



Hi3519DV500 超高清智慧视觉 SoC

用户指南

文档版本 00B02

发布日期 2023-03-10

版权所有 © 上海海思技术有限公司 2023。保留一切权利。

未经本公司书面许可，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海思技术有限公司的商标。

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。

上海海思技术有限公司

地址：上海市青浦区虹桥港路 2 号 101 室 邮编：201721

网址：<http://www.hisilicon.com/cn/>

客户服务邮箱：support@hisilicon.com



目 录

前 言.....1



前言

概述

本文档介绍了 Hi3519DV500 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 Hi3519DV500。

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
Hi3519D	V500

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

表格内容约定

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

寄存器访问类型约定

类型	说明	类型	说明
RO	只读, 不可写。	W0C	可读, 写 0 清零, 写 1 保持不变。
WO	只写。	W1S	可读, 写 1 置 1, 写 0 保持不变。
RW	可读可写。	W0S	可读, 写 0 置 1, 写 1 保持不变。
RC	读清零。	OSW	可读, 写 1 后片内自清零, 即产生一个脉冲。
W1C	可读, 写 1 清零, 写 0 保持不变。		
RW_CLR	可读可写, 逻辑清除	RW_LOCK	可读可写, 写 1 后锁定
RW_SEC	安全 CPU 可读可写	RW_SEC_LOCK	安全 CPU 可读可写, 写 1 后锁定

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



类别	符号	对应的数值
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
2023-03-10	00B02	第 2 次临时版本发布。 第 1 章 产品概述 1.4.2 小节刷新。 第 3 章 系统 3.10.6.1 小节步骤 5 更新。 删除 3.10.9 小节。 第 5 章 FMC 控制器 5.4.5 小节，图 5-10 涉及修改。
2022-12-10	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
1.3 架构	1-2
1.3.1 逻辑框图	1-2
1.3.2 处理器内核	1-3
1.3.3 系统级加速模块	1-3
1.3.4 智能视频分析	1-3
1.3.5 视频编解码	1-4
1.3.6 数字图像处理 (ISP)	1-4
1.3.7 视频与图形处理	1-5
1.3.8 视频输入接口	1-5
1.3.9 视频输出接口	1-6
1.3.10 音频接口与处理	1-6
1.3.11 安全隔离与引擎	1-6
1.3.12 网络接口	1-7
1.3.13 外围接口	1-7
1.3.14 存储器接口	1-7
1.3.15 SDK	1-8
1.3.16 芯片物理规格	1-8
1.4 启动和升级模式	1-8
1.4.1 概述	1-8
1.4.2 启动介质和上电锁存值关系	1-9
1.4.3 启动模式	1-10



1.5 地址空间映射 1-11



插图目录

图 1-1 Hi3519DV500 超高清智慧视觉 SoC 方案	1-2
图 1-2 Hi3519DV500 芯片逻辑框图	1-3



表格目录

表 1-1 启动介质.....	1-10
表 1-2 启动模式.....	1-10
表 1-3 地址空间映射表.....	1-11



1 产品概述

1.1 概述

Hi3519DV500 是一颗面向视觉行业推出的超高清智能 SoC。该芯片最高支持四路 sensor 输入，支持最高 4K@30fps 的 ISP 图像处理能力，支持 2F WDR、多级降噪、六轴防抖、全景拼接、多光谱融合等多种传统图像增强和处理算法，支持通过 AI 算法对输入图像进行实时降噪等处理，为用户提供了卓越的图像处理能力。支持热红外、结构光和 ToF sensor 的接入和处理。

Hi3519DV500 内置双核 A55，提供高效、丰富和灵活的 CPU 资源，以满足客户计算和控制需求。

Hi3519DV500 集成了高效的神经网络推理引擎，最高 2.5Tops NN 算力，支持业界主流的神经网络框架。

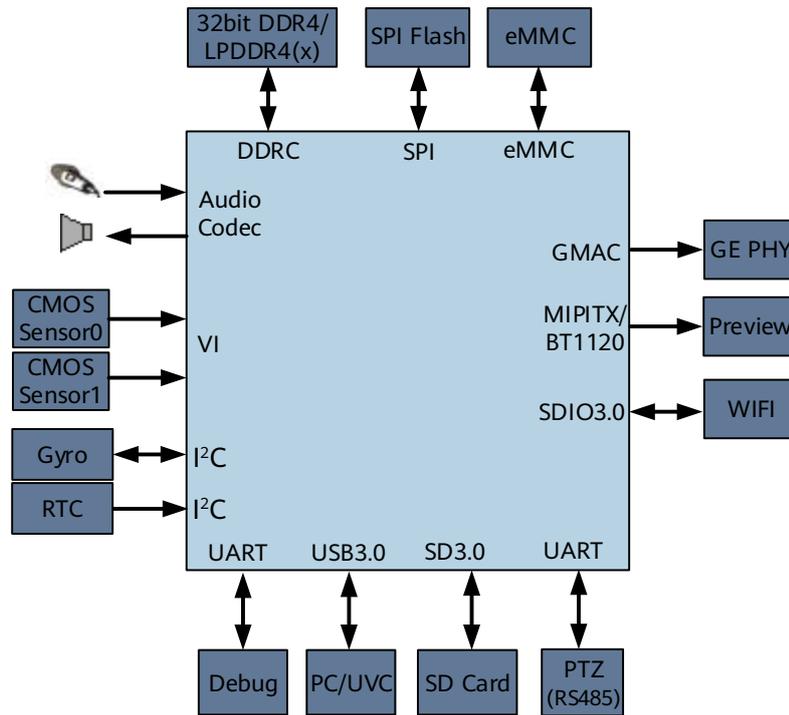
Hi3519DV500 提供稳定、易用的 SDK 软件开发包，支撑客户产品快速量产。

1.2 应用场景

Hi3519DV500 超高清智慧视觉 SoC 方案如图 1-1 所示。



图1-1 Hi3519DV500 超高清智慧视觉 SoC 方案

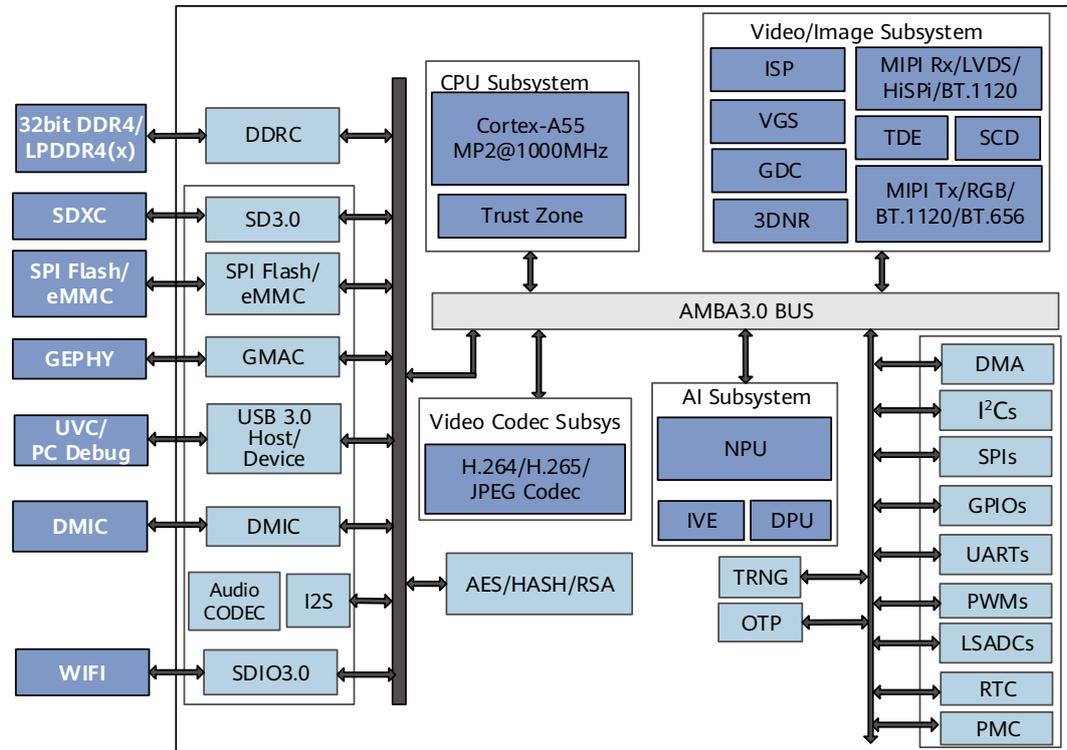


1.3 架构

1.3.1 逻辑框图

Hi3519DV500 芯片逻辑框图如图 1-2 所示。

图1-2 Hi3519DV500 芯片逻辑框图



1.3.2 处理器内核

- 双核 ARM Cortex A55@1000MHz
- 32KB I-Cache, 32KB D-Cache, 256KB L3 cache
- 支持 NEON 加速, 集成 FPU 处理单元
- 支持 TrustZone

1.3.3 系统级加速模块

- 集成硬化的标准 CRC32/CRC16/CRC8 多项式运算单元
- 集成硬化的高速直接数据搬移模块(DMA)

1.3.4 智能视频分析

- 神经网络
支持完整的 API 和工具链, 易于客户开发
- 升级 IVE 算子, 支持特征点检测、周界、光流及多种计算机形态学算子



- 升级 DPU 算法实现双目深度图加速单元，最大分辨率 2048 x 2048，最大视差 224，处理性能 720p@30fps

1.3.5 视频编解码

- 支持 H.264 BP/MP/HP Level 5.1
- 支持 H.265 Main Profile Level 5.1
- H.264/H.265 编解码最大分辨率为 6144x 6144
- 支持 I/P 帧
- H.264/H.265 多码流编解码典型性能如下：
 - 3840 x 2160@30fps(编码)+1920 x 1080@30fps(编码)+720 x 480@30fps(编码)
 - 3840 x 2160@30fps(编码)+720 x 480@30fps(编码)+1920 x 1080@30fps(解码)
 - 3840 x 2160@30fps(解码)
- 支持 8 个区域的编码前 OSD 叠加
- 支持 CBR/VBR/AVBR/FIXQP/QPMAP 等多种码率控制模式
- 输出码率最大值 80Mbps
- 支持 8 个感兴趣区域 (ROI) 编码
- 支持视频前端叠加 mosaic 编码
- 支持数字水印
- 支持 PVC 感知编码降低码流
- 支持 JPEG Baseline 编解码
- JPEG 编解码最大分辨率 16384 x 16384
- JPEG 最大性能
 - 编码：3840 x 2160@60fps(YUV420)
 - 解码：3840 x 2160@30fps(YUV420)

1.3.6 数字图像处理 (ISP)

- 支持多路 sensor 同时处理
- 支持 3A (AE/AWB/AF) 功能，3A 参数用户可调节
- 支持去固定模式噪声 (FPN)



- 支持坏点校正和镜头阴影校正
- 支持两帧 WDR 及 Advanced Local Tone Mapping, 支持强光抑制和背光补偿
- 支持多级 3D 去噪
- 支持图像边缘增强
- 支持去雾
- 支持动态对比度增强
- 支持 3D-LUT 色彩调节
- 支持新一代镜头畸变校正
- 支持鱼眼等任意形状几何矫正
- 支持 6-DoF 数字防抖
- 支持陀螺仪防抖和 Rolling-Shutter 校正
- 支持图像 Mirror、Flip、90 度/270 度旋转
- 支持使用神经网络对图像进行实时 DRC、BNR、3DNR 或 DM 处理
- 支持黑白与彩色两路图像双光融合
- 提供 PC 端 ISP 调节工具

1.3.7 视频与图形处理

- 支持图形和图像 1/15.5 ~ 16x 缩放功能
- 支持水平方向全景拼接
 - 输入 2 路 2560×1440@30fps, 输出 3840x2160@30fps
 - 输入 4 路 1920x1080@30fps, 输出 3840x2160@30fps
- 支持视频层、图形层叠加
- 支持色彩空间转换

1.3.8 视频输入接口

- 支持 8-Lane image sensor 串行输入, 支持 MIPI/LVDS/Sub-LVDS/HiSPi 多种接口
- 支持 2x4-Lane 或 4x2-Lane 等多种组合, 最高支持 4 路 sensor 输入
- 支持 8/10/12/14 Bit RGB Bayer DC 时序视频输入, 时钟频率最高 148.5MHz
- 支持 BT.601、BT.656、BT.1120 视频输入接口
- 支持通过 MIPI 虚拟通道输入 1~4 路 YUV



- 支持主流 CMOS 电平热成像传感器接入
- 支持结构光图像模组
- 支持 cw ToF 图像传感器

1.3.9 视频输出接口

- 支持一路 BT.1120 或 BT.656 接口输出，其中 BT.1120 最高性能 1920x1080@60fps
- 支持 6/8bit 串行或 16/18/24bit RGB 并行输出，最高频率 74.25MHz
- 支持 4-Lane Mipi DSI/CSI 接口输出，最高 1.8Gbps/lane，性能 3840 x 2160@30fps
- 支持 Gamma 校正和水平方向 sharpen

1.3.10 音频接口与处理

- 内置 Audio codec，支持 16bit 双路差分语音输入和双路单端语音输出
- 支持 1 路 I2S 接口，兼容多声道时分复用传输模式（TDM）
- 支持 8 路数字 MIC 阵列输入
- 支持多协议语音编解码
- 支持音频 3A（AEC/ANR/ALC）处理

1.3.11 安全隔离与引擎

- 支持安全启动
- 支持基于 TrustZone 的 REE/TEE 硬件隔离方案
- 支持神经网络模型与数据保护
- 硬件实现 AES128/256 对称加密算法
- 硬件实现 RSA3072/4096 签名校验算法
- 硬件实现 ECC256/384/512 椭圆曲线算法
- 硬件实现 SHA256/384/512、HMAC_SHA256/384/512 算法
- 硬件实现 SM2/3/4 国密算法
- 硬件实现真随机数发生器
- 提供 28Kbit OTP 存储空间供客户使用



1.3.12 网络接口

- 1 个千兆以太网接口
 - 支持 RGMII、RMII 两种接口模式
 - 支持 TSO、UFO、COE 等加速单元

1.3.13 外围接口

- 2 个 SDIO3.0 接口
 - SDIO0 支持 SDXC 卡, 最大容量 2TB
 - SDIO1 支持对接 wifi 模组
- 1 个 USB3.0/USB2.0 接口
 - USB Host/Device 可切换
- 支持上电复位 (POR) 和外部输入复位
- 集成独立供电 RTC
- 集成精简上下电控制逻辑, 实现芯片待机唤醒
- 集成 4 通道 LSADC
- 集成 RGB 小屏专用三线控制接口
- 多个 UART、I²C、SPI、PWM、GPIO 接口

1.3.14 存储器接口

- DDR4/LPDDR4/LPDDR4x 接口
 - 支持 2 x 16bit DDR4
 - 支持 1 x 32bit LPDDR4/LPDDR4x
 - DDR4 最高速率 2666Mbps
 - LPDDR4/LPDDR4x 最高速率 2666Mbps
 - 最大容量 4GB
- SPI Nor/SPI Nand Flash 接口
 - 支持 1、2、4 线模式
 - SPI Nor Flash 支持 3Byte、4Byte 地址模式
- eMMC5.1 接口, 最大容量 2TB
- 可选择从 eMMC、SPI Nor/SPI Nand Flash 启动



1.3.15 SDK

支持 Linux5.10 SDK 包

1.3.16 芯片物理规格

- 功耗
2.5W 典型功耗(编码 4K30 + 2Tops)
- 工作电压
 - 内核电压为 0.9V
 - IO 电压为 1.8/3.3V
 - DDR4/LPDDR4/LPDDR4x 接口电压分别为 1.2/1.1/0.6V
- 封装形式
 - RoHS, FCCSP 15mm x 15mm 封装
 - 管脚间距: 0.65mm

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

芯片包含多种外设接口可用于启动介质接口:

- SPI Nand/Nor Flash 存储接口。
- eMMC 存储接口。

启动介质的选择由 SFC_EMMC_BOOT_MODE/BOOT_SEL1/BOOT_SEL0 信号决定。

烧写 (升级)

芯片可以通过 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 FAST_BOOT_MODE 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 启动介质和上电锁存值关系

启动/升级介质由

FAST_BOOT_MODE/SFC_EMMC_BOOT_MODE/BOOT_SEL1/BOOT_SEL0 和 UPDATE_MODE_N 信号来确定。

说明

- FAST_BOOT_MODE 信号为 SENSOR1_CLK 管脚的上电锁存值；
- SFC_EMMC_BOOT_MODE 信号为 RGMII_TXD1 管脚的上电锁存值；
- BOOT_SEL1 信号为 RGMII_TXD0 管脚的上电锁存值；
- BOOT_SEL0 信号为 SENSOR1_VS 管脚的上电锁存值；
- UPDATE_MODE_N 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。
- BOOT_PARA_SEL_0、BOOT_PARA_SEL_1、BOOT_PARA_SEL_2、BOOT_PARA_SEL_3 信号分别为 SENSOR1_HS、EPHY_CLK、MDCK、RGMII_TXEN 管脚的上电锁存值。
- 以上上电锁存值不受系统软复位影响。
- SFC_EMMC_BOOT_MODE & BOOT_SEL1 & BOOT_SEL0 决定了启动或者烧写的目标介质。
- FAST_BOOT_MODE 用于选择是否进入串口烧写。

通过系统控制寄存器 SYSSTAT[4:2]和 SYSSTAT[11]可以获取当前的 FAST_BOOT_MODE、SFC_EMMC_BOOT_MODE、BOOT_SEL1、BOOT_SEL0 的状态。通过系统控制寄存器 SYSSTAT[29:27]可以获取当前的 BOOT_PARA_SEL_0、BOOT_PARA_SEL_1、BOOT_PARA_SEL_2 的状态，从而选择某个 DDR 参数表格启动。通过系统控制寄存器 SYSSTAT[30]可以获取当前的 BOOT_PARA_SEL_3 的状态，从而确认 SFC 器件是 3.3V 还是 1.8V。

说明

请参考“系统”章节的 SYSSTAT 寄存器中的描述。
相关信号名请参考《Hi3519DV500_PINOUT_CN》。

启动介质和这些信号的关系如表 1-1 所示。



表1-1 启动介质

UPDATE_MODE	FAST_BOOT_MODE	BOOT_SEL1	BOOT_SEL0	启动介质
1	0	0	0	从 SPI NOR Flash 启动
1	0	0	1	从 SPI NAND Flash 启动
1	0	1	1	从 EMMC 启动
1	1	0	0	串口烧写模式，烧写 SPI NOR Flash
1	1	0	1	串口烧写模式，烧写 SPI NAND Flash
1	1	1	1	串口烧写模式，烧写 EMMC
0	X	0	0	从 SDIO/USB 升级 SPI Nor Flash
0	X	0	1	从 SDIO/USB 升级 SPI Nand Flash
0	X	1	1	从 SDIO/USB 升级 eMMC Flash

1.4.3 启动模式

芯片支持如表 1-2 所示启动模式，其中 soc_tee_enable 由寄存器 SOC_TEE_ENABLE[7:0]（绝对地址：0x0_101E_0010）值来决定（源于 OTP 中的启动模式标志位）。

说明

寄存器 SOC_TEE_ENABLE 详见《xxxx 安全启动使用指南》。

表1-2 启动模式

soc_tee_enable [7:0]	启动模式
非 0x42	安全启动模式
0x42	非安全启动的普通模式



须知

推荐使用安全启动模式，以降低产品的安全风险。

1.5 地址空间映射

地址空间参见表 1-3 所示。

表1-3 地址空间映射表

起始地址	结束地址	功能	大小	说明
0x0_0000_0000	0x0_0001_3FFF	启动地址空间 BOOTROM	80KB	
0x0_0001_4000	0x0_03FF_FFFF	保留	-	
0x0_0400_0000	0x0_0401_3FFF	启动地址空间 BOOTROM	80KB	
0x0_0401_4000	0x0_0401_FFFF	保留	-	
0x0_0402_0000	0x0_0403_FFFF	启动地址空间 BOOTRAM	128KB	
0x0_0404_0000	0x0_0EFE_FFFF	保留	-	
0x0_0EFF_0000	0x0_0EFF_FFFF	IO0_CFG 寄存器	64KB	IO0 管脚配置
0x0_0F00_0000	0x0_0FFF_FFFF	FMC_MEM 空间	16MB	
0x0_1000_0000	0x0_1000_FFFF	FMC 寄存器	64KB	
0x0_1001_0000	0x0_1001_FFFF	保留	-	
0x0_1002_0000	0x0_1002_FFFF	EMMC 寄存器	64KB	
0x0_1003_0000	0x0_1003_FFFF	SDIO0 寄存器	64KB	
0x0_1004_0000	0x0_1004_FFFF	SDIO1 寄存器	64KB	
0x0_1005_0000	0x0_101D_FFFF	保留	-	
0x0_101E_0000	0x0_101E_1FFF	OTP Ctrl 寄存器	8KB	



起始地址	结束地址	功能	大小	说明
0x0_101E_2000	0x0_101E_FFFF	保留	-	
0x0_101F_0000	0x0_101F_FFFF	SPACC 寄存器	64KB	
0x0_1020_0000	0x0_1025_FFFF	保留	-	
0x0_1026_0000	0x0_1026_FFFF	IO1_CFG 寄存器	64KB	IO1 管脚配置
0x0_1027_0000	0x0_1027_1FFF	MIPI Tx 寄存器	8KB	
0x0_1027_2000	0x0_1027_3FFF	MIPI Tx PHY 寄存器	8KB	
0x0_1027_4000	0x0_1027_FFFF	保留	-	
0x0_1028_0000	0x0_1028_FFFF	DMA 寄存器	64KB	
0x0_1029_0000	0x0_1029_FFFF	GMAC 寄存器	64KB	
0x0_102A_0000	0x0_1031_FFFF	保留	-	
0x0_1032_0000	0x0_1032_FFFF	USB3.0 寄存器	64KB	
0x0_1033_0000	0x0_10FF_0FFF	保留	-	
0x0_1100_0000	0x0_1100_0FFF	TIMER01 寄存器	4KB	
0x0_1100_1000	0x0_1100_1FFF	TIMER23 寄存器	4KB	
0x0_1100_2000	0x0_1100_2FFF	TIMER45 寄存器	4KB	
0x0_1100_3000	0x0_1100_3FFF	TIMER67 寄存器	4KB	
0x0_1100_4000	0x0_1100_4FFF	SEC_TIMER01 寄存器	4KB	
0x0_1100_5000	0x0_1100_5FFF	SEC_TIMER23 寄存器	4KB	
0x0_1100_6000	0x0_1100_7FFF	保留	-	
0x0_1100_8000	0x0_1100_8FFF	CRC 寄存器	4KB	
0x0_1100_9000	0x0_1100_FFFF	保留	-	
0x0_1101_0000	0x0_1101_FFFF	CRG 寄存器	64KB	
0x0_1102_0000	0x0_1102_3FFF	SYS CTRL 寄存器	16KB	



起始地址	结束地址	功能	大小	说明
0x0_1102_4000	0x0_1102_8FFF	SOC MISC 寄存器	20KB	
0x0_1102_9000	0x0_1102_9FFF	SVB_PWM 寄存器	4KB	
0x0_1102_A000	0x0_1102_AFFF	Tsensor_CTRL 寄存器	4KB	
0x0_1102_B000	0x0_1102_BFFF	HPM_CTRL 寄存器	4KB	
0x0_1102_C000	0x0_1102_CFFF	STOPWATCH 寄存器	4KB	
0x0_1102_D000	0x0_1102_FFFF	保留	-	
0x0_1103_0000	0x0_1103_0FFF	WDG 寄存器	4KB	
0x0_1103_1000	0x0_1103_FFFF	保留	-	
0x0_1104_0000	0x0_1104_0FFF	UART0 寄存器	4KB	
0x0_1104_1000	0x0_1104_1FFF	UART1 寄存器	4KB	
0x0_1104_2000	0x0_1104_2FFF	UART2 寄存器	4KB	
0x0_1104_3000	0x0_1104_3FFF	UART3 寄存器	4KB	
0x0_1104_4000	0x0_1104_4FFF	UART4 寄存器	4KB	-
0x0_1104_5000	0x0_1104_5FFF	UART5 寄存器	4KB	-
0x0_1104_6000	0x0_1105_FFFF	保留	-	
0x0_1106_0000	0x0_1106_0FFF	I2C0 寄存器	4KB	
0x0_1106_1000	0x0_1106_1FFF	I2C1 寄存器	4KB	
0x0_1106_2000	0x0_1106_2FFF	I2C2 寄存器	4KB	
0x0_1106_3000	0x0_1106_3FFF	I2C3 寄存器	4KB	-
0x0_1106_4000	0x0_1106_4FFF	I2C4 寄存器	4KB	-
0x0_1106_5000	0x0_1106_5FFF	I2C5 寄存器	4KB	-
0x0_1106_6000	0x0_1106_6FFF	I2C6 寄存器	4KB	-
0x0_1106_7000	0x0_1106_7FFF	I2C7 寄存器	4KB	



起始地址	结束地址	功能	大小	说明
0x0_1106_8000	0x0_1106_FFFF	保留	-	
0x0_1107_0000	0x0_1107_0FFF	SPI0 寄存器	4KB	
0x0_1107_1000	0x0_1107_1FFF	SPI1 寄存器	4KB	-
0x0_1107_2000	0x0_1107_2FFF	SPI2 寄存器	4KB	
0x0_1107_3000	0x0_1107_3FFF	SPI3 寄存器	4KB	
0x0_1107_4000	0x0_1107_4FFF	SPI_TFT 寄存器	4KB	
0x0_1107_5000	0x0_1107_5FFF	SPI_3WIRE 寄存器	4KB	
0x0_1107_6000	0x0_1107_FFFF	保留	-	
0x0_1108_0000	0x0_1108_0FFF	PWM0 寄存器	4KB	
0x0_1108_1000	0x0_1108_1FFF	PWM1 寄存器	4KB	
0x0_1108_2000	0x0_1108_2FFF	PWM2 寄存器	4KB	
0x0_1108_3000	0x0_1108_FFFF	保留	-	
0x0_1109_0000	0x0_1109_0FFF	GPIO0 寄存器	4KB	
0x0_1109_1000	0x0_1109_1FFF	GPIO1 寄存器	4KB	
0x0_1109_2000	0x0_1109_2FFF	GPIO2 寄存器	4KB	
0x0_1109_3000	0x0_1109_3FFF	GPIO3 寄存器	4KB	
0x0_1109_4000	0x0_1109_4FFF	GPIO4 寄存器	4KB	
0x0_1109_5000	0x0_1109_5FFF	GPIO5 寄存器	4KB	
0x0_1109_6000	0x0_1109_6FFF	GPIO6 寄存器	4KB	
0x0_1109_7000	0x0_1109_7FFF	GPIO7 寄存器	4KB	
0x0_1109_8000	0x0_1109_8FFF	GPIO8 寄存器	4KB	
0x0_1109_9000	0x0_1109_9FFF	GPIO9 寄存器	4KB	
0x0_1109_A000	0x0_1109_AFFF	GPIO10 寄存器	4KB	
0x0_1109_B000	0x0_1109_BFFF	GPIO11 寄存器	4KB	



起始地址	结束地址	功能	大小	说明
0x0_1109_C000	0x0_1109_CFFF	GPIO12 寄存器	4KB	
0x0_1109_D000	0x0_1109_DFFF	GPIO13 寄存器	4KB	
0x0_1109_E000	0x0_1109_EFFF	GPIO14 寄存器	4KB	
0x0_1109_F000	0x0_110F_FFFF	保留	-	
0x0_1110_0000	0x0_1110_FFFF	LSADC 寄存器	64KB	
0x0_1111_0000	0x0_1111_FFFF	RTC 寄存器	64KB	
0x0_1112_0000	0x0_1112_FFFF	PMC 寄存器	64KB	
0x0_1113_0000	0x0_1113_FFFF	保留	-	
0x0_1114_0000	0x0_1115_FFFF	MDDRC 寄存器	128KB	
0x0_1116_0000	0x0_1116_FFFF	DDRT 寄存器	64KB	
0x0_1117_0000	0x0_11FF_FFFF	保留	-	
0x0_1200_0000	0x0_123F_FFFF	CPU 寄存器	4MB	
0x0_1240_0000	0x0_125F_FFFF	GIC600 寄存器	2MB	
0x0_1260_0000	0x0_13FF_EFFF	保留	-	
0x0_13FF_F000	0x0_13FF_FFFF	NPU_SEC_PRESENT 寄存器	4KB	
0x0_1400_0000	0x0_147F_FFFF	NPU 寄存器	8MB	
0x0_1480_0000	0x0_16FF_FFFF	保留	-	
0x0_1700_0000	0x0_1700_FFFF	IVE 寄存器	64KB	
0x0_1701_0000	0x0_1702_FFFF	保留	-	
0x0_1703_0000	0x0_1703_FFFF	DPU 寄存器	64KB	
0x0_1704_0000	0x0_170D_FFFF	保留	-	
0x0_170E_0000	0x0_170E_FFFF	保留	-	
0x0_170F_0000	0x0_170F_FFFF	GZIP 寄存器	64KB	
0x0_1710_0000	0x0_1710_FFFF	SCD 寄存器	64KB	



起始地址	结束地址	功能	大小	说明
0x0_1711_0000	0x0_1713_FFFF	保留	-	
0x0_1714_0000	0x0_1714_FFFF	VEDU 寄存器	64KB	
0x0_1715_0000	0x0_1717_FFFF	保留	-	
0x0_1718_0000	0x0_1718_FFFF	JPGD 寄存器	64KB	
0x0_1719_0000	0x0_171B_FFFF	保留	-	
0x0_171C_0000	0x0_171C_FFFF	JPGE 寄存器	64KB	
0x0_171D_0000	0x0_1723_FFFF	保留	-	
0x0_1724_0000	0x0_1725_FFFF	VGS 寄存器	128KB	
0x0_1726_0000	0x0_1734_FFFF	保留	-	
0x0_1735_0000	0x0_1735_FFFF	USB2_PHY 寄存器	64KB	
0x0_1736_0000	0x0_1736_0FFF	USB3_CTRL_APB 寄存器	4KB	
0x0_1736_1000	0x0_1736_1FFF	UPS_PHY 寄存器	4KB	
0x0_1736_2000	0x0_173B_FFFF	保留	-	
0x0_173C_0000	0x0_173C_FFFF	MIPI RX 寄存器	64KB	
0x0_173D_0000	0x0_173D_0FFF	热成像控制器寄存器	4KB	
0x0_173D_1000	0x0_173F_FFFF	保留	-	
0x0_1740_0000	0x0_175F_FFFF	VICAP 寄存器	2MB	
0x0_1760_0000	0x0_177F_FFFF	保留	-	
0x0_1780_0000	0x0_1783_FFFF	VIPROC 寄存器	256KB	
0x0_1784_0000	0x0_178F_FFFF	保留	-	
0x0_1790_0000	0x0_1790_FFFF	VPSS 寄存器	64KB	
0x0_1791_0000	0x0_179C_FFFF	保留	-	
0x0_179D_0000	0x0_179D_FFFF	保留	-	
0x0_179E_0000	0x0_179E_FFFF	保留	-	



起始地址	结束地址	功能	大小	说明
0x0_179F_0000	0x0_179F_FFFF	IO2_CFG 寄存器	64KB	IO2 管脚配置
0x0_17A0_0000	0x0_17A3_FFFF	VDP 寄存器	256KB	
0x0_17A4_0000	0x0_17A4_0FFF	VI_MIPI 及 SPI0/1 寄存器	4KB	
0x0_17A4_1000	0x0_17BF_FFFF	保留	-	
0x0_17C0_0000	0x0_17C0_FFFF	AIAO 寄存器	64KB	
0x0_17C1_0000	0x0_17C1_3FFF	DMIC 寄存器	16KB	
0x0_17C1_4000	0x0_17C3_FFFF	保留	-	
0x0_17C4_0000	0x0_17C4_FFFF	Audio Codec 寄存器	64KB	
0x0_17C5_0000	0x0_3FFF_FFFF	保留	-	
0x0_4000_0000	0x3_FFFF_FFFF	DDR 地址空间	15360MB	



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	2-5
2.2 管脚信息描述	2-6
2.3 焊接工艺建议	2-6
2.3.1 无铅回流焊工艺参数要求	2-6
2.3.2 混合回流焊工艺参数要求	2-8
2.4 潮敏参数	2-9
2.4.1 上海海思产品防潮包装	2-10
2.4.2 存放与使用	2-11
2.4.3 重新烘烤	2-12
2.5 电性能参数	2-13
2.5.1 功耗参数	2-13
2.5.2 温度和热阻参数	2-14
2.5.3 工作条件	2-15
2.5.4 上下电时序	2-18
2.5.5 DC/AC 电气参数	2-18
2.5.6 MIPI/LVDS Rx 电气参数	2-34
2.5.7 MIPI Tx 电气参数	2-37
2.5.8 AUDIO CODEC 电气参数	2-40
2.5.9 USB3 电气参数	2-41
2.5.10 USB2 电气参数	2-41
2.6 接口时序	2-42



2.6.1 DDR 接口时序	2-42
2.6.2 SPI FLASH 接口时序.....	2-46
2.6.3 Ethernet MAC 接口时序.....	2-48
2.6.4 VI 接口时序.....	2-53
2.6.5 VO 接口时序	2-53
2.6.6 AIAO 接口时序.....	2-55
2.6.7 I ² C 时序.....	2-57
2.6.8 SPI 接口时序	2-58
2.6.9 SPI_TFT 接口时序	2-62
2.6.10 MIPI Rx 接口时序.....	2-66
2.6.11 MIPI Tx 接口时序.....	2-68
2.6.12 SDIO/MMC 接口时序	2-70
2.6.13 热成像接口时序	2-81



插图目录

图 2-1 芯片封装顶视图.....	2-2
图 2-2 芯片封装底视图.....	2-3
图 2-3 芯片封装侧视图.....	2-3
图 2-4 Detail A 放大图.....	2-4
图 2-5 无铅回流焊接工艺曲线.....	2-6
图 2-6 封装体测温示意图.....	2-8
图 2-7 干燥真空包装材料示意图.....	2-10
图 2-8 DDR4/LPDDR4/LPDDR4X 命令和地址相对于 CK 的写操作时序图.....	2-42
图 2-9 DDR4/LPDDR4/LPDDR4X DQS 相对于 CK 的写操作时序图.....	2-42
图 2-10 DDR4 CK 相对于 DQS 的读操作时序图.....	2-43
图 2-11 LPDDR4/LPDDR4X CK 相对于 DQS 的读操作时序图.....	2-43
图 2-12 DDR4/LPDDR4/LPDDR4X DQS 相对于 DQ 的读操作时序图.....	2-44
图 2-13 SPI FLASH 输入方向时序图-SDR 模式.....	2-46
图 2-14 SPI FLASH 输入方向时序图-DDR 模式.....	2-46
图 2-15 SPI FLASH 输出方向时序图-SDR 模式.....	2-47
图 2-16 SPI FLASH 输出方向时序图-DDR 模式.....	2-47
图 2-17 RMII 接口 100Mbit/s 接收时序.....	2-48
图 2-18 RMII 接口 100Mbit/s 发送时序.....	2-49
图 2-19 RMII 接口 10Mbit/s 接收时序.....	2-49
图 2-20 RMII 接口 10Mbit/s 发送时序.....	2-49



