 读写内部 buffer 错误中断使能。 0: 禁止; 1: 使能。	0x0
[6]	-	reserved	保留。	0x0
[5]	RW	dma_err_int_en	DMA 传输总线出现错误中断使能。 0: 禁止; 1: 使能。	0x0
[4]	RW	err_alarm_int_en	ECC 校验告警中断。当错误的 bit 达到设定的阈值，产生中断。 0: 禁止; 1: 使能。	0x0
[3]	RW	err_inval_int_en	ECC 校验不可纠正错误，产生中断。 0: 禁止; 1: 使能。	0x0
[2]	RW	err_val_int_en	ECC 校验可纠正的错误，产生中断。 0: 禁止; 1: 使能。	0x0
[1]	RW	op_fail_int_en	编程操作失败中断使能。 0: 禁止; 1: 使能。	0x0
[0]	RW	op_done_int_en	FMC 本次操作结束中断。 0: 禁止; 1: 使能。	0x0

FMC_INT_CLR

FMC_INT_CLR 为中断清除寄存器。

Offset Address: 0x0020 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	WO	ahb_op_int_clr	清除 ahb_op_err 中断，写 1 清除。	0x0



Bits	Access	Name	Description	Reset
[6]	-	reserved	保留。	0x0
[5]	WO	dma_err_int_clr	清除 DMA 传输总线错误中断，写 1 清除。	0x0
[4]	WO	err_alarm_int_clr	清除 err_alarm 中断，写 1 清除。	0x0
[3]	WO	err_inval_int_clr	清除 err_invalid 中断，写 1 清除。	0x0
[2]	WO	err_val_int_clr	清除 err_valid 中断，写 1 清除。	0x0
[1]	WO	op_fail_int_clr	清除 op_fail 中断，写 1 清除。	0x0
[0]	WO	op_done_int_clr	清除 op_done 中断，写 1 清除。	0x0

FMC_CMD

FMC_CMD 为命令字配置寄存器。

Offset Address: 0x0024 Total Reset Value: 0x0000_3000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:8]	-	reserved	保留。	0x30
[7:0]	RW	cmd1	Controller 发给 SPI NOR/NAND Flash 的操作命令。	0x00

FMC_ADDRH

FMC_ADDRH 为器件地址配置寄存器高 byte。

Offset Address: 0x0028 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	addrh	对于 SPI NAND Flash，配置 Flash 的操作高 byte 地址。	0x00

FMC_ADDRL

FMC_ADDRL 为器件地址配置寄存器低 4byte。

Offset Address: 0x002C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	addrl	配置 Flash 器件的操作地址的低 4byte。对于 SPI NOR Flash，配置器件地址。	0x00000000

FMC_OP_CFG

FMC_OP_CFG 为操作配置寄存器。

Offset Address: 0x0030 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13]	RW	multi_cycle_en	SPI 接口冗余周期使能。 0: 不使能; 1: 使能。 输出时钟超过 50MHz 时必须配置为 1。	0x0
[12]	-	reserved	保留。	0x0
[11]	RW	fm_cs	选择操作的 Flash 器件。 0: cs0; 1: cs1。	0x0
[10]	RW	force_cs_en	cs 片选强制使能选择。 0: 不使能; 1: 使能。	0x0
[9:7]	RW	mem_if_type	读操作, SPI FLASH 接口类型选择。 000: Standard SPI 接口类型 001: Dual-Input/Dual-Output SPI 010: Dual-I/O SPI 011: Quad-Input/Quad-Output SPI 100: Quad-I/O SPI 101-111: 保留。	0x0
[6:4]	RW	addr_num	发给 Flash 的地址 Byte 数。	0x0
[3:0]	RW	dummy_num	对于 dummy_en 的操作 Byte 个数。 (1Byte=4 线 2 时钟周期=2 线 4 时钟周期=1 线 8 时钟周期)	0x0



SPI_OP_ADDR

SPI_OP_ADDR 为操作地址配置寄存器。

Offset Address: 0x0034 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	spi_op_addr	配置 SPI NOR/NAND Flash 的操作地址。每次操作下发器件的操作地址。不同于 FMC_ADDRL 和 FMC_ADDRH。	0x0000_0000

FMC_DATA_NUM

FMC_DATA_NUM 为数据长度寄存器。

Offset Address: 0x0038 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13:0]	RW	op_data_num	一次操作所需要处理的数据长度。在有数据传输的操作时需要配置，对于 DMA 操作或 AHB 总线直接访问时不需要配置。仅对于 ECC0 有效。	0x0000

FMC_OP

FMC_OP 为操作寄存器。

Offset Address: 0x003C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:9]	-	reserved	保留。	0x000000
[8]	RW	dummy_en	传输 dummy byte 使能 0: 禁止; 1: 使能。 同时使能 dummy byte 和地址时, dummy 在地址操作后发出。	0x0
[7]	RW	cmd1_en	向 flash 发 command1 命令使能。 0: 禁止; 1: 使能。	0x0
[6]	RW	addr_en	向 FLASH 写操作地址使能。	0x0



Bits	Access	Name	Description	Reset
			0: 禁止; 1: 使能。	
[5]	RW	write_data_en	向 Flash 写数据使能。 read_data_en 和 write_data_en 不能同时为 1。 0: 禁止; 1: 使能。	0x0
[4:3]	-	reserved	保留。	0x0
[2]	RW	read_data_en	从 Flash 读数据使能。 read_data_en 和 write_data_en 不能同时为 1。 0: 禁止; 1: 使能。	0x0
[1]	RW	read_status_en	读器件状态寄存器使能。 0: 禁止; 1: 使能。 该标志为 1 时, 控制器读器件状态寄存器值存放在 FMC_FLASH_INFO 的 fm_status 域中 (而不写入内部 buffer)。	0x0
[0]	RWSC	reg_op_start	下发 OP 操作控制器状态。 0: 控制器就绪(软件下发操作时只能将该位域配置为 1, 使能逻辑); 1: 控制器忙(控制器操作完成后会自动将这个 bit 设置为 0, 表示逻辑完成)。	0x0

FMC_DMA_LEN

FMC_DMA_LEN 为 DMA 操作长度寄存器。

Offset Address: 0x0040 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27:0]	RW	dma_len	DMA 操作数据搬运长度, 以 byte 为单位。 用于 SPI NAND ecc0 时, 配置为 spare 区域数据长度。 用于 SPI NOR 的 dma 读写操作时, 配置为 dma 所需数据长度。	0x0000000



FMC_DMA_AHB_CTRL

FMC_DMA_AHB_CTRL 为 DMA AHB 总线操作控制寄存器。

Offset Address: 0x0048 Total Reset Value: 0x0000_0007

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2]	RW	burst16_en	BURST16 使能。 0: 禁止; 1: 使能。	0x1
[1]	RW	burst8_en	BURST8 使能。 0: 禁止; 1: 使能。	0x1
[0]	RW	burst4_en	BURST4 使能。 0: 禁止; 1: 使能。	0x1

FMC_DMA_SADDR_D0

FMC_DMA_SADDR_D0 为 DMA 操作 DDR 起始地址寄存器 0。

Offset Address: 0x004C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	dma_mem_saddr_d0	对于 SPI NOR Flash, 指示 DMA 操作 DDR 起始地址寄存器。对于 SPI NAND Flash, 指示 dma 操作, DDR 操作数据基地址。	0x0000_0000

FMC_DMA_SADDR_OOB

FMC_DMA_SADDR_OOB 为 DMA 操作 DDR 存放 OOB 信息起始地址寄存器。

Offset Address: 0x005C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	dma_mem_saddr_oob	读写 OOB 区数据的 DDR 基地址。	0x0000_0000



FMC_OP_CTRL

FMC_OP_CTRL 为操作控制寄存器。

Offset Address: 0x0068 Total Reset Value: 0x0003_0200

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:16]	RW	rd_opcode	SPI NAND/NOR Flash, 非总线方式, DMA 读操作命令。 (FAST_READ/READ/DUAL_READ)	0x03
[15:8]	RW	wr_opcode	对于 SPI NAND/NOR Flash, 非总线方式, DMA 写操作命令。	0x02
[7:6]	-	reserved	保留。	0x0
[5:4]	RW	rd_op_sel	读操作时, 读数据区域选择。 00: 读整个 page; 01: 只读 OOB 操作; 其他: 保留。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	RW	rw_op	DMA 读写模式选择。 0: DMA 读; 1: DMA 写。	0x0
[0]	RWSC	dma_op_ready	下发操作控制器状态。 0: 控制器就绪; (软件只能配置为 1, 使能逻辑) 1: 控制器忙。(逻辑只能配置为 0, 表示逻辑完成)	0x0

FMC_TIMEOUT_WR

FMC_TIMEOUT_WR 为写操作 TIMEOUT 时间寄存器。

Offset Address: 0x006C Total Reset Value: 0x00FF_FFFF

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RW	timeout_wr	编程操作 busy 等待期间, timeout 时间长度。 (器件响应超时时间)其单位为 1 个接口时钟周期。 (SPI NAND/SPI NOR Flash)	0x000000



FMC_OP_PARA

FMC_OP_PARA 为 OP 操作参数选择寄存器。

Offset Address: 0x0070 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	rd_oob_only	OP 下发寄存器读 flash 操作时，只读 OOB 相关的 sector。 0: 读 page; 1: 只读 OOB。	0x0
[0]	-	reserved	保留。	0x0

FMC_BOOT_SET

FMC_BOOT_SET 为 boot 设定寄存器。

Offset Address: 0x0074 Total Reset Value: 0x0000_0005

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x00000000
[4]	RW	boot_otp_cfg	器件配置信息，使用 OTP 直接配置 FMC_CFG 寄存器的方式。 0: 使用自适应 Boot 模式; 1: 使用 OTP 模式。	0x0
[3]	-	reserved	保留。	0x0
[2]	-	reserved	保留。	0x1
[1]	RW	boot_quad_mode	对于 SPI NAND FLASH，BOOT 是否使用四线模式启动。复位值由管脚决定。 0: 不使用，使用单线模式启动; 1: 使用。	0x0
[0]	RW	boot_page0_cfg	Boot 模式选择(SPI NAND Flash)。 0: 默认启动方式; 1: 省管脚启动方式。	0x1



FMC_LP_CTRL

FMC_LP_CTRL 为低功耗控制寄存器。

Offset Address: 0x0078 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	mem_gate_en	SRAM 时钟关断选择 0: 不关闭; 1: 关闭。	0x0
[0]	RW	clk_gate_en	时钟门控选择。打开此功能, 使用低功耗设计, 逻辑选择关闭模块工作时钟。 0: 时钟全部使能; 1: 根据低功耗设计, 关闭时钟。	0x0

FMC_ERR_THD

FMC_ERR_THD 为 ECC 告警阈值寄存器。

Offset Address: 0x00A8 Total Reset Value: 0x0000_00FF

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	fmc_err_thd	ECC 校验告警阈值配置寄存器。当错误 bit 达到此阈值时, 触发 ECC 校验告警中断。 注: 1.不可纠错时, 无论此寄存器值是多少, 都上报为不可纠错中断; 2.设置为 0 和 1 等价, 即只要有错误发生, 都上报为告警中断; 3.设置值大于可纠错范围时, 不论出现多少 bit 错误, 告警中断均不上报, 只有可纠错和不可纠错中断产生。	0xFF

FMC_FLASH_INFO

FMC_FLASH_INFO 为器件状态寄存器值。

Offset Address: 0x00AC Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:24]	RO	exp_bb_flag	扩展操作下发的只读 OOB 或者读 page 时，当前页坏块标志对应的未纠错数据	0x00
[23:16]	RO	bb_flag	只读 OOB 或者读 page 时，当前页坏块标志对应的未纠错数据	0x00
[15:8]	RO	exp_flash_status	扩展操作，控制器从器件读状态寄存器值。	0x00
[7:0]	RO	flash_status	器件读状态寄存器值。(比特含义详见各 flash 器件手册)	0x00

FMC_VERSION

FMC_VERSION 为版本寄存器。

Offset Address: 0x00BC Total Reset Value: 0x0000_0100

Bits	Access	Name	Description	Reset
[31:0]	RO	version	FMC 版本 V100。	0x0000_0100

FMC_ERR_NUM0_BUF0

FMC_ERR_NUM0_BUF0 为 SPI NAND Flash 第一次 buffer 纠错信息 0 统计寄存器。

Offset Address: 0x00C0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	err_num0_buf0	对于 2KB, 4KB, 8KB 及 16KB page_size 第一次 buffer 操作，前 4KB 数据错误统计寄存器。 bit[31:24]: 第四 KB 数据纠错，错误数据统计寄存器。 bit[23:16]: 第三 KB 数据纠错，错误数据统计寄存器。 bit[15:8]: 第二 KB 数据纠错，错误数据统计寄存器。 bit[7:0]: 第一 KB 数据纠错，错误数据统计寄存器。	0x0000_0000

FMC_ERR_ALARM_ADDRH

FMC_ERR_ALARM_ADDRH 为 ECC 告警 Flash 地址高 byte 寄存器。



Offset Address: 0x00D0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RO	fmc_err_alarm_addrh	当出现 ECC 告警中断时，最后一次触发中断的 Flash 操作地址高 byte 寄存器。	0x00

FMC_ERR_ALARM_ADDRL

FMC_ERR_ALARM_ADDRL 为 ECC 告警 flash 地址低 byte 寄存器。

Offset Address: 0x00D4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	fmc_err_alarm_addrl	当出现 ECC 告警中断时，最后一次触发中断的 Flash 操作地址低 4byte 寄存器。	0x0000_0000

FMC_ECC_INVALID_ADDRH

FMC_ECC_INVALID_ADDRH 为 ECC 不可纠错地址高 byte 寄存器。

Offset Address: 0x00D8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RO	fmc_ecc_invalid_addrh	当出现 ECC 不可纠错时，最后一次触发中断的 Flash 操作地址高 byte 寄存器。	0x00

FMC_ECC_INVALID_ADDRL

FMC_ECC_INVALID_ADDRL 为 ECC 不可纠错地址低 4byte 寄存器。

Offset Address: 0x00DC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	fmc_ecc_invalid_addrl	当出现 ECC 不可纠错时，最后一次触发中断的 Flash 操作地址低 4byte 寄存器。	0x0000_0000



目 录

5 ETH	5-1
5.1 概述.....	5-1
5.2 功能描述.....	5-1
5.3 信号描述.....	5-2
5.4 工作方式.....	5-3
5.4.1 收帧过程	5-3
5.4.2 发帧过程	5-6
5.4.3 中断管理	5-9
5.4.4 流量控制	5-9
5.4.5 典型应用	5-9
5.5 寄存器概览.....	5-13
5.6 寄存器描述.....	5-17
5.6.1 MDIO 控制寄存器.....	5-17
5.6.2 MAC 控制寄存器	5-21
5.6.3 全局控制寄存器.....	5-27
5.6.4 统计计数控制寄存器.....	5-47
5.6.5 统计结果寄存器.....	5-49



插图目录

图 5-1 Ethernet 模块逻辑框图 5-2

图 5-2 中断方式收帧流程图.....5-5

图 5-3 查询方式收帧流程图.....5-6

图 5-4 CPU 发帧流程图.....5-7



表格目录

表 5-1 MDIO 接口信号描述.....	5-2
表 5-2 MII 接口信号描述	5-3
表 5-3 CPU 接收帧描述子数据结构	5-3
表 5-4 CPU 发送帧描述子数据结构	5-7
表 5-5 MDIO 控制寄存器概览（基地址是 0x1001_0000）	5-13
表 5-6 MAC 控制寄存器概览（基地址是 0x1001_0000）	5-13
表 5-7 全局控制寄存器概览（基地址是 0x1001_0000）	5-13
表 5-8 统计计数控制寄存器概览（基地址是 0x1001_0000）	5-15
表 5-9 统计结果寄存器概览（基地址是 0x1001_0000）	5-15



5 ETH

5.1 概述

以太网模块提供了 1 个 Ethernet MAC，实现网络接口数据的接收和发送，可以工作在 10Mbit/s 或 100Mbit/s 模式下，支持全双工或者半双工工作模式，提供 RMII 接口，提供可配置的 8 组 DMAC 地址过滤表，可以对网口的输入帧进行选择过滤接收，同时实现了对 CPU 端口流量限制功能，保护 CPU 不被大流量攻击。

注意：Hi3559V200/Hi3556V200 不支持 ETH 模块。

5.2 功能描述

Ethernet 模块有如下功能特点：

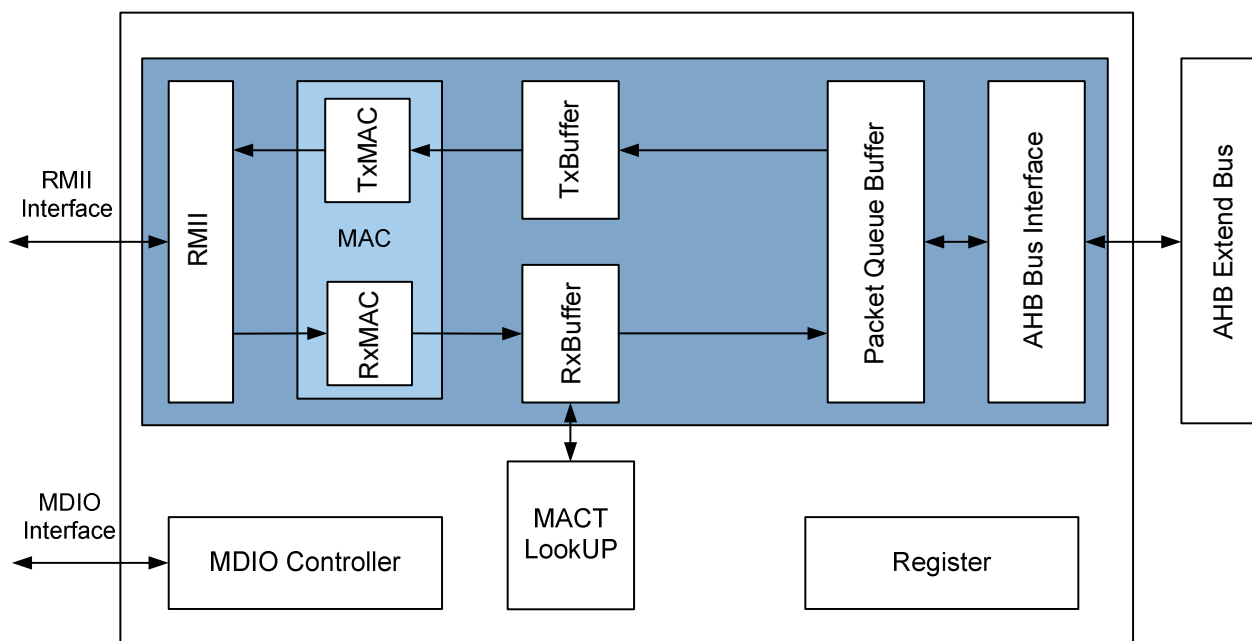
- 提供 1 个 Ethernet MAC。
- 支持 10Mbit/s 或 100Mbit/s 速率。
- 可工作在全双工或半双工模式。
- 支持 RMII 接口。
- 支持半双工模式下的碰撞回退重传和 late collision。
- 支持全双工模式下的流控帧的发送。
- 支持帧长有效性检测，丢弃超长帧和超短帧。
- 支持对输入帧进行 CRC 校验，可丢弃校验错的帧。
- 支持对输出帧添加 CRC 校验。
- 支持短帧填充功能。
- 支持端口全双工模式下的外环回。
- 提供自适应功能。
- 提供 MDIO 接口，MDIO 接口时钟频率可调整。
- 提供 64 个接收和发送共用的帧管理队列。
- 提供流量限制功能，防止针对 CPU 的流量攻击。
- 支持对接收和发送帧进行统计计数。



- 支持 8 个可配置的 DMAC 地址过滤表。
- 对广播帧、组播帧和单播帧支持可配置是否转发或者丢弃。
- 支持 SG(Scatter Gather) 分散聚合功能。
- 接收、发送方向都支持 COE（Checksum Offload Engine）校验和卸载引擎功能。
- 支持 TSO（TCP Segment Offload）TCP 分段卸载功能。
- 支持 UFO（UDP Fragment Offload）UDP 分段卸载功能。

Ethernet 模块逻辑框图如图 5-1 所示。

图5-1 Ethernet 模块逻辑框图



5.3 信号描述

Ethernet 接口信号如表 5-1 和表 5-2 所示。

表5-1 MDIO 接口信号描述

信号名称	方向	描述	对应管脚
MDCK	O	MDIO 接口时钟输出。	MDCK
MDIO	I/O	MDIO 接口的输入/输出信号。	MDIO



表5-2 MII 接口信号描述

信号名称	方向	描述	对应管脚
RMII_REF_CLK	I/O	上行 RMII 接口参考时钟。	RMII_CLK
RMII_TXD[1:0]	O	RMII 发送数据。	RMII_TXD1~ RMII_TXD0
RMII_TXEN	O	RMII 发送数据有效。	RMII_TX_EN
RMII_RXD[1:0]	I	RMII 接口接收数据。	RMII_RXD1~ RMII_RXD0
RMII_CRS_DV	I	RMII 接口接收数据有效。	RMII_RX_DV

5.4 工作方式

5.4.1 收帧过程

初始化时软件需要完成以下操作：

- 软件需要申请与配置的输入队列深度个数相同的缓冲区，每个缓冲区的大小为 2KB，然后将缓冲区首地址依次写入到帧输入队列中，写入的次数应刚好等于配置的输入队列的深度。
- 软件需要保证配置的缓冲区在收帧时不能释放。如果配置的首地址为非 word 对齐地址时，需确保该地址对应的字节地址为可写地址。

当 CPU 获知有 1 帧数据需要接收时，执行以下步骤：

- 步骤 1. 读取寄存器 [UD_GLB_IQFRM_DES](#) 中的帧描述子（包含输入帧的起始地址和帧长度信息）。
- 步骤 2. 处理数据，并将 [GLB_IRQ_RAW](#)[iraw_rx_up]写 1 清 0（表示 CPU 收帧结束）。

----结束

软件收完 1 帧数据后需要重新申请一个 2KB 的缓冲区，并将首地址重新写入到当前输入队列帧描述子中。否则，实际可以使用的输入队列的深度不等于 CPU 配置的数值，而等于 CPU 实际分配的缓冲区个数。

CPU 接收帧描述子数据结构如表 5-3 所示。

表5-3 CPU 接收帧描述子数据结构

位	名称	描述
[63:32]	rxfrm_saddr	接收帧的起始地址。
[31:18]	reserved	保留



位	名称	描述
[17:12]	fd_in_addr	输入队列（iq）待接收帧的相对地址，作为帧存放绝对地址的索引值（0~iq_len-1）。
[11:0]	fd_in_len	输入队列待接收帧帧长度。

 说明

查询 [UD_GLB_ADDRQ_STAT](#) 可获得接收队列使用长度。

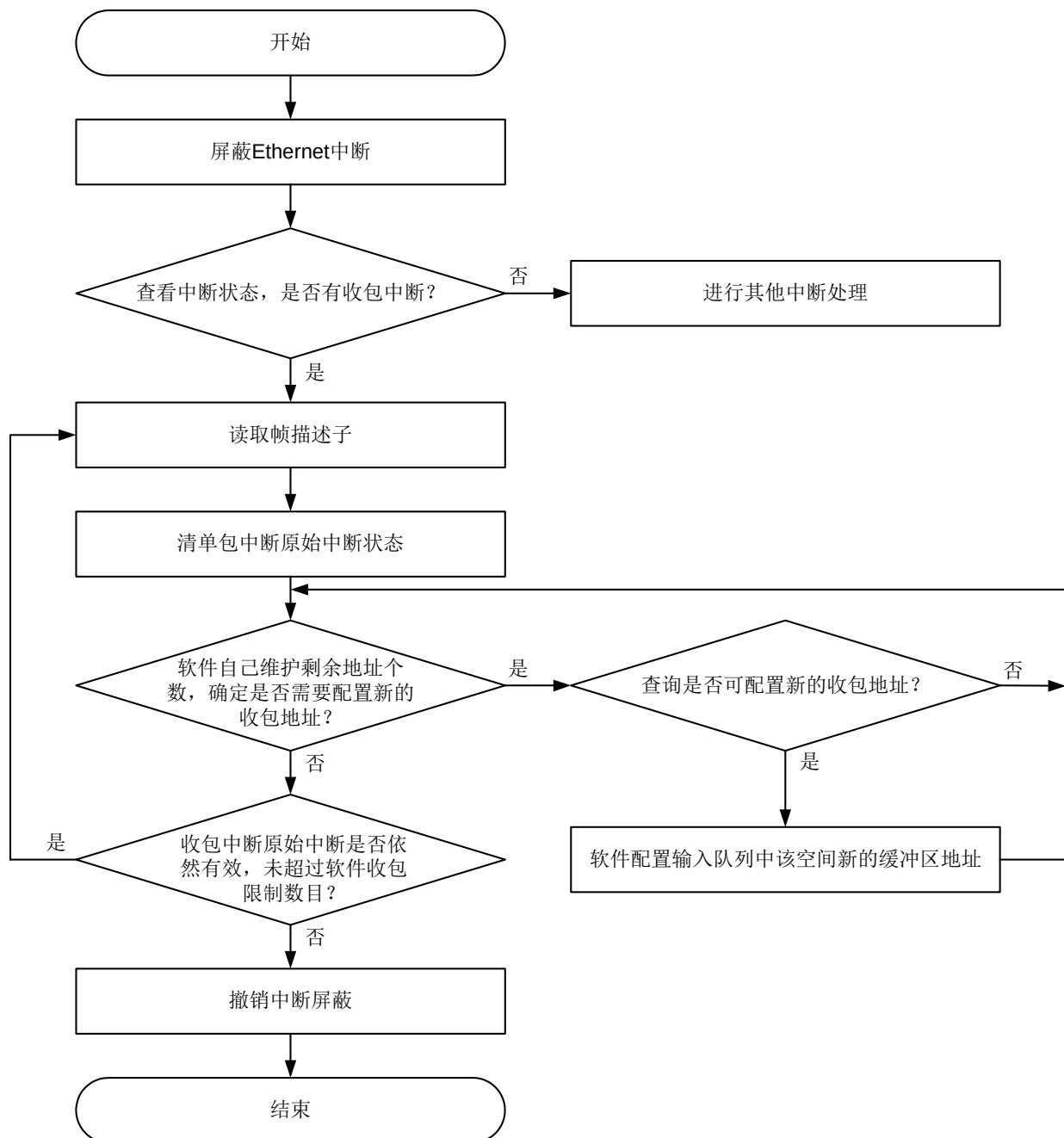
CPU 收帧可以采用中断方式或查询方式。

a. 中断方式收帧

CPU 使能帧接收中断时，根据是否有帧需要接收，硬件会产生帧接收中断（单包中断和多包中断）[int_rx_up](#)（每接收 1 个报文上报 1 个中断）和 [int_rxd_up](#)（每接收到指定报文数目时上报 1 个中断）通知 CPU 收帧。中断方式收帧流程如[图 5-2](#) 所示。



图5-2 中断方式收帧流程图

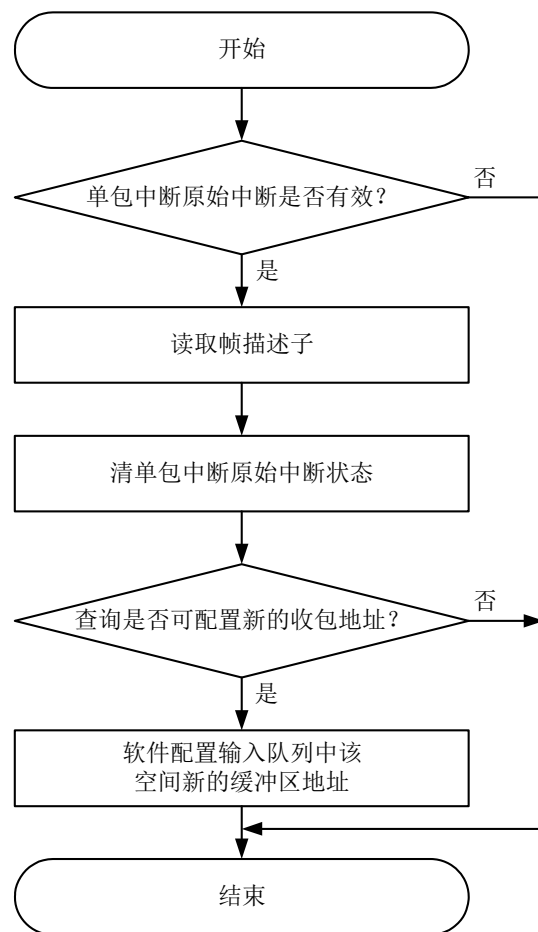


b. 查询方式收帧

CPU 不使能帧接收中断 `GLB_IRQ_ENA[ien_rx_up]`，CPU 自动查询 `GLB_IRQ_RAW[iraw_rx_up]`，为 1 表示有帧需要 CPU 接收。查询方式收帧流程如图 5-3 所示。



图5-3 查询方式收帧流程图



5.4.2 发帧过程

CPU 有帧待发送时，判断当前队列是否有空间发送。如果有空间发送，则分别写该帧 Buffer 的首地址和帧长到输出队列帧描述子中。CPU 先写入首地址，然后写入发送帧的帧长。写发送帧的帧长触发硬件将该输出帧的首地址和帧长信息写入到输出队列中等待发送。因此，软件应该控制不能任意写帧长寄存器，每写该寄存器一次会导致发送一个数据包。

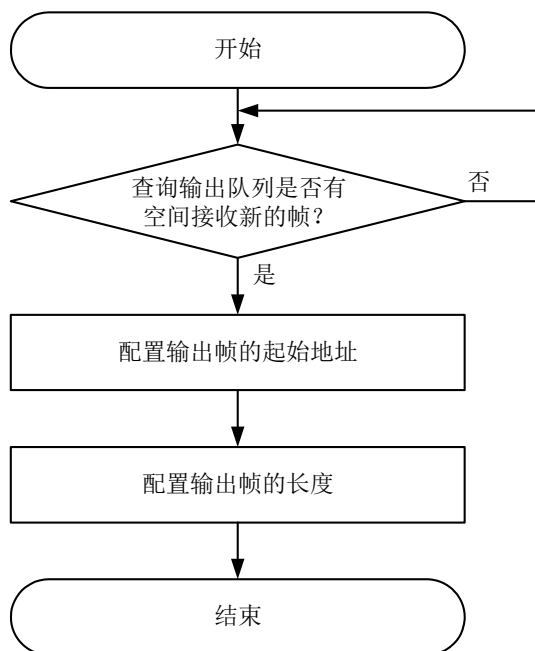
帧格式如下：

目的 MAC	源 MAC	Type	Data	FCS

CPU 发帧流程如图 5-4 所示。



图5-4 CPU 发帧流程图



CPU 发出的帧在 SDRAM 中缓存时，不包括帧描述子。写帧描述子到 [UD_GLB_EQ_ADDR](#) 及 [UD_GLB_EQFRM_LEN](#)，通知 Ethernet 将该帧（描述子）入队。CPU 发送帧描述子数据结构如表 5-4 所示。

表5-4 CPU 发送帧描述子数据结构

位	名称	描述
[63:32]	start_addr_eq	发送一个帧的首地址。
[31]	tso_flag	是否执行 TSO 操作 0: 不执行 TSO 操作 1: 执行 TSO 操作
[30]	vlan_flag	当前报文是否有 VLAN 标识 0: 当前报文无 VLAN 标识 1: 当前报文有 VLAN 标识
[29]	ip_version	当前报文 IP 协议版本 0: IPv4 1: IPv6
[28]	protocol_type	当前报文传输层协议类型 0: TCP 1: UDP



位	名称	描述
[27]	tx_coe_flag	是否执行 TxCOE 0: 不执行 TxCOE 操作 1: 执行 TxCOE 操作
[26]	sg_flag	是否执行 SG 操作 0: 不执行 SG 操作 1: 执行 SG 操作
[25:24]	reserve	保留
[23:20]	ip_hdr_len	当前报文 IP 首部长度的 word (4 字节) 为单位 若当前报文为 IPv4 报文, 其长度为 5~15word 若当前报文为 IPv6 报文, 其长度为 10word
[19:16]	protocol_hdr_len	若当前报文为 TCP 报文, 表示其 TCP 首部长度的 word (4 字节) 为单位, 取值为 5~15word; 若当前报文为 UDP 报文, 表示其 UDP 首部长度的 word (4 字节) 为单位, 且取值为 2word
[15:11]	nfrags_num	当前描述子指示的 SKB 中含有的 nfrags 的个数, 最大为 16 个
[10:0]	Length	当为 TSO 报文时, Length 代表切片报文的 MSS 若为 SG 或 BYPASS 报文, Length 代表整个帧长

注: length 小于 20byte 以及大于 1900byte 的帧将被丢弃, 即可发送范围在 20~1900 之间。

说明

查询 [UD_GLB_ADDRQ_STAT](#) 可获得当前 CPU 发送队列使用情况。

CPU 发帧可以采用中断方式或查询方式。

a. 中断方式发帧

CPU 使能 Ethernet 发包队列由不空到空中断 (int_freeeq_up), 同时设置允许该中断通知 CPU。此时如果 Ethernet 输出队列由不空到空, 说明 Ethernet 可以发送帧, 硬件产生中断通知 CPU, 表示 CPU 可以发帧。

如果软件需要发送帧, 但是当前输出队列已满, 则软件可以使能该中断。当输出队列为空后, 产生中断, 通知软件发送等待的帧。软件可以使用该中断一次发送一组帧, 同时在该中断有效时释放上次发送的一组帧的缓冲区。

b. 查询方式发帧

软件查询自己内部的发帧计数, 如果计数小于配置的输出队列长度, 则可以直接将待发送的帧通知 Ethernet, 同时建立对应的发帧索引表, 索引表项内容为写入到 Ethernet 输出队列的输出帧首地址。当 Ethernet 发送完 1 帧后, 通过输出队列的地址通知 CPU



释放对应的发送缓冲区，CPU 通过该输出队列的地址查找到对应的发送缓冲区并释放。

5.4.3 中断管理

中断状态寄存器

指示产生的中断类型，具体内容请参见“5.6.3 全局控制寄存器”中的“GLB_IRQ_STAT”。

中断使能寄存器

控制是否产生相应的中断，具体内容请参见“5.6.3 全局控制寄存器”中的“GLB_IRQ_ENA”。如果使能某中断，其中断状态会写入到相应的中断状态寄存器中。

原始中断状态寄存器

可以读取某种类型的原始中断传送给 CPU，具体内容请参见“5.6.3 全局控制寄存器”中的“GLB_IRQ_RAW”。在需要清除中断状态时，必须先清除该中断的原始中断，原始中断清除后，中断状态会自动清除。

5.4.4 流量控制

对于软件接收的帧，当单位时间间隔内接收到的帧超过软件配置上限时，则将后续接收的帧按配置进行选择性地丢弃。软件可配置超过流量限制后丢弃广播帧、组播帧或者单播帧（通过 UD_GLB_FC_DROPCTRL 配置）。流量限制阈值通过 UD_GLB_FC_RXLIMIT 设置。

软件可配置寄存器 UD_GLB_FC_TIMECTRL 来设置流量限制的时间间隔。使用 1 个 10bit 时间间隔寄存器，最大可以是 1023 个时间粒度。时间间隔粒度采用 17bit 计数器，对主时钟进行计数，默认值是 100000，对于 100MHz 主时钟时间粒度为 1ms。软件可配置（20bit 寄存器）流量限制上限值的大小，若配置为 0 表示不进行流量限制全部接收。

5.4.5 典型应用

时钟门控



说明

在不使用 ETH 模块时，可关断其时钟，以降低功耗。

关断 Ethernet 时钟的步骤如下：

- 步骤 1. 关闭端口链路状态，使 Ethernet 无法收发包。
- 步骤 2. 清除端口收包队列，使 Ethernet 再无收包中断上报。
- 步骤 3. 软件下发复位 Ethernet 逻辑命令，复位不撤消。
- 步骤 4. 设置 PERI_CRG91 [eth_cken] 为 0，关闭 Ethernet 时钟。

----结束



打开 Ethernet 时钟的步骤如下：

- 步骤 1. 保持复位不撤消。设置 PERI_CRG91[eth_cken_pre]为 1，打开 Ethernet 时钟。
- 步骤 2. 设置 PERI_CRG91 [hrst_eth_s]为 0，撤消复位。
- 步骤 3. 打开端口链路状态，Ethernet 恢复正常工作。

----结束

软复位

Ethernet 全局软复位的步骤如下：

- 步骤 1. 关闭 Ethernet 的端口链路状态以及收包中断使能，使软件无法收包，同时软件不再发包。
- 步骤 2. 等到软件处理完端口的当前收发包后，清除收发包队列，同时队列长度保持软复位之前的值，相关指针、队列计数值均要回 0。
- 步骤 3. 设置 PERI_CRG91 [hrst_eth_s]为 1，下发 Ethernet 软复位命令。
- 步骤 4. 设置 PERI_CRG91 [hrst_eth_s]为 0，撤消 Ethernet 软复位。
- 步骤 5. 如果要重新开始收发包，软件还需对端口收发包队列进行初始化配置。
- 步骤 6. 打开端口链路状态，Ethernet 恢复正常工作。

----结束

初始化

端口的初始化的步骤如下：

- 步骤 1. 配置端口状态获取方式。

Ethernet 可以自适应物理层芯片工作状态或者由软件配置其工作状态。在初始化时通过配置 UD_MAC_PORTSEL[stat_ctrl]进行选择：

- 当配置为 1 时，选择软件配置端口工作状态，执行步骤 2。
- 当配置为 0 时，选择自适应获取工作状态，执行步骤 3。

复位时选择软件设置端口工作状态。

- 步骤 2. 配置工作状态和物理层芯片工作状态。

- 如果配置端口通过软件配置工作状态，则软件需要根据实际运用环境配置 UD_MAC_PORTSET 中的速度、连接状态和双工模式状态信息，同时将这些信息配置到物理层芯片的相关寄存器中。
- Ethernet 提供 MDIO 接口，实现对物理层芯片的读写控制。软件操作时将数据、物理层芯片的地址、寄存器地址和相关控制信息写入到 MDIO_RWCTRL 寄存器中，然后查询到 MDIO_RWCTRL[finish]为 1，表示硬件完成对物理层芯片的读写操作。具体的配置内容需要参考相关物理层芯片的数据手册。

配置完成后执行步骤 4。



步骤 3. 配置自适应工作状态。

如果配置端口通过自适应方式获取物理层芯片的工作状态，需要指明物理层芯片速度、双工模式和有无连接寄存器的地址以及这些状态位各自寄存器中的偏移地址。通过配置 `UD_MDIO_ANEG_CTRL` 实现。

步骤 4. 接收和发送帧队列深度设置。

设置寄存器 `UD_GLB_QLEN_SET` 中的输入队列深度和输出队列深度：

- 输入队列深度表示接收数据时可以缓存的帧的最大的数量。
- 输出队列深度表示发送数据时可以缓存的帧的最大的数量。

由于输入队列和输出队列共用 64 个管理空间，因此设置的输入队列深度和输出队列深度之和不能超过 64，输入队列深度和输出队列深度至少为 1。如果软件设置的输入队列深度和输出队列深度之和大于 64，则优先保证软件先配置的项，另外一项则为 64 减去先配置项的结果。

另外，可以设置多包中断配置寄存器，设置寄存器 `UD_GLB_IRQN_SET[int_frm_cnt]` 来配置每隔多少个包上报一次多包中断，同时配置老化时间寄存器 `UD_GLB_IRQN_SET[int_timer]`。

步骤 5. 初始化接收帧队列缓冲区。

复位后，软件需要申请与配置的输入队列深度个数相同的缓冲区，每个缓冲区大小为 2KB，然后将缓冲区首地址依次写入到帧输入队列中，写入的次数应刚好等于配置的输入队列的深度。

步骤 6. 模块软复位。

软件配置 ETH 模块软复位，可以对 Ethernet 内部的逻辑电路和帧管理队列进行复位，使 ETH 模块回到初始化状态，但 Ethernet 内部所有的寄存器保留软复位前的值。软件解除软件复位后，需要重新申请收包缓冲区并对输入队列进行初始化，否则 Ethernet 不能接收网络包。



说明

- Ethernet 软复位后，软件设置的寄存器是不变的。具体哪些寄存器可以软复位，请参见每个寄存器的说明。

----结束

中断收帧流程

中断收帧的步骤如下：

步骤 1. 进入中断处理程序后，屏蔽 Ethernet 中断。

步骤 2. 查看中断状态 `GLB_IRQ_STAT[int_rx_up]` 是否有收帧中断。如果有，则转步骤 3；如果没有，则进行其他 Ethernet 中断处理。

步骤 3. 读取帧描述子 `UD_GLB_IQFRM_DES`，根据 `fd_in_addr` 对应的帧首地址读取对应长度 (`fd_in_len`) 的帧数据。

步骤 4. 清单包中断原始中断信号 `GLB_IRQ_RAW[iraw_rx_up]` 为 0，通知硬件接收完成。



- 步骤 5. 软件根据自己维护的剩余可用地址个数，确定是否需要配置新的收包地址。如果不需要配置，则转步骤 8。完成一次收包。
- 步骤 6. 读取 `UD_GLB_QSTAT[cpu_addr_in_rdy]`，查看是否可配新的收包地址。如果不可配，则返回步骤 5。
- 步骤 7. 软件通过寄存器 `UD_GLB_IQ_ADDR` 将新的缓冲区地址配置给输入队列。返回步骤 5。
- 步骤 8. 读取并判断单包中断原始中断信号 `GLB_IRQ_RAW[iraw_rx_up]`。如果该位有效且软件可继续收包（未超过收包限制数目），则返回步骤 3。
- 步骤 9. 撤消 Ethernet 中断屏蔽。

----结束

查询收帧流程

不使能帧接收中断 `GLB_IRQ_ENA[ien_cpu_rx]`，CPU 自动查询单包中断原始中断信号 `GLB_IRQ_RAW[iraw_rx_up]`，为 1 表示有帧需要接收。

查询收帧的步骤如下：

- 步骤 1. 读取单包中断原始中断信号 `GLB_IRQ_RAW[iraw_rx_up]`。如果该位无效，则直接结束。
- 步骤 2. 读取帧描述子 `UD_GLB_IQFRM_DES`，根据 `fd_in_addr` 对应的帧首地址读取对应长度（`fd_in_len`）的帧数据。
- 步骤 3. 写 1 清除单包中断原始中断信号 `GLB_IRQ_RAW[iraw_rx_up]`，通知硬件接收完成。
- 步骤 4. 读取 `UD_GLB_QSTAT[cpu_addr_in_rdy]`，查看是否可配新的收包地址。如果无效，则直接结束。
- 步骤 5. 软件通过寄存器 `UD_GLB_IQ_ADDR` 将新的缓冲区地址配置给输入队列。

----结束

发帧流程

发帧的步骤如下：

- 步骤 1. 读取 `UD_GLB_ADDRQ_STAT[eq_in_rdy]`，检查 Ethernet 输出队列是否有剩余空间接收新的发送帧。如果没有，则继续等待并查询。
- 步骤 2. 配置待输出帧的首地址 `UD_GLB_EQ_ADDR`。
- 步骤 3. 配置待输出帧的长度 `UD_GLB_EQFRM_LEN`。完成 1 帧的发送配置。

----结束



5.5 寄存器概览

MDIO 控制寄存器

MDIO 控制寄存器概览如表 5-5 所示。

表5-5 MDIO 控制寄存器概览（基地址是 0x1001_0000）

偏移地址	名称	描述	页码
0x1100	MDIO_RWCTRL	MDIO 命令字寄存器	5-17
0x1104	MDIO_RO_DATA	MDIO 读数据寄存器	5-18

MAC 控制寄存器

MAC 控制寄存器概览如表 5-6 所示。

表5-6 MAC 控制寄存器概览（基地址是 0x1001_0000）

偏移地址	名称	描述	页码
0x0200	UD_MAC_PORTSEL	端口工作状态控制寄存器	5-21
0x0204	UD_MAC_RO_STAT	端口状态寄存器	5-21
0x0208	UD_MAC_PORTSET	端口设定工作状态寄存器	5-22
0x020C	UD_MAC_STAT_CHANGE	端口状态改变指示寄存器	5-22
0x0210	UD_MAC_SET	MAC 功能设置寄存器	5-23

全局控制寄存器

Ethernet 全局控制寄存器概览如表 5-7 所示。

表5-7 全局控制寄存器概览（基地址是 0x1001_0000）

偏移地址	名称	描述	页码
0x1300	GLB_HOSTMAC_L32	本机 MAC 地址低 32bit 寄存器	5-27
0x1304	GLB_HOSTMAC_H16	本机 MAC 地址寄存器高 16bit	5-27
0x1308	GLB_SOFT_RESET	内部软复位寄存器	5-27
0x1310	GLB_FWCTRL	转发控制寄存器	5-27



偏移地址	名称	描述	页码
0x1314	GLB_MACTCTRL	MAC 过滤表控制寄存器	5-28
0x1318	GLB_ENDIAN_MOD	大小端控制寄存器	5-29
0x1330	GLB_IRQ_STAT	中断状态寄存器	5-29
0x1334	GLB_IRQ_ENA	中断使能寄存器	5-31
0x1338	GLB_IRQ_RAW	原始中断寄存器	5-33
0x1400	GLB_MAC0_L32	MAC 过滤器 0	5-34
0x1404	GLB_MAC0_H16	MAC 过滤器 0	5-34
0x1408	GLB_MAC1_L32	MAC 过滤器 1	5-35
0x140C	GLB_MAC1_H16	MAC 过滤器 1	5-35
0x1410	GLB_MAC2_L32	MAC 过滤器 2	5-35
0x1414	GLB_MAC2_H16	MAC 过滤器 2	5-35
0x1418	GLB_MAC3_L32	MAC 过滤器 3	5-36
0x141C	GLB_MAC3_H16	MAC 过滤器 3	5-36
0x1420	GLB_MAC4_L32	MAC 过滤器 4	5-37
0x1424	GLB_MAC4_H16	MAC 过滤器 4	5-37
0x1428	GLB_MAC5_L32	MAC 过滤器 5	5-37
0x142C	GLB_MAC5_H16	MAC 过滤器 5	5-38
0x1430	GLB_MAC6_L32	MAC 过滤器 6	5-38
0x1434	GLB_MAC6_H16	MAC 过滤器 6	5-38
0x1438	GLB_MAC7_L32	MAC 过滤器 7	5-39
0x143C	GLB_MAC7_H16	MAC 过滤器 7	5-39
0x0340	UD_GLB_IRQN_SET	多包中断配置寄存器	5-40
0x0344	UD_GLB_QLEN_SET	队列长度配置寄存器	5-40
0x0348	UD_GLB_FC_LEVEL	流控控制寄存器	5-41
0x034C	UD_GLB_CAUSE	报文上送 CPU 的 Cause 寄存器	5-41
0x0350	UD_GLB_RXFRM_SADDR	输入帧起始地址寄存器	5-42
0x0354	UD_GLB_IQFRM_DES	输入帧描述子寄存器	5-42
0x0358	UD_GLB_IQ_ADDR	输入帧首地址寄存器	5-42



偏移地址	名称	描述	页码
0x035C	UD_GLB_BFC_STAT	转发缓存流控状态和多包中断老化时间计数器	5-43
0x0360	UD_GLB_EQ_ADDR	输出队列首地址寄存器	5-43
0x0364	UD_GLB_EQFRM_LEN	输出队列帧长配置寄存器	5-44
0x0368	UD_GLB_QSTAT	队列状态寄存器	5-44
0x036C	UD_GLB_ADDRQ_STAT	地址队列状态寄存器	5-45
0x0370	UD_GLB_FC_TIMECTRL	流量限制时间配置寄存器	5-45
0x0374	UD_GLB_FC_RXLIMIT	流量限制水线配置寄存器	5-46
0x0378	UD_GLB_FC_DROPCTRL	流量限制丢包控制寄存器	5-46
0x0380	UD_GLB_RX_COE_EN	接收方向 coe 使能开关寄存器	5-47

统计计数控制寄存器

统计计数控制寄存器概览如表 5-8 所示。

表5-8 统计计数控制寄存器概览（基地址是 0x1001_0000）

偏移地址	名称	描述	页码
0x0584	UD_STS_PORTCNT	端口部分状态计数器	5-47
0x05A0	UD_PORT2CPU_PKTS	CPU 接收总包数寄存器	5-48
0x05A4	UD_CPU2IQ_ADDRCNT	CPU 配置收包地址队列次数寄存器	5-48
0x05A8	UD_RX_IRQCNT	上报单包中断次数寄存器	5-48
0x05AC	UD_CPU2EQ_PKTS	CPU 发往发送端口总包数寄存器	5-49

统计结果寄存器

统计结果寄存器概览如表 5-9 所示。

表5-9 统计结果寄存器概览（基地址是 0x1001_0000）

偏移地址	名称	描述	页码
0x0600	UD_RX_DVCNT	RXDV 上升沿次数寄存器	5-49
0x0604	UD_RX_OCTS	接收的总字节数寄存器	5-49



偏移地址	名称	描述	页码
0x0608	UD_RX_RIGHTOCTS	接收正确包总字节数寄存器	5-50
0x060C	UD_HOSTMAC_PKTS	匹配本机 MAC 的包数寄存器	5-50
0x0610	UD_RX_RIGHTPKTS	端口接收的总包数寄存器	5-50
0x0614	UD_RX_BROADPKTS	正确的广播包数寄存器	5-50
0x0618	UD_RX_MULTPKTS	正确的组播包数寄存器	5-51
0x061C	UD_RX_UNIPKTS	正确的单播包数寄存器	5-51
0x0620	UD_RX_ERRPKTS	错包总包数寄存器	5-51
0x0624	UD_RX_CRCERR_PKTS	CRC 错误次数寄存器	5-51
0x0628	UD_RX_LENERR_PKTS	包长无效的包数寄存器	5-52
0x062C	UD_RX_OCRCERR_PKTS	奇数 Nibble CRC 错的包数寄存器	5-52
0x0630	UD_RX_PAUSE_PKTS	接收的流控包数寄存器	5-52
0x0634	UD_RF_OVERCNT	RXFIFO 溢出次数寄存器	5-52
0x0638	UD_FLUX_TOL_IPKTS	流量限制入口的总包数寄存器	5-53
0x063C	UD_FLUX_TOL_DPKTS	流量限制丢弃的总包数寄存器	5-53
0x064C	UD_MN2CPU_PKTS	由于 MAC 限制不转发给 CPU 端口的包数寄存器	5-53
0x0780	UD_TX_PKTS	发送成功的总包数寄存器	5-53
0x0784	UD_TX_BROADPKTS	发送成功的广播包数寄存器	5-54
0x0788	UD_TX_MULTPKTS	发送成功的组播包数寄存器	5-54
0x078C	UD_TX_UNIPKTS	发送成功的单播包数寄存器	5-54
0x0790	UD_TX_OCTS	发送的总字节数寄存器	5-54
0x0794	UD_TX_PAUSE_PKTS	发送的流控帧数目寄存器	5-55
0x0798	UD_TX_RETRYCNT	发送过程中总重传次数寄存器	5-55



偏移地址	名称	描述	页码
0x079C	UD_TX_COLCNT	发生冲突的总次数寄存器	5-55
0x07A0	UD_TX_LC_PKTS	发生 LateCollision 的包数寄存器	5-55
0x07A4	UD_TX_COLOK_PKTS	有冲突但发送成功的包数寄存器	5-55
0x07A8	UD_TX_RETRY15_PKTS	重传超 15 次丢弃的包数寄存器	5-56
0x07AC	UD_TX_RETRYN_PKTS	冲突次数等于域值的包数寄存器	5-56

5.6 寄存器描述

5.6.1 MDIO 控制寄存器

MDIO_RWCTRL

MDIO_RWCTRL 为 MDIO 命令字寄存器。该寄存器不支持软复位。

Offset Address: 0x1100

Total Reset Value: 0x0000_8000

Bits	Access	Name	Description	Reset
[31:16]	RW	cpu_data_in	MDIO 模块对 PHY 进行写操作的数据。 进行写操作时，CPU 将要对 MDIO 写入的 16bit 数据先写入到该寄存器中。	0x0000
[15]	RW	finish	PHY 的读/写操作完成。 0: 未完成; 1: 完成。 当要进行第 2 次读/写操作时，CPU 要先对该位进行清 0。	0x1
[14]	-	reserved	保留。	0x0
[13]	RW	rw	控制对 PHY 的访问为读操作还是写操作。 0: 读操作; 1: 写操作。	0x0
[12:8]	RW	phy_exaddr	对外部操作的 PHY 的对应外部物理地址。 1 个 MDIO 可以对外面的多个 PHY 进行读写访问。每个 PHY 有 1 个相应的地址。当	0x00



Bits	Access	Name	Description	Reset
			外部只连接 1 个 PHY 芯片时，其等价于 UD_MDIO_PHYADDR[phy0_addr] 或者 UD_MDIO_PHYADDR[phy1_addr] 。	
[7:5]	RW	frq_dv	对外部 PHY 进行读写操作时，对 MDC（MDIO 接口时钟）的分频系数。 以主时钟频率 100MHz 为例，frq_dv 与 MDC 频率对应关系如下： 000：对工作主时钟 50 分频，分频后的频率为 2MHz； 001：对工作主时钟 100 分频，分频后的频率为 1MHz； 010：对工作主时钟 200 分频，分频后的频率为 500kHz； 011：对工作主时钟 400 分频，分频后的频率为 250kHz； 100：对工作主时钟 800 分频，分频后的频率为 125kHz； 101：对工作主时钟 1600 分频，分频后的频率为 62.5kHz； 110：对工作主时钟 3200 分频，分频后的频率为 31.25kHz； 111：对工作主时钟 6400 分频，分频后的频率为 15.625kHz。	0x0
[4:0]	RW	phy_inaddr	对外部操作的 PHY 芯片的内部寄存器地址，用 5bit 二进制数表示。	0x00

MDIO_RO_DATA

MDIO_RO_DATA MDIO 为读数据寄存器。该寄存器不支持软复位。

Offset Address: 0x1104 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RO	cpu_data_out	用于 MDIO 模块对 PHY 进行读操作的数据寄存器，将从 PHY 读取的 16bit 数据先写入到该寄存器中。	0x0000



UD_MDIO_PHYADDR

UD_MDIO_PHYADDR 为 PHY 物理地址寄存器。该寄存器不支持软复位。

Offset Address: 0x0108 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x00000000
[4:0]	RW	phy_addr	外部连接的物理层 PHY 芯片的外部物理地址。	0x01

UD_MDIO_RO_STAT

UD_MDIO_RO_STAT 为 PHY 芯片状态寄存器。该寄存器不支持软复位。

Offset Address: 0x010C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2]	RO	speed_mdio2mac	从 MDIO 接口得到的端口速度工作状态，处于 10Mbit/s 或 100Mbit/s 工作方式。 0: 10Mbit/s 工作方式； 1: 100Mbit/s 工作方式。	0x0
[1]	RO	link_mdio2mac	从 MDIO 接口得到的端口链接状态。 0: 无链接； 1: 有链接。	0x0
[0]	RO	duplex_mdio2mac	从 MDIO 接口得到的端口双工工作状态。 0: 半双工； 1: 全双工。	0x0

UD_MDIO_ANEG_CTRL

UD_MDIO_ANEG_CTRL PHY 芯片各状态的偏移地址设置寄存器。该寄存器不支持软复位。



说明

如果 PHY 芯片速度状态位于其地址为 17 的寄存器的 bit[14]，即可配置 internal_addr_speed 为 0x11、speed_index 为 0xE，则此时可通过 MDIO 接口将该 bit 值读出作为 PHY 当前工作的速度模式信息。

Offset Address: 0x0110 Total Reset Value: 0x0463_1EA9



Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x00
[26:22]	RW	internal_addr_speed	PHY 芯片存储状态信息（速度）的寄存器地址。缺省值按照 Intel 9785 设置。	0x11
[21:17]	RW	internal_addr_link	PHY 芯片存储状态信息（链接）的寄存器地址。缺省值按照 Intel 9785 设置。	0x11
[16:12]	RW	internal_addr_duplex	PHY 芯片存储状态信息（双工）的寄存器地址。缺省值按照 Intel 9785 设置。	0x11
[11:8]	RW	speed_index	PHY 芯片状态寄存器中存储速度信息的偏移地址。缺省值按照 Intel 9785 设置。	0xE
[7:4]	RW	link_index	PHY 芯片状态寄存器中存储链接信息的偏移地址。缺省值按照 Intel 9785 设置。	0xA
[3:0]	RW	duplex_index	PHY 芯片状态寄存器中存储双工信息的偏移地址。缺省值按照 Intel 9785 设置。	0x9

UD_MDIO_IRQENA

UD_MDIO_IRQENA 为端口状态变化扫描屏蔽寄存器。该寄存器不支持软复位。

说明

- 对于端口所连接 PHY 状态信息无法通过配置 [UD_MDIO_ANEG_CTRL](#) 进行扫描获取的，用户可通过 [MDIO_RWCTRL](#) 扫描 PHY 状态寄存器来获取端口状态是否发生变化，并产生中断通知软件进行处理。
- link_partner 状态变化是指 PHY 状态的 link、speed、duplex 任意一位发生变化。

Offset Address: 0x0114 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RW	link_partner_ch_mask	端口 link partner 状态扫描变化中断屏蔽。 0: 屏蔽; 1: 不屏蔽。	0x0
[2]	RW	speed_ch_mask	端口速度模式扫描变化中断屏蔽。 0: 屏蔽; 1: 不屏蔽。	0x0
[1]	RW	link_ch_mask	端口连接模式扫描变化中断屏蔽。 0: 屏蔽; 1: 不屏蔽。	0x0



Bits	Access	Name	Description	Reset
[0]	RW	duplex_ch_mask	端口双工模式扫描变化中断屏蔽。 0: 屏蔽; 1: 不屏蔽。	0x0

5.6.2 MAC 控制寄存器

MAC 控制寄存器为端口控制类寄存器。当端口状态有效时，对 MAC 控制寄存器进行配置后，需要进行一次软件复位。

UD_MAC_PORTSEL

UD_MAC_PORTSEL 为端口工作状态控制寄存器。该寄存器不支持软复位。

Offset Address: 0x0200 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	stat_ctrl	端口工作状态信息选择控制寄存器。 0: 使用从 MDIO 接口获得的状态信息; 1: 使用 CPU 设定的状态信息。	0x1

UD_MAC_RO_STAT

UD_MAC_RO_STAT 为端口状态寄存器。该寄存器不支持软复位。

Offset Address: 0x0204 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2]	RO	speed_stat	端口当前速度模式。 0: 10Mbit/s 模式; 1: 100Mbit/s 模式。	0x0
[1]	RO	link_stat	端口当前链接状态。 0: 无链接; 1: 有链接。	0x0
[0]	RO	duplex_stat	端口当前双工模式。 0: 半双工; 1: 全双工。	0x0



UD_MAC_PORTSET

UD_MAC_PORTSET 为端口设定工作状态寄存器。该寄存器不支持软复位。

Offset Address: 0x0208 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2]	RW	speed_stat_dio	CPU 设定端口速度模式。 0: 10Mbit/s 模式; 1: 100Mbit/s 模式。	0x0
[1]	RW	link_stat_dio	CPU 设定端口链接状态。 0: 无链接; 1: 有链接。	0x0
[0]	RW	duplex_stat_dio	CPU 设定端口双工模式。 0: 半双工; 1: 全双工。	0x0

UD_MAC_STAT_CHANGE

UD_MAC_STAT_CHANGE 为端口状态改变指示寄存器。该寄存器不支持软复位。

Offset Address: 0x020C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2]	WC	speed_stat_ch	端口速度模式改变指示。 0: 无改变; 1: 改变。 该寄存器写 1 清 0。	0x0
[1]	WC	link_stat_ch	端口链接状态改变指示。 0: 无改变; 1: 改变。 该寄存器写 1 清 0。	0x0
[0]	WC	duplex_stat_ch	端口双工模式改变指示。 0: 无改变; 1: 改变。	0x0



Bits	Access	Name	Description	Reset
			该寄存器写 1 清 0。	

UD_MAC_SET

UD_MAC_SET 为 MAC 功能设置寄存器。该寄存器不支持软复位。

Offset Address: 0x0210 Total Reset Value: 0x2027_55EE

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29]	RW	add_pad_en	端口发送时自动添加 PAD 使能。 0: 禁止; 1: 使能。	0x1
[28]	RW	crcgen_dis	端口 CRC 生成禁止控制。 0: 输出帧重新计算 CRC; 1: 输出帧不重新计算 CRC。	0x0
[27]	RW	cntr_rdclr_en	端口统计计数器读清空使能。 0: 禁止; 1: 使能。	0x0
[26]	RW	cntr_clr_all	端口统计计数器清空控制。 0: 不清空; 1: 清空。  说明 若 cntr_clr_all 为 1, 则在下次全清时需要置 0 后再置为 1 才会执行全清操作。	0x0
[25]	RW	cntr_roll_dis	端口统计非循环计数使能。 0: 禁止; 1: 使能。	0x0
[24:21]	RW	colthreshold	端口冲突次数统计阈值。 缺省为 0x1, 表示出现一次冲突的帧的个数。	0x1
[20]	RW	in_loop_en	端口内环回使能。 0: 禁止; 1: 使能。  说明 外环回和内环回不可同时配置使能。当网口处于正常工作状态时, 内、外环回配置改变之后	0x0



Bits	Access	Name	Description	Reset
			需对模块进行软件复位。	
[19]	RW	ex_loop_en	端口外环回使能。 0: 禁止; 1: 使能。 说明 外环回和内环回不可同时配置使能。当网口处于正常工作状态时, 内、外环回配置改变之后需对模块进行软件复位。	0x0
[18]	RW	pause_en	端口流控帧发送使能。 0: 禁止; 1: 使能。	0x1
[17]	RW	rx_shframe_en	端口短帧接收使能。 0: 禁止; 1: 使能。 说明 rx_shframe_en 配置为 1 时, 端口允许接收的最小帧长为 rx_min_thr 配置帧长; rx_shframe_en 配置为 0 时, 端口允许接收的最小帧长缺省为 64byte (含 CRC)。	0x1
[16:11]	RW	rx_min_thr	端口允许接收的最小帧长。 取值范围为 42byte~63byte, 缺省为 42byte。 说明 若 rx_min_thr 配置值小于 42, 将取值为 42。	0x2A
[10:0]	RW	len_max	端口允许接收的最大帧长, 缺省为 1518byte。 取值范围为 1518byte~1535byte。 说明 若 len_max 配置大于 2000, 将取值 2000; 若 len_max 配置小于 256, 将取值 256。	0x5EE

UD_MAC_EEE_INT

UD_MAC_EEE_INT 为 EEE 原始中断寄存器。

Offset Address: 0x0480 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x000000



Bits	Access	Name	Description	Reset
[4]	RO	tx_entry_start	ETH TX方向允许进入LPI的原始中断。	0x0
[3]	RO	rx_leave_lpi	RX方向退出LPI的原始中断。	0x0
[2]	RO	rx_entry_lpi	RX方向进入LPI的原始中断。	0x0
[1]	RO	tx_leave_lpi	TX方向退出LPI的原始中断。	0x0
[0]	RO	tx_entry_lpi	TX方向进入LPI的原始中断。	0x0

UD_MAC_EEE_INTEN

UD_MAC_EEE_INTEN 为 EEE 中断使能寄存器。

Offset Address: 0x0484 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	RO	reserved	保留。	0x000000
[9]	RW	tx_entry_start_msk	ETH TX方向允许进入LPI的原始中断掩码。	0x0
[8]	RW	rx_leave_lpi_msk	RX方向退出LPI的原始中断掩码。	0x0
[7]	RW	rx_entry_lpi_msk	RX方向进入LPI的原始中断掩码。	0x0
[6]	RW	tx_leave_lpi_msk	TX方向退出LPI的原始中断掩码。	0x0
[5]	RW	tx_entry_lpi_msk	TX方向进入LPI的原始中断掩码。	0x0
[4]	RO	tx_entry_start_en	ETH TX方向允许进入LPI的原始中断使能。	0x0
[3]	RO	rx_leave_lpi_en	RX方向退出LPI的原始中断使能。	0x0
[2]	RO	rx_entry_lpi_en	RX方向进入LPI的原始中断使能。	0x0
[1]	RO	tx_leave_lpi_en	TX方向退出LPI的原始中断使能。	0x0
[0]	RO	tx_entry_lpi_en	TX方向进入LPI的原始中断使能。	0x0

UD_MAC_EEE_ENA

UD_MAC_EEE_ENA 为 EEE 使能寄存器。

Offset Address: 0x0488 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:4]	RW	eee_ls_timer	LS_TIMER	0x0000000
[3:2]	RW	reserved	保留。	0x0
[1]	RW	eee_assert	EEE 进入配置。	0x0
[0]	RW	eee_enable	EEE使能开关。	0x0

UD_MAC_EEE_TIMER

UD_MAC_EEE_TIMER 为 EEE 需要设置的 TIMER 寄存器。

Offset Address: 0x048C Total Reset Value: 0x001E_2710

Bits	Access	Name	Description	Reset
[31:16]	RW	tx_wk_timer	TX_WK_TIMER	0x001E
[15:0]	RW	lpi_cond_timer	LPI_COND_TIMER	0x2710

UD_MAC_EEE_LINK_STATUS

UD_MAC_EEE_LINK_STATUS 为专为 EEE 设置的网口 link 状态寄存器。

Offset Address: 0x0490 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x0000_0000
[0]	RW	phy_link_status	PHY link 状态。	0x0

UD_MAC_EEE_CLK_CNT

UD_MAC_EEE_CLK_CNT 为 EEE 时钟单元计数器。

Offset Address: 0x0494 Total Reset Value: 0x0000_0063

Bits	Access	Name	Description	Reset
[31:0]	RW	eee_clk_cnt	EEE时钟单元计数器。	0x0000_0063



5.6.3 全局控制寄存器

GLB_HOSTMAC_L32

GLB_HOSTMAC_L32 为本机 MAC 地址低 32bit 寄存器。该寄存器不支持软复位。

Offset Address: 0x1300 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	local_mac	本机 MAC 的低 32bit。	0x0000_0000

GLB_HOSTMAC_H16

GLB_HOSTMAC_H16 为本机 MAC 地址高 16bit 寄存器。该寄存器不支持软复位。

Offset Address: 0x1304 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	local_mac[47:32]	本机 MAC 的高 16bit。	0x0000

GLB_SOFT_RESET

GLB_SOFT_RESET 为内部软复位寄存器。该寄存器不支持软复位。

Offset Address: 0x1308 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	soft_reset	内部软复位。 0: 不复位; 1: 复位。 进入软复位状态后, 若要撤消软复位, 需将该位置 0。	0x0



说明

所有软复位时间须保持 2ms 以上。

GLB_FWCTRL

GLB_FWCTRL 为转发控制寄存器。该寄存器不支持软复位。



Offset Address: 0x1310 Total Reset Value: 0x0000_0020

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RW	fwall2cpu_up	端口所有输入的有效帧是否都强制转发给 CPU 端口。 0: 否; 1: 是。	0x0
[6]	-	reserved	保留。	0x0
[5]	RW	fw2cpu_ena_up	端口输入帧转发给 CPU 端口使能。 0: 禁止; 1: 使能。	0x1
[4:0]	-	reserved	保留。	0x00

GLB_MACTCTRL

GLB_MACTCTRL 为 MAC 过滤表控制寄存器。该寄存器不支持软复位。

说明

- 目的 MAC 地址最高字节为偶数是单播帧。
- 目的 MAC 地址最高字节为奇数是组播帧。
- 目的 MAC 地址所有字节为全 0xFF 的是广播帧。

Offset Address: 0x1314 Total Reset Value: 0x0000_0020

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RW	mact_ena_up	端口所有 MAC 过滤器总使能位。 0: 不使用任何 MAC 过滤器; 1: 使用 MAC 过滤器。	0x0
[6]	-	reserved	保留。	0x0
[5]	RW	broad2cpu_up	端口输入的广播帧是否转发给 CPU 端口。 0: 不转发给 CPU 端口; 1: 转发给 CPU 端口。	0x1
[4]	-	reserved	保留。	0x0
[3]	RW	multi2cpu_up	端口输入的非过滤表中的组播帧是否转发给 CPU 端口。 0: 不转发给 CPU 端口;	0x0



Bits	Access	Name	Description	Reset
			1: 转发给 CPU 端口。	
[2]	-	reserved	保留。	0x0
[1]	RW	uni2cpu_up	端口输入的非过滤表中的单播帧是否转发给 CPU 端口。 0: 不转发给 CPU 端口; 1: 转发给 CPU 端口。	0x0
[0]	-	reserved	保留。	0x0

GLB_ENDIAN_MOD

GLB_ENDIAN_MOD 为大小端控制寄存器。该寄存器不支持软复位。

Offset Address: 0x1318 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x0000_0000
[1]	RW	in_edian	收包写 SDRAM 大小端配置。 0: big-endian 模式; 1: little-endian 模式。 数据按字节翻转。	0x1
[0]	RW	out_edian	发包读 SDRAM 大小端配置。 0: big-endian 模式; 1: little-endian 模式。	0x1

GLB_IRQ_STAT

GLB_IRQ_STAT 为中断状态寄存器。该寄存器不支持软复位。

Offset Address: 0x1330 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RO	int_mdio_finish	MDIO 完成 CPU 操作指示。 0: MDIO 未完成 CPU 操作; 1: MDIO 已完成 CPU 操作, 产生中断。 进入该中断程序后, 查询 MDIO_RWCTRL[finish] , 判断 MDIO 操作	0x0



Bits	Access	Name	Description	Reset
			是否完成。	
[11:8]	-	reserved	保留。	0x0
[7]	RO	int_rxd_up	端口有（多个）帧等待 CPU 接收中断指示（多包中断）。 0：中断无效； 1：中断有效，输入队列中有帧等待 CPU 接收。 当发生该中断后，需要查询 GLB_IRQ_RAW[iraw_rxd_up] 信号是否有效来判断是否有帧等待接收。	0x0
[6]	RO	int_freeeq_up	端口输出队列由不空到空中断，表示输出队列缓冲区由不空到空，指示 CPU 可以写入新的待发送的一组帧。 0：没有产生中断； 1：输出队列由不空到空中断。 进入该中断程序后，软件通过查询 UD_GLB_ADDRQ_STAT[eq_cnt] 判断当前输出队列是否为空，如果不为空，说明该中断为无效中断。	0x0
[5]	RO	int_stat_up	端口状态改变中断信号，表示 MDIO 自适应到 PHY 芯片的速度变化、双工模式变化和连接状态变化时产生中断。 0：没有产生中断； 1：端口状态改变中断。 进入该中断程序后，软件根据 UD_MDIO_IRQENA 的配置值决定是什么状态发生变化。	0x0
[4]	RO	int_duplex_up	端口双工模式变化中断指示。 0：双工模式没有变化； 1：双工模式发生变化，产生中断。 进入该中断程序后，查询 UD_MAC_STAT_CHANGE[duplex_stat_ch]，判断是否发生双工状态改变。	0x0
[3]	RO	int_speed_up	端口速度模式变化中断指示。 0：速度模式没有变化； 1：速度模式发生变化，产生中断。 进入该中断程序后，查询 UD_MAC_STAT_CHANGE[speed_stat_ch]，判断是否发生速度状态改变。	0x0



Bits	Access	Name	Description	Reset
[2]	RO	int_link_up	端口连接状态变化中断指示。 0: 连接状态没有变化; 1: 连接状态发生变化, 产生中断。 进入该中断程序后, 查询 UD_MAC_STAT_CHANGE [link_stat_ch], 判断是否发生连接状态改变。	0x0
[1]	RO	int_tx_up	端口发送完来自 CPU 的 1 帧数据指示。 0: 未发送完; 1: 已发送完, 产生中断。 当发生该中断后, 需要查询 UD_GLB_QSTAT 中的输出队列当前的出 队地址 eq_out_index, 判断是否释放输出帧 的缓存空间。	0x0
[0]	RO	int_rx_up	端口有帧等待 CPU 接收中断指示。 0: 中断无效; 1: 中断有效, 输入队列中有帧等待 CPU 接收。 当发生该中断后, 需要查询 GLB_IRQ_RAW [iraw_rxd_up]信号是否有 效来判断是否有帧等待接收。	0x0

GLB_IRQ_ENA

GLB_IRQ_ENA 为中断使能寄存器。该寄存器不支持软复位。

Offset Address: 0x1334 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000000
[19]	RW	ien_all	所有中断使能。 0: 禁止 (所有中断都不能上报); 1: 使能 (所有中断按配置对应上报)。	0x0
[18]	RW	ien_up	所有上行端口中断使能。 0: 禁止 (所有上行端口中断都不能上 报); 1: 使能 (所有上行端口中断按配置对应上 报)。	0x0
[17:13]	-	reserved	保留。	0x00



Bits	Access	Name	Description	Reset
[12]	RW	ien_mdio_finish	MDIO 完成 CPU 操作指示使能。 0: 禁止; 1: 使能。	0x0
[11:8]	-	reserved	保留。	0x0
[7]	RW	ien_rxd_up	上行端口有（多个）帧等待 CPU 接收使能（多包中断）。 0: 禁止; 1: 使能。	0x0
[6]	RW	ien_freeeq_up	上行端口输出队列由不空到空中断信号使能。 0: 禁止; 1: 使能。	0x0
[5]	RW	ien_stat_up	上行端口状态改变中断信号使能。 0: 禁止; 1: 使能。	0x0
[4]	RW	ien_duplex_up	上行端口双工模式变化中断使能。 0: 禁止; 1: 使能。	0x0
[3]	RW	ien_speed_up	上行端口速度模式变化中断使能。 0: 禁止; 1: 使能。	0x0
[2]	RW	ien_link_up	上行端口连接状态变化中断使能。 0: 禁止; 1: 使能。	0x0
[1]	RW	ien_tx_up	上行端口发送完来自 CPU 的 1 帧数据指示使能。 0: 禁止; 1: 使能。	0x0
[0]	RW	ien_rx_up	上行端口有帧等待 CPU 接收使能。 0: 禁止; 1: 使能。	0x0



GLB_IRQ_RAW

GLB_IRQ_RAW 为原始中断寄存器。该寄存器不支持软复位，是写 1 清 0 寄存器。

Offset Address: 0x1338 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x000000
[12]	WC	iraw_mdio_finish	MDIO 完成 CPU 操作的原始中断状态。 0: 无中断; 1: 有中断。	0x0
[11:8]	-	reserved	保留。	0x0
[7]	WC	iraw_rxd_up	上行端口有（多个）帧等待 CPU 接收的原始中断状态（多包中断）。 0: 无中断; 1: 有中断。	0x0
[6]	WC	iraw_freeeq_up	上行端口输出队列由不空到空的原始中断状态，表示输出队列缓冲区由不空到空，指示 CPU 可以写入新的待发送的一组帧。 0: 无中断; 1: 有中断。	0x0
[5]	WC	iraw_stat_up	上行端口状态改变的原始中断状态，表示 MDIO 自适应到 PHY 芯片的速度变化、双工模式变化和连接状态变化时产生中断。 0: 无中断; 1: 有中断。	0x0
[4]	WC	iraw_duplex_up	上行端口双工模式变化的原始中断状态。 0: 无中断; 1: 有中断。	0x0
[3]	WC	iraw_speed_up	上行端口速度模式变化的原始中断状态。 0: 中断无效; 1: 速度模式发生变化，产生中断。 该寄存器写 1 清 0。	0x0
[2]	WC	iraw_link_up	上行端口连接状态变化的原始中断状态。 0: 无中断; 1: 有中断。	0x0
[1]	WC	iraw_tx_up	上行端口发送完来自 CPU 的原始中断状态。	0x0



Bits	Access	Name	Description	Reset
			0: 无中断; 1: 有中断。	
[0]	WC	iraw_rx_up	上行端口有帧等待 CPU 接收的原始中断状态。 0: 无中断; 1: 有中断。	0x0

GLB_MAC0_L32

GLB_MAC0_L32 为过滤表 MAC0 的低 32bit 寄存器。

Offset Address: 0x1400 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	flt_mac0	过滤表 MAC0 的低 32bit。	0x0000_0000

GLB_MAC0_H16

GLB_MAC0_H16 为过滤表 MAC0 的高 16bit 寄存器。

Offset Address: 0x1404 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:22]	-	reserved	保留。	0x000
[21]	RW	fw2cpu_up	上行端口使能本过滤器时，上行端口输入帧命中本过滤器是否转发给 CPU 端口。 0: 否; 1: 是。	0x0
[20:18]	-	reserved	保留。	0x0
[17]	RW	mac0_up	本过滤器配置为上行端口使用控制。 0: 上行端口不使用该过滤器; 1: 上行端口使用该过滤器。	0x0
[16]	-	reserved	保留。	0x0
[15:0]	RW	flt_mac0	过滤表 MAC0 的高 16bit。	0x0000



GLB_MAC1_L32

GLB_MAC1_L32 为过滤表 MAC1 的低 32bit 寄存器。

Offset Address: 0x1408 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	flt_mac1	过滤表 MAC1 的低 32bit。	0x0000_0000

GLB_MAC1_H16

GLB_MAC1_H16 为过滤表 MAC1 的高 16bit 寄存器。

Offset Address: 0x140C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:22]	-	reserved	保留。	0x000
[21]	RW	fw2cpu_up	上行端口使能本过滤器时，上行端口输入帧命中本过滤器是否转发给 CPU 端口。 0: 否； 1: 是。	0x0
[20:18]	RO	reserved	保留。	0x0
[17]	RW	mac1_up	本过滤器配置为上行端口使用控制。 0: 上行端口不使用该过滤器； 1: 上行端口使用该过滤器。	0x0
[16]	RO	reserved	保留。	0x0
[15:0]	RW	flt_mac1	过滤表 MAC1 的高 16bit。	0x0000

GLB_MAC2_L32

GLB_MAC2_L32 为过滤表 MAC2 的低 32bit 寄存器。

Offset Address: 0x1410 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	flt_mac2	过滤表 MAC2 的低 32bit。	0x0000_0000

GLB_MAC2_H16

GLB_MAC2_H16 为过滤表 MAC2 的高 16bit 寄存器。



Offset Address: 0x1414 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:22]	-	reserved	保留。	0x000
[21]	RW	fw2cpu_up	上行端口使能本过滤器时，上行端口输入帧命中本过滤器是否转发给 CPU 端口。 0: 否； 1: 是。	0x0
[20:18]	RO	reserved	保留。	0x0
[17]	RW	mac2_up	本过滤器配置为上行端口使用控制。 0: 上行端口不使用该过滤器； 1: 上行端口使用该过滤器。	0x0
[16]	-	reserved	保留。	0x0
[15:0]	RW	flt_mac2	过滤表 MAC2 的高 16bit。	0x0000

GLB_MAC3_L32

GLB_MAC3_L32 为过滤表 MAC3 的低 32bit 寄存器。

Offset Address: 0x1418 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	flt_mac3	过滤表 MAC3 的低 32bit。	0x0000_0000

GLB_MAC3_H16

GLB_MAC3_H16 为过滤表 MAC3 的高 16bit 寄存器。

Offset Address: 0x141C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:22]	-	reserved	保留。	0x000
[21]	RW	fw2cpu_up	上行端口使能本过滤器时，上行端口输入帧命中本过滤器是否转发给 CPU 端口。 0: 否； 1: 是。	0x0
[20:18]	-	reserved	保留。	0x0



[17]	RW	mac3_up	本过滤器配置为上行端口使用控制。 0: 上行端口不使用该过滤器; 1: 上行端口使用该过滤器。	0x0
[16]	-	reserved	保留。	0x0
[15:0]	RW	flt_mac3	过滤表 MAC3 的高 16bit。	0x0000

GLB_MAC4_L32

GLB_MAC4_L32 为过滤表 MAC4 的低 32bit 寄存器。

Offset Address: 0x1420 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	flt_mac4	过滤表 MAC4 的低 32bit。	0x0000_0000

GLB_MAC4_H16

GLB_MAC4_H16 为过滤表 MAC4 的高 16bit 寄存器。

Offset Address: 0x1424 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:22]	-	reserved	保留。	0x000
[21]	RW	fw2cpu_up	上行端口使能本过滤器时，上行端口输入帧命中本过滤器是否转发给 CPU 端口。 0: 否; 1: 是。	0x0
[20:18]	-	reserved	保留。	0x0
[17]	RW	mac4_up	本过滤器配置为上行端口使用控制。 0: 上行端口不使用该过滤器; 1: 上行端口使用该过滤器。	0x0
[16]	-	reserved	保留。	0x0
[15:0]	RW	flt_mac4	过滤表 MAC4 的高 16bit。	0x0000

GLB_MAC5_L32

GLB_MAC5_L32 为过滤表 MAC5 的低 32bit 寄存器。



Offset Address: 0x1428 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	flt_mac5	过滤表 MAC5 的低 32bit。	0x0000_0000

GLB_MAC5_H16

GLB_MAC5_H16 为过滤表 MAC5 的高 16bit 寄存器。

Offset Address: 0x142C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:22]	-	reserved	保留。	0x000
[21]	RW	fw2cpu_up	上行端口使能本过滤器时，上行端口输入帧命中本过滤器是否转发给 CPU 端口。 0: 否； 1: 是。	0x0
[20:18]	-	reserved	保留。	0x0
[17]	RW	mac5_up	本过滤器配置为上行端口使用控制。 0: 上行端口不使用该过滤器； 1: 上行端口使用该过滤器。	0x0
[16]	-	reserved	保留。	0x0
[15:0]	RW	flt_mac5	过滤表 MAC5 的高 16bit。	0x0000

GLB_MAC6_L32

GLB_MAC6_L32 为过滤表 MAC6 的低 32bit 寄存器。

Offset Address: 0x1430 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	flt_mac6	过滤表 MAC6 的低 32bit。	0x0000_0000

GLB_MAC6_H16

GLB_MAC6_H16 为过滤表 MAC6 的高 16bit 寄存器。

Offset Address: 0x1434 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:22]	-	reserved	保留。	0x000
[21]	RW	fw2cpu_up	上行端口使能本过滤器时，上行端口输入帧命中本过滤器是否转发给 CPU 端口。 0: 否； 1: 是。	0x0
[20:18]	-	reserved	保留。	0x0
[17]	RW	mac6_up	本过滤器配置为上行端口使用控制。 0: 上行端口不使用该过滤器； 1: 上行端口使用该过滤器。	0x0
[16]	-	reserved	保留。	0x0
[15:0]	RW	flt_mac6	过滤表 MAC6 的高 16bit。	0x0000

GLB_MAC7_L32

GLB_MAC7_L32 为过滤表 MAC7 的低 32bit 寄存器。

Offset Address: 0x1438 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	flt_mac7	过滤表 MAC7 的低 32bit。	0x0000_0000

GLB_MAC7_H16

GLB_MAC7_H16 为过滤表 MAC7 的高 16bit 寄存器。

Offset Address: 0x143C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:22]	-	reserved	保留。	0x000
[21]	RW	fw2cpu_up	上行端口使能本过滤器时，上行端口输入帧命中本过滤器是否转发给 CPU 端口。 0: 否； 1: 是。	0x0
[20:18]	-	reserved	保留。	0x0
[17]	RW	mac7_up	本过滤器配置为上行端口使用控制。 0: 上行端口不使用该过滤器；	0x0



Bits	Access	Name	Description	Reset
			1: 上行端口使用该过滤器。	
[16]	-	reserved	保留。	0x0
[15:0]	RW	flt_mac7	过滤表 MAC7 的高 16bit。	0x0000

UD_GLB_IRQN_SET

UD_GLB_IRQN_SET 为多包中断配置寄存器。该寄存器不支持软复位。

Offset Address: 0x0340 Total Reset Value: 0x0800_003A

Bits	Access	Name	Description	Reset
[31:29]	RO	reserved	保留。	0x0
[28:24]	RW	int_frm_cnt	该设置用于多包中断功能，定义为每收到多少个包上报一次多包中断。  说明 int_frm_cnt 最小值可配置为 1，此时多包中断等同于单包中断。	0x00
[23:16]	RO	reserved	保留。	0x00
[15:0]	RW	age_timer	打开多包中断功能后，每经过一定的时间收包数量还未达到上报多包中断所要求的包数，该时间段定义为多包中断产生的老化时间。  说明 age_timer 以主时钟 256 分频周期为计数单位。	0x003A

UD_GLB_QLEN_SET

UD_GLB_QLEN_SET 为队列长度配置寄存器。该寄存器不支持软复位。

Offset Address: 0x0344 Total Reset Value: 0x0000_2020

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x0000
[13:8]	RW	iq_len	输入（收包）队列长度配置。  说明 iq_len 不得配置为 0，否则将会强行置 1；配置的 iq_len 和 eq_len 之和不得大于 64，否则优先满足 iq_len 的值（非 0），eq_len 取 64-iq_len。	0x20
[7:6]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[5:0]	RW	eq_len	输出（发包）队列长度配置。  说明 eq_len 不得配置为 0，否则将会强行置 1。	0x20

UD_GLB_FC_LEVEL

UD_GLB_FC_LEVEL 为流控制寄存器。该寄存器不支持软复位。

Offset Address: 0x0348 Total Reset Value: 0x3018_0508

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x6030
[14]	RW	qlimit_ena	接收队列流控使能。 0: 禁止（不根据接收队列状态来发送流控信息）； 1: 使能（可根据接收队列状态来发送流控信息）。	0x0
[13:8]	RW	qlimit_up	接收队列流控上限。当输入队列剩余可用队列空间数小于该上限值时，若收包队列流控使能，则发送流控信息给对端。  说明 - 若上限值 qlimit_up 配置为 0，将无法进入流控状态； - 上限值 qlimit_up 必须比下限值 qlimit_down 小。	0x05
[7:6]	-	reserved	保留。	0x0
[5:0]	RW	qlimit_down	接收队列解除流控下限。当输入队列剩余可用队列空间数大于或者等于该下限值同时正处于接收流控状态时，解除当前流控状态。	0x08

UD_GLB_CAUSE

UD_GLB_CAUSE 为报文上送 CPU 的 Cause 寄存器。该寄存器不支持软复位。

Offset Address: 0x034C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x0000_0000



Bits	Access	Name	Description	Reset
[2:0]	RO	mact_cause	报文查 MAC 表的匹配结果类型。 000: 强制转发; 001: 目的 MAC 是本机 MAC 的报文; 010: 广播报文; 011: 匹配 MAC 表的报文; 100: 不匹配 MAC 表的组播报文; 101: 不匹配 MAC 表的单播报文; 其他: 保留。	0x0

UD_GLB_RXFRM_SADDR

UD_GLB_RXFRM_SADDR 为输入帧起始地址寄存器。该寄存器不支持软复位。

Offset Address: 0x0350 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	rxfrm_saddr	接收帧的起始地址。	0x0000_0000

UD_GLB_IQFRM_DES

UD_GLB_IQFRM_DES 为输入帧描述子寄存器。该寄存器不支持软复位。

Offset Address: 0x0354 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:18]	-	reserved	保留。	0x0000_0000
[17:12]	RO	fd_in_addr	输入队列 (iq) 中等待接收的第一个帧的相对地址, 作为帧存放绝对地址的索引值 (0~iq_len-1)。	0x00
[11:0]	RO	fd_in_len	输入队列待接收帧的帧长度。	0x000

UD_GLB_IQ_ADDR

UD_GLB_IQ_ADDR 为输入帧首地址寄存器。该寄存器不支持软复位。



说明

如果软件分配的地址非 word 对齐时，逻辑会按照 word 地址写入数据，只是前面写入的数据为无效数据。例如，配置的帧首地址为 0xF000_8002（非 word 对齐地址），逻辑会在 0xF000_8000 和 0xF000_8001 地址都写入 0x00（也可能是其他数据），在 0xF000_8002 写入接收帧的第 1 个字节（有效数据），在 0xF000_8003 写入接收帧的第 2 个字节（有效数据），后续数据依次写入缓冲区。如果是其他非 word 对齐地址，操作方式类似。

Offset Address: 0x0358 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	startaddr_iq	CPU 配置的输入帧对应的存储空间的首地址。接收帧根据此地址来申请总线。	0x0000_0000

UD_GLB_BFC_STAT

UD_GLB_BFC_STAT 为转发缓存流控状态和多包中断老化时间计数器。该寄存器不支持软复位。

Offset Address: 0x035C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	timerover_cnt	多包中断老化时间计数器溢出（计数到配置值）次数计数器。 说明 单位时间内如果 timerover_cnt 值过大，说明 UD_GLB_IRQN_SET[int_frm_cnt] 设置不合理，多包中断主要由老化时间触发，需要减少配置值。	0x0000
[15:0]	RO	flowctrl_cnt	上行端口（或者下行端口）的转发缓存进入流控状态次数计数器。 说明 单位时间内如果 flowctrl_cnt 值过大，说明 UD_GLB_FC_LEVEL[blimit_up] 或者 UD_GLB_FC_LEVEL[blimit_down] 设置过小，或者外部网络状态恶化，可能需要减少配置值。	0x0000

UD_GLB_EQ_ADDR

UD_GLB_EQ_ADDR 为输出队列首地址寄存器。该寄存器不支持软复位。



说明

如果输出帧的首地址为非 word 对齐时，逻辑会按照 word 地址读入数据，只是前面读取的数据为无效数据而丢弃。例如，配置的发送帧首地址为 0xF000_8102（非 word 对齐地址），逻辑会把从地址 0xF000_8100 和 0xF000_8101 读入的字节数据直接丢弃，在 0xF000_8102 读入的数据当作发送帧的第 1 个字节（有效数据），在 0xF000_8103 读入的数据为发送帧的第 2 个字节（有效数据），后续数据都为有效数据（直到读取指定帧长的数据）。如果是其他非 word 对齐地址，操作方式类似。

Offset Address: 0x0360 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	add_fd_addr_out	CPU 添加到输出队列的发送帧的首地址。	0x00000000

UD_GLB_EQFRM_LEN

UD_GLB_EQFRM_LEN 为输出队列帧长配置寄存器。该寄存器不支持软复位。

Offset Address: 0x0364 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留。	0x000000
[10:0]	RW	add_fd_len_out	CPU 添加到输出队列的发送帧的帧长。 配置该寄存器触发硬件将发送帧的首地址和帧长写入到输出队列中等待发送，软件发送帧时必须先写入帧的首地址，然后再写入该帧的帧长。 注意：add_fd_len_out 小于 20byte 以及大于 1600byte 的帧将被丢弃，即可发送范围在 20byte~1600byte 之间。	0x000

UD_GLB_QSTAT

UD_GLB_QSTAT 为队列状态寄存器。该寄存器不支持软复位。

Offset Address: 0x0368 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x000000
[29:24]	RO	iq_in_index	输入（收包）队列入队 index。	0x000
[23:22]	-	reserved	保留。	
[21:16]	RO	cpuw_index	输入（收包）队列帧首地址入队 index。	
[15:14]	-	reserved	保留。	



Bits	Access	Name	Description	Reset
[13:8]	RO	eq_in_index	输出（发包）队列描述子入队 index。	
[7:6]	-	reserved	保留。	
[5:0]	RO	eq_out_index	输出（发包）队列描述子出队 index。	

UD_GLB_ADDRQ_STAT

UD_GLB_ADDRQ_STAT 为地址队列状态寄存器。该寄存器不支持软复位。

Offset Address: 0x036C Total Reset Value: 0x0300_0000

Bits	Access	Name	Description	Reset
[31:26]	RO	reserved	保留。	0x01
[25]	RO	cpuaddr_in_rdy	CPU 是否可配置输入队列帧首地址。 0: 否; 1: 是。  说明 cpuaddr_in_rdy、eq_in_rdy 复位时为 0，但复位后将立即被电路置为 1。即复位后 iq 地址队列和 eq 描述子队列可配。	0x1
[24]	RO	eq_in_rdy	CPU 是否可配置输出队列帧描述子（帧首地址及帧长） 0: 否; 1: 是。	0x0
[23:22]	-	reserved	保留。	0x0
[21:16]	RO	cpu_cnt	输入队列对 CPU 分配的可用帧首地址计数。	0x00
[15:14]	-	reserved	保留。	0x0
[13:8]	RO	iq_cnt	输入队列已经使用长度（0~iq_len）。	0x00
[7:6]	-	reserved	保留。	0x0
[5:0]	RO	eq_cnt	输出队列已经使用长度（0~eq_len）。	0x00

UD_GLB_FC_TIMECTRL

UD_GLB_FC_TIMECTRL 为流量限制时间配置寄存器。该寄存器不支持软复位。

Offset Address: 0x0370 Total Reset Value: 0x07FF_86A0



Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x00
[26:17]	RW	flux_timer_cfg	流量限制时间间隔计数器，对 flux_timer_inter 产生的分频时钟进行计数。配置为 0 不进行流量限制。	0x3FF
[16:0]	RW	flux_timer_inter	流量限制时间粒度计数器，对 ETH 工作时钟（请参见 PERI_CRG91 寄存器）进行计数，默认值为 100000。 例如，当 ETH 工作时钟为 54MHz 时，如需配置粒度为 1ms，则寄存器设置为 54000，即 $1\text{ms}/(1/54\text{MHz})=[1 \times 10^4(-3)]/[(1/54) \times 10^4(-6)]=54 \times 10^3=54000$ 。	0x186A0

UD_GLB_FC_RXLIMIT

UD_GLB_FC_RXLIMIT 为流量限制水线配置寄存器。该寄存器不支持软复位。

Offset Address: 0x0374 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19:0]	RW	flux_cfg	流量限制上限阈值寄存器，该组对流量限制时间间隔内的软件接收帧数进行限制，超过该配置值的帧根据配置选择性进行丢弃或者接收。当配置为全 0 时，表示不进行流量限制。	0x00000

UD_GLB_FC_DROPCTRL

UD_GLB_FC_DROPCTRL 为流量限制丢包控制寄存器。该寄存器不支持软复位。

Offset Address: 0x0378 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x0000000
[2]	RW	flux_uni	流量限制超过上限值时是否丢弃单播包。 0: 否; 1: 是。	0x0
[1]	RW	flux_multi	流量限制超过上限值时是否丢弃组播包。 0: 否;	0x0



Bits	Access	Name	Description	Reset
			1: 是。	
[0]	RW	flux_broad	流量限制超过上限值时是否丢弃广播包。 0: 否; 1: 是。	0x0

UD_GLB_RX_COE_EN

UD_GLB_RX_COE_EN 为接收方向 coe 使能开关寄存器。该寄存器不支持软复位。

Offset Address: 0x0380 Total Reset Value: 0x8000_E000

Bits	Access	Name	Description	Reset
[31]	RW	rx_coe_en	接收方向 coe 使能开关。	0x1
[30:16]	-	reserved	保留。	0x0000
[15]	RW	iphdr_drop	ipv4 ip 头 checksum 错报文丢弃使能开关。 0: 透传 1: 丢包	0x1
[14]	RW	pro_drop	tcp/udp checksum 错报文丢弃使能开关。 0: 透传 1: 丢包	0x1
[13]	RW	i6udp_drop	ipv6 udp checksum 域为 0 非法报文丢弃使能开关。 0: 透传 1: 丢包	0x1
[12:0]	-	reserved	保留。	0x0000

5.6.4 统计计数控制寄存器

UD_STS_PORTCNT

UD_STS_PORTCNT 为端口部分状态计数器。该寄存器不支持软复位。

Offset Address: 0x0584 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:28]	RO	rxsof_cnt	端口接收到帧头计数。	0x0



[27:24]	RO	rxeof_cnt	端口接收到帧尾计数。	0x0
[23:20]	RO	rxcrcok_cnt	端口接收到 CRC 校验正确计数。	0x0
[19:16]	RO	rxcrbad_cnt	端口接收到 CRC 校验错误计数。	0x0
[15:12]	RO	txsof_cnt	端口发送帧头计数。	0x0
[11:8]	RO	txeof_cnt	端口发送帧尾计数。	0x0
[7:4]	RO	txcrcok_cnt	端口发送 CRC 校验正确计数。	0x0
[3:0]	RO	txcrbad_cnt	端口发送 CRC 校验错误计数。	0x0

UD_PORT2CPU_PKTS

UD_PORT2CPU_PKTS 为 CPU 接收上行或下行端口总包数寄存器。该寄存器不支持软复位。

Offset Address: 0x05A0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	WC	pkts_cpu	CPU 端口接收到的所有包数，该寄存器写 0 清 0，写 1 无效。	0x0000

UD_CPU2IQ_ADDRCNT

UD_CPU2IQ_ADDRCNT 为 CPU 配置收包地址队列次数寄存器。该寄存器不支持软复位。

Offset Address: 0x05A4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	WC	addr_cpu	CPU 成功配置收包地址队列的次数，该寄存器写 0 清 0，写 1 无效。	0x0000

UD_RX_IRQCNT

UD_RX_IRQCNT 为上行或下行端口上报单包中断次数寄存器。该寄存器不支持软复位。

Offset Address: 0x05A8 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	WC	pkts_port	上报的收帧中断计数，该寄存器写 0 清 0，写 1 无效。	0x0000

UD_CPU2EQ_PKTS

UD_CPU2EQ_PKTS 为 CPU 发往上行或下行发送端口总包数寄存器。该寄存器不支持软复位。

Offset Address: 0x05AC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	WC	pkts_cpu2tx	CPU 端口发往端口所有报文计数，该寄存器写 0 清 0，写 1 无效。	0x0000

5.6.5 统计结果寄存器

统计结果可配置只读和读清 0 两种模式（UD_MAC_SET[cnt_rdcclr_en] 设置为 1 表示读清 0 模式，设置为 0 表示只读模式），以下对寄存器的介绍针对只读模式。

UD_RX_DVCNT

UD_RX_DVCNT 为 RXDV 上升沿次数寄存器。该寄存器不支持软复位。

Offset Address: 0x0600 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	rxdvrise	所有的 rxdv 上升沿的计数。	0x00000000

UD_RX_OCTS

UD_RX_OCTS 为接收的总字节数寄存器。该寄存器不支持软复位。

Offset Address: 0x0604 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RO	ifinoctets	所有接收字节的计数，包括正确帧、错误帧以及 Preamble 中的字节。未检测到有效 SFD（Start of Frame Delimiter）的帧不进行统计。	0x00000000

UD_RX_RIGHTOCTS

UD_RX_RIGHTOCTS 为接收正确包总字节数寄存器。该寄存器不支持软复位。

Offset Address: 0x0608 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	octets_rx	接收字节计数，包括正确帧和错误帧，但不包括 Preamble 字节。未检测到有效 SFD 的帧不进行统计。	0x00000000

UD_HOSTMAC_PKTS

UD_HOSTMAC_PKTS 为匹配本机 MAC 的包数寄存器。该寄存器不支持软复位。

Offset Address: 0x060C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	local_mac_match	接收帧中目的 MAC 和本机 MAC 地址相同的正确帧的计数，不包括超短帧、超长帧、CRC 错帧、流控帧和传输错帧。	0x00000000

UD_RX_RIGHTPKTS

UD_RX_RIGHTPKTS 为端口接收的总包数寄存器。该寄存器不支持软复位。

Offset Address: 0x0610 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	pkts	所有帧的计数。	0x00000000

UD_RX_BROADPKTS

UD_RX_BROADPKTS 为正确的广播包数寄存器。该寄存器不支持软复位。

Offset Address: 0x0614 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RO	broadcastpkts	帧长有效并且 CRC 正确的广播帧的计数，但不包括流控帧、传输错帧。	0x00000000

UD_RX_MULTPKTS

UD_RX_MULTPKTS 为正确的组播包数寄存器。该寄存器不支持软复位。

Offset Address: 0x0618 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	multicastpkts	帧长有效并且 CRC 正确的组播帧的计数，但不包括流控帧、传输错帧。	0x00000000

UD_RX_UNIPKTS

UD_RX_UNIPKTS 为正确的单播包数寄存器。该寄存器不支持软复位。

Offset Address: 0x061C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	ifinucastpkts	帧长有效并且 CRC 正确的单播帧的计数，但不包括流控帧、传输错帧。	0x00000000

UD_RX_ERRPKTS

UD_RX_ERRPKTS 为错包总包数寄存器。该寄存器不支持软复位。

Offset Address: 0x0620 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	ifinerrors	接收的所有错帧计数，包括 CRC 错帧、超短帧、超长帧和传输错帧。	0x00000000

UD_RX_CRCERR_PKTS

UD_RX_CRCERR_PKTS 为 CRC 错误次数寄存器。该寄存器不支持软复位。

Offset Address: 0x0624 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RO	crcerr	接收帧的帧长有效（非超短、超长帧），但其 CRC 或 Alignment 检查出错的帧的个数。	0x00000000

UD_RX_LENERR_PKTS

UD_RX_LENERR_PKTS 为包长无效的包数寄存器。该寄存器不支持软复位。

Offset Address: 0x0628 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	abnormalsizepkts	帧长无效（小于设定的最小有效帧长，或者帧长大于设定的最大有效帧长）的个数（超短帧、超长帧）。	0x00000000

UD_RX_OCRCERR_PKTS

UD_RX_OCRCERR_PKTS 为奇数 Nibble 的 CRC 错的包数寄存器。该寄存器不支持软复位。

Offset Address: 0x062C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	dot3alignmenterr	接收到的奇数个 Nibble 的 CRC 错帧。	0x00000000

UD_RX_PAUSE_PKTS

UD_RX_PAUSE_PKTS 为接收的流控包数寄存器。该寄存器不支持软复位。

Offset Address: 0x0630 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	dot3pause	接收的流控帧数。	0x00000000

UD_RF_OVERCNT

UD_RF_OVERCNT 为 RXFIFO 溢出次数寄存器。该寄存器不支持软复位。

Offset Address: 0x0634 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RO	dropevents	帧接收的过程中，RXFIFO 溢出事件的累计次数。	0x00000000

UD_FLUX_TOL_IPKTS

UD_FLUX_TOL_IPKTS 为流量限制入口的总包数寄存器。该寄存器不支持软复位。

Offset Address: 0x0638 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	flux_frame_cnt	流量限制输入正确帧的总数计数，不包括超短帧、超长帧、CRC 错帧、流控帧和传输错帧。	0x00000000

UD_FLUX_TOL_DPKTS

UD_FLUX_TOL_DPKTS 为流量限制丢弃的总包数寄存器。该寄存器不支持软复位。

Offset Address: 0x063C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	flux_drop_cnt	流量限制丢弃正确帧的计数，不包括超短帧、超长帧、CRC 错帧、流控帧和传输错帧。	0x00000000

UD_MN2CPU_PKTS

UD_MN2CPU_PKTS 为 MAC 限制不转发给 CPU 端口的包数寄存器。该寄存器不支持软复位。

Offset Address: 0x064C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	mac_not2cpu_pkts	由于 MAC 限制不转发 CPU 端口的包数。	0x00000000

UD_TX_PKTS

UD_TX_PKTS 为发送成功的总包数寄存器。该寄存器不支持软复位。

Offset Address: 0x0780 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RO	pkts_tx	所有配置发送帧的计数，不包含超时丢弃帧以及 UD_GLB_EQFRM_LEN 配置长度不在有效范围之内的发送帧。	0x00000000

UD_TX_BROADPKTS

UD_TX_BROADPKTS 为发送成功的广播包数寄存器。该寄存器不支持软复位。

Offset Address: 0x0784 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	broadcastpkts_tx	发送成功的广播帧的计数（不含重传）。	0x00000000

UD_TX_MULTPKTS

UD_TX_MULTPKTS 为发送成功的组播包数寄存器。该寄存器不支持软复位。

Offset Address: 0x0788 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	multicastpkts_tx	发送成功的组播帧的计数（不含重传）。	0x00000000

UD_TX_UNIPKTS

UD_TX_UNIPKTS 为发送成功的单播包数寄存器。该寄存器不支持软复位。

Offset Address: 0x078C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	ifoutucastpkts_tx	发送成功的单播帧的计数（不含重传）。	0x00000000

UD_TX_OCTS

UD_TX_OCTS 为发送的总字节数寄存器。该寄存器不支持软复位。

Offset Address: 0x0790 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	octets_tx	发送的总字节数，包含重发帧、正确帧和错误帧，但 Preamble 字节不计在内。	0x00000000



UD_TX_PAUSE_PKTS

UD_TX_PAUSE_PKTS 为发送的流控帧数目寄存器。该寄存器不支持软复位。

Offset Address: 0x0794 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	dot3outpause	发送的流控帧数目。	0x00000000

UD_TX_RETRYCNT

UD_TX_RETRYCNT 为发送过程中总重传次数寄存器。该寄存器不支持软复位。

Offset Address: 0x0798 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	retry_times_tx	配置发送帧的总的重传次数。	0x00000000

UD_TX_COLCNT

UD_TX_COLCNT 为发生冲突的总次数寄存器。该寄存器不支持软复位。

Offset Address: 0x079C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	collisions	冲突发生的次数。	0x00000000

UD_TX_LC_PKTS

UD_TX_LC_PKTS 为发生 LateCollision 的包数寄存器。该寄存器不支持软复位。

Offset Address: 0x07A0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	dot3latecol	发生 Latecollision 的包数目。	0x00000000

UD_TX_COLOK_PKTS

UD_TX_COLOK_PKTS 为有冲突但发送成功的包数寄存器。该寄存器不支持软复位。

Offset Address: 0x07A4 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RO	dot3col_ok	发生冲突后发送成功的包数目。	0x00000000

UD_TX_RETRY15_PKTS

UD_TX_RETRY15_PKTS 为重传超 15 次丢弃的包数寄存器。该寄存器不支持软复位。

Offset Address: 0x07A8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	dot3excessivecol	由于重传次数大于 15 而丢弃的包数目。	0x00000000

UD_TX_RETRYN_PKTS

UD_TX_RETRYN_PKTS 为冲突次数等于域值的包数寄存器。该寄存器不支持软复位。

Offset Address: 0x07AC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	dot3colcnt	冲突次数等于冲突阈值的包数目，由 UD_MAC_SET[colthreshold] 设定。	0x00000000



目 录

6 视频编解码.....6-1

6.1 VEDU.....6-1

6.1.1 概述 6-1

6.1.2 特点 6-1

6.1.3 功能描述 6-4

6.2 JPGE/JPGD.....6-6

6.2.1 概述 6-6

6.2.2 特点 6-6

6.2.3 功能描述 6-8

6.2.4 工作方式 6-9



插图目录

图 6-1 VEDU 编码功能框图.....	6-5
图 6-2 VEDU 解码功能框图.....	6-6
图 6-3 JPGE 功能框图	6-8
图 6-4 JPGD 总体结构图	6-9
图 6-5 JPEG 码流结构图	6-10



6 视频编解码

6.1 VEDU

6.1.1 概述

VEDU (Video Encode and Decode Unit) 是一个硬件实现的支持 H.265/H.264 视频标准的编解码器。VEDU 具有 CPU 占用率低、总线带宽占用小、低延时、低功耗等优点。

6.1.2 特点

VEDU 编码器具有以下特点：

- 支持 ITU-T H.265 Main Profile @Level 5 Main Tier 编码
 - 支持 1/2、1/4 像素精度运动补偿
 - 支持 I、P 帧
 - 支持最多两个参考帧，支持长期参考帧
 - 帧间预测支持 64x64、32x32、16x16、8x8 四种 PU 类型
 - 帧内预测支持 32x32、16x16、8x8、4x4 四种 PU 类型
 - 支持最多候选数目为 2 的 Merge/SKIP 模式
 - 支持 32x32、16x16、8x8、4x4 四种 TU 类型
 - 支持量化矩阵
 - 支持 CABAC 熵编码
 - 支持 De-blocking 滤波
 - 支持 SAO
 - 支持 QPMap/SkipMap
- 支持 ITU-T H.264 High Profile/Main Profile/Baseline Profile@Level 5.1 编码
 - 支持 1/2、1/4 像素精度运动补偿
 - 支持 I、P 帧
 - 支持最多两个参考帧，支持长期参考帧
 - 支持帧间预测 16x16、8x8 两种子块类型
 - 支持所有 Intra4x4、Intra8x8、Intra16x16 预测模式



- 支持 Trans4x4、Trans8x8
- 支持 CABAC、CAVLC 熵编码
- 支持 De-blocking 滤波
- 支持 H.264 SVC 时域分层 (SVC-T)
- 支持如下输入图像格式
 - Semi-Planar YCbCr4:2:0
- H.265/H.264 多码流编码性能:
 - 2688x1536@30fps+720x480@30fps+360x240@30fps 编码
 - 2688x1944@20fps+720x480@20fps+360x240@20fps 编码
- 支持图像分辨率可配置
 - 最小图像分辨率: 114x114 (H.265/H.264)
 - 最大图像分辨率: 2688x1944
 - 图像宽度/高度的配置步长为 2
- 支持感兴趣区域编码
 - 支持最多 8 个区域的感兴趣区域编码
 - 每个感兴趣区域编码功能可独立使能/禁止
- 支持 OSD 区域编码保护
OSD 区域编码保护功能可使能/禁止
- 支持视频前端 OSD 叠加处理
 - 支持最多 8 个区域的编码前 OSD 叠加
 - 支持任意大小, 任意位置 (不超出图像大小和位置) OSD 叠加
 - 支持 129 级的 alpha 叠加
 - OSD 叠加功能可使能/禁止
 - 支持 OSD ARGB1555、ARGB4444 格式、2BPP 格式
- 支持彩转灰编码功能
- 支持输入低延时功能
- 支持输入 Y 分量, 自动转为 YCbCr4:2:0 灰度图 (色度全为 0x80) 并编码
- 支持 CBR/VBR/AVBR/FIXQP/ QPMAP 五种码率控制模式
- 输出码率 H.265 最高 50Mbps, H.264 最高 50Mbps。
- 支持 slice 输出中断

VEDU 解码具有以下特点:

- 支持 ITU-T H.265Main Profile @Level 5 Main Tier 解码
 - 支持 1/2、1/4 像素精度运动补偿
 - 支持 I、P 帧
 - 支持最多两个参考帧, 支持长期参考帧
 - 帧间预测支持 64x64、32x32、16x16、8x8 四种 PU 类型
 - 帧内预测支持 32x32、16x16、8x8、4x4 四种 PU 类型



- 支持最多候选数目为 2 的 Merge/SKIP 模式
- 支持 32x32、16x16、8x8、4x4 四种 TU 类型
- 支持量化矩阵
- 支持 CABAC 熵解码
- 支持 De-blocking 滤波
- 支持 SAO
- 不支持 B slice 解码
- 支持 ITU-T H.264 High Profile/Main Profile/Baseline Profile@Leve 5.1 解码
 - 支持 1/2、1/4 像素精度运动补偿
 - 支持支持帧间预测 16x16、8x8 两种子块类型
 - 支持所有 Intra4x4、Intra8x8、Intra16x16 预测模式
 - 支持 Trans4x4、Trans8x8
 - 支持 CABAC、CAVLC 熵解码
 - 支持 De-blocking 滤波
 - 支持 IPCM 解码
 - 不支持 ASO/FMO
 - 不支持 B slice 解码
 - 不支持场解码
 - 不支持 MBAFF
 - 不支持加权预测
- 支持 H.265/H.264 多码流解码性能
 - 支持 2688x1536@30fps 解码
 - 支持 2688x1944@20fps 解码
- 支持图像分辨率
 - 最小图像分辨率：114x114
 - 最大图像分辨率：2688x1944



说明

2BPP 格式说明如下：

bit[1]表示 Alpha-blending，与 ARGB1555 中 1bit 类似

bit[0]用于映射 RGB 值（具体值由寄存器指定）



注意

VEDU 的区域编码保护功能只对 VEDU 叠加的 OSD 起作用。



6.1.3 功能描述

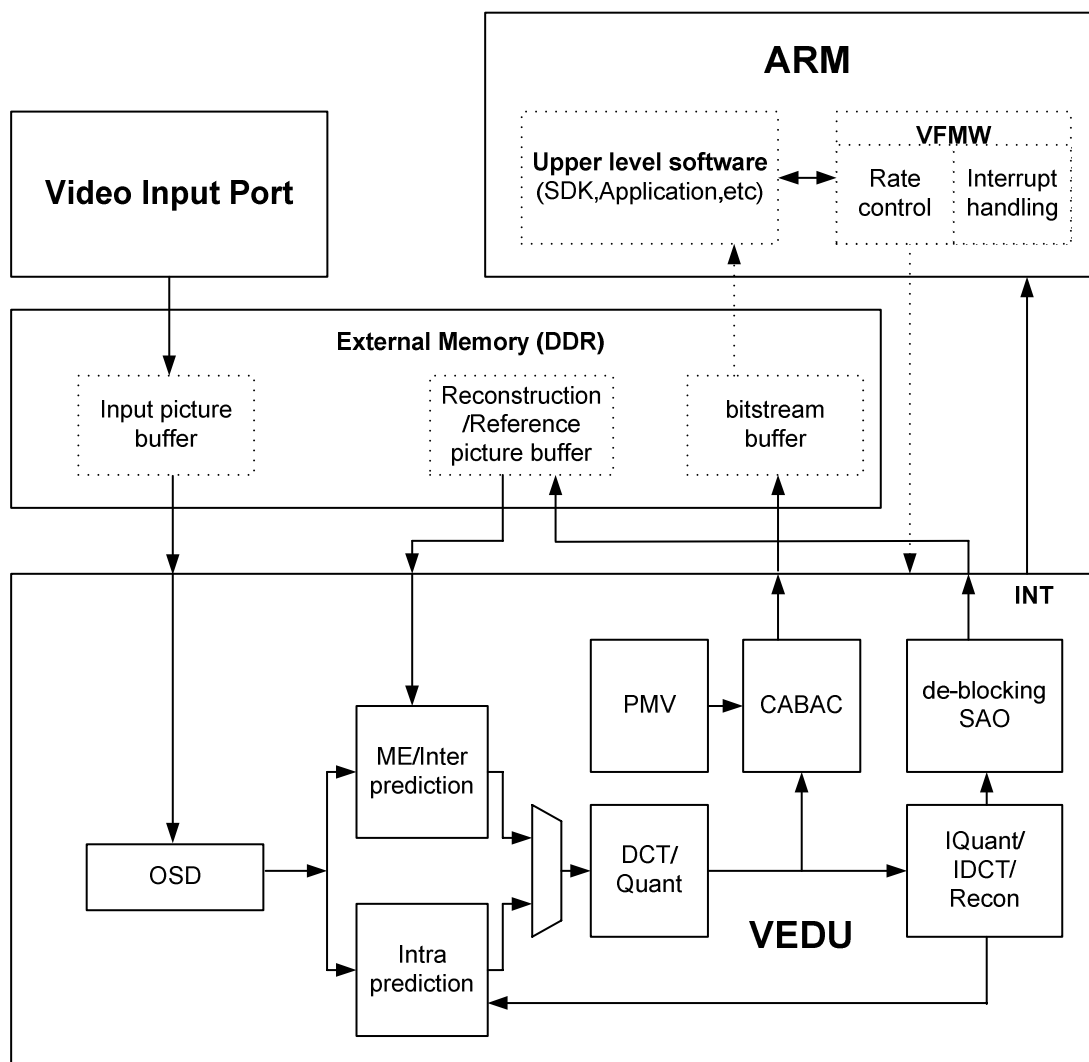
VEDU 功能框图如图 6-1 所示。

VEDU 编码实现了运动估计/帧间预测、帧内预测、运动矢量预测、变换/量化、反量化/反变换、熵编码及码流生成、de-blocking 滤波、SAO 等协议/算法处理（H.265），ARM 软件则完成码率控制和中断处理等编码控制处理。

在启动 VEDU 进行视频编码前，软件需要为其在外部存储器（DDR SDRAM）中分配以下三种类型的缓冲区。

- 输入图像缓冲区
VEDU 在编码过程中会从该缓冲区读取待编码的原始图像。该缓冲区通常由视频输入单元写入。
- 重构图像/参考图像缓冲区
VEDU 在编码过程中会向该缓冲区中写入重构图像、以作为后续图像的参考图像，在进行 P 帧编码时会从该缓冲区读取参考图像。
- 码流缓冲区
该缓冲区用于存放编码输出的码流。VEDU 在编码过程中会将码流写入该缓冲区。该缓冲区通常由软件读取。

图6-1 VEDU 编码功能框图



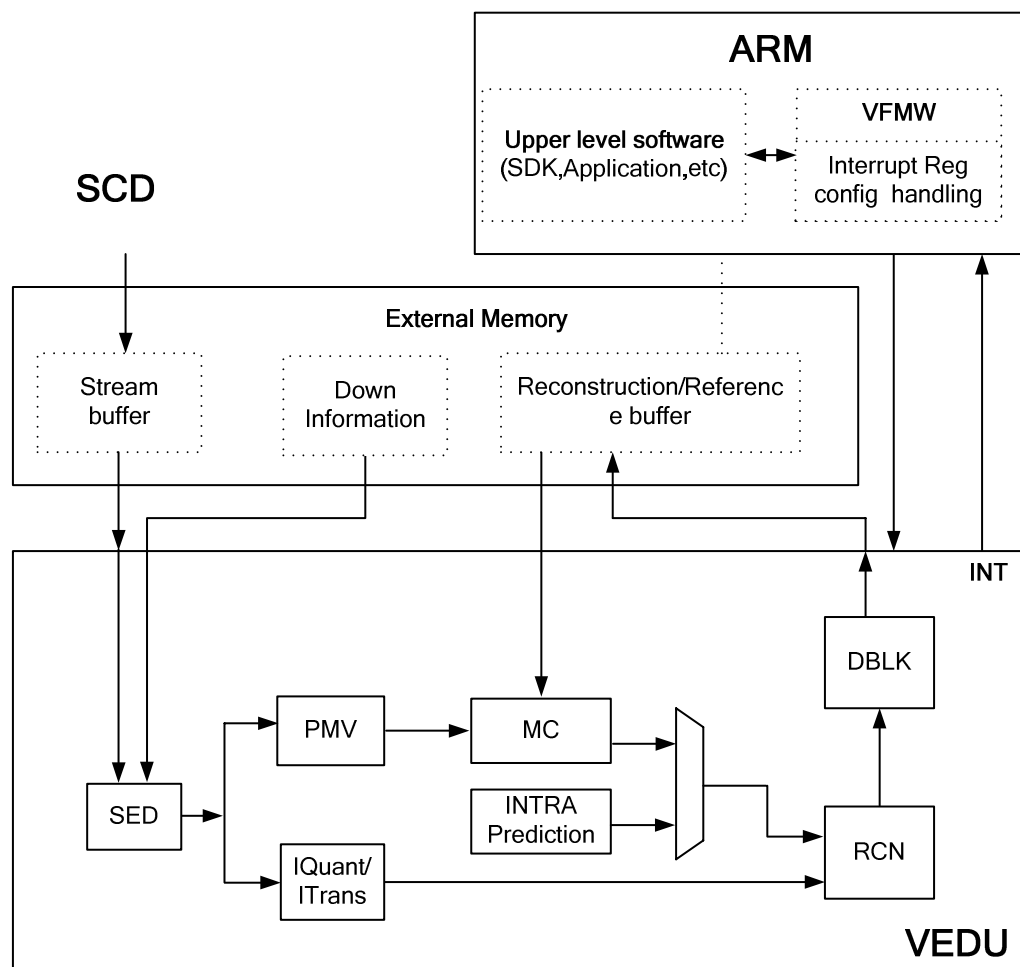
VEDU 解码功能框图如图 6-2 所示。

VEDU 解码器实现了对 H264/H265 协议按一批 slice 进行交互完成解码，VFMW 完成 slice header 及以上的解码，VEDU 硬件完成 slice data 及以下的解码。

在启动 VEDU 进行视频解码前，软件需要为其在外部存储器（DDR SDRAM）中分配以下三种类型的缓冲区。

- 重构图像/参考图像缓冲区
VEDU 在解码过程中会向该缓冲区中写入重构图像、以作为后续图像的参考图像，在进行 P 帧解码时会从该缓冲区读取参考图像。
- 码流缓冲区
该缓冲区用于存放解码输入的码流。该片空间码流来之 SCD（起始码检测模块）模块的输出
- 下行消息缓冲区
该缓冲区用于存放解码输入的码流以及 slice data 层次以上信息。

图6-2 VEDU 解码功能框图



6.2 JPGE/JPGD

6.2.1 概述

JPGE 是一个硬件实现的高性能 Jpeg 编码器，可实现图片抓拍、高清图像 MJPEG 编码业务。

JPGD (JPEG Decoder) 模块是 JPEG (Joint Photographic Experts Group) 静态数字图像解码模块，该模块在高清芯片中的作用是支持 JPEG/Motion-JPEG 图像的解码。

6.2.2 特点

JPGE 具有以下特点：

- 支持 ISO/IEC 10918-1(CCITT T.81) Baseline Process(DCT Sequential)编码
 - 支持 Semi-PlanarYCbCr4:2:0、Semi-PlanarYCbCr4:2:2 两种色度采样格式的图像编码



- 量化表可配置：Y、Cb、Cr 三个分量各提供一张量化表
- 支持彩转灰功能
- 支持输入低延时功能
- 支持输入 Y 分量，自动转为 YCbCr4:2:0 灰度图（色度全为 0x80）并编码
- 支持如下输入图像格式
Semi-Planar YCbCr4:2:0
Semi-Planar YCbCr4:2:2
- 编码性能
最高支持 16M(4608x3456)@10fps
- 支持的图像分辨率
最小图像分辨率：32x32
最大图像分辨率：8192x8192
图像宽度/高度的配置步长为 2
- 支持视频前端 OSD 叠加处理
支持最多 8 个区域的编码前 OSD 叠加
支持任意大小，任意位置（不超出图像大小和位置）OSD 叠加
支持 129 级的 alpha 叠加
OSD 叠加功能可使能/禁止
支持 OSD ARGB1555、ARGB4444 格式、2BPP 格式
- MJPEG 输出码率范围：2kbps~200Mbps

JPGD 模块具有以下功能特点：

- 部分支持 ITU-T81 Baseline profile 解码。即：
 - 支持 YUV 三分量的 JPEG 图像解码，支持 YUV 4:0:0、YUV4:2:0、YUV4:2:2 1x2、YUV4:2:2 2x1、YUV 4:4:4 五种输入格式。
 - 最多支持 4 张 Huffman 表，其中包括 2 张 DC 表和 2 张 AC 表。
 - 最多支持 3 张量化表。
 - 支持 sequential 格式解码。
 - 支持基于 DCT 变换的 JPEG 格式解码。
 - 支持 8bit 采样精度。
 - 支持交织的扫描方式。
- 输入输出最大支持分辨率为 8192x8192 大小的静态图像解码，最小支持分辨率为 8 x 8 大小的静态图像解码。
- 支持 semi-planar420 的输出。
- 提供 1 路 Motion-JPEG 16M(4608 x 3456)@10fps 的解码能力。



说明

2BPP 格式说明如下：

bit[1]表示 Alpha-blending，与 ARGB1555 中 1bit 类似

bit[0]用于映射 RGB 值（具体值由寄存器指定）



6.2.3 功能描述

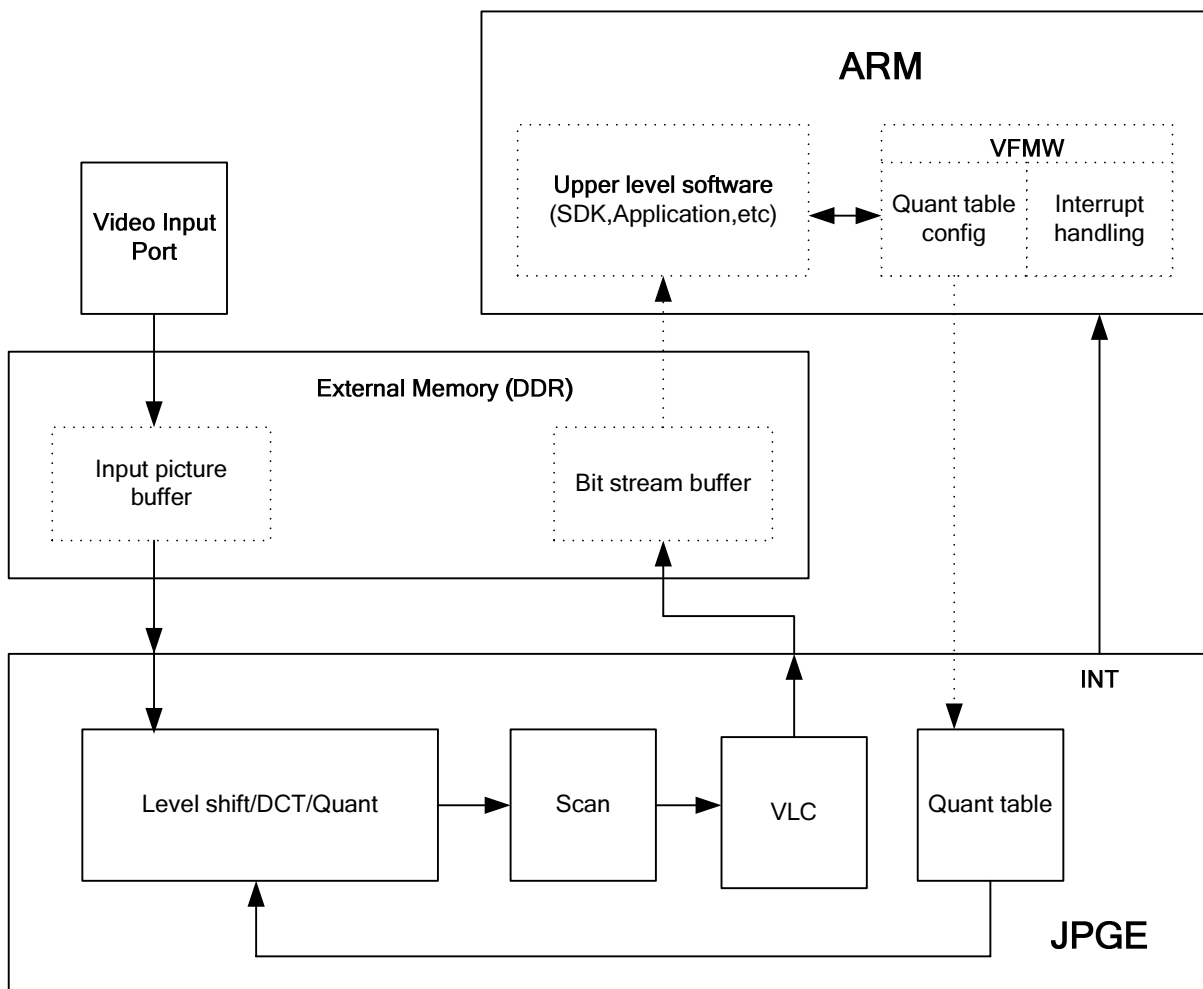
JPGE 功能如图 6-3 所示。

由图可见，JPGE 硬件实现了 OSD、level shift、DCT、量化、扫描、VLC 编码及码流生成等运算量较大的协议处理，而 ARM 软件则完成量化表配置和中断处理等编码控制处理。

在启动 JPGE 进行视频编码前，软件需要为其在外部存储器（DDR SDRAM）中分配以下两种类型的缓冲区：

- 输入图像缓冲区
JPGE 在编码过程中会从该缓冲区读取待编码的原始图像。该缓冲区通常由视频输入单元写入。
- 码流缓冲区
该缓冲区用于存放编码输出的码流。JPGE 在编码过程中会将码流写入该缓冲区。该缓冲区通常由软件读取。

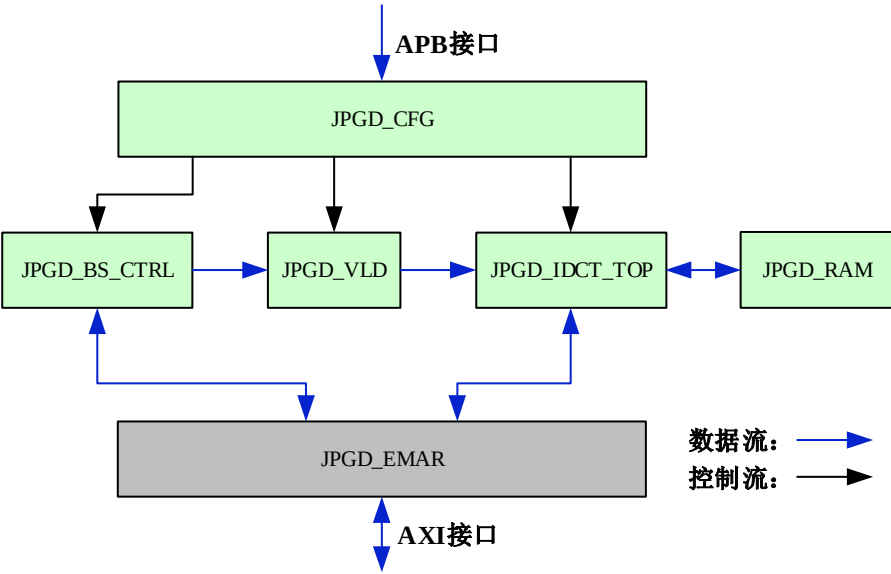
图6-3 JPGE 功能框图



JPGD 总体结构如图 6-4 所示。



图6-4 JPGD 总体结构图



JPGD 内部模块说明如表 6-1 所示。

表6-1 JPGD 内部模块说明

模块名称	功能
JPGD_BS_CTRL	码流的读取和移位处理，内含一个 Barrel-Shift，将有效码流送给下游模块进行解码。
JPGD_VLD	Huffman 变长码解码，同时将解码后的系数进行反扫描和反量化。
JPGD_IDCT_TOP	进行 IDCT 变换。
JPGD_CFG	接收 HOST 的配置信息，并将配置信息配置给各功能模块。同时负责整个解码器的启动、中断的产生以及向 HOST 反馈解码器的内部状态。
JPGD_EMAR	与 AXI 总线相关的接口 IP，完成总线异步处理，数据存取等。

6.2.4 工作方式

6.2.4.1 软硬件划分

JPEG 码流结构如图 6-5 所示。



图6-5 JPEG 码流结构图

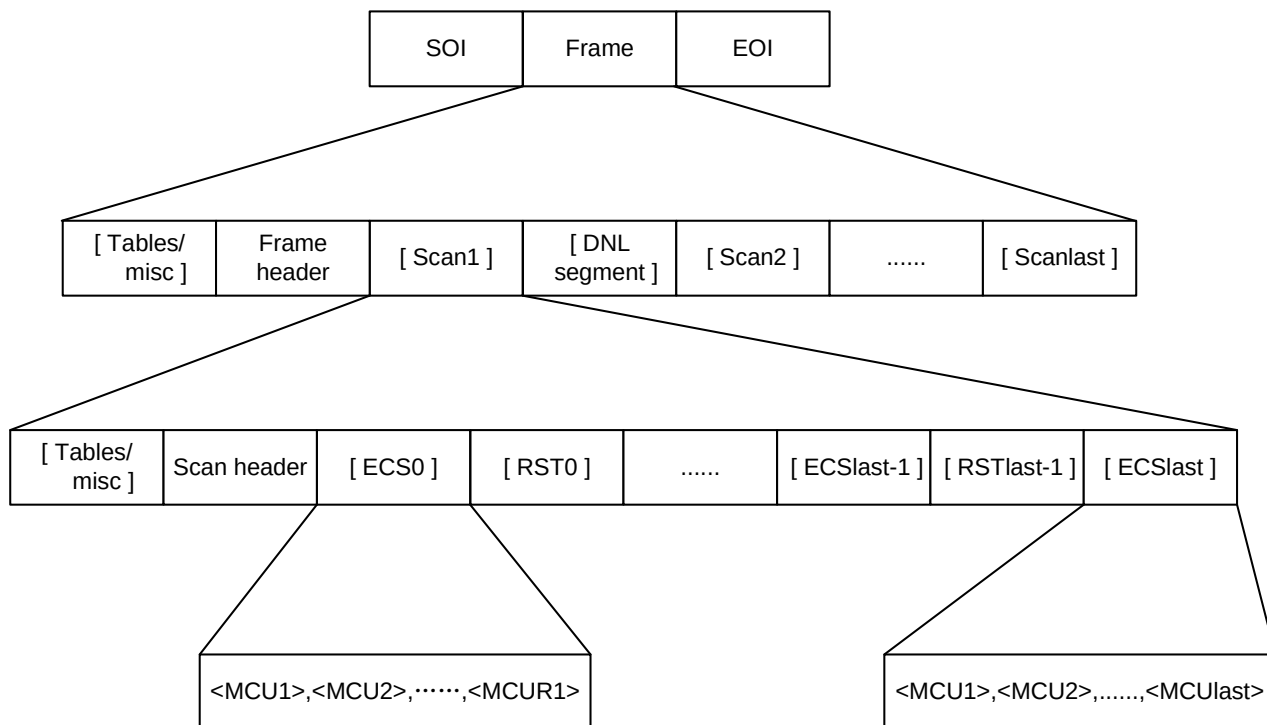


图 6-5 为广义结构图，对于 JPEG 码流，由软件解析 Scan header 及其以上部分，硬件解析 ECS 层和 RSTn 标志。

6.2.4.2 软硬件交互

JPEG 解码由软硬件共同完成，所以在解码中存在软硬件交互。

- 软硬件的交互以帧级进行：
 - 对于 JPEG 图像，每帧图像交互一次。
 - 对于 Motion-JPEG，每帧图像交互一次。
- 软硬件可以通过查询和中断两种方式进行交互，中断产生方式有如下几种：
 - 当前图像解码完成产生的中断，表示当前图像已经全部解码完成并写入 DDR 中，JPEG 解码工作结束（因为 Baseline 图像只有一个扫描层，因此一个扫描层解码结束也表示一幅图像解码结束）。
 - 当前图像解码错误中断，表示当前图像解码过程中发生了错误，JPGD 无法继续解码，工作结束。