图 2-21 RMII 接口时序参数.....	2-50
图 2-22 RGMII 接口 1000Mbit/s 接收时序.....	2-51
图 2-23 RGMII 接口 1000Mbit/s 发送时序.....	2-51
图 2-24 MDIO 接口读时序.....	2-52
图 2-25 MDIO 接口写时序.....	2-52
图 2-26 MDIO 接口接收时序参数.....	2-52
图 2-27 VI 接口时序图.....	2-53
图 2-28 BT.656 接口时序.....	2-54
图 2-29 RGB 接口时序.....	2-54
图 2-30 BT.1120 接口时序.....	2-55
图 2-31 I ² S 接口接收时序图.....	2-55
图 2-32 I ² S 接口发送时序图.....	2-55
图 2-33 PCM 接口接收时序图.....	2-56
图 2-34 PCM 接口发送时序图.....	2-56
图 2-35 I ² C 传输时序图.....	2-57
图 2-36 SPICK 时序.....	2-58
图 2-37 SPI 主模式下接口时序 (sph=1)	2-59
图 2-38 SPI 主模式下接口时序 (sph=0)	2-59
图 2-39 SPI_TFT_CLK 时序.....	2-62
图 2-40 SPI_TFT 接口写数据时序 (sph=1)	2-63
图 2-41 SPI_TFT 接口读数据时序 (sph=1)	2-63
图 2-42 SPI_TFT 接口写数据时序 (sph=0)	2-63
图 2-43 SPI_TFT 接口读数据时序 (sph=0)	2-64
图 2-44 MIPI Rx DPHY/Sub-lvds/LVDS/HiSPi 接口时钟数据时序图.....	2-67
图 2-45 MIPI Tx 接口时钟数据时序图.....	2-69
图 2-46 eMMC DS/HS 模式 CMD/DATA 输入/输出方向时序图.....	2-70



图 2-47 HS200 模式的 CMD/DATA 输入/输出方向时序图	2-72
图 2-48 HS400 模式 DATA 输出方向时序图.....	2-73
图 2-49 HS400 模式 DATA 输入方向时序图.....	2-74
图 2-50 HS400 (Enhanced data strobe mode) 模式 CMD 输入方向时序图	2-75
图 2-51 SDIO DS 模式 CMD/DATA 输出方向时序图	2-76
图 2-52 SDIO DS 模式 CMD/DATA 输入方向时序图	2-77
图 2-53 SDIO HS/SDR12/SDR25/SDR50/SDR104 模式 CMD/DATA 输出方向时序图	2-77
图 2-54 SDIO HS/SDR12/SDR25/SDR50/SDR104 模式 CMD/DATA 输入方向时序图	2-77
图 2-55 T0 模式发送方向时序图	2-81
图 2-56 T0 模式接收方向时序图	2-82
图 2-57 T1/T2 模式发送方向时序图.....	2-82
图 2-58 T1/T2 模式接收方向时序图.....	2-83



表格目录

表 2-1 封装参数说明	2-4
表 2-2 Hi3519DV500 管脚数目统计表	2-5
表 2-3 无铅回流焊工艺参数	2-7
表 2-4 IPC/JEDEC 020D 中的无铅器件封装体耐温标准	2-7
表 2-5 混装回流焊工艺参数表	2-8
表 2-6 有铅器件封装体耐温标准	2-9
表 2-7 floor life 参照表	2-11
表 2-8 重新烘烤参考表	2-12
表 2-9 各场景的功耗参数	2-13
表 2-10 各场景具体配置表	2-14
表 2-11 工作环境参数	2-14
表 2-12 结温参数	2-15
表 2-14 使用 SVB 方案时, 电源工作条件	2-16
表 2-15 常压电源工作条件	2-16
表 2-16 破坏性电压值	2-17
表 2-17 DC 电气参数表 (DVDD33=3.3V, IO2)	2-19
表 2-18 DC 电气参数表 (DVDD18=1.8V, IO2)	2-20
表 2-19 DC 电气参数表 (DVDD33=3.3V, IO6)	2-21
表 2-20 DC 电气参数表 (DVDD18=1.8V, IO6)	2-22
表 2-21 DC 电气参数表 (DVDD33=3.3V, EMMC IO)	2-23



表 2-22 DC 电气参数表 (DVDD18=1.8V , EMMC IO)	2-25
表 2-23 DC 电气参数表 (DVDD33=3.3V , IO7_1)	2-27
表 2-24 DC 电气参数表 (DVDD18=1.8V , IO7_1)	2-28
表 2-25 DC 电气参数表 (DVDD18=1.8V , IO7_2)	2-29
表 2-26 AC 和 DC 电气参数表-数据线 (VDDIO_DDR =1.2V , DDR4 模式)	2-30
表 2-27 AC 和 DC 电气参数表-地址/命令线 (VDDIO_DDR =1.2V , DDR4 模式)	2-30
表 2-28 AC 和 DC 电气参数表-数据线 (VDDIO_DDR =1.1V , LPDDR4 模式)	2-31
表 2-29 AC 和 DC 电气参数表 CKE/RESET (VDDIO_DDR =1.1V , LPDDR4 模式)	2-32
表 2-30 AC 和 DC 电气参数表-数据线 (VDDIO_DDR =0.6V , LPDDR4X 模式)	2-33
表 2-31 AC 和 DC 电气参数表 CKE/RESET (VDDIO_DDR =0.6V , LPDDR4X 模式)	2-33
表 2-32 差分 DC 电气参数表	2-34
表 2-33 MIPI DPHY HS (High Speed) DC 参数表	2-35
表 2-34 MIPI DPHY HS (High Speed) AC 参数表	2-36
表 2-35 MIPI DPHY LP (Low Power) DC 参数表	2-37
表 2-36 MIPI DPHY LP (Low Power) AC 参数表	2-37
表 2-37 MIPI Tx LP DC 参数表	2-37
表 2-38 MIPI Tx LP AC 参数表	2-38
表 2-39 MIPI Tx HS DC 参数表	2-39
表 2-40 MIPI Tx HS AC 参数表	2-39
表 2-41 总体指标表	2-40
表 2-42 DAC 主要指标表	2-40
表 2-43 ADC 主要指标表	2-41
表 2-44 MICBIAS 主要指标表	2-41
表 2-45 MICPGA 主要指标	2-41
表 2-46 DDR4 时钟参数表	2-44
表 2-47 LPDDR4/LPDDR4X 时钟参数表	2-45



表 2-48 DDR4 SDRAM 存储器参数表 (DDR4-2666)	2-45
表 2-49 LPDDR4/LPDDR4X SDRAM 存储器参数表 (LPDDR4/LPDDR4X-2666)	2-45
表 2-50 SPI FLASH 输入方向时序参数表 (电压=3.3V)	2-46
表 2-51 SPI FLASH 输入方向时序参数表 (电压=1.8V)	2-47
表 2-52 SPI FLASH 输出方向时序参数表 (电压=3.3V)	2-48
表 2-53 SPI FLASH 输出方向时序参数表 (电压=1.8V)	2-48
表 2-54 RMII 接口时序参数说明	2-50
表 2-55 RGMII 接口时序参数说明	2-51
表 2-56 MDIO 接口时序参数.....	2-53
表 2-57 VI 接口时序参数表.....	2-53
表 2-58 BT.656 接口时序参数表.....	2-54
表 2-59 RGB 接口时序参数表.....	2-54
表 2-60 BT.1120 接口时序参数表 (仅支持 3.3V)	2-55
表 2-61 I ² S 接口时序参数表.....	2-56
表 2-62 PCM 接口时序参数表.....	2-57
表 2-63 I ² C 接口时序参数表.....	2-57
表 2-64 SPI 接口时序参数	2-59
表 2-65 SPI_TFT 接口时序参数	2-64
表 2-66 MIPI Rx DPHY/Sub-lvds/LVDS/HiSpi 时序参数表.....	2-67
表 2-67 MIPI Tx DATA-CLOCK 时序参数表.....	2-69
表 2-68 eMMC DS 模式 CMD/DATA 输入/输出时序参数表.....	2-71
表 2-69 eMMC HS 模式 CMD/DATA 输入/输出时序参数表.....	2-71
表 2-70 HS200 模式 CMD/DATA 输入/输出时钟时序参数表.....	2-72
表 2-71 HS400 模式 DATA 输出方向时序参数表.....	2-74
表 2-72 HS400 模式 DATA 输入方向时序参数表.....	2-74
表 2-73 HS400 (Enhanced data strobe mode) 模式 CMD 输入时序参数表	2-75



表 2-74 SDIO DS 模式输入/输出时序参数表.....	2-78
表 2-75 SDIO HS 模式 CMD/DATA 输入/输出时钟时序参数表.....	2-78
表 2-76 SDIO SDR12 模式 CMD/DATA 输入/输出时钟时序参数表.....	2-79
表 2-77 SDIO SDR25 模式 CMD/DATA 输入/输出时钟时序参数表.....	2-79
表 2-78 SDIO SDR50 模式 CMD/DATA 输入/输出时钟时序参数表.....	2-80
表 2-79 SDIO SDR104 模式 CMD/DATA 输入/输出时钟时序参数表.....	2-81
表 2-80 T0 模式发送方向时序参数表.....	2-81
表 2-81 T0 模式接收方向时序参数表.....	2-82
表 2-82 T1/T2 模式发送方向时序参数表.....	2-83
表 2-83 T1/T2 模式接收方向时序参数表.....	2-83



2 硬件特性

2.1 封装与管脚分布

2.1.1 封装

芯片采用 FCCSP 封装，封装尺寸为 15mmx15mm，管脚间距为 0.65mm。管脚总数为 436 个。

详细封装请参见图 2-1 ~ 图 2-4，封装尺寸参数请参见表 2-1。



图2-1 芯片封装顶视图

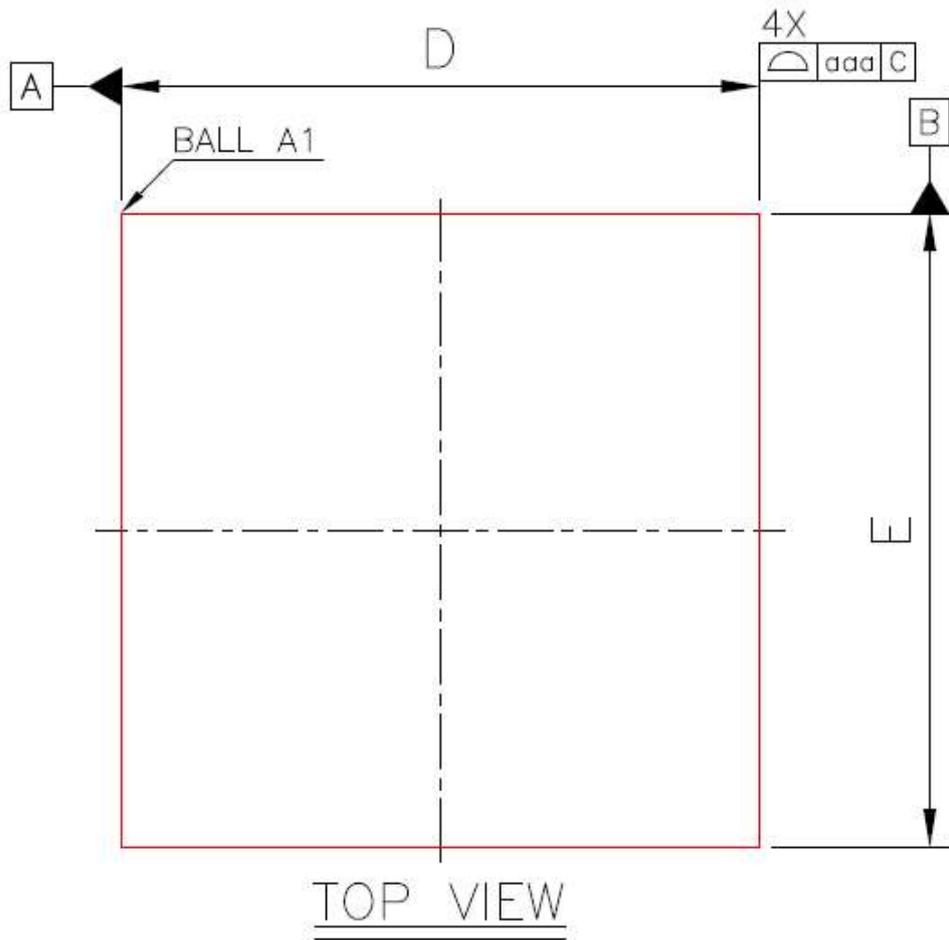




图2-2 芯片封装底视图

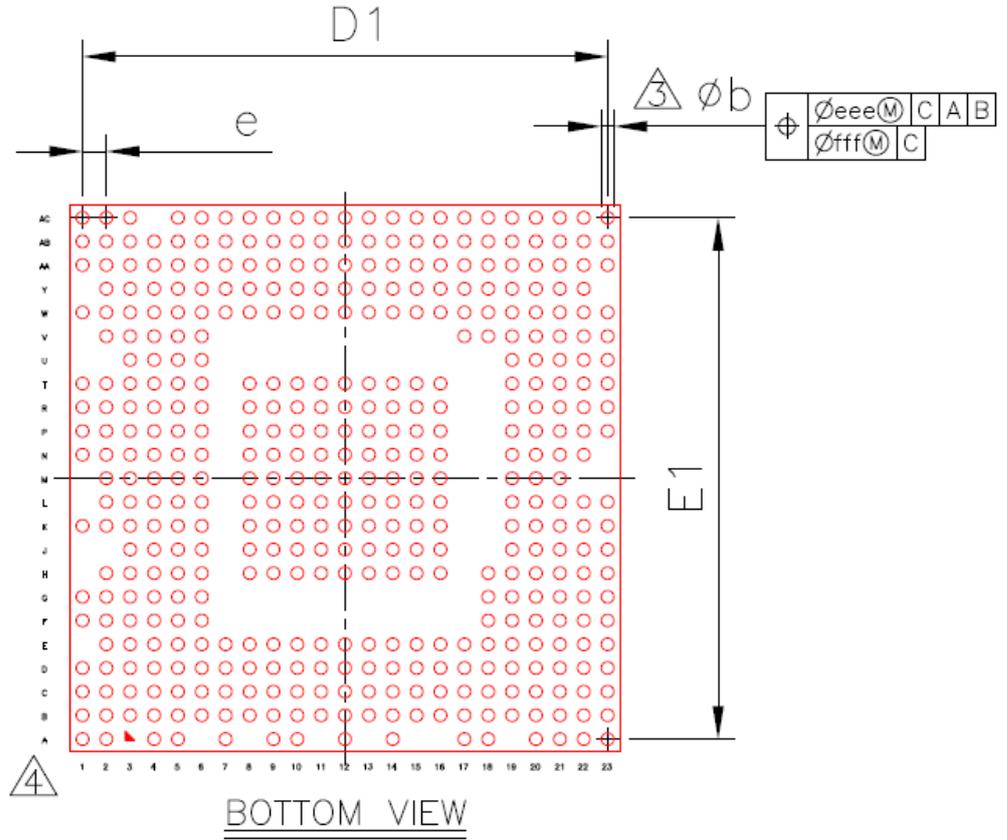


图2-3 芯片封装侧视图

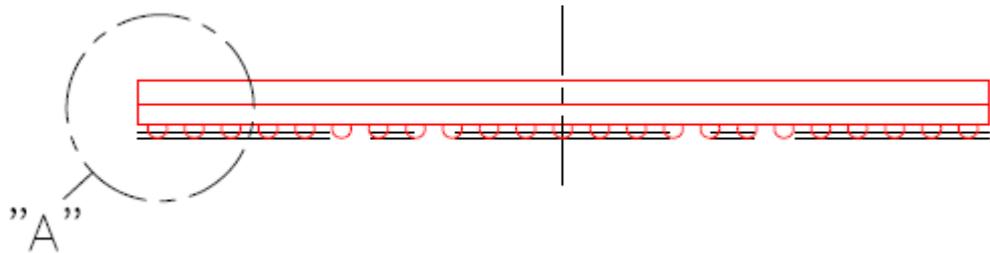




图2-4 Detail A 放大图

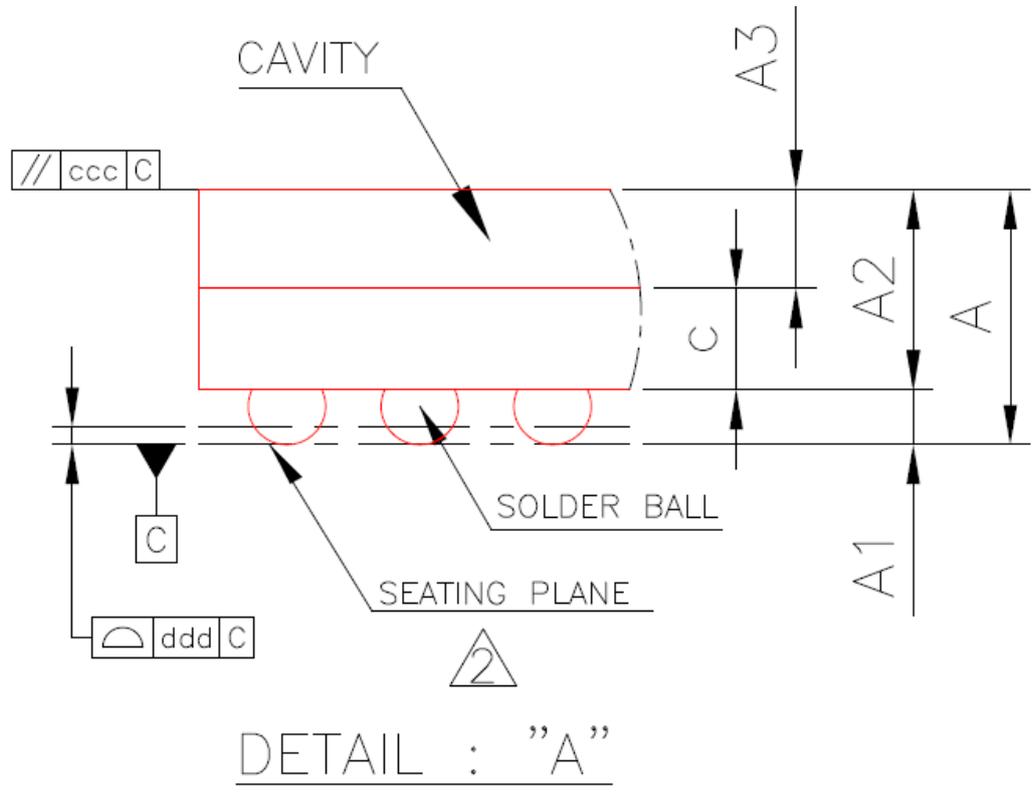


表2-1 封装参数说明

Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	0.969	1.040	1.111	0.038	0.041	0.044
A1	0.200	0.250	0.300	0.008	0.010	0.012
A2	0.740	0.790	0.840	0.029	0.031	0.033
A3	0.420	0.450	0.480	0.017	0.018	0.019
c	0.300	0.340	0.380	0.012	0.013	0.015
D	14.900	15.000	15.100	0.587	0.591	0.594
E	14.900	15.000	15.100	0.587	0.591	0.594
D1	---	14.300	---	---	0.563	---
E1	---	14.300	---	---	0.563	---
e	---	0.650	---	---	0.026	---



Symbol	Dimension in mm			Dimension in inch		
b	0.300	0.350	0.400	0.012	0.014	0.016
aaa	0.15			0.006		
ccc	0.20			0.008		
ddd	0.10			0.004		
eee	0.15			0.006		
fff	0.08			0.003		
MD/ME	23/23					

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. b is solder ball diameter = 0.35 mm (before reflow) bga pad solder mask opening = 0.30 mm pkg ball diameter = 0.36 mm (after reflow)

2.1.2 管脚分布

Hi3519DV500 的管脚有 436 个，管脚数目统计表如表 2-2 所示。

表2-2 Hi3519DV500 管脚数目统计表

管脚类别	数量
I/O	224
数字电源	48
数字地	122
其他/模拟电源	10
其他/模拟地	32
DDR 参考电源 (VREF)	0
NA	0
总计	436



2.2 管脚信息描述

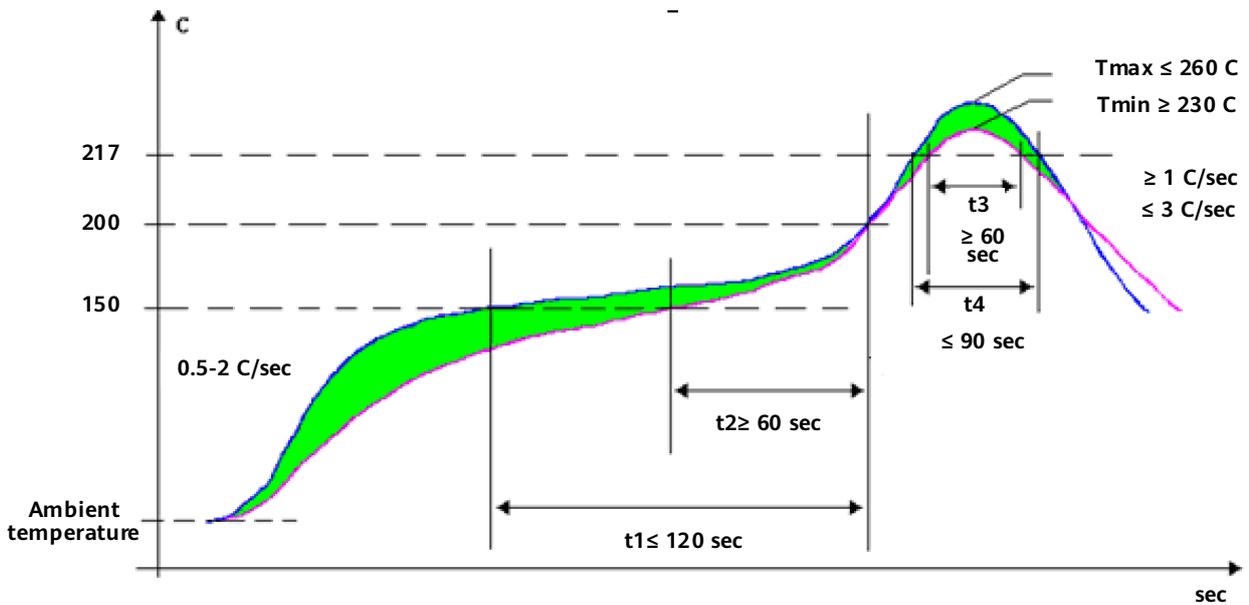
芯片管脚信息、管脚电源域信息、数字管脚默认状态描述请参考文档：
《Hi3519DV500_PINOUT_CN》。

2.3 焊接工艺建议

2.3.1 无铅回流焊工艺参数要求

无铅回流焊焊接工艺曲线如图 2-5 所示。

图2-5 无铅回流焊焊接工艺曲线



无铅回流焊工艺参数如表 2-3 所示。



表2-3 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°C)	60 ~ 150s	≤2.0°C/s	-	-
均温区 (150~200°C)	60 ~ 120s	< 1.0°C/s	-	-
回流区 (> 217°C)	60 ~ 90s	-	230-260°C	-
冷却区 (Tmax ~ 180°C)	-	-	-	1.0°C/s ≤ Slope ≤ 4.0°C/s

说明

- 预热区：温度由 40°C ~ 150°C，温度上升速率控制在 2°C/s 左右，该温区时间为 60 ~ 150s。
- 均温区：温度由 150°C ~ 200°C，稳定缓慢升温，温度上升速率小于 1°C/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°C ~ Tmax ~ 217°C，整个区间时间控制在 60 ~ 90s。
- 冷却区：温度由 Tmax ~ 180°C，温度下降速率最大不能超过 4°C/s。
- 温度从室温 25°C 升温到 250°C 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 2-4 所示。

表2-4 IPC/JEDEC 020D 中的无铅器件封装体耐温标准

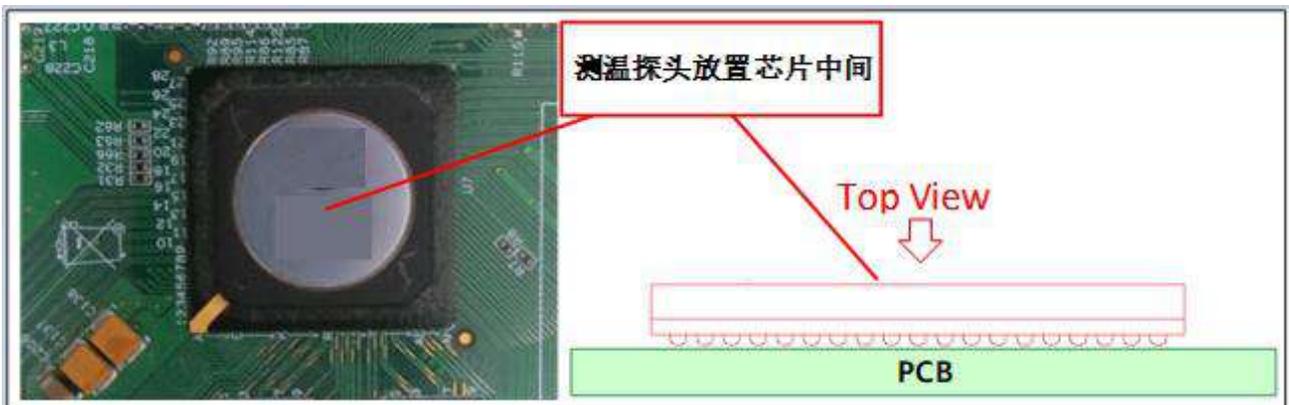
Package Thickness	Volume mm ³ < 350	Volume mm ³ 350~2000	Volume mm ³ > 2000
< 1.6mm	260°C	260°C	260°C
1.6mm ~ 2.5mm	260°C	250°C	245°C
> 2.5mm	250°C	245°C	245°C

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 2-6 所示。

图2-6 封装体测温示意图



说明

如果是 QFP 封装的芯片，直接将测温探头放在管脚处即可。

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 2-5 所示。

表2-5 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°C)	时间	60-150Sec		
	升温斜率	< 2.5°C/Sec		
均温区	时间	30-90Sec		



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

须知

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 020D 中的有铅器件封装体耐温标准如表 2-6。

表2-6 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”。

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: 上海海思产品允许在车间保留的最长时间（环境<30°C/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。
- 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

【潮敏参数】

潮敏等级为 4 级。

2.4.1 上海海思产品防潮包装

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2-7 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 2-8 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 2-8 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 $<30^{\circ}\text{C}/60\% \text{RH}$ 下。

【shelf life】(防潮包装后正常存储时间)

存放环境 $<30^{\circ}\text{C}/60\% \text{RH}$ 下, 真空包装存放, shelf life (存储期限) 不少于 12 个月。

【floor life】

在环境条件 $<30^{\circ}\text{C}/60\%$ 下, floor life 参照表 2-7 如下。

表2-7 floor life 参照表

MSL	Floor life(out of bag) at factory ambient $\leq 30^{\circ}\text{C}/60\% \text{RH}$ or as stated
1	Unlimited at $\leq 30^{\circ}\text{C}/85\% \text{RH}$
2	1 year
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2-8 重新烘烤参考表

Body thickness	level	bake@125°C	bake@90°C $\leq 5\% \text{RH}$	bake@40°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
$\leq 4.5\text{mm}$	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 2-8 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

Hi3519DV500 的功耗参数如表 2-9 所示。

须知

单板电源设计必须参照《Hi3519DV500 硬件设计用户指南》

表2-9 各场景的功耗参数

符号	参数	典型值			最大值	单位
		场景 1	场景 2	场景 3		
DVDD_CPU	CPU 电源	TBD	TBD	TBD	TBD	mW
DVDD	CORE 内核电源	TBD	TBD	TBD	TBD	mW
DVDD_NPU	NPU 内核电源	TBD	TBD	TBD	TBD	mW
VDDIO_DDR	DDR IO 电源	TBD	TBD	TBD	TBD	mW
DVDD18	I/O 电源	TBD	TBD	TBD	TBD	mW
DVDD33	I/O 电源	TBD	TBD	TBD	TBD	mW



说明

以上功耗测试数据基于 Hi3519DV500DMEB 单板测试，实际应用中，不可能出现所有电源域都是最大值的场景，各场景具体配置如表 2-10 所示。功耗参数详情请见《Hi3519DV500 功耗测试报告》。

表2-10 各场景具体配置表

场景	具体配置	业务描述
场景 1	TBD	TBD
场景 2	TBD	TBD

2.5.2 温度和热阻参数

温度和热阻参数如表 2-11、表 2-12 和表 2-13 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2-11 工作环境参数

芯片	工作环境温度 (T_A)	
	最小值 ($^{\circ}\text{C}$)	最大值 ($^{\circ}\text{C}$)
Hi3519DV500	0	70



表2-12 结温参数

芯片	工作结温 (T _J)		破坏性结温 (°C)
	最小值 (°C)	最大值 (°C)	
Hi3519DV500	0	110	125

⚠ 注意

- 在任何条件下芯片的结温都不能大于表 2-12 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。

Hi3519DV500 热阻参数如表 2-13 所示。

表2-13 热阻参数

参数	符号	典型值	单位
Junction-to-ambient thermal resistance	θ_{JA}	17.5	°C/W
Junction-to-board thermal resistance	θ_{JB}	5.5	°C/W
Junction-to-case thermal resistance	θ_{JC}	4.05	°C/W

2.5.3 工作条件

Hi3519DV500 的工作条件如表 2-14 和表 2-15 所示。



须知

- 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。
- 必须验证 SVB 电路的准确性，参考文档《Hi35xxVxxx SVB 电压和寄存器对应关系》

表2-14 使用 SVB 方案时，电源工作条件

符号	描述	SVB 电压范围			单位
		最小值	最大值	容忍的误差范围	
DVDD	CORE 内核电源	TBD	TBD	±0.03	V
DVDD_CPU	CPU 内核电源	TBD	TBD	±0.03	V
DVDD_NPU	NPU 内核电源	TBD	TBD	±0.03	V

表2-15 常压电源工作条件

符号	描述	最小值	典型值	最大值	单位
DVDD	CORE 电源	0.85	0.9	TBD	V
DVDD_CPU	CPU 内核电源	TBD	0.9	TBD	V
DVDD_NPU	NPU 内核电源	TBD	0.9	TBD	V
AVDD_BAT	RTC 电池电源	1.6	3	3.6	V
AVDD33_USB	USB 3.3V 模拟电源	2.97	3.3	3.63	V
AVDD09_USB3	USB 0.9V 模拟电源	0.81	0.9	0.99	V
AVDD18_MIPIRX	MIPIRX 1.8V 模拟电源	1.62	1.8	1.98	V
DVDD3318_MIPITX	MIPITX 3.3V/1.8V 数字电源	2.97/1.62	3.3/1.8	3.63/1.98	V
AVDD18_MIPITX	MIPITX 1.8V 模拟电源	1.62	1.8	1.98	V



符号	描述	最小值	典型值	最大值	单位
AVDD33_AC	Audio Codec 3.3V 模拟电源	2.97	3.3	3.63	V
AVDD33_PLL	PLL 3.3V 模拟电源	2.97	3.3	3.63	V
AVDD09_PLL	PLL 0.9V 模拟电源	0.81	0.9	0.99	V
DVDD33_1	3.3V 数字信号电源 1	2.97	3.3	3.63	V
DVDD33_2	3.3V 数字信号电源 2	2.97	3.3	3.63	V
DVDD33_3	3.3V 数字信号电源 3	2.97	3.3	3.63	V
DVDD33_4	3.3V 数字信号电源 4	2.97	3.3	3.63	V
DVDD18_SDIO0	SDIO0 1.8V 电源	1.62	1.8	1.98	V
SDIO0_VOUT	SDIO0 3.3V/1.8V 电源	2.97/1.62	3.3/1.8	3.63/1.98	V
DVDD3318_EMMC	EMMC 3.3/1.8V 电源	2.97/1.62	3.3/1.8	3.63/1.98	V
DVDD3318_PMC	PMC 3.3/1.8V 电源	2.97/1.62	3.3/1.8	3.63/1.98	V
DVDD3318_SDIO1	SDIO1 3.3/1.8V 电源	2.97/1.62	3.3/1.8	3.63/1.98	V
DVDD3318_SENSOR	SENSOR 3.3/1.8V 电源	2.97/1.62	3.3/1.8	3.63/1.98	V
VDDIO_DDR	DDR4/LPDDR4/ LPDDR4x IO 电源	1.14/1.045/ 0.57	1.2/1.1/ 0.6	1.26/1.155/ 0.65	V
VDDIO_DDR_CK	DDR4/LPDDR4/ LPDDR4x CLK IO 电源	1.14/1.045/ 0.57	1.2/1.1/ 0.6	1.26/1.155/ 0.65	V
VDDIOA_DDR	DDR4/LPDDR4/ LPDDR4x IO 电源 A	1.14/1.14/ 1.14	1.2/1.1/ 1.1	1.26/1.26/ 1.26	V
AVDD33_DDR_PLL	DDR PLL 电源	2.97	3.3	3.63	V

表2-16 破坏性电压值

参数描述	电源管脚名称	最小值 (V)	最大值 (V)
core 电源供电电压	DVDD/DVDD_CPU/DVDD_N PU	-0.2	1.17



参数描述	电源管脚名称	最小值 (V)	最大值 (V)
IP core 电源 供电电压	AVDD09_PLL/ AVDD09_USB3	-0.2	1.17
DDR IO 电 源供电电压	VDDIO_DDR/ VDDIO_DDR_CK/ VDDIOA_DDR	-0.2	1.44 (DDR4) 1.32 (LPDDR4) 0.72 (LPDDR4X)
1.8V IO 电 源供电电压	DVDD18_SDIO0/ AVDD18_MIPITX/ AVDD18_MIPIRX	-0.2	2.16
3.3V IO 电 源供电电压	DVDD33_1/ DVDD33_2/ DVDD33_3/ DVDD33_4/ AVDD33_DDR_PLL/ AVDD33_PLL/ AVDD33_AC/ AVDD33_USB	-0.2	3.96
3.3V/1.8V IO 电源供电 电压	DVDD3318_SDIO1 DVDD3318_MIPITX DVDD3318_EMMC DVDD3318_SENSOR DVDD3318_PMC	-0.2	2.16(1.8V 供电时) 3.96(3.3V 供电时)

⚠ 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 上下电时序

请参考《Hi3519DV500 硬件设计用户指南》上下电时序章节。

2.5.5 DC/AC 电气参数

📖 说明

以下表中涉及的 IO2/IO6/EMMC IO 与接口的对应关系请参见《Hi3519DV500_PINOUT_CN》。

数字接口 DC 电气参数如表 2-17~表 2-25 所示。



表2-17 DC 电气参数表 (DVDD33=3.3V, IO2)

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	3.63	V	-
V _{IH}	高电平输入电压	0.65*DVDD33	-	3.63	V	接口不兼容 5V 输入
V _{IL}	低电平输入电压	-0.3	-	0.35*DVDD33	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	4.67	-	-	mA	IO2_档位 1
		9.15	-	-	mA	IO2_档位 2
		13.49	-	-	mA	IO2_档位 3
		17.61	-	-	mA	IO2_档位 4
I _{OL}	低电平输出电流	3.98	-	-	mA	IO2_档位 1
		7.57	-	-	mA	IO2_档位 2
		10.7	-	-	mA	IO2_档位 3
		13.54	-	-	mA	IO2_档位 4
R _{PU1}	内部上拉电阻	32.34	41.84	54.89	kΩ	-
R _{PD1}	内部下拉电阻	29.64	38.51	52.35	kΩ	-
R _{PD5}	内部下拉电阻	20.91	29.69	47.96	kΩ	-
R _{PU6}	内部上拉电阻	15.47	20.94	30.03	kΩ	-
R _{PD6}	内部下拉电阻	57.23	89.66	162.8	kΩ	-
R _{PU7}	内部上拉电阻	16.93	22.72	32.52	kΩ	-
R _{PD7}	内部下拉电阻	58.19	91.15	165.1	kΩ	-



表2-18 DC 电气参数表 (DVDD18=1.8V, IO2)

符号	参数	最小值	典型值	最大值	单位	说明
DVDD18	接口电压	1.62	1.8	1.98	V	-
V _{IH}	高电平输入电压	0.65*DVDD18	-	1.98	V	接口不兼容 5V 输入
V _{IL}	低电平输入电压	-0.3	-	0.35*DVDD18	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	2.01	-	-	mA	IO2_档位 1
		3.97	-	-	mA	IO2_档位 2
		5.89	-	-	mA	IO2_档位 3
		7.76	-	-	mA	IO2_档位 4
I _{OL}	低电平输出电流	1.87	-	-	mA	IO2_档位 1
		3.66	-	-	mA	IO2_档位 2
		5.35	-	-	mA	IO2_档位 3
		6.51	-	-	mA	IO2_档位 4
R _{PU1}	内部上拉电阻	41.81	59.05	87.99	kΩ	-
R _{PD1}	内部下拉电阻	35.94	55.68	100.2	kΩ	-
R _{PD5}	内部下拉电阻	37.05	62.45	116.2	kΩ	-
R _{PU6}	内部上拉电阻	28.36	41.95	64.08	kΩ	-
R _{PD6}	内部下拉电阻	116.9	213.3	430.5	kΩ	-
R _{PU7}	内部上拉电阻	30.37	44.73	68.5	kΩ	-



符号	参数	最小值	典型值	最大值	单位	说明
R _{PD7}	内部下拉电阻	119.2	216.9	436.7	kΩ	-

表2-19 DC 电气参数表 (DVDD33=3.3V , IO6)

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	3.63	V	-
V _{IH}	高电平输入电压	0.65*DVDD33	-	3.63	V	接口不兼容 5V 输入
V _{IL}	低电平输入电压	-0.3	-	0.35*DVDD33	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	6.161	-	-	mA	IO6_档位 1
		12.32	-	-	mA	IO6_档位 2
		24.64	-	-	mA	IO6_档位 3
		30.8	-	-	mA	IO6_档位 4
		49.29	-	-	mA	IO6_档位 5
		55.45	-	-	mA	IO6_档位 6
		67.77	-	-	mA	IO6_档位 7
		73.93	-	-	mA	IO6_档位 8
I _{OL}	低电平输出电流	2.813	-	-	mA	IO6_档位 1
		5.262	-	-	mA	IO6_档位 2
		11.25	-	-	mA	IO6_档位 3



符号	参数	最小值	典型值	最大值	单位	说明
		14.07	-	-	mA	IO6_档位 4
		22.5	-	-	mA	IO6_档位 5
		25.32	-	-	mA	IO6_档位 6
		30.94	-	-	mA	IO6_档位 7
		33.76	-	-	mA	IO6_档位 8
R _{PU11}	内部上拉电阻	23.24	30.02	38.07	kΩ	-
R _{PD11}	内部下拉电阻	22.11	28.53	36.48	kΩ	-

表2-20 DC 电气参数表 (DVDD18=1.8V, IO6)

符号	参数	最小值	典型值	最大值	单位	说明
DVDD18	接口电压	1.62	1.8	1.98	V	-
V _{IH}	高电平输入电压	0.65*DVDD33	-	1.98	V	接口不兼容 5V 输入
V _{IL}	低电平输入电压	-0.3	-	0.35*DVDD33	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	1.888	-	-	mA	IO6_档位 1
		3.775	-	-	mA	IO6_档位 2
		7.551	-	-	mA	IO6_档位 3
		9.438	-	-	mA	IO6_档位 4
		15.1	-	-	mA	IO6_档位 5



符号	参数	最小值	典型值	最大值	单位	说明
		16.99	-	-	mA	IO6_档位 6
		20.76	-	-	mA	IO6_档位 7
		22.65	-	-	mA	IO6_档位 8
I _{OL}	低电平输出电流	1.435	-	-	mA	IO6_档位 1
		2.871	-	-	mA	IO6_档位 2
		5.741	-	-	mA	IO6_档位 3
		7.177	-	-	mA	IO6_档位 4
		11.48	-	-	mA	IO6_档位 5
		12.92	-	-	mA	IO6_档位 6
		15.79	-	-	mA	IO6_档位 7
		17.22	-	-	mA	IO6_档位 8
R _{PU11}	内部上拉电阻	25.54	33.9	45.02	kΩ	-
R _{PD11}	内部下拉电阻	23.35	31.44	44.53	kΩ	-

表2-21 DC 电气参数表 (DVDD33=3.3V, EMMC IO)

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	3.63	V	-
V _{IH}	高电平输入电压	0.65*DVDD33	-	3.63	V	接口不兼容 5V 输入
V _{IL}	低电平输入电压	-0.3	-	0.35*DVDD33	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	-



符号	参数	最小值	典型值	最大值	单位	说明
V_{OL}	低电平输出电压	-	-	0.4	V	-
I_{OH}	高电平输出电流	7.676	-	-	mA	EMMC IO_档位 1
		15.35	-	-	mA	EMMC IO_档位 2
		23.03	-	-	mA	EMMC IO_档位 3
		30.71	-	-	mA	EMMC IO_档位 4
		46.06	-	-	mA	EMMC IO_档位 5
		53.74	-	-	mA	EMMC IO_档位 6
		61.41	-	-	mA	EMMC IO_档位 7
		69.09	-	-	mA	EMMC IO_档位 8
		76.77	-	-	mA	EMMC IO_档位 9
		84.44	-	-	mA	EMMC IO_档位 10
		92.12	-	-	mA	EMMC IO_档位 11
		99.8	-	-	mA	EMMC IO_档位 12
		115.2	-	-	mA	EMMC IO_档位 13
		122.8	-	-	mA	EMMC IO_档位 14
		130.5	-	-	mA	EMMC IO_档位 15
		138.2	-	-	mA	EMMC IO_档位 16
I_{OL}	低电平输出电流	3.487	-	-	mA	EMMC IO_档位 1
		6.974	-	-	mA	EMMC IO_档位 2
		10.46	-	-	mA	EMMC IO_档位 3
		13.95	-	-	mA	EMMC IO_档位 4
		20.92	-	-	mA	EMMC IO_档位 5
		24.41	-	-	mA	EMMC IO_档位 6



符号	参数	最小值	典型值	最大值	单位	说明
		27.89	-	-	mA	EMMC IO_档位 7
		31.38	-	-	mA	EMMC IO_档位 8
		34.87	-	-	mA	EMMC IO_档位 9
		38.35	-	-	mA	EMMC IO_档位 10
		41.84	-	-	mA	EMMC IO_档位 11
		45.33	-	-	mA	EMMC IO_档位 12
		52.3	-	-	mA	EMMC IO_档位 13
		55.79	-	-	mA	EMMC IO_档位 14
		59.27	-	-	mA	EMMC IO_档位 15
		62.76	-	-	mA	EMMC IO_档位 16
R _{PU20}	内部上拉电阻	23.26	30.16	38.34	kΩ	-
R _{PD20}	内部下拉电阻	22.15	28.68	36.76	kΩ	-
R _{PU21}	内部上拉电阻	23.26	30.16	38.34	kΩ	-
R _{PD21}	内部下拉电阻	22.15	28.68	36.76	kΩ	-

表2-22 DC 电气参数表 (DVDD18=1.8V , EMMC IO)

符号	参数	最小值	典型值	最大值	单位	说明
DVDD18	接口电压	1.62	1.8	1.98	V	-
V _{IH}	高电平输入电压	0.65*DVDD33	-	1.98	V	接口不兼容 5V 输入
V _{IL}	低电平输入电压	-0.3	-	0.35*DVDD33	V	-
I _L	输入漏电流	-	-	±10	μA	-
I _{oz}	三态输出漏电流	-	-	±10	μA	-



符号	参数	最小值	典型值	最大值	单位	说明
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	2.354	-	-	mA	EMMC IO_档位 1
		4.078	-	-	mA	EMMC IO_档位 2
		7.064	-	-	mA	EMMC IO_档位 3
		9.418	-	-	mA	EMMC IO_档位 4
		14.13	-	-	mA	EMMC IO_档位 5
		16.48	-	-	mA	EMMC IO_档位 6
		18.84	-	-	mA	EMMC IO_档位 7
		21.19	-	-	mA	EMMC IO_档位 8
		23.55	-	-	mA	EMMC IO_档位 9
		25.9	-	-	mA	EMMC IO_档位 10
		28.26	-	-	mA	EMMC IO_档位 11
		30.61	-	-	mA	EMMC IO_档位 12
		35.32	-	-	mA	EMMC IO_档位 13
		37.68	-	-	mA	EMMC IO_档位 14
		40.03	-	-	mA	EMMC IO_档位 15
		42.39	-	-	mA	EMMC IO_档位 16
I_{OL}	低电平输出电流	1.788	-	-	mA	EMMC IO_档位 1
		3.577	-	-	mA	EMMC IO_档位 2
		5.364	-	-	mA	EMMC IO_档位 3
		7.153	-	-	mA	EMMC IO_档位 4
		10.73	-	-	mA	EMMC IO_档位 5



符号	参数	最小值	典型值	最大值	单位	说明
		12.52	-	-	mA	EMMC IO_档位 6
		14.3	-	-	mA	EMMC IO_档位 7
		16.09	-	-	mA	EMMC IO_档位 8
		17.88	-	-	mA	EMMC IO_档位 9
		19.67	-	-	mA	EMMC IO_档位 10
		21.45	-	-	mA	EMMC IO_档位 11
		23.24	-	-	mA	EMMC IO_档位 12
		26.82	-	-	mA	EMMC IO_档位 13
		28.61	-	-	mA	EMMC IO_档位 14
		30.39	-	-	mA	EMMC IO_档位 15
		32.18	-	-	mA	EMMC IO_档位 16
R _{PU20}	内部上拉电阻	25.56	34.04	44.74	kΩ	-
R _{PD20}	内部下拉电阻	23.36	31.58	44.74	kΩ	-
R _{PU21}	内部上拉电阻	25.56	34.04	44.74	kΩ	-
R _{PD21}	内部下拉电阻	23.36	31.58	44.74	kΩ	-

表2-23 DC 电气参数表 (DVDD33=3.3V , IO7_1)

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	3.63	V	-
V _{IH}	高电平输入电压	0.65*DVDD33	-	3.63	V	接口不兼容 5V 输入
V _{IL}	低电平输入电压	-0.3	-	0.35*DVDD33	V	-



符号	参数	最小值	典型值	最大值	单位	说明
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	7.101	-	-	mA	IO7_1 档位 1
		14.2	-	-	mA	IO7_1 档位 2
		21.3	-	-	mA	IO7_1 档位 3
		28.41	-	-	mA	IO7_1 档位 4
I_{OL}	低电平输出电流	10.59	-	-	mA	IO7_1 档位 1
		21.17	-	-	mA	IO7_1 档位 2
		31.76	-	-	mA	IO7_1 档位 3
		42.34	-	-	mA	IO7_1 档位 4

表2-24 DC 电气参数表 (DVDD18=1.8V, IO7_1)

符号	参数	最小值	典型值	最大值	单位	说明
DVDD18	接口电压	1.62	1.8	1.98	V	-
V_{IH}	高电平输入电压	$0.65 \cdot DVDD18$	-	1.98	V	接口不兼容 5V 输入
V_{IL}	低电平输入电压	-0.3	-	$0.35 \cdot DVDD18$	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	-



符号	参数	最小值	典型值	最大值	单位	说明
V _{OL}	低电平输出电压	-	-	0.45	V	-
I _{OH}	高电平输出电流	4.331	-	-	mA	IO7_1 档位 1
		8.661	-	-	mA	IO7_1 档位 2
		12.99	-	-	mA	IO7_1 档位 3
		17.32	-	-	mA	IO7_1 档位 4
I _{OL}	低电平输出电流	3.737	-	-	mA	IO7_1 档位 1
		7.474	-	-	mA	IO7_1 档位 2
		11.21	-	-	mA	IO7_1 档位 3
		14.95	-	-	mA	IO7_1 档位 4

表2-25 DC 电气参数表 (DVDD18=1.8V , IO7_2)

符号	参数	最小值	典型值	最大值	单位	说明
DVDD18	接口电压	1.62	1.8	1.98	V	-
V _{IH}	高电平输入电压	0.65*DVDD18	-	1.98	V	接口不兼容 5V 输入
V _{IL}	低电平输入电压	-0.3	-	0.35*DVDD18	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	1.537	-	-	mA	IO7_2_档位 1
		3.951	-	-	mA	IO7_2_档位 2
I _{OL}	低电平输出电流	1.976	-	-	mA	IO7_2_档位 1



符号	参数	最小值	典型值	最大值	单位	说明
		3.074	-	-	mA	IO7_2_档位 2

DDR 模式下，AC 和 DC 电气参数如表 2-26~表 2-31 所示。

表2-26 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_DDR	接口电压	1.14	1.2	1.26	V	-
VDDIOA_DDR	接口电压	1.14	1.2	1.26	V	-
Vref	参考电压	0.4 * VDDIO_DDR	0.5 * VDDIO_DDR	0.8 * VDDIO_DDR	V	-
V _{OH(DC)}	高电平输出电压	-	0.9 * VDDIO_DDR	-	V	基于 ODT=600hm
V _{OL(DC)}	低电平输出电压	-	0.5* VDDIO_DDR	-	V	
V _{OH(AC)}	高电平输出电压	-	1.1 * VDDIO_DDR	-	V	
V _{OL(AC)}	低电平输出电压	-	0.3* VDDIO_DDR	-	V	
R _{ON}	输出阻抗	40	-	240	Ω	-

表2-27 AC 和 DC 电气参数表-地址/命令线 (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_DDR	接口电压	1.14	1.2	1.26	V	-
VDDIOA_DDR	接口电压	1.14	1.2	1.26	V	-
Vref	参考电压	0.49 *	0.5 *	0.51 *	V	-



符号	参数	最小值	典型值	最大值	单位	说明
		VDDIO_DDR	VDDIO_DDR	VDDIO_DDR		
$V_{IH(DC65)}$	高电平输入电压	$V_{ref} + 0.065$	-	VDDIO_DDR	V	-
$V_{IL(DC65)}$	低电平输入电压	0	-	$V_{ref} - 0.065$	V	-
$V_{IH(AC90)}$	高电平输入电压	$V_{ref} + 0.09$	-	VDDIO_DDR	V	-
$V_{IL(AC90)}$	低电平输入电压	0	-	$V_{ref} - 0.09$	V	-
$V_{OH(DC)}$	高电平输出电压	-	$0.9 * V_{DDIO_DDR}$	-	V	基于 ODT=60Ohm
$V_{OL(DC)}$	低电平输出电压	-	$0.5 * V_{DDIO_DDR}$	-	V	
$V_{OH(AC)}$	高电平输出电压	-	$1.1 * V_{DDIO_DDR}$	-	V	
$V_{OL(AC)}$	低电平输出电压	-	$0.3 * V_{DDIO_DDR}$	-	V	
R_{ON}	输出阻抗	40	-	240	Ω	-

表2-28 AC 和 DC 电气参数表-数据线 (VDDIO_DDR =1.1V, LPDDR4 模式)

符号	参数	最小值	典型值	最大值	单位	说明
VDDIO_DDR	接口电压	1.045	1.1	1.155	V	-
VDDIOA_DDR	接口电压	1.045	1.1	1.155	V	-
Vref	参考电压	$0.1 * V_{DDIO_D}$ DR	$0.272 * V_{DDIO_DDR}$	$0.42 * V_{DDIO_DDR}$	V	-
$V_{IH(AC)}$	高电平输入	$V_{ref} + 0.07$	-	-	mV	-



符号	参数	最小值	典型值	最大值	单位	说明
	入摆幅					
$V_{IL(AC)}$	低电平输入摆幅	0	-	$V_{ref}-0.07$	mV	-
$V_{OH(DC)}$	高电平输出电压	-	$0.3*VDDIO_DDR$	-	V	基于 ODT=60Ohm
$V_{OL(DC)}$	低电平输出电压	-	$0.1*VDDIO_DDR$	-	V	
$V_{OH(AC)}$	高电平输出电压	-	$0.5*VDDIO_DDR$	-	V	
$V_{OL(AC)}$	低电平输出电压	-	$-0.1*VDDIO_DDR$	-	V	
R_{ON}	输出阻抗	40	-	240	Ω	-

表2-29 AC 和 DC 电气参数表 CKE/RESET ($VDDIO_DDR = 1.1V$, LPDDR4 模式)

符号	参数	最小值	典型值	最大值	单位	说明
$VDDIO_DDR$	接口电压	1.045	1.1	1.155	V	-
$VDDIOA_DDR$	接口电压	1.045	1.1	1.155	V	-
$V_{OH(AC)}$	高电平输出电压	$0.75*VDDIO_DDR$	-	$VDDIO_DDR+0.2$	V	-
$V_{OL(AC)}$	低电平输出电压	-0.2	-	$0.25*VDDIO_DDR$	V	-
$V_{OH(DC)}$	高电平输出电压	$0.65*VDDIO_DDR$	-	$VDDIO_DDR+0.2$	V	-
$V_{OL(DC)}$	低电平输出电压	-0.2	0	$0.35*VDDIO_DDR$	V	-
R_{ON}	输出阻抗	40	-	240	Ω	-



表2-30 AC 和 DC 电气参数表-数据线 (VDDIO_DDR =0.6V, LPDDR4X 模式)

符号	参数	最小值	典型值	最大值	单位	说明
VDDIO_DDR	接口电压	0.57	0.6	0.63	V	-
VDDIOA_DDR	接口电压	1.045	1.1	1.155	V	-
Vref	参考电压	0.15 * VDDIO_D DR	-	0.449* VDDIO_DDR	-	-
V _{IH}	高电平输入 电压	Vref + 0.07	-	VDDIO_DDR	V	-
V _{IL}	低电平输入 电压	0	-	Vref -0.07	V	-
V _{OH(DC)}	高电平输出 电压	-	0.5*VDDIO_DDR	-	V	基于 ODT=60Oh m
V _{OL(DC)}	低电平输出 电压	-	0.15*VDDIO_DDR	-	V	
V _{OH(AC)}	高电平输出 电压	-	0.8*VDDIO_DDR	-	V	
V _{OL(AC)}	低电平输出 电压	-	-0.1*VDDIO_DDR	-	V	
R _{ON}	输出阻抗	40	-	240	Ω	-

表2-31 AC 和 DC 电气参数表 CKE/RESET (VDDIO_DDR =0.6V, LPDDR4X 模式)

符号	参数	最小值	典型值	最大值	单位	说明
VDDIO_DDR	接口电压	0.57	0.6	0.63	V	-
VDDIOA_DDR	接口电压	1.045	1.1	1.155	V	-
V _{OH(AC)}	高电平输 出电压	0.75*VDDIO_DDR	-	VDDIO_DDR+0.2	V	-



符号	参数	最小值	典型值	最大值	单位	说明
$V_{OL(AC)}$	低电平输出电压	-0.2	-	$0.25 \cdot V_{DDIO_DDR}$	V	-
$V_{OH(DC)}$	高电平输出电压	$0.65 \cdot V_{DDIO_DDR}$	-	$V_{DDIO_DDR} + 0.2$	V	-
$V_{OL(DC)}$	低电平输出电压	-0.2	0	$0.35 \cdot V_{DDIO_DDR}$	V	-
R_{ON}	输出阻抗	40	-	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 2-32 所示。

表2-32 差分 DC 电气参数表

符号	参数	最小值	典型值	最大值	单位	
WIDTH(SL)	Differential Input Threshold Voltage $ (V_P - V_M) $	Sub-LVDS	70	-	mV	
WIDTH(HS)		HiSpi(SLV S)	70	-		
WIDTH(HiV CM)		HiSpi(HiV CM)	100	-		
WIDTH(DP)		D-PHY HS	70	-		
WIDTH(LV)		LVDS	100	-		
WIDTH(ML)		Mini-LVDS	100	-		
VCM(SL)	Common Mode Voltage Range $(V_P + V_M)/2$	Sub-LVDS	0.5	0.9	V	
VCM(HS)		HiSpi(SLV S)	0.07	0.2		0.35
VCM(HiV CM)		HiSpi(HiV CM)	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



符号	参数		最小值	典型值	最大值	单位
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(SLV S)	-0.04		0.49	
VCM(HiVCM)		HiSPi(HiV CM)	0.55		1.35	
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	
ZID(HS)	HiSPi(SLV S)		125			
ZID(HiVCM)	HiSPi(HiV CM)		125			
ZID(LV)	LVDS		120			
ZID(ML)	Mini-LVDS		120			
ZID(DP)	D-PHY HS		125			

MIPI 参数如表 2-33~表 2-36 所示。

表2-33 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
VCMRX(DC)	Common-mode voltage HS receive mode	70	-	330	mV
VIDTH	Differential input high threshold	-	-	70(data rate <= 1.5Gbps)	mV



符号	参数	最小值	典型值	最大值	单位
		-	-	40(data rate >1.5Gbps)	mV
VIDTL	Differential input high threshold	-70(data rate ≤ 1.5Gbps)	-	-	mV
		-40(data rate >1.5Gbps)	-	-	mV
VIHHS	Single-ended input high voltage	-	-	460	mV
VILHS	Single-ended input low voltage	-40	-	-	mV

表2-34 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 when datarate ≤1.5Gbps	-	-	100	mV
	Common-mode interface beyond 450MHz when datarate >1.5Gbps			50	
Δ VCMRX(LF)	Common-mode interface 50MHz-450MHz when datarate ≤1.5Gbps	-50	-	50	mV
	Common-mode interface 50MHz-450MHz when datarate >1.5Gbps	-25		25	
CCM	Common-mode termination	-	-	60	pF



表2-35 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 when datarate $\leq 1.5\text{Gbps}$	880	-	-	mV
	Logic 1 input voltage when datarate $> 1.5\text{Gbps}$	740	-	-	
VILLP	Logic 0 input voltage	-	-	550	
VHYST	Input hysteresis	25	-	-	

表2-36 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}	Minimum 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 2-37 所示。

表2-37 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 (data rate $\leq 1.5\text{Gbps}$)	1.1	1.2	1.3	V
	Thevenin output high level (data rate $> 1.5\text{Gbps}$)	0.95	-	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 2-38 所示。

表2-38 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 <70pF	-	-	25	ns
T _{FLP}	15%-85% fall time, C _L <70pF	-	-	25	ns
T _{REOT}	30%-85% rise time and fall time, C _L <70pF	-	-	35	ns
T _{LP-PER-TX}	Period of the LP exclusive-OR clock	90	-	-	ns
C _{LOAD}	Load capacitance	0	-	70	pF
ΔV / Δt _{SR}	15% to 85%, C _L <70pF	-	-	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 2-39 所示。



表2-39 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_{\text{CMTX}}(1,0) $	Vcmtx mismatch when output is differential-1 or differential-0	-	-	5	mV
$ V_{\text{OD}} $	HS transmit differential voltage	140	200	270	mV
$ \Delta V_{\text{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 2-40 所示。

表2-40 MIPI Tx HS AC 参数表

符号	参数	最小值	典型值	最大值	单位	备注
$\Delta V_{\text{CMTX(HF)}}$	common-level variations above 450Mhz	-	-	15	mV_{RMS}	
$\Delta V_{\text{CMTX(LF)}}$	common-level variations between 50-450Mhz	-	-	25	mV_{PEAK}	-
t_{R} and t_{F}	20%-80% rise time and fall time	-	-	0.3	UI	Data rates $\leq 1\text{Gbps}$
		-	-	0.35	UI	$1\text{Gbps} < \text{Data rates} \leq 1.5\text{Gbps}$



符号	参数	最小值	典型值	最大值	单位	备注
		100	-	-	ps	Data rates <=1.5Gbps. However, data rates < 1 Gbps, not blow 150 ps
		-	-	0.4	UI	Data rates > 1.5Gbps
		50	-	-	ps	Data rate> 1.5 Gbps. However, data rates <= 1.5 Gbps, not below 100 ps and data rates <= 1 Gbps not below 150 ps.

2.5.8 AUDIO CODEC 电气参数

Audio Code 电性能参数如表 2-41、表 2-42、表 2-43、表 2-44 和表 2-45 所示。

表2-41 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	2.97	3.3	3.63	V	相对于 AGND
VREF	-	AVDD/2	-	V	相对于 AGND

表2-42 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	-	1	-	Vrms	单端最大输出信号摆幅



表2-43 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	-	2	-	Vrms	ADC 最大差分输入信号摆幅

表2-44 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	$2.1 \times AVD$ D/3.3	-	$2.8 \times AV$ DD/3.3	V	麦克风偏置电压
最大输出电流	-	-	3	mA	-

表2-45 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	-	2	-	Vrms	最大差分输入信号摆幅
输入阻抗	-	28.44	-	kΩ	MICPGA 的输入阻抗

2.5.9 USB3 电气参数

USB3.0 标准电气特性请参考标准协议《Universal Serial Bus 3.0 Specification》中第 6 章节，协议下载链接 <http://www.softelectro.ru/usb30.pdf>。

📖 说明

电气特性完全符合 USB3.0 版本特性规范要求。

2.5.10 USB2 电气参数

USB2 标准电气特性请参考标准协议《Universal Serial Bus 2.0 Specification》中 7 章节，协议下载链接 <https://www.usb.org/document-library/usb-20-specification>

📖 说明

电气特性完全符合 USB2.0 版本特性规范要求。



2.6 接口时序

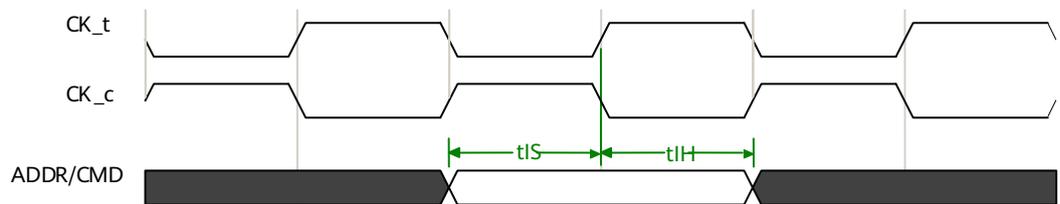
2.6.1 DDR 接口时序

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。DDR4/LPDDR4/LPDDR4X 命令和地址相对于 CK 的写操作时序如图 2-8 所示。

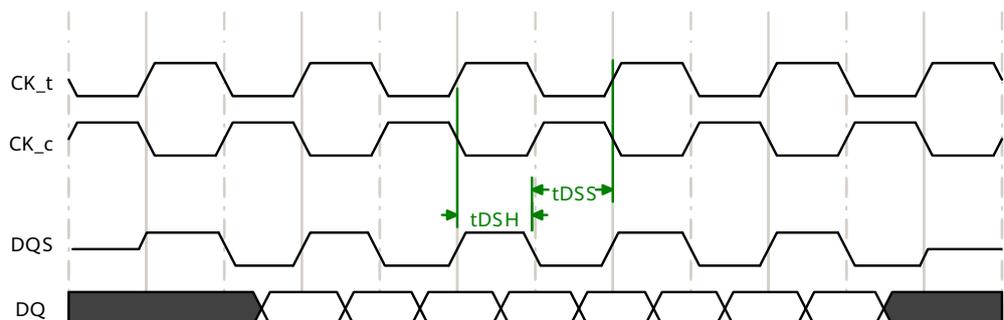
图2-8 DDR4/LPDDR4/LPDDR4X 命令和地址相对于 CK 的写操作时序图



DQS 相对于 CK 的写操作时序

对于 DDR4/LPDDR4/LPDDR4X 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 2-9 所示。

图2-9 DDR4/LPDDR4/LPDDR4X DQS 相对于 CK 的写操作时序图





2.6.1.2 读操作时序

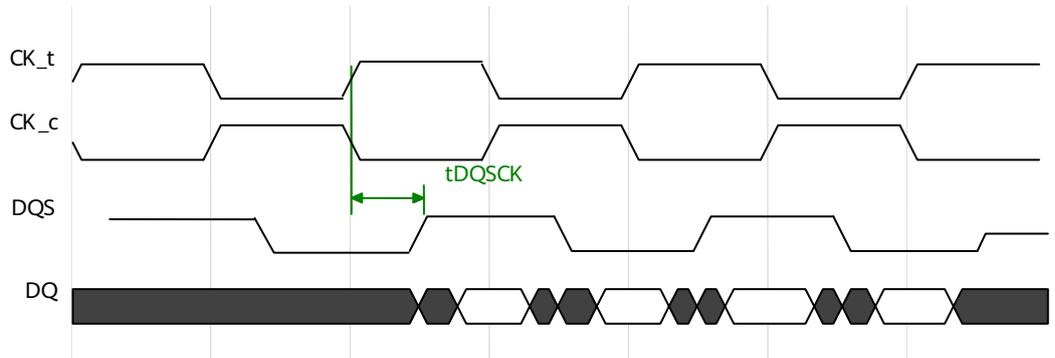
命令和地址相对于 CK 的读操作时序

“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

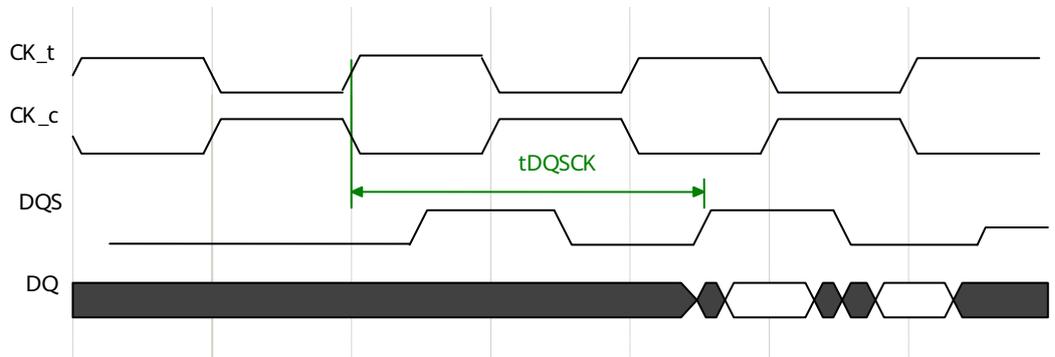
对于 DDR4, CK 相对于 DQS 的读操作时序, t_{DQSCK} 为有效 DQS 相对于 CK 的偏斜。DDR4 CK 相对于 DQS 的读操作时序如图 2-10 所示。

图2-10 DDR4 CK 相对于 DQS 的读操作时序图



对于 LPDDR4/LPDDR4X, t_{DQSCK} 允许的偏斜值更大。LPDDR4/LPDDR4X CK 相对于 DQS 的读操作时序如图 2-11 所示。

图2-11 LPDDR4/LPDDR4X CK 相对于 DQS 的读操作时序图

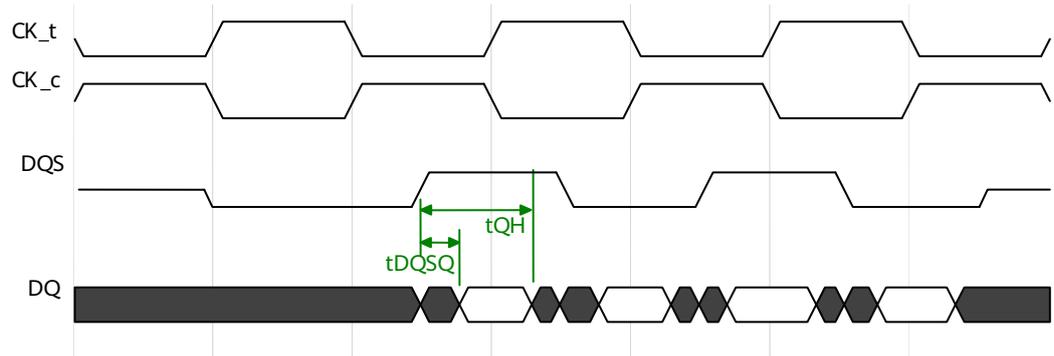




DQS 相对于 DQ 的读操作时序

对于 DDR4/LPDDR4/LPDDR4X 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。DDR4/LPDDR4/LPDDR4X DQS 相对于 DQ 的读操作时序如图 2-12 所示。

图2-12 DDR4/LPDDR4/LPDDR4X 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 标准协议，本文中描述的时序都是 DDR PHY 侧输出的时序。

时钟参数如表 2-46 和表 2-47 所示，DDR4-2666 SDRAM 存储器参数如表 2-48 所示，LPDDR4/LPDDR4X-2666 SDRAM 存储器参数如表 2-49 所示。

表2-46 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	-	1333	-	MHz
锁相环抖动 $t_{JIT(cc)}$	-	-	75	ps
锁相环抖动 $t_{JIT(per)}$	-	-	38	ps
锁相环占空比	45.000		55.000	%



表2-47 LPDDR4/LPDDR4X 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	-	1333	-	MHz
锁相环抖动 tJIT(cc)	-	-	80	ps
锁相环抖动 tJIT(per)	-	-	40	ps
锁相环占空比	43.000		57.000	%

表2-48 DDR4 SDRAM 存储器参数表 (DDR4-2666)

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	-	0.18	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)	55	-	ps
参考 Vref 电平的地址和命令相对 DDR 时钟的建立时间	tIS(Vref)	145	-	ps
tIH(Vref)	tIH(base)	80	-	-
参考 vref 电平地址和命令相对 DDR 时钟的保持时间	tIH(Vref)	145	-	-
DQS 输出时相对 DDR 时钟的偏斜	tDQSCK	-170	170	ps

表2-49 LPDDR4/LPDDR4X SDRAM 存储器参数表 (LPDDR4/LPDDR4X-2666)

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	0.2	-	tCK



参数	符号	最小值	最大值	单位
DQS 下降沿相对 DDR 时钟的保持时间	tDSH	0.2	-	tCK
DQS 与 DQ 的偏斜	tDQSQ	-	0.18	tCK
DQS 输出时相对 DDR 时钟的偏斜	tDQSCK	1500	3500	ps

2.6.2 SPI FLASH 接口时序

SPI FLASH 输入方向时序如图 2-13 和图 2-14 所示。

图2-13 SPI FLASH 输入方向时序图-SDR 模式

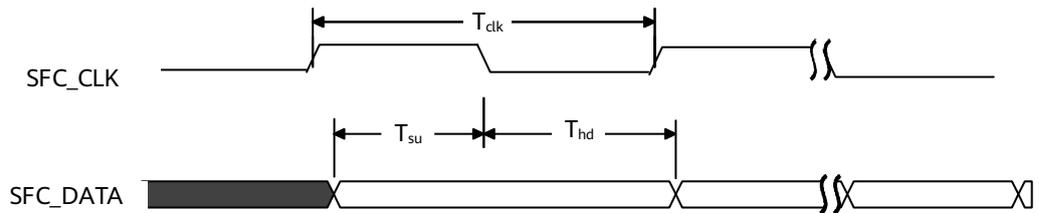
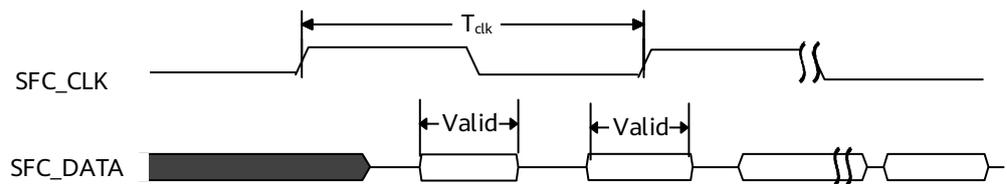


图2-14 SPI FLASH 输入方向时序图-DDR 模式



SPI FLASH 输入方向时序参数如表 2-50 和表 2-51 所示。

表2-50 SPI FLASH 输入方向时序参数表 (电压=3.3V)

参数	符号	最小值	典型值	最大值	单位
SFC_CLK 时钟周期 (SDR)	T_{clk}	10.1	-	83.2	ns
SFC_CLK 时钟周期 (DDR)	T_{clk}	10.1	-	83.2	ns