目 录

目 录.....	i
7 视频及图形处理.....	7-1
7.1 TDE.....	7-1
7.1.1 概述	7-1
7.1.2 功能描述	7-1
7.2 VPSS.....	7-2
7.2.1 概述	7-2
7.2.2 功能描述	7-2
7.3 VGS.....	7-3
7.3.1 概述	7-3
7.4 GDC	7-3
7.4.1 概述	7-3



7

视频及图形处理

7.1 TDE

7.1.1 概述

2D 图形加速引擎 TDE（Two Dimensional Engine）利用硬件进行图形绘制，可以大大减少对 CPU 的占用，同时提高了内存带宽的资源利用率。

图形数据接口包括两条通路，其功能如下：

- 通路 1 在单源操作时完成直接拷贝与直接填充的功能。
- 通路 2 在单源操作时可完成各种复杂的操作，如图像缩放等。
- 通路 1 与通路 2 协同工作时可以完成颜色混合等操作。

7.1.2 功能描述

TDE 模块有以下功能特点：

- 源位图 1 支持输入格式：ARGB4444、ARGB1555、ARGB8888、RGB444、RGB565、RGB888、ARGB8565、CLUT1、CLUT2、CLUT4、CLUT8、A1、A8、byte、halfword。
- 源位图 2 支持输入格式：ARGB4444、ARGB1555、ARGB8888、RGB444、RGB565、RGB888、ARGB8565、CLUT1、CLUT2、CLUT4、CLUT8、A1、A8、semi-planner 420。
- 输出图像支持 ARGB4444、ARGB1555、ARGB8888、RGB444、RGB565、RGB888、ARGB8565、CLUT1、CLUT4、CLUT8、A1、A8、byte、halfword、packge422（YUYV、VYUY）的格式。
- 像素大于等于 1byte 的只支持小端系统，像素小于 1byte 的大小端都支持。
- 支持源位图 1、源位图 2 和输出位图格式分别可配。
- 支持 Gamma 校正、亮度对比度的调节。
- 支持直接拷贝。
- 支持直接填充。
- 支持 2D-resize 操作。
- 支持 clip 操作。



- 支持 alpha blending 操作。
- 支持 colorkey 操作。
- 支持 clip mask 功能。
- 支持旋转功能(90°、180°、270°)。
- 支持 mirror、flip。
- 最大分辨率支持到 1920x1080。

7.2 VPSS

7.2.1 概述

视频处理子系统 VPSS（Video Processing Sub System）实现视频处理功能。支持在线（VICAP-VIPROC-VPSS 全在线）和离线（VPSS 离线或 VIPROC-VPSS 之间在线）两种工作模式。包含视频遮挡、3D 降噪、视频马赛克处理、视频裁剪、缩放、亮度单分量处理、压缩、解压缩、mirror、flip、LBA（Letter Box）幅型比转换功能。

VPSS 特性如下：

- 支持最大宽度 2688 的视频源。
- 支持最多 3 个通道视频输出。
- 支持 3D 降噪功能。
- 输出支持 4 个区域的马赛克处理。
- 输出支持 8 个区域的视频任意四边形实心遮挡，其中凹四边形处理成三角形。
- 支持 3 个输出通道的视频裁剪和 mirror/flip 功能。
- 支持视频数据压缩/解压缩。
- 输入数据为 Semiplanar 420/422/单分量。
- 输入数据支持 linear 格式和 tile 格式。
- 输出数据为 Semiplanar 420/422/单分量。
- 输出数据支持 linear 格式。
- 支持亮度单分量处理。
- 支持输出通道低延时模式。
- 支持安全特性。
 - 支持安全 CPU 操作（寄存器隔离）。
 - 支持安全的 DDR 读写操作。

7.2.2 功能描述

- 缩放功能：输入输出分辨率不同的低频滤波处理。
 - 离线模式下，水平、垂直最大缩放倍数各支持：CH0 支持 1~16 倍放大，CH1~CH2 支持 1~1/15 倍缩小。
 - 在线模式下，CH0 仅支持 1:1 缩放，CH1~CH2 支持 1~1/15 倍缩小。



- **LBA 幅型比转换：**通过在图像上下边缘或者左右边缘添加固定背景色的边框，从而调整图像幅型比，背景色可配。仅离线模式支持。

7.3 VGS

7.3.1 概述

视频图形系统 VGS（Video Graphics System）实现视频及图形处理功能。包含 OSD 叠加、缩放、区域亮度和统计、旋转、多区域拼接、视频裁剪、视频遮挡功能。

VGS 特性如下：

- 支持离线处理最大输入宽度 2688 的视频源。
- 支持一个通道视频输出。
- 支持 1 个区域的 OSD 与视频叠加，支持 OSD 反色。
- OSD 的输入格式为 ARGB1555/ ARGB4444/ ARGB8888
- 输入/输出数据支持 linear 格式。
- 输入/输出支持视频数据压缩解压缩。
- 支持 1 区域的视频遮挡，遮挡形状可支持实心或空心的任意四边形，凹四边形处理成三角形。
- 支持视频裁剪。
- 支持区域亮度和统计。
- 支持单分量处理。
- 支持多区域拼接。
- 支持 90 度、180 度、270 度旋转。
- 支持缩放：输入输出分辨率不同的低频滤波处理。缩放倍数支持 1/30~16 倍。
- 支持特征点检测和全局运动估计。
- 支持安全 CPU。
- 支持安全 DDR 属性可配置。

7.4 GDC

7.4.1 概述

几何畸变矫正 GDC（Geometric Distortion Correction）实现图像畸变矫正功能。

GDC 特性如下：

- 支持离线处理最大输入宽度 2688 的视频源。
- 支持 YUV semi planer420 格式输入输出图像。
- 支持单分量图像处理。
- 支持任意角度旋转。



- 支持桶形和枕形畸变矫正。
 - 支持最大 40%的桶形镜头畸变矫正
 - 支持最大 20%的枕形镜头畸变矫正
 - 支持桶形展宽功能
- 支持防抖矫正。
 - 支持 6 个自由度，其中 4 个自由度校准参数可配
 - 支持平行四边形及拉伸形变的畸变矫正
 - 支持陀螺仪（Gyro）、数字、混合（Gyro+数字）模式
- 输入/输出数据支持 linear 格式。
- 支持非压缩图像输入，支持压缩图像数据输出。
- 支持安全 CPU。
- 支持安全 DDR 属性可配置。



目 录

8 智能加速引擎.....	8-1
8.1 IVE.....	8-1
8.1.1 概述	8-1
8.1.2 功能描述	8-2
8.1.3 工作方式	8-3
8.2 NNIE 加速引擎	8-17
8.2.1 概述	8-17
8.2.2 特点	8-18



插图目录

图 8-1 IVE 在系统中的位置	8-1
图 8-2 SP422 存储格式	8-3
图 8-3 SP420 存储格式	8-3
图 8-4 8bit 单分量数据在 Memory 中的存储	8-4
图 8-5 Package 存储格式	8-4
图 8-6 Plannar 存储格式	8-5
图 8-7 16bit 单分量数据在 Memory 中的存储	8-5
图 8-8 32bit 单分量数据在 Memory 中的存储	8-6
图 8-9 64bit 单分量数据在 Memory 中的存储	8-6
图 8-10 NCC 输出数据在 Memory 中的存储	8-6
图 8-11 CCL 统计信息在 Memory 中的存储（顺序存放）	8-6
图 8-12 16bit 水平和垂直方向的间插数据在 Memory 中的存储	8-7
图 8-13 HOG 输出格式 1	8-7
图 8-14 HOG 输出格式 2	8-8
图 8-15 NNIE 加速引擎在系统中的位置	8-18



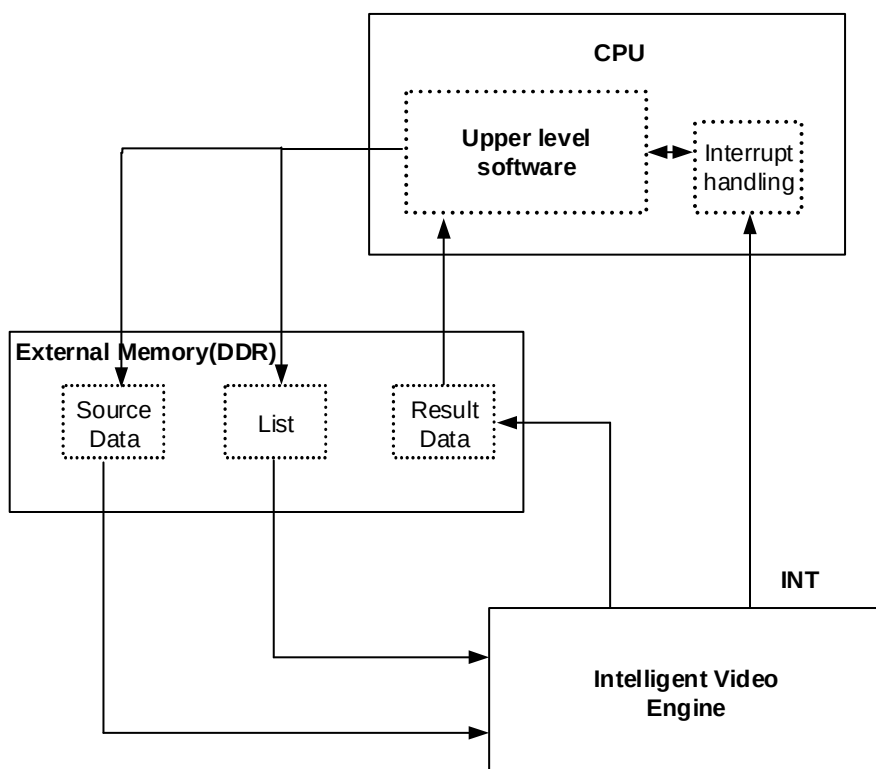
8 智能加速引擎

8.1 IVE

8.1.1 概述

IVE (Intelligent Video Engine)模块提供智能分析算法中所用到的一系列基础运算功能，以及部分耗时较大的特殊功能，是智能分析系统中的硬件加速模块，IVE 在系统中的位置如图 8-1 所示。

图8-1 IVE 在系统中的位置





8.1.2 功能描述

IVE 模块支持如下功能特点：

- DMA：支持直接拷贝、间隔拷贝、内存填充。
- Filter：支持 5x5 模板滤波。
- CSC：支持 YUV2RGB、YUV2HSV、YUV2LAB、RGB2YUV 颜色空间转换。
- FilterAndCSC：支持 5x5 模板滤波和 CSC 的复合功能。
- Sobel：支持 5x5 模板 Sobel-like 梯度计算。
- MagAndAng\Canny：支持 5x5 模板梯度幅值和幅角计算、Canny 边缘提取。
- Erode：支持 5x5 模板腐蚀。
- Dilate：支持 5x5 模板膨胀。
- Thresh\Thresh_S16\Thresh_U16：支持图象阈值化处理。
- And\Or\Xor：支持两幅图象相与、或、异或。
- Add\Sub：支持两幅图象相加加权加、减。
- Integ：支持积分图计算。
- Hist：支持直方图统计。
- Map：支持对图像通过 256 级 map 映射赋值。
- 16BitTo8Bit：支持 16bit 数据到 8bit 数据线性转换。
- OrdStatFilter：支持顺序统计量滤波：中值滤波、最大值滤波、最小值滤波。
- NCC：支持两相同大小图像互相关系数计算。
- CCL：支持连通区域标记。
- LBP：支持简单局部二值模式计算。
- NormGrad：支持归一化梯度计算。
- LKOpticalFlow：支持 LK 光流跟踪。
- STCorner：支持 ShiTomasi 角点检测。
- SAD：支持分块计算两幅图像对应像素差值绝对值的累加和。
- Resize：支持双线性缩放。
- GMM2：支持灰度图、RGB 图的快速混合高斯背景建模。
- PSP：支持 nonreflective similiarity、similiarity、affine 三种变换。
- HOG：支持方向梯度直方图特征提取。
- KCF：支持对 HOG 特征进行核相关滤波。
- 支持单独进行软复位。
- 支持 128bit AXI 总线和 32 bit APB 总线。
- 支持链表级中断、节点级中断和超时中断。
- 支持 SP400、SP420 (semi-plannar 420)、SP422 (semi-plannar 422)、package、planar 等输入格式。
- 支持 SP400、SP420、SP422、package、plannar 等输出格式。
- 部分算子支持读写地址非 16-byte 对齐。



8.1.3 工作方式

8.1.3.1 输入、输出数据格式

图 8-2~图 8-14 中的 w、h 均指的是图像以像素为单位的宽、高。如无特别说明，数据存放顺序均是在小端系统（little endian）的内存存放顺序，为了方便描述，统一使用 word 作为存储单位进行描述，实际应用中不同的算子对数据存储对齐格式有特殊要求。8.1.3.2 中所描述的不同算子功能，对应的输入输出格式也不相同。

图8-2 SP422 存储格式

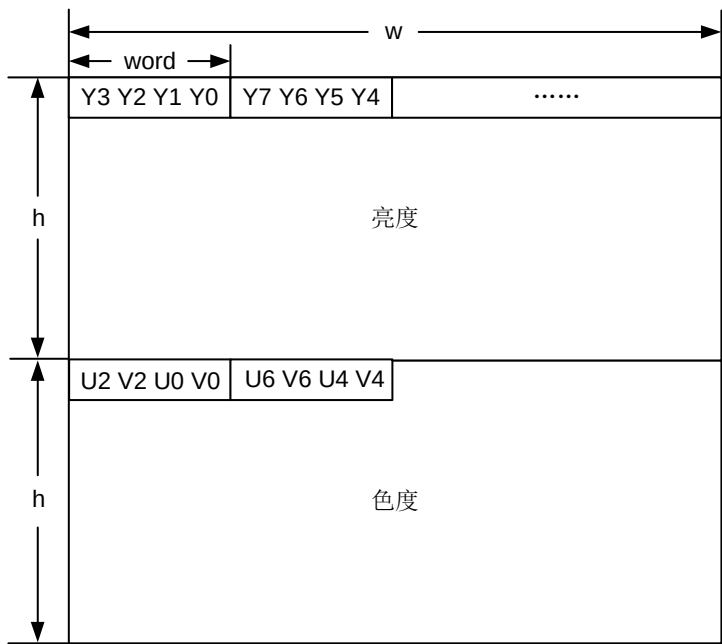


图8-3 SP420 存储格式

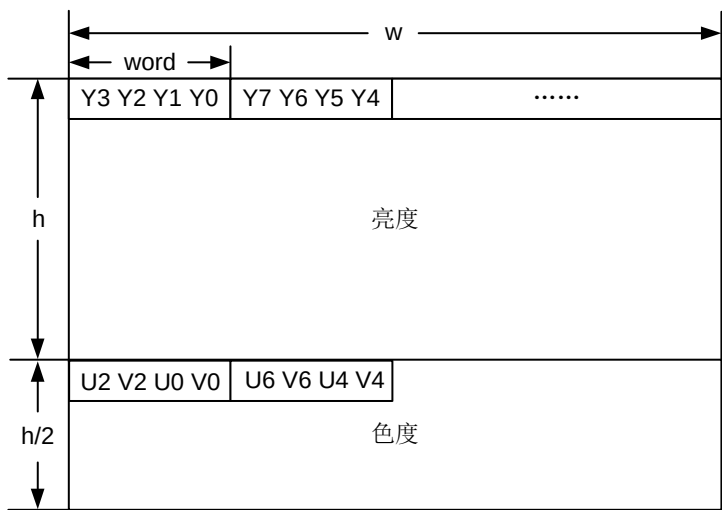




图8-4 8bit 单分量数据在 Memory 中的存储

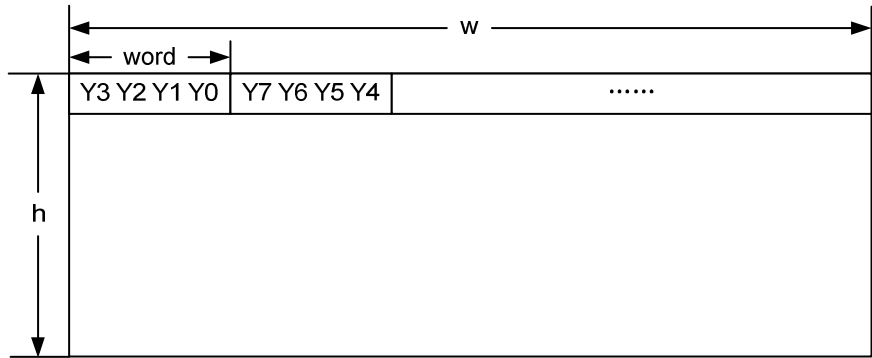


图8-5 Package 存储格式

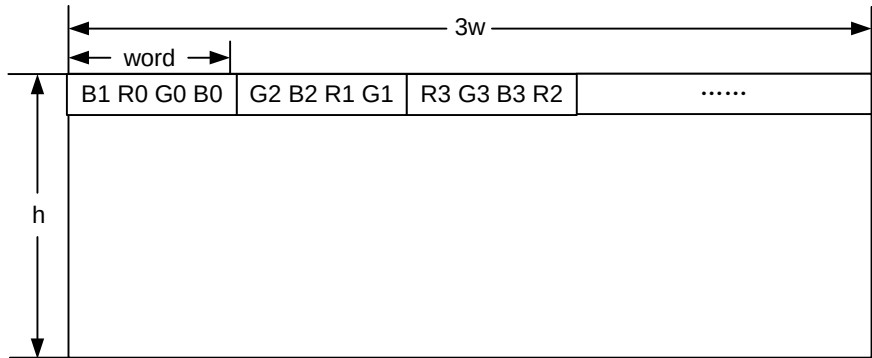




图8-6 Plannar 存储格式

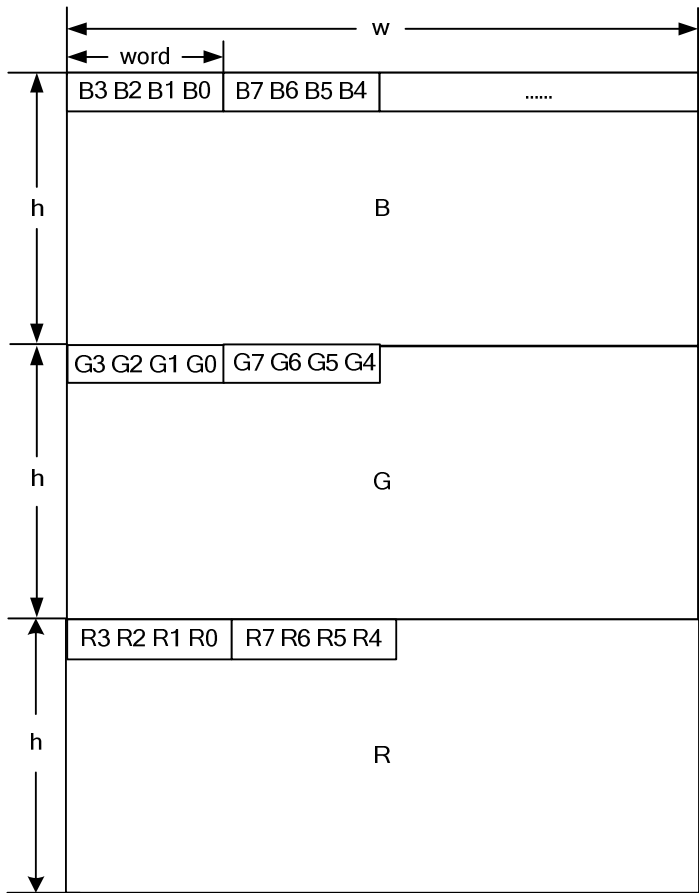


图8-7 16bit 单分量数据在 Memory 中的存储

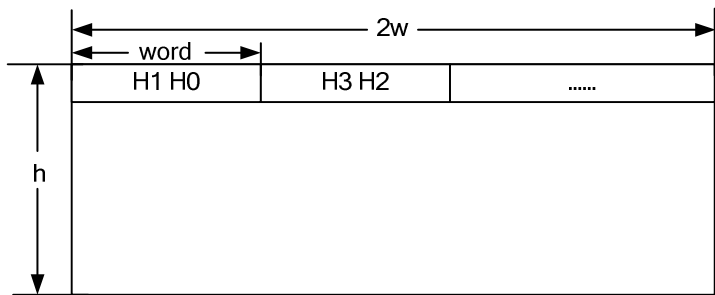




图8-8 32bit 单分量数据在 Memory 中的存储

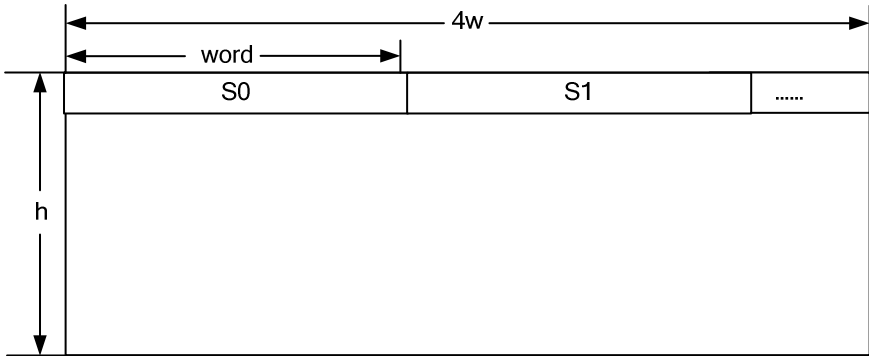


图8-9 64bit 单分量数据在 Memory 中的存储

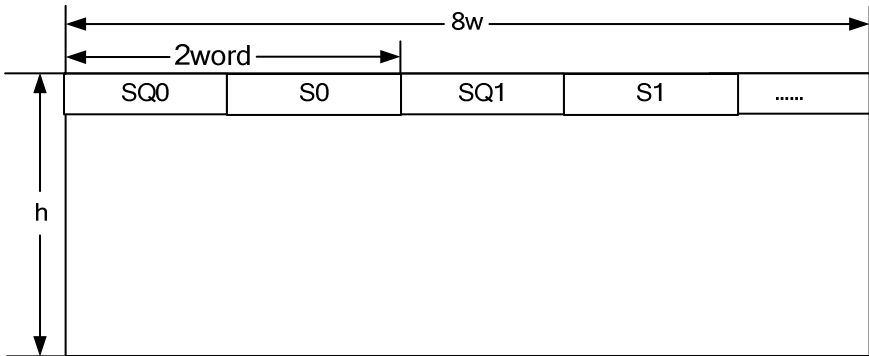


图8-10 NCC 输出数据在 Memory 中的存储

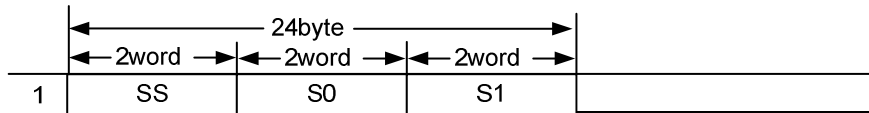


图8-11 CCL 统计信息在 Memory 中的存储（顺序存放）

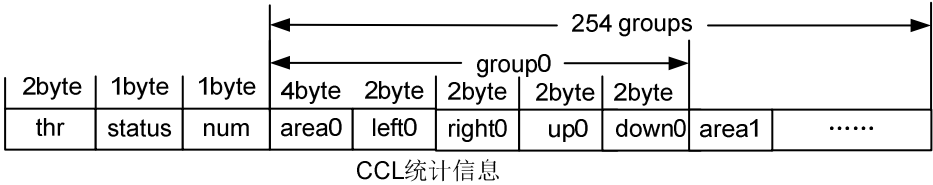




图8-12 16bit 水平和垂直方向的间插数据在 Memory 中的存储

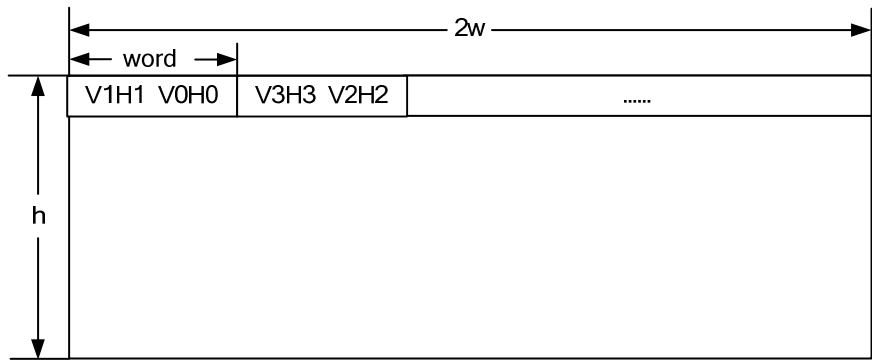


图8-13 HOG 输出格式 1

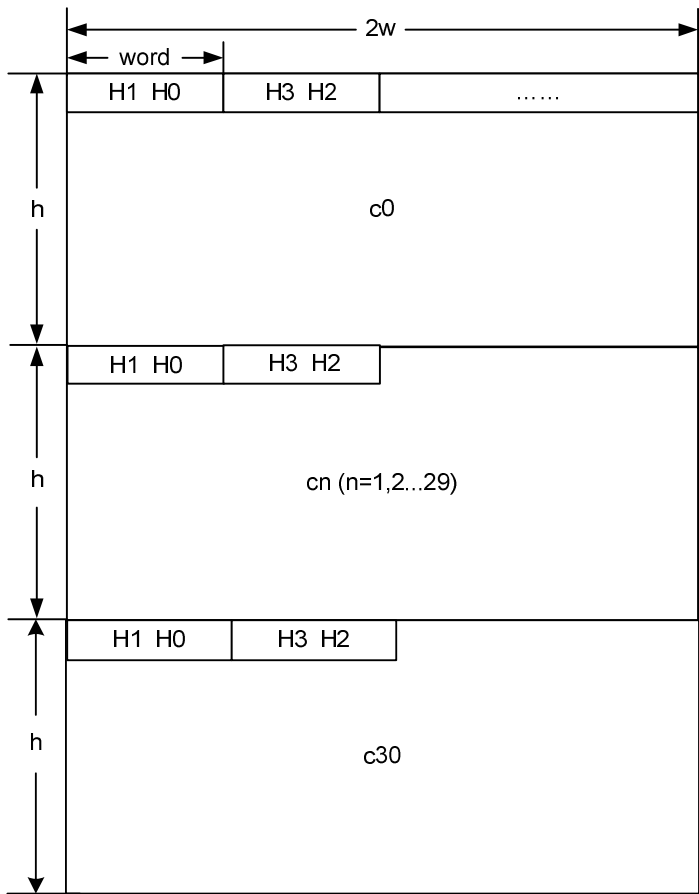
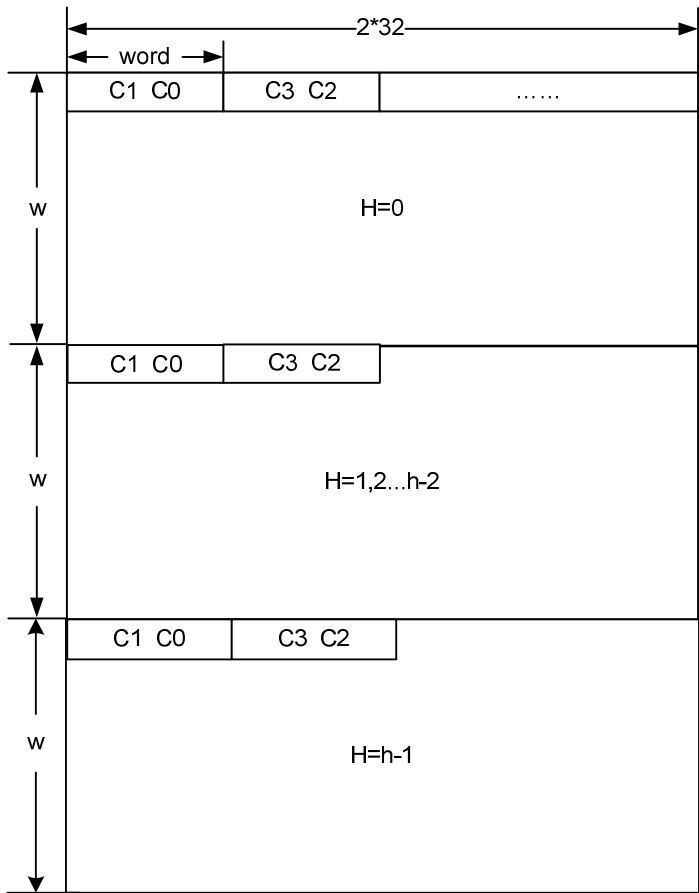




图8-14 HOG 输出格式 2



8.1.3.2 支持的功能



说明

本章节描述的各算子的支持分辨率请参考《HiIVE API 参考》。

DMA

(1) 直接拷贝模式

实现矩形数据区域的快速搬移功能。源数据在此模式下将会直接通过 IVE 内部快速通路，搬移到目的区域，并直接覆盖目标区域数据。

- 地址对齐方式：输入输出地址都要求 byte 对齐，输入输出 stride 要求 16byte 对齐
- 输入输出格式：输入和输出都是 8bit 单分量数据，如图 8-4 所示

(2) 间隔拷贝模式

实现矩形区域的数据间隔搬移的功能。源数据在该模式下将在水平和垂直方向上都间隔给定的距离搬运固定大小数据到目的区域。

- 地址对齐方式：输入输出地址都要求 byte 对齐，输入输出 stride 要求 16byte 对齐
- 输入输出格式：输入和输出都是 8bit 单分量数据，如图 8-4 所示



- 其他：源数据宽 width 必须为 distance 的整数倍

(3) 3 字节 memset 模式

实现矩形数据区域的 memory set 功能，以 3 字节为单位填充目的区域。

- 地址对齐方式：输入输出地址都要求 byte 对齐，输入输出 stride 要求 16byte 对齐
- 输入输出格式：无输入数据，输出 8bit 单分量数据，如图 8-4 所示

(4) 8 字节 memset 模式

实现矩形数据区域的 memory set 功能，以 8 字节为单位填充目的区域。

- 地址对齐方式：输入输出地址都要求 byte 对齐，输入输出 stride 要求 16byte 对齐
- 输入输出格式：无输入数据，输出 8bit 单分量数据，如图 8-4 所示

Filter

将源图像以 5x5 模板作滤波后输出。

- 地址对齐方式：输入输出地址与 stride 都要求 16-byte 对齐，输出各分量配置相同 stride
- 输入输出格式有 3 种不同情况：
 - 输入和输出都是 8bit 单分量数据，如图 8-4 所示
 - 输入和输出都是 SP420 数据，如图 8-3 所示
 - 输入和输出都是 SP422 数据，如图 8-2 所示

CSC

颜色空间转换，支持 YUV2RGB, YUV2HSV, YUV2LAB, RGB2YUV 空间的转换。

- 地址对齐方式：输入输出地址与 stride 都要求 16-byte 对齐，输出各分量配置相同 stride
- 输入输出格式：
 - SP420 --> package; SP420 --> plannar
 - SP422 --> package; SP422 --> plannar
 - package-->SP420; package-->SP422
 - plannar-->SP420; plannar-->SP422SP420 或 SP422 格式时，如图 8-3 和图 8-2 所示。
package 格式时，如图 8-5 所示。
plannar 格式时，如图 8-6 所示。

FilterAndCSC

将 YUV SP420/SP422 图象以 5x5 模板作滤波，然后再作 YUV2RGB 的颜色空间转换后输出。

- 地址对齐方式：输入输出地址与 stride 都要求 16-byte 对齐，输出各分量配置相同 stride



- 输入输出格式：
 - SP420 --> package; SP420 --> plannar
 - SP422 --> package; SP422 --> plannar

SP420 或 SP422 格式时，如图 8-2 和图 8-3 所示；package 格式时，如图 8-5 所示；plannar 格式时，如图 8-6 所示

Sobel

对垂直和水平方向做 5x5 模板的 sobel 滤波。

- 地址对齐方式：输入输出地址与 stride 都要求 16-byte 对齐，输出各分量配置相同 stride
- 输入输出格式：
 - 输入为 8bit 单分量图，如图 8-4 所示
 - 只输出 H 或 V 时，如图 8-7 所示
 - H 和 V 都输出时，如图 8-7 所示

MagAndAng

计算图像梯度的幅值、幅角，支持对幅值图 TO_ZERO thresh 操作。

- 地址对齐方式：输入输出地址与 stride 都要求 16-byte 对齐，输出各分量配置相同 stride
- 输入输出格式：
 - 输入为 8bit 单分量图，如图 8-5 所示
 - 输出为 16bit 单分量幅值图，如图 8-7 所示
 - 输出为 8bit 单分量幅角图，如图 8-4 所示

Dilate

二值图像 5x5 模板膨胀。

- 地址对齐方式：输入输出地址都要求 16-byte 对齐
- 输入输出格式：输入和输出都是 8bit 单分量数据，如图 8-4 所示

Erode

二值图像 5x5 模板腐蚀。

- 地址对齐方式：输入输出地址都要求 16-byte 对齐
- 输入输出格式：输入和输出都是 8bit 单分量数据，如图 8-4 所示

Thresh

使用固定阈值对图像进行阈值化操作，共有 8 种可配置模式。

- 地址对齐方式：输入输出地址都要求 byte 对齐，输入输出 stride 要求 16byte 对齐
- 输入输出格式：输入和输出都为 8bit 单分量数据，如图 8-4 所示



And

源 2 数据与源 1 数据进行与运算，并将结果输出到目的区域。

- 图象分辨率：要求输入的两图像分辨率相同
- 地址对齐方式：输入输出地址都要求 byte 对齐，输入输出 stride 要求 16byte 对齐
- 输入输出格式：
 - 输入为源 1 8bit 单分量数据
 - 输入为源 2 8bit 单分量数据
 - 输出为运算后的结果数据，为 8bit 单分量数据，如图 8-4 所示

Sub

源 2 数据与源 1 数据进行减运算，并将结果输出到目的区域。

- 图象分辨率：要求输入的两图像分辨率相同
- 地址对齐方式：输入输出地址都要求 byte 对齐，输入输出 stride 要求 16byte 对齐
- 输入输出格式：
 - 被减数 8bit 单分量
 - 减数 8bit 单分量
 - 结果数据 8bit 单分量，如图 8-4 所示

Or

源图 2 数据与源图 1 数据进行进行或运算，并将结果输出到目的区域。

- 图象分辨率：要求输入的两图像分辨率相同
- 地址对齐方式：输入输出地址都要求 byte 对齐，输入输出 stride 要求 16byte 对齐
- 输入输出格式：
 - 输入两幅图都是 8bit 单分量图
 - 输出 8bit 单分量图，如图 8-4 所示

Integral

对图像求积分图、平方和积分图，支持和积分图单独输出、平方和积分图单独输出、和积分图与平方和积分图拼在一起输出（分量累加和占低 28 比特，分量平方累加和高 36 比特）。

- 地址对齐方式：输入输出地址与 stride 都要求 16byte 对齐
- 输入输出格式：
 - 输入为 8bit 单分量图，如图 8-4 所示
 - 和积分图单独输出时，输入为 32bit 单分量图，如图 8-8 所示；平方和积分图单独输出或者和积分图与平方和积分图拼在一起输出时，输出为 64bit 单分量图，如图 8-9 所示



Histogram

256 级直方图统计，输入为单分量，输出为位宽为 32bit 的 256 级直方图统计值。

- 地址对齐方式：输入输出地址与 stride 都要求 16byte 对齐
- 输入输出格式：
 - 输入为 8bit 单分量，如[图 8-4](#) 所示
 - 输出统计结果 32bit 单分量数据，如[图 8-8](#) 所示

Thresh_S16

Signed 16bit 数据到 8bit 数据阈值化，支持 4 种比较模式。

- 地址对齐方式：输入输出地址与 stride 都要求 16byte 对齐
- 输入输出格式：
 - 输入 16bit 单分量数据，如[图 8-7](#)
 - 输出转换后的 8bit 单分量数据，如[图 8-4](#)

Thresh_U16

Unsigned 16bit 数据到 unsigned 8bit 数据阈值化，支持 2 种比较模式。

- 地址对齐方式：输入输出地址与 stride 都要求 16byte 对齐
- 输入输出格式：
 - 输入 16bit 单分量数据，如[图 8-7](#) 所示
 - 输出转换后的 8bit 单分量数据，如[图 8-4](#) 所示

16BitTo8Bit

16bit 数据到 8bit 的线性转化，支持 4 种比较模式。

- 地址对齐方式：输入输出地址与 stride 都要求 16byte 对齐
- 输入输出格式：
 - 输入 16bit 数据，如[图 8-7](#) 所示
 - 输出转换后的 8bit 数据，如[图 8-4](#) 所示

OrdStatFilter

3x3 模板顺序统计量滤波，支持中值滤波、最大值、最小值。

- 地址对齐方式：输入输出地址与 stride 都要求 16byte 对齐
- 输入输出格式：输入和输出都为 8bit 单分量图，如[图 8-4](#) 所示

Map

数据映射赋值，将源数据通过用户配置的 256 级 unsigned 8bit Map、unsigned 16bit Map 或 signed 16bit Map 表赋予新的值输出。



- 地址对齐方式：所有输入输出地址和 stride 都要求 16byte 对齐。
- 输入输出数据格式：
 - 输入 8bit 单分量数据，如图 8-4 所示
 - 输入 8bit 或 16bit 位宽的映射表，表项长度固定 256 项
 - 输出映射后的 8bit 单分量或 16bit 数据。如图 8-4 和图 8-7 所示

Add

两幅灰度图像的加权和，可分别配置各自的权重。

- 图象分辨率：要求输入的两图像分辨率相同
- 地址对齐方式：输入输出地址都要求 byte 对齐，输入输出 stride 都要求 16byte 对齐
- 输入输出数据格式：
 - 输入两幅 8bit 单分量图像数据，如图 8-4 所示
 - 输出相加后的 8bit 单分量图像数据。如图 8-4 所示

Xor

两幅二值图像对应数据相异或。

- 图象分辨率：要求输入的两图像分辨率相同
- 地址对齐方式：输入输出地址都要求 byte 对齐，输入输出 stride 都要求 16byte 对齐
- 输入输出数据格式：
 - 输入两幅 8bit 单分量图像数据如图 8-4 所示
 - 输出异或后的 8bit 单分量图像数据，如图 8-4 所示

NCC

求取两相同分辨率灰度图像的归一化互相关系数。

- 图象分辨率：要求输入的两图像分辨率相同
- 地址对齐方式：输入地址都要求 byte 对齐，输出地址和输入输出 stride 都要求 16byte 对齐
- 输入输出数据格式：
 - 输入两幅 8bit 单分量图像数据，如图 8-4 所示
 - 输出共有 3 个数据，按先后顺序分别为两图像乘积累加值、图像 1 的平方和累加值、图像 2 的平方和累加值，如图 8-10 所示

CCL

二值图的 8-连通/4-连通区域标记。

- 图象分辨率：widthxheight < 1024x1024
- 地址对齐方式：所有输入输出地址与 stride 都要求 16byte 对齐



- 输入输出数据格式：
 - 输入 8bit 单分量图像，如[图 8-4](#) 所示，输出 16bit 数据表示输出的有效连通区域最小面积
 - 输出 8bit 数据表示检测到的连通区域是否超出阈值；
 - 输出各个连通区域的外接矩形坐标及面积，如[图 8-11](#) 所示，同时将原图改写为连通区域标记后的 8bit 单分量图像，如[图 8-4](#) 所示

CannyHysEdge

Canny 边缘检测的磁滞阈值化、非极大抑制部分，输出强弱边缘图及强边缘坐标信息。

- 地址对齐方式：所有输入输出地址与 stride 都要求 16byte 对齐
- 输入输出数据格式：
 - 输入 8bit 单分量幅角图，如[图 8-4](#) 所示
 - 输入 16bit 单分量幅值图，如[图 8-7](#) 所示
 - 输出 8bit 单分量边缘标记图，如[图 8-4](#) 所示
 - 输出 32bit 单分量数据表示的栈，如[图 8-8](#) 所示
 - 输出栈大小，用一个 32bit 数据表示

LBP

LBP 局部二值模式，提取图像局部的纹理信息。

- 地址对齐方式：所有输入输出地址与 stride 都要求 16byte 对齐
- 输入数据格式：
 - 输入 8bit 单分量数据
 - 输出 8bit 单分量 8bit 结果数据，如[图 8-4](#) 所示

NormGrad

归一化梯度计算，梯度分量均归一化到 8bit。

- 地址对齐方式：所有输入输出地址与 stride 都要求 16byte 对齐，要求输出各分量配置相同 stride
- 输入输出数据格式：
 - 输入 8bit 单分量数据，如[图 8-4](#) 所示
 - 当 out_fmt 为 0x00、0x01 或 0x02 时，输出为 8bit 单分量数据，如[图 8-4](#) 所示；当 out_fmt 为 0x03 时，输出为 16bit H 和 V 间插数据，如[图 8-12](#) 所示

LKOpticalFlow

LK 光流运动估计，支持多层金字塔输入，1~4 层可配，支持最多 500 个角点。

- 地址对齐方式：所有输入输出地址与 stride 都要求 16byte 对齐
- 输入输出数据格式：



- 输入前一帧 8bit 单分量 单/多层金字塔图像，每层图像如图 8-4 所示
- 输入当前帧 8bit 单分量 单/多层金字塔图像，每层图像如图 8-4 所示
- 输入前一帧角点坐标，每个坐标用两个 32bit 表示，分别表示水平坐标、垂直坐标，有效位[21:0]
- 当前帧角点坐标既作为输入（仅在使用初始光流情况下），也作为输出
- 输出模式可配，0 为只输出当前帧角点坐标，1 为输出当前帧角点坐标和 status 信息，2 为输出当前帧角点坐标，status 信息，err 信息

STBoxFltAndEigCalc

Shi-tamosi-like 角点计算过程中的盒子滤波、求取角点响应值及最大角点响应值。

- 地址对齐方式：所有输入输出地址与 stride 都要求 16byte 对齐
- 输入输出数据格式：
 - 输入 16bit 水平与垂直方向间插数据，如图 8-12 所示
 - 输出 16bit 单分量数据，如图 8-7 所示
 - 输出 16bit 的 maxEig

STCandiCorner

实现从 Shi-Tomasi-like 候选角点筛选出角点。

- 图像分辨率：要求输入两图像的分辨率相同
- 地址对齐方式：所有输入输出地址要求 byte 对齐，stride 要求 16byte 对齐
- 输入输出数据格式：
 - 输入为 8bit 单分量图像
 - 输出为 8bit 单分量图像，如图 8-4 所示

SAD

分块计算两幅图像对应像素差值绝对值的累加和。

- 图像分辨率：宽高要求配置为块大小的整数倍
- 地址对齐方式：所有输入地址与 stride 要求 byte 对齐，输出地址要求 byte 对齐
- 输入输出数据格式：
 - 输入为 2 帧相同分辨率 8bit 单分量图像，如图 8-4 所示。
 - dst1 和 dst2 的结果取决于 out_fmt 配置：
 - a. 当 out_fmt 配置为 0x00 时，dst1 为 sad 值，16bit 单分量，如图 8-7 所示，dst2 为二值化图，8bit 单分量，如图 8-4 所示。
 - b. 当 out_fmt 配置为 0x01 时，dst1 为 sad 值，8bit 单分量，如图 8-4 所示，dst2 为二值化图，8bit 单分量，如图 8-4 所示。
 - c. 当 out_fmt 配置为 0x02 时，dst1 为 sad 值，16bit 单分量，如图 8-7 所示。
 - d. 当 out_fmt 配置为 0x03 时，dst1 为 sad 值，8bit 单分量，如图 8-4 所示。



- e. 当 out_fmt 配置为 0x04 时, dst1 为二值化结果, 8bit 单分量, 如图 8-4 所示。

Resize

支持双线性插值(bilinear), 支持最多 64 张图像输入, 水平缩放倍数[1, 16]、垂直缩放倍数[1, 16]。

- 地址对齐方式: 所有输入输出地址与 stride 都要求 16byte 对齐
- 输入输出数据格式:
 - 输入 8bit 单分量图像或 RGB Plannar 图像, 如图 8-4 和图 8-6 所示。
 - 输出缩放后的图像, 对应于输入图像格式。
- 缩放模式:
 - 一次任务中的图像缩放模式必须相同

GMM2

GMM 背景建模, 支持灰度图、RGB package 图像输入, Gauss 模型个数支持 1-5 个。

- 地址对齐方式: 所有输入输出地址与 stride 都要求 16byte 对齐
- 输入输出数据格式:
 - 输入 8bit 单分量图像或 RGB package 图像, 如图 8-4 和图 8-5 所示
 - 输入 16bit 像素因子图, 像素因子包含高 8bit 更新时长、低 8bit 灵敏度 sensitivity 信息, 如图 8-7 所示
 - 输入模型数据
 - 输出 8bit 单分量前景二值图, 如图 8-4 所示
 - 输出更新后的模型数据
 - 输出背景图像, 对应于输入图像类型
 - 输出 8bit 模型命中信息, 如图 8-4 所示, 低 1 比特表示模型命中与否, 高 7bit 表示命中模型索引;

PSP

投影校正, 支持 nonreflective、similarity、similarity affine 三种变换。输入为点对以及待校正的图像, 输出为校正后的图像。

- 图像分辨率: 输入分辨率: 20x20~104x1024; 输出分辨率: 20x20~256x256。
- 地址对齐方式: 输入任意对齐, 输出 16byte 对齐
- 输入输出数据格式:
 - 8bit 单分量—>8bit 单分量;
 - YUV SP420 —> YUV SP420 ;
 - YUV SP420 —> RGB Package。

8bit 单分量如图 8-4, YUV SP420 如图 8-3, RGB Package 如图 8-5 所示。



HOG

方向梯度直方图特征提取。输入为待求 HOG 特征的框信息，包括左上角坐标、框的宽和高。当框的宽和高超过 136 时，缩放到 136。

- 框的分辨率：40x40~104x1024。
- 地址对齐方式：输入任意对齐，输出 16byte 对齐
- 输入输出数据格式：
 - YUV SP420 → HOG 输出格式 1；
 - YUV SP420 → HOG 输出格式 2。

HOG 输出格式 1 如图 8-13，HOG 输出格式 2 如图 8-14 所示。

KCF

核相关滤波。只支持本模块的 HOG 特征输入。输入限制参考 HOG。

输出数据格式：

- 第一个 word 为水平方向偏移量 dx。
- 第二个 word 为垂直方向偏移量 dy。
- 第三个 word 为响应值。
- 第四个 word 为指示标志，1 表示框完全移出图像，0 表示框未完全移出图像。

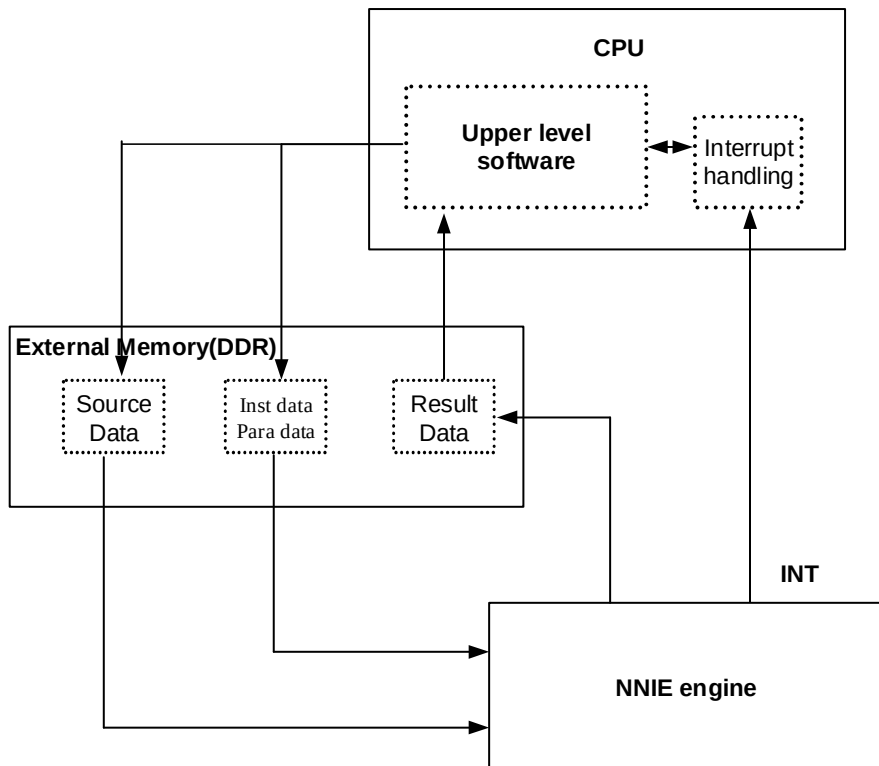
8.2 NNIE 加速引擎

8.2.1 概述

加速引擎 NNIE 是一款基于 CNN、RCNN 等神经网络结构的深度学习专用加速器，可用于图片分类、目标检测等应用场景，NNIE 加速引擎在系统中的位置如图 8-15 所示。芯片支持 1 个 NNIE。



图8-15 NNIE 加速引擎在系统中的位置



8.2.2 特点

NNIE 加速引擎特性如下:

- 支持 $N * N$ ^{注1} 卷积
- 支持 Pooling (Max 和 Average)
- 支持 Stride^{注2}
- 支持 Pad
- 支持激活函数 (Relu、Sigmoid 和 TanH)
- 支持 LRN 运算
- 支持 BN(Batch Normalization)
- 支持向量与矩阵的乘加运算 (Inner Product)
- 支持 Concat
- 支持 Eltwise
- 支持 8bit 的数据与参数模式
- 支持数据与参数位宽可配置
- 支持参数压缩和参数稀疏
- 支持输入图像为单通道 (灰度图) 和三通道 (RGB 格式)
- 支持图像预处理 (均值化和像素值缩放)
- 支持图像批处理



- 支持中间层结果上报



说明

注 1: $N * N$ 表示卷积框为正方形状。

注 2: 此处 Stride 特指神经网络专用术语。



目 录

9 视频接口.....	9-1
9.1 VI	9-1
9.1.1 概述	9-1
9.1.2 特点	9-1
9.1.3 模式功能描述.....	9-3
9.1.4 工作方式	9-10
9.1.5 VICAP 寄存器概览.....	9-13
9.1.6 VICAP 寄存器描述.....	9-19
9.2 VDP.....	9-58
9.2.1 概述	9-58
9.2.2 架构描述	9-58
9.2.3 工作方式	9-59
9.2.4 功能描述	9-60
9.2.5 VDP 寄存器概览.....	9-63
9.2.6 VDP 寄存器描述.....	9-69
9.3 MIPI Rx.....	9-114
9.3.1 概述	9-114
9.3.2 特点	9-115
9.3.3 功能描述	9-115
9.3.4 MIPI Rx 控制器工作方式.....	9-128
9.3.5 MIPI Rx 寄存器概览.....	9-130
9.3.6 MIPI Rx 寄存器描述.....	9-138
9.4 MIPI Tx.....	9-200
9.4.1 概述	9-200
9.4.2 特点	9-201
9.4.3 功能描述	9-201
9.4.4 配置流程	9-209
9.4.5 MIPI Tx 寄存器概览.....	9-211
9.4.6 MIPI Tx 寄存器描述.....	9-212
9.4.7 MIPI Tx PHY 寄存器概览.....	9-240



9.4.8 MIPI Tx PHY 寄存器描述.....9-242



插图目录

图 9-1 VI 功能框图	9-1
图 9-2 高清接口输入时序水平时序	9-3
图 9-3 高清接口输入时序垂直时序	9-4
图 9-4 720 行视频系统垂直时序	9-5
图 9-5 ITU-R BT.601 水平时序图	9-6
图 9-6 NTSC 制式垂直同步时序图	9-6
图 9-7 PAL 制式垂直同步时序图	9-7
图 9-8 数字摄像头水平时序	9-7
图 9-9 数字摄像头垂直时序脉冲方式	9-8
图 9-10 数字摄像头垂直时序行有效方式	9-8
图 9-11 有效图像区域与水平垂直消隐关系图	9-8
图 9-12 YUV4:2:2 的存储模式	9-9
图 9-13 RAW DATA 的存储模式	9-10
图 9-14 little endian 图像存储模式	9-10
图 9-15 VICAP 的硬件工作流程	9-11
图 9-16 软件操作流程	9-12
图 9-17 VIPROC 离线硬件工作流程	9-12
图 9-18 VIPROC 离线模式软件工作流程	9-13
图 9-19 VDP 总体框图	9-58
图 9-18 MIPI Rx 功能框图及在系统中的位置	9-115
图 9-19 CSI-2 数据包的传输机制	9-117
图 9-20 CSI-2 长包的格式	9-118
图 9-21 CSI-2 短包的格式	9-118
图 9-22 MIPI 接口图像格式	9-119
图 9-23 MIPI 接口宽动态数据传输（使用 VC）	9-120



图 9-24 MIPI 接口宽动态数据传输（使用 DT）	9-120
图 9-25 MIPI DOL 数据传输	9-121
图 9-26 LVDS 同步码和图像传输模式	9-123
图 9-27 LVDS 单个像素点时序	9-124
图 9-28 LVDS 同步方式 1	9-124
图 9-29 LVDS 同步方式 2	9-125
图 9-30 LVDS 宽动态模式 1（2 帧）	9-126
图 9-31 LVDS 宽动态模式 2（2 帧）	9-126
图 9-32 HiSPi Packetized-SP 数据格式	9-127
图 9-33 HiSPi 宽动态模式	9-128
图 9-34 MIPI Tx 功能框图及在系统中的位置	9-201
图 9-35 RGB 16-bit 格式	9-202
图 9-36 RGB 18-bit non-loosely 格式	9-203
图 9-37 RGB 18-bit loosely 格式	9-203
图 9-38 RGB 24-bit 格式	9-203
图 9-39 YUV4:2:0 12-bit 格式	9-204
图 9-40 YUV4:2:2 16-bit 格式	9-204
图 9-41 YUV4:2:2 20-bit 格式	9-204
图 9-42 Legacy YUV420 8-bit 格式	9-205
图 9-43 YUV420 8-bit 格式	9-205
图 9-44 YUV422 8-bit 格式	9-206
图 9-45 Burst Mode	9-207
图 9-46 Non-Burst Mode with Sync Pulses	9-207
图 9-47 Non-Burst Mode with Sync Events	9-208
图 9-48 CSI-2 帧时序	9-208



表格目录

表 9-1 ITU-R BT.656 YUV4:2:2 行数据格式.....	9-4
表 9-2 SAV/EAV 格式	9-5
表 9-1 各模块的寄存器偏移地址变量表.....	9-13
表 9-2 VICAP 寄存器概览（基址是 0x1130_0000）	9-13
表 9-3 VDP 寄存器概览（基址是 0x1144_0000）	9-63
表 9-7 MIPI Rx 支持的接口类型	9-116
表 9-8 LVDS 同步码格式	9-122
表 9-9 LVDS 同步码第 4 个字段举例.....	9-122
表 9-10 HiSPi 传输模式.....	9-127
表 9-11 MIPI Rx 寄存器偏移地址变量表.....	9-130
表 9-12 MIPI Rx 寄存器概览（基址是 0x113A_0000）	9-130
表 9-13 MPI Tx 寄存器概览（基址是 0x1127_0000）	9-211
表 9-14 MIPI Tx PHY 寄存器概览（基址是 0x00）	9-240



9 视频接口

9.1 VI

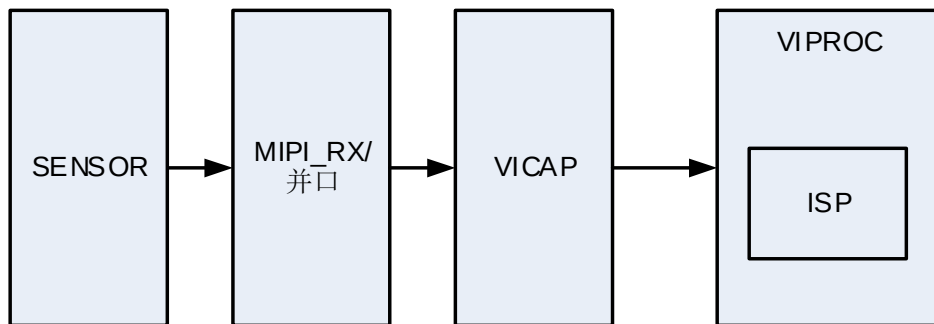
9.1.1 概述

视频输入单元 VI (Video Input)，可以通过 MIPI Rx (包含 MIPI、LVDS、HiSPi) 接口、BT.656/601、BT.1120 接口和 DC (Digital Camera) 接收视频数据，存入指定的内存区域。VI 内嵌 ISP 图像处理单元，可以直接对接外部原始数据 (BAYER RGB 数据)，VI 的功能框图如图 9-1 所示。

VI 分成两个物理子模块：捕获子模块 VICAP 和处理子模块 VIPROC 组成。其中：

- VICAP 完成多路视频输入的数据捕获功能，并将捕获的数据存放到 DDR 或者在线送给 VIPROC。VIPROC 用以支持离线模式（从 DDR 读取数据）或者在线模式（从 VICAP 接收在线）视频数据处理。
- VIPROC 支持将处理数据写到 DDR 或者直接在线送给 VPSS。

图9-1 VI 功能框图



9.1.2 特点

VICAP 子模块有以下特点：



说明

VICAP 工作时钟请参见 3.2.6 CRG 寄存器章节。



- 最大支持 2 路 sensor 数据输入。
 - 接口 0 最大支持 2688 x 1944@20fps 的 2F-WDR RAW 输入
 - 接口 1 最大支持 2048 x 1536@30fps 的 2F-WDR RAW 输入
- 输入支持最大数据位宽为 14bit。
- 支持逐行输入模式。
- 支持 BT.656、BT.601、BT.1120 和 DC 接口。
- 支持 MIPI、LVDS、HiSPi 接口。
- 支持 MIPI 接口输入 YUV 格式
 - YUV422、YUV420 格式
- 支持 flash trigger
- 支持 shutter trigger
- 支持 sensor 主模式和从模式
- 支持数据压缩
- 支持在线输出图像数据到 VIPROC。
- 支持离线输出图像数据到 DDR。
 - 支持 4 个写出通道
 - CH0~3 只支持 RAW 格式写出
 - CH2 只支持 YUV 格式写出
- 离线输出数据格式支持：
 - Semi-planar YUV4:2:2 模式
 - Semi-planar YUV4:2:0 模式
 - Semi-planar YUV4:0:0 模式
 - RAW 模式
- 支持安全特性（可开关）
 - 寄存器仅支持安全 CPU 访问
 - 支持各读写通道的安全属性独立可配

VIPROC 子模块有以下特点：

- 支持在线模式（数据来自 VICAP 在线传输）。
- 支持离线模式处理（数据来自 DDR）。
- 最大处理性能为 2688 x 1536@30fps（2 路总性能）
- 内嵌 ISP 处理功能。
- 支持一个通道的视频输出。
- 支持数据压缩
- 支持输出图像数据到 DDR。
- 输出数据格式支持：
 - Semi-planar YUV4:2:2 模式
 - Semi-planar YUV4:2:0 模式



- Semi-planar YUV4:0:0 模式
- RAW 模式
- 支持安全特性（可开关）
 - 寄存器仅支持安全 CPU 访问
 - 支持各读写通道的安全属性独立可配

9.1.3 模式功能描述

9.1.3.1 典型应用

VI 是一个支持多种时序输入的视频输入采集处理单元，系统可以配置不同的功能模式，使之可以灵活地适应不同的输入视频接口。

VI 模块最多可以支持 2 路视频输入信号。

VI 的典型输入有以下 3 种：

- 1 路 4M(2688x1536)@30fps RAW（或 2F-WDR）输入
- 1 路 5M(2688x1944)@20fps RAW（或 2F-WDR）输入
- 2 路 2M（1920x1080）@30fps RAW（或 2F-WDR）输入

9.1.3.2 功能原理

BT 1120 接口时序

VICAP 支持 Y/C 分开输入的 BT.1120 接口时序，同步码和 BT.656 时序一致。此时需要用到 16bit 数据线，8bit 用来传输亮度，另外 8bit 用来传输色度，如图 9-2 和图 9-3 所示。

图9-2 高清接口输入时序水平时序

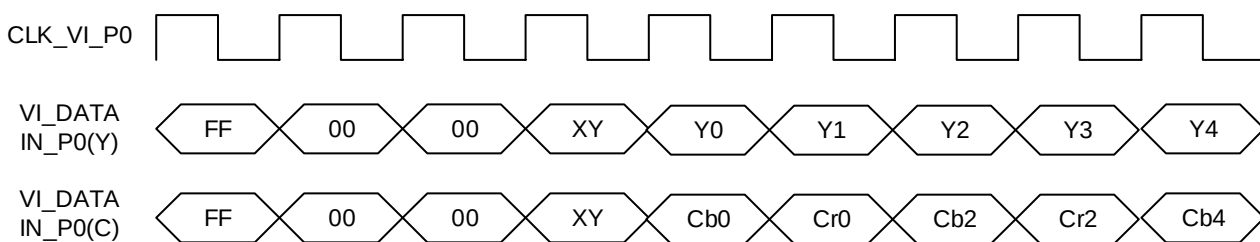
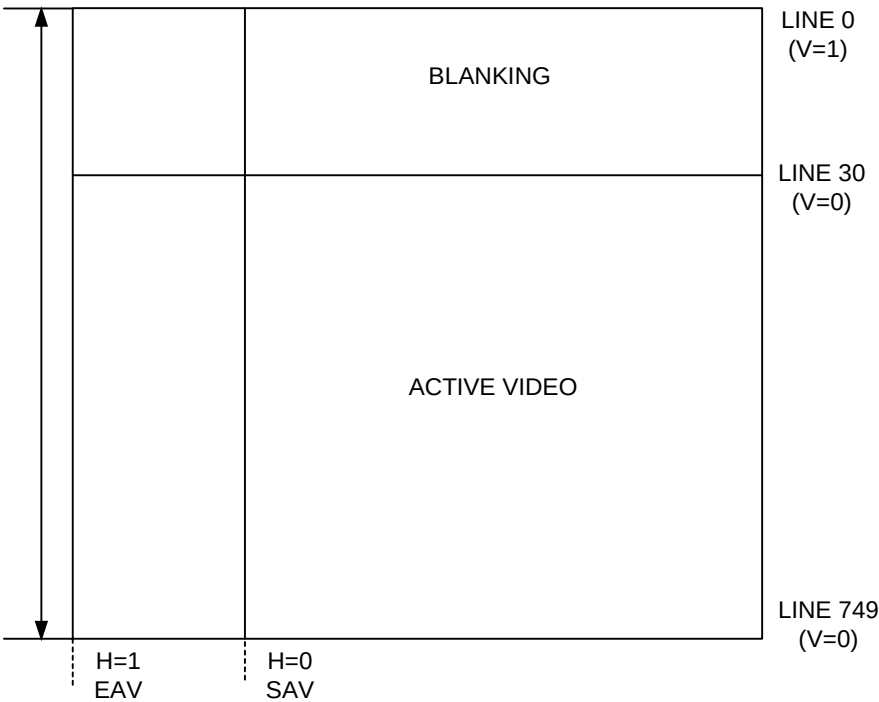




图9-3 高清接口输入时序垂直时序



BT.656 接口时序

VICAP 支持以 BT.656 时序格式输入的逐行 YUV422 数据。

- 水平时序

在 ITU-R BT.656 协议中，同步信号集成在数据流中，在数据流中的特殊字节 SAV 和 EAV 分别表示有效行数据的开始和结束。在视频数据流中，由 FF 00 00（FF、00 为图像编码数据的保留值，为非图像数据）构成的定时基准码字的码头来标志紧接着的一个字节为 SAV 或者 EAV，ITU-R BT. 656 的行数据流格式如表 9-1 所示。

表9-1 ITU-R BT.656 YUV4:2:2 行数据格式

定时基准码				行消隐区				定时基准码				720 有效像素 YUV4:2:2							
FF	00	00	EAV	80	10	...	80	10	FF	00	00	SAV	U0	Y0	V0	Y1	...	V718	Y719

SAV 和 EAV 的相互区别由 SAV/EAV 的特殊比特位“H”区分，SAV/EAV 还包含了垂直消隐位“V”和场指示位“F”。SAV/EAV 的具体描述见表 9-2 所示。



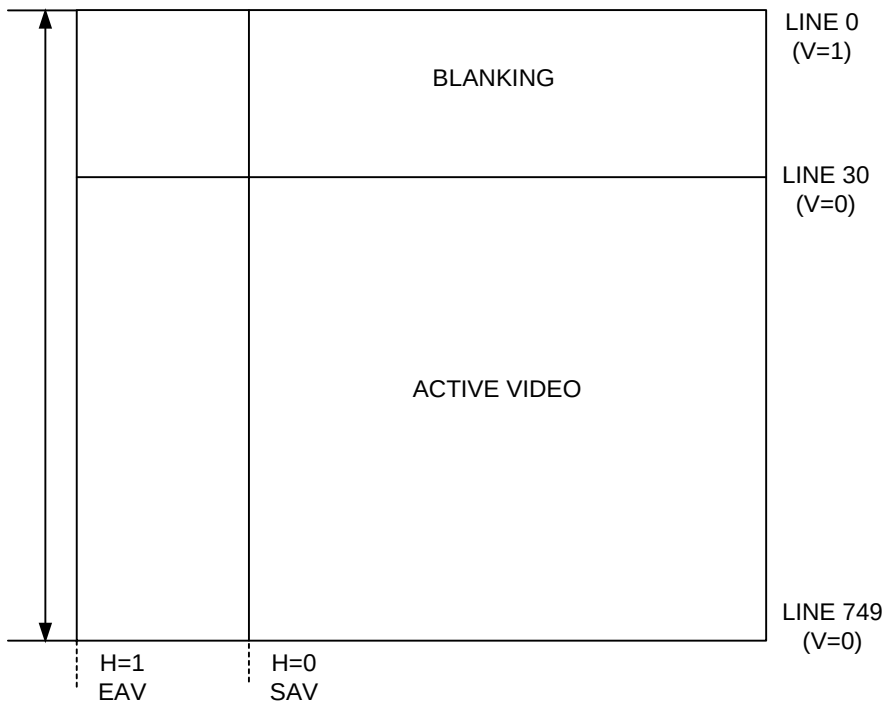
表9-2 SAV/EAV 格式

bit[7]	bit[6](F)	bit[5](V)	bit[4](H)	bit[3:0](P3~P0)
固定值 1	场指示位 1st field: F=0 2nd field: F=1	垂直消隐位 VBI: V=1 Active video: V=0	SAV: H=0 EAV: H=1	校验位

● 垂直时序

垂直时序也是通过定时基准码 SAV/EAV 的 “V” 来实现，典型的 720 行的行视频系统垂直时序如图 9-4 所示。

图9-4 720 行视频系统垂直时序



注：VICAP 内部只根据 SAV/EAV 来识别垂直时序，不会限制具体在哪一行。

BT.601 接口时序

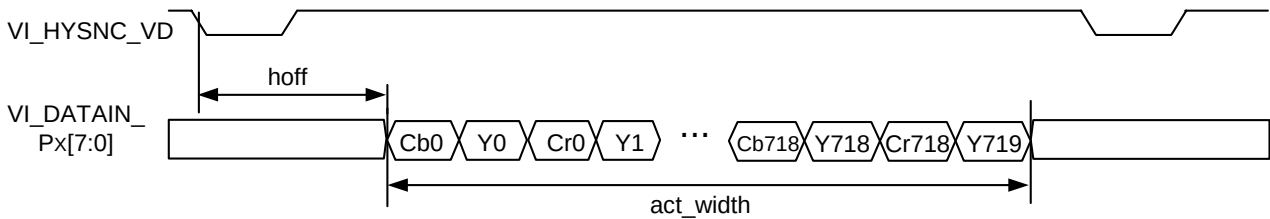
VICAP 支持以 BT.601 时序输入的 YUV422 逐行数据。

● 水平时序

水平脉冲表示新的一行开始如图 9-5 所示。经过 hoff（可配，NTSC525 行系统该值为 244，PAL625 行系统该值为 264 个时钟）后，行前消隐区结束，进入行有效数据区，经过 act_width（可配，典型值为 720 或 704）个时钟后行有效区域结束，进入行后消隐区。水平同步极性可配。



图9-5 ITU-R BT.601 水平时序图

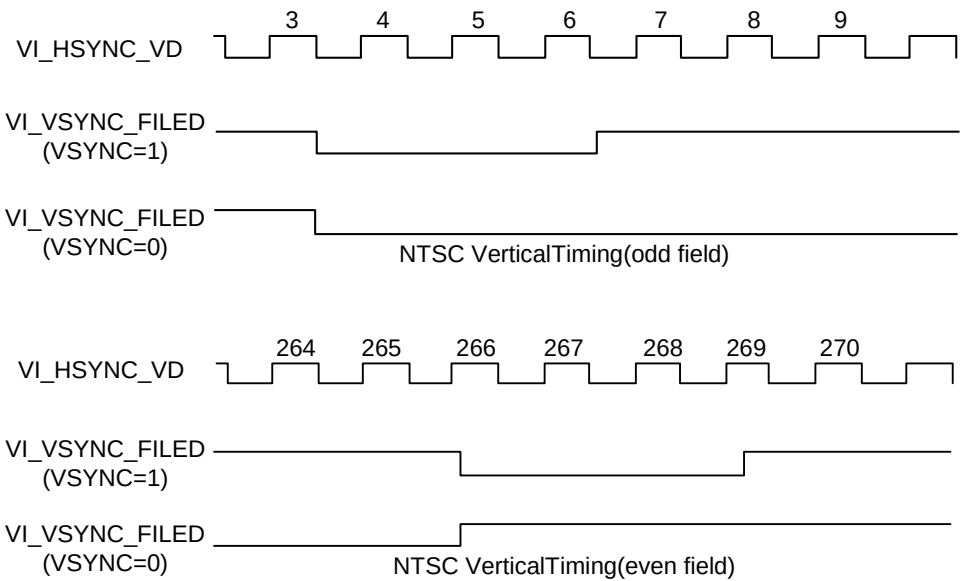


● 垂直时序

ITU-R BT.601 建议：信号 VSYNC 作为垂直同步信号。VSYNC 的脉冲标志帧的开始，VICAP 支持以下 2 种垂直同步方法。

VICAP 在 NTSC 制式（525 行）和 PAL 制式（625 行）的垂直时序关系图如图 9-6 和图 9-7 所示，其中 VI_HSYNC_VD 为水平同步脉冲，VI_VSYNC_FIELD 在 VSYNC=1 时为垂直同步脉冲；在 VSYNC=0 时为场同步信号。

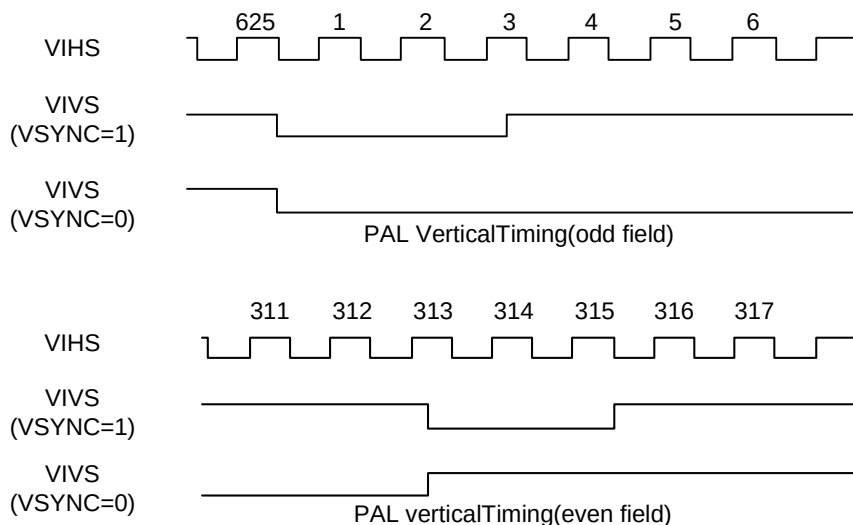
图9-6 NTSC 制式垂直同步时序图



在 NTSC 隔行扫描制式下，第 1 场的垂直同步信号在第 4 行的起始位置变为低电平，持续 3 行低电平后，在第 7 行的起始位置变为高电平。VICAP 接收从第 22 行开始到第 261 行结束的 240 行数据。第 2 场的垂直同步信号在第 266 行的中间位置变为低电平，持续 3 行低电平后，在第 269 行中间位置变为高电平。VICAP 接收从第 285 行开始第 524 行结束的 240 行数据。



图9-7 PAL 制式垂直同步时序图



在 PAL 隔行扫描制式下，第 1 场的垂直同步信号在第 1 行起始位置变为低电平，持续 2.5 行低电平后，在第 3 行的中间位置变为高电平。VICAP 接收从第 24 行开始到第 310 行结束的 288 行数据。第 2 场的垂直同步信号在第 313 行的中间位置变为低电平，持续 2.5 行低电平后，在第 316 行的起始位置变为高电平。VICAP 接收从第 336 行开始到第 623 行结束的 288 行数据。

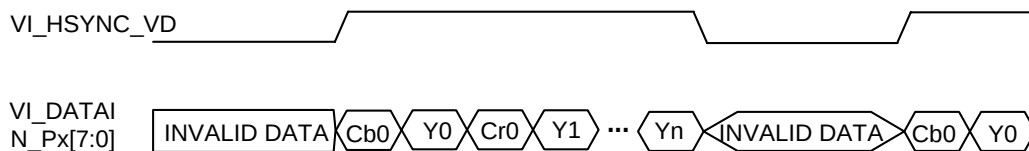
上面两种为典型的 BT.601 垂直时序，VICAP 内部支持场开始到场有效行开始之间的行数可配，场有效行的行数可配，垂直同步的极性可配。

数字摄像头（DC）接口时序

• 水平时序

VICAP 接数字摄像头时，VI_HSYNC_VD 表示数据有效信号，数据有效信号极性可配，水平时序如图 9-8 所示。

图9-8 数字摄像头水平时序



• 垂直时序

VICAP 支持两种垂直时序脉冲方式和行有效方式，如图 9-9 和图 9-10 所示。垂直同步极性可配。



图9-9 数字摄像头垂直时序脉冲方式

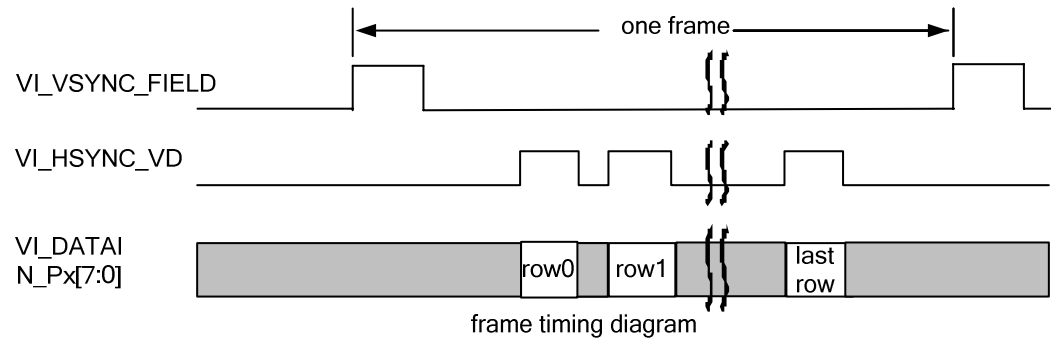
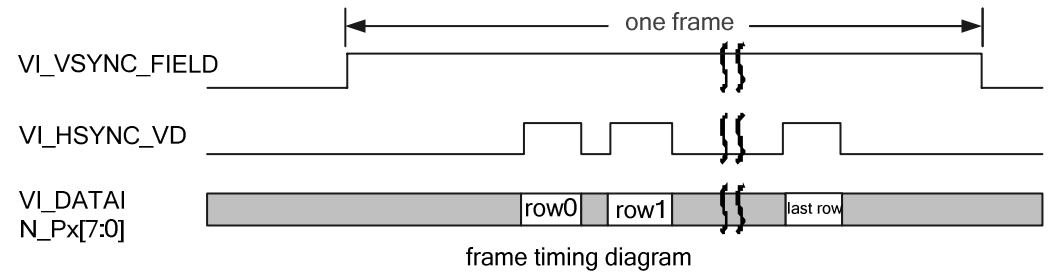


图9-10 数字摄像头垂直时序行有效方式

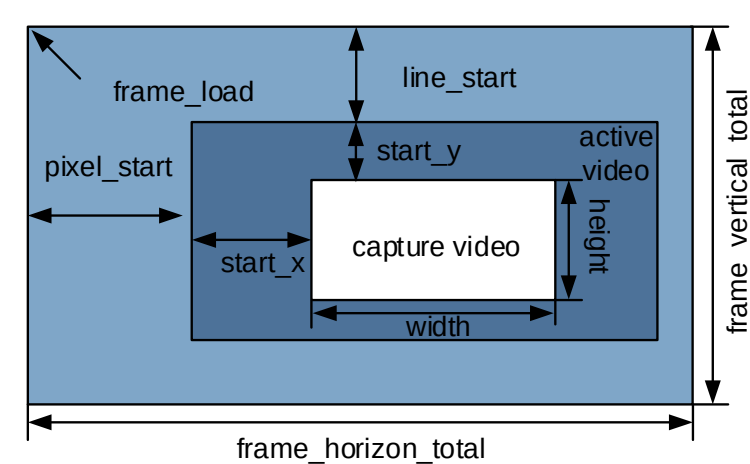


对于 VICAP 内部处理来说，这两种时序相同，VICAP 内部检测到一个上升沿或下降沿，这认为是一帧的开始，然后检测数据有效信号，来判断当前数据是否有效。

9.1.3.3 图像 CROP

有效视频范围如图 9-11 所示。开始于水平消隐和垂直消隐之后。而实际显示的视图区域常常包含在有效视频范围之内。

图9-11 有效图像区域与水平垂直消隐关系图





9.1.3.4 图像存储模式

图像存储模式包括：

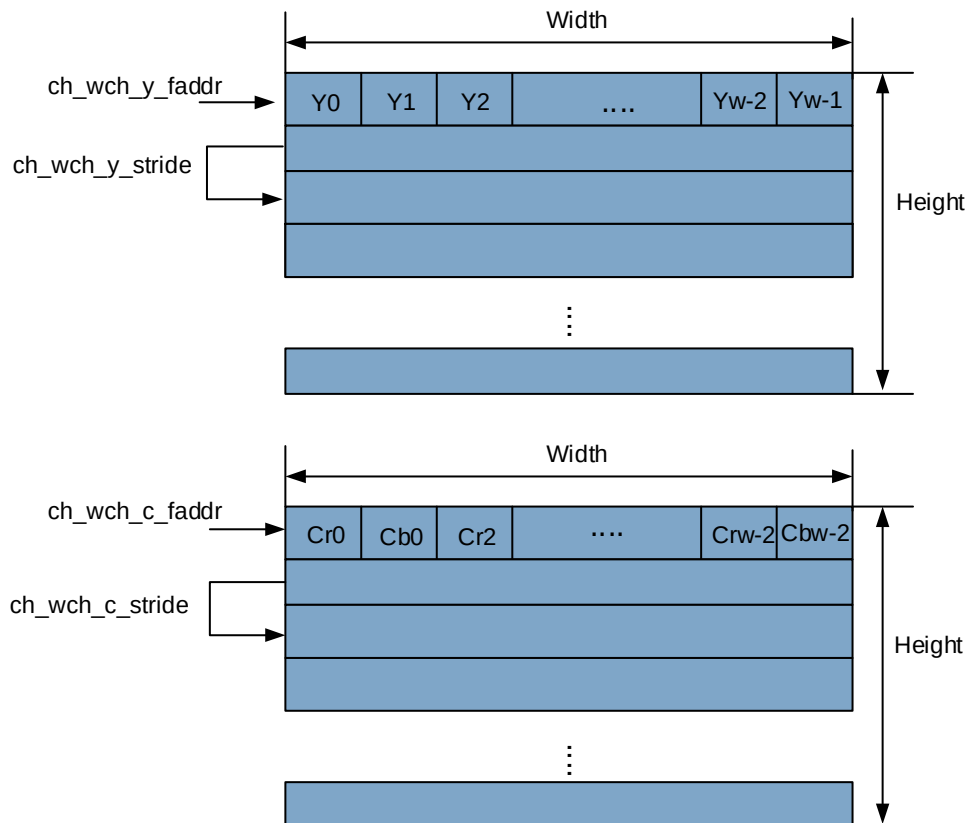
1. YUV 数据存储

- semi-planar YUV 存储。

系统设定了视图区域后，对读入数据按照 semi-planar 方式存储，即亮度分量和色度分量分别存储在 DDR 中的亮度存储空间和色度存储空间。

- 在 1 行内，亮度、色度分量各自连续存储。
- 连续 2 行之间的存储，可以通过系统定义的行首与行首之间的存储间隔参数 stride 定义。
- 亮度和色度分量在 DDR 中的存储位置由起始地址 base_addr 来指示。VI 捕获的 YUV4:2:2 数据的存储结构如图 9-12 所示。

图9-12 YUV4:2:2 的存储模式



2. RAW 数据存储

- 按单分量方式存储。
- 1 行内，RAW 数据连续存储。
- 连续 2 行之间的存储，可以通过系统定义的行首与行首之间的存储间隔参数 stride 定义。

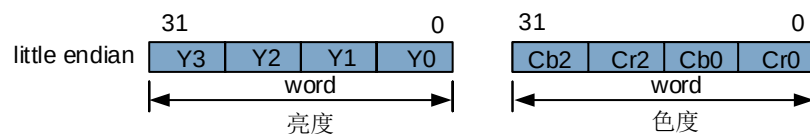
- DDR 中的存储位置由起始地址 `base_addr` 来指示。
VI 捕获的 RAW 数据的存储结构如图 9-12 所示。

图9-13 RAW DATA 的存储模式



在 DDR 中，数据的存储是以 word（32bit）为单位。由 4 个 8bit 像素组成一个 32bit 的 word。如图 9-14。

图9-14 little endian 图像存储模式



VICAP 支持 little endian 方式存储数据到 DDR，存储地址为 16byte 对齐。

9.1.4 工作方式

9.1.4.1 VICAP 的 `reg_newer` 功能

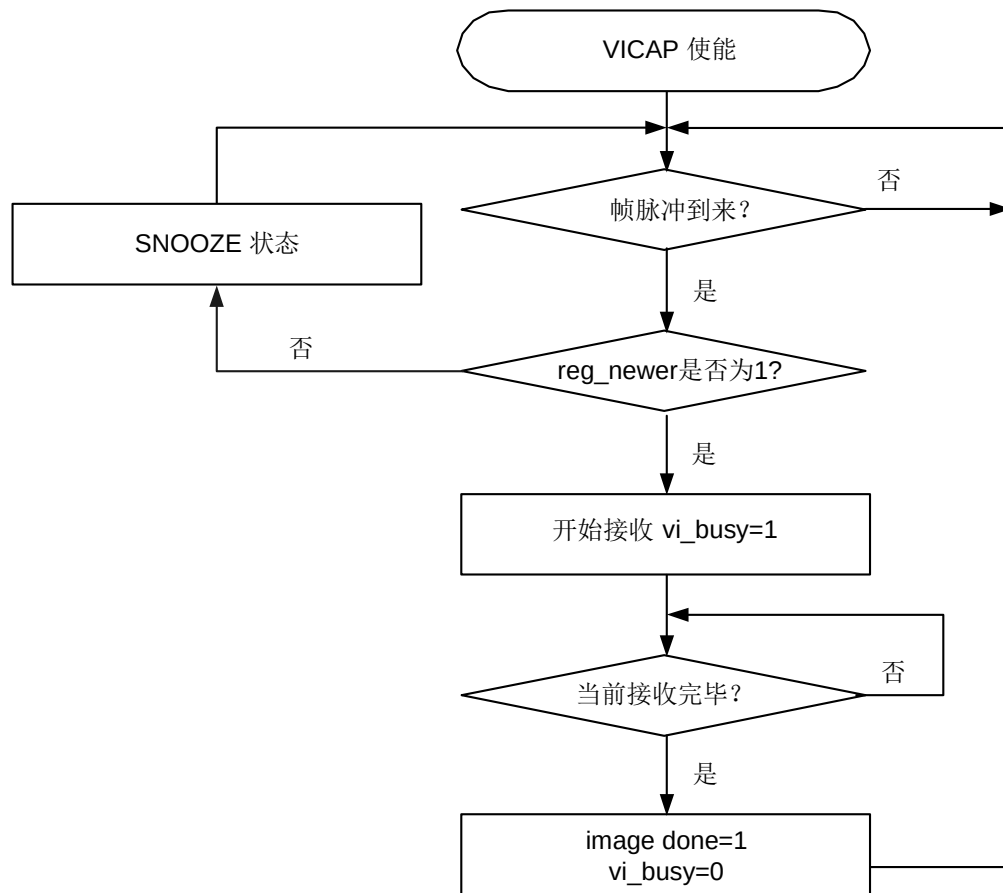
- 在软件使能 VICAP 通道之前，软件应该完成以下操作：
 - 完成对 VICAP 的属性寄存器的配置操作。
 - 写 `reg_newer` 位为“1”，通知 VICAP 模块当前的寄存器已经准备就绪。
- 使能 VICAP 后，VICAP 逻辑开始工作，当一帧到来的时候，则有：
 - 如果 `reg_newer` 为 0，则 VICAP 将不会接收数据，置硬件状态为 SNOOZE（以下各硬件状态为图 9-15 所用），等待下一帧的数据的到来。
 - 如果 `reg_newer` 为 1，则开始接收数据，同时给出寄存器更新中断 (`reg_update_int`)，并设置硬件状态为 busy。
- 当接收完毕当前数据后，清除硬件 busy 状态。等到下一帧到来的时候，则：
 - 如果 `reg_newer` 为 0，则放弃下一帧数据的接收。
 - 如果 `reg_newer` 为 1，则可以紧接着前一次数据继续接收下一帧的数据。



9.1.4.2 在线模式硬件工作流程

VICAP 的硬件工作流程如图 9-15 所示。

图9-15 VICAP 的硬件工作流程



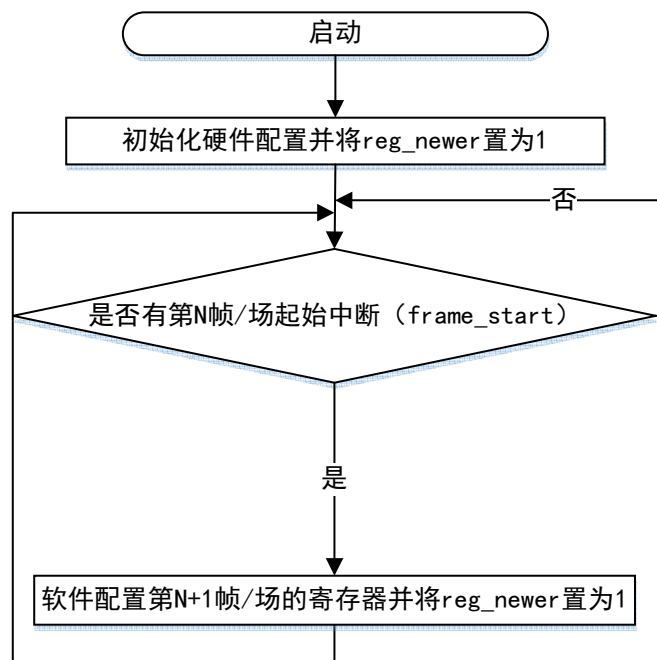
在 VICAP 工作时，每接收完一帧数据，在下一帧的到来时，VICAP 将检测 `reg_newer` 位。如果 `reg_newer` 位为 1（表示软件已经更新或者确认 VICAP 的寄存器），VICAP 将自动 load 软件所配置的寄存器值到工作寄存器（工作寄存器软件不可访问），然后将 `reg_newer` 位清 0，并开始接收下一帧数据。否则，只有等到 `reg_newer` 为 1 且新的一帧到来时开始接收数据。

9.1.4.3 在线模式软件配置流程

在中断方式下，软件的操作流程如图 9-16 所示。



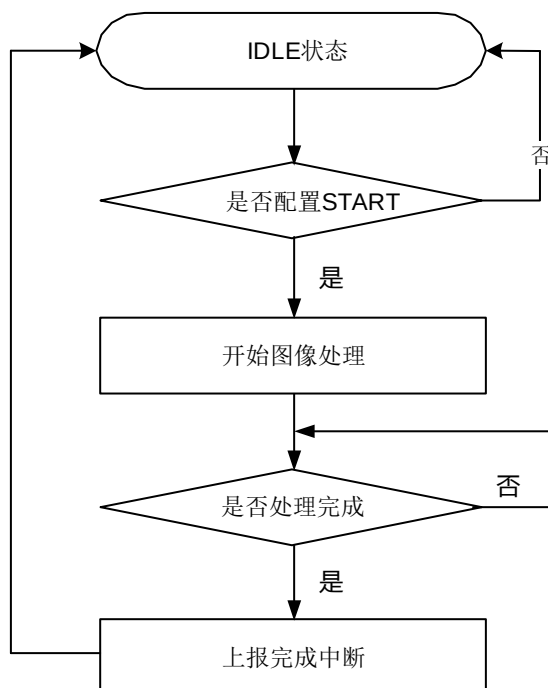
图9-16 软件操作流程



9.1.4.4 VIPROC 离线模式硬件工作流程

VIPROC 子模块支持离线模式，以时分复用的方式处理多路数据。如图 9-17 所示。

图9-17 VIPROC 离线硬件工作流程

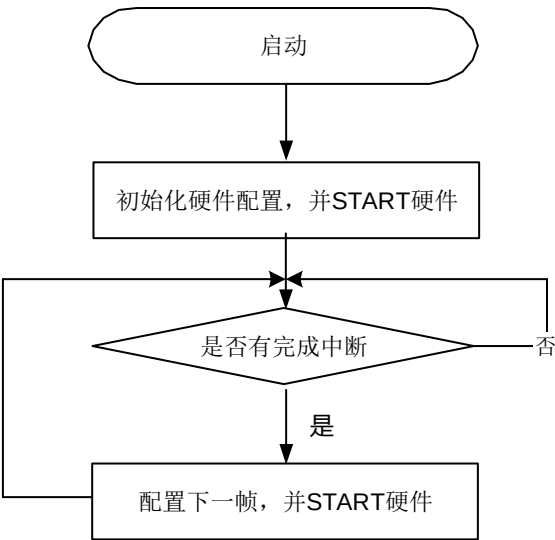




9.1.4.5 VIPROC 离线模式软件配置流程

软件配置流程和如下图 9-18 所示。

图9-18 VIPROC 离线模式软件工作流程



9.1.5 VICAP 寄存器概览

各模块的寄存器偏移地址中变量的取值范围和含义如表 9-1 所示。

表9-1 各模块的寄存器偏移地址变量表

变量名称	取值范围	描述
CH_N	0~3	VICAP 模块的 4 个输出 CH 通道
PT_N	0~1	VICAP 模块的 2 个输入通道

VICAP 寄存器概览如表 9-2 所示。

表9-2 VICAP 寄存器概览（基址是 0x1130_0000）

偏移地址	名称	描述	页码
0x0000	WK_MODE	全局工作配置寄存器	9-19
0x0010	AXI_CFG	总线配置寄存器	9-19
0x001C	AXI_WR_PROT	AXI 总线写通道安全属性寄存器	9-20



偏移地址	名称	描述	页码
0x0050	CMUX_SEL	CMUX 通路选择寄存器	9-21
0x0054	CMUX_ID_CFG	CMUX 帧 ID 选择寄存器	9-21
0x0058	ISP_MODE	ISP 模式选择寄存器	9-22
0x00F0	VICAP_INT	中断指示寄存器	9-23
0x00F8	VICAP_INT_MASK	中断屏蔽寄存器	9-25
0x0100	SLAVE0_MODE_CFG	SENSOR0 从模式配置寄存器	9-26
0x0104	SLAVE0_MODE_VS_TIME_H	SENSOR0 从模式输出 VSYNC 的周期配置寄存器	9-27
0x0108	SLAVE0_MODE_VS_TIME_L	SENSOR0 从模式输出 VSYNC 的周期配置寄存器	9-27
0x010C	SLAVE0_MODE_HS_TIME	SENSOR0 从模式输出 HSYNC 的周期配置寄存器	9-28
0x0110	SLAVE0_MODE_VS_OUTPUT_TIME_H	SENSOR0 从模式输出 VS_OUT 的周期配置寄存器	9-28
0x0114	SLAVE0_MODE_VS_OUTPUT_TIME_L	SENSOR0 从模式输出 VS_OUT 的周期配置寄存器	9-28
0x0118	SLAVE0_MODE_VS_CYC_H	SENSOR0 从模式输出 VSYNC 脉冲有效信号持续时间配置寄存器	9-28
0x011C	SLAVE0_MODE_VS_CYC_L	SENSOR0 从模式输出 VSYNC 脉冲有效信号持续时间配置寄存器	9-29
0x0120	SLAVE0_MODE_HS_CYC	SENSOR0 从模式输出 HSYNC 脉冲有效信号持续时间配置寄存器	9-29
0x0124	SLAVE0_MODE_VS_OUTPUT_CYC_H	SENSOR0 从模式输出 VS_OUT 脉冲有效信号持续时间配置寄存器	9-29
0x0128	SLAVE0_MODE_VS_OUTPUT_CYC_L	SENSOR0 从模式输出 VS_OUT 脉冲有效信号持续时间配置寄存器	9-29
0x012C	SLAVE0_MODE_HS_DELAY_CYC	SENSOR0 从模式输出 HSYNC 脉冲输出相对 VSYNC 脉冲的延迟周期配置	9-30



偏移地址	名称	描述	页码
0x0130	SLAVE0_MODE_VS_OUT_DLY_CYC_H	SENSOR0 从模式输出 VS_OUT 脉冲输出相对 VSYNC 脉冲的延迟周期配置	9-30
0x0134	SLAVE0_MODE_VS_OUT_DLY_CYC_L	SENSOR0 从模式输出 VS_OUT 脉冲输出相对 VSYNC 脉冲的延迟周期配置	9-30
0x01FC	SLAVE_MODE_SYN	SENSOR 同步信号，上升沿有效	9-30
0x0200	ONLINE0_CFG	在线通道 0 控制寄存器	9-31
0x02AC	ONLINE0_SIZE	ONLINE0 输出有效区大小寄存器	9-31
0x0FE4	VICAP_AXI0_WR_ERROR	AXI0 写错误状态寄存器	9-31
0x0FF0	VICAP_DBG_INT	系统异常中断状态寄存器	9-32
0x0FF8	VICAP_DBG_INT_MASK	系统异常状态屏蔽寄存器	9-32
0x1000+ PT_N×0x100	PT_INTF_MOD	接口模式寄存器	9-33
0x1010+ PT_N×0x100	PT_OFFSET0	0 分量 offset 寄存器	9-33
0x1014+ PT_N×0x100	PT_OFFSET1	1 分量 offset 寄存器	9-34
0x1018+ PT_N×0x100	PT_OFFSET2	2 分量 offset 寄存器	9-34
0x1020+ PT_N×0x100	PT_BT656_CFG	BT656 配置寄存器	9-34
0x1038+ PT_N×0x100	PT_UNIFY_TIMING_CFG	时序配置寄存器	9-35
0x103C+ PT_N×0x100	PT_GEN_TIMING_CFG	时序恢复模块配置寄存器	9-37
0x1040+ PT_N×0x100	PT_UNIFY_DATA_CFG	数据配置寄存器	9-38
0x1044+ PT_N×0x100	PT_GEN_DATA_CFG	数据产生模块配置寄存器	9-38
0x1048+ PT_N×0x100	PT_GEN_DATA_COEF	数据产生模块系数寄存器	9-39



偏移地址	名称	描述	页码
0x104C + PT_N × 0x100	PT_GEN_DATA_INIT	数据产生初始值配置寄存器	9-40
0x1050 + PT_N × 0x100	PT_YUV444_CFG	YUV444 配置寄存器	9-40
0x1060 + PT_N × 0x100	PT_FSTART_DLY	端口 fstart 中断延时寄存器	9-41
0x1064 + PT_N × 0x100	PT_FSTART_H_DLY	端口 fstart 中断行延时延时寄存器	9-41
0x1080 + PT_N × 0x100	PT_INTF_HFB	水平前消隐区宽度寄存器	9-41
0x1084 + PT_N × 0x100	PT_INTF_HACT	水平有效区宽度寄存器	9-41
0x1088 + PT_N × 0x100	PT_INTF_HBB	水平后消隐区宽度寄存器	9-42
0x108C + PT_N × 0x100	PT_INTF_VFB	垂直 前消隐区宽度寄存器	9-42
0x1090 + PT_N × 0x100	PT_INTF_VACT	垂直有效区宽度寄存器	9-42
0x1094 + PT_N × 0x100	PT_INTF_VBB	垂直后消隐区宽度寄存器	9-42
0x1098 + PT_N × 0x100	PT_INTF_VBFB	垂直底场前消隐区宽度寄存器	9-43
0x109C + PT_N × 0x100	PT_INTF_VBACT	垂直底场有效区宽度寄存器	9-43
0x10A0 + PT_N × 0x100	PT_INTF_VBBB	垂直底场后消隐区宽度寄存器	9-43
0x10A4 + PT_N × 0x100	PT_ID_CFG	ID 配置寄存器	9-43
0x10C0 + PT_N × 0x100	PT_FLASH_CFG	闪光灯配置寄存器	9-44
0x10C4 + PT_N × 0x100	PT_FLASH_CYC0	闪光灯时序 0 宽度寄存器	9-45
0x10C8 + PT_N × 0x100	PT_FLASH_CYC1	闪光灯时序 1 宽度寄存器	9-45
0x10D0 + PT_N × 0x100	PT_SHUTTER_CYC0	快门时序 0 宽度寄存器	9-45



偏移地址	名称	描述	页码
0x10D4+ PT_N×0x100	PT_SHUTTER_CYC1	快门时序 1 宽度寄存器	9-46
0x10D8+ PT_N×0x100	PT_SHUTTER_CYC2	快门时序 2 宽度寄存器	9-46
0x10DC+ PT_N×0x100	PT_SHUTTER_CYC3	快门时序 3 宽度寄存器	9-46
0x10E0+ PT_N×0x100	PT_STATUS	端口状态寄存器	9-46
0x10E4+ PT_N×0x100	PT_BT656_STATUS	BT656 状态寄存器	9-47
0x10EC+ PT_N×0x100	PT_SIZE	输入图像有效区大小指示寄存器	9-47
0x10F0+ PT_N×0x100	PT_INT	端口中断指示寄存器	9-47
0x10F8+ PT_N×0x100	PT_INT_MASK	端口中断屏蔽寄存器	9-48
0x10000+ CH_N× 0x1000	CH_CTRL	通道控制寄存器	9-49
0x10004+ CH_N× 0x1000	CH_REG_NEWER	通道更新寄存器	9-49
0x10034+ CH_N× 0x1000	CH_DLY_CFG	通道输入图像起始中断延迟配置寄存器	9-49
0x10040+ CH_N× 0x1000	CH_CROP_CFG	通道 CROP 使能寄存器	9-50
0x10044+ CH_N× 0x1000	CH_CROP0_START	通道 0 区域 CROP 起始位置寄存器	9-50
0x10048+ CH_N× 0x1000	CH_CROP0_SIZE	通道 0 区域 CROP 大小置寄存器	9-50
0x10050+ CH_N× 0x1000	CH_SKIP_Y_CFG	通道 Y SKIP 配置寄存器	9-50



偏移地址	名称	描述	页码
0x10058+ CH_N× 0x1000	CH_SKIP_C_CFG	通道 C SKIP 配置寄存器	9-51
0x10080+ CH_N× 0x1000	CH_WCH_Y_CFG	Y 分量配置寄存器	9-51
0x10084+ CH_N× 0x1000	CH_WCH_Y_SIZE	Y 分量存储大小寄存器	9-52
0x10094+ CH_N× 0x1000	CH_WCH_Y_FADDR_ L	Y 分量存储基地址低 32bit 寄存器	9-52
0x100A0+ CH_N× 0x1000	CH_WCH_Y_STRIDE	Y 分量行偏移寄存器	9-52
0x100B0+ CH_N× 0x1000	CH_WCH_C_CFG	C 分量配置寄存器	9-53
0x100B4+ CH_N× 0x1000	CH_WCH_C_SIZE	C 分量存储大小寄存器	9-53
0x100C4+ CH_N× 0x1000	CH_WCH_C_FADDR_ L	C 分量存储基地址低 32bit 寄存器	9-53
0x100D0+ CH_N× 0x1000	CH_WCH_C_STRIDE	C 分量行偏移寄存器	9-54
0x100E8+ CH_N× 0x1000	CH_Y_ACT_SIZE	亮度通道输出图像有效区宽高指示寄存器	9-54
0x100EC+ CH_N× 0x1000	CH_C_ACT_SIZE	色度通道输出图像有效区宽高指示寄存器	9-54
0x100F0+ CH_N× 0x1000	CH_INT	通道原始中断寄存器	9-54
0x100F8+ CH_N× 0x1000	CH_INT_MASK	通道中断屏蔽寄存器	9-55



偏移地址	名称	描述	页码
0x10200+ CH_N× 0x1000	CH_WCH_STT_CFG	ISP 统计信息写出配置寄存器	9-56
0x10208+ CH_N× 0x1000	CH_WCH_STT_AE_HI ST_ADDR_L	ISP AE_HIST 统计信息存储基地址低 32bit 寄存器	9-56
0x10210+ CH_N× 0x1000	CH_WCH_STT_AE_A VER_R_GR_ADDR_L	ISP AE_AVER_R_GR 统计信息存储基地址低 32bit 寄存器	9-57
0x10218+ CH_N× 0x1000	CH_WCH_STT_AE_A VER_GB_B_ADDR_L	ISP AE_AVER_GB_B 统计信息存储基地址低 32bit 寄存器	9-57

9.1.6 VICAP 寄存器描述

WK_MODE

WK_MODE 为全局工作配置寄存器。

Offset Address: 0x0000 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	-	reserved	保留。	0x1
[0]	RW	power_mode	时钟模式。 0: 低功耗关闭; 1: 低功耗打开。	0x1

AXI_CFG

AXI_CFG 为总线配置寄存器。

Offset Address: 0x0010 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:4]	RW	w_outstanding	写请求 outstanding 个数。 outstanding 的取值范围为[0, 7]。	0x0



Bits	Access	Name	Description	Reset
[3:0]	-	reserved	保留。	0x0

AXI_WR_PROT

AXI_WR_PROT 为 AXI 总线写通道安全属性寄存器。

Offset Address: 0x001C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:12]	-	reserved	保留。	0x000
[11]	RW	isp3_stt_prot	ISP3 STT 通道安全属性： 0: 安全； 1: 非安全。	0x0
[10]	RW	isp2_stt_prot	ISP2 STT 通道安全属性： 0: 安全； 1: 非安全。	0x0
[9]	RW	isp1_stt_prot	ISP1 STT 通道安全属性： 0: 安全； 1: 非安全。	0x0
[8]	RW	isp0_stt_prot	ISP0 STT 通道安全属性： 0: 安全； 1: 非安全。	0x0
[7:4]	-	reserved	保留。	0x00
[3]	RW	ch3_prot	CH3 通道安全属性： 0: 安全； 1: 非安全。	0x0
[2]	RW	ch2_prot	CH2 通道安全属性： 0: 安全； 1: 非安全。	0x0
[1]	RW	ch1_prot	CH1 通道安全属性： 0: 安全； 1: 非安全。	0x0
[0]	RW	ch0_prot	CH0 通道安全属性： 0: 安全；	0x0



Bits	Access	Name	Description	Reset
			1: 非安全。	

CMUX_SEL

CMUX_SEL 为 CMUX 通路选择寄存器。

Offset Address: 0x0050 Total Reset Value: 0x0054_3210

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0054
[15:12]	RW	chn3_sel	CHN3 选择配置寄存器 0x0: pt0 输入数据; 0x1: pt1 输入数据; 其他: 保留。	0x3
[11:8]	RW	chn2_sel	CHN2 选择配置寄存器 0x0: pt0 输入数据; 0x1: pt1 输入数据; 其他: 保留。	0x2
[7:4]	RW	chn1_sel	CHN1 选择配置寄存器 0x0: pt0 输入数据; 0x1: pt1 输入数据; 其他: 保留。	0x1
[3:0]	RW	chn0_sel	CHN0 选择配置寄存器 0x0: pt0 输入数据; 0x1: pt1 输入数据; 其他: 保留。	0x0

CMUX_ID_CFG

CMUX_ID_CFG 为 CMUX 帧 ID 选择寄存器。

Offset Address: 0x0054 Total Reset Value: 0x00FF_FFFF

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00



Bits	Access	Name	Description	Reset
[23:16]	-	reserved	保留。	0xFF
[15:12]	RW	chn3_id	CHN3 ID 选择寄存器。 0: 过滤该 ID 的数据; 1: 接收该 ID 的数据。 bit[0]: 通道接收 ID 为 0 的数据使能; bit[1]: 通道接收 ID 为 1 的数据使能; bit[2]: 通道接收 ID 为 2 的数据使能; bit[3]: 通道接收 ID 为 3 的数据使能。	0xF
[11:8]	RW	chn2_id	CHN2 ID 选择寄存器。 0: 过滤该 ID 的数据; 1: 接收该 ID 的数据。 bit[0]: 通道接收 ID 为 0 的数据使能; bit[1]: 通道接收 ID 为 1 的数据使能; bit[2]: 通道接收 ID 为 2 的数据使能; bit[3]: 通道接收 ID 为 3 的数据使能。	0xF
[7:4]	RW	chn1_id	CHN1 ID 选择寄存器。 0: 过滤该 ID 的数据; 1: 接收该 ID 的数据。 bit[0]: 通道接收 ID 为 0 的数据使能; bit[1]: 通道接收 ID 为 1 的数据使能; bit[2]: 通道接收 ID 为 2 的数据使能; bit[3]: 通道接收 ID 为 3 的数据使能。	0xF
[3:0]	RW	chn0_id	CHN0 ID 选择寄存器。 0: 过滤该 ID 的数据; 1: 接收该 ID 的数据。 bit[0]: 通道接收 ID 为 0 的数据使能; bit[1]: 通道接收 ID 为 1 的数据使能; bit[2]: 通道接收 ID 为 2 的数据使能; bit[3]: 通道接收 ID 为 3 的数据使能。	0xF

ISP_MODE

ISP_MODE 为 ISP 模式选择寄存器。

Offset Address: 0x0058 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19]	RW	isp3_in_yuv_mode	ISP3 模式选择配置寄存器 0: 输入 RAW; 1: 输入 YUV。	0x0
[18]	RW	isp2_in_yuv_mode	ISP2 模式选择配置寄存器 0: 输入 RAW; 1: 输入 YUV。	0x0
[17]	RW	isp1_in_yuv_mode	ISP1 模式选择配置寄存器 0: 输入 RAW; 1: 输入 YUV。	0x0
[16]	RW	isp0_in_yuv_mode	ISP0 模式选择配置寄存器 0: 输入 RAW; 1: 输入 YUV。	0x0
[15:4]	-	reserved	保留。	0x000
[3]	RW	isp3_mode	ISP3 模式选择配置寄存器 0: 数据经过 ISP 处理; 1: BYPASS ISP。	0x0
[2]	RW	isp2_mode	ISP2 模式选择配置寄存器 0: 数据经过 ISP 处理; 1: BYPASS ISP。	0x0
[1]	RW	isp1_mode	ISP1 模式选择配置寄存器 0: 数据经过 ISP 处理; 1: BYPASS ISP。	0x0
[0]	RW	isp0_mode	ISP0 模式选择配置寄存器 0: 数据经过 ISP 处理; 1: BYPASS ISP。	0x0

VICAP_INT

VICAP_INT 为中断指示寄存器。

Offset Address: 0x00F0 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x00
[26]	RO	int_dbg	系统总线异常中断指示。 0: 无中断; 1: 有中断。	0x0
[25:20]	-	reserved	保留。	0x00
[19]	RO	int_isp3	ISP3 中断指示。 0: 无中断; 1: 有中断。	0x0
[18]	RO	int_isp2	ISP2 中断指示。 0: 无中断; 1: 有中断。	0x0
[17]	RO	int_isp1	ISP1 中断指示。 0: 无中断; 1: 有中断。	0x0
[16]	RO	int_isp0	ISP0 中断指示。 0: 无中断; 1: 有中断。	0x0
[15:12]	-	reserved	保留。	0x00
[11]	RO	int_ch3	通道 3 中断指示。 0: 无中断; 1: 有中断。	0x0
[10]	RO	int_ch2	通道 2 中断指示。 0: 无中断; 1: 有中断。	0x0
[9]	RO	int_ch1	通道 1 中断指示。 0: 无中断; 1: 有中断。	0x0
[8]	RO	int_ch0	通道 0 中断指示。 0: 无中断; 1: 有中断。	0x0
[7:2]	-	reserved	保留。	0x00
[1]	RO	int_pt1	端口 1 中断指示。 0: 无中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	
[0]	RO	int_pt0	端口 0 中断指示。 0: 无中断; 1: 有中断。	0x0

VICAP_INT_MASK

VICAP_INT_MASK 为中断屏蔽寄存器。

Offset Address: 0x00F8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x00
[26]	RW	mask_dbg	系统总线异常中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[25:20]	-	reserved	保留。	0x0
[19]	RW	mask_isp3	ISP3 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[18]	RW	mask_isp2	ISP2 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[17]	RW	mask_isp1	ISP1 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[16]	RW	mask_isp0	ISP0 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[15:12]	-	reserved	保留。	0x00
[11]	RW	mask_ch3	通道 3 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[10]	RW	mask_ch2	通道 2 中断使能。	0x0



Bits	Access	Name	Description	Reset
			0: 屏蔽中断; 1: 使能中断。	
[9]	RW	mask_ch1	通道 1 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[8]	RW	mask_ch0	通道 0 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[7:2]	-	reserved	保留。	0x00
[1]	RW	mask_pt1	端口 1 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	mask_pt0	端口 0 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0

SLAVE0_MODE_CFG

SLAVE0_MODE_CFG 为 SENSOR0 从模式配置寄存器。

Offset Address: 0x0100 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	slave0_vs_enable	VS 输出使能。 0: 关闭; 1: 使能。	0x0
[30]	RW	slave0_hs_enable	HS 输出使能。 0: 关闭; 1: 使能。	0x0
[29]	RW	slave0_vs_out_enabl e	VS_OUT 输出使能。 0: 关闭; 1: 使能。	0x0
[28:18]	-	reserved	保留。	0x000
[17]	RW	slave0_vs_inv	VS 输出极性。 0: 高有效;	0x0



Bits	Access	Name	Description	Reset
			1: 低有效。	
[16]	RW	slave0_hs_inv	HS 输出极性。 0: 高有效; 1: 低有效。	0x0
[15]	RW	slave0_vs_out_inv	VS_OUT 输出极性。 0: 高有效; 1: 低有效。	0x0
[14]	RW	slave0_vs_mux	VS/Vs_OUT 输出信号切换控制。 0: VS/Vs_OUT 输出保持不变; 1: VS/Vs_OUT 输出切换。	0x0
[13:1]	-	reserved	保留。	0x0000
[0]	RW	slave0_update_mode	更新模式选择。 0: 立即更新模式; 1: 帧更新模式。	0x0

SLAVE0_MODE_VS_TIME_H

SLAVE0_MODE_VS_TIME_H 为 SENSOR0 从模式输出 VSYNC 的周期配置寄存器。

Offset Address: 0x0104 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave0_vs_time_h	VS 输出周期配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE0_MODE_VS_TIME_L

SLAVE0_MODE_VS_TIME_L 为 SENSOR0 从模式输出 VSYNC 的周期配置寄存器。

Offset Address: 0x0108 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave0_vs_time_l	VS 输出周期配置(单位: SENSOR 工作时钟周期)。	0x00000000



SLAVE0_MODE_HS_TIME

SLAVE0_MODE_HS_TIME 为 SENSOR0 从模式输出 HSYNC 的周期配置寄存器。

Offset Address: 0x010C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave0_hs_time	HS 输出周期配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE0_MODE_VS_OUT_TIME_H

SLAVE0_MODE_VS_OUT_TIME_H 为 SENSOR0 从模式输出 VS_OUT 的周期配置寄存器。

Offset Address: 0x0110 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave0_vs_out_time_h	VS_OUT 输出周期配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE0_MODE_VS_OUT_TIME_L

SLAVE0_MODE_VS_OUT_TIME_L 为 SENSOR0 从模式输出 VS_OUT 的周期配置寄存器。

Offset Address: 0x0114 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave0_vs_out_time_l	VS_OUT 输出周期配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE0_MODE_VS_CYC_H

SLAVE0_MODE_VS_CYC_H 为 SENSOR0 从模式输出 VSYNC 脉冲有效信号持续时间配置寄存器。

Offset Address: 0x0118 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave0_vs_cyc_h	VS 输出有效信号持续时间配置(单位: SENSOR 工作时钟周期)。	0x00000000



SLAVE0_MODE_VS_CYC_L

SLAVE0_MODE_VS_CYC_L 为 SENSOR0 从模式输出 VSYNC 脉冲有效信号持续时间配置寄存器。

Offset Address: 0x011C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave0_vs_cyc_l	VS 输出有效信号持续时间配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE0_MODE_HS_CYC

SLAVE0_MODE_HS_CYC 为 SENSOR0 从模式输出 HSYNC 脉冲有效信号持续时间配置寄存器。

Offset Address: 0x0120 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave0_hs_cyc	HS 输出有效信号持续时间配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE0_MODE_VS_OUT_CYC_H

SLAVE0_MODE_VS_OUT_CYC_H 为 SENSOR0 从模式输出 VS_OUT 脉冲有效信号持续时间配置寄存器。

Offset Address: 0x0124 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave0_vs_out_cyc_h	VS_OUT 输出有效信号持续时间配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE0_MODE_VS_OUT_CYC_L

SLAVE0_MODE_VS_OUT_CYC_L 为 SENSOR0 从模式输出 VS_OUT 脉冲有效信号持续时间配置寄存器。

Offset Address: 0x0128 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave0_vs_out_cyc_l	VS_OUT 输出有效信号持续时间配置(单位: SENSOR 工作时钟周期)。	0x00000000



SLAVE0_MODE_HS_DLY_CYC

SLAVE0_MODE_HS_DLY_CYC 为 SENSOR0 从模式输出 HSYNC 脉冲输出相对 VSYNC 脉冲的延迟周期配置

Offset Address: 0x012C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave0_hs_dly_cyc	HSYNC 脉冲输出相对 VSYNC 脉冲的延迟周期配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE0_MODE_VS_OUT_DLY_CYC_H

SLAVE0_MODE_VS_OUT_DLY_CYC_H 为 SENSOR0 从模式输出 VS_OUT 脉冲输出相对 VSYNC 脉冲的延迟周期配置

Offset Address: 0x0130 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave0_vs_out_dly_cyc_h	VS_OUT 脉冲输出相对 VSYNC 脉冲的延迟周期配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE0_MODE_VS_OUT_DLY_CYC_L

SLAVE0_MODE_VS_OUT_DLY_CYC_L 为 SENSOR0 从模式输出 VS_OUT 脉冲输出相对 VSYNC 脉冲的延迟周期配置

Offset Address: 0x0134 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave0_vs_out_dly_cyc_l	VS_OUT 脉冲输出相对 VSYNC 脉冲的延迟周期配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE_MODE_SYN

SLAVE_MODE_SYN 为 SENSOR 同步信号, 上升沿有效。

Offset Address: 0x01FC Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	sensor1_sync	SENSOR1 同步信号。	0x0
[0]	RW	sensor0_sync	SENSOR0 同步信号。	0x0

ONLINE0_CFG

ONLINE0_CFG 为在线通道 0 控制寄存器。

Offset Address: 0x0200 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31]	RW	online0_en	在线输出使能 0: 关闭; 1: 开启。	0x0
[30:2]	-	reserved	保留。	0x00000000
[1]	RW	online0_vsync_mode	vsync 是否进行延时处理。 0: 不处理; 1: 处理。	0x1
[0]	RW	online0_hsync_mode	hsync 是否进行延时处理。 0: 不处理; 1: 处理。	0x1

ONLINE0_SIZE

ONLINE0_SIZE 为 ONLINE0 输出有效区大小寄存器。

Offset Address: 0x02AC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	online0_height	图像有效区高度。	0x0000
[15:0]	RO	online0_width	图像有效区宽度。	0x0000

VICAP_AXI0_WR_ERROR

VICAP_AXI0_WR_ERROR 为 AXI0 写错误状态寄存器。

Offset Address: 0x0FE4 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25]	WC	isp3_stt_bus_err	ISP3 STT 通道非法访问错误，写 1 清 0。	0x0
[24]	WC	isp2_stt_bus_err	ISP2 STT 通道非法访问错误，写 1 清 0。	0x0
[23]	WC	isp1_stt_bus_err	ISP1 STT 通道非法访问错误，写 1 清 0。	0x0
[22]	WC	isp0_stt_bus_err	ISP0 STT 通道非法访问错误，写 1 清 0。	0x0
[21:7]	-	reserved	保留。	0x0
[6]	WC	ch3_y_bus_err	CH3_Y 通道非法访问错误，写 1 清 0。	0x0
[5]	WC	ch2_c_bus_err	CH2_C 通道非法访问错误，写 1 清 0。	0x0
[4]	WC	ch2_y_bus_err	CH2_Y 通道非法访问错误，写 1 清 0。	0x0
[3]	-	reserved	保留。	0x0
[2]	WC	ch1_y_bus_err	CH1_Y 通道非法访问错误，写 1 清 0。	0x0
[1]	-	reserved	保留。	0x0
[0]	WC	ch0_y_bus_err	CH0_Y 通道非法访问错误，写 1 清 0。	0x0

VICAP_DBG_INT

VICAP_DBG_INT 为系统异常中断状态寄存器。

Offset Address: 0x0FF0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	WC	apb_bus_err	APB 非法访问错误中断状态，写 1 清零。 0: 无中断; 1: 有中断。	0x0
[0]	WC	axi_bus_err	AXI 非法访问错误中断状态，写 1 清零。 0: 无中断; 1: 有中断。	0x0

VICAP_DBG_INT_MASK

VICAP_DBG_INT_MASK 为系统异常状态屏蔽寄存器。

Offset Address: 0x0FF8 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	apb_bus_err_mask	APB 非法访问错误中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	axi_bus_err_mask	AXI 非法访问错误中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0

PT_INTF_MOD

PT_INTF_MOD 为接口模式寄存器。

Offset Address: $0x1000 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	pt_enable	端口使能。 0: 关闭; 1: 使能。	0x0
[30:1]	-	reserved	保留。	0x00000000
[0]	RW	pt_mode	时序模式配置。 0: 外同步; 1: 内同步。	0x0

PT_OFFSET0

PT_OFFSET0 为 0 分量 offset 寄存器。

Offset Address: $0x1010 + PT\_N \times 0x100$ Total Reset Value: 0xFFFF0_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	offset0_mask	0 分量 mask。	0xFFFF0
[15]	RW	offset0_rev	数据线是否翻转。 0: 不反转; 1: 反转。	0x0
[14:6]	-	reserved	保留。	0x000



[5:0]	RW	offset0	0 分量偏移。	0x00

PT_OFFSET1

PT_OFFSET1 为 1 分量 offset 寄存器。

Offset Address: $0x1014 + PT\_N \times 0x100$ Total Reset Value: 0xFFF0_0010

Bits	Access	Name	Description	Reset
[31:16]	RW	offset1_mask	1 分量 mask。	0xFFF0
[15]	RW	offset1_rev	数据线是否翻转。 0: 不反转; 1: 反转。	0x0
[14:6]	-	reserved	保留。	0x000
[5:0]	RW	offset1	1 分量偏移。	0x10

PT_OFFSET2

PT_OFFSET2 为 2 分量 offset 寄存器。

Offset Address: $0x1018 + PT\_N \times 0x100$ Total Reset Value: 0xFFF0_0020

Bits	Access	Name	Description	Reset
[31:16]	RW	offset2_mask	2 分量 mask。	0xFFF0
[15]	RW	offset2_rev	数据线是否翻转。 0: 不反转; 1: 反转。	0x0
[14:6]	-	reserved	保留。	0x000
[5:0]	RW	offset2	2 分量偏移。	0x20

PT_BT656_CFG

PT_BT656_CFG 为 BT656 配置寄存器。

Offset Address: $0x1020 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0303

Bits	Access	Name	Description	Reset
[31]	RW	bt656_enable	BT656 使能寄存器。	0x0



Bits	Access	Name	Description	Reset
			0: 关闭; 1: 使能。	
[30:11]	-	reserved	保留。	0x00000
[10]	RW	bt656_field_inv	Field 反向控制。 0: 不反向; 1: 反向。	0x0
[9]	RW	bt656_vsync_inv	VS 反向控制。 0: 不反向; 1: 反向。	0x1
[8]	RW	bt656_hsync_inv	HS 反向控制。 0: 不反向; 1: 反向。	0x1
[7:4]	-	reserved	保留。	0x0
[3:0]	RW	bt656_mode	模式选择寄存器。 bt656_mode[0] 0: hsync 不作为有效信号; 1: hsync 作为有效信号。 bt656_mode[1] 0: hsync 输出低有效; 1: hsync 输出高有效。 bt656_mode[3:2] 00: 解析 0 分量; 01: 解析 1 分量; 10: 解析 2 分量; 11: 保留。	0x3

PT_UNIFY_TIMING_CFG

PT_UNIFY_TIMING_CFG 为时序配置寄存器。

Offset Address: $0x1038 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x00
[26]	RW	unify_field_inv	field 是否反相(field 第 1 级处理)。	0x0



Bits	Access	Name	Description	Reset
			0: 不反相; 1: 反相。	
[25:24]	RW	unify_field_sel	field 来源选择 (field 第 0 级处理)。 00: 输入 field; 01: 输入 vsync; 10: 跟据 vsync 和 hsync 的关系检测; 11: 恒为 0。	0x0
[23:21]	-	reserved	保留。	0x0
[20:19]	RW	unify_vsync_mode	vsync 处理模式(vsync 第 2 级处理)。 00: 不处理; 01: 取上升沿; 10: 取上升沿和下降沿; 11: 保留。	0x0
[18]	RW	unify_vsync_inv	vsync 是否反相(vsync 第 1 级处理)。 0: 不反相; 1: 反相。	0x0
[17:16]	RW	unify_vsync_sel	vsync 来源选择 (vsync 第 0 级处理) 00: 输入 vsync; 01: 输入 field; 10: 恒为 0; 11: 保留。	0x0
[15]	-	reserved	保留。	0x0
[14:13]	RW	unify_hsync_mode	hsync 处理模式(hsync 第 3 级处理)。 0: 不处理; 1: 取上升沿。	0x0
[12:11]	RW	unify_hsync_and	hsync 是否与 vsync 第 1 级处理的结果进行运算(hsync 第 2 级处理)。 00: 不处理; 01: 相与; 10: 异或; 11: 保留。	0x0
[10]	RW	unify_hsync_inv	hsync 是否反相(hsync 第 1 级处理)。 0: 不反相; 1: 反相。	0x0



Bits	Access	Name	Description	Reset
[9:8]	RW	unify_hsync_sel	hsync 来源选择 (hsync 第 0 级处理)。 00: 输入 hsync; 01: 输入 de; 10: 恒为 0; 11: 保留。	0x0
[7:3]	-	reserved	保留。	0x00
[2]	RW	unify_de_inv	de 是否反相 (de 第 1 级处理)。 0: 不反相; 1: 反相。	0x0
[1:0]	RW	unify_de_sel	de 来源选择 (de 第 0 级处理)。 00: 输入 de; 01: hsync 第 2 级处理的结果; 10: 恒为 1; 11: 恒为 0。	0x0

PT_GEN_TIMING_CFG

PT_GEN_TIMING_CFG 为时序恢复模块配置寄存器。

Offset Address: $0x103C + PT\_N \times 0x100$ Total Reset Value: 0x4000_0006

Bits	Access	Name	Description	Reset
[31]	RW	gen_enable	时序恢复使能，根据时序参数恢复时序。 0: 关闭; 1: 使能。	0x0
[30]	RW	gen_mode	时序恢复模式，根据时序参数恢复时序。 0: 依赖 PT 输入的 Valid 信号来产生时序; 1: 内部自动计算产生时序。	0x1
[29:3]	-	reserved	保留。	0x0000000
[2]	RW	gen_vsync_mode	是否恢复 vsync。 0: 不恢复; 1: 恢复。	0x1



[1]	RW	gen_hsync_mode	是否恢复 hsync。 0: 不恢复; 1: 恢复。	0x1
[0]	-	reserved	保留。	0x0

PT_UNIFY_DATA_CFG

PT_UNIFY_DATA_CFG 为数据配置寄存器。

Offset Address: $0x1040 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	unify_data_enable	数据分离使能。 0: 关闭; 1: 使能。	0x0
[30:4]	-	reserved	保留。	0x00000000
[3]	RW	unify_data_uv_seq	CbCr 顺序。 0: CbCr; 1: CrCb。	0x0
[2]	RW	unify_data_yc_seq	YC 顺序。 0: CY; 1: YC。	0x0
[1:0]	RW	unify_data_comp_num	数据分量选择。 00: 1 分量; 01: 2 分量; 10: 3 分量; 11: 保留。	0x0

PT_GEN_DATA_CFG

PT_GEN_DATA_CFG 为数据产生模块配置寄存器。

Offset Address: $0x1044 + PT\_N \times 0x100$ Total Reset Value: 0x0000_00E9

Bits	Access	Name	Description	Reset
[31]	RW	gen_data_enable	数据产生使能，根据数据产生参数产生数据。	0x0



Bits	Access	Name	Description	Reset
			0: 关闭; 1: 使能。	
[30:8]	-	reserved	保留。	0x000000
[7]	RW	gen_data0_move	data0 是否递增。 0: 不递增; 1: 递增。	0x1
[6]	RW	gen_data1_move	data1 是否递增。 0: 不递增; 1: 递增。	0x1
[5]	RW	gen_data2_move	data2 是否递增。 0: 不递增; 1: 递增。	0x1
[4]	RW	gen_data_vsync_reset	数据是否根据 vsync 信号复位。 0: 不复位; 1: 复位。	0x0
[3]	RW	gen_data_hsync_reset	数据是否根据 hsync 信号复位。 0: 不复位; 1: 复位。	0x1
[2]	RW	gen_data_vsync_move	数据是否根据 vsync 递增。 0: 不递增; 1: 递增。	0x0
[1]	RW	gen_data_hsync_move	数据是否根据 hsync 递增。 0: 不递增; 1: 递增。	0x0
[0]	RW	gen_data_de_move	数据是否根据 de 信号递增。 0: 不递增; 1: 递增。	0x1

PT_GEN_DATA_COEF

PT_GEN_DATA_COEF 为数据产生模块系数寄存器。

Offset Address: $0x1048 + PT\_N \times 0x100$ Total Reset Value: 0x0100_0100



Bits	Access	Name	Description	Reset
[31:24]	RW	inc_frame	数据帧间递增值，递增值累加在数据的高 8bit。	0x01
[23:16]	RW	step_frame	数据帧间递增的间隔。配置值为实际值减 1，配为 0 表示每帧递增。	0x00
[15:8]	RW	inc_space	数据像素间递增值，递增值累加在数据的高 10bit。	0x01
[7:0]	RW	step_space	数据像素间递增的间隔，为 0 是每像素递增。	0x00

PT_GEN_DATA_INIT

PT_GEN_DATA_INIT 为数据产生初始值配置寄存器。

Offset Address: $0x104C + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:16]	RW	data2	V/B 的初始值。	0x00
[15:8]	RW	data1	U/G 的初始值。	0x00
[7:0]	RW	data0	Y/R 的初始值。	0x00

PT_YUV444_CFG

PT_YUV444_CFG 为 YUV444 配置寄存器。

Offset Address: $0x1050 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	yuv44_enable	YUV 使能，将 YUV422 信号调整为 YUV444 信号。 0: 关闭; 1: 使能。	0x0
[30:0]	-	reserved	保留。	0x00000000



PT_FSTART_DLY

PT_FSTART_DLY 为端口 fstart 中断延时寄存器。

Offset Address: $0x1060 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	pt_fstart_dly	fstart 中断延时时间，以端口时钟为单位。	0x00000000

PT_FSTART_H_DLY

PT_FSTART_H_DLY 为端口 fstart 中断行延时寄存器。

Offset Address: $0x1064 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	fstart_h_dly	fstart 中断行延时时间，以有效区行号为单位。	0x0000
[15:4]	RW	reserved	保留。	0x000
[3:0]	RW	fstart_h_dly_id	计数帧 ID 选择寄存器。 bit[0]: 通过 ID 为 0 的时序触发计数； bit[1]: 通过 ID 为 1 的时序触发计数； bit[2]: 通过 ID 为 2 的时序触发计数； bit[3]: 通过 ID 为 3 的时序触发计数。	0x0

PT_INTF_HFB

PT_INTF_HFB 为水平前消隐区宽度寄存器。

Offset Address: $0x1080 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0058

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	hfb	水平前消隐区宽度。	0x0058

PT_INTF_HACT

PT_INTF_HACT 为水平有效区宽度寄存器。

Offset Address: $0x1084 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0780



Bits	Access	Name	Description	Reset
[31:0]	RW	hact	水平有效区宽度(单位: 时钟周期)。	0x00000780

PT_INTF_HBB

PT_INTF_HBB 为水平后消隐区宽度寄存器。

Offset Address: $0x1088 + PT\_N \times 0x100$ Total Reset Value: 0x0000_00C0

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	hbb	水平后消隐区宽度。	0x00C0

PT_INTF_VFB

PT_INTF_VFB 为垂直 前消隐区宽度寄存器。

Offset Address: $0x108C + PT\_N \times 0x100$ Total Reset Value: 0x0000_0004

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	vfb	垂直前消隐区宽度。	0x0004

PT_INTF_VACT

PT_INTF_VACT 为垂直有效区宽度寄存器。

Offset Address: $0x1090 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0438

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	vact	垂直有效区宽度。	0x0438

PT_INTF_VBB

PT_INTF_VBB 为垂直后消隐区宽度寄存器。

Offset Address: $0x1094 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0029



Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	vbb	垂直后消隐区宽度。	0x0029

PT_INTF_VBFB

PT_INTF_VBFB 为垂直底场前消隐区宽度寄存器。

Offset Address: $0x1098 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0004

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	vbfb	垂直底场前消隐区宽度。	0x0004

PT_INTF_VBACT

PT_INTF_VBACT 为垂直底场有效区宽度寄存器。

Offset Address: $0x109C + PT\_N \times 0x100$ Total Reset Value: 0x0000_0438

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	vbact	垂直底场有效区宽度。	0x0438

PT_INTF_VBBB

PT_INTF_VBBB 为垂直底场后消隐区宽度寄存器。

Offset Address: $0x10A0 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0029

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	vbbb	垂直底场后消隐区宽度。	0x0029

PT_ID_CFG

PT_ID_CFG 为 ID 配置寄存器。

Offset Address: $0x10A4 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31]	RW	id_enable	生成 ID 功能使能。 0: 不使能; 1: 使能。	0x0
[30]	RW	id_mode	生成 ID 模式选择。 0: 自动模式, ID 每帧自动增加; 1: 非自动模式, ID 由配置值决定。	0x0
[29]	WO	id_reset	自动模式时, 配置寄存器为 1 可使下一帧的 ID 恢复初始值, 恢复后自动清 0。	0x0
[28:4]	-	reserved	保留。	0x000000
[3:2]	RW	id_max	自动模式时为 ID 增加的时最大值。	0x0
[1:0]	RW	id	自动模式时为 ID 初始值, 非自动模式时为生成的 ID 值。	0x0

PT_FLASH_CFG

PT_FLASH_CFG 为闪光灯配置寄存器。

Offset Address: 0x10C0+PT_N×0x100 Total Reset Value: 0x1000_0000

Bits	Access	Name	Description	Reset
[31:28]	RW	id_sel	触发帧 ID 选择寄存器。 0001: 通过 ID 为 0 的帧起始中断触发; 0010: 通过 ID 为 1 的帧起始中断触发; 0100: 通过 ID 为 2 的帧起始中断触发; 1000: 通过 ID 为 3 的帧起始中断触发。 其他: 保留。	0x1
[27:19]	-	reserved	保留。	0x000
[18]	RW	shutter_times	快门脉冲次数。 0: 2 次; 1: 1 次。	0x0
[17]	RW	shutter_phase	快门信号是否反向。 0: 不反向; 1: 反向。	0x0



[16]	WO	shutter_en	触发快门控制，自动清零。 0：关闭； 1：使能。	0x0
[15:2]	-	reserved	保留。	0x0000
[1]	RW	flash_phase	闪光灯信号是否反向。 0：不反向； 1：反向。	0x0
[0]	WO	flash_en	触发闪光灯控制，自动清零。 0：关闭； 1：使能。	0x0

PT_FLASH_CYC0

PT_FLASH_CYC0 为闪光灯时序 0 宽度寄存器。

Offset Address: $0x10C4 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	pt_flash_cyc0	闪光灯时序 0 的时间，以端口时钟为单位，配置值为实际值减 1。	0x00000000

PT_FLASH_CYC1

PT_FLASH_CYC1 为闪光灯时序 1 宽度寄存器。

Offset Address: $0x10C8 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	pt_flash_cyc1	闪光灯时序 1 的时间，以端口时钟为单位，配置值为实际值减 1。	0x00000000

PT_SHUTTER_CYC0

PT_SHUTTER_CYC0 为快门时序 0 宽度寄存器。

Offset Address: $0x10D0 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	pt_shutter_cyc0	快门时序 0 的时间，以端口时钟为单位，配置值为实际值减 1。	0x00000000

PT_SHUTTER_CYC1

PT_SHUTTER_CYC1 为快门时序 1 宽度寄存器。

Offset Address: $0x10D4 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	pt_shutter_cyc1	快门时序 1 的时间，以端口时钟为单位，配置值为实际值减 1。	0x00000000

PT_SHUTTER_CYC2

PT_SHUTTER_CYC2 为快门时序 2 宽度寄存器。

Offset Address: $0x10D8 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	pt_shutter_cyc2	快门时序 2 的时间，以端口时钟为单位，配置值为实际值减 1。	0x00000000

PT_SHUTTER_CYC3

PT_SHUTTER_CYC3 为快门时序 3 宽度寄存器。

Offset Address: $0x10DC + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	pt_shutter_cyc3	快门时序 3 的时间，以端口时钟为单位，配置值为实际值减 1。	0x00000000

PT_STATUS

PT_STATUS 为端口状态寄存器。

Offset Address: $0x10E0 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5:4]	RO	pt_status_id	端口输出的 id。	0x0
[3]	RO	pt_status_field	端口输出的 field。	0x0
[2]	RO	pt_status_vsync	端口输出的 vsync。	0x0
[1]	RO	pt_status_hsync	端口输出的 hsync。	0x0
[0]	RO	pt_status_de	端口输出的 de。	0x0

PT_BT656_STATUS

PT_BT656_STATUS 为 BT656 状态寄存器。

Offset Address: $0x10E4 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:8]	RO	pt_bt656_seav	同步码。	0x00
[7:0]	-	reserved	保留。	0x00

PT_SIZE

PT_SIZE 为输入图像有效区大小指示寄存器。

Offset Address: $0x10EC + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	pt_height	图像有效区高度。	0x0000
[15:0]	RO	pt_width	图像有效区宽度。	0x0000

PT_INT

PT_INT 为端口中断指示寄存器。

Offset Address: $0x10F0 + PT\_N \times 0x100$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000



[3]	WC	pt_fstart_h_dly	帧/场起始行延时中断状态，写 1 清零。 0: 无中断; 1: 有中断。	0x0
[2]	WC	pt_height_err	图像高度变化中断状态，写 1 清零。 0: 无中断; 1: 有中断。	0x0
[1]	WC	pt_width_err	图像宽度变化中断状态，写 1 清零。 0: 无中断; 1: 有中断。	0x0
[0]	WC	pt_fstart	帧/场起始中断状态，写 1 清零。 0: 无中断; 1: 有中断。	0x0

PT_INT_MASK

PT_INT_MASK 为端口中断屏蔽寄存器。

Offset Address: 0x10F8+PT_N×0x100 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RW	pt_mask_fstart_h_dly	帧/场起始行延时中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RW	pt_mask_height_err	图像高度变化中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	RW	pt_mask_width_err	图像宽度变化中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	pt_mask_fstart	帧/场起始中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0



CH_CTRL

CH_CTRL 为通道控制寄存器。

Offset Address: $0x10000 + CH\_N \times 0x1000$ Total Reset Value: 0x0002_0000

Bits	Access	Name	Description	Reset
[31]	RW	ch_enable	通道使能寄存器。 0: 不使能; 1: 使能。	0x0
[30]	RW	ch_reg_newer_mode	使能模式控制寄存器。 0: reg_new 有效且模块使能时输出; 1: 模块使能时输出。	0x0
[29:18]	-	reserved	保留。	0x000
[17]	RW	ch_finish_mode	完成中断上报时间点控制寄存器。 0: 完成中断立即上报; 1: 完成中断在下帧的帧起始上报。	0x1
[16:0]	-	reserved	保留。	0x00000

CH_REG_NEWER

CH_REG_NEWER 为通道更新寄存器。

Offset Address: $0x10004 + CH\_N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	ch_reg_newer	通道更新寄存器, 每帧自动清 0。	0x0

CH_DLY_CFG

CH_DLY_CFG 为通道输入图像起始中断延迟配置寄存器。

Offset Address: $0x10034 + CH\_N \times 0x1000$ Total Reset Value: 0x0010_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	ch_v_dly_cfg	垂直行数 delay 寄存器。	0x0010
[15:0]	-	reserved	保留。	0x0000



CH_CROP_CFG

CH_CROP_CFG 为通道 CROP 使能寄存器。

Offset Address: $0x10040 + CH\_N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	ch_crop_n0_en	0 区域使能。 0: 禁止; 1: 使能。	0x0

CH_CROP0_START

CH_CROP0_START 为通道 0 区域 CROP 起始位置寄存器。

Offset Address: $0x10044 + CH\_N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	ch_crop_y_start	开始获取图像的行号。	0x0000
[15:0]	RW	ch_crop_x_start	开始获取图像的像素号。	0x0000

CH_CROP0_SIZE

CH_CROP0_SIZE 为通道 0 区域 CROP 大小置寄存器。

Offset Address: $0x10048 + CH\_N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	ch_crop_height	获取图像的高度(以行为单位), 配置值为实际值减 1。	0x0000
[15:0]	RW	ch_crop_width	获取图像的宽度(以像素为单位), 配置值为实际值减 1。	0x0000

CH_SKIP_Y_CFG

CH_SKIP_Y_CFG 为通道 Y SKIP 配置寄存器。

Offset Address: $0x10050 + CH\_N \times 0x1000$ Total Reset Value: 0xFFFF_FFFF



Bits	Access	Name	Description	Reset
[31:0]	RW	ch_skipy_cfg	SKIP 配置。	0xFFFFFFFF

CH_SKIP_C_CFG

CH_SKIP_C_CFG 为通道 C SKIP 配置寄存器。

Offset Address: $0x10058 + CH\_N \times 0x1000$ Total Reset Value: 0xFFFF_FFFF

Bits	Access	Name	Description	Reset
[31:0]	RW	ch_skipc_cfg	SKIP 配置。	0xFFFFFFFF

CH_WCH_Y_CFG

CH_WCH_Y_CFG 为 Y 分量配置寄存器。

Offset Address: $0x10080 + CH\_N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	wch_y_enable	写通道使能寄存器。 0: 不使能; 1: 使能。	0x0
[30:29]	-	reserved	保留。	0x0
[28:15]	RW	wch_y_wrap_line	写通道卷绕 buffer 行数配置。	0x0000
[14]	RW	wch_y_wrap_en	写通道卷绕 buffer 使能寄存器。 0: 不使能; 1: 使能。	0x0
[13]	RW	wch_y_cmp_mode	写通道压缩选择模式选择寄存器。 0: 帧压缩; 1: 行压缩。	0x0
[12]	RW	wch_y_cmp_en	写通道压缩使能寄存器。 0: 不使能; 1: 使能。	0x0
[11:6]	RW	reserved	保留。	0x00
[5:0]	RW	wch_y_bit_width	数据位宽。 0x08: 8bit;	0x00



Bits	Access	Name	Description	Reset
			0x0A: 10bit; 0x0C: 12bit; 0x0E: 14bit; 0x10: 16bit; 其他: 保留。	

CH_WCH_Y_SIZE

CH_WCH_Y_SIZE 为 Y 分量存储大小寄存器。

Offset Address: $0x10084 + CH\_N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	wch_y_height	存储图像的高度(以行为单位), 配置值为实际值减 1。	0x0000
[15:0]	RW	wch_y_width	存储图像的宽度(以像素为单位), 配置值为实际值减 1。	0x0000

CH_WCH_Y_FADDR_L

CH_WCH_Y_FADDR_L 为 Y 分量存储基地址低 32bit 寄存器。

Offset Address: $0x10094 + CH\_N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	wch_y_faddr_l	Y 分量存储基地址低 32bit。	0x00000000

CH_WCH_Y_STRIDE

CH_WCH_Y_STRIDE 为 Y 分量行偏移寄存器。

Offset Address: $0x100A0 + CH\_N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	wch_y_stride	图像存储 Y 分量的 stride, 以 byte 为单位。	0x0000



CH_WCH_C_CFG

CH_WCH_C_CFG 为 C 分量配置寄存器。

Offset Address: $0x100B0 + CH\_N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	wch_c_enable	写通道使能寄存器。 0: 不使能; 1: 使能。	0x0
[30:6]	-	reserved	保留。	0x0000000
[5:0]	RW	wch_c_bit_width	写通道数据位宽。 0x08: 8bit; 0x0A: 10bit; 0x0C: 12bit; 0x0E: 14bit; 0x10: 16bit; 其他: 保留。	0x00

CH_WCH_C_SIZE

CH_WCH_C_SIZE 为 C 分量存储大小寄存器。

Offset Address: $0x100B4 + CH\_N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	wch_c_height	存储图像的高度(以行为单位), 配置值为实际值减 1。	0x0000
[15:0]	RW	wch_c_width	存储图像的宽度(以像素为单位), 配置值为实际值减 1。	0x0000

CH_WCH_C_FADDR_L

CH_WCH_C_FADDR_L 为 C 分量存储基地址低 32bit 寄存器。

Offset Address: $0x100C4 + CH\_N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	wch_c_faddr_l	C 分量存储基地址低 32bit。	0x00000000



CH_WCH_C_STRIDE

CH_WCH_C_STRIDE 为 C 分量行偏移寄存器。

Offset Address: $0x100D0 + CH\_N \times 0x1000$ Total Reset Value: $0x0000\_0000$

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	wch_c_stride	图像存储 C 分量的 stride，以 byte 为单位。	0x0000

CH_Y_ACT_SIZE

CH_Y_ACT_SIZE 为亮度通道输出图像有效区宽高指示寄存器。

Offset Address: $0x100E8 + CH\_N \times 0x1000$ Total Reset Value: $0x0000\_0000$

Bits	Access	Name	Description	Reset
[31:16]	RO	ch_y_height_act	亮度有效区高度。	0x0000
[15:0]	RO	ch_y_width_act	亮度有效区宽度。	0x0000

CH_C_ACT_SIZE

CH_C_ACT_SIZE 为色度通道输出图像有效区宽高指示寄存器。

Offset Address: $0x100EC + CH\_N \times 0x1000$ Total Reset Value: $0x0000\_0000$

Bits	Access	Name	Description	Reset
[31:16]	RO	ch_c_height_act	色度有效区高度。	0x0000
[15:0]	RO	ch_c_width_act	色度有效区宽度。	0x0000

CH_INT

CH_INT 为通道原始中断寄存器。

Offset Address: $0x100F0 + CH\_N \times 0x1000$ Total Reset Value: $0x0000\_0000$

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x000
[16]	WC	ch_stt_ae_finish	AE 统计信息完成中断状态，写 1 清零。	0x0



Bits	Access	Name	Description	Reset
			0: 无中断; 1: 有中断。	
[15]	WC	ch_int_fstart_dly	delay 后的场/帧起始中断状态, 写 1 清零。 0: 无中断; 1: 有中断。	0x0
[14:5]	-	reserved	保留。	0x000
[4]	WC	ch_int_update_cfg	寄存器更新中断状态, 写 1 清零。 0: 无中断; 1: 有中断。	0x0
[3]	WC	ch_int_field_throw	场/帧丢失中断状态, 写 1 清零。 0: 无中断; 1: 有中断。	0x0
[2]	WC	ch_int_buf_ovf	内部 FIFO 溢出错误中断状态, 写 1 清零。 0: 无中断; 1: 有中断。	0x0
[1]	WC	ch_int_cc_int	获取完毕中断状态, 写 1 清零。 0: 无中断; 1: 有中断。	0x0
[0]	WC	ch_int_fstart	场/帧起始中断状态, 写 1 清零。 0: 无中断; 1: 有中断。	0x0

CH_INT_MASK

CH_INT_MASK 为通道中断屏蔽寄存器。

Offset Address: 0x100F8+CH_N×0x1000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x000
[16]	RW	mask_stt_ae_finish	AE 统计信息完成中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[15]	RW	mask_fstart_dly	delay 后的场/帧起始中断使能。 0: 屏蔽中断;	0x0



Bits	Access	Name	Description	Reset
			1: 使能中断。	
[14:5]	-	reserved	保留。	0x000
[4]	RW	mask_update_cfg	寄存器更新中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[3]	RW	mask_field_throw	场/帧丢失中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RW	mask_buf_ovf	内部 FIFO 溢出错误中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	RW	mask_cc_int	获取完毕中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	mask_fstart	场/帧起始中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0

CH_WCH_STT_CFG

CH_WCH_STT_CFG 为 ISP 统计信息写出配置寄存器。

Offset Address: $0x10200 + CH\_N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留	0x0
[30]	RW	ch_wch_stt_32b_align	STT 写通道是 32byte 对齐使能。 0: 不使能; 1: 使能。	0x0
[29:0]	-	reserved	保留。	0x00000000

CH_WCH_STT_AE_HIST_ADDR_L

CH_WCH_STT_AE_HIST_ADDR_L 为 ISP AE_HIST 统计信息存储基地址低 32bit 寄存器。



Offset Address: $0x10208 + CH\_N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	ch_wch_stt_ae_hist_addr_l	ISP AE_HIST 统计信息存储基地址低 32bit。 (仅 CH0/CH1/CH2/CH3 支持)	0x00000000

CH_WCH_STT_AE_AVER_R_GR_ADDR_L

CH_WCH_STT_AE_AVER_R_GR_ADDR_L 为 ISP AE_AVER_R_GR 统计信息存储基地址低 32bit 寄存器。

Offset Address: $0x10210 + CH\_N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	ch_wch_stt_ae_aver_r_gr_addr_l	ISP AE_AVER_R_GR 统计信息存储基地址低 32bit。 (仅 CH0/CH1/CH2/CH3 支持)	0x00000000

CH_WCH_STT_AE_AVER_GB_B_ADDR_L

CH_WCH_STT_AE_AVER_GB_B_ADDR_L 为 ISP AE_AVER_GB_B 统计信息存储基地址低 32bit 寄存器。

Offset Address: $0x10218 + CH\_N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	ch_wch_stt_ae_aver_gb_b_addr_l	ISP AE_AVER_GB_B 统计信息存储基地址低 32bit。 (仅 CH0/CH1/CH2/CH3 支持)	0x00000000

CH_WCH_STT_AF_STAT_H1_ADDR_L

CH_WCH_STT_AF_STAT_H1_ADDR_L 为 ISP AF_STAT_H1 统计信息存储基地址低 32bit 寄存器。

Offset Address: $0x10220 + CH\_N \times 0x1000$ Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	ch_wch_stt_af_stat_h1_addr_l	ISP AF_STAT_H1 统计信息存储基地址低 32bit。 (仅 CH0/CH1/CH2/CH3 支持)	0x00000000

CH_WCH_STT_AF_STAT_H2_ADDR_L

CH_WCH_STT_AF_STAT_H2_ADDR_L 为 ISP AF_STAT_H2 统计信息存储基地址低 32bit 寄存器。

Offset Address: 0x10228+CH_N×0x1000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	ch_wch_stt_af_stat_h2_addr_l	ISP AF_STAT_H2 统计信息存储基地址低 32bit。 (仅 CH0/CH1/CH2/CH3 支持)	0x00000000

9.2 VDP

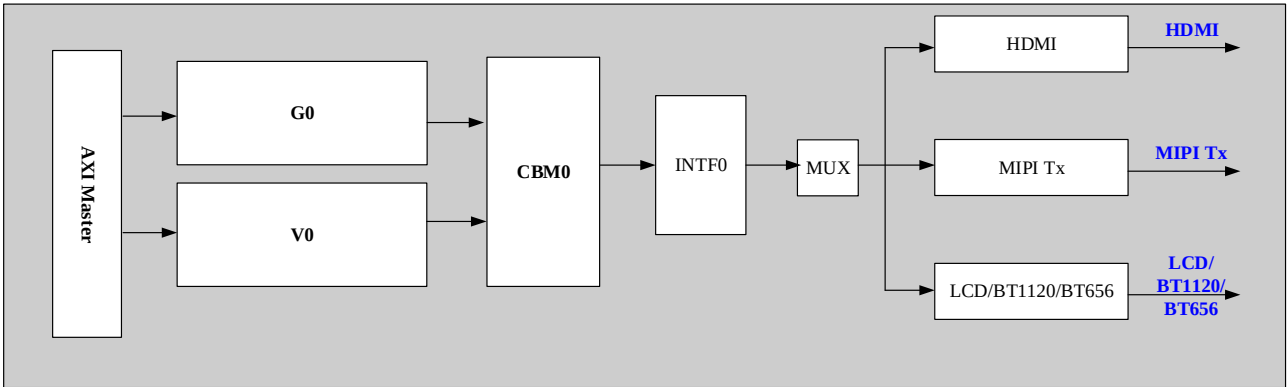
9.2.1 概述

VDP（Video Display Processor）模块主动从内存相应位置读取视频和图形数据，将视频层和图形层数据叠加后通过显示通道送出。

9.2.2 架构描述

VDP 的总体框图如图 9-19 所示。

图9-19 VDP 总体框图





- Surface: 总线输入的数据通路。其功能包括单个图层的总线数据读取、数据处理。Surface 包括: 视频层 V0、图形层 G0。
- Display Channel: 显示通道。包括高清显示通道 DHD0。
- CBM (CrossBar and Mixer, 选通叠加器): 视频层 V0 图形层 G0 叠加 CBM0。
- CAS (Control And Status, 控制和状态): 该模块主要是通过 APB (Advanced Peripheral Bus, 高级外围总线) 总线完成对寄存器的配置, 并且各模块的状态信息通过该模块上报给 CPU。

VDP 寄存器主要按照模块全局、Surface 和 Display Channel 来分配:

- 模块全局寄存器
包括总线相关配置、中断。
- Surface 寄存器
包括视频层、图形层配置寄存器。
- Display Channel 寄存器
包括 DHD0 配置寄存器。

VDP 的特点如下:

- HDMI 1.4 输出接口: 最大支持 1080P@60fps YUV/RGB 输出
- MIPI Tx 输出接口: 最大支持 1080P@60fps 输出显示。
- 数字输出接口: 支持 ITU-R BT.1120 输出, 支持 ITU-R BT.656 输出, 支持 LCD24bit 并行/18bit 并行/16bit 并行/6bit 串行/8bit 串行输出。
- 视频层 (Video Surface): V0 层。
- 图形层 (Graphics Surface): G0 层。
- 叠加特性: V0、G0 叠加。
- VDP 包含 1 个显示通道, 有 1 个垂直时序中断以及 1 个低带宽中断。

9.2.3 工作方式

9.2.3.1 时钟配置

VDP 的可配时钟寄存器有 1 个:

具体寄存器参见 3.2.6 章节 CRG 寄存器: PERI_CRG66。

9.2.3.2 复位

VDP 的复位包括 1 个硬件复位、1 个软件复位。



注意

在进行 AXI 总线复位之前:



- 将所有的层关闭。
- 在下一帧/场中断起来（到达更新点）后，再配置总线复位请求。

9.2.3.3 输出接口

VDP 支持以下三种接口输出：

- HDMI 1.4
- MIPI Tx
- ITU-R BT.1120/ITU-R BT.656/LCD 输出

9.2.3.4 中断

VDP 中断分 2 类：

- 垂直时序中断
- 低带宽中断

垂直时序中断

VDP 支持垂直时序中断，中断产生位置可灵活配置：

- VDP 包含 1 个显示通道，对应有 1 个垂直时序中断。
- 支持中断屏蔽可配。
- 支持垂直时序中断门限可配。
- 每个中断可清除，写 1 清 0。

低带宽中断

VDP 支持以中断方式上报低带宽状态：

- VDP 包含 1 个显示通道，对应有 1 个低带宽中断，标志帧的低带宽信息。
- 支持中断屏蔽可配。
- 每个中断可清除，写 1 清 0。

9.2.4 功能描述

9.2.4.1 视频层功能

视频层 V0 特性

- 支持非压缩、压缩两种数据源。
- 支持的输入像素格式：semi-planar400、semi-planar420、semi-planar422。
- 最小输入分辨率 32x32，最大输入分辨率为 1920x1080。
- 最小输出分辨率 32x32，最大输出分辨率为 1920x1080。
- 支持输入数据位宽：8bit。
- 输入的水平分辨率为 2 的倍数。



- 输入垂直分辨率为 2 的倍数。
- 读取模式与接口绑定。
- 源起始地址可配（分亮度、色度地址），地址 16byte 对齐。
- 源 stride 可配（分亮度、色度 stride），为 16byte 对齐。
- 支持垂直色度上采样功能 CVFIR，滤波系数可配置。
- 支持水平色度上采样 HFIR，复制模式、双线性插值、4 阶半带滤波模式可选。
- 支持色彩空间转换，支持对比度/色调/饱和度调节。
- 支持视频层 BT.601、BT.709 色域转换。
- 支持全局 alpha 可配置，配置范围 0~255。

9.2.4.2 图形层功能

图形层 G0 特性

- 支持的输入像素格式：ARGB1555、ARGB4444、ARGB8888。
- 最小输入分辨率 32x32，ARGB1555 和 ARGB4444 最大输入分辨率为 1920x1080，ARGB8888 最大输入分辨率为 960x540。
- 最小输出分辨率 32x32，最大输出分辨率为 1920x1080。
- 输入水平垂直分辨率为 2 的倍数。
- 读取模式跟接口绑定。
- 源起始地址可配，地址为 16byte 对齐。
- 源 stride 可配，为 16byte 对齐。
- 支持放大功能，水平、垂直范围均为 1x~15x。
- 支持色彩空间转换。
- 支持显示位置可配：在屏幕的任意位置显示。
- 支持全局 alpha 可配置，配置范围 0~255。
- 支持 colorkey 处理。
- 支持预乘格式处理。

9.2.4.3 叠加处理

VDP 只支持 V0、G0 叠加。

叠加特性

- 支持 bottom-to-top 的叠加方式
- 叠加背景色可配
- 叠加 surface 的优先级可配

9.2.4.4 显示通道

显示通道特性

- DHD0 可作为高清、标清的输出通道。



- DHD0 通道只能选择一个输出接口输出。
- DHD0 支持以下典型输出时序：1080P@60fps、720P@60fps。

时序配置

VDP 的输出接口支持配置各种典型及非典型时序，以适应不同的对接芯片接口。



注意

所有时序参数配置时，接口都应该关闭，配置完成后再打开接口。

9.2.4.5 高清输出接口

HDMI 1.4 特性

- 支持 8bit 输出。
- 时钟频率为 74.25MHz~297MHz
- 最大支持 1080P@60fps YUV444/RGB 输出。

MIPI Tx 特性

- 支持 8bit 输出。
- 最大支持 1080P@60fps 显示。
- 支持显示以下典型输出时序：1080P@60fps、720P@60fps。

9.2.4.6 高清输出接口 BT.1120

BT.1120 特性

- 支持 YUV422 8bit 输出。
- 支持数据钳位的 clip，根据接口协议，Y 钳位范围 16~235，C 钳位范围 16~240。
- 支持以下典型输出时序：720P、1080P。
- 支持 bypass 模式，dither、dfir 分别采用截位、丢点的方式处理。
- 16bit 数据，默认 Y 在高位，C 在低位，YC 位置可互换。
- 支持输出随路时钟反相。

9.2.4.7 标清输出接口 BT.656

BT.656 特性

- 支持 YUV422 8bit 输出。
- 支持数据钳位的 clip，根据接口协议，Y 钳位范围 16~235，C 钳位范围 16~240。
- 支持以下典型输出时序：NTSC、PAL。



9.2.4.8 LCD 输出接口 RGB

- 支持 8bit/6bit 串行 RGB 输出 bit 位顺序、倒序可调。
- 支持 16bit/18bit/24bit 并行 RGB 输出，默认 RGB 顺序，BGR 顺序输出可选。
- 支持 RGB 串行输出，输出顺序 R、G、B，16bit 中的低 8bit 输出。
- 支持 RGB 并行输出，输出顺序 RGB、BGR 可选。

9.2.5 VDP 寄存器概览

VDP 寄存器概览如表 9-3 所示。

表9-3 VDP 寄存器概览（基址是 0x1144_0000）

偏移地址	名称	描述	页码
0x0000	VOCTRL	VDP 控制寄存器	9-69
0x0004	VOINTSTA	VDP 中断状态寄存器	9-69
0x0008	VOMSKINTSTA	VDP 经过 Mask 的中断状态寄存器	9-70
0x000C	VOINTMSK	VDP 中断屏蔽寄存器	9-70
0x0014	VOINTSTA1	VDP 中断状态 1 寄存器	9-70
0x0018	VOMSKINTSTA1	VDP 经过 Mask 的离线中断状态寄存器 1	9-71
0x001C	VOINTMSK1	VDP 中断屏蔽寄存器 1	9-71
0x0300	CBM_BKG1	CBM Mixer1 叠加背景色寄存器	9-72
0x0308	CBM_MIX1	CBM Mixer1 优先级配置寄存器	9-72
0x1000	V0_CTRL	V0 控制寄存器，为非即时寄存器	9-73
0x1004	V0_UPD	V0 通道更新使能寄存器	9-73
0x1008	V0_0RESO_READ	V0 层输出寄存器	9-74
0x1080	V0_DFPOS	Surface 在显示窗口的起始位置（First Position），以像素为单位，非即时寄存器	9-74
0x1084	V0_DLPOS	Surface 在显示窗口的结束位置（Last Position），以像素为单位，非即时寄存器	9-74
0x1088	V0_VFPOS	Surface 真实内容在显示窗口的起始位置（First Position），以像素为单位，非即时寄存器	9-75



偏移地址	名称	描述	页码
0x108C	V0_VLPOS	Surface 真实内容在显示窗口的结束位置 (Last Position)，以像素为单位，非即时寄存器	9-75
0x1090	V0_BK	surface 的背景色寄存器	9-75
0x1094	V0_ALPHA	surface 的背景填充色 ALPHA 寄存器	9-76
0x1200	V0_HIPP_CSC_CTRL	CSC 控制寄存器	9-76
0x1204	V0_HIPP_CSC_COEF_00	HIPP_CSC 系数寄存器 00	9-76
0x1208	V0_HIPP_CSC_COEF_01	HIPP_CSC 系数寄存器 01	9-76
0x120C	V0_HIPP_CSC_COEF_02	HIPP_CSC 系数寄存器 02	9-77
0x1210	V0_HIPP_CSC_COEF_10	HIPP_CSC 系数寄存器 10	9-77
0x1214	V0_HIPP_CSC_COEF_11	HIPP_CSC 系数寄存器 11	9-77
0x1218	V0_HIPP_CSC_COEF_12	HIPP_CSC 系数寄存器 12	9-78
0x121C	V0_HIPP_CSC_COEF_20	HIPP_CSC 系数寄存器 20	9-78
0x1220	V0_HIPP_CSC_COEF_21	HIPP_CSC 系数寄存器 21	9-78
0x1224	V0_HIPP_CSC_COEF_22	HIPP_CSC 系数寄存器 22	9-78
0x1228	V0_HIPP_CSC_SCALE	HIPP_CSC 精度寄存器	9-79
0x122C	V0_HIPP_CSC_IDC0	HIPP_CSC 输入直流分量寄存器 0	9-79
0x1230	V0_HIPP_CSC_IDC1	HIPP_CSC 输入直流分量寄存器 1	9-79
0x1234	V0_HIPP_CSC_IDC2	HIPP_CSC 输入直流分量寄存器 2	9-79
0x1238	V0_HIPP_CSC_ODC0	HIPP_CSC 输出直流分量寄存器 0	9-80
0x123C	V0_HIPP_CSC_ODC1	HIPP_CSC 输出直流分量寄存器 1	9-79
0x1240	V0_HIPP_CSC_ODC2	HIPP_CSC 输出直流分量寄存器 2	9-79
0x1244	V0_HIPP_CSC_MIN_Y	HIPP_CSC Y 分量 CLIP 最小值寄存器	9-81
0x1248	V0_HIPP_CSC_MIN_C	HIPP_CSC C 分量 CLIP 最小值寄存器	9-81



偏移地址	名称	描述	页码
0x124C	V0_HIPP_CSC_MAX_Y	HIPP_CSC Y 分量 CLIP 最大值寄存器	9-81
0x1250	V0_HIPP_CSC_MAX_C	HIPP_CSC C 分量 CLIP 最大值寄存器	9-81
0x1400	V0_CVFIR_VINFO	亮度垂直位置偏移寄存器	9-82
0x1404	V0_CVFIR_VSP	垂直缩放参数配置（Vertical Scaling Parameter）寄存器	9-82
0x1500	V0_HFIR_CTRL	hfir 控制寄存器	9-83
0x1504	V0_HFIRCOEF01	HFIR 滤波系数 01 寄存器	9-83
0x1508	V0_HFIRCOEF23	HFIR 滤波系数 23 寄存器	9-83
0x150C	V0_HFIRCOEF45	HFIR 滤波系数 45 寄存器	9-84
0x1510	V0_HFIRCOEF67	HFIR 滤波系数 67 寄存器	9-84
0x7000	G0_CTRL	G0 控制寄存器，为非即时寄存器	9-84
0x7004	G0_UPD	图形层更新使能寄存器	9-85
0x700C	G0_0RESO_READ	输出分辨率寄存器	9-85
0x7080	G0_DFPOS	Surface 在显示窗口的起始位置（First Position），以像素为单位，非即时寄存器	9-85
0x7084	G0_DLPOS	Surface 在显示窗口的结束位置（Last Position），以像素为单位，非即时寄存器	9-86
0x7088	G0_VFPOS	Surface 真实内容在显示窗口的起始位置（First Position），以像素为单位，非即时寄存器	9-86
0x708C	G0_VLPOS	Surface 真实内容在显示窗口的结束位置（Last Position），以像素为单位，非即时寄存器	9-86
0x7090	G0_BK	图形层的背景色寄存器	9-87
0x7094	G0_ALPHA	图形层的背景填充色 ALPHA 寄存器	9-87
0x7100	G0_HIPP_CSC_CTRL	CSC 控制寄存器	9-87
0x7104	G0_HIPP_CSC_COEF_00	HIPP_CSC 系数寄存器 00	9-88
0x7108	G0_HIPP_CSC_COEF_01	HIPP_CSC 系数寄存器 01	9-88



偏移地址	名称	描述	页码
0x710C	G0_HIPP_CSC_COEF_02	HIPP_CSC 系数寄存器 02	9-88
0x7110	G0_HIPP_CSC_COEF_10	HIPP_CSC 系数寄存器 10	9-88
0x7114	G0_HIPP_CSC_COEF_11	HIPP_CSC 系数寄存器 11	9-89
0x7118	G0_HIPP_CSC_COEF_12	HIPP_CSC 系数寄存器 12	9-89
0x711C	G0_HIPP_CSC_COEF_20	HIPP_CSC 系数寄存器 20	9-89
0x7120	G0_HIPP_CSC_COEF_21	HIPP_CSC 系数寄存器 21	9-90
0x7124	G0_HIPP_CSC_COEF_22	HIPP_CSC 系数寄存器 22	9-90
0x7128	G0_HIPP_CSC_SCAL_E	HIPP_CSC 精度寄存器	9-90
0x712C	G0_HIPP_CSC_IDC0	HIPP_CSC 输入直流分量寄存器 0	9-90
0x7130	G0_HIPP_CSC_IDC1	HIPP_CSC 输入直流分量寄存器 1	9-91
0x7134	G0_HIPP_CSC_IDC2	HIPP_CSC 输入直流分量寄存器 2	9-91
0x7138	G0_HIPP_CSC_ODC0	HIPP_CSC 输出直流分量寄存器 0	9-91
0x713C	G0_HIPP_CSC_ODC1	HIPP_CSC 输出直流分量寄存器 1	9-91
0x7140	G0_HIPP_CSC_ODC2	HIPP_CSC 输出直流分量寄存器 2	9-91
0x7144	G0_HIPP_CSC_MIN_Y	HIPP_CSC Y 分量 CLIP 最小值寄存器	9-92
0x7148	G0_HIPP_CSC_MIN_C	HIPP_CSC C 分量 CLIP 最小值寄存器	9-92
0x714C	G0_HIPP_CSC_MAX_Y	HIPP_CSC Y 分量 CLIP 最大值寄存器	9-93
0x7150	G0_HIPP_CSC_MAX_C	HIPP_CSC C 分量 CLIP 最大值寄存器	9-93
0x7300	G0_ZME_HINFO	亮度水平位置偏移寄存器	9-93
0x7304	G0_ZME_HSP	水平缩放参数配置（Horizontal Scaling Parameter）寄存器	9-93
0x7380	G0_ZME_VINFO	亮度垂直位置偏移寄存器。	9-94
0x7384	G0_ZME_VSP	垂直缩放参数配置（Vertical Scaling Parameter）寄存器。	9-95



偏移地址	名称	描述	页码
0xD000	DHD0_CTRL	显示通道的总体控制寄存器	9-96
0xD004	DHD0_VSYNC1	显示通道的垂直时序控制寄存器	9-96
0xD008	DHD0_VSYNC2	显示通道的垂直时序控制寄存器	9-97
0xD00C	DHD0_HSYNC1	显示通道的水平时序控制 1 寄存器	9-97
0xD010	DHD0_HSYNC2	显示通道的水平时序控制 2 寄存器	9-97
0xD014	DHD0_VPLUS1	显示通道隔行输出的底场垂直时序控制寄存器	9-98
0xD018	DHD0_VPLUS2	显示通道隔行输出的底场垂直时序控制寄存器	9-98
0xD01C	DHD0_PWR	同步信号脉冲宽度寄存器，非即时寄存器	9-98
0xD024	DHD0_VTTHD	垂直时序门限值（Vertical Timing Threshold），为即时寄存器	9-99
0xD028	DHD0_PARATHD	PARA 系数更新点阈值寄存器	9-99
0xD030	DHD0_START_POS	DHD0 通道 start 信号起始位置寄存器	9-100
0xD088	DHD0_STATE	DHD0 状态寄存器	9-100
0xD090	VO_MUX	VDP 输出接口复选寄存器	9-101
0xD0B0	INTF0_DITHER_CTRL	Dither 控制寄存器	9-101
0xD204	INTF_BT_UPD	INTF_BT 更新寄存器	9-103
0xD208	INTF_BT_SYNC_INV	外同步时序输出时，同步信号极性配置寄存器	9-103
0xD210	BT_CLIP0_L	BT.1120 Clip 处理最低门限值寄存器，为即时寄存器	9-103
0xD214	BT_CLIP0_H	BT.1120 Clip 处理最高门限值寄存器，为即时寄存器	9-104
0xD300	INTF_LCD_CTRL	接口通道配置寄存器	9-104
0xD304	INTF_LCD_UPD	INTF_LCD 更新寄存器	9-105
0xD308	INTF_LCD_SYNC_INV	外同步时序输出时，同步信号极性配置寄存器	9-105
0x10000	MAC_OUTSTANDING	OUTSTANDING 寄存器	9-106



偏移地址	名称	描述	页码
0x10200	VID_READ_CTRL	VID FDR 读取控制寄存器，为非即时寄存器	9-106
0x10210	VID_OUT_CTRL	VID FDR 输出控制寄存器，为非即时寄存器	9-107
0x10240	VID_SRC_INFO	VID FDR 输入图像属性寄存器，为非即时寄存器	9-107
0x10244	VID_SRC_RESO	原始图像分辨率，非及时寄存器	9-108
0x10248	VID_SRC_CROP	源图的 CROP 寄存器，非及时寄存器	9-108
0x1024C	VID_IN_RESO	输入图像分辨率寄存器，为非即时寄存器	9-109
0x10380	ICE_V3R2_SEG_DCM P_GLB_INFO	亮度 DCMP 控制寄存器	9-109
0x103C0	ICE_V3R2_SEG_DCM P_GLB_INFO_CHM	色度 DCMP 控制寄存器	9-109
0x10E08	GFX_OUT_CTRL	GFX FDR 输出控制寄存器，为非即时寄存器	9-110
0x10E28	GFX_1555_ALPHA	ARGB1555 格式 alpha 值，为非即时寄存器	9-111
0x10E40	GFX_SRC_INFO	GFX FDR 源图像属性寄存器，为非即时寄存器	9-111
0x10E44	GFX_SRC_RESO	原始图像分辨率，非及时寄存器	9-112
0x10E48	GFX_SRC_CROP	源图的 CROP 寄存器，非及时寄存器	9-112
0x10E4C	GFX_IRESO	输入分辨率寄存器，为非即时寄存器	9-113
0x10F00	GFX_CKEY_MAX	color key 最大值，为非即时寄存器	9-113
0x10F04	GFX_CKEY_MIN	color key 最小值，为非即时寄存器	9-113
0x10F08	GFX_CKEY_MASK	color key mask 值，为非即时寄存器	9-114

9.2.6 VDP 寄存器描述

VOCTRL

VOCTRL 为 VDP 控制寄存器。

Offset Address: 0x0000 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31]	RW	vo_ck_gt_en	VDP 时钟门控使能，既低功耗模式寄存器。 0: 时钟门控关闭; 1: VDP 的内部时钟门控打开。	0x0
[30:0]	-	reserved	保留。	0x00000000

VOINTSTA

VOINTSTA 为 VDP 中断状态寄存器。

Offset Address: 0x0004 Total Reset Value: 0x0000_0044

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000
[3]	RO	dhd0uf_int	HD0 通道低带宽告警中断。 0: 无中断; 1: 上报中断。	0x0
[2]	-	reserved	保留。	0x1
[1]	-	reserved	保留。	0x0
[0]	RO	dhd0vtthd1_int	HD0 通道垂直时序中断 1。 0: 无中断; 1: 上报中断。	0x0

VOMSKINTSTA

VOMSKINTSTA 为 VDP 经过 Mask 的中断状态寄存器。写 1 清零。

Offset Address: 0x0008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000
[3]	WC	dhd0uf_clr	HD0 通道低带宽告警中断。 0: 无中断; 1: 上报中断。	0x0
[2]	-	reserved	保留。	0x0
[1]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[0]	WC	dhd0vtthd1_clr	HD0 通道垂直时序中断 1。 0: 无中断; 1: 上报中断。	0x0

VOINTMSK

VOINTMSK 为 VDP 中断屏蔽寄存器。与 VOINTSTA 对应。相应比特为 1 表示中断打开，为 0 表示中断屏蔽。

Offset Address: 0x000C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000
[3]	RW	dhd0uf_intmask	HD0 通道低带宽告警中断屏蔽。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	-	reserved	保留。	0x0
[1]	-	reserved	保留。	0x0
[0]	RW	dhd0vtthd1_intmask	HD0 通道垂直时序中断 1 屏蔽。 0: 屏蔽中断; 1: 使能中断。	0x0

VOINTSTA1

VOINTSTA1 为 VDP 中断状态 1 寄存器。

Offset Address: 0x0014 Total Reset Value: 0x0000_0044

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RO	dhd0uf_int	HD0 通道低带宽告警中断。 0: 无中断; 1: 上报中断。	0x0
[2]	-	reserved	保留。	0x1
[1]	-	reserved	保留。	0x0
[0]	RO	dhd0vtthd1_int	HD0 通道垂直时序中断 1。	0x0



Bits	Access	Name	Description	Reset
			0: 无中断; 1: 上报中断。	

VOMSKINTSTA1

VOMSKINTSTA1 为 VDP 经过 Mask 的离线中断状态寄存器 1。写 1 清零。

Offset Address: 0x0018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	WC	dhd0uf_clr	HD0 通道低带宽告警中断。 0: 无中断; 1: 上报中断。	0x0
[2]	-	reserved	保留。	0x0
[1]	-	reserved	保留。	0x0
[0]	WC	dhd0vtthd1_clr	HD0 通道垂直时序中断 1。 0: 无中断; 1: 上报中断。	0x0

VOINTMSK1

VOINTMSK1 为 VDP 中断屏蔽寄存器 1。与 VOINTSTA 对应。

Offset Address: 0x001C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RW	dhd0uf_intmask	HD0 通道低带宽告警中断屏蔽。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	-	reserved	保留。	0x0
[1]	-	reserved	保留。	0x0
[0]	RW	dhd0vtthd1_intmask	HD0 通道垂直时序中断 1 屏蔽。 0: 屏蔽中断; 1: 使能中断。	0x0



CBM_BKG1

CBM_BKG1 为 CBM Mixer1 叠加背景色寄存器。

Offset Address: 0x0300 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29:20]	RW	cbm_bkgy1	CBM Mixer1 叠加背景色，Y 分量。	0x000
[19:10]	RW	cbm_bkgcb1	CBM Mixer1 叠加背景色，Cb 分量。	0x000
[9:0]	RW	cbm_bkgcr1	CBM Mixer1 叠加背景色，Cr 分量。	0x000

CBM_MIX1

CBM_MIX1 为 CBM Mixer1 优先级配置寄存器。在 vsync 处更新有效。mixer_prio_x 表示第 x 个优先级所配置的层。

该寄存器为即时寄存器。

Offset Address: 0x0308 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:4]	RW	mixer_prio1	CBM Mixer1 的叠加层优先级配置，表示优先级 1 的驱动层。 0x0: 表示没有层驱动； 0x1: V0； 0x2: G0； 其他：保留。	0x0
[3:0]	RW	mixer_prio0	CBM Mixer1 的叠加层优先级配置，表示优先级 0 的驱动层。 0x0: 表示没有层驱动； 0x1: V0； 0x2: G0； 其他：保留。	0x1

V0_CTRL

V0_CTRL 为 V0 控制寄存器，为非即时寄存器。



Offset Address: 0x1000 Total Reset Value: 0x4000_00FF

Bits	Access	Name	Description	Reset
[31]	RW	surface_en	surface 使能。非即时寄存器。 0: 禁止; 1: 使能。	0x0
[30]	-	reserved	保留。	0x1
[29]	-	reserved	保留。	0x0
[28:8]	-	reserved	保留。	0x00000
[7:0]	RW	galpha	叠加全局 alpha 值。取值范围为 0~255, 255 为全不透明, 0 为全透明。	0xFF

V0_UPD

V0_UPD 为 V0 通道更新使能寄存器。

Offset Address: 0x1004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	WC	regup	surface 的寄存器更新。当软件配置完该层 寄存器后写 1 更新, 更新完成后, 硬件自 动清零。	0x0

V0_0RESO_READ

V0_0RESO_READ 为 V0 输出分辨率读取寄存器,

Offset Address: 0x1008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	oh	高度, 行为单位。实际高度减 1。 以帧高度为参考, 以行为单位。	0x0000
[15:0]	RO	ow	宽度, 像素单位。实际宽度减 1。 以帧宽度为参考, 以列为单位。	0x0000



V0_DFPOS

V0_DFPOS 为 Surface 在显示窗口的起始位置（First Position），以像素为单位，非即时寄存器。

Offset Address: 0x1080 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	disp_yfpos	显示行起始坐标。 以帧高度为参考，以行为单位。	0x0000
[15:0]	RW	disp_xfpos	显示列起始坐标。 以帧宽度为参考，以列为单位。	0x0000

V0_DLPOS

V0_DLPOS 为 Surface 在显示窗口的结束位置（Last Position），以像素为单位，非即时寄存器。

Offset Address: 0x1084 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	disp_ylpos	显示行结束坐标。 以帧高度为参考，以行为单位。	0x0000
[15:0]	RW	disp_xlpos	显示列结束坐标。 以帧宽度为参考，以列为单位。	0x0000

V0_VFPOS

V0_VFPOS 为 Surface 真实内容在显示窗口的起始位置（First Position），以像素为单位，非即时寄存器。

Offset Address: 0x1088 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	video_yfpos	视频内容行起始坐标。 以帧高度为参考，以行为单位。	0x0000
[15:0]	RW	video_xfpos	视频内容列起始坐标。 以帧宽度为参考，以列为单位。	0x0000



V0_VLPOS

V0_VLPOS 为 Surface 真实内容在显示窗口的结束位置 (Last Position)，以像素为单位，非即时寄存器。

Offset Address: 0x108C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	video_ylpos	视频内容行结束坐标。 以帧高度为参考，以行为单位。	0x0000
[15:0]	RW	video_xlpos	视频内容列结束坐标。 以帧宽度为参考，以列为单位。	0x0000

V0_BK

V0_BK 为 surface 的背景色寄存器。

Offset Address: 0x1090 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29:20]	RW	vbk_y	Y 分量。	0x000
[19:10]	RW	vbk_cb	Cb 分量。	0x000
[9:0]	RW	vbk_cr	Cr 分量。	0x000

V0_ALPHA

V0_ALPHA 为 surface 的背景填充色 ALPHA 寄存器。

Offset Address: 0x1094 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	vbk_alpha	surface 的背景填充色 0~255 级。	0x00

V0_HIPP_CSC_CTRL

V0_HIPP_CSC_CTRL 为 CSC 控制寄存器。

Offset Address: 0x1200 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留	0x00000000
[0]	RW	hipp_csc_en	Hipp CSC 使能寄存器。 0: 不使能; 1: 使能。	0x0

V0_HIPP_CSC_COEF00

V0_HIPP_CSC_COEF00 为 CSC 系数寄存器 00。

Offset Address: 0x1204 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef00	CSC 系数寄存器 00 带符号位 s1.14。	0x0000

V0_HIPP_CSC_COEF01

V0_HIPP_CSC_COEF01 为 CSC 系数寄存器 01。

Offset Address: 0x1208 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef01	CSC 系数寄存器 01 带符号位 s1.14。	0x0000

V0_HIPP_CSC_COEF02

V0_HIPP_CSC_COEF02 为 CSC 系数寄存器 02。

Offset Address: 0x120C Total Reset Value: 0x0000_0000

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



[15:0]	RW	hipp_csc_coef02	CSC 系数寄存器 02 带符号位 s1.14。	0x0000

V0_HIPP_CSC_COEF10

V0_HIPP_CSC_COEF10 为 CSC 系数寄存器 10。

Offset Address: 0x1210 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef10	CSC 系数寄存器 10 带符号位 s1.14。	0x0000

V0_HIPP_CSC_COEF11

V0_HIPP_CSC_COEF11 为 CSC 系数寄存器 11。

Offset Address: 0x1214 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef11	CSC 系数寄存器 11 带符号位 s1.14。	0x0000

V0_HIPP_CSC_COEF12

V0_HIPP_CSC_COEF12 为 CSC 系数寄存器 12。

Offset Address: 0x1218 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef12	CSC 系数寄存器 12 带符号位 s1.14。	0x0000



V0_HIPP_CSC_COEF20

V0_HIPP_CSC_COEF20 为 CSC 系数寄存器 20。

Offset Address: 0x121C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef20	CSC 系数寄存器 20 带符号位 s1.14。	0x0000

V0_HIPP_CSC_COEF21

V0_HIPP_CSC_COEF21 为 CSC 系数寄存器 21。

Offset Address: 0x1220 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef21	CSC 系数寄存器 21 带符号位 s1.14。	0x0000

V0_HIPP_CSC_COEF22

V0_HIPP_CSC_COEF22 为 CSC 系数寄存器 22。

Offset Address: 0x1224 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef22	CSC 系数寄存器 22 带符号位 s1.14。	0x0000



V0_HIPP_CSC_SCALE

V0_HIPP_CSC_SCALE 为 CSC 精度寄存器。

Offset Address: 0x1228 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留	0x0000000
[3:0]	RW	hipp_csc_scale	CSC 精度寄存器 无符号数 u4.0。	0x0

V0_HIPP_CSC_IDC0

V0_HIPP_CSC_IDC0 为 CSC 输入直流分量寄存器 0。

Offset Address: 0x122C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留	0x0000000
[10:0]	RW	hipp_csc_idc0	Hipp CSC 输入直流分量寄存器 0，带符号位 s10.0。	0x000

V0_HIPP_CSC_IDC1

V0_HIPP_CSC_IDC1 为 CSC 输入直流分量寄存器 1。

Offset Address: 0x1230 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留	0x0000000
[10:0]	RW	hipp_csc_idc1	Hipp CSC 输入直流分量寄存器 1，带符号位 s10.0。	0x000

V0_HIPP_CSC_IDC2

V0_HIPP_CSC_IDC2 为 CSC 输入直流分量寄存器 2。

Offset Address: 0x1234 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留	0x0000000



[10:0]	RW	hipp_csc_idc2	Hipp CSC 输入直流分量寄存器 2，带符号位 s10.0。	0x000

V0_HIPP_CSC_ODC0

V0_HIPP_CSC_ODC0 为 CSC 输出直流分量寄存器 0。

Offset Address: 0x1238 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留	0x000000
[10:0]	RW	hipp_csc_odc0	Hipp CSC 输出直流分量寄存器 0，带符号位 s10.0。	0x000

V0_HIPP_CSC_ODC1

V0_HIPP_CSC_ODC1 为 CSC 输出直流分量寄存器 1。

Offset Address: 0x123C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留	0x000000
[10:0]	RW	hipp_csc_odc1	Hipp CSC 输出直流分量寄存器 1，带符号位 s10.0。	0x000

V0_HIPP_CSC_ODC2

V0_HIPP_CSC_ODC2 为 CSC 输出直流分量寄存器 2。

Offset Address: 0x1240 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留	0x000000
[10:0]	RW	hipp_csc_odc2	Hipp CSC 输出直流分量寄存器 2，带符号位 s10.0。	0x000

V0_HIPP_CSC_MIN_Y

V0_HIPP_CSC_MIN_Y 为 CSC Y 分量 CLIP 最小值寄存器。



Offset Address: 0x1244 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留	0x000000
[9:0]	RW	hipp_csc_min_y	CSC Y 分量 CLIP 最小值寄存器 无符号数 u10.0。	0x000

V0_HIPP_CSC_MIN_C

V0_HIPP_CSC_MIN_C 为 CSC C 分量 CLIP 最小值寄存器。

Offset Address: 0x1248 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留	0x000000
[9:0]	RW	hipp_csc_min_c	CSC C 分量 CLIP 最小值寄存器 无符号数 u10.0。	0x000

V0_HIPP_CSC_MAX_Y

V0_HIPP_CSC_MAX_Y 为 CSC Y 分量 CLIP 最大值寄存器。

Offset Address: 0x124C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留	0x000000
[9:0]	RW	hipp_csc_max_y	CSC Y 分量 CLIP 最大值寄存器 无符号数 u10.0。	0x000

V0_HIPP_CSC_MAX_C

V0_HIPP_CSC_MAX_C 为 CSC C 分量 CLIP 最大值寄存器。

Offset Address: 0x1250 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留	0x000000
[9:0]	RW	hipp_csc_max_c	CSC C 分量 CLIP 最大值寄存器 无符号数 u10.0。	0x000



V0_CVFIR_VINFO

V0_CVFIR_VINFO 为亮度垂直位置偏移寄存器。用于 pan-scan。为非即时寄存器。

Offset Address: 0x1400 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:19]	-	reserved	保留。	0x000
[18]	RW	out_pro	垂直缩放输出逐隔行 0: 隔行; 1: 逐行;	0x0
[17:16]	RW	out_fmt	垂直缩放输出数据格式。 00: 420; 01: 422; 10: 444; 11: 保留。	0x0
[15:0]	RW	out_height	垂直亮度单眼输出高度。	0x0000

V0_CVFIR_VSP

V0_CVFIR_VSP 为垂直缩放参数配置（Vertical Scaling Parameter）寄存器。为非即时寄存器。

Offset Address: 0x1404 Total Reset Value: 0x0000_1000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30]	RW	cvfir_en	垂直色度缩放使能。 0: 禁止; 1: 使能。	0x0
[29]	-	reserved	保留。	0x0
[28]	RW	cvmid_en	垂直色度缩放中值滤波使能(当 chfir_en 无效时, 该比特不起作用)。 0: 禁止; 1: 使能。	0x0
[27]	-	reserved	保留。	0x0
[26]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[25]	RW	cvfir_mode	垂直色度缩放模式。 0: 复制模式(滤波不使能); 1: 滤波模式(滤波使能)。	0x0
[24:16]	-	reserved	保留。	0x00
[15:0]	-	reserved	保留。	0x1000

V0_HFIR_CTRL

V0_HFIR_CTRL 为 hfir 控制寄存器。

Offset Address: 0x1500 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x0000000
[2:1]	RW	hfir_mode	水平色度 HFIR 模式。 00: 保留; 01: 色度 HFIR 拷贝模式; 10: 双线性插值; 11: 6 阶 FIR。	0x0
[0]	-	reserved	保留。	0x0

V0_HFIRCOEF01

V0_HFIRCOEF01 为 HFIR 滤波系数 01 寄存器。

Offset Address: 0x1504 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:16]	RW	coef1	HFIR 滤波系数 1。	0x000
[15:10]	-	reserved	保留。	0x00
[9:0]	RW	coef0	HFIR 滤波系数 0。	0x000

V0_HFIRCOEF23

V0_HFIRCOEF23 为 HFIR 滤波系数 23 寄存器。



Offset Address: 0x1508 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:16]	RW	coef3	HFIR 滤波系数 3。	0x000
[15:10]	-	reserved	保留。	0x00
[9:0]	RW	coef2	HFIR 滤波系数 2。	0x000

V0_HFIRCOEF45

V0_HFIRCOEF45 为 HFIR 滤波系数 45 寄存器。

Offset Address: 0x150C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:16]	RW	coef5	HFIR 滤波系数 5。	0x000
[15:10]	-	reserved	保留。	0x00
[9:0]	RW	coef4	HFIR 滤波系数 4。	0x000

V0_HFIRCOEF67

V0_HFIRCOEF67 为 HFIR 滤波系数 67 寄存器。

Offset Address: 0x1510 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:16]	RW	coef7	HFIR 滤波系数 7。	0x000
[15:10]	-	reserved	保留。	0x00
[9:0]	RW	coef6	HFIR 滤波系数 6。	0x000

G0_CTRL

G0_CTRL 为 G0 控制寄存器，为非即时寄存器。

Offset Address: 0x7000 Total Reset Value: 0x4000_00FF



Bits	Access	Name	Description	Reset
[31]	RW	surface_en	surface 使能，非即时寄存器。 0: 禁止； 1: 使能。	0x0
[30]	-	reserved	保留。	0x1
[29:8]	-	reserved	保留。	0x00000
[7:0]	RW	galpha	叠加全局 alpha 值。取值范围为 0~255， 255 为全不透明，0 为全透明。	0xFF

G0_UPD

G0_UPD 为图形层更新使能寄存器。

Offset Address: 0x7004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	WC	regup	surface 的寄存器更新。当软件配置完该层 寄存器后写 1 更新，更新完成后，硬件自 动清零。	0x0

G0_0RESO_READ

G0_0RESO_READ 为输出分辨率寄存器，

Offset Address: 0x700C Total Reset Value: 0x0000_077F

Bits	Access	Name	Description	Reset
[31:16]	RO	oh	高度，行为单位。实际高度减 1。 以帧高度为参考，以行为单位。	0x0000
[15:0]	RO	ow	宽度，像素单位。实际宽度减 1。 以帧宽度为参考，以列为单位。	0x077F

G0_DFPOS

G0_DFPOS 为 Surface 在显示窗口的起始位置（First Position），以像素为单位，非即时寄存器。

Offset Address: 0x7080 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	RW	disp_yfpos	显示行起始坐标。 以帧高度为参考，以行为单位。	0x0000
[15:0]	RW	disp_xfpos	显示列起始坐标。 以帧宽度为参考，以列为单位。	0x0000

G0_DLPOS

G0_DLPOS 为 Surface 在显示窗口的结束位置（Last Position），以像素为单位，非即时寄存器。

Offset Address: 0x7084 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	disp_ylpos	显示行结束坐标。 以帧高度为参考，以行为单位。	0x0000
[15:0]	RW	disp_xlpos	显示列结束坐标。 以帧宽度为参考，以列为单位。	0x0000

G0_VFPOS

G0_VFPOS 为 Surface 真实内容在显示窗口的起始位置（First Position），以像素为单位，非即时寄存器。

Offset Address: 0x7088 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	video_yfpos	视频行起始坐标。 以帧高度为参考，以行为单位。	0x0000
[15:0]	RW	video_xfpos	视频列起始坐标。 以帧宽度为参考，以列为单位。	0x0000

G0_VLPOS

G0_VLPOS 为 Surface 真实内容在显示窗口的结束位置（Last Position），以像素为单位，非即时寄存器。

Offset Address: 0x708C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	RW	video_ylpos	视频行结束坐标。 以帧高度为参考，以行为单位。	0x0000
[15:0]	RW	video_xlpos	视频列结束坐标。 以帧宽度为参考，以列为单位。	0x0000

G0_BK

G0_BK 为图形层的背景色寄存器。

Offset Address: 0x7090 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29:20]	RW	vbk_y	Y 分量。	0x000
[19:10]	RW	vbk_cb	Cb 分量。	0x000
[9:0]	RW	vbk_cr	Cr 分量。	0x000

G0_ALPHA

G0_ALPHA 为图形层的背景填充色 ALPHA 寄存器。

Offset Address: 0x7094 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	vbk_alpha	图形层的背景填充色 0~255 级。	0x00

G0_HIPP_CSC_CTRL

G0_HIPP_CSC_CTRL 为 CSC 控制寄存器。

Offset Address: 0x7100 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留	0x00000000
[0]	RW	hipp_csc_en	Hipp CSC 使能寄存器。 0: 不使能;	0x0



Bits	Access	Name	Description	Reset
			1: 使能。	

G0_HIPP_CSC_COEF00

G0_HIPP_CSC_COEF00 为 CSC 系数寄存器 00。

Offset Address: 0x7104 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef00	带符号位 s1.14。	0x0000

G0_HIPP_CSC_COEF01

G0_HIPP_CSC_COEF01 为 CSC 系数寄存器 01。

Offset Address: 0x7108 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef01	带符号位 s1.14。	0x0000

G0_HIPP_CSC_COEF02

G0_HIPP_CSC_COEF02 为 CSC 系数寄存器 02。

Offset Address: 0x710C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef02	带符号位 s1.14。	0x0000

G0_HIPP_CSC_COEF10

G0_HIPP_CSC_COEF10 为 CSC 系数寄存器 10。



Offset Address: 0x7110 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef10	带符号位 s1.14。	0x0000

G0_HIPP_CSC_COEF11

G0_HIPP_CSC_COEF11 为 CSC 系数寄存器 11。

Offset Address: 0x7114 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef11	带符号位 s1.14。	0x0000

G0_HIPP_CSC_COEF12

G0_HIPP_CSC_COEF12 为 CSC 系数寄存器 12。

Offset Address: 0x7118 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef12	带符号位 s1.14。	0x0000

G0_HIPP_CSC_COEF20

G0_HIPP_CSC_COEF20 为 CSC 系数寄存器 20。

Offset Address: 0x711C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef20	带符号位 s1.14。	0x0000



G0_HIPP_CSC_COEF21

G0_HIPP_CSC_COEF21 为 CSC 系数寄存器 21。

Offset Address: 0x7120 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef21	带符号位 s1.14。	0x0000

G0_HIPP_CSC_COEF22

G0_HIPP_CSC_COEF22 为 CSC 系数寄存器 22。

Offset Address: 0x7124 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留	0x0000
[15:0]	RW	hipp_csc_coef22	带符号位 s1.14。	0x0000

G0_HIPP_CSC_SCALE

G0_HIPP_CSC_SCALE 为 CSC 精度寄存器。

Offset Address: 0x7128 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留	0x0000000
[3:0]	RW	hipp_csc_scale	无符号数 u4.0。	0x0

G0_HIPP_CSC_IDC0

G0_HIPP_CSC_IDC0 为 CSC 输入直流分量寄存器 0。

Offset Address: 0x712C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留	0x000000
[10:0]	RW	hipp_csc_idc0	Hipp CSC 输入直流分量寄存器 0，带符号位 s10.0。	0x000

G0_HIPP_CSC_IDC1

G0_HIPP_CSC_IDC1 为 CSC 输入直流分量寄存器 1。

Offset Address: 0x7130 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留	0x000000
[10:0]	RW	hipp_csc_idc1	Hipp CSC 输入直流分量寄存器 1，带符号位 s10.0。	0x000

G0_HIPP_CSC_IDC2

G0_HIPP_CSC_IDC2 为 CSC 输入直流分量寄存器 2。

Offset Address: 0x7134 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留	0x000000
[10:0]	RW	hipp_csc_idc2	Hipp CSC 输入直流分量寄存器 2，带符号位 s10.0。	0x000

G0_HIPP_CSC_ODC0

G0_HIPP_CSC_ODC0 为 CSC 输出直流分量寄存器 0。

Offset Address: 0x7138 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留	0x000000
[10:0]	RW	hipp_csc_odc0	Hipp CSC 输出直流分量寄存器 0，带符号位 s10.0。	0x000



G0_HIPP_CSC_ODC1

G0_HIPP_CSC_ODC1 为 CSC 输出直流分量寄存器 1。

Offset Address: 0x713C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留	0x000000
[10:0]	RW	hipp_csc_odc1	Hipp CSC 输出直流分量寄存器 1，带符号位 s10.0。	0x000

G0_HIPP_CSC_ODC2

G0_HIPP_CSC_ODC2 为 CSC 输出直流分量寄存器 2。

Offset Address: 0x7140 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留	0x000000
[10:0]	RW	hipp_csc_odc2	Hipp CSC 输出直流分量寄存器 2，带符号位 s10.0。	0x000

G0_HIPP_CSC_MIN_Y

G0_HIPP_CSC_MIN_Y 为 CSC Y 分量 CLIP 最小值寄存器。

Offset Address: 0x7144 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留	0x000000
[9:0]	RW	hipp_csc_min_y	无符号数 u10.0。	0x000

G0_HIPP_CSC_MIN_C

G0_HIPP_CSC_MIN_C 为 CSC C 分量 CLIP 最小值寄存器。

Offset Address: 0x7148 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留	0x000000
[9:0]	RW	hipp_csc_min_c	无符号数 u10.0。	0x000



G0_HIPP_CSC_MAX_Y

G0_HIPP_CSC_MAX_Y 为 CSC Y 分量 CLIP 最大值寄存器。

Offset Address: 0x714C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留	0x000000
[9:0]	RW	hipp_csc_max_y	无符号数 u10.0。	0x000

G0_HIPP_CSC_MAX_C

G0_HIPP_CSC_MAX_C 为 CSC C 分量 CLIP 最大值寄存器。

Offset Address: 0x7150 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留	0x000000
[9:0]	RW	hipp_csc_max_c	无符号数 u10.0。	0x000

G0_ZME_HINFO

G0_ZME_HINFO 为亮度水平位置偏移寄存器。用于 pan-scan。为非即时寄存器。

Offset Address: 0x7300 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	out_width	水平单眼输出宽度。	0x0000

G0_ZME_HSP

G0_ZME_HSP 为水平缩放参数配置（horizontal Scaling Parameter）寄存器。为非即时寄存器。

缩放比率 = 输入宽度 / 输出宽度。

Offset Address: 0x7304 Total Reset Value: 0x0010_0000



Bits	Access	Name	Description	Reset
[31]	RW	hfir_en	水平亮度缩放使能。 0: 禁止; 1: 使能。	0x0
[30]	RW	ahfir_mid_en	α 值水平缩放中值滤波使能(当 hfir_en 无效时, 该比特不起作用)。 0: 禁止; 1: 使能。	0x0
[29]	RW	lhfir_mid_en	水平缩放中值滤波使能(当 hfir_en 无效时, 该比特不起作用)。 0: 禁止; 1: 使能。	0x0
[28]	RW	chfir_mid_en	水平缩放中值滤波使能(当 hfir_en 无效时, 该比特不起作用)。 0: 禁止; 1: 使能。	0x0
[27]	-	reserved	保留。	0x0
[26]	RW	lhfir_mode	亮度和色度水平缩放模式。 0: 复制模式(滤波不使能); 1: 滤波模式(滤波使能)。	0x0
[25]	RW	ahfir_mode	水平 α 缩放模式。 0: 复制模式(滤波不使能); 1: 滤波模式(滤波使能), 与 Y 分量采用同一滤波模式。	0x0
[24]	-	reserved	保留。	0x0
[23:0]	RW	hratio	水平缩放倍数, (u,4,20)格式。	0x100000

G0_ZME_VINFO

G0_ZME_VINFO 为亮度垂直位置偏移寄存器。用于 pan-scan。为非即时寄存器。

Offset Address: 0x7380 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:19]	-	reserved	保留。	0x0000



[18]	RW	out_pro	垂直缩放输出逐隔行 0: 隔行; 1: 逐行;	0x0
[17:16]	-	reserved	保留。	0x0
[15:0]	RW	out_height	垂直亮度单眼输出高度。	0x0000

G0_ZME_VSP

G0_ZME_VSP 为垂直缩放参数配置（Vertical Scaling Parameter）寄存器。为非即时寄存器。

缩放比率=输入宽度/输出宽度。

Offset Address: 0x7384 Total Reset Value: 0x0000_1000

Bits	Access	Name	Description	Reset
[31]	RW	vfir_en	垂直亮度缩放使能。 0: 禁止; 1: 使能。	0x0
[30]	RW	avfir_mid_en	垂直 α 缩放中值滤波使能 0: 禁止; 1: 使能。	0x0
[29]	RW	lvfir_mid_en	垂直亮度缩放中值滤波使能 0: 禁止; 1: 使能。	0x0
[28]	RW	cvfir_mid_en	垂直色度缩放中值滤波使能 0: 禁止; 1: 使能。	0x0
[27]	RO	reserved	保留。	0x0
[26]	RW	lvfir_mode	亮度和色度垂直缩放模式。 0: 复制模式(滤波不使能); 1: 滤波模式(滤波使能)。	0x0
[25]	RW	vafir_mode	水平 α 缩放垂直模式。 0: 复制模式(滤波不使能); 1: 滤波模式(滤波使能), 与 Y 分量采用同一滤波模式。	0x0
[24:16]	-	reserved	保留。	0x000



Bits	Access	Name	Description	Reset
[15:0]	RW	vratio	垂直缩放倍数, (u,4,12)格式。	0x1000

DHD0_CTRL

DHD0_CTRL 为显示通道的总体控制寄存器。该寄存器的所有比特的配置必须不晚于 DHD0_CTRL.regup 比特, 否则配置无法生效。

Offset Address: 0xD000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	intf_en	显示接口使能。 0: 禁止; 1: 使能。	0x0
[30]	RW	cbar_en	color bar 使能。使能时, 接口输出 color bar。 0: 禁止; 1: 使能。	0x0
[29]	RW	cbar_sel	输出 color bar 色彩空间选择信号, 即时寄存器。 0: VGA; 1: YPbPr。	0x0
[28:1]	-	reserved	保留。	0x00000000
[0]	WC	regup	DHD0 的寄存器更新。当软件配置完该层寄存器后写 1 更新, 更新完成后, 硬件自动清零。	0x0

DHD0_VSYNC1

DHD0_VSYNC1 为显示通道的垂直时序控制寄存器。为隔行输出时, 该寄存器表示顶场垂直同步时序; 逐行输出时, 该寄存器表示帧垂直同步时序。非即时寄存器。

Offset Address: 0xD004 Total Reset Value: 0x0028_0437

Bits	Access	Name	Description	Reset
[31:16]	RW	vbb	隔行输出时: 顶场垂直消隐后肩; 逐行输出时: 垂直消隐后肩加垂直脉冲的宽度。	0x0028



[15:0]	RW	vact	隔行输出时：顶场的活动图象的高度； 逐行输出时：一帧的活动图象的高度。该寄存器值为实际值减 1。	0x0437

DHD0_VSYNC2

DHD0_VSYNC2 为显示通道的垂直时序控制寄存器。为隔行输出时，该寄存器表示顶场垂直同步时序；逐行输出时，该寄存器表示帧垂直同步时序。非即时寄存器。

Offset Address: 0xD008 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	vfb	隔行输出时：顶场垂直消隐前肩； 逐行输出时：垂直消隐前肩。	0x0003

DHD0_HSYNC1

DHD0_HSYNC1 为显示通道的水平时序控制 1 寄存器。隔行或逐行输出情况下，水平同步配置,非即时寄存器。

Offset Address: 0xD00C Total Reset Value: 0x00BF_077F

Bits	Access	Name	Description	Reset
[31:16]	RW	hbb	水平消隐后肩，单位为像素。	0x00BF
[15:0]	RW	hact	活动区域水平像素数。	0x077F

DHD0_HSYNC2

DHD0_HSYNC2 为显示通道的水平时序控制 2 寄存器。隔行或逐行输出情况下，水平同步配置，非即时寄存器。

Offset Address: 0xD010 Total Reset Value: 0x0000_0057

Bits	Access	Name	Description	Reset
[31:16]	RW	hmid	底场垂直同步有效像素值(有效数据区)。	0x0000
[15:0]	RW	hfb	水平消隐前肩，单位为像素。	0x0057



DHD0_VPLUS1

DHD0_VPLUS1 为显示通道隔行输出的底场垂直时序控制寄存器。隔行输出时，该寄存器表示底场垂直同步时序，非即时寄存器。

Offset Address: 0xD014 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	bvbb	隔行输出时：底场垂直消隐后肩垂直脉冲的宽度。	0x0000
[15:0]	RW	bvact	隔行输出时：底场的活动图象的高度。 该寄存器值为实际值减 1。	0x0000

DHD0_VPLUS2

DHD0_VPLUS2 为显示通道隔行输出的底场垂直时序控制寄存器。隔行输出时，该寄存器表示底场垂直同步时序，非即时寄存器。

Offset Address: 0xD018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	bvfb	隔行输出时：底场垂直消隐后肩垂直脉冲的宽度。	0x0000

DHD0_PWR

DHD0_PWR 为同步信号脉冲宽度寄存器，非即时寄存器。

Offset Address: 0xD01C Total Reset Value: 0x0804_002B

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	保留。	0x0
[28:27]	RW	multichn_en	DHD0 接口输出模式选择寄存器。 00：一拍四像素； 01：一拍一像素； 10：一拍二像素； 11：一拍四像素。	0x1
[26:24]	RO	reserved	保留。	0x0
[23:16]	RW	vpw	垂直脉冲宽度减 1。	0x04



[15:0]	RW	hpw	水平脉冲宽度减 1。	0x002B

DHD0_VTTHD

DHD0_VTTHD 为垂直时序门限值（Vertical Timing Threshold），为即时寄存器。该寄存器中包含两个门限配置，可以分别独立的产生两个中断。

Offset Address: 0xD024 Total Reset Value: 0x0001_0001

Bits	Access	Name	Description	Reset
[31]	RW	thd2_mode	门限 2 中断产生的模式。 0: 帧模式，以帧为单位做门限计数； 1: 场模式，在隔行显示时以场为单位做门限计数。	0x0
[30:29]	-	reserved	保留。	0x0
[28:16]	RW	vtmgthd2	垂直时序门限值 2，当垂直时序计数器到达该门限时触发 VOINTSTA[dhdvtthd2_int]中断。	0x0001
[15]	RW	thd1_mode	门限 1 中断产生的模式。 0: 帧模式，以帧为单位做门限计数； 1: 场模式，在隔行显示时以场为单位做门限计数。	0x0
[14:13]	-	reserved	保留。	0x0
[12:0]	RW	vtmgthd1	垂直时序门限值 1，当垂直时序计数器到达该门限时触发 VOINTSTA[dhdvtthd1_int]中断。	0x0001

DHD0_PARATHD

DHD0_PARATHD 为 PARA 系数更新点阈值寄存器。

Offset Address: 0xD028 Total Reset Value: 0x0000_0002

Bits	Access	Name	Description	Reset
[31]	RW	dfs_en	DFS 使能控制位。 0: 禁止； 1: 使能。	0x0
[30:8]	-	reserved	保留。	0x000000



[7:0]	RW	para_thd	PARA 系数更新点阈值寄存器，表示从消隐区开始的第几行产生该更新点。从 0 开始计数。	0x02

DHD0_START_POS

DHD0_START_POS 为 DHD0 通道 start 信号起始位置寄存器。

Offset Address: 0xD030 Total Reset Value: 0x00A3_030A

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x00A
[19:16]	-	reserved	保留。	0x3
[15:8]	RW	timing_start_pos	timing_gen 的状态机在有效区的前几行开始工作。	0x03
[7:0]	RW	start_pos	表示从消隐区开始的第几行产生该 start 信号。从 0 开始计数。	0x0A

DHD0_STATE

DHD0_STATE 为 DHD0 状态寄存器。

Offset Address: 0xD088 Total Reset Value: 0x0100_0006

Bits	Access	Name	Description	Reset
[31:25]	-	reserved	保留。	0x00
[24]	-	reserved	保留。	0x1
[23:16]	RO	count_int	DHD0 中断统计计数器。每上报一次垂直时序中断加 1。	0x00
[15:3]	RO	vcnt	DHD0 显示行有效计数，起始 0 行：垂直前消隐到垂直后消隐为起始 0 行	0x0000
[2]	RO	bottom_field	DHD0 显示顶底场标识。 0：顶场； 1：底场。	0x1
[1]	RO	vblank	DHD0 显示消隐区标识。 0：有效区； 1：消隐区。	0x1



[0]	RO	vback_blank	DHD0 显示后消隐标识。 0: 非后消隐区; 1: 后消隐区。	0x0

VO_MUX

VO_MUX 为 VDP 输出接口复选寄存器。

Offset Address: 0xD090 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:28]	RW	digital_sel	digital 接口数据选择(默认 00)。 00: BT1120; 01: BT656; 10: LCD; 其余: 保留。	0x0
[27:12]	-	reserved	保留。	0x0000
[11:8]	RW	bt_sel	BT1120/656 数据选择(默认 0)。 0: DHD0 BT1120/656; 1: 保留。	0x0
[7:4]	RW	lcd_sel	LCD 数据选择(默认 0)。 0: DHD0 LCD; 1: 保留。	0x0
[3:0]	RW	mipi_sel	MIPI 数据选择(默认 0)。 00: DHD0 MIPI; 01: 保留; 10: DHD0 HDMI 420。	0x0

INTF0_DITHER_CTRL

INTF0_DITHER_CTRL 为 Dither 控制寄存器。

Offset Address: 0xD0B0 Total Reset Value: 0x0000_1500

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12:10]	RW	o_data_width_dither	输出像素数据位宽	0x5



Bits	Access	Name	Description	Reset
			000: 5; 001: 6; 010: 7; 011: 8; 100: 9; 101: 10。	
[9:7]	RW	i_data_width_dither	输入像素数据位宽 000: 8; 001: 9; 010: 10; 011: 11; 100: 12。	0x2
[6]	RW	dither_round_unlim	不在门限范围内的像素四舍五入使能寄存器。 0: 四舍五入关闭; 1: 四舍五入使能。	0x0
[5]	RW	dither_en	Dither 使能选择。 1: 模块使能; 0: 模块不使能, 直接 bypass。	0x0
[4]	RW	dither_mode	Dither 输出模式选择。 0: 12bits->10bits 输出, 丢低位; 1: 12bits->8bits 输出, 丢低位。	0x0
[3]	RW	dither_round	Dither 四舍五入使能寄存器。 0: 四舍五入关闭; 1: 四舍五入使能。	0x0
[2]	RW	dither_domain_mode	Dither 时域和空域模式寄存器。 0: 空域; 1: 时域和空域的组合。	0x0
[1:0]	RW	dither_tap_mode	Dither 周期配置。 00: 周期为 7; 01: 周期为 15; 10: 周期为 23; 11: 周期为 31。	0x0



INTF_BT_UPD

INTF_BT_UPD 为 INTF_BT 更新寄存器。

Offset Address: 0xD204 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	WC	regup	更新寄存器。当软件配置完该层寄存器后写 1 更新，更新完成后，硬件自动清零。	0x0

INTF_BT_SYNC_INV

INTF_BT_SYNC_INV 为外同步时序输出时，同步信号极性配置寄存器。该寄存器配置后立即生效，将马上影响到对应同步信号的极性。

Offset Address: 0xD208 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000
[3]	RW	f_inv	奇偶场指示信号输出反相使能，即时寄存器。 0: 禁止; 1: 使能。	0x0
[2]	RW	vs_inv	垂直同步脉冲输出反相使能，即时寄存器。 0: 禁止; 1: 使能。	0x0
[1]	RW	hs_inv	水平同步脉冲输出反相使能，即时寄存器。 0: 禁止; 1: 使能。	0x0
[0]	RW	dv_inv	数据有效信号输出反相使能，即时寄存器。 0: 禁止; 1: 使能。	0x0

BT_CLIP0_L

BT_CLIP0_L 为 BT.1120 Clip 处理最低门限值寄存器，为即时寄存器。



Offset Address: 0xD210 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	clip_en	clip 使能寄存器。 0: 不使能; 1: 使能。	0x0
[30]	-	reserved	保留。	0x0
[29:20]	RW	clip_cl2	分量 2 最低门限值 Y/R, 无符号整数。	0x000
[19:10]	RW	clip_cl1	分量 1 最低门限值 Cb/G, 无符号整数。	0x000
[9:0]	RW	clip_cl0	分量 0 最低门限值 Cr/B, 无符号整数。	0x000

BT_CLIP0_H

BT_CLIP0_H 为 BT.1120 Clip 处理最高门限值寄存器, 为即时寄存器。

Offset Address: 0xD214 Total Reset Value: 0x3FFF_FFFF

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29:20]	RW	clip_ch2	分量 2 最高门限值 Y/R, 无符号整数。	0x3FF
[19:10]	RW	clip_ch1	分量 1 最高门限值 Cb/G, 无符号整数。	0x3FF
[9:0]	RW	clip_ch0	分量 0 最高门限值 Cr/B, 无符号整数。	0x3FF

INTF_LCD_CTRL

INTF_LCD_CTRL 为接口通道配置寄存器。

Offset Address: 0xD300 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30]	-	reserved	保留。	0x0
[29]	RW	lcd_serial_mode	LCD 串行模式。 0: 并行; 1: 串行。	0x0
[28]	RW	lcd_serial_perd	LCD 串行输出单像素时钟周期数。 0: 3 个周期;	0x0



Bits	Access	Name	Description	Reset
			1: 4 个周期。	
[27]	RW	lcd_parallel_order	LCD 并行输出顺序。 0: RGB; 1: BGR。	0x0
[26]	RW	lcd_data_inv	LCD 输出 bit 反相。 0: 高位到低位 15~0bit; 1: 高位到低位 0~15bit。	0x0
[25]	RW	lcd_parallel_mode	LCD 并行输出模式。 0: RGB565; 1: RGB888。	0x0
[24:0]	RO	reserved	保留。	0x000000

INTF_LCD_UPD

INTF_LCD_UPD 为 INTF_LCD 更新寄存器。

Offset Address: 0xD304 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	WC	regup	更新寄存器。当软件配置完该层寄存器后写 1 更新，更新完成后，硬件自动清零。	0x0

INTF_LCD_SYNC_INV

INTF_LCD_SYNC_INV 为外同步时序输出时，同步信号极性配置寄存器。该寄存器配置后立即生效，将马上影响到对应同步信号的极性。

Offset Address: 0xD308 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000
[3]	RW	f_inv	奇偶场指示信号输出反相使能，即时寄存器。 0: 禁止; 1: 使能。	0x0



[2]	RW	vs_inv	垂直同步脉冲输出反相使能，即时寄存器。 0：禁止； 1：使能。	0x0
[1]	RW	hs_inv	水平同步脉冲输出反相使能，即时寄存器。 0：禁止； 1：使能。	0x0
[0]	RW	dv_inv	数据有效信号输出反相使能，即时寄存器。 0：禁止； 1：使能。	0x0

MAC_OUTSTANDING

MAC_OUTSTANDING 为 OUTSTANDING 寄存器。

Offset Address: 0x10000 Total Reset Value: 0x0000_007F

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:4]	RW	mstr0_woutstanding	master0 写 outstanding。	0x7
[3:0]	RW	mstr0_routstanding	master0 读 outstanding。	0xF

VID_READ_CTRL

VID_READ_CTRL 为 VID FDR 读取控制寄存器，为非即时寄存器。

Offset Address: 0x10200 Total Reset Value: 0x0002_0000

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	保留。	0x000040
[6:4]	RW	lm_rmode	色度逐隔行读取模式寄存器。 0：读取模式跟接口绑定； 1：读取逐行； 2：读取顶场； 3：读取底场。 4~7：保留。	0x0



Bits	Access	Name	Description	Reset
[3]	-	reserved	保留。	0x0
[2:0]	RW	chm_rmode	色度逐隔行读取模式寄存器。 0: 读取模式跟接口绑定; 1: 读取逐行; 2: 读取顶场; 3: 读取底场。 4~7: 保留。	0x0

VID_OUT_CTRL

VID_OUT_CTRL 为 VID FDR 输出控制寄存器，为非即时寄存器。

Offset Address: 0x10210 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	uv_order_en	输入 YUV 数据格式的 UV 顺序寄存器。 0: U 在高位; 1: V 在高位。	0x0
[3:0]	RO	reserved	保留。	0x0

VID_SRC_INFO

VID_SRC_INFO 为 VID FDR 输入图像属性寄存器，为非即时寄存器。

Offset Address: 0x10240 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:21]	-	reserved	保留。	0x000
[20]	RW	dcmp_en	输入图像压缩类型: 0: 非压缩; 1: 压缩。	0x0
[19:5]	-	reserved	保留。	0x0000
[4:3]	RW	data_fmt	输入数据排布方式。 00: LINEAR; 01: TILE;	0x0



Bits	Access	Name	Description	Reset
			10: PKG; 11: 保留, 等同于 0。	
[2:0]	RW	data_type	输入数据格式类型: 000: 保留, 默认 422; 001: YUV400; 010: YUV420; 011: YUV422; 100: YUV444; 101: RGB888; 110: ARGB8888; 111: 保留, 等同于 0。	0x0

VID_SRC_RESO

VID_SRC_RESO 为原始图像分辨率, 非及时寄存器。

Offset Address: 0x10244 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	src_h	高度, 行为单位。实际高度减 1。 以帧高度为参考, 以行为单位。	0x0000
[15:0]	RW	src_w	宽度, 像素单位。实际宽度减 1。	0x0000

VID_SRC_CROP

VID_SRC_CROP 为源图的 CROP 寄存器, 非及时寄存器。

Offset Address: 0x10248 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	src_crop_y	高度, 行为单位, 为实际 Crop 值, 不需要减 1。	0x0000
[15:0]	RW	src_crop_x	宽度, 像素单位, 为实际 Crop 值, 不需要减 1。	0x0000



VID_IN_RESO

VID_IN_RESO 为输入图像分辨率寄存器，为非即时寄存器。

Offset Address: 0x1024C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	ireso_h	高度，行为单位。实际高度减 1。 以帧高度为参考，以行为单位。	0x0000
[15:0]	RW	ireso_w	宽度，像素单位。实际宽度减 1。	0x0000

ICE_V3R2_SEG_DCMP_GLB_INFO

ICE_V3R2_SEG_DCMP_GLB_INFO 为 DCMP 控制寄存器

Offset Address: 0x10380 Total Reset Value: 0x0000_0005

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留	0x00000
[2]	RW	cmp_mode	压缩模式： 0: 压缩； 1: raw 模式；	0x1
[1]	RW	is_lossless	是否无损压缩： 0: 有损定带宽模式； 1: 无损压缩模式；	0x0
[0]	RW	ice_en	cmp 使能。 0:不使能； 1:使能。	0x1

ICE_V3R2_SEG_DCMP_GLB_INFO_CHM

ICE_V3R2_SEG_DCMP_GLB_INFO_CHM 为 DCMP 控制寄存器。

Offset Address: 0x103C0 Total Reset Value: 0x0000_0005

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留	0x00000



[2]	RW	cmp_mode	压缩模式： 0: 压缩； 1: raw 模式；	0x1
[1]	RW	is_lossless	是否无损压缩： 0: 有损定带宽模式； 1: 无损压缩模式；	0x0
[0]	RW	ice_en	cmp 使能。 0: 不使能； 1: 使能。	0x1

GFX_OUT_CTRL

GFX_OUT_CTRL 为 GFX FDR 输出控制寄存器，为非即时寄存器。

Offset Address: 0x10E08 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留。	0x00000
[10]	RW	premulti_en	图形层预乘使能寄存器。 0: 使能； 1: 禁止。	0x0
[9:8]	RW	bitext	层输入位图 Bit 位扩展模式。 00: 低位扩展 0； 10: 低位扩展最高 bit 位； 11: 低位扩展最高若干 bit 位。	0x0
[7:6]	-	reserved	保留。	0x0
[5]	RW	key_en	color key 使能。 0: 禁止； 1: 使能。	0x0
[4]	RW	key_mode	color key 模式。 0: 满足 $\text{Keymin} \leq \text{Pixel} \leq \text{Keymax}$ 时， 处理为关键色； 1: 满足 $\text{Pixel} \leq \text{Keymin}$ 或者 $\text{Pixel} \geq \text{Keymax}$ 时处理为关键色。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	RW	palpha_en	像素 alpha 使能。	0x0



Bits	Access	Name	Description	Reset
			0: 禁止; 1: 使能。	
[0]	RW	palpha_range	像素的 alpha 范围选择。 0: 表示像素的 alpha 范围为 0~128; 1: 表示像素的 alpha 范围为 0~255。	0x0

GFX_1555_ALPHA

GFX_1555_ALPHA 为 ARGB1555 格式 alpha 值，为非即时寄存器。

Offset Address: 0x10E28 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:8]	RW	alpha_1	alpha1 值。当数据格式为 alphaRGB1555 时，alpha 值为 1 时，用该值替换。	0x00
[7:0]	RW	alpha_0	alpha0 值。当数据格式为 alphaRGB1555 时，alpha 值为 1 时，用该值替换。	0x00

GFX_SRC_INFO

GFX_SRC_INFO 为 GFX FDR 源图像属性寄存器，为非即时寄存器。

Offset Address: 0x10E40 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:21]	-	reserved	保留。	0x000
[20]	RW	dcmp_en	输入图像压缩类型： 0: 非压缩; 1: 压缩。	0x0
[19:8]	RO	reserved	保留。	0x000
[7:0]	RW	ifmt	输入数据格式。 0x00: CLUT 1bpp; 0x10: CLUT 2bpp; 0x20: CLUT 4bpp; 0x30: CLUT 8bpp; 0x38: ACLUT44;	0x00



Bits	Access	Name	Description	Reset
			0x40: RGB444; 0x41: RGB555; 0x42: RGB565; 0x43: PackageUYVY(Cb0Y0Cr0Y1); 0x44: PackageYUYV(Y0Cb0Y1Cr0); 0x45: PackageYVYU(Y0Cr0Y1Cb0); 0x46: ACLUT88; 0x48: ARGB4444; 0x49: ARGB1555; 0x50: RGB888(24bpp); 0x51: YUV888(24bpp); 0x5a: ARGB8565(24bpp); 0x60: KRGB888; 0x68: ARGB8888; 0x69: AYUV8888; 0xc8: RGBA4444; 0xc9: RGBA5551; 0xda: RGBA5658(24bpp); 0xe8: RGBA8888; 0xe9: YUVA8888; 0xef: ABGR8888; 其他: 保留。	

GFX_SRC_RESO

GFX_SRC_RESO 为原始图像分辨率，非及时寄存器。

Offset Address: 0x10E44 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	src_h	高度，行为单位。实际高度减 1。 以帧高度为参考，以行为单位。	0x0000
[15:0]	RW	src_w	宽度，像素单位。实际宽度减 1。	0x0000

GFX_SRC_CROP

GFX_SRC_CROP 为源图的 CROP 寄存器，非及时寄存器。



Offset Address: 0x10E48 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	src_crop_y	高度，行为单位，为源图上方 Crop 值，不需要减 1。	0x0000
[15:0]	RW	src_crop_x	宽度，像素单位，为源图左边 Crop 值，不需要减 1(涵盖源坐标偏移功能)。	0x0000

GFX_IRESO

GFX_IRESO 为输入分辨率寄存器，为非即时寄存器。

Offset Address: 0x10E4C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	ireso_h	高度，行为单位。实际高度减 1。 注意：层的实际高度必须是偶数。	0x0000
[15:0]	RW	ireso_w	宽度，像素单位。实际宽度减 1。 注意：层的实际宽度必须是偶数。	0x0000

GFX_CKEY_MAX

GFX_CKEY_MAX 为 color key 最大值，为非即时寄存器。

Offset Address: 0x10F00 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:16]	RW	key_r_max	color key R 分量最大值。	0x00
[15:8]	RW	key_g_max	color key G 分量最大值。	0x00
[7:0]	RW	key_b_max	color key B 分量最大值。	0x00

GFX_CKEY_MIN

GFX_CKEY_MIN 为 color key 最小值，为非即时寄存器。

Offset Address: 0x10F04 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:16]	RW	key_r_min	color key R 分量最小值。	0x00
[15:8]	RW	key_g_min	color key G 分量最小值。	0x00
[7:0]	RW	key_b_min	color key B 分量最小值。	0x00

GFX_CKEY_MASK

GFX_CKEY_MASK 为 color key mask 值，为非即时寄存器。相应 bit 为 1 表示在 Key 的比较过程中，pixel 的相应 bit 不变；相应 bit 为 0 表示在 Key 的比较过程中，pixel 的相应 bit 不管是 0 还是 1，都强制设置为 0。

Offset Address: 0x10F08 Total Reset Value: 0x00FF_FFFF

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:16]	RW	key_r_msk	color key mask 的 R 分量。	0xFF
[15:8]	RW	key_g_msk	color key mask 的 G 分量。	0xFF
[7:0]	RW	key_b_msk	color key mask 的 B 分量。	0xFF

9.3 MIPI Rx

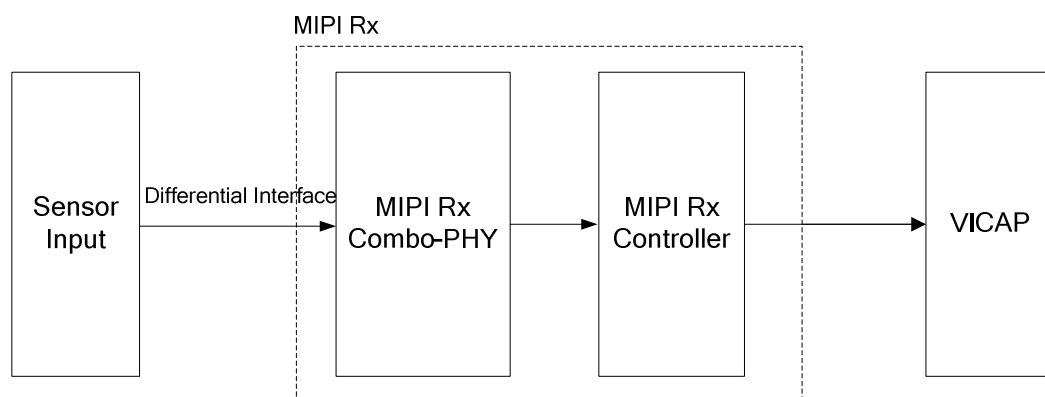
9.3.1 概述

移动行业处理器接口 MIPI Rx（Mobile Industry Processor Interface Receiver），通过低电压差分信号接收原始视频数据（BAYER RGB 数据），并将其转化为 DC（Digital Camera）时序后传递给下一级模块 VICAP（Video Capture）。

MIPI Rx 支持 MIPI D-PHY、LVDS（Low-Voltage Differential Signal）、HiSPi（High-Speed Serial Pixel Interface）等串行视频信号输入，串行视频接口可以提供更高的传输带宽，增强传输的稳定性。

MIPI Rx 包含 combo-PHY 和 Controller 两部分，功能框图及在系统中的位置如图 9-18 所示。

图9-18 MIPI Rx 功能框图及在系统中的位置



9.3.2 特点



说明

以下章节会多次出现 Lane、Link、通道的概念。

- Lane 指差分数据对，MIPI_Rx 最多有 4 条 Lane。
- Link 指 Lane 的分组，每个分组中包含 4 对数据，MIPI_Rx 有 1 个 Link。
- 通道指 MIPI_Rx 内部的数据处理通道，对应连接到 VICAP 通道，每个通道单独处理一个 Sensor 的数据。

MIPI Rx 有以下特点：

- 支持 MIPI DPHY-ver1.1
- 可同时支持 2 路 sensor 输入
- 单路最多支持 4-Lane MIPI D-PHY 接口，最大支持 1.5Gbps/Lane
- 单路最多支持 4-Lane LVDS/ sub-LVDS /HiSPi 接口，最大支持 1.5Gbps/Lane
- 支持 RAW8/ RAW10/ RAW12/ RAW14/ RAW16 数据类型的解析
- 支持 YUV420 8-bit/ YUV422 8-bit/ 数据类型的解析
- 最多支持 2 帧 WDR，支持多种 WDR 时序
- 支持 LVDS/HiSPi 模式像素/同步码大小端配置
- 支持 Lane 数和 Lane 顺序可配置

9.3.3 功能描述

9.3.3.1 典型应用

MIPI Rx 是一个支持多种差分视频输入接口的采集单元，主要功能是接口时序的转换。通过不同的功能配置，MIPI Rx 可以接收 MIPI/LVDS/sub-LVDS/HiSPi 接口的数据，并且支持多种速度和分辨率的传输需求，兼容多种图像传感器。

MIPI Rx 包括 1 个 D-PHY，有 2 对差分随路时钟，每对时钟对应 4 对数据。因此 MPI Rx 可以同时支持 2 路 Sensor 输入。



- 当采用 Mode0 时，最大支持 4Lane 输入
Sensor0 的时钟对接到 PHY_CLK0
- 当采用 Mode1 时，支持 2Lane+2Lane 输入
 - Sensor0 的时钟对接到 PHY_CLK0，数据对接到 lane0/lane2
 - Sensor1 的时钟对接到 PHY_CLK1，数据对接到 lane1/lane3

不同模式下，时钟的接法不同，时钟的共享方式也不同，可以通过寄存器配置时钟的来源与共享方式。详细配置说明请参见 [9.3.4.1 MIPI 模式配置流程](#)和 [9.3.4.2 LVDS 和 HiSPi 模式配置流程](#)，只要 PHY 的时钟是独立外接的，则可以单独对接 Sensor。

说明

MIPI/LVDS 支持 Lane 数和 Lane 顺序可配置，但有一定的限制，Lane 的乱序只能限制在同一个 sensor 内，不同 sensor 间之间的 lane 没有相关性。例如：对于 mode0，为 4lane 模式，当接 sensor0 时，所有 lane 都可以乱序。而 mode1，sensor0 接 lane0/2，sensor1 接 lane1/3，0/2 可以乱序，1/3 可以乱序。

MIPI Rx 支持的接口类型如表 9-7 所示。

表9-7 MIPI Rx 支持的接口类型

Interface Type	Common mode voltage	Differential mode voltage	Maximum clock frequency	Maximum data rate per lane
MIPI D-PHY	200mV	200mV	750MHz	1.5Gbps
sub-LVDS	900mV	150mV	750MHz	1.5Gbps
LVDS	1.25V	350mV	750MHz	1.5Gbps
HiSPi(HiVCM)	900mV	280mV	750MHz	1.5Gbps
HiSPi(SLVS)	200mV	200mV	750MHz	1.5Gbps

MIPI Rx 只完成接口的时序转换，不处理图像的数据格式。在满足带宽的前提下可以支持任意分辨率和帧率。MIPI Rx 的带宽有两部分限制：combo-PHY 的接口数据率和内部处理速度。两路 Sensor 输入接口都是最大支持 1.5Gbps/Lane，内部处理速度最大 300M*1pixels/s。

说明

Combo-PHY 支持的最大速率为 1.5Gbps/Lane，总共支持 4Lane 同时传输。对于 mipi_rx 控制器的来说，通道的最大工作时钟为 300MHz。最大通道带宽为 300Mpixels/s。

9.3.3.2 功能原理

MIPI 接口数据格式

MIPI 规范由不同的工作组负责开发和维护，涵盖了不同的应用需求，分别面向不同领域的应用。MIPI Rx 支持 D-PHY 和 CSI-2 (Camera Serial Interface)。D-PHY 规定了物理层传输规范，CSI-2 规定了 Camera 输出数据包的格式和协议。

- D-PHY

D-PHY 是 MIPI 联盟发布的高速物理层标准，规定了主机和外设的物理层的物理特性及传输协议。D-PHY 采用 200mV 源同步的低压差分信令技术，每个 Lane 的数据率范围为 80~1500Mbps。D-PHY 可以工作在低功耗（Low Power, LP）和高速（High Speed, HS）两种模式下。

- CSI-2

CSI-2 是针对摄像头的数据协议，规定了主机与外设通信的数据包格式。

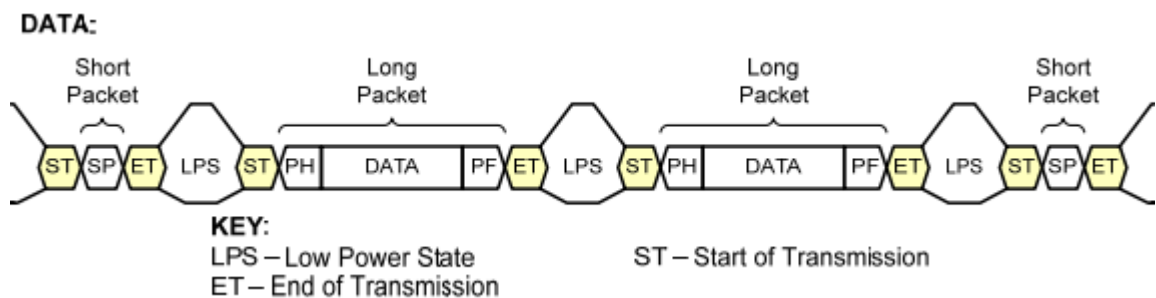
CSI-2 可以支持不同像素格式的图像应用，数据传输的最小粒度是字节。为增加 CSI-2 性能，可以选择数据 Lane 的数量，CSI-2 协议规定了发送端将像素数据打包成字节的机制，并指明多个数据 Lane 分配和管理的方式。字节数据以数据包的形式组织，数据包在 SoT 和 EoT 之间传输。接收端根据协议解析相应的数据包，恢复出原始的像素数据。

MIPI Rx 支持 RAW8/RAW10/RAW12/RAW14/RAW16 格式的像素数据解析。支持 YUV420 8-bit/ YUV422 8-bit/数据类型的解析。

CSI-2 的数据包分为长包和短包两种，包含有校验码，能进行误码纠正和错误检测。

长包和短包都是在 SoT 和 EoT 之间传输，在数据传送的间隙，D-PHY 处于 LP 模式。CSI-2 数据包的传输机制如图 9-19 所示。PH 和 PF 分别表示 Packet Header 和 Packet Footer。

图9-19 CSI-2 数据包的传输机制



长包用于传输有效像素数据，分为五部分：Data ID, Word Count, ECC, PAYLOAD, Checksum。

Data ID 包含 Virtual Channel 和 Data Type。Virtual Channel 控制传输所用的通道，可以指定通道的复用，让不同的通道传输不同的数据。Data Type 指定数据的类型。

Word Count 指示接收端需要接收到的数据量。

ECC 是 8 位纠错码，可以纠正或检测 Data Type 和 Word Count 的误码。

Payload Data 是需要传输的像素数据。

Checksum 是利用线性反馈移位寄存器产生的校验和，用于 Payload 数据的校验。

长包的结构如图 9-20 所示。

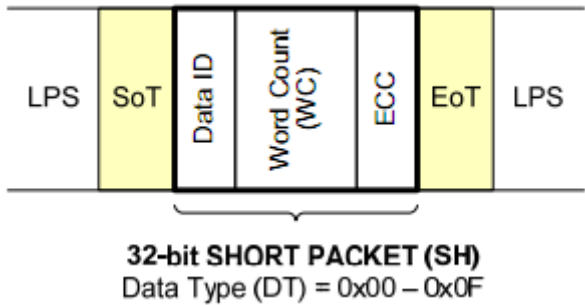


图9-20 CSI-2 长包的格式



短包的作用是传输同步信息，包含 Data ID, Word Count 和 ECC 三部分。其格式如图 9-21 所示。

图9-21 CSI-2 短包的格式

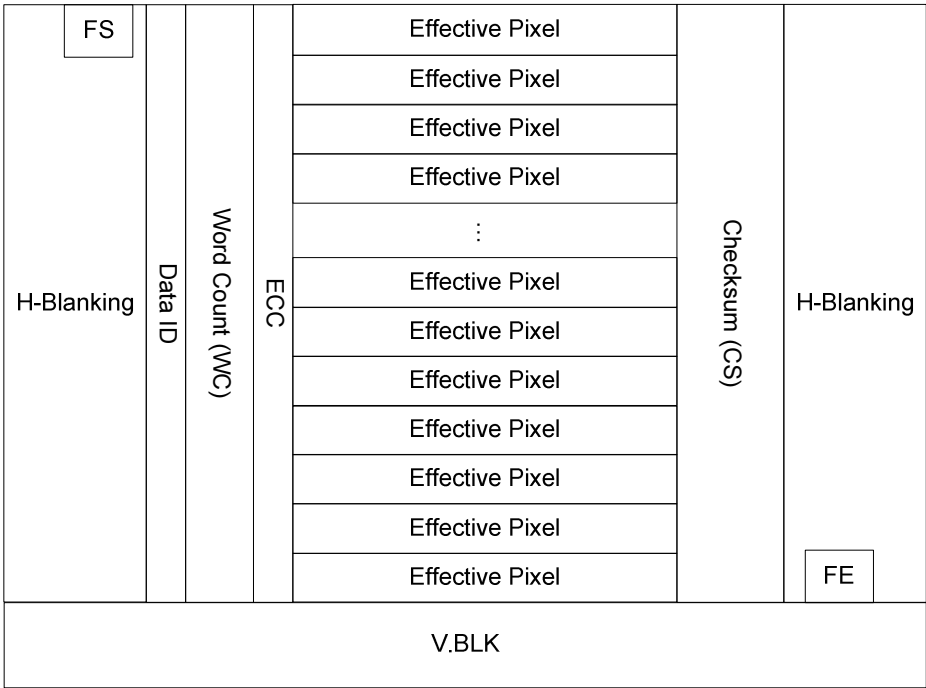


MIPI 接口线性模式

MIPI 接口的线性模式视频传输格式如图 9-22 所示。Frame Start(FS)表示帧起始，Frame End(FE)表示帧结束。每一行的数据包头是 32-bit，包含当前行的 Virtual Channel 和 Data Type 等信息。



图9-22 MIPI 接口图像格式



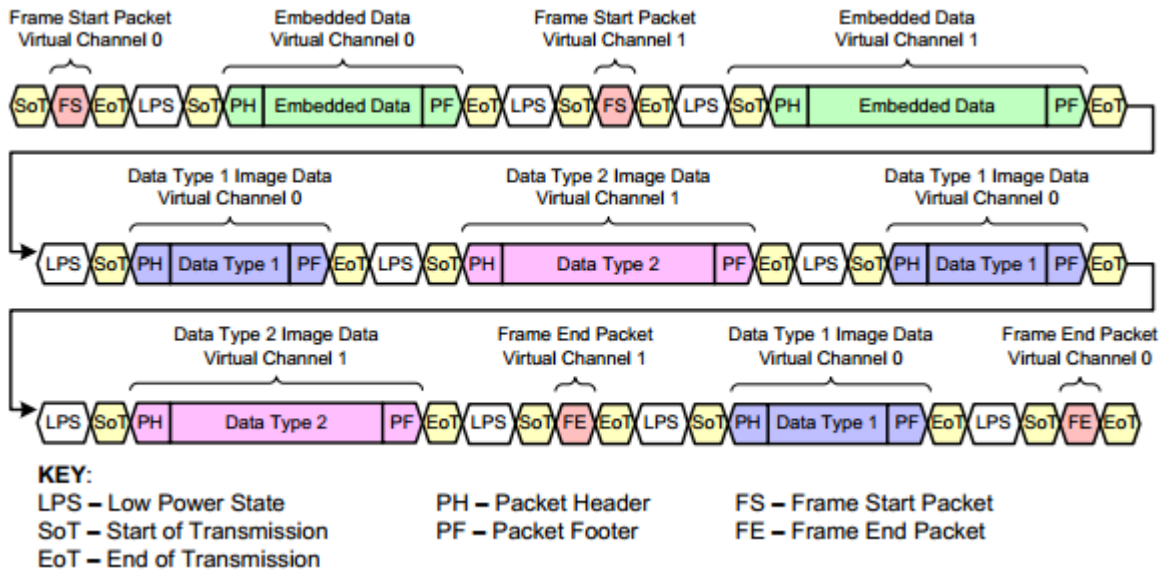
MIPI 接口宽动态模式

MIPI Rx 支持 3 种 MIPI 接口的宽动态（WDR）模式，分别为：

- 1) MIPI 宽动态模式 1：使用 VC（Virtual Channel）区分长/短曝光数据。
- 2) MIPI 宽动态模式 2：使用 DT（Data Type）区分长/短曝光数据。
- 3) MIPI 宽动态模式 3：DOL 宽动态传输格式。

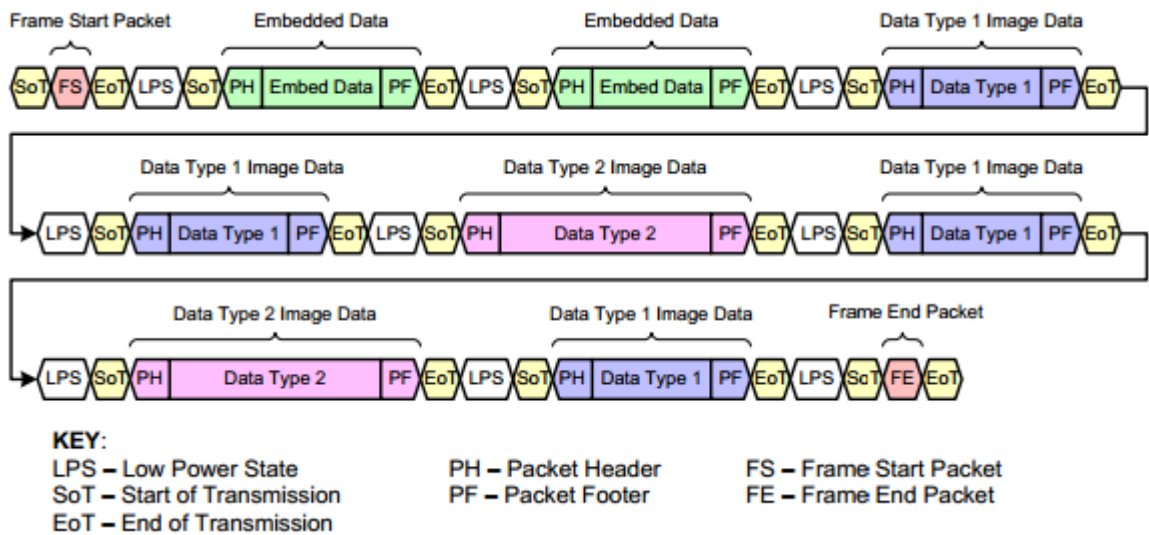
使用 VC 的 WDR 传输方式如[图 9-23](#)所示，FS/FE 短包和数据长包里都包含有 VC 信息。MIPI Rx 控制器根据短包/长包里的 VC 值检测出长/短曝光数据，并把每一行数据都加上 2-bit 信号标识，伴随 DC 时序输出到 VICAP，在 VICAP 内实现解复用。

图9-23 MIPI 接口宽动态数据传输（使用 VC）



使用 DT 的 WDR 传输方式如图 9-24 所示，不同曝光长度的同一帧共用一组 FS/FE 短包，长包的包头有 DT 信息。DT0 和 DT1 的 RAW DATA 位宽可以不同。MIPI Rx 控制器根据 DT 值区分出长/短曝光数据，并把每一行数据都加上 2-bit 信号标识，伴随 DC 时序输出到 VICAP，在 VICAP 内实现解复用。

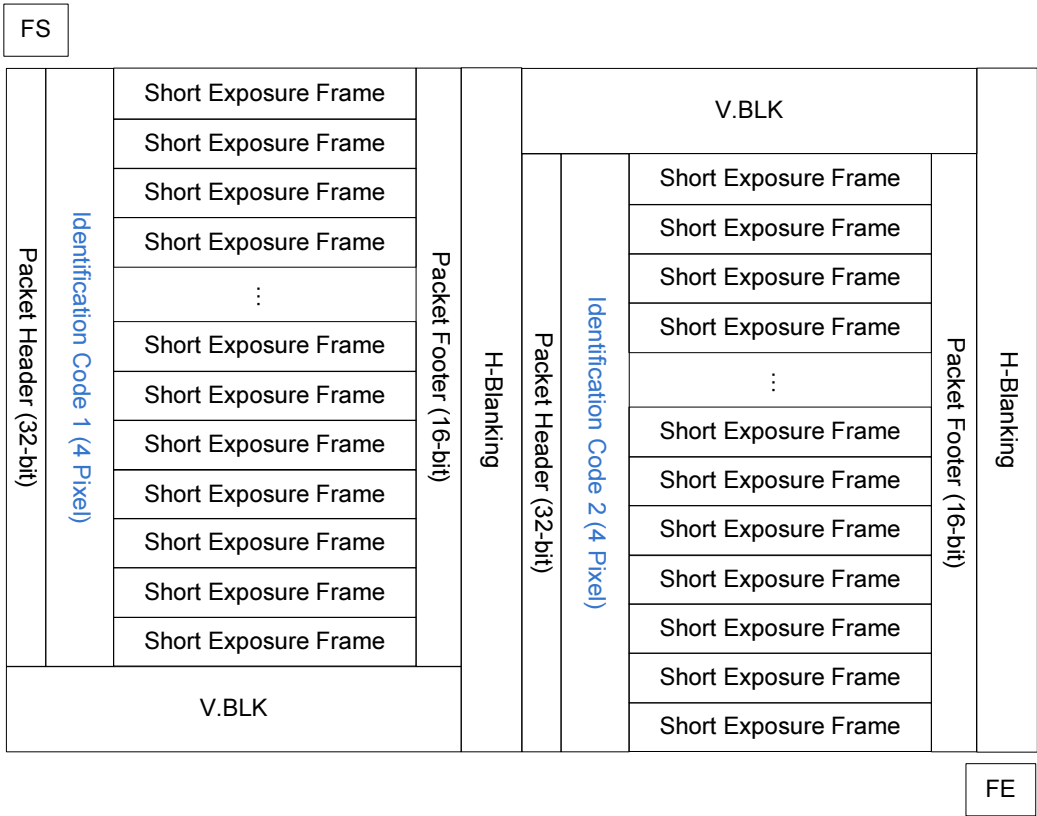
图9-24 MIPI 接口宽动态数据传输（使用 DT）



MIPI DOL 宽动态传输格式如图 9-25 所示。长短曝光数据共用一组 FS/FE 短包，每行起始的 4 个像素点作为识别码（Identification Code）用于区分长/短曝光的数据。



图9-25 MIPI DOL 数据传输



说明

图像传感器有两种宽动态模式，帧模式与行模式。帧模式的宽动态在时序和数据格式上与线性模式相同，这里介绍的宽动态都是针对行模式。

LVDS 接口数据格式

低电压差分信号 LVDS（Low-Voltage Differential Signal）普遍应用于前端摄像头。通过同步码区分消隐区和有效区的数据。



说明

LVDS 只约束了电气传输规范，没有时序和数据格式方面的标准协议。sub-LVDS 是一种超低压摆幅的差分信号技术，共模/差模电压比 LVDS 更低，更适合图像传感器的应用。可以把 sub-LVDS 看作是 LVDS 的一种，在此统称为 LVDS。

MIPI Rx 的 combo-PHY 将差分串行数据转换为并行数据，MIPI Rx 控制器把并行数据拆分、拼接，然后提取同步码、解析出像素数据。

在 LVDS 传输模式中，行场同步信号集成在数据流中，在数据流中的特殊码型 SOF 和 EOF 分别表示帧的起始和结束，SOL 和 EOL 分别表示行的起始和结束。在数据流中，SOF/EOF/SOL/EOL 由 4 个字段构成，每个字段的位宽与像素数据保持一致，前 3 个字段为固定的基准码字，根据第 4 个字段来区分帧/行的起始或结束。LVDS 同步码格式如表 9-8 所示。



表9-8 LVDS 同步码格式

Field	Bit Width	Sync code			
		SOL/SAV (Valid line)	EOL/EAV (Valid line)	SOF/SAV (Invalid line)	EOF/EAV (Invalid line)
1st code	8bit	FFh	FFh	FFh	FFh
	10bit	3FFh	3FFh	3FFh	3FFh
	12bit	FFFh	FFFh	FFFh	FFFh
	14bit	3FFFh	3FFFh	3FFFh	3FFFh
	16bit	FFFFh	FFFFh	FFFFh	FFFFh
2nd code	8bit	00h	00h	00h	00h
	10bit	000h	000h	000h	000h
	12bit	000h	000h	000h	000h
	14bit	0000h	0000h	0000h	0000h
	16bit	0000h	0000h	0000h	0000h
3rd code	8bit	00h	00h	00h	00h
	10bit	000h	000h	000h	000h
	12bit	000h	000h	000h	000h
	14bit	0000h	0000h	0000h	0000h
	16bit	0000h	0000h	0000h	0000h
4th code	8bit	XXh	XXh	XXh	XXh
	10bit	XXXh	XXXh	XXXh	XXXh
	12bit	XXXh	XXXh	XXXh	XXXh
	14bit	XXXXh	XXXXh	XXXXh	XXXXh
	16bit	XXXXh	XXXXh	XXXXh	XXXXh



说明

同步码前三个字段固定，第 4 个字段标识行场的起始或结束。第 4 个字段的值由图像传感器厂商确定，不同的厂商会使用不同的数值。表 9-9 是其中一种实现方式。

表9-9 LVDS 同步码第 4 个字段举例

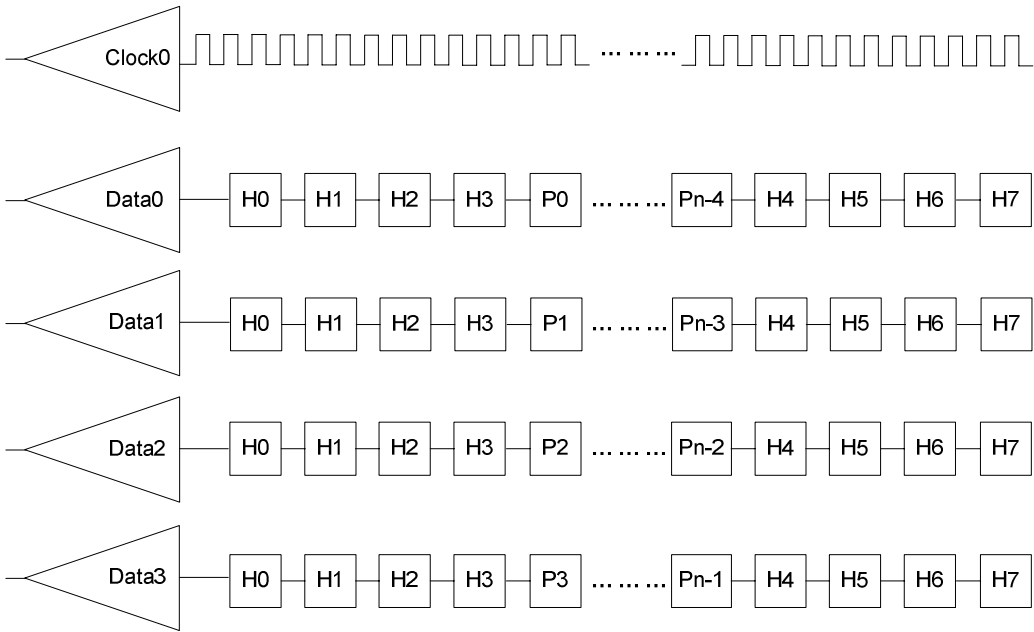
Field	Bit Width	Sync code			
		SAV(Valid line)	EAV(Valid line)	SAV(Invalid line)	EAV(Invalid line)
4th	8bit	80h	9Dh	ABh	B6h



Field	Bit Width	Sync code			
		SAV(Valid line)	EAV(Valid line)	SAV(Invalid line)	EAV(Invalid line)
code	10bit	200h	274h	2ACh	2D8h
	12bit	800h	9D0h	AB0h	B60h
	14bit	2000h	2740h	2AC0h	2D80h
	16bit	8000h	9D00h	AB00h	B600h

以 4 个 Lane 为例，LVDS 同步码和像素数据在各个 Lane 上传输方式如图 9-26 所示，图中 H 表示同步码，P 表示像素，H 和 P 的位宽与图像传感器输出单个像素的位宽一致。各个数据通道首先传输 4 个像素位宽的同步码，紧接着是像素数据，像素数据的分布与通道数有关。

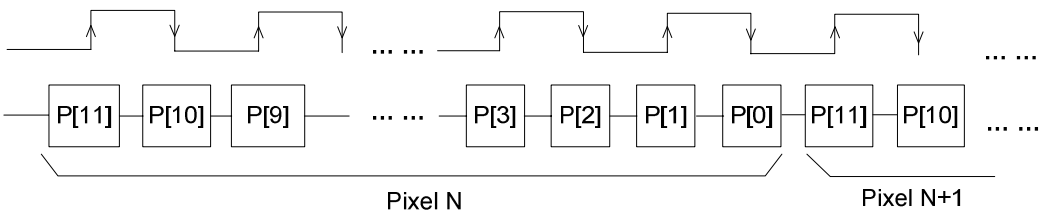
图9-26 LVDS 同步码和图像传输模式



同步码和像素数据的传输是串行的，MIPI Rx 支持数据的大小端可配置。以 RAW12、大端模式为例，图像传感器输出单个像素点的时序如图 9-27 所示。



图9-27 LVDS 单个像素点时序



LVDS 接口线性模式

LVDS 有两种同步方式，其中一种用 SAV(Invalid)和 EAV(Invalid)标识消隐区的无效数据，使用 SAV(Valid)和 EAV(Valid)标识有效区像素数据，这种同步方式如图 9-28 所示。

图9-28 LVDS 同步方式 1

H.BLK	SAV (Invalid line)	V.BLK	EAV (Invalid line)	H.BLK
H.BLK		V.BLK		H.BLK
H.BLK		V.BLK		H.BLK
H.BLK	SAV (Valid line)	Effective Pixel	EAV (Valid line)	H.BLK
H.BLK		Effective Pixel		H.BLK
⋮		⋮		⋮
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK	SAV (Invalid line)	V.BLK	EAV (Invalid line)	H.BLK
⋮		⋮		⋮
H.BLK		V.BLK		H.BLK
H.BLK		V.BLK		H.BLK

另一种同步方式用 SOF 标识有效区的第一行起始，用 EOF 标识有效区最后一行的结束，其他有效区分别用 SOL 和 EOL 作为起始和结束，这种同步方式如图 9-29 所示。



图9-29 LVDS 同步方式 2

V.BLK				
H.BLK	SOF	Effective Pixel	EOL	H.BLK
H.BLK	SOL	Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
⋮		⋮		⋮
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel	EOF	H.BLK
V.BLK				

LVDS 接口宽动态模式

MIPI Rx 支持 3 种 LVDS 的宽动态传输模式，分别为：

- 1) LVDS 宽动态模式 1: SOF-EOF 标识，长短曝光帧有独立的同步码。
- 2) LVDS 宽动态模式 2: SAV-EAV 标识，DOL 模式，四个字段的同步码，长短曝光帧有独立的同步码。
- 3) LVDS 宽动态模式 3: SAV-EAV 标识，DOL 模式，五个字段的同步码，长短曝光帧有独立的同步码。

MIPI Rx 根据不同的模式配置，将接收到的同步码的值与寄存器中的预设值比对，从而区分出接收到的图像属于长/短曝光并以特定的方式将其标识出，在 VICAP 内实现解复用。

LVDS 宽动态模式 1 如图 9-30 所示。长/短曝光视频数据同步码的码型不同，利用同步码区分不同曝光长度的数据。



图9-30 LVDS 宽动态模式 1（2 帧）

SOF_0	Long Frame Line 1	EOL_0	H.BL K	V.BLK				
SOL_0	Long Frame Line 2			SOL_1	SOF_1	Short Frame Line 1	EOL_1	H.BL K
	Long Frame Line 3				Short Frame Line 2			
	Long Frame Line 4				Short Frame Line 3			
	⋮				Short Frame Line 4			
	Long Frame Line n-6				⋮			
	Long Frame Line n-5				Short Frame Line n-6			
	Long Frame Line n-4				Short Frame Line n-5			
	Long Frame Line n-3				Short Frame Line n-4			
	Long Frame Line n-2				Short Frame Line n-3			
	Long Frame Line n-1				Short Frame Line n-2			
	Long Frame Line n				Short Frame Line n-1			
EOF_0			Short Frame Line n	EOF_1				
V.BLK				V.BLK				

LVDS 宽动态模式 2 如图 9-31 所示。使用 SAV-EAV 作为同步方式，长短曝光有独立的同步码，第 N 帧和第 N+1 帧的同步码也不同。

LVDS 宽动态模式 3 与模式 2 基本相同，区别仅仅在于模式 3 的同步码为 5 个字段。

图9-31 LVDS 宽动态模式 2（2 帧）

Horizontal Blanking	SAV_0	blanking	EAV_0	Horizontal Blanking	SAV_0	blanking	EAV_0	Horizontal Blanking	Frame N
	SAV_2	Long N	EAV_2		SAV_3	Short N	EAV_3		
	SAV_0	blanking	EAV_0		SAV_0	blanking	EAV_0		
	SAV_1	blanking	EAV_1		SAV_1	blanking	EAV_1	Horizontal Blanking	Frame N+1
	SAV_4	Long N+1	EAV_4		SAV_5	Short N+1	EAV_5		
	SAV_1	blanking	EAV_1		SAV_1	blanking	EAV_1		
	SAV_1	blanking	EAV_1		SAV_1	blanking	EAV_1		

HiSPi 接口数据格式

High-Speed Serial Pixel (HiSPi)协议包括两部分：HiSPi 物理层协议和 HiSPi 规范。HiSPi 物理层协议中规定了电气特性和时序等参数，HiSPi 规范中规定了数据的打包方式。



HiSPi 规范中包含两种物理层电气标准和四种不同的数据传输方式。电气标准分别为：HiVCM 和 SLVS；数据传输方式分别为：Packetized-SP、Streaming-SP、Streaming-S 和 ActiveStart-SP8。

MIPI Rx 同时支持 HiVCM 和 SLVS 电气标准，支持 Packetized-SP、ActiveStart-SP8 和 Streaming-SP 数据传输方式。

HiSPi 接口线性模式

Packetized-SP 模式下，图像传感器用 SOF 标识图像有效区的第一行起始，用 EOF 标识有效区最后一行的结束，其他有效区分别用 SOL 和 EOL 作为起始和结束。这种同步方式与图 9-29 类似，区别在于 Packetized-SP 模式的数据包内可以添加 CRC 和 FLR。其数据格式如图 9-32 所示。

MIPI Rx 可以校验 SOF-EOF、SOL-EOL 的配对是否正常，但不处理 CRC 和 FLR 数据。

图9-32 HiSPi Packetized-SP 数据格式

SOF		Effective Pixel			
SOL	FLR	Effective Pixel	EOL	CRC	H.BLK
		Effective Pixel			
		Effective Pixel			
		Effective Pixel			
		⋮			
		Effective Pixel			
		Effective Pixel			
		Effective Pixel			
		Effective Pixel			
		Effective Pixel			
		Effective Pixel	EOF		
V.BLK					

HiSPi 另外 2 种模式（Streaming-SP, ActiveStart-SP8）的传输方式与 Packetized-SP 类似，区别在于同步码的标定方式不同。他们之间的区别如表 9-10 所示。其中，SOF 和 EOF 分别表示图像有效区的帧起始和帧结束；SOL 和 EOL 分别表示图像有效区的行起始和行结束；SAV 表示图像消隐区的行起始。

表9-10 HiSPi 传输模式

同步码	Packetized-SP	Streaming-SP	ActiveStart-SP8
SOF	Required	Required	Required
SOL	Required	Required	Required
EOF	Required	Unsupported	Unsupported



同步码	Packetized-SP	Streaming-SP	ActiveStart-SP8
EOL	Required	Unsupported	Unsupported
SAV	Unsupported	Required	Unsupported

MIPI Rx 可以支持以上 3 种模式，同时支持所有通道都传输 4 个字段的同步码（如图 9-26）。

HiSPi 接口宽动态模式

HiSPi 接口宽动态模式的同步码与线性模式相同，长曝光与短曝光数据之间是消隐区。短曝光的起始几行不是有效像素区，而是以固定值填充。HiSPi 接口宽动态模式的时序如所示。

图9-33 HiSPi 宽动态模式

SOF	Long Frame Line 1	EOL	H.BLK	SOL	T2 Padding	EOL	H.BLK
SOL	Long Frame Line 2				Short Frame Line 1		
	Long Frame Line 3				Short Frame Line 2		
	Long Frame Line 4				Short Frame Line 3		
	⋮				Short Frame Line 4		
	Long Frame Line n-3				⋮		
	Long Frame Line n-2				Short Frame Line n-3		
	Long Frame Line n-1				Short Frame Line n-2		
	Long Frame Line n				Short Frame Line n-1		
	T1 Padding				Short Frame Line n	EOF	
SOV	V.BLK	EOV		SOV	V.BLK	EOV	

9.3.4 MIPI Rx 控制器工作方式

MIPI Rx 控制器支持 MIPI、LVDS 和 HiSPi 模式，每种模式下的软件配置包含两部分：控制器和 combo-PHY。

9.3.4.1 MIPI 模式配置流程

MIPI 模式下需要配置 PHY 的工作模式、数据传输所用的通道数、数据类型及宽动态模式。MIPI 模式的帧/行同步信息包含于数据包中，由控制器完成数据包的解析，从而恢复出像素数据。MIPI 模式软件操作流程如下。

步骤 1. 上电启动。



- 步骤 2. 根据使用场景将 MISC_CTRL6 寄存器中相应通道的 mipi_work_mode 配置为 MIPI 模式。
- 步骤 3. 配置 CRG 寄存器中的 PERI_CRG62, 打开 mipi_bus_clken、mipi_cil_clken 以及对应通道的 mipi_pix_clken。配置 mipi 总线软复位, 撤销复位; 配置对应通道 pix_core 复位, 撤销复位。
- 步骤 4. 配置 CRG 寄存器中的 PERI_CRG60, 配置 sensor 复位, 撤销复位; 打开 sensor 时钟门控, 并配置时钟频率。
- 步骤 5. 配置 CRG 寄存器中的 PERI_CRG61, 选择 MIP_Rx 通道时钟频率。
- 步骤 6. 配置 MIPI_Rx 的 Lane 数 (MIPI_LANES_NUM) 和 Lane ID (LANE_ID*_CHN*)。
- 步骤 7. 配置接收数据类型、WDR 模式等。
- 步骤 8. 配置 PHY 的工作模式 (PHY_MODE_LINK*)、PHY 通道延迟调节 (PHY_SKEW_LINK*)、PHY 通道使能 (PHY_EN_LINK*)、PHY 性能调节 (PHY_CFG_LINK*)。
- 步骤 9. 配置系统控制寄存器。场景模式选择 (HS_MODE_SELECT)、PHY_EN、LANE_EN, 打开 PHY_CIL_CTRL、选择 PHYCFG_MODE (对于 MIPI 模式, 应选择 0 或者 4)。
- 步骤 10. 配置对应的 PHYCFG_EN。
- 步骤 11. 配置 sensor 序列。

----结束

9.3.4.2 LVDS 和 HiSPi 模式配置流程

LVDS/HiSPi 模式下需要配置 RAW DATA 类型、数据大小端、同步方式、WDR 类型和图像宽高等寄存器。LVDS 模式依靠同步码识别帧/行同步信息, 根据 RAW DATA 类型的不同, 同步码可以为 8/10/12/14/16-bit。

LVDS 和 HiSPi 模式软件的操作流程如下。

- 步骤 1. 上电启动。
- 步骤 2. 根据使用场景将 MISC_CTRL6 寄存器中相应通道的 mipi_work_mode 配置为 LVDS 模式。
- 步骤 3. 配置 CRG 寄存器中的 PERI_CRG62, 打开 mipi_bus_clken、mipi_cil_clken 以及对应通道的 mipi_pix_clken。配置 mipi 总线软复位, 撤销复位; 配置对应通道 pix_core 复位, 撤销复位。
- 步骤 4. 配置 CRG 寄存器中的 PERI_CRG60, 配置 sensor 复位, 撤销复位; 打开 sensor 时钟门控, 并配置时钟频率。
- 步骤 5. 配置 CRG 寄存器中的 PERI_CRG61, 选择 MIP_Rx 通道时钟频率。
- 步骤 6. 配置接收数据类型、WDR 模式, 图像宽高 (LVDS 模式下, 配置的宽度是图像实际宽度除以 Lane 数-1)、同步头、Lane ID 等信息。



- 步骤 7. 配置 PHY 的工作模式（PHY_MODE_LINK*）、PHY 通道延迟调节（PHY_SKEW_LINK*）、PHY 通道使能（PHY_EN_LINK*）、PHY 性能调节（PHY_CFG_LINK*）；配置 LVDS 模式 Lane 同步头信息（PHY_SYNC_CODE*_LINK*）。
- 步骤 8. 配置系统控制寄存器。场景模式选择（HS_MODE_SELECT）、PHY_EN、LANE_EN，打开 PHY_CIL_CTRL、选择 PHYCFG_MODE（对于 LVDS 模式，应选择 1）。
- 步骤 9. 配置对应的 PHYCFG_EN。
- 步骤 10. 配置 sensor 序列。

----结束

9.3.5 MIPI Rx 寄存器概览

MIPI Rx 寄存器偏移地址中变量的取值范围和含义如表 9-11 所示。

表9-11 MIPI Rx 寄存器偏移地址变量表

变量名称	取值范围	描述
M	0~3	Lane 号 当 N=0 时，M 取值：[0,3]； 当 N=1 时，M 取值：[0,1]。
N	0~1	通道号

MIPI Rx 寄存器概览如表 9-12 所示。

表9-12 MIPI Rx 寄存器概览（基址是 0x113A_0000）

偏移地址	名称	描述	页码
0x0000	PHY_MODE_LINK	Link PHY 工作模式寄存器	9-138
0x0004	PHY_SKEW_LINK	Link PHY 通道延迟调节寄存器	9-139
0x000C	PHY_EN_LINK	Link PHY 通道使能寄存器	9-140
0x0018	PHY_DATA_LINK	Link PHY 输出的并行数据寄存器	9-141
0x001C	PHY_PH_MIPI_LINK	Link MIPI 数据包头寄存器	9-142
0x0020	PHY_DATA_MIPI_LINK	Link MIPI 模式数据寄存器	9-142
0x0024	PHY_SYNC_DCT_LINK	Link PHY LVDS 模式同步头检测控制寄存器	9-142



偏移地址	名称	描述	页码
0x0030	PHY_SYNC_SOF0_LINK	Link PHY LVDS 模式 Lane0 同步头寄存器	9-143
0x0034	PHY_SYNC_SOF1_LINK	Link PHY LVDS 模式 Lane1 同步头寄存器	9-144
0x0038	PHY_SYNC_SOF2_LINK	Link PHY LVDS 模式 Lane2 同步头寄存器	9-144
0x003C	PHY_SYNC_SOF3_LINK	Link PHY LVDS 模式 Lane3 同步头寄存器	9-144
0x0118	PHY_ST2_LINK	Link PHY 状态 2 寄存器	9-144
0x01F0	MIPI_CIL_INT_RAW_LINK	Link MIPI CIL 原始中断状态寄存器	9-145
0x01F4	MIPI_CIL_INT_LINK	Link MIPI CIL 中断状态寄存器	9-146
0x01F8	MIPI_CIL_INT_MSK_LINK	Link MIPI CIL 中断屏蔽寄存器	9-148
0x0800	HS_MODE_SELECT	HS ID 选择寄存器	9-149
0x0808	PHY_EN	PHY 使能寄存器	9-149
0x080C	LANE_EN	LANE_EN 使能寄存器	9-150
0x0810	PHY_CIL_CTRL	PHY CIL 控制寄存器	9-150
0x0818	PHYCFG_MODE	PHY 配置模式寄存器	9-151
0x081C	PHYCFG_EN	PHY 配置使能寄存器	9-152
0x0820	CHN0_MEM_CTRL	CHN0 MEMORY 控制寄存器	9-152
0x0824	CHN0_CLR_EN	CHN0 强制置位使能寄存器	9-153
0x0828	CHN1_MEM_CTRL	CHN1 MEMORY 控制寄存器	9-153
0x082C	CHN1_CLR_EN	CHN1 强制置位使能寄存器	9-154
0x0FF0	MIPI_INT_RAW	MIPI 系统原始中断状态寄存器	9-154
0x0FF4	MIPI_INT_ST	MIPI 系统中断状态寄存器	9-154
0x0FF8	MIPI_INT_MSK	MIPI 系统中断屏蔽寄存器	9-155
0x1004 + N × 0x1000	MIPI(N)_LANES_NUM	MIPI(N)数据通道数使能寄存器	9-156