参数	符号	最小值	典型值	最大值	单位
SFC_CLK 时钟占空比 (DDR)	Duty	45	-	55	%
输入信号建立时间要求(SDR)	T_{su}	1	-	-	ns
输入信号保持时间要求(SDR)	T_{hd}	1	-	-	ns
输入信号有效时间要求(DDR)	Valid	3	-	-	ns

表2-51 SPI FLASH 输入方向时序参数表 (电压=1.8V)

参数	符号	最小值	典型值	最大值	单位
SFC_CLK 时钟周期 (SDR)	T_{clk}	13.33	-	83.2	ns
SFC_CLK 时钟周期 (DDR)	T_{clk}	13.33	-	83.2	ns
SFC_CLK 时钟占空比 (DDR)	Duty	45	-	55	%
输入信号建立时间要求(SDR)	T_{su}	1.9	-	-	ns
输入信号保持时间要求(SDR)	T_{hd}	0	-	-	ns
输入信号有效时间要求(DDR)	Valid	3	-	-	ns

SPI FLASH 输出方向时序如图 2-15 和图 2-16 所示。

图2-15 SPI FLASH 输出方向时序图-SDR 模式

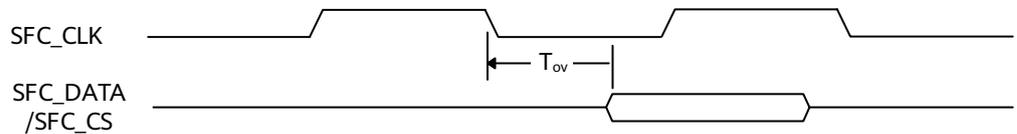
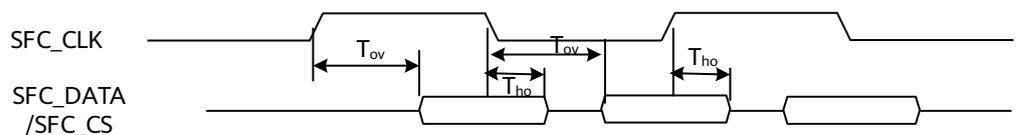


图2-16 SPI FLASH 输出方向时序图-DDR 模式





SPI FLASH 输出方向时序参数如表 2-52 和表 2-53 所示。

表2-52 SPI FLASH 输出方向时序参数表 (电压=3.3V)

参数	符号	最小值	典型值	最大值	单位
SFCCLK 时钟周期 (SDR)	T	10.1	-	83.2	ns
SFCCLK 时钟周期 (DDR)	T	10.1	-	83.2	ns
输出信号延时(SDR)	T_{ov}	-1	-	3	ns
输出信号延时(DDR)	T_{ov}	-	-	T/2-1.5	ns
输出信号保持时间(DDR)	T_{ho}	1.5	-	-	ns

表2-53 SPI FLASH 输出方向时序参数表 (电压=1.8V)

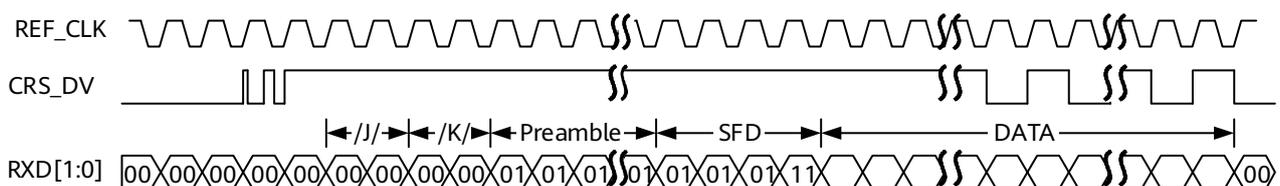
参数	符号	最小值	典型值	最大值	单位
SFCCLK 时钟周期 (SDR)	T	13.33	-	83.2	ns
SFCCLK 时钟周期 (DDR)	T	13.33	-	83.2	ns
输出信号延时(SDR)	T_{ov}	-1	-	3	ns
输出信号延时(DDR)	T_{ov}	-	-	T/2-1.5	ns
输出信号保持时间(DDR)	T_{ho}	1.5	-	-	ns

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 2-17 所示。

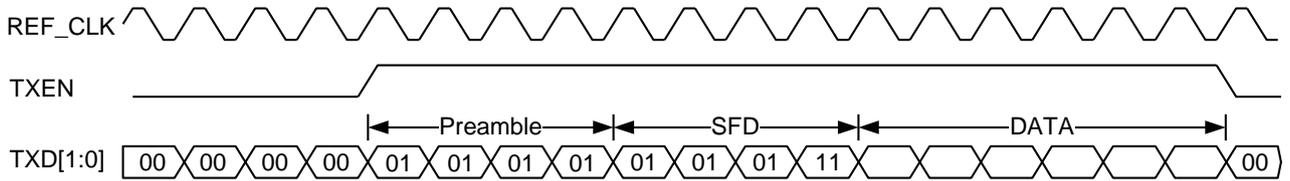
图2-17 RMII 接口 100Mbit/s 接收时序





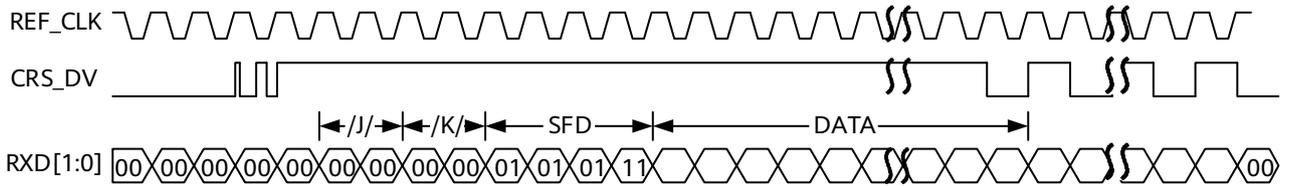
RMII 接口 100Mbit/s 发送时序如图 2-18 所示。

图2-18 RMII 接口 100Mbit/s 发送时序



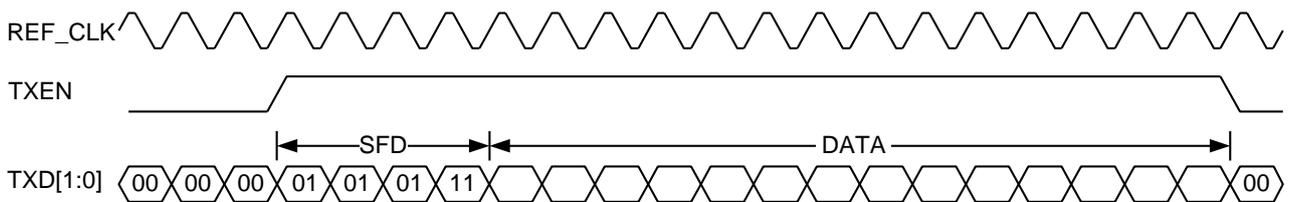
RMII 接口 10Mbit/s 接收时序如图 2-19 所示。

图2-19 RMII 接口 10Mbit/s 接收时序



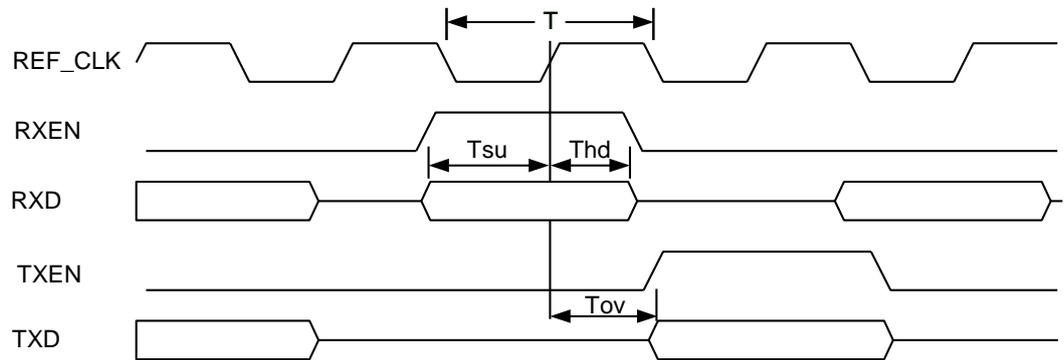
RMII 接口 10Mbit/s 发送时序如图 2-20 所示。

图2-20 RMII 接口 10Mbit/s 发送时序



RMII 接口时序参数如图 2-21 所示。

图2-21 RMII 接口时序参数



RMII 接口时序参数说明如表 2-54 所示。

表2-54 RMII 接口时序参数说明

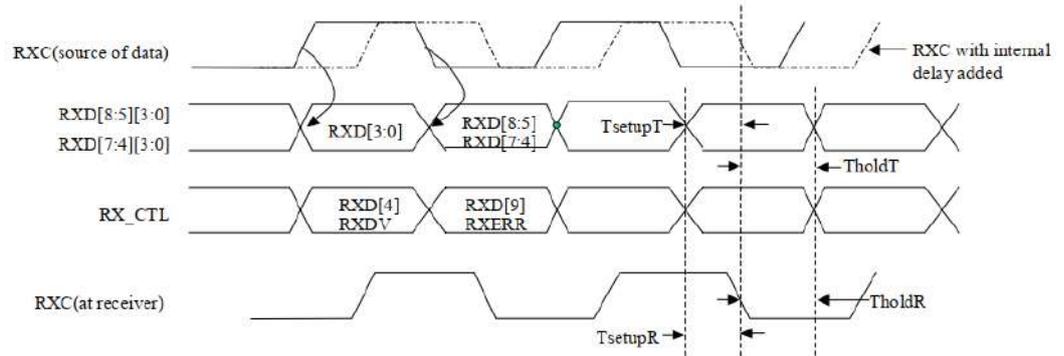
参数	符号	信号	最小值	典型值	最大值	单位
RMII 时钟周期	T	REF_CLK	-	20	-	ns
RMII 信号建立时间	Tsu (RX)	CRS_DV/RX D[1:0]	4	-	-	ns
RMII 信号保持时间	Thd (RX)	CRS_DV/RX D[1:0]	2	-	-	ns
RMII 输出信号延时	Tov (TX)	TXEN/TXD[1 :0]	3	-	16	ns

2.6.3.2 RGMII 接口时序

RGMII 接口 1000Mbit/s 接收时序如图 2-22 所示。

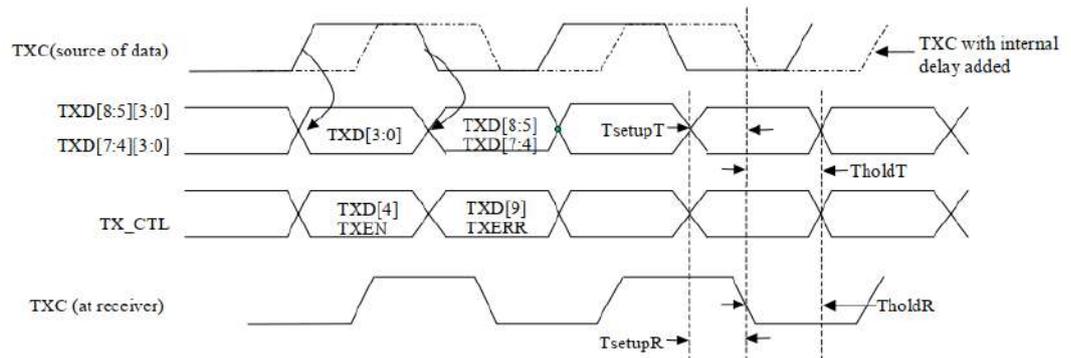


图2-22 RGMII 接口 1000Mbit/s 接收时序



RGMII 接口 1000Mbit/s 发送时序如图 2-23 所示。

图2-23 RGMII 接口 1000Mbit/s 发送时序



RGMII 接口时序参数说明如表 2-55 所示。

表2-55 RGMII 接口时序参数说明

参数	符号	信号	最小值	典型值	最大值	单位
RGMII 时钟周期	T	RXCK、TXCK	-	8	-	ns
RGMII 信号建立时间	Tsu (RX)	RXER、RXDV、RXD[3:0]	1	-	-	ns
RGMII 信号保持时间	Thd (RX)	RXER、RXDV、RXD[3:0]	1	-	-	ns

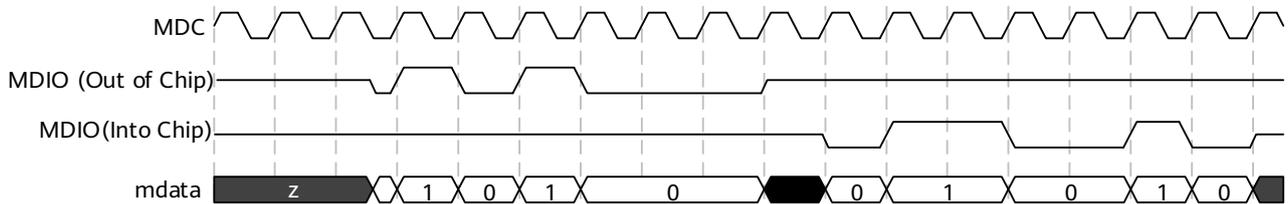


参数	符号	信号	最小值	典型值	最大值	单位
RGMI I 输出信号延时	Tov (TX)	TXD[3:0]、TXEN	-0.5	-	0.5	ns

2.6.3.3 MDIO 接口时序

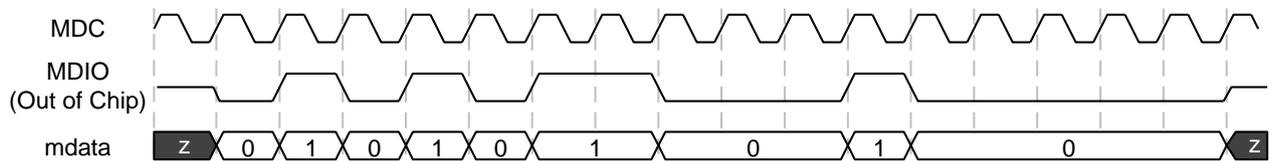
MDIO 接口读时序如图 2-24 所示。

图2-24 MDIO 接口读时序



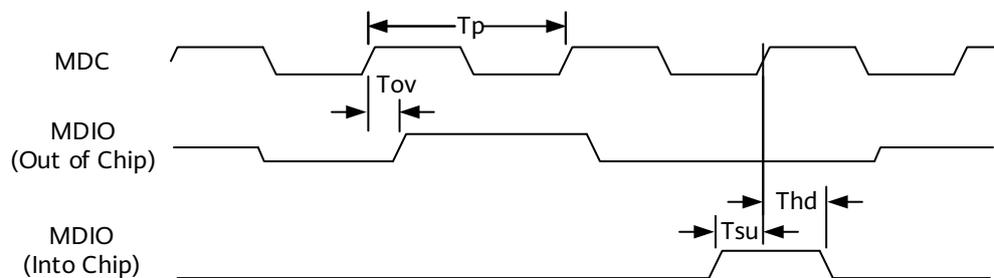
MDIO 接口写时序如图 2-25 所示。

图2-25 MDIO 接口写时序



MDIO 接口时序参数如图 2-26 所示。

图2-26 MDIO 接口接收时序参数



MDIO 接口时序参数说明如表 2-56 所示。



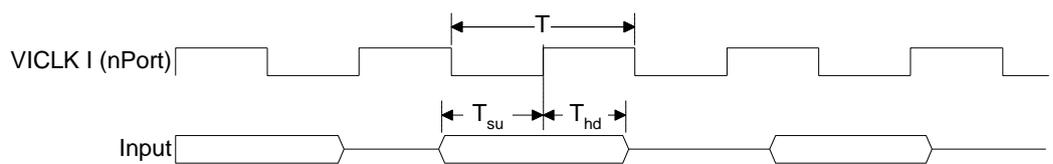
表2-56 MDIO 接口时序参数

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	-	300	ns
MDIO 时钟周期	Tp	MDCK	-	400	ns
MDIO 发送数据建立时间	Tsu	MDIO	10	-	ns
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 2-27 所示。

图2-27 VI 接口时序图



VI 接口时序参数，如表 2-57 所示。

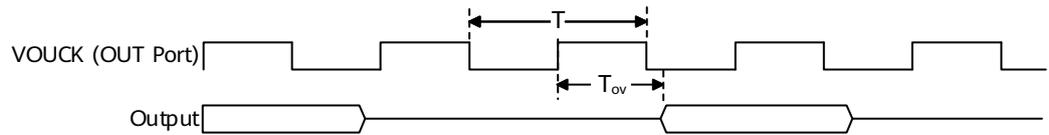
表2-57 VI 接口时序参数表

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}	1.93	-	-	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 2-28 所示。

图2-28 BT.656 接口时序



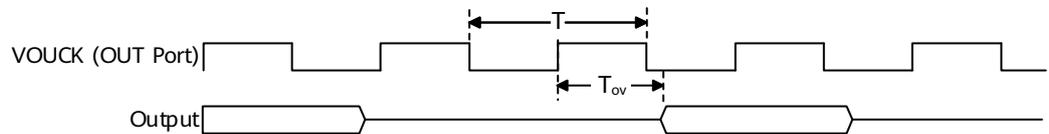
BT.656 接口时序参数如表 2-58 所示。

表2-58 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	13.46	-	-	ns
输出信号延时	T _{ov}	T/2-1.5	-	T/2+1.5	ns

RGB 接口时序如图 2-29 所示。

图2-29 RGB 接口时序



RGB 接口时序参数如表 2-59 所示。

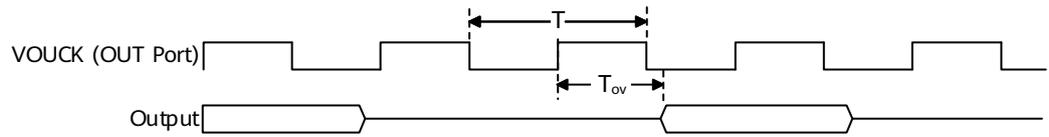
表2-59 RGB 接口时序参数表

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

BT.1120 接口时序如图 2-30 所示。



图2-30 BT.1120 接口时序



BT.1120 接口时序参数如表 2-60 所示。

表2-60 BT.1120 接口时序参数表 (仅支持 3.3V)

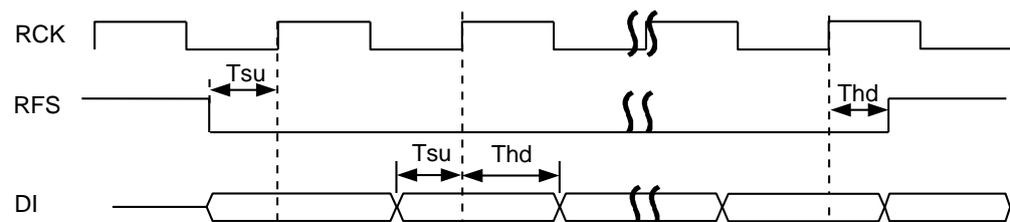
参数	符号	最小值	典型值	最大值	单位
VOUCLK 时钟周期	T	6.75	-	-	ns
输出信号延时	T_{ov}	$T/2-1.7$	-	$T/2+1.7$	ns

2.6.6 AIAO 接口时序

2.6.6.1 I²S 接口时序

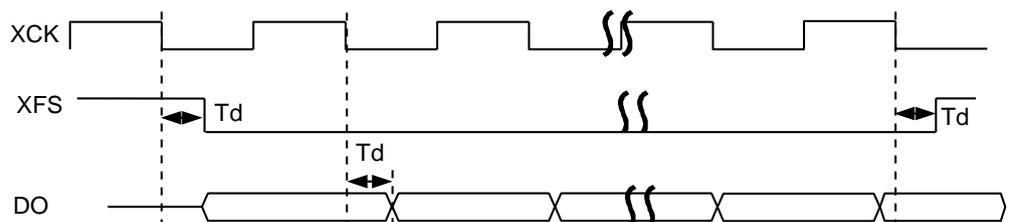
I²S 接口接收时序如图 2-31 所示。

图2-31 I²S 接口接收时序图



I²S 接口发送时序如图 2-32 所示。

图2-32 I²S 接口发送时序图





I²S 接口时序参数如表 2-61 所示。

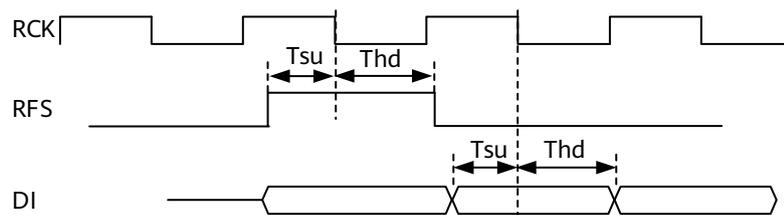
表2-61 I²S 接口时序参数表

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

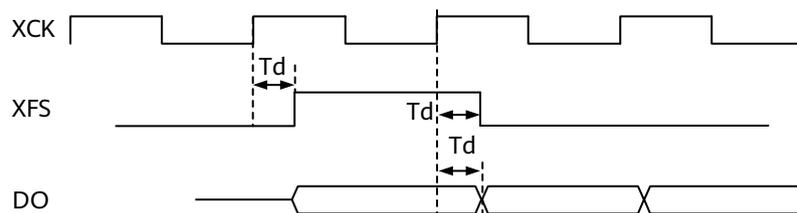
PCM 接口接收时序如图 2-33 所示。

图2-33 PCM 接口接收时序图



PCM 接口发送时序如图 2-34 所示。

图2-34 PCM 接口发送时序图



PCM 接口时序参数如表 2-62 所示。



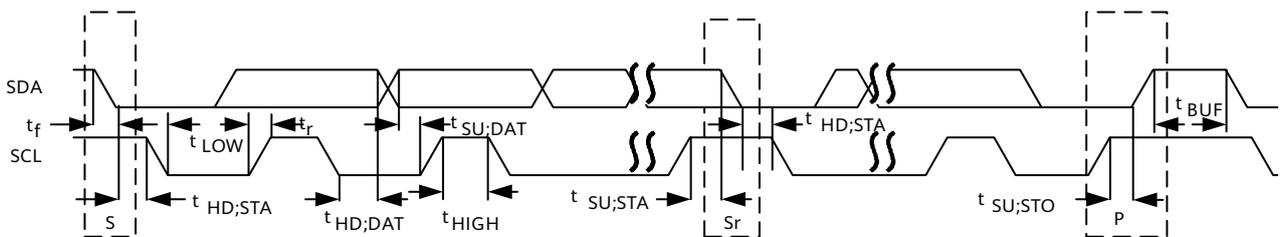
表2-62 PCM 接口时序参数表

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 2-35 所示。

图2-35 I²C 传输时序图



I²C 接口时序参数如表 2-63 所示。

表2-63 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	$t_{HD;STA}$	4.0	-	0.6	-	μ s
SCL 低电平周期	t_{LOW}	4.7	-	1.3	-	μ s
SCL 高电平周期	t_{HIGH}	4.0	-	0.6	-	μ s
启动建立时间	$t_{SU;STA}$	4.7	-	0.6	-	μ s



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	
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

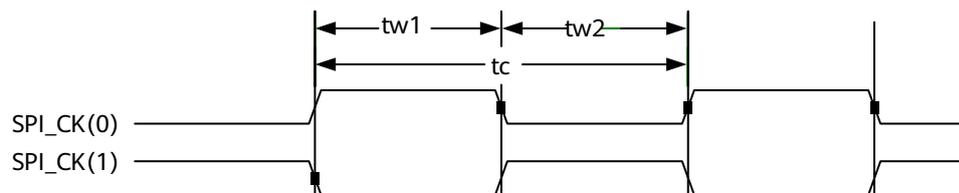
说明

图 2-36 ~ 图 2-38 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 2-36 所示。

图2-36 SPICK 时序



SPI 主模式下接口时序分别如图 2-37 和图 2-38 所示。



图2-37 SPI 主模式下接口时序 (sph=1)

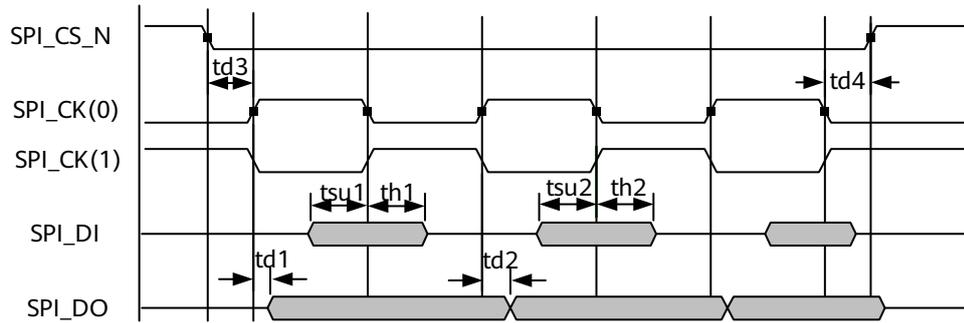
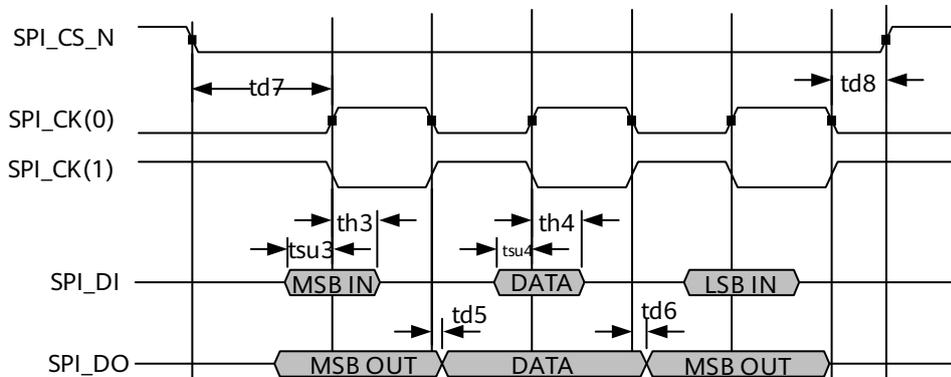


图2-38 SPI 主模式下接口时序 (sph=0)



SPI 接口时序参数如表 2-64 所示。

表2-64 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	spi0:2000 /Fsspclk spi1/2/3: 4000/Fsspclk	-	spi0/1/2/3: 65024000/ Fsspclk	ns	Fsspclk 以 MHz 为单位。 Fsspclk 频率值具体请参考本文档“外围设备”章节
Pulse duration, SPI_CK high (All Master Modes)	tw1	-	1/2 tc	-	ns	



参数	符号	最小值	典型值	最大值	单位	备注
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	spi0:7 spi1:12 spi2:12 spi3:12	-	-	ns	
Setup time, SPI_DI (input) valid before SPI_CK (output) rising edge	tsu2	spi0:7 spi1:12 spi2:12 spi3:12	-	-	ns	
Hold time, SPI_DI (input) valid after SPI_CK (output) falling edge	th1	0	-	-	ns	
Hold time, SPI_DI (input) valid after SPI_CK (output) rising edge	th2	0	-	-	ns	
Delay time, SPI_CK (output) rising edge to SPI_DO (output) transition	td1	-	-	spi0:5 spi1:10 spi2:10 spi3:10	ns	
Delay time, SPI_CK (output) falling edge to SPI_DO (output) transition	td2	-	-	spi0:5 spi1:10 spi2:10 spi3:10	ns	
Delay time, SPI_CS_N (output) falling edge to first SPI_CK (output) rising or falling edge	td3	spi0: (1/2 tc)-5 spi1: (1/2 tc)-10 spi2: (1/2 tc)-10 spi3: (1/2 tc)-10	-	-	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	spi0:tc-5 spi1:tc-10 spi2:tc-10 spi3:tc-10	-	-	ns	
Setup time, SPI_DI (input) valid before SPI_CK (output) rising edge	tsu3	spi0:7 spi1:12 spi2:12 spi3:12	-	-	ns	
Setup time, SPI_DI (input) valid before SPI_CK (output) falling edge	tsu4	spi0:7 spi1:12 spi2:12 spi3:12	-	-	ns	
Hold time, SPI_DI (input) valid after SPI_CK (output) rising edge	th3	0	-	-	ns	
Hold time, SPI_DI (input) valid after SPI_CK (output) falling edge	th4	0	-	-	ns	
Delay time, SPI_CK (output) falling edge to SPI_DO (output) transition	td5	-	-	spi0:5 spi1:10 spi2:10 spi3:10	ns	
Delay time, SPI_CK (output) rising edge to SPI_DO (output) transition	td6	-	-	spi0:5 spi1:10 spi2:10 spi3:10	ns	
Delay time, SPI_CS_N (output) falling edge to first SPI_CK (output) rising or falling edge	td7	spi0:tc-5 spi1:tc-10 spi2:tc-10 spi3:tc-10	-	-	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	td8	spi0: (1/2 tc)-5 spi1: (1/2 tc)-10 spi2: (1/2 tc)-10 spi3: (1/2 tc)-10	-	-	ns	

2.6.9 SPI_TFT 接口时序

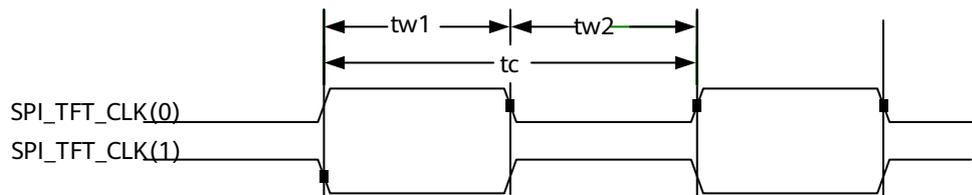
说明

图 2-39 ~ 图 2-43 中，以下缩略语或字母意义不变：

- MSB: Most Significant Bit
- LSB: Least Significant Bit
- SPI_TFT_SCL(0):spo=0
- SPI_TFT_SCL(1):spo=1
- SPI_TFT 只能作为 master 使用。

SPI 接口时钟时序如图 2-39 所示。

图2-39 SPI_TFT_CLK 时序



SPI_TFT 接口时序分别如图 2-40 和图 2-43 所示。



图2-40 SPI_TFT 接口写数据时序 (sph=1)

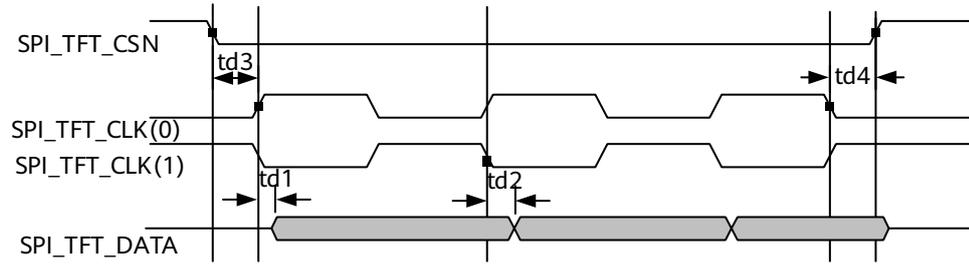


图2-41 SPI_TFT 接口读数据时序 (sph=1)

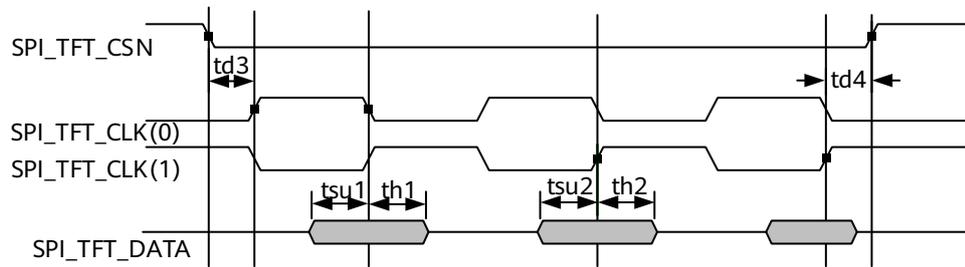


图2-42 SPI_TFT 接口写数据时序 (sph=0)

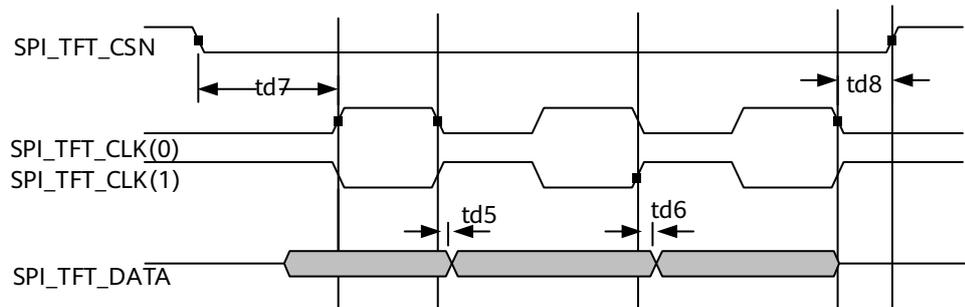
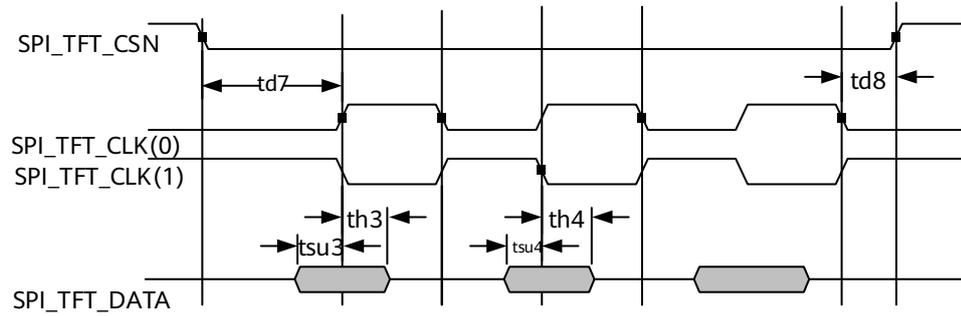




图2-43 SPI_TFT 接口读数据时序 (sph=0)



SPI_TFT 接口时序参数如表 2-65 所示。

表2-65 SPI_TFT 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_TFT_CLK	tc	10/ Fsspclk	-	4000/ Fsspclk	ns	Fsspclk 以 MHz 为单位。 Fsspclk 频率值具体请参考本文档“外围设备”章节
Pulse duration, SPI_TFT_CLK high (All Master Modes)	tw1	-	1/2tc	-	ns	
Pulse duration, SPI_TFT_CLK low (All Master Modes)	tw2	-	1/2tc	-	ns	
Setup time, SPI_TFT_DATA (input) valid before SPI_TFT_CLK(output) falling edge	tsu1	25	-	-	ns	
Setup time, SPI_TFT_DATA (input) valid before SPI_TFT_CLK (output) rising edge	tsu2	25	-	-	ns	



参数	符号	最小值	典型值	最大值	单位	备注
Hold time, SPI_TFT_DATA (input) valid after SPI_CTFT_CLK(output) falling edge	th1	0	-	-	ns	
Hold time, SPI_TFT_DATA (input) valid after SPI_TFT_CLK (output) rising edge	th2	0	-	-	ns	
Delay time, SPI_TFT_CLK(output) rising edge to SPI_TFT_DATA (output) transition	td1	-	-	25	ns	
Delay time, SPI_TFT_CLK (output) falling edge to SPI_TFT_DATA (output) transition	td2	-	-	25	ns	
Delay time, SPI_TFT_CSN (output) falling edge to first SPI_TFT_CLK (output) rising or falling edge	td3	25	-	-	ns	
Delay time, SPI_TFT_CLK (output) rising or falling edge to SPI_TFT_CSN (output) rising edge	td4	15	-	-	ns	
Setup time, SPI_TFT_DATA (input) valid before SPI_TFT_CLK (output) rising edge	tsu3	25	-	-	ns	
Setup time, SPI_TFT_DATA (input) valid before SPI_TFT_CLK (output) falling edge	tsu4	25	-	-	ns	
Hold time, SPI_TFT_DATA (input) valid after SPI_TFT_CLK (output) rising edge	th3	0	-	-	ns	
Hold time, SPI_TFT_DATA (input) valid after SPI_TFT_CLK (output) falling edge	th4	0	-	-	ns	