偏移地址	名称	描述	页码
0x100C + N × 0x1000	MIPI(N)_MAIN_INT_ST	MIPI(N)总中断状态寄存器	9-156
0x1010 + N × 0x1000	MIPI(N)_DI_1	MIPI(N)控制器 Data ID 1 寄存器	9-157
0x1014 + N × 0x1000	MIPI(N)_DI_2	MIPI(N)控制器 Data ID 2 寄存器	9-158
0x1060 + N × 0x1000	MIPI(N)_PKT_INTR_ST	MIPI(N)包中断状态寄存器	9-159
0x1064 + N × 0x1000	MIPI(N)_PKT_INTR_MSK	MIPI(N)包中断屏蔽寄存器	9-160
0x1070 + N × 0x1000	MIPI(N)_PKT_INTR_2_ST	MIPI(N)包中断状态 2 寄存器	9-160
0x1074 + N × 0x1000	MIPI(N)_PKT_INTR_2_MSK	MIPI(N)包中断屏蔽 2 寄存器	9-161
0x1080 + N × 0x1000	MIPI(N)_FRAME_INTR_ST	MIPI(N)帧中断状态寄存器	9-162
0x1084 + N × 0x1000	MIPI(N)_FRAME_INTR_MSK	MIPI(N)帧中断屏蔽寄存器	9-164
0x1090 + N × 0x1000	MIPI(N)_LINE_INTERRUPT_ST	MIPI(N)行中断状态寄存器	9-165
0x1094 + N × 0x1000	MIPI(N)_LINE_INTERRUPT_MSK	MIPI(N)行中断屏蔽寄存器	9-167
0x1100 + N × 0x1000	MIPI(N)_USERDEFINED_DT	MIPI(N)用户自定义 data type 对应的像素位宽配置寄存器	9-169
0x1104 + N × 0x1000	MIPI(N)_USER_DEFINED_DT_TYPE_CFG	MIPI(N)用户自定义 data type 的类型使能配置寄存器	9-170



偏移地址	名称	描述	页码
0x1108+N×0x1000	MIPI(N)_CTRL_MODE_HS	MIPI(N)工作模式使能寄存器	9-171
0x1200+N×0x1000	MIPI(N)_DOL_ID_CODE0	MIPI(N) DOL 模式帧识别 0 寄存器	9-172
0x1204+N×0x1000	MIPI(N)_DOL_ID_CODE1	MIPI(N) DOL 模式帧识别 1 寄存器	9-172
0x1208+N×0x1000	MIPI(N)_DOL_ID_CODE2	MIPI(N) DOL 模式帧识别 2 寄存器	9-172
0x1210+N×0x1000	MIPI(N)_CROP_START_CHN0	MIPI(N) CHN0 CROP 宽高起始点寄存器	9-172
0x1214+N×0x1000	MIPI(N)_CROP_START_CHN1	MIPI(N) CHN1 CROP 宽高起始点寄存器	9-173
0x1218+N×0x1000	MIPI(N)_CROP_START_CHN2	MIPI(N) CHN2 CROP 宽高起始点寄存器	9-173
0x121C+N×0x1000	MIPI(N)_CROP_START_CHN3	MIPI(N) CHN3 CROP 宽高起始点寄存器	9-173
0x1224+N×0x1000	MIPI(N)_IMGSIZE	MIPI(N)图像宽高寄存器	9-173
0x1230+N×0x1000	MIPI(N)_CTRL_MODE_PIXEL	MIPI(N)输出工作模式使能寄存器	9-174
0x1240+N×0x1000	MIPI(N)_DUMMY_PIX_REG	MIPI(N) dummy 行像素值寄存器	9-175
0x1244+N×0x1000	MIPI(N)_YUVEVEN_THOLD	MIPI(N) YUV 奇偶行阈值寄存器	9-175
0x1250+N×0x1000	MIPI(N)_IMGSIZE0_STATIS	MIPI(N) VC0 传输的图像宽高寄存器	9-175



偏移地址	名称	描述	页码
0x1254+N×0x1000	MIPI(N)_IMGSIZE1_STATIS	MIPI(N) VC1 传输的图像宽高寄存器	9-176
0x1258+N×0x1000	MIPI(N)_IMGSIZE2_STATIS	MIPI(N) VC2 传输的图像宽高寄存器	9-176
0x125C+N×0x1000	MIPI(N)_IMGSIZE3_STATIS	MIPI(N) VC3 传输的图像宽高寄存器	9-176
0x12F0+N×0x1000	MIPI(N)_CTRL_INT_RAW	MIPI(N)读数据错误原始中断状态寄存器	9-176
0x12F4+N×0x1000	MIPI(N)_CTRL_INT	MIPI(N)读数据错误中断状态寄存器	9-177
0x12F8+N×0x1000	MIPI(N)_CTRL_INT_MSK	MIPI(N)读数据错误中断屏蔽寄存器	9-178
0x1300+N×0x1000	LVDS(N)_WDR	LVDS(N) WDR 控制寄存器	9-179
0x1304+N×0x1000	LVDS(N)_DOLSCD_HBLK	LVDS(N) SCD 控制寄存器	9-180
0x1308+N×0x1000	LVDS(N)_CTRL	LVDS(N)控制寄存器	9-180
0x130C+N×0x1000	LVDS(N)_IMGSIZE	LVDS(N)图像宽高寄存器	9-181
0x1310+N×0x1000	LVDS(N)_CROP_ST_ART0	LVDS(N)第一帧裁剪寄存器	9-182
0x1314+N×0x1000	LVDS(N)_CROP_ST_ART1	LVDS(N)第二帧裁剪寄存器	9-182
0x1318+N×0x1000	LVDS(N)_CROP_ST_ART2	LVDS(N)第三帧裁剪寄存器	9-182



偏移地址	名称	描述	页码
0x131C + N × 0x1000	LVDS(N)_CROP_ST ART3	LVDS(N)第四帧裁剪寄存器	9-182
0x1320 + M × 0x20 + N × 0x1000	LVDS(N)_LANE(M) _SOF_01	LVDS/HiSPi 模式下, lane(M)帧起始同步码配置寄存器	9-183
0x1324 + M × 0x20 + N × 0x1000	LVDS(N)_LANE(M) _SOF_23	LVDS/HiSPi 模式下, lane(M)帧起始同步码配置寄存器	9-183
0x1328 + M × 0x20 + N × 0x1000	LVDS(N)_LANE(M) _EOF_01	LVDS/HiSPi 模式下, lane(M)帧结束同步码配置寄存器	9-184
0x132C + M × 0x20 + N × 0x1000	LVDS(N)_LANE(M) _EOF_23	LVDS/HiSPi 模式下, lane(M)帧结束同步码配置寄存器	9-184
0x1330 + M × 0x20 + N × 0x1000	LVDS(N)_LANE(M) _SOL_01	LVDS/HiSPi 模式下, lane(M)行起始同步码配置寄存器	9-184
0x1334 + M × 0x20 + N × 0x1000	LVDS(N)_LANE(M) _SOL_23	LVDS/HiSPi 模式下, lane(M)行起始同步码配置寄存器	9-185
0x1338 + M × 0x20 + N × 0x1000	LVDS(N)_LANE(M) _EOL_01	LVDS/HiSPi 模式下, lane(M)行结束同步码配置寄存器	9-185
0x133C + M × 0x20 + N × 0x1000	LVDS(N)_LANE(M) _EOL_23	LVDS/HiSPi 模式下, lane(M)行结束同步码配置寄存器	9-186
0x1520 + M × 0x20 + N × 0x1000	LVDS(N)_LANE(M) _NXT_SOF_01	LVDS/HiSPi 模式下, 第 N+1 帧 lane(M)帧起始同步码配置寄存器	9-186



偏移地址	名称	描述	页码
0x1524+ M×0x20 +N× 0x1000	LVDS(N)_LANE(M) _NXT_SOF_23	LVDS/HiSPi 模式下, 第 N+1 帧 lane(M) 帧起始同步码配置寄存器	9-187
0x1528+ M×0x20 +N× 0x1000	LVDS(N)_LANE(M) _NXT_EOF_01	LVDS/HiSPi 模式下, 第 N+1 帧 lane(M) 帧结束同步码配置寄存器	9-187
0x152C+ M×0x20 +N× 0x1000	LVDS(N)_LANE(M) _NXT_EOF_23	LVDS/HiSPi 模式下, 第 N+1 帧 lane(M) 帧结束同步码配置寄存器	9-187
0x1530+ M×0x20 +N× 0x1000	LVDS(N)_LANE(M) _NXT_SOL_01	LVDS/HiSPi 模式下, 第 N+1 帧 lane(M) 行起始同步码配置寄存器	9-188
0x1534+ M×0x20 +N× 0x1000	LVDS(N)_LANE(M) _NXT_SOL_23	LVDS/HiSPi 模式下, 第 N+1 帧 lane(M) 行起始同步码配置寄存器	9-188
0x1538+ M×0x20 +N× 0x1000	LVDS(N)_LANE(M) _NXT_EOL_01	LVDS/HiSPi 模式下, 第 N+1 帧 lane(M) 行结束同步码配置寄存器	9-189
0x153C+ M×0x20 +N× 0x1000	LVDS(N)_LANE(M) _NXT_EOL_23	LVDS/HiSPi 模式下, 第 N+1 帧 lane(M) 行结束同步码配置寄存器	9-189
0x1720+ N× 0x1000	LVDS(N)_LI_WOR D0	LVDS DOL 模式第 0 帧 LI 寄存器	9-190
0x1724+ N× 0x1000	LVDS(N)_LI_WOR D1	LVDS DOL 模式第 1 帧 LI 寄存器	9-190
0x1728+ N× 0x1000	LVDS(N)_LI_WOR D2	LVDS DOL 模式第 2 帧 LI 寄存器	9-190
0x172C+ N× 0x1000	LVDS(N)_LI_WOR D3	LVDS DOL 模式第 3 帧 LI 寄存器	9-190



偏移地址	名称	描述	页码
0x1780+N×0x1000	LVDS(N)_IMGSIZE0_STATIS	LVDS LEF 图像宽高统计寄存器	9-191
0x1784+N×0x1000	LVDS(N)_IMGSIZE1_STATIS	LVDS SEF1 图像宽高统计寄存器	9-191
0x1788+N×0x1000	LVDS(N)_IMGSIZE2_STATIS	LVDS SEF2 图像宽高统计寄存器	9-191
0x178C+N×0x1000	LVDS(N)_IMGSIZE3_STATIS	LVDS SEF3 图像宽高统计寄存器	9-192
0x17F0+N×0x1000	LVDS(N)_CTRL_INT_RAW	LVDS 读数据原始中断状态寄存器	9-192
0x17F4+N×0x1000	LVDS(N)_CTRL_INT	LVDS 读数据中断状态寄存器	9-193
0x17F8+N×0x1000	LVDS(N)_CTRL_INT_MSK	LVDS 读数据中断屏蔽寄存器	9-195
0x1800+N×0x1000	LANE_ID0_CHN(N)	LANE 优先级配置寄存器 0	9-196
0x18F0+N×0x1000	ALIGN(N)_INT_RAW	MIPI_ALIGN 原始中断状态寄存器	9-196
0x18F4+N×0x1000	ALIGN(N)_INT	MIPI_ALIGN 中断状态寄存器	9-197
0x18F8+N×0x1000	ALIGN(N)_INT_MSK	MIPI_ALIGN 中断屏蔽寄存器	9-198
0x1FF0+N×0x1000	CHN(N)_INT_RAW	通道(N)原始中断状态寄存器	9-198
0x1FF4+N×0x1000	CHN(N)_INT	通道(N)中断状态寄存器	9-199



偏移地址	名称	描述	页码
0x1FF8+N×0x1000	CHN(N)_INT_MASK	通道(N)中断屏蔽寄存器	9-200

9.3.6 MIPI Rx 寄存器描述

PHY_MODE_LINK

PHY_MODE_LINK 为 Link PHY 工作模式寄存器。

Offset Address: 0x0000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25]	RW	phy0_rg_mipi_mode1	MIPI/LVDS 输入模式选择。当 PHY 工作于 LVDS 模式且 Common voltage 大于等于 1200mV 时设置为 0。 0: Common voltage 在 1200mV 以上; 1: Common voltage 在 1200mV 或以下。	0x0
[24]	RW	phy0_rg_mipi_mode0	MIPI/LVDS 输入模式选择。当 PHY 工作于 LVDS 模式且 Common voltage 大于等于 900mV 时设置为 0。 0: Common voltage 在 1200mV 以上; 1: Common voltage 在 1200mV 或以下。	0x0
[23:21]	-	reserved	保留。	0x0
[20]	RW	phy0_rg_en_2l2l	差分时钟源选择。 0: 外部输入一对差分时钟; 1: 外部输入两对差分时钟。	0x0
[19:18]	-	reserved	保留。	0x0
[17]	RW	phy0_rg_faclk1_en	PHY 输出数据的随路 clock1 相位。 0: 时钟上升沿输出数据; 1: 时钟下降沿输出数据。	0x0
[16]	RW	phy0_rg_faclk0_en	PHY 输出数据的随路 clock0 相位。 0: 时钟上升沿输出数据; 1: 时钟下降沿输出数据。	0x0
[15:14]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[13]	RW	phy0_rg_en_lp1	LP1 模式使能。MIPI 模式时使能，其他模式时禁止。 0: 关闭 PHY 的 LP1 模式; 1: 使能 PHY 的 LP1 模式。	0x0
[12]	RW	phy0_rg_en_lp0	LP0 模式使能。MIPI 模式时使能，其他模式时禁止。 0: 关闭 PHY 的 LP0 模式; 1: 使能 PHY 的 LP0 模式。	0x0
[11:9]	-	reserved	保留。	0x0
[8]	RW	phy0_rg_en_cmos	CMOS 模式使能。 0: 不使能 PHY 的 CMOS 模式; 1: 使能 PHY 的 CMOS 模式。	0x0
[7:6]	-	reserved	保留。	0x0
[5]	RW	phy0_rg_en_clk1	Clock1 Lane 使能。 0: 禁止 Clock1 Lane; 1: 使能 Clock1 Lane。	0x0
[4]	RW	phy0_rg_en_clk0	Clock0 Lane 使能。 0: 禁止 Clock0 Lane; 1: 使能 Clock0 Lane。	0x0
[3:0]	RW	phy0_rg_en_d	Data Lane 使能。 0: 禁止相应的 Data Lane; 1: 使能相应的 Data Lane。	0x0

PHY_SKEW_LINK

PHY_SKEW_LINK 为 Link PHY 通道延迟调节寄存器。

Offset Address: 0x0004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:23]	-	reserved	保留。	0x000
[22:20]	RW	phy0_d3_skew	Data lane 3 的时序延迟调节。此寄存器的值每加 1，相位向后延迟约 62.5ps。	0x0
[19]	-	reserved	保留。	0x0
[18:16]	RW	phy0_d2_skew	Data lane 2 的时序延迟调节。此寄存器的	0x0



Bits	Access	Name	Description	Reset
			值每加 1，相位向后延迟约 62.5ps。	
[15]	-	reserved	保留。	0x0
[14:12]	RW	phy0_d1_skew	Data lane 1 的时序延迟调节。此寄存器的值每加 1，相位向后延迟约 62.5ps。	0x0
[11]	-	reserved	保留。	0x0
[10:8]	RW	phy0_d0_skew	Data lane 0 的时序延迟调节。此寄存器的值每加 1，相位向后延迟约 62.5ps。	0x0
[7]	-	reserved	保留。	0x0
[6:4]	RW	phy0_clk1_skew	Clock1 Lane 的时序延迟调节。此寄存器的值每加 1，相位向后延迟约 62.5ps。	0x0
[3]	-	reserved	保留。	0x0
[2:0]	RW	phy0_clk0_skew	Clock0 Lane 的时序延迟调节。此寄存器的值每加 1，相位向后延迟约 62.5ps。	0x0

PHY_EN_LINK

PHY_EN_LINK 为 Link PHY 通道使能寄存器。

Offset Address: 0x000C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13]	RW	phy_clk1_term_en	Clock1 Lane 终接匹配阻抗使能。 0: 关闭内部匹配电阻; 1: 打开内部匹配电阻。	0x0
[12]	RW	phy_clk0_term_en	Clock0 Lane 终接匹配阻抗使能。 0: 关闭内部匹配电阻; 1: 打开内部匹配电阻。	0x0
[11]	RW	phy_d3_term_en	Data lane3 终接匹配阻抗使能。 0: 关闭内部匹配电阻; 1: 打开内部匹配电阻。	0x0
[10]	RW	phy_d2_term_en	Data lane2 终接匹配阻抗使能。 0: 关闭内部匹配电阻; 1: 打开内部匹配电阻。	0x0



Bits	Access	Name	Description	Reset
[9]	RW	phy_d1_term_en	Data lane1 终接匹配阻抗使能。 0: 关闭内部匹配电阻; 1: 打开内部匹配电阻。	0x0
[8]	RW	phy_d0_term_en	Data lane0 终接匹配阻抗使能。 0: 关闭内部匹配电阻; 1: 打开内部匹配电阻。	0x0
[7:4]	-	reserved	保留。	0x0
[3]	RW	phy_da_d3_valid	Data lane3 高速模式使能。 0: 高速模式关闭; 1: 高速模式使能。	0x0
[2]	RW	phy_da_d2_valid	Data lane2 高速模式使能。 0: 高速模式关闭; 1: 高速模式使能。	0x0
[1]	RW	phy_da_d1_valid	Data lane1 高速模式使能。 0: 高速模式关闭; 1: 高速模式使能。	0x0
[0]	RW	phy_da_d0_valid	Data lane0 高速模式使能。 0: 高速模式关闭; 1: 高速模式使能。	0x0

PHY_DATA_LINK

PHY_DATA_LINK 为 Link PHY 输出的并行数据寄存器。

Offset Address: 0x0018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RO	phy_data3_mipi	Data Lane3 接收到的数据。	0x00
[23:16]	RO	phy_data2_mipi	Data Lane2 接收到的数据。	0x00
[15:8]	RO	phy_data1_mipi	Data Lane1 接收到的数据。	0x00
[7:0]	RO	phy_data0_mipi	Data Lane0 接收到的数据。	0x00



PHY_PH_MIPI_LINK

PHY_PH_MIPI_LINK 为 Link MIPI 数据包头寄存器。

Offset Address: 0x001C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RO	phy_ph3_mipi	Data Lane3 接收到的数据包头。	0x00
[23:16]	RO	phy_ph2_mipi	Data Lane2 接收到的数据包头。	0x00
[15:8]	RO	phy_ph1_mipi	Data Lane1 接收到的数据包头。	0x00
[7:0]	RO	phy_ph0_mipi	Data Lane0 接收到的数据包头。	0x00

PHY_DATA_MIPI_LINK

PHY_DATA_MIPI_LINK 为 Link MIPI 模式数据寄存器。

Offset Address: 0x0020 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RO	phy_data3_mipi_hs	Data Lane3 MIPI 格式的数据。	0x00
[23:16]	RO	phy_data2_mipi_hs	Data Lane2 MIPI 格式的数据。	0x00
[15:8]	RO	phy_data1_mipi_hs	Data Lane1 MIPI 格式的数据。	0x00
[7:0]	RO	phy_data0_mipi_hs	Data Lane0 MIPI 格式的数据。	0x00

PHY_SYNC_DCT_LINK

PHY_SYNC_DCT_LINK 为 Link PHY LVDS 模式同步头检测控制寄存器。

Offset Address: 0x0024 Total Reset Value: 0x0010_1101

Bits	Access	Name	Description	Reset
[31:21]	-	reserved	保留。	0x000
[20]	RW	cil_code_big_endian 1	LVDS/HiSPi 模式下，传输的 RAW data 同步码(sync_code)的串行比特位传输顺序 1。 0: LSB，低比特先传，实际收到的同步码(sync_code)串行数据顺序为 bit0, bit1, ... bit11; 1: MSB，高比特先传，实际收到的同步码(sync_code)串行数据顺序为 bit11, bit10, ...bit0。	0x1



Bits	Access	Name	Description	Reset
[19:15]	-	reserved	保留。	0x00
[14:12]	RW	cil_raw_type1	LVDS/HiSPi 模式下，传输的 RAW data type1。 001: Raw 8bit; 010: Raw 10bit; 011: Raw 12bit; 100: Raw 14bit; 101: Raw 16bit。	0x1
[11:9]	-	reserved	保留。	0x0
[8]	RW	cil_code_big_endian 0	LVDS/HiSPi 模式下，传输的 RAW data 同步码(sync_code)的串行比特位传输顺序 0。 0: LSB, 低比特先传，实际收到的同步码(sync_code)串行数据顺序为 bit0, bit1, ... bit11; 1: MSB, 高比特先传，实际收到的同步码(sync_code)串行数据顺序为 bit11, bit10, ...bit0。	0x1
[7:3]	-	reserved	保留。	0x00
[2:0]	RW	cil_raw_type0	LVDS/HiSPi 模式下，传输的 RAW data type0。 001: Raw 8bit; 010: Raw 10bit; 011: Raw 12bit; 100: Raw 14bit; 101: Raw 16bit。	0x1

PHY_SYNC_SOF0_LINK

PHY_SYNC_SOF0_LINK 为 Link PHY LVDS 模式 Lane0 同步头寄存器。

Offset Address: 0x0030 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	cil_sof1_word4_0	Lane0 帧起始同步码(N+1 Frame)。	0x0000
[15:0]	RW	cil_sof0_word4_0	Lane0 帧起始同步码(N Frame)。	0x0000



PHY_SYNC_SOF1_LINK

PHY_SYNC_SOF1_LINK 为 Link PHY LVDS 模式 Lane1 同步头寄存器。

Offset Address: 0x0034 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	cil_sof1_word4_1	Lane1 帧起始同步码(N+1 Frame)。	0x0000
[15:0]	RW	cil_sof0_word4_1	Lane1 帧起始同步码(N Frame)。	0x0000

PHY_SYNC_SOF2_LINK

PHY_SYNC_SOF2_LINK 为 Link PHY LVDS 模式 Lane2 同步头寄存器。

Offset Address: 0x0038 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	cil_sof1_word4_2	Lane2 帧起始同步码(N+1 Frame)。	0x0000
[15:0]	RW	cil_sof0_word4_2	Lane2 帧起始同步码(N Frame)。	0x0000

PHY_SYNC_SOF3_LINK

PHY_SYNC_SOF3_LINK 为 Link PHY LVDS 模式 Lane3 同步头寄存器。

Offset Address: 0x003C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	cil_sof1_word4_3	Lane3 帧起始同步码(N+1 Frame)。	0x0000
[15:0]	RW	cil_sof0_word4_3	Lane3 帧起始同步码(N Frame)。	0x0000

PHY_ST2_LINK

PHY_ST2_LINK 为 Link PHY 状态 2 寄存器。

Offset Address: 0x0118 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:14]	RO	reserved	保留。	0x00000
[13]	RO	physt_clk1_term_en	Clock2 Lane 终接匹配阻抗使能。 0: 关闭内部匹配电阻;	0x0



Bits	Access	Name	Description	Reset
			1: 打开内部匹配电阻。	
[12]	RO	physt_clk0_term_en	Clock Lane 终接匹配阻抗使能。 0: 关闭内部匹配电阻; 1: 打开内部匹配电阻。	0x0
[11]	RO	physt_d3_term_en	Data lane3 终接匹配阻抗使能。 0: 关闭内部匹配电阻; 1: 打开内部匹配电阻。	0x0
[10]	RO	physt_d2_term_en	Data lane2 终接匹配阻抗使能。 0: 关闭内部匹配电阻; 1: 打开内部匹配电阻。	0x0
[9]	RO	physt_d1_term_en	Data lane1 终接匹配阻抗使能。 0: 关闭内部匹配电阻; 1: 打开内部匹配电阻。	0x0
[8]	RO	physt_d0_term_en	Data lane0 终接匹配阻抗使能。 0: 关闭内部匹配电阻; 1: 打开内部匹配电阻。	0x0
[7:0]	-	reserved	保留。	0x00

MIPI_CIL_INT_RAW_LINK

MIPI_CIL_INT_RAW_LINK 为 Link MIPI CIL 原始中断状态寄存器。

Offset Address: 0x01F0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13]	WC	err0_timeout_ck1_raw	Clock1 Lane FSM timeout 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[12]	WC	err0_timeout_ck0_raw	Clock0 Lane FSM timeout 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[11]	WC	err0_timeout_d3_raw	Data Lane3 FSM timeout 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0



Bits	Access	Name	Description	Reset
[10]	WC	err0_timeout_d2_raw	Data Lane2 FSM timeout 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[9]	WC	err0_timeout_d1_raw	Data Lane1 FSM timeout 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[8]	WC	err0_timeout_d0_raw	Data Lane0 FSM timeout 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[7:6]	-	reserved	保留。	0x0
[5]	WC	err0_escape_ck1_raw	Clock1 Lane escape 序列原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[4]	WC	err0_escape_ck0_raw	Clock0 Lane escape 序列原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[3]	WC	err0_escape_d3_raw	Data Lane3 escape 序列原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[2]	WC	err0_escape_d2_raw	Data Lane2 escape 序列原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[1]	WC	err0_escape_d1_raw	Data Lane1 escape 序列原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[0]	WC	err0_escape_d0_raw	Data Lane0 escape 序列原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0

MIPI_CIL_INT_LINK

MIPI_CIL_INT_LINK 为 Link MIPI CIL 中断状态寄存器。

Offset Address: 0x01F4 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13]	RO	err0_timeout_ck1_st	Clock1 Lane FSM timeout 中断状态。 0: 无中断; 1: 有中断。	0x0
[12]	RO	err0_timeout_ck0_st	Clock0 Lane FSM timeout 中断状态。 0: 无中断; 1: 有中断。	0x0
[11]	RO	err0_timeout_d3_st	Data Lane3 FSM timeout 中断状态。 0: 无中断; 1: 有中断。	0x0
[10]	RO	err0_timeout_d2_st	Data Lane2 FSM timeout 中断状态。 0: 无中断; 1: 有中断。	0x0
[9]	RO	err0_timeout_d1_st	Data Lane1 FSM timeout 中断状态。 0: 无中断; 1: 有中断。	0x0
[8]	RO	err0_timeout_d0_st	Data Lane0 FSM timeout 中断状态。 0: 无中断; 1: 有中断。	0x0
[7:6]	-	reserved	保留。	0x0
[5]	RO	err0_escape_ck1_st	Clock1 Lane escape 序列中断状态。 0: 无中断; 1: 有中断。	0x0
[4]	RO	err0_escape_ck0_st	Clock0 Lane escape 序列中断状态。 0: 无中断; 1: 有中断。	0x0
[3]	RO	err0_escape_d3_st	Data Lane3 escape 序列中断状态。 0: 无中断; 1: 有中断。	0x0
[2]	RO	err0_escape_d2_st	Data Lane2 escape 序列中断状态。 0: 无中断; 1: 有中断。	0x0
[1]	RO	err0_escape_d1_st	Data Lane1 escape 序列中断状态。	0x0



Bits	Access	Name	Description	Reset
			0: 无中断; 1: 有中断。	
[0]	RO	err0_escape_d0_st	Data Lane0 escape 序列中断状态。 0: 无中断; 1: 有中断。	0x0

MIPI_CIL_INT_MSK_LINK

MIPI_CIL_INT_MSK_LINK 为 Link MIPI CIL 中断屏蔽寄存器。

Offset Address: 0x01F8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13]	RW	err3_timeout_ck1_msk	Clock1 Lane FSM timeout 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[12]	RW	err3_timeout_ck0_msk	Clock0 Lane FSM timeout 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[11]	RW	err3_timeout_d3_msk	Data Lane3 FSM timeout 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[10]	RW	err3_timeout_d2_msk	Data Lane2 FSM timeout 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[9]	RW	err3_timeout_d1_msk	Data Lane1 FSM timeout 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[8]	RW	err3_timeout_d0_msk	Data Lane0 FSM timeout 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[7:6]	-	reserved	保留。	0x0
[5]	RW	err3_escape_ck1_msk	Clock1 Lane escape 序列中断状态使能。 0: 屏蔽中断;	0x0



Bits	Access	Name	Description	Reset
			1: 使能中断。	
[4]	RW	err3_escape_ck0_msk	Clock0 Lane escape 序列中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[3]	RW	err3_escape_d3_msk	Data Lane3 escape 序列中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RW	err3_escape_d2_msk	Data Lane2 escape 序列中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	RW	err3_escape_d1_msk	Data Lane1 escape 序列中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	err3_escape_d0_msk	Data Lane0 escape 序列中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0

HS_MODE_SELECT

HS_MODE_SELECT 为 HS ID 选择寄存器。

Offset Address: 0x0800 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	hs_mode	场景模式选择。 0x0: 4Lane; 0x1: 2Lane+2Lane; 其他: 保留。	0x0

PHY_EN

PHY_EN 为 PHY 使能寄存器。

Offset Address: 0x0808 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	phy0_en	PHY0 使能控制。 0: 禁止; 1: 使能。	0x0

LANE_EN

LANE_EN 为 LANE_EN 使能寄存器。

Offset Address: 0x080C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000
[3]	RW	lane3_en	Lane3 工作使能。 0: 禁止; 1: 使能。	0x0
[2]	RW	lane2_en	Lane2 工作使能。 0: 禁止; 1: 使能。	0x0
[1]	RW	lane1_en	Lane1 工作使能。 0: 禁止; 1: 使能。	0x0
[0]	RW	lane0_en	Lane0 工作使能。 0: 禁止; 1: 使能。	0x0

PHY_CIL_CTRL

PHY_CIL_CTRL 为 PHY CIL 控制寄存器。

Offset Address: 0x0810 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x000000



[9]	RW	cil1_rst_req	cil1 复位。 0: 撤掉复位; 1: 复位。	0x0
[8]	RW	cil0_rst_req	cil0 复位。 0: 撤掉复位; 1: 复位。	0x0
[7:1]	-	reserved	保留。	0x00
[0]	RW	phycil0_cken	PHYCIL0 时钟门控。 0: 关闭时钟; 1: 打开时钟。	0x0

PHYCFG_MODE

PHYCFG_MODE 为 PHY 配置模式寄存器。

Offset Address: 0x0818 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	保留。	0x0000000
[6:4]	RW	phycil0_1_cfg_mode	PHY0 Lane1/Lane3 配置模式。 000: PHY 的配置受控于 PHYCFG_EN 和 FSM; (mipi) 001: PHY 的配置受控于 PHYCFG_EN; (lvds) 010: PHY 的配置直接受控于寄存器的值; 011: PHY 的配置受控于 PHYCFG_EN 和 FSM, 时钟通道的使能受控于 PHYCFG_EN。	0x0
[3]	RW	phycil0_cfg_mode_sel	PHY0 配置模式。 0: 受控于 PHYCFG_EN; 1: 受控于寄存器的值。(cmos)	0x0



[2:0]	RW	phycil0_0_cfg_mode	PHY0 Lane0/Lane2 配置模式。 000: PHY 的配置受控于 PHYCFG_EN 和 FSM; (mipi) 001: PHY 的配置受控于 PHYCFG_EN; (lvds) 010: PHY 的配置直接受控于寄存器的值; 011: PHY 的配置受控于 PHYCFG_EN 和 FSM, 时钟通道的使能受控于 PHYCFG_EN。	0x0

PHYCFG_EN

PHYCFG_EN 为 PHY 配置使能寄存器。

Offset Address: 0x081C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	WO	phycil0_cfg_en	PHY0 配置使能寄存器。 0: 不使能 PHY 的配置; 1: 使能 PHY 的配置。	0x0

CHN0_MEM_CTRL

CHN0_MEM_CTRL 为 CHN0 MEMORY 控制寄存器。

Offset Address: 0x0820 Total Reset Value: 0x0000_4420

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:12]	-	reserved	保留。	0x4
[11]	-	reserved	保留。	0x0
[10:8]	-	reserved	保留。	0x4
[7:6]	-	reserved	保留。	0x0
[5]	-	reserved	保留。	0x1
[4:1]	-	reserved	保留。	0x0



[0]	RW	chn0_mem_ck_gt	CHN0 MEMORY 时钟门控端口。 0: 关闭 memory 时钟; 1: 打开 memory 时钟。	0x0

CHN0_CLR_EN

CHN0_CLR_EN 为 CHN0 强制置位使能寄存器。

Offset Address: 0x0824 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	chn0_clr_en_align	CHN0 ALIGN 模块的强制置位控制。 0: 关闭 CHN0 ALIGN 的置位请求; 1: 使能 CHN0 ALIGN 的置位请求。	0x0
[0]	RW	chn0_clr_en_lvds	CHN0 LVDS_CTRL 模块的强制置位控制。 0: 关闭 CHN0 LVDS_CTRL 的置位请求; 1: 使能 CHN0 LVDS_CTRL 的置位请求。	0x0

CHN1_MEM_CTRL

CHN1_MEM_CTRL 为 CHN1 MEMORY 控制寄存器。

Offset Address: 0x0828 Total Reset Value: 0x0000_4420

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:12]	-	reserved	保留。	0x4
[11]	-	reserved	保留。	0x0
[10:8]	-	reserved	保留。	0x4
[7:6]	-	reserved	保留。	0x0
[5]	-	reserved	保留。	0x1
[4:1]	-	reserved	保留。	0x0
[0]	RW	chn1_mem_ck_gt	CHN1 MEMORY 时钟门控端口。 0: 关闭 memory 时钟; 1: 打开 memory 时钟。	0x0



CHN1_CLR_EN

CHN1_CLR_EN 为 CHN1 强制置位使能寄存器。

Offset Address: 0x082C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	chn1_clr_en_align	CHN1 ALIGN 模块的强制置位控制。 0: 关闭 CHN1 ALIGN 的置位请求; 1: 使能 CHN1 ALIGN 的置位请求。	0x0
[0]	RW	chn1_clr_en_lvds	CHN1 LVDS_CTRL 模块的强制置位控制。 0: 关闭 CHN1 LVDS_CTRL 的置位请求; 1: 使能 CHN1 LVDS_CTRL 的置位请求。	0x0

MIPI_INT_RAW

MIPI_INT_RAW 为 MIPI 系统原始中断状态寄存器。

Offset Address: 0x0FF0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x00000000
[5]	WC	int_chn1_raw	通道 1 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[4]	WC	int_chn0_raw	通道 0 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	WC	int_phycil0_raw	PHYCIL0 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0

MIPI_INT_ST

MIPI_INT_ST 为 MIPI 系统中断状态寄存器。



Offset Address: 0x0FF4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5]	RO	int_chn1_st	通道 1 中断状态。 0: 无中断; 1: 有中断。	0x0
[4]	RO	int_chn0_st	通道 0 中断状态。 0: 无中断; 1: 有中断。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RO	int_phycil0_st	PHYCIL0 中断状态。 0: 无中断; 1: 有中断。	0x0

MIPI_INT_MSK

MIPI_INT_MSK 为 MIPI 系统中断屏蔽寄存器。

Offset Address: 0x0FF8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5]	RW	int_chn1_mask	通道 1 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[4]	RW	int_chn0_mask	通道 0 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	int_phycil0_mask	PHYCIL0 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0



MIPI(N)_LANES_NUM

MIPI(N)_LANES_NUM 为 MIPI(N)数据通道数使能寄存器。

Offset Address: $0x1004 + N \times 0x1000$ Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	lane_num	使能的数据通道数。 000: 数据通道数为 1; 001: 数据通道数为 2; 010: 数据通道数为 3; 011: 数据通道数为 4; 100: 数据通道数为 5; 101: 数据通道数为 6; 110: 数据通道数为 7; 111: 数据通道数为 8。	0x3

MIPI(N)_MAIN_INT_ST

MIPI(N)_MAIN_INT_ST 为 MIPI(N)总中断状态寄存器。

Offset Address: $0x100C + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:19]	-	reserved	保留。	0x0000
[18]	RO	status_int_line	行中断状态寄存器，若寄存器 MIPI(N)_LINE_INTR_ST 有任意一位置 1，则该位置 1，读清 0。若要清掉上报的中断，则应清掉该位及 MIPI_LINE_INTR_ST。	0x0
[17]	RO	status_int_pkt	包中断状态寄存器，若寄存器 MIPI(N)_PKT_INTR2_ST 有任意一位置 1，则该位置 1，读清 0。若要清掉上报的中断，则应清掉该位及 MIPI_PKT_INTR2_ST。	0x0
[16:3]	-	reserved	保留。	0x0000



[2]	RO	status_int_frame_fatal	帧中断状态寄存器，若寄存器 MIPI(N)_FRAME_INTR_ST 有任意一位位置 1，则该位置 1，读清 0。若要清掉上报的中断，则应清掉该位及 MIPI(N)_FRAME_INTR_ST。	0x0
[1]	RO	status_int_pkt_fatal	包中断状态寄存器，若寄存器 MIPI(N)_PKT_INTR_ST 有任意一位位置 1，则该位置 1，读清 0。若要清掉上报的中断，则应清掉该位及 MIPI(N)_PKT_INTR_ST。	0x0
[0]	-	reserved	保留。	0x0

MIPI(N)_DI_1

MIPI(N)_DI_1 为 MIPI(N)控制器 Data ID 1 寄存器。

Offset Address: $0x1010 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	RW	di3_vc	Data ID3 的虚拟通道号。	0x0
[29:24]	RW	di3_dt	Data ID3 的数据类型。 0x2A: RAW8; 0x2B: RAW10; 0x2C: RAW12; 0x2D: RAW14; 其他: 保留。	0x00
[23:22]	RW	di2_vc	Data ID2 的虚拟通道号。	0x0
[21:16]	RW	di2_dt	Data ID2 的数据类型。 0x2A: RAW8; 0x2B: RAW10; 0x2C: RAW12; 0x2D: RAW14; 其他: 保留。	0x00
[15:14]	RW	di1_vc	Data ID1 的虚拟通道号。	0x0
[13:8]	RW	di1_dt	Data ID1 的数据类型。 0x2A: RAW8; 0x2B: RAW10; 0x2C: RAW12;	0x00



Bits	Access	Name	Description	Reset
			0x2D: RAW14; 其他: 保留。	
[7:6]	RW	di0_vc	Data ID0 的虚拟通道号。	0x0
[5:0]	RW	di0_dt	Data ID0 的数据类型。 0x2A: RAW8; 0x2B: RAW10; 0x2C: RAW12; 0x2D: RAW14; 其他: 保留。	0x00

MIPI(N)_DI_2

MIPI(N)_DI_2 为 MIPI(N)控制器 Data ID 2 寄存器。

Offset Address: $0x1014 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	RW	di7_vc	Data ID7 的虚拟通道号。	0x0
[29:24]	RW	di7_dt	Data ID7 的数据类型。 0x2A: RAW8; 0x2B: RAW10; 0x2C: RAW12; 0x2D: RAW14; 其他: 保留。	0x00
[23:22]	RW	di6_vc	Data ID6 的虚拟通道号。	0x0
[21:16]	RW	di6_dt	Data ID6 的数据类型。 0x2A: RAW8; 0x2B: RAW10; 0x2C: RAW12; 0x2D: RAW14; 其他: 保留。	0x00
[15:14]	RW	di5_vc	Data ID5 的虚拟通道号。	0x0
[13:8]	RW	di5_dt	Data ID5 的数据类型。 0x2A: RAW8; 0x2B: RAW10;	0x00



Bits	Access	Name	Description	Reset
			0x2C: RAW12; 0x2D: RAW14; 其他: 保留。	
[7:6]	RW	di4_vc	Data ID4 的虚拟通道号。	0x0
[5:0]	RW	di4_dt	Data ID4 的数据类型。 0x2A: RAW8; 0x2B: RAW10; 0x2C: RAW12; 0x2D: RAW14; 其他: 保留。	0x00

MIPI(N)_PKT_INTR_ST

MIPI(N)_PKT_INTR_ST 为 MIPI(N)包中断状态寄存器。

Offset Address: $0x1060 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x0000
[16]	RO	err_ecc_double	Header 至少 2 个错误, ECC 无法纠错。读清零。 0: 无 ECC 错误; 1: 有 ECC 错误。	0x0
[15:4]	-	reserved	保留。	0x000
[3]	RO	vc3_err_crc	VC3 数据 CRC 错误。读清零。 0: VC3 数据无 CRC 错误; 1: VC3 数据有 CRC 错误。	0x0
[2]	RO	vc2_err_crc	VC2 数据 CRC 错误。读清零。 0: VC2 数据无 CRC 错误; 1: VC2 数据有 CRC 错误。	0x0
[1]	RO	vc1_err_crc	VC1 数据 CRC 错误。读清零。 0: VC1 数据无 CRC 错误; 1: VC1 数据有 CRC 错误。	0x0
[0]	RO	vc0_err_crc	VC0 数据 CRC 错误。读清零。 0: VC0 数据无 CRC 错误;	0x0



Bits	Access	Name	Description	Reset
			1: VC0 数据有 CRC 错误。	

MIPI(N)_PKT_INTR_MSK

MIPI(N)_PKT_INTR_MSK 为 MIPI(N)包中断屏蔽寄存器。

Offset Address: $0x1064 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x0000
[16]	RW	mask_err_ecc_double	err_ecc_double 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[15:4]	-	reserved	保留。	0x000
[3]	RW	mask_vc3_err_crc	vc3_err_crc 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RW	mask_vc2_err_crc	vc2_err_crc 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	RW	mask_vc1_err_crc	vc1_err_crc 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	mask_vc0_err_crc	vc0_err_crc 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0

MIPI(N)_PKT_INTR2_ST

MIPI(N)_PKT_INTR2_ST 为 MIPI(N)包中断状态 2 寄存器。

Offset Address: $0x1070 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19]	RO	vc3_err_ecc_correcte	VC3 通道 Header 错误, ECC 已纠正。读清	0x0



Bits	Access	Name	Description	Reset
		d	零。 0: VC3 通道 Header 无 ECC 错误; 1: VC3 通道 Header 有 ECC 错误并已纠正。	
[18]	RO	vc2_err_ecc_corrected	VC2 通道 Header 错误, ECC 已纠正。读清零。 0: VC2 通道 Header 无 ECC 错误; 1: VC2 通道 Header 有 ECC 错误并已纠正。	0x0
[17]	RO	vc1_err_ecc_corrected	VC1 通道 Header 错误, ECC 已纠正。读清零。 0: VC1 通道 Header 无 ECC 错误; 1: VC1 通道 Header 有 ECC 错误并已纠正。	0x0
[16]	RO	vc0_err_ecc_corrected	VC0 通道 Header 错误, ECC 已纠正。读清零。 0: VC0 通道 Header 无 ECC 错误; 1: VC0 通道 Header 有 ECC 错误并已纠正。	0x0
[15:4]	-	reserved	保留。	0x000
[3]	RO	err_id_vc3	VC3 通道数据类型。读清零。 0: 支持的数据类型; 1: 不支持的数据类型。	0x0
[2]	RO	err_id_vc2	VC2 通道数据类型。读清零。 0: 支持的数据类型; 1: 不支持的数据类型。	0x0
[1]	RO	err_id_vc1	VC1 通道数据类型。读清零。 0: 支持的数据类型; 1: 不支持的数据类型。	0x0
[0]	RO	err_id_vc0	VC0 通道数据类型。读清零。 0: 支持的数据类型; 1: 不支持的数据类型。	0x0

MIPI(N)_PKT_INTR2_MSK

MIPI(N)_PKT_INTR2_MSK 为 MIPI(N)包中断屏蔽 2 寄存器。



Offset Address: $0x1074 + N \times 0x1000$ Total Reset Value: $0x0000\_0000$

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19]	RW	mask_vc3_err_ecc_corrected	vc3_err_ecc_corrected 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[18]	RW	mask_vc2_err_ecc_corrected	vc2_err_ecc_corrected 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[17]	RW	mask_vc1_err_ecc_corrected	vc1_err_ecc_corrected 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[16]	RW	mask_vc0_err_ecc_corrected	vc0_err_ecc_corrected 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[15:4]	-	reserved	保留。	0x000
[3]	RW	mask_err_id_vc3	err_id_vc3 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RW	mask_err_id_vc2	err_id_vc2 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	RW	mask_err_id_vc1	err_id_vc1 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	mask_err_id_vc0	err_id_vc0 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0

MIPI(N)_FRAME_INTR_ST

MIPI(N)_FRAME_INTR_ST 为 MIPI(N)帧中断状态寄存器。

Offset Address: $0x1080 + N \times 0x1000$ Total Reset Value: $0x0000\_0000$



Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19]	RO	err_frame_data_vc3	最后一帧数据中，VC3 通道数据至少有一个 CRC 错误。 读清零。 0: 无 CRC 错误; 1: 有 CRC 错误。	0x0
[18]	RO	err_frame_data_vc2	最后一帧数据中，VC2 通道数据至少有一个 CRC 错误。 读清零。 0: 无 CRC 错误; 1: 有 CRC 错误。	0x0
[17]	RO	err_frame_data_vc1	最后一帧数据中，VC1 通道数据至少有一个 CRC 错误。 读清零。 0: 无 CRC 错误; 1: 有 CRC 错误。	0x0
[16]	RO	err_frame_data_vc0	最后一帧数据中，VC0 通道数据至少有一个 CRC 错误。 读清零。 0: 无 CRC 错误; 1: 有 CRC 错误。	0x0
[15:12]	-	reserved	保留。	0x0
[11]	RO	err_f_seq_vc3	VC3 的帧序出错。读清零。 0: 帧序无误; 1: 帧序有误。	0x0
[10]	RO	err_f_seq_vc2	VC2 的帧序出错。读清零。 0: 帧序无误; 1: 帧序有误。	0x0
[9]	RO	err_f_seq_vc1	VC1 的帧序出错。读清零。 0: 帧序无误; 1: 帧序有误。	0x0
[8]	RO	err_f_seq_vc0	VC0 的帧序出错。读清零。 0: 帧序无误; 1: 帧序有误。	0x0



Bits	Access	Name	Description	Reset
[7:4]	-	reserved	保留。	0x0
[3]	RO	err_f_bndry_match_vc3	VC3 通道的帧起始和帧结束不匹配。读清零。 0: 帧起始和帧结束短包匹配; 1: 帧起始和帧结束短包不匹配。	0x0
[2]	RO	err_f_bndry_match_vc2	VC2 通道的帧起始和帧结束不匹配。读清零。 0: 帧起始和帧结束短包匹配; 1: 帧起始和帧结束短包不匹配。	0x0
[1]	RO	err_f_bndry_match_vc1	VC1 通道的帧起始和帧结束不匹配。读清零。 0: 帧起始和帧结束短包匹配; 1: 帧起始和帧结束短包不匹配。	0x0
[0]	RO	err_f_bndry_match_vc0	VC0 通道的帧起始和帧结束不匹配。读清零。 0: 帧起始和帧结束短包匹配; 1: 帧起始和帧结束短包不匹配。	0x0

MIPI(N)_FRAME_INTR_MSK

MIPI(N)_FRAME_INTR_MSK 为 MIPI(N)帧中断屏蔽寄存器。

Offset Address: $0x1084 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19]	RW	mask_err_frame_data_vc3	err_frame_data_vc3 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[18]	RW	mask_err_frame_data_vc2	err_frame_data_vc2 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[17]	RW	mask_err_frame_data_vc1	err_frame_data_vc1 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0



Bits	Access	Name	Description	Reset
[16]	RW	mask_err_frame_data_vc0	err_frame_data_vc0 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[15:12]	-	reserved	保留。	0x0
[11]	RW	mask_err_f_seq_vc3	err_f_seq_vc3 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[10]	RW	mask_err_f_seq_vc2	err_f_seq_vc2 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[9]	RW	mask_err_f_seq_vc1	err_f_seq_vc1 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[8]	RW	mask_err_f_seq_vc0	err_f_seq_vc0 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[7:4]	-	reserved	保留。	0x0
[3]	RW	mask_err_f_bndry_match_vc3	err_f_bndry_match_vc3 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RW	mask_err_f_bndry_match_vc2	err_f_bndry_match_vc2 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	RW	mask_err_f_bndry_match_vc1	err_f_bndry_match_vc1 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	mask_err_f_bndry_match_vc0	err_f_bndry_match_vc0 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0

MIPI(N)_LINE_INTR_ST

MIPI(N)_LINE_INTR_ST 为 MIPI(N)行中断状态寄存器。



Offset Address: $0x1090 + N \times 0x1000$ Total Reset Value: $0x0000\_0000$

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23]	RO	err_l_seq_di7	VC7 和 DT7 行序出错。读清零。 0: 行序正常; 1: 行序出错。	0x0
[22]	RO	err_l_seq_di6	VC6 和 DT6 行序出错。读清零。 0: 行序正常; 1: 行序出错。	0x0
[21]	RO	err_l_seq_di5	VC5 和 DT5 行序出错。读清零。 0: 行序正常; 1: 行序出错。	0x0
[20]	RO	err_l_seq_di4	VC4 和 DT4 行序出错。读清零。 0: 行序正常; 1: 行序出错。	0x0
[19]	RO	err_l_seq_di3	VC3 和 DT3 行序出错。读清零。 0: 行序正常; 1: 行序出错。	0x0
[18]	RO	err_l_seq_di2	VC2 和 DT2 行序出错。读清零。 0: 行序正常; 1: 行序出错。	0x0
[17]	RO	err_l_seq_di1	VC1 和 DT1 行序出错。读清零。 0: 行序正常; 1: 行序出错。	0x0
[16]	RO	err_l_seq_di0	VC0 和 DT0 行序出错。读清零。 0: 行序正常; 1: 行序出错。	0x0
[15:8]	-	reserved	保留。	0x00
[7]	RO	err_l_bndry_match_di7	VC7 和 DT7 的行起始和行结束不匹配。读清零。 0: 行起始和行结束短包匹配; 1: 行起始和行结束短包不匹配。	0x0
[6]	RO	err_l_bndry_match_di6	VC6 和 DT6 的行起始和行结束不匹配。读清零。	0x0



Bits	Access	Name	Description	Reset
			0: 行起始和行结束短包匹配; 1: 行起始和行结束短包不匹配。	
[5]	RO	err_l_bndry_match_di5	VC5 和 DT5 的行起始和行结束不匹配。读清零。 0: 行起始和行结束短包匹配; 1: 行起始和行结束短包不匹配。	0x0
[4]	RO	err_l_bndry_match_di4	VC4 和 DT4 的行起始和行结束不匹配。读清零。 0: 行起始和行结束短包匹配; 1: 行起始和行结束短包不匹配。	0x0
[3]	RO	err_l_bndry_match_di3	VC3 和 DT3 的行起始和行结束不匹配。读清零。 0: 行起始和行结束短包匹配; 1: 行起始和行结束短包不匹配。	0x0
[2]	RO	err_l_bndry_match_di2	VC2 和 DT2 的行起始和行结束不匹配。读清零。 0: 行起始和行结束短包匹配; 1: 行起始和行结束短包不匹配。	0x0
[1]	RO	err_l_bndry_match_di1	VC1 和 DT1 的行起始和行结束不匹配。读清零。 0: 行起始和行结束短包匹配; 1: 行起始和行结束短包不匹配。	0x0
[0]	RO	err_l_bndry_match_di0	VC0 和 DT0 的行起始和行结束不匹配。读清零。 0: 行起始和行结束短包匹配; 1: 行起始和行结束短包不匹配。	0x0

MIPI(N)_LINE_INTR_MSK

MIPI(N)_LINE_INTR_MSK 为 MIPI(N)行中断屏蔽寄存器。

Offset Address: $0x1094 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23]	RW	mask_err_l_seq_di7	err_l_seq_di7 中断屏蔽寄存器。	0x0



Bits	Access	Name	Description	Reset
			0: 屏蔽中断; 1: 使能中断。	
[22]	RW	mask_err_l_seq_di6	err_l_seq_di6 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[21]	RW	mask_err_l_seq_di5	err_l_seq_di5 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[20]	RW	mask_err_l_seq_di4	err_l_seq_di4 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[19]	RW	mask_err_l_seq_di3	err_l_seq_di3 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[18]	RW	mask_err_l_seq_di2	err_l_seq_di2 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[17]	RW	mask_err_l_seq_di1	err_l_seq_di1 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[16]	RW	mask_err_l_seq_di0	err_l_seq_di0 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[15:8]	-	reserved	保留。	0x00
[7]	RW	mask_err_l_bndry_match_di7	err_l_bndry_match_di7 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[6]	RW	mask_err_l_bndry_match_di6	err_l_bndry_match_di6 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[5]	RW	mask_err_l_bndry_match_di5	err_l_bndry_match_di5 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0



Bits	Access	Name	Description	Reset
[4]	RW	mask_err_l_bndry_match_di4	err_l_bndry_match_di4 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[3]	RW	mask_err_l_bndry_match_di3	err_l_bndry_match_di3 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RW	mask_err_l_bndry_match_di2	err_l_bndry_match_di2 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	RW	mask_err_l_bndry_match_di1	err_l_bndry_match_di1 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	mask_err_l_bndry_match_di0	err_l_bndry_match_di0 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0

MIPI(N)_USERDEF_DT

MIPI(N)_USERDEF_DT 为 MIPI(N)用户自定义 data type 对应的像素位宽配置寄存器。

Offset Address: $0x1100 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:12]	RW	user_def3_dt	data type 为 user_def3 时传输的像素位宽。 000:8bit; 001:10bit; 010:12bit; 011:14bit; 100:16bit; 101:embed_data 其他: 保留。	0x0
[11]	-	reserved	保留。	0x0
[10:8]	RW	user_def2_dt	data type 为 user_def2 时传输的像素位宽。 000:8bit;	0x0



Bits	Access	Name	Description	Reset
			001:10bit; 010:12bit; 011:14bit; 100:16bit; 101:embed_data 其他：保留。	
[7]	-	reserved	保留。	0x0
[6:4]	RW	user_def1_dt	data type 为 user_def1 时传输的像素位宽： 000:8bit 001:10bit 010:12bit 011:14bit 100:16bit 101:embed_data 其他：保留。	0x0
[3]	-	reserved	保留。	0x0
[2:0]	RW	user_def0_dt	data type 为 user_def0 时传输的像素位宽。 000:8bit; 001:10bit; 010:12bit; 011:14bit; 100:16bit; 101:embed_data 其他：保留。	0x0

MIPI(N)_USER_DEF

MIPI(N)_USER_DEF 为 MIPI(N)用户自定义 data type 的类型使能配置寄存器。

Offset Address: $0x1104 + N \times 0x1000$ Total Reset Value: 0x1036_3534

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29:24]	RW	user_def3	用户自定义数据类型 3。 此寄存器的值用于匹配传感器输出的数据类型，与 user_def3_dt 配合使用。	0x10



[23:22]	-	reserved	保留。	0x0
[21:16]	RW	user_def2	用户自定义数据类型 2。 此寄存器的值用于匹配传感器输出的数据类型，与 user_def2_dt 配合使用。	0x36
[15:14]	-	reserved	保留。	0x0
[13:8]	RW	user_def1	用户自定义数据类型 1。 此寄存器的值用于匹配传感器输出的数据类型，与 user_def1_dt 配合使用。	0x35
[7:6]	-	reserved	保留。	0x0
[5:0]	RW	user_def0	用户自定义数据类型 0。 此寄存器的值用于匹配传感器输出的数据类型，与 user_def0_dt 配合使用。	0x34

MIPI(N)_CTRL_MODE_HS

MIPI(N)_CTRL_MODE_HS 为 MIPI(N)工作模式使能寄存器。

Offset Address: $0x1108 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:9]	-	reserved	保留。	0x000000
[8]	RW	user_def_en	user define 模式使能。 0:不使能; 1:使能。	0x0
[7:5]	-	reserved	保留。	0x0
[4]	RW	vc_mode	当使用 Virtual Channel 区分不同曝光长度的数据时，Share Vsync 模式使能。 0:不使能; 1:使能。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	hdr_mode	使能 Data Type 区分不同曝光长度的数据。 0:不使能; 1:使能。	0x0



MIPI(N)_DOL_ID_CODE0

MIPI(N)_DOL_ID_CODE0 为 MIPI(N) DOL 模式帧识别 0 寄存器。

Offset Address: $0x1200 + N \times 0x1000$ Total Reset Value: 0x0242_0241

Bits	Access	Name	Description	Reset
[31:16]	RW	id_code_reg1	mipi dol 模式第 N 帧 SEF1 帧 ID 码。	0x0242
[15:0]	RW	id_code_reg0	mipi dol 模式第 N 帧 LEF 帧 ID 码。	0x0241

MIPI(N)_DOL_ID_CODE1

MIPI(N)_DOL_ID_CODE1 为 MIPI(N) DOL 模式帧识别 1 寄存器。

Offset Address: $0x1204 + N \times 0x1000$ Total Reset Value: 0x0251_0244

Bits	Access	Name	Description	Reset
[31:16]	RW	id_code_reg3	mipi dol 模式第 N+1 帧 LEF 帧 ID 码。	0x0251
[15:0]	RW	id_code_reg2	mipi dol 模式第 N 帧 SEF2 帧 ID 码。	0x0244

MIPI(N)_DOL_ID_CODE2

MIPI(N)_DOL_ID_CODE2 为 MIPI(N) DOL 模式帧识别 2 寄存器。

Offset Address: $0x1208 + N \times 0x1000$ Total Reset Value: 0x0254_0252

Bits	Access	Name	Description	Reset
[31:16]	RW	id_code_reg5	mipi dol 模式第 N+1 帧 SEF2 帧 ID 码。	0x0254
[15:0]	RW	id_code_reg4	mipi dol 模式第 N+1 帧 SEF1 帧 ID 码。	0x0252

MIPI(N)_CROP_START_CHN0

MIPI(N)_CROP_START_CHN0 为 MIPI(N) CHN0 CROP 宽高起始点寄存器。

Offset Address: $0x1210 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	mipi_start_y_chn0	chn0 裁剪起始点高度。	0x0000
[15:0]	RW	mipi_start_x_chn0	chn0 裁剪起始点宽度。	0x0000



MIPI(N)_CROP_START_CHN1

MIPI(N)_CROP_START_CHN1 为 MIPI(N) CHN1 CROP 宽高起始点寄存器。

Offset Address: $0x1214 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	mipi_start_y_chn1	chn1 裁剪起始点高度。	0x0000
[15:0]	RW	mipi_start_x_chn1	chn1 裁剪起始点宽度。	0x0000

MIPI(N)_CROP_START_CHN2

MIPI(N)_CROP_START_CHN2 为 MIPI(N) CHN2 CROP 宽高起始点寄存器。

Offset Address: $0x1218 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	mipi_start_y_chn2	chn2 裁剪起始点高度。	0x0000
[15:0]	RW	mipi_start_x_chn2	chn2 裁剪起始点宽度。	0x0000

MIPI(N)_CROP_START_CHN3

MIPI(N)_CROP_START_CHN3 为 MIPI(N) CHN3 CROP 宽高起始点寄存器。

Offset Address: $0x121C + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	mipi_start_y_chn3	chn3 裁剪起始点高度。	0x0000
[15:0]	RW	mipi_start_x_chn3	chn3 裁剪起始点宽度。	0x0000

MIPI(N)_IMGSIZE

MIPI(N)_IMGSIZE 为 MIPI(N) 图像宽高寄存器。

Offset Address: $0x1224 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	mipi_imgheight	图像高度，配置值为图像的实际高度减 1。	0x0000
[15:0]	RW	mipi_imgwidth	图像宽度，配置值为图像的实际宽度减 1。	0x0000



MIPI(N)_CTRL_MODE_PIXEL

MIPI(N)_CTRL_MODE_PIXEL 为 MIPI(N) 输出工作模式使能寄存器。

Offset Address: $0x1230 + N \times 0x1000$ Total Reset Value: $0x0000\_0000$

Bits	Access	Name	Description	Reset
[31:23]	-	reserved	保留。	0x000
[22]	RW	mipi_yuv420_even_detect	YUV420 NO Legacy 奇偶行检测使能。 0: 不使能; 1: 使能。	0x0
[21]	RO	reserved	保留。	0x0
[20]	RW	mipi_yuv_420_legacy_en	YUV420 Legacy 使能。 0: 不使能; 1: 使能。	0x0
[19]	RW	mipi_yuv_420_nolegacy_en	YUV420 NO Legacy 使能。 0: 不使能; 1: 使能。	0x0
[18]	RW	mipi_yuv_422_en	YUV422 使能。 0: 不使能; 1: 使能。	0x0
[17:15]	-	reserved	保留。	0x0
[14:13]	RW	stagger_frm_num	stagger hdr mode 帧数。 01: 两帧 hdr; 10: 三帧 hdr; 11: 四帧 hdr; 其他: 保留。	0x0
[12]	RW	stagger_hdr_mode	omnivision stagger hdr mode 模式使能。 0: 不使能; 1: 使能。	0x0
[11]	RW	sync_clear_en	帧结束自清使能。 0: 不使能; 1: 使能。	0x0
[10:5]	-	reserved	保留。	0x00
[4]	RW	mipi_dol_mode	mipi dol 模式使能。 0: 不使能;	0x0



Bits	Access	Name	Description	Reset
			1: 使能。	
[3:1]	-	reserved	保留。	0x0
[0]	RW	crop_en	裁剪使能。 0: 不使能; 1: 使能。	0x0

MIPI(N)_DUMMY_PIX_REG

MIPI(N)_DUMMY_PIX_REG 为 MIPI(N) dummy 行像素值寄存器。

Offset Address: $0x1240 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	dummy_pix_reg	stagger hdr 模式 dummy 行像素值。	0x0000

MIPI(N)_YUVEVEN_THOLD

MIPI(N)_YUVEVEN_THOLD 为 MIPI(N) YUV 奇偶行阈值寄存器。

Offset Address: $0x1244 + N \times 0x1000$ Total Reset Value: 0x0000_09D0

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	mipi_yuv_even_thold	yuv 奇偶行阈值，单位 byte；大于该值判断成偶行。	0x09D0

MIPI(N)_IMGSIZE0_STATIS

MIPI(N)_IMGSIZE0_STATIS 为 MIPI(N) VC0 传输的图像宽高寄存器。

Offset Address: $0x1250 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	imgheight_statiscvc0	MIPI 模式下，virtual channel0 传输的上一帧图像高度。	0x0000
[15:0]	RO	imgwidth_statiscvc0	MIPI 模式下，virtual channel0 传输的上一帧图像宽度。	0x0000



MIPI(N)_IMGSIZE1_STATIS

MIPI(N)_IMGSIZE1_STATIS 为 MIPI(N) VC1 传输的图像宽高寄存器。

Offset Address: $0x1254 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	imgheight_stat1_vc1	MIPI 模式下, virtual channel1 传输的上一帧图像高度。	0x0000
[15:0]	RO	imgwidth_stat1_vc1	MIPI 模式下, virtual channel1 传输的上一帧图像宽度。	0x0000

MIPI(N)_IMGSIZE2_STATIS

MIPI(N)_IMGSIZE2_STATIS 为 MIPI(N) VC2 传输的图像宽高寄存器。

Offset Address: $0x1258 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	imgheight_stat2_vc2	MIPI 模式下, virtual channel2 传输的上一帧图像高度。	0x0000
[15:0]	RO	imgwidth_stat2_vc2	MIPI 模式下, virtual channel2 传输的上一帧图像宽度。	0x0000

MIPI(N)_IMGSIZE3_STATIS

MIPI(N)_IMGSIZE3_STATIS 为 MIPI(N) VC3 传输的图像宽高寄存器。

Offset Address: $0x125C + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	imgheight_stat3_vc3	MIPI 模式下, virtual channel3 传输的上一帧图像高度。	0x0000
[15:0]	RO	imgwidth_stat3_vc3	MIPI 模式下, virtual channel3 传输的上一帧图像宽度。	0x0000

MIPI(N)_CTRL_INT_RAW

MIPI(N)_CTRL_INT_RAW 为 MIPI(N) 读数据错误原始中断状态寄存器。



Offset Address: $0x12F0 + N \times 0x1000$ Total Reset Value: $0x0000\_0000$

Bits	Access	Name	Description	Reset
[31:18]	-	reserved	保留。	0x0000
[17]	WC	int_dfifo_rderr_raw	MIPI CTRL 读数据 FIFO 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[16]	WC	int_cfifo_rderr_raw	MIPI CTRL 读命令 FIFO 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[15:5]	-	reserved	保留。	0x000
[4]	WC	int_vsync_raw	MIPI CTR vsync 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	WC	int_dfifo_wrerr_raw	MIPI CTRL 写数据 FIFO 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[0]	WC	int_cfifo_wrerr_raw	MIPI CTRL 写命令 FIFO 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0

MIPI(N)_CTRL_INT

MIPI(N)_CTRL_INT 为 MIPI(N) 读数据错误中断状态寄存器。

Offset Address: $0x12F4 + N \times 0x1000$ Total Reset Value: $0x0000\_0000$

Bits	Access	Name	Description	Reset
[31:18]	-	reserved	保留。	0x0000
[17]	RO	int_dfifo_rderr_st	MIPI CTRL 读数据 FIFO 中断状态。 0: 无中断; 1: 有中断。	0x0
[16]	RO	int_cfifo_rderr_st	MIPI CTRL 读命令 FIFO 中断状态。 0: 无中断; 1: 有中断。	0x0



Bits	Access	Name	Description	Reset
[15:5]	-	reserved	保留。	0x000
[4]	RO	int_vsync_st	MIPI CTR vsync 中断状态。 0: 无中断; 1: 有中断。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	RO	int_dfifo_wrerr_st	MIPI CTRL 写数据 FIFO 中断状态。 0: 无中断; 1: 有中断。	0x0
[0]	RO	int_cfifo_wrerr_st	MIPI CTRL 写命令 FIFO 中断状态。 0: 无中断; 1: 有中断。	0x0

MIPI(N)_CTRL_INT_MSK

MIPI(N)_CTRL_INT_MSK 为 MIPI(N) 读数据错误中断屏蔽寄存器。

Offset Address: $0x12F8 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:18]	-	reserved	保留。	0x0000
[17]	RW	int_dfifo_rderr_msk	MIPI CTRL 读数据 FIFO 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[16]	RW	int_cfifo_rderr_msk	MIPI CTRL 读命令 FIFO 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[15:5]	-	reserved	保留。	0x000
[4]	RW	int_vsync_msk	MIPI CTR vsync 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	RW	int_dfifo_wrerr_msk	MIPI CTRL 写数据 FIFO 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0



Bits	Access	Name	Description	Reset
[0]	RW	int_cfifo_wrerr_msk	MIPI CTRL 写命令 FIFO 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0

LVDS(N)_WDR

LVDS(N)_WDR 为 LVDS(N) WDR 控制寄存器。

Offset Address: $0x1300 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:12]	RW	lvds_wdr_id_shift	IMX136 Frame ID 标识移位寄存器, 标识向右移动多少位使 Frame ID 为第 0 位。	0x0
[11:8]	RW	lvds_wdr_mode	LVDS WDR 模式选择。 0x0: SOF-EOF 标识的 WDR, 长短曝光都有独立的同步码; 0x2: HiSPi 接口的 WDR 模式, 长短曝光共用一对 SOF-EOF 标识, 短曝光的起始几行用 0x04 填充; 0x4: SONY DOL 模式, 4 个字段的同步码, SAV-EAV 标识方式, 长短曝光有独立的同步码; 0x5: SONY DOL 模式, 4 个字段的同步码, SAV-EAV 标识方式, 长短曝光共用一组同步码, 长短曝光之间是消隐区; 0x6: SONY DOL 模式, 5 个字段的同步码, 第 5 个字段标识长短帧; 0x8: SONY IMX136 模式, 帧 WDR 模式, 不要打开 WDR_EN; 0x9: SHARP 长短曝光两行交错方式, 长帧在前短帧在后; 0xA: SHARP 长短曝光两行交错方式, 短帧在前长帧在后; 其他: 保留。	0x0
[7:6]	-	reserved	保留。	0x0
[5:4]	RW	lvds_wdr_num	WDR 模式配置信号。 00: 保留;	0x0



Bits	Access	Name	Description	Reset
			01: 2 帧 WDR; 10: 3 帧 WDR; 11: 4 帧 WDR。	
[3:1]	-	reserved	保留。	0x0
[0]	RW	lvds_wdr_en	WDR 使能信号。 0: Linear 模式; 1: WDR 模式。	0x0

LVDS(N)_DOLSCD_HBLK

LVDS(N)_DOLSCD_HBLK 为 LVDS(N) SCD 控制寄存器。

Offset Address: $0x1304 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	dol_hblank2	DOL 模式时 HBLANK2 的值。	0x0000
[15:0]	RW	dol_hblank1	DOL 模式时 HBLANK1 的值。	0x0000

LVDS(N)_CTRL

LVDS(N)_CTRL 为 LVDS(N) 控制寄存器。

Offset Address: $0x1308 + N \times 0x1000$ Total Reset Value: 0x0000_0330

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RW	lvds_crop_en	LVDS/HiSPi 图像裁剪使能。 0: 不使能; 1: 使能。	0x0
[11:10]	-	reserved	保留。	0x0
[9]	RW	lvds_code_big_endian	LVDS/HiSPi 模式下, 传输的 RAW data 同步码(sync_code)的串行比特位传输顺序。 0: LSB, 低比特先传, 实际收到的同步码(sync_code)串行数据顺序为 bit0, bit1, ... bit11; 1: MSB, 高比特先传, 实际收到的同步码(sync_code)串行数据顺序为 bit11,	0x1



Bits	Access	Name	Description	Reset
			bit10, ...bit0。	
[8]	RW	lvds_pix_big_endian	LVDS/HiSPi 模式下，传输的 RAW data 有效像素 pixel 的串行比特位的传输顺序。 0: LSB, 低比特先传，实际收到的有效像素 pixel 串行数据顺序为 bit0, bit1, ... bit11; 1: MSB, 高比特先传，实际收到有效像素 pixel 串行数据顺序为 bit11, bit10, ... bit0。	0x1
[7]	-	reserved	保留。	0x0
[6:4]	RW	lvds_raw_type	LVDS/HiSPi 模式下，传输的 RAW data type。 001: Raw 8bit; 010: Raw 10bit; 011: Raw 12bit; 100: Raw 14bit; 101: Raw 16bit; 其他: 保留。	0x3
[3:1]	-	reserved	保留。	0x0
[0]	RW	lvds_sync_mode	LVDS 模式下，帧/行同步模式。 0: 同步方式为 SOF/EOF/SOL/EOL 模式；SOF 标识有效区的第一行起始，用 EOF 标识有效区最后一行的结束，其他有效区分别用 SOL 和 EOL 作为起始和结束； 1: 同步方式为 SAV/EAV 模式；用 SAV(Invalid)和 EAV(Invalid)标识消隐区的无效数据，使用 SAV(Valid)和 EAV(Valid)标识有效区像素数据。	0x0

LVDS(N)_IMGSIZE

LVDS(N)_IMGSIZE 为 LVDS(N) 图像宽高寄存器。

Offset Address: $0x130C + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lvds_imgheight	图像高度减 1。	0x0000
[15:0]	RW	lvds_imgwidth_lane	每个通道传输的图像宽度减 1。	0x0000



LVDS(N)_CROP_START0

LVDS(N)_CROP_START0 为 LVDS(N) 第一帧裁剪寄存器。

Offset Address: $0x1310 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lvds_start_y0	第一帧裁剪起始点高度。	0x0000
[15:0]	RW	lvds_start_x0_lane	第一帧裁剪起始点宽度。配置值为需要裁剪的起始像素点除以通道数。	0x0000

LVDS(N)_CROP_START1

LVDS(N)_CROP_START1 为 LVDS(N) 第二帧裁剪寄存器。

Offset Address: $0x1314 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lvds_start_y1	第二帧裁剪起始点高度。	0x0000
[15:0]	RW	lvds_start_x1_lane	第二帧裁剪起始点宽度。配置值为需要裁剪的起始像素点除以通道数。	0x0000

LVDS(N)_CROP_START2

LVDS(N)_CROP_START2 为 LVDS(N) 第三帧裁剪寄存器。

Offset Address: $0x1318 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lvds_start_y2	第三帧裁剪起始点高度。	0x0000
[15:0]	RW	lvds_start_x2_lane	第三帧裁剪起始点宽度。配置值为需要裁剪的起始像素点除以通道数。	0x0000

LVDS(N)_CROP_START3

LVDS(N)_CROP_START3 为 LVDS(N) 第四帧裁剪寄存器。

Offset Address: $0x131C + N \times 0x1000$ Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	RW	lvds_start_y3	第四帧裁剪起始点高度。	0x0000
[15:0]	RW	lvds_start_x3_lane	第四帧裁剪起始点宽度。配置值为需要裁剪的起始像素点除以通道数。	0x0000

LVDS(N)_LANE(M)_SOF_01

LVDS(N)_LANE(M)_SOF_01 为 LVDS/HiSPi 模式下，lane(M)帧起始同步码配置寄存器。

Offset Address: $0x1320 + M \times 0x20 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(M)_sof_1	LVDS、HiSPi 模式下帧 1 的帧起始同步码，lane(M) SOF of frame1; 仅在 WDR 模式下可配置，linear 模式下配置无效。	0x0000
[15:0]	RW	lane(M)_sof_0	LVDS、HiSPi 模式下帧 0 的帧起始同步码，lane(M) SOF of frame0; WDR 模式下为第 0 帧 SOF，linear 模式下为 SOF。	0x0000

LVDS(N)_LANE(M)_SOF_23

LVDS(N)_LANE(M)_SOF_23 为 LVDS/HiSPi 模式下，lane(M)帧起始同步码配置寄存器。

Offset Address: $0x1324 + M \times 0x20 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(M)_sof_3	LVDS、HiSPi 模式下帧 3 的帧起始同步码，lane(M) SOF of frame3; 仅在 WDR 模式下可配置，linear 模式下配置无效。	0x0000
[15:0]	RW	lane(M)_sof_2	LVDS、HiSPi 模式下帧 2 的帧起始同步码，lane(M) SOF of frame2; 仅在 WDR 模式下可配置，linear 模式下配置无效。	0x0000



LVDS(N)_LANE(M)_EOF_01

LVDS(N)_LANE(M)_EOF_01 为 LVDS/HiSPi 模式下, lane(M)帧结束同步码配置寄存器。

Offset Address: $0x1328 + M \times 0x20 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(M)_eof_1	LVDS、HiSPi 模式下帧 1 的帧结束同步码, lane(M) EOF of frame1; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000
[15:0]	RW	lane(M)_eof_0	LVDS、HiSPi 模式下帧 0 的 0 帧结束同步码, lane(M) EOF of frame0; WDR 模式下为第 0 帧 EOF, linear 模式下为 EOF。	0x0000

LVDS(N)_LANE(M)_EOF_23

LVDS(N)_LANE(M)_EOF_23 为 LVDS/HiSPi 模式下, lane(M)帧结束同步码配置寄存器。

Offset Address: $0x132C + M \times 0x20 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(M)_eof_3	LVDS、HiSPi 模式下帧 3 的帧结束同步码, lane(M) EOF of frame3; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000
[15:0]	RW	lane(M)_eof_2	LVDS、HiSPi 模式下帧 2 的帧结束同步码, lane(M) EOF of frame2; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000

LVDS(N)_LANE(M)_SOL_01

LVDS(N)_LANE(M)_SOL_01 为 LVDS/HiSPi 模式下, lane(M)行起始同步码配置寄存器。

Offset Address: $0x1330 + M \times 0x20 + N \times 0x1000$ Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	RW	lane(M)_sol_1	LVDS、HiSPi 模式下帧 1 的行起始同步码, lane(M) SOL of frame1; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000
[15:0]	RW	lane(M)_sol_0	LVDS、HiSPi 模式下帧 0 的行起始同步码, lane(M) SOL of frame0; WDR 模式下为第 0 帧 SOL, linear 模式下为 SOL。	0x0000

LVDS(N)_LANE(M)_SOL_23

LVDS(N)_LANE(M)_SOL_23 为 LVDS/HiSPi 模式下, lane(M)行起始同步码配置寄存器。

Offset Address: $0x1334 + M \times 0x20 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(M)_sol_3	LVDS、HiSPi 模式下帧 3 的行起始同步码, lane(M) SOL of frame3; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000
[15:0]	RW	lane(M)_sol_2	LVDS、HiSPi 模式下帧 2 的行起始同步码, lane(M) SOL of frame2; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000

LVDS(N)_LANE(M)_EOL_01

LVDS(N)_LANE(M)_EOL_01 为 LVDS/HiSPi 模式下, lane(M)行结束同步码配置寄存器。

Offset Address: $0x1338 + M \times 0x20 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(M)_eol_1	LVDS、HiSPi 模式下帧 1 的行结束同步码, lane(M) EOL of frame1; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000



[15:0]	RW	lane(M)_eol_0	LVDS、HiSPi 模式下帧 0 的行结束同步码，lane(M) EOL of frame0; WDR 模式下为第 0 帧 EOL，linear 模式下为 EOL。	0x0000

LVDS(N)_LANE(M)_EOL_23

LVDS(N)_LANE(M)_EOL_23 为 LVDS/HiSPi 模式下，lane(M)行结束同步码配置寄存器。

Offset Address: $0x133C + M \times 0x20 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(M)_eol_3	LVDS、HiSPi 模式下帧 3 的行结束同步码，lane(M) EOL of frame3; 仅在 WDR 模式下可配置，linear 模式下配置无效。	0x0000
[15:0]	RW	lane(M)_eol_2	LVDS、HiSPi 模式下帧 2 的行结束同步码，lane(M) EOL of frame2; 仅在 WDR 模式下可配置，linear 模式下配置无效。	0x0000

LVDS(N)_LANE(M)_NXT_SOF_01

LVDS(N)_LANE(M)_NXT_SOF_01 为 LVDS/HiSPi 模式下，第 N+1 帧 lane(M)帧起始同步码配置寄存器。

Offset Address: $0x1520 + M \times 0x20 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(M)_nxt_sof_1	LVDS、HiSPi 模式下帧 1 的帧起始同步码，lane(M) SOF of frame1; 仅在 WDR 模式下可配置，linear 模式下配置无效。	0x0000
[15:0]	RW	lane(M)_nxt_sof_0	LVDS、HiSPi 模式下帧 0 的帧起始同步码，lane(M) SOF of frame0; WDR 模式下为第 0 帧 SOF，linear 模式下为 SOF。	0x0000



LVDS(N)_LANE(M)_NXT_SOF_23

LVDS(N)_LANE(M)_NXT_SOF_23 为 LVDS/HiSPi 模式下, 第 N+1 帧 lane(M)帧起始同步码配置寄存器。

Offset Address: $0x1524 + M \times 0x20 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(M)_nxt_sof_3	LVDS、HiSPi 模式下帧 3 的帧起始同步码, lane(M) SOF of frame3; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000
[15:0]	RW	lane(M)_nxt_sof_2	LVDS、HiSPi 模式下帧 2 的帧起始同步码, lane(M) SOF of frame2; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000

LVDS(N)_LANE(M)_NXT_EOF_01

LVDS(N)_LANE(M)_NXT_EOF_01 为 LVDS/HiSPi 模式下, 第 N+1 帧 lane(M)帧结束同步码配置寄存器。

Offset Address: $0x1528 + M \times 0x20 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(M)_nxt_eof_1	LVDS、HiSPi 模式下帧 1 的帧结束同步码, lane(M) EOF of frame1; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000
[15:0]	RW	lane(M)_nxt_eof_0	LVDS、HiSPi 模式下帧 0 的 0 帧结束同步码, lane(M) EOF of frame0; WDR 模式下为第 0 帧 EOF, linear 模式下为 EOF。	0x0000

LVDS(N)_LANE(M)_NXT_EOF_23

LVDS(N)_LANE(M)_NXT_EOF_23 为 LVDS/HiSPi 模式下, 第 N+1 帧 lane(M)帧结束同步码配置寄存器。

Offset Address: $0x152C + M \times 0x20 + N \times 0x1000$ Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	RW	lane(M)_nxt_eof_3	LVDS、HiSPi 模式下帧 3 的帧结束同步码, lane(M) EOF of frame3; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000
[15:0]	RW	lane(M)_nxt_eof_2	LVDS、HiSPi 模式下帧 2 的帧结束同步码, lane(M) EOF of frame2; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000

LVDS(N)_LANE(M)_NXT_SOL_01

LVDS(N)_LANE(M)_NXT_SOL_01 为 LVDS/HiSPi 模式下, 第 N+1 帧 lane(M)行起始同步码配置寄存器。

Offset Address: $0x1530 + M \times 0x20 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(M)_nxt_sol_1	LVDS、HiSPi 模式下帧 1 的行起始同步码, lane(M) SOL of frame1; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000
[15:0]	RW	lane(M)_nxt_sol_0	LVDS、HiSPi 模式下帧 0 的行起始同步码, lane(M) SOL of frame0; WDR 模式下为第 0 帧 SOL, linear 模式下为 SOL。	0x0000

LVDS(N)_LANE(M)_NXT_SOL_23

LVDS(N)_LANE(M)_NXT_SOL_23 为 LVDS/HiSPi 模式下, 第 N+1 帧 lane(M)行起始同步码配置寄存器。

Offset Address: $0x1534 + M \times 0x20 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(M)_nxt_sol_3	LVDS、HiSPi 模式下帧 3 的行起始同步码, lane(M) SOL of frame3; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000



[15:0]	RW	lane(M)_nxt_sol_2	LVDS、HiSPi 模式下帧 2 的行起始同步码, lane(M) SOL of frame2; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000

LVDS(N)_LANE(M)_NXT_EOL_01

LVDS(N)_LANE(M)_NXT_EOL_01 为 LVDS/HiSPi 模式下, 第 N+1 帧 lane(M)行结束同步码配置寄存器。

Offset Address: $0x1538 + M \times 0x20 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(M)_nxt_eol_1	LVDS、HiSPi 模式下帧 1 的行结束同步码, lane(M) EOL of frame1; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000
[15:0]	RW	lane(M)_nxt_eol_0	LVDS、HiSPi 模式下帧 0 的行结束同步码, lane(M) EOL of frame0; WDR 模式下为第 0 帧 EOL, linear 模式下为 EOL。	0x0000

LVDS(N)_LANE(M)_NXT_EOL_23

LVDS(N)_LANE(M)_NXT_EOL_23 为 LVDS/HiSPi 模式下, 第 N+1 帧 lane(M)行结束同步码配置寄存器。

Offset Address: $0x153C + M \times 0x20 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(M)_nxt_eol_3	LVDS、HiSPi 模式下帧 3 的行结束同步码, lane(M) EOL of frame3; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000
[15:0]	RW	lane(M)_nxt_eol_2	LVDS、HiSPi 模式下帧 2 的行结束同步码, lane(M) EOL of frame2; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000



LVDS(N)_LI_WORD0

LVDS(N)_LI_WORD0 为 LVDS DOL 模式第 0 帧 LI 寄存器。

Offset Address: $0x1720 + N \times 0x1000$ Total Reset Value: 0x0211_0201

Bits	Access	Name	Description	Reset
[31:16]	RW	li_word0_1	DOL 模式 LEF 的 Line Information 的值 (N+1 Frame)。	0x0211
[15:0]	RW	li_word0_0	DOL 模式 LEF 的 Line Information 的值(N Frame)。	0x0201

LVDS(N)_LI_WORD1

LVDS(N)_LI_WORD1 为 LVDS DOL 模式第 1 帧 LI 寄存器。

Offset Address: $0x1724 + N \times 0x1000$ Total Reset Value: 0x0212_0202

Bits	Access	Name	Description	Reset
[31:16]	RW	li_word1_1	DOL 模式 SEF1 的 Line Information 的值 (N+1 Frame)。	0x0212
[15:0]	RW	li_word1_0	DOL 模式 SEF1 的 Line Information 的值(N Frame)。	0x0202

LVDS(N)_LI_WORD2

LVDS(N)_LI_WORD2 为 LVDS DOL 模式第 2 帧 LI 寄存器。

Offset Address: $0x1728 + N \times 0x1000$ Total Reset Value: 0x0214_0204

Bits	Access	Name	Description	Reset
[31:16]	RW	li_word2_1	DOL 模式 SEF2 的 Line Information 的值 (N+1 Frame)。	0x0214
[15:0]	RW	li_word2_0	DOL 模式 SEF2 的 Line Information 的值(N Frame)。	0x0204

LVDS(N)_LI_WORD3

LVDS(N)_LI_WORD3 为 LVDS DOL 模式第 3 帧 LI 寄存器。

Offset Address: $0x172C + N \times 0x1000$ Total Reset Value: 0x0218_0208



Bits	Access	Name	Description	Reset
[31:16]	RW	li_word3_1	DOL 模式 SEF3 的 Line Information 的值 (N+1 Frame)。	0x0218
[15:0]	RW	li_word3_0	DOL 模式 SEF3 的 Line Information 的值(N Frame)。	0x0208

LVDS(N)_IMGSIZE0_STATIS

LVDS(N)_IMGSIZE0_STATIS 为 LVDS LEF 图像宽高统计寄存器。

Offset Address: $0x1780 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	lvds_imgheight0	LVDS/HiSPi 模式下, virtual channel0 传输的图像高度。	0x0000
[15:0]	RO	lvds_imgwidth0	LVDS/HiSPi 模式下, virtual channel0 传输的图像宽度。	0x0000

LVDS(N)_IMGSIZE1_STATIS

LVDS(N)_IMGSIZE1_STATIS 为 LVDS SEF1 图像宽高统计寄存器。

Offset Address: $0x1784 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	lvds_imgheight1	LVDS/HiSPi 模式下, virtual channel1 传输的图像高度。	0x0000
[15:0]	RO	lvds_imgwidth1	LVDS/HiSPi 模式下, virtual channel1 传输的图像宽度。	0x0000

LVDS(N)_IMGSIZE2_STATIS

LVDS(N)_IMGSIZE2_STATIS 为 LVDS SEF2 图像宽高统计寄存器。

Offset Address: $0x1788 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	lvds_imgheight2	LVDS/HiSPi 模式下, virtual channel2 传输的图像高度。	0x0000



[15:0]	RO	lvds_imgwidth2	LVDS/HiSPi 模式下, virtual channel2 传输的图像宽度。	0x0000

LVDS(N)_IMGSIZE3_STATIS

LVDS(N)_IMGSIZE3_STATIS 为 LVDS SEF3 图像宽高统计寄存器。

Offset Address: $0x178C + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	lvds_imgheight3	LVDS/HiSPi 模式下, virtual channel3 传输的图像高度。	0x0000
[15:0]	RO	lvds_imgwidth3	LVDS/HiSPi 模式下, virtual channel3 传输的图像宽度。	0x0000

LVDS(N)_CTRL_INT_RAW

LVDS(N)_CTRL_INT_RAW 为 LVDS 读数据原始中断状态寄存器。

Offset Address: $0x17F0 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	保留。	0x0
[28]	WC	lvds_vsync_raw	lvds vsync 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[27]	WC	cmd_rd_err_raw	lvds CMD_FIFO 读错误原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[26]	WC	cmd_wr_err_raw	lvds CMD_FIFO 写错误原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[25]	WC	pop_err_raw	lvds 读取 line_buf 错误原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[24]	WC	lvds_state_err_raw	lvds 各 lane 同步错误原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0



Bits	Access	Name	Description	Reset
[23:21]	-	reserved	保留。	0x0
[20]	WC	link0_rd_err_raw	Link 读 FIFO 错误原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[19:17]	-	reserved	保留。	0x0
[16]	WC	link0_wr_err_raw	Link 写 FIFO 错误原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[15:4]	-	reserved	保留。	0x000
[3]	WC	lane3_sync_err_raw	Lane3 同步错误原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[2]	WC	lane2_sync_err_raw	Lane2 同步错误原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[1]	WC	lane1_sync_err_raw	Lane1 同步错误原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[0]	WC	lane0_sync_err_raw	Lane0 同步错误原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0

LVDS(N)_CTRL_INT

LVDS(N)_CTRL_INT 为 LVDS 读数据中断状态寄存器。

Offset Address: $0x17F4 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	保留。	0x0
[28]	RO	lvds_vsync_st	lvds vsync 中断状态。 0: 无中断; 1: 有中断。	0x0
[27]	RO	cmd_rd_err_st	lvds CMD_FIFO 读错误中断状态。	0x0



Bits	Access	Name	Description	Reset
			0: 无中断; 1: 有中断。	
[26]	RO	cmd_wr_err_st	lvds CMD_FIFO 写错误中断状态。 0: 无中断; 1: 有中断。	0x0
[25]	RO	pop_err_st	lvds 读取 line_buf 错误中断状态。 0: 无中断; 1: 有中断。	0x0
[24]	RO	lvds_state_err_st	lvds 各 lane 同步错误中断状态。 0: 无中断; 1: 有中断。	0x0
[23:21]	-	reserved	保留。	0x0
[20]	RO	link0_rd_err_st	Link 读 FIFO 错误中断状态。 0: 无中断; 1: 有中断。	0x0
[19:17]	RO	reserved	保留。	0x0
[16]	RO	link0_wr_err_st	Link 写 FIFO 错误中断状态。 0: 无中断; 1: 有中断。	0x0
[15:4]	-	reserved	保留。	0x000
[3]	RO	lane3_sync_err_st	Lane3 同步错误中断状态。 0: 无中断; 1: 有中断。	0x0
[2]	RO	lane2_sync_err_st	Lane2 同步错误中断状态。 0: 无中断; 1: 有中断。	0x0
[1]	RO	lane1_sync_err_st	Lane1 同步错误中断状态。 0: 无中断; 1: 有中断。	0x0
[0]	RO	lane0_sync_err_st	Lane0 同步错误中断状态。 0: 无中断; 1: 有中断。	0x0



LVDS(N)_CTRL_INT_MSK

LVDS(N)_CTRL_INT_MSK 为 LVDS 读数据中断屏蔽寄存器。

Offset Address: $0x17F8 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	保留。	0x0
[28]	RW	lvds_vsync_msk	lvds vsync 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[27]	RW	cmd_rd_err_msk	lvds CMD_FIFO 读错误中断状态使能。 0: 无中断; 1: 有中断。	0x0
[26]	RW	cmd_wr_err_msk	lvds CMD_FIFO 写错误中断状态使能。 0: 无中断; 1: 有中断。	0x0
[25]	RW	pop_err_msk	lvds 读取 line_buf 错误中断状态使能。 0: 无中断; 1: 有中断。	0x0
[24]	RW	lvds_state_err_msk	lvds 各 lane 同步错误中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[23:21]	-	reserved	保留。	0x0
[20]	RW	link0_rd_err_msk	Link 读 FIFO 错误中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[19:17]	-	reserved	保留。	0x0
[16]	RW	link0_wr_err_msk	Link 写 FIFO 错误中断状态使能。 0: 无原始中断; 1: 有原始中断。	0x0
[15:4]	-	reserved	保留。	0x000
[3]	RW	lane3_sync_err_msk	Lane3 同步错误中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RW	lane2_sync_err_msk	Lane2 同步错误中断状态使能。	0x0



Bits	Access	Name	Description	Reset
			0: 屏蔽中断; 1: 使能中断。	
[1]	RW	lane1_sync_err_msk	Lane1 同步错误中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	lane0_sync_err_msk	Lane0 同步错误中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0

LANE_ID0_CHN(N)

LANE_ID0_CHN(N)为 LANE 优先级配置寄存器 0。

Offset Address: $0x1800 + N \times 0x1000$ Total Reset Value: 0x0000_3210

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:12]	RW	lane3_id	指示 Lane3 的通道号，取值范围为 0~15。 取值表示为连接到 Lane3 的图像传感器的实际通道号。	0x3
[11:8]	RW	lane2_id	指示 Lane2 的通道号，取值范围为 0~15。 取值表示为连接到 Lane2 的图像传感器的实际通道号。	0x2
[7:4]	RW	lane1_id	指示 Lane1 的通道号，取值范围为 0~15。 取值表示为连接到 Lane1 的图像传感器的实际通道号。	0x1
[3:0]	RW	lane0_id	指示 Lane0 的通道号，取值范围为 0~15。 取值表示为连接到 Lane0 的图像传感器的实际通道号。	0x0

ALIGN(N)_INT_RAW

ALIGN(N)_INT_RAW 为 MIPI_ALIGN 原始中断状态寄存器。

Offset Address: $0x18F0 + N \times 0x1000$ Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x00000000
[4]	WC	err_lane3_raw	ALIGN lane3 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[3]	WC	err_lane2_raw	ALIGN lane2 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[2]	WC	err_lane1_raw	ALIGN lane1 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[1]	WC	err_lane0_raw	ALIGN lane0 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[0]	WC	err_full_raw	ALIGN FIFO 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0

ALIGN(N)_INT

ALIGN(N)_INT 为 MIPI_ALIGN 中断状态寄存器。

Offset Address: $0x18F4 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x00000000
[4]	RO	err_lane3_st	ALIGN lane3 中断状态。 0: 无中断; 1: 有中断。	0x0
[3]	RO	err_lane2_st	ALIGN lane2 中断状态。 0: 无中断; 1: 有中断。	0x0
[2]	RO	err_lane1_st	ALIGN lane1 中断状态。 0: 无中断; 1: 有中断。	0x0



[1]	RO	err_lane0_st	ALIGN lane0 中断状态。 0: 无中断; 1: 有中断。	0x0
[0]	RO	err_full_st	ALIGN FIFO 中断状态。 0: 无中断; 1: 有中断。	0x0

ALIGN(N)_INT_MSK

ALIGN(N)_INT_MSK 为 MIPI_ALIGN 中断屏蔽寄存器。

Offset Address: $0x18F8 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	err_lane3_mask	ALIGN lane3 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[3]	RW	err_lane2_mask	ALIGN lane2 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RW	err_lane1_mask	ALIGN lane1 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	RW	err_lane0_mask	ALIGN lane0 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	err_full_mask	ALIGN FIFO 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0

CHN(N)_INT_RAW

CHN(N)_INT_RAW 为通道(N)原始中断状态寄存器。

Offset Address: $0x1FF0 + N \times 0x1000$ Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000
[3]	WC	int_data_align_raw	DATA_ALIGN 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[2]	WC	int_mipi_ctrl_raw	MIPI_CTRL 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[1]	WC	int_mipi_csi_raw	MIPI_CSI 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[0]	WC	int_lvds_ctrl_raw	LVDS_CTRL 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0

CHN(N)_INT

CHN(N)_INT 为通道(N)中断状态寄存器。

Offset Address: $0x1FF4 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000
[3]	RO	int_data_align_st	DATA_ALIGN 中断状态。 0: 无中断; 1: 有中断。	0x0
[2]	RO	int_mipi_ctrl_st	MIPI_CTRL 中断状态。 0: 无中断; 1: 有中断。	0x0
[1]	RO	int_mipi_csi_st	MIPI_CSI 中断状态。 0: 无中断; 1: 有中断。	0x0
[0]	RO	int_lvds_ctrl_st	LVDS_CTRL 中断状态。 0: 无中断; 1: 有中断。	0x0



CHN_INT_MASK

CHN_INT_MASK 为通道中断(N)屏蔽寄存器。

Offset Address: $0x1FF8 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RW	int_data_align_mask	DATA_ALIGN 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RW	int_mipi_ctrl_mask	MIPI_CTRL 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	RW	int_mipi_csi_mask	MIPI_CSI 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	int_lvds_ctrl_mask	LVDS_CTRL 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0

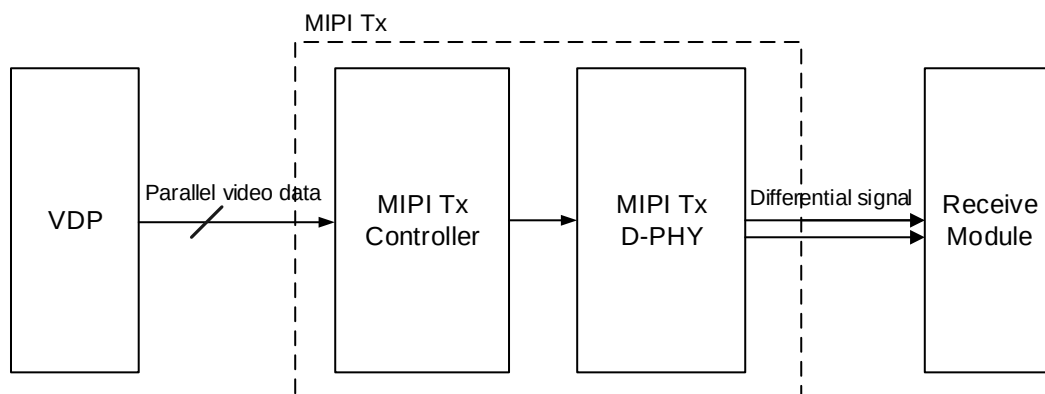
9.4 MIPI Tx

9.4.1 概述

The Display Serial Interface (DSI) 接口是移动行业处理器接口联盟 (Mobile Industry Processor Interface alliance, MIPI 联盟) 定义的一种高速串行接口, 主要用于处理器和显示模块之间的连接。Camera Serial Interface 2 (CSI-2) 是 MIPI 联盟定义的另一种高速接口, 主要用于图像采集模块和处理器之间的连接。MIPI Tx 接口实现的是 DSI 接口和 CSI-2 接口的发送端, 支持 MIPI D-PHY V1.0 串行信号输出。

MIPI Tx 包含模拟 PHY 和 Controller 两部分, 功能框图及在系统中的位置如图 9-34 所示。

图9-34 MIPI Tx 功能框图及在系统中的位置



9.4.2 特点

MIPI Tx 有以下特点：

- 最多支持 4-Lane MIPI D-PHY 接口，支持 Data Lane 个数可配置，顺序不可配
- 高速模式最大支持 1000Mbps/Lane
- 仅 Data Lane0 支持低速的发送和接收，Bus Turn-Around 功能。低速模式速率最高 10Mbps
- 支持 DSI RGB 16-bit/18-bit/24-bit 和 YUV4:2:0 12-bit/YUV 4:2:2 16-bit/YUV4:2:2 20-bit 数据类型的输出
- 支持 DSI video mode 和 command mode，video mode 支持 Burst mode、Non-burst Mode with Sync Pulses 和 Non-burst Mode with Sync Events
- 支持 CSI-2 Legacy YUV4:2:0 8-bit/YUV4:2:0 8-bit/YUV4:2:2 8-bit/数据类型的输出

9.4.3 功能描述

MIPI Tx 包括 Tx D-PHY 和 Tx Controller 两部分。

- Tx D-PHY 支持 MIPI D-PHY ver1.0 协议，主要实现了物理层的传输规范。
- Tx Controller 根据 MIPI DSI 和 CSI-2 协议要求对数据格式进行封装。

9.4.3.1 Tx D-PHY

Tx D-PHY 可以工作在低速和高速（High Speed, HS）两种模式下：

- video mode/CSI-2 的数据通过高速模式传输，
- command mode 的数据可以通过高速或者低速模式来传输。

高速模式每个通道（lane）的数据速率范围为 80~1000Mbps，低速模式的速率最大为 10Mbps。

高速模式最大支持 4 个 data lane，实际使用的 data lane 个数可以配置，顺序不可配。当仅使用一个 data lane 时，data lane0 有效；当使用 2 个 data lane 时，data lane0 和 data lane1 有效；当使用 3 个 data lane 时，data lane0、data lane1 和 data lane2 有效；当使用



4 个 data lane 时，data lane0、data lane1、data lane2 和 data lane3 有效。仅 data lane0 支持低速模式的发送、接收和 Bus Turn-Around。

9.4.3.2 Tx Controller

数据包的发送

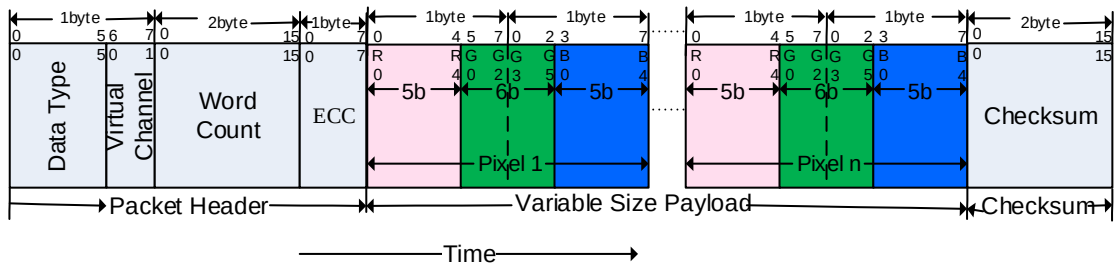
当多个数据包需要发送时，控制器会根据间隔时间自动进行 HS 和 LP 模式的切换。控制器支持 EoT 包（End of Transmission, EoTp）发送的使能和关闭，具体配置寄存器 PCKHDL_CFG bit[0]，可以在每个 HS 传输结束时发送一个 EoT 包，也可以不发送。

数据类型

控制器支持 DSI RGB 16-bit/18-bit/24-bit 和 YUV4:2:0 12-bit/YUV4:2:2 16-bit/YUV4:2:2 20-bit 的发送，支持 CSI-2 Legacy YUV420 8-bit/YUV420 8-bit/YUV422 8-bit 的发送，各种数据类型的数据组成格式如图 9-35~图 9-44 所示。

DSI RGB 16-bit，每两个字节包含一个像素。

图9-35 RGB 16-bit 格式



DSI RGB 18-bit，分为两种时序，一种是连续排列的（non-loosely），每 4 个 pixel 占用 9 个字节，每个数据包的 pixel 数是 4 的倍数，如图 9-21 所示。另一种是每个分量占用一个字节（loosely），一个像素占用 3 个字节，每个字节的低两 bit 是无效的，如图 9-22 所示。



图9-36 RGB 18-bit non-loosely 格式

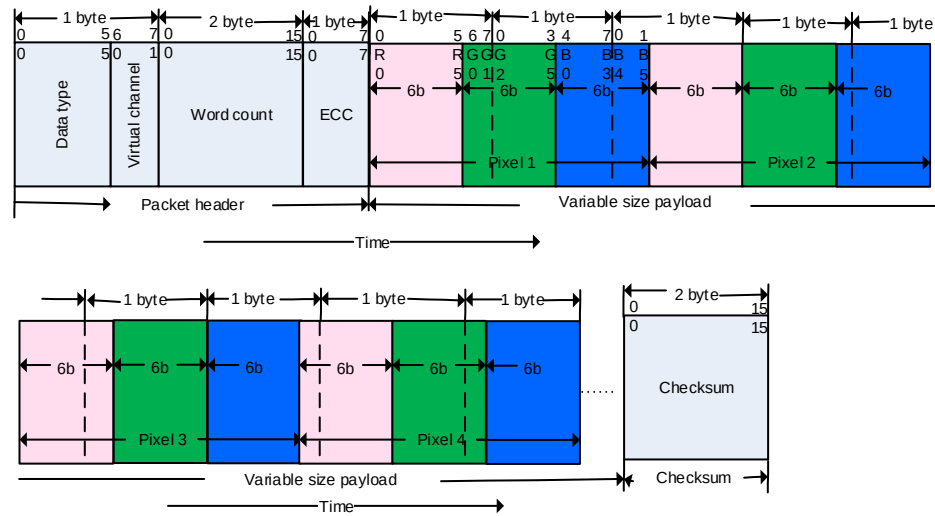
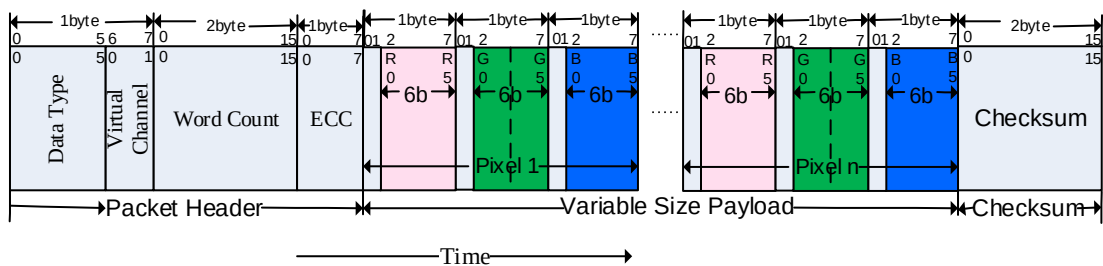
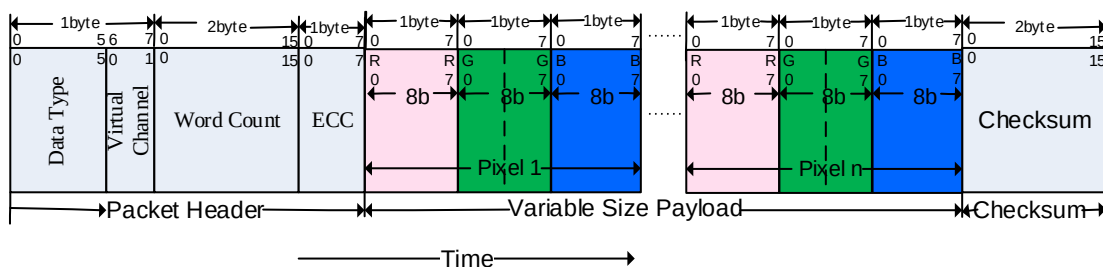


图9-37 RGB 18-bit loosely 格式



DSI RGB 24-bit, 一个分量占用一个字节, 一个 pixel 占用 3 个字节。

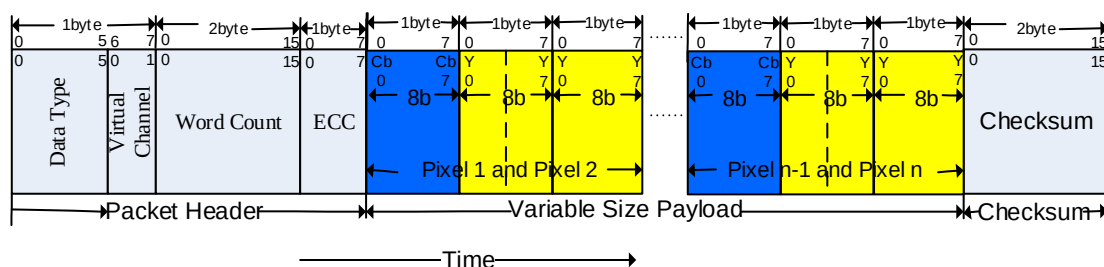
图9-38 RGB 24-bit 格式



DSI YUV4:2:0 12-bit, 奇数行发 Cb 和 Y 分量, 偶数行发 Cr 和 Y 分量。每个数据包的像素个数是 2 的倍数。

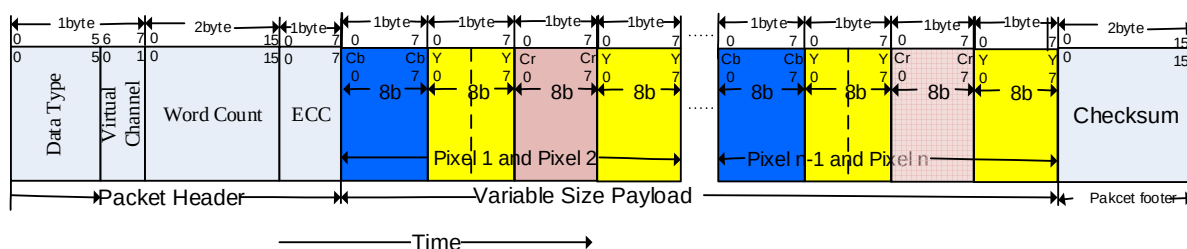


图9-39 YUV4:2:0 12-bit 格式



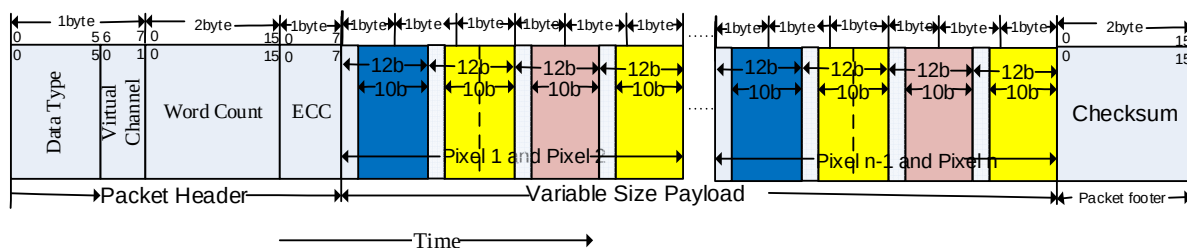
DSI YUV4:2:2 16-bit, 每行的 Cb 和 Cr 分量交替传输。每个数据包的像素个数是 2 的倍数。

图9-40 YUV4:2:2 16-bit 格式



DSI YUV4:2:2 20-bit, 每行的 Cb 和 Cr 分量交替传输。每个 10bit 的有效分量之间插入 2bit 的 0。每个数据包的像素个数是 2 的倍数。

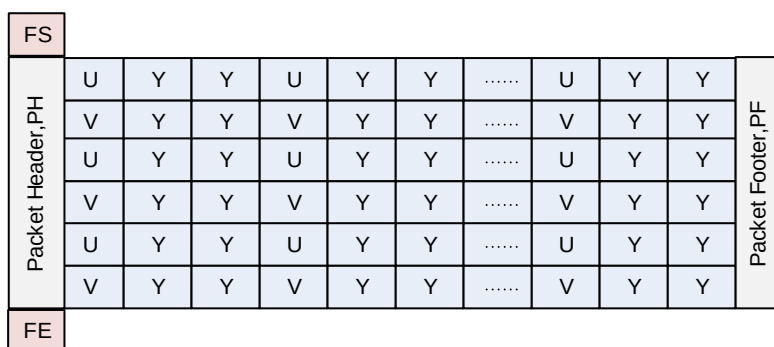
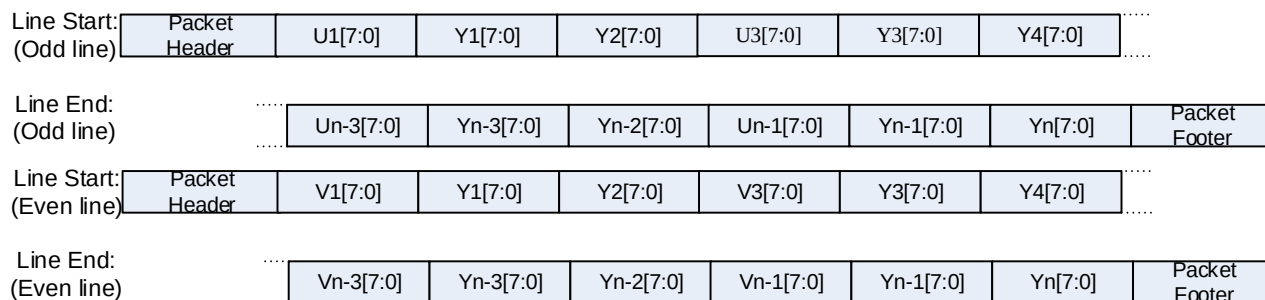
图9-41 YUV4:2:2 20-bit 格式



CSI-2 Legacy YUV420 8-bit, 奇数行发 U 和 Y 分量，偶数行发 V 和 Y 分量。每个数据包的像素个数是 2 的倍数。

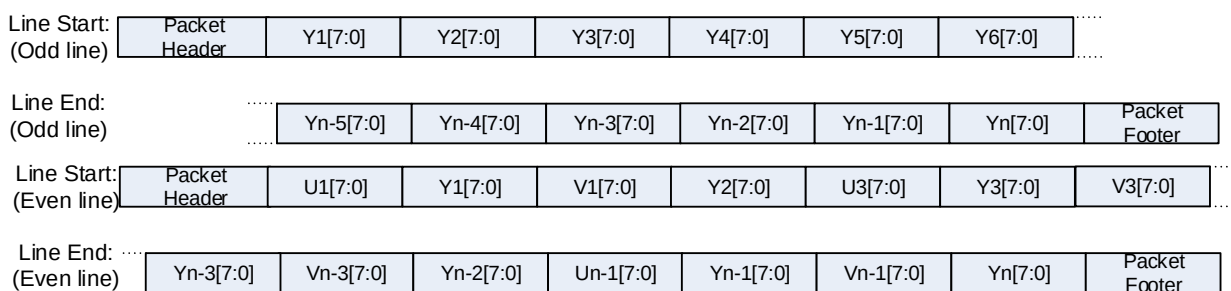


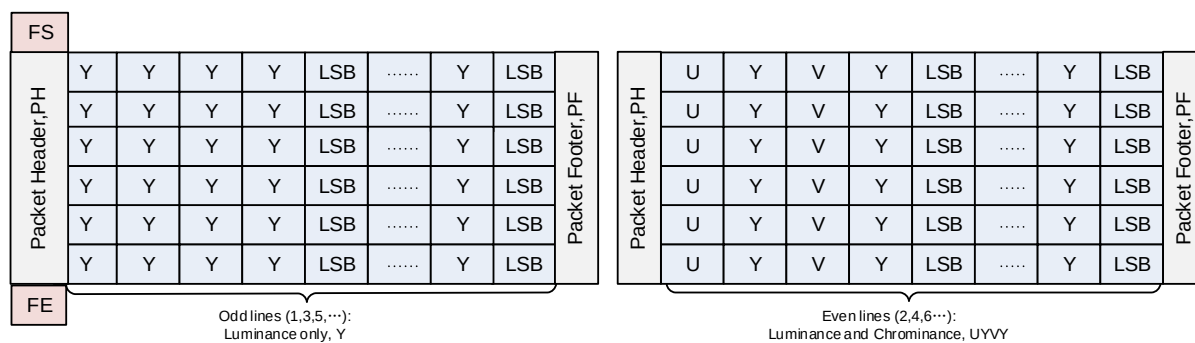
图9-42 Legacy YUV420 8-bit 格式



CSI-2 YUV420 8-bit, 奇数行只发 Y 分量, 偶数行发 U、V 和 Y 分量, 奇数行的有效数据是偶数行的有效数据的一半。每个数据包的像素个数是 2 的倍数。

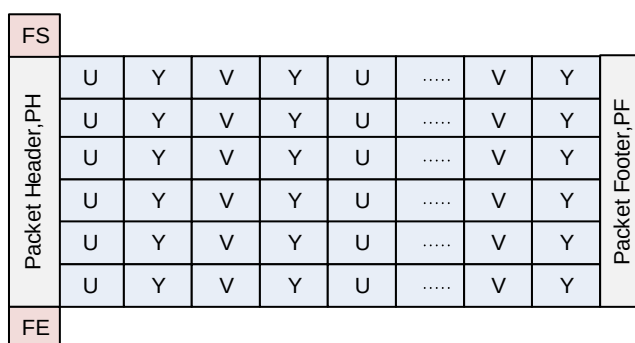
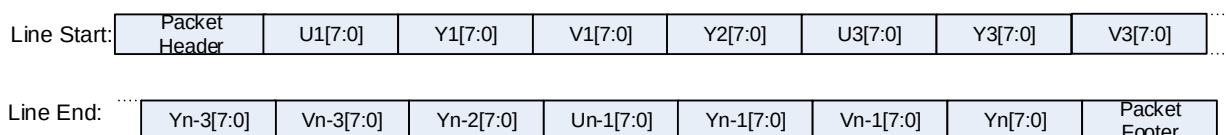
图9-43 YUV420 8-bit 格式





CSI-2 YUV422 8-bit，每行的 U 和 V 分量交替传输。每个数据包的像素个数是 2 的倍数。

图9-44 YUV422 8-bit 格式



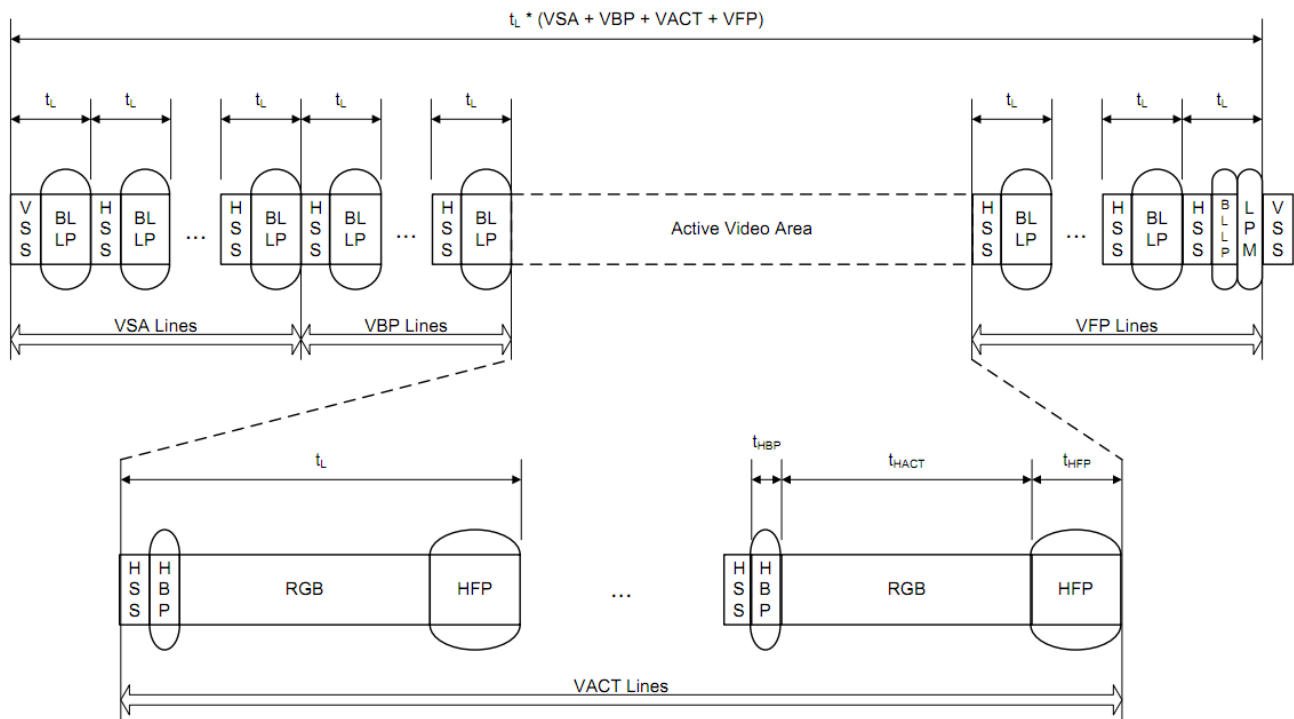
接口时序

Video mode 的 Burst Mode 时序如图 9-45 所示，一帧图像包含 VSA，VBP，VACT 和 VFP 四部分，VSA 是帧 sync active 行，VBP 是帧后消隐行，VACT 是有效数据行，VFP 是帧前消隐行。有效数据行包括行 sync start，行后消隐区，行有效区，BLLP (Blanking or Low-Power Interval)，行前消隐区五部分，无效数据行只包含行 sync start 和 BLLP。使用 Burst Mode 传输数据可以充分利用带宽，在有效数据行可以进入 BLLP 区，此时 PHY 处于 LP 模式，降低功耗。



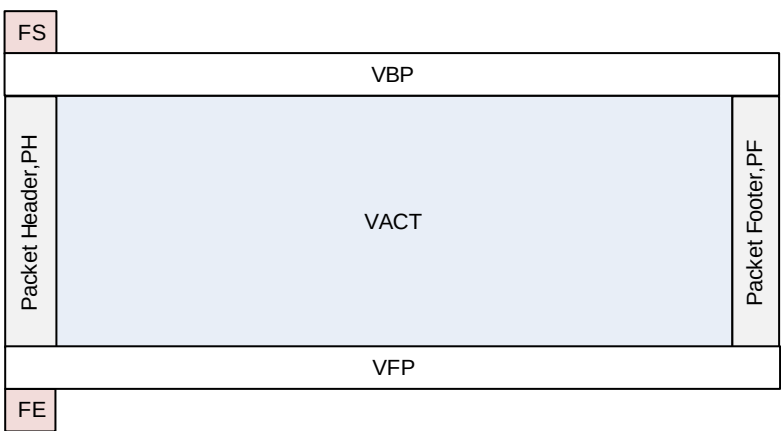
Video mode 的 Non-Burst Mode with Sync Events 时序如图 9-47 所示，一帧图像包含 VSA, VBP, VACT 和 VFP 四部分。有效数据行包括行 sync start, 行后消隐区, 行有效区, 行前消隐区四部分, 无效数据行只包含行 sync start 和 BLLP。

图9-47 Non-Burst Mode with Sync Events



CSI-2 发送的帧时序如图 9-48 所示，FS 包和有效像素的长包之间是后消隐区，后消隐区大小可配，最小是 1 行。FE 包可以在 VACT 区结束后的第一行发送，也可以在帧前消隐区 VFP 的最后发送。

图9-48 CSI-2 帧时序





9.4.4 配置流程

软件操作流程如下：

步骤 1. 打开时钟门控，配置 PHY 的参考时钟，输出时钟频率，时间参数，Lane 个数。

通过时钟复位配置寄存器 PERI_CRG67 打开时钟门控，撤销复位，选择 PHY PLL 的参考时钟。

Lane 的个数由 [PHY_IF_CFG](#) 配置。PHY 的寄存器配置通过控制器的寄存器来完成，配置方法为（这里为一个 PHY 寄存器的配置方法，当需要配置多个寄存器时，每个寄存器的配置方法类似）：

```
PHY_TST_CTRL1 = 0x100XX (XX为PHY的寄存器地址)
PHY_TST_CTRL0 = 0x2
PHY_TST_CTRL0 = 0x0
PHY_TST_CTRL1 = 0xYY (YY为PHY的寄存器XX的配置值)
PHY_TST_CTRL0 = 0x2
PHY_TST_CTRL0 = 0x0
```

当需要读 PHY 的寄存器时，方法为执行下面的配置后，读 [PHY_TST_CTRL1](#) 寄存器的值，该寄存器的 bit15~bit8 即是 PHY XX 寄存器的值（这里为一个 PHY 寄存器的读方法，当需要读多个寄存器时，每个寄存器的读方法类似）：

```
PHY_TST_CTRL1 = 0x100XX (XX为PHY的寄存器地址)
PHY_TST_CTRL0 = 0x2
PHY_TST_CTRL0 = 0x0
```

PHY 的输出时间参数需要配置的寄存器有：

- **clk lane:** [PRE_DELAY_HSCLK](#), [POST_DELAY_HSCLK](#), [TLPX_CLK](#), [TCK_PREPARE](#), [TCLK-ZERO](#), [TCLK-TRAIL](#), [TWAKEUP-CLK](#);
- **datalane**（以下是 datalane0 的寄存器，其他 datalane 类似）：[PRE_DELAY_D0HS](#), [POST_DELAY_D0HS](#), [TLPX_D0](#), [THS_PREPARE-D0](#), [THS-ZERO-D0](#), [THS-TRAIL-D0](#), [TTA-GO-D0](#), [TTA-GET-D0](#), [TWAKEUP-D0](#)。

应该根据不同的场景来配置，寄存器具体描述请参考 [9.4.6 MIPI Tx 寄存器描述](#) 章节。

PHY 输出的时钟频率通过配置 PHY 的 PLL 来完成，输出数据的速率范围是 80Mbps~1000Mbps。PLL 的参考时钟有两档可选：19.8MHz 和 27MHz。PLL 配置涉及到的寄存器有 [PLL_SET0](#)、[PLL_SET1](#)、[PLL_SET2](#)、[PLL_SET4](#)、[PLL_SET5](#)。

$$\text{Data Rate} = f_{\text{ref\_clk}} * \text{pll\_m} * 2 / (\text{pll\_n} * \text{pll\_q})$$

其中：

```
pll_m = RG_PLL_PRE_P*2+RG_PLL_FBD_S
pll_n = RG_PLL_FBD_P
pll_q = (RG_HSTX_CKG_SEL == 3'h4) ? 2 : (RG_HSTX_CKG_SEL == 3'h5) ? 4 :
        (RG_HSTX_CKG_SEL == 3'h6) ? 8 : (RG_HSTX_CKG_SEL == 3'h7) ? 16 : 1
```

$f_{\text{ref\_clk}}$ 为 PHY 参考时钟频率。



步骤 2. 如果对接设备需要控制器配置属性，配置设备属性。

DSI 接口的设备，设备属性一般也是由 D-PHY 接口配置，配置命令可通过高速和低速模式完成，由寄存器 [CMD_MODE_CFG](#) 选择。

配置方法：配置命令写入寄存器 GEN_HDR，长包的配置数据写入寄存器 [GEN_PLD_DATA](#)，一个命令的数据最多支持 800 字节。先写配置数据，再写配置命令。

用 generic long write (data type: 29h) 写 6 个字节数据：

GEN_PLD_DATA = 32'h04030201; GEN_PLD_DATA = 32'h00000605; GEN_HDR = 32'h00000629;

用 DCS Short WRITE, 1parameter (data type: 15h) 写数据 35h: GEN_HDR = 32'h00003515;

当用读命令获取设备信息时，读操作完成后（可查看 [CMD_PKT_STATUS](#) 的 bit6，为 1 表示正在进行读操作），通过 [GEN_PLD_DATA](#) 获取读回的信息，当获取的设备信息大于 4 字节时，每读取一次 [GEN_PLD_DATA](#) 返回 4 字节。一个读命令支持的最大字节数是 160。

步骤 3. 配置控制器工作模式，分辨率。

- 控制器支持多种工作模式，应根据需要选择不同的模式，涉及的寄存器有 [MODE_CFG](#)，[VID_MODE_CFG](#)，[CMD_MODE_CFG](#)，[CSI_CTRL](#)，[DATATYPE0](#)，[OPERATION_MODE](#)。
- 分辨率的配置涉及到的寄存器有 [VID_PKT_SIZE](#)，[VID_NUM_CHUNKS](#)，[VID_NULL_SIZE](#)，[VID_HSA_TIME](#)，[VID_HBP_TIME](#)，[VID_HLINE_TIME](#)，[VID_VSA_LINES](#)，[VID_VBP_LINES](#)，[VID_VFP_LINES](#)，[VID_VACTIVE_LINES](#)，[EDPI_CMD_SIZE](#)，[PHY_TMR_LPCLK_CFG](#)，[PHY_TMR_CFG](#)。

应注意各参数的单位，详细参考寄存器描述。

选择 Non-Burst mode 时，从前级模块 VDP 输入的带宽和控制器输出的带宽需要满足下面表达式， $hscclkperiod = 1 / (Data\ Rate / 8)$ ，vid_num_chunks 由寄存器 [VID_NUM_CHUNKS](#) 配置，vid_pkt_size 由寄存器 [VID_PKT_SIZE](#) 配置，vid_null_size 由寄存器 [VID_NULL_SIZE](#) 配置，bytes_per_pixel 是每个 pixel 占用的字节数，vdpcclkperiod 是 MIPI DSI Tx 输入的时钟，由寄存器 [PERI_CRG67](#) 配置。

$$hscclkperiod * vid\_num\_chunks (vid\_pkt\_size * bytes\_per\_pixel + 12 + vid\_null\_size)$$

/number_of_lanes = pixels_per_line * vdpcclkperiod 当 null packets 使能;

$$hscclkperiod * vid\_num\_chunks (vid\_pkt\_size * bytes\_per\_pixel + 6)$$

/number_of_lanes = pixels_per_line * vdpcclkperiod 当 null packets 不使能;

选择 Burst mode、CSI-2 时，需要满足：

$$hscclkperiod * (vid\_pkt\_size * bytes\_per\_pixel + 6)$$

/number_of_lanes < pixels_per_line * vdpcclkperiod

步骤 4. 配置前面模块（VDP），给 TX 输入视频内容。



----结束

9.4.5 MIPI Tx 寄存器概览

MPI Tx 寄存器概览如表 9-13 所示。

表9-13 MIPI Tx 寄存器概览（基址是 0x1127_0000）

偏移地址	名称	描述	页码
0x0004	PWR_UP	复位寄存器	9-212
0x0008	CLKMGR_CFG	时钟分频寄存器	9-213
0x000C	VCID	virtual channel ID 配置寄存器	9-213
0x0010	COLOR_CODING	数据格式配置寄存器	9-213
0x0018	LP_CMD_TIM	video 模式 LP 命令可用时间配置寄存器	9-214
0x002C	PCKHDL_CFG	包参数配置寄存器	9-214
0x0030	GEN_VCID	Generic 接口读数据的 virtual channel ID 配置寄存器	9-215
0x0034	MODE_CFG	video 和 command 模式选择寄存器	9-215
0x0038	VID_MODE_CFG	video 模式操作配置寄存器	9-216
0x003C	VID_PKT_SIZE	packet 大小配置寄存器	9-217
0x0040	VID_NUM_CHUNKS	一行 chunks 个数配置寄存器	9-218
0x0044	VID_NULL_SIZE	null packet 大小配置寄存器	9-218
0x0048	VID_HSA_TIME	HSA 时间配置寄存器	9-218
0x004C	VID_HBP_TIME	HBP 时间配置寄存器	9-218
0x0050	VID_HLINE_TIME	HLINE 时间配置寄存器	9-219
0x0054	VID_VSA_LINES	VSA 行数配置寄存器	9-219
0x0058	VID_VBP_LINES	VBP 行数配置寄存器	9-219
0x005C	VID_VFP_LINES	VFP 行数配置寄存器	9-219
0x0060	VID_VACTIVE_LINES	VACTIVE 行数配置寄存器	9-220
0x0064	EDPI_CMD_SIZE	WMC SIZE 配置寄存器	9-220
0x0068	CMD_MODE_CFG	command mode 操作配置寄存器	9-220
0x006C	GEN_HDR	Generic interface header 设置寄存器	9-222



偏移地址	名称	描述	页码
0x0070	GEN_PLD_DATA	Generic interface 数据寄存器	9-222
0x0074	CMD_PKT_STATUS	Generic interface FIFO 状态寄存器	9-223
0x0088	LP_WR_TO_CNT	LP 写回复超时配置寄存器	9-223
0x008C	BTA_TO_CNT	BTA 回复超时配置寄存器	9-224
0x0094	LPCLK_CTRL	clklane 控制寄存器	9-224
0x0098	PHY_TMR_LPCLK_CFG	clklane HS 和 LP 切换时间配置寄存器	9-224
0x009C	PHY_TMR_CFG	datalane HS 和 LP 切换时间配置寄存器	9-225
0x00A0	PHY_RSTZ	PHY PLL 和复位配置寄存器	9-225
0x00A4	PHY_IF_CFG	PHY 接口配置寄存器	9-226
0x00A8	PHY_ULPS_CTRL	PHY 的 ULPS 状态配置寄存器	9-226
0x00B0	PHY_STATUS	PHY 状态寄存器	9-227
0x00B4	PHY_TST_CTRL0	PHY 配置接口 0 寄存器	9-228
0x00B8	PHY_TST_CTRL1	PHY 配置接口 1 寄存器	9-229
0x00BC	INT_ST0	中断状态 0 寄存器	9-229
0x00C0	INT_ST1	中断状态 1 寄存器	9-232
0x00C4	INT_MSK0	中断 0 屏蔽寄存器	9-234
0x00C8	INT_MSK1	中断 1 屏蔽寄存器	9-236
0x0208	OPERATION_MODE	工作模式控制寄存器	9-238
0x0224	V_H_SEND	发送的宽高寄存器	9-239
0x0230	DATATYPE0	datatype 配置寄存器 0	9-239
0x0238	CSI_CTRL	CSI 时序配置寄存器	9-240

9.4.6 MIPI Tx 寄存器描述



说明

寄存器描述中的 hsclk period=1/(Data Rate/8)。

PWR_UP

PWR_UP 为复位寄存器。

Offset Address: 0x0004 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	pwr_up	复位使能。 0: 复位; 1: 撤销复位。	0x0

CLKMGR_CFG

CLKMGR_CFG 为时钟分频寄存器。

Offset Address: 0x0008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	tx_esc_clk_division	TX escape clock 分频倍数，时钟源为 hsclock，配 0 和 1 将不会产生时钟。	0x00

VCID

VCID 为 virtual channel ID 配置寄存器。

Offset Address: 0x000C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1:0]	RW	vcid	virtual channel ID 配置值。	0x0

COLOR_CODING

COLOR_CODING 为数据格式配置寄存器。

Offset Address: 0x0010 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:9]	-	reserved	保留。	0x000000
[8]	RW	loosely18_en	loosely packed 18-bit 配置使能。 0: 不使能; 1: 使能。	0x0



Bits	Access	Name	Description	Reset
[7:4]	-	reserved	保留。	0x0
[3:0]	RW	color_coding	数据格式配置，command mode 不支持 RGB30-bit 和 YUV 格式。 0000: DSI RGB16-bit; 0001: CSI YUV422 8-bit; 0010: CSI YUV422 10-bit; 0011: DSI RGB18-bit; 0101: DSI RGB24-bit; 0110: DSI YUV422 20-bit; 1000: DSI YUV422 16-bit; 1001: DSI RGB30-bit; 1011: DSI YUV420 12-bit; 1100: CSI YUV420 8-bit legacy; 1101: CSI YUV420 8-bit; 1110: CSI YUV420 10-bit; 其他：保留。	0x0

LP_CMD_TIM

LP_CMD_TIM 为 video 模式 LP 命令可用时间配置寄存器。

Offset Address: 0x0018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:16]	RW	outvact_lpcmd_time	VSA、VBP 和 VFP 区域可以发送的 LP 命令的最大字节数。	0x00
[15:8]	-	reserved	保留。	0x00
[7:0]	RW	invact_lpcmd_time	VACT 区域可以发送的 LP 命令的最大字节数。	0x00

PCKHDL_CFG

PCKHDL_CFG 为包参数配置寄存器。

Offset Address: 0x002C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x00000000
[4]	RW	crc_rx_en	CRC 接收使能。 0: 不使能; 1: 使能。	0x0
[3]	RW	ecc_rx_en	ECC 接收使能。 0: 不使能; 1: 使能。	0x0
[2]	RW	bta_en	Bus Turn-Around 请求。 0: 不使能; 1: 使能。	0x0
[1]	RW	eotp_rx_en	EoTp 接收使能。 0: 不使能; 1: 使能。	0x0
[0]	RW	eotp_tx_en	EoTp 发送使能。 0: 不使能; 1: 使能。	0x0