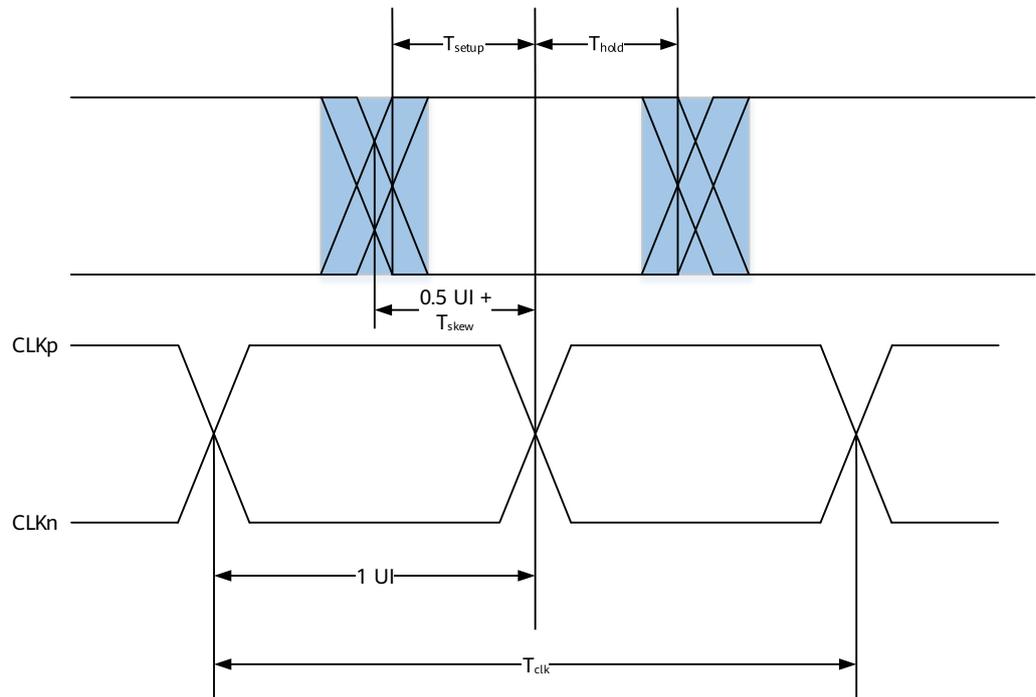


参数	符号	最小值	典型值	最大值	单位	备注
Delay time, SPI_TFT_CLK (output) falling edge to SPI_TFT_DATA (output) transition	td5	-	-	25	ns	
Delay time, SPI_TFT_CLK (output) rising edge to SPI_TFT_DATA (output) transition	td6	-	-	25	ns	
Delay time, SPI_TFT_CSN (output) falling edge to first SPI_TFT_CLK (output) rising or falling edge	td7	25	-	-	ns	
Delay time, SPI_TFT_CLK (output) rising or falling edge to SPI_TFT_CSN (output) rising edge	td8	15	-	-	ns	

2.6.10 MIPI Rx 接口时序

MIPI Rx DPHY/Sub-lvds/LVDS/HiSPi 接口时钟时序如图 2-44 所示。

图2-44 MIPI Rx DPHY/Sub-lvds/LVDS/HiSPi 接口时钟数据时序图



图中的 Tskew 包括 Tdynamic_skew 和 Tstatic_skew

MIPI Rx DPHY/Sub-lvds/LVDS/HiSPi 时序参数如表 2-66 所示。

表2-66 MIPI Rx DPHY/Sub-lvds/LVDS/HiSPi 时序参数表

符号	参数	最小值	典型值	最大值	单位
FMAX	数据率	-	-	2.5G	bps
Tclk	差分时钟周期	0.8	T	-	ns
T _{static_skew}	静态 skew 时间 (1.5Gbps <datarate<=2.5Gbps)	-0.2	-	0.2	UI
T _{dynamic_skew}	动态 skew 时间 (T _{setup} +T _{hold}) (1.5Gbps <datarate<=2.5Gbps)	0.5	-	-	UI



符号	参数	最小值	典型值	最大值	单位
Tsetup	差分时钟建立时间	0.15 (80Mbps <datarate<=1.0Gbps)	-	-	UI
		0.2 (1.0Gbps <datarate<=1.5Gbps)			
Thold	差分时钟保持时间	0.15 (80Mbps <datarate<=1.0Gbps)	-	-	UI
		0.2 (1.0Gbps <datarate<=1.5Gbps)			

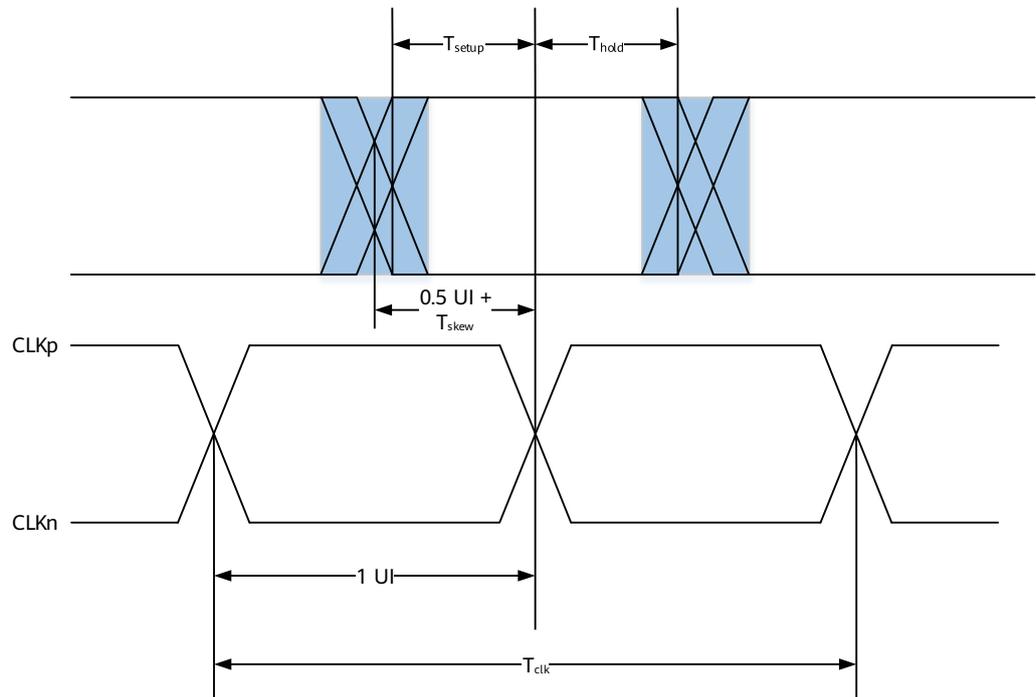
说明

UI is equal to T/2.

2.6.11 MIPI Tx 接口时序

MIPI Tx 接口时钟时序如图 2-45 所示。

图2-45 MIPI Tx 接口时钟数据时序图



图中的 T_{skew} 包括 $T_{dynamic_skew}$ 和 T_{static_skew}

MIPI Tx DATA-CLOCK 时序参数如表 2-67 所示。

表2-67 MIPI Tx DATA-CLOCK 时序参数表

符号	参数	最小值	最大值	Datarate	单位
Tskew	TX skew	-0.15	0.15	80Mbps-1Gbps	UI
		-0.2	0.2	1Gbps-1.5Gbps	UI
Tstatic_skew	TX 静态 skew 时间	-0.2	0.2	>1.5Gbps	UI
Tdynamic_skew	TX 动态 skew 时间	-0.15	0.15	>1.5Gbps	UI
Tsetup	差分时钟建立时间	0.35	-	80Mbps-1Gbps	UI
		0.3	-	1Gbps-1.5Gbps	UI
		0.15	-	>1.5Gbps	UI



符号	参数	最小值	最大值	Datarate	单位
Thold	差分时钟保持时间	0.35	-	80Mbps-1Gbps	UI
		0.3	-	1Gbps-1.5Gbps	UI
		0.15	-	>1.5Gbps	UI

说明

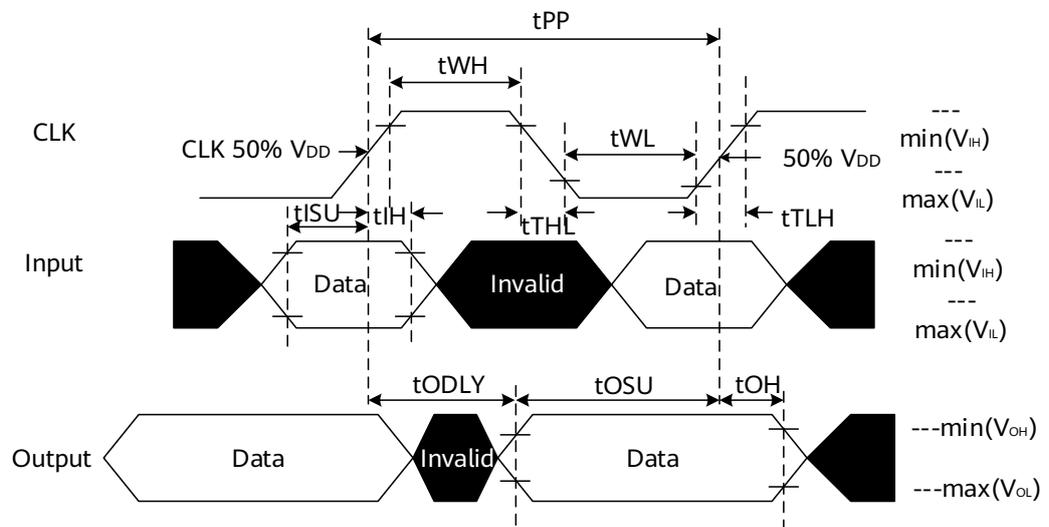
1. 以上值均为对接 100Ω差分端的参数。
2. UI is equal to T/2.

2.6.12 SDIO/MMC 接口时序

eMMC 接口时序

eMMC DS/HS 模式 CMD/DATA 输入/输出方向时序如图 2-46 所示。

图2-46 eMMC DS/HS 模式 CMD/DATA 输入/输出方向时序图



eMMC DS 模式 CMD/DATA 输入/输出方向时序参数如表 2-68 所示。



表2-68 eMMC DS 模式 CMD/DATA 输入/输出时序参数表

符号	参数	最小值	典型值	最大值	单位	备注
Clock CLK						
t_{pp}	Cycle time data transfer mode	38.5	40.4	-	ns	-
t_{WH}	Clock high time	10	18	-	ns	-
t_{WL}	Clock low time	10	18	-	ns	-
Outputs DAT (referenced to CLK)						
t_{OSU}	Output set-up time	5	18	-	ns	-
t_{OH}	Output hold time	5	18	-	ns	-
Inputs DAT (referenced to CLK)						
t_{ISU}	Input set-up time	11.7	-	-	ns	-
t_{IH}	Input hold time	8.3	-	-	ns	-

eMMC HS 模式 CMD/DATA 输入/输出方向时序参数如表 2-69 所示。

表2-69 eMMC HS 模式 CMD/DATA 输入/输出时序参数表

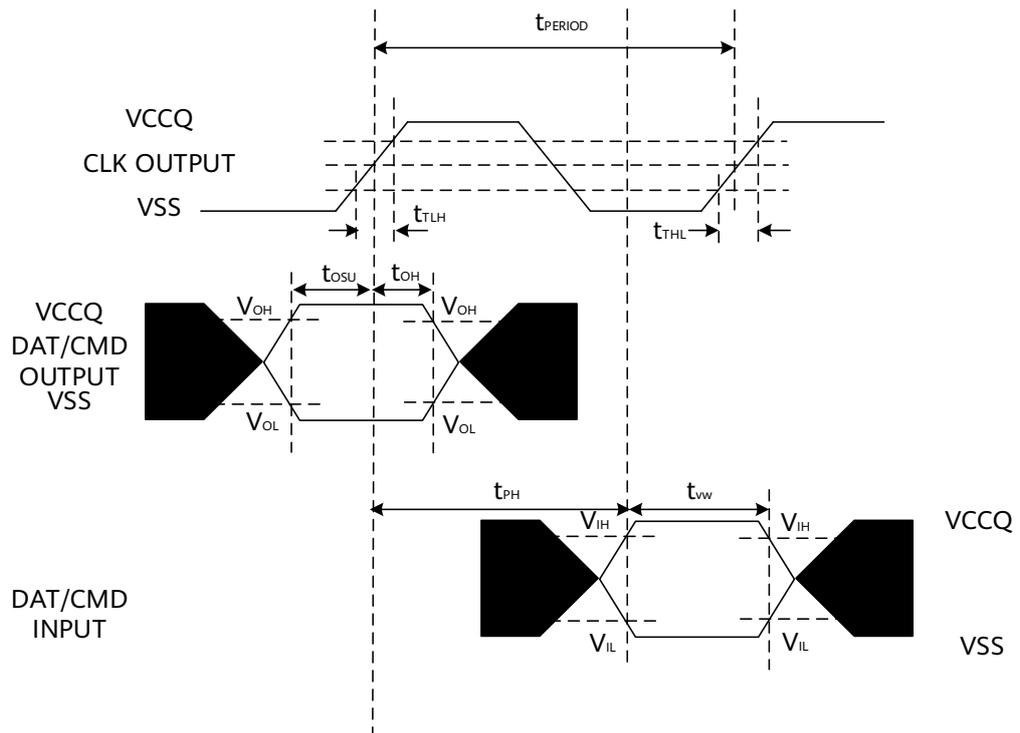
符号	参数	最小值	典型值	最大值	单位	备注
Clock CLK						
t_{pp}	Cycle time data transfer mode	19.2	20.2	-	ns	-
t_{WH}	Clock high time	6.5	9	-	ns	-
t_{WL}	Clock low time	6.5	8	-	ns	-
Outputs DAT (referenced to CLK)						
t_{OSU}	Output set-up time	5	7.2	-	ns	-
t_{OH}	Output hold time	5	9	-	ns	-
Inputs CMD, DAT (referenced to CLK)						
t_{IDLY}	Input delay time during data transfer	-	-	13.7	ns	



符号	参数	最小值	典型值	最大值	单位	备注
tIH	Input hold time	2.5	-	-	ns	

eMMC HS200 模式 CMD/DATA 输入输出方向时序如图 2-47 所示。

图2-47 HS200 模式的 CMD/DATA 输入/输出方向时序图



eMMC HS200 模式 CMD/DATA 输入/输出方向时序参数如表 2-70 所示。

表2-70 HS200 模式 CMD/DATA 输入/输出时钟时序参数表

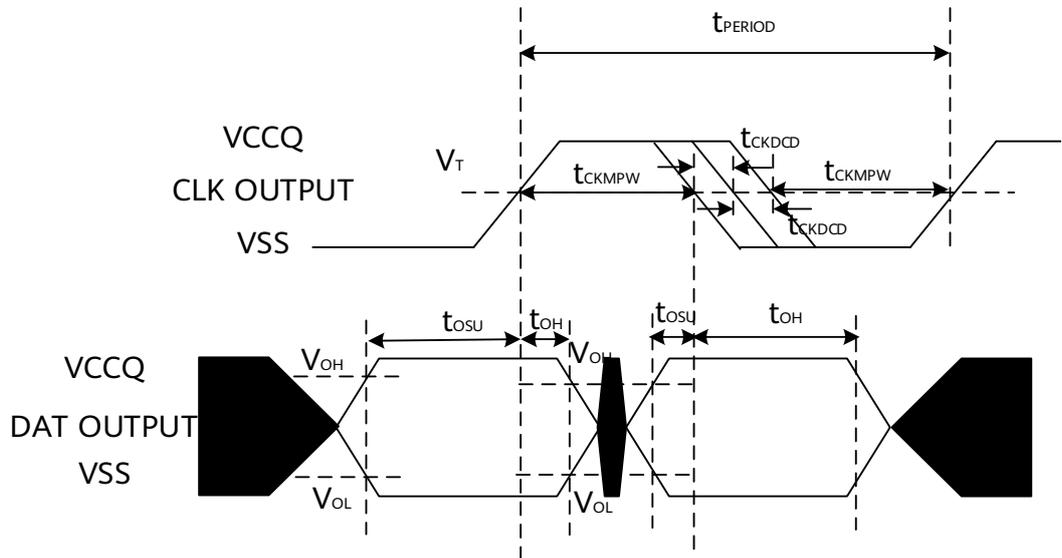
符号	参数	最小值	典型值	最大值	单位	备注
Clock CLK						
t_{PERIOD}	Cycle time data transfer mode	6.73	6.73	-	ns	-
Duty Cycle	Clock duty cycle	30	50	70	%	-
Outputs DAT (referenced to CLK)						



符号	参数	最小值	典型值	最大值	单位	备注
t_{OSU}	Output set-up time	TBD	TBD	-	ns	
t_{OH}	Output hold time	TBD	TBD	-	ns	
Inputs CMD, DAT (referenced to CLK)						
t_{PH}	器件输出 CMD/DA 与接收 CLK 相位差	0	-	2	UI	UI 是单位标称时间, 例如 148.5MHz 时 UI=6.73ns。
t_{VW}	Input valid data window	0.575	-	-	UI	148.5MHz 时 t_{VW} = 3.87ns

eMMC HS400 模式 DATA 输出方向时序如图 2-48 所示。

图2-48 HS400 模式 DATA 输出方向时序图



eMMC HS400 模式 DATA 输出方向时序参数如表 2-71 所示。

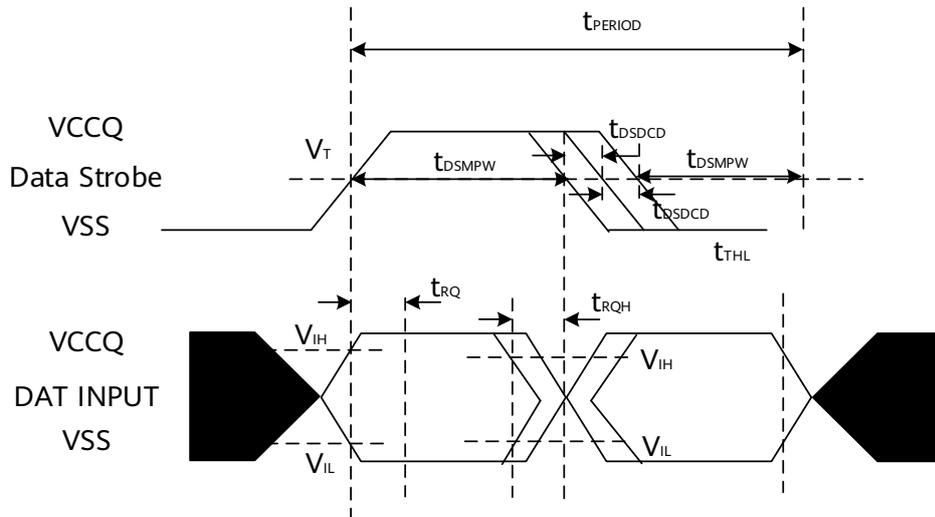


表2-71 HS400 模式 DATA 输出方向时序参数表

符号	参数	最小值	典型值	最大值	单位	备注
Clock CLK						
t_{PERIOD}	Cycle time data transfer mode	6.73	6.73	-	ns	-
t_{CKMPW}	Minimum pulse width	2.2	2.5	-	ns	-
Output DAT (referenced to CLK)						
t_{OSUddr}	Output set-up time	TBD	TBD	-	ns	
t_{OHddr}	Output hold time	TBD	TBD	-	ns	

eMMC HS400 模式 DATA 输入方向时序如图 2-49 所示。

图2-49 HS400 模式 DATA 输入方向时序图



eMMC HS400 模式 DATA 输入方向时序参数如表 2-72 所示。

表2-72 HS400 模式 DATA 输入方向时序参数表

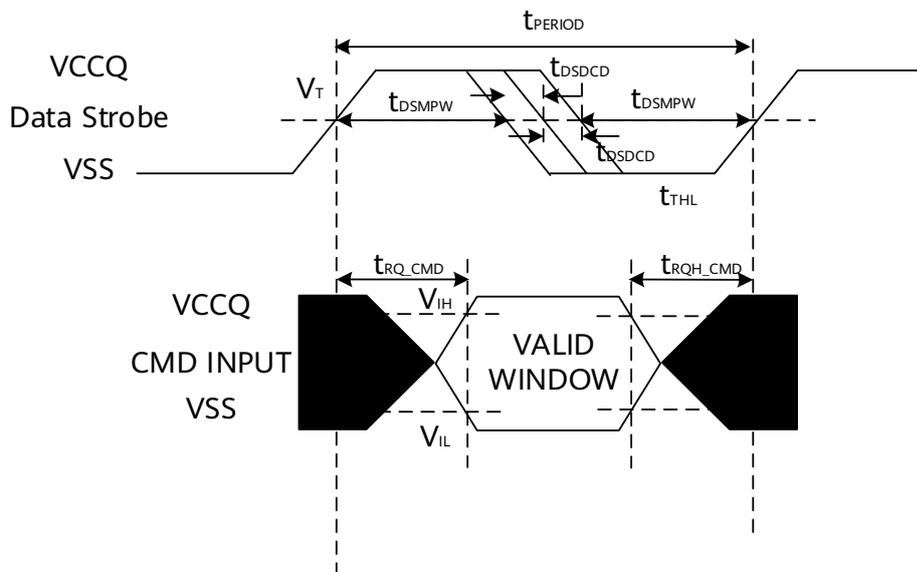
符号	参数	最小值	典型值	最大值	单位
Data Strobe					
t_{PERIOD}	Cycle time data transfer mode	6.73	-	-	-



符号	参数	最小值	典型值	最大值	单位
t_{DSDCD}	Duty cycle distortion	0.0	-	0.2	ns
t_{DSMPW}	Minimum pulse width	2.0	-	-	ns
Input DAT (referenced to Data Strobe)					
t_{RQ}	Input skew	-	-	0.4	ns
t_{RQH}	Input hold skew	-	-	0.4	ns

eMMC HS400 模式 CMD 输入/输出方向时序同 HS200 模式, HS400 (Enhanced data strobe mode) 模式 CMD 输入方向时序如图 2-50 所示。

图2-50 HS400 (Enhanced data strobe mode) 模式 CMD 输入方向时序图



eMMC HS400 (Enhanced data strobe mode) 模式 CMD 输入方向时序参数如表 2-73 所示。

表2-73 HS400 (Enhanced data strobe mode) 模式 CMD 输入时序参数表

符号	参数	最小值	典型值	最大值	单位
Data Strobe					
t_{PERIOD}	Cycle time data transfer mode	6.73	-	-	-



符号	参数	最小值	典型值	最大值	单位
t_{DSDCD}	Duty cycle distortion	0.0	-	0.2	ns
t_{DSMPW}	Minimum pulse width	2.0	-	-	ns
Input CMD (referenced to Data Strobe)					
t_{RQ_CMD}	Input skew(CMD)	-	-	0.4	ns
t_{RQH_CMD}	Input hold skew(CMD)	-	-	0.4	ns

SDIO 接口时序

SDIO DS 模式 CMD/DATA 输出方向时序如图 2-51 所示，DS 模式 CMD/DATA 输入方向时序如图 2-52 所示，HS/SDR12/SDR25/SDR50/SDR104 模式 CMD/DATA 输出方向时序如图 2-53 所示，HS/SDR12/SDR25/SDR50/SDR104 模式 CMD/DATA 输入方向时序如图 2-54 所示。

图2-51 SDIO DS 模式 CMD/DATA 输出方向时序图

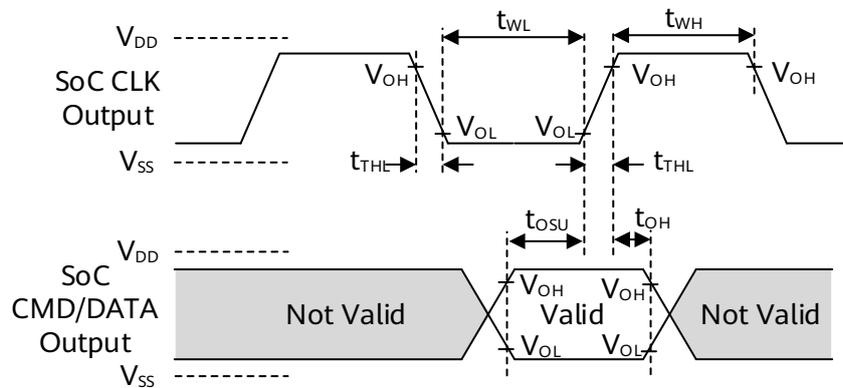




图2-52 SDIO DS 模式 CMD/DATA 输入方向时序图

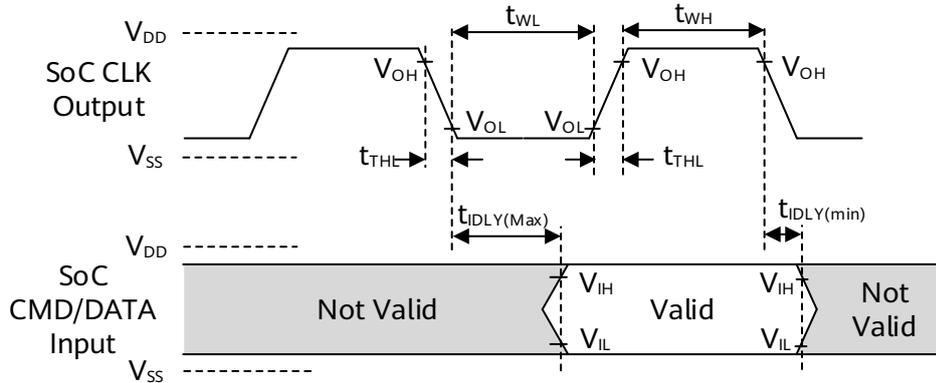


图2-53 SDIO HS/SDR12/SDR25/SDR50/SDR104 模式 CMD/DATA 输出方向时序图

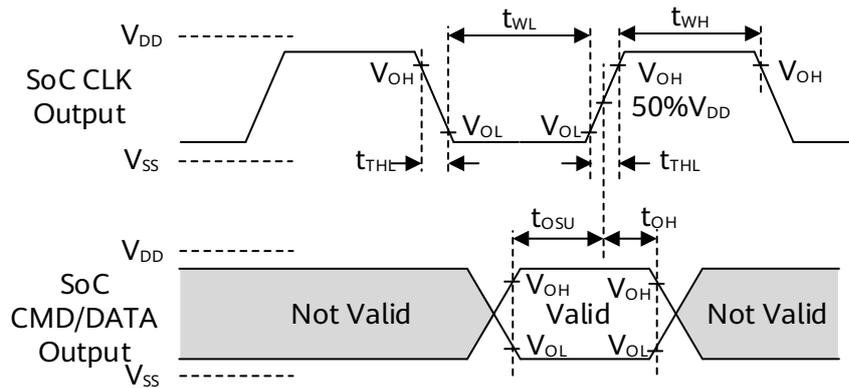
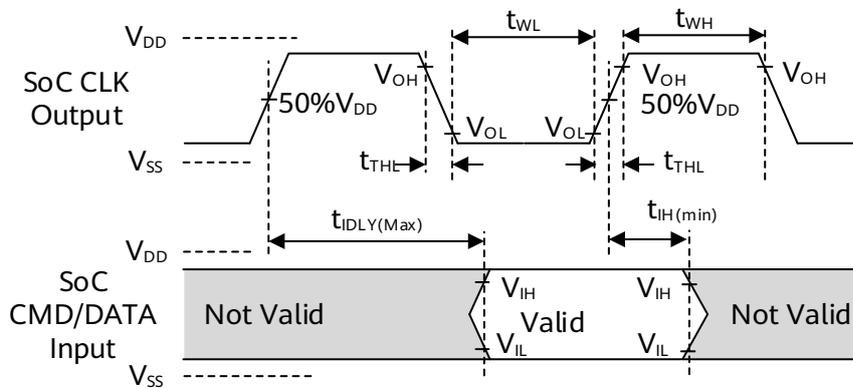


图2-54 SDIO HS/SDR12/SDR25/SDR50/SDR104 模式 CMD/DATA 输入方向时序图





SDIO DS 模式 CMD/DATA 输入/输出时序参数如表 2-74 所示。

表2-74 SDIO DS 模式输入/输出时序参数表

符号	参数	最小值	典型值	最大值	单位	备注
Clock CLK						
t_{CLK}	Cycle time data transfer mode	40	40.4	-	ns	-
t_{WH}	Clock high time	10	19	-	ns	-
t_{WL}	Clock low time	10	19	-	ns	-
Outputs CMD/DAT (referenced to CLK)						
t_{OSU}	Output set-up time	5	18	-	ns	-
t_{OH}	Output hold time	5	18	-	ns	-
Inputs CMD/DAT (referenced to CLK)						
t_{IDLY}	Input Delay time during Data Transfer Mode	0	-	14	ns	-
t_{IDLY}	Input Delay time during Identification Mode	0	-	50	ns	-

SDIO HS 模式 CMD/DATA 输入/输出方向时序参数如表 2-75 所示。

表2-75 SDIO HS 模式 CMD/DATA 输入/输出时钟时序参数表

符号	参数	最小值	典型值	最大值	单位	备注
Clock CLK						
t_{CLK}	Cycle time data transfer mode	20	20.2	-	ns	-
t_{WH}	Clock high time	7	7.89	-	ns	-
t_{WL}	Clock low time	7	7.89	-	ns	-
Outputs CMD/DAT (referenced to CLK)						
t_{OSU}	Output set-up time	6	8	-	ns	-
t_{OH}	Output hold time	2	9	-	ns	-



符号	参数	最小值	典型值	最大值	单位	备注
Inputs CMD/DAT (referenced to CLK)						
t_{IDLY}	Input Delay time during Data Transfer Mode	-	-	14	ns	-
t_{IH}	Input Hold time	2.5	-	-	ns	-

表2-76 SDIO SDR12 模式 CMD/DATA 输入/输出时钟时序参数表

符号	参数	最小值	典型值	最大值	单位	备注
Clock CLK						
t_{CLK}	Cycle time data transfer mode	40	40.4	-	ns	-
Duty Cycle	Clock duty cycle	30	50	70	%	-
Outputs CMD/DAT (referenced to CLK)						
t_{OSU}	Output set-up time	5	18	-	ns	-
t_{OH}	Output hold time	5	18	-	ns	-
Inputs CMD/DAT (referenced to CLK)						
t_{IDLY}	Input Delay time during Data Transfer Mode	0	-	14	ns	-
t_{IH}	Input Hold time	1.5	-	-	ns	-

表2-77 SDIO SDR25 模式 CMD/DATA 输入/输出时钟时序参数表

符号	参数	最小值	典型值	最大值	单位	备注
Clock CLK						
t_{CLK}	Cycle time data transfer mode	20	20.2	-	ns	-



符号	参数	最小值	典型值	最大值	单位	备注
Duty Cycle	Clock duty cycle	30	50	70	%	-
Outputs CMD/DAT (referenced to CLK)						
t_{OSU}	Output set-up time	6	8	-	ns	-
t_{OH}	Output hold time	2	9	-	ns	-
Inputs CMD/DAT (referenced to CLK)						
t_{IDLY}	Input Delay time during Data Transfer Mode	-	-	14	ns	-
t_{IH}	Input Hold time	2.5	-	-	ns	-

表2-78 SDIO SDR50 模式 CMD/DATA 输入/输出时钟时序参数表

符号	参数	最小值	典型值	最大值	单位	备注
Clock CLK						
t_{CLK}	Cycle time data transfer mode	10	10.1	-	ns	-
Duty Cycle	Clock duty cycle	30	50	70	%	-
Outputs CMD/DAT (referenced to CLK)						
t_{OSU}	Output set-up time	3.0	4.0	-	ns	-
t_{OH}	Output hold time	1.0	4.5	-	ns	-
Inputs CMD/DAT (referenced to CLK)						
t_{IDLY}	Input Delay time during Data Transfer Mode	-	-	14	ns	-
t_{IH}	Input Hold time	1.5	-	-	ns	-



表2-79 SDIO SDR104 模式 CMD/DATA 输入/输出时钟时序参数表

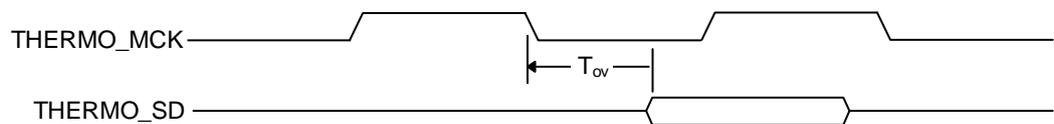
符号	参数	最小值	典型值	最大值	单位	备注
Clock CLK						
t_{CLK}	Cycle time data transfer mode	6.73	6.73	-	ns	-
Duty Cycle	Clock duty cycle	30	50	70	%	-
Outputs CMD/DAT (referenced to CLK)						
t_{OSU}	Output set-up time	1.4	1.6	-	ns	-
t_{OH}	Output hold time	0.8	1.6	-	ns	-
Inputs CMD/DAT (referenced to CLK)						
t_{IDLY}	Input Delay time during Data Transfer Mode	-	-	7.5	ns	-
t_{IH}	Input Hold time	1.5	-	-	ns	-

2.6.13 热成像接口时序

T0 模式

T0 模式发送数据时：芯片输出数据和时钟，发送方向时序如图 2-55 所示。

图2-55 T0 模式发送方向时序图



T0 模式发送方向时序参数，如表 2-80 所示。

表2-80 T0 模式发送方向时序参数表

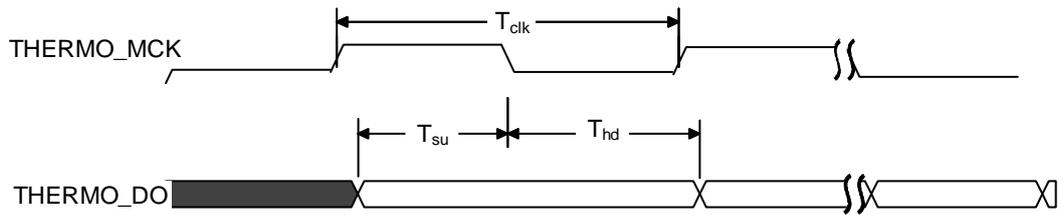
参数	符号	最小值	典型值	最大值	单位
时钟周期	T	22.2	-	-	ns



参数	符号	最小值	典型值	最大值	单位
输出信号延时	T_{ov}	6	-	15	ns

T0 模式接收数据时：对接器件用芯片发送的时钟发送数据，接收方向时序如图 2-56 所示。

图2-56 T0 模式接收方向时序图



接收方向时序参数，如表 2-81 所示。

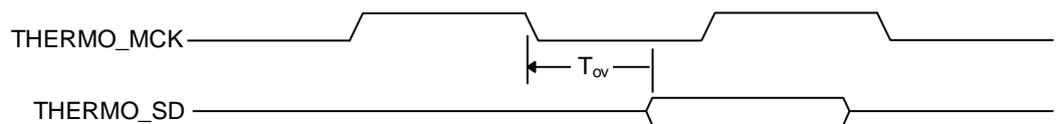
表2-81 T0 模式接收方向时序参数表

参数	符号	最小值	典型值	最大值	单位
时钟周期	T	22.2	-	-	ns
输入信号建立时间要求	T_{su}	15	-	-	ns
输入信号保持时间要求	T_{hd}	10	-	-	ns

T1/T2 模式

T1/T2 模式发送数据时：芯片输出数据和时钟，发送方向时序如图 2-57 所示。

图2-57 T1/T2 模式发送方向时序图



T1/T2 模式发送方向时序参数，如表 2-82 所示。

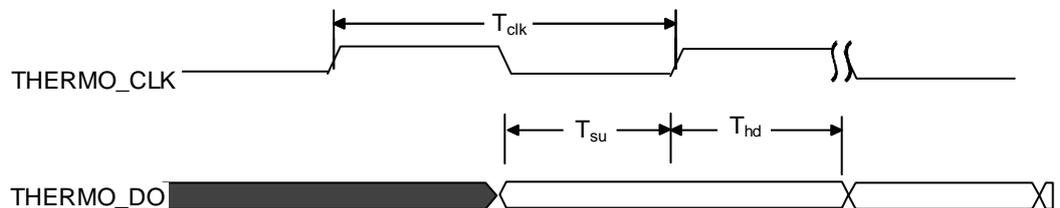


表2-82 T1/T2 模式发送方向时序参数表

参数	符号	最小值	典型值	最大值	单位
时钟周期	T	22.2	-	-	ns
输出信号延时	T _{ov}	2	-	3	ns

T1/T2 模式接收数据时：芯片同时接收对接器件的时钟和数据，接收方向时序如图 2-58 所示。

图2-58 T1/T2 模式接收方向时序图



T1/T2 模式接收方向时序参数，如表 2-83 所示。

表2-83 T1/T2 模式接收方向时序参数表

参数	符号	最小值	典型值	最大值	单位
时钟周期	T	13.5	-	-	ns
输入信号建立时间要求	T _{su}	4	-	-	ns
输入信号保持时间要求	T _{hd}	2	-	-	ns



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	3-3
3.2 时钟	3-4
3.2.1 概述	3-4
3.2.2 功能框图	3-4
3.2.3 时钟资源分布	3-5
3.2.4 PLL 配置	3-10
3.2.5 CRG 寄存器概览	3-12
3.2.6 CRG 寄存器描述	3-16
3.3 处理器子系统	3-102
3.4 中断系统	3-102
3.5 系统控制器	3-105
3.5.1 概述	3-105
3.5.2 特点	3-105
3.5.3 功能描述	3-106
3.5.4 系统控制器寄存器	3-106
3.5.5 SOC MISC 寄存器	3-123
3.5.6 VI_MIPI 及 SPI0/1 寄存器	3-153
3.6 DMA 控制器	3-156
3.6.1 概述	3-156
3.6.2 特点	3-156



3.6.3 功能描述.....	3-156
3.6.4 工作方式.....	3-164
3.6.5 DMAC 寄存器概览.....	3-171
3.6.6 DMAC 寄存器描述.....	3-173
3.7 CRC 循环冗余校验.....	3-192
3.7.1 概述.....	3-192
3.7.2 特点.....	3-192
3.7.3 功能描述.....	3-193
3.7.4 工作方式.....	3-193
3.7.5 CRC 寄存器概览.....	3-194
3.7.6 CRC 寄存器描述.....	3-195
3.8 定时器.....	3-199
3.8.1 概述.....	3-199
3.8.2 特点.....	3-199
3.8.3 功能描述.....	3-199
3.8.4 工作方式.....	3-200
3.8.5 TIMER 寄存器概览.....	3-201
3.8.6 Timer 寄存器描述.....	3-202
3.9 WatchDog.....	3-206
3.9.1 概述.....	3-206
3.9.2 特点.....	3-206
3.9.3 功能描述.....	3-206
3.9.4 工作方式.....	3-208
3.9.5 WDG 寄存器概览.....	3-209
3.9.6 WDG 寄存器描述.....	3-210
3.10 实时时钟.....	3-212
3.10.1 概述.....	3-212
3.10.2 特点.....	3-213
3.10.3 功能描述.....	3-213
3.10.4 工作方式.....	3-214
3.10.5 RTC 寄存器概览.....	3-216



3.10.6 RTC 内部寄存器描述.....	3-218
3.11 电源管理与低功耗模式控制.....	3-234
3.11.1 概述.....	3-234
3.11.2 PMC.....	3-235
3.11.3 时钟门控和时钟频率调整.....	3-251
3.11.4 模块级低功耗控制.....	3-252
3.11.5 DDR 低功耗控制.....	3-252
3.11.6 SVB 功能描述.....	3-252
3.11.7 SVB_PWM 寄存器概览.....	3-254
3.11.8 SVB_PWM 寄存器描述.....	3-255
3.11.9 Tsensor_CTRL 内部寄存器概览.....	3-257
3.11.10 Tsensor_CTRL 寄存器描述.....	3-258



插图目录

图 3-1 复位信号控制图.....	3-2
图 3-2 内部复位和外部复位示意图.....	3-3
图 3-3 时钟管理模块功能框图.....	3-5
图 3-4 时钟资源分布框图.....	3-6
图 3-5 SOC MISC 寄存器（基址是 0x0_1102_4000）	3-123
图 3-6 链表在 DDR 内存放的格式示意图.....	3-162
图 3-7 WatchDog 应用框图.....	3-207
图 3-8 使用 PMC 时的电源网络粗略框图	3-236
图 3-9 不使用 PMC 功能但使用 PMC 管脚.....	3-237
图 3-10 不使用 PMC 功能且不使用管脚时外围管脚的处理	3-238
图 3-11 PWR_BUTTON、PWR_STARTUP 第一次上电时序.....	3-239
图 3-12 PWR_BUTTON、PWR_STARTUP 上电时序.....	3-240
图 3-13 PWR_WAKEUP 上电时序（以上升沿触发为例）	3-241
图 3-14 RTC 定时中断上电时序.....	3-241
图 3-15 软件配置下电时序(以下电模式 1 为例)	3-242
图 3-16 异常下电时序(以下电顺序模式 1 为例)	3-243



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.....	3-6
表 3-3 PLL 对应的配置寄存器	3-10
表 3-4 PLL 频率计算方法.....	3-10
表 3-5 CRG 寄存器概览 (基址: 0x0_1101_0000)	3-12
表 3-6 中断源分配表	3-102
表 3-1 系统控制器(SYS CTRL)寄存器概览 (基址: 0x0_1102_0000)	3-106
表 3-2 VI_MIPI 及 SPI0/1 寄存器概览 (基址是 0x0_17A4_0000)	3-153
表 3-3 DMAC 外设 Single 和 Burst 硬件请求线编号说明.....	3-156
表 3-4 DMAC 外设接收 Last Single 硬件请求线编号说明.....	3-158
表 3-5 DMAC 访问空间说明	3-159
表 3-6 各模块的寄存器偏移地址变量表	3-171
表 3-7 DMAC 寄存器概览(基址: 0x0_1028_0000)	3-171
表 3-8 CRC 多项式规格	3-192
表 3-9 CRC 校验码输入要求.....	3-194
表 3-10 CRC 寄存器概览 (基址是 0x0_1100_8000)	3-194
表 3-11 Timer 寄存器概览	3-202
表 3-12 WatchDog 寄存器概览 (基址是 0x0_1103_0000)	3-209
表 3-13 RTC 内部寄存器概览 (基址是 0x0_1111_0000)	3-216
表 3-14 PMC 寄存器概览 (基址是 0x0_1112_0000)	3-244



表 3-15 寄存器偏移地址变量表	3-245
表 3-16 SVB_PWM 寄存器概览 (基址是 0x0_1102_9000)	3-254
表 3-17 Tsensor_CTRL 寄存器概览 (基址是 0x0_1102_A000)	3-257



3 系统

3.1 复位

3.1.1 概述

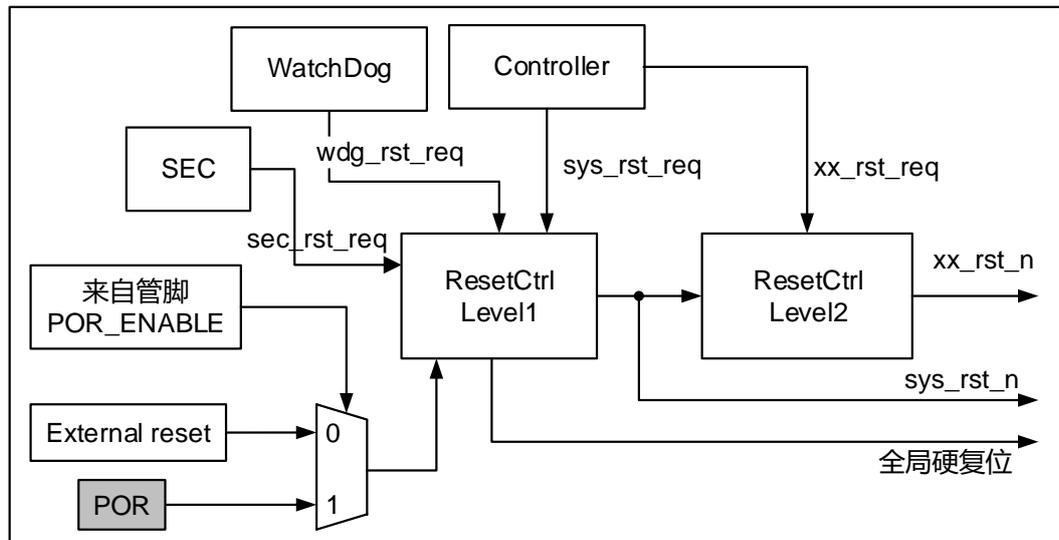
复位管理模块对整个芯片的复位、各功能模块的复位进行统一的管理，包括：

- 上电复位的管理和控制
- 外部复位的管理和控制
- WatchDog 复位管理和控制
- 安全系统复位管理和控制
- 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。
External reset	来自管脚的外部复位信号。
POR_ENABLE	来自管脚, 选择 POR 复位生效还是 External reset 生效。
wdg_rst_req	WatchDog 复位请求。
sec_rst_req	安全系统复位请求。
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、全局硬复位	复位信号。

复位信号分类如表 3-1 所示。

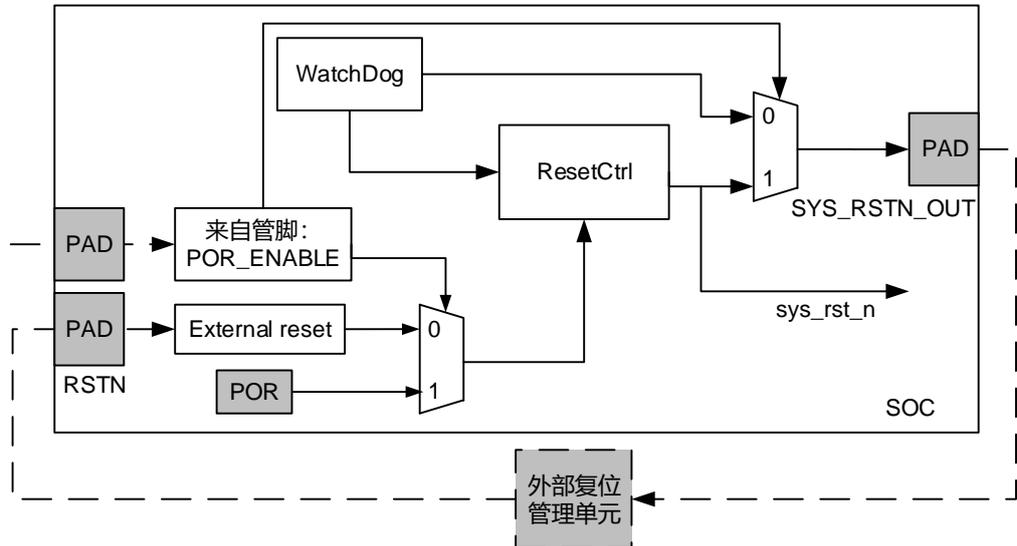
表3-1 复位信号分类表

复位信号类型	产生方式	用途
全局硬复位	WatchDog 复位请求、安全系统复位请求以及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。



内部复位和外部复位示意图如图 3-2 所示。

图3-2 内部复位和外部复位示意图



- POR_ENABLE 管脚接 1 时，选择 POR 复位生效，外部复位不生效，SYS_RSTN_OUT 管脚固定输出复位 sys_rst_n。
- POR_ENABLE 管脚接 0 时，选择外部复位生效，POR 复位不生效。

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。
- 外部复位。



- WatchDog 复位。
- 安全系统复位。
- 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。

须知

- 系统软复位请求发出后，整芯片必须等待至少约 20ms 才完成复位撤消。在这 20ms 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：

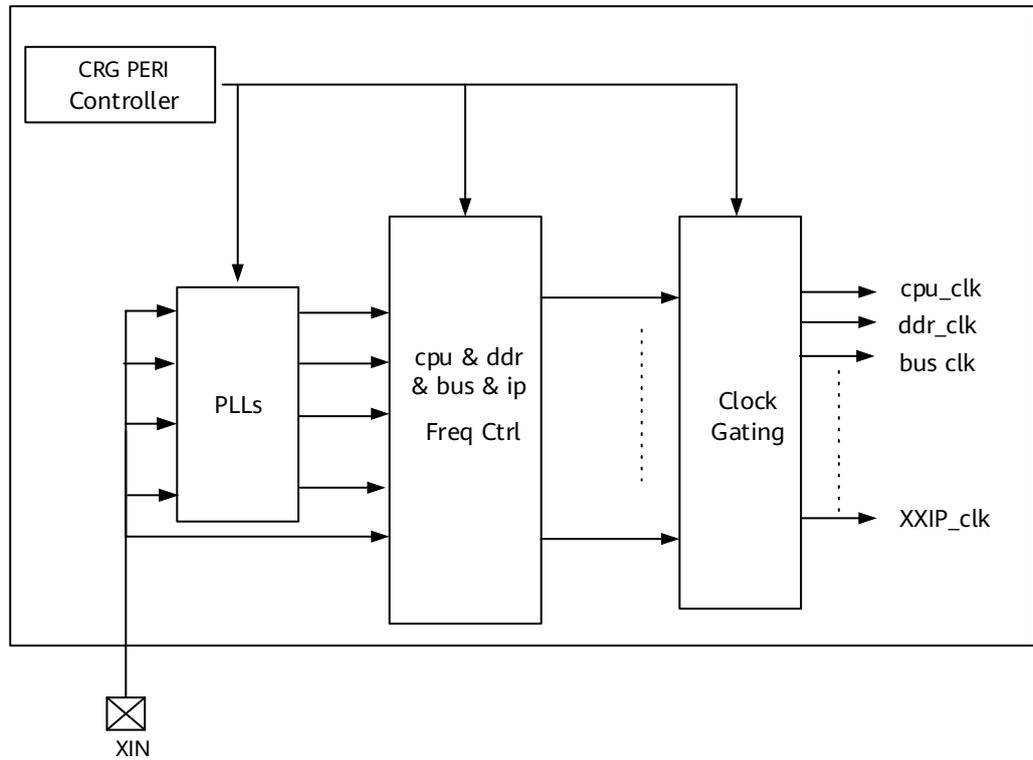
- 时钟输入的管理和控制
- 时钟分频和控制
- 生成各模块的工作时钟

3.2.2 功能框图

时钟管理模块功能框图如图 3-3 所示。



图3-3 时钟管理模块功能框图

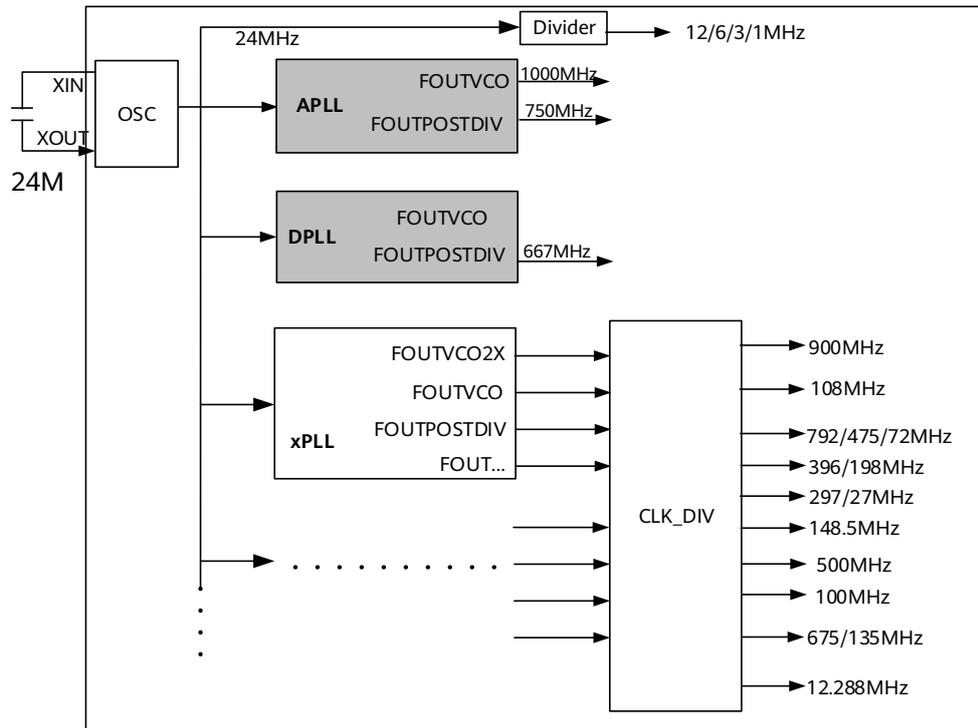


注：图中 XIN 为 PLL 输入时钟，固定连接 24MHz 晶体。

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 3-4 所示。

图3-4 时钟资源分布框图



注：图中灰色的 PLL 用户可编程配置。

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	备注
A55	24MHz	24MHz	-
		675MHz	
		792MHz	
		833MHz	
		900MHz	
		1000MHz	
A55_FCM	24MHz	24MHz	-
		540MHz	



模块	默认频点	所有频点	备注
		675MHz	
		792MHz	
DDR	24MHz	24MHz	-
		357MHz	
		594MHz	
		DPLL_POSTDIV	
DDR BUS	24MHz	24MHz	DDR 内部总线频点
		396MHz	
		500MHz	
		594MHz	
		675MHz	
		DPLL_POSTDIV	
DDR AXI	24MHz	24MHz	高性能媒体模块总线接口频点
		198MHz	
		300MHz	
		396MHz	
		475MHz	
		540MHz	
		594MHz	
CFG AXI	24MHz	24MHz	-
		100MHz	
		150MHz	
		198MHz	
DATA AXI	24MHz	24MHz	SEC(安全)、
		198MHz	FMC、eMMC、
		297MHz	SDIO、GMAC、
		339MHz	USB 总线接口频点
MIPI RX	100MHz	100MHz	MIPI RX CTRL



模块	默认频点	所有频点	备注
CTRL0/1/2/3		150MHz	0/1/2/3 独立可选
		198MHz	
		237MHz	
		300MHz	
		339MHz	
		475MHz	
		594MHz	
VICAP	150MHz	150MHz	Pixel Process Clock
		198MHz	
		300MHz	
		396MHz	
		475MHz	
		540MHz	
		594MHz	
ISPO	150MHz	150MHz	
		198MHz	
		300MHz	
		396MHz	
		475MHz	
		500MHz	
		540MHz	
		594MHz	
AIAO	786.432MHz	786.432MHz	实际工作频率为此时钟的分频
		1188MHz	
FMC	24MHz	24MHz	-
		100MHz	-
		150MHz	-
		198MHz	SDR 模式仅 3.3V



模块	默认频点	所有频点	备注
			可选
		237MHz	仅 DDR 模式可选
		300MHz	仅 DDR 模式可选
		396MHz	仅 DDR 模式 3.3V 可选
SDIO0/1	1600KHz	1600KHz	-
		100MHz	
		198MHz	
		396MHz	
		594MHz	
eMMC	1600KHz	1600KHz	-
		100MHz	
		198MHz	
		396MHz	
		594MHz	
VDP	66MHz	66MHz	通道源时钟
		74.25MHz	
		88MHz	
		108MHz	
		135MHz	
		148.5MHz	
		297MHz	
		RGB 分频器时钟	
SENSOR0/1/2/3	74.25MHz	74.25MHz	SOC 输出给 sensor 的参考时钟
		72MHz	
		54MHz	
		50MHz	



模块	默认频点	所有频点	备注
		37.125MHz	
		36MHz	
		27MHz	
		25MHz	
		24MHz	
		12MHz	

3.2.4 PLL 配置

芯片内部使用了 2 个可配置的 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_PLL96	PERI_CRG_PLL97

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	$FREF \times (fbdiv + frac / 2^{24}) / refdiv$	PLL 工作频率，要求大于等于 800MHz，且小于等于 3GHz
FOUTVCO2X	$FOUTVCO \times 2$	-
FOUTPOSTDIV	$FOUTVCO / (pstdiv1 \times pstdiv2)$	$pstdiv1 \geq pstdiv2$
FOUT1PH0	$FOUTVCO / (pstdiv1 \times pstdiv2 \times 2)$	-



PLL Pin	计算方法描述	注意事项
FOUT2	$FOUTVCO / (pstdiv1 \times pstdiv2 \times 4)$	-
FOUT3	$FOUTVCO / (pstdiv1 \times pstdiv2 \times 6)$	-
FOUT4	$FOUTVCO / (pstdiv1 \times pstdiv2 \times 8)$	-
<p>备注:</p> <p>fbdiv: 倍频系数的整数部分;</p> <p>frac: 倍频系数的小数部分;</p> <p>refdiv: 参考时钟分频系数;</p> <p>pstdiv1: 第一级输出分频系数;</p> <p>pstdiv2: 第二级输出分频系数。</p> <p>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。</p>		

注: 考虑 PLL 抗噪能力和更小的时钟 Jitter, 推荐 refdiv 配置为 1。

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 $postdiv2 = 1$, $postdiv1 = 2$
- 配置 FOUTVCO 输出 1200MHz, 根据 $FREF \times (fbdiv + frac/2^{24}) / refdiv = 1200MHz$, 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 需要参考表 3-3 寄存器, 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, 或者关闭业务时钟门控。

步骤 2 将对应 PLL 配置寄存器 1 的 bit[20]配置为 power down 状态。

步骤 3 更改 PLL 配置。

步骤 4 将对应 PLL 配置寄存器 1 的 bit[20]配置为正常工作状态。

步骤 5 等待 0.1ms 后观测对应 PLL LOCK 状态寄存器的 bit[4] lock_final 标志位。

步骤 6 如果 PLL LOCK 状态寄存器(PERI_CRG_PLL110/PERI_CRG_PLL14)的 bit[4] lock_final 标志位为 LOCK 状态, PLL 配置完成, 否则回到步骤 2, 直到 PLL LOCK 寄存器的 bit[4] lock_final 标志位为 LOCK 状态。



---结束

3.2.5 CRG 寄存器概览

CRG 寄存器概览如表 3-5 所示。

表3-5 CRG 寄存器概览 (基址: 0x0_1101_0000)

偏移地址	名称	描述	页码
0x0000	PERI_CRG_PLL0	APLL 配置寄存器 0	3-16
0x0004	PERI_CRG_PLL1	APLL 配置寄存器 1	3-17
0x0038	PERI_CRG_PLL14	APLL LOCK 状态寄存器	3-18
0x0068	PERI_CRG26	APLL REE 侧写权限控制寄存器	3-18
0x0180	PERI_CRG_PLL96	DPLL 配置寄存器 0	3-19
0x0184	PERI_CRG_PLL97	DPLL 配置寄存器 1	3-19
0x01B8	PERI_CRG_PLL110	DPLL LOCK 状态寄存器	3-20
0x01E8	PERI_CRG122	DPLL REE 侧写权限控制寄存器	3-21
0x2000	PERI_CRG2048	SOC 总线频率配置寄存器	3-21
0x2020	PERI_CRG2056	SOC 总线频率指示寄存器	3-23
0x2040	PERI_CRG2064	CPU_SUBSYS 时钟复位配置寄存器	3-24
0x2044	PERI_CRG2065	CPU_SUBSYS 频率指示寄存器	3-25
0x2048	PERI_CRG2066	CPU CORE0 时钟复位配置寄存器	3-26
0x204C	PERI_CRG2067	CPU CORE1 时钟复位配置寄存器	3-26
0x2058	PERI_CRG2070	CPU REE 侧写权限控制寄存器	3-27
0x2240	PERI_CRG2192	CPU Debug (Coresight) 时钟配置寄存器	3-27
0x2280	PERI_CRG2208	DDR 时钟复位配置寄存器	3-28
0x2284	PERI_CRG2209	DDR 频率指示寄存器	3-29



偏移地址	名称	描述	页码
0x22A0	PERI_CRG2216	DDR TEST 时钟复位配置寄存器	3-29
0x2A80	PERI_CRG2720	DMAC 时钟及软复位控制寄存器	3-30
0x34C0	PERI_CRG3376	eMMC 接口时钟复位控制寄存器	3-30
0x34C4	PERI_CRG3377	eMMC p4 DLL 控制寄存器	3-32
0x34C8	PERI_CRG3378	eMMC DRV DLL 控制寄存器	3-32
0x34CC	PERI_CRG3379	eMMC SAM DLL 控制寄存器	3-34
0x34D0	PERI_CRG3380	eMMC DS DLL 控制寄存器	3-35
0x34D4	PERI_CRG3381	eMMC 边沿检测时钟相位选择寄存器	3-36
0x34D8	PERI_CRG3382	eMMC 状态寄存器	3-37
0x35C0	PERI_CRG3440	SDIO0 接口时钟复位控制寄存器	3-38
0x35C4	PERI_CRG3441	SDIO0 p4 DLL 控制寄存器	3-39
0x35C8	PERI_CRG3442	SDIO0 DRV DLL 控制寄存器	3-40
0x35CC	PERI_CRG3443	SDIO0 SAM DLL 控制寄存器	3-41
0x35D4	PERI_CRG3445	SDIO0 边沿检测时钟相位选择寄存器	3-42
0x35D8	PERI_CRG3446	SDIO0 状态寄存器	3-43
0x36C0	PERI_CRG3504	SDIO1 接口时钟复位控制寄存器	3-44
0x36C4	PERI_CRG3505	SDIO1 p4 DLL 控制寄存器	3-45
0x36C8	PERI_CRG3506	SDIO1 DRV DLL 控制寄存器	3-45
0x36CC	PERI_CRG3507	SDIO1 SAM DLL 控制寄存器	3-47
0x36D4	PERI_CRG3509	SDIO1 边沿检测时钟相位选择寄存器	3-48
0x36D8	PERI_CRG3510	SDIO1 状态寄存器	3-49
0x37C0	PERI_CRG3568	网络时钟及软复位控制寄存器	3-50
0x37C4	PERI_CRG3569	GMAC 时钟及软复位控制寄存器	3-50



偏移地址	名称	描述	页码
0x37CC	PERI_CRG3571	FEPHY 时钟及软复位控制寄存器	3-51
0x38C0	PERI_CRG3632	USB2.0 PHY 时钟及软复位控制寄存器	3-51
0x3940	PERI_CRG3664	USB3.0 兼容(USB2.0) CTRL 时钟及软复位控制寄存器	3-53
0x3944	PERI_CRG3665	USB3.0 PHY 时钟及软复位控制寄存器	3-54
0x3F40	PERI_CRG4048	FMC 时钟及软复位控制寄存器	3-55
0x3F44	PERI_CRG4049	FMC 频率指示寄存器	3-56
0x4180	PERI_CRG4192	UART0 时钟及复位控制寄存器	3-57
0x4188	PERI_CRG4194	UART1 时钟及复位控制寄存器	3-57
0x4190	PERI_CRG4196	UART2 时钟及复位控制寄存器	3-58
0x4198	PERI_CRG4198	UART3 时钟及复位控制寄存器	3-59
0x41A0	PERI_CRG4200	UART4 时钟及复位控制寄存器	3-60
0x41A8	PERI_CRG4202	UART5 时钟及复位控制寄存器	3-60
0x4280	PERI_CRG4256	I2C0 时钟及复位控制寄存器	3-61
0x4288	PERI_CRG4258	I2C1 时钟及复位控制寄存器	3-62
0x4290	PERI_CRG4260	I2C2 时钟及复位控制寄存器	3-62
0x4298	PERI_CRG4262	I2C3 时钟及复位控制寄存器	3-63
0x42A0	PERI_CRG4264	I2C4 时钟及复位控制寄存器	3-64
0x42A8	PERI_CRG4266	I2C5 时钟及复位控制寄存器	3-64
0x42B0	PERI_CRG4268	I2C6 时钟及复位控制寄存器	3-65
0x42B8	PERI_CRG4270	I2C7 时钟及复位控制寄存器	3-66
0x4380	PERI_CRG4320	SPI_TFT 时钟及复位控制寄存器	3-66
0x4480	PERI_CRG4384	SPI0 时钟及复位控制寄存器	3-67
0x4488	PERI_CRG4386	SPI1 时钟及复位控制寄存器	3-68



偏移地址	名称	描述	页码
0x4490	PERI_CRG4388	SPI2 时钟及复位控制寄存器	3-68
0x4498	PERI_CRG4390	SPI3 时钟及复位控制寄存器	3-69
0x44A0	PERI_CRG4392	SPI 3WIRE 时钟及复位控制寄存器	3-69
0x4580	PERI_CRG4448	SVB PWM 时钟软复位控制寄存器	3-70
0x4588	PERI_CRG4450	PWM0 时钟软复位控制寄存器	3-70
0x4590	PERI_CRG4452	PWM1 时钟软复位控制寄存器	3-71
0x4598	PERI_CRG4454	PWM2 时钟软复位控制寄存器	3-72
0x4648	PERI_CRG4498	THERMO 热成像时钟软复位控制寄存器	3-72
0x46C0	PERI_CRG4528	LSADC 时钟软复位控制寄存器	3-74
0x8140	PERI_CRG8272	MIPI TX 时钟及软复位控制寄存器 1	3-75
0x8148	PERI_CRG8274	MIPI TX 时钟及软复位控制寄存器 2	3-76
0x8240	PERI_CRG8336	VDP 工作时钟及复位控制寄存器	3-77
0x8248	PERI_CRG8338	VDP BT 输出时钟相位配置寄存器	3-78
0x8250	PERI_CRG8340	VDP HD 时钟分频系数及时钟相位	3-78
0x8254	PERI_CRG8341	VDP PPC 时钟选择控制寄存器	3-79
0x8268	PERI_CRG8346	VDP RGB 时钟配置寄存器	3-80
0x8270	PERI_CRG8348	VDP BT 时钟门控及时钟选择控制寄存器	3-81
0x8280	PERI_CRG8352	MIPITX PIXEL 时钟控制寄存器	3-81
0x8440	PERI_CRG8464	SENSOR0 时钟复位配置寄存器	3-82
0x8460	PERI_CRG8472	SENSOR1 时钟复位配置寄存器	3-83
0x8480	PERI_CRG8480	SENSOR2 时钟复位配置寄存器	3-84
0x84A0	PERI_CRG8488	SENSOR3 时钟复位配置寄存器	3-85
0x8540	PERI_CRG8528	MIPI RX 时钟复位配置寄存器	3-86



偏移地址	名称	描述	页码
0x8560	PERI_CRG8536	MIPI_RX PIX0 时钟复位配置寄存器	3-88
0x8580	PERI_CRG8544	MIPI_RX PIX1 时钟复位配置寄存器	3-88
0x85A0	PERI_CRG8552	MIPI_RX PIX2 时钟复位配置寄存器	3-89
0x85C0	PERI_CRG8560	MIPI_RX PIX3 时钟复位配置寄存器	3-89
0x9140	PERI_CRG9296	VICAP 时钟及复位控制寄存器	3-90
0x9144	PERI_CRG9297	VICAP 复位状态寄存器	3-91
0x9148	PERI_CRG9298	VI CH 和 PT 时钟控制寄存器	3-91
0x9150	PERI_CRG9300	VI ISP0 时钟及复位控制寄存器	3-93
0x9154	PERI_CRG9301	VI ISP1 时钟及复位控制寄存器	3-94
0x9158	PERI_CRG9302	VI ISP2 时钟及复位控制寄存器	3-94
0x915C	PERI_CRG9303	VI ISP3 时钟及复位控制寄存器	3-95
0x9160	PERI_CRG9304	VI COMS0 时钟及复位控制寄存器	3-96
0x9164	PERI_CRG9305	VI PORT0 时钟及复位控制寄存器	3-96
0x9184	PERI_CRG9313	VI PORT1 时钟及复位控制寄存器	3-97
0x91A4	PERI_CRG9321	VI PORT2 时钟及复位控制寄存器	3-98
0x91C4	PERI_CRG9329	VI PORT3 时钟及复位控制寄存器	3-99
0xA880	PERI_CRG10784	AIAO 时钟复位控制寄存器	3-100
0xA884	PERI_CRG10785	AIAO DMIC 时钟复位配置寄存器	3-101
0xAA80	PERI_CRG10912	AUDIO CODEC 时钟复位控制寄存器	3-101

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: 0x1280_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。	0x800000

PERI_CRG_PLL1

PERI_CRG_PLL1 为 APLL 配置寄存器 1。

Offset Address: 0x0004 Total Reset Value: 0x0010_103E

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x00
[26]	RW	apll_bypass	APLL 时钟分频旁路(bypass)控制。 0: 非旁路; 1: 旁路。	0x0
[25]	RW	apll_dsmpd	APLL 小数分频控制。 0: 小数模式; 1: 整数模式。	0x0
[24]	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
[23]	RW	apll_postdivpd	APLL POSTDIV 输出 Power Down 控制。 0: 正常输出时钟; 1: 不输出时钟。	0x0
[22]	RW	apll_fout4phasepd	APLL FOUT 输出 Power Down 控制。	0x0



Bits	Access	Name	Description	Reset
			0: 正常输出时钟; 1: 不输出时钟。	
[21]	RW	apll_foutvcopd	APLL VCO 输出 Power Down 控制。 0: 正常输出时钟; 1: 不输出时钟。	0x0
[20]	RW	apll_pd	APLL Power Down 控制。 0: 正常工作状态; 1: power down 工作状态。	0x1
[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。	0x03E

PERI_CRG_PLL14

PERI_CRG_PLL14 为 APLL LOCK 状态寄存器。

Offset Address: 0x0038 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RO	apll_lock_final	APLL LOCK 状态。 0: Unlock; 1: Locked。	0x0
[3:0]	-	reserved	保留。	0x0

PERI_CRG26

PERI_CRG26 为 APLL REE 侧写权限控制寄存器。

Offset Address: 0x0068 Total Reset Value: 0x0000_000A



Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3:0]	RW	apll_cfg_nonsec_w r_disable	APLL REE 侧写权限配置信号。 0xA: 允许 REE 写; 其他: 不允许 REE 写。	0xA

PERI_CRG_PLL96

PERI_CRG_PLL96 为 DPLL 配置寄存器 0。

Offset Address: 0x0180 Total Reset Value: 0x1295_5556

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30:28]	RW	dpll_postdiv2	DPLL 第二级输出分频系数。	0x1
[27]	-	reserved	保留。	0x0
[26:24]	RW	dpll_postdiv1	DPLL 第一级输出分频系数。	0x2
[23:0]	RW	dpll_frac	DPLL 倍频系数小数部分。	0x955556

PERI_CRG_PLL97

PERI_CRG_PLL97 为 DPLL 配置寄存器 1。

Offset Address: 0x0184 Total Reset Value: 0x0010_1037

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x00
[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_dsmpd	DPLL 小数分频控制。	0x0



Bits	Access	Name	Description	Reset
			0: 小数模式; 1: 整数模式。	
[24]	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
[23]	RW	dpll_postdivpd	DPLL POSTDIV 输出 Power Down 控制。 0: 正常时钟输出; 1: 不输出时钟。	0x0
[22]	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
[21]	RW	dpll_foutvcopd	DPLL VCO 输出 Power Down 控制。 0: 正常输出时钟; 1: 不输出时钟。	0x0
[20]	RW	dpll_pd	DPLL Power Down 控制。 0: 正常工作状态; 1: power down 工作状态。	0x1
[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。	0x037