GEN_VCID

GEN_VCID 为 Generic 接口读数据的 virtual channel ID 配置寄存器。

Offset Address: 0x0030 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1:0]	RW	gen_vcid_rx	Generic 接口读数据的 virtual channel ID 值。	0x0

MODE_CFG

MODE_CFG 为 video 和 command 模式选择寄存器。

Offset Address: 0x0034 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000



Bits	Access	Name	Description	Reset
[0]	RW	cmd_video_mode	video 和 command 模式选择。 0: video 模式; 1: command 模式。	0x0

VID_MODE_CFG

VID_MODE_CFG 为 video 模式操作配置寄存器。

Offset Address: 0x0038 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:25]	-	reserved	保留。	0x00
[24]	RW	vpg_orientation	colorbar 方向选择。 0: 垂直模式; 1: 水平模式。	0x0
[23:21]	-	reserved	保留。	0x0
[20]	RW	vpg_mode	test pattern 模式选择。 0: colorbar; 1: BER pattern。	0x0
[19:17]	-	reserved	保留。	0x0
[16]	RW	vpg_en	video 模式 test pattern 使能。 0: 不使能; 1: 使能。	0x0
[15]	RW	lp_cmd_en	使用 LP 模式传输命令使能。 0: 不使能; 1: 使能。	0x0
[14]	RW	frame_bta_ack_en	帧结束 ACK 回复请求使能。 0: 不使能; 1: 使能。	0x0
[13]	RW	lp_hfp_en	如果时间足够, HFP 区域返回 low-power 状态使能。 0: 不使能; 1: 使能。	0x0
[12]	RW	lp_hbp_en	如果时间足够, HBP 区域返回 low-power 状态使能。	0x0



Bits	Access	Name	Description	Reset
			0: 不使能; 1: 使能。	
[11]	RW	lp_vact_en	如果时间足够, VACT 区域返回 low-power 状态使能。 0: 不使能; 1: 使能。	0x0
[10]	RW	lp_vfp_en	如果时间足够, VFP 区域返回 low-power 状态使能。 0: 不使能; 1: 使能。	0x0
[9]	RW	lp_vbp_en	如果时间足够, VBP 区域返回 low-power 状态使能。 0: 不使能; 1: 使能。	0x0
[8]	RW	lp_vsa_en	如果时间足够, VSA 区域返回 low-power 状态使能。 0: 不使能; 1: 使能。	0x0
[7:2]	-	reserved	保留。	0x00
[1:0]	RW	vid_mode_type	video 模式传输类型选择。 00: Non-burst with sync pulse; 01: Non-burst with sync events; 其他: Burst mode。	0x0

VID_PKT_SIZE

VID_PKT_SIZE 为 packet 大小配置寄存器。

Offset Address: 0x003C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13:0]	RW	vid_pkt_size	packet 大小, 单位 pixel。	0x0000



VID_NUM_CHUNKS

VID_NUM_CHUNKS 为一行 chunks 个数配置寄存器。

Offset Address: 0x0040 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12:0]	RW	vid_num_chunks	一行的 chunks 个数，一个 chunks 包括一个数据 packet 和一个 null packet。如果为 0，一行只有一个数据 packet；如果为 1，一行为一个数据 packet+一个 null packet(如果 vid_null_size>0)。	0x0000

VID_NULL_SIZE

VID_NULL_SIZE 为 null packet 大小配置寄存器

Offset Address: 0x0044 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12:0]	RW	vid_null_size	null packet 大小，单位字节。如果为 0，不发送 null packet。	0x0000

VID_HSA_TIME

VID_HSA_TIME 为 HSA 时间配置寄存器。

Offset Address: 0x0048 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:12]	-	reserved	保留。	0x00000
[11:0]	RW	vid_hsa_time	HSA 周期，单位 hsclock cycles。	0x000

VID_HBP_TIME

VID_HBP_TIME 为 HBP 时间配置寄存器。

Offset Address: 0x004C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:12]	-	reserved	保留。	0x00000
[11:0]	RW	vid_hbp_time	HBP 周期, 单位 hclk cycles。	0x000

VID_HLINE_TIME

VID_HLINE_TIME 为 HLINE 时间配置寄存器。

Offset Address: 0x0050 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:0]	RW	vid_hline_time	一行总周期(HSA+HBP+HACT+HFP), 单位 hclk cycles。	0x0000

VID_VSA_LINES

VID_VSA_LINES 为 VSA 行数配置寄存器。

Offset Address: 0x0054 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x000000
[9:0]	RW	vsa_lines	VSA 行数。	0x000

VID_VBP_LINES

VID_VBP_LINES 为 VBP 行数配置寄存器。

Offset Address: 0x0058 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x000000
[9:0]	RW	vbp_lines	VBP 行数。	0x000

VID_VFP_LINES

VID_VFP_LINES 为 VFP 行数配置寄存器。



Offset Address: 0x005C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x000000
[9:0]	RW	vfp_lines	VFP 行数。	0x000

VID_VACTIVE_LINES

VID_VACTIVE_LINES 为 VACTIVE 行数配置寄存器。

Offset Address: 0x0060 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13:0]	RW	v_active_lines	VACTIVE 行数。	0x0000

EDPI_CMD_SIZE

EDPI_CMD_SIZE 为 WMC SIZE 配置寄存器。

Offset Address: 0x0064 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	edpi_allowed_cmd_size	write memory command 最大 size 值，单位 pixel。	0x0000

CMD_MODE_CFG

CMD_MODE_CFG 为 command mode 操作配置寄存器。

Offset Address: 0x0068 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:25]	-	reserved	保留。	0x00
[24]	RW	max_rd_pkt_size	maximun read packet size command 传输类型。 0: high-speed; 1: low-power。	0x0



Bits	Access	Name	Description	Reset
[23:20]	-	reserved	保留。	0x0
[19]	RW	dcslw_tx	DCS long write packet command 传输类型。 0: high-speed; 1: low-power。	0x0
[18]	RW	dcssr_0p_tx	DCS short read packet with zero parameter command 传输类型。 0: high-speed; 1: low-power。	0x0
[17]	RW	dcssw_1p_tx	DCS short write packet with one parameter command 传输类型。 0: high-speed; 1: low-power。	0x0
[16]	RW	dcssw_0p_tx	DCS short write packet with zero parameter command 传输类型。 0: high-speed; 1: low-power。	0x0
[15]	-	reserved	保留。	0x0
[14]	RW	genlw_tx	Generic long write packet command 传输类型。 0: high-speed; 1: low-power。	0x0
[13]	RW	gensr_2p_tx	Generic short read packet with two parameters command 传输类型。 0: high-speed; 1: low-power。	0x0
[12]	RW	gensr_1p_tx	Generic short read packet with one parameters command 传输类型。 0: high-speed; 1: low-power。	0x0
[11]	RW	gensr_0p_tx	Generic short read packet with zero parameters command 传输类型。 0: high-speed; 1: low-power。	0x0
[10]	RW	gensw_2p_tx	Generic short write packet with two parameters command 传输类型。 0: high-speed;	0x0



Bits	Access	Name	Description	Reset
			1: low-power。	
[9]	RW	gen_sw_1p_tx	Generic short write packet with one parameters command 传输类型。 0: high-speed; 1: low-power。	0x0
[8]	RW	gen_sw_0p_tx	Generic short write packet with zero parameters command 传输类型。 0: high-speed; 1: low-power。	0x0
[7:0]	-	reserved	保留。	0x0000

GEN_HDR

GEN_HDR 为 Generic interface header 设置寄存器。

Offset Address: 0x006C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:16]	RW	gen_wc_msbyte	配置长包长度高字节，或者短包的 data 1。	0x00
[15:8]	RW	gen_wc_lsbyte	配置长包长度低字节，或者短包的 data 0。	0x00
[7:6]	RW	gen_vc	包的 virtual channel ID。	0x0
[5:0]	RW	gen_dt	包的 data type。	0x00

GEN_PLD_DATA

GEN_PLD_DATA 为 Generic interface 数据寄存器。

Offset Address: 0x0070 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RW	gen_pld_b4	包数据的第 4 字节。	0x00
[23:16]	RW	gen_pld_b3	包数据的第 3 字节。	0x00
[15:8]	RW	gen_pld_b2	包数据的第 2 字节。	0x00
[7:0]	RW	gen_pld_b1	包数据的第 1 字节。	0x00



CMD_PKT_STATUS

CMD_PKT_STATUS 为 Generic interface FIFO 状态寄存器。

Offset Address: 0x0074 Total Reset Value: 0x0000_0010

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	保留。	0x0000000
[6]	RW	gen_rd_cmd_busy	Generic interface 读忙标志。 0: 空闲; 1: 忙。	0x0
[5]	RW	gen_pld_r_full	Generic interface 读 FIFO 满标志。 0: 不满; 1: 满。	0x0
[4]	RW	gen_pld_r_empty	Generic interface 读 FIFO 空标志。 0: 不空; 1: 空。	0x1
[3]	RW	gen_pld_w_full	Generic interface 写 FIFO 满标志。 0: 不满; 1: 满。	0x0
[2]	RW	gen_pld_w_empty	Generic interface 写 FIFO 空标志。 0: 不空; 1: 空。	0x0
[1]	RW	gen_cmd_full	Generic interface 命令 FIFO 满标志。 0: 不满; 1: 满。	0x0
[0]	RW	gen_cmd_empty	Generic interface 命令 FIFO 空标志。 0: 不空; 1: 空。	0x0

LP_WR_TO_CNT

LP_WR_TO_CNT 为 LP 写回复超时配置寄存器。

Offset Address: 0x0088 Total Reset Value: 0x0000_0000

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



Bits	Access	Name	Description	Reset
[15:0]	RW	lp_wr_to_cnt	LP 发送写命令后等待时间，单位为 hclk。	0x0000

BTA_TO_CNT

BTA_TO_CNT 为 BTA 回复超时配置寄存器。

Offset Address: 0x008C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	bta_to_cnt	Bus Turn-Around 后 host 等待时间，单位为 hclk。	0x0000

LPCLK_CTRL

LPCLK_CTRL 为 clk lane 控制寄存器。

Offset Address: 0x0094 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	auto_clk lane_ctrl	clk lane 空闲时间时钟自动关断使能。 0: 不使能; 1: 使能。	0x0
[0]	RW	phy_txrequestclkhs	hclk 请求使能。 0: 不使能; 1: 使能。	0x0

PHY_TMR_LPCLK_CFG

PHY_TMR_LPCLK_CFG 为 clk lane HS 和 LP 切换时间配置寄存器。

Offset Address: 0x0098 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:16]	RW	phy_clkhs2lp_time	clk lane hs 转换到 lp 模式需要的时间，单位 hclk。	0x000



Bits	Access	Name	Description	Reset
[15:10]	-	reserved	保留。	0x00
[9:0]	RW	phy_clklp2hs_time	clklane lp 转换到 hs 模式需要的时间，单位 hscclk。	0x000

PHY_TMR_CFG

PHY_TMR_CFG 为 datalane HS 和 LP 切换时间配置寄存器。

Offset Address: 0x009C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:16]	RW	phy_hs2lp_time	datalane hs 转换到 lp 模式需要的时间，单位 hscclk。	0x000
[15:10]	-	reserved	保留。	0x00
[9:0]	RW	phy_lp2hs_time	datalane lp 转换到 hs 模式需要的时间，单位 hscclk。	0x000

PHY_RSTZ

PHY_RSTZ 为 PHY PLL 和复位配置寄存器。

Offset Address: 0x00A0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RW	phy_forcepll	PHY PLL 处于有效状态使能。 0: 不使能; 1: 使能。	0x0
[2]	RW	phy_enableclk	PHY clklane 使能。 0: 不使能; 1: 使能。	0x0
[1]	RW	phy_rstz	PHY 数字电路复位使能。 0: 复位; 1: 撤销复位。	0x0
[0]	RW	phy_shutdownz	PHY 整体 powerdown 使能。	0x0



Bits	Access	Name	Description	Reset
			0: power down; 1: power up。	

PHY_IF_CFG

PHY_IF_CFG 为 PHY 接口配置寄存器。

Offset Address: 0x00A4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:8]	RW	phy_stop_wait_time	进入 stopstate 到发起 HS 传输的最小等待时间。	0x00
[7:2]	-	reserved	保留。	0x00
[1:0]	RW	n_lanes	datalane 个数配置。 00: 1lane(0); 01: 2lanes(0,1); 10: 3lanes(0,1,2); 11: 4lanes(0,1,2,3)。	0x0

PHY_ULPS_CTRL

PHY_ULPS_CTRL 为 PHY 的 ULPS 状态配置寄存器。

Offset Address: 0x00A8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000
[3]	RW	phy_txexitulpslan	datalanes 退出 ULPS 模式使能。 0: 不使能; 1: 使能。	0x0
[2]	RW	phy_txrequlpslan	datalanes 进入 ULPS 模式请求使能。 0: 不使能; 1: 使能。	0x0
[1]	RW	phy_txexitulpsclk	clk lane 退出 ULPS 模式使能。 0: 不使能;	0x0



Bits	Access	Name	Description	Reset
			1: 使能。	
[0]	RW	phy_txrequpsclk	cklane 进入 ULPS 模式请求使能。 0: 不使能; 1: 使能。	0x0

PHY_STATUS

PHY_STATUS 为 PHY 状态寄存器。

Offset Address: 0x00B0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RO	phy_ulpsactivenot3lane	PHY lane3 ulps 状态指示信号。 0: 处于 ulps 状态; 1: 不处于 ulps 状态。	0x0
[11]	RO	phy_stopstate3lane	PHY lane3 stopstate 状态指示信号。 0: 不处于 stopstate 状态; 1: 处于 stopstate 状态。	0x0
[10]	RO	phy_ulpsactivenot2lane	PHY lane2 ulps 状态指示信号。 0: 处于 ulps 状态; 1: 不处于 ulps 状态。	0x0
[9]	RO	phy_stopstate2lane	PHY lane2 stopstate 状态指示信号。 0: 不处于 stopstate 状态; 1: 处于 stopstate 状态。	0x0
[8]	RO	phy_ulpsactivenot1lane	PHY lane1 ulps 状态指示信号。 0: 处于 ulps 状态; 1: 不处于 ulps 状态。	0x0
[7]	RO	phy_stopstate1lane	PHY lane1 stopstate 状态指示信号。 0: 不处于 stopstate 状态; 1: 处于 stopstate 状态。	0x0
[6]	RO	phy_rxulpsesc0lane	PHY lane0 rxulps 状态指示信号。 0: 处于 ulps 状态; 1: 不处于 ulps 状态。	0x0
[5]	RO	phy_ulpsactivenot0lane	PHY lane0 ulps 状态指示信号。	0x0



Bits	Access	Name	Description	Reset
		ne	0: 处于 ulps 状态; 1: 不处于 ulps 状态。	
[4]	RO	phy_stopstate0lane	PHY lane0 stopstate 状态指示信号。 0: 不处于 stopstate 状态; 1: 处于 stopstate 状态。	0x0
[3]	RO	phy_ulpsactivenotclk	PHY clklane ulps 状态指示信号。 0: 处于 ulps 状态; 1: 不处于 ulps 状态。	0x0
[2]	RO	phy_stopstateclklane	PHY clklane stopstate 状态指示信号。 0: 不处于 stopstate 状态; 1: 处于 stopstate 状态。	0x0
[1]	RO	phy_direction	PHY direction 信号状态。 0: 发送状态; 1: 接收状态。	0x0
[0]	RO	phy_lock	PHY lock 信号状态。 0: PHY PLL lock 无效; 1: PHY PLL lock 有效。	0x0

PHY_TST_CTRL0

PHY_TST_CTRL0 为 PHY 配置接口 0 寄存器。

Offset Address: 0x00B4 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	phy_testclk	phy 寄存器配置接口时钟控制位。 0: 低电平; 1: 高电平。	0x0
[0]	RW	phy_testclr	配置清除。 0: 不清除; 1: 清除。	0x1



PHY_TST_CTRL1

PHY_TST_CTRL1 为 PHY 配置接口 1 寄存器。

Offset Address: 0x00B8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x0000
[16]	RW	phy_testen	testen 信号控制。 0: 在 testclk 的上升沿采集数据; 1: 在 testclk 的下降沿采集地址。	0x0
[15:8]	RO	phy_testdout	PHY 配置接口输出数据。	0x00
[7:0]	RW	phy_testdin	PHY 配置接口输入数据。	0x00

INT_ST0

INT_ST0 为中断状态 0 寄存器。

Offset Address: 0x00BC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:21]	-	reserved	保留。	0x000
[20]	RC	dphy_errors_4	Lane0 LP1 contention 错误中断，读清零。 0: 无中断; 1: 有中断。	0x0
[19]	RC	dphy_errors_3	Lane0 LP0 contention 错误，读清零。 0: 无中断; 1: 有中断。	0x0
[18]	RC	dphy_errors_2	错误状态序列检测中断，读清零。 0: 无中断; 1: 有中断。	0x0
[17]	RC	dphy_errors_1	LP 数据传输同步中断，如果数据 bit 个数不是 8 的倍数，报中断，读清零。 0: 无中断; 1: 有中断。	0x0
[16]	RC	dphy_errors_0	Escape Entry 错误中断，如果接收到未定义命令时报中断，读清零。 0: 无中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	
[15]	RC	ack_with_err_15	设备错误报告(Error Reporting)中的 protocol violation 中断, 当预期的 EoTp 或者读命令后的 BTA 序列没有接收到时, 设备报告错误, 读清零。 0: 无中断; 1: 有中断。	0x0
[14]	RC	ack_with_err_14	设备错误报告(Error Reporting)中的 Reserved 位, 读清零。	0x0
[13]	RC	ack_with_err_13	设备错误报告(Error Reporting)中的 invalid transmission length 位, 当设备接收到的数据长度和预期不符时, 报错误, 读清零。 0: 无中断; 1: 有中断。	0x0
[12]	RC	ack_with_err_12	设备错误报告(Error Reporting)中的 DSI VC ID Invalid 位, 当设备接收到不能识别的 ID 号时报错误, 读清零。 0: 无中断; 1: 有中断。	0x0
[11]	RC	ack_with_err_11	设备错误报告(Error Reporting)中的 DSI Data Type Not Recognized 位, 当设备接收到不能识别的数据类型时报错误, 读清零。 0: 无中断; 1: 有中断。	0x0
[10]	RC	ack_with_err_10	设备错误报告(Error Reporting)中的 Checksum Error 位, 当设备接收到的数据有 CRC 错误时报错误, 读清零。 0: 无中断; 1: 有中断。	0x0
[9]	RC	ack_with_err_9	设备错误报告(Error Reporting)中的 ECC Error, multi-bit 位, 当设备接收到的数据有多位 ECC 错误时报错误, 读清零。 0: 无中断; 1: 有中断。	0x0
[8]	RC	ack_with_err_8	设备错误报告(Error Reporting)中的 ECC Error, single-bit 位, 当设备接收到的数据	0x0



Bits	Access	Name	Description	Reset
			有 1 位 ECC 错误时报错误，读清零。 0: 无中断; 1: 有中断。	
[7]	RC	ack_with_err_7	设备错误报告(Error Reporting)中的 Contention Detected 位，当设备有 LP contention 时报错误，读清零。 0: 无中断; 1: 有中断。	0x0
[6]	RC	ack_with_err_6	设备错误报告(Error Reporting)中的 False Control Error，当设备接收到错误状态序列时报错误，读清零。 0: 无中断; 1: 有中断。	0x0
[5]	RC	ack_with_err_5	设备错误报告(Error Reporting)中的 Peripheral Timeout Error 位，设备发生超时时报错误，读清零。 0: 无中断; 1: 有中断。	0x0
[4]	RC	ack_with_err_4	设备错误报告(Error Reporting)中的 Low-Power Transmit Sync Error 位，LP 传输时如果设备接收到的 bit 数不是 8 的倍数时报错误，读清零。 0: 无中断; 1: 有中断。	0x0
[3]	RC	ack_with_err_3	设备错误报告(Error Reporting)中的 Escape Mode Entry Command Error 位，设备接收到错误的 ESC 命令时报错误，读清零; 0: 无中断; 1: 有中断。	0x0
[2]	RC	ack_with_err_2	设备错误报告(Error Reporting)中的 EoT Sync Error 位，设备接收到错误的 EoT 序列时报错误，读清零。 0: 无中断; 1: 有中断。	0x0
[1]	RC	ack_with_err_1	设备错误报告(Error Reporting)中的 SoT Sync Error 位，设备接收到错误的 SoT 序列时报错误，读清零。 0: 无中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	
[0]	RC	ack_with_err_0	设备错误报告(Error Reporting)中的 SoT Error 位, 设备接收到错误的 EoT 序列但仍能接收到正确的同步信息时报错误, 读清零。 0: 无中断; 1: 有中断。	0x0

INT_ST1

INT_ST1 为中断状态 1 寄存器。

Offset Address: 0x00C0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:25]	-	reserved	保留。	0x00
[24]	RC	hss_abnormal	hss 包不在包起始阶段发送中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[23]	RC	rxtrigger_3	01100010 trigger 中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[22]	RC	rxtrigger_2	01011101 trigger 中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[21]	RC	rxtrigger_1	00100001 trigger 中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[20]	RC	rxtrigger_0	10100000 trigger 中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[19:17]	-	reserved	保留。	0x0
[16]	RC	vsstart	帧脉冲中断, 读清零。 0: 无中断; 1: 有中断。	0x0



Bits	Access	Name	Description	Reset
[15:13]	-	reserved	保留。	0x0
[12]	RC	gen_pld_recev_err	接收 fifo 满写错误中断，读清零。 0: 无中断; 1: 有中断。	0x0
[11]	RC	gen_pld_rd_err	接收 fifo 空读错误中断，读清零。 0: 无中断; 1: 有中断。	0x0
[10]	RC	gen_pld_send_err	payload fifo 空读错误中断，读清零。 0: 无中断; 1: 有中断。	0x0
[9]	RC	gen_pld_wr_err	payload fifo 满写错误中断，读清零。 0: 无中断; 1: 有中断。	0x0
[8]	RC	gen_cmd_wr_err	命令 fifo 满写错误中断，读清零。 0: 无中断; 1: 有中断。	0x0
[7]	RC	dpi_pld_wr_err	video mode fifo 满写错误中断，读清零。 0: 无中断; 1: 有中断。	0x0
[6]	RC	eopt_err	EoTp 接收错误中断，读清零。 0: 无中断; 1: 有中断。	0x0
[5]	RC	pkt_size_err	包大小接收错误中断，读清零。 0: 无中断; 1: 有中断。	0x0
[4]	RC	crc_err	接收数据 CRC 错误中断，读清零。 0: 无中断; 1: 有中断。	0x0
[3]	RC	ecc_multi_err	接收数据 ECC 多位错误中断，读清零。 0: 无中断; 1: 有中断。	0x0
[2]	RC	ecc_single_err	接收数据 ECC 单 bit 错误中断，读清零。 0: 无中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	
[1]	RC	to_lp_rx	LP 接收超时错误中断，读清零。 0: 无中断; 1: 有中断。	0x0
[0]	RC	to_hs_tx	HS 发送超时错误中断，读清零。 0: 无中断; 1: 有中断。	0x0

INT_MSK0

INT_MSK0 为中断 0 屏蔽寄存器。

Offset Address: 0x00C4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:21]	-	reserved	保留。	0x000
[20]	RW	mask_dphy_errors_4	dphy_errors_4 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[19]	RW	mask_dphy_errors_3	dphy_errors_3 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[18]	RW	mask_dphy_errors_2	dphy_errors_2 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[17]	RW	mask_dphy_errors_1	dphy_errors_1 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[16]	RW	mask_dphy_errors_0	dphy_errors_0 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[15]	RW	mask_ack_with_err_15	ack_with_err_15 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[14]	RW	mask_ack_with_err_	ack_with_err_14 中断屏蔽使能。	0x0



Bits	Access	Name	Description	Reset
		14	0: 屏蔽; 1: 使能。	
[13]	RW	mask_ack_with_err_13	ack_with_err_13 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[12]	RW	mask_ack_with_err_12	ack_with_err_12 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[11]	RW	mask_ack_with_err_11	ack_with_err_11 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[10]	RW	mask_ack_with_err_10	ack_with_err_10 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[9]	RW	mask_ack_with_err_9	ack_with_err_9 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[8]	RW	mask_ack_with_err_8	ack_with_err_8 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[7]	RW	mask_ack_with_err_7	ack_with_err_7 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[6]	RW	mask_ack_with_err_6	ack_with_err_6 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[5]	RW	mask_ack_with_err_5	ack_with_err_5 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[4]	RW	mask_ack_with_err_4	ack_with_err_4 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[3]	RW	mask_ack_with_err_3	ack_with_err_3 中断屏蔽使能。	0x0



Bits	Access	Name	Description	Reset
			0: 屏蔽; 1: 使能。	
[2]	RW	mask_ack_with_err_2	ack_with_err_2 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[1]	RW	mask_ack_with_err_1	ack_with_err_1 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[0]	RW	mask_ack_with_err_0	ack_with_err_0 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0

INT_MSK1

INT_MSK1 为中断 1 屏蔽寄存器。

Offset Address: 0x00C8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:25]	-	reserved	保留。	0x00
[24]	RW	mask_hss_abnormal	hss_abnormal 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[23]	RW	mask_trigger_3	01100010 trigger 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[22]	RW	mask_trigger_2	01011101 trigger 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[21]	RW	mask_trigger_1	00100001 trigger 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[20]	RW	mask_trigger_0	10100000 trigger 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0



Bits	Access	Name	Description	Reset
[19:17]	-	reserved	保留。	0x0
[16]	RW	mask_vsstart	vsstart 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[15:13]	-	reserved	保留。	0x0
[12]	RW	mask_gen_pld_recev_err	gen_pld_recev_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[11]	RW	mask_gen_pld_rd_err	gen_pld_rd_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[10]	RW	mask_gen_pld_send_err	gen_pld_send_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[9]	RW	mask_gen_pld_wr_err	gen_pld_wr_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[8]	RW	mask_gen_cmd_wr_err	gen_cmd_wr_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[7]	RW	mask_dpi_pld_wr_err	dpi_pld_wr_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[6]	RW	mask_eopt_err	eopt_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[5]	RW	mask_pkt_size_err	pkt_size_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[4]	RW	mask_crc_err	crc_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[3]	RW	mask_ecc_multi_err	ecc_multi_err 中断屏蔽使能。	0x0



Bits	Access	Name	Description	Reset
			0: 屏蔽; 1: 使能。	
[2]	RW	mask_ecc_single_err	ecc_single_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[1]	RW	mask_to_lp_rx	to_lp_rx 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[0]	RW	mask_to_hs_tx	to_hs_tx 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0

OPERATION_MODE

OPERATION_MODE 为工作模式控制寄存器。

Offset Address: 0x0208 Total Reset Value: 0x0000_0002

Bits	Access	Name	Description	Reset
[31]	RW	input_en	前级模块输入数据有效使能: 0: 不使能; 1: 使能。	0x0
[30:21]	-	reserved	保留。	0x000
[20]	RW	hss_abnormal_rst	hss 包不在包起始阶段发送,自动纠正使能: 0: 不使能; 1: 使能。	0x0
[19]	-	reserved	保留。	0x0
[18]	RW	buf_clr_en	buf 每帧清除使能。 0: 不使能; 1: 使能。	0x0
[17]	RO	reserved	保留。	0x0
[16]	RW	read_empty_vsync_en	fifo 每帧读空使能。 0: 不使能; 1: 使能。	0x0



Bits	Access	Name	Description	Reset
[15:6]	-	reserved	保留。	0x000
[5]	RW	init_skew_en	initial skew 使能。 0: 不使能; 1: 使能。	0x0
[4]	RO	reserved	保留。	0x0
[3:2]	RO	reserved	保留。	0x0
[1]	RW	mem_ck_en	memory 门控使能信号。 0: memory 门控一直打开; 1: memory 门控在对 memory 读写的时候打开, 空闲时关闭。	0x1
[0]	RO	reserved	保留。	0x0

V_H_SEND

V_H_SEND 为发送的宽高寄存器。

Offset Address: 0x0224 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	v_send	发送的高度。	0x0000
[15:0]	RO	h_send	发送的宽度。	0x0000

DATATYPE0

DATATYPE0 为 datatype 配置寄存器 0。

Offset Address: 0x0230 Total Reset Value: 0x0111_213D

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29:24]	RW	dt_vss	帧起始 datatype。	0x01
[23:22]	-	reserved	保留。	0x0
[21:16]	RW	dt_vse	帧结束 datatype。	0x11
[15:14]	-	reserved	保留。	0x0
[13:8]	RW	dt_hss	行起始 datatype。	0x21



[7:6]	-	reserved	保留。	0x0
[5:0]	RO	reserved	保留。	0x3D

CSI_CTRL

CSI_CTRL 为 CSI 时序配置寄存器。

Offset Address: 0x0238 Total Reset Value: 0x0001_0100

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x0000
[16]	RO	reserved	保留。	0x1
[15:9]	-	reserved	保留。	0x00
[8]	RO	reserved	保留。	0x1
[7:1]	-	reserved	保留。	0x00
[0]	RW	csi_en	CSI 时序使能。 0: 不使能; 1: 使能。	0x0

9.4.7 MIPI Tx PHY 寄存器概览

MIPI Tx PHY 寄存器概览如表 9-14 所示。

表9-14 MIPI Tx PHY 寄存器概览（基址是 0x00）

偏移地址	名称	描述	页码
0x0F	LP_RX_EN	LP RX 使能配置寄存器	9-242
0x10	TLPX_CLK	clocklane Tlpx 时间配置寄存器	9-242
0x11	TCK_PREPARE	clocklane Tck-prepare 时间配置寄存器	9-243
0x12	TCLK-ZERO	clocklane Tclk-zero 时间配置寄存器	9-243
0x13	TCLK-TRAIL	clocklane Tclk-trail 时间配置寄存器	9-243
0x14	TWAKEUP-CLK	clocklane Twakeup 时间配置寄存器	9-243
0x17	PRE_DELAY_HSCLK	HS 时钟有效延时配置寄存器	9-244
0x18	POST_DELAY_HSCLK	HS 时钟无效延时配置寄存器	9-244



偏移地址	名称	描述	页码
0x20	TLPX_D0	data lane0 Tlpx 时间配置寄存器	9-244
0x21	THS_PREPARE-D0	data lane0 Ths-prepare 时间配置寄存器	9-244
0x22	THS-ZERO-D0	data lane0 Ths-zero 时间配置寄存器	9-245
0x23	THS-TRAIL-D0	data lane0 Ths-trail 时间配置寄存器	9-245
0x24	TTA-GO-D0	data lane0 Tta-go 时间配置寄存器	9-245
0x25	TTA-GET-D0	data lane0 Tta-get 时间配置寄存器	9-245
0x26	TWAKEUP-D0	data lane0 Twakeup 时间配置寄存器	9-246
0x28	PRE_DELAY_D0HS	data lane0 HS 有效延时配置寄存器	9-246
0x29	POST_DELAY_D0HS	data lane0 HS 无效延时配置寄存器	9-246
0x30	TLPX_D1	data lane1 Tlpx 时间配置寄存器	9-246
0x31	THS_PREPARE-D1	data lane1 Ths-prepare 时间配置寄存器	9-246
0x32	THS-ZERO-D1	data lane1 Ths-zero 时间配置寄存器	9-247
0x33	THS-TRAIL-D1	data lane1 Ths-trail 时间配置寄存器	9-247
0x34	TTA-GO-D1	data lane1 Tta-go 时间配置寄存器	9-247
0x35	TTA-GET-D1	data lane1 Tta-get 时间配置寄存器	9-247
0x36	TWAKEUP-D1	data lane1 Twakeup 时间配置寄存器	9-248
0x38	PRE_DELAY_D1HS	data lane1 HS 有效延时配置寄存器	9-248
0x39	POST_DELAY_D1HS	data lane1 HS 无效延时配置寄存器	9-248
0x40	TLPX_D2	data lane2 Tlpx 时间配置寄存器	9-248
0x41	THS_PREPARE-D2	data lane2 Ths-prepare 时间配置寄存器	9-249
0x42	THS-ZERO-D2	data lane2 Ths-zero 时间配置寄存器	9-249
0x43	THS-TRAIL-D2	data lane2 Ths-trail 时间配置寄存器	9-249
0x44	TTA-GO-D2	data lane2 Tta-go 时间配置寄存器	9-249
0x45	TTA-GET-D2	data lane2 Tta-get 时间配置寄存器	9-250
0x46	TWAKEUP-D2	data lane2 Twakeup 时间配置寄存器	9-250
0x48	PRE_DELAY_D2HS	data lane2 HS 有效延时配置寄存器	9-250
0x49	POST_DELAY_D2HS	data lane2 HS 无效延时配置寄存器	9-250
0x50	TLPX_D3	data lane3 Tlpx 时间配置寄存器	9-251



偏移地址	名称	描述	页码
0x51	THS_PREPARE-D3	data lane3 Ths-prepare 时间配置寄存器	9-251
0x52	THS_ZERO-D3	data lane3 Ths-zero 时间配置寄存器	9-251
0x53	THS_TRAIL-D3	data lane3 Ths-trail 时间配置寄存器	9-251
0x54	TTA_GO-D3	data lane3 Tta-go 时间配置寄存器	9-251
0x55	TTA_GET-D3	data lane3 Tta-get 时间配置寄存器	9-252
0x56	TWAKEUP-D3	data lane3 Twakeup 时间配置寄存器	9-252
0x58	PRE_DELAY_D3HS	data lane3 HS 有效延时配置寄存器	9-252
0x59	POST_DELAY_D3HS	data lane3 HS 无效延时配置寄存器	9-252
0x60	PLL_SET0	PLL 参数 0 配置寄存器	9-253
0x64	PLL_SET1	PLL 参数 1 配置寄存器	9-253
0x65	PLL_SET2	PLL 参数 2 配置寄存器	9-254
0x66	PLL_SET4	PLL 参数 4 配置寄存器	9-254
0x67	PLL_SET5	PLL 参数 5 配置寄存器	9-254

9.4.8 MIPI Tx PHY 寄存器描述

 说明
寄存器描述中的 hclk period=1/(Data Rate/8)。

LP_RX_EN

LP_RX_EN 为 LP RX 使能配置寄存器。

Offset Address: 0x0F Total Reset Value: 0x01

Bits	Access	Name	Description	Reset
[7:1]	-	reserved	保留。	0x0
[0]	RW	lane0_rx_en	lane0 RX 使能： 0：不使能； 1：使能。	0x1

TLPX_CLK

TLPX_CLK 为 clocklane Tlpx 时间配置寄存器。



Offset Address: 0x10 Total Reset Value: 0x01

Bits	Access	Name	Description	Reset
[7:0]	RW	Tlpx-clk	clocklane Tlpx 时间，单位 hclk 周期。	0x01

TCK_PREPARE

TCK_PREPARE 为 clocklane Tck-prepare 时间配置寄存器。

Offset Address: 0x11 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	Tck-prepare	clocklane Tck-prepare 时间，单位 hclk 周期。	0x00

TCLK-ZERO

TCLK-ZERO 为 clocklane Tclk-zero 时间配置寄存器。

Offset Address: 0x12 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	Tclk-zero	clocklane Tclk-zero 时间，单位 hclk 周期。	0x00

TCLK-TRAIL

TCLK-TRAIL 为 clocklane Tclk-trail 时间配置寄存器。

Offset Address: 0x13 Total Reset Value: 0x0A

Bits	Access	Name	Description	Reset
[7:0]	RW	Tclk-trail	clocklane Tclk-trail 时间，单位 hclk 周期。	0x0A

TWAKEUP-CLK

TWAKEUP-CLK 为 clocklane Twakeup 时间配置寄存器。

Offset Address: 0x14 Total Reset Value: 0x20



Bits	Access	Name	Description	Reset
[7:0]	RW	Twakeup-clk	clocklane Twakeup 时间高 8 位，{0x26[7:0],3'h0}，单位 cfg_clk 周期。	0x20

PRE_DELAY_HSCLK

PRE_DELAY_HSCLK 为 HS 时钟有效延时配置寄存器。

Offset Address: 0x17 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	pre_delay_hsclk	clock lane 从 txrequesthsclk 有效到进入 HS 模式的延时时间，单位 hsclk 周期。	0x00

POST_DELAY_HSCLK

POST_DELAY_HSCLK 为 HS 时钟无效延时配置寄存器。

Offset Address: 0x18 Total Reset Value: 0x20

Bits	Access	Name	Description	Reset
[7:0]	RW	post_delay_hsclk	clock lane 从 txrequesthsclk 无效到退出 HS 模式的延时时间，单位 hsclk 周期。	0x20

TLPX_D0

TLPX_D0 为 data lane0 Tlpx 时间配置寄存器。

Offset Address: 0x20 Total Reset Value: 0x01

Bits	Access	Name	Description	Reset
[7:0]	RW	Tlpx-d0	data lane0 Tlpx 时间，单位 hsclk 周期。	0x01

THS_PREPARE-D0

THS_PREPARE-D0 为 data lane0 Ths-prepare 时间配置寄存器。

Offset Address: 0x21 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	Ths-prepare-d0	data lane0 Tck-prepare 时间，单位 hsclk 周	0x00



			期。	

THS-ZERO-D0

THS-ZERO-D0 为 data lane0 Ths-zero 时间配置寄存器。

Offset Address: 0x22 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	Ths-zero-d0	data lane0 Ths-zero 时间，单位 hclk 周期。	0x00

THS-TRAIL-D0

THS-TRAIL-D0 为 data lane0 Ths-trail 时间配置寄存器。

Offset Address: 0x23 Total Reset Value: 0x08

Bits	Access	Name	Description	Reset
[7:0]	RW	Ths-trail-d0	data lane0 Ths-trail 时间，单位 hclk 周期。	0x08

TTA-GO-D0

TTA-GO-D0 为 data lane0 Tta-go 时间配置寄存器。

Offset Address: 0x24 Total Reset Value: 0x03

Bits	Access	Name	Description	Reset
[7:0]	RW	Tta-go-d0	data lane0 Tta-go 时间，单位 txclkesc 周期。	0x03

TTA-GET-D0

TTA-GET-D0 为 data lane0 Tta-get 时间配置寄存器。

Offset Address: 0x25 Total Reset Value: 0x04

Bits	Access	Name	Description	Reset
[7:0]	RW	Tta-get-d0	data lane0 Tta-get 时间，单位 txclkesc 周期。	0x04



TWAKEUP-D0

TWAKEUP-D0 为 data lane0 Twakeup 时间配置寄存器。

Offset Address: 0x26 Total Reset Value: 0x20

Bits	Access	Name	Description	Reset
[7:0]	RW	Twakeup-d0	data lane0 Twakeup 时间高 8 位， {0x38[7:0],3'h0}，单位 cfg_clk 周期。	0x20

PRE_DELAY_D0HS

PRE_DELAY_D0HS 为 data lane0 HS 有效延时配置寄存器。

Offset Address: 0x28 Total Reset Value: 0x10

Bits	Access	Name	Description	Reset
[7:0]	RW	pre_delay_d0hs	data lane0 从 txrequesths_lane0 有效到进入 HS 模式的延时时间，单位 hsclock 周期。	0x10

POST_DELAY_D0HS

POST_DELAY_D0HS 为 data lane0 HS 无效延时配置寄存器。

Offset Address: 0x29 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	post_delay_d0hs	data lane0 从 txrequesths_lane0 无效到退出 HS 模式的延时时间，单位 hsclock 周期。	0x00

TLPX_D1

TLPX_D1 为 data lane1 Tlpx 时间配置寄存器。

Offset Address: 0x30 Total Reset Value: 0x01

Bits	Access	Name	Description	Reset
[7:0]	RW	Tlpx-d1	data lane1 Tlpx 时间，单位 hsclock 周期。	0x01

THS_PREPARE-D1

THS_PREPARE-D1 为 data lane1 Ths-prepare 时间配置寄存器。



Offset Address: 0x31 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	Ths-prepare-d1	data lane1 Tck-prepare 时间，单位 hclk 周期。	0x00

THS-ZERO-D1

THS-ZERO-D1 为 data lane1 Ths-zero 时间配置寄存器。

Offset Address: 0x32 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	Ths-zero-d1	data lane1 Ths-zero 时间，单位 hclk 周期。	0x00

THS-TRAIL-D1

THS-TRAIL-D1 为 data lane1 Ths-trail 时间配置寄存器。

Offset Address: 0x33 Total Reset Value: 0x08

Bits	Access	Name	Description	Reset
[7:0]	RW	Ths-trail-d1	data lane1 Ths-trail 时间，单位 hclk 周期。	0x08

TTA-GO-D1

TTA-GO-D1 为 data lane1 Tta-go 时间配置寄存器。

Offset Address: 0x34 Total Reset Value: 0x03

Bits	Access	Name	Description	Reset
[7:0]	RW	Tta-go-d1	data lane1 Tta-go 时间，单位 txclkesc 周期。	0x03

TTA-GET-D1

TTA-GET-D1 为 data lane1 Tta-get 时间配置寄存器。

Offset Address: 0x35 Total Reset Value: 0x04



Bits	Access	Name	Description	Reset
[7:0]	RW	Tta-get-d1	data lane1 Tta-get 时间，单位 txclkesc 周期。	0x04

TWAKEUP-D1

TWAKEUP-D1 为 data lane1 Twakeup 时间配置寄存器。

Offset Address: 0x36 Total Reset Value: 0x20

Bits	Access	Name	Description	Reset
[7:0]	RW	Twakeup-d1	data lane1 Twakeup 时间高 8 位，{0x48[7:0],3'h0}，单位 cfg_clk 周期。	0x20

PRE_DELAY_D1HS

PRE_DELAY_D1HS 为 data lane1 HS 有效延时配置寄存器。

Offset Address: 0x38 Total Reset Value: 0x10

Bits	Access	Name	Description	Reset
[7:0]	RW	pre_delay_d1hs	data lane1 从 txrequesths_lane1 有效到进入 HS 模式的延时时间，单位 hsclock 周期。	0x10

POST_DELAY_D1HS

POST_DELAY_D1HS 为 data lane1 HS 无效延时配置寄存器。

Offset Address: 0x39 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	post_delay_d1hs	data lane1 从 txrequesths_lane1 无效到退出 HS 模式的延时时间，单位 hsclock 周期。	0x00

TLPX_D2

TLPX_D2 为 data lane2 Tlpx 时间配置寄存器。

Offset Address: 0x40 Total Reset Value: 0x01



Bits	Access	Name	Description	Reset
[7:0]	RW	Tlpx-d2	data lane2 Tlpx 时间，单位 hclk 周期。	0x01

THS_PREPARE-D2

THS_PREPARE-D2 为 data lane2 Ths-prepare 时间配置寄存器。

Offset Address: 0x41 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	Ths-prepare-d2	data lane2 Tck-prepare 时间，单位 hclk 周期。	0x00

THS_ZERO-D2

THS_ZERO-D2 为 data lane2 Ths-zero 时间配置寄存器。

Offset Address: 0x42 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	Ths-zero-d2	data lane2 Ths-zero 时间，单位 hclk 周期。	0x00

THS_TRAIL-D2

THS_TRAIL-D2 为 data lane2 Ths-trail 时间配置寄存器。

Offset Address: 0x43 Total Reset Value: 0x08

Bits	Access	Name	Description	Reset
[7:0]	RW	Ths-trail-d2	data lane2 Ths-trail 时间，单位 hclk 周期。	0x08

TTA_GO-D2

TTA_GO-D2 为 data lane2 Tta-go 时间配置寄存器。

Offset Address: 0x44 Total Reset Value: 0x03

Bits	Access	Name	Description	Reset
[7:0]	RW	Tta-go-d2	data lane2 Tta-go 时间，单位 txclkesc 周期。	0x03



TTA-GET-D2

TTA-GET-D2 为 data lane2 Tta-get 时间配置寄存器。

Offset Address: 0x45 Total Reset Value: 0x04

Bits	Access	Name	Description	Reset
[7:0]	RW	Tta-get-d2	data lane2 Tta-get 时间，单位 txclkesc 周期。	0x04

TWAKEUP-D2

TWAKEUP-D2 为 data lane2 Twakeup 时间配置寄存器。

Offset Address: 0x46 Total Reset Value: 0x20

Bits	Access	Name	Description	Reset
[7:0]	RW	Twakeup-d2	data lane2 Twakeup 时间高 8 位，{0x58[7:0],3'h0}，单位 cfg_clk 周期。	0x20

PRE_DELAY_D2HS

PRE_DELAY_D2HS 为 data lane2 HS 有效延时配置寄存器。

Offset Address: 0x48 Total Reset Value: 0x10

Bits	Access	Name	Description	Reset
[7:0]	RW	pre_delay_d2hs	data lane2 从 txrequesths_lane2 有效到进入 HS 模式的延时时间，单位 hsclk 周期。	0x10

POST_DELAY_D2HS

POST_DELAY_D2HS 为 data lane2 HS 无效延时配置寄存器。

Offset Address: 0x49 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	post_delay_d2hs	data lane2 从 txrequesths_lane2 无效到退出 HS 模式的延时时间，单位 hsclk 周期。	0x00



TLPX_D3

TLPX_D3 为 data lane3 Tlpx 时间配置寄存器。

Offset Address: 0x50 Total Reset Value: 0x01

Bits	Access	Name	Description	Reset
[7:0]	RW	Tlpx-d3	data lane3 Tlpx 时间，单位 hclk 周期。	0x01

THS_PREPARE-D3

THS_PREPARE-D3 为 data lane3 Ths-prepare 时间配置寄存器。

Offset Address: 0x51 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	Ths-prepare-d3	data lane3 Tck-prepare 时间，单位 hclk 周期。	0x00

THS_ZERO-D3

THS_ZERO-D3 为 data lane3 Ths-zero 时间配置寄存器。

Offset Address: 0x52 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	Ths-zero-d3	data lane3 Ths-zero 时间，单位 hclk 周期。	0x00

THS-TRAIL-D3

THS-TRAIL-D3 为 data lane3 Ths-trail 时间配置寄存器。

Offset Address: 0x53 Total Reset Value: 0x08

Bits	Access	Name	Description	Reset
[7:0]	RW	Ths-trail-d3	data lane3 Ths-trail 时间，单位 hclk 周期。	0x08

TTA-GO-D3

TTA-GO-D3 为 data lane3 Tta-go 时间配置寄存器。

Offset Address: 0x54 Total Reset Value: 0x03



Bits	Access	Name	Description	Reset
[7:0]	RW	Tta-go-d3	data lane3 Tta-go 时间，单位 txclkesc 周期。	0x03

TTA-GET-D3

TTA-GET-D3 为 data lane3 Tta-get 时间配置寄存器。

Offset Address: 0x55 Total Reset Value: 0x04

Bits	Access	Name	Description	Reset
[7:0]	RW	Tta-get-d3	data lane3 Tta-get 时间，单位 txclkesc 周期。	0x04

TWAKEUP-D3

TWAKEUP-D3 为 data lane3 Twakeup 时间配置寄存器。

Offset Address: 0x56 Total Reset Value: 0x20

Bits	Access	Name	Description	Reset
[7:0]	RW	Twakeup-d3	data lane3 Twakeup 时间高 8 位，{0x68[7:0],3'h0}，单位 cfg_clk 周期。	0x20

PRE_DELAY_D3HS

PRE_DELAY_D3HS 为 data lane3 HS 有效延时配置寄存器。

Offset Address: 0x58 Total Reset Value: 0x10

Bits	Access	Name	Description	Reset
[7:0]	RW	pre_delay_d3hs	data lane3 从 txrequesths_lane3 有效到进入 HS 模式的延时时间，单位 hsclock 周期。	0x10

POST_DELAY_D3HS

POST_DELAY_D3HS 为 data lane3 HS 无效延时配置寄存器。

Offset Address: 0x59 Total Reset Value: 0x00



Bits	Access	Name	Description	Reset
[7:0]	RW	post_delay_d3hs	data lane3 从 txrequesths_lane3 无效到退出 HS 模式的延时时间，单位 hsclock 周期。	0x00

PLL_SET0

PLL_SET0 为 PLL 参数 0 配置寄存器。

Offset Address: 0x60 Total Reset Value: 0x0E

Bits	Access	Name	Description	Reset
[7:4]	-	reserved	保留。	0x0
[3:1]	RW	RG_HSTX_CKG_SEL	分频倍数： 0xx: 一分频，当 data rate=1000Mbit/s~750Mbit/s; 100: 两分频，当 data rate=750Mbit/s~375Mbit/s; 101: 四分频，当 data rate=375Mbit/s~188Mbit/s; 110: 八分频，当 data rate=188Mbit/s~94Mbit/s; 111: 十六分频，当 data rate=94Mbit/s~47Mbit/s。	0x7
[0]	-	reserved	保留。	0x0

PLL_SET1

PLL_SET1 为 PLL 参数 1 配置寄存器。

Offset Address: 0x64 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:4]	RW	RG_PLL_FBD_S	Feedback divided 配置，不能配置为 0。	0x0
[3]	-	reserved	保留。	0x0
[2]	RW	RG_PLL_EN_O	RG_PLL_EN_O 配置值。	0x0
[1]	RW	RG_PLL_BW	RG_PLL_BW 配置值。	0x0



PLL_SET2

PLL_SET2 为 PLL 参数 2 配置寄存器。

Offset Address: 0x65 Total Reset Value: 0x22

Bits	Access	Name	Description	Reset
[7]	-	reserved	保留。	0x0
[6:0]	RW	RG_PLL_FBD_P	RG_PLL_FBD_P 配置值。	0x22

PLL_SET4

PLL_SET4 为 PLL 参数 4 配置寄存器。

Offset Address: 0x66 Total Reset Value: 0x01

Bits	Access	Name	Description	Reset
[7:1]	-	reserved	保留。	0x00
[0]	RW	RG_PLL_FBD_DIV 1F	设置为 0。	0x1

PLL_SET5

PLL_SET5 为 PLL 参数 5 配置寄存器。

Offset Address: 0x67 Total Reset Value: 0x01

Bits	Access	Name	Description	Reset
[7]	RO	reserved	保留。	0x0
[6:0]	RW	RG_PLL_PRE_P	RG_PLL_PRE_P 配置值。	0x01



目 录

10 ISP	10-1
10.1 功能简介	10-1
10.2 总体概要	10-2
10.3 工作模式	10-4
10.4 ISP 中断系统	10-6
10.5 模块功能	10-7



插图目录

图 10-1 ISP 整体结构图.....	10-2
图 10-2 ISP_FE 结构图.....	10-3
图 10-3 ISP_BE 结构图.....	10-4
图 10-4 中断时序示意图.....	10-7
图 10-5 有效图像区域与水平垂直消隐关系图.....	10-8
图 10-6 AWB 有效像素颜色空间.....	10-11
图 10-7 Gamma 曲线.....	10-13
图 10-8 锐化、过冲说明.....	10-14



表格目录

表 10-1 外接 ISP 关键参数.....	10-5
表 10-2 位置可调模块关键参数.....	10-5
表 10-3 中断指示寄存器.....	10-6
表 10-4 中断时序	10-7
表 10-5 AE 区域统计信息	10-9
表 10-6 AE 直方图统计信息	10-10
表 10-7 AWB 区域统计方式 A 信息.....	10-12



10 ISP

10.1 功能简介

ISP 模块支持标准的 Sensor 图像数据处理，包括自动白平衡、自动曝光、Demosaic、坏点矫正及镜头阴影矫正等基本功能，也支持 WDR、DRC、降噪等高级处理功能。ISP 主要支持的图像处理功能如下：

- 支持黑电平校正
- 支持静态以及动态坏点校正，坏点簇矫正
- 支持 Bayer 降噪
- 支持固定噪声消除
- 支持 Demosaic 处理
- 支持紫边校正（CAC）
- 支持 Gamma 校正
- 支持动态范围压缩（DRC）
- 支持 Sensor 内部合成宽动态功能（Sensor Built-in WDR）
- 最大支持 2 合 1 宽动态功能(WDR)
- 支持自动白平衡
- 支持自动曝光
- 支持 3A 相关统计信息输出
- 支持镜头阴影校正
- 支持图像锐化
- 支持自动去雾处理
- 支持颜色三维查找表增强
- 支持局部对比度增强
- 支持色彩自适应（CA）
- 支持 3D 降噪
- 支持高光屏蔽(HLC)

处理能力以及其他功能：



- 最大支持 14 bit bayer 数据输入
- Built-in WDR 最大支持 16 bit bayer 数据输入
- 支持最大图像分辨率 2688x1944
- 支持最小图像分辨率 120x120(AF 关闭时)、256x120 (AF 打开时)
- 最小水平消隐区 64 像素
- 最小垂直消隐区 40 行
- 最大性能：2688x1536@30fps/2 路 1920x1080@30fps

10.2 总体概要

功能框图

ISP 的功能结构图如图 10-1、图 10-2、图 10-3 所示。此图与本文中提到的 ISP_FE 均代指 ISP pipeline 中 FPN（不包含）之前的部分，ISP_BE 均代指 ISP pipeline 中 FPN（包含）之后的部分。



说明

本文中 ISP 采用 U*.*、S*.*表示无符号数和有符号数。如：U8.8 表示数据类型为无符号数，整数部分 8bit，小数部分 8bit；同理，S8.8 表示数据类型为有符号数，整数部分 8bit（包括 1bit 符号位），小数部分 8bit。

图10-1 ISP 整体结构图

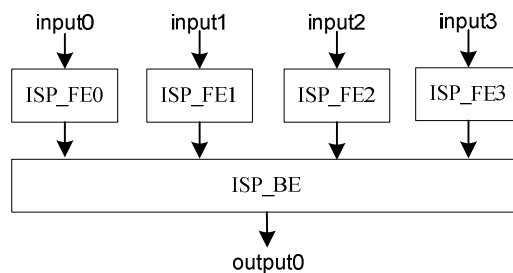




图10-2 ISP_FE 结构图

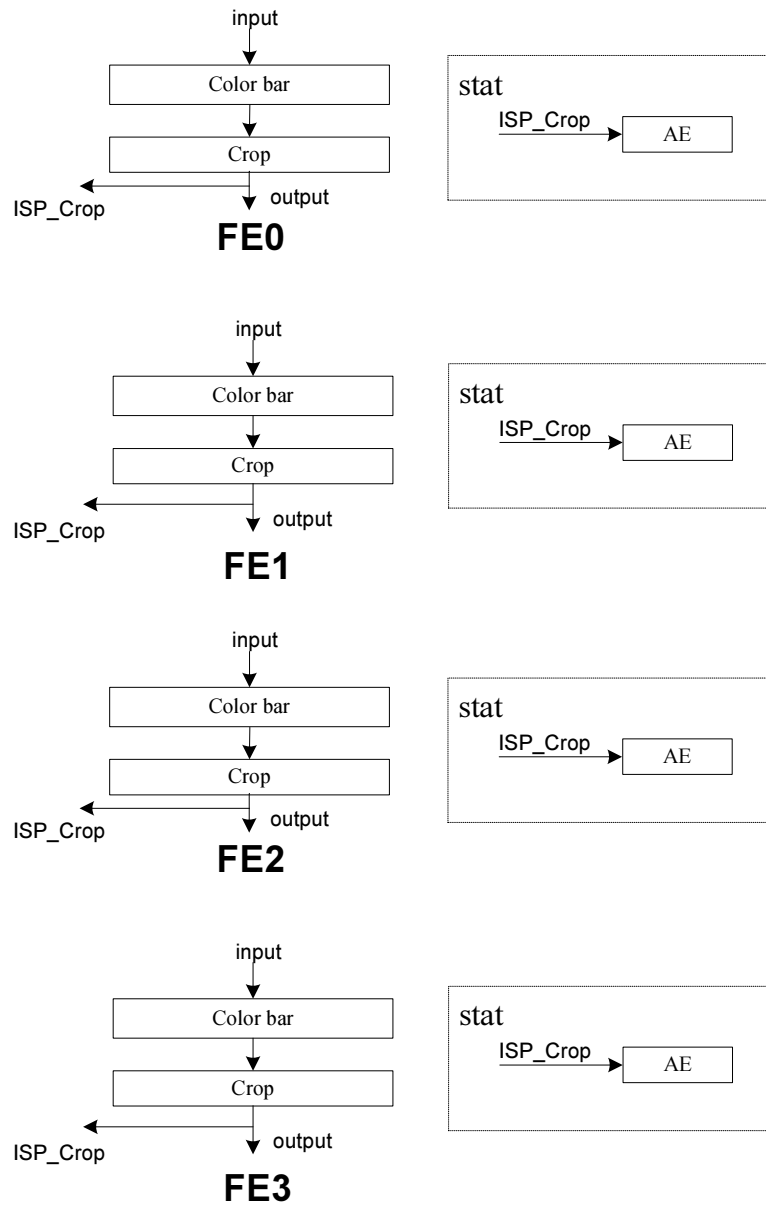
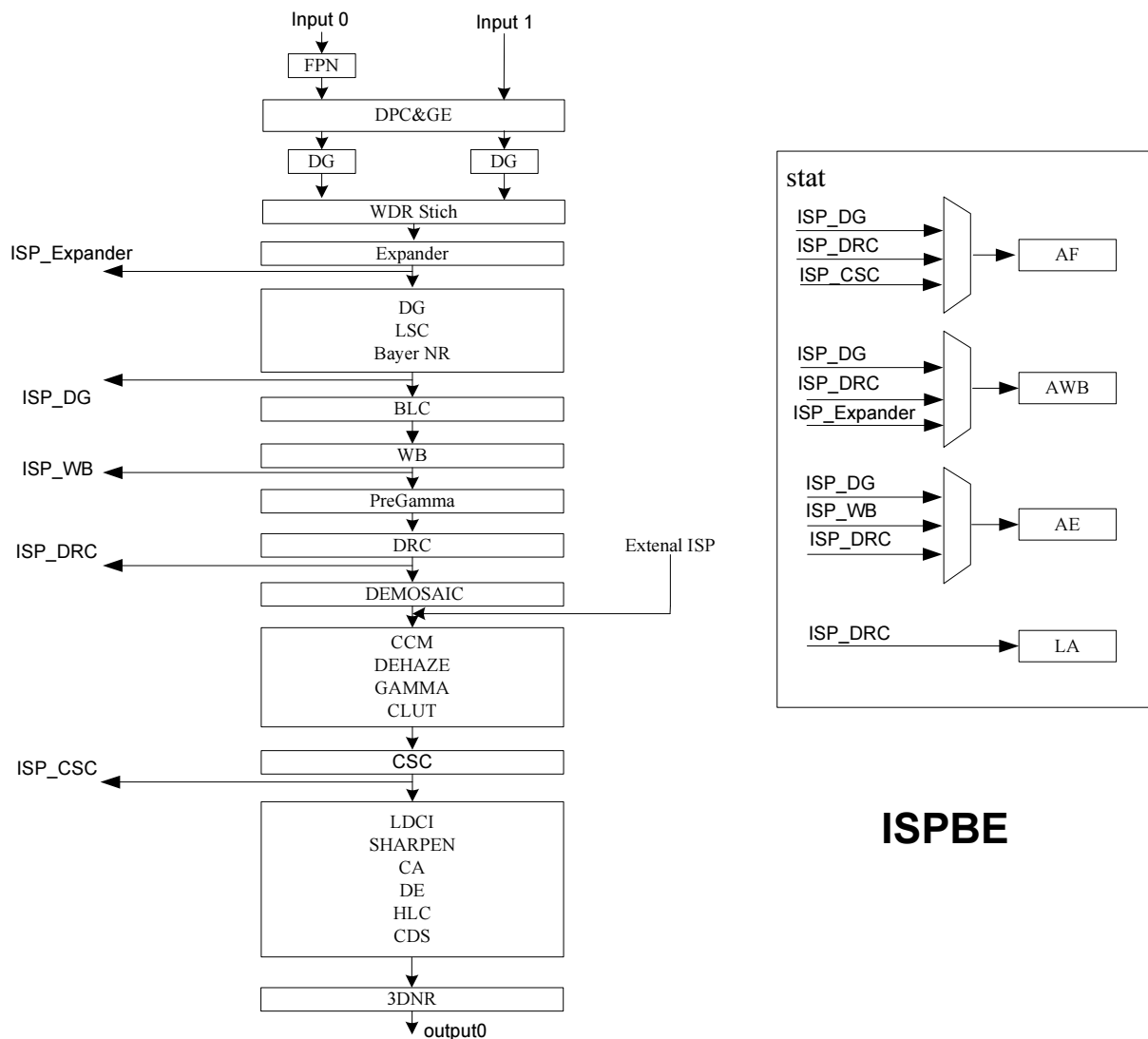




图10-3 ISP_BE 结构图



ISPBE

10.3 工作模式

- **支持最大 14bit bayer 数据输入**

当输入数据位宽小于 14bit，自动对输入图像数据进行高位对齐、低位补零处理。此模式下支持任意 RG_rG_bB 顺序。

- **支持 Sensor Bulit-in WDR**

Sensor Bulit-in WDR 模式下，支持最大 16bit bayer 数据输入。

- **支持 2 合 1 WDR**

- **支持亮度单分量模式**

支持丢弃 C (U、V) 分量，只输出 Y 分量图像数据。



- **ISP_BE 支持分时复用**

分时复用应用场景举例如下：

例 1：4 路 ISP_FE 输入，ISP_FE0 时钟频率 300M，ISP_FE1 时钟频率 300M。
ISP_FE2 时钟频率 300M，ISP_FE3 时钟频率 300M。ISP_BE 可以分时处理
ISP_FE0，ISP_FE1，ISP_FE2，ISP_FE3 4 路数据。



说明

ISP_FE/ISP_BE 频率可分别配置，请参考 3.2.6 CRG 小节的寄存器。

- **支持外接 ISP**

支持外接 ISP 模式，关键参数如下：

表10-1 外接 ISP 关键参数

参数名称	描述
ISP_BE_INPUT_MUX[16]	ISP 数据输入选择。 0 为 RAW 输入； 1 为 YUV 输入。

- **支持模块位置可调**

AE、AWB、AF 支持位置可调。

表10-2 位置可调模块关键参数

参数名称	参数说明
ISP_BE_MODULE_POS[5:4]	AF 位置调节寄存器。 00：AF 在 DG 后面； 01：AF 在 DRC 后面。 10：AF 在 CSC 后面。
ISP_BE_MODULE_POS[3:2]	AWB 位置调节寄存器。 00：AWB 在 DG 后面； 01：AWB 在 Expander 后面； 10：AWB 在 DRC 后面。
ISP_BE_MODULE_POS[1:0]	AE 位置调节寄存器。 00：AE 在 DG 后面； 01：AE 在 WB 后面； 10：AE 在 DRC 后面。



10.4 ISP 中断系统

功能描述

现 ISP 有 20 个硬件中断事件，详细见表 10-3 所示。

表10-3 中断指示寄存器

偏移地址	32bit	描述（写 1 清零，0：无中断 1：有中断）
0x3200F0+FE_N*0x2000 0	bit[4]	AE 统计完成中断。
	bit[3]	ISP FE 触发位置可配中断。可支持输入图像有效区以行为单位的任意位置触发。
	bit[2]	ISP FE 寄存器配置丢失中断。
	bit[1]	ISP FE 寄存器更新中断。
	bit[0]	ISP FE 帧起始中断。
0x00310	bit[28]	LA 统计完成中断。
	bit[26]	DEHAZE 统计完成中断。
	bit[25]	LDCI 统计完成中断。
	bit[22]	AF 统计完成中断。
	bit[21]	AWB 统计完成中断。
	bit[20]	AE 统计完成中断。
	bit[18]	ISP BE 寄存器配置丢失中断。
	bit[17]	ISP BE 寄存器更新中断。
	bit[16]	ISP BE 帧起始中断。



说明

* FE_N, 分别对应 ISP 整体架构图中 FE0, FE1, FE2, FE3, 当 FE_N 取 0~3 时, 分别对应配置 FE0~FE3 的寄存器, 基址是 0x1100_0000。

中断时序

很多中断的位置是由 ISP 模块开关、寄存器配置决定。图 10-4 为各个中断位置的示意图，注意图中中断位置是以一帧内时间前后关系排列。表 10-4 为图 10-4 中断序号与中断事件的对应表格。



图10-4 中断时序示意图

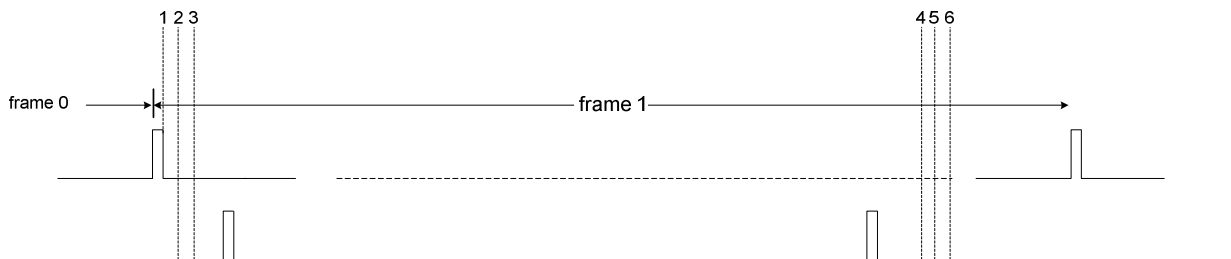


表10-4 中断时序

序号	中断事件
1	fstart(ISP_FE)/fstart(ISP_BE)
2	cfg_loss(ISP_FE)/cfg_loss(ISP_BE) update_cfg(ISP_FE) /update_cfg(ISP_BE)
3	fstart_delay(ISP_FE)
4	ae_int
5	af_int
6	LA 统计完成中断 DEHAZE 统计完成中断 LDCI 统计完成中断

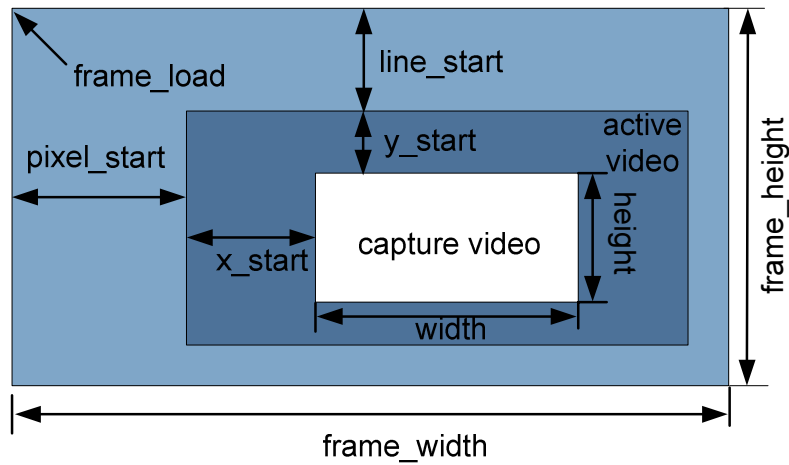
10.5 模块功能

Crop

该模块实现对输入图像裁剪的功能。实际显示的视图区域常常包含在有效视频范围之内，相对有效视频的边界有若干像素缩小。如图 10-5 所示。



图10-5 有效图像区域与水平垂直消隐关系图



FPN

本模块通过标定的黑帧或黑行对 Sensor 输入的图像进行校正，达到去除 Sensor FPN 的目的。

FPN 支持帧模式的标定和校正。在对接 FPN 比较明显的 Sensor 需要开启，Sensor 如果 FPN 不明显，则不需要开。

BLC

提供 Sensor 相关的黑电平校正。分别提供 4 个分量（R，Gr，Gb，B）的偏移量设置。

DPC

本模块提供对静态坏点和动态坏点的检测和校正功能。

- 对于静态坏点，用户可在软件和硬件配合下对 Sensor 的静态坏点进行检测，将全部坏点坐标保存在外部存储器，ISP 工作时根据坐标位置对坏点进行校正处理。
- 对于动态坏点，本模块可根据用户设置的阈值进行自动检测并实时校正，本模块支持单通道 2 像素以内的坏点簇校正。

GE

GE 模块校正 Gr 与 Gb 两个通道的失衡，提高部分场景的图像质量。

WDR

提供多帧合成宽动态功能。在动态范围较大的场景仍然可以看到亮区和暗区细节。



Expander

将 sensor 内部压缩的数据，进行解压缩。

Bayer NR

该模块在 Bayer domain 中实现对图像的去噪，目的是去除噪声的同时，保留细节。该模块可根据用户提供的噪声模型有针对性的对 Sensor 进行噪声消除。

LSC

用于镜头阴影校正。由于镜头的光学特性，会导致 Sensor 影像边缘区域接收的光强比中心小，因此需要根据像素的位置做一个增益的补偿。

LSC 可以提供 4 个分量（R，Gr，Gb，B）的增益，每个增益的精度可调（由 meshScale 控制）。通过对整幅图像进行上下左右对称分窗（分窗数为 32x32），并提供窗口顶点增益，其余的点通过插值得到。插值求得的增益与对应的像素点值相乘输出补偿后的图像数据。

DG

提供数字增益。分别提供 4 个分量（R，Gr，Gb，B）的增益设置，精度为 U8.8。

LA

LA 模块统计 DRC 后分块均值，与 AE 统计分块均值相比，可以得出分块均值增益最大值。LA 统计信息包含 8bit 精度分块 R/Gr/Gb/B 均值统计，分块最大支持 17*15。

AE

实现自动曝光信息的统计，软件根据统计信息调节 Sensor 可实现自动曝光的功能。AE 统计信息包含分区间的 R/Gr/Gb/B 均值统计、带权重的全局 R/Gr/Gb/B 均值、1024 段直方图信息。

- AE 区域统计信息

AE 最大支持 17×15 分块，最小支持 1×1 分块，每个分块均可以输出 R，Gr，Gb，B 均值（Average R/Gr/Gb/B）。区块的统计信息读取地址与数据的对应关系如表 10-5 所示。m 表示水平分区个数，n 表示垂直分区个数。

表10-5 AE 区域统计信息

存储器地址	Zone	存储器读数据			
		MEM_AVER_R_GR [31:16]	MEM_AVER_R_GR [15:0]	_MEM_AVER_GB_B [31:16]	_MEM_AVER_GB_B [15:0]
0	0	Average R	Average Gr	Average Gb	Average B
1	1	Average R	Average Gr	Average Gb	Average B



存储器地址	Zone	存储器读数据			
		MEM_AVER_R_GR [31:16]	MEM_AVER_R_GR [15:0]	_MEM_AVER_GB_B [31:16]	_MEM_AVER_GB_B [15:0]
2	2	Average R	Average Gr	Average Gb	Average B
3	3	Average R	Average Gr	Average Gb	Average B
.....					
m*n-3	m*n-3	Average R	Average Gr	Average Gb	Average B
m*n-2	m*n-2	Average R	Average Gr	Average Gb	Average B
m*n-1	m*n-1	Average R	Average Gr	Average Gb	Average B

- 带权重的 AE 全局统计信息
与 AE 的区域统计信息基本一致，对于整幅图像，AE 也提供了 4 个全局统计信息。其物理意义与分块统计一致。
- 带权重的 AE 直方图信息
AE 提供 1024 阶直方图统计 0-1023，直方图信息读取地址与数据的对应关系如表 10-6 所示。

表10-6 AE 直方图统计信息

存储器地址	Zone	存储器读数据	
		[31:29]	[28:0]
0	0	0	Pixel Value 为 0 的个数
1	1	0	Pixel Value 为 1 的个数
2	2	0	Pixel Value 为 2 的个数
3	3	0	Pixel Value 为 3 的个数
.....			
1022	1022	0	Pixel Value 为 1022 的个数
1023	1023	0	Pixel Value 为 1023 的个数



AF 统计信息

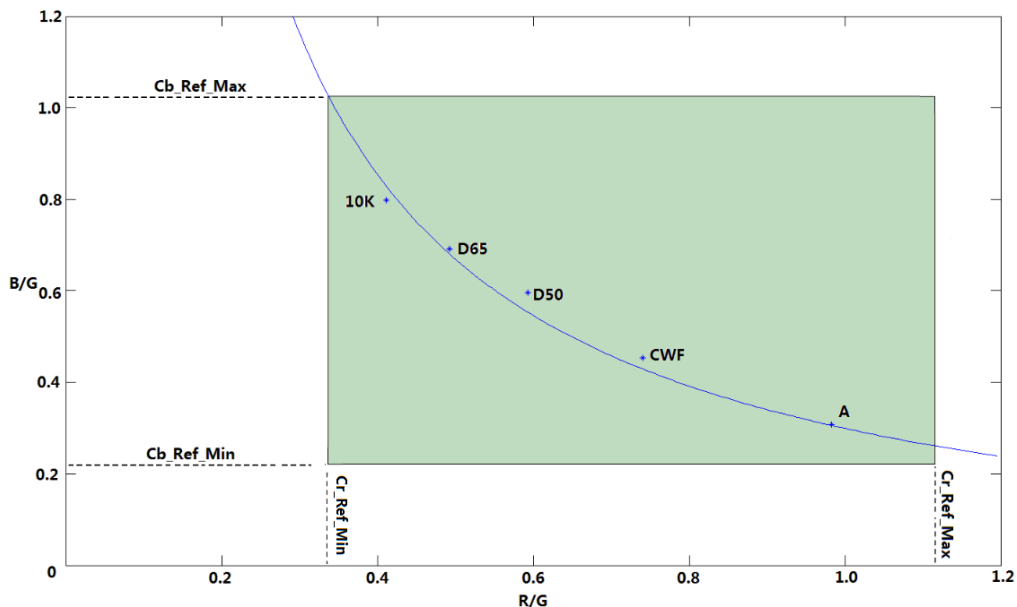
该模块支持图像清晰度评价信息统计。图像区块数目可配，区块个数最大 17×15 ，区块宽高最小 32×32 ，区块宽高最大 1023×1023 ，每个区域分别提供清晰度评价信息。

AWB

AWB 统计信息包含全局统计信息和区域统计信息。

- 全局统计信息：整幅图像的 R，G，B 均值，以及有效统计点的个数。
- 区域统计信息的分块统计方式：
 - 支持最大区块个数 32×32 ，每个分块均可以输出 RGB 均值，以及有效统计点的个数。
 - 有效统计点定义：亮度要满足配置的 RGB 上限与下限要求，颜色需要满足配置的有效像素颜色空间限制要求。

图10-6 AWB 有效像素颜色空间



- AWB 区域统计信息：

AWB 中最大支持图像的 32×32 分块，每个分块均可以输出 RGB 均值（Average R/G/B）。其中统计的像素点均为 16bit，分块内统计点的个数统计值为有效点个数 Count All。点数统计值均已根据图像大小做了归一化处理，归一化到 16bit。区块的统计信息读取地址与数据的对应关系如表 10-7 所示。



表10-7 AWB 区域统计方式 A 信息

存储器地址	Zone	存储器读数据	
		[31:16]	[15:0]
0	0	Average G	Average R
1	0	Count All	Average B
2	1	Average G	Average R
.....			
2046	1023	Average G	Average R
2047	1023	Count All	Average B

- AWB 全局统计信息:

与 AWB 的区域统计信息基本一致，对于整幅图像，AWB 也提供了四个全局统计信息。其物理意义与分块统计一致。

WB

提供白平衡功能。分别提供 4 个分量（R，G_r，G_b，B）的增益设置，精度为 U4.8。

DRC

该模块用于图像动态范围进行压缩（Dynamic Range Compression）。用于调整图像的显示动态范围，使之在显示设备上的显示效果与人眼感知一致。

CAC

CAC 模块用来校正由镜头引入的轴向色差（紫边）与横向色差（物体相对两侧带有不同颜色的彩色边缘）。

DEMOSAIC

Demosaic 模块将 Bayer 格式的 Raw 图像转换到 RGB 图像。

该模块可以分析图像内部边缘特征，利用像素相关性插值，在保证较高分辨率和清晰度的同时，有效抑制伪色彩的产生。

CCM

CCM 通过标准 3×3 的矩阵和矢量偏移量可完成颜色空间的线性校正。

利用预先计算出的几组 CCM 系数，Firmware 根据图像色温动态计算出当前图像的 CCM 系数。Firmware 也可根据当前图像的亮度情况，利用 CCM 矩阵动态调整饱和度。

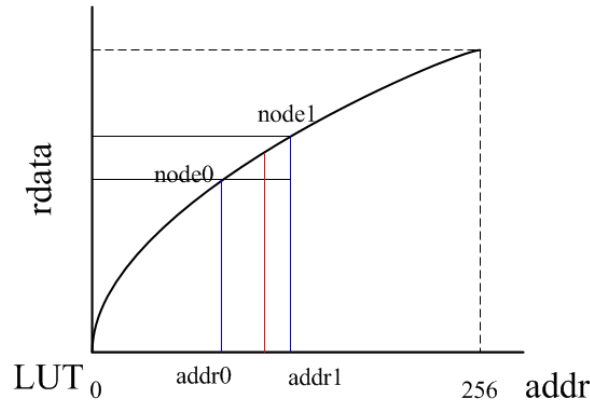
$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} coef00 & coef01 & coef02 \\ coef10 & coef11 & coef12 \\ coef20 & coef21 & coef22 \end{pmatrix} \times \begin{pmatrix} R+in\_dc0 \\ G+in\_dc1 \\ B+in\_dc2 \end{pmatrix} + \begin{pmatrix} out\_dc0 \\ out\_dc1 \\ out\_dc2 \end{pmatrix}$$

R、G、B 是输入数据，coef 是一个 3×3 的可配置矩阵系数。系数均为 15bit，格式 S5.10。

GAMMA

Gamma/preGamma 模块应用于 rgb/bayer 阀，输出 Gamma/Pregamma 调节结果。该模块根据伽马曲线调整亮度。颜色通道都会根据伽马曲线调整。

图10-7 Gamma 曲线



Gamma 曲线由 257 个节点组成，即 0...256，每个节点对应的曲线值为 12bit 无符号数，两个节点之间的点通过线性插值得到。gamma[0]= 0、gamma[256]= 0xFF。

DEHAZE

本模块提供强大的分区域去雾能力以改善雾霾场景下视频的对比度和清晰度。

模块对区域内的图像特性进行分析，得到各区域内的对比度指标，然后对区域内的像素进行增强处理，每个像素的增强强度取决于本区域以及周边区域的对比度指标。

CSC

CSC 通过标准 3×3 的矩阵和矢量偏移量将输入{R, G, B}转换为{Y, U, V}。

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} coef00 & coef01 & coef02 \\ coef10 & coef11 & coef12 \\ coef20 & coef21 & coef22 \end{pmatrix} \times \begin{pmatrix} R+in\_dc0 \\ G+in\_dc1 \\ B+in\_dc2 \end{pmatrix} + \begin{pmatrix} out\_dc0 \\ out\_dc1 \\ out\_dc2 \end{pmatrix}$$

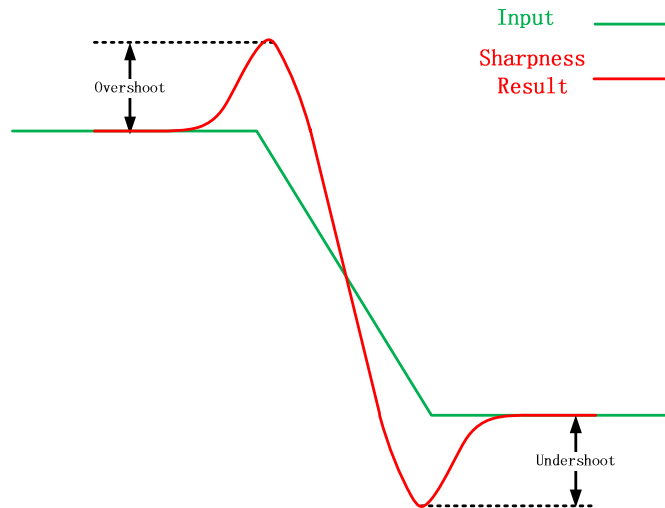
可根据对转换格式的需求改变参数。



SHARPEN

Sharpen 模块实现图像的锐化，提高图像的清晰度。通过控制参数 `edge_amt` 和 `sharp_amt`，可以调节锐化强度，但是太强的锐化可能会放大噪声。软件根据 ISO 值调整参数配置，达到清晰度和抑制噪声的平衡，提升图像的视觉效果。

图10-8 锐化、过冲说明



另外，锐化可能造成边沿正向、负向增幅过大，出现白边和黑边现象，如图 10-8。通过调整 `overshoot` 和 `undershoot` 控制参数，能够抑制白边、黑边现象。

CDS

CDS 模块实现 YUV444 转换到 YUV422 或 YUV420。CDS 对色度水平方向进行多阶滤波，当需要输出 YUV420 时，CDS 对色度垂直方向进行平均下采样。通过配置合适的滤波参数，使图像转换的视觉损失最小。

CA

提供饱和度调整功能。

CLUT

COLOR_3D_LUT 是利用 17x17x17 大小的 3D LUT 实现复杂的颜色调整操作，比如亮度的调整，饱和度的调整，阴影区域，中间亮度区域，高亮区域分别调整。

LDCI

LDCI 基于局域直方图均衡的方法来增强局部的对比度，提升暗区细节，同时对图像中高频进行一定的增强，提升对比度。



HLC

将图像中亮度最高的区域拉灰，以减少类似于车灯等高亮光源对人眼的刺激。

3DNR

把图像中的高斯噪声去除，使得图像变得平滑，同时降低了编码码率。

DE

改善噪声形态，使得噪声均匀细碎，同时改善边缘毛糙。



目 录

11 音频接口.....	11-1
11.1 AIAO	11-1
11.1.1 概述.....	11-1
11.1.2 特点.....	11-2
11.1.3 功能描述.....	11-2
11.1.4 工作方式.....	11-5
11.1.5 AIAO 寄存器概览	11-7
11.1.6 AIAO 寄存器描述	11-8
11.2 Audio Codec	11-30
11.2.1 概述.....	11-30
11.2.2 特点.....	11-30
11.2.3 Audio Codec 寄存器概览	11-31
11.2.4 Audio Codec 寄存器描述	11-31



插图目录

图 11-1 AIAO 框图.....	11-1
图 11-2 通过 I ² S 接口主模式与内置 codec 连接示意图.....	11-3
图 11-3 通过 I ² S/PCM 接口主/从模式连接外置 codec 示意图	11-3
图 11-4 AOP1 和 HDMI 连接示意图	11-3
图 11-5 I ² S 接口时序.....	11-4
图 11-6 PCM 接口标准模式时序	11-4
图 11-7 PCM 接口自定义模式时序	11-4



表格目录

表 11-1 AIAO 寄存器概览（基址是 0x113B_0000）	11-7
表 11-2 Audio Codec 寄存器概览（基址是 0x113C_0000）	11-31



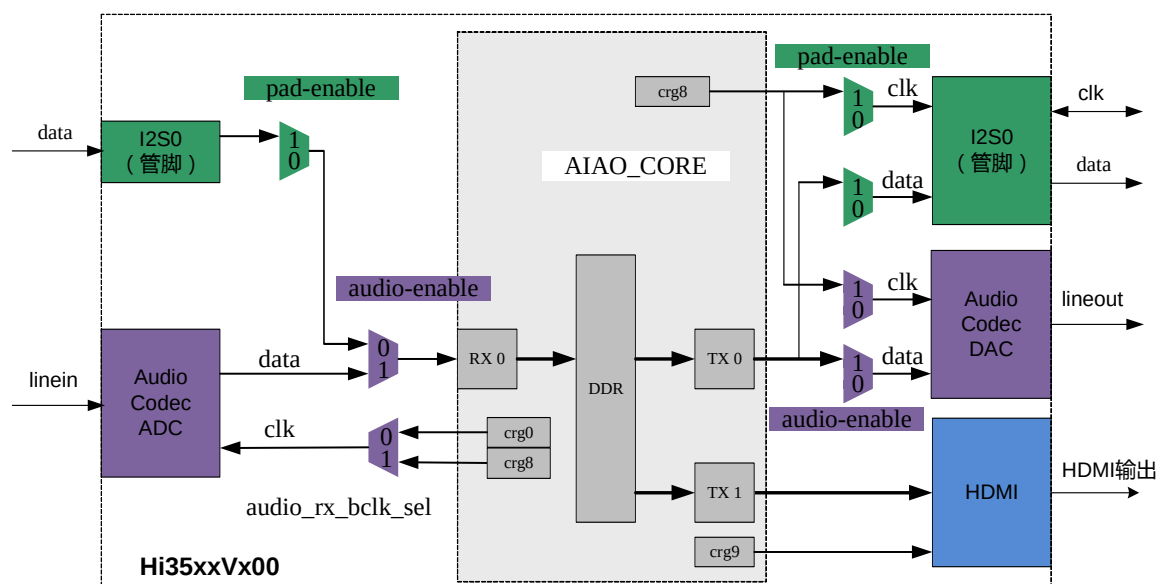
11 音频接口

11.1 AIAO

11.1.1 概述

音频输入输出接口 AIAO (Audio Input/Audio Output)，用于和片内或片外 AudioCodec 的对接，完成音频数据的输入和输出，以实现录音、对讲、回放等功能。芯片内部集成 1 个 AIAO，包含 1 个 AIP (Audio Input Port) 和 2 个 AOP (Audio Output Port)，支持立体声输入输出。基本模块框图如图 11-1 所示：

图11-1 AIAO 框图





说明

绿色：外置 codec 相关通路；
紫色：内置 codec 相关通路；
灰色：内部逻辑；
蓝色：HDMI 播放通路；
Mux 悬空的管脚表示接数字 0，固定值。

11.1.2 特点

AIAO 接口支持 I²S 和 PCM（Pulse Code Modulation）两种模式，采用 DMA 方式存取数据。

PCM 接口

PCM 接口有如下特点：

- 支持主模式单声道 16bit 线性 PCM 编码的发送和接收。
- PCM 接口帧同步信号仅支持短脉冲同步信号（同步信号的持续时间为 1 个时钟周期），支持标准和自定义 2 种模式。
- 接收（AIP）和发送（AOP）相互独立，可以单独使能或关闭。
- 接收（AIP）和发送（AOP）均采用 DMA 操作，通过软件开辟的循环缓冲区存取数据，循环缓冲区大小和水线可调。

I²S 接口

I²S 接口有如下特点：

- 支持主/从模式立体声 16/24bit 数据的发送和接收。
- 支持 8kHz~96kHz 采样率。
- 接收（AIP）和发送（AOP）相互独立，可以单独使能或关闭。
- 接收（AIP）和发送（AOP）均采用 DMA 操作，通过软件开辟的循环缓冲区存取数据，循环缓冲区大小和水线可调。

11.1.3 功能描述

典型应用

芯片内部集成 1 个 AIP 和 2 个 AOP，其中：

- AIP 支持上述 PCM 或 I2S 模式下的数据接收。
- AOP0 支持上述 PCM 或 I2S 模式下的音乐播放。
- AOP1 通过 I2S 模式和 HDMI 连接，将音频数据发送给 HDMI。
- AIP 支持 I2S 主模式对接内部 Audio Codec 或者 I2S/PCM 主/从模式对接外部 ADC 作声音采集。
- AOP0 支持 I2S 主模式对接内部 Audio Codec 或者 I2S/PCM 主/从模式对接外部 DAC 作音乐播放。



- AOP1 支持 I2S 主模式和内部 HDMI tx 模块对接，通过 HDMI 播放音频。

图11-2 通过 I²S 接口主模式与内置 codec 连接示意图

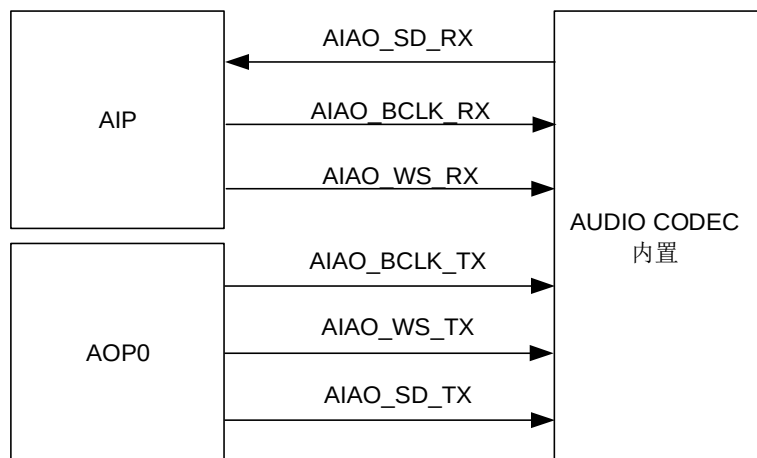


图11-3 通过 I²S/PCM 接口主/从模式连接外置 codec 示意图

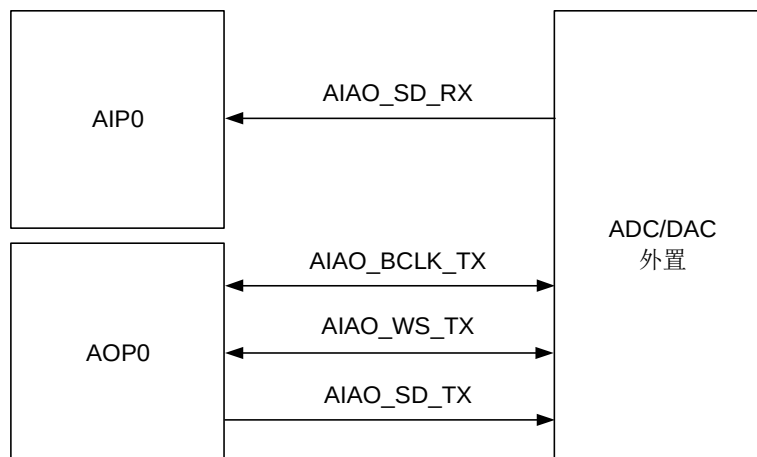
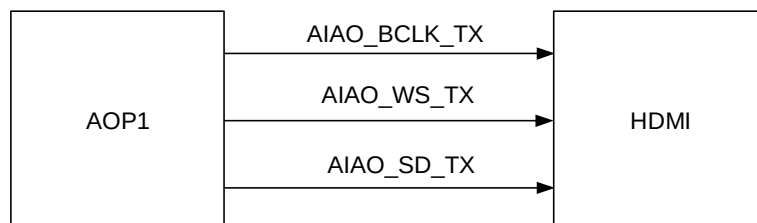


图11-4 AOP1 和 HDMI 连接示意图





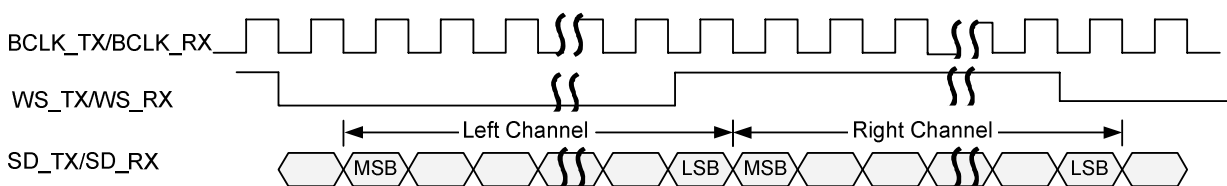
功能原理

AIP 通过 I²S 或 PCM 接口接收对接内部集成的 Audio Codec 或外置 ADC 进行 AD (Analog-to-Digital) 转换后的音频数据，存入为 AIP 开辟的循环缓冲区，然后由 CPU 取走并存储，从而完成录音功能。

AOP0/1 从循环缓冲区中读取音频数据，然后按照设定的采样率，把音频数据通过 I²S 或 PCM 接口传送给对接内部集成的 Audio Codec、外置 DAC 或 HDMI 进行 DA (Digital-to-Analog) 转换后进行声音播放。

对接外部 I2S 接口时，支持的 I2S 时序如图 11-5。

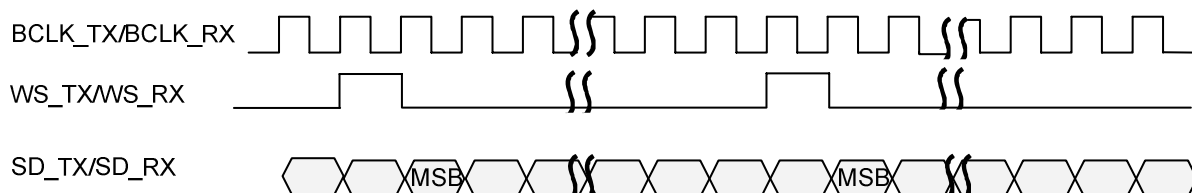
图11-5 I²S 接口时序



I2S 采用 MSB FIRST 方式传输，数据与 WS 信号使用 BCLK 的下降沿发出，在 BCLK 的上升沿采样。数据相对 WS 信号延迟一个 BCLK 周期。

对接外部 PCM 接口时，支持 PCM 标准时序和数据左对齐时序。标准时序如图 11-6。

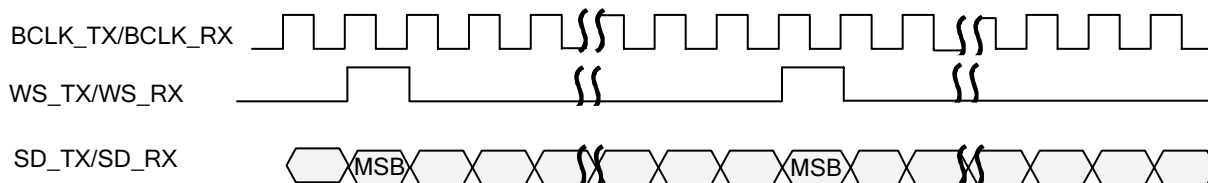
图11-6 PCM 接口标准模式时序



PCM 模式采用 MSB FIRST 方式传输，数据与 WS 信号使用 BCLK 上升沿发出，在 BCLK 下降沿被采样。标准模式中数据相对于 WS 信号延迟一个 BCLK 周期。

PCM 自定义模式时序如图 11-7 所示。

图11-7 PCM 接口自定义模式时序



自定义模式时，数据与 WS 脉冲同拍开始发出。



11.1.4 工作方式

通路复用配置

内部集成的 Audio Codec 与外部 I2S/PCM 对接的 ADC/DAC 不能同时工作。

配置 `i2s_pad_enable` 寄存器为 1 和 `audio_enable` 寄存器为 0 时，连接 AIP/AOP0 通路到外部 I2S/PCM 管脚。反之配置 `i2s_pad_enable` 寄存器为 0 和 `audio_enable` 寄存器为 1 时，连接 AIP/AOP0 到内部集成的 Audio Codec。

AOP1 和 HDMI 的连接关系是不可配置的。包括时钟和数据都是固定连接关系。

时钟门控及时钟配置

在使能 AIAO 进行录音或播放时，必须先打开 AIAO 中对应通道（AIP/AOP0/AOP1）的时钟门控。具体步骤如下：

- 步骤 1. 配置系统 CRG 寄存器的 `PERI_CRG103` 为 0xa，解除 AIAO 复位，并打开时钟门控。
- 步骤 2. 配置 AIAO 寄存器的 `I2S_CRG_CFG0_00`、`I2S_CRG_CFG1_00`、`I2S_CRG_CFG0_08`、`I2S_CRG_CFG1_08`、`I2S_CRG_CFG0_09`、`I2S_CRG_CFG1_09`。选择合适的分频系数。注意使用内部集成的 Audio Codec 时，Audio Codec 配置的 `i2s1_fs_sel` 寄存器的 MCLK/WS 分频比要与此处配置的一致。

----结束

软复位

AIAO 内部的 3 个通道（AIP、AOP0、AOP1）支持独立的软复位，当复位 AIAO 模块时，3 条通道同时复位。

录音工作流程

录音步骤如下：（假设场景为 I2S 模式 48K，2 声道 16bit 精度，假设 AIAO PLL 源头时钟为 1188M，系统控制器 AIAO 时钟已经使能）：

- 步骤 1. 配置寄存器 `I2S_CRG_CFG0_00` 为 0x152EF0，此时时钟输出 MCLK 频率为 12.288MHz。
- 步骤 2. 配置寄存器 `I2S_CRG_CFG1_00` 为 0x0000c133，此时使能时钟，同时把 BCLK 配置为 MCLK 的 4 分频，FCLK 配置为 BCLK 的 64 分频，此时 FCLK 频率为 48KHz。
- 步骤 3. 配置 `RX_IF_ATTRI` 为 0xE4800014，此时设置接收通道工作在 I²S 模式下，两声道，采样精度为 16bit。
- 步骤 4. 配置 `RX_BUFF_SADDR` 寄存器为分配 DDR 的起始地址，比如 0x00000100 配置 `RX_BUFF_SIZE` 寄存器为分配 DDR_BUF 的大小，比如 0x0000F000，配置 `RX_BUFF_WPTR` 寄存器和 `RX_BUFF_RPTR` 寄存器为 0x0，初始化读写指针。配置 `RX_TRANS_SIZE` 寄存器，比如 0x00000F00。
- 步骤 5. 根据需要，使能接收通路相应的中断位，即配置寄存器 `RX_INT_ENA`，比如配置为 0x00000001，只是能 `trans_int` 中断。



- 步骤 6. 配置寄存器 `RX_DSP_CTRL` 为 0x10000000，使能接收通道，接收通道开始工作，录音开始。
- 步骤 7. 通过读取 `RX_BUFF_WPTR` 和 `RX_BUFF_RPTR` 的值判断循环缓冲区的空/满状态以及有效数据量；要保证在循环缓冲区满之前将数据取走，并将更新后的循环缓冲区读地址写入 `RX_BUFF_RPTR`，否则可能会造成循环缓冲区溢出，声音不连续。
- 步骤 8. 录音完成后，写寄存器 `RX_DSP_CTRL` 为 0x00000000，不断查询 `RX_DSP_CTRL` 寄存器，直到看到其变为 0x20000000 后，说明接收通道停止完成。

----结束



注意

启动 AIP 之前，必须完成 AIP 时钟的配置，以保证 AIAO_BCLK_RX 和 AIAO_WS_RX 正常。

播放工作流程

播放步骤如下：（假设场景为 I²S 模式 48K2 声道 16bit 精度，假设 AIAO PLL 源头时钟为 1188M，系统控制器 AIAO 时钟已经使能；使用 AOP0 进行播放）：

- 步骤 1. 配置寄存器 `I2S_CRG_CFG0_08` 为 0x152EF0，此时时钟输出 MCLK 频率为 12.288MHz。
- 步骤 2. 配置寄存器 `I2S_CRG_CFG1_08` 为 0x0000c133，此时使能时钟，同时把 BCLK 配置为 MCLK 的 4 分频，FCLK 配置为 BCLK 的 64 分频，此时 FCLK 频率为 48KHz。
- 步骤 3. 配置 `TX_IF_ATTRI` 寄存器为 0xE4000014，即配置发送接口工作于 I²S 模式下，两声道，16bit 采样精度。
- 步骤 4. 配置 `TX_BUFF_SADDR` 为分 BUF 的起始地址，比如 0x00000100，`TX_BUFF_SIZE` 为分配 BUF 的大小，初始化 `TX_BUFF_WPTR` 为 0x0，`TX_BUFF_RPTR` 为 0x0，设置 `TX_TRANS_SIZE`。（这一步配置可以参考录音流程）
- 步骤 5. 根据需要，使能接收通路相应的中断位，即配置寄存器 `TX_INT_ENA`，比如配置为 0x00000001，只使能 trans_int 中断。
- 步骤 6. 配置寄存器 `TX_DSP_CTRL` 为 0x10000000，使能播放通道。
- 步骤 7. 通过读取 `TX_BUFF_WPTR` 和 `TX_BUFF_RPTR` 的值判断循环缓冲区的空/满状态以及有效数据量；要保证在循环缓冲区空之前将新的音频数据填入，并将更新后的循环缓冲区写地址写入 `TX_BUFF_WPTR`，否则可能会造成循环缓冲区下溢出，声音不连续。
- 步骤 8. 播放结束后，配置寄存器 `TX_DSP_CTRL` 为 0x00000000，停止播放通道，查询 `TX_DSP_CTRL` 寄存器，当看到其值变为 0x20000000 时，说明通道停止完成。

----结束



注意

启动 AOP 前，必须完成 AOP 时钟的配置，以保证 AIAO_BCLK_TX 和 AIAO_WS_TX 正常。

向 AOP 的循环缓冲区写入数据以及更新 [TX_BUFF_WPTR](#) 时，必须保证 AOP 的循环缓冲区空闲空间不小于 32 字节。

11.1.5 AIAO 寄存器概览

AIAO 寄存器概览如[表 11-1](#)所示。

表11-1 AIAO 寄存器概览（基址是 0x113B_0000）

偏移地址	名称	描述	页码
0x0000	AIAO_INT_ENA	AIAO 模块中断使能寄存器	11-8
0x0004	AIAO_INT_STATUS	AIAO 模块中断状态寄存器	11-9
0x0008	AIAO_INT_RAW	AIAO 模块原始中断寄存器	11-9
0x0100	I2S_CRG_CFG0_00	I2S00 CRG 配置 0 号寄存器	11-10
0x0104	I2S_CRG_CFG1_00	I2S00 CRG 配置 1 号寄存器	11-10
0x0140	I2S_CRG_CFG0_08	I2S08 CRG 配置 0 号寄存器	11-11
0x0144	I2S_CRG_CFG1_08	I2S08 CRG 配置 1 号寄存器	11-11
0x0148	I2S_CRG_CFG0_09	I2S09 CRG 配置 0 号寄存器	11-11
0x014C	I2S_CRG_CFG1_09	I2S09 CRG 配置 1 号寄存器	11-11
0x1000	RX_IF_ATTRI	接收通道的接口属性设置寄存器	11-14
0x1004	RX_DSP_CTRL	接收处理通道的控制寄存器	11-15
0x1080	RX_BUFF_SADDR	接收通道的 DDR 缓存起始地址寄存器	11-16
0x1084	RX_BUFF_SIZE	接收通道的 DDR 缓存大小寄存器	11-16
0x1088	RX_BUFF_WPTR	接收通道的 DDR 缓存写地址寄存器	11-16
0x108C	RX_BUFF_RPTR	接收通道的 DDR 缓存读地址寄存器	11-17
0x1090	RX_BUFF_ALFULL_TH	接收通道的 DDR 缓存几乎满水线寄存器	11-17
0x1094	RX_TRANS_SIZE	接收通道的数据传输长度寄存器	11-18
0x10A0	RX_INT_ENA	接收通道的中断使能寄存器	11-18



偏移地址	名称	描述	页码
0x10A4	RX_INT_RAW	接收通道的原始中断寄存器	11-19
0x10A8	RX_INT_STATUS	接收通道的中断状态寄存器	11-19
0x10AC	RX_INT_CLR	接收通道的中断清除寄存器	11-20
0x2000	TX_IF_ATTRI	发送通道的接口属性设置寄存器	11-21
0x2004	TX_DSP_CTRL	发送处理通道的控制寄存器	11-22
0x2080	TX_BUFF_SADDR	发送通道的 DDR 缓存起始地址寄存器	11-24
0x2084	TX_BUFF_SIZE	发送通道的 DDR 缓存大小寄存器	11-24
0x2088	TX_BUFF_WPTR	发送通道的 DDR 缓存写地址寄存器	11-25
0x208C	TX_BUFF_RPTR	发送通道的 DDR 缓存读地址寄存器	11-25
0x2090	TX_BUFF_ALEMP Y_TH	发送通道的 DDR 缓存几乎空水线寄存器	11-25
0x2094	TX_TRANS_SIZE	发送通道的数据传输长度寄存器	11-26
0x20A0	TX_INT_ENA	发送通道的中断使能寄存器	11-26
0x20A4	TX_INT_RAW	发送通道的原始中断寄存器	11-27
0x20A8	TX_INT_STATUS	发送通道的中断状态寄存器	11-28
0x20AC	TX_INT_CLR	发送通道的中断清除寄存器	11-29

11.1.6 AIAO 寄存器描述

AIAO_INT_ENA

AIAO_INT_ENA 为 AIAO 模块中断使能寄存器。

Offset Address: 0x0000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:18]	-	reserved	保留。	0x000
[17]	RW	tx_ch1_int_ena	发送通道 1 的中断使能。 0: 不使能; 1: 使能。	0x0
[16]	RW	tx_ch0_int_ena	发送通道 0 的中断使能。 0: 不使能; 1: 使能。	0x0



Bits	Access	Name	Description	Reset
[15:1]	RO	reserved	保留。	0x0000
[0]	RW	rx_ch0_int_ena	接收通道 0 的中断使能。 0: 不使能; 1: 使能。	0x0

AIAO_INT_STATUS

AIAO_INT_STATUS 为 AIAO 模块中断状态寄存器。

Offset Address: 0x0004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:18]	-	reserved	保留。	0x000
[17]	RO	tx_ch1_int_status	发送通道 1 的中断状态。 0: 无中断状态; 1: 有中断状态。	0x0
[16]	RO	tx_ch0_int_status	发送通道 0 的中断状态。 0: 无中断状态; 1: 有中断状态。	0x0
[15:1]	RO	reserved	保留。	0x0000
[0]	RO	rx_ch0_int_status	接收通道 0 的中断状态。 0: 无中断状态; 1: 有中断状态。	0x0

AIAO_INT_RAW

AIAO_INT_RAW 为 AIAO 模块原始中断寄存器。

Offset Address: 0x0008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:18]	-	reserved	保留。	0x000
[17]	RO	tx_ch1_int_raw	发送通道 1 的原始中断。 0: 无原始中断; 1: 有原始中断。	0x0
[16]	RO	tx_ch0_int_raw	发送通道 0 的原始中断。	0x0



Bits	Access	Name	Description	Reset
			0: 无原始中断; 1: 有原始中断。	
[15:1]	RO	reserved	保留。	0x0000
[0]	RO	rx_ch0_int_raw	接收通道 0 的原始中断。 0: 无原始中断; 1: 有原始中断。	0x0

I2S_CRG_CFG0_00

I2S_CRG_CFG0_00 为 I2S00 CRG 配置 0 号寄存器。

Offset Address: 0x0100 Total Reset Value: 0x00AA_AAAA

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x0
[26:0]	RW	aiao_mclk_div	mclk 的分频时钟配置值，配置值为(目标 mclk 频率/mclk 时钟源头 pll 频率) $\times 2^{27}$ 。其中 mclk 时钟源头 pll 频率请参见 3.2 时钟，默认为 1188M。	0x0AA_AAAA

I2S_CRG_CFG1_00

I2S_CRG_CFG1_00 为 I2S00 CRG 配置 1 号寄存器。

Offset Address: 0x0104 Total Reset Value: 0x0000_0131

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x0000000
[9]	RW	aiao_srst_req	RX0 通道的独立软复位请求。 0: 撤消复位; 1: 复位。	0x0
[8]	RW	aiao_cken	CRG00 的 MCLK/BCLK/WS 时钟门控。 0: 关闭; 1: 打开。	0x1
[7]	-	reserved	保留。	0x0
[6:4]	RW	aiao_fsclk_div	位时钟 BCLK 与采样时钟 FS 的分频关系。 000: FS 是 BCLK 的 16 分频;	0x3



Bits	Access	Name	Description	Reset
			001: FS 是 BCLK 的 32 分频; 010: FS 是 BCLK 的 48 分频; 011: FS 是 BCLK 的 64 分频; 100: FS 是 BCLK 的 128 分频; 101: FS 是 BCLK 的 256 分频; 其他: FS 是 BCLK 的 8 分频。	
[3:0]	RW	aiao_bclk_div	主时钟 MCLK 与位时钟 BCLK 分频关系。 0000: BCLK 是 MCLK 的 1 分频; 0001: BCLK 是 MCLK 的 3 分频; 0010: BCLK 是 MCLK 的 2 分频; 0011: BCLK 是 MCLK 的 4 分频; 0100: BCLK 是 MCLK 的 6 分频; 0101: BCLK 是 MCLK 的 8 分频; 0110: BCLK 是 MCLK 的 12 分频; 0111: BCLK 是 MCLK 的 16 分频; 1000: BCLK 是 MCLK 的 24 分频; 1001: BCLK 是 MCLK 的 32 分频; 1010: BCLK 是 MCLK 的 48 分频; 1011: BCLK 是 MCLK 的 64 分频; 其他: BCLK 是 MCLK 的 8 分频。	0x1

I2S_CRG_CFG0_08

I2S_CRG_CFG0_08 为 I2S08 CRG 配置 0 号寄存器。

Offset Address: 0x0140 Total Reset Value: 0x00AA_AAAA

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x0
[26:0]	RW	aiao_mclk_div	mclk 的分频时钟配置值，配置值为(目标 mclk 频率/mclk 时钟源头 pll 频率) $\times 2^{27}$ 。其中 mclk 时钟源头 pll 频率请参见 3.2 时钟，默认为 1188M。	0x0AA_AAAA

I2S_CRG_CFG1_08

I2S_CRG_CFG1_08 为 I2S08 CRG 配置 1 号寄存器。



Offset Address: 0x0144 Total Reset Value: 0x0000_0131

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x0000000
[9]	RW	aiao_srst_req	TX0 通道的独立软复位请求。 0: 撤消复位; 1: 复位。	0x0
[8]	RW	aiao_cken	CRG08 的 MCLK/BCLK/WS 时钟门控。 0: 关闭; 1: 打开。	0x1
[7]	-	reserved	保留。	0x0
[6:4]	RW	aiao_fsclk_div	位时钟 BCLK 与采样时钟 FS 的分频关系。 000: FS 是 BCLK 的 16 分频; 001: FS 是 BCLK 的 32 分频; 010: FS 是 BCLK 的 48 分频; 011: FS 是 BCLK 的 64 分频; 100: FS 是 BCLK 的 128 分频; 101: FS 是 BCLK 的 256 分频; 其他: FS 是 BCLK 的 8 分频。	0x3
[3:0]	RW	aiao_bclk_div	主时钟 MCLK 与位时钟 BCLK 分频关系。 0000: BCLK 是 MCLK 的 1 分频; 0001: BCLK 是 MCLK 的 3 分频; 0010: BCLK 是 MCLK 的 2 分频; 0011: BCLK 是 MCLK 的 4 分频; 0100: BCLK 是 MCLK 的 6 分频; 0101: BCLK 是 MCLK 的 8 分频; 0110: BCLK 是 MCLK 的 12 分频; 0111: BCLK 是 MCLK 的 16 分频; 1000: BCLK 是 MCLK 的 24 分频; 1001: BCLK 是 MCLK 的 32 分频; 1010: BCLK 是 MCLK 的 48 分频; 1011: BCLK 是 MCLK 的 64 分频; 其他: BCLK 是 MCLK 的 8 分频。	0x1



I2S_CRG_CFG0_09

I2S_CRG_CFG0_09 为 I2S09 CRG 配置 0 号寄存器。

Offset Address: 0x0148 Total Reset Value: 0x00AA_AAAA

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x0
[26:0]	RW	aiao_mclk_div	mclk 的分频时钟配置值，配置值为(目标 mclk 频率/mclk 时钟源头 pll 频率) $\times 2^{27}$ 。其中 mclk 时钟源头 pll 频率请参见 3.2 时钟，默认为 1188M。	0x0AA_AAAA

I2S_CRG_CFG1_09

I2S_CRG_CFG1_09 为 I2S09 CRG 配置 1 号寄存器。

Offset Address: 0x014C Total Reset Value: 0x0000_0131

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x0000000
[9]	RW	aiao_srst_req	TX1 通道的独立软复位请求。 0: 撤消复位; 1: 复位。	0x0
[8]	RW	aiao_cken	CRG09 的 MCLK/BCLK/WS 时钟门控。 0: 关闭; 1: 打开。	0x1
[7]	-	reserved	保留。	0x0
[6:4]	RW	aiao_fsclk_div	位时钟 BCLK 与采样时钟 FS 的分频关系。 000: FS 是 BCLK 的 16 分频; 001: FS 是 BCLK 的 32 分频; 010: FS 是 BCLK 的 48 分频; 011: FS 是 BCLK 的 64 分频; 100: FS 是 BCLK 的 128 分频; 101: FS 是 BCLK 的 256 分频; 其他: FS 是 BCLK 的 8 分频。	0x3



[3:0]	RW	aiao_bclk_div	主时钟 MCLK 与位时钟 BCLK 分频关系。 0000: BCLK 是 MCLK 的 1 分频; 0001: BCLK 是 MCLK 的 3 分频; 0010: BCLK 是 MCLK 的 2 分频; 0011: BCLK 是 MCLK 的 4 分频; 0100: BCLK 是 MCLK 的 6 分频; 0101: BCLK 是 MCLK 的 8 分频; 0110: BCLK 是 MCLK 的 12 分频; 0111: BCLK 是 MCLK 的 16 分频; 1000: BCLK 是 MCLK 的 24 分频; 1001: BCLK 是 MCLK 的 32 分频; 1010: BCLK 是 MCLK 的 48 分频; 1011: BCLK 是 MCLK 的 64 分频; 其他: BCLK 是 MCLK 的 8 分频。	0x1

RX_IF_ATTRI

RX_IF_ATTRI 为接收通道的接口属性设置寄存器。

Offset Address: 0x1000 Total Reset Value: 0xE400_0004

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0xE4
[23:20]	RW	rx_sd_source_sel	正常工作配置为 0x1000。	0x0
[19]	-	reserved	保留。	0x0
[18:16]	RW	rx_trackmode	I ² S 模式下，左右声道模式控制。 000: 不做处理; 001: 两个声道全部为左声道声音; 010: 两个声道全部为右声道声音; 011: 两个声道互换; 100: 左右两个声道输出为左右声道相加; 101: 左声道静音，右声道播放原右声道声音; 110: 右声道静音，左声道播放原左声道声音; 111: 左右声道静音。  说明 1ch 接收时，trackmode 仍然有效。	0x0



Bits	Access	Name	Description	Reset
[15:8]	RW	rx_sd_offset	PCM 模式下，数据相对 WS 信号延迟 BCLK 周期数。 0x0: 左对齐模式； 0x1: 标准模式； 其他：保留。	0x00
[7]	RO	reserved	保留。	0x0
[6:4]	RW	rx_ch_num	接收的声道数选择。 00: 1 路(ch)接收； 01: 2 路(ch)接收； 其他：保留。	0x0
[3:2]	RW	rx_i2s_precision	数据采样精度配置位。 I2S 模式： 01: 16bit； 10: 24bit； 其他：保留。 PCM 模式： 01: 16bit； 其他：保留。	0x1
[1:0]	RW	rx_mode	接收通道的接口模式选择。 00: I ² S 模式； 01: PCM 模式； 其他：保留。	0x0

RX_DSP_CTRL

RX_DSP_CTRL 为接收处理通道的控制寄存器。

Offset Address: 0x1004 Total Reset Value: 0x2000_0000

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29]	RO	rx_disable_done	接收通道的停止完成标识位。 0: disable 未完成； 1: disable 完成。	0x1



[28]	RW	rx_enable	接收通道的启动、停止控制位。 0: 停止; 1: 启动。	0x0
[27]	RW	bypass_en	数据处理禁止位, 控制功能仍然生效。 0: 正常进行数据处理; 1: 不进行 trackmode 等运算。	0x0
[26:0]	-	reserved	保留。	0x000000

RX_BUFF_SADDR

RX_BUFF_SADDR 为接收通道的 DDR 缓存起始地址寄存器。

Offset Address: 0x1080 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	rx_buff_saddr	接收通道 0 的 DDR 缓存起始地址, 以字节为单位。  说明 DDR 缓存起始地址要求 128 × 8 比特对齐。	0x0000_0000

RX_BUFF_SIZE

RX_BUFF_SIZE 为接收通道的 DDR 缓存大小寄存器。

Offset Address: 0x1084 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RW	rx_buff_size	接收通道的 DDR 缓存大小, 以字节为单位。  说明 要求 rx_buff_size 是 128 字节的整数倍。	0x000000

RX_BUFF_WPTR

RX_BUFF_WPTR 为接收通道的 DDR 缓存写地址寄存器。

Offset Address: 0x1088 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RW	rx_buff_wptr	接收通道的 DDR 缓存写地址，以字节为单位。  说明 - 接收方向的写地址由逻辑维护，是相对于 DDR 缓存起始地址的偏移地址。 - 要求 128×2 比特对齐。	0x000000

RX_BUFF_RPTR

RX_BUFF_RPTR 为接收通道的 DDR 缓存读地址寄存器。

Offset Address: 0x108C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RW	rx_buff_rptr	接收通道的 DDR 缓存读地址，以字节为单位。  说明 - 接收方向的读地址由软件维护，是相对于 DDR 缓存起始地址的偏移地址。 - 软件按照字节为单位，硬件内部按照 128×2 比特对齐操作	0x000000

RX_BUFF_ALFULL_TH

RX_BUFF_ALFULL_TH 为接收通道的 DDR 缓存几乎满水位寄存器。

Offset Address: 0x1090 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RW	rx_buff_alfull_th	接收通道的 DDR 缓存几乎满水位，以字节为单位。当 DDR 缓存可写空间小于几乎满水位时，产生几乎满原始中断。  说明 如果使用 rx_alfull_int 中断，要求 rx_buff_alfull_th 配置为 16 字节的整数倍，且大于或等于 0x40。	0x000000



RX_TRANS_SIZE

RX_TRANS_SIZE 为接收通道的数据传输长度寄存器。

Offset Address: 0x1094 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RW	rx_trans_size	接收通道，当完成 rx_trans_size 长度(以字节为单位)的音频数据接收时，产生传输完成中断。	0x000000

RX_INT_ENA

RX_INT_ENA 为接收通道的中断使能寄存器。

Offset Address: 0x10A0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	RO	reserved	保留。	0x000000
[7]	RW	rx_if_full_lost_int_ena	接收通道的接口数据满丢失原始中断使能。 0: 不使能; 1: 使能。	0x0
[6]	-	reserved	保留。	0x0
[5]	RW	rx_stop_int_ena	接收通道的停止中断使能。 0: 不使能; 1: 使能。	0x0
[4]	-	reserved	保留。	0x0
[3]	RW	rx_bfifo_full_int_ena	接收通道的总线 fifo 上溢中断使能。 0: 不使能; 1: 使能。	0x0
[2]	RW	rx_alfull_int_ena	接收通道的 DDR 缓存几乎满中断使能。 0: 不使能; 1: 使能。	0x0
[1]	RW	rx_full_int_ena	接收通道的 DDR 缓存满中断使能。 0: 不使能; 1: 使能。	0x0



[0]	RW	rx_trans_int_ena	接收通道的传输完成中断使能。 0: 不使能; 1: 使能。	0x0

RX_INT_RAW

RX_INT_RAW 为接收通道的原始中断寄存器。

Offset Address: 0x10A4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RO	rx_if_full_lost_int_raw	接收通道的接口数据满丢失原始中断。 0: 无原始中断; 1: 有原始中断。	0x0
[6]	-	reserved	保留。	0x0
[5]	RO	rx_stop_int_raw	接收通道的停止原始中断。 0: 无原始中断; 1: 有原始中断。	0x0
[4]	-	reserved	保留。	0x0
[3]	RO	rx_bfifo_full_int_raw	接收通道的总线 fifo 上溢原始中断。 0: 无原始中断; 1: 有原始中断。	0x0
[2]	RO	rx_alfull_int_raw	接收通道的 DDR 缓存几乎满原始中断。 0: 无原始中断; 1: 有原始中断。	0x0
[1]	RO	rx_full_int_raw	接收通道的 DDR 缓存满原始中断。 0: 无原始中断; 1: 有原始中断。	0x0
[0]	RO	rx_trans_int_raw	接收通道的传输完成原始中断。 0: 无原始中断; 1: 有原始中断。	0x0

RX_INT_STATUS

RX_INT_STATUS 为接收通道的中断状态寄存器。



Offset Address: 0x10A8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RO	tx_if_full_lost_int_status	发送通道的接口数据满丢失中断状态。 0: 无中断状态; 1: 有中断状态。	0x0
[6]	-	reserved	保留。	0x0
[5]	RO	rx_stop_int_status	接收通道的停止中断状态。 0: 无中断状态; 1: 有中断状态。	0x0
[4]	-	reserved	保留。	
[3]	RO	rx_bfifo_full_int_status	接收通道的总线 fifo 上溢中断状态。 0: 无中断状态; 1: 有中断状态。	0x0
[2]	RO	rx_alfull_int_status	接收通道的 DDR 缓存几乎满中断状态。 0: 无中断状态; 1: 有中断状态。	0x0
[1]	RO	rx_full_int_status	接收通道的 DDR 缓存满中断状态。 0: 无中断状态; 1: 有中断状态。	0x0
[0]	RO	rx_trans_int_status	接收通道的传输完成中断状态。 0: 无中断状态; 1: 有中断状态。	0x0

RX_INT_CLR

RX_INT_CLR 为接收通道的中断清除寄存器。

Offset Address: 0x10AC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	WC	tx_if_full_lost_int_clear	发送通道的接口数据满丢失中断清除。 0: 不清除; 1: 清除停止中断。	0x0



Bits	Access	Name	Description	Reset
[6]	-	reserved	保留。	0x0
[5]	WC	rx_stop_int_clear	接收通道的停止中断清除位。 0: 不清除; 1: 清除停止中断。	0x0
[4]	-	reserved	保留。	0x0
[3]	WC	rx_bfifo_full_int_clear	接收通道的总线 fifo 上溢中断清除位。 0: 不清除; 1: 清除 fifo 上溢中断。	0x0
[2]	WC	rx_alfull_int_clear	接收通道的 DDR 缓存几乎满中断清除位。 0: 不清除; 1: 清除 DDR 缓存几乎满中断。	0x0
[1]	WC	rx_full_int_clear	接收通道的 DDR 缓存满中断清除位。 0: 不清除; 1: 清除 DDR 缓存满中断。	0x0
[0]	WO	rx_trans_int_clear	接收通道的传输完成中断清除位。 0: 不清除; 1: 清除传输完成中断。	0x0

TX_IF_ATTRI

TX_IF_ATTRI 为发送通道的接口属性设置寄存器。

Offset Address: 0x2000 Total Reset Value: 0xE400_0004

Bits	Access	Name	Description	Reset
[31:19]	-	reserved	保留。	0xE400
[18:16]	RW	tx_trackmode	I ² S 模式下，左右声道模式控制。 000: 不做处理; 001: 两个声道全部为左声道声音; 010: 两个声道全部为右声道声音; 011: 两个声道互换; 100: 左右两个声道输出为左右声道相加; 101: 左声道静音，右声道播放原右声道声音; 110: 右声道静音，左声道播放原左声道声音	0x0



Bits	Access	Name	Description	Reset
			音; 111: 左右声道静音。  说明 1ch 接收时, trackmode 仍然有效。	
[15:8]	RW	tx_sd_offset	PCM 模式下, 数据相对 WS 信号延迟 BCLK 周期数。 0x0: 左对齐模式; 0x1: 标准模式; 其他: 保留。	0x00
[7:6]	-	reserved	保留。	0x0
[5:4]	RW	tx_ch_num	发送路数选择。 00: 单声道发送; 01: 立体声发送; 其他: 保留。	0x0
[3:2]	RW	tx_i2s_precision	数据采样精度配置位。 I2S 模式: 01: 16bit; 10: 24bit; 其他: 保留。 PCM 模式: 01: 16bit; 其他: 保留。	0x1
[1:0]	RW	tx_mode	发送通道的接口模式选择。 00: I2S 模式; 01: PCM 模式; 其他: 保留。	0x0

TX_DSP_CTRL

TX_DSP_CTRL 为发送处理通道的控制寄存器。

Offset Address: 0x2004 Total Reset Value: 0x2000_0000

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29]	RO	tx_disable_done	发送通道的停止完成标识。	0x1



Bits	Access	Name	Description	Reset
			0: 未完成; 1: 完成。	
[28]	RW	tx_enable	发送通道的启动、停止控制位。 0: 停止; 1: 启动。	0x0
[27]	RW	bypass_en	数据处理禁止位, 控制功能仍然生效。 0: 正常进行数据处理; 1: 不进行 trackmode、音量处理等运算。	0x0
[26:24]	-	reserved	保留。	0x0
[23:20]	RW	fade_out_rate	淡出速度。 0000: 1 个采样点改变一次; 0001: 2 个采样点改变一次; 0010: 4 个采样点改变一次; 0011: 8 个采样点改变一次; 0100: 16 个采样点改变一次; 0101: 32 个采样点改变一次; 0110: 64 个采样点改变一次; 0111: 128 个采样点改变一次; 其他: 保留。	0x0
[19:16]	RW	fade_in_rate	淡入速度。 0000: 1 个采样点改变一次; 0001: 2 个采样点改变一次; 0010: 4 个采样点改变一次; 0011: 8 个采样点改变一次; 0100: 16 个采样点改变一次; 0101: 32 个采样点改变一次; 0110: 64 个采样点改变一次; 0111: 128 个采样点改变一次; 其他: 保留。	0x0
[15]	-	reserved	保留。	0x0
[14:8]	RW	volume	音量, 每分度 1DB。 0x00~0x28: 静音; 0x29: - 80dB; 0x2A: - 79dB;	0x00



Bits	Access	Name	Description	Reset
			0x75: -4dB; 0x77: -2dB; 0x79: +0dB; 0x7B: +2dB; 0x7D: +4dB; 0x7F: +6dB (最大)。	
[7:2]	-	reserved	保留。	0x00
[1]	RW	mute_fade_en	静音淡入淡出控制。 0: 淡入淡出功能关闭; 1: 淡入淡出功能打开。	0x0
[0]	RW	mute_en	静音控制。 0: 静音撤销; 1: 静音使能。	0x0

TX_BUFF_SADDR

TX_BUFF_SADDR 为发送通道的 DDR 缓存起始地址寄存器。

Offset Address: 0x2080 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tx_buff_saddr	发送通道的 DDR 缓存起始地址，以字节为单位。  说明 DDR 缓存起始地址要求 128 × 8 比特对齐。	0x0000_0000

TX_BUFF_SIZE

TX_BUFF_SIZE 为发送通道的 DDR 缓存大小寄存器。

Offset Address: 0x2084 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RW	tx_buff_size	发送通道的 DDR 缓存大小，以字节为单位。  说明 要求 tx_buff_size 是 128 字节的整数倍。	0x000000



TX_BUFF_WPTR

TX_BUFF_WPTR 为发送通道的 DDR 缓存写地址寄存器。

Offset Address: 0x2088 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RW	tx_buff_wptr	发送通道的 DDR 缓存写地址。  说明 - 发送方向的写地址由软件维护，是相对于 DDR 缓存起始地址的偏移地址。 - 软件必须保证 TX_BUF 空闲空间不小于 128 字节。 - 软件按照字节为单位，硬件内部按照 128×2 比特对齐操作。	0x000000

TX_BUFF_RPTR

TX_BUFF_RPTR 为发送通道的 DDR 缓存读地址寄存器。

Offset Address: 0x208C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RW	tx_buff_rptr	发送通道的 DDR 缓存读地址。  说明 发送方向的读地址由逻辑维护，是相对于 DDR 缓存起始地址的偏移地址。注意：要求 128 × 2 比特对齐。	0x000000

TX_BUFF_ALEMPY_TH

TX_BUFF_ALEMPY_TH 为发送通道的 DDR 缓存几乎空水准寄存器。

Offset Address: 0x2090 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RW	tx_buff_alempy_th	发送通道的 DDR 缓存几乎空水准，以字节为单位。当 DDR 缓存可读空间小于几乎空	0x000000



Bits	Access	Name	Description	Reset
			水线时，产生几乎空原始中断。  说明 如果使用 tx_alempy_int 中断，要求 tx_buff_alempy_th 配置为 16 字节的整数倍，且大于或等于 0x20。	

TX_TRANS_SIZE

TX_TRANS_SIZE 为发送通道的数据传输长度寄存器。

Offset Address: 0x2094 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RW	tx_trans_size	发送通道，当完成 tx_trans_size 长度(以字节为单位)的音频数据发送时，产生传输完成中断。	0x000000

TX_INT_ENA

TX_INT_ENA 为发送通道的中断使能寄存器。

Offset Address: 0x20A0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RW	tx_dat_break_int_ena	发送通道的接口数据断流中断使能。 0: 无原始中断; 1: 有原始中断。	0x0
[6]	RW	tx_mfade_int_ena	发送通道的静音淡入淡出完成中断使能。 0: 不使能; 1: 使能。	0x0
[5]	RW	tx_stop_int_ena	发送通道的停止中断使能。 0: 不使能; 1: 使能。	0x0
[4]	RW	tx_ififo_empty_int_ena	发送通道的接口 fifo 下溢中断使能。 0: 不使能; 1: 使能。	0x0



Bits	Access	Name	Description	Reset
[3]	RW	tx_bfifo_empty_int_ena	发送通道的总线 fifo 下溢中断使能。 0: 不使能; 1: 使能。	0x0
[2]	RW	tx_aleempty_int_ena	发送通道的 DDR 缓存几乎空中断使能。 0: 不使能; 1: 使能。	0x0
[1]	RW	tx_empty_int_ena	发送通道的 DDR 缓存空中断使能。 0: 不使能; 1: 使能。	0x0
[0]	RW	tx_trans_int_ena	发送通道的传输完成中断使能。 0: 不使能; 1: 使能。	0x0

TX_INT_RAW

TX_INT_RAW 为发送通道的原始中断寄存器。

Offset Address: 0x20A4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RO	tx_dat_break_int_raw	发送通道的接口数据断流原始中断。 0: 无原始中断; 1: 有原始中断。	0x0
[6]	RO	tx_mfade_int_raw	发送通道的静音淡入淡出完成原始中断。 0: 无原始中断; 1: 有原始中断。	0x0
[5]	RO	tx_stop_int_raw	发送通道的停止原始中断。 0: 无原始中断; 1: 有原始中断。	0x0
[4]	RO	tx_ififo_empty_int_raw	发送通道的接口 fifo 下溢原始中断。 0: 无原始中断; 1: 有原始中断。	0x0
[3]	RO	tx_bfifo_empty_int_raw	发送通道的总线 fifo 下溢原始中断。 0: 无原始中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有原始中断。	
[2]	RO	tx_alempy_int_raw	发送通道的 DDR 缓存几乎空原始中断。 0: 无原始中断; 1: 有原始中断。	0x0
[1]	RO	tx_empty_int_raw	发送通道的 DDR 缓存空原始中断。 0: 无原始中断; 1: 有原始中断。	0x0
[0]	RO	tx_trans_int_raw	发送通道的传输完成原始中断。 0: 无原始中断; 1: 有原始中断。	0x0

TX_INT_STATUS

TX_INT_STATUS 为发送通道的中断状态寄存器。

Offset Address: 0x20A8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RO	tx_dat_break_int_status	发送通道的接口数据断流中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[6]	RO	tx_mfade_int_status	发送通道的静音淡入淡出完成中断状态位。 0: 无中断状态; 1: 有中断状态。	0x0
[5]	RO	tx_stop_int_status	发送通道的停止中断状态。 0: 无中断状态; 1: 有中断状态。	0x0
[4]	RO	tx_ififo_empty_int_status	发送通道的接口 fifo 下溢中断状态。 0: 无中断状态; 1: 有中断状态。	0x0
[3]	RO	tx_bfifo_empty_int_status	发送通道的总线 fifo 下溢中断状态。 0: 无中断状态; 1: 有中断状态。	0x0



Bits	Access	Name	Description	Reset
[2]	RO	tx_alempy_int_status	发送通道的 DDR 缓存几乎空中断状态。 0: 无中断状态; 1: 有中断状态。	0x0
[1]	RO	tx_empty_int_status	发送通道的 DDR 缓存空中断状态。 0: 无中断状态; 1: 有中断状态。	0x0
[0]	RO	tx_trans_int_status	发送通道的传输完成中断状态。 0: 无中断状态; 1: 有中断状态。	0x0

TX_INT_CLR

TX_INT_CLR 为发送通道的中断清除寄存器。

Offset Address: 0x20AC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RO	tx_dat_break_int_clear	发送通道的接口数据断流中断清除。 0: 无原始中断; 1: 有原始中断。	0x0
[6]	WC	tx_mfade_int_clear	发送通道的静音淡入淡出完成中断清除位。 0: 不清除; 1: 清除停止中断。	0x0
[5]	WC	tx_stop_int_clear	发送通道的停止中断清除位。 0: 不清除; 1: 清除停止中断。	0x0
[4]	WC	tx_ififo_empty_int_clear	发送通道的接口 fifo 下溢中断清除位。 0: 不清除; 1: 清除 fifo 下溢中断。	0x0
[3]	WC	tx_bfifo_empty_int_clear	发送通道的总线 fifo 下溢中断清除位。 0: 不清除; 1: 清除 fifo 下溢中断。	0x0
[2]	WC	tx_alempy_int_clear	发送通道的 DDR 缓存几乎空中断清除位。	0x0



Bits	Access	Name	Description	Reset
			0: 不清除; 1: 清除 DDR 缓存几乎空中断。	
[1]	WC	tx_empty_int_clear	发送通道的 DDR 缓存空中断清除位。 0: 不清除; 1: 清除 DDR 缓存空中断。	0x0
[0]	WC	tx_trans_int_clear	发送通道的传输完成中断清除位。 0: 不清除; 1: 清除传输完成中断。	0x0

11.2 Audio Codec

11.2.1 概述

芯片集成高性能的 Audio Codec，包括高品质立体声回放 DAC（96dB DR A-Weighted），支持两路单端 lineout 输出；高品质立体声录音 ADC（93dB DR A-Weighted），支持立体声单端输入或一路差分输入，麦克风输入支持-1.5~30dB，其中-1.5dB~0dB 为 1.5dB 步长，0dB~30dB 为 2dB 步长的增益控制，另外一档 Boost gain 为 20dB。I2S 数据接口，支持 8kHz 到 192kHz 的标准采样率，可支持两种采样率同时工作，并支持数字混音。