PERI_CRG_PLL110

PERI_CRG_PLL110 为 DPLL LOCK 状态寄存器。

Offset Address: 0x01B8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000



Bits	Access	Name	Description	Reset
[4]	RO	dpll_lock_final	DPLL LOCK 状态。 0: Unlock; 1: Locked。	0x0
[3:0]	-	reserved	保留。	0x0

PERI_CRG122

PERI_CRG122 为 DPLL REE 侧写权限控制寄存器。

Offset Address: 0x01E8 Total Reset Value: 0x0000_000A

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3:0]	RW	dpll_cfg_nonsec_w r_disable	DPLL REE 侧写权限配置信号。 0xA: 允许 REE 写; 其他: 不允许 REE 写。	0xA

PERI_CRG2048

PERI_CRG2048 为 SOC 总线频率配置寄存器。

Offset Address: 0x2000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:19]	-	reserved	保留。	0x0000
[18:16]	RW	ddr_cksel	DDR 内部总线时钟选择。 000: 24MHz; 001: DPLL_POSTDIV; 010: 396MHz; 011: 500MHz; 100: 594MHz;	0x0



Bits	Access	Name	Description	Reset
			101: 675MHz; 其他: 保留。	
[15]	-	reserved	保留。	0x0
[14:12]	RW	ddraxi_cksel	DDR AXI 时钟选择。 000: 24MHz; 001: 198MHz; 010: 300MHz; 011: 396MHz; 100: 475MHz; 101: 540MHz; 110: 594MHz; 其他: 保留。	0x0
[11:10]	-	reserved	保留。	0x0
[9:8]	RW	cfgaxi_cksel	CFG AXI 时钟选择。 00: 24MHz; 01: 100MHz; 10: 150MHz; 11: 198MHz。	0x0
[7:6]	-	reserved	保留。	0x0
[5:4]	RW	dataaxi_cksel	DATA AXI 时钟选择。 00: 24MHz; 01: 198MHz; 10: 297MHz; 11: 339MHz。	0x0
[3:0]	-	reserved	保留。	0x0



PERI_CRG2056

PERI_CRG2056 为 SOC 总线频率指示寄存器。

Offset Address: 0x2020 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:19]	-	reserved	保留。	0x0000
[18:16]	RO	ddr_sc_seled	DDR 模块内部总线时钟切换完成指示信号。 000: 24MHz; 001: DPLL_POSTDIV; 010: 396MHz; 011: 500MHz; 100: 594MHz; 101: 675MHz; 其他: 保留。	0x0
[15]	-	reserved	保留。	0x0
[14:12]	RO	ddraxi_sc_seled	DDR AXI 时钟切换完成指示信号。 000: 24MHz; 001: 198MHz; 010: 300MHz; 011: 396MHz; 100: 475MHz; 101: 540MHz; 110: 594MHz。 其他: 保留。	0x0
[11:10]	-	reserved	保留。	0x0
[9:8]	RO	cfgaxi_sc_seled	CFG AXI 时钟切换完成指示信号。 00: 24MHz; 01: 100MHz;	0x0



Bits	Access	Name	Description	Reset
			10: 150MHz; 11: 198MHz。	
[7:6]	-	reserved	保留。	0x0
[5:4]	RO	dataaxi_sc_seled	DATA AXI 时钟切换完成指示信号。 00: 24MHz; 01: 198MHz; 10: 297MHz; 11: 339MHZ。	0x0
[3:0]	-	reserved	保留。	0x0

PERI_CRG2064

PERI_CRG2064 为 CPU_SUBSYS 时钟复位配置寄存器。

Offset Address: 0x2040 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13:11]	RW	a55_fcm_cksel	A55 SCU 时钟选择。 000: 24MHz; 001: 540MHz; 010: 675MHz; 011: APLL_POSTDIV(只有当 cpu_cksel_limit_en=0xA 才可选); 100: 792MHz (只有当 cpu_cksel_limit_en=0xA 才可选); 其他: 保留。	0x0
[10:8]	RW	a55_cksel	A55 Core 时钟选择。 000: 24MHz; 001: 675MHz;	0x0



Bits	Access	Name	Description	Reset
			010: APLL POSTDIV; 011: 792MHz; 100: 833MHz; 101: 900MHz; 110: 1000MHz(只有当 cpu_cksel_limit_en=0xA 才可选)。 其他: 保留	
[7:0]	-	reserved	保留。	0x00

PERI_CRG2065

PERI_CRG2065 为 CPU_SUBSYS 频率指示寄存器。

Offset Address: 0x2044 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留。	0x000000
[10:8]	RO	a55_fcm_sc_seled	A55 SCU 时钟切换状态指示。 000: 24MHz; 001: 540MHz; 010: 675MHz; 011: APLL POSTDIV(只有当 cpu_cksel_limit_en=0xA 才可选); 100: 792M(只有当 cpu_cksel_limit_en=0xA 才可选); 其他: 保留。	0x0
[7:3]	-	reserved	保留。	0x00
[2:0]	RO	a55_sc_seled	A55 Core 时钟切换状态指示。 000: 24MHz; 001: 675MHz;	0x0



Bits	Access	Name	Description	Reset
			010: APLL_POSTDIV; 011: 792MHz; 100: 833MHz; 101: 900MHz; 110: 1000MHz(只有当 cpu_cksel_limit_en=0xA 才可选); 其他: 保留。	

PERI_CRG2066

PERI_CRG2066 为 CPU CORE0 时钟复位配置寄存器。

Offset Address: 0x2048 Total Reset Value: 0x0000_0010

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	core0_cken	core0 全局门控请求。 0: 禁止; 1: 使能。	0x1
[3:2]	-	reserved	保留。	0x0
[1]	RW	core0_srst_req	core0 除 debug 外逻辑软复位请求。 0: 不复位; 1: 复位。	0x0
[0]	RW	core0_po_srst_req	core0 上电(全局)软复位请求。 0: 不复位; 1: 复位。	0x0

PERI_CRG2067

PERI_CRG2067 为 CPU CORE1 时钟复位配置寄存器。



Offset Address: 0x204C Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x00000000
[4]	RW	core1_cken	core1 全局门控请求。 0: 禁止; 1: 使能。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	RW	core1_srst_req	core1 除 debug 外逻辑软复位请求。 0: 不复位; 1: 复位。	0x1
[0]	RW	core1_po_srst_req	core1 上电(全局)软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG2070

PERI_CRG2070 为 CPU REE 侧写权限控制寄存器。

Offset Address: 0x2058 Total Reset Value: 0x0000_000A

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000
[3:0]	RW	cpu_cfg_nonsec_w r_disable	CPU REE 侧写权限配置信号。 0xA: 允许 REE 写; 其他: 不允许 REE 写。	0xA

PERI_CRG2192

PERI_CRG2192 为 CPU Debug(Coresight)时钟配置寄存器。

Offset Address: 0x2240 Total Reset Value: 0x0100_0000



Bits	Access	Name	Description	Reset
[31:25]	-	reserved	保留。	0x00
[24]	RW	soc_cs_ckgate_bypass	调试通路 soc 全局门控。 0: 不旁路; 1: 旁路自动门控。	0x1
[23:0]	-	reserved	保留。	0x000000

PERI_CRG2208

PERI_CRG2208 为 DDR 时钟复位配置寄存器。

Offset Address: 0x2280 Total Reset Value: 0x0000_0004

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13:12]	RW	dfi_cksel	DDR DFI 时钟选择。 00: 24MHz; 01: 357MHz; 10: 594MHz; 11: DPLL POSTDIV.	0x0
[11:9]	-	reserved	保留。	0x0
[8]	RW	ddr_apb_cken	DDR APB 门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[7:5]	-	reserved	保留。	0x0
[4]	RW	ddr_cken	DDR 门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[2]	RW	ddr_apb_srst_req	DDR APB 软复位请求。 0: 撤消复位; 1: 复位。	0x1
[1:0]	-	reserved	保留。	0x0

PERI_CRG2209

PERI_CRG2209 为 DDR 频率指示寄存器。

Offset Address: 0x2284 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x000000
[9:8]	RO	dfi_sc_seled	DDR DFI (DDR PHY Interface)时钟切换完成指示信号。 00: 24MHz; 01: 357MHz; 10: 594MHz; 11: DPLL_POSTDIV.	0x0
[7:0]	-	reserved	保留。	0x00

PERI_CRG2216

PERI_CRG2216 为 DDR TEST 时钟复位配置寄存器。

Offset Address: 0x22A0 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	ddrtest_cken	DDRTEST 时钟门控配置寄存器。 0: 关闭时钟;	0x0



Bits	Access	Name	Description	Reset
			1: 打开时钟。	
[3:1]	-	reserved	保留。	0x0
[0]	RW	ddrtest_srst_req	DDRTEST 的软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG2720

PERI_CRG2720 为 DMAC 时钟及软复位控制寄存器。

Offset Address: 0x2A80 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5]	RW	edma_axi_cken	DMAC AXI 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[4]	RW	edma_apb_cken	DMAC APB 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	edma_srst_req	DMAC 的软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG3376

PERI_CRG3376 为 eMMC 接口时钟复位控制寄存器。

Offset Address: 0x34C0 Total Reset Value: 0x0007_0000



Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x00
[26:24]	RW	emmc_clk_sel	eMMC 4X 时钟选择，接口时钟是这个时钟的 4 分频。 000: 1.6MHz; 001: 100MHz; 010: 198MHz; 011: 396MHz; 100: 594MHz; 其他: 保留。	0x0
[23:19]	-	reserved	保留。	0x00
[18]	RW	emmc_mmc_tx_srst_req	eMMC TX 方向软复位请求。 0: 不复位; 1: 复位。	0x1
[17]	RW	emmc_mmc_rx_srst_req	eMMC RX 方向软复位请求。 0: 不复位; 1: 复位。	0x1
[16]	RW	emmc_mmc_srst_req	eMMC 的软复位请求。 0: 不复位; 1: 复位。	0x1
[15:2]	-	reserved	保留。	0x0000
[1]	RW	emmc_mmc_ahb_cken	eMMC 总线时钟门控配置。 0: 关闭; 1: 打开。	0x0
[0]	RW	emmc_mmc_cken	eMMC 时钟门控配置。 0: 关闭; 1: 打开。	0x0



PERI_CRG3377

PERI_CRG3377 为 eMMC p4 DLL 控制寄存器。

Offset Address: 0x34C4 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	emmc_mmc_dll_sr st_req	eMMC DLL(Master 和 Slave) 软复位请求。 0: 不复位; 1: 复位。	0x1
[0]	RW	emmc_p4_dll_stop	DLL Master 持续动态跟踪检测的使能信号。 0: 动态检测, 并根据检测参数实时校准相位; 1: Lock 后停止动态检测, 即 DLL 初始化后只检测一次。	0x1

PERI_CRG3378

PERI_CRG3378 为 eMMC DRV DLL 控制寄存器。

Offset Address: 0x34C8 Total Reset Value: 0x0008_0400

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19:15]	RW	emmc_drv_clk_pha se_sel	eMMC DRV 时钟相位配置, 默认 180°。 0x00: 0°; 0x01: 11.25°; 0x02: 22.5°; ... 0x1E: 337.5°; 0x1F: 348.75°。	0x10



Bits	Access	Name	Description	Reset
[14:11]	RW	emmc_drv_dll_tune	eMMC DRV DLL 时钟相位微调信号。 0x0: 不校准; 0x1: 增加 1 级 delay; 0x2: 增加 2 级 delay; 0x3: 增加 3 级 delay; ... 0x7: 增加 7 级 delay; 0x8: 不校准; 0x9: 减少 1 级 delay; 0xA: 减少 2 级 delay; 0xB: 减少 3 级 delay; ... 0xF: 减少 7 级 delay.	0x0
[10]	RW	emmc_drv_dll_slave_en	eMMC DRV DLL Slave 使能信号。 0: 不使能 DLL Slave; 1: 使能 DLL Slave.	0x1
[9]	RW	emmc_drv_dll_bypass	eMMC DRV DLL Slave 旁路选择信号。 0: 正常模式, 输出时钟相对输入时钟移相, 具体移相度数由 SDIO 控制器相关寄存器配置; 1: 旁路掉 DRV DLL Slave.	0x0
[8]	RW	emmc_drv_dll_mode	eMMC DRV DLL Slave 模式选择信号。 0: 正常模式; 1: 由 drv_dll_sel 控制 drv slave line 级数。	0x0
[7:0]	RW	emmc_drv_dll_ssel	eMMC DRV DLL Slave line 级数选择, 其中[7:6]保留。 0x00: DLL slave 设置 1 级 delay cell; 0x01: DLL slave 设置 1 级 delay cell;	0x00



Bits	Access	Name	Description	Reset
			0x02: DLL slave 设置 2 级 delay cell; 0x3f: DLL slave 设置 63 级 delay cell。 其他: 保留。	

PERI_CRG3379

PERI_CRG3379 为 eMMC SAM DLL 控制寄存器。

Offset Address: 0x34CC Total Reset Value: 0x0000_0400

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:11]	RW	emmc_sam_dll_tune	eMMC SAM DLL 时钟相位微调信号。 0x0: 不校准; 0x1: 增加 1 级 delay; 0x2: 增加 2 级 delay; 0x3: 增加 3 级 delay; ... 0x7: 增加 7 级 delay; 0x8: 不校准; 0x9: 减少 1 级 delay; 0xA: 减少 2 级 delay; 0xB: 减少 3 级 delay; ... 0xF: 减少 7 级 delay。	0x0
[10]	RW	emmc_sam_dll_slave_en	eMMC sam DLL Slave 使能信号。 0: 不使能 DLL Slave; 1: 使能 DLL Slave。	0x1
[9]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[8]	RW	emmc_sam_dll_mode	eMMC SAM DLL Slave 模式选择信号。 0: 正常模式; 1: 由 sam_dll_ssel 控制 sam slave line 级数。	0x0
[7:0]	RW	emmc_sam_dll_ssel	eMMC SAM DLL Slave line 级数选择, 其中[7:6]保留。 0x00: DLL slave 设置 1 级 delay cell; 0x01: DLL slave 设置 1 级 delay cell; 0x02: DLL slave 设置 2 级 delay cell; 0x3F: DLL slave 设置 63 级 delay cell。	0x00

PERI_CRG3380

PERI_CRG3380 为 eMMC DS DLL 控制寄存器。

Offset Address: 0x34D0 Total Reset Value: 0x0000_0400

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:11]	RW	emmc_ds_dll_tune	eMMC ds DLL 时钟相位微调信号。 0x0: 不校准; 0x1: 增加 1 级 delay; 0x2: 增加 2 级 delay; 0x3: 增加 3 级 delay; ... 0x7: 增加 7 级 delay; 0x8: 不校准; 0x9: 减少 1 级 delay; 0xA: 减少 2 级 delay;	0x0



Bits	Access	Name	Description	Reset
			0xB: 减少 3 级 delay; ... 0xF: 减少 7 级 delay。	
[10]	RW	emmc_ds_dll_slave_en	eMMC ds DLL Slave 使能信号。 0: 不使能 DLL Slave; 1: 使能 DLL Slave。	0x1
[9]	RW	emmc_ds_dll_bypass	eMMC ds DLL Slave 旁路选择信号。 0: 正常模式, 输出时钟相对输入时钟移相, 具体移相度数由 SDIO 控制器相关寄存器配置; 1: 旁路掉 DRV DLL Slave。	0x0
[8]	RW	emmc_ds_dll_mode	eMMC ds DLL Slave 模式选择信号。 0: 正常模式; 1: 由 ds_dll_ssel 控制 ds slave line 级数。	0x0
[7:0]	RW	emmc_ds_dll_ssel	eMMC ds DLL Slave line 级数选择, 其中 [7]保留。 0x00: DLL slave 设置 1 级 delay cell; 0x01: DLL slave 设置 1 级 delay cell; 0x02: DLL slave 设置 2 级 delay cell; 0x3F: DLL slave 设置 63 级 delay cell;	0x00

PERI_CRG3381

PERI_CRG3381 为 eMMC 边沿检测时钟相位选择寄存器。

Offset Address: 0x34D4 Total Reset Value: 0x0000_0008

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000



Bits	Access	Name	Description	Reset
[4:0]	RW	emmc_sample_b_clk_sel	边沿检测时钟相位(相对于 sample 时钟)选择信号: 0x04: 45°; 0x08: 90°(上电默认值); 0x1C: 315°; 注意: [1:0]必须为 00。	0x08

PERI_CRG3382

PERI_CRG3382 为 eMMC 状态寄存器。

Offset Address: 0x34D8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RO	emmc_sam_dll_ready	eMMC sam DLL ready 信号。 0: DLL SLAVE not ready; 1: DLL SLAVE ready。	0x0
[11]	RO	emmc_drv_dll_ready	eMMC drv DLL ready 信号。 0: DLL SLAVE not ready; 1: DLL SLAVE ready。	0x0
[10]	RO	emmc_ds_dll_ready	eMMC DS DLL ready 信号。 0: DLL SLAVE not ready; 1: DLL SLAVE ready。	0x0
[9]	RO	emmc_p4_dll_locked	eMMC P4 DLL lock 信号。 0: DLL MASTER unlock; 1: DLL MASTER locked。	0x0
[8]	RO	emmc_p4_dll_overflow	eMMC P4 DLL overflow 信号。	0x0



Bits	Access	Name	Description	Reset
			0: DLL MASTER unoverflow; 1: DLL MASTER overflow。	
[7:0]	RO	emmc_p4_dll_mdly_tap	eMMC P4 DLL mdly_tap 信号。 DLL MASTER LINE tap 值。	0x00

PERI_CRG3440

PERI_CRG3440 为 SDIO0 接口时钟复位控制寄存器。

Offset Address: 0x35C0 Total Reset Value: 0x0007_0000

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x00
[26:24]	RW	sdio0_clk_sel	SDIO0 4X 时钟选择，接口时钟是这个时钟的 4 分频。 000: 1.6MHz; 001: 100MHz; 010: 198MHz; 011: 396MHz; 100: 594MHz; 其他: 保留。	0x0
[23:19]	-	reserved	保留。	0x00
[18]	RW	sdio0_mmc_tx_srst_req	SDIO0 TX 方向软复位请求。 0: 不复位; 1: 复位。	0x1
[17]	RW	sdio0_mmc_rx_srst_req	SDIO0 RX 方向软复位请求。 0: 不复位; 1: 复位。	0x1
[16]	RW	sdio0_mmc_srst_req	SDIO0 软复位请求。	0x1



Bits	Access	Name	Description	Reset
			0: 不复位; 1: 复位。	
[15:2]	-	reserved	保留。	0x0000
[1]	RW	sdio0_mmc_ahb_cken	SDIO0 总线时钟门控配置。 0: 关闭; 1: 打开。	0x0
[0]	RW	sdio0_mmc_cken	SDIO0 时钟门控配置。 0: 关闭; 1: 打开。	0x0

PERI_CRG3441

PERI_CRG3441 为 SDIO0 p4 DLL 控制寄存器。

Offset Address: 0x35C4 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	sdio0_mmc_dll_srst_req	SDIO0 DLL(Master 和 Slave) 软复位请求。 0: 不复位; 1: 复位。	0x1
[0]	RW	sdio0_p4_dll_stop	DLL Master 持续动态跟踪检测的使能信号。 0: 动态检测, 并根据检测参数实时校准相位; 1: Lock 后停止动态检测, 即 DLL 初始化后只检测一次。	0x1



PERI_CRG3442

PERI_CRG3442 为 SDIO0 DRV DLL 控制寄存器。

Offset Address: 0x35C8 Total Reset Value: 0x0008_0400

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19:15]	RW	sdio0_drv_clk_phase_sel	SDIO0 DRV 时钟相位配置，默认 180°。 0x00: 0°; 0x01: 11.25°; 0x02: 22.5°; ... 0x1E: 337.5°; 0x1F: 348.75°。	0x10
[14:11]	RW	sdio0_drv_dll_tune	SDIO0 DRV DLL 时钟相位微调信号。 0x0: 不校准; 0x1: 增加 1 级 delay; 0x2: 增加 2 级 delay; 0x3: 增加 3 级 delay; ... 0x7: 增加 7 级 delay; 0x8: 不校准; 0x9: 减少 1 级 delay; 0xA: 减少 2 级 delay; 0xB: 减少 3 级 delay; ... 0xF: 减少 7 级 delay。	0x0
[10]	RW	sdio0_drv_dll_slave_en	SDIO0 DRV DLL Slave 使能信号。 0: 不使能 DLL Slave; 1: 使能 DLL Slave。	0x1
[9]	RW	sdio0_drv_dll_bypa	SDIO0 DRV DLL Slave 旁路选择信号。	0x0



Bits	Access	Name	Description	Reset
		ss	0: 正常模式, 输出时钟相对输入时钟移相, 具体移相度数由 SDIO0 控制器相关寄存器配置; 1: 旁路掉 DRV DLL Slave。	
[8]	RW	sdio0_drv_dll_mode	SDIO0 DRV DLL Slave 模式选择信号。 0: 正常模式; 1: 由 drv_dll_sel 控制 drv slave line 级数。	0x0
[7:0]	RW	sdio0_drv_dll_ssel	SDIO0 DRV DLL Slave line 级数选择, 其中[7:6]保留。 0x00: DLL slave 设置 1 级 delay cell; 0x01: DLL slave 设置 1 级 delay cell; 0x02: DLL slave 设置 2 级 delay cell; 0x3F: DLL slave 设置 63 级 delay cell。	0x00

PERI_CRG3443

PERI_CRG3443 为 SDIO0 SAM DLL 控制寄存器。

Offset Address: 0x35CC Total Reset Value: 0x0000_0400

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:11]	RW	sdio0_sam_dll_tune	SDIO0 SAM DLL 时钟相位微调信号。 0x0: 不校准; 0x1: 增加 1 级 delay; 0x2: 增加 2 级 delay; 0x3: 增加 3 级 delay; ... 0x7: 增加 7 级 delay;	0x0



Bits	Access	Name	Description	Reset
			0x8: 不校准; 0x9: 减少 1 级 delay; 0xA: 减少 2 级 delay; 0xB: 减少 3 级 delay; ... 0xF: 减少 7 级 delay。	
[10]	RW	sdio0_sam_dll_slave_en	SDIO0 sam DLL Slave 使能信号。 0: 不使能 DLL Slave; 1: 使能 DLL Slave。	0x1
[9]	-	reserved	保留。	0x0
[8]	RW	sdio0_sam_dll_mode	SDIO0 SAM DLL Slave 模式选择信号。 0: 正常模式; 1: 由 sam_dll_ssel 控制 sam slave line 级数。	0x0
[7:0]	RW	sdio0_sam_dll_ssel	SDIO0 SAM DLL Slave line 级数选择, 其中[7:6]保留。 0x00: DLL slave 设置 1 级 delay cell; 0x01: DLL slave 设置 1 级 delay cell; 0x02: DLL slave 设置 2 级 delay cell; 0x3F: DLL slave 设置 63 级 delay cell。	0x00

PERI_CRG3445

PERI_CRG3445 为 SDIO0 边沿检测时钟相位选择寄存器。

Offset Address: 0x35D4 Total Reset Value: 0x0000_0008

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000



Bits	Access	Name	Description	Reset
[4:0]	RW	sdio0_sample_b_cclk_sel	边沿检测时钟相位(相对于 sample 时钟)选择信号; 0x04: 45°; 0x08: 90° (上电默认值); 0x1C: 315°; 注意: [1:0]必须为 00。	0x08

PERI_CRG3446

PERI_CRG3446 为 SDIO0 状态寄存器。

Offset Address: 0x35D8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RO	sdio0_sam_dll_ready	SDIO0 sam DLL ready 信号。 0: DLL SLAVE not ready; 1: DLL SLAVE ready。	0x0
[11]	RO	sdio0_drv_dll_ready	SDIO0 drv DLL ready 信号。 0: DLL SLAVE not ready; 1: DLL SLAVE ready。	0x0
[10]	-	reserved	保留。	0x0
[9]	RO	sdio0_p4_dll_locked	SDIO0 P4 DLL lock 信号。 0: DLL MASTER unlock; 1: DLL MASTER locked。	0x0
[8]	RO	sdio0_p4_dll_overflow	SDIO0 P4 DLL overflow 信号。 0: DLL MASTER unoverflow; 1: DLL MASTER overflow。	0x0



Bits	Access	Name	Description	Reset
[7:0]	RO	sdio0_p4_dll_mdly_tap	SDIO0 P4 DLL mdly_tap 信号。 DLL MASTER LINE tap 值。	0x00

PERI_CRG3504

PERI_CRG3504 为 SDIO1 接口时钟复位控制寄存器。

Offset Address: 0x36C0 Total Reset Value: 0x0007_0000

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x00
[26:24]	RW	sdio1_clk_sel	SDIO1 4X 时钟选择, 接口时钟是这个时钟的 4 分频。 000: 1.6MHz; 001: 100MHz; 010: 198MHz; 011: 396MHz; 100: 594MHz; 其他: 保留。	0x0
[23:19]	-	reserved	保留。	0x00
[18]	RW	sdio1_mmc_tx_srst_req	SDIO1 TX 方向软复位请求。 0: 不复位; 1: 复位。	0x1
[17]	RW	sdio1_mmc_rx_srst_req	SDIO1 RX 方向软复位请求。 0: 不复位; 1: 复位。	0x1
[16]	RW	sdio1_mmc_srst_req	SDIO1 的软复位请求。 0: 不复位; 1: 复位。	0x1



Bits	Access	Name	Description	Reset
[15:2]	-	reserved	保留。	0x0000
[1]	RW	sdio1_mmc_ahb_cken	SDIO1 总线时钟门控配置。 0: 关闭; 1: 打开。	0x0
[0]	RW	sdio1_mmc_cken	SDIO1 时钟门控配置。 0: 关闭; 1: 打开。	0x0

PERI_CRG3505

PERI_CRG3505 为 SDIO1 p4 DLL 控制寄存器。

Offset Address: 0x36C4 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	sdio1_mmc_dll_srst_req	SDIO1 DLL(Master 和 Slave) 软复位请求。 0: 不复位; 1: 复位。	0x1
[0]	RW	sdio1_p4_dll_stop	DLL Master 持续动态跟踪检测的使能信号。 0: 动态检测, 并根据检测参数实时校准相位; 1: Lock 后停止动态检测, 即 DLL 初始化后只检测一次。	0x1

PERI_CRG3506

PERI_CRG3506 为 SDIO1 DRV DLL 控制寄存器。



Offset Address: 0x36C8 Total Reset Value: 0x0008_0400

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19:15]	RW	sdio1_drv_clk_phase_sel	SDIO1 DRV 时钟相位配置，默认 180C。 0x00: 0°; 0x01: 11.25°; 0x02: 22.5°; ... 0x1E: 337.5°; 0x1F: 348.75°。	0x10
[14:11]	RW	sdio1_drv_dll_tune	SDIO1 DRV DLL 时钟相位微调信号。 0x0: 不校准; 0x1: 增加 1 级 delay; 0x2: 增加 2 级 delay; 0x3: 增加 3 级 delay; ... 0x7: 增加 7 级 delay; 0x8: 不校准; 0x9: 减少 1 级 delay; 0xA: 减少 2 级 delay; 0xB: 减少 3 级 delay; ... 0xF: 减少 7 级 delay。	0x0
[10]	RW	sdio1_drv_dll_slave_en	SDIO1 DRV DLL Slave 使能信号。 0: 不使能 DLL Slave; 1: 使能 DLL Slave。	0x1
[9]	RW	sdio1_drv_dll_bypass	SDIO1 DRV DLL Slave 旁路选择信号。 0: 正常模式，输出时钟相对输入时钟移相，具体移相度数由 SDIO1 控制器相关寄	0x0



Bits	Access	Name	Description	Reset
			寄存器配置; 1: 旁路掉 DRV DLL Slave。	
[8]	RW	sdio1_drv_dll_mode	SDIO1 DRV DLL Slave 模式选择信号。 0: 正常模式; 1: 由 drv_dll_sel 控制 drv slave line 级数。	0x0
[7:0]	RW	sdio1_drv_dll_ssel	SDIO1 DRV DLL Slave line 级数选择, 其中[7:6]保留。 0x00: DLL slave 设置 1 级 delay cell; 0x01: DLL slave 设置 1 级 delay cell; 0x02: DLL slave 设置 2 级 delay cell; 0x3F: DLL slave 设置 63 级 delay cell。	0x00

PERI_CRG3507

PERI_CRG3507 为 SDIO1 SAM DLL 控制寄存器。

Offset Address: 0x36CC Total Reset Value: 0x0000_0400

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:11]	RW	sdio1_sam_dll_tune	SDIO1 SAM DLL 时钟相位微调信号。 0x0: 不校准; 0x1: 增加 1 级 delay; 0x2: 增加 2 级 delay; 0x3: 增加 3 级 delay; ... 0x7: 增加 7 级 delay; 0x8: 不校准;	0x0



Bits	Access	Name	Description	Reset
			0x9: 减少 1 级 delay; 0xA: 减少 2 级 delay; 0xB: 减少 3 级 delay; ... 0xF: 减少 7 级 delay。	
[10]	RW	sdio1_sam_dll_slave_en	SDIO1 sam DLL Slave 使能信号。 0: 不使能 DLL Slave; 1: 使能 DLL Slave。	0x1
[9]	-	reserved	保留。	0x0
[8]	RW	sdio1_sam_dll_mode	SDIO1 SAM DLL Slave 模式选择信号。 0: 正常模式; 1: 由 sam_dll_ssel 控制 sam slave line 级数。	0x0
[7:0]	RW	sdio1_sam_dll_ssel	SDIO1 SAM DLL Slave line 级数选择, 其中[7:6]保留。 0x00: DLL slave 设置 1 级 delay cell; 0x01: DLL slave 设置 1 级 delay cell; 0x02: DLL slave 设置 2 级 delay cell; 0x3F: DLL slave 设置 63 级 delay cell。	0x00

PERI_CRG3509

PERI_CRG3509 为 SDIO1 边沿检测时钟相位选择寄存器。

Offset Address: 0x36D4 Total Reset Value: 0x0000_0008

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4:0]	RW	sdio1_sample_b_clk_sel	边沿检测时钟相位(相对于 sample 时钟)选	0x08



Bits	Access	Name	Description	Reset
			择信号。 0x04: 45°; 0x08: 90° (上电默认值); 0x1C: 315°; 注意: [1:0] 必须为 00。	

PERI_CRG3510

PERI_CRG3510 为 SDIO1 状态寄存器。

Offset Address: 0x36D8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RO	sdio1_sam_dll_ready	SDIO1 sam DLL ready 信号。 0: DLL SLAVE not ready; 1: DLL SLAVE ready。	0x0
[11]	RO	sdio1_drv_dll_ready	SDIO1 drv DLL ready 信号。 0: DLL SLAVE not ready; 1: DLL SLAVE ready。	0x0
[10]	-	reserved	保留。	0x0
[9]	RO	sdio1_p4_dll_locked	SDIO1 P4 DLL lock 信号。 0: DLL MASTER unlock; 1: DLL MASTER locked。	0x0
[8]	RO	sdio1_p4_dll_overflow	SDIO1 P4 DLL overflow 信号。 0: DLL MASTER unoverflow; 1: DLL MASTER overflow。	0x0
[7:0]	RO	sdio1_p4_dll_mdly_tap	SDIO1 P4 DLL mdly_tap 信号。	0x00



Bits	Access	Name	Description	Reset
			DLL MASTER LINE tap 值。	

PERI_CRG3568

PERI_CRG3568 为网络时钟及软复位控制寄存器。

Offset Address: 0x37C0 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RW	rmii_cksel	RMII 时钟选择。 0: 选择 CRG 时钟; 1: 选择 PAD 输入时钟。	0x0
[11:5]	-	reserved	保留。	0x00
[4]	RW	gmac_if_cken	MAC_IF 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	gmac_if_srst_req	MAC_IF 的软复位请求。 0: 撤消复位; 1: 复位。	0x1

PERI_CRG3569

PERI_CRG3569 为 GMAC 时钟及软复位控制寄存器。

Offset Address: 0x37C4 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	gmac_cken	GMAC 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。	
[3:1]	-	reserved	保留。	0x0
[0]	RW	gmac_srst_req	GMAC 的软复位请求。 0: 撤消复位; 1: 复位。	0x1

PERI_CRG3571

PERI_CRG3571 为 FEPHY 时钟及软复位控制寄存器。

Offset Address: 0x37CC Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13:12]	RW	ext_fephy_cksel	外接 FEPHY 时钟选择。 00: 25MHz; 01: 50MHz; 其他: 保留。	0x0
[11:1]	-	reserved	保留。	0x000
[0]	RW	ext_fephy_srst_req	外接 FEPHY 的软复位请求。 0: 撤消复位; 1: 复位。	0x1

PERI_CRG3632

PERI_CRG3632 为 USB2.0 PHY 时钟及软复位控制寄存器。

Offset Address: 0x38C0 Total Reset Value: 0x0000_0107



Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x000000
[9:8]	RW	usb2_phy_pll_ksel	USB2.0 PHY PLL 参考时钟源选择。 00: 保留; 01: 60MHz; 10: 30MHz; 11: 20MHz。	0x1
[7]	-	reserved	保留。	0x0
[6]	RW	usb2_phy_apb_cke n	USB2.0 PHY APB 时钟门控。 0: 关闭时钟; 1: 打开时钟。	0x0
[5]	-	reserved	保留。	0x0
[4]	RW	usb2_phy_xtal_cke n	USB2.0 PHY 晶振参考时钟门控。 0: 关闭时钟; 1: 打开时钟。	0x0
[3]	-	reserved	保留。	0x0
[2]	RW	usb2_phy_apb_srst _req	USB2.0 PHY APB 软复位请求。 0: 不复位; 1: 复位。	0x1
[1]	RW	usb2_phy_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_CRG3664

PERI_CRG3664 为 USB3.0 兼容(USB2.0) CTRL 时钟及软复位控制寄存器。

Offset Address: 0x3940 Total Reset Value: 0x0001_0003

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19]	RW	usb3_phy_clk_pctrl	USB3.0 PHY PIPE 时钟相位控制，默认正向。 0: 时钟不取反; 1: 时钟取反。	0x0
[18]	RW	usb3_pclk_occ_sel	USB3.0 时钟源选择。 0: 选择 PHY 时钟; 1: 选择内部 CRG OCC 时钟。	0x0
[17]	-	reserved	保留。	0x0
[16]	RW	usb3_freeclk_cksel	USB3.0 控制器时钟源选择。 0: 选择 USB2.0 PHY UTMI 时钟; 1: 选择 USB2.0 PHY FREECLK 时钟。	0x1
[15:13]	-	reserved	保留。	0x0
[12]	RW	usb3_pipe_cken	USB3.0 控制器 PIPE 时钟门控。 0: 关闭时钟; 1: 打开时钟。	0x0
[11:9]	-	reserved	保留。	0x0
[8]	RW	usb3_utmi_cken	USB3.0 控制器 UTMI 时钟门控。 0: 关闭时钟; 1: 打开时钟。	0x0
[7]	-	reserved	保留。	0x0
[6]	RW	usb3_suspend_cken	USB3.0 控制器 SUSPEND 时钟门控。 0: 关闭时钟;	0x0



Bits	Access	Name	Description	Reset
			1: 打开时钟。	
[5]	RW	usb3_ref_cken	USB3.0 控制器 REF 时钟门控。 0: 关闭时钟; 1: 打开时钟。	0x0
[4]	RW	usb3_bus_cken	USB3.0 控制器总线时钟门控。 0: 关闭时钟; 1: 打开时钟。	0x0
[3]	-	reserved	保留。	0x0
[2]	RW	usb3_pclk_cken	USB3.0 配置总线时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[1]	RW	usb3_pclk_srst_req	USB3.0 配置总线时钟域的软复位请求。 0: 撤消复位; 1: 复位。	0x1
[0]	RW	usb3_srst_req	USB3.0 控制器软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG3665