11.2.2 特点

Audio Codec 模块有如下特点：

- 96dBA DR 立体声 DAC
- 立体声单端 Lineout 输出
- DAC 数字音量控制范围：-121dB~6dB，1dB 步长
- 93dBA DR 立体声 ADC
- ADC 通路模拟音量控制范围：-1.5~30dB，2dB 步长，另外一档 Boost gain 为 20dB
- ADC 通路数字音量控制范围：-96dB~30dB，1dB 步长
- 立体声单端输入或一路差分输入
- 提供内部麦克风偏置
- 支持主从模式 I2S 数据接口，支持 24/20/18/16bits，二进制码
- 音频采样率：支持 48kHz、44.1kHz、32kHz 三个系列的采样率。其中各系列采样率情况如下：
 - 32kHz 系列采样率包括 8kHz、16kHz、32kHz、64kHz、128kHz；



- 44.1kHz 系列采样率包括 11.025kHz、22.05kHz、44.1kHz、88.2kHz、176.4kHz;
- 48kHz 系列采样率包括 12kHz、24kHz、48kHz、96kHz、192kHz。
- 50psRMS 的模拟时钟 CLKIN 的 Jitter 容忍度。

11.2.3 Audio Codec 寄存器概览

Audio Codec 寄存器概览如表 11-2 所示。

表11-2 Audio Codec 寄存器概览（基址是 0x113C_0000）

偏移地址	名称	描述	页码
0x0014	AUDIO_ANA_CTRL_0	AudioCodec 模拟 0 到 3 号寄存器	11-31
0x0018	AUDIO_ANA_CTRL_1	AudioCodec 模拟 4 到 7 号寄存器	11-34
0x001c	AUDIO_ANA_CTRL_2	AudioCodec 模拟 8 到 11 号寄存器	11-36
0x0020	AUDIO_ANA_CTRL_3	AudioCodec 模拟 12 到 15 号寄存器	11-38
0x0024	AUDIO_ANA_CTRL_4	AudioCodec 模拟 16 到 19 号寄存器	11-41
0x0030	AUDIO_CTRL_REG_1	Audio Codec 控制寄存器 1	11-46
0x0034	AUDIO_DAC_REG_0	Audio Codec 控制寄存器 2	11-48
0x0038	AUDIO_DAC_REG_1	Audio Codec 控制寄存器 3	11-49
0x003c	AUDIO_ADC_REG_0	Audio Codec 控制寄存器 4	11-50
0x0048	ACODEC_REG18	I2S 通路选择控制寄存器(不可软复位)	11-52

11.2.4 Audio Codec 寄存器描述

AUDIO_ANA_CTRL_0

AUDIO_ANA_CTRL_0 为 AudioCodec 模拟 0 到 3 号寄存器。

Offset Address: 0x0014 Total Reset Value: 0x0405_78E1

Bits	Access	Name	Description	Reset
[31]	RW	IBADJ_CTCM	CTCM, MICBIAS, LINEIN 偏置电流控制 0: 5uA; (推荐值) 1: 10uA;	0x0
[30:29]	RW	SEL_CLK_CHOP_L INEOUT_D	Lineout 差分模式 chopper 时钟频率选择 00: 96kHz;	0x0



Bits	Access	Name	Description	Reset
			01: 192kHz; 10: 384kHz; 11: 768kHz。	
[28]	RW	CTRL_CLK_DAC_PH	DAC 的工作时钟沿选择(主时钟二分频) 0: 上升沿工作; 1: 下降沿工作;	0x0
[27:26]	RW	IBADJ_DAC	DAC 偏置电流控制信号 00: 3uA; 01: 5uA(推荐值); 10: 7uA; 11: 10uA。	0x1
[25:24]	-	reserved	保留。	0x0
[23:22]	RW	SEL_CLK_CHOP_DAC_VREF	DAC VREF chopper 时钟频率选择 00: 96kHz; 01: 192kHz; 10: 384kHz; 11: 768kHz。	0x0
[21]	-	reserved	保留。	0x0
[20]	RW	BYP_CHOP_DAC_VREF	DAC_VREF 的 Chopper 功能控制 0: 打开 Chopper; 1: 关闭 Chopper。	0x0
[19]	RW	ANA_LOOP	模拟环回使能控制 0: 正常工作模式(经数字 codec); 1: 模拟 ADC 到模拟 DAC 直接模拟环回。	0x0
[18]	RW	DACL_POP_EN	左通路 POP 电路使能控制 0: 去 pop 音功能关闭; 1: 开启去 POP 音功能; 注意: DACR_POP_EN 仅在上下电抑制 POP 流程的过程中按照要求置 1, 而在 DAC 通路正常工作的过程中 DACR_POP_EN 需要置 0。	0x1
[17]	-	reserved	保留。	0x0
[16]	RW	DACR_POP_EN	右通路 POP 电路使能控制 0: 去 pop 音功能关闭; 1: 开启去 POP 音功能;	0x1



Bits	Access	Name	Description	Reset
			注意：DACR_POP_EN 仅在上下电抑制 POP 流程的过程中按照要求置 1，而在 DAC 通路正常工作的过程中 DACR_POP_EN 需要置 0。	
[15]	-	reserved	保留。	0x0
[14]	RW	MUTE_DACR	右通路 DAC 静音控制 0: 正常工作; 1: 下电。	0x1
[13]	RW	MUTE_DACL	左通路 DAC 静音控制 0: 正常工作; 1: 下电。	0x1
[12]	RW	PD_DACR	右通路 DAC 上下电控制 0: 正常工作; 1: 下电。	0x1
[11]	RW	PD_DACL	左通路 DAC 上下电控制 0: 正常工作; 1: 下电。	0x1
[10]	RW	reserved	保留	0x0
[9]	RW	PA_DACR_DFF	右通路 DAC 触发器上下电控制 0: 正常工作; 1: 下电。	0x0
[8]	RW	PA_DACL_DFF	左通路 DAC 触发器上下电控制 0: 正常工作; 1: 下电。	0x0
[7]	RW	PD_MICBIAS2	MICBIAS2 上下电控制 0: 正常工作; 1: 下电。	0x1
[6]	RW	PD_IBIAS	ADC 的 RC 系数校正电路上下电控制 0: 正常工作; 1: 下电。	0x1
[5]	RW	PD_MICBIAS1	MICBIAS1 上下电控制 0: 正常工作; 1: 下电。	0x1
[4:2]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[1]	RW	PDB_CTCM_IBIAS	CTCM_BUFFER 的偏置电流开关控制 0: 无外部偏置电流; 1: 正常偏置。	0x0
[0]	RW	PD_VREF	VREF 上下电控制 0: 正常工作; 1: 下电。	0x1

AUDIO_ANA_CTRL_1

AUDIO_ANA_CTRL_1 为 AudioCodec 模拟 4 到 7 号寄存器。

Offset Address: 0x0018 Total Reset Value: 0x1102_0004

Bits	Access	Name	Description	Reset
[31:28]	RW	ADC_DAC_RL_MIS_SELN	ADC 中反馈 DAC 负载电阻选择控制信号 (负端) 0000: 负载电阻增大 15%; 0001: 负载电阻增大 14%; 0010: 负载电阻增大 13%; 0011: 负载电阻增大 12%; 0100: 负载电阻增大 11%; 0101: 负载电阻增大 10%; 0110: 负载电阻增大 9%; 0111: 负载电阻增大 8%; 1000: 负载电阻增大 7%; 1001: 负载电阻增大 6%; 1010: 负载电阻增大 5%; 1011: 负载电阻增大 4%; 1100: 负载电阻增大 3%; 1101: 负载电阻增大 2%; 1110: 负载电阻增大 1%; 1111: 负载电阻不变。	0x1
[27:24]	RW	ADC_DAC_RL_MIS_SELP	ADC 中反馈 DAC 负载电阻选择控制信号 (正端) 0000: 负载电阻增大 15%; 0001: 负载电阻增大 14%; 0010: 负载电阻增大 13%;	0x1



Bits	Access	Name	Description	Reset
			0011: 负载电阻增大 12%; 0100: 负载电阻增大 11%; 0101: 负载电阻增大 10%; 0110: 负载电阻增大 9%; 0111: 负载电阻增大 8%; 1000: 负载电阻增大 7%; 1001: 负载电阻增大 6%; 1010: 负载电阻增大 5%; 1011: 负载电阻增大 4%; 1100: 负载电阻增大 3%; 1101: 负载电阻增大 2%; 1110: 负载电阻增大 1%; 1111: 负载电阻不变。	
[23]	RW	LPGA_BOOST	左通路 ADC boost 使能控制信号 0: boost 不使能; 1: boost 使能。	0x0
[22]	RW	RPGA_BOOST	右通路 ADC boost 使能控制信号 0: boost 不使能; 1: boost 使能。	0x0
[21:20]	RW	ADC_DAC_IB_SEL	ADC 中的反馈 DAC 电流调节 00: 参考电阻 $R=90\% \cdot R_f$; 01: 参考电阻 $R=95\% \cdot R_f$; 10: 参考电阻 $R=100\% \cdot R_f$; 11: 参考电阻 $R=105\% \cdot R_f$ 。	0x0
[19:18]	RW	reserved	保留	0x0
[17]	RW	RX_CTCM_PD	上行通路 CTCM_BUFFER 的上下电控制 0: 正常工作 1: 下电	0x1
[16]	-	reserved	保留。	0x0
[15]	-	reserved	保留。	0x0
[14]	RW	PGA_RIN_MAN_EN	手动控制 LINEIN 模块未选通通路输入电阻悬空或接共模的使能信号 0: 手动控制模式关闭 1: 手动控制模块开启	0x0



Bits	Access	Name	Description	Reset
[13:8]	RW	PGA_RIN_MAN_F LOAT	当 PGA_RIN_MAN_EN 为高时该寄存器有效，分别控制 LINEIN 模块 6 条通路的输入电阻悬空或接共模 0: 接共模； 1: 悬空。	0x00
[7]	RW	CTRL_MCLK_PH	IP 的工作时钟沿选择 0: IP 使用主时钟的下降沿； 1: IP 使用主时钟的上升沿。	0x0
[6]	RW	CTRL_CLK_ADCD AC_PH	ADC 的反馈 DAC 工作 CLK 沿选择 0: 反馈 DAC 的 CLK 由主时钟上升沿同步； 1: 反馈 DAC 的 CLK 由主时钟下降沿同步。	0x0
[5:4]	-	reserved	保留。	0x0
[3:2]	RW	IBADJ_ADC	ADC 的偏置电流调节 00: 3uA； 01: 5uA(推荐值)； 10: 7uA； 11: 10uA。	0x1
[1]	RW	SEL_CLK_CHOP_C TCM	CTCM BUFFER 的 chopper 时钟频率选择 0: 96kHz； 1: 192kHz。	0x0
[0]	RW	BYP_CHOP_CTCM	CTCM BUFFER 的 Chopper 功能控制 0: 打开 Chopper； 1: 关闭 Chopper。	0x0

AUDIO_ANA_CTRL_2

AUDIO_ANA_CTRL_2 为 AudioCodec 模拟 8 到 11 号寄存器。

Offset Address: 0x001c Total Reset Value: 0x40C0_0013

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30]	RW	PD_CTCM	下行 CTCM 上下电控制信号 0: 下行 CTCM 正常工作；	0x1



Bits	Access	Name	Description	Reset
			1: 下行 CTCM 下电。	
[29:27]	RW	MICBIAS2_ADJ	MICBIAS2 的输出电压选择 000: 2.1V; 001: 2.2V; 010: 2.3V; 011: 2.4V; 110: 2.7V; 111: 2.8V。	0x0
[26:24]	RW	MICBIAS1_ADJ	MICBIAS1 的输出电压选择 000: 2.1V; 001: 2.2V; 010: 2.3V; 011: 2.4V; 110: 2.7V; 111: 2.8V。	0x0
[23]	RW	RST	CODEC 的模拟全局复位信号 0: 正常工作; 1: 复位。	0x1
[22:18]	RW	VREF_SEL	VREF 启动时间控制 00000:0.38s; 00100:1.543s; 11111:10.7s。	0x10
[17]	RW	VREF_FS	VREF 上电速度控制信号 0: VREF 正常上电; 1: VREF 快速上电。	0x0
[16]	-	reserved	保留。	0x0
[15]	RW	BYP_CHOPPER_LI NEIN	输入 PGA chopper 使能控制信号 0: 使能; 1: 不使能。	0x0
[14]	RW	BYP_CHOPPER_A DC	ADC 中积分器运放 chopper 使能控制信号 0: 使能;	0x0



Bits	Access	Name	Description	Reset
			1: 不使能。	
[13]	RW	BYP_CHOP_CTCM_RX	上行 CTCM chopper 使能控制信号 0: 使能; 1: 不使能。	0x0
[12:8]	RW	reserved	保留	0x00
[7]	RW	ADCL_MODE_SEL	ADCL 模式选择信号 0: 6.144MHz 采样率; 1: 4.096MHz 采样率。	0x0
[6]	RW	ADCR_MODE_SEL	ADCR 模式选择信号 0: 6.144MHz 采样率; 1: 4.096MHz 采样率。	0x0
[5:4]	RW	ADC_VREF	ADC 模块 flashADC 参考电压选择 00: 0.5333V; 01: 0.5V; 10: 0.47V; 11: 0.4444V。	0x1
[3:2]	RW	LDO_SEL	LDO 模块输出电压选择信号 00: 1.0V; 01: 0.95V; 10: 0.9V(推荐值); 11: 0.85V。	0x0
[1]	RW	LDO_EN_BK	LDO 模块备份参考电压产生电路(直接电源电压分压)使能信号 0: 不使能; 1: 使能。	0x1
[0]	RW	LDO_PD	LDO 上下电控制信号 0: LDO 模块正常工作; 1: LDO 模块下电。	0x1

AUDIO_ANA_CTRL_3

AUDIO_ANA_CTRL_3 为 AudioCodec 模拟 12 到 15 号寄存器。

Offset Address: 0x0020 Total Reset Value: 0x1111_FE00



Bits	Access	Name	Description	Reset
[31:29]	RW	RPGA_SEL_09	右通路输入 PGA 通路选择信号 000: MIC 差分输入; 001: MIC_R 单端输入; 010: MIC_L 单端输入; 011: LINEIN_R 单端输入; 100: LINEIN_L 单端输入; 101: LINEIN 差分输入。	0x0
[28:24]	RW	RPGA_GAIN_09	右通路输入 PGA 增益选择信号 00000: 0dB; 00001: 2dB; 00010: 4dB; 00011: 6dB; 00100: 8dB; 00101: 10dB; 00110: 12dB; 00111: 14dB; 01000: 16dB; 01001: 18dB; 01010: 20dB; 01011: 22dB; 01100: 24dB; 01101: 26dB; 01110: 28dB; 01111: 30dB; 10000: -1.6dB。	0x11
[23:21]	RW	LPGA_SEL_09	左通路输入 PGA 通路选择信号 000: MIC 差分输入; 001: MIC_L 单端输入; 010: MIC_R 单端输入; 011: LINEIN_R 单端输入; 100: LINEIN_L 单端输入; 101: LINEIN 差分输入。	0x0
[20:16]	RW	LPGA_GAIN_09	左通路输入 PGA 增益选择信号 00000: 0dB; 00001: 2dB;	0x11



Bits	Access	Name	Description	Reset
			00010: 4dB; 00011: 6dB; 00100: 8dB; 00101: 10dB; 00110: 12dB; 00111: 14dB; 01000: 16dB; 01001: 18dB; 01010: 20dB; 01011: 22dB; 01100: 24dB; 01101: 26dB; 01110: 28dB; 01111: 30dB; 10000: -1.6dB。	
[15]	RW	ADCL_PD_09	左通路 ADC 上下电控制信号 0: 左通路 ADC 正常工作; 1: 左通路 ADC 下电。	0x1
[14]	RW	ADCR_PD_09	右通路 ADC 上下电控制信号 0: 右通路 ADC 正常工作; 1: 右通路 ADC 下电。	0x1
[13]	RW	LPGA_PD_09	左通路输入 PGA 上下电控制信号 0: 左通路输入 PGA 正常工作; 1: 左通路输入 PGA 下电。	0x1
[12]	RW	RPGA_PD_09	右通路输入 PGA 上下电控制信号 0: 右通路输入 PGA 正常工作; 1: 右通路输入 PGA 下电。	0x1
[11]	RW	PD_ADC_TUNE_09	ADC RC tune 模块上下电控制信号 0: RC tune 模块正常工作; 1: RC tune 模块下电。	0x1
[10]	RW	LPGA_MUTE_09	左通路输入 PGA 的 MUTE 控制信号 0: 左通路输入 PGA 正常工作; 1: 左通路输入 PGA mute。	0x1
[9]	RW	RPGA_MUTE_09	右通路输入 PGA 的 MUTE 控制信号	0x1



Bits	Access	Name	Description	Reset
			0: 右通路输入 PGA 正常工作; 1: 右通路输入 PGA mute。	
[8]	-	reserved	保留。	0x0
[7]	RW	VREF_EXMODE	当 PD_VREF 为 1 时, 该位配置生效 0: 有下拉, 正常 PD; 1: 无下拉, 可外供 VREF。	0x0
[6:5]	RW	POP_RAMPCLK_SEL	popramp 电路工作时钟。 00: 48kHz; 0: 24kHz; 10: 12kHz; 11: 6kHz。	0x0
[4:3]	RW	POP_RES_SEL	popfree 电路与主输出间间隔电阻。 00: 100; 01: 1K; 10: 10K; 11: 100K。	0x0
[2]	RW	DACL_POP_DIRECTION	左通路 DAC POP 电路爬坡方向。 0: POP 电路下拉 GND; 1: POP 电路上拉 VCM。	0x0
[1]	RW	reserved	保留。	0x0
[0]	RW	DACR_POP_DIRECTION	右通路 DAC POP 电路爬坡方向。 0: POP 电路下拉 GND; 1: POP 电路上拉 VCM。	0x0

AUDIO_ANA_CTRL_4

AUDIO_ANA_CTRL_4 为 AudioCodec 模拟 16 到 19 号寄存器。

Offset Address: 0x0024 Total Reset Value: 0x0000_105C

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:12]	RW	ADC_TUNE_09	ADC 中积分电容控制码, 码值与 R 和 C 的 corner 相关 0101: R 和 C 均为 tt corner;	0x1



Bits	Access	Name	Description	Reset
			1100: R 和 C 均为 ff corner; 0000: R 和 C 均为 ss corner。	
[11]	RW	ADC_TUNE_EN_09	ADC 的 RC 系数校正电路工作使能信号 0: RC 校正模块不使能; 1: RC 校正模块开始工作。	0x0
[10]	RW	ADC_TUNE_SEL_09	ADC 的 RC 系数校正码值来源控制 0: 选择从寄存器中输入; 1: 选择 RC 系数校正电路的输出作为输入源。	0x0
[9]	RW	CTRL_CLK_ADC_PH_09	ADC 的工作时钟沿选择(主时钟二分频) 0: 上升沿工作; 1: 下降沿工作。	0x0
[8]	-	reserved	保留。	0x0
[7:6]	RW	PGA_CHOP_CLK_SEL_09	LINEIN 模块 chop 时钟频率选择信号 00: 48kHz; 01: 96kHz; 10: 192kHz; 11: 384kHz。	0x1
[5:4]	RW	ADC_CHOP_CLK_SEL_09	ADC 模块 chop 时钟频率选择信号 00: 48kHz; 01: 96kHz; 10: 192kHz; 11: 384kHz。	0x1
[3]	RW	ADC_CHOP_CLK_PHSEL_09	ADC 模块 chop 时钟相位选择信号 0: ADC 工作时钟同向相位; 1: ADC 工作时钟反向相位。	0x1
[2]	RW	ADC_THE_CLK_PHSEL_09	ADC 模块温度计码输出相位选择信号 0: ADC 工作时钟同向相位; 1: ADC 工作时钟反向相位。	0x1
[1]	RW	ADC_BIN_CLK_PHSEL_09	ADC 模块二进制码输出相位选择信号 0: ADC 工作时钟同向相位; 1: ADC 工作时钟反向相位。	0x0
[0]	RW	ADC_DWA_BPS_09	ADC 的 DWA 模块使能控制信号 0: DWA 使能;	0x0



Bits	Access	Name	Description	Reset
			1: DWA 不使能。	

AUDIO_CTRL_REG_1

AUDIO_CTRL_REG_1 为 Audio Codec 控制寄存器 1。

Offset Address: 0x0030 Total Reset Value: 0x00F3_5A4A

Bits	Access	Name	Description	Reset
[31]	RW	dac1_rst_n	DACL 复位信号。 0: 复位有效; 1: 复位无效。	0x0
[30]	RW	dacr_rst_n	DACR 复位信号。 0: 复位有效; 1: 复位无效。	0x0
[29]	RW	adcl_rst_n	ADCL 复位信号。 0: 复位有效; 1: 复位无效。	0x0
[28]	RW	adcr_rst_n	ADCR 复位信号。 0: 复位有效; 1: 复位无效。	0x0
[27]	RW	dac1_en	DACL 使能信号。 0: 不使能; 1: 使能。	0x0
[26]	RW	dacr_en	DACR 使能信号。 0: 不使能; 1: 使能。	0x0
[25]	RW	adcl_en	ADCL 使能信号。 0: 不使能; 1: 使能。	0x0
[24]	RW	adcr_en	ADCR 使能信号。 0: 不使能; 1: 使能。	0x0
[23:22]	RW	i2s1_data_bits	I2S 通道 1 数据接口宽度。 00: 16bit;	0x3



Bits	Access	Name	Description	Reset
			01: 18bit; 10: 20bit; 11: 24bit。	
[21:20]	RW	i2s2_data_bits	I2S 通道 2 数据接口宽度。 00: 16bit; 01: 18bit; 10: 20bit; 11: 24bit。	0x3
[19]	RW	dig_bypass	数字部分旁路控制，用于模拟测试模式。 0: 数字部分正常工作; 1: 数字部分被旁路。	0x0
[18]	RW	dig_loop	数字环回控制信号。 0: 数字环回无效; 1: 数字环回有效。	0x0
[17:13]	RW	i2s1_fs_sel	I2S 通道采样率选择。 11000: mclk/512/2, 即 MCLK 为 1024 FS 时选择; 11001: mclk/256/2, 即 MCLK 为 512 FS 时选择; 11010: mclk/128/2, 即 MCLK 为 256 FS 时选择; 11011: mclk/64/2, 即 MCLK 为 128 FS 时选择; 111xx: mclk/32/2, 即 MCLK 为 64 FS 时选择。	0x1A
[12:8]	RW	i2s2_fs_sel	I2S 通道 2 采样率选择。 11000: mclk/512/2, 即 MCLK 为 1024 FS 时选择; 11001: mclk/256/2, 即 MCLK 为 512 FS 时选择; 11010: mclk/128/2, 即 MCLK 为 256 FS 时选择; 11011: mclk/64/2, 即 MCLK 为 128 FS 时选择; 111xx: mclk/32/2, 即 MCLK 为 64 FS 时选择。	0x1A
[7:0]	-	reserved	保留。	0x4A



AUDIO_DAC_REG_0

AUDIO_DAC_REG_0 为 Audio Codec 控制寄存器 2。

Offset Address: 0x0034 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	axi0_aline32_en	axi0 总线写操作 32BYTE 对齐使能控制。 0: 不进行对齐操作; 1: 进行 32byte 对齐操作。	0x0

AUDIO_DAC_REG_1

AUDIO_DAC_REG_1 为 Audio Codec 控制寄存器 3。

Offset Address: 0x0038 Total Reset Value: 0x0606_2424

Bits	Access	Name	Description	Reset
[31]	RW	dac1_mute	DACL 数字静音控制。 0: 正常工作; 1: 静音。	0x0
[30:24]	RW	dac1_vol	DACL 数字音量控制, 不建议使用, 建议使用 AIAO 内的数字音量调节功能进行数字音量输出控制。 计算公式为: $(6-dac1\_vol)db$ 。 当 dac1_vol 为 0x7F 时, DACL 数字静音。 0x00: 6dB; 0x01: 5dB; 0x02: 4dB; ... 0x7E: -120dB; 0x7F: mute。	0x06
[23]	RW	dacr_mute	DACR 数字静音控制。 0: 正常工作; 1: 静音。	0x0
[22:16]	RW	dacr_vol	DACR 数字音量控制, 不建议使用, 建议使用 AIAO 内的数字音量调节功能进行数	0x06



Bits	Access	Name	Description	Reset
			字音量输出控制。 计算公式为: $(6 - \text{dacr\_vol} * 1) \text{db}$ 。 当 dacr_vol 为 0x7F 时, DACR 数字静音。 0x00: 6dB; 0x01: 5dB; 0x02: 4dB; ... 0x7E: -120dB; 0x7F: mute。	
[15]	RW	dacr2dacl_en	DACR2DACL mixer 控制信号。 0: 关闭; 1: 打开。	0x0
[14:8]	RW	dacr2dacl_vol	DACR to DACL 音量控制位 计算公式为: $(36 - \text{dacr2dacl\_vol}) \text{db}$ 。 00: 36dB; 01: 35dB; 02: 34dB; ... 7E: -90dB; 7F: -91dB。	0x24
[7]	RW	dacl2dacr_en	DACL to DACR mixer 控制信号。 0: 关闭; 1: 打开。	0x0
[6:0]	RW	dacl2dacr_vol	DACL to DACR 音量控制位 计算公式为: $(36 - \text{dacl2dacr\_vol}) \text{db}$ 。 00: 36dB; 01: 35dB; 02: 34dB; ... 7E: -90dB; 7F: -91dB。	0x24



AUDIO_ADC_REG_0

AUDIO_ADC_REG_0 为 Audio Codec 控制寄存器 4。

Offset Address: 0x003c Total Reset Value: 0x1E1E_0001

Bits	Access	Name	Description	Reset
[31]	RW	adcl_mute	ADCL 数字静音控制位。 0: ADCL unmute; 1: ADCL mute。	0x0
[30:24]	RW	adcl_vol	ADCL 音量控制位，不建议使用，建议使用 CODEC ADC 模拟音量调节实现输入音量控制。 计算公式: (30-adcl_vol) db。 00: 30dB; 01: 29dB; 02: 28dB; ... 7E: -96dB; 7F: -97dB。	0x1E
[23]	RW	adcr_mute	ADCR 数字静音控制位。 0: ADCR 不静音; 1: ADCR 静音。	0x0
[22:16]	RW	adcr_vol	ADCR 音量控制位，不建议使用，建议使用 CODEC ADC 模拟音量调节实现输入音量控制。 计算公式: (30-adcr_vol) db。 00: 36dB; 01: 35dB; 02: 34dB; ... 7E: -90dB; 7F: -91dB。	0x1E
[15]	RW	adcl_hpf_en	ADCL 高通滤波器使能控制。 0: 关闭高通滤波器; 1: 使能高通滤波器。	0x0
[14]	RW	adcr_hpf_en	ADCR 高通滤波器使能控制。 0: 关闭高通滤波器; 1: 使能高通滤波器。	0x0



Bits	Access	Name	Description	Reset
[13:4]	RW	reserved	保留。	0x000
[3]	RW	adcl_i2ssel	ADCL I2S 接口选择(高优先级)。 0: 选择 I2S1; 1: 选择 I2S2。	0x0
[2]	RW	adcl_lrsl	ADCL 左右声道数据选择。 0: 选择左声道; 1: 选择右声道。	0x0
[1]	RW	adcr_i2ssel	ADCR I2S 接口选择(低优先级)。 0: 选择 I2S1; 1: 选择 I2S2。	0x0
[0]	RW	adcr_lrsl	ADCR 左右声道数据选择。 0: 选择左声道; 1: 选择右声道。	0x1

ACODEC_REG18

ACODEC_REG18 为 I2S 通路选择控制寄存器(不可软复位)。

Offset Address: 0x0048 Total Reset Value: 0x0000_0008

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	audio_rx_bclk_sel	audiocodec rx bclk 和 ws 源选择: 0: 选择 CRG0 的时钟。 1: 选择 CRG8 时钟。推荐此方式, 在此方式下, audio tx 和 rx 共时钟, AIAO 可以数字环回。	0x0
[3]	RW	i2s_pad_mclk_sel	芯片 i2s 接口的 mclk 选择控制。 0: aiao_i2s_rx_mclk; 1: aiao_i2s_tx_mclk。	0x1
[2]	RW	audio_mclk_sel	芯片内部 audiocodec 的 mclk 选择控制。 0: aiao_i2s_rx_mclk; 1: aiao_i2s_tx_mclk。	0x0
[1]	RW	i2s_pad_enable	芯片 i2s 接口与 aiao 通道的使能控制。 0: 关闭芯片 i2s 接口与 aiao 的通道, 芯片	0x0



Bits	Access	Name	Description	Reset
			i2s 接口没有输出； 1: 打开芯片 i2s 接口与 aiao 的通道，芯片 i2s 接口与 aiao 的 i2s 相接。 注意：i2s_pad_enable 和 audio_enable 只能同时配置一个为 1，不能同时都配置为 1。	
[0]	RW	audio_enable	芯片内部 audicodec 的 i2s 接口与 aiao 通道的使能控制。 0: 关闭芯片内部 audicodec 的 i2s 接口与 aiao 的通道，芯片内部 audicodec 的 i2s 接口没有输出； 1: 打开芯片内部 audicodec 的 i2s 接口与 aiao 的通道，芯片内部 audicodec 的 i2s 接口与 aiao 的 i2s 相接。 注意：i2s_pad_enable 和 audio_enable 只能同时配置一个为 1，不能同时都配置为 1。	0x0



目 录

12 外围设备.....	12-1
12.1 I ² C	12-1
12.1.1 概述	12-1
12.1.2 功能描述.....	12-1
12.1.3 功能框图.....	12-2
12.1.4 时序描述原理.....	12-2
12.1.5 工作方式.....	12-10
12.1.6 I ² C 寄存器概览	12-14
12.1.7 I ² C 寄存器描述	12-15
12.2 UART	12-27
12.2.1 概述	12-27
12.2.2 特点	12-28
12.2.3 功能描述.....	12-28
12.2.4 工作方式.....	12-30
12.2.5 UART 寄存器概览	12-32
12.2.6 UART 寄存器描述	12-33
12.3 SPI.....	12-44
12.3.1 概述	12-44
12.3.2 特点	12-44
12.3.3 功能描述.....	12-45
12.3.4 三种外设总线时序.....	12-45
12.3.5 工作方式.....	12-52
12.3.6 寄存器概览.....	12-55
12.3.7 寄存器描述.....	12-56
12.4 3WIRE SPI.....	12-63
12.4.1 概述	12-63
12.4.2 工作方式.....	12-63
12.4.3 spi_3wire 寄存器概览	12-64
12.4.4 spi_3wire 寄存器描述	12-64
12.5 eMMC/SD/SDIO 控制器.....	12-65



12.5.1 功能描述.....	12-65
12.5.2 应用说明.....	12-72
12.5.3 寄存器概览.....	12-85
12.5.4 寄存器描述.....	12-86
12.6 红外接口.....	12-108
12.6.1 概述	12-108
12.6.2 特点	12-108
12.6.3 功能描述.....	12-109
12.6.4 工作方式.....	12-116
12.6.5 IR 寄存器概览.....	12-118
12.6.6 IR 寄存器描述.....	12-119
12.7 GPIO	12-133
12.7.1 概述	12-133
12.7.2 特点	12-134
12.7.3 工作方式.....	12-134
12.7.4 GPIO 寄存器概览	12-135
12.7.5 GPIO 寄存器描述	12-136
12.8 USB DRD	12-139
12.8.1 概述	12-139
12.8.2 功能描述.....	12-140
12.8.3 工作方式.....	12-141
12.8.4 寄存器概览.....	12-142
12.8.5 USB 寄存器描述.....	12-144
12.9 LSADC.....	12-174
12.9.1 概述	12-174
12.9.2 特点	12-174
12.9.3 工作方式.....	12-175
12.9.4 LSADC 寄存器概览.....	12-177
12.9.5 LSADC 寄存器描述.....	12-177
12.10 PWM	12-182
12.10.1 概述	12-182
12.10.2 特点	12-182
12.10.3 工作方式.....	12-182
12.10.4 PWM 寄存器概览	12-183
12.10.5 PWM 寄存器描述	12-184



插图目录

图 12-1 控制器的功能框图.....	12-2
图 12-2 7bit 寻址，写操作时序图	12-4
图 12-3 7bit 寻址，直接读时序图	12-5
图 12-4 10bit 寻址，写操作时序图	12-6
图 12-5 10bit 寻址，组合读时序图	12-8
图 12-6 非标准时序图	12-10
图 12-7 UART 的典型应用框图.....	12-29
图 12-8 UART 帧格式.....	12-29
图 12-9 SPI 接 Slave 时的应用	12-45
图 12-10 SPI 单帧帧格式（SPO=0、SPH=0）	12-46
图 12-11 SPI 连续帧帧格式（SPO=0、SPH=0）	12-46
图 12-12 SPI 单帧帧格式（SPO=0、SPH=1）	12-47
图 12-13 SPI 连续帧帧格式（SPO=0、SPH=1）	12-47
图 12-14 SPI 单帧帧格式（SPO=1、SPH=0）	12-48
图 12-15 SPI 连续帧帧格式（SPO=1、SPH=0）	12-48
图 12-16 SPI 单帧帧格式（SPO=1、SPH=1）	12-49
图 12-17 SPI 连续帧帧格式（SPO=1、SPH=1）	12-49
图 12-18 SPI 接口时序图.....	12-50
图 12-19 TI 同步串行单帧帧格式.....	12-50
图 12-20 TI 同步串行连续帧帧格式	12-51
图 12-21 National Semiconductor Microwire 单帧帧格式	12-51
图 12-22 National Semiconductor Microwire 连续帧帧格式	12-52
图 12-23 MMC 功能框图.....	12-67
图 12-24 MMC 典型应用图.....	12-68
图 12-25 MMC 指令格式.....	12-68



图 12-26 MMC 指令响应格式.....	12-69
图 12-27 MMC 非数据指令操作.....	12-69
图 12-28 单块与多块读操作.....	12-70
图 12-29 单块与多块写操作.....	12-71
图 12-30 1bit 数据线传输模式下的块数据格式.....	12-71
图 12-31 4bit 数据线传输模式下的块数据格式.....	12-72
图 12-32 双 buffer 结构示意图.....	12-79
图 12-33 链结构示意图.....	12-79
图 12-34 32bit 位宽的描述子的结构	12-79
图 12-35 发送单个 NEC with simple repeat code 码的帧格式.....	12-111
图 12-36 持续按键连续发送 NEC with simple repeat code 码的帧格式.....	12-111
图 12-37 NEC with simple repeat code 码 bit0 和 bit1 定义	12-112
图 12-38 NEC with simple repeat code 码单发代码格式.....	12-112
图 12-39 NEC with simple repeat code 码连发代码格式.....	12-112
图 12-40 发送单个 NEC with full repeat code 码的帧格式.....	12-112
图 12-41 持续按键连续发送 NEC with full repeat code 码的帧格式.....	12-113
图 12-42 NEC with full repeat code 码 bit0 和 bit1 定义	12-113
图 12-43 NEC with full repeat code 码单发代码格式.....	12-113
图 12-44 发送单个 TC9012 码的帧格式.....	12-114
图 12-45 持续按键连续发送 TC9012 码的帧格式.....	12-114
图 12-46 TC9012 码 bit0 和 bit1 定义	12-114
图 12-47 TC9012 码单发代码格式.....	12-115
图 12-48 TC9012 码连发代码格式 (C0=1)	12-115
图 12-49 TC9012 码连发代码格式 (C0=0)	12-115
图 12-50 发送单个 SONY 帧格式.....	12-115
图 12-51 持续按键连续发送 SONY 码帧格式.....	12-116
图 12-52 bit0 和 bit1 定义	12-116
图 12-53 IR 模块初始化操作流程.....	12-117
图 12-54 读取解码数据的操作流程.....	12-118
图 12-55 USB 2.0 DRD 逻辑框图	12-140
图 12-56 单次扫描处理流程.....	12-175
图 12-57 连续扫描模式下通道轮询扫描示意图.....	12-176



图 12-58 连续扫描处理流程.....12-176



表格目录

表 12-1 时序命令	12-2
表 12-2 7bit 寻址，写操作对应的时序描述配置.....	12-4
表 12-3 7bit 寻址，直接读操作的时序描述配置.....	12-5
表 12-4 10bit 寻址，写操作对应的时序描述配置.....	12-6
表 12-5 10bit 寻址，组合读对应的时序描述配置.....	12-8
表 12-6 非标准时序对应时序描述配置.....	12-10
表 12-7 I ² C 寄存器偏移地址变量表	12-14
表 12-8 I ² C 寄存器概览	12-14
表 12-9 UART 寄存器概览	12-32
表 12-10 SPI 接口时序参数.....	12-50
表 12-11 SPI 寄存器概览.....	12-55
表 12-12 spi_3wire 寄存器概览（基地址：0x120C_3000）	12-64
表 12-13 芯片中 3 个 MMC 控制器对应的功能信号和管脚名	12-66
表 12-14 非数据传输指令时的寄存器 MMC_CMD 配置参考（默认值）	12-75
表 12-15 单块或多块读数据时的寄存器 MMC_CMD 配置参考（默认值）	12-76
表 12-16 单块或多块写数据时的寄存器 MMC_CMD 配置参考（默认值）	12-77
表 12-17 DES0 各 bit 的含义.....	12-80
表 12-18 DES1 各 bit 的含义.....	12-80
表 12-19 DES2 各 bit 的含义.....	12-81
表 12-20 DES3 各 bit 的含义.....	12-81
表 12-21 Resume 操作时的寄存器 MMC_CMDARG 配置参考.....	12-84
表 12-22 MMC 寄存器概览（EMMC 基址是 0x1010_0000，SDIO0 基址是 0x100F_0000，SDIO1 基址是 0x1002_0000）	12-85
表 12-23 红外接收数据码型的统计表（NEC with simple repeat code）	12-109
表 12-24 红外接收数据码型的统计表（NEC with full repeat code）	12-110



表 12-25 红外接收数据码型的统计表（TC9012 和 SONY 码）	12-110
表 12-26 IR 寄存器概览（基址是 0x120F_0000）	12-119
表 12-27 芯片 GPIO 寄存器对应的基地址	12-135
表 12-28 GPIO 寄存器概览	12-136
表 12-29 USB 寄存器概览（USB2.0 基址是 0x100E_0000）	12-142
表 12-30 LSADC 寄存器概览（基址：0x120E_0000）	12-177
表 12-31 PWM _n 的寄存器基地址表	12-183
表 12-32 PWM0 寄存器概览	12-183



12 外围设备

12.1 I²C

12.1.1 概述

I²C(Inter-Integrated Circuit)控制器为 Master 接口，完成 CPU 对 I²C 总线上连接的从设备的读写访问。

芯片共有 8 个 I²C 控制器。

12.1.2 功能描述

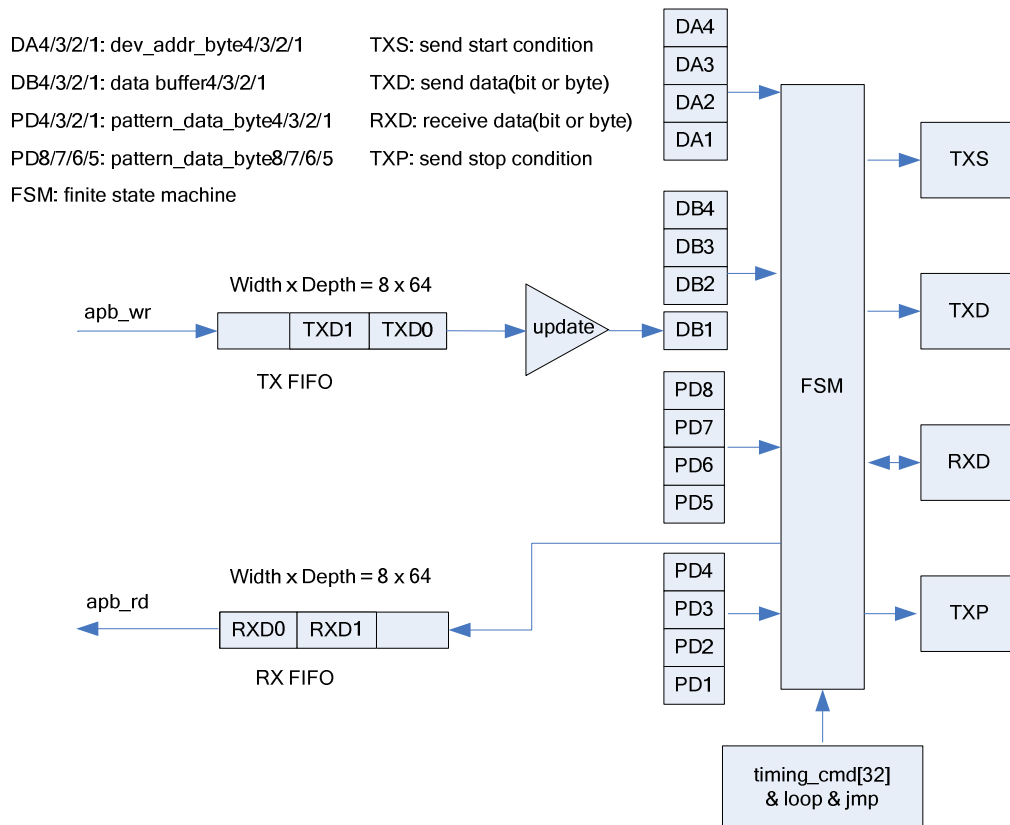
I²C 控制器具有以下功能特点：

- 芯片的 I²C 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。



12.1.3 功能框图

图12-1 控制器的功能框图



12.1.4 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 12-1 所示。

表12-1 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



时序命令编号	时序命令代号	时序命令含义
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 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。
0x13	RACK	接收低电平应答
0x14	RNACK	接收高电平非应答
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



时序命令编号	时序命令代号	时序命令含义
0x1B	UNDEF	未定义
0x1C	UNDEF	未定义
0x1D	UDB1	从 TX FIFO 更新数据到 data_buf_byte1 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 I²C 总线的状态。
0x1E	SR	总线 repeated START
0x1F	P	总线 STOP

标准时序----7bit 寻址，写操作

图12-2 7bit 寻址，写操作时序图

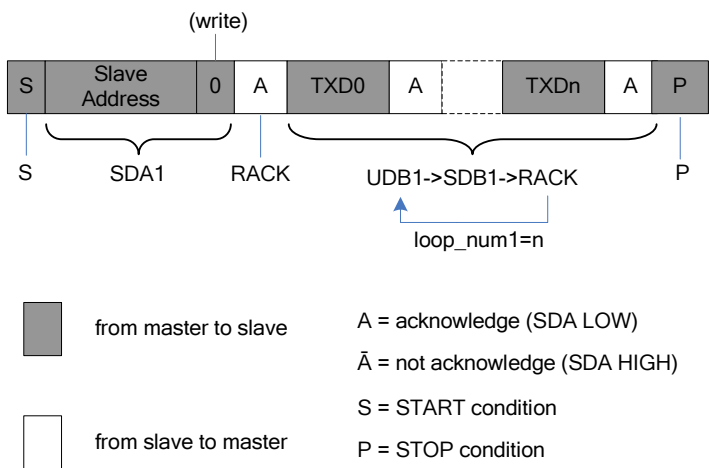


图 12-2 对应的时序描述配置如表 12-2 所示。

表12-2 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



类型	寄存器	配置
	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
时序数据准备	I2C_DEV_ADDR	(Slave Address+ ‘0’) 共 8bit
	I2C_TX_FIFO	TXD0、TXD1、…TXDn 依次写入。

标准时序----7bit 寻址，直接读操作

图12-3 7bit 寻址，直接读时序图

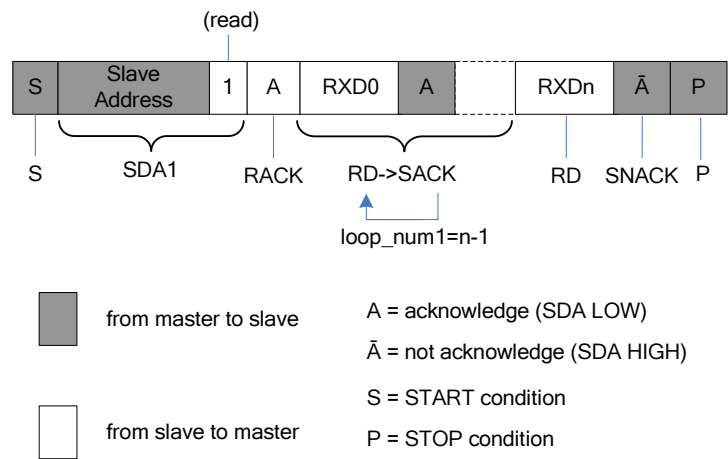


图 12-3 对应的时序描述配置如表 12-3 所示。

表12-3 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



类型	寄存器	配置
	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12-4 10bit 寻址，写操作时序图

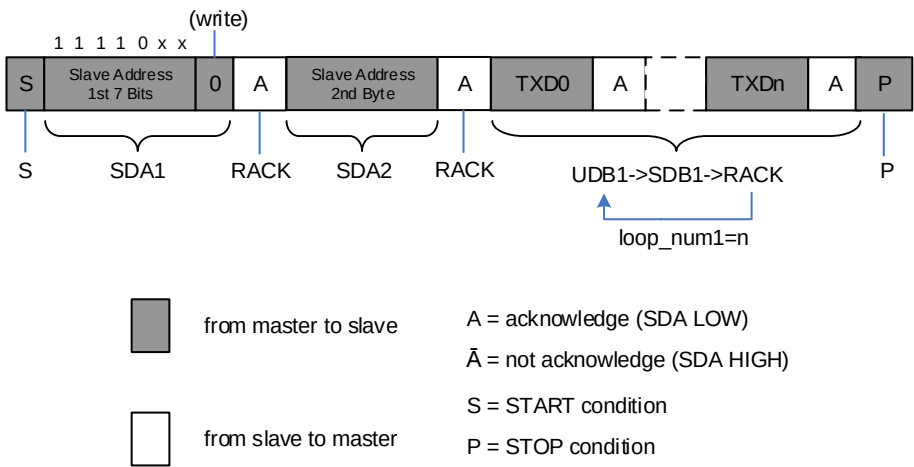


图 12-5 对应的时序描述配置如表 12-5 所示。

表12-4 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



类型	寄存器	配置
	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
	I2C_DST1	5
时序数据准备	I2C_DEV_ADDR	(Slave Address 1st 7Bits + ‘0’) 共 8bit 写入 dev_addr_byte1; (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12-5 10bit 寻址，组合读时序图

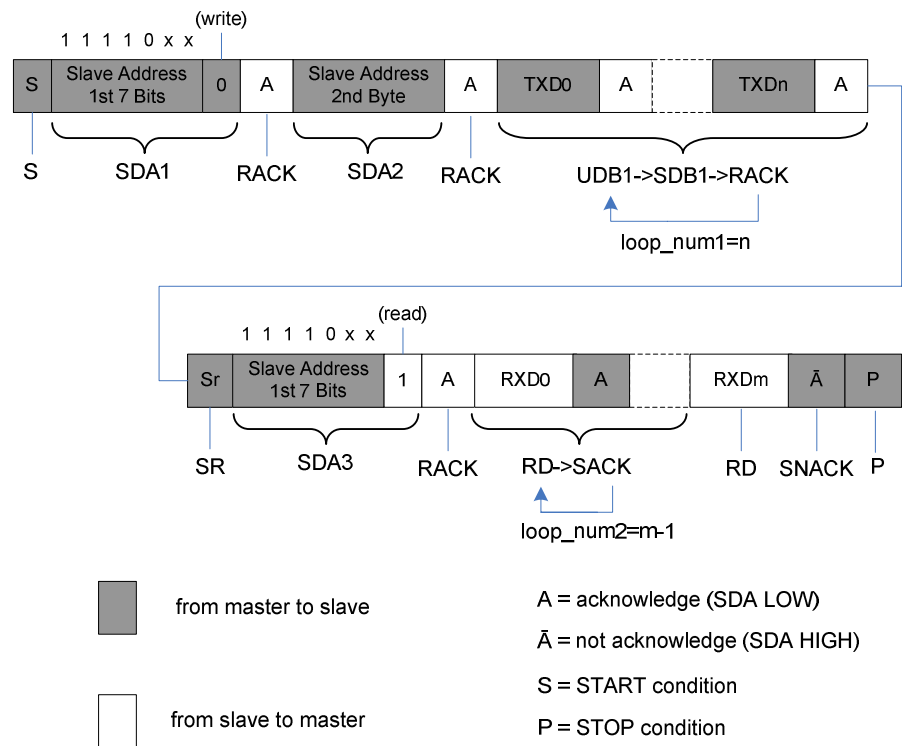


图 12-5 对应的时序描述配置如表 12-5 所示。

表12-5 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; (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 12-6 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12-6 非标准时序图

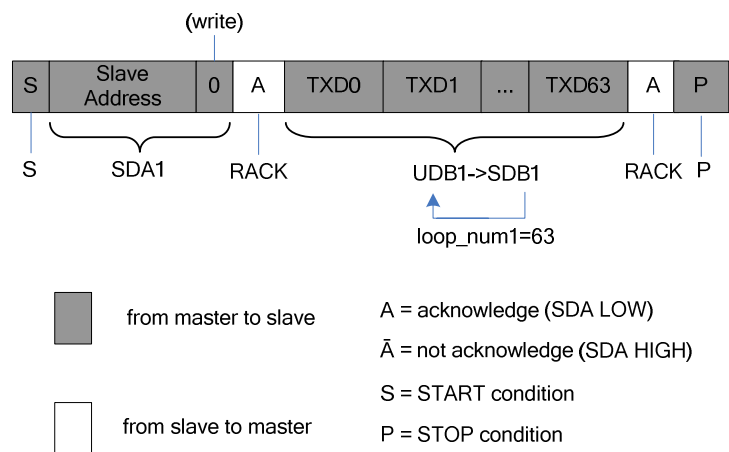


图 12-6 对应的时序描述配置如表 12-6 所示。

表12-6 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
	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。

12.1.5 工作方式

12.1.5.1 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 [12.1.4 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 0x811。
- 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_lt_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 0x10，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。
 - 如果已经没有待写数据，则配置 [I2C_INTR_EN](#) 为 0x1801，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 0x1fff，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

12.1.5.2 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 [12.1.4 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 0x1fff，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

12.1.5.3 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 12.1.4 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 0x1805。
- 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 0x4，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 0x1fff，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

12.1.5.4 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 12.1.4 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 0x1fff，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

12.1.5.5 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 [12.1.4 时序描述原理](#) 完成时序配置。
- 步骤 7. 配置 `I2C_CTRL1` 为 0x201，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 0x1fff，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

12.1.5.6 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 [12.1.4 时序描述原理](#) 完成时序配置。
- 步骤 7. 配置 `I2C_CTRL1` 为 0x301，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 0x1fff，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

12.1.5.7 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 0x1fff，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



12.1.6 I²C 寄存器概览

I²C 基地址如下：

- I²C0 寄存器基地址为 0x120B_0000。
- I²C1 寄存器基地址为 0x120B_1000。
- I²C2 寄存器基地址为 0x120B_2000。
- I²C3 寄存器基地址为 0x120B_3000。
- I²C4 寄存器基地址为 0x120B_4000。
- I²C5 寄存器基地址为 0x120B_5000。
- I²C6 寄存器基地址为 0x120B_6000。
- I²C7 寄存器基地址为 0x120B_7000。

I²C 寄存器偏移地址中变量的取值范围和含义如表 12-7 所示。

表12-7 I²C 寄存器偏移地址变量表

变量名称	取值范围	描述
n	0~31	I ² C 时序命令，最多支持 32 个时序命令

表12-8 I²C 寄存器概览

偏移地址	名称	描述	页码
0x0000	I2C_GLB	I ² C 全局配置寄存器	12-15
0x0004	I2C_HCNT	I ² C 高电平时长寄存器	12-16
0x0008	I2C_LCNT	I ² C 低电平时长寄存器	12-16
0x0010	I2C_DEV_ADDR	I ² C 器件地址寄存器	12-17
0x0014	I2C_DATA_BUF	I ² C 数据 buf 寄存器	12-17
0x0018	I2C_PATTERN_DATA1	I ² C PATTERN 数据 1 寄存器	12-17
0x001C	I2C_PATTERN_DATA2	I ² C PATTERN 数据 2 寄存器	12-18
0x0020	I2C_TX_FIFO	I ² C TX FIFO 数据寄存器	12-18
0x0024	I2C_RX_FIFO	I ² C RX FIFO 数据寄存器	12-18
0x0030+n x 4	I2C_TIMING_CMD	I ² C 时序命令寄存器	12-19
0x00B0	I2C_LOOP1	I ² C 循环次数 1 寄存器	12-20
0x00B4	I2C_DST1	I ² C 跳转目的 1 寄存器	12-20
0x00B8	I2C_LOOP2	I ² C 循环次数 2 寄存器	12-21



偏移地址	名称	描述	页码
0x00BC	I2C_DST2	I ² C 跳转目的 2 寄存器	12-21
0x00C0	I2C_LOOP3	I ² C 循环次数 3 寄存器	12-21
0x00C4	I2C_DST3	I ² C 跳转目的 3 寄存器	12-21
0x00C8	I2C_TX_WATERMARK	I ² C TX FIFO 水线寄存器	12-21
0x00CC	I2C_RX_WATERMARK	I ² C RX FIFO 水线寄存器	12-22
0x00D0	I2C_CTRL1	I ² C 控制寄存器 1	12-22
0x00D4	I2C_CTRL2	I ² C 控制寄存器 2	12-23
0x00D8	I2C_FIFO_STAT	I ² C FIFO 状态寄存器	12-24
0x00E0	I2C_INTR_RAW	I ² C 原始中断寄存器	12-24
0x00E4	I2C_INTR_EN	I ² C 中断使能寄存器	12-25
0x00E8	I2C_INTR_STAT	I ² C 中断状态寄存器	12-26

12.1.7 I²C 寄存器描述

I2C_GLB

I2C_GLB 为 I²C 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
[23:8]	RW	sda_hold_duration	SDA hold time 时长。 参考时钟 PCLK 为 50MHz。配置为 0xa，不允许配置其他值。 注意：只能控制从主机发往从设备的数据的 hold time 时长。	0x0000
[7:1]	RO	reserved	保留。	0x00
[0]	RW	i2c_enable	I ² C 使能控制。 0：禁止；此时会清空 TX FIFO 和 RX FIFO。 1：使能。 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。	0x0



I2C_HCNT

I2C_HCNT 为 I2C 高电平时长寄存器。

Offset Address: 0x0004 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。 参考时钟在 CRG 寄存器 PERI_CRG105 内配，可选 50MHz 或 100MHz。 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。 标准模式下： $i2c\_high\_duration = (F_{PCLK} / F_{SCL}) * 0.5$ ； 快速模式下： $i2c\_high\_duartion = (F_{PCLK} / F_{SCL}) * 0.36$ 。(F_{SCL} 为 SCL 总线的频率) 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: 0x0008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	保留。	0x0000
[15:0]	RW	i2c_low_duration	SCL 低电平时长。 参考时钟在 CRG 寄存器 PERI_CRG105 内配置，可选 50MHz 或 100MHz。 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。 标准模式下： $i2c\_low\_duartion = (F_{PCLK} / F_{SCL}) * 0.5$ ； 快速模式下：	0x0000



Bits	Access	Name	Description	Reset
			$i2c\_low\_duration = (F_{PCLK} / F_{SCL}) * 0.64$ 。 (F_{SCL} 为 SCL 总线的频率) 以参考时钟为 50MHz 且工作在快速模式下 ($F_{SCL} = 400KHz$)为例, $i2c\_scl\_lcnt = (50MHz / 400KHz) \times 0.64 = 80$ 。	

I2C_DEV_ADDR

I2C_DEV_ADDR 为 I²C 器件地址寄存器。

Offset Address: 0x0010 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 I²C 数据 buf 寄存器。

Offset Address: 0x0014 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_PATTERN_DATA1

I2C_PATTERN_DATA1 为 I²C PATTERN 数据 1 寄存器。

Offset Address: 0x0018 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 I²C PATTERN 数据 2 寄存器。

Offset Address: 0x001C 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

I2C_TX_FIFO

I2C_TX_FIFO 为 I²C TX FIFO 数据寄存器。

Offset Address: 0x0020 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。 写入的数据是待发送出去的数据。	0x00

I2C_RX_FIFO

I2C_RX_FIFO 为 I²C RX FIFO 数据寄存器。

Offset Address: 0x0024 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。 读出的数据是 I2C 总线上收到的数据。	0x00

I2C_TIMING_CMD

I2C_TIMING_CMD 为 I²C 时序命令寄存器。

Offset Address: 0x0030+n x 4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	RO	reserved	保留。	0x00000000
[4:0]	RW	timing_cmd	时序命令。n = 0, 1, 2, ... 31. 0x00: EXIT(结束, 用于逻辑退出); 0x01: S (总线 START); 0x02: WDA4 (发送 dev_addr_byte4); 0x03: WDA3 (发送 dev_addr_byte3); 0x04: WDA2 (发送 dev_addr_byte2); 0x05: WDA1 (发送 dev_addr_byte1); 0x06: WDB4 (发送 data_buf_byte4); 0x07: WDB3 (发送 data_buf_byte3); 0x08: WDB2 (发送 data_buf_byte2); 0x09: WDB1 (发送 data_buf_byte1); 0x0A: WPD8 (发送 pattern_data_byte8); 0x0B: WPD7 (发送 pattern_data_byte7); 0x0C: WPD6 (发送 pattern_data_byte6); 0x0D: WPD5 (发送 pattern_data_byte5); 0x0E: WPD4 (发送 pattern_data_byte4); 0x0F: WPD3 (发送 pattern_data_byte3); 0x10: WPD2 (发送 pattern_data_byte2); 0x11: WPD1 (发送 pattern_data_byte1); 0x12: RD (接收 1 字节数据); 0x13: RACK (接收低电平应答); 0x14: RNACK (接收高电平非应答); 0x15: RNC (接收应答, 高低电平都无所谓); 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	0x00



			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); 0x1E: SR (总线 repeated START); 0x1F: P(总线 STOP)。 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 I2C 总线的状态。 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。	

I2C_LOOP1

I2C_LOOP1 为 I²C 循环次数 1 寄存器。

Offset Address: 0x00B0 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 I²C 跳转目的 1 寄存器。

Offset Address: 0x00B4 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 I²C 循环次数 2 寄存器。

Offset Address: 0x00B8 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 I²C 跳转目的 2 寄存器。

Offset Address: 0x00BC 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 I²C 循环次数 3 寄存器。

Offset Address: 0x00C0 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 I²C 跳转目的 3 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_cmd3	指定跳转到哪个时序命令寄存器。	0x00

I2C_TX_WATERMARK

I2C_TX_WATERMARK 为 I²C TX FIFO 水线寄存器。



Offset Address: 0x00C8 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。当 TX FIFO 里的数据数目比 tx_watermark 小时，tx_lt_watermark_raw 原始中断会被置 1。	0x00

I2C_RX_WATERMARK

I2C_RX_WATERMARK 为 I²C RX FIFO 水线寄存器。

Offset Address: 0x00CC 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，rx_gt_watermark_raw 原始中断会被置 1。	0x00

I2C_CTRL1

I2C_CTRL1 为 I²C 控制寄存器 1。

Offset Address: 0x00D0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	RO	reserved	保留。	0x000000
[9:8]	RW	dma_operation	DMA 操作控制。 00: 非 DMA 方式; 01: 保留; 10: DMA 写方式; 11: DMA 读方式。	0x0
[7:1]	RO	reserved	保留。	0x00
[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。 0: 不启动; 1: 启动。	0x0



I2C_CTRL2

I2C_CTRL2 为 I²C 控制寄存器 2。

Offset Address: 0x00D4 Total Reset Value: 0x11110011

Bits	Access	Name	Description	Reset
[31:29]	RO	reserved	保留。	0x0
[28]	RO	i2c_scl_oen	监测内部 I2C 的 SDA 输出电平。 0: SDA 输出低电平; 1: SDA 输出高电平。	0x1
[27:25]	RO	reserved	保留。	0x0
[24]	RO	i2c_sda_oen	监测内部 I2C 的 SCL 输出电平。 0: SCL 输出低电平; 1: SCL 输出高电平。	0x1
[23:21]	RO	reserved	保留。	0x0
[20]	RO	i2c_scl_in	监测外部 I2C 总线 SCL 的电平。 0: 总线上 SCL 为低电平; 1: 总线上 SCL 为高电平。	0x1
[19:17]	RO	reserved	保留。	0x0
[16]	RO	i2c_sda_in	监测外部 I2C 总线 SDA 的电平。 0: 总线上 SDA 为低电平; 1: 总线上 SDA 为高电平。	0x1
[15:9]	RO	reserved	保留。	0x00
[8]	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。 0: 禁止; 1: 使能。	0x0
[7:5]	RO	reserved	保留。	0x0
[4]	RW	force_scl_oen	在 gpio_mode 有效时控制 SCL 管脚电平。 0: SCL 管脚为低电平; 1: SCL 管脚为高电平。	0x1
[3:1]	RO	reserved	保留。	0x0
[0]	RW	force_sda_oen	在 gpio_mode 有效时控制 SDA 管脚电平。	0x1



Bits	Access	Name	Description	Reset
			0: SDA 管脚为低电平; 1: SDA 管脚为高电平。	

I2C_FIFO_STAT

I2C_FIFO_STAT 为 I²C FIFO 状态寄存器。

Offset Address: 0x00D8 Total Reset Value: 0x000A_0000

Bits	Access	Name	Description	Reset
[31:20]	RO	reserved	保留	0x000
[19]	RO	tx_fifo_not_full	TX FIFO 非满指示。 0: 满; 1: 非满。	0x1
[18]	RO	tx_fifo_not_empty	TX FIFO 非空指示。 0: 空; 1: 非空。	0x0
[17]	RO	rx_fifo_not_full	RX FIFO 非满指示。 0: 满; 1: 非满。	0x1
[16]	RO	rx_fifo_not_empty	RX FIFO 非空指示。 0: 空; 1: 非空。	0x0
[15]	RO	reserved	保留。	0x0
[14:8]	RO	tx_fifo_vld_num	TX FIFO 里的有效数目。	0x00
[7]	RO	reserved	保留。	0x0
[6:0]	RO	rx_fifo_vld_num	RX FIFO 里的有效数目。	0x00

I2C_INTR_RAW

I2C_INTR_RAW 为 I²C 原始中断寄存器。

Offset Address: 0x00E0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:13]	RO	reserved	保留。	0x00000



Bits	Access	Name	Description	Reset
[12]	RWC	all_cmd_done_raw	时序命令序列执行到 EXIT 命令时产生此中断。 0: 无中断; 1: 有中断。	0x0
[11]	RWC	arb_lost_raw	仲裁丢失。 0: 无中断; 1: 有中断。	0x0
[10]	RWC	start_det_raw	检查到 START。 0: 无中断; 1: 有中断。	0x0
[9]	RWC	stop_det_raw	检查到 STOP。 0: 无中断; 1: 有中断。	0x0
[8:5]	RO	reserved	保留。	0x0
[4]	RWC	tx_lt_watermark_raw	发送数据时, TX FIFO 里的数据数目低于 水线。 0: 无中断; 1: 有中断。	0x0
[3]	RO	reserved	保留。	0x0
[2]	RWC	rx_gt_watermark_raw	接收数据时, RX FIFO 里的数据数目高于 水线。 0: 无中断; 1: 有中断。	0x0
[1]	RO	reserved	保留。	0x0
[0]	RWC	ack_bit_unmatch_raw	应答位不符合预期。 0: 无中断; 1: 有中断。	0x0

I2C_INTR_EN

I2C_INTR_EN 为 I²C 中断使能寄存器。

Offset Address: 0x00E4 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:13]	RO	reserved	保留。	0x00000
[12]	RW	all_cmd_done_en	all_cmd_done 中断使能。 0: 不使能; 1: 使能。	0x0
[11]	RW	arb_lost_en	arb_lost 中断使能。 0: 不使能; 1: 使能。	0x0
[10]	RW	start_det_en	start_det 中断使能。 0: 不使能; 1: 使能。	0x0
[9]	RW	stop_det_en	STOP 中断使能。 0: 不使能; 1: 使能。	0x0
[8:5]	RO	reserved	保留。	0x0
[4]	RW	tx_lt_watermark_en	tx_lt_watermark 中断使能。 0: 不使能; 1: 使能。	0x0
[3]	RO	reserved	保留。	0x0
[2]	RW	rx_gt_watermark_en	rx_gt_watermark 中断使能。 0: 不使能; 1: 使能。	0x0
[1]	RO	reserved	保留。	0x0
[0]	RW	ack_bit_unmatch_en	ack_bit_unmatch 中断使能。 0: 不使能; 1: 使能。	0x0

I2C_INTR_STAT

I2C_INTR_STAT 为 I²C 中断状态寄存器。

Offset Address: 0x00E8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:13]	RO	reserved	保留。	0x00000



Bits	Access	Name	Description	Reset
[12]	RO	all_cmd_done	所有时序命令正常完成。 0: 无中断; 1: 有中断。	0x0
[11]	RO	arb_lost	仲裁丢失。 0: 无中断; 1: 有中断。	0x0
[10]	RO	start_det	检测到 START。 0: 无中断; 1: 有中断。	0x0
[9]	RO	stop_det	检测到 STOP。 0: 无中断; 1: 有中断。	0x0
[8:5]	RO	reserved	保留。	0x0
[4]	RO	tx_lt_watermark	发送数据时, TX FIFO 里的数据数目低于 水线。 0: 无中断; 1: 有中断。	0x0
[3]	RO	reserved	保留。	0x0
[2]	RO	rx_gt_watermark	接收数据时, RX FIFO 里的数据数目高于 水线。 0: 无中断; 1: 有中断。	0x0
[1]	RO	reserved	保留。	0x0
[0]	RO	ack_bit_unmatch	应答位不符合预期。 0: 无中断; 1: 有中断。	0x0

12.2 UART

12.2.1 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 5 个 UART 控制器：

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



注意

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。具体的配置寄存器请参考 Hi35xxVxxx_PINOUT_CN.xlsx（Hi35xxVxxx 表示具体项目名称）。

例如：当使用系统的 UART0 时，需要配置 iocfg_reg55 的 bit[8] 为高。

12.2.2 特点

UART 模块有以下特点：

- 支持 64 x 8bit 的发送 FIFO 和 64 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。

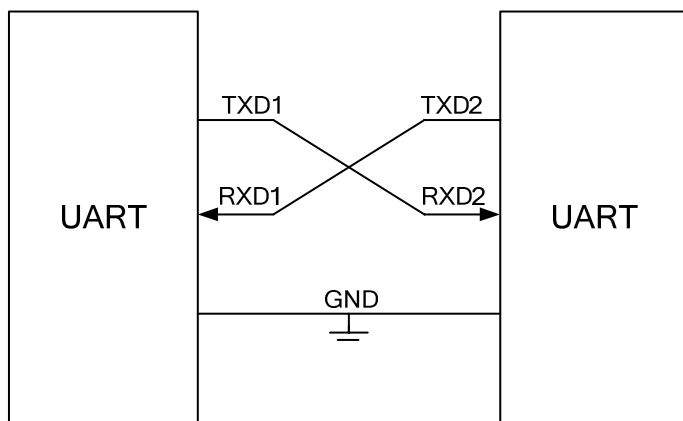
12.2.3 功能描述

应用框图

UART 的典型应用框图如图 12-7 所示。



图12-7 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 12-8 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12-8 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。

12.2.4 工作方式

12.2.4.1 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 6bit `UART_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。

12.2.4.2 软复位



说明

复位寄存器请参考 3.2.6 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。

12.2.4.3 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。

12.2.4.4 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_DMAR` 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_DMAR` 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

12.2.5 UART 寄存器概览

地址空间内：

- UART0 寄存器基地址为：0x120A_0000。
- UART1 寄存器基地址为：0x120A_1000。
- UART2 寄存器基地址为：0x120A_2000。
- UART3 寄存器基地址为：0x120A_3000。
- UART4 寄存器基地址为：0x120A_4000

UART 寄存器概览如表 12-9 所示。

表12-9 UART 寄存器概览

偏移地址	名称	描述	页码
0x000	UART_DR	数据寄存器	12-33
0x004	UART_RSR	接收状态寄存器/错误清除寄存器	12-34
0x018	UART_FR	标志寄存器	12-35
0x024	UART_IBRD	整数波特率寄存器	12-36
0x028	UART_FBRD	小数波特率寄存器	12-36



偏移地址	名称	描述	页码
0x02C	UART_LCR_H	线控寄存器	12-36
0x030	UART_CR	控制寄存器	12-38
0x034	UART_IFLS	中断 FIFO 阈值选择寄存器	12-39
0x038	UART_IMSC	中断屏蔽寄存器	12-40
0x03C	UART_RIS	原始中断状态寄存器	12-41
0x040	UART_MIS	屏蔽后中断状态寄存器	12-42
0x044	UART_ICR	中断清除寄存器	12-43
0x048	UART_DMACR	DMA 控制寄存器	12-44

12.2.6 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。 0: 无溢出错误; 1: 有溢出错误, 接收 FIFO 满后接收到了数据。	0x0
[10]	RO	be	Break 错误。 0: 无 break 错误; 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。 0: 无校验错误; 1: 有校验错误。	0x0
[8]	RO	fe	帧错误。 0: 无帧错误; 1: 有帧错误(错误的停止位)。	0x0



Bits	Access	Name	Description	Reset
[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。 0: 无溢出错误; 1: 溢出错误。 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。 0: 无 break 错误; 1: break 错误。 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、parity、stop bit)还要长。	0x0
[1]	RW	pe	校验错误。 0: 无校验错误; 1: 接收数据的校验错误。 FIFO 模式下, 该错误与 FIFO 顶部的数据相关联。	0x0
[0]	RW	fe	帧错误。 0: 无帧错误; 1: 接收到的数据的停止位错误(有效的停	0x0



			止位为 1)。	

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。 如果 UART_LCR_H[fen] 为 0，则当发送 holding register 空时该位置 1； 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。 如果 UART_LCR_H[fen] 为 0，则当接收 holding register 满时该位置 1； 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。 如果 UART_LCR_H[fen] 为 0，则当发送 holding register 满时该位置 1； 如果 UART_LCR_H[fen] 为 1，当发送 FIFO 为满时该位置 1。	0x0
[4]	RO	rxfe	该位的含义由 UART_LCR_H[fen] 的状态决定。 如果 UART_LCR_H[fen] 为 0，则当接收 holding register 空时该 bit 置 1； 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。 0: UART 空闲或者完成发送数据； 1: UART 正忙于发送数据。 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。	0x0



Bits	Access	Name	Description	Reset
			一旦发送 FIFO 非空该位就置位，不管 UART 使能与否。	
[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。



注意

- 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。 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； 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。 当本寄存器的 bit[7]为 0 时，stick parity 禁止。 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
[6:5]	RW	wlen	指示发送和接收一个帧里数据比特的数目。 00: 5bit; 01: 6bit; 10: 7bit; 11: 8bit。	0x0
[4]	RW	fen	发送和接收 FIFO 使能控制。 0: 发送的帧尾有 1bit 停止位； 1: 发送和接收 FIFO 使能。	0x0
[3]	RW	stp2	发送帧尾 2bit 停止位判断。 0: 发送的帧尾没有 2bit 停止位； 1: 发送的帧尾有 2bit 停止位。 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。 0: 在发送和接收过程中生成奇校验或检查奇校验； 1: 在发送和接收过程中生成偶校验或检查偶校验。 当 UART_LCR_H[pen]为 0 时，该位不起作用。	0x0
[1]	RW	pen	校验选择位。 0: 不作校验； 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。	0x0



Bits	Access	Name	Description	Reset
			0: 无效; 1: 在完成当前数据的发送后, UTXD 连续输出低电平。 注意: 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[uarten]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[uarten]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。 0: 不使能 CTS 硬件流控; 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。 0: 不使能 RTS 硬件流控; 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。 该 bit 为 UART modem 状态输出信号 nUARTRTS 的取反。 0: 输出信号不变;	0x0



Bits	Access	Name	Description	Reset
			1: 即该 bit 配置为 1, 则输出信号为 0。	
[10]	RW	dtr	数据发送准备。 该 bit 为 UART modem 状态输出信号 nUARTDTR 的取反。 0: 输出信号不变; 1: 即该 bit 配置为 1, 则输出信号为 0。	0x0
[9]	RW	rx	UART 接收使能。 0: 禁止; 1: 使能。 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。 0: 禁止; 1: 使能。 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。 0: 禁止; 1: UARTTXD 输出环回到 UARTRXD。	0x0
[6:1]	RO	reserved	保留。	0x00
[0]	RW	uarten	UART 使能。 0: 禁止; 1: 使能。 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	rxiflsel	接收中断 FIFO 的阈值选择, 接收中断的触	0x2



			发点如下。 000: 接收 FIFO $\geq 1/8$ full; 001: 接收 FIFO $\geq 1/4$ full; 010: 接收 FIFO $\geq 1/2$ full; 011: 接收 FIFO $\geq 3/4$ full; 100: 接收 FIFO $\geq 7/8$ full; 101~111: 保留。	
[2:0]	RW	txiflssel	发送中断 FIFO 的阈值选择, 发送中断的触发点如下。 000: 发送 FIFO $\leq 1/8$ full; 001: 发送 FIFO $\leq 1/4$ full; 011: 发送 FIFO $\leq 3/4$ full; 010: 发送 FIFO $\leq 1/2$ full; 100: 发送 FIFO $\leq 7/8$ full; 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。 0: 屏蔽该中断; 1: 不屏蔽该中断。	0x0
[9]	RW	beim	break 错误中断的屏蔽状态。 0: 屏蔽该中断; 1: 不屏蔽该中断。	0x0
[8]	RW	peim	校验中断的屏蔽状态。 0: 屏蔽该中断; 1: 不屏蔽该中断。	0x0
[7]	RW	feim	帧错误中断的屏蔽状态。 0: 屏蔽该中断; 1: 不屏蔽该中断。	0x0
[6]	RW	rtim	接收超时中断的屏蔽状态。	0x0



Bits	Access	Name	Description	Reset
			0: 屏蔽该中断; 1: 不屏蔽该中断。	
[5]	RW	txim	发送中断的屏蔽状态。 0: 屏蔽该中断; 1: 不屏蔽该中断。	0x0
[4]	RW	rxim	接收中断的屏蔽状态。 0: 屏蔽该中断; 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
[10]	RO	oeris	原始的溢出错误中断状态。 0: 未产生中断; 1: 已产生中断。	0x0
[9]	RO	beris	原始的 break 错误中断状态。 0: 未产生中断; 1: 已产生中断。	0x0
[8]	RO	peris	原始的校验中断状态。 0: 未产生中断; 1: 已产生中断。	0x0
[7]	RO	feris	原始的错误中断状态。 0: 未产生中断; 1: 已产生中断。	0x0
[6]	RO	rtris	原始的接收超时中断状态。 0: 未产生中断; 1: 已产生中断。	0x0
[5]	RO	txris	原始的发送中断状态。 0: 未产生中断;	0x0



Bits	Access	Name	Description	Reset
			1: 已产生中断。	
[4]	RO	rxris	原始接收中断状态。 0: 未产生中断; 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。 0: 未产生中断; 1: 已产生中断。	0x0
[9]	RO	bemis	屏蔽后的 break 错误中断状态。 0: 未产生中断; 1: 已产生中断。	0x0
[8]	RO	pemis	屏蔽后的校验中断状态。 0: 未产生中断; 1: 已产生中断。	0x0
[7]	RO	femis	屏蔽后的错误中断状态。 0: 未产生中断; 1: 已产生中断。	0x0
[6]	RO	rtmis	屏蔽后的接收超时中断状态。 0: 未产生中断; 1: 已产生中断。	0x0
[5]	RO	txmis	屏蔽后的发送中断状态。 0: 未产生中断; 1: 已产生中断。	0x0
[4]	RO	rxmis	屏蔽后的接收中断状态。 0: 未产生中断;	0x0



Bits	Access	Name	Description	Reset
			1: 已产生中断。	
[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。 0: 无效; 1: 清除中断。	0x0
[9]	WO	beic	清除 break 错误中断。 0: 无效; 1: 清除中断。	0x0
[8]	WO	peic	清除校验中断。 0: 无效; 1: 清除中断。	0x0
[7]	WO	feic	清除错误中断。 0: 无效; 1: 清除中断。	0x0
[6]	WO	rtic	清除接收超时中断。 0: 无效; 1: 清除中断。	0x0
[5]	WO	txic	清除发送中断。 0: 无效; 1: 清除中断。	0x0
[4]	WO	rxic	清除接收中断。 0: 无效; 1: 清除中断。	0x0
[3:0]	RO	reserved	保留。	0x0



UART_DMACR

UART_DMACR 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。 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； 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。 0: 禁止； 1: 使能。	0x0
[0]	RW	rxdmae	接收 FIFO 的 DMA 使能控制。 0: 禁止； 1: 使能。	0x0

12.3 SPI

12.3.1 概述

SPI 控制器实现数据的串并、并串转换，可以作为 Master 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。

12.3.2 特点



注意

- 芯片总共有 3 组 SPI 接口。
- 工作参考时钟为 100M，SPI 输出的 SPI_CLK 最大支持 50MHz。

SPI 的功能特点有：

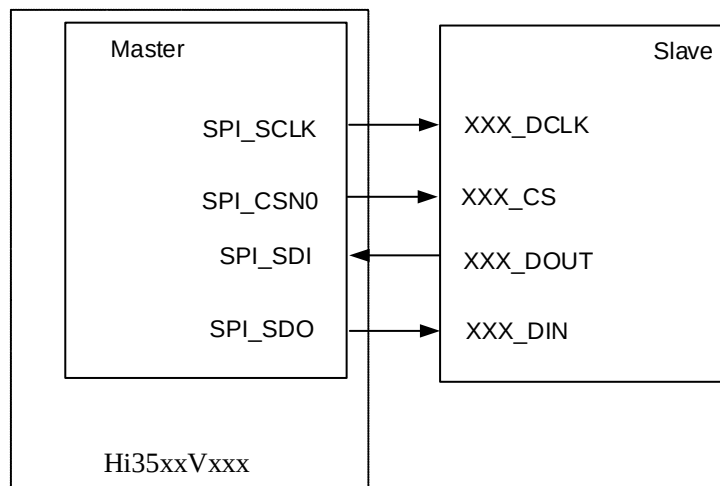
- 接口时钟频率可编程。
- 收/发分开的宽度 16bit、深度为 256 的 FIFO。
- 串行数据帧长度可编程：4bit~16bit。
- 内部提供环回测试模式。
- 支持 DMA 操作。
- 支持 SPI、MicroWire、TI 同步串行三种接口，支持单帧和连续帧格式。
- 支持 SPI 全双工工作模式，时钟极性、相位可配置。
- 支持 MicroWire 半双工工作模式。
- 支持 TI 同步串行接口全双工工作模式。

12.3.3 功能描述

典型应用

SPI 接 Slave 时的应用框图如图 12-9 所示。

图12-9 SPI 接 Slave 时的应用



12.3.4 三种外设总线时序

图 12-10~图 12-17 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



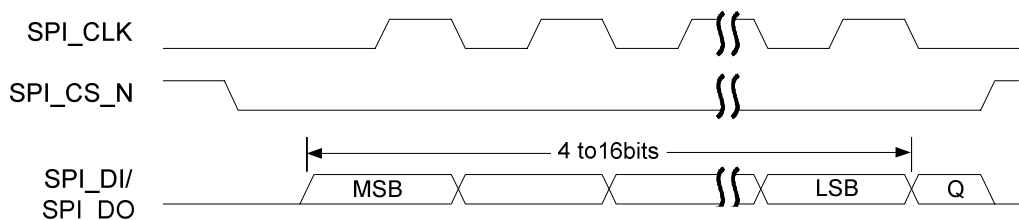
说明

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

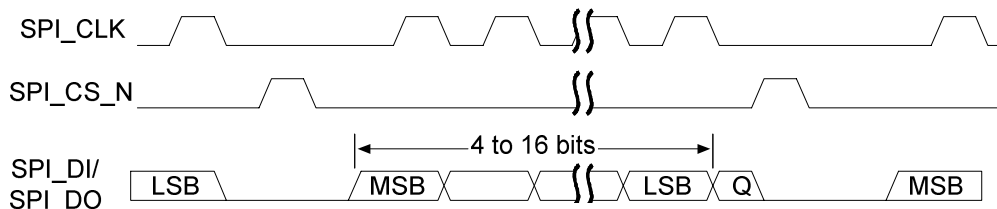
SPI 单帧格式如图 12-10 所示。

图12-10 SPI 单帧格式（SPO=0、SPH=0）



SPI 连续帧格式如图 12-11 所示。

图12-11 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。

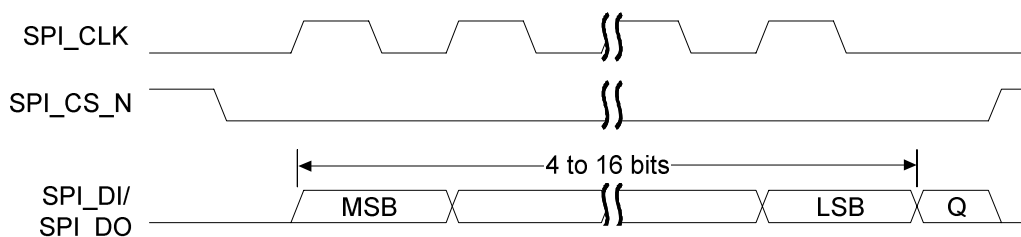


如果是连续的传输，SPI_CS_N 信号在每个 word 传输之间必须将 SPI_CLK 时钟拉高一个时钟周期。这是因为 SPH 为 0 时，Salve 选择管脚会固定其内部串行设备寄存器的数据，使它不会变化。因此在连续传输时，主设备必须在每个 word 传输之间将 SPI_CS_N 信号拉高。连续传输结束时，SPI_CS_N 在捕获到最后 1bit 之后的 1 个 SPI_CLK 时钟周期之后恢复为高电平。

(2) SPO=0、SPH=1

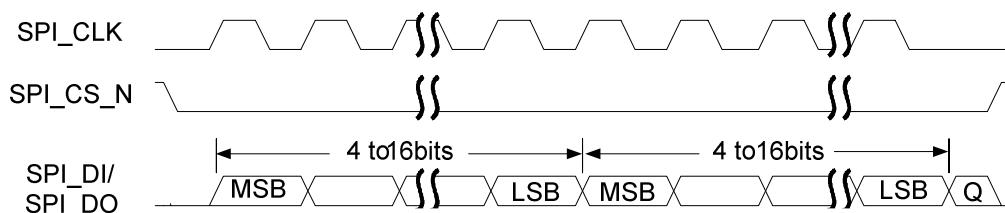
SPI 单帧格式如图 12-12 所示。

图12-12 SPI 单帧格式（SPO=0、SPH=1）



SPI 连续帧格式如图 12-13 所示。

图12-13 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。

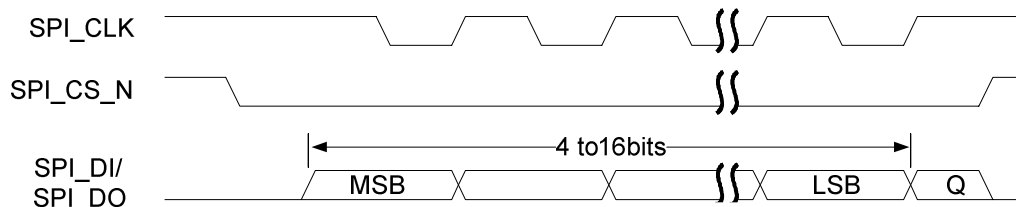
当连续传输时，在传输数据 word 之间 SPI_CS_N 保持为低。连续传输结束时，SP_CS_N 在最后 1bit 捕获之后的 1 个 SPI_CLK 时钟之后恢复为高电平。



(3) SPO=1、SPH=0

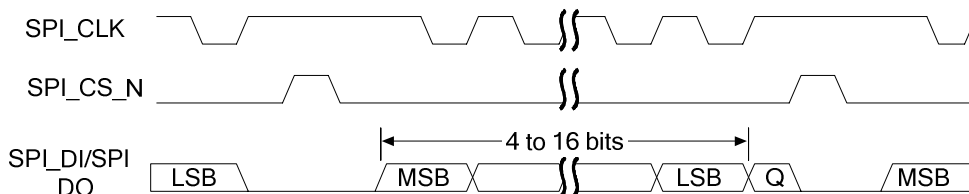
SPI 单帧格式如图 12-14 所示。

图12-14 SPI 单帧格式 (SPO=1、SPH=0)



SPI 连续帧格式如图 12-15 所示。

图12-15 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。

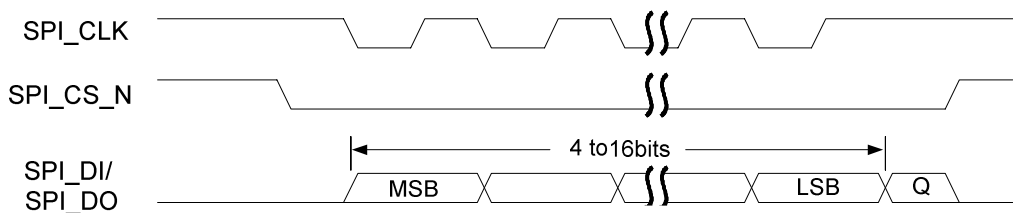
如果是连续的传输，SPI_CS_N 信号在每个 word 传输之间必须拉高。这是因为当 SPH 为 0 时，Slave 选择管脚固定其内部串行设备寄存器的数据，使它不会变化。SPI_CS_N 在捕捉到最后 1bit 数据之后的 1 个 SPI_CLK 时钟周期之后恢复为高电平。

(4) SPO=1、SPH=1

SPI 单帧格式如图 12-16 所示。

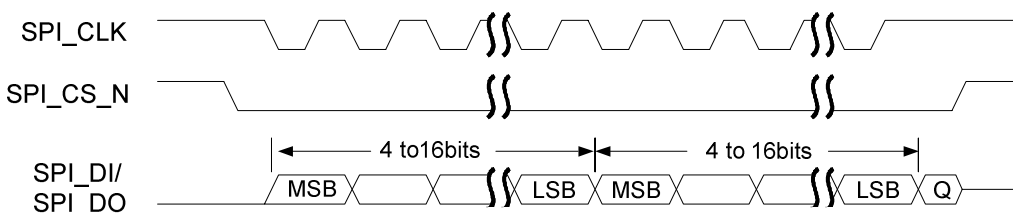


图12-16 SPI 单帧格式 (SPO=1、SPH=1)



SPI 连续帧格式如图 12-17 所示。

图12-17 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12-18 SPI 接口时序图

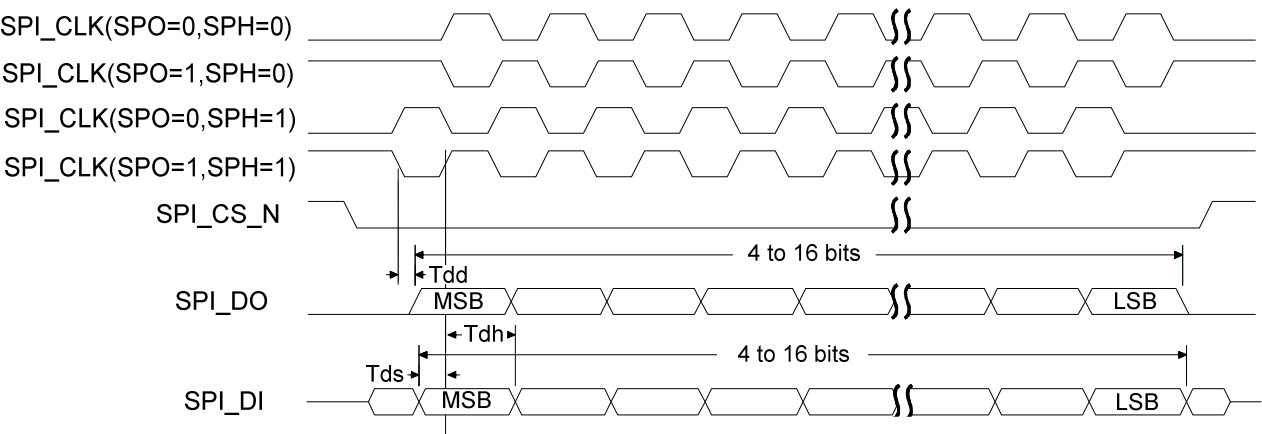


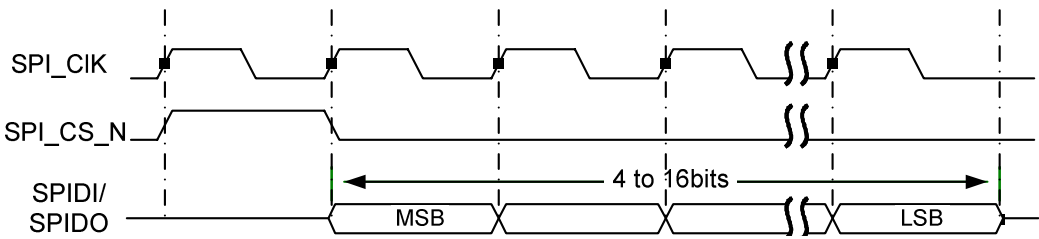
表12-10 SPI 接口时序参数

参数	描述	最小值	最大值	单位
Tdd	输出数据延迟	-3.5	5	ns
Tds	输入控制信号建立时间	23	-	ns
Tdh	输入控制信号保持时间	0	-	ns

TI 同步串行接口

TI 同步串行单帧格式如图 12-19 所示。

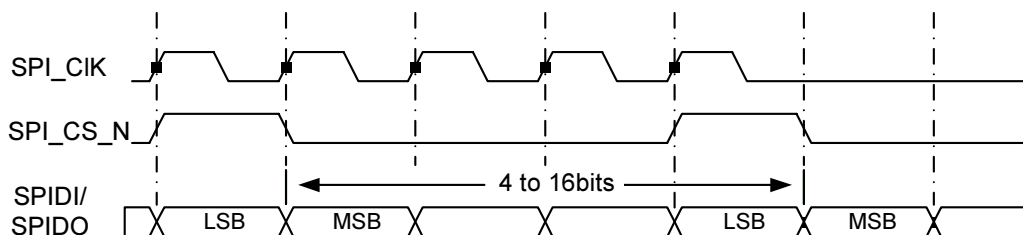
图12-19 TI 同步串行单帧格式



TI 同步串行连续帧格式如图 12-20 所示。



图12-20 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。

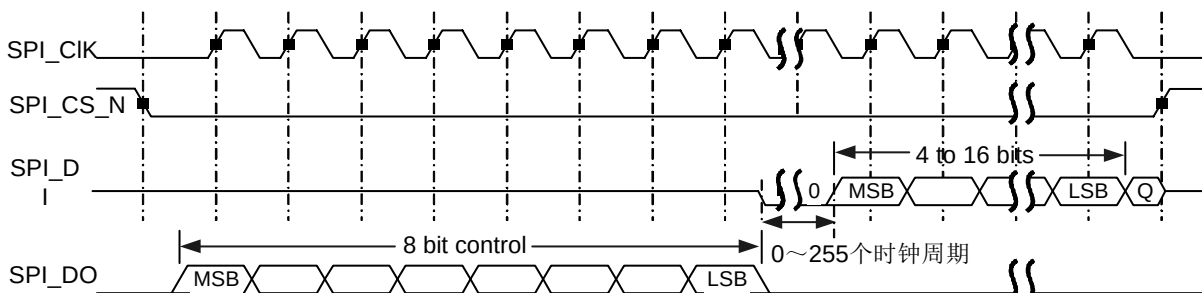
一旦发送 FIFO 有数据，SPICSN 就会产生一个 SPICK 时钟周期的高电平脉冲，将被发送的数据就会从发送 FIFO 传送到发送逻辑串行移位寄存器。在 SPICK 时钟的下一个上升沿，4bit~16bit 数据帧的 MSB 就会从 SPIDO 移位输出。同样，从外部串行 slave 设备接收数据的 MSB 会从 SPIDI 管脚移位输入。

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 单帧帧格式如图 12-21 所示。

图12-21 National Semiconductor Microwire 单帧帧格式

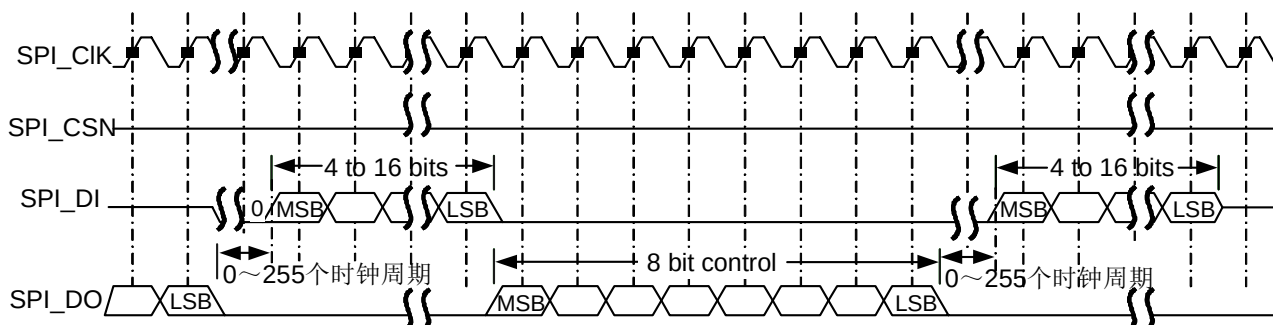


在 SPIDO LSB 结束和 SPIDI MSB 开始之间可以延迟 0~255 个时钟周期。

National Semiconductor Microwire 连续帧帧格式如图 12-22 所示。



图12-22 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。

12.3.5 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：

$$F_{\text{ssclkout}} = F_{\text{ssclk}} / (\text{CPSDVRx} \times (1 + \text{SCR}))$$

Fssclk: SPI 的工作参考时钟 100M。

CPSDVR、SCR 请查询相应寄存器。

支持独立软复位：由寄存器 PERI_CRG111 的 bit[17:15]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[14:12]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 个或更少的有效数据时，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）。



注意

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。



注意

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
- a. 配置该 DMAC 通道的配置寄存器和控制寄存器中的相关参数。
 - b. 启动 DMAC，响应 SPI 发送 FIFO 的 DMA 请求进行数据传输。
 - c. 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
- a. 配置该 DMAC 通道的配置寄存器和控制寄存器中的相关参数。
 - b. 启动 DMAC，响应 SPI 接收 FIFO 的 DMA 请求进行数据传输。
 - c. 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

12.3.6 寄存器概览

SPI 寄存器概览如[表 12-11](#) 所示。

- SPI0 基地址是 0x120C_0000
- SPI1 基地址是 0x120C_1000
- SPI2 基地址是 0x120C_2000

表12-11 SPI 寄存器概览

偏移地址	名称	描述	页码
0x000	SPICR0	控制寄存器 0	12-56
0x004	SPICR1	控制寄存器 1	12-57
0x008	SPIDR	数据寄存器	12-58
0x00C	SPISR	状态寄存器	12-58
0x010	SPICPSR	时钟分频寄存器	12-59
0x014	SPIIMSC	中断屏蔽寄存器	12-59
0x018	SPIRIS	原始中断状态寄存器	12-60
0x01C	SPIMIS	屏蔽后中断状态寄存器	12-60



偏移地址	名称	描述	页码
0x020	SPIICR	中断清除寄存器	12-61
0x024	SPIDMACR	DMA 控制寄存器	12-61
0x028	SPITXFIFO CR	发送 FIFO 控制寄存器	12-61
0x02C	SPIRXFIFO CR	接收 FIFO 控制寄存器	12-62

12.3.7 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 $F_{SSPCLK}/(CPSDVSR \times (1+SCR))$ 。 CPSDVSR 是一个 2~254 之间的偶数，由寄存器 SPICPSR 配置。	0x00
[7]	RW	SPH	SPICLKOUT 相位，具体含义请参见“ 12.3.4 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“ 12.3.4 三种外设总线时序 ”的 SPI 帧格式。	0x0
[5:4]	RW	FRF	帧格式选择。 00: Motorola SPI 帧格式； 01: TI 同步串行帧格式； 10: National Microwire 帧格式； 11: 保留。	0x0
[3:0]	RW	DSS	设置数据位宽。 0011: 4bit； 1000: 9bit； 1101: 14bit； 0100: 5bit； 1001: 10bit；	0x0



			1110: 15bit; 0101: 6bit; 1010: 11bit; 1111: 16bit; 0110: 7bit; 1011: 12bit; 0111: 8bit; 1100: 13bit; 其他: 保留。	

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。 0: 不使能; 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 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。 0: 数据顺序为 MSB 到 LSB; 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。 0: Master 模式（默认）；	0x0



Bits	Access	Name	Description	Reset
			1: 保留。	
[1]	RW	SSE	设置 SPI 使能。 0: 不使能; 1: 使能。	0x0
[0]	RW	LBM	设置环回模式。 0: 正常的串行接口操作使能; 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。 读: 接收 FIFO; 写: 发送 FIFO。 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。 0: 空闲; 1: 忙。	0x0
[3]	RW	RFF	接收 FIFO 是否已满。 0: 未滿; 1: 已滿。	0x0



Bits	Access	Name	Description	Reset
[2]	RW	RNE	接收 FIFO 是否未空。 0: 已空; 1: 未空。	0x0
[1]	RW	TNF	发送 FIFO 是否未滿。 0: 已滿; 1: 未滿。	0x1
[0]	RW	TFE	发送 FIFO 是否已空。 0: 未空; 1: 已空。	0x1

SPICPSR

SPICPSR 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	CPSDVSR	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。 0: 半空或更少情况下中断被屏蔽; 1: 半空或更少情况下中断未被屏蔽。	0x0
[2]	RW	RXIM	接收 FIFO 中断屏蔽。 0: 半空或更少情况下中断被屏蔽; 1: 半空或更少情况下中断未被屏蔽。	0x0



Bits	Access	Name	Description	Reset
[1]	RW	RTIM	接收超时中断。 0: 接收超时中断屏蔽; 1: 接收超时中断不屏蔽。	0x0
[0]	RW	RORIM	接收溢出中断屏蔽。 0: 接收 FIFO 溢出中断屏蔽; 1: 接收 FIFO 溢出中断不屏蔽。 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。 0: 禁止; 1: 使能。	0x0
[0]	WO	RXDMAE	DMA 接收 FIFO 使能寄存器。 0: 禁止; 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。 000: 1; 001: 4; 010: 8;	0x1



Bits	Access	Name	Description	Reset
			011: 16; 100: 32; 101: 64; 110: 128; 111: 128。	
[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。 000: 1; 001: 4; 010: 8; 011: 16; 100: 32; 101: 64; 110: 64; 111: 64。	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。 000: 1; 001: 4; 010: 8; 011: 16; 100: 32; 101: 64; 110: 128;	0x1



Bits	Access	Name	Description	Reset
			111: 192。	
[2:0]	RW	DMARXBRSIZE	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 DMARXBRSIZE 所配置的字数时，DMARXBREQ 有效。 000: 1; 001: 4; 010: 8; 011: 16; 100: 32; 101: 64; 110: 96; 111: 128。	0x1

12.4 3WIRE SPI

12.4.1 概述

芯片提供 1 个 3 线 SPI 接口,用于与对接 MN34120 Sensor。

12.4.2 工作方式

3 线 SPI 内部工作于 APB 总线时钟。初始化步骤如下：

根据需要计算出 SPI 接口时钟,把配置写入寄存器 [SPI_3WIRE_COEF0](#)。

例如：需要 SPI 接口时钟为 1MHz，spi_clk_div 应该配置为(50MHz/1MHz)/2 -1
=24,所以配置 spi_clk_div 为 0x18。

读操作步骤如下：

- 步骤 1. 把读寄存器的地址写入 [SPI_3WIRE_COEF1](#) [spi_add], [SPI_3WIRE_COEF1](#) [spi_rw]写 1。
- 步骤 2. 向 [SPI_3WIRE_COEF2](#) [start]写 1,启动读操作。
- 步骤 3. 查询 [SPI_3WIRE_COEF2](#) [spi_busy],直到该状态位为 0。
- 步骤 4. 从 [SPI_3WIRE_COEF2](#) [spi_rdata]读取需要的数据。

----结束

写操作步骤如下：



- 步骤 1. 把写寄存器的地址和数据写入 [SPI_3WIRE_COEF1 \[spi_add\]](#)和 [SPI_3WIRE_COEF1 \[spi_wdata\]](#), [SPI_3WIRE_COEF1 \[spi_rw\]](#)写 0。
- 步骤 2. 向 [SPI_3WIRE_COEF2 \[start\]](#)写 1,启动写操作。
- 步骤 3. 查询 [SPI_3WIRE_COEF2 \[spi_busy\]](#),直到该状态位为 0。

----结束

12.4.3 spi_3wire 寄存器概览

spi_3wire 寄存器概览如[表 12-12](#) 所示。

表12-12 spi_3wire 寄存器概览（基地址：0x120C_3000）

偏移地址	名称	描述	页码
0x0000	SPI_3WIRE_COEF0	3 线 SPI 配置寄存器 0	12-64
0x0004	SPI_3WIRE_COEF1	3 线 SPI 配置寄存器 1	12-65
0x0008	SPI_3WIRE_COEF2	3 线 SPI 配置寄存器 2	12-65

12.4.4 spi_3wire 寄存器描述

SPI_3WIRE_COEF0

SPI_3WIRE_COEF0 为 3 线 SPI 配置寄存器 0。

Offset Address: 0x0000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	RO	reserved	保留。	0x000000
[7:0]	RW	spi_clk_div	SPI 接口的时钟率。取值范围为 1~255。 spi_clk_div 的值用来产生 SPI 时钟，公式为 $FSPICLK = FAPBCLK / (2 \times (spi\_clk\_div + 1))$ 其中 FAPBCLK 为 APB 总线的时钟频率。例如，APB 总线时钟为 50MHz，期望 SPI 的时钟为 1MHz，则 spi_clk_div 应该配置为 $(50MHz/1MHz)/2 - 1 = 24$。	0x00



SPI_3WIRE_COEF1

SPI_3WIRE_COEF1 为 3 线 SPI 配置寄存器 1。

Offset Address: 0x0004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	spi_rw	SPI 读写操作选择。 0: 写; 1: 读。	0x0
[30:16]	RW	spi_add	SPI 操作地址。	0x0000
[15:0]	RW	spi_wdata	SPI 写数据。	0x0000

SPI_3WIRE_COEF2

SPI_3WIRE_COEF2 为 3 线 SPI 配置寄存器 2。

Offset Address: 0x0008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:18]	RO	reserved	保留。	0x00000
[17]	RO	spi_busy	SPI 操作状态位。 1: SPI 总线上有操作; 0: SPI 总线处于 IDLE。	0x0
[16]	RW	start	启动 SPI 读写操作。只有 spi_busy 为 0 时才能启动，回读值为 0，无意义。	0x0
[15:0]	RO	spi_rdata	SPI 回读数据。	0x0000

12.5 eMMC/SD/SDIO 控制器

12.5.1 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 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 12-13 所示。



表12-13 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
SDIO0	SDIO0_CCLK_OUT	SDIO0_CCLK_OUT
	SDIO0_CARD_POWER_EN	SDIO0_CARD_POWER_EN
	SDIO0_CARD_DETECT	SDIO0_CARD_DETECT
	SDIO0_CCMD	SDIO0_CCMD
	SDIO0_CDATA0	SDIO0_CDATA0
	SDIO0_CDATA1	SDIO0_CDATA1
	SDIO0_CDATA2	SDIO0_CDATA2
	SDIO0_CDATA3	SDIO0_CDATA3
SDIO1	SDIO1_CCLK_OUT	SDIO1_CCLK_OUT
	SDIO1_CCMD	SDIO1_CCMD
	SDIO1_CDATA0	SDIO1_CDATA0
	SDIO1_CDATA1	SDIO1_CDATA1
	SDIO1_CDATA2	SDIO1_CDATA2
	SDIO1_CDATA3	SDIO1_CDATA3
eMMC	EMMC_CLK	SFC_CLK
	EMMC_CMD	SFC_HOLD_IO3
	EMMC_DATA0	SFC_WP_IO2
	EMMC_DATA1	SFC_MISO_IO1
	EMMC_DATA2	SFC_CSN
	EMMC_DATA3	SFC_MOSI_IO0
	EMMC_RST_N	SYS_RSTN_OUT

SDIO0 支持符合以下协议的设备：

- Secure Digital Memory（SD mem-version3.0）

SDIO1 支持符合以下协议的设备：

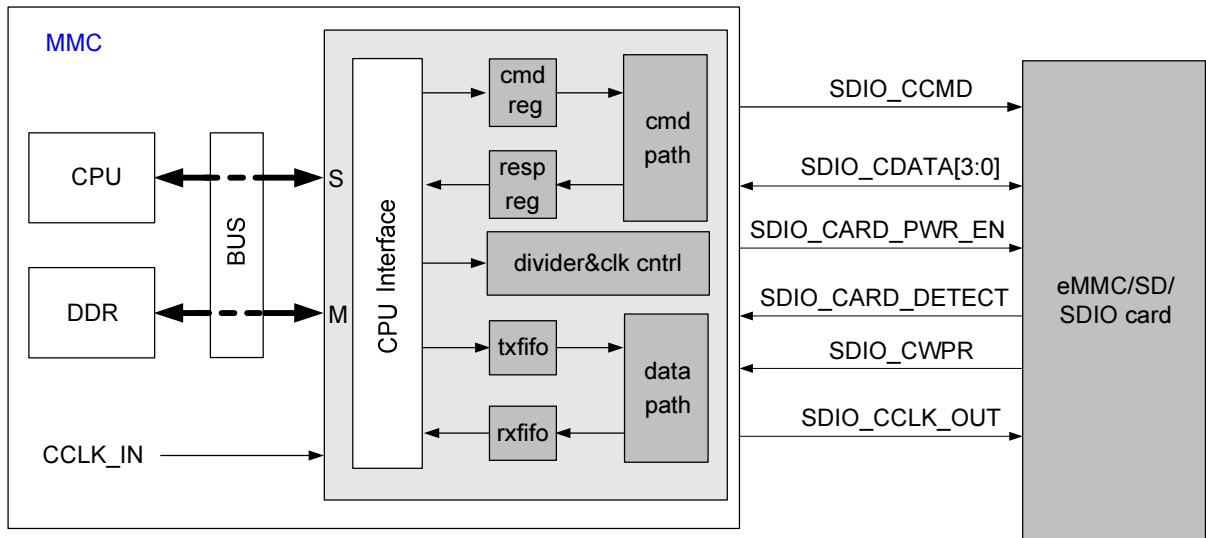
- Secure Digital I/O（SDIO version3.0）

eMMC 支持符合协议的设备：

- MultiMediaCard（eMMC-version 4.41, eMMC-version 4.5）

MMC 的功能框图如图 12-23 所示。

图12-23 MMC 功能框图



注：S：slave 接口；M：master 接口。

MMC 的功能特点有：

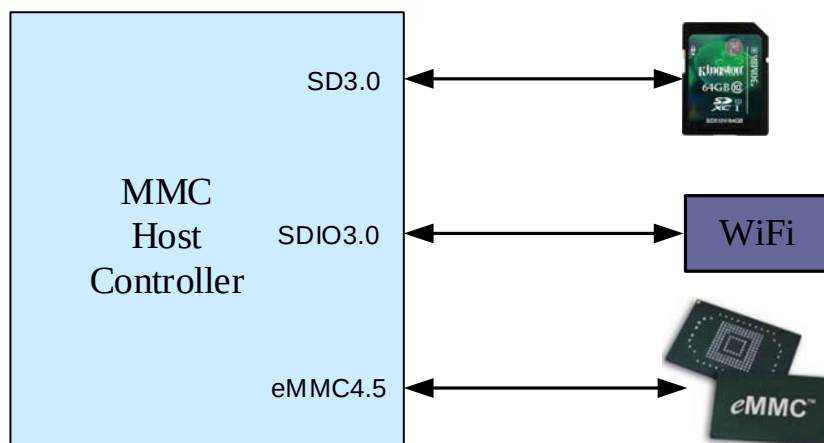
- 最大可支持 2TB 容量的 SD 卡和 eMMC 器件。
- 支持内部 DMA 数据传输方式。
- 支持 FIFO 阈值可配，DMA 传输时 Burst 大小可配。
- 支持 FIFO 上溢出与下溢出中断告警，防止数据传输错误。
- 支持命令与数据的 CRC 生成与校验。
- 接口时钟频率可编程。
- 支持低功耗模式关闭 MMC 时钟和接口时钟。
- 数据位宽支持 1、4bit，可根据对接器件选择。
- 支持 1byte~65535byte 的块数据读写操作。
- 支持 MMC 卡流数据读写方式。
- 支持 SDIO 的 suspend 操作、resume 操作和 read wait 操作。
- eMMC4.41 支持 default speed、high speed、DDR 4bit、GO_PRE_IDLE_STATE command 和 Hardware Reset。
- eMMC4.5 支持的特性：HS200（4bit），支持 CMD21，CMD49，支持 1.8/3.3V 操作控制，HS200 支持 1.8V IO 电压。

典型应用

MMC 的典型应用如图 12-24 所示。



图12-24 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12-25 MMC 指令格式

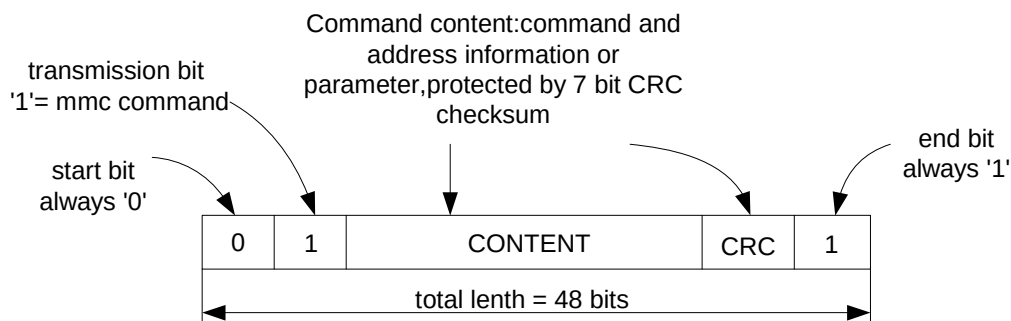
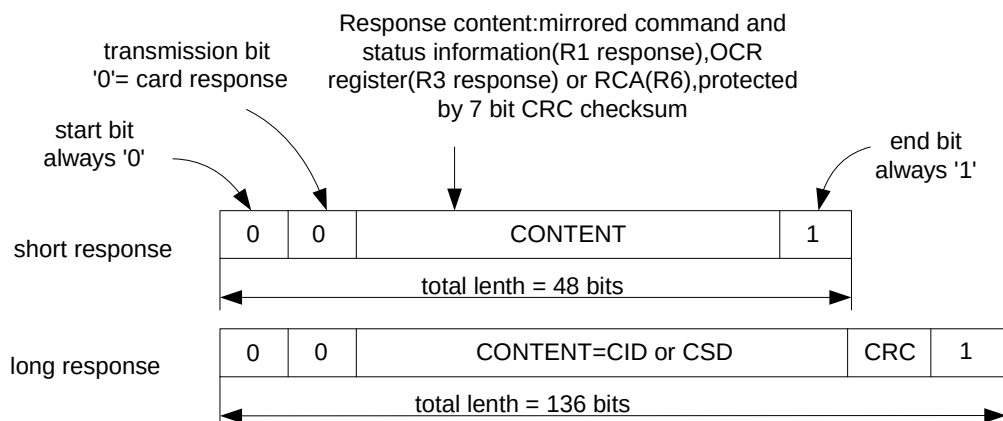




图12-26 MMC 指令响应格式



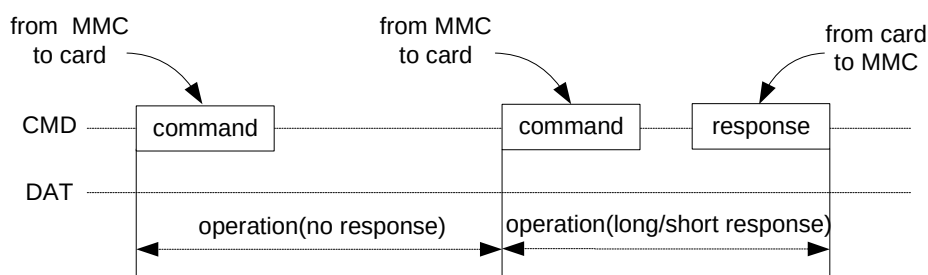
根据是否有数据传输，指令分为以下 2 种：

- 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 12-27 所示。

图12-27 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。两种方式的差别在于 MMC 通知卡结束一次传输的方式不同。

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。

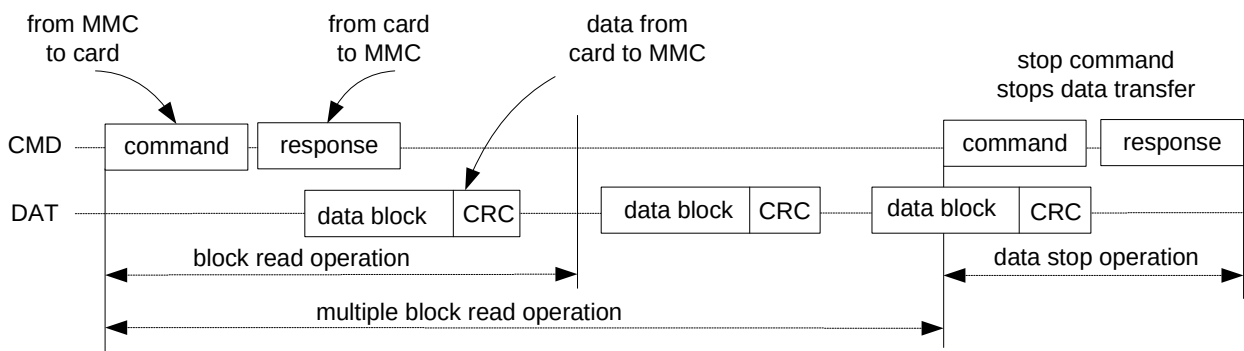
说明

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 12-28 和图 12-29 所示。

(1) 单块与多块读操作

图12-28 单块与多块读操作



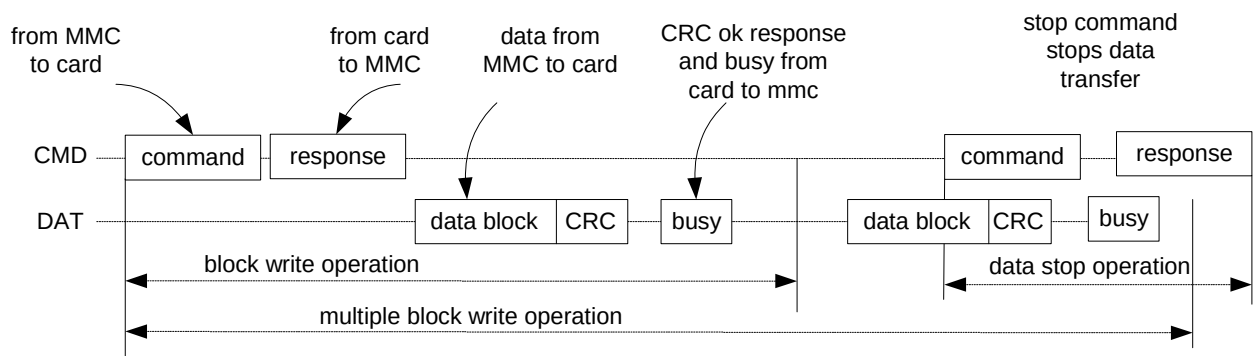
MMC 向卡发送单块或多块读指令。在接收响应的过程中，接收以块为单位的数据，其中每块数据中均包含有 CRC 校验位，以保证数据传输的完整性。

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12-29 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 MMC_CTYPE 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 12-30 和图 12-31 所示。

图12-30 1bit 数据线传输模式下的块数据格式

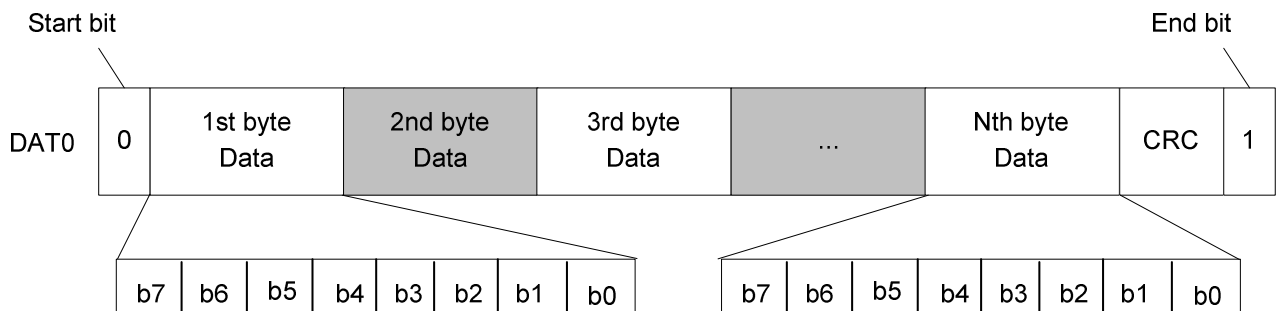
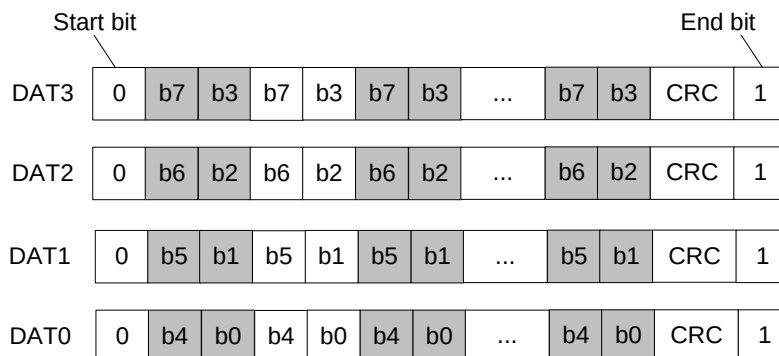




图12-31 4bit 数据线传输模式下的块数据格式



12.5.2 应用说明



说明

时钟复位寄存器请参考 3.2.6 "CRG 寄存器描述"。

时钟门控

在软件完成当前命令或数据传输，并且未启动新传输时可关闭 SDIO_CCLK_OUT 时钟，但需要确保 MMC 已处于空闲状态。

具体步骤如下：

- 步骤 1. 读寄存器 [MMC_STATUS](#)。
- 步骤 2. 若 [MMC_STATUS](#)[Command fsm states]和 [MMC_STATUS](#)[data_state_mc_busy]均为 0，则向 [MMC_CTRL](#) 写 “0”，屏蔽 MMC 中断、DMA 请求使能等，进入步骤 3。若其中有一个为非 0，则延时等待，返回步骤 1。
- 步骤 3. 向 CRG 寄存器 PERI_CRG82 bit[1]写 “0”，关闭 EMMC 时钟；向 CRG 寄存器 PERI_CRG88 bit[1]写 “0”，关闭 SDIO0 时钟；向 CRG 寄存器 PERI_CRG85 bit[1]写 “0”，关闭 SDIO1 时钟。



说明

如果需要重新开启 EMMC 工作时钟，则向 CRG 寄存器 PERI_CRG82 bit[1]写 “1”；如需重新开启 SDIO0 工作时钟，则向 CRG 寄存器 PERI_CRG88 bit[1]写 1，如果需要重新开启 SDIO1 工作时钟，则向 CRG 寄存器 PERI_CRG85 bit[1]写 “1”。

----结束

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，配置 CRG 寄存器 PERI_CRG82 bit[0]写 “1” 对 EMMC 模块进行软复位，配置 CRG 寄存器 PERI_CRG88 bit[0]写 “1” 对 SDIO0 模块进行软复位，配置 CRG 寄存器 PERI_CRG85 bit[0]写 “1” 对 SDIO1 模块进行软复位。查询寄存器 [MMC_STATUS](#)[Data_busy]确认控制器是否处于空闲状态。

建议在使用 MMC 之前和热插拔卡后，软复位 MMC。



工作时钟配置

在使用 MMC 前，需为其配置合适的工作时钟频率。eMMC 通过 CRG 寄存器 PERI_CRG82 bit[3:2]配置，SDIO0 通过 CRG 寄存器 PERI_CRG88 bit[3:2]配置，SDIO1 通过 CRG 寄存器 PERI_CRG85 bit[5:4]配置。

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。MMC 内部提供一个偶数分频器以便于将工作时钟分频至合适的接口时钟。控制器工作时钟 CCLK_IN 与接口时钟 SDIO_CCLK_OUT 的频率关系为：

$$F_{SDIO\_CCLK\_OUT} = F_{CCLK\_IN} / (2 \times clk\_divider)$$

其中，分频因子 clk_divider 为寄存器 MMC_CLKDIV[clk_divider]的值。不同卡类型支持的时钟频率不同，卡侧 $F_{SDIO\_CCLK\_OUT}$ 最高不超过 50MHz。

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 MMC_CLKENA 配置为 0x0000_0000，并将寄存器 MMC_CMD[Start_cmd]、MMC_CMD[Update_clk_regs_only]和 MMC_CMD[Wait_prvdata_complete]置“1”，等待寄存器 MMC_CMD[Start_cmd]被自动清零。

步骤 2. 设置分频因子。

根据所需要的时钟频率设置寄存器 MMC_CLKDIV，并将寄存器 MMC_CMD[Start_cmd]和 MMC_CMD[Update_clk_regs_only]置“1”，等待寄存器 MMC_CMD[Start_cmd]被自动清零。

步骤 3. 重新使能接口时钟。

将寄存器 MMC_CLKENA 配置为 0x0000_0001，并将寄存器 MMC_CMD[Start_cmd]和 MMC_CMD[Update_clk_regs_only]置“1”，等待寄存器 MMC_CMD[Start_cmd]被自动清零。

----结束



注意

只有当寄存器 MMC_CMD[Start_cmd]和 MMC_CMD[Update_clk_only]置“1”时，寄存器 MMC_CLKDIV、MMC_CLKENA 的值才会被载入。当载入成功以后，MMC 会自动清零寄存器 MMC_CMD[Start_cmd]。如果此时有其他指令正在执行，则会产生 HLE (Hardware Locked Error) 中断。若产生 HLE 中断，清除中断后重新发命令即可。

当有指令执行和数据传输时，不能改变卡的时钟参数。



初始化

与卡进行命令和数据的交互前，需要初始化 MMC，步骤如下：

- 步骤 1. 配置 MMC 工作时钟频率。请参见“12.5.2 应用说明”中的“工作时钟配置”。
- 步骤 2. 当卡上电，指令和数据信号线上拉稳定后，软复位 MMC。请参见“12.5.2 应用说明”中的“软复位”。
- 步骤 3. 清中断。将寄存器 `MMC_RINTSTS` bit[15:0]所有位置“1”，清除原始中断状态位。
- 步骤 4. 设置寄存器 `MMC_INTMASK`。将寄存器 `MMC_INTMASK` bit[15:0]所有位置“1”，使能各中断源。

若使用 DMA 方式进行数据传输，应将 `MMC_INTMASK` bit[4]、`MMC_INTMASK` bit[5]置“0”，以屏蔽接收/发送 FIFO 数据请求中断。
- 步骤 5. 将寄存器 `MMC_CTRL`[Int_enable]置“1”，使能 MMC 中断。
- 步骤 6. 配置超时参数寄存器 `MMC_TMOUT`。
- 步骤 7. 配置 FIFO 参数寄存器 `MMC_FIFOTH`。

----结束

完成以上步骤后，就可以配置接口时钟，往卡发送指令了。

非数据传输指令

MMC 在指令发送后，一旦收到响应（无论对错或超时），就会将寄存器 `MMC_RINTSTS` bit[2]置位。短响应保存到寄存器 `MMC_RESP0` 中，长响应保存到寄存器 `MMC_RESP0~MMC_RESP3` 中，寄存器 `MMC_RESP3` bit[31]为最高位，`MMC_RESP0` bit[0]为最低位。当指令发出以后，其错误是由指令响应以及寄存器 `MMC_RINTSTS` 的错误位反映。

发送非数据传输指令的步骤如下：

- 步骤 1. 在寄存器 `MMC_CMDARG` 中设置相应的指令参数。
- 步骤 2. 根据表 12-14 设置指令寄存器 `MMC_CMD`。
- 步骤 3. 等待指令被 MMC 执行。如果指令已执行，MMC 自动清零寄存器 `MMC_CMD`[Start_cmd]。
- 步骤 4. 检查寄存器 `MMC_RINTSTS` bit[12]是否产生 HLE 中断。
- 步骤 5. 等待指令执行完毕。MMC 收到响应（无论对错或超时）时将寄存器 `MMC_RINTSTS` bit[2]置“1”，表示 Command Done。
- 步骤 6. 检查是否有响应异常，必要时可读取响应值。

读取寄存器 `MMC_RINTSTS` bit[8]、`MMC_RINTSTS` bit[6]和 `MMC_RINTSTS` bit[1]，检查响应超时、响应 CRC 错误和响应错误。

----结束



注意

只有当寄存器 **MMC_CMD**[Start_cmd]置“1”、**MMC_CMD**[Update_clock_registers_only]置“0”时，寄存器 **MMC_BYTCNT**、**MMC_BLKSIZE**、**MMC_CMDARG** 和 **MMC_CMD** 的值才能被载入。载入成功后，MMC 自动清零 **MMC_CMD**[Start_cmd]。

如果有其他指令正在执行，会产生 HLE 中断，此时重新执行操作即可。在非数据传输指令执行时，寄存器 **MMC_BYTCNT** 和 **MMC_BLKSIZE** 的值被忽略。

表12-14 非数据传输指令时的寄存器 MMC_CMD 配置参考（默认值）

参数	取值	描述
Start_cmd	1	启动指令发送。
Update_clock_registers_only	0	非时钟参数更新指令。
data_transfer_expected	0	非数据传输指令。
card_number	0	-
cmd_index	Cmd index	指令序号。
send_initialization	0	当指令为卡复位时置“1”，如指令 CMD0。
stop_abort_cmd	0	当指令为停止数据传输时置“1”，如指令 CMD12。
response_length	0	当响应为长响应类型时置“1”。
response_expect	1	当指令无响应时置“0”，如：指令 CMD0、指令 CMD4、指令 CMD15。
Wait_prvdata_complete	1 or 0	在发送指令之前，MMC 必须等待正在处理的数据传输指令结束。建议此位一直置“1”，除非该指令是为了在数据传输时查询卡状态或停止当前数据的传输。
Check_response_crc	1 or 0	MMC 是否检查响应的 CRC 校验位。

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. 向寄存器 **MMC_CTRL**[fifo_reset]写“1”，复位 FIFO 指针，查询等待该位自动清零。



步骤 2. 向寄存器 [MMC_BYTCNT](#) 写入待传输数据的字节数。

步骤 3. 向寄存器 [MMC_BLKSIZE](#) 写入块的大小。

步骤 4. 向寄存器 [MMC_CMDARG](#) 写入读取数据的起始地址。

步骤 5. 根据表 12-15 设置寄存器 [MMC_CMD](#)。

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 [MMC_CMD](#) 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. 检查寄存器 [MMC_RINTSTS](#) bit[5]和 [MMC_RINTSTS](#) bit[10]，如果其中之一为 1 或都为 1，则从寄存器 [MMC_DATA](#) 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 [MMC_RINTSTS](#) bit[7]、[MMC_RINTSTS](#) bit[9]、[MMC_RINTSTS](#) bit[13]和 [MMC_RINTSTS](#) bit[15]。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 [MMC_RINTSTS](#) bit[3]为 1 时，数据传输完成，从寄存器 [MMC_DATA](#) 中读取残留在 FIFO 中的数据。

步骤 8. 若执行指令时已将寄存器 [MMC_CMD](#)[Send_auto_stop]置“1”，MMC 自动发送停止指令结束一次数据传输，请参见“12.5.2 应用说明”中的“Auto-stop 使用配置”。

----结束

表12-15 单块或多块读数据时的寄存器 [MMC_CMD](#) 配置参考（默认值）

参数	取值	描述
Start_cmd	1	启动指令发送。
Update_clock_registes_only	0	非时钟参数更新指令。
card_number	0	-
send_initialization	0	当指令为卡复位时置“1”，如指令 CMD0。
stop_abort_cmd	0	当指令为停止数据传输时置“1”，如指令 CMD12。
send_auto_stop	0 or 1	请参见“12.5.2 应用说明”中的“Auto-stop 使用配置”。
transfer_mode	0	块传输。
read/write	0	从卡中读取数据。
response_length	0	数据指令均为短响应。
data_transfer_expected	1	数据传输指令。



参数	取值	描述
response_expect	1	当指令无响应时置“0”，如：指令 CMD0、CMD4、CMD15。
cmd_index	Cmd index	命令序号。
Wait_prvdata_complete	1 or 0	在发送指令之前，主设备必须等待正在处理的数据传输指令结束。建议此位一直置1，除非该指令是为了询问卡状态或停止当前数据的传输。
Check_response_crc	1 or 0	MMC 是否检查响应的 CRC 校验位。

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. 向寄存器 [MMC_CTRL\[fifo_reset\]](#) 写“1”，复位 FIFO 指针，查询等待直至该位自动清零。
- 步骤 2. 向寄存器 [MMC_BYTCNT](#) 写入待传输数据的大小。
- 步骤 3. 向寄存器 [MMC_BLKSI](#) 写入块的大小。
- 步骤 4. 向寄存器 [MMC_CMDARG](#) 写入数据的起始地址。
- 步骤 5. 将数据写入 FIFO，即写寄存器 [MMC_DATA](#)，通常在开始时应写满 FIFO。
- 步骤 6. 根据表 12-16 设置寄存器 [MMC_CMD](#)。

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 7. 检查寄存器 [MMC_RINTSTS](#) bit[4]和 [MMC_RINTSTS](#) bit[10]，如果其中之一为 1 或都为 1，写寄存器 [MMC_DATA](#) 往 FIFO 填充数据；同时应检查数据错误中断，即检查寄存器 [MMC_RINTSTS](#) bit[7]、[MMC_RINTSTS](#) bit[9]、[MMC_RINTSTS](#) bit[13]和 [MMC_RINTSTS](#) bit[15]。如果有需要，程序可以发送一个停止指令以中止数据的传输。当寄存器 [MMC_RINTSTS](#) bit[3]为 1，数据传输结束。
- 步骤 8. 若执行指令时已将寄存器 [MMC_CMD\[Send_auto_stop\]](#)置“1”，MMC 自动发送停止指令结束一次数据传输。请参见“12.5.2 应用说明”中的“Auto-stop 使用配置”。
- 步骤 9. 查询并等待寄存器 [MMC_STATUS\[data_busy\]](#)由 1 变为 0。

----结束

表12-16 单块或多块写数据时的寄存器 [MMC_CMD](#) 配置参考（默认值）

参数	取值	描述
Start_cmd	1	启动指令发送。



参数	取值	描述
Update_clock_registes_only	0	非时钟参数更新指令。
card_number	0	-
send_initialization	0	当指令为卡复位时置“1”，如指令 CMD0。
stop_abort_cmd	0	当指令为停止数据传输时置“1”，如指令 CMD12。
send_auto_stop	0 or 1	请参见“12.5.2 应用说明”中的“Auto-stop 使用配置”。
transfer_mode	0	块传输。
read_write	1	往卡写入数据。
response_length	0	数据指令均为短响应。
data_transfer_expected	1	数据传输指令。
response_expect	1	当指令无响应时置“0”，如：指令 CMD0、CMD4、CMD15。
cmd_index	Cmd index	-
Wait_prvdata_complete	1 or 0	在发送指令之前，主设备必须等待直到正在处理的数据传输指令结束。建议此位一直置“1”，除非该指令是为了询问卡状态或停止当前数据的传输。
Check_response_crc	1 or 0	MMC 是否检查响应的 CRC 校验位。

流数据读写

流数据的读写方式，除了将寄存器 MMC_CMD[Transfer_mode]置“1”外，其他与块数据的读写方式一致。对于流数据的传输，通常需要使用 Auto-stop 功能。

内置 DMA 方式数据传输

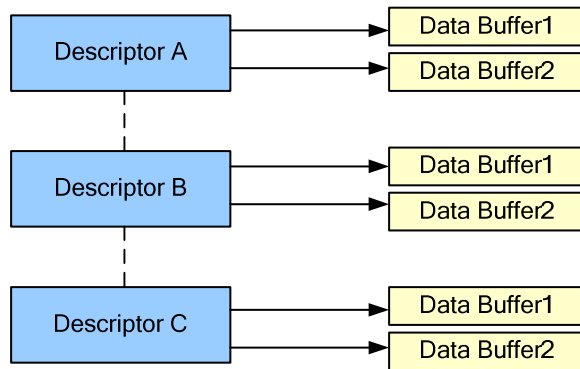
MMC 控制器含有内置 DMA 控制器（IDMAC），IDMAC 可以根据指定的描述子把数据从原地址搬移到目的地址。

描述子

IDMAC 可以使用如下两种类型的描述子：

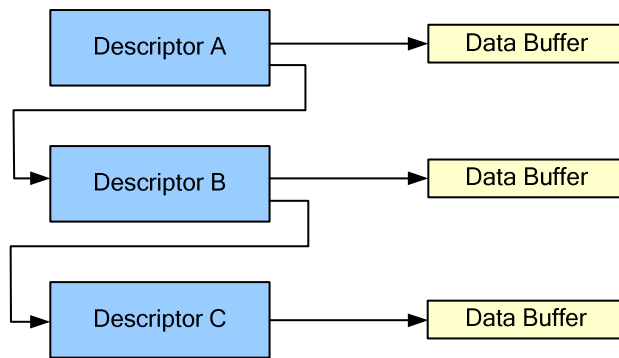
- 双 Buffer 结构：两个描述子之间的间隔根据 MMC_BMOD 寄存器的 DSL 位来指定。双 buffer 结构的示意图如图 12-32 所示。

图12-32 双 buffer 结构示意图



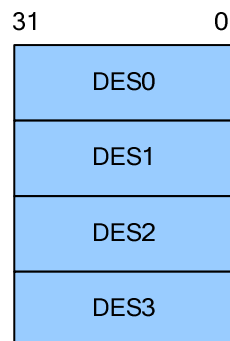
- **链结构**：每个描述子指向唯一的 `buffer` 和下一个描述子。链结构的示意图如图 12-33 所示。

图12-33 链结构示意图



描述子必须以 word 对齐，每个描述子包含 16bytes 的控制和状态信息。描述子的内部结构如图 12-34 所示。

图12-34 32bit 位宽的描述子的结构



DES0 保护控制和状态信息，具体含义如表 12-17 所示。



表12-17 DES0 各 bit 的含义

位	名称	说明
31	OWN	描述子属性指示。 0: 表示该描述子属于 CPU; 1: 表示该描述子属于 IDMAC。 使用内部 DMA 完成数据传输后, IDMAC 将该位清 0
30	CES	读写卡时出错的状态指示。 0: 没有错; 1: 出错。
29:6	RES	保留。
5	ER	描述子链路结束指示。 0: 不是最后一个; 1: 该链路最后一个描述子。 只对双 buffer 结构的描述子有意义。
4	CH	第二个地址 (DES3 的地址) 的含义。 0: DES3 中的第二个地址是指第二个 buffer 的地址; 1: DES3 中的第二个地址是指下一个描述子的地址。 但该位为 1 时, DES1[25:13]应该是 0。
3	FS	为 1 表示该描述子包含第一个数据 buffer, 如果第 1 个数据 buffer 的大小为 0, 那么下一个描述子包含起始数据。
2	LD	为 1 表示该描述子指向的 buffer 是最后一个数据 buffer。
1	DIC	为 1 时阻止数据传输结束上报中断。
0	RES	保留。

DES1 指示 buffer 大小, 具体含义如表 12-18 所示。

表12-18 DES1 各 bit 的含义

位	名称	说明
31:26	RES	保留。
25:13	BS2	第二个数据 buffer 的 byte 数, 必须是 4 的整数倍。DES0[4]为 1 时该位无意义。
12:0	BS1	第一个数据 buffer 的 byte 数, 必须是 4 的整数倍。



DES2 表示第一个数据 buffer 的地址指针，具体含义如表 12-19 所示。

表12-19 DES2 各 bit 的含义

位	名称	说明
31:0	BAP1	第一个数据 buffer 的物理地址，必须 word 对齐。

DES3 表示第二个地址，具体含义如表 12-20 所示。

表12-20 DES3 各 bit 的含义

位	名称	说明
31:0	BAP2	当使用的是双 buffer 结构时，表示第二个数据 buffer 的物理地址。如果 DES0[4]是 1，该地址表示下一个描述子的物理地址。

初始化

- 步骤 1. 配置 MMC_BMOD 寄存器，设置总线参数。
- 步骤 2. 配置 MMC_IDINTEN 寄存器，屏蔽不需要的寄存器。
- 步骤 3. 创建发送或接收描述子链表，然后配置 MMC_DBADDR 寄存器，设置起始地址。
- 步骤 4. IDMAC 尝试从描述子链表中获取描述子。

----结束

发送

- 步骤 1. CPU 创建描述子（DES0~DES3）并将 DES0[31]位 OWN 置为 1，同时准备好数据 buffer。
- 步骤 2. 把写数据命令写入 MMC_CMD 寄存器。
- 步骤 3. 通过 MMC_FIFOTH 寄存器设置 TX_Wmark。
- 步骤 4. IDMAC 获取描述子并判断 OWN 是否为 1，如果 OWN 不为 1 则需要等待 CPU 释放描述子，同时，IDMAC 会进入 suspend 状态，需要 CPU 配置 MMC_PLDMND 寄存器使 IDMAC 重新获取描述子。
- 步骤 5. OWN 为 1 时，IDMAC 从数据 buffer 中搬移数据到 MMC 内部 FIFO。
- 步骤 6. 如果中断已经使能，数据搬移完成后 IDMAC 状态寄存器 MMC_IDSTS 的相应位将被更新，同时 OWN 位被清零。

----结束



接收

- 步骤 1. CPU 创建描述子 (DES0~DES3) 并将 DES0[31]位 OWN 置为 1。
- 步骤 2. 把读数据命令写入 [MMC_CMD](#) 寄存器。
- 步骤 3. 通过 [MMC_FIFOTH](#) 寄存器设置 RX_WMark。
- 步骤 4. IDMAC 获取描述子并判断 OWN 是否为 1, 如果 OWN 不为 1 则需要等待 CPU 释放描述子, 同时, IDMAC 会进入 suspend 状态, 需要 CPU 配置 [MMC_PLDMND](#) 寄存器使 IDMAC 重新获取描述子。
- 步骤 5. OWN 为 1 时, IDMAC 从 MMC 内部 FIFO 中搬移数据到外部的数据 buffer。
- 步骤 6. 如果中断已经使能, 数据搬移完成后 IDMAC 状态寄存器 [MMC_IDSTS](#) 的相应位将被更新, 同时 OWN 位被清零。

----结束

Auto-stop 使用配置

在多块读写指令操作中, 需使用停止指令完成一次数据传输。停止指令的发送可以通过非数据传输指令的方式发送, 也可以使用 Auto-stop 功能发送。

Auto-stop 功能的应用场合如下:

- SD 卡
多块读写操作, 如指令 CMD18 和指令 CMD25。
- eMMC 器件
 - 流数据读写操作。
 - open-ended 方式多块读写操作, 如指令 CMD18 和指令 CMD25。

建议使用控制器提供的 Auto-stop 功能, 配置过程如下:

- 步骤 1. 在执行块数据传输指令操作中, 将寄存器 [MMC_CMD](#)[Send_auto_stop]置“1”。
- 步骤 2. 在所有数据传输完成后, MMC 自动发送一次停止指令, 以便卡能返回相应的状态。
- 步骤 3. 检查寄存器 [MMC_RINTSTS](#)[auto_cmd_done], 判断该停止指令是否完成。其响应保存在寄存器 [MMC_RESP1](#) 中。

----结束

停止或中止数据传输

停止指令用于打断 MMC 与卡之间的数据传输, 中止指令用于打断 I/O 数据的传输 (仅用于 SDIO_IOONLY 或 SDIO_COMBO)。

这两种指令的用法如下:

- 停止指令



该指令可以在数据传送的任何阶段进行发送。因为该指令用于停止数据传输，所以需要将寄存器 `MMC_CMD` bit[5:0] 设为指令 `CMD12`，将寄存器 `MMC_CMD` bit[14] 设为 1，将寄存器 `MMC_CMD` bit[13] 设为 0。

- 中止指令

该指令仅用于 `SDIO_IOONLY` 或 `SDIO_COMBO`。为了中止数据的传输，需要通过指令 `CMD52` 设置 `SDIO` 卡的寄存器 `CCCR[ASx]`。

Suspend 和 Resume 操作

对于 `SDIO` 卡（内部最多可容纳 7 个功能设备），`MMC` 可通过 `Suspend` 操作暂停某一功能设备的数据传输，将 `SD` 接口总线出让给另一个有更高优先级的功能设备。高优先级的功能设备完成数据传输后，`MMC` 通过 `Resume` 操作恢复前一功能设备未完成的数据传输。

`Suspend` 与 `Resume` 操作通过设置 `SDIO` 卡的寄存器 `CCCR` 来实现。读写寄存器 `CCCR`，使用指令 `CMD52`。

`Suspend` 操作步骤如下：

- 步骤 1. 查询寄存器 `CCCR[SBS]`，判断 `SDIO` 卡是否支持 `suspend/resume` 操作。
- 步骤 2. 查询寄存器 `CCCR[FSx]` 和 `[BS]`，判断待暂停的功能设备是否正在进行数据传输。

如果 `[BS]` 为 1，则 `[FSx]` 位所指定的功能设备正在进行数据传输。

- 步骤 3. 将寄存器 `CCCR[BR]` 置 “1”，暂停当前数据传输。
- 步骤 4. 检测寄存器 `CCCR[BS]` 和 `[BR]` 状态是否清零。

`[BS]`（Bus 状态）在数据总线正被使用时保持为 1。`[BR]`（Bus 释放）在总线完全释放之前保持为 1。当 `[BR]` 与 `[BS]` 都为 0 时，所选功能设备的数据传输被暂停。

- 步骤 5. 如果暂停正在进行的读操作，在 `suspend` 操作成功完成以后，必须置位寄存器 `MMC_CTRL[Abort_read_data]` 来复位 `MMC` 的数据传输功能。复位完成后，寄存器 `MMC_CTRL[Abort_read_data]` 自动清零。
- 步骤 6. 读寄存器 `MMC_TCBCNT` 获取已传输数据字节数。

----结束

`Resume` 操作步骤如下：

- 步骤 1. 检查卡是否处于非传输状态，以确认总线处于空闲状态。
- 步骤 2. 如果卡处于 `disconnect` 状态，使用指令 `CMD7` 将它选中。卡的状态可以通过指令 `CMD52/CMD53` 获取。
- 步骤 3. 检查待恢复的功能设备是否准备好进行数据传输（查询寄存器 `CCCR[RF]`）。如果 `[RF]=1`，则该功能设备已准备好进行数据传输。
- 步骤 4. 为了恢复传输，使用指令 `CMD52` 将功能设备号写入寄存器 `CCCR[FS]`。发送指令 `CMD52` 的同时应启动 `MMC` 进入数据传输状态，即向寄存器 `MMC_BLKSIZE` 写入块的大小，向寄存器 `MMC_BYTCNT` 写入剩余待传输数据量。



寄存器 [MMC_CMDARG](#) 的配置如表 12-21 所示，寄存器 [MMC_CMD](#) 的配置与块传输类似。

- 步骤 5. 当指令 [CMD52](#) 成功发送以后，数据传输恢复正常。读取 [SDIO](#) 设备的 [DF](#)（Resume Data Flag）标志位，如果为 1，则在功能被恢复的同时，数据开始传输；如果为 0，则已无数据需要传输。
- 步骤 6. 如果 [DF](#) 标志位为 0，在读数据的情况下，[MMC](#) 会等待一段时间后产生数据超时错误中断。

----结束

表12-21 Resume 操作时的寄存器 [MMC_CMDARG](#) 配置参考

MMC_CMDARG	取值	描述
Bit[31]	1	读写标志。
Bit[30:28]	0	功能设备号，访问寄存器 CCCR 。
Bit[27]	1	实时标志，先写后读。
Bit[26]	-	-
Bit[25:9]	0x0D	寄存器地址。
Bit[8]	-	-
Bit[7:0]	被恢复的功能号	写数据。



注意

系统处于低功耗模式后不能通过 [MMC](#) 唤醒。

Read wait 操作

[Read wait](#) 操作用于暂停 [SDIO](#) 卡当前功能设备的数据传输。[MMC](#) 根据需要决定暂停数据传输的时间长度。

[Read wait](#) 操作步骤如下：

- 步骤 1. 检查卡是否支持 [read wait](#) 操作。
- 使用指令 [CMD52](#) 读取寄存器 [CCCR\[SRW\]](#)。如果为 1，则卡的所有功能设备都支持 [read wait](#) 操作。
- 步骤 2. 将寄存器 [MMC_CTRL\[Read_wait\]](#)置“1”。
- 步骤 3. 如需恢复数据传输，清零寄存器 [MMC_CTRL\[Read_wait\]](#)。

----结束



12.5.3 寄存器概览

MMC 寄存器概览如表 12-22 所示。

表12-22 MMC 寄存器概览（EMMC 基址是 0x1010_0000，SDIO0 基址是 0x100F_0000，SDIO1 基址是 0x1002_0000）

偏移地址	名称	描述	页码
0x0000	MMC_CTRL	MMC 控制寄存器	12-86
0x0004	MMC_PWREN	Power_en 控制寄存器	12-88
0x0008	MMC_CLKDIV	时钟分频系数寄存器，显示模块输出时钟与输入时钟分频比	12-88
0x0010	MMC_CLKENA	时钟使能寄存器	12-88
0x0014	MMC_TMOUT	超时时间寄存器	12-89
0x0018	MMC_CTYPE	卡类型寄存器	12-89
0x001C	MMC_BLKSIZE	块大小配置寄存器	12-90
0x0020	MMC_BYTCNT	块传输计数寄存器	12-90
0x0024	MMC_INTMASK	MMC 中断屏蔽寄存器	12-90
0x0028	MMC_CMDARG	MMC 指令参数寄存器	12-91
0x002C	MMC_CMD	MMC 命令寄存器	12-91
0x0030	MMC_RESP0	MMC 响应寄存器 0	12-94
0x0034	MMC_RESP1	MMC 响应寄存器 1	12-95
0x0038	MMC_RESP2	MMC 响应寄存器 2	12-95
0x003C	MMC_RESP3	MMC 响应寄存器 3	12-95
0x0040	MMC_MINTSTS	MMC 屏蔽后中断状态寄存器	12-95
0x0044	MMC_RINTSTS	MMC 原始中断状态寄存器	12-96
0x0048	MMC_STATUS	MMC 状态寄存器	12-97
0x004C	MMC_FIFOTH	FIFO 水位值寄存器	12-99
0x0050	MMC_CDETECT	卡检测寄存器	12-100
0x005C	MMC_TCBCNT	发送到卡的 byte 数目寄存器	12-100
0x0060	MMC_TBBCNT	BIU FIFO 传送数据的 byte 数目寄存器	12-100
0x0064	MMC_DEBNCE	去抖动计数器	12-101



偏移地址	名称	描述	页码
0x0074	MMC_UHS_REG	UHS-1 寄存器	12-101
0x0078	MMC_CARD_RSTN	eMMC 器件复位控制寄存器	12-101
0x0080	MMC_BMOD	总线模式寄存器	12-102
0x0084	MMC_PLDMND	Poll demand 寄存器	12-103
0x0088	MMC_DBADDR	描述子链表的基地址寄存器	12-103
0x008C	MMC_IDSTS	IDMAC 状态寄存器	12-103
0x0090	MMC_IDINTEN	IDMAC 中断使能寄存器	12-104
0x0094	MMC_DSCADDR	当前描述子地址寄存器	12-105
0x0098	MMC_BUFADDR	当前数据 buffer 地址寄存器	12-105
0x0100	MMC_CARDTHRCTL	阈值控制寄存器	12-106
0x0108	MMC_UHS_REG_EXT	UHS 扩展寄存器	12-106
0x010C	MMC_DDR_REG	eMMC4.5 DDR START bit 检测控制寄存器	12-107
0x0110	MMC_ENABLE_SHIFT	相移寄存器	12-107
0x0200	MMC_DATA	数据寄存器，为 FIFO 入口地址	12-108

12.5.4 寄存器描述

MMC_CTRL

MMC_CTRL 为 MMC 控制寄存器。

Offset Address: 0x0000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25]	RW	use_internal_dmac	是否使用内置 DMA 搬移数据。 0: CPU 使用 slave 接口搬移数据; 1: 使用内部 DMA 搬移数据。	0x0
[24:9]	-	reserved	保留。	0x00
[8]	RW	abort_read_data	读数据过程中是否中止。 0: 无效; 1: 在读操作过程中发送 suspend 指令后,	0x0



Bits	Access	Name	Description	Reset
			软件让卡去查找 suspend 何时出现。 一旦 suspend 出现, 软件把该位置“1”, 使数据传输状态机回到 Idle 状态, 等待下一个块传输。 状态机回到 Idle 后状态, 此比特自动清零。	
[7]	RW	send_irq_response	发送中断响应控制。 0: 无效; 1: 发送自动 IRQ 回复。 当回复已经发送, 此比特自动清零。 为了等待 MMC 产生中断, 主机发送指令 CMD40 并等待来自 MMC 的中断回复。同时, 如果主机希望不再停留在中断等待状态, 可将该比特置“1”, 发送指令 CMD40 并回到 IDLE 状态。	0x0
[6]	RW	read_wait	读等待控制。 0: 禁止读等待; 1: 使能读等待。 此位仅用在具有 SDIO 功能的卡上, 要求卡支持读等待功能。	0x0
[5]	-	reserved	保留。	0x0
[4]	RW	int_enable	全局中断使能。 0: 禁止; 1: 使能。 当该比特有效且有中断源被使能时, 中断输出才有效。	0x0
[3]	-	reserved	保留。	0x0
[2]	RW	dma_reset	内部 DMAC 软复位控制。 0: 无效; 1: 复位内部 DMA 接口。 该比特在两个 AHB 时钟周期后自动复位。	0x0
[1]	RW	fifo_reset	内部 FIFO 软复位控制。 0: 无效; 1: 复位 FIFO 指针。 当复位操作结束后, 该比特自动复位。	0x0
[0]	RW	controller_reset	控制器软复位控制。	0x0



Bits	Access	Name	Description	Reset
			0: 无效; 1: 复位 eMMC/SD/SDIO Host 模块。	

MMC_PWREN

MMC_PWREN 为 Power_en 控制寄存器。

Offset Address: 0x0004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留。	0x00000000
[0]	RW	power_enable	POWER 控制。 0: 关闭电源; 1: 打开电源。	0x0

MMC_CLKDIV

MMC_CLKDIV 为时钟分频系数寄存器，显示模块输出时钟与输入时钟分频比。假如模块输入时钟为 40MHz，寄存器配置为 1，则输出时钟为 20MHz。时钟分频系数值为 $2 \times N$ 。比如，N 为 0x0 时， $2 \times 0 = 0$ （无分频）；N 为 0x1 时为 2 分频；N 为 0xFF 时为 510 分频。

Offset Address: 0x0008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x00000000
[7:0]	RW	clk_divider0	时钟分频系数 0。时钟分频系数是 $2 \times n$ ，比如，0 为无分频，1 为 2 分频，ff 即为 510 分频等等。	0x0

MMC_CLKENA

MMC_CLKENA 为时钟使能寄存器。

Offset Address: 0x0010 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x00
[16]	RW	cclk_low_power	卡的低功耗控制，用于关闭卡时钟。	0x0



			0: 无低功耗模式; 1: 低功耗模式。 当卡处于 IDLE 状态时, 停止该卡时钟。该功能一般只用于 SD 卡、eMMC 器件。对于 SDIO, 为了能检测到中断, 时钟不能停止。	
[15:1]	-	reserved	保留。	0x00
[0]	RW	cclk_enable	卡的时钟使能控制。 0: 时钟关闭; 1: 时钟使能。	0x0

MMC_TMOUT

MMC_TMOUT 为超时时间寄存器。

Offset Address: 0x0014 Total Reset Value: 0xFFFF_FF40

Bits	Access	Name	Description	Reset
[31:8]	RW	data_timeout	卡数据传输超时时间, 该值同时用作 CPU 的 data starvation 超时时间。单位为该卡的 mmc_clk 周期。	0xFFFFFFFF
[7:0]	RW	response_timeout	回复超时时间, 单位为该卡的 mmc_clk 周期。	0x40

MMC_CTYPE

MMC_CTYPE 为卡类型寄存器。

Offset Address: 0x0018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x0000
[16]	RW	card_width_0	配置卡的总线宽度, 必须配置为 0。 0: 非 8bit 模式; 1: 8bit 模式。 下面为卡配置说明: 如果 bit[16]=1, 卡将被配置成 8bit 模式。此时 bit[0]的值被忽略。 如果 bit[16]=0, 卡将会是 1bit 或者 4bit 模式, 取决于 bit[0]的配置。	0x0



[15:1]	-	reserved	保留。	0x0000
[0]	RW	card_width_1	配置卡的总线宽度。 0: 1bit 模式; 1: 4bit 模式。	0x0

MMC_BLKSIZE

MMC_BLKSIZE 为块大小配置寄存器。

Offset Address: 0x001C Total Reset Value: 0x0000_0200

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	block_size	块大小值，初始值为 512byte/block。	0x0200

MMC_BYTCNT

MMC_BYTCNT 为块传输计数寄存器。

Offset Address: 0x0020 Total Reset Value: 0x0000_0200

Bits	Access	Name	Description	Reset
[31:0]	RW	byte_count	被传输的字节数，应该为块大小的整数倍。 如果为非块传输，本寄存器需要置“0”，此时必须由软件发出 stop/abort 命令来控制数据传输操作。	0x0000_0200

MMC_INTMASK

MMC_INTMASK 为 MMC 中断屏蔽寄存器。

Offset Address: 0x0024 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x000
[16]	RW	sdio_int_mask	屏蔽 SDIO 中断。 0: 屏蔽; 1: 使能。	0x0



[15:0]	RW	int_mask	屏蔽中断。 0：屏蔽； 1：使能。 每 bit 对应的含义如下： Bit[15]: End-bit error (read)/Write no CRC (EBE) Bit[14]: Auto command done (ACD) Bit[13]: Start-bit error (SBE)/Busy Complete Interrupt(BCI) Bit[12]: Hardware locked write error (HLE) Bit[11]: FIFO underrun/overflow error (FRUN) Bit[10]: Data starvation-by-host timeout(HTO)/Vlt_switch_int Bit[9]: Data read timeout (DRTO) Bit[8]: Response timeout (RTO) Bit[7]: Data CRC error (DCRC) Bit[6]: Response CRC error (RCRC) Bit[5]: Receive FIFO data request (RXDR) Bit[4]: Transmit FIFO data request (TXDR) Bit[3]: Data transfer over (DTO) Bit[2]: Command done (CD) Bit[1]: Response error (RE) Bit[0]: Card detect (CD)	0x0000

MMC_CMDARG

MMC_CMDARG 为 MMC 指令参数寄存器。

Offset Address: 0x0028 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	cmd_arg	配置传输给卡的指令参数。指令参数与协议相关，每个指令都对应一个具体的指令参数。	0x0000_0000

MMC_CMD

MMC_CMD 为 MMC 命令寄存器。



Offset Address: 0x002C Total Reset Value: 0x2000_0000

Bits	Access	Name	Description	Reset
[31]	RW	start_cmd	启动控制。 0: 不启动; 1: 启动命令。当命令已经被传到 CIU, 该比特被清零。 CPU 不允许修改此寄存器。如果修改, 产生 hardware lock error 中断。 CPU 发送完命令以后需要查询该比特, 查到该比特变为 0 以后再发下一个命令。	0x0
[30]	-	reserved	保留。	0x0
[29]	RW	use_hold_reg	Use Hold 寄存器。 0: 发送到卡的 CMD 和 DATA 信号不经过 HOLD 寄存器; 1: 发送到卡的 CMD 和 DATA 信号经过 HOLD 寄存器。	0x1
[28]	RW	volt_switch	电压切换控制。 0: 禁止电压切换; 1: 使能电压切换。	0x0
[27]	RW	boot_mode	Boot 模式。 0: 强制 Boot 模式; 1: 交替 Boot 模式。	0x0
[26]	RW	disable_boot	禁用 Boot。 当软件同时使能此 bit 和 Start_cmd 时, 控制器终止 Boot 操作。 不允许同时使能 Enable_boot 和 Disable_boot。	0x0
[25]	RW	expect_boot_ack	使能 Boot 响应。 当软件同时使能此 bit 和 Enable_boot 时, 控制器将检测 Boot 响应信号, 即 “0-1-0” 序列。	0x0
[24]	RW	enable_boot	使能 Boot。 此 bit 只能使用在 “强制 Boot 模式”。当软件同时使能此 bit 和 Start_cmd 时, 控制器拉低 CMD 信号启动 Boot 流程。 不允许同时使能 Enable_boot 和 Disable_boot。	0x0
[23:22]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[21]	RW	update_clock_registers_only	是否自动更新。 0: 正常命令顺序, 寄存器 MMC_CMD、MMC_CMDARG、MMC_TMOUT、MMC_CTYPE、MMC_BLKSIZE、MMC_BYTCNT 的值会从 BIU 传到 CIU。CIU 在新命令中使用寄存器新值; 1: 不发送命令, 只更新卡时钟域的时钟寄存器值。寄存器 MMC_CLKDIV、MMC_CLKENA 的值被转换到卡时钟域。 无需发命令给卡即可转换卡时钟(转换频率和时钟开关), 用于调整卡时钟频率以及控制卡时钟开关。 每次改变卡时钟时, 需要将该比特设为“1”。此时不会有命令被传送给卡, 也不会产生 Command Done 中断。	0x0
[20:16]	RW	card_number	正在使用的卡的序号。	0x00
[15]	RW	send_initialization	是否发初始序列。 0: 在发送 Send_initialization 命令前不送出初始序列(80 个时钟周期的“1”); 1: 在发送 Send_initialization 命令前送出初始序列。 卡上电时, 在发送任何命令以前, 都必须先送出初始序列来做初始化, 即该比特置“1”。	0x0
[14]	RW	stop_abort_cmd	当 open_end 或定长的数据传输操作正在进行时, 该位取值含义如下。 0: 不发送 stop/abort 命令; 1: 发送 stop/abort 命令, 终止正在进行的数据传输。	0x0
[13]	RW	wait_prvdata_complete	是否立即发送指令。 0: 立即发送命令(即使前一个数据传输还没完成); 1: 等到前一个数据传输完成才发送命令。 “0”为典型值, 用于数据传输时读取状态或中断传输。	0x0
[12]	RW	send_auto_stop	是否发送 stop 命令。 0: 数据传完以后不发 stop 命令; 1: 数据传完以后发 stop 命令。 在非数据传输时, 该比特被忽略。	0x0



Bits	Access	Name	Description	Reset
[11]	RW	transfer_mode	传输模式。 0: block 传输命令; 1: stream 传输命令。 在非数据传输时, 该比特被忽略。	0x0
[10]	RW	read_write	读写控制。 0: 从卡读取数据; 1: 往卡写数据。 在非数据传输时, 该比特被忽略。	0x0
[9]	RW	data_transfer_expected	数据传输指示。 0: 没有数据从卡输出; 1: 有数据从卡输出。	0x0
[8]	RW	check_response_crc	是否 CRC 检查。 0: 不检查 CRC response; 1: 检查 CRC response。 一些命令回复时没有返回有效的 CRC。为了禁止 Host 对 CRC 进行检查, 软件需要针对这些命令禁止该功能。	0x0
[7]	RW	response_length	Response 长度。 0: 短 response 从卡输出; 1: 长 response 从卡输出。 长 response 是 128bit, 短 response 是 32bit。	0x0
[6]	RW	response_expect	是否有 response。 0: 无 response 从卡输出; 1: 有 response 从卡输出。	0x0
[5:0]	RW	cmd_index	指令序号。	0x00

MMC_RESP0

MMC_RESP0 为 MMC 响应寄存器 0。

Offset Address: 0x0030 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	response0	Response 的 bit[31:0]。	0x0000_0000



MMC_RESP1

MMC_RESP1 为 MMC 响应寄存器 1。

Offset Address: 0x0034 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	response1	长 Response 的 bit[63:32]。 当 CIU 发出 Auto-stop 命令，response 就会被保存在该寄存器。上一条命令的 response 仍然会被保存在寄存器 MMC_RESP0 内。 Auto-stop 只供数据传输使用，回复类型一直是短 response。	0x0000_0000

MMC_RESP2

MMC_RESP2 为 MMC 响应寄存器 2。

Offset Address: 0x0038 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	response2	长 Response 的 bit[95:64]。	0x0000_0000

MMC_RESP3

MMC_RESP3 为 MMC 响应寄存器 3。

Offset Address: 0x003C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	response3	长 Response 的 bit[127:96]。	0x0000_0000

MMC_MINTSTS

MMC_MINTSTS 为 MMC 屏蔽后中断状态寄存器。

Offset Address: 0x0040 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x000
[16]	RO	sdio_interrupt	SDIO 中断屏蔽状态。 SDIO 中断只有在对应的 MMC_INTMASK	0x0



			[sdio_int_mask]被使能才有效。 0: 无 SDIO 中断从卡输出; 1: 有 SDIO 中断从卡输出。	
[15:0]	RO	int_status	各中断状态。 Bit[15]: End-bit error(read)/write no CRC(EBE) Bit[14]: Auto command done(ACD) Bit[13]: Start bit error(SBE)/Busy Complete Interrupt(BCI) Bit[12]: Hardware locked write error(HLE) Bit[11]: FIFO underrun/overflow error(FRUN) Bit[10]: Data starvation by host timeout(HTO)/Vlt_switch_int Bit[9]: Data read timeout(DRTO) Bit[8]: Response timeout(RTO) Bit[7]: Data CRC error(DCRC) Bit[6]: Response CRC error(RCRC) Bit[5]: Receive FIFO data request(RXDR) Bit[4]: Transmit FIFO data request(TXDR) Bit[3]: Data transfer over(DTO) Bit[2]: Command done(CD) Bit[1]: Response error(RE) Bit[0]: Card detect(CD)	0x0000

MMC_RINTSTS

MMC_RINTSTS 为 MMC 原始中断状态寄存器。

Offset Address: 0x0044 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x000
[16]	RW	sdio_interrupt	SDIO 中断原始状态。 0: 无 SDIO 中断从卡输出; 1: 有 SDIO 中断从卡输出。 中断状态位的值与中断屏蔽无关。	0x0
[15:0]	RW	int_status	各中断的原始状态。写“1”清零, 写“0”不修改。中断状态位的值与中断屏蔽	0x0000



			无关。 Bit[15]: End-bit error (read)/write no CRC (EBE) Bit[14]: Auto command done (ACD) Bit[13]: Start bit error(SBE)/Busy Complete Interrupt(BCI) Bit[12]: Hardware locked write error (HLE) Bit[11]: FIFO underrun/overflow error (FRUN) Bit[10]: Data starvation by host timeout(HTO)/Volt_switch_int Bit[9]: Data read timeout (DRTO)/Boot Data Start(BDS) Bit[8]: Response timeout (RTO)/Boot Ack Received(BAR) Bit[7]: Data CRC error (DCRC) Bit[6]: Response CRC error (RCRC) Bit[5]: Receive FIFO data request (RXDR) Bit[4]: Transmit FIFO data request (TXDR) Bit[3]: Data transfer over (DTO) Bit[2]: Command done (CD) Bit[1]: Response error (RE) Bit[0]: Card detect (CD)	

MMC_STATUS

MMC_STATUS 为 MMC 状态寄存器。

Offset Address: 0x0048 Total Reset Value: 0x0000_0106

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29:17]	RO	fifo_count	FIFO 计数值。	0x000
[16:11]	RO	response_index	前一个 response 的序号，包括 Auto-stop 的 response。	0x00
[10]	RO	data_state_mc_busy	数据发送和接收状态机状态。 0: 数据发送/接收状态机空闲； 1: 数据发送/接收状态机正忙。	0x0
[9]	RO	data_busy	DAT[0]指示的 data_busy 的状态。	0x0



Bits	Access	Name	Description	Reset
			0: 卡闲; 1: 卡忙。	
[8]	-	reserved	保留。	0x1
[7:4]	RO	commandfsm_states	Command 状态机状态。 0x0: Idle; 0x1: Send init sequence; 0x2: Tx cmd start bit; 0x3: Tx cmd tx bit; 0x4: Tx cmd index +arg; 0x5: Tx cmd crc7; 0x6: Tx cmd end bit; 0x7: Rx resp start bit; 0x8: Rx resp IRQ response; 0x9: Rx resp tx bit; 0xA: Rx resp cmd idx; 0xB: Rx resp data; 0xC: Rx resp crc7; 0xD: Rx resp end bit; 0xE: Cmd path wait NCC; 0xF: Wait, CMD-to-response turnaround。	0x0
[3]	RO	fifo_full	FIFO 满标志。 0: FIFO 空; 1: FIFO 满。	0x0
[2]	RO	fifo_empty	FIFO 空标志。 0: FIFO 非空; 1: FIFO 空。	0x1
[1]	RO	fifo_tx_watermark	FIFO 是否到达 Transmit watermark level。 0: 未达到水线; 1: 达到水线。	0x1
[0]	RO	fifo_rx_watermark	FIFO 是否到达 Receive watermark level。 0: 未达到水线; 1: 达到水线。	0x0



MMC_FIFOTH

MMC_FIFOTH 为 FIFO 水线值寄存器。

Offset Address: 0x004C Total Reset Value: 0x00FF_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30:28]	RW	Multiple_Transaction_Size	传输的突发长度。 000: 1; 001: 4; 010: 8; 011: 16; 100: 32; 101: 64; 110: 128; 111: 256。	0x0
[27:16]	RW	rx_wmark	读数据时的 FIFO threshold watermark level。当 FIFO 计数大于该值时，使能 DMA 请求。数据传输结束后为了完成剩余的数据，不论该值为多少，都会产生 DMA 请求。 在非 DMA 模式，使能 RXDR 中断。在数据传输完成后如果 FIFO 计数没有大于该值，不会产生中断。这需要软件主动查询 DTD 中断来读取剩余的数据。 在 DMA 模式，当数据传输结束时，即使剩余的数据比门限低，DMA 还是会发出 single 传送请求读取数据，直到产生 DTD 中断。 限制：rx_wmark $\leq$ (FIFO_DEPTH - 2)。 建议：配置为(FIFO_DEPTH/2) - 1。 注意：FIFO_DEPTH = 256（下同）	0x0FF
[15:12]	RW	reserved	保留。	0x0
[11:0]	RW	tx_wmark	发送数据时的 FIFO threshold watermark level。当 FIFO 计数小于该值时，使能 DMA 请求。数据传输结束后为了完成剩余的数据，不论该值为多少，都会产生 DMA 请求。 在非 DMA 模式，使能 RXDR 中断。在数据传输完成后如果 FIFO 计数没有大于该值，不会产生中断。这需要软件主动查询	0x000



			DTD 中断来读取剩余的数据。 在 DMA 模式，当数据传输结束时，即使剩余的数据比门限低，DMA 还是会发出 single 传送请求读取数据，直到产生 DTD 中断。 限制：tx_wmark(FIFO_DEPTH) - 2。 建议：大于(FIFO_DEPTH/2) - 1 时发出请求。	

MMC_CDETECT

MMC_CDETECT 为卡检测寄存器。

Offset Address: 0x0050 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x0000_0000
[0]	RO	card_detect_n	卡检测信号。 值由管脚 SDIO0_CARD_DETECT 决定。	0x1

MMC_TCBCNT

MMC_TCBCNT 为发送到卡的 byte 数目寄存器。

Offset Address: 0x005C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	trans_card_byte_count	已经从 CIU 发送到卡的 byte 数目。 32bit AHB 总线对该寄存器进行访问时，应该一次读完 32bit，避免出现 read-coherency 错误。	0x0000_0000

MMC_TBBCNT

MMC_TBBCNT 为 BIU FIFO 传送数据的 byte 数目寄存器。

Offset Address: 0x0060 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	trans_fifo_byte_count	在 CPU/DMA 与 BIU FIFO 之间传送数据的 byte 数目。	0x0000_0000



			32bit AHB 总线对该寄存器进行访问时，应该一次读完 32bit，避免出现 read-coherency 错误。	

MMC_DEBNCE

MMC_DEBNCE 为去抖动计数器。

Offset Address: 0x0064 Total Reset Value: 0x00FF_FFFF

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RW	debounce_count	去抖动过滤逻辑使用的总线时钟周期的数目，通常的去抖动时间为 5ms-25ms。	0xFFFFFFFF

MMC_UHS_REG

MMC_UHS_REG 为 UHS-1 寄存器。

Offset Address: 0x0074 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x0000
[16]	RW	ddr_reg	DDR 模式控制寄存器。 0: 非 DDR 模式; 1: DDR 模式。	0x0
[15:1]	RW	reserved	保留。	0x0000
[0]	RW	volt_reg	电压模式控制寄存器。 0: 3.3V; 1: 1.8V。	0x0

MMC_CARD_RSTN

MMC_CARD_RSTN 为 eMMC 器件复位控制寄存器。

Offset Address: 0x0078 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	RW	reserved	保留。	0x00000000



[0]	RW	card_reset	eMMC 器件复位控制器，控制管脚 CARD_RESET。 0: 复位； 1: 撤销复位。	0x1

MMC_BMOD

MMC_BMOD 为总线模式寄存器。

Offset Address: 0x0080 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留。	0x000
[10:8]	RO	pbl	IDMAC burst 传输的长度。 000: 1; 001: 4; 010: 8; 011: 16; 100: 32; 101: 64; 110: 128; 111: 256。	0x0
[7]	RW	de	IDMAC 使能。 0: 不使能; 1: 使能。	0x0
[6:2]	RW	dsl	描述子跨越的长度。即两个非链接的描述子之间间隔多少个 WORD。该参数只用于双 buffer 结构描述子。	0x00
[1]	RW	fb	固定 burst 长度。 0: 使用 SINGLE 和 INCR burst 类型; 1: 使用 SINGLE、INCR4、INCR8、INCR16 burst 类型。	0x0
[0]	RW	swr	IDMAC 内部寄存器软复位控制。 0: 不复位; 1: 复位。置位后 1 个时钟周期自动清零。	0x0



MMC_PLDMND

MMC_PLDMND 为 Poll demand 寄存器。

Offset Address: 0x0084 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	WO	pd	如果描述子 DES0[OWN]为 0，IDMAC 会进入 suspend 状态，CPU 往该寄存器中写任何值都可以使 IDMAC 重新获取描述子。	0x0000_0000

MMC_DBADDR

MMC_DBADDR 为描述子链表的基地址寄存器。

Offset Address: 0x0088 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	sdl	描述子链表的起始地址。即第一个描述子的基地址，需 Word 地址对齐。	0x0000_0000

MMC_IDSTS

MMC_IDSTS 为 IDMAC 状态寄存器。

Offset Address: 0x008C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x000
[16:13]	RO	fsm	IDMAC 状态机当前状态。 0x0: DMA_IDLE; 0x1: DMA_SUSPEND; 0x2: DESC_RD; 0x3: DESC_CHK; 0x4: DMA_RD_REQ_WAIT; 0x5: DMA_WR_REQ_WAIT; 0x6: DMA_RD; 0x7: DMA_WR; 0x8: DESC_CLOSE; 其他: 保留。	0x0



[12:10]	RW	eb	总线错误类型指示。 001: 发送被中止; 010: 接收被中止; 其它: 保留。	0x0
[9]	RW	ais	异常总中断。是 FBE、DU、CES 取或。写 1 清 0。	0x0
[8]	RW	nis	正常总中断。是 TI、RI 取或。写 1 清 0。	0x0
[7:6]	RO	reserved	保留。	0x0
[5]	RW	ces	卡出错指示。指示卡在接收发送数据过程中状态。	0x0
[4]	RW	du	描述子无效中断。当 DES0[OWN]为 0 时, 该位被置位。写 1 清 0。	0x0
[3]	-	reserved	保留。	0x0
[2]	RW	fbe	致命的总线错误中断。当该 bit 被置位时, IDMAC 停止总线访问。写 1 清 0。	0x0
[1]	RW	ri	接收完成中断。指示一个描述子的数据接收完成。写 1 清 0。	0x0
[0]	RW	ti	发送完成中断。指示一个描述子的数据发送完成。写 1 清 0。	0x0

MMC_IDINTEN

MMC_IDINTEN 为 IDMAC 中断使能寄存器。

Offset Address: 0x0090 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x000
[9]	RW	ai	异常中断使能位。 0: 不使能; 1: 使能 FBE/DU/CES 中断。	0x0
[8]	RW	ni	正常中断使能位。 0: 不使能; 1: 使能 TI/RI 中断。	0x0
[7:6]	-	reserved	保留。	0x0
[5]	RW	ces	卡出错中断使能位。	0x0



			0: 不使能; 1: 使能。	
[4]	RW	du	描述子无效中断使能位。 0: 不使能; 1: 使能。	0x0
[3]	-	reserved	保留。	0x0
[2]	RW	fbe	致命总线错中断使能位。 0: 不使能; 1: 使能。	0x0
[1]	RW	ri	接收中断使能位。 0: 不使能; 1: 使能。	0x0
[0]	RW	ti	发送中断使能位。 0: 不使能; 1: 使能。	0x0

MMC_DSCADDR

MMC_DSCADDR 为当前描述子地址寄存器。

Offset Address: 0x0094 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	had	描述子指针。在数据传输过程中自动更新。该寄存器指向当前正要被 IDMAC 使用的描述子的起始地址。	0x0000_0000

MMC_BUFADDR

MMC_BUFADDR 为当前数据 buffer 地址寄存器。

Offset Address: 0x0098 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	hba	数据 buffer 指针。在数据传输过程中自动更新，该寄存器指向当前正被 IDMAC 访问的数据 buffer 地址。	0x0000_0000



MMC_CARDTHRCTL

MMC_CARDTHRCTL 为阈值控制寄存器。

Offset Address: 0x0100 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27:16]	RW	cardrdthreshold	读阈值大小。 最大只能配置为 512。	0x000
[15:2]	-	reserved	保留。	0x0000
[1]	RW	BsyClrIntEn	Busy Clear Interrupt 使能。 0: 不使能; 1: 使能。	0x0
[0]	RW	cardrdthr_en	读阈值使能控制。 0: 不使能; 1: 使能。	0x0

MMC_UHS_REG_EXT

MMC_UHS_REG_EXT 为 UHS 扩展寄存器。

Offset Address: 0x0108 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:23]	RW	clk_drv_phase_ctrl	控制 clk_in_drv 的时钟相位。 000: 0° ; 001: 45° ; 010: 90° ; 011: 135° ; 100: 180° ; 101: 225° ; 110: 270° ; 111: 315° 。	0x0
[22:19]	-	reserved	保留。	0x0
[18:16]	RW	clk_smpl_phase_ctrl	控制 clk_in_sample 的时钟相位。 000: 0° ;	0x0



			001: 45° ; 010: 90° ; 011: 135° ; 100: 180° ; 101: 225° ; 110: 270° ; 111: 315° 。	
[15:0]	RO	reserved	保留。	0x0000

MMC_DDR_REG

MMC_DDR_REG 为 eMMC4.5 DDR START bit 检测控制寄存器。

Offset Address: 0x010C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	half_start_bit	DWC_mobile_storage 内部 start bit 检测机制控制。对于 eMMC 4.5 器件，start bit 可以是： 0: 全周期。 1: 不足一个周期。 对于上述的 eMMC 4.5 器件，设置 half_start_bit 为 1，对于 SD 应用设置为 0。 当 SATRT_BIT_SAMPLING_DDR 或者 ASYNC_SAMPLE_DATA_PATH 设置为 1 情况下的 DDR 模式此 bit 被忽略。	0x0000

MMC_ENABLE_SHIFT

MMC_ENABLE_SHIFT 为相移寄存器。

Offset Address: 0x0110 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x0000_0000
[1:0]	RW	Enable Phase Shift Register	相移控制寄存器。 00: 默认相移； 01: 相移生效在下一个上升沿；	0x0



			10: 相移生效在下一个下降沿; 11: 保留。	

MMC_DATA

MMC_DATA 为数据寄存器，为 FIFO 入口地址。在读写 FIFO 时，应先读取 FIFO 溢出。

Offset Address: 0x0200 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	data	读写 FIFO 的地址。地址范围 0x200~0x200+FIFO_DEPTH，均选中 FIFO。	0x0000_0000

12.6 红外接口

12.6.1 概述

红外遥控接收单元 IR（Infrared Remoter）通过红外接口接收红外数据。

12.6.2 特点

IR 模块具有以下特点：

- 软件可配置关闭红外遥控接收模块。
- 支持 2 种工作模式：
 - 模式 0: 支持 NEC with simple repeat code、NEC with full repeat code、SONY 和 TC9012 四种数据格式解码，及接收数据错误检测和红外遥控唤醒等功能。
 - 模式 1: 支持任意数据格式的 symbol 电平宽度检测。
- 模式 0 时，支持接收数据帧溢出中断、接收数据帧格式错误中断、接收数据帧中断、按键释放的中断、各种中断构成的组合中断。
- 模式 1 时，支持接收 symbol 溢出中断、接收到 symbol 中断、symbol 超时中断、各种中断构成的组合中断。
- 支持初始中断状态查询和屏蔽后中断状态查询。
- 支持中断清除和屏蔽（写清）。
- 支持红外遥控唤醒。
- 支持参考时钟频率 1MHz~128MHz 可选，软件可编程控制分频因子使工作时钟预分频到 1MHz。



12.6.3 功能描述

当 IR 模块接收到红外遥控器发射的红外信号时，便对其进行解码，然后传送给 ARM 系统。ARM 系统再根据接收到的码的不同进行相应的操作，实现期望的功能。IR 模块连接在 ARM 子系统内的 APB 总线上，当芯片处于低功耗状态时（CPU 处于低频模式），IR 模块会在接收一个完整的帧数据后，产生中断信号送给 CPU，实现红外遥控唤醒功能。

通过对多种红外遥控器发出的信号进行分析，发现在不同的遥控器发出的红外指令中，引导码各不相同，而且后面的控制指令也有较大差别，甚至指令码的位数也不相同，这是因为这些红外遥控器的设计没有遵循统一的红外遥控标准。尽管遵循的标准不同，但是基本的编码思想是相同的，都是采用不同的周期和不同占空比的脉冲分别表示 0 和 1。不同遥控器占空比可能不同，且脉冲周期也不相同。根据这些不同，对一些码型类似的红外数据进行分类：NEC with simple repeat code 的数据格式、NEC with full repeat code 的数据格式、TC9012 的数据格式和 SONY 的数据格式。

红外接收数据码型统计情况如表 12-23～表 12-25 所示。



说明

此统计表以芯片实测为准，表格上的数据只是以前的测量值，并不完全正确。

表12-23 红外接收数据码型的统计表（NEC with simple repeat code）

数据格式		NEC with simple repeat code			
		uPD6121G	D6121/BU5777/D1913	LC7461M-C13	AEHA
引导码（10μs）	LEAD_S	900	900	900	337.6
	LEAD_E	450	450	450	168.8
bit0（10μs）	B0_L	56	56	56	42.2
	B0_H	56	56	56	42.2
bit1（10μs）	B1_L	56	56	56	42.2
	B1_H	169	169	169	126.6
simple repeat code（10μs）	SLEAD_S	900	900	900	337.6
	SLEAD_E	225	225	225	337.6
burst（10μs）		55	55	55	42.2
帧长（10μs）		10800	10800	10800	8777.6～12828.8
有效数据位		32	32	42	48



表12-24 红外接收数据码型的统计表（NEC with full repeat code）

数据格式		NEC with full repeat code						
		uPD6121G	LC7461 M-C13	MN602 4-C5D6	MN6014 -C6D6	MATNEW	MN6030	PANA SONIC
引导码 (10μs)	LEAD_S	900	900	337.6	349.2	348.8	349	352
	LEAD_E	450	450	337.6	349.2	374.4	349	352
bit0 (10μs)	B0_L	56	56	84.4	87.3	43.6	87.3	88
	B0_H	56	56	84.4	87.3	43.6	87.3	88
bit1 (10μs)	B1_L	56	56	84.4	87.3	43.6	87.3	88
	B1_H	169	169	253.2	174.6	130.8	261.9	264
simple repeat code (10μs)	SLEAD_S	无	无	无	无	无	无	无
	SLEAD_E							
burst (10μs)		55	55	84.4	87.3	43.6	87.3	88
帧长 (10μs)		10800	10800	10130	10470	12413.6~ 16594.4	10500	10400
有效数据位		32	42	22	24	48	22	22

表12-25 红外接收数据码型的统计表（TC9012 和 SONY 码）

数据格式		TC9012	SONY			
		TC9012F/9243	SONY-D7C5	SONY-D7C6	SONY-D7C8	SONY-D7C13
引导码 (10μs)	LEAD_S	450	240	240	240	240
	LEAD_E	450	60	60	60	60
bit0 (10μs)	B0_L	56	60	60	60	60
	B0_H	56	60	60	60	60
bit1 (10μs)	B1_L	56	120	120	120	120
	B1_H	169	60	60	60	60
simple repeat code (10μs)	SLEAD_S	无	无	无	无	无
	SLEAD_E					
burst (10μs)		56	无	无	无	无



数据格式	TC9012	SONY			
	TC9012F/9243	SONY-D7C5	SONY-D7C6	SONY-D7C8	SONY-D7C13
帧长（10μs）	10800	4500	4500	4500	4500
有效数据位	32	12	13	15	20

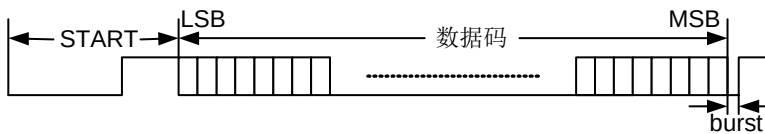
12.6.3.1 NEC with simple repeat code 数据格式

帧格式

NEC with simple repeat code 数据格式是由 START（引导码）、数据码和 burst 三部分组成，其中 START 是由一个起始码（低电平），一个结束码（高电平）组成；数据码的有效位数以及某位表示的含义由具体的码型而定，它是按照 LSB first 的顺序接收的；burst 信号用于接收最后一个数据码位。

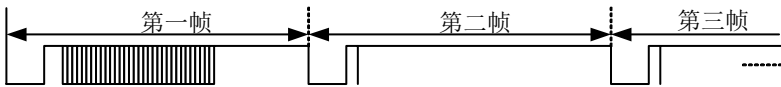
发送单个 NEC with simple repeat code 的帧格式如图 12-35 所示。

图12-35 发送单个 NEC with simple repeat code 码的帧格式



如果按键时间持续超过一帧的时间，则在收到完整数据帧后，接下来收到的数据帧仅由简化的引导码和 burst 信号组成。引导码也是由起始码（低电平）和结束码（高电平）组成，持续按键连续发送 NEC with simple repeat code 码的帧格式如图 12-36 所示。

图12-36 持续按键连续发送 NEC with simple repeat code 码的帧格式

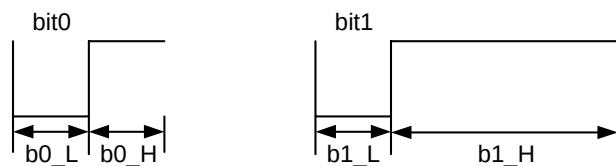


码格式

NEC with simple repeat code 的 bit0 或 bit1 定义如图 12-37 所示。



图12-37 NEC with simple repeat code 码 bit0 和 bit1 定义



NEC simple repeat code 单发代码格式和连发代码格式分别如图 12-38 和图 12-39 所示。

图12-38 NEC with simple repeat code 码单发代码格式

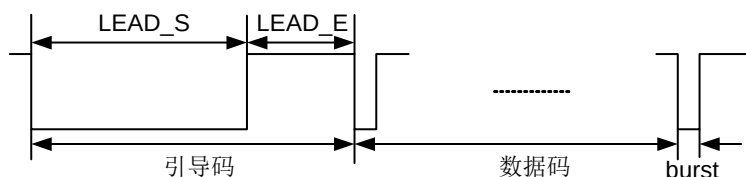


图12-39 NEC with simple repeat code 码连发代码格式



注 1：图中高低电平脉宽的宽度以及帧长均有各个具体码型决定，请参见表 12-23～表 12-25。

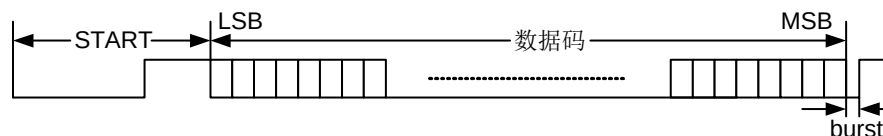
注 2：帧长不能大于 160ms，否则无法识别简化引导码。

12.6.3.2 NEC with full repeat code 数据格式

帧格式

NEC with full repeat code 的数据格式是由 START（引导码）、数据码和 burst 三部分组成。START 是由一个起始码（低电平）和一个结束码（高电平）组成；数据码的有效位数以及某位表示的含义由具体的码型而定，它是按照 LSB first 的顺序接收的；burst 信号用于接收最后一个数据码位。发送单个 NEC with full repeat code 帧格式如图 12-40 所示。

图12-40 发送单个 NEC with full repeat code 码的帧格式





如果按键时间持续超过一帧的时间，则在收到完整数据帧（第一帧）后，接下来收到的数据帧还是一个完整的数据帧格式（即按照帧间隔重复发送第一帧数据），持续按键连续发送 NEC with full repeat code 码的帧格式如图 12-41 所示。

图12-41 持续按键连续发送 NEC with full repeat code 码的帧格式

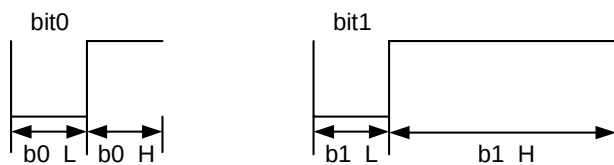


通过图 12-40 和图 12-41 可以看出：NEC with simple repeat code 与 NEC with full repeat code 唯一不同之处就是重复帧的格式，NEC with simple repeat code 发送的是简化的引导码，而 NEC with full repeat code 发送的是完整帧格式，第一帧和重复帧完全相同。

码格式

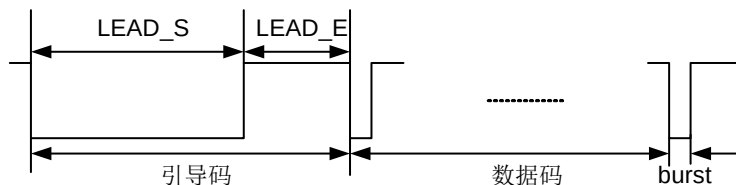
NEC with full repeat code 码 bit0 或 bit1 定义如图 12-42 所示。

图12-42 NEC with full repeat code 码 bit0 和 bit1 定义



NEC with full repeat code 码单发代码格式如图 12-43 所示。

图12-43 NEC with full repeat code 码单发代码格式



注：图中高低电平的脉宽宽度以及帧长均有各个具体码型决定，请参见表 12-23～表 12-25。

12.6.3.3 TC9012 数据格式

帧格式

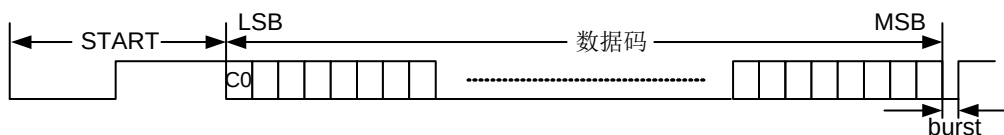


注意

根据 TC9012 码的数据格式特点，所有按键编码的第一位都必须全是 1 或者全是 0，否则会产生不需要的持续按键帧。

TC9012 的数据格式是由 START（引导码）、数据码和 burst 三部分组成，其中 START 是由一个起始码（低电平），一个结束码（高电平）组成；数据码的有效位数以及某位表示的含义由具体的码型而定，其是按照 LSB first 的顺序接收的；burst 信号用于接收最后一个数据码位。发送单个 TC9012 码的帧格式如图 12-44 所示。

图12-44 发送单个 TC9012 码的帧格式



如果按键时间持续超过一帧的时间，则在收到完整数据帧后，接下来收到的数据帧由引导码、一个数据位和 burst 信号三部分组成。引导码也是由起始码（低电平）和结束码（高电平）组成；该数据位是上一帧接收的第一个数据位（C0）的反码。发送连续 TC9012 码的帧格式如图 12-45 所示。

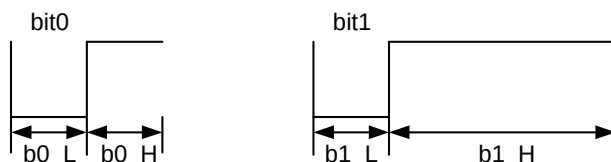
图12-45 持续按键连续发送 TC9012 码的帧格式



码格式

TC9012 码 bit0 或 bit1 定义如图 12-46 所示。

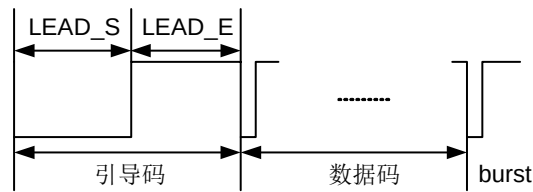
图12-46 TC9012 码 bit0 和 bit1 定义



TC9012 码单发代码格式如图 12-47 所示。

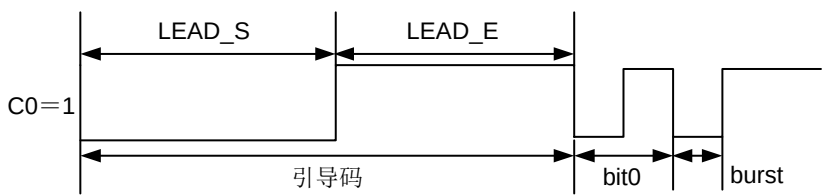


图12-47 TC9012 码单发代码格式



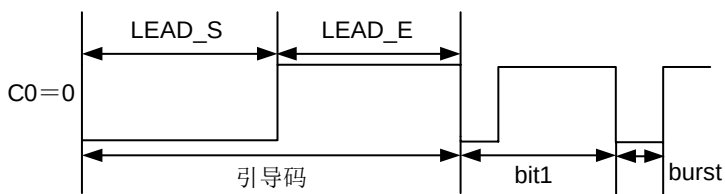
C0=1 时，TC9012 码连发代码格式如图 12-48 所示。

图12-48 TC9012 码连发代码格式（C0=1）



C0=0 时，TC9012 码连发代码格式如图 12-49 所示。

图12-49 TC9012 码连发代码格式（C0=0）



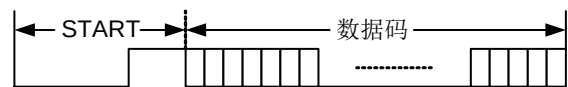
注：图中高低电平的脉宽宽度以及帧长均有各个具体码型决定。请参见表 12-23～表 12-25。另外值得注意的是帧长不能大于 160ms，否则无法识别重复帧。

12.6.3.4 SONY 的数据格式

帧格式

SONY 码数据格式是由 START（引导码）和数据码两部分组成。其中 START 由一个起始码（低电平），一个结束码（高电平）组成；数据码的有效位数以及某位表示的含义由具体的码型而定，其是按照 LSB first 的顺序接收的。发送单个 SONY 码帧格式如图 12-50 所示。

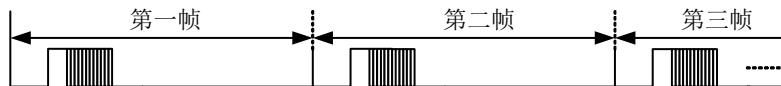
图12-50 发送单个 SONY 帧格式





如果按键时间持续超过一帧的时间，则再收到完整数据帧后，接下来收到的数据帧还是一个完整的数据帧格式。持续按键连续发送 SONY 码帧格式如图 12-51 所示。

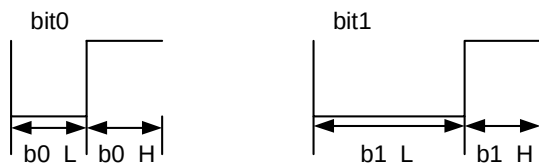
图12-51 持续按键连续发送 SONY 码帧格式



码格式

SONY 码 bit0 或 bit1 定义如图 12-52 所示。

图12-52 bit0 和 bit1 定义



注：图中高低电平的脉宽宽度以及帧长均有各个具体码型决定。请参见表 12-23～表 12-25。

12.6.4 工作方式

工作时钟和软复位

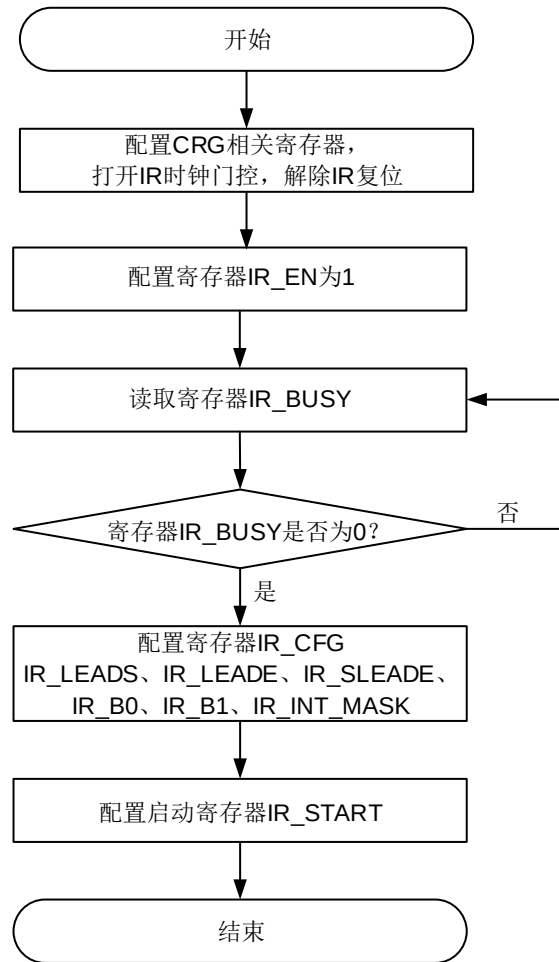
配置 CRG 寄存器 PERI_CRG111[ir_clken]为 1，打开 IR 模块的时钟门控。配置 CRG 寄存器 PERI_CRG111[ir_srst_req]为 1，对 IR 模块单独软复位。复位后各配置寄存器的值均复位为默认值，因此复位后需要重新对这些寄存器进行初始化配置。

寄存器配置实例

IR 模块初始化操作流程如图 12-53 所示。



图12-53 IR 模块初始化操作流程



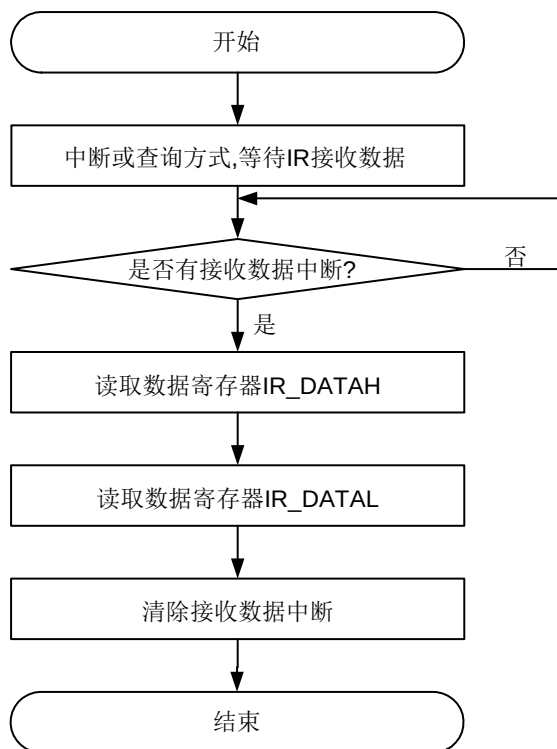
IR 模块初始化操作流程如下：

- 步骤 1. 配置 CRG 相关寄存器，打开 IR 时钟门控，解除 IR 复位。
- 步骤 2. 配置 `IR_EN` bit[0] 为 1，打开 IR 接收模块。
- 步骤 3. 读 `IR_BUSY`，判断 IR 模块配置的当前状态。
 - 若读取的值为 1，表明 IR 模块处于配置忙状态，则继续查询 `IR_BUSY`（**注意：此时软件不要对 IR 模块的其他控制寄存器进行配置，否则配置无效。**）。
 - 若读取的值为 0，表明 IR 模块处于配置空闲状态，则执行步骤 4。
- 步骤 4. 配置 `IR_CFG`、`IR_LEADS`、`IR_LEADE`、`IR_SLEADE`、`IR_B0`、`IR_B1`、`IR_INT_MASK`。**注意：用户可以根据需要更新相应寄存器，如果不更新，则寄存器保持原值。**
- 步骤 5. 配置 `IR_START`。必须要等所有的 IR 控制寄存器都配置完成后，才能配置 `IR_START`，因为它被用来产生启动信号，只要对其进行配置，IR 模块就会根据控制寄存器的值进行红外数据接收。

----结束



图12-54 读取解码数据的操作流程



读取解码数据的操作流程如下：

步骤 1. 选中 IR 模块地址空间。

步骤 2. 中断或查询方式等待接收数据帧。

- 中断方式下，当 CPU 接收到 IR 模块的中断请求信号时，查询 [IR_INT_STATUS \[intms_rcv\]](#) 的值。若读取的值为 1 表明 IR 模块接收到一个数据帧，执行步骤 3；若读取的值为 0，重新执行步骤 2，继续等待中断。
- 查询方式下，软件不停（或每间隔一定时间）读取 [IR_INT_STATUS \[intrs_rcv\]](#) 的值，若读取的值为 1 表明 IR 模块接收到一个数据帧，执行步骤 3；若读取的值为 0 时，表明 IR 模块尚未接收到数据帧，重新执行步骤 2，继续查询。

步骤 3. 读取数据寄存器 [IR_DATAH](#)。（如果一帧内的数据位数不大于 32 位，可以省略此步骤）

步骤 4. 读取数据寄存器 [IR_DATA_L](#)。

步骤 5. 清除接收数据中断。

----结束

12.6.5 IR 寄存器概览

IR 寄存器概览如表 12-26 所示。



表12-26 IR 寄存器概览（基址是 0x120F_0000）

偏移地址	名称	描述	页码
0x000	IR_EN	IR 接收使能控制寄存器	12-119
0x004	IR_CFG	IR 配置寄存器	12-120
0x008	IR_LEADS	引导码起始位裕量配置寄存器（只在 IR_CFG [ir_mode]=0 时使用）	12-122
0x00C	IR_LEADE	引导码结束位裕量配置寄存器（只在 IR_CFG [ir_mode]=0 时使用）	12-122
0x010	IR_SLEADE	简化引导码结束位裕量配置寄存器（只在 IR_CFG [ir_mode]=0 时使用）	12-123
0x014	IR_B0	数据 0 的判断电平裕量配置寄存器（只在 IR_CFG [ir_mode]=0 时使用）	12-124
0x018	IR_B1	数据 1 的判断电平裕量配置寄存器（只在 IR_CFG [ir_mode]=0 时使用）	12-125
0x01C	IR_BUSY	配置忙标志寄存器	12-126
0x020	IR_DATAH	IR 接收解码数据的高 16 位寄存器（当 IR_CFG[ir_mode]=0 时）或 symbol FIFO 中的 symbol 个数寄存器（当 IR_CFG[ir_mode]=1 时）	12-126
0x024	IR_DATAH	IR 接收解码数据的低 32 位寄存器（当 IR_CFG[ir_mode]=0 时）或 IR 模块接收到的 symbol 宽度寄存器（当 IR_CFG[ir_mode]=1 时）	12-127
0x028	IR_INT_MASK	IR 中断屏蔽寄存器	12-127
0x02C	IR_INT_STATUS	IR 中断状态寄存器	12-129
0x030	IR_INT_CLR	IR 中断清除寄存器	12-131
0x034	IR_START	IR 启动配置寄存器	12-133

12.6.6 IR 寄存器描述

IR_EN

IR_EN 为 IR 接收使能控制寄存器。



注意

软件必须先配置寄存器 `IR_EN[ir_en]=1`，才能配置其他寄存器，否则配置无效。当寄存器 `IR_EN[ir_en]=0` 时，其他寄存器只可读不可写，且读出值为寄存器的复位值。

Offset Address: 0x000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留。	0x00000000
[0]	RW	ir_en	IR 接收模块的使能。 0: 关闭 IR 接收模块; 1: 打开 IR 接收模块。	0x0

IR_CFG

IR_CFG 为 IR 配置寄存器。



注意

必须在确保 `IR_BUSY[ir_busy]=0` 并且 `IR_EN[ir_en]=1` 时，才能配置此寄存器，否则配置无效，寄存器保持原值。

IR 支持的参考时钟频率为 1MHz~128MHz，其与分频因子 `ir_freq` 的对应关系是：

- 当参考时钟频率为 1MHz 时，分频因子 `ir_freq` 需配置为 0x00。
- 当参考时钟频率为 128MHz 时，分频因子 `ir_freq` 需配置为 0x7F。

当 IR 的参考时钟为 1MHz~128MHz 内的非整数倍频率时，选用四舍五入的方法选择相应的分频因子。举例：参考时钟为 12.1MHz，选用分频因子为 0x0B；参考时钟为 12.8MHz，选用分频因子为 0x0C。

对于频偏和计数偏差的关系：基频 f ，频率变化 Df ，则频偏率 $ratio=Df/f$ ；计数器计数偏差 $Dcnt$ ；判断电平宽度 s （ μs 为单位），则计数偏差： $Dcnt = \lceil 0.1 \times s \times ratio \rceil$ 。所以在时钟有频偏的情况下，参数值的有效范围要移位，如果频率上升，相应的裕量值应改为： $[min+Dcnt, max+Dcnt]$ ，其中 min 和 max 为无偏移时的裕量值；如果频率下降，相应的裕量值应改为： $[min-Dcnt, max-Dcnt]$ 。以引导码的起始位裕量举例来说：假如基频为 100MHz，频率上漂 0.1MHz，那么 $ratio=0.1/100=0.001$ ，设 $s=9000 \mu s$ ，则 $Dcnt = \lceil 0.1 \times 9000 \times 0.001 \rceil = 1$ ，则 `ir_leads` 的裕量值应改为 $[0x033D, 0x3CD]$ 。

Offset Address: 0x004 Total Reset Value: 0x3E80_1F0B



Bits	Access	Name	Description	Reset
[31:16]	RW	ir_max_level_width	当 IR_CFG [ir_mode]=0 时，无效； 当 IR_CFG [ir_mode]=1 时，表示 symbol 的最大电平宽度（单位 10 μ s），用以确定一个 symbol stream 结束。	0x3E80
[15:14]	RW	ir_format	当 IR_CFG [ir_mode]=0 时，表示数据码型。 00: NEC with simple repeat code 的数据格式； 01: TC9012 的数据格式； 10: NEC with full repeat code 的数据格式； 11: SONY 的数据格式。 关于具体码型属于哪类码族，请参见表 12-23~表 12-25。 当 IR_CFG [ir_mode]=1 时，表示 symbol 格式。 bit[15]: 保留； bit[14]的含义如下： 0: symbol 的格式为先低后高，symbol stream 结束在高电平； 1: symbol 的格式为先高后低，symbol stream 结束在低电平。	0x0
[13:8]	RW	ir_bits	当 IR_CFG [ir_mode]=0 时，表示一帧内的数据位数。 0x00~0x2F: 分别对应一帧内包含 1~48 个数据位； 0x30~0x3F: 保留。 如果软件对该域配置 0x30~0x3F 范围内的值，则配置无效，ir_bits 保持原值不变。 当 IR_CFG [ir_mode]=1 时，表示接收到 symbol 的中断水线。 bit[13]: 保留； bit[12:8]: 0x0~0x1F: 分别对应 FIFO 中至少有 1~32 个 symbol 时报中断。	0x1F
[7]	RW	ir_mode	IR 工作模式。 0: 输出解码后的完整数据帧； 1: 只输出 symbol 宽度。	0x0
[6:0]	RW	ir_freq	工作时钟分频因子。 0x00~0x7F 分别对应工作时钟分频因子	0xB



Bits	Access	Name	Description	Reset
			1~128。	

IR_LEADS

IR_LEADS 为引导码起始位裕量配置寄存器（只在表 12-23~表 12-25 [ir_mode]=0 时使用）。



注意

必须在确保 IR_BUSY[ir_busy]=0 并且 IR_EN[ir_en]=1 时，才能配置此寄存器，否则配置无效，寄存器保持原值；另外寄存器保留值配置无效，仍然保持原值不变。

为了准确判断引导码的起始位，需要在具体码型的典型值左右考虑一定的裕量，具体码型的典型值请参见表 12-23~表 12-25 中 LEAD_S 的值。

- 对于典型值不小于 400（其精度为 10 μs）的脉宽，建议裕量范围设为典型值的 8%。举例说明：D6121 码型，其 LEAD_S 的典型值为 900，那么相应的 cnt_leads_min=900 x 92%=828=0x33C，cnt_leads_max=900 x 108%=972=0x3CC。
- 对于典型值小于 400（其精度为 10 μs）的脉宽，建议裕量范围设为典型值的 20%。举例说明：SONY-D7C5 码型，其 LEAD_S 的典型值为 240，那么相应的 cnt_leads_min=240 x 80%=192=0xC0，cnt_leads_max=240 x 120%=288=0x120。

基本的配置原则：cnt_leads_max 不小于 cnt_leads_min，并且 cnt_leads_min 大于 cnt0_b_max 和 cnt1_b_max

Offset Address: 0x008 Total Reset Value: 0x033C_03CC

Bits	Access	Name	Description	Reset
[31:26]	RO	reserved	保留。	0x00
[25:16]	RW	cnt_leads_min	引导码起始位的最小脉宽。 0x000~0x007：保留。	0x33C
[15:10]	RO	reserved	保留。	0x00
[9:0]	RW	cnt_leads_max	引导码起始位的最大脉宽。 0x000~0x007：保留。	0x3CC

IR_LEADE

IR_LEADE 为引导码结束位裕量配置寄存器（只在 IR_CFG [ir_mode]=0 时使用）。



注意

- 必须在确保 **IR_BUSY** [ir_busy]=0 并且 **IR_EN** [ir_en]=1 时，才能配置此寄存器，否则配置无效，寄存器保持原值；另外寄存器保留值配置无效，仍然保持原值不变。
- 对于 NEC with simple repeat code 的码族，其 cnt_sleade 的裕量范围和 cnt_leade 的裕量范围设置不能重合，否则当实际计数值落入重合范围内时，无法识别简化引导码，会导致帧格式错误。

为了准确判断引导码的结束位，需要在具体码型的典型值左右考虑一定的裕量，裕量范围约为典型值的 8%。具体码型的典型值请参见表 12-23~表 12-25 中 LEAD_E 的值。

- 对于典型值不小于 400（其精度为 10 μs）的脉宽，建议裕量范围设为典型值的 8%。举例说明：D6121 码型，其 LEAD_E 的典型值为 450，那么相应的 cnt_leade_min=450 x 92%=414=0x19E，cnt_leade_max=450 x 108%=486=0x1E6。
- 对于典型值小于 400（其精度为 10 μs）的脉宽，建议裕量范围设为典型值的 20%。举例说明：SONY-D7C5 码型，其 LEAD_E 的典型值为 60，那么相应的 cnt_leade_min=60 x 80%=48=0x030，cnt_leade_max=60 x 120%=72=0x048。

基本的配置原则是：cnt_leade_max 不小于 cnt_leade_min 的值。

Offset Address: 0x00C Total Reset Value: 0x019E_01E6

Bits	Access	Name	Description	Reset
[31:25]	RO	reserved	保留。	0x00
[24:16]	RW	cnt_leade_min	引导码结束位的最小脉宽。 0x000~0x007：保留。	0x19E
[15:9]	RO	reserved	保留。	0x00
[8:0]	RW	cnt_leade_max	引导码结束位的最大脉宽。 0x000~0x007：保留。	0x1E6

IR_SLEADE

IR_SLEADE 为简化引导码结束位裕量配置寄存器（只在 **IR_CFG** [ir_mode]=0 时使用）。



注意

- 必须在确保 **IR_BUSY** [ir_busy]=0 并且 **IR_EN** [ir_en]=1 时，才能配置此寄存器，否则配置无效，寄存器保持原值；另外寄存器保留值配置无效，仍然保持原值不变。



- 对于 NEC with simple repeat code 的码族，cnt_sleade 的裕量范围和 cnt_leade 的裕量范围设置不能重合，否则当实际计数值落入重合范围内时，无法识别简化引导码，会导致帧格式错误。
- 对于 NEC with simple repeat code 的数据格式，才需配置此寄存器；对于其他格式，无需配置此寄存器。

为了准确判断简化引导码的结束位，需要在具体码型的典型值左右考虑一定的裕量。具体码型的典型值请参见表 12-23～表 12-25 中 SLEAD_E 的值。

- 对于典型值不小于 225（其精度为 10μs）的脉宽，建议裕量范围设为典型值的 8%。举例说明：D6121 码型，其 SLEAD_E 的典型值为 225，那么相应的 cnt_sleade_min=225 x 92%=207=0xCF，cnt_sleade_max=225 x 108%=243=0xF3。
- 对于典型值小于 225（其精度为 10μs）的脉宽，建议裕量范围设为典型值的 20%。举例说明：比如某种码型其 SLEAD_E 的典型值为 60，那么相应的 cnt_sleade_min=60 x 80%=48=0x30，cnt_sleade_max=60 x 120%=72=0x48。

基本的配置原则是：cnt_sleade_max 不小于 cnt_sleade_min 的值。

Offset Address: 0x010 Total Reset Value: 0x00CF_00F3

Bits	Access	Name	Description	Reset
[31:25]	RO	reserved	保留。	0x00
[24:16]	RW	cnt_sleade_min	简化引导码结束位的最小脉宽。 0x000～0x007：保留。	0x0CF
[15:9]	RO	reserved	保留。	0x00
[8:0]	RW	cnt_sleade_max	简化引导码起始位的最大脉宽。 0x000～0x007：保留。	0x0F3

IR_B0

IR_B0 为数据 0 的判断电平裕量配置寄存器（只在 IR_CFG [ir_mode]=0 时使用）。



注意

- 必须在确保 IR_BUSY [ir_busy]=0 并且 IR_EN [ir_en]=1 时，才能配置此寄存器，否则配置无效，寄存器保持原值；另外寄存器保留值配置无效，仍然保持原值不变。
- 对于四类码型，bit0 判断电平裕量范围和 bit1 判断电平范围设置不能重合，否则当实际计数值落入重合范围内时，无法识别 bit1，只能被误解接收到 bit0。

为了准确判断 bit0，需要在具体码型的典型值左右考虑一定的裕量，裕量范围约为典型值的 20%。



- 对于 NEC with simple repeat code、NEC with full repeat code 和 TC9012 这三类码型，其包含的具体码型的典型值请参见表 12-23~表 12-25 中 B0_H 的值。举例说明：D6121 码型，其 B0_H 的典型值为 56（其精度为 10μs），那么相应的 $\text{cnt0\_b\_min}=56 \times 80\%=45=0x2D$ ， $\text{cnt0\_b\_max}=56 \times 120\%=67=0x43$ 。
- 对于 SONY 的数据格式，其包含的具体码型的典型值请参见表 12-23~表 12-25 中 B0_L 的值。举例说明：SONY-D7C5 码型，其 B0_L 的典型值为 60（其精度为 10μs），那么相应的 $\text{cnt0\_b\_min}=60 \times 80\%=48=0x30$ ， $\text{cnt0\_b\_max}=60 \times 120\%=72=0x48$ 。

基本的配置原则是：cnt0_b_max 不小于 cnt0_b_min 的值。

Offset Address: 0x014 Total Reset Value: 0x002D_0043

Bits	Access	Name	Description	Reset
[31:25]	RO	reserved	保留。	0x00
[24:16]	RW	cnt0_b_min	bit0 判断电平的最小脉宽。 0x000~0x007：保留。	0x02D
[15:9]	RO	reserved	保留。	0x00
[8:0]	RW	cnt0_b_max	bit0 判断电平的最大脉宽。 0x000~0x007：保留。	0x043

IR_B1

IR_B1 为数据 1 的判断电平裕量配置寄存器（只在 IR_CFG [ir_mode]=0 时使用）。



注意

- 必须在确保 IR_BUSY [0]=0 并且 IR_EN [0]=1 时，才能配置此寄存器，否则配置无效，寄存器保持原值；另外寄存器保留值配置无效，仍然保持原值不变。
- 对于四类码型，bit0 判断电平裕量范围和 bit1 判断电平范围设置不能重合，否则当实际计数值落入重合范围内时，无法识别 bit1，只能被误解接收到 bit0。

为了准确判断 bit1，需要在具体码型的典型值左右考虑一定的裕量，裕量范围约为典型值的 20%。

- 对于 NEC with simple repeat code、NEC with full repeat code 和 TC9012 这三类码族，其包含的具体码型的典型值请参见表 12-23~表 12-25 中 B1_H 的值。举例说明：D6121 码型，其 B1_H 的典型值为 169（其精度为 10μs），那么相应的 $\text{cnt1\_b\_min}=169 \times 80\%=135=0x87$ ， $\text{cnt1\_b\_max}=169 \times 120\%=203=0xCB$ 。
- 对于 SONY 的数据格式，其包含的具体码型的典型值请参见表 12-23~表 12-25 中 B1_L 的值。举例说明：SONY-D7C5 码型，其 B1_L 的典型值为 120（其精度为



10μs)，那么相应的 cnt1_b_min=120 x 80%=96=0x60，cnt1_b_max=120 x 120%=144=0x90。

基本的配置原则是：cnt1_b_max 不小于 cnt1_b_min 的值。

Offset Address: 0x018 Total Reset Value: 0x0087_00CB

Bits	Access	Name	Description	Reset
[31:25]	RO	reserved	保留。	0x00
[24:16]	RW	cnt1_b_min	bit1 判断电平的最小脉宽。 0x000~0x007：保留。	0x087
[15:9]	RO	reserved	保留。	0x00
[8:0]	RW	cnt1_b_max	bit1 判断电平的最大脉宽。 0x000~0x007：保留。	0x0CB

IR_BUSY

IR_BUSY 为配置忙标志寄存器。

Offset Address: 0x01C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留。	0x00000000
[0]	RO	ir_busy	忙状态标志。 0：空闲状态，软件可以配置数据； 1：忙状态，软件不可以配置数据。	0x0

IR_DATAH

IR_DATAH 为 IR 接收解码数据的高 16 位寄存器（当 [IR_CFG](#) [ir_mode]=0 时）或 symbol FIFO 中的 symbol 个数寄存器（当 [IR_CFG](#) [ir_mode]=1 时）。

IR_DATAH 是接收到的解码数据的高 16 位，[IR_DATAH](#) 是接收到的解码数据的低 32 位。具体哪些数据位有效取决于具体码型一帧内包含的有效数据位数，请参见[表 12-23~表 12-25](#) 的有效数据位。

数据存储原则：按照由高到低的顺序存储在 IR_DATAH 和 [IR_DATAH](#) 中（MSB……LSB），先存满 [IR_DATAH](#)，然后再存放 IR_DATAH，未用到的高位作为保留位。软件读取数据的顺序必须是：先读 [IR_DATAH](#)，然后再读 [IR_DATAH](#)。

对于具体每个数据位表示的含义，硬件不做判断，仅负责接收所有数据位，最终由软件统一处理。

Offset Address: 0x020 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	保留。	0x0000
[15:0]	RO	ir_datah	当 IR_CFG [ir_mode]=0 时，表示接收到的解码数据的高 16 位数据。 当 IR_CFG [ir_mode]=1 时，表示 symbol FIFO 中的 symbol 个数。 bit[15:6]：保留； bit[5:0]：symbol FIFO 中的 symbol 个数。	0x0000

IR_DATA1

IR_DATA1 为 IR 接收解码数据的低 32 位寄存器（当 [IR_CFG](#) [ir_mode]=0 时）或，IR 模块接收到的 symbol 宽度寄存器（当 [IR_CFG](#) [ir_mode]=1 时）。

Offset Address: 0x024 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	ir_data1	当 IR_CFG [ir_mode]=0 时，表示接收到的解码数据的低 32 位数据。 当 IR_CFG [ir_mode]=1 时，表示 IR 模块接收到的 symbol 宽度。 bit[31:16]的含义如下： symbol 的格式为先低后高时，表示接收到的 symbol 的高电平宽度（单位是 10 μ s）； symbol 的格式为先高后低时，表示接收到的 symbol 的低电平宽度（单位是 10 μ s）。 bit[15:0]的含义如下： symbol 的格式为先低后高时，表示接收到的 symbol 的低电平宽度（单位是 10 μ s）； symbol 的格式为先高后低时，表示接收到的 symbol 的高电平宽度（单位是 10 μ s）。	0x0000_0000

IR_INT_MASK

IR_INT_MASK 为 IR 中断屏蔽寄存器。



注意

- 必须在确保 **IR_EN** [ir_en]=1 时，才能配置此寄存器，否则配置无效，寄存器保持原值。
- 如果中断全部屏蔽后，无法支持红外遥控唤醒功能。
- **IR_CFG** [ir_mode]=0 时，**IR_INT_MASK** bit[3:0]有效；**IR_CFG** [ir_mode]=1 时，**IR_INT_MASK** bit[18:16]有效。

涉及到的中断定义如下：

- 接收数据溢出中断
如果 CPU 没有及时响应取走当前帧的数据，而下一帧数据也已经收到的情况，下一帧数据将会覆盖当前帧数据，同时上报屏蔽前接收数据溢出错中断请求。
- 接收数据帧格式错误中断
如果接收到的数据帧不完整以及数据脉宽不满足裕量范围，则会上报屏蔽前的接收帧格式错误中断请求。
- 接收到数据帧中断
当接收到一个完整的帧数据后，则会上报屏蔽前接收到数据帧中断请求。
- 支持按键释放的检测中断
对于 NEC with simple repeat code 和 TC9012 码族的数据格式，在检测到一个有效起始同步码之后的 160ms 内，如果没有再次检测到起始同步码，或者检测到非简化引导码而是有效数据帧时，则会上报屏蔽前遥控器按键释放中断。对于 NEC with full repeat code 和 SONY 两种码制不支持按键释放中断。
- 接收 symbol 溢出中断
如果 CPU 没有及时响应取走数据，导致 symbol FIFO 满，而下一个 symbol 已经收到，则会上报屏蔽前接收 symbol 溢出错中断请求。
- 接收到 symbol 中断
当接收到一个完整的 symbol 后，且 symbol FIFO 中的 symbol 个数超过 **IR_CFG** [ir_bits]设置的水线，则会上报屏蔽前接收到 symbol 中断请求。
- symbol 超时中断
在接收到一个有效的 symbol 后，**IR_CFG** [ir_max_level_width]设置的时间内没有再接收到新的 symbol 的中断请求，则会上报屏蔽前 symbol 超时中断请求。

硬件没有中断优先级仲裁，任何一个或多个屏蔽后的中断源有效，都会产生中断。

Offset Address: 0x028 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:19]	RO	reserved	保留。	0x0000
[18]	RW	intm_overrun	当 IR_CFG [ir_mode]=1 时，symbol 溢出中断屏蔽。 0：不屏蔽； 1：屏蔽。	0x0



Bits	Access	Name	Description	Reset
[17]	RW	intm_time_out	当 IR_CFG [ir_mode]=1 时，symbol 超时中断屏蔽。 0：不屏蔽； 1：屏蔽。	0x0
[16]	RW	intm_symb_rcv	当 IR_CFG [ir_mode]=1 时，接收到 N 个 symbol 中断屏蔽。 0：不屏蔽； 1：屏蔽。	0x0
[15:4]	RO	reserved	保留。	0x0000
[3]	RW	intm_release	当 IR_CFG [ir_mode]=0 时，按键释放中断屏蔽。 0：不屏蔽； 1：屏蔽。	0x0
[2]	RW	intm_overflow	当 IR_CFG [ir_mode]=0 时，接收数据溢出中断屏蔽。 0：不屏蔽； 1：屏蔽。	0x0
[1]	RW	intm_frame	当 IR_CFG [ir_mode]=0 时，接收数据帧格式错误中断屏蔽。 0：不屏蔽； 1：屏蔽。	0x0
[0]	RW	intm_rcv	当 IR_CFG [ir_mode]=0 时，接收到数据帧中断屏蔽。 0：不屏蔽； 1：屏蔽。	0x0

IR_INT_STATUS

IR_INT_STATUS 为 IR 中断状态寄存器。



注意

- [IR_CFG](#) [ir_mode]=0 时，IR_INT_STATUS bit[3:0]和 IR_INT_STATUS bit[19:16]有效；



- **IR_CFG** [ir_mode]=1 时，IR_INT_STATUS bit[10:8]和 IR_INT_STATUS bit[26:24]有效。

Offset Address: 0x02C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:27]	RO	reserved	保留。	0x00
[26]	RO	intms_overrun	当 IR_CFG [ir_mode]=1 时，屏蔽后的 symbol 溢出中断状态。 0: 无中断; 1: 有中断。	0x0
[25]	RO	intms_time_out	当 IR_CFG [ir_mode]=1 时，屏蔽后的 symbol 超时中断状态。 0: 无中断; 1: 有中断。	0x0
[24]	RO	intms_symb_rcv	当 IR_CFG [ir_mode]=1 时，屏蔽后的接收到 symbol 的中断状态。 0: 无中断; 1: 有中断。	0x0
[23:20]	RO	reserved	保留。	0x0
[19]	RO	intms_release	当 IR_CFG [ir_mode]=0 时，屏蔽后的按键释放的中断状态。 0: 无中断; 1: 有中断。	0x0
[18]	RO	intms_overflow	当 IR_CFG [ir_mode]=0 时，屏蔽后的接收数据溢出错中断状态。 0: 无中断; 1: 有中断。	0x0
[17]	RO	intms_framerr	当 IR_CFG [ir_mode]=0 时，屏蔽后的接收数据帧格式错误中断状态。 0: 无中断; 1: 有中断。	0x0
[16]	RO	intms_rcv	当 IR_CFG [ir_mode]=0 时，屏蔽后的接收到数据帧中断状态。 0: 无中断; 1: 有中断。	0x0
[15:11]	RO	reserved	保留。	0x0
[10]	RO	intrs_overrun	当 IR_CFG [ir_mode]=1 时，屏蔽前的	0x0



Bits	Access	Name	Description	Reset
			symbol 溢出中断状态。 0: 无中断; 1: 有中断。	
[9]	RO	intrs_time_out	当 IR_CFG [ir_mode]=1 时, 屏蔽前的 symbol 超时中断状态。 0: 无中断; 1: 有中断。	0x0
[8]	RO	intrs_symb_rcv	当 IR_CFG [ir_mode]=1 时, 屏蔽前的接收到 symbol 的中断状态。 0: 无中断; 1: 有中断。	0x0
[7:4]	RO	reserved	保留。	0x0
[3]	RO	intrs_release	当 IR_CFG [ir_mode]=0 时, 屏蔽前的按键释放的中断状态。 0: 无中断; 1: 有中断。	0x0
[2]	RO	intrs_overflow	当 IR_CFG [ir_mode]=0 时, 屏蔽前的接收数据溢出错中断状态。 0: 无中断; 1: 有中断。	0x0
[1]	RO	intrs_framerr	当 IR_CFG [ir_mode]=0 时, 屏蔽前的接收数据帧格式错误中断状态。 0: 无中断; 1: 有中断。	0x0
[0]	RO	intrs_rcv	当 IR_CFG [ir_mode]=0 时, 屏蔽前的接收到数据帧中断状态。 0: 无中断; 1: 有中断。	0x0

IR_INT_CLR

IR_INT_CLR 为 IR 中断清除寄存器。



注意

- [IR_CFG](#) [ir_mode]=0 时，IR_INT_CLR bit[3:0]有效；
- [IR_CFG](#) [ir_mode]=1 时，IR_INT_CLR bit[18:16]有效。

Offset Address: 0x030 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:19]	RO	reserved	保留。	0x0
[18]	WC	intc_overflow	IR_CFG [ir_mode]=1 时，清除 symbol 溢出中断请求。 0: 无影响； 1: 清除。	0x0
[17]	WC	intc_time_out	IR_CFG [ir_mode]=1 时，清除 symbol 超时中断请求。 0: 无影响； 1: 清除。	0x0
[16]	WC	intc_symb_rcv	IR_CFG [ir_mode]=1 时，清除接收到 symbol 中断请求。 0: 无影响； 1: 清除。	0x0
[15:4]	RO	reserved	保留。	0x0000
[3]	WC	intc_release	IR_CFG [ir_mode]=0 时，清除遥控器按键释放中断请求。 0: 无影响； 1: 清除。	0x0
[2]	WC	intc_overflow	IR_CFG [ir_mode]=0 时，清除接收数据溢出错误中断请求。 0: 无影响； 1: 清除。	0x0
[1]	WC	intc_framerr	IR_CFG [ir_mode]=0 时，清除接收数据帧格式错误中断请求。 0: 无影响； 1: 清除。	0x0
[0]	WC	intc_rcv	IR_CFG [ir_mode]=0 时，清除接收到数据帧中断请求。 0: 无影响； 1: 清除。	0x0



Bits	Access	Name	Description	Reset
			如果接收数据帧中断请求产生后，软件未读走 IR_DATAL 中的数据就直接对本位进行写 1 操作，无法清除该中断请求。	

IR_START

IR_START 为 IR 启动配置寄存器。

在其他寄存器的值配置完成后，启动 IR 模块时，只要往该地址进行一次写操作（写操作数可以为任意值），就可以启动配置寄存器。

Offset Address: 0x034 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留。	0x00000000
[0]	WO	ir_start	IR 启动配置寄存器。	0x0

12.7 GPIO

12.7.1 概述

芯片支持 12 组 GPIO（General Purpose Input/Output），即 GPIO0~GPIO11。每组 GPIO 提供 8 个可编程的输入输出管脚（GPIO11 只有 4 个）。

每个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。



注意

- GPIO 具体管脚个数、管脚与其他管脚复用及相关复用控制的说明请参见 Hi35xxVxxx_PINOUT_CN.xlsx（Hi35xxVxxx 表示具体项目名称）。
- 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。
- 当使用 GPIO11 时，需要确保 RTC 处于工作模式（即 32768Hz 的 RTC 晶体必须接上）和 PMC 处于工作模式（即 PMC 的复位输入为高电平），还需要把 PWR_CTRL0[gpio_at_pmc_enable]寄存器配置为 1（寄存器详情请参见 3.10.2.3 小节）。
- 当使用 GPIO9、GPIO10_0 和 GPIO10_1 时，需要先配置 PHY_RSTZ 和 PHY_TST_CTRL0 寄存器分别为 0x0 和 0x1（寄存器详情请参见 9.4.6 小节），使 MIPI TX PHY 处于关闭模式。

12.7.2 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。

12.7.3 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



注意

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“3.3 中断系统”。

GPIO_IS、**GPIO_IBE**、**GPIO_IIEV** 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_IIEV**，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



注意

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。

12.7.4 GPIO 寄存器概览

芯片的 GPIO 基地址如表 12-27 所示。

表12-27 芯片 GPIO 寄存器对应的基地址

GPIO 控制器	基地址
GPIO11	0x120D_B000
GPIO10	0x120D_A000
GPIO9	0x120D_9000



GPIO 控制器	基地址
GPIO8	0x120D_8000
GPIO7	0x120D_7000
GPIO6	0x120D_6000
GPIO5	0x120D_5000
GPIO4	0x120D_4000
GPIO3	0x120D_3000
GPIO2	0x120D_2000
GPIO1	0x120D_1000
GPIO0	0x120D_0000

 说明
GPIO_n 对应的寄存器地址为：GPIO_n 基地址+该寄存器偏移地址。

表 12-28 是单组 GPIO 内部寄存器的偏移地址以及定义，GPIO0~GPIO_n 具有相同的寄存器组。

表12-28 GPIO 寄存器概览

偏移地址	名称	描述	页码
0x000~0x3FC	GPIO_DATA	GPIO 数据寄存器	12-136
0x400	GPIO_DIR	GPIO 方向控制寄存器	12-137
0x404	GPIO_IS	GPIO 中断触发寄存器	12-137
0x408	GPIO_IBE	GPIO 双沿触发中断寄存器	12-137
0x40C	GPIO_IEV	GPIO 触发中断条件寄存器	12-138
0x410	GPIO_IE	GPIO 中断屏蔽寄存器	12-138
0x414	GPIO_RIS	GPIO 原始中断状态寄存器	12-138
0x418	GPIO_MIS	GPIO 屏蔽状态中断寄存器	12-139
0x41C	GPIO_IC	GPIO 中断清除寄存器	12-139

12.7.5 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。



注意

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。 0：输入； 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。 0：边沿触发中断； 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。 0：单边沿触发中断，具体是上升沿还是下降沿触发由 GPIO_IEV 控制； 1：双边触发中断。	0x00

GPIO_IEV

GPIO_IEV 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_iev	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。 0：下降沿或低电平触发中断； 1：上升沿或高电平触发中断。	0x00

GPIO_IE

GPIO_IE 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。	0x00



			0: 屏蔽中断; 1: 不屏蔽中断。	

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。 0: 未发生中断; 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。 0: 中断无效; 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。 0: 无影响;	0x00



			1: 清除中断。	

12.8 USB DRD

12.8.1 概述

该系统支持 1 个 USB2.0 DRD (Dule Role Device)模块。

USB2.0 DRD 支持 USB2.0 Host 和 USB2.0 Device 的静态操作。

只需要配置相关寄存器就能使该模块工作在 Host 或者 Device 模式间切换；

USB2.0 部分支持 USB2.0 协议规定的 480Mbit/s 传输速率以及高速传输的 UTMI 接口协议；USB2.0 部分还支持 XHCI 1.0 协议；USB2.0 内部集成了一个 Root Hub，可以扩展 USB 接口或者其他 Hub；

两个 USB DRD 控制器均可以完成：

- 完成对传输的控制和处理
- 对数据包的解析和打包
- 对 USB 传输信号的编码和解码
- 为驱动程序提供了中断向量等接口

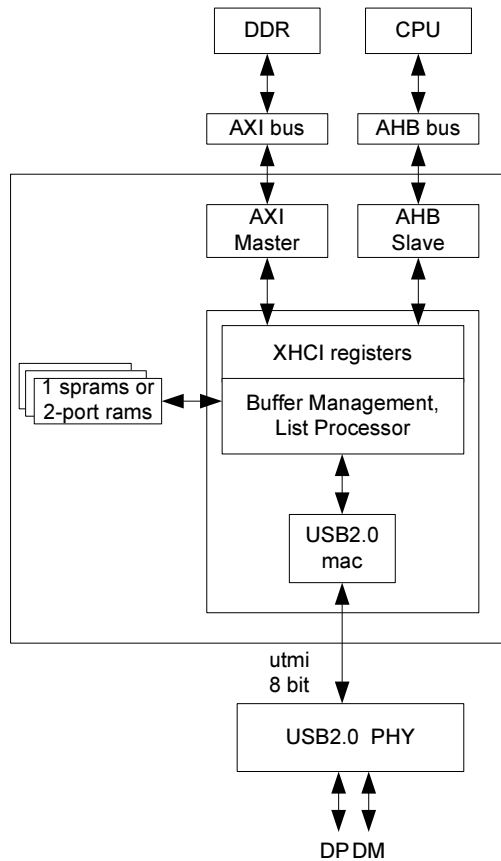
12.8.2 功能描述

逻辑框图

USB 2.0 DRD 逻辑框图如图 12-55 所示。



图12-55 USB 2.0 DRD 逻辑框图



UTMI: USB2.0 Transceiver Macrocell Interface
XHCI: eXtensible Host Controller 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。
- 可单独工作在 Host 或者 Device 模式。
- 可以支持、High-speed、Full-speed、Low-speed（Device 模式不支持）四种设备。
- 支持 USB 2.0 低功耗的解决方案。
- 支持 Host 工作模式下 Control Transfer、Bulk Transfer、Isochronous Transfer、Interrupt Transfer 四种基本数据传输类型。支持 Device 工作模式下的 Control Transfer、Bulk Transfer、Isochronous Transfer 三种基本数据传输类型。
- 支持内部 DMA 控制器。
- 可以通过连接 USB Hub，连接最多 127 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，某些特定型号的 USB 设备的控制传输可能用到其他的端点。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，依赖于设备和传输速度。
- **Bulk Transfer（批量传输）**
主要用于没有带宽和间隔时间要求的情况下发送和接收大量的数据，这种类型的设备适合于传输非常慢和大量被延迟的传输，可以等到所有其他类型的数据的传送完成之后再传送和接收数据。它的特点是以错误检测和重传的方式保证 USB Host 与 USB Device 的数据被无差错地发送。
- **Isochronous Transfer（同步传输）**
主要用于时间严格并具有较强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。

当 USB2.0 DRD 工作于 Device 工作模式时，支持 Control Transfer（控制传输）、Bulk Transfer（批量传输）和 Isochronous Transfer（同步传输），不支持 Interrupt Transfer（中断传输）。

12.8.3 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[2]写入 0x1，打开 U2PHY 的参考时钟
- 步骤 2. 向 PERI_CRG80[0]写入 0x0，撤离 U2PHY 的复位。
- 步骤 3. 向 PERI_CRG80[1]写入 0x0，撤离 U2PHY 的数字复位；
- 步骤 4. 向 PERI_CRG80[3]写入 0x0，撤离 U2 控制器的复位。

----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。



注意

Host/Device 工作模式切换只支持静态切换，不支持动态切换。

12.8.4 寄存器概览

USB 寄存器概览如表 12-29 所示。

表12-29 USB 寄存器概览（USB2.0 基址是 0x100E_0000）

偏移地址	名称	描述	页码
0xC100	PERI_USB2_GSBUS_CFG0	全局 SOC 总线配置寄存器 0	12-144
0xC104	PERI_USB2_GSBUS_CFG1	全局 SOC 总线配置寄存器 1	12-146
0xC108	PERI_USB2_GTXT_HRCFG	全局发送门限控制寄存器	12-146
0xC10C	PERI_USB2_GRXT_HRCFG	全局接收门限控制寄存器	12-147
0xC110	PERI_USB2_GCTL	全局 core 控制寄存器	12-148
0xC118	PERI_USB2_GSTS	全局状态寄存器	12-151
0xC11C	PERI_USB2_GUCTL_1	全局用户控制寄存器 1	12-152
0xC124	PERI_USB2_GGPIO	全局 GPIO 寄存器	12-153
0xC128	PERI_USB2_GUID	全局用户 ID 寄存器	12-153
0xC12C	PERI_USB2_GUCTL	全局用户控制寄存器	12-153
0xC130	PERI_USB2_GBUSE_RRADDR_HI	全局总线错误地址高 32 位寄存器	12-155
0xC134	PERI_USB2_GBUSE_RRADDR_LO	全局总线错误地址低 32 位寄存器	12-155
0xC138	PERI_USB2_GPRTB_IMAP_HI	超速端口-总线映射高 32 位寄存器	12-156
0xC13C	PERI_USB2_GPRTB_IMAP_LO	超速端口-总线映射低 32 位寄存器	12-156
0xC180	PERI_USB2_GPRTB_IMAP_HS_HI	高速端口-总线映射高 32 位寄存器	12-156
0xC184	PERI_USB2_GPRTB_IMAP_HS_LO	高速端口-总线映射低 32 位寄存器	12-156



偏移地址	名称	描述	页码
0xC188	PERI_USB2_GPRTB IMAP_FS_HI	全速端口-总线映射高 32 位寄存器	12-156
0xC18C	PERI_USB2_GPRTB IMAP_FS	全速端口-总线映射低 32 位寄存器	12-157
0xC200	PERI_USB2_GUSB2 PHYCFGN	全局 USB2.0 PHY 配置寄存器	12-157
0xC304	PERI_USB2_GTXFI FOSIZN	全局发送 FIFO 大小寄存器	12-159
0xC384	PERI_USB2_GRXFI FOSIZN	全局接收 FIFO 大小寄存器	12-159
0xC410	PERI_USB2_GEVN TADR_N_HI	全局 event buffer 的地址高 32 位寄存器	12-160
0xC414	PERI_USB2_GEVN TADR_N_LO	全局 event buffer 的地址低 32 位寄存器	12-160
0xC418	PERI_USB2_GEVN TSIZN	全局 event buffer 的大小寄存器	12-160
0xC41C	PERI_USB2_GEVN TCOUNTN	全局 event buffer 的计数寄存器	12-160
0xC610	PERI_USB2_GTXFI FOPRIDEV	外设的全局 TX FIFO DMA 优先寄存器	12-161
0xC618	PERI_USB2_GTXFI FOPRIHST	host 的全局 TX FIFO DMA 优先寄存器	12-161
0xC61C	PERI_USB2_GRXFI FOPRIHST	host 的全局 RX FIFO DMA 优先寄存器	12-161
0xC620	PERI_USB2_GFIFO PRIDBC	host 的全局 Debug 性能时 DMA 优先寄存器	12-162
0xC624	PERI_USB2_GDMA HLRATIO	host 的全局 FIFO DMA 高、低优先权比例寄存器	12-162
0xC630	PERI_USB2_GFLADJ	GFLADJ 为全局帧长度调整寄存器	12-162
0xC700	PERI_USB2_DCFG	外设配置寄存器	12-163
0xC704	PERI_USB2_DCTL	外设控制寄存器	12-164
0xC708	PERI_USB2_DEVTE N	外设事件使能寄存器	12-168
0xC70C	PERI_USB2_DSTS	外设状态寄存器	12-169
0xC710	PERI_USB2_DGCM DPAR	外设类命令参数寄存器	12-170



偏移地址	名称	描述	页码
0xC714	PERI_USB2_DGCM D	外设类命令寄存器	12-171
0xC718	PERI_USB2_DALEP ENA	外设 USB 端点使能寄存器	12-172
0xC810	PERI_USB2_DEPC MDPAR2N	外设端点命令参数寄存器 2	12-172
0xC814	PERI_USB2_DEPC MDPAR1N	外设端点命令参数寄存器 1	12-172
0xC818	PERI_USB2_DEPC MDPAR0N	外设端点命令参数寄存器 0	12-173
0xC81C	PERI_USB2_DEPC MDN	外设物理端点命令寄存器	12-173

12.8.5 USB 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。 0: 小端; 1: 大端。	0x0
[10]	RW	descbigend	链表存取大小端选择。 0: 小端; 1: 大端。	0x0



Bits	Access	Name	Description	Reset
[9:8]	-	reserved	保留。	0x0
[7]	RW	incr256brstena	AHB master INCR 进行 128 beat burst 传输使能信号。 0: 不使能; 1: 使能。	0x0
[6]	RW	incr128brstena	AHB master INCR 进行 128 beat burst 传输使能信号。 0: 不使能; 1: 使能。	0x0
[5]	RW	incr64brstena	AHB master INCR 进行 64 beat burst 传输使能信号。 0: 不使能; 1: 使能。	0x0
[4]	RW	incr32brstena	AHB master INCR 进行 32 beat burst 传输使能信号。 0: 不使能; 1: 使能。	0x0
[3]	RW	incr16brstena	AHB master INCR 进行 16 beat burst 传输使能信号。 0: 不使能; 1: 使能。	0x0
[2]	RW	incr8brstena	AHB master INCR 进行 8 beat burst 传输使能信号。 0: 不使能; 1: 使能。	0x0
[1]	RW	incr4brstena	AHB master INCR 进行 4 beat burst 传输使能信号。 0: 不使能; 1: 使能。	0x0
[0]	RW	incrbrstena	AHB master INCR 进行 1 beat burst 传输使能信号。 0: 不使能; 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。 0: 4K Byte 边界; 1: 1K Byte 边界。	0x0
[11:8]	RW	pipetranslimit	AXI master outstanding 请求数目。 0x0: 1 个请求; 0x1: 2 个请求; 0x2: 3 个请求; 0x3: 4 个请求; ... 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。 0: USB 只在 全部的包被读取到既定的 TXFIFO 后, 才开始进行传输; 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 以内。	0x0
[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	0x00



Bits	Access	Name	Description	Reset
			在 1-16 之间。	
[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。 0: USB 只在 全部的包被读取到既定的 RXFIFO 后，才开始进行传输； 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	pwrndnscale	Suspend_clk 设置，GCTL[31:19] * 16K = Suspend_clk 说明：32kHz<Suspend_clk<125MHz。	0x0000
[18]	RW	masterfiltbypass	滤波功能选择。 0: 当 DWC_USB2_EN_BUS_FILTERS 值	0x0



Bits	Access	Name	Description	Reset
			为 1 时，选择滤波功能； 1: 无论当 DWC_USB2_EN_BUS_FILTERS 值为多少，关闭滤波功能。	
[17]	RW	bypssetaddr	Device 模式下 SetAddress 命令选择。 0: Host 正常发送 SetAddress 命令给 Device; 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。 说明：只在仿真时设置此 bit 位。	0x0
[16]	RW	u2rstecn	超时连接选择。 0: 当超速连接失败时，设备处于 HS 模式; 1: 当超速连接失败时，Device 将花费大于 3 个 cycle 的时间等待连接。 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： 00: 125us; 01: 62.5us; 10: 31.25us; 11: 15.625us; 其他：保留。 对于 FS 模式，将上述值*8 即可。 当选择 xHCI Debug 模式时，可配置仿真时 bulk in 和 bulkout 传输的 MaxPacketSize, 00: 1024 bytes; 01: 512 bytes; 10: 256 bytes; 11: 128bytes; 其他：保留。	0x0
[13:12]	RW	prtcapdir	端口配置类型。 00: 保留; 01: Host 配置; 10: Device 配置; 11: Otg 配置。	0x0



Bits	Access	Name	Description	Reset
[11]	RW	coresoftreset	Core 软复位选择。 0: 不进行软复位; 1: 对 core 软复位。 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: -GCTL -GUCTL -GSTS -GSNPSID -GGPIO -GUID -GUSB2PHYCFGn -DCFG -DCTL -DEVTEN -DSTS	0x0
[10]	RW	sofitpsync	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; 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。 说明: 此 bit 只在控制器工作于 Host 模式时有效。	0x0
[9]	RW	u1u2timerscale	U1/U2 timer scaledown 选择。 0: 不关闭; 1: 如果 PERI_USB2_GCTL[5:4] (ScaleDown) = X1, 则关闭 U1/U2 的反应时间的 scaledown。	0x0
[8]	RW	debugattach	Debug Attach 信号 当此位被置 1 时 -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 -Link LFPS polling 的超时 时间有限 -TS1 的 Polling 超时时间有限。	0x0
[7:6]	RW	ramclksel	RAM Clock 选择。 00: bus clock;	0x0



Bits	Access	Name	Description	Reset
			01: pipe clock; 10: pipe/2 clock; 11: reserved。 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。	
[5:4]	RW	scaledown	scale-down timing 选择。 HS/FS/LS 模式下： 00: 关闭掉所有的 scale-down timing，使用实际的 timing 进行仿真； 01: 启用除以下功能外地所有的 scale-down timing： -speed enumeration -HNP/SRP -Host 模式 的 suspend 和 resume； 10: 仅仅开启 device 模式时 suspend 和 resume 功能时的 scale-down timing； 11: 打开所有的 scale-down timing。 SS 模式下 HS/FS/LS 模式下 00: 关闭掉所有的 scale-down timing，使用实际的 timing 进行仿真； 01: 开启 SS 的 scale-down timing，包括： -减少 TxEq training sequences 到 8 -减少 LFPS polling burst time 到 100ns -减少 LFPS warm reset receive 到 30us 10: 不发送 TxEq training sequences； 11: 打开所有的 scale-down timing。	0x0
[3]	RW	dissscrumble	关闭加扰功能。 1: 关闭； 0: 不关闭。	0x0
[2]	RW	u2exit_lfps	U2 状态退出信号。 0: link 将把 248ns 的 LFPS 信号当做有效的 U2 退出状态信号； 1: link 层在检测到有效的 U2 退出信号前等待 8us。	0x0



Bits	Access	Name	Description	Reset
[1]	RW	gblhibernationen	休眠使能。 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; 1: 打开全局休眠功能。	0x0
[0]	RW	dsblclkgtng	内部 Clock Gating 选择。 0: 选择内部 clock gating; 1: 当 core 处于 lpm 模式时, 关闭掉内部 clock gating。 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
[19:11]	-	reserved	保留。	0x000
[10]	RO	otg_ip	指示 OEVT 寄存器中有一个与 OTG 有关的中断等待处理。 0: 无指示; 1: 有指示。	0x0
[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。 0: 无指示; 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。 0: 无指示; 1: 有指示。	0x0
[7]	RO	host_ip	指示 Host event queue 中有一个与 xHCI 有关的中断等待处理。 0: 无指示;	0x0



Bits	Access	Name	Description	Reset
			1: 有指示。	
[6]	RO	device_ip	指示 Device event queue 中有一个与 xHCI 有关的中断等待处理。 0: 无指示; 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。 0: 无指示; 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。 0: 无指示; 1: 有指示。	0x0
[3:2]	-	reserved	保留。	0x0
[1:0]	RO	curmod	当前的工作模式: 00: Device 模式; 01: Host 模式; 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
[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： 系统版本； 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 例如：ref_clk=125MHz,则此处为 1/125MHz=8ns。	0x000
[21]	RW	noextrdl_	SOF 包和第一个包之间的额外延时选择。 0: Host 在 SOF 包之后等待 2ms 再发送第一个包； 1: Host 在 SOF 之后不延时直接发送第一个包。	0x0
[20:18]	-	reserved	保留	0x0
[17]	RW	sprscrtltransen	分散控制传输使能。 0: 禁止； 1: 使能。 一些 Device 对控制传输的响应很慢，在 1	0x0



Bits	Access	Name	Description	Reset
			帧/微帧内进行多次传输的话会导致 Device 行为紊乱。当此位被置 1 时，host 控制器将把一个控制传输分散在不同的帧或者微帧中。	
[16]	RW	resbwhseps	保留 85%的带宽给高速周期性端点，只在 HOST 模式或者 DRD 模式下的 Host 操作模式下有效。	0x0
[15]	RW	cmdevaddr	Device 地址的模式。 1: 根据每个设备分配地址命令递增； 0: 设备地址等于 Slot ID。	0x0
[14]	RW	usbhstlnautoretryen	Host 输入传输自动重传使能。 0: 自动重传功能关闭； 如果 host 的输入传输发生错误，host 变自动回复给 device 一个终结的 ACK(Retry = 1 and NumP = 0)； 1: 自动重传功能开启。 当自动重传功能开启后，如果 host 的输入传输发生错误，host 变自动回复给 device 一个不是终结的 ACK(Retry = 1 and NumP != 0)。	0x0
[13]	RW	enoverlapchk	LFPS 叠加信号检测使能。 0: 不检测 LFPS 叠加信号； 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。 0: 不插入； 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： 00: 0us； 01: 500us； 10: 1.5ms； 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，	0x000



Bits	Access	Name	Description	Reset
			T=DTFT*256*8 us。	

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。 说明：只在 GSTS.BusErrAddrVld 为 1 时有效 只在复位时清 0，只支持 AHB 和 AXI 的总线配置。	0x00000000

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。 说明：只在 GSTS.BusErrAddrVld 为 1 时有效 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。	0x0000000
[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]	RW	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。	0x0000000
[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。	0x0000000
[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。 0: 不提供 free clock; 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。 0: PHY 使用内部 Vbus 作为比较电压; 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。 0: PHY 用内部 VBUS 电压驱动; 1: PHY 用外部 VBUS 电压驱动。	0x0
[16]	RO	reserved	保留。	0x0
[15]	RW	ulpi_auto_res	ULPI 自动唤醒。 0: PHY 不启用自动唤醒功能; 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	0x0



Bits	Access	Name	Description	Reset
			请求 Packet FIFO Controller (PFC)从 DFIFO (SPRAM)取回数据的响应时间。 当 16-bit UTMI+ 时: 0x5 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。 0: utmi_sleep_n 和 utmi_l1_suspend_n 信号不接 PHY; 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。 0: USB2.0 高速 UTMI+ 或者 ULPI PHY; 1: USB1.1 全速串行接口; 当作为只写时, 此 bit 为 1。	0x0
[6]	RW	susphy	USB2.0 HS/FS/LS PHY 挂起选择。 0: 不挂起; 1: 挂起。 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。 0: 6-pin 单向全速串行传输接口; 1: 3-pin 双向全速串行传输接口。 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。 0: UTMI+ ; 1: ULPI。	0x0
[3]	RW	phyif	UTMI 接口数据位宽选择。 0: 8bits; 1: 16bits。	0x0
[2:0]	RW	toutcal	HS/FS Timeout 校准。 每个 PHY clock 加上相应的 bit time	0x0



Bits	Access	Name	Description	Reset
			High-speed 模式: One 30-MHz PHY clock = 16 bit times One 60-MHz PHY clock = 8 bit times Full-speed 模式: One 30-MHz PHY clock = 0.4 bit times One 60-MHz PHY clock = 0.2 bit times One 48-MHz PHY clock = 0.25 bit times	

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 在 memory 中的起始地址。	0x0000
[15:0]	RW	txfdep_n	Transmit FIFO 深度。 最小: 32 MDWIDTH-bit words; 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 在 memory 中的起始地址。	0x0000
[15:0]	RW	rxfdep_n	Receive FIFO 深度。 最小: 32 MDWIDTH-bit words; 最大: 32768 MDWIDTH-bit words。	0x0000

PERI_USB2_GEVNTADR_N_HI

PERI_USB2_GEVNTADR_N_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。 0: low; 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。 0: low; 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。 0: low; 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。 00: Low; 01: Normal; 10: High; 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
[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。 0: 余数小于 0.5; 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: GFLADJ_REFCLK_240MHZ_DECR = 240/ref_clk_frequency。	0x00
[23]	RW	gfladj_refclk_lpm_sel	SOF/ITP 计数时钟选择。 若此位被置 1, 则 SOF/ITP 已 ref_clk 为时钟计数。	0x0
[22]	-	reserved	保留	0x0



[21:8]	RW	gfladj_refclk_fladj	当 bit[23]为 1 时，SOF/ITP 校准值 $FLADJ\_REF\_CLK\_FLADJ = ((125000/ref\_clk\_period\_integer) - (125000/ref\_clk\_period)) * ref\_clk\_period$ 。	0x0000
[7]	RW	gfladj_30mhz_reg_sel	SOF/ITP 校准选择信号。 0: 控制器用 fladj_30mhz_reg 的值来对 SOF/ITP 进行校准; 1: 控制器用 GFLADJ.GFLADJ_30MHZ 的值来对 SOF/ITP 进行校准。	0x0
[6]	-	reserved	保留。	0x0
[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。 0:不忽略; 1: 忽略。	0x0
[22]	RW	lpm_capable	LPM 功能选择。 0: LPM 功能开启; 1: LPM 功能关闭。	0x0
[21:17]	RW	number_of_receive_buffers	接收 Buffer 号。 指示 ACK TP 中返回的接收 buffer 号。	0x04
[16:12]	RW	interrupt_number	中断号指示。 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。 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。 000: 高速; 001: 全速; 100: 超速; 其他: 保留。	0x4

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	设备开启/停止选择。 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。 Run/stop 比特的必须在如下的情况下操作: 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; 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; 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。	0x0
[30]	RWSC	soft_core_reset	core 软复位。 将对所有时钟域的中断和寄存器做如下操作: 1: 清除所有中断和复位出一下寄存器以外的所有寄存器; - GCTL - GUCTL - GSTS - GSNPSID - GGPI0 - GUID - GUSB2PHYCFGn registers	0x0



Bits	Access	Name	Description	Reset
			- GUSB2PIPECTLn registers - DCFG - DCTL - DEVTEN - DSTS 2: 所有状态机将回归 IDLE 状态, 所有的 FIFO 将清 0; 3: SoC 总线上的所有传输将立刻终止。	
[29]	-	reserved	保留。	0x0
[28:24]	RW	hird_threshold	HIRD 门限值 core 将在满足以下的条件下产生 utmi_l1_suspend_n 和 utmi_sleep_n 信号。 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: -HIRD 的值大于或者等于 DCTL.HIRD_Thres[3:0] -HIRD_Thres[4]的值为 1 2.当满足以下条件时, core 在 L1 状态将产生 utmi_sleep_n: -HIRD 的值小于 HIRD_Thres[3:0] -HIRD_Thres[4]的值为 0	0x00
[23]	RW	appl1res	LPM 响应配置。 当 PERI_USB2_DCFG[lpm_capable]为 0: core 没有响应; 当 PERI_USB2_DCFG[lpm_capable]为 1: 0: 当满足下面的条件时, LPM 收发成功后将返回 ACK -没有校验错误 -接收到有效的 bLinkState = 0001B (L1) -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) 1: 当满足下面的条件时, LPM 收发成功后将返回 ACK -没有校验错误 -接收到有效的 bLinkState = 0001B (L1)。	0x0
[22:20]	-	reserved	保留。	0x0
[19]	RW	keepconnect	保持连接状态。	0x0



Bits	Access	Name	Description	Reset
			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。	
[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] 中配置的门限值，device core 将产生一个 Hibernation Request Event。	0x0
[17]	RW	controller_restore_state	控制器恢复状态。 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。 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。 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。 说明：当读此比特的时候，返回 0。	0x0
[15:13]	-	reserved	保留。	0x0
[12]	RW	initiate_u2_enable	初始化 U2 使能寄存器。 0：不初始化 U2； 1：初始化 U2。	0x0
[11]	RW	accept_u2_enable	U2 状态使能。 0：core 拒绝进入 U1 状态除非；Force_LinkPM_Accep 为 1 1：如果用户端没有任何设备在等待，则 core 可以接受允许进入 U2 状态。 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。 0：不初始化 U1； 1：初始化 U1。	0x0
[9]	RW	accept_u1_enable	U1 状态使能。 0：core 拒绝进入 U1 状态除非；Force_LinkPM_Accep 为 1； 1：如果用户端没有任何设备在等待，则 core 可以接受允许进入 U1 状态	0x0



Bits	Access	Name	Description	Reset
			说明：USB 复位后，硬件自动清除此比特位。	
[8:5]	WO	ulstchngrq	USB/Link 状态切换请求。 软件对配置此处发起相应的状态切换请求 SS 模式： 0x0: No Action; 0x4: SS.Disabled; 0x5: Rx.Detect; 0x6: SS.Inactive; 0x8: Recovery; 0xA: Compliance; 其他：保留。 HS/FS/LS 模式： 0x8: Remote wakeup request; 其他：保留。	0x0
[4:1]	RW	test_control	测试控制。 0x0: Test mode 关闭; 0x1: Test_J mode; 0x2: Test_K mode; 0x3: Test_SE0_NAK mode; 0x4: Test_Packet mode; 0x5: Test_Force_Enable; 其他：保留。	0x0
[0]	-	reserved	保留。	0x0

PERI_USB2_DEV TEN

PERI_USB2_DEV TEN 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。 0: 禁止; 1: 使能。	0x0



Bits	Access	Name	Description	Reset
[8]	-	reserved	保留。	0x0
[7]	RW	sof_en	SOF 包使能。 0: 禁止; 1: 使能。	0x0
[6]	RW	u3l2l1_susp_en	U3/L2-L1 挂起事件使能。 0: 禁止; 1: 使能。	0x0
[5]	RW	hibernation_req_evt_en	休眠请求使能。 0: 禁止; 1: 使能。	0x0
[4]	RW	wkup_evt_en	恢复/远程唤醒检测事件使能。 0: 禁止; 1: 使能。	0x0
[3]	RW	ulst_cng_en	USB/LINK 状态切换事件使能。 0: 禁止; 1: 使能。	0x0
[2]	RW	connect_done_en	连接完成使能。 0: 禁止; 1: 使能。	0x0
[1]	RW	usbrst_en	USB 复位使能。 0: 禁止; 1: 使能。	0x0
[0]	RW	disconn_evt_en	断开检测使能。 0: 禁止; 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



Bits	Access	Name	Description	Reset
[29]	RO	dcnrd	Device 控制器未准备好。 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。 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。 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。 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。 当 PERI_USB2_DCTL[run_stop]为 1 的时候，此位置 0； 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。 SS 模式： 0x0: U0; 0x1: U1; 0x2: U2; 0x3: U3; 0x4: SS_DIS; 0x5: RX_DET; 0x6: SS_INACT; 0x7: POLL; 0x8: RECOV; 0x9: HRESET; 0xa: CMPPLY; 0xb: LPBK; 0xf: Resume/Reset; 其他：保留。 HS/FS/LS 模式： 0x0: On 状态;	0x4



Bits	Access	Name	Description	Reset
			0x2: Sleep (L1) 状态; 0x3: Suspend (L2) 状态; 0x4: Disconnected 状态 (默认状态); 0x5: Early Suspend 状态(仅当休眠功能关闭是有效); 0xe: Reset (仅当休眠功能开启时有效); 0xf: Resume (仅当休眠功能开启时有效); 其他: 保留。	
[17]	-	reserved	保留。	0x1
[16:3]	RO	soffn	SOF 帧/微帧数目: HS 模式 [16:6]指示帧数目 [5:3]指示微帧数目 全速模式 [16:14]不用 [13:3]指帧数目	0x0000
[2:0]	RO	connect_spd	连接速度指示。 000: 高速(PHY clock 工作在 30 或者 60MHz); 001: 全速(PHY clock 工作在 30 或者 60MHz); 010: 低速 (PHY clock 工作在 6MHz); 010: 低速 (PHY clock 工作在 48MHz)100: 超速(PHY clock 工作在 125 或者 250MHz)。	0x4

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。 0: Device 控制器处理命令成功; 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 控制器执行完命令后将此位置 0。	0x0
[9]	-	reserved	保留。	0x0
[8]	WO	cmdloc	命令完成中断。 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。 0x0: 保留; 0x1: 设置端点的配置; 0x2: 设置端点传输源配置; 0x3: 获取端点状态; 0x4: 设置挂起; 0x5: 清除挂起; 0x6: 开始传输; 0x7: 跟新传输; 0x8: 结束传输; 0x9: 开始新的配置; 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： Bit[0]: USB 输出端点 0 Bit[1]: USB 输入端点 0 Bit[2]: USB 输出端点 1 Bit[3]: USB 输入端点 1 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。 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。	0x00000000

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	这些 bit 表示命令的参数，往寄存器写相关值其意义如下： 对于开始传输命令： [31: 16]: 表示为本次传输申明的 USB StreamID。 对于实时端点的开始传输命令： [31: 16]: 表示首个 TRB 应用时的微帧(帧)的数目。 对于更新传输、结束传输和开始新的配置命令： [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。	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	HighPriority: 仅在开始传输命令才有效。 ForceRM: 仅在结束传输命令才有效。	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。 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	0x0



Bits	Access	Name	Description	Reset
			明先前发送的命令都已实现。	
[9]	RO	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。 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。 0x0：保留； 0x1：设置端点配置(64 或 96bit 参数)； 0x2：设置端点传输资源配置(32bit 参数)； 0x3：得到端点状态(不需要参数)； 0x4：设置 Stall(不需要参数)； 0x5：清除 Stall(不需要参数)； 0x6：开始传输(64bit 参数)； 0x7：更新传输(不需要参数)； 0x8：结束传输(不需要参数)； 0x9：开始新的配置(不需要参数)； 其他：保留。	0x0

12.9 LSADC

12.9.1 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。

芯片提供 1 个 LSADC，2 个独立通道；

12.9.2 特点

LSADC 具有以下特点：

- 电源电压 3.3V/1.8V；
- 扫描频率不能高于 200K/s；
- 10bit 采样精度，4 个独立通道；
- 支持单次扫描和连续扫描模式；



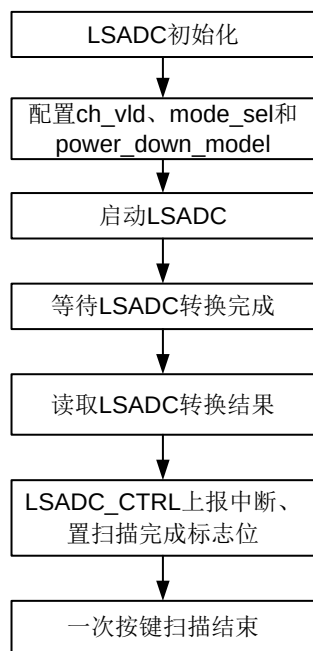
- 扫描完成自动上报中断。

12.9.3 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12-56 单次扫描处理流程



连续扫描处理流程

在连续读取模式（[LSADC_CTRL0](#) [model_sel]=1），CPU 根据应用场景设置连续扫描的时间间隔 T_{scan} 、毛刺宽度（ T_{glitch} ）、有效通道号（ ch_vld ），启动 LSADC。LSADC 在一个时间间隔 T_{scan} 内完成一个有效通道（配置 [LSADC_CTRL0](#) 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，各通道的轮询如[图 12-57](#)所示。

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 12-57](#)所示。

图12-57 连续扫描模式下通道轮询扫描示意图

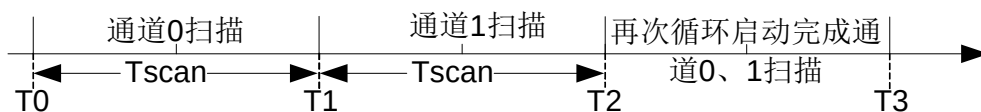
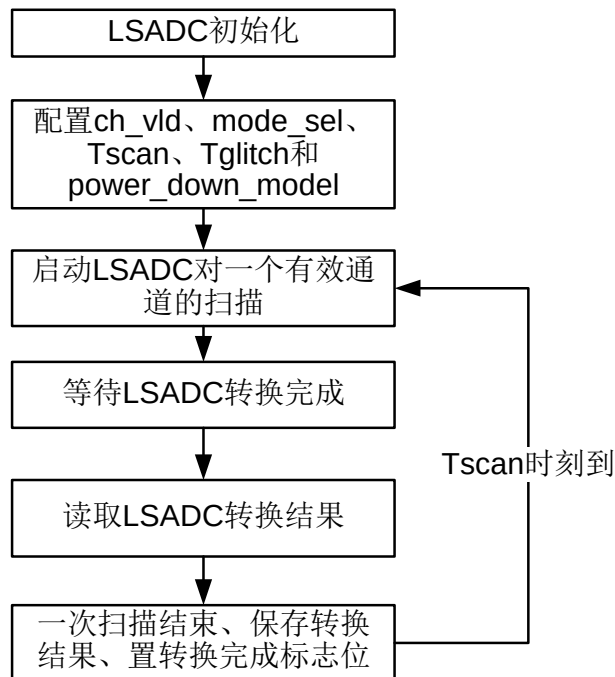




图12-58 连续扫描处理流程



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](#) [9:0]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 10bit 时，采样结果 10bit 全部有效。
- 当采样精度设置小于 10bit 时，对应的采样结果高位有效，例如：当采样精度设置成 8bit 时，采样结果的高 8bit 才有效。

12.9.4 LSADC 寄存器概览

LSADC 寄存器概览如表 12-30 所示。



表12-30 LSADC 寄存器概览（基地址：0x120E_0000）

偏移地址	名称	描述	页码
0x0000	LSADC_CTRL0	LSADC 配置寄存器	12-177
0x0004	LSADC_CTRL1	滤毛刺配置寄存器	12-178
0x0008	LSADC_CTRL2	扫描间隔配置寄存器	12-178
0x0010	LSADC_CTRL4	中断使能寄存器	12-179
0x0014	LSADC_CTRL5	中断状态寄存器	12-179
0x0018	LSADC_CTRL6	中断清除寄存器	12-180
0x001C	LSADC_CTRL7	Start 配置寄存器	12-180
0x0020	LSADC_CTRL8	Stop 配置寄存器	12-180
0x0024	LSADC_CTRL9	转换结果精度寄存器	12-181
0x0028	LSADC_CTRL10	LSADC_ZERO 寄存器	12-181
0x002C	LSADC_CTRL11	LSADC 通道 0 数据寄存器	12-181
0x0030	LSADC_CTRL12	LSADC 通道 1 数据寄存器	12-181

12.9.5 LSADC 寄存器描述

LSADC_CTRL0

LSADC_CTRL0 为 LSADC 配置寄存器。

Offset Address: 0x0000 Total Reset Value: 0x0000_80FF

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式下使用) 0: 使能滤毛刺功能; 1: bypass。	0x0
[16]	-	reserved	保留。	0x0



[15]	RW	lsadc_reset	设置 LSADC 是否进入复位状态。 1: 进入复位状态; 0: 退出复位状态。	0x1
[14]	-	reserved	保留。	0x0
[13]	RW	model_sel	LSADC 扫描模式选择。 0: 单次扫描模式; 1: 连续扫描模式;	0x0
[12:10]	-	reserved	保留。	0x0
[9]	RW	ch_1_vld	LSADC 通道 1 是否有效。 0: 无效; 1: 有效;	0x0
[8]	RW	ch_0_vld	LSADC 通道 0 是否有效。 0: 无效; 1: 有效。	0x0
[7:0]	-	reserved	保留。	0xFF

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)。	0x0000_0000

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 $Tscan = time\_scan * 0.333\mu s$ (连续扫描时间间	0x0000_0000



			隔 应该大于 LSADC 的转换时间,即配置值不 能少于 20)。	

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。	0x0000_0000
[0]	RW	int_enable	扫描值有效中断使能位。 0: 不使能; 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。 写无效, 读为 0。	0x0000_0000
[4]	RO	lsadc_auto_busy	在自动扫描模式下, LSADC busy 状态指示。 0: IDLE; 1: BUSY。	0x0
[3:2]	-	reserved	保留	0x0
[1]	RO	int_flag_in1	通道 1 扫描值有效中断标志位。 0: 无中断; 1: 有中断。	0x0
[0]	RO	int_flag_in0	通道 0 扫描值有效中断标志位。 0: 无中断; 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
[31:2]	-	reserved	保留。	0x0000_0000
[1]	WO	clr_int_flag_in1	通道 1 中断清除寄存器。 0: 不清除; 1: 清除中断。	0x0
[0]	RO	clr_int_flag_in0	通道 0 中断清除寄存器。 0: 不清除; 1: 清除中断。	0x0

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)。	0x0000_0000

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)	0x0000_0000

LSADC_CTRL9

LSADC_CTRL9 为转换结果精度寄存器。

Offset Address: 0x0024 Total Reset Value: 0x0000_03FF



Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x000000
[9:0]	RW	lsadc_active_bit	LSADC 转换结果精度。 10'b1111111111 表示 10 比特精度； 10'b1111111110 表示 9 比特精度； 10'b1000000000 表示 1 比特精度	0x3FF

LSADC_CTRL10

LSADC_CTRL10 为 LSADC_ZERO 寄存器。

Offset Address: 0x0028 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x000000
[9:0]	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
[31:10]	-	reserved	保留。	0x000000
[9:0]	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
[31:10]	-	reserved	保留。	0x000000
[9:0]	RW	lsadc_data_in1	LSADC 通道 1 扫描值。	0x000



12.10 PWM

12.10.1 概述

芯片提供 1 组 2 路独立的脉宽调制信号输出。

12.10.2 特点

对于每组 PWM 输出：

- 3MHz、24MHz 和 50MHz 时钟源可选。
- 内部有 26bit 计数器，输出周期可配置。支持最高 25MHz（50MHz/2）输出，最低约 0.045Hz（3MHz/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。

12.10.3 工作方式

PWM 内部工作于 3MHz 或者 24MHz 或者 50MHz，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_CFG0`、`PWM0_CFG1`、`PWM0_CFG2` 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。

选择 3MHz 时钟为时钟源，周期数配置为 $3\text{MHz} / 3\text{kHz} = 1000$ ，四舍五入后为 1000，十六进制为 `0x00003E8`。高电平数配置为 $1000 \times 72.5\%$ （占空比）= 725，四舍五入后为 725，十六进制为 `0x00002D5`。

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 `0x0`，选择 PWM 的时钟源为 3MHz，打开 PWM 时钟。

步骤 2. 读取 `PWM0_STATE2` `bit[10]`，等待 `bit[10]` 为 0（表示 PWM 空闲，可以输出方波）

步骤 3. 向 `PWM0_CFG0` 写入 `0x0000_03E8`。

步骤 4. 向 `PWM0_CFG1` 写入 `0x0000_02D5`。

步骤 5. 向 `PWM0_CFG2` 写入 `0x0000_000A`。



- 步骤 6. 向 **PWM0_CTRL** 写入 0x1。（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_STATE2** bit[10]，等待 bit10 位为 1（表示 PWM 正在输出方波）
- 步骤 8. 读取 **PWM0_STATE0** 和 0x0000_03E8 进行校验。
- 步骤 9. 读取 **PWM0_STATE1** 和 0x0000_02D5 进行校验。
- 步骤 10. 读取 **PWM0_STATE2** bit[9:0]和 0x0A 进行校验(当 **PWM0_STATE2** bit[10]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

12.10.4 PWM 寄存器概览

各路 PWM 寄存器的基地址如表 12-31 所示。

表12-31 PWMn 的寄存器基地址表

PWM 路数	基地址
PWM0	0x1207_0000
PWM1	0x1207_0020

以 PWM0 寄存器的描述为例，寄存器概览如表 12-32 所示。

表12-32 PWM0 寄存器概览

偏移地址	名称	描述	页码
0x0000	PWM0_CFG0	PWM0 的配置 0 寄存器	12-184
0x0004	PWM0_CFG1	PWM0 的配置 1 寄存器	12-184
0x0008	PWM0_CFG2	PWM0 的配置 2 寄存器	12-184
0x000C	PWM0_CTRL	PWM0 的控制寄存器	12-185
0x0010	PWM0_STATE0	PWM0 的状态 0 寄存器	12-185
0x0014	PWM0_STATE1	PWM0 的状态 1 寄存器	12-185
0x0018	PWM0_STATE2	PWM0 的状态 2 寄存器	12-185

12.10.5 PWM 寄存器描述

PWM0_CFG0

PWM0_CFG0 为 PWM0 的配置 0 寄存器。



Offset Address: 0x0000 Total Reset Value: 0x0000_018F

Bits	Access	Name	Description	Reset
[31:0]	RW	pwm0_period	PWM0 的周期数。  说明 要求配置值必须 ≥ 2 。	0x0000_018F

PWM0_CFG1

PWM0_CFG1 为 PWM0 的配置 1 寄存器。

Offset Address: 0x0004 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。 注意:要求配置值必须≥ 1。	0x0000_00C7

PWM0_CFG2

PWM0_CFG2 为 PWM0 的配置 2 寄存器。

Offset Address: 0x0008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	RO	reserved	保留。	0x000000
[9:0]	RW	pwm0_num	PWM0 输出方波的数目。	0x000

PWM0_CTRL

PWM0_CTRL 为 PWM0 的控制寄存器。

Offset Address: 0x000C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	RO	reserved	保留。	0x00000000
[2]	RW	pwm0_keep	PWM 输出模式。 0: PWM0 输出固定数目的方波; 1: PWM0 一直输出方波。	0x0
[1]	RW	pwm0_inv	PWM 输出正反相控制。	0x0



			0: PWM0 正常输出方波; 1: PWM0 反相输出方波。	
[0]	RW	pwm0_enable	PWM 使能控制。 0: PWM0 模块关闭; 1: PWM0 模块使能。	0x0

PWM0_STATE0

PWM0_STATE0 为 PWM0 的状态 0 寄存器。

Offset Address: 0x0010 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

PWM0_STATE1

PWM0_STATE1 为 PWM0 的状态 1 寄存器。

Offset Address: 0x0014 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_STATE2

PWM0_STATE2 为 PWM0 的状态 2 寄存器。

Offset Address: 0x0018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:22]	RO	reserved	保留。	0x000
[21:12]	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
[11]	RO	pwm0_keep_st	PWM0 的内部模块采用的输出方波的模式。 0: 输出固定数目的方波; 1: 一直输出方波。	0x0



[10]	RO	pwm0_busy	PWM0 模块的工作状态。 0: 空闲，输出完成; 1: 正在输出方波。	0x0
[9:0]	RO	pwm0_period_st	PWM0 的内部模块采用的输出方波数目。	0x000



目 录

13 安全模块.....	13-1
13.1 SPACC	13-1
13.1.1 概述	13-1
13.1.2 特点	13-1
13.1.3 功能描述.....	13-2
13.1.4 工作方式.....	13-12
13.1.5 SPACC 寄存器概览	13-12
13.1.6 SPACC 寄存器描述	13-16
13.2 HISEC_TRNG_CTRL	13-46
13.2.1 概述	13-46
13.2.2 特点	13-46
13.2.3 寄存器概览.....	13-46
13.2.4 寄存器描述.....	13-47
13.3 RSA.....	13-50
13.3.1 概述	13-50
13.3.2 特点	13-50
13.3.3 寄存器概览.....	13-51
13.3.4 RSA 寄存器描述.....	13-51
13.4 OTP Ctrl.....	13-55
13.4.1 工作方式.....	13-55
13.4.2 OTP Ctrl 寄存器概览	13-59
13.4.3 OTP CTRL 寄存器描述	13-59



插图目录

图 13-1 3 个密钥和 2 个密钥的 3DES 加密操作	13-2
图 13-2 3 个密钥和 2 个密钥的 3DES 解密操作	13-3
图 13-3 AES/DES 的电子密码本（ECB）模式	13-3
图 13-4 3DES 的电子密码本（ECB）模式	13-4
图 13-5 AES/DES 的密码分组链接（CBC）模式	13-5
图 13-6 3DES 的密码分组链接（CBC）模式	13-6
图 13-7 AES/DES 的 s 位密码反馈（CFB）模式	13-7
图 13-8 3DES 的 s 位密码反馈（CFB）模式	13-8
图 13-9 AES 的输出反馈（OFB）模式	13-9
图 13-10 DES 的 s 位输出反馈（OFB）模式	13-10
图 13-11 3DES 的 s 位输出反馈（OFB）模式	13-11
图 13-12 AES 的 CTR 模式	13-12



表格目录

表 13-1 各模块的寄存器偏移地址变量表.....	13-12
表 13-2 SPACC 寄存器概览（基址是 0x100C_0000）	13-13
表 13-3 HISEC_TRNG_CTRL 寄存器概览（基址是 0x1009_0000）	13-46
表 13-4 RSA 寄存器概览（RSA0 基址是 0x1008_0000； RSA1 基址是 0x100D_0000）	13-51
表 13-5 OTP_CTRL 寄存器概览（基址是 0x100B_0000）	13-59



13 安全模块

13.1 SPACC

13.1.1 概述

SPACC 实现对称算法和 HASH 算法功能规格。

- 对称算法功能是一个实现 DES（Data Encryption Standard）/3DES/AES（Advanced Encryption Standard）加解密处理的模块，DES/3DES 和 AES 算法的实现符合 FIPS46-3/FIPS 197 标准。DES/3DES 和 AES 的工作模式符合 FIPS -81/NIST special800-38a 标准。
- 对称算法适用于进行大量数据的高效加解密处理，可支持一次实现单个分组的加解密或多个分组的加解密。
- HASH 算法是一个实现 SHA1/SHA224/SHA256/SHA384/SHA512/ HMAC_SHA1/ HMAC_SHA224/ HMAC_SHA256/ HMAC_SHA384/ HMAC_SHA512 算法的模块。SHA1/SHA224/SHA256/SHA384/SHA512 算法的实现符合 FIPS180-2 标准。HMAC_SHA1/ HMAC_SHA224/ HMAC_SHA256/ HMAC_SHA384/ HMAC_SHA512 算法的实现符合 RFC2104 标准。
- HASH 算法适用于数据完整性认证和构建数字签名。

13.1.2 特点

SPACC 模块有如下特点：

- 支持算法 AES ECB/CBC/OFB/CFB/CTR。
- 支持算法 3DES/DES ECB/CBC/OFB/CFB。
- 支持算法 SHA1、SHA224、SHA256、SHA384、SHA512、HMAC-SHA1、HMAC-SHA224、HMAC-SHA256、HMAC-SHA384、HMAC-SHA512。
- AES 密钥长度支持 128 位、192 位、256 位，由芯片密钥管理模块配置密钥时，密钥长度支持 128 位、192 位、256 位。
- DES 密钥长度支持 64 位，其中 0、8、16、24、32、40、48、56 位的值是每个 byte 的奇偶校验值，实际加解密操作中不使用。
- 3DES 支持 3 个密钥的方式，也支持 2 个密钥的方式，由芯片密钥管理模块配置密钥时只支持 2 个密钥的方式。



- 支持 8 个通道，1 个单块的 CPU 通道和 7 个多块 DMA 通道。
- 支持 CPU 配置输入数据和 DMA 方式读取输入数据。
- 多块加解密通道支持字节地址。
- 多块加解密通道支持循环链表结构，支持拼接多个链表数据，每个链表的长度用 32 bit 表示。
- 提供中断状态查询、中断屏蔽和中断清除功能。
- 每个通道提供单独的中断处理和控制在。

13.1.3 功能描述

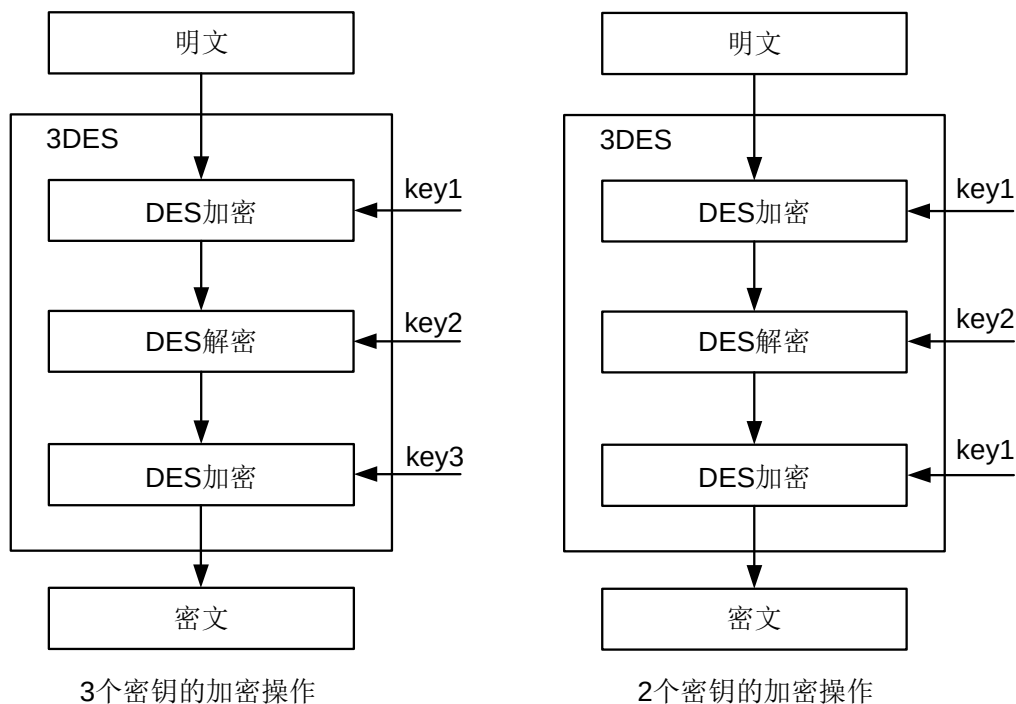
DES/3DES、AES 算法支持的几种工作模式分别符合 FIPS-81 标准和 NIST special800-38a/c/d 标准，对于 DES/3DES 和 AES 算法，ECB、CBC 和 CFB 工作模式相同，OFB 和 CTR（只有 AES 算法中包括）工作模式略有区别。

3DES 算法

3DES 支持 3 个密钥和 2 个密钥的运算，2 个密钥的运算可以看作 3 个密钥的一种简化情况，在 2 个密钥的操作中的第三个密钥（key3）都使用第一个密钥（key1）代替。

3 个密钥和 2 个密钥的 3DES 加密运算过程如图 13-1 所示。

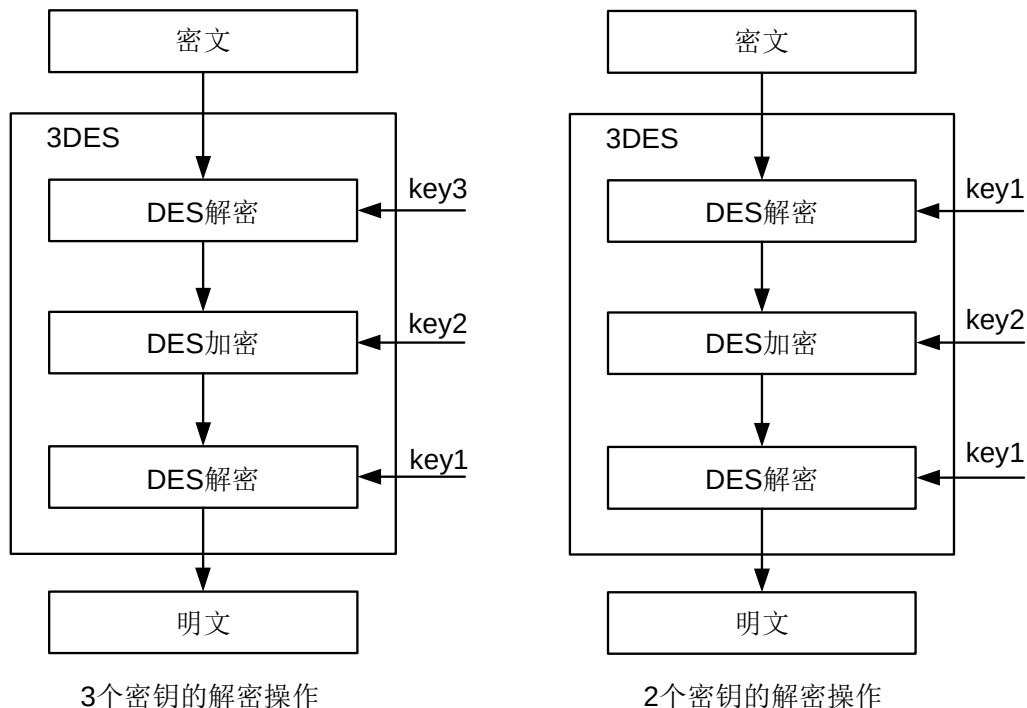
图13-1 3个密钥和2个密钥的3DES加密操作



3 个密钥和 2 个密钥的 3DES 解密运算过程如图 13-2 所示。



图13-2 3个密钥和2个密钥的3DES 解密操作



ECB 模式

ECB (Electronic CodeBook) 模式中，加、解密算法是直接应用到各个分组数据，而且各个分组的运算均独立。这个特点使得明文的加密操作和密文的解密操作可以并行进行。AES/DES 和 3DES 的电子密码本 (ECB) 模式分别如图 13-3 和图 13-4 所示。

图13-3 AES/DES 的电子密码本 (ECB) 模式

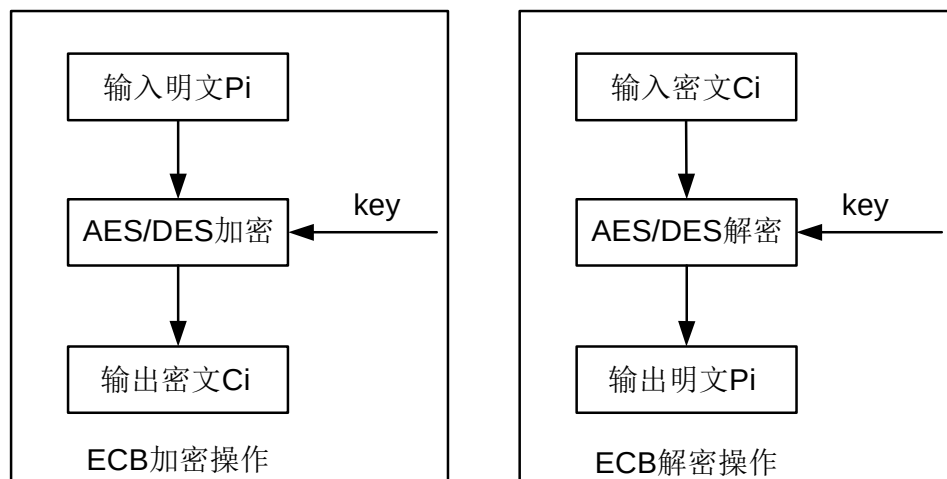
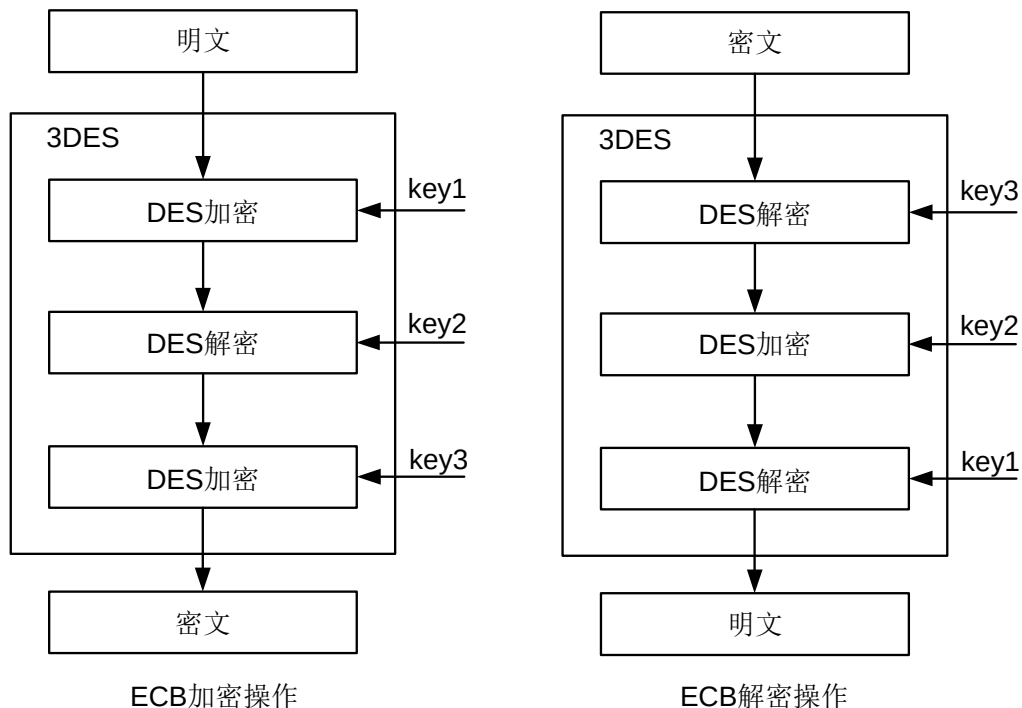




图13-4 3DES 的电子密码本（ECB）模式



CBC 模式

CBC (Cipher Block Chaining) 模式下，加密的输入明文分组需要先与输入向量 IV (Initialization Vector) 进行异或操作，才进入加密操作，而每个明文分组的加密处理都与上一个明文分组处理的结果（即密文）相关，因此 CBC 模式下的加密操作是不能进行并行处理的。但是解密操作不依赖于上一个分组的明文输出，是可以进行并行处理的。AES/DES 和 3DES 的密码分组链接（CBC）模式分别如图 13-5 和图 13-6 所示。



图13-5 AES/DES 的密码分组链接（CBC）模式

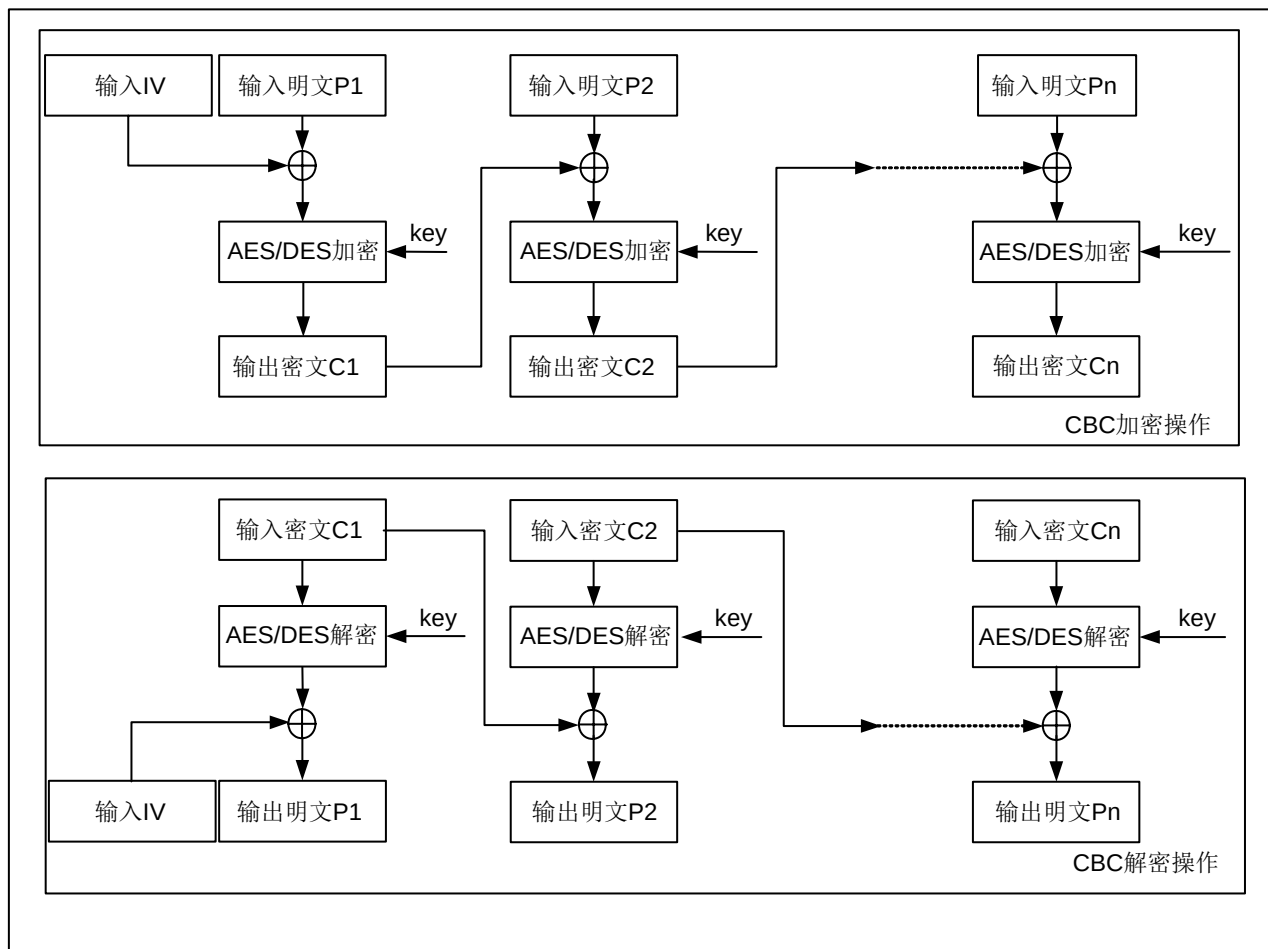
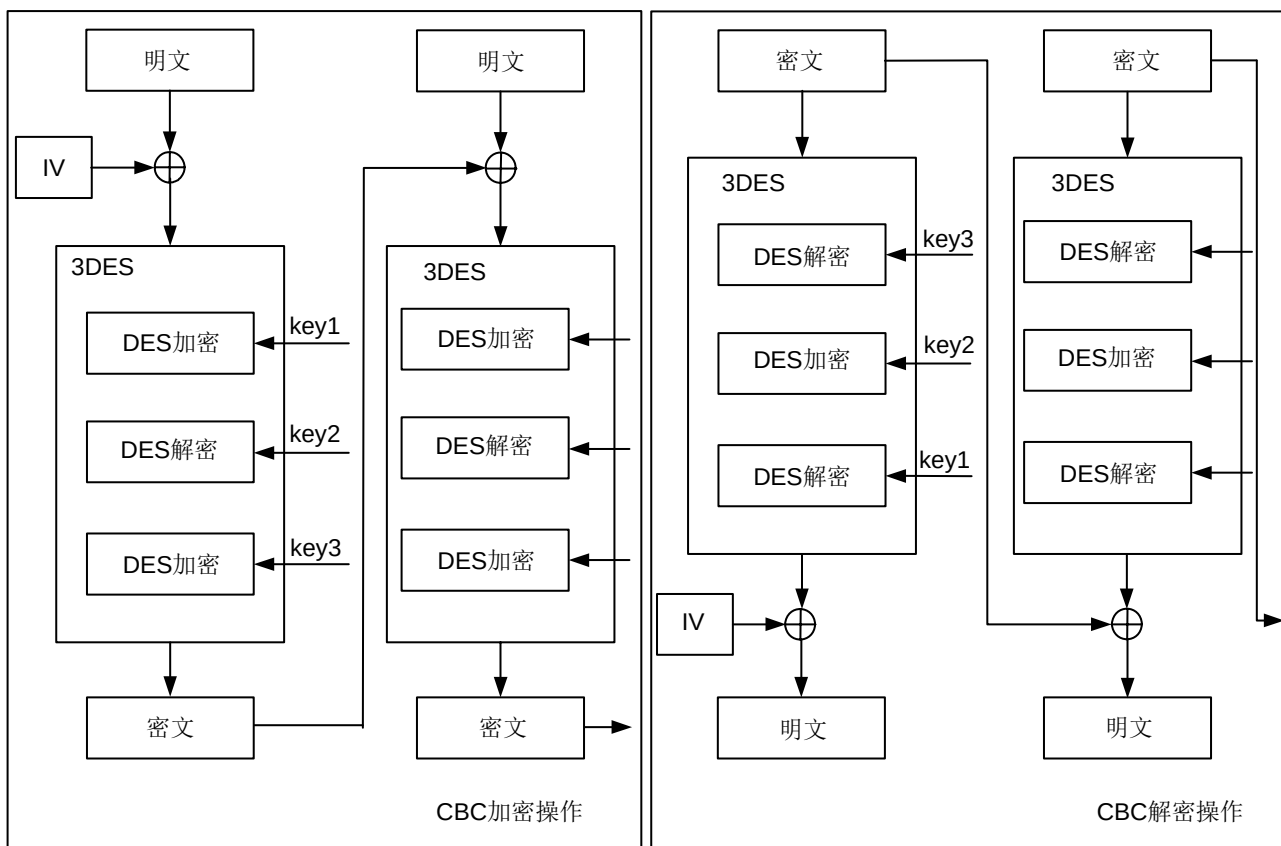




图13-6 3DES 的密码分组链接（CBC）模式



CFB 模式

CFB (Cipher FeedBack) 模式是将分组密码转换成流密码的一种工作模式，可以通过选择 CFB 的操作位数来实现。移位操作的位数用 s 位表示，关于 s 位，存在以下 2 种情况：

- 对于 DES/3DES， s 位可以是 1 位、8 位或 64 位。
- 对于 AES， s 位可以是 1 位、8 位或 128 位。

AES/DES 的 s 位密码反馈 (CFB) 模式和 3DES 的 s 位密码反馈 (CFB) 模式分别如图 13-7 和图 13-8 所示。



图13-7 AES/DES 的 s 位密码反馈（CFB）模式

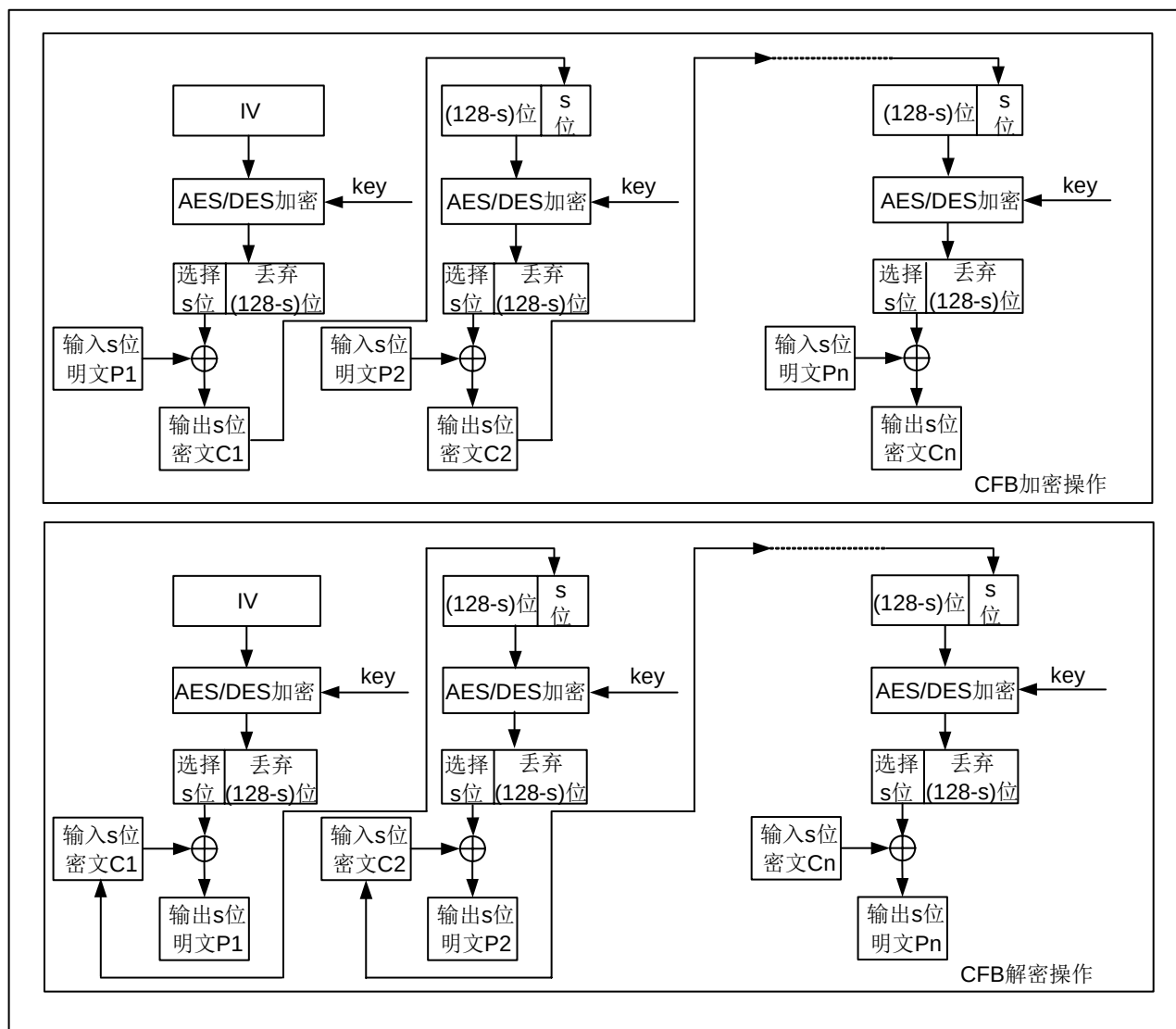
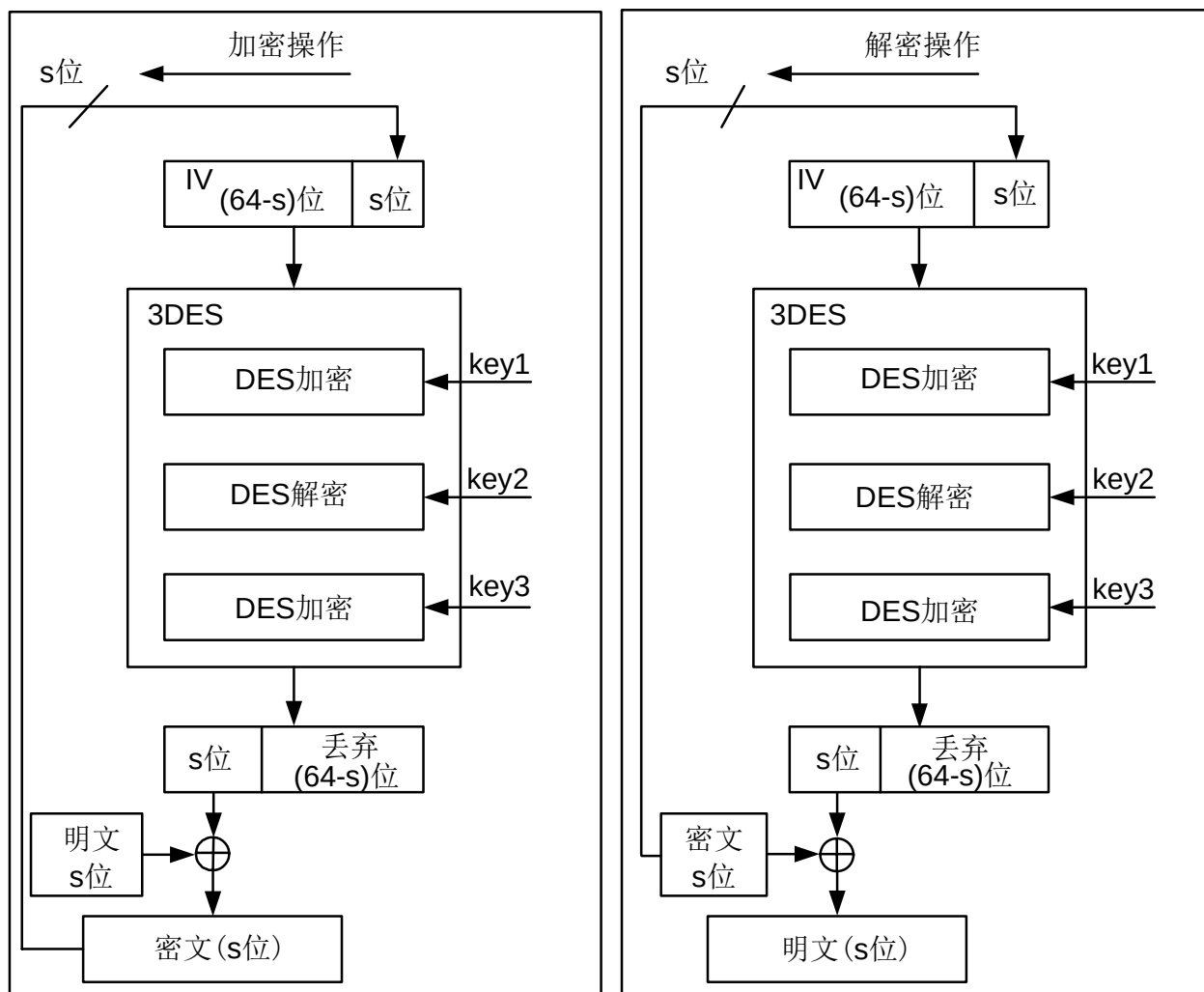




图13-8 3DES 的 s 位密码反馈（CFB）模式



OFB 模式

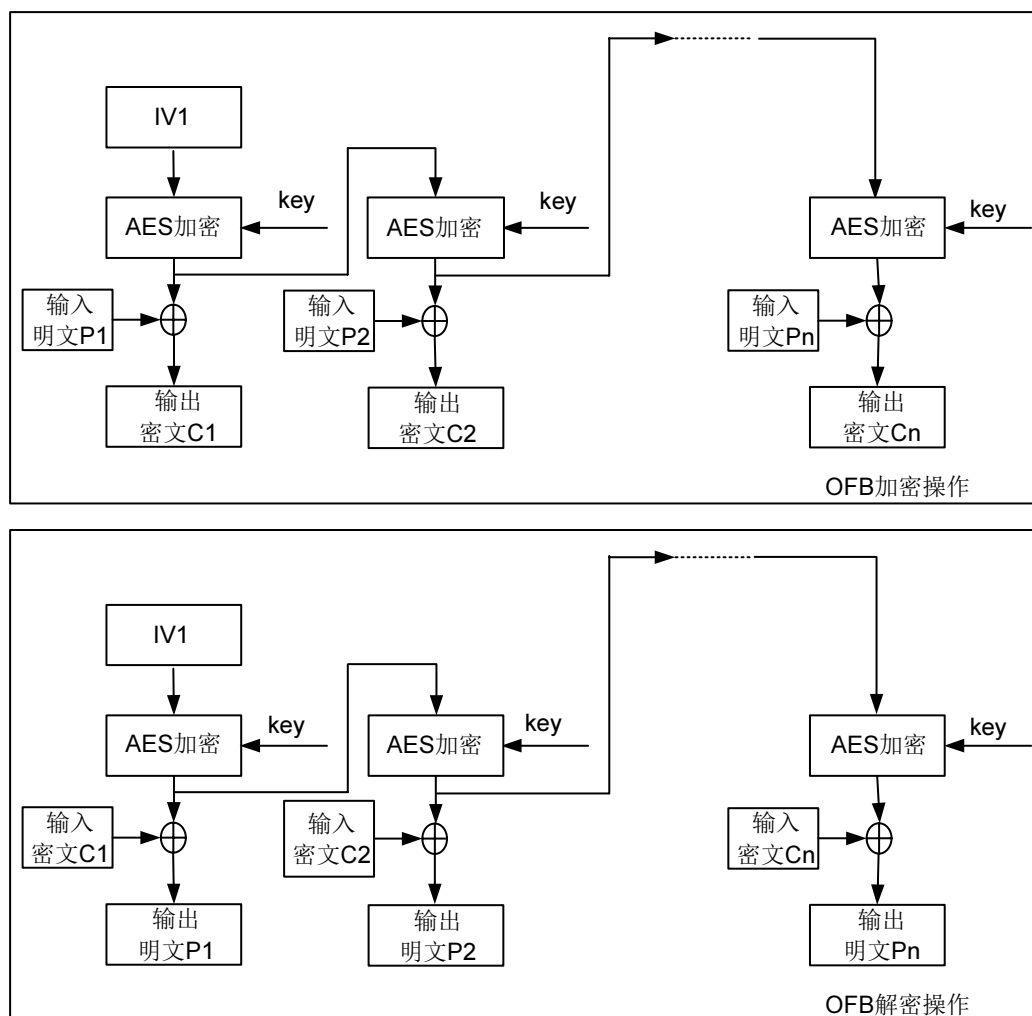
OFB (Output FeedBack) 模式下, 将 IV 直接作为加密操作的输入, 因此对同一个密钥的操作情况下, 应该使用不相同的 IV, 避免降低操作的安全性。关于 s 位, 存在以下 2 种情况:

- 对于 DES/3DES, s 位可以是 1 位、8 位或 64 位。
- 对于 AES, s 位只能是 128 位。



AES 的输出反馈（OFB）模式如图 13-9 所示。

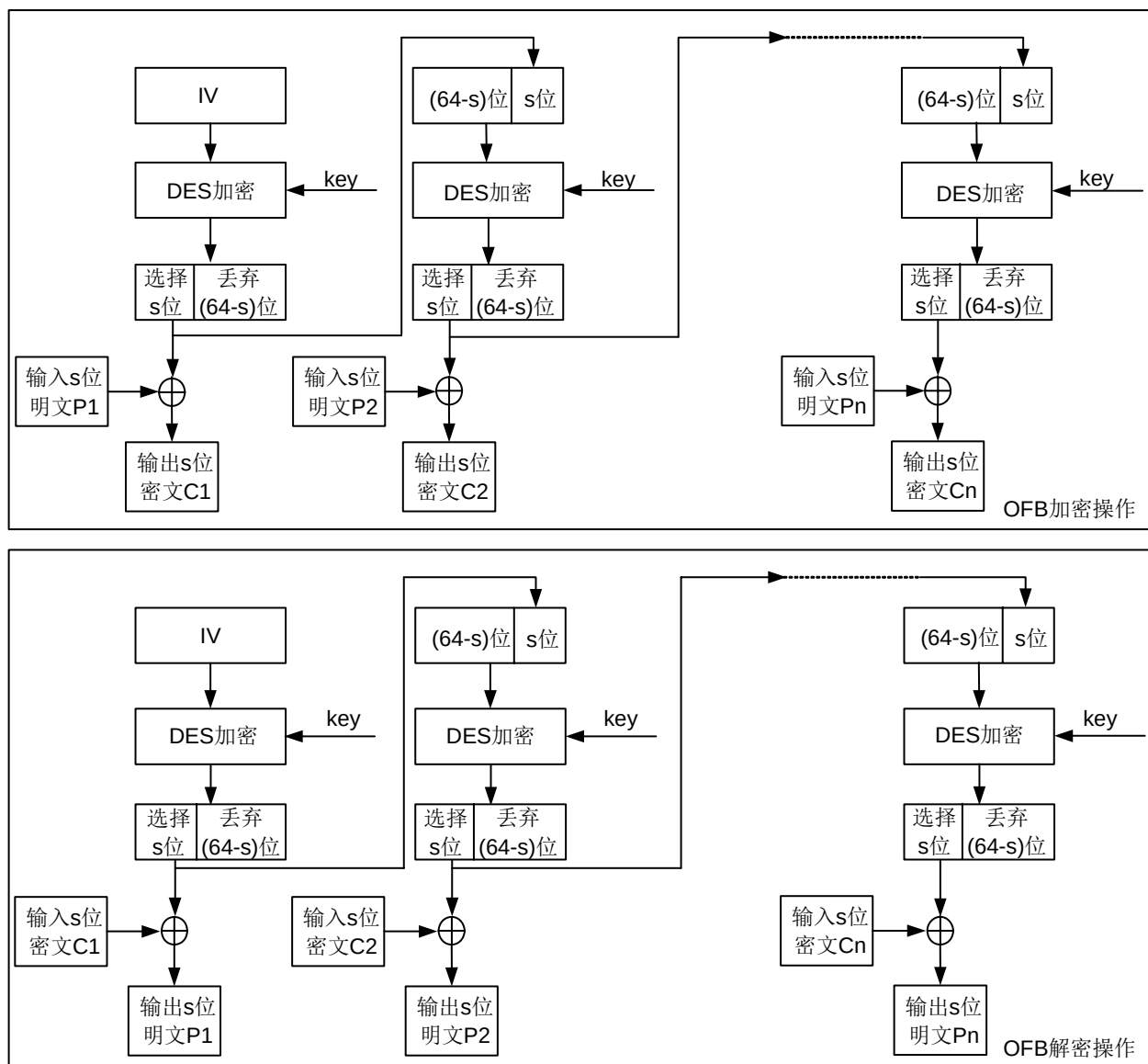
图13-9 AES 的输出反馈（OFB）模式



DES 的 s 位输出反馈模式如图 13-10 所示。



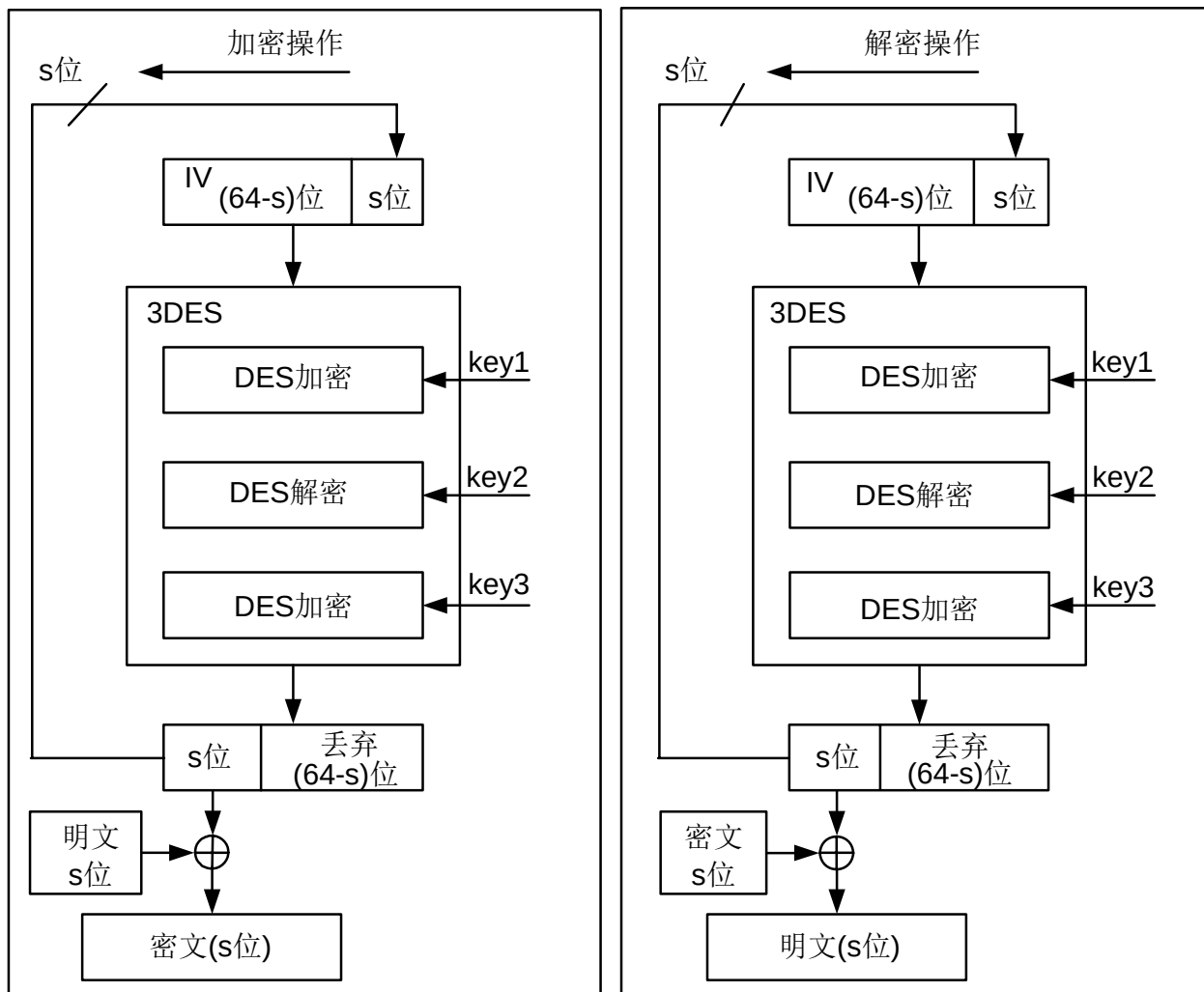
图13-10 DES 的 s 位输出反馈（OFB）模式



3DES 的 s 位输出反馈模式如图 13-11 所示。



图13-11 3DES 的 s 位输出反馈 (OFB) 模式



CTR 模式

CTR (Counter) 模式下，向 AES 加密或解密处理模块输入不同的数据来保证数据处理的安全性，这种数据可以是计数的值。因此，计数值 CTR_n 的选取也决定了这种方式应用的安全性。

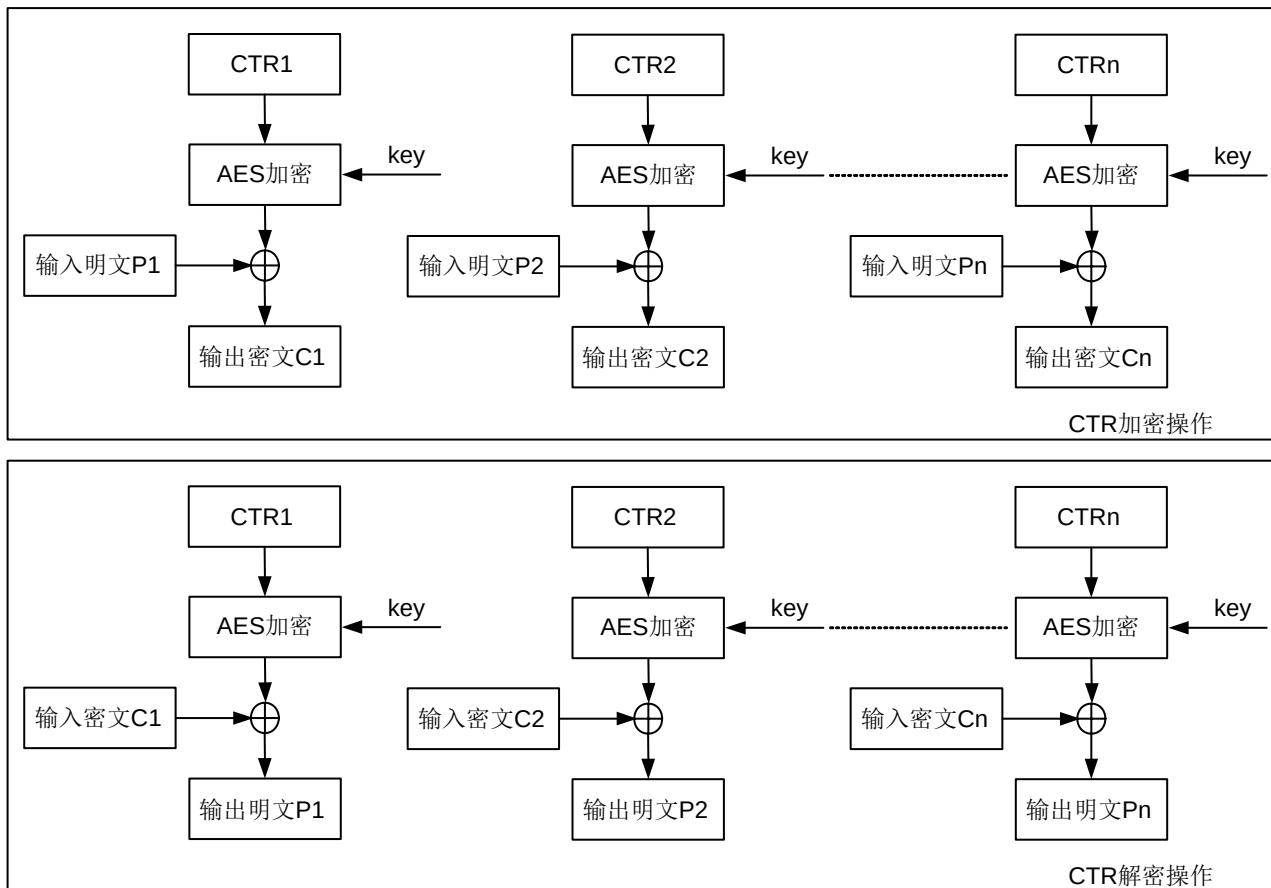
说明

CTR_n 一般采用累加计数的方式获取。

AES 的 CTR 模式如图 13-12 所示。



图13-12 AES 的 CTR 模式



13.1.4 工作方式

时钟门控

当不需要进行加密操作，且 SPACC 模块处于空闲状态时，可以通过配置 3.2.6 章节的 CRG 寄存器，关断 SPACC 模块时钟以降低功耗。

软复位

可以通过配置 3.2.6 章节的 CRG 寄存器对 SPACC 模块进行软复位。

13.1.5 SPACC 寄存器概览

各模块的寄存器偏移地址中变量的取值范围和含义如表 13-1 所示。

表13-1 各模块的寄存器偏移地址变量表

变量名称	取值范围	描述
j	0~7	256bit 数据的 word 数 0~7



变量名称	取值范围	描述
k	0~3	128bit 数据的 word 数 0~3
m	0~7	通道号 0~7
n	1~7	通道号 1~7

SPACC 寄存器概览如表 13-2 所示。

表13-2 SPACC 寄存器概览（基址是 0x100C_0000）

偏移地址	名称	描述	页码
0x0+k×0x04	CHAN0_CIPHER_IV	对称算法通道 0 的向量输入输出寄存器	13-16
0x0000+n×0x10+k×0x04	CHANN_CIPHER_IVOUT	通道 n(1~7)对称算法的向量输出寄存器	13-17
0x80+k×0x04	CHAN0_CIPHER_DOUT	对称算法通道 0 输出寄存器	13-17
0x0100+m×0x20+j×0x04	CIPHER_KEY	对称算法 CPU 配置密钥寄存器	13-18
0x0304	SEC_CHN_CFG	安全通道配置寄存器	13-19
0x0308	SPACC_VERSION	SPACC 版本寄存器	13-20
0x0310	MEM_EMA_CFG	memory 的 ema 配置	13-20
0x0314	KEY_ST	core 使用的 key 的状态信号	13-21
0x0318	CALC_ST0	计算状态寄存器	13-21
0x0320	CALC_ERR	算法 core 告警状态寄存器	13-21
0x340+m×0x8	CHANN_HASH_STATE_VAL	通道 m(0~7)的 HASH 输入输出状态值	13-22
0x344+m×0x8	CHANN_HASH_STATE_VAL_ADDR	通道 m(0~7)的 hash 输入输出状态值地址	13-22
0x039c	SPACC_CALC_CRG_CFG	SPACC 内部算法 core 时钟门控与复位配置	13-22
0x0400	CHAN0_CIPHER_CTRL	对称算法通道 0 的加解密控制寄存器	13-24



偏移地址	名称	描述	页码
0x0404	CIPHER_INT_STAT US	对称算法中断状态寄存器	13-27
0x0408	CIPHER_INT_EN	对称算法中断使能寄存器	13-28
0x040C	CIPHER_INT_RAW	对称算法中断状态寄存器	13-28
0x0410	CIPHER_IN_SMMU _EN	通道 1~7 对称算法的输入数据读地址 bypass	13-29
0x0414	OUT_SMMU_EN	通道 1~7 对称算法的输出结果读地址 bypass	13-29
0x420+k ×0x04	CHAN0_CIPHER_DI N	对称算法通道 0 数据输入寄存器	13-30
0x0440	NORM_SMMU_STA RT_ADDR	非安全 SMMU 起始地址寄存器	13-31
0x0444	SEC_SMMU_STAR T_ADDR	安全 SMMU 起始地址寄存器	13-31
0x0450	IN_ST	输入状态寄存器	13-32
0x0454	OUT_ST	输出状态寄存器	13-32
0x400+n ×0x80	CHANN_CIPHER_C TRL	通道 n 对称算法的加解密控制寄存器	13-32
0x404+n ×0x80	CHANN_CIPHER_I N_NODE_CFG	通道 n 的输入节点配置寄存器	13-35
0x408+n ×0x80	CHANN_CIPHER_I N_NODE_START_A DDR	通道 n 的输入节点起始地址寄存器	13-35
0x40C+n ×0x80	CHANN_CIPHER_I N_BUF_RPTR	通道 n 的输入 BUF 读地址寄存器	13-35
0x410+n ×0x80	CHANN_CIPHER_I N_LEFT_BYTE	通道 n 当前读入数据中不足一块的剩余 数据	13-35
0x414+n ×0x80	CHANN_CIPHER_I N_LEFT_WORD0	通道 n 当前读入数据中不足一块的剩余 数据	13-36
0x418+n ×0x80	CHANN_CIPHER_I N_LEFT_WORD1	通道 n 当前读入数据中不足一块的剩余 数据	13-36
0x41C+n ×0x80	CHANN_CIPHER_I N_LEFT_WORD2	通道 n 当前读入数据中不足一块的剩余 数据	13-36
0x420+n ×0x80	CHANN_CIPHER_I N_BUF_REST_LEN	通道 n 输入 buf 的剩余长度	13-36



偏移地址	名称	描述	页码
0x430+n ×0x80	CHANN_CIPHER_O UT_NODE_CFG	通道 n 的输出节点配置寄存器	13-37
0x434+n ×0x80	CHANN_CIPHER_O UT_NODE_START_ ADDR	通道 n 的输出节点起始地址寄存器	13-37
0x438+n ×0x80	CHANN_CIPHER_O UT_BUF_RPTR	通道 n 的输出 BUF 读地址寄存器	13-37
0x440+n ×0x80	CHANN_CIPHER_O UT_LEFT_BYTE	通道 n 当前输出数据中不足一块的剩余数据	13-38
0x444+n ×0x80	CHANN_CIPHER_O UT_LEFT_WORD0	通道 n 当前数据中不足一块的剩余数据	13-38
0x448+n ×0x80	CHANN_CIPHER_O UT_LEFT_WORD1	通道 n 当前数据中不足一块的剩余数据	13-38
0x44C+n ×0x80	CHANN_CIPHER_O UT_LEFT_WORD2	通道 n 当前数据中不足一块的剩余数据	13-38
0x450+n ×0x80	CHANN_CIPHER_O UT_BUF_REST_LE N	通道 n 输入 buf 的剩余长度	13-39
0x460+n ×0x80	CHANN_CIPHER_I N_NODE_START_A DDR_HIGH	通道 n 的输入节点起始地址寄存器	13-39
0x464+n ×0x80	CHANN_CIPHER_I N_BUF_RPTR_HIG H	通道 n 的输入 BUF 读地址寄存器	13-39
0x470+n ×0x80	CHANN_CIPHER_O UT_NODE_START_ ADDR_HIGH	通道 n 的输出节点起始地址寄存器	13-39
0x474+n ×0x80	CHANN_CIPHER_O UT_BUF_RPTR_HI GH	通道 n 的输出 BUF 读地址寄存器	13-40
0x0800	CHAN0_HASH_CT RL	通道 0 的 HASH 控制寄存器	13-40
0x0804	HASH_INT_STATU S	HASH 中断状态寄存器	13-41
0x0808	HASH_INT_EN	HASH 中断使能寄存器	13-41
0x080C	HASH_INT_RAW	HASH 原始中断状态寄存器	13-42
0x0810	HASH_IN_SMMU_E N	通道 1~7 的读地址 bypass	13-42



偏移地址	名称	描述	页码
0x0818	CHAN0_HASH_DATA_IN	CPU 输入数据	13-43
0x081C	CHAN0_HASH_TO_TAL_DAT_LEN	CPU 输入数据长度	13-43
0x0820	CHAN0_HASH_IN_DAT_LEN	通道 0 的已输入数据长度	13-43
0x800+n ×0x80	CHANN_HASH_CTL	通道 n(1~7)的 HASH 控制寄存器	13-43
0x804+n ×0x80	CHANN_HASH_IN_NODE_CFG	通道 n(1~7)的输入节点配置寄存器	13-44
0x808+n ×0x80	CHANN_HASH_IN_NODE_START_ADDR	通道 n(1~7)的输入节点起始地址寄存器	13-45
0x80C+n ×0x80	CHANN_HASH_IN_BUF_RPTR	通道 n(1~7)的输入 BUF 读地址寄存器	13-45
0x810+n ×0x80	CHANN_HASH_IN_BUF_REST_LEN	通道 n 输入 buf 的剩余长度	13-45
0x820+n ×0x80	CHANN_HASH_IN_NODE_START_ADDR_HIGH	通道 n(1~7)的输入节点起始地址寄存器	13-45
0x824+n ×0x80	CHANN_HASH_IN_BUF_RPTR_HIGH	通道 n(1~7)的输入 BUF 读地址寄存器	13-46

13.1.6 SPACC 寄存器描述

CHAN0_CIPHER_IV

CHAN0_CIPHER_IV 为对称算法通道 0 的向量输入输出寄存器。

Offset Address: 0x0+k×0x04 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	chan0_cipher_iv	通道 0 的对称算法 IV 输入或输出向量，小端格式。 1)如果选择不需要进行更新向量配置 (CHAN0_CIPHER_CTRL[sym_ch0_ivin_sel]='b00)，则逻辑不需要重新读取该寄存器。 2)如果选择需要进行更新向量配置 (CHAN0_CIPHER_CTRL[sym_ch0_ivin_sel]='b01)，则逻辑需要重新读取该寄存器。对于 AES/SM1/SM4 运算，chan0_cipher_ivin[127:0]为有效数据；对于 DES/3DES 运算,chan0_cipher_ivin[63:0]为有效数据。	0x00000000

CHANN_CIPHER_IVOUT

CHANN_CIPHER_IVOUT 为通道 n(1~7)对称算法的向量输出寄存器。

Offset Address: $0x0000 + n \times 0x10 + k \times 0x04$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	chann_cipher_ivout	对称算法的向量输出,0x00n0~0x00nC 对应通道 n，小端格式； 1)如果执行的是 ECB 工作模式，不需要关注此寄存器； 2)如果选择进行 AES/SM1/SM4 运算，128 位数据均有效； 3)如果选择进行 DES 或 3DES 运算，chann_cipher_ivout[63:0]为有效数据。	0x00000000

CHAN0_CIPHER_DOUT

CHAN0_CIPHER_DOUT 为对称算法通道 0 输出寄存器。

Offset Address: $0x80 + k \times 0x04$ Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RO	chan0_cipher_dout	对称算法通道 0 输出，小端格式。 AES/SM1: 1)如果选择 1-CFB 模式， chan0_cipher_dout[0]为有效数据； 2)如果选择 8-CFB 模式， chan0_cipher_dout[7:0]为有效数据； 3)如果选择 128-CFB 操作，128 位数据均有效； 4)其它模式下 128 位数据均有效。 SM4:128 位数据均有效。 DES/3DES: 1)如果选择 1-CFB 或 1-OFB 模式， chan0_cipher_dout[0]为有效数据； 2)如果选择 8-CFB 或 8-OFB 模式， chan0_cipher_dout[7:0]为有效数据； 3)如果选择 64-CFB 或 64-OFB 模式， chan0_cipher_dout[63:0]为有效数据； 4)其它模式下低 64 位数据有效， chan0_cipher_dout[63:0]为有效数据。	0x00000000

CIPHER_KEY

CIPHER_KEY 为对称算法 CPU 配置密钥寄存器。

Offset Address: $0x0100 + m \times 0x20 + j \times 0x04$ Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	WO	cipher_key	对称算法的密钥输入，无法读取(读结果为0)，小端格式。 0x0100~0x011C: 通道 0 的 host_key; 其它以此类推。 1)选择 DES 运算时，cipher_key[63:0]为有效数据。 2)选择 3DES 运算： a)选择 3 个密钥运算时，此时 cipher_key[63:0],cipher_key[127:64],cipher_key[191:128]分别表示第 1,2,3 个密钥； b)选择 2 个密钥运算时，此时 cipher_key[63:0],cipher_key[127:64]分别表示第 1,2 个密钥。 3)选择 AES 运算时： a)如果选择 128 位密钥操作，cipher_key[127:0]为有效数据； b)如果选择 192 位密钥操作，cipher_key[191:0]为有效数据； c)如果选择 256 位密钥操作，256 位数据均有效。 4)选择 SM1 运算时： 256 位数据均有效，cipher_key[255:0]表示 AK EK。 4)选择 SM4 运算时： cipher_key[127:0]为有效数据。	0x00000000

SEC_CHN_CFG

SEC_CHN_CFG 为安全通道配置寄存器。

Offset Address: 0x0304 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:25]	-	reserved	保留。	0x00
[24]	RW	hash_sec_chn_cfg_lock	安全通道配置锁定，锁定所有通道的 sec_chn_cfg，只有安全 cpu 可以写 1(不能写 0)，系统复位清除 hash_sec_chn_cfg_lock；	0x0
[23:16]	RW	hash_sec_chn_cfg	安全通道配置，必须在通道不工作的时候配置好。	0x00



Bits	Access	Name	Description	Reset
			hash_sec_chn_cfg[i]为 0, 表示 i 通道为非安全通道; hash_sec_chn_cfg[i]为 1, 表示 i 通道为安全通道; 其中 i=0~7, 表示 0~7 通道。	
[15:9]	-	reserved	保留。	0x00
[8]	RW	cipher_sec_chn_cfg_lock	安全通道配置锁定, 锁定所有通道的 hash_sec_chn_cfg, 只有安全 cpu 可以写 1(不能写 0), 系统复位清除 hash_sec_chn_cfg_lock;	0x0
[7:0]	RW	cipher_sec_chn_cfg	安全通道配置, 必须在通道不工作的时候配置好。 hash_sec_chn_cfg[i]为 0, 表示 i 通道为非安全通道; hash_sec_chn_cfg[i]为 1, 表示 i 通道为安全通道; 其中 i=0~7, 表示 0~7 通道。	0x00

SPACC_VERSION

SPACC_VERSION 为 SPACC 版本寄存器。

Offset Address: 0x0308 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	spacc_version	SPACC 版本寄存器, 使用宏定义。	0x00000000

MEM_EMA_CFG

MEM_EMA_CFG 为 memory 的 ema 配置。

Offset Address: 0x0310 Total Reset Value: 0x0000_A413

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15]	RW	rft_colldisn	rft 的 colldisn 信号。	0x1
[14]	RW	rft_emasa	rft 的 emasa 信号。	0x0
[13:11]	RW	rft_emab	rft 的 emab 信号。	0x4



Bits	Access	Name	Description	Reset
[10:8]	RW	rft_ema	rft 的 ema 信号。	0x4
[7:6]	-	reserved	保留。	0x0
[5:4]	RW	rfs_emaw	rfs 的 emaw 信号。	0x1
[3]	-	reserved	保留。	0x0
[2:0]	RW	rfs_ema	rfs 的 ema 信号。	0x3

KEY_ST

KEY_ST 为 core 使用的 key 的状态信号。

Offset Address: 0x0314 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1:0]	RO	key_req_cur_st	key 的请求状态机。	0x0

CALC_ST0

CALC_ST0 为计算状态寄存器。

Offset Address: 0x0318 Total Reset Value: 0x0000_5000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:12]	RO	hdcp_key_ksv_crc4	hdcp key 的 KSV 的 CRC4 结果。	0x5
[11:8]	RO	hash_calc_cur_st	hash 的状态机状态。	0x0
[7:4]	-	reserved	保留。	0x0
[3:0]	RO	cipher_calc_cur_st	cipher 的状态机状态。	0x0

CALC_ERR

CALC_ERR 为算法 core 告警状态寄存器

Offset Address: 0x0320 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x00000000
[4]	RO	klad_key_write_err	key 写出错	0x0
[3]	RO	out_smmu_page_unv lid	节点或或者数据的 smmu 无效。	0x0
[2]	RO	smmu_page_unvlid	节点或或者数据的 smmu 无效。	0x0
[1]	RO	alg_len_err	选择全空命令或者有对称算法时，长度为 0 中断使能；HASH 的长度为 0；	0x0
[0]	RO	klad_key_use_err	key 使用错误。	0x0

CHANN_HASH_STATE_VAL

CHANN_HASH_STATE_VAL 为通道 m(0~7)的 HASH 输入输出状态值。

Offset Address: $0x340 + m \times 0x8$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	hash_state_val	HASH 结果状态值。	0x00000000

CHANN_HASH_STATE_VAL_ADDR

CHANN_HASH_STATE_VAL_ADDR 为通道 m(0~7)的 hash 输入输出状态值地址。

Offset Address: $0x344 + m \times 0x8$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000
[3:0]	RW	hash_state_val_addr	HASH 状态值地址，最多支持 16word，即 512bit。	0x0

SPACC_CALC_CRG_CFG

SPACC_CALC_CRG_CFG 为 SPACC 内部算法 core 时钟门控与复位配置

Offset Address: 0x039c Total Reset Value: 0x8000_2AAA

Bits	Access	Name	Description	Reset
[31]	RW	spacc_core_auto_cke n_bypass	算法 core 自动时钟门控开关：	0x1



Bits	Access	Name	Description	Reset
			0: 打开自动时钟门控功能; 1: 关闭自动时钟门控功能。	
[30]	RW	spacc_rft_mem_wr_clk_gt_en	rft 类型 memory 写时钟门控: 0: 关闭自动时钟门控功能; 1: 打开自动时钟门控功能。	0x0
[29]	RW	spacc_rft_mem_rd_clk_gt_en	rft 类型 memory 读时钟门控: 0: 关闭自动时钟门控功能; 1: 打开自动时钟门控功能。	0x0
[28]	RW	spacc_rfs_mem_clk_gt_en	rfs 类型 memory 时钟门控: 0: 关闭自动时钟门控功能; 1: 打开自动时钟门控功能。	0x0
[27:14]	-	reserved	保留。	0x0000
[13]	RW	spacc_aes_core_clk_en	aes 算法 core 时钟使能: 0: 不使能; 1: 使能。	0x1
[12]	RW	spacc_aes_core_srst_req	aes 算法 core 软复位请求: 0: 不复位; 1: 复位。	0x0
[11]	RW	spacc_des_core_clk_en	des 算法 core 时钟使能: 0: 不使能; 1: 使能。	0x1
[10]	RW	spacc_des_core_srst_req	des 算法 core 软复位请求: 0: 不复位; 1: 复位。	0x0
[9]	RW	spacc_sm1_core_clk_en	sm1 算法 core 时钟使能: 0: 不使能; 1: 使能。	0x1
[8]	RW	spacc_sm1_core_srst_req	sm1 算法 core 软复位请求: 0: 不复位; 1: 复位。	0x0
[7]	RW	spacc_sm3_core_clk_en	sm3 算法 core 时钟使能: 0: 不使能; 1: 使能。	0x1



Bits	Access	Name	Description	Reset
[6]	RW	spacc_sm3_core_srst_req	sm3 算法 core 软复位请求: 0: 不复位; 1: 复位。	0x0
[5]	RW	spacc_sm4_core_clk_en	sm4 算法 core 时钟使能: 0: 不使能; 1: 使能。	0x1
[4]	RW	spacc_sm4_core_srst_req	sm4 算法 core 软复位请求: 0: 不复位; 1: 复位。	0x0
[3]	RW	spacc_sha1_core_clk_en	sha1 算法 core 时钟使能: 0: 不使能; 1: 使能。	0x1
[2]	RW	spacc_sha1_core_srst_req	sha1 算法 core 软复位请求: 0: 不复位; 1: 复位。	0x0
[1]	RW	spacc_sha2_core_clk_en	sha2 算法 core 时钟使能: 0: 时钟不使能; 1: 时钟使能。	0x1
[0]	RW	spacc_sha2_core_srst_req	sha2 算法 core 软复位请求: 0: 不复位; 1: 复位。	0x0

CHAN0_CIPHER_CTRL

CHAN0_CIPHER_CTRL 为对称算法通道 0 的加解密控制寄存器。

Offset Address: 0x0400 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:25]	-	reserved	保留。	0x00
[24]	RW	sym_ch0_ccm_gcm_pc_last	0: 非 last 块; 1: last 块。	0x0
[23:20]	RW	sym_ch0_gcm_iv_len	gcm 的 iv 长度为(gcm_iv_len+1)Byte	0x0
[19:17]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[16]	RW	sym_ch0_odd_even_key_sel	通道 0 的奇偶 KEY 选择。 0: 选择偶 key 进行计算; 1: 选择奇 key 进行计算。	0x0
[15]	RW	sym_ch0_ivin_sel	ivin 的输入选择控制, 只对通道 0 有效。 0: ivin 不需要进行配置; 1: ivin 需要配置。	0x0
[14]	RW	sym_ch0_key_sel	密钥来源选择。 0: CPU 配置的密钥; 1: 硬件配置的密钥。	0x0
[13:12]	RW	sym_ch0_ccm_gcm_input_flag	在 GCM 或 CCM 模式下。 1)mode 为 GCM: 00: A; 01: P 或者 C; 10: len(A) len(C); 其他: A。 2)mode 为 CCM: 00: N; 01: A; 10: P 或者 C; 其他: N。	0x0
[11:10]	RW	sym_ch0_key_length	密钥长度控制。 AES: 00: 128 位密钥长度; 01: 192 位密钥长度; 10: 256 位密钥长度; 11: 128 位密钥长度。 3DES: 00: 3 个密钥; 01: 3 个密钥; 10: 3 个密钥; 11: 2 个密钥。 SM1: 固定 384 位密钥长度。 SM4: 固定 128 位密钥长度。	0x0



Bits	Access	Name	Description	Reset
[9:8]	RW	sym_ch0_dat_width	数据位宽控制。 DES/3DES: 00: 64 位模式; 01: 8 位模式; 10: 1 位模式; 11: 64 位模式。 AES/SM1: 00: 128 位模式; 01: 8 位模式; 10: 1 位模式; 11: 128 位模式。 SM4: 固定 128 位模式。	0x0
[7]	RW	sym_ch0_decrypt	加解密控制。 0: 加密; 1: 解密。	0x0
[6:4]	RW	sym_ch0_alg_sel	算法类型选择控制。 000: DES 运算; 001: 3DES 运算; 010: AES 运算; 011: SM4 运算; 100: SM1 运算; 其它: AES 运算。	0x0
[3:1]	RW	sym_ch0_alg_mode	工作模式控制。 AES: 000: ECB 模式; 001: CBC 模式; 010: CFB 模式; 011: OFB 模式; 100: CTR 模式; 101: CCM 模式; 110: GCM 模式; 其它: ECB 模式。 TDES/DES/SM1: 000: ECB 模式;	0x0



Bits	Access	Name	Description	Reset
			001: CBC 模式; 010: CFB 模式; 011: OFB 模式; 其它: ECB 模式。 SM4: 000: ECB 模式; 001: CBC 模式; 100: CTR 模式; 其它: :ECB 模式;	
[0]	RW	sym_ch0_start	通道 0 加解密启动(输入传输结束)信号。 0: 不启动(输入传输结束); 1: 启动(输入传输未结束)。	0x0

CIPHER_INT_STATUS

CIPHER_INT_STATUS 为对称算法中断状态寄存器。

Offset Address: 0x0404 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x0000
[16]	RO	ahb_bus_err_int	ahb 总线的错误中断。 0: 无中断; 1: 有中断;	0x0
[15:8]	RO	cipher_chn_obuf_int	通道 m 输出队列数据中断。 0: 无中断; 1: 有中断; bit[15:8]对应通道 7~0。	0x00
[7:1]	RO	cipher_chn_ibuf_int	通道 n 输入队列数据中断。 0: 无中断; 1: 有中断; bit[7:1]对应通道 7~1。	0x00
[0]	-	reserved	保留。	0x0



CIPHER_INT_EN

CIPHER_INT_EN 为对称算法中断使能寄存器。

Offset Address: 0x0408 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	cipher_nsec_int_en	Cipher 模块的非安全通道的总中断使能。 0: 不使能; 1: 使能;	0x0
[30]	RW	cipher_sec_int_en	Cipher 模块的安全通道总安全中断使能, 安全 CPU 才能写。 0: 不使能; 1: 使能;	0x0
[29]	RW	ahb_bus_int_en	ahb 总线的总中断使能。 0: 不使能; 1: 使能;	0x0
[28:17]	-	reserved	保留。	0x000
[16]	RW	ahb_bus_err_int_en	ahb 总线的错误中断使能。 0: 不使能; 1: 使能;	0x0
[15:8]	RW	cipher_chn_obuf_en	通道 m 输出队列数据中断使能。 0: 不使能; 1: 使能; bit[15:8]对应通道 7~0。	0x00
[7:1]	RW	cipher_chn_ibuf_en	通道 n 输入队列数据中断使能。 0: 不使能; 1: 使能; bit[7:1]对应通道 7~1。	0x00
[0]	-	reserved	保留。	0x0

CIPHER_INT_RAW

CIPHER_INT_RAW 为对称算法中断状态寄存器。

Offset Address: 0x040C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x0000
[16]	RWC	ahb_bus_err_int_raw	ahb 总线的错误原始中断。 0: 无中断; 1: 有中断;	0x0
[15:8]	RWC	cipher_chn_obuf_raw	通道 m 输出队列数据原始中断。 0: 无中断; 1: 有中断; bit[15:8]对应通道 7~0。	0x00
[7:1]	RWC	cipher_chn_ibuf_raw	通道 n 输入队列数据原始中断。 0: 无中断; 1: 有中断; bit[7:1]对应通道 7~1。	0x00
[0]	-	reserved	保留。	0x0

CIPHER_IN_SMMU_EN

CIPHER_IN_SMMU_EN 为通道 1~7 对称算法的输入数据读地址 bypass。

Offset Address: 0x0410 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:23]	-	reserved	保留。	0x000
[22:16]	RW	cipher_in_chann_rd_node_smmu_en	输入的通道 n 的读节点 smmu 使能。 0: 不使能; 1: 使能; bit[22:16]对应通道 7~1。	0x00
[15:7]	-	reserved	保留。	0x000
[6:0]	RW	cipher_in_chann_rd_dat_smmu_en	输入的通道 n 的读数据 smmu 使能。 0: 不使能; 1: 使能; bit[6:0]对应通道 7~1。	0x00

OUT_SMMU_EN

OUT_SMMU_EN 为通道 1~7 对称算法的输出结果读地址 bypass。



Offset Address: 0x0414 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:23]	-	reserved	保留。	0x000
[22:16]	RW	out_chann_rd_node_smmu_en	输出的通道 n 的读节点 smmu 使能。 0: 不使能; 1: 使能; bit[22:16]对应通道 7~1。	0x00
[15:7]	-	reserved	保留。	0x000
[6:0]	RW	out_chann_wr_dat_smmu_en	输出的通道 7 的写数据 smmu 使能。 0: 不使能; 1: 使能; bit[6:0]对应通道 7~1。	0x00

CHAN0_CIPHER_DIN

CHAN0_CIPHER_DIN 为对称算法通道 0 数据输入寄存器。

Offset Address: 0x420+k×0x04 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	chan0_cipher_din	通道 0 的对称算法数据输入，小端格式。 AES/SM1: 1)如果选择 1-CFB 操作，chan0_cipher_din[0]为有效数据。 2)如果选择 8-CFB 操作，chan0_cipher_din[7:0]为有效数据。 3)如果选择 128-CFB 操作，128 位数据均有效。 4)如果选择其他操作模式，128 位数据均有效。 DES/3DES: 1)如果选择 1-CFB/1-OFB 操作，chan0_cipher_din[0]为有效数据。 2)如果选择 8-CFB/8-OFB 操作，chan0_cipher_din[7:0]为有效数据。 3)如果选择 64-CFB/64-OFB 操作，chan0_cipher_din[63:0]为有效数据。 4)如果选择其他操作模式，低 64 位数有效，chan0_cipher_din[63:0]为有效数据。 SM4: chan0_cipher_din[127:0]为有效数据。	0x00000000

NORM_SMMU_START_ADDR

NORM_SMMU_START_ADDR 为非安全 SMMU 起始地址寄存器。

Offset Address: 0x0440 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	norm_smmu_start_addr	非安全 smmu 页表的起始地址(若为 SYM 算法则为前 128 个节点对应的页表起始地址)。	0x00000000

SEC_SMMU_START_ADDR

SEC_SMMU_START_ADDR 为安全 SMMU 起始地址寄存器。

Offset Address: 0x0444 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	sec_smmu_start_addr	安全 smmu 页表的起始地址(若为 SYM 算法则为前 128 个节点对应的页表起始地址)。	0x00000000

IN_ST

IN_ST 为输入状态寄存器。

Offset Address: 0x0450 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:12]	-	reserved	保留。	0x00000
[11:8]	RO	sym_hash_req_cur_st	sym 和 hash 的请求状态。	0x0
[7:4]	RO	sym_in_ctrl_cur_st	对称算法 IN_CTRL 的状态机状态。	0x0
[3:0]	RO	hash_in_ctrl_cur_st	HASH IN_CTRL 的状态机状。	0x0

OUT_ST

OUT_ST 为输出状态寄存器。

Offset Address: 0x0454 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4:0]	RO	out_cur_st	OUT_CTRL 的状态机状态	0x00

CHANN_CIPHER_CTRL

CHANN_CIPHER_CTRL 为通道 n 对称算法的加解密控制寄存器。

Offset Address: 0x400+n×0x80 Total Reset Value: 0x0040_0000

Bits	Access	Name	Description	Reset
[31:22]	RW	weight	当前通道的权重,weight * 4block(0 的时候也是 4block)。	0x001
[21:18]	-	reserved	保留。	0x0
[17]	RW	sym_chn_din_byte_s	输入数据 Byte 顺序交换。	0x0



Bits	Access	Name	Description	Reset
		wap_en	0: 不使能; 1: 使能。	
[16]	RW	sym_chn_dout_byte_swap_en	输出数据 Byte 顺序交换。 0: 不使能; 1: 使能。	0x0
[15]	-	reserved	保留。	0x0
[14]	RW	sym_chn_key_sel	密钥来源选择。 0: CPU 配置的密钥; 1: 硬件配置的密钥。	0x0
[13:12]	-	reserved	保留。	0x0
[11:10]	RW	sym_chn_key_length	密钥长度控制。 AES: 00: 128 位密钥长度; 01: 192 位密钥长度; 10: 256 位密钥长度; 11: 128 位密钥长度。 3DES: 00: 3 个密钥; 01: 3 个密钥; 10: 3 个密钥; 11: 2 个密钥。 SM1: 固定 384 位密钥长度。 SM4: 固定 128 位密钥长度。	0x0
[9:8]	RW	sym_chn_dat_width	数据位宽控制。 DES/3DES: 00: 64 位模式; 01: 8 位模式; 10: 1 位模式; 11: 64 位模式。 AES/SM1: 00: 128 位模式; 01: 8 位模式; 10: 1 位模式;	0x0



Bits	Access	Name	Description	Reset
			11: 128 位模式。 SM4: 固定 128 位模式。	
[7]	RW	sym_chn_decrypt	加解密控制。 0: 加密; 1: 解密。	0x0
[6:4]	RW	sym_chn_alg_sel	算法类型选择控制。 000: DES 运算; 001: 3DES 运算; 010: AES 运算; 011: SM4 运算; 100: SM1 运算; 101: DMA 拷贝; 其它: AES 运算。	0x0
[3:1]	RW	sym_chn_alg_mode	工作模式控制。 AES: 000: ECB 模式; 001: CBC 模式; 010: CFB 模式; 011: OFB 模式; 100: CTR 模式; 101: CCM 模式; 110: GCM 模式; 其它: ECB 模式。 TDDES/DES/SM1: 000: ECB 模式; 001: CBC 模式; 010: CFB 模式; 011: OFB 模式; 其它: ECB 模式。 SM4: 000: ECB 模式; 001: CBC 模式; 100: CTR 模式; 其它: ECB 模式;	0x0



Bits	Access	Name	Description	Reset
[0]	-	reserved	保留。	0x0

CHANN_CIPHER_IN_NODE_CFG

CHANN_CIPHER_IN_NODE_CFG 为通道 n 的输入节点配置寄存器。

Offset Address: $0x404 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RW	cipher_in_node_total_num	输入节点总深度。	0x00
[23:16]	RW	cipher_in_node_wptr	输入节点写指针。	0x00
[15:8]	RO	cipher_in_node_rptr	输入节点读指针。	0x00
[7:0]	RW	cipher_in_node_mpa ckage_int_level	输入节点多包中断水准。	0x00

CHANN_CIPHER_IN_NODE_START_ADDR

CHANN_CIPHER_IN_NODE_START_ADDR 为通道 n 的输入节点起始地址寄存器。

Offset Address: $0x408 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	cipher_in_node_start_addr	输入节点起始地址，通过此地址读取节点。	0x00000000

CHANN_CIPHER_IN_BUF_RPTR

CHANN_CIPHER_IN_BUF_RPTR 为通道 n 的输入 BUF 读地址寄存器。

Offset Address: $0x40C + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	cipher_in_buf_rptr	输入 buffer 的读地址。	0x00000000

CHANN_CIPHER_IN_LEFT_BYTE

CHANN_CIPHER_IN_LEFT_BYTE 为通道 n 当前读入数据中不足一块的剩余数据。



Offset Address: $0x410 + n \times 0x80$ Total Reset Value: $0x0000\_0000$

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000
[3:0]	RO	in_left_byte	读数据的剩余字节个数。	0x0

CHANN_CIPHER_IN_LEFT_WORD0

CHANN_CIPHER_IN_LEFT_WORD0 为通道 n 当前读入数据中不足一块的剩余数据。

Offset Address: $0x414 + n \times 0x80$ Total Reset Value: $0x0000\_0000$

Bits	Access	Name	Description	Reset
[31:0]	RO	cipher_in_left_word 0	剩余数据的第 1 个 word。	0x00000000

CHANN_CIPHER_IN_LEFT_WORD1

CHANN_CIPHER_IN_LEFT_WORD1 为通道 n 当前读入数据中不足一块的剩余数据。

Offset Address: $0x418 + n \times 0x80$ Total Reset Value: $0x0000\_0000$

Bits	Access	Name	Description	Reset
[31:0]	RO	cipher_in_left_word 1	剩余数据的第 2 个 word。	0x00000000

CHANN_CIPHER_IN_LEFT_WORD2

CHANN_CIPHER_IN_LEFT_WORD2 为通道 n 当前读入数据中不足一块的剩余数据。

Offset Address: $0x41C + n \times 0x80$ Total Reset Value: $0x0000\_0000$

Bits	Access	Name	Description	Reset
[31:0]	RO	cipher_in_left_word 2	剩余数据的第 3 个 word。	0x00000000

CHANN_CIPHER_IN_BUF_REST_LEN

CHANN_CIPHER_IN_BUF_REST_LEN 为通道 n 输入 buf 的剩余长度。

Offset Address: $0x420 + n \times 0x80$ Total Reset Value: $0x0000\_0000$



Bits	Access	Name	Description	Reset
[31:0]	RO	cipher_in_buf_rest_len	通道 n 输入 buf 的剩余长度，以字节为单位。	0x00000000

CHANN_CIPHER_OUT_NODE_CFG

CHANN_CIPHER_OUT_NODE_CFG 为通道 n 的输出节点配置寄存器。

Offset Address: $0x430 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RW	cipher_out_node_total_num	输出节点总深度。	0x00
[23:16]	RW	cipher_out_node_wptr	输出节点写指针。	0x00
[15:8]	RO	cipher_out_node_rptr	输出节点读指针。	0x00
[7:0]	RW	cipher_out_node_multicast_int_level	输出节点多包中断水线。	0x00

CHANN_CIPHER_OUT_NODE_START_ADDR

CHANN_CIPHER_OUT_NODE_START_ADDR 为通道 n 的输出节点起始地址寄存器。

Offset Address: $0x434 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	cipher_out_node_start_addr	输出节点起始地址，通过此地址读取节点。	0x00000000

CHANN_CIPHER_OUT_BUF_RPTR

CHANN_CIPHER_OUT_BUF_RPTR 为通道 n 的输出 BUF 读地址寄存器。

Offset Address: $0x438 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	cipher_out_buf_rptr	输出 buffer 的读地址。	0x00000000



CHANN_CIPHER_OUT_LEFT_BYTE

CHANN_CIPHER_OUT_LEFT_BYTE 为通道 n 当前输出数据中不足一块的剩余数据。

Offset Address: $0x440 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	保留。	0x00000000
[6:0]	RO	out_left_byte	写数据的剩余字节个数。	0x00

CHANN_CIPHER_OUT_LEFT_WORD0

CHANN_CIPHER_OUT_LEFT_WORD0 为通道 n 当前数据中不足一块的剩余数据。

Offset Address: $0x444 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	cipher_out_left_word0	剩余数据的第 1 个 word。	0x00000000

CHANN_CIPHER_OUT_LEFT_WORD1

CHANN_CIPHER_OUT_LEFT_WORD1 为通道 n 当前数据中不足一块的剩余数据。

Offset Address: $0x448 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	cipher_out_left_word1	剩余数据的第 2 个 word。	0x00000000

CHANN_CIPHER_OUT_LEFT_WORD2

CHANN_CIPHER_OUT_LEFT_WORD2 为通道 n 当前数据中不足一块的剩余数据。

Offset Address: $0x44C + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	cipher_out_left_word2	剩余数据的第 3 个 word。	0x00000000



CHANN_CIPHER_OUT_BUF_REST_LEN

CHANN_CIPHER_OUT_BUF_REST_LEN 为通道 n 输入 buf 的剩余长度。

Offset Address: $0x450 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	cipher_out_buf_rest_len	通道 n 输入 buf 的剩余长度，以字节为单位。	0x00000000

CHANN_CIPHER_IN_NODE_START_ADDR_HIGH

CHANN_CIPHER_IN_NODE_START_ADDR_HIGH 为通道 n 的输入节点起始地址寄存器。

Offset Address: $0x460 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	cipher_in_node_start_addr_high	输入节点起始地址高 32bit，通过此地址读取节点。	0x00000000

CHANN_CIPHER_IN_BUF_RPTR_HIGH

CHANN_CIPHER_IN_BUF_RPTR_HIGH 为通道 n 的输入 BUF 读地址寄存器。

Offset Address: $0x464 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	cipher_in_buf_rptr_high	输入 buffer 的读地址高 32bit。	0x00000000

CHANN_CIPHER_OUT_NODE_START_ADDR_HIGH

CHANN_CIPHER_OUT_NODE_START_ADDR_HIGH 为通道 n 的输出节点起始地址寄存器。

Offset Address: $0x470 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	cipher_out_node_start_addr_high	输出节点起始地址高 32bit，通过此地址读取节点。	0x00000000



CHANN_CIPHER_OUT_BUF_RPTR_HIGH

CHANN_CIPHER_OUT_BUF_RPTR_HIGH 为通道 n 的输出 BUF 读地址寄存器。

Offset Address: $0x474 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	cipher_out_buf_rptr_high	输出 buffer 的读地址高 32bit。	0x00000000

CHAN0_HASH_CTRL

CHAN0_HASH_CTRL 为通道 0 的 HASH 控制寄存器。

Offset Address: 0x0800 Total Reset Value: 0x0000_0200

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12:10]	RW	hash_ch0_hmac_key_addr	在 hmac key 计算的时候, key 的地址, 地址 0~7 对应通道 0~7。	0x0
[9]	-	reserved	保留。	0x1
[8:7]	-	reserved	保留。	0x0
[6]	RW	hash_ch0_key_sel	HMAC 的 key 选择。 0: CPU 配置的 key; 1: 硬件配置的 key。	0x0
[5]	RW	hash_ch0_mode	计算模式选择。 0: RAW HASH 计算; 1: HMAC 计算。	0x0
[4]	RW	hash_ch0_hmac_calc_step	如果选择 HMAC, 为 HMAC 计算阶段选择: 0: key 计算; 1: message 的 hash 计算。	0x0



[3:1]	RW	hash_ch0_agl_sel	算法模式选择。 000: SHA1; 001: SHA244; 010: SHA256; 011: SHA384; 100: SHA512; 101: SM3; 其它: SHA1。	0x0
[0]	RW	hash_ch0_start	hash 的 block 启动信号。 0: 不启动; 1: 启动。	0x0

HASH_INT_STATUS

HASH_INT_STATUS 为 HASH 中断状态寄存器。

Offset Address: 0x0804 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:18]	RO	hash_chn_oram_int	通道 n 输出 RAM 数据中断。 0: 无中断; 1: 有中断; bit[25:18]对应通道 7~0。	0x00
[17:0]	RO	reserved	保留。	0x00000

HASH_INT_EN

HASH_INT_EN 为 HASH 中断使能寄存器。

Offset Address: 0x0808 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	hash_int_en	HASH 的总中断使能。 0: 不使能; 1: 使能。	0x0



[30]	RW	hash_sec_int_en	HASH 的总安全中断使能，安全访问才能写。 0: 不使能; 1: 使能。	0x0
[29:26]	-	reserved	保留。	0x0
[25:18]	RW	hash_chn_oram_en	通道 n 输出队列数据中断使能，写入结果 RAM 报中断。 0: 不使能; 1: 使能; bit[25:18]对应通道 7~0。	0x00
[17:0]	RO	reserved	保留。	0x00000

HASH_INT_RAW

HASH_INT_RAW 为 HASH 原始中断状态寄存器。

Offset Address: 0x080C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:18]	RW	hash_chn_oram_raw	通道 n 输出队列数据原始中断。 0: 无中断; 1: 有中断; bit[25:18]对应通道 7~0。	0x00
[17:0]	RO	reserved	保留。	0x00000

HASH_IN_SMMU_EN

HASH_IN_SMMU_EN 为通道 1~7 的读地址 bypass。

Offset Address: 0x0810 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:23]	-	reserved	保留。	0x000
[22:16]	RW	hash_in_chann_rd_node_smmu_en	输入的通道 n 的读节点 smmu 使能。 0: 不使能; 1: 使能; bit[22:16]对应通道 7~1。	0x00



[15:7]	-	reserved	保留。	0x000
[6:0]	RW	hash_in_chann_rd_d at_smmu_en	输入的通道 n 的读数据 smmu 使能。 0: 不使能; 1: 使能; bit[6:0]对应通道 7~1。	0x00

CHAN0_HASH_DAT_IN

CHAN0_HASH_DAT_IN 为 CPU 输入数据。

Offset Address: 0x0818 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	WO	chan0_hash_data_in	chan0 的 CPU 数据输入寄存器。	0x00000000

CHAN0_HASH_TOTAL_DAT_LEN

CHAN0_HASH_TOTAL_DAT_LEN 为 CPU 输入数据长度。

Offset Address: 0x081C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	chan0_hash_total_da t_len	chan0 的 CPU 数据长度寄存器，单位为 byte。	0x00000000

CHAN0_HASH_IN_DAT_LEN

CHAN0_HASH_IN_DAT_LEN 为通道 0 的已输入数据长度。

Offset Address: 0x0820 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	chan0_hash_in_dat_l en	chan0 的已输入数据长度，单位为 hash block。	0x00000000

CHANN_HASH_CTRL

CHANN_HASH_CTRL 为通道 n(1~7)的 HASH 控制寄存器。

Offset Address: 0x800+n×0x80 Total Reset Value: 0x0000_0200



Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x00000
[9]	-	reserved	保留。	0x1
[8]	-	reserved	保留。	0x0
[7]	RW	hash_chn_dat_in_byte_swap_en	输入数据 Byte 顺序交换。 0: 不使能; 1: 使能。	0x0
[6]	RW	hash_chn_key_sel	HMAC 的 key 选择。 0: CPU 配置的 key; 1: 硬件配置的 key。	0x0
[5]	RW	hash_chn_mode	计算模式选择。 0: RAW HASH 计算; 1: HMAC 计算。	0x0
[4]	-	reserved	保留。	0x0
[3:1]	RW	hash_chn_agl_sel	算法模式选择。 000: SHA1; 001: SHA244; 010: SHA256; 011: SHA384; 100: SHA512; 101: SM3; 其它: SHA1。	0x0
[0]	-	reserved	保留。	0x0

CHANN_HASH_IN_NODE_CFG

CHANN_HASH_IN_NODE_CFG 为通道 n(1~7)的输入节点配置寄存器。

Offset Address: $0x804 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RW	hash_in_node_total_num	输入节点总深度。	0x00
[23:16]	RW	hash_in_node_wptr	输入节点写指针。	0x00
[15:8]	RO	hash_in_node_rptr	输入节点读指针。	0x00



[7:0]	RW	hash_in_node_mpac kage_int_level	输入节点多包中断水线。	0x00

CHANN_HASH_IN_NODE_START_ADDR

CHANN_HASH_IN_NODE_START_ADDR 为通道 n(1~7)的输入节点起始地址寄存器。

Offset Address: $0x808 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	hash_in_node_start_ addr	输入节点起始地址，通过此地址读取节点。	0x00000000

CHANN_HASH_IN_BUF_RPTR

CHANN_HASH_IN_BUF_RPTR 为通道 n(1~7)的输入 BUF 读地址寄存器。

Offset Address: $0x80C + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	hash_in_buf_rptr	输入 buffer 的读地址。	0x00000000

CHANN_HASH_IN_BUF_REST_LEN

CHANN_HASH_IN_BUF_REST_LEN 为通道 n 输入 buf 的剩余长度。

Offset Address: $0x810 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	hash_in_buf_rest_len	通道 n 输入 buf 的剩余长度，以字节为单位。	0x00000000

CHANN_HASH_IN_NODE_START_ADDR_HIGH

CHANN_HASH_IN_NODE_START_ADDR_HIGH 为通道 n(1~7)的输入节点起始地址寄存器。

Offset Address: $0x820 + n \times 0x80$ Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	hash_in_node_start_addr_high	输入节点起始地址高 32bit，通过此地址读取节点。	0x00000000

CHANN_HASH_IN_BUF_RPTR_HIGH

CHANN_HASH_IN_BUF_RPTR_HIGH 为通道 n(1~7)的输入 BUF 读地址寄存器。

Offset Address: $0x824 + n \times 0x80$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	hash_in_buf_rptr_high	输入 buffer 的读地址高 32bit。	0x00000000

13.2 HISEC_TRNG_CTRL

13.2.1 概述

HISEC_TRNG_CTRL 是一个能够产生真随机数的模块。产生的真随机数符合 FIPS140-1 的随机测试标准。

13.2.2 特点

HISEC_TRNG_CTRL 模块有如下特点：

- 支持真随机数产生。
- 支持三个随机源选择，CPU 可配置。

13.2.3 寄存器概览

HISEC_TRNG_CTRL 寄存器概览如表 13-3 所示。

表13-3 HISEC_TRNG_CTRL 寄存器概览（基址是 0x1009_0000）

偏移地址	名称	描述	页码
0x0200	HISEC_COM_TRNG_CTRL	TRNG 控制寄存器。	13-47
0x0204	HISEC_COM_TRNG_FIFO_DATA	TRNG 的 FIFO 数据寄存器。	13-48
0x0208	HISEC_COM_TRNG_DATA_STA	TRNG 的 FIFO 数据寄存器的状态。	13-48



偏移地址	名称	描述	页码
0x020C	HISEC_COM_TRNG_ER R0_CNT	RNG 随机数异常检测为全 0 的次数统计。	13-49
0x0210	HISEC_COM_TRNG_ER R1_CNT	RNG 随机数异常检测为全 1 的次数统计。	13-49
0x0214	HISEC_COM_TRNG_ER R2_CNT	RNG 随机数异常检测为相邻数据出现相同的次数统计。	13-50
0x0218	HISEC_COM_TRNG_ER R3_CNT	RNG 随机数异常检测为相邻 256bit 数据的低 32bit 出现相同的次数统计。	13-50

13.2.4 寄存器描述

HISEC_COM_TRNG_CTRL

HISEC_COM_TRNG_CTRL 为 TRNG 控制寄存器

Offset Address: 0x0200 Total Reset Value: 0x0000_000A

Bits	Access	Name	Description	Reset
[31:18]	-	reserved	保留。	0x0000
[17]	RW	trng_cfg_lock	trng_ctrl 的配置 (HISEC_COM_TRNG_CTRL[16:0])锁定信号，写 1 之后，不能写 0，全局复位才能复位。 0: 不锁定； 1: 锁定。	0x0
[16]	RW	trng_sel	熵源 TRNG 输出选择。 0: 选择海思的 OSC 模拟随机源数据输出； 1: 选择 RO 随机源数据输出。	0x0
[15:8]	RW	post_process_depth	后处理混合深度，越大随机效果越好，但是速度越慢。	0x00
[7]	RW	post_process_enable	随机数前处理使能，使用此功能后随机数产生速度降低(post_process_depth+1)倍。 0: 不使能； 1: 使能。	0x0
[6]	RW	mix_enable	数据混合。	0x0



Bits	Access	Name	Description	Reset
			0: 不使能; 1: 使能。	
[5]	RW	drop_enable	丢掉不等概率数据使能, 使用此功能后随机数产生速度降低 3 倍。 0: 不使能; 1: 使能。	0x0
[4]	RW	pre_process_enable	随机数前处理使能, 使用此功能后随机数产生速度降低 4 倍。 0: 不使能; 1: 使能。	0x0
[3]	RW	drbg_enable	DRBG 后处理使能。 0: 不使能; 1: 使能。	0x1
[2]	WC	cleardata	写 1 清空 fifo 中的随机数	0x0
[1:0]	RW	osc_sel	随机源选择。 00: disable RNG; 01: 使用随机源 1; 10: 使用随机源 2; 11: 使用随机源 3。	0x2

HISEC_COM_TRNG_FIFO_DATA

HISEC_COM_TRNG_FIFO_DATA 为 TRNG 的 FIFO 数据寄存器

Offset Address: 0x0204 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	trng_fifo_data	外边的接口可以直接读取本寄存器, 直接返回 fifo 中的 32bit 数据。	0x0000_0000

HISEC_COM_TRNG_DATA_ST

HISEC_COM_TRNG_DATA_ST 为 TRNG 的 FIFO 数据寄存器的状态

Offset Address: 0x0208 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:8]	RO	trng_fifo_data_cnt	fifo 中随机数的个数。	0x00
[7:4]	-	reserved	保留。	0x0
[3]	WC	low_ro_st1	为 1 表示低频振荡器时钟输出一直为 1(系统时钟检测中硬 RO 源连续出现 512 个 1)，写清零	0x0
[2]	WC	low_ro_st0	为 1 表示低频振荡器时钟输出一直为 0(系统时钟检测中硬 RO 源连续出现 512 个 0)，写清零	0x0
[1]	WC	low_osc_st1	为 1 表示低频振荡器时钟输出一直为 1，写清零	0x0
[0]	WC	low_osc_st0	为 1 表示低频振荡器时钟输出一直为 0，写清零	0x0

HISEC_COM_TRNG_ERR0_CNT

HISEC_COM_TRNG_ERR0_CNT 为 RNG 随机数异常检测为全 0 的次数统计

Offset Address: 0x020C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RO	rng_err0_cnt	RNG 随机数异常检测为全 0 的次数统计	0x0000

HISEC_COM_TRNG_ERR1_CNT

HISEC_COM_TRNG_ERR1_CNT 为 RNG 随机数异常检测为全 1 的次数统计

Offset Address: 0x0210 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RO	rng_err1_cnt	RNG 随机数异常检测为全 1 的次数统计	0x0000



HISEC_COM_TRNG_ERR2_CNT

HISEC_COM_TRNG_ERR2_CNT 为 RNG 随机数异常检测为相邻数据出现相同的次数统计

Offset Address: 0x0214 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RO	rng_err2_cnt	RNG 随机数异常检测为相邻数据出现相同的次数统计	0x0000

HISEC_COM_TRNG_ERR3_CNT

HISEC_COM_TRNG_ERR3_CNT 为 RNG 随机数异常检测为相邻 256bit 数据的低 32bit 出现相同的次数统计

Offset Address: 0x0218 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RO	rng_err3_cnt	RNG 随机数异常检测为相邻 256bit 数据的低 32bit 出现相同的次数统计	0x0000

13.3 RSA

13.3.1 概述

RSA 是一个公钥加解密算法，主要通过模幂计算实现加解密。加密： $C = M^E \bmod N$ ；解密： $M = C^D \bmod N$ 。M 是明文，C 是密文，(N,E)为公钥，(N,D) 为私钥。

RSA 支持 RSA1024/2048/4096，符合 PKCS#1 V1.5/2.1 标准。

RSA 模块适用于数据加解密、数字签名校验。

13.3.2 特点

RSA 具备以下特点：

- 支持 1024bit、2048bit、4096bit 的密钥位宽。
- 支持 2048 密钥位宽性能最低为 40 ops/s（40 次/秒）。
- 支持 RSA 内部密钥 RAM、报文 RAM、结果 RAM 的清 0。



13.3.3 寄存器概览

RSA 寄存器概览如表 13-4 所示。

表13-4 RSA 寄存器概览（RSA0 基址是 0x1008_0000；RSA1 基址是 0x100D_0000）

偏移地址	名称	描述	页码
0x50	SEC_RSA_BUSY_REG	SEC_RSA 模块 BUSY 状态寄存器	13-51
0x54	SEC_RSA_MOD_REG	SEC_RSA 模块工作模式寄存器	13-51
0x58	SEC_RSA_WSEC_REG	SEC_RSA 模块写密钥寄存器	13-52
0x5C	SEC_RSA_WPKT_REG	SEC_RSA 模块写报文数据寄存器	13-52
0x60	SEC_RSA_RPKT_REG	SEC_RSA 模块读报文寄存器	13-53
0x64	SEC_RSA_RRSULT_REG	SEC_RSA 模块读计算结果寄存器	13-53
0x68	SEC_RSA_START_REG	SEC_RSA 模块启动模幂配置寄存器	13-53
0x6C	SEC_RSA_ADDR_REG	SEC_RSA 模块密钥、报文、结果的 RAM 地址寄存器	13-53
0x70	SEC_RSA_ERROR_REG	SEC_RSA 模块错误报警状态寄存器	13-54
0x74	SEC_CRC16_REG	SEC_RSA 的密钥 CRC 计算结果	13-55

13.3.4 RSA 寄存器描述

SEC_RSA_BUSY_REG

SEC_RSA_BUSY_REG 为 SEC_RSA 模块 BUSY 状态寄存器。

Offset Address: 0x50 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留。	0x0000_0000
[0]	RO	sec_rsa_busy_reg	SEC_RSA 的 BUSY 状态寄存器。 0: 模块处于空闲状态； 1: 模块处于 BUSY 状态。	0x0

SEC_RSA_MOD_REG

SEC_RSA_MOD_REG 为 SEC_RSA 模块工作模式寄存器。

Offset Address: 0x54 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:7]	-	reserved	保留。	0x000000
[6]	RW	sec_rsa_data2_clr	清除 RSA 结果 RAM 使能。 0: 不对结果 RAM 进行清 0; 1: 清除存储结果的 RAM 数据。	0x0
[5]	RW	sec_rsa_data1_clr	清除 RSA 报文 RAM 使能。 0: 不对报文 RAM 进行清 0; 1: 清除存储报文的 RAM 数据。	0x0
[4]	RW	sec_rsa_data0_clr	清除 RSA 密钥 RAM 使能。 0: 不对密钥 RAM 进行清 0; 1: 清除存储密钥的 RAM 数据。	0x0
[3:2]	RW	sec_rsa_key_width	密钥位宽选择。 00: 密钥位宽为 1024; 01: 密钥位宽为 2048; 10: 密钥位宽为 4096; 11: 密钥位宽为 1024。	0x0
[1:0]	RW	sec_rsa_mod_sel	工作模式选择。 00: 模幂加解密计算; 01: 保留; 10: 对 RAM 进行清 0 操作; 11: 对密钥进行 CRC16 计算。	0x0

SEC_RSA_WSEC_REG

SEC_RSA_WSEC_REG 为 SEC_RSA 模块写密钥寄存器。

Offset Address: 0x58 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	WO	sec_rsa_wsec_reg	CPU 配置密钥寄存器，需要根据配置的 RSA 密钥位宽确定配置的次数。	0x0000_0000

SEC_RSA_WPKT_REG

SEC_RSA_WPKT_REG 为 SEC_RSA 模块写报文数据寄存器。

Offset Address: 0x5C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	WO	sec_rsa_wpkt_reg	CPU 配置报文寄存器，需要根据配置的 RSA 密钥位宽确定配置的次数。	0x0000_0000

SEC_RSA_RPKT_REG

SEC_RSA_RPKT_REG 为 SEC_RSA 模块读报文寄存器。

Offset Address: 0x60 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	sec_rsa_rpkt_reg	CPU 读取报文寄存器，需要根据配置的 RSA 密钥位宽确定配置的次数。	0x0000_0000

SEC_RSA_RRSULT_REG

SEC_RSA_RRSULT_REG 为 SEC_RSA 模块读计算结果寄存器。

Offset Address: 0x64 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	sec_rsa_rrslt_reg	CPU 运算结果寄存器，需要根据配置的 RSA 密钥位宽确定配置的次数。	0x0000_0000

SEC_RSA_START_REG

SEC_RSA_START_REG 为 SEC_RSA 模块启动模幂配置寄存器。

Offset Address: 0x68 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x0000_0000
[0]	RW	sec_rsa_start_reg	CPU 配置启动。 0: 不启动 RSA; 1: 启动 RSA。	0x0

SEC_RSA_ADDR_REG

SEC_RSA_ADDR_REG 为 SEC_RSA 模块密钥、报文、结果的 RAM 地址寄存器。



Offset Address: 0x6C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RO	sec_rsa_addere_rslt	结果 RAM 地址。 每次地址偏移值为 4B；即如果该值指示为 1，则表示内部已存储了 4B 的数据；由于从 0 开始计数，存储到密钥位宽最大值时，该值为 0。	0x00
[23:16]	RO	sec_rsa_addre_pkt	报文 RAM 地址。 每次地址偏移值为 4B；即如果该值指示为 1，则表示内部已存储或读取了 4B 的数据；由于从 0 开始计数，存储到密钥位宽最大值时，该值为 0。	0x00
[15:8]	RO	sec_rsa_addre_d	密钥参数 E/D RAM 地址。 每次地址偏移值为 4B；即如果该值指示为 1，则表示内部已存储了 4B 的数据；由于从 0 开始计数，存储到密钥位宽最大值时，该值为 0。	0x00
[7:0]	RO	sec_rsa_addr_n	密钥参数 N RAM 地址。 每次地址偏移值为 4B；即如果该值指示为 1，则表示内部已存储了 4B 的数据；由于从 0 开始计数，存储到密钥位宽最大值时，该值为 0。	0x00

SEC_RSA_ERROR_REG

SEC_RSA_ERROR_REG 为 SEC_RSA 模块错误报警状态寄存器。

Offset Address: 0x70 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x0000_0000
[1]	RO	sec_rsa_err1	当前外部 RSA 输入密钥 N 偶数告警。(密钥 RAM 清 0 时，不进行检测) 0: 外部输入密钥无异常； 1: 外部输入密钥 N 为偶数。	0x0
[0]	-	reserved	保留。	0x0



SEC_CRC16_REG

SEC_CRC16_REG 为 SEC_RSA 的密钥 CRC 计算结果。

Offset Address: 0x74 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	保留。	0x0000
[15:0]	RO	sec_rsa_crc16_dat	密钥的 CRC 计算结果。	0x0000

13.4 OTP Ctrl

13.4.1 工作方式

OTP (One Time Programmable) Ctrl 用于对 OTP 进行编程和读取。OTP Ctrl 有七种工作模式：

- 读取 OTP 中的 lock 状态模式；
- 加载 OTP 的 key 到加密模块模式；
- 烧写 key 和 jtag ID/PASSWORD 模式；
- key 和 jtag ID/PASSWORD 的 CRC 校验模式；
- 使能标志位烧写模式；
- 用户预留空间烧写模式；
- 用户预留空间回读模式。

13.4.1.1 读取 OTP 中的 lock 状态

- 步骤 1. 读取 [OTP_USER_CTRL_STA](#)[otp_op_busy]位，直到回读值为 0，表示 OTP 控制器空闲，可以发起操作。
- 步骤 2. 往 [OTP_USER_WORK_MODE](#) 写 0x0(读取 OTP 中的 lock 状态模式)；
- 步骤 3. 往 [OTP_USER_OP_START](#) 写 0x1ACC_E551，启动命令；
- 步骤 4. 读取 [OTP_USER_CTRL_STA](#)[otp_user_cmd_finish]位，直到回读值为 1，表示操作完成；
- 步骤 5. 读取 [OTP_USER_LOCK_STA0](#) 的值，可得到 OTP 的 LOCK 状态。

----结束

13.4.1.2 加载 OTP 的 key 到加密模块

- 步骤 1. 读取 [OTP_USER_CTRL_STA](#)[otp_op_busy]位，直到回读值为 0，表示 OTP 控制器空闲，可以发起操作。



- 步骤 2. 往 **OTP_USER_KEY_INDEX** 写需要加载的 Key (0x0: key0; 0x1: key1; 0x2: key2; 0x3: key3; 其他值: key0;)
- 步骤 3. 往 **OTP_USER_WORK_MODE** 写 0x1(加载 OTP 的 key 到加密模块模式);
- 步骤 4. 往 **OTP_USER_OP_START** 写 0x1ACC_E551, 启动命令;
- 步骤 5. 读取 **OTP_USER_CTRL_STA**[otp_user_cmd_finish]位, 直到回读值为 1, 表示操作完成;

----结束

13.4.1.3 烧写 KEY、JTAG ID、PASSWORD 到 OTP

烧写之前, OTP 中的数据全部为 0; 烧写使能后, OTP Ctrl 将根据 **OTP_USER_KEY_INDEX** 的值将 OTP_USER_KEY_DATA[n](n=0~8)烧写到 OTP。

- 步骤 1. 按照 13.4.1.1 的方法读取要烧写的 key 的 lock 状态; 如果 lock 状态位为 1, 则对应的 key 不能烧写; 如果 lock 状态位为 0, 继续步骤 2;
- 步骤 2. 读取 **OTP_USER_CTRL_STA**[otp_op_busy]位, 直到回读值为 0, 表示 OTP 控制器空闲, 可以发起操作;
- 步骤 3. 往 **OTP_USER_KEY_INDEX** 写需要烧写的 otp_index;
- 步骤 4. 往 OTP_USER_KEY_DATA[n](n=0~8)写数据, 数据填充方式如下:

数据寄存器	64bit DATA (jtag_pw_id)	128bit DATA (hmackey)	256bit DATA (key0~key3、jtag_pw、rootkey)
OTP_USER_KEY_DATA0	DATA[31:0]	DATA[31:0]	DATA[31:0]
OTP_USER_KEY_DATA1	DATA[63:32]	DATA[63:32]	DATA[63:32]
OTP_USER_KEY_DATA2	{16'h0,CRC16}	DATA[95:64]	DATA[95:64]
OTP_USER_KEY_DATA3	32'h0	DATA[127:96]	DATA[127:96]
OTP_USER_KEY_DATA4	32'h0	{16'h0,CRC16}	DATA[159:128]
OTP_USER_KEY_DATA5	32'h0	32'h0	DATA[191:160]
OTP_USER_KEY_DATA6	32'h0	32'h0	DATA[223:192]
OTP_USER_KEY_DATA7	32'h0	32'h0	DATA[255:224]
OTP_USER_KEY_DATA8	32'h0	32'h0	{16'h0,CRC16}



说明

- jtag_pw_id 的数据位宽为 64bit;
- hmackey 的数据位宽为 128bit;
- key0~key3、jtag_pw、rootkey 的数据位宽为 256bit。jtag_pw 做密钥比对时实际比对的是 data[127:0], 做 crc 校验时是 data[255:0]的所有数据都做 CRC 计算。

- 步骤 5. 往 **OTP_USER_WORK_MODE** 写 0x2(烧写 key 和 jtag ID/PASSWORD 模式);



- 步骤 6. 往 [OTP_USER_OP_START](#) 写 0x1ACC_E551, 启动命令;
- 步骤 7. 读取 [OTP_USER_CTRL_STA](#)[otp_user_cmd_finish]位, 直到回读值为 1, 表示操作完成。

----结束

13.4.1.4 KEY、JTAG ID、PASSWORD 的 CRC 校验

如果烧写 key 和 jtag ID/PASSWORD 时附加了 CRC 校验位, 那么我们可以使用 CRC 校验模式来确定烧写的值是否正确。

- 步骤 1. 读取 [OTP_USER_CTRL_STA](#)[otp_op_busy]位, 直到回读值为 0, 表示 OTP 控制器空闲, 可以发起操作;
- 步骤 2. 往 [OTP_USER_KEY_INDEX](#) 写需要做 CRC 校验的 otp_index;
- 步骤 3. 往 [OTP_USER_WORK_MODE](#) 写 0x3(key 和 jtag ID/PASSWORD 的 CRC 校验模式);
- 步骤 4. 往 [OTP_USER_OP_START](#) 写 0x1ACC_E551, 启动命令;
- 步骤 5. 读取 [OTP_USER_CTRL_STA](#)[otp_user_cmd_finish]位, 直到回读值为 1, 表示操作完成;
- 步骤 6. 读取 [OTP_USER_CTRL_STA](#)[key_crc_check_ok_flag]的值, 为 1 表示 CRC 校验通过; 为 0 表示 CRC 校验出错。

----结束

13.4.1.5 使能标志位烧写模式

操作流程如下:

- 步骤 1. 按照 [13.4.1.1](#) 的方法读取要烧写的 key 的 lock 状态; 如果 lock 状态位为 1, 则对应的 key 不能烧写; 如果 lock 状态位为 0, 继续步骤 2;
- 步骤 2. 读取 [OTP_USER_CTRL_STA](#)[otp_op_busy]位, 直到回读值为 0, 表示 OTP 控制器空闲, 可以发起操作。
- 步骤 3. 往 [OTP_USER_FLAG_INDEX](#) 写需要烧写的 flag_index;
- 步骤 4. 往 [OTP_USER_FLAG_VALUE](#) 写对应使能位的数据
- 步骤 5. 往 [OTP_USER_WORK_MODE](#) 写 0x4(使能标志位烧写模式);
- 步骤 6. 往 [OTP_USER_OP_START](#) 写 0x1ACC_E551, 启动命令;
- 步骤 7. 读取 [OTP_USER_CTRL_STA](#)[otp_user_cmd_finish]位, 直到回读值为 1, 表示操作完成;

----结束



注意

Jtag_prt_mode 区域必须配置锁定状态位为 1 才会锁定，其他只要写过一次即锁定。

Jtag_prt_mode 在没有锁定的情况下可以向更高密级配置：

JTAG OPEN→JTAG PASSWORD→JTAG CLOSE

比如第一次配置 `OTP_USER_FLAG_VALUE[jtag_prt_mode]` 的值为 3'b000，JTAG 处于 OPEN 模式且没有锁定，则可以向更高密级配置如配置成 3'b001(密码模式)。如果配置 `OTP_USER_FLAG_VALUE[jtag_prt_mode]` 的值为 3'b100，则意味着 JTAG 配置为了 OPEN 模式且不能更改。

13.4.1.6 用户预留空间烧写模式

芯片给用户预留了 8Kbit 的 OTP 空间供用户自定义使用，默认值为 0，用户预留空间每个地址烧写后自动锁定。

- 步骤 1. 读取 `OTP_USER_CTRL_STA[otp_op_busy]` 位，直到回读值为 0，表示 OTP 控制器空闲，可以发起操作。
- 步骤 2. 往 `OTP_USER_REV_ADDR` 写需要烧写的用户预留空间地址；
- 步骤 3. 往 `OTP_USER_REV_WDATA` 写数据；
- 步骤 4. 往 `OTP_USER_WORK_MODE` 写 0x5(用户预留空间烧写模式)；
- 步骤 5. 往 `OTP_USER_OP_START` 写 0x1ACC_E551，启动命令；
- 步骤 6. 读取 `OTP_USER_CTRL_STA[otp_user_cmd_finish]` 位，直到回读值为 1，表示操作完成；同时判断 `OTP_USER_CTRL_STA[otp_user_lock_err]`，为 0 表示用户配置 OTP 正常，为 1 表示用户配置了已经锁定的空间，配置失败。

----结束

13.4.1.7 用户预留空间回读模式

- 步骤 1. 读取 `OTP_USER_CTRL_STA[otp_op_busy]` 位，直到回读值为 0，表示 OTP 控制器空闲，可以发起操作；
- 步骤 2. 往 `OTP_USER_REV_ADDR` 写需要回读的用户预留空间地址；
- 步骤 3. 往 `OTP_USER_WORK_MODE` 写 0x6(用户预留空间回读模式)；
- 步骤 4. 往 `OTP_USER_OP_START` 写 0x1ACC_E551，启动命令；
- 步骤 5. 读取 `OTP_USER_CTRL_STA[otp_user_cmd_finish]` 位，直到回读值为 1，表示操作完成；
- 步骤 6. 读取 `OTP_USER_REV_RDATA` 的值。

----结束



13.4.2 OTP Ctrl 寄存器概览

OTP CTRL 寄存器概览如表 13-5 所示。

表13-5 OTP CTRL 寄存器概览（基址是 0x100B_0000）

偏移地址	名称	描述	页码
0x0000	OTP_USER_WORK_MODE	OTP 用户接口寄存器 0	13-59
0x0004	OTP_USER_OP_START	OTP 用户接口寄存器 1	13-60
0x0008	OTP_USER_KEY_INDEX	OTP 用户接口寄存器 2	13-60
0x000C	OTP_USER_KEY_DATA0	OTP 用户接口寄存器 3	13-61
0x0010	OTP_USER_KEY_DATA1	OTP 用户接口寄存器 4	13-61
0x0014	OTP_USER_KEY_DATA2	OTP 用户接口寄存器 5	13-61
0x0018	OTP_USER_KEY_DATA3	OTP 用户接口寄存器 6	13-62
0x001C	OTP_USER_KEY_DATA4	OTP 用户接口寄存器 7	13-62
0x0020	OTP_USER_KEY_DATA5	OTP 用户接口寄存器 8	13-62
0x0024	OTP_USER_KEY_DATA6	OTP 用户接口寄存器 9	13-62
0x0028	OTP_USER_KEY_DATA7	OTP 用户接口寄存器 10	13-62
0x002C	OTP_USER_KEY_DATA8	OTP 用户接口寄存器 11	13-63
0x0030	OTP_USER_FLAG_VALUE	OTP 用户接口寄存器 12	13-63
0x0034	OTP_USER_FLAG_INDEX	OTP 用户接口寄存器 13	13-64
0x0038	OTP_USER_REV_ADDR	OTP 用户接口寄存器 14	13-64
0x003C	OTP_USER_REV_WDATA	OTP 用户接口寄存器 15	13-65
0x0040	OTP_USER_REV_RDATA	OTP 用户接口寄存器 16	13-65
0x0044	OTP_USER_LOCK_STA0	OTP 用户接口寄存器 17	13-65
0x004C	OTP_USER_CTRL_STA	OTP 用户接口寄存器 19	13-67

13.4.3 OTP CTRL 寄存器描述

OTP_USER_WORK_MODE

OTP_USER_WORK_MODE 为 OTP 用户接口寄存器 0。

Offset Address: 0x0000 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000
[3:0]	RW	otp_user_work_mode	0x0: 回读 lock 状态模式 0x1: 加载 cipher key 模式 0x2: 烧写 key 和写 jtag id/password 模式 0x3: key 和 jtag id/password 的 CRC 校验模式 0x4: 使能标志配置模式 0x5: 用户预留空间烧写模式 0x6: 用户预留空间回读模式 其他: 保留	0x0

OTP_USER_OP_START

OTP_USER_OP_START 为 OTP 用户接口寄存器 1。

Offset Address: 0x0004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	otp_user_op_start	写 32'h1acce551 启动	0x00000000

OTP_USER_KEY_INDEX

OTP_USER_KEY_INDEX 为 OTP 用户接口寄存器 2。

Offset Address: 0x0008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000



[3:0]	RW	otp_index	0x0: key0 0x1: key1 0x2: key2 0x3: key3 0x4: jtag_pw_id 0x5: jtag_pw 0x6: rootkey 0x7: hmackey 其他: 保留 说明: jtag_pw_id 的数据位宽为 64bit; hmackey 的数据位宽为 128bit; key0~key3、jtag_pw、rootkey 的数据位宽为 256bit。	0x0

OTP_USER_KEY_DATA0

OTP_USER_KEY_DATA0 为 OTP 用户接口寄存器 3。

Offset Address: 0x000C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	otp_user_key_data0	OTP 配置数据 0。	0x00000000

OTP_USER_KEY_DATA1

OTP_USER_KEY_DATA1 为 OTP 用户接口寄存器 4。

Offset Address: 0x0010 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	otp_user_key_data1	OTP 配置数据 1。	0x00000000

OTP_USER_KEY_DATA2

OTP_USER_KEY_DATA2 为 OTP 用户接口寄存器 5。

Offset Address: 0x0014 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	otp_user_key_data2	OTP 配置数据 2。	0x00000000



OTP_USER_KEY_DATA3

OTP_USER_KEY_DATA3 为 OTP 用户接口寄存器 6。

Offset Address: 0x0018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	otp_user_key_data3	OTP 配置数据 3。	0x00000000

OTP_USER_KEY_DATA4

OTP_USER_KEY_DATA4 为 OTP 用户接口寄存器 7。

Offset Address: 0x001C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	otp_user_key_data4	OTP 配置数据 4。	0x00000000

OTP_USER_KEY_DATA5

OTP_USER_KEY_DATA5 为 OTP 用户接口寄存器 8。

Offset Address: 0x0020 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	otp_user_key_data5	OTP 配置数据 5。	0x00000000

OTP_USER_KEY_DATA6

OTP_USER_KEY_DATA6 为 OTP 用户接口寄存器 9。

Offset Address: 0x0024 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	otp_user_key_data6	OTP 配置数据 6。	0x00000000

OTP_USER_KEY_DATA7

OTP_USER_KEY_DATA7 为 OTP 用户接口寄存器 10。

Offset Address: 0x0028 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	otp_user_key_data7	OTP 配置数据 7。	0x00000000

OTP_USER_KEY_DATA8

OTP_USER_KEY_DATA8 为 OTP 用户接口寄存器 11。

Offset Address: 0x002C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	otp_user_key_data8	OTP 配置数据 8。	0x00000000

OTP_USER_FLAG_VALUE

OTP_USER_FLAG_VALUE 为 OTP 用户接口寄存器 12。

Offset Address: 0x0030 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RW	secure_iso_en	安全隔离配置写数据 0: 不使能安全隔离; 1: 使能安全隔离。	0x0
[6]	RW	jtag_prt_mode_lock_en	jtag_prt_mode 区域配置锁定使能位 0:不 lock, jtag_prt_mode 区域为 0 的 bit 位还能配置为 1 1: lock, jtag_prt_mode 将不能再修改	0x0
[5:4]	RW	jtag_prt_mode	jtag_prt_mode[1:0]配置数据寄存器 00:jtag open 模式 01: jtag password 模式 1x: jtag 永久关闭模式	0x0
[3]	-	reserved	保留。	0x0
[2]	RW	jtag_ca_en	jtag_ca_en 配置数据寄存器。 0: 不使能 JTAG 安全模式; 1: 使能 JTAG 安全模式。	0x0



[1]	RW	ddr_ca_en	ddr_ca_en 配置数据寄存器。 0: 不使能 DDR 安全模式; 1: 使能 DDR 安全模式。	0x0
[0]	RW	secure_boot_en	secure_boot_en 配置数据寄存器。 0: 不使能安全启动模式; 1: 使能安全启动模式。	0x0

OTP_USER_FLAG_INDEX

OTP_USER_FLAG_INDEX 为 OTP 用户接口寄存器 13。

Offset Address: 0x0034 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	flag_index	使能标志位索引。 000: secure_boot_en 的索引; 001: ddr_ca_en 的索引; 010: jtag_ca_en 的索引; 011: jtag_prt_mode 的索引; 100: secure_iso_en 的索引; 其他: 保留。	0x0

OTP_USER_REV_ADDR

OTP_USER_REV_ADDR 为 OTP 用户接口寄存器 14。

Offset Address: 0x0038 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x00000000
[7:0]	RW	otp_user_reserved_addr	OTP 用户预留空间地址。一个地址对应 32bit OTP 空间的数据。	0x00

OTP_USER_REV_WDATA

OTP_USER_REV_WDATA 为 OTP 用户接口寄存器 15。

Offset Address: 0x003C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	otp_user_reserved_wdata	OTP 用户预留空间写数据。	0x00000000

OTP_USER_REV_RDATA

OTP_USER_REV_RDATA 为 OTP 用户接口寄存器 16。

Offset Address: 0x0040 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	otp_user_reserved_rdata	OTP 用户预留空间读数据。	0x00000000

OTP_USER_LOCK_STA0

OTP_USER_LOCK_STA0 为 OTP 用户接口寄存器 17。

Offset Address: 0x0044 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15]	RO	hmac_key_lock	hmac key 锁定状态指示位。 0: 可配置; 1: 已经配置过并锁定。	0x0
[14:12]	-	reserved	保留。	0x0
[11]	RO	secure_iso_en_lock	secure_iso_en 锁定状态指示位。 0: 可配置; 1: 已经配置过并锁定。	0x0
[10]	RO	jtag_prt_mode_lock	jtag_prt_mode 锁定状态指示位。 0: 可配置; 1: 已配置并锁定。	0x0
[9]	RO	jtag_ca_en_lock	jtag_ca_en 锁定状态指示位。 0: 可配置; 1: 已配置并锁定。	0x0
[8]	RO	key3_lock	Key3 锁定状态指示位。 0: 可配置; 1: 已配置并锁定。	0x0
[7]	RO	key2_lock	Key2 锁定状态指示位。	0x0



Bits	Access	Name	Description	Reset
			0: 可配置; 1: 已配置并锁定。	
[6]	RO	key1_lock	Key1 锁定状态指示位。 0: 可配置; 1: 已配置并锁定。	0x0
[5]	RO	key0_lock	Key0 锁定状态指示位。 0: 可配置; 1: 已配置并锁定。	0x0
[4]	RO	root_key_lock	root_key 锁定状态指示位。 0: 可配置; 1: 已配置并锁定。	0x0
[3]	RO	jtag_pw_lock	jtag_pw 锁定状态指示位。 0: 可配置; 1: 已配置并锁定。	0x0
[2]	RO	jtag_pw_id_lock	jtag_pw_id 锁定状态指示位。 0: 可配置; 1: 已配置并锁定。	0x0
[1]	RO	ddr_ca_en_lock	ddr_ca_en 锁定状态指示位。 0: 可配置; 1: 已配置并锁定。	0x0
[0]	RO	secure_boot_en_lock	secure_boot_en 锁定状态指示位。 0: 可配置; 1: 已配置并锁定。	0x0

OTP_USER_CTRL_STA

OTP_USER_CTRL_STA 为 OTP 用户接口寄存器 19。

Offset Address: 0x004C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:24]	RO	flag_index_finish	对应完成命令的标志位地址。	0x0
[23:16]	RO	user_rev_addr_finish	对应完成命令的用户保留空间地址。	0x00



Bits	Access	Name	Description	Reset
[15:12]	RO	key_index_finish	对应完成命令的 key 地址。	0x0
[11]	RO	rd_lock_sta_finish	读 lock 状态模式是否完成。 0: 未完成; 1: 完成。	0x0
[10]	RO	key_load_finish	key 加载模式是否完成。 0: 未完成; 1: 完成。	0x0
[9]	RO	key_pgm_finish	key 写模式是否完成。 0: 未完成; 1: 完成。	0x0
[8]	RO	key_crc_check_finish	CRC 校验模式是否完成 0: 未完成; 1: 完成。	0x0
[7]	RO	flag_pgm_finish	标志位写模式是否完成 0: 未完成; 1: 完成。	0x0
[6]	RO	user_rev_pgm_finish	用户保留空间写模式是否完成 0: 未完成; 1: 完成。	0x0
[5]	RO	user_rev_rd_finish	用户保留空间读模式是否完成 0: 未完成; 1: 完成。	0x0
[4]	RO	key_crc_check_ok_flag	用户 key 的 CRC 校验是否通过。 0: 不通过; 1: 通过。	0x0
[3]	RO	key_rd_back_check_err	用户配置的 key 回读状态指示。 0: 回读正常; 1: 回读错误。	0x0
[2]	RO	otp_user_lock_err	用户配置 OTP 错误状态指示位。 0: 正常完成; 1: 用户配置了已经锁定的 OTP 空间。	0x0
[1]	RO	otp_user_cmd_finish	OTP 控制器操作是否完成。 0: 未完成;	0x0



Bits	Access	Name	Description	Reset
			1: 完成，可查询命令运行结果。	
[0]	RO	otp_op_busy	OTP 控制器空忙指示位。 0: 空闲，可以接受新的操作； 1: OTP 控制器忙，此时不可操作 OTP 控制器。	0x0



目 录

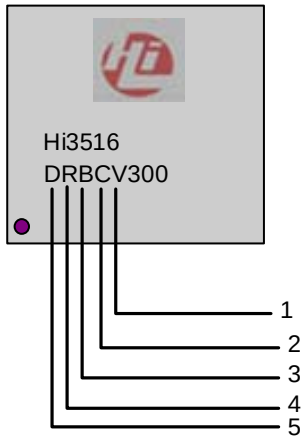
A 订购须知	A-1



A 订购须知

Hi3516DV300 芯片标识如图 A-1 所示。

图A-1 Hi3516DV300 芯片标识



Hi3516DV300 芯片标识定义如表 A-1 所示。

表A-1 Hi3516DV300 芯片标识定义

编号	标识	含义
1	版本号	芯片版本号。
2	温度标识	C: Commercial(商业用品)
3	封装标识	B: BGA
4	环保标识	R: RoHS
5	产品区分	芯片产品名称最后一位，即产品区分位。