PERI_CRG3665 为 USB3.0 PHY 时钟及软复位控制寄存器。

Offset Address: 0x3944 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:9]	-	reserved	保留。	0x000000
[8]	RW	usb3phy_refclk_sel	USB3.0 PHY 参考时钟选择: 0: 选择 100MHz 时钟; 1: 选择晶振时钟。	0x0



Bits	Access	Name	Description	Reset
[7:5]	-	reserved	保留。	0x0
[4]	RW	usb3phy_ref_cken	USB3.0 PHY 参考时钟门控。 0: 关闭时钟; 1: 打开时钟。	0x0
[3]	-	reserved	保留。	0x0
[2]	RW	usb3phy_apb_cken	USB3.0 PHY APB 时钟门控 0: 时钟关闭; 1: 时钟开启。	0x0
[1]	RW	usb3phy_test_srst_req	USB3.0 PHY TEST 软复位请求。 0: 不复位; 1: 复位。	0x1
[0]	RW	usb3phy_srst_req	USB3.0 PHY 端口软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG4048

PERI_CRG4048 为 FMC 时钟及软复位控制寄存器。

Offset Address: 0x3F40 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:12]	RW	fmc_cksel	FMC 时钟源选择(SDR 模式下, 接口时钟为以下时钟的 2 分频; DDR 模式下, 接口时钟为以下时钟的 4 分频)。 000: 24MHz; 001: 100MHz; 010: 150MHz;	0x0



Bits	Access	Name	Description	Reset
			011: 198MHz(SDR 模式仅 3.3V 可选); 100: 237MHz(仅 DDR 模式可选); 101: 300MHz(仅 DDR 模式可选); 110: 396MHz(仅 DDR 模式 3.3V 可选); 111: 保留。	
[11:5]	-	reserved	保留。	0x00
[4]	RW	fmc_cken	FMC 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	fmc_srst_req	FMC 的软复位请求。 0: 撤消复位; 1: 复位。	0x1

PERI_CRG4049

PERI_CRG4049 为 FMC 频率指示寄存器。

Offset Address: 0x3F44 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留。	0x000000
[10:8]	RO	fmc_sc_seled	FMC 时钟切换完成指示信号。 000: 24MHz; 001: 100MHz; 010: 150MHz; 011: 198MHz(SDR 模式仅 3.3V 可选); 100: 237MHz(仅 DDR 模式可选); 101: 300MHz(仅 DDR 模式可选);	0x0



Bits	Access	Name	Description	Reset
			110: 396MHz(仅 DDR 模式 3.3V 可选); 111: 保留。	
[7:0]	-	reserved	保留。	0x00

PERI_CRG4192

PERI_CRG4192 为 UART0 时钟及复位控制寄存器。

Offset Address: 0x4180 Total Reset Value: 0x0000_2001

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13:12]	RW	uart0_cksel	UART0 时钟选择。 00: 100MHz; 01: 50MHz; 10: 24MHz; 11: 3MHz。	0x2
[11:5]	-	reserved	保留。	0x00
[4]	RW	uart0_cken	UART0 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	uart0_srst_req	UART0 的软复位请求。 0: 撤消复位; 1: 复位。	0x1

PERI_CRG4194

PERI_CRG4194 为 UART1 时钟及复位控制寄存器。



Offset Address: 0x4188 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13:12]	RW	uart1_cksel	UART1 时钟选择。 00: 100MHz; 01: 50MHz; 10: 24MHz; 11: 3MHz。	0x0
[11:5]	-	reserved	保留。	0x00
[4]	RW	uart1_cken	UART1 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	uart1_srst_req	UART1 的软复位请求。 0: 撤消复位; 1: 复位。	0x1

PERI_CRG4196

PERI_CRG4196 为 UART2 时钟及复位控制寄存器。

Offset Address: 0x4190 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13:12]	RW	uart2_cksel	UART2 时钟选择。 00: 100MHz; 01: 50MHz; 10: 24MHz; 11: 3MHz。	0x0



[11:5]	-	reserved	保留。	0x00
[4]	RW	uart2_cken	UART2 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	uart2_srst_req	UART2 的软复位请求。 0: 撤消复位; 1: 复位。	0x1

PERI_CRG4198

PERI_CRG4198 为 UART3 时钟及复位控制寄存器。

Offset Address: 0x4198 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13:12]	RW	uart3_cksel	UART3 时钟选择。 00: 100MHz; 01: 50MHz; 10: 24MHz; 11: 3MHz。	0x0
[11:5]	-	reserved	保留。	0x00
[4]	RW	uart3_cken	UART3 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	uart3_srst_req	UART3 的软复位请求。 0: 撤消复位;	0x1



			1: 复位。	
--	--	--	--------	--

PERI_CRG4200

PERI_CRG4200 为 UART4 时钟及复位控制寄存器。

Offset Address: 0x41A0 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x0000
[13:12]	RW	uart4_cksel	UART4 时钟选择。 00: 100MHz; 01: 50MHz; 10: 24MHz; 11: 3MHz。	0x0
[11:5]	-	reserved	保留。	0x00
[4]	RW	uart4_cken	UART4 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	uart4_srst_req	UART4 的软复位请求。 0: 撤消复位; 1: 复位。	0x1

PERI_CRG4202

PERI_CRG4202 为 UART5 时钟及复位控制寄存器。

Offset Address: 0x41A8 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x0000



Bits	Access	Name	Description	Reset
[13:12]	RW	uart5_cksel	UART5 时钟选择。 00: 100MHz; 01: 50MHz; 10: 24MHz; 11: 3MHz。	0x0
[11:5]	-	reserved	保留。	0x00
[4]	RW	uart5_cken	UART5 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	uart5_srst_req	UART5 的软复位请求。 0: 撤消复位; 1: 复位。	0x1

PERI_CRG4256

PERI_CRG4256 为 I2C0 时钟及复位控制寄存器。

Offset Address: 0x4280 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RW	i2c0_cksel	I2C0 时钟选择。 0: 50MHz; 1: 100MHz。	0x0
[11:5]	-	reserved	保留	0x00
[4]	RW	i2c0_cken	I2C0 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0



Bits	Access	Name	Description	Reset
[3:1]	-	reserved	保留	0x0
[0]	RW	i2c0_srst_req	I2C0 的软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG4258

PERI_CRG4258 为 I2C1 时钟及复位控制寄存器。

Offset Address: 0x4288 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RW	i2c1_cksel	I2C1 时钟选择。 0: 50MHz; 1: 100MHz。	0x0
[11:5]	-	reserved	保留	0x00
[4]	RW	i2c1_cken	I2C1 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留	0x0
[0]	RW	i2c1_srst_req	I2C1 的软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG4260

PERI_CRG4260 为 I2C2 时钟及复位控制寄存器。

Offset Address: 0x4290 Total Reset Value: 0x0000_0001



Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RW	i2c2_cksel	I2C2 时钟选择。 0: 50MHz; 1: 100MHz。	0x0
[11:5]	-	reserved	保留	0x00
[4]	RW	i2c2_cken	I2C2 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留	0x0
[0]	RW	i2c2_srst_req	I2C2 的软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG4262

PERI_CRG4262 为 I2C3 时钟及复位控制寄存器。

Offset Address: 0x4298 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RW	i2c3_cksel	I2C3 时钟选择。 0: 50MHz; 1: 100MHz。	0x0
[11:5]	-	reserved	保留	0x00
[4]	RW	i2c3_cken	I2C3 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0



Bits	Access	Name	Description	Reset
[3:1]	-	reserved	保留	0x0
[0]	RW	i2c3_srst_req	I2C3 的软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG4264

PERI_CRG4264 为 I2C4 时钟及复位控制寄存器。

Offset Address: 0x42A0 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RW	i2c4_cksel	I2C4 时钟选择。 0: 50MHz; 1: 100MHz。	0x0
[11:5]	-	reserved	保留	0x00
[4]	RW	i2c4_cken	I2C4 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留	0x0
[0]	RW	i2c4_srst_req	I2C4 的软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG4266

PERI_CRG4266 为 I2C5 时钟及复位控制寄存器。

Offset Address: 0x42A8 Total Reset Value: 0x0000_0001



Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RW	i2c5_cksel	I2C5 时钟选择。 0: 50MHz; 1: 100MHz。	0x0
[11:5]	-	reserved	保留	0x00
[4]	RW	i2c5_cken	I2C5 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留	0x0
[0]	RW	i2c5_srst_req	I2C5 的软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG4268

PERI_CRG4268 为 I2C6 时钟及复位控制寄存器。

Offset Address: 0x42B0 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RW	i2c6_cksel	I2C6 时钟选择。 0: 50MHz; 1: 100MHz。	0x0
[11:5]	-	reserved	保留	0x00
[4]	RW	i2c6_cken	I2C6 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0



Bits	Access	Name	Description	Reset
[3:1]	-	reserved	保留	0x0
[0]	RW	i2c6_srst_req	I2C6 的软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG4270

PERI_CRG4270 为 I2C7 时钟及复位控制寄存器。

Offset Address: 0x42B8 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RW	i2c7_cksel	I2C7 时钟选择。 0: 50MHz; 1: 100MHz。	0x0
[11:5]	-	reserved	保留	0x00
[4]	RW	i2c7_cken	I2C7 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留	0x0
[0]	RW	i2c7_srst_req	I2C7 的软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG4320

PERI_CRG4320 为 SPI_TFT 时钟及复位控制寄存器。

Offset Address: 0x4380 Total Reset Value: 0x0000_0001



Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RW	tspi_cksel	SPI_TFT 工作时钟选择寄存器。 0: 24MHz; 1: 100MHz。	0x0
[11:5]	-	reserved	保留。	0x00
[4]	RW	tspi_cken	SPI_TFT 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	tspi_srst_req	SPI_TFT 的软复位请求。 0: 撤消复位; 1: 复位。	0x1

PERI_CRG4384

PERI_CRG4384 为 SPI0 时钟及复位控制寄存器。

Offset Address: 0x4480 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	ssp0_cken	SPI0 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	ssp0_srst_req	SPI0 的软复位请求。 0: 撤消复位; 1: 复位。	0x1



PERI_CRG4386

PERI_CRG4386 为 SPI1 时钟及复位控制寄存器。

Offset Address: 0x4488 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	ssp1_cken	SPI1 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	ssp1_srst_req	SPI1 的软复位请求。 0: 撤消复位; 1: 复位。	0x1

PERI_CRG4388

PERI_CRG4388 为 SPI2 时钟及复位控制寄存器。

Offset Address: 0x4490 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	ssp2_cken	SPI2 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	ssp2_srst_req	SPI2 的软复位请求。 0: 撤消复位; 1: 复位。	0x1



PERI_CRG4390

PERI_CRG4390 为 SPI3 时钟及复位控制寄存器。

Offset Address: 0x4498 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	ssp3_cken	SPI3 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	ssp3_srst_req	SPI3 的软复位请求。 0: 撤消复位; 1: 复位。	0x1

PERI_CRG4392

PERI_CRG4392 为 SPI 3WIRE 时钟及复位控制寄存器。

Offset Address: 0x44A0 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	ssp_3w_cken	SPI 3WIRE 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	ssp_3w_srst_req	SPI 3WIRE 软复位请求。 0: 撤消复位; 1: 复位。	0x1



PERI_CRG4448

PERI_CRG4448 为 SVB PWM 时钟软复位控制寄存器。

Offset Address: 0x4580 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13:12]	RW	svb_pwm_cksel	SVB PWM 时钟门控选择。 00: 3MHz; 01: 24MHz; 10: 50MHz; 11: 保留。	0x0
[11:5]	-	reserved	保留。	0x00
[4]	RW	svb_pwm_cken	SVB PWM 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	svb_pwm_srst_req	SVB PWM 软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG4450

PERI_CRG4450 为 PWM0 时钟软复位控制寄存器。

Offset Address: 0x4588 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:12]	RW	pwm0_cksel	PWM0 时钟选择。	0x0



Bits	Access	Name	Description	Reset
			0x0: 198MHz; 0x4: sensor0 clk; 0x5: sensor1 clk; 0x6: sensor2 clk; 0x7: sensor3 clk; 其他: 保留。	
[11:5]	-	reserved	保留。	0x00
[4]	RW	pwm0_cken	PWM0 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	pwm0_srst_req	PWM0 软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG4452

PERI_CRG4452 为 PWM1 时钟软复位控制寄存器。

Offset Address: 0x4590 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:12]	RW	pwm1_cksel	PWM1 时钟选择。 0x0: 198MHz; 0x4: sensor0 clk; 0x5: sensor1 clk; 0x6: sensor2 clk; 0x7: sensor3 clk;	0x0



Bits	Access	Name	Description	Reset
			其他：保留。	
[11:5]	-	reserved	保留。	0x00
[4]	RW	pwm1_cken	PWM1 时钟门控配置寄存器。 0：关闭时钟； 1：打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	pwm1_srst_req	PWM1 软复位请求。 0：不复位； 1：复位。	0x1

PERI_CRG4454

PERI_CRG4454 为 PWM2 时钟软复位控制寄存器。

Offset Address: 0x4598 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x00000000
[4]	RW	pwm2_cken	PWM2 时钟门控配置寄存器。 0：关闭时钟； 1：打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	pwm2_srst_req	PWM2 软复位请求。 0：不复位； 1：复位。	0x1

PERI_CRG4498

PERI_CRG4498 为 THERMO (热成像)时钟软复位控制寄存器。



Offset Address: 0x4648 Total Reset Value: 0x0020_000F

Bits	Access	Name	Description	Reset
[31:22]	-	reserved	保留。	0x000
[21]	RW	vi_ir_rx_pctrl	THERMO RX 方向时钟相位控制，默认反向。 0: 正向时钟; 1: 反向时钟。	0x1
[20]	RW	vi_ir_tx_pctrl	THERMO TX 方向时钟相位控制，默认正向。 0: 正向时钟; 1: 反向时钟。	0x0
[19:16]	-	reserved	保留。	0x0
[15:12]	RW	vi_ir_cksel	THERMO 时钟选择。 0x0: 4MHz; 0x1: 15MHz; 0x2: 22MHz; 0x3: 25MHz; 0x4: 30MHz; 0x5: 40MHz; 0x6: 45MHz; 其他: 保留。	0x0
[11:7]	-	reserved	保留。	0x00
[6]	RW	vi_ir_apb_cken	THERMO APB 总线时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[5]	RW	vi_ir_rx_cken	THERMO 控制器 RX 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0



Bits	Access	Name	Description	Reset
[4]	RW	vi_ir_tx_cken	THERMO 控制器 TX 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3]	RW	vi_ir_sensor_srst_req	THERMO sensor 软复位请求。 0: 不复位; 1: 复位。	0x1
[2]	RW	vi_ir_apb_srst_req	THERMO APB 总线时钟域软复位请求。 0: 不复位; 1: 复位。	0x1
[1]	RW	vi_ir_rx_srst_req	THERMO 控制器 RX 逻辑软复位请求。 0: 不复位; 1: 复位。	0x1
[0]	RW	vi_ir_tx_srst_req	THERMO 控制器 TX 逻辑软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG4528

PERI_CRG4528 为 LSADC 时钟软复位控制寄存器。

Offset Address: 0x46C0 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	lsadc_cken	LSADC 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	lsadc_srst_req	LSADC 软复位请求。	0x1



Bits	Access	Name	Description	Reset
			0: 不复位; 1: 复位。	

PERI_CRG8272

PERI_CRG8272 为 MIPI TX 时钟及软复位控制寄存器。

Offset Address: 0x8140 Total Reset Value: 0x0000_0002

Bits	Access	Name	Description	Reset
[31:24]	RW	mipitx_refdiv	MIPI TX PHY 参考高频时钟分频系数, 当 mipitx_ref_cksel 为 2'b10 时生效。 n: n+1 分频。	0x00
[23:6]	-	reserved	保留。	0x00000
[5]	RW	mipitx_div_cken	MIPITX REF 时钟的 DIV 分支时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[4]	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
[3:2]	RW	mipitx_ref_cksel	MIPITX PHY 参考时钟选择。 00: 19.2MHz; 01: 27MHz; 10: PIXEL 分频时钟; 11: 保留。	0x0
[1]	RW	mipitx_srst_req	MIPITX CTRL 软复位请求。 0: 不复位; 1: 复位。	0x1
[0]	RW	mipitx_cken	MIPITX 时钟门控。	0x0



Bits	Access	Name	Description	Reset
			0: 时钟关闭; 1: 时钟打开。	

PERI_CRG8274

PERI_CRG8274 为 MIPITX 工作时钟及软复位控制寄存器。

Offset Address: 0x8148 Total Reset Value: 0x0000_0101

Bits	Access	Name	Description	Reset
[31:25]	-	reserved	保留。	0x00
[24]	RW	mipitx_xtal_cken	MIPITX_XTAL 时钟门控。 0: 不复位; 1: 复位。	0x0
[23:21]	-	reserved	保留。	0x0
[20]	RW	mipitx_ad_ref_cken	MIPITX_AD_REF 时钟门控。 0: 不复位; 1: 复位。	0x0
[19:17]	-	reserved	保留。	0x0
[16]	RW	mipitx_ad_fb_cken	MIPITX_AD_FB 时钟门控。 0: 不复位; 1: 复位。	0x0
[15:13]	-	reserved	保留。	0x0
[12]	RW	mipitx_ad_ssc_cken	MIPITX_AD_SSC 时钟门控。 0: 不复位; 1: 复位。	0x0
[11:9]	-	reserved	保留。	0x0
[8]	RW	mipi_pcs_tx_srst_req	MIPI PCS TX 软复位请求。	0x1



Bits	Access	Name	Description	Reset
			0: 不复位; 1: 复位。	
[7:5]	-	reserved	保留。	0x0
[4]	RW	mipi_dpi_cken	MIPI TX dpi 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	mipitx_pcs_srst_req	MIPITX PCS 软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG8336

PERI_CRG8336 为 VDP 工作时钟及复位控制寄存器。

Offset Address: 0x8240 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x000000
[9]	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
[8]	RW	vdp_pcken	VDP APB 总线时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[7:1]	-	reserved	保留。	0x00
[0]	RW	vdp_srst_req	VDP 软复位请求。 0: 不复位; 1: 复位。	0x1



PERI_CRG8338

PERI_CRG8338 为 VDP BT(BT.656/BT.1120)输出时钟相位配置寄存器。

Offset Address: 0x8248 Total Reset Value: 0x0010_0000

Bits	Access	Name	Description	Reset
[31:21]	-	reserved	保留。	0x000
[20]	RW	voout_hd_pctrl	VDP 输出时钟(芯片输出时钟)相位控制。 0: 时钟不取反; 1: 时钟取反。	0x1
[19:0]	-	reserved	保留。	0x00000

PERI_CRG8340

PERI_CRG8340 为 VDP HD 时钟分频系数及时钟相位。

Offset Address: 0x8250 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:24]	RW	vdp_hd_div	VDP HD 输出随路时钟分频比。 00: 一分频; 01: 二分频; 10: 三分频; 11: 四分频。	0x0
[23:16]	-	reserved	保留。	0x00
[15:12]	RW	vdp_hd_cksel	VDP HD 时钟选择。 0x0: 66MHz; 0x1: 74.25MHz; 0x2: 88MHz;	0x0



Bits	Access	Name	Description	Reset
			0x3: 108MHz; 0x4: 135MHz; 0x5: 148.5MHz; 0x6: 297MHz; 0x7: RGB 分频器时钟; 其他: 保留。	
[11:7]	-	reserved	保留。	0x00
[6]	RW	vdp_ppc_hd_cfg_cken	VDP PPC CFG 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[5]	RW	vdp_ppc_hd_cken	VDP PPC 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[4]	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
[3:0]	-	reserved	保留。	0x0

PERI_CRG8341

PERI_CRG8341 为 VDP PPC 时钟选择控制寄存器。

Offset Address: 0x8254 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	保留。	0x0
[28:24]	RW	vdp_ppc_div	VDP PPC 时钟 CLK_DIV_ICG 分频比。 0x00: 1 分频; 0x01: 2 分频;	0x00



Bits	Access	Name	Description	Reset
			0x02: 3 分频; 0x03: 4 分频; 0x1f: 32 分频。	
[23:15]	-	reserved	保留。	0x000
[14:12]	RW	vdp_ppc_cksel	VDP PPC 时钟源头选择。 000: 50MHz; 001: 100MHz; 010: 150MHz; 011: 300MHz; 100: 339MHz; 其他: 保留。	0x0
[11:1]	-	reserved	保留。	0x000
[0]	RW	vdp_ppc_div_srst_req	VDP PPC 时钟 CLK_DIV_ICG 模块软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG8346

PERI_CRG8346 为 VDP RGB 时钟配置寄存器。

Offset Address: 0x8268 Total Reset Value: 0x0015_E4C3

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



Bits	Access	Name	Description	Reset
[26:0]	RW	lcd_mclk_div	RGB 时钟可配分频值。 假定目标频率 X(MHz;), 则 $lcd_mclk_div = (X/1188) * 2^{27}$ 。	0x015E4C3

PERI_CRG8348

PERI_CRG8348 为 VDP BT(BT.1120/BT.656)时钟门控及时钟选择控制寄存器。

Offset Address: 0x8270 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5]	RW	vdp_bt_cken	VDP BT 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[4]	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
[3:0]	-	reserved	保留。	0x0

PERI_CRG8352

PERI_CRG8352 为 MIPITX PIXEL 时钟控制寄存器。

Offset Address: 0x8280 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5]	RW	mipi_vdp_cken	VDP 的 MIPI 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0



Bits	Access	Name	Description	Reset
[4]	RW	mipi_pixel_cken	MIPI pixel 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:0]	-	reserved	保留。	0x0

PERI_CRG8464

PERI_CRG8464 为 SENSOR0 时钟复位配置寄存器。

Offset Address: 0x8440 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:12]	RW	sensor0_cksel	Sensor0 时钟(芯片输出给 sensor 的参考时钟)选择。 0x0: 74.25MHz; 0x1: 72MHz; 0x2: 54MHz; 0x3: 50MHz; 0x4: 24MHz; 0x8: 37.125MHz; 0x9: 36MHz; 0xA: 27MHz; 0xB: 25MHz; 0xC: 12MHz; 其他: 保留。	0x0
[11:5]	-	reserved	保留。	0x00
[4]	RW	sensor0_cken	Sensor0 时钟(芯片输出给 sensor 的参考时钟)门控。 0: 时钟关闭;	0x0



Bits	Access	Name	Description	Reset
			1: 时钟打开。	
[3:2]	-	reserved	保留。	0x0
[1]	RW	sensor0_ctrl_srst_req	Sensor0 从模式控制模块软复位请求。 0: 不复位; 1: 复位。	0x1
[0]	RW	sensor0_srst_req	Sensor0 软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG8472

PERI_CRG8472 为 SENSOR1 时钟复位配置寄存器。

Offset Address: 0x8460 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:12]	RW	sensor1_cksel	Sensor1 时钟(芯片输出给 sensor 的参考时钟)选择。 0x0: 74.25MHz; 0x1: 72MHz; 0x2: 54MHz; 0x3: 50MHz; 0x4: 24MHz; 0x8: 37.125MHz; 0x9: 36MHz; 0xA: 27MHz; 0xB: 25MHz; 0xC: 12MHz; 其他: 保留。	0x0



Bits	Access	Name	Description	Reset
[11:5]	-	reserved	保留。	0x00
[4]	RW	sensor1_cken	Sensor1 时钟(芯片输出给 sensor 的参考时钟)门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	RW	sensor1_ctrl_srst_req	Sensor1 从模式控制模块软复位请求。 0: 不复位; 1: 复位。	0x1
[0]	RW	sensor1_srst_req	Sensor1 软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG8480

PERI_CRG8480 为 SENSOR2 时钟复位配置寄存器。

Offset Address: 0x8480 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:12]	RW	sensor2_cksel	Sensor2 时钟(芯片输出给 sensor 的参考时钟)选择。 0x0: 74.25MHz; 0x1: 72MHz; 0x2: 54MHz; 0x3: 50MHz; 0x4: 24MHz; 0x8: 37.125MHz;	0x0



Bits	Access	Name	Description	Reset
			0x9: 36MHz; 0xA: 27MHz; 0xB: 25MHz; 0xC: 12MHz; 其他: 保留。	
[11:5]	-	reserved	保留。	0x00
[4]	RW	sensor2_cken	Sensor2 时钟(芯片输出给 sensor 的参考时钟)门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	RW	sensor2_ctrl_srst_req	sensor2 从模式控制模块软复位请求。 0: 不复位; 1: 复位。	0x1
[0]	RW	sensor2_srst_req	sensor2 软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG8488

PERI_CRG8488 为 SENSOR3 时钟复位配置寄存器。

Offset Address: 0x84A0 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:12]	RW	sensor3_cksel	Sensor3 时钟(芯片输出给 sensor 的参考时钟)选择。 0x0: 74.25MHz;	0x0



Bits	Access	Name	Description	Reset
			0x1: 72MHz; 0x2: 54MHz; 0x3: 50MHz; 0x4: 24MHz; 0x8: 37.125MHz; 0x9: 36MHz; 0xA: 27MHz; 0xB: 25MHz; 0xC: 12MHz; 其他: 保留。	
[11:5]	-	reserved	保留。	0x00
[4]	RW	sensor3_cken	Sensor3 时钟(芯片输出给 sensor 的参考时钟)门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	RW	sensor3_ctrl_srst_req	Sensor3 从模式控制模块软复位请求。 0: 不复位; 1: 复位。	0x1
[0]	RW	sensor3_srst_req	sensor3 软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG8528

PERI_CRG8528 为 MIPI RX 时钟复位配置寄存器。

Offset Address: 0x8540 Total Reset Value: 0x0000_0001



Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23]	RW	mipi_hs3_ctrl	MIPIPHY HS0 接口时钟相位控制，默认正向。 0: 时钟不取反; 1: 时钟取反。	0x0
[22]	RW	mipi_hs2_ctrl	MIPIPHY HS1 接口时钟相位控制，默认正向。 0: 时钟不取反; 1: 时钟取反。	0x0
[21]	RW	mipi_hs1_ctrl	MIPIPHY HS2 接口时钟相位控制，默认正向。 0: 时钟不取反; 1: 时钟取反。	0x0
[20]	RW	mipi_hs0_ctrl	MIPIPHY HS3 接口时钟相位控制，默认正向。 0: 时钟不取反; 1: 时钟取反。	0x0
[19:6]	-	reserved	保留。	0x0000
[5]	RW	mipi_bus_cken	MIPI 总线逻辑时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[4]	RW	cil_cken	MIPI CIL 逻辑时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3]	RW	cbar_cken	MIPI CBAR 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0



Bits	Access	Name	Description	Reset
[2]	RW	cal_cken	MIPI CAL 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[1]	-	reserved	保留。	0x0
[0]	RW	mipi_bus_srst_req	MIPI 总线逻辑软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG8536

PERI_CRG8536 为 MIPI RX PIX0 时钟复位配置寄存器。

Offset Address: 0x8560 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	mipi_pix0_cken	MIPI PIX0 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	mipi_pix0_core_srst_req	MIPI PIX0 的 CORE 逻辑软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG8544

PERI_CRG8544 为 MIPI_RX PIX1 时钟复位配置寄存器。

Offset Address: 0x8580 Total Reset Value: 0x0000_0001



Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	mipi_pix1_cken	MIPI PIX1 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	mipi_pix1_core_srst_req	MIPI PIX1 的 CORE 逻辑软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG8552

PERI_CRG8552 为 MIPI_RX PIX2 时钟复位配置寄存器。

Offset Address: 0x85A0 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	mipi_pix2_cken	MIPI PIX2 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	mipi_pix2_core_srst_req	MIPI PIX2 的 CORE 逻辑软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG8560

PERI_CRG8560 为 MIPI_RX PIX3 时钟复位配置寄存器。

Offset Address: 0x85C0 Total Reset Value: 0x0000_0001



Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	mipi_pix3_cken	MIPI PIX3 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	mipi_pix3_core_srst_req	MIPI PIX3 的 CORE 逻辑软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG9296

PERI_CRG9296 为 VICAP 时钟及复位控制寄存器。

Offset Address: 0x9140 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:12]	RW	vi_ppc_cksel	VICAP 工作时钟选择。 000: 150MHz; 001: 198MHz; 010: 300MHz; 011: 396MHz; 100: 475MHz; 101: 540MHz; 110: 594MHz; 111: 保留。	0x0
[11:6]	-	reserved	保留。	0x00
[5]	RW	vi_bus_cken	VICAP BUS 时钟门控。 0: 时钟关闭;	0x0



Bits	Access	Name	Description	Reset
			1: 时钟打开。	
[4]	RW	vi_ppc_cken	VICAP PPC 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	RW	vi_bus_srst_req	VICAP BUS 软复位请求。 0: 不复位; 1: 复位。	0x1
[0]	-	reserved	保留。	0x1

PERI_CRG9297

PERI_CRG9297 为 VICAP 复位状态寄存器。

Offset Address: 0x9144 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留	0x00000000
[0]	RO	vicap_softrst_state	VICAP 复位状态。 0: 复位未完成; 1: 复位完成, 可以撤销软复位。	0x0

PERI_CRG9298

PERI_CRG9298 为 VI CH(Channel)和 PT(Port)时钟控制寄存器。

Offset Address: 0x9148 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14]	RW	vi_ppc_ch6_cken	VI PPC CH6 时钟门控。	0x0



Bits	Access	Name	Description	Reset
			0: 时钟关闭; 1: 时钟打开。	
[13]	RW	vi_ppc_ch5_cken	VI PPC CH5 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[12]	RW	vi_ppc_ch4_cken	VI PPC CH4 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[11]	RW	vi_ppc_ch3_cken	VI PPC CH3 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[10]	RW	vi_ppc_ch2_cken	VI PPC CH2 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[9]	RW	vi_ppc_ch1_cken	VI PPC CH1 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[8]	RW	vi_ppc_ch0_cken	VI PPC CH0 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[7]	RW	vi_ppc_pt3_cken	VI PPC PT3 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[6]	RW	vi_ppc_pt2_cken	VI PPC PT2 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[5]	RW	vi_ppc_pt1_cken	VI PPC PT1 时钟门控。	0x0



Bits	Access	Name	Description	Reset
			0: 时钟关闭; 1: 时钟打开。	
[4]	RW	vi_ppc_pt0_cken	VI PPC PT0 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:0]	-	reserved	保留。	0x0

PERI_CRG9300

PERI_CRG9300 为 VI ISP0 时钟及复位控制寄存器。

Offset Address: 0x9150 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:12]	RW	isp0_cksel	ISP0 时钟选择。 000: 150MHz; 001: 198MHz; 010: 300MHz; 011: 396MHz; 100: 475MHz; 101: 500MHz; 110: 540MHz; 111: 594MHz。	0x0
[11:5]	-	reserved	保留。	0x00
[4]	RW	isp0_cken	ISP0 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:2]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[1]	RW	isp0_core_srst_req	ISP0 core 软复位请求。 0: 不复位; 1: 复位。	0x1
[0]	RW	isp0_cfg_srst_req	ISP0 cfg 软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG9301

PERI_CRG9301 为 VI ISP1 时钟及复位控制寄存器。

Offset Address: 0x9154 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	isp1_cken	ISP1 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	RW	isp1_core_srst_req	ISP1 core 软复位请求。 0: 不复位; 1: 复位。	0x1
[0]	RW	isp1_cfg_srst_req	ISP1 cfg 软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG9302

PERI_CRG9302 为 VI ISP2 时钟及复位控制寄存器。



Offset Address: 0x9158 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x00000000
[4]	RW	isp2_cken	ISP2 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	RW	isp2_core_srst_req	ISP2 core 软复位请求。 0: 不复位; 1: 复位。	0x1
[0]	RW	isp2_cfg_srst_req	ISP2 cfg 软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG9303

PERI_CRG9303 为 VI ISP3 时钟及复位控制寄存器。

Offset Address: 0x915C Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x00000000
[4]	RW	isp3_cken	ISP3 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	RW	isp3_core_srst_req	ISP3 core 软复位请求。 0: 不复位; 1: 复位。	0x1



Bits	Access	Name	Description	Reset
[0]	RW	isp3_cfg_srst_req	ISP3 cfg 软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG9304

PERI_CRG9304 为 VI COMS0 时钟及复位控制寄存器。

Offset Address: 0x9160 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:21]	-	reserved	保留。	0x000
[20]	RW	vi_cmos0_pctrl	VI CMOS 时钟相位控制。 0: 时钟不取反; 1: 时钟取反。	0x0
[19:0]	-	reserved	保留。	0x00000

PERI_CRG9305

PERI_CRG9305 为 VI PORT0 时钟及复位控制寄存器。

Offset Address: 0x9164 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:12]	RW	vi_p0_cksel	VI PORT0 时钟选择。 000: 100MHz; 001: 150MHz; 010: 198MHz; 011: 237MHz; 100: 300MHz;	0x0



Bits	Access	Name	Description	Reset
			101: 339MHz; 110: 475MHz; 111: 594MHz。	
[11:5]	-	reserved	保留。	0x00
[4]	RW	vi_p0_cken	VI PORT0 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	vi_p0_srst_req	VI PORT0 软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG9313

PERI_CRG9313 为 VI PORT1 时钟及复位控制寄存器。

Offset Address: 0x9184 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:12]	RW	vi_p1_cksel	VI PORT1 时钟选择。 000: 100MHz; 001: 150MHz; 010: 198MHz; 011: 237MHz; 100: 300MHz; 101: VI_PORT1(由 MISC_CTRL 的 vi_port1_input_sel 选择为 CMOS 时钟或 THERMO 时钟); 110: 475MHz;	0x0



Bits	Access	Name	Description	Reset
			111: 594MHz。	
[11:5]	-	reserved	保留。	0x00
[4]	RW	vi_p1_cken	VI PORT1 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	vi_p1_srst_req	VI PORT1 软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG9321

PERI_CRG9321 为 VI PORT2 时钟及复位控制寄存器。

Offset Address: 0x91A4 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:12]	RW	vi_p2_cksel	VI PORT2 时钟选择。 000: 100MHz; 001: 150MHz; 010: 198MHz; 011: 237MHz; 100: 300MHz; 101: 339MHz; 110: 475MHz; 111: 594MHz。	0x0
[11:5]	-	reserved	保留。	0x00
[4]	RW	vi_p2_cken	VI PORT2 时钟门控。	0x0



Bits	Access	Name	Description	Reset
			0: 时钟关闭; 1: 时钟打开。	
[3:1]	-	reserved	保留。	0x0
[0]	RW	vi_p2_srst_req	VI PORT2 软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG9329

PERI_CRG9329 为 VI PORT3 时钟及复位控制寄存器。

Offset Address: 0x91C4 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:12]	RW	vi_p3_cksel	VI PORT3 时钟选择。 000: 100MHz; 001: 150MHz; 010: 198MHz; 011: 237MHz; 100: 300MHz; 101: cmos 时钟; 110: 475MHz; 111: 594MHz。	0x0
[11:5]	-	reserved	保留。	0x00
[4]	RW	vi_p3_cken	VI PORT3 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:1]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[0]	RW	vi_p3_srst_req	VI PORT3 软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG10784

PERI_CRG10784 为 AIAO 时钟复位控制寄存器。

Offset Address: 0xA880 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13]	RW	aiao_work_cksel	AIAO 控制器工作时钟选择。 0: 50MHz; 1: 100MHz。	0x0
[12]	RW	aiao_pll_cksel	AIAO 时钟选择配置寄存器。 0: 786.432MHz; 1: 1188MHz。	0x0
[11:6]	-	reserved	保留。	0x00
[5]	RW	aiao_cken	AIAO 时钟门控配置寄存器。 0: 关闭时钟; 1: 打开时钟。	0x0
[4]	RW	aiao_pll_cken	AIAO PLL 时钟门控。 0: 关闭时钟; 1: 打开时钟。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	aiao_srst_req	AIAO 软复位请求。 0: 撤消复位; 1: 复位。	0x1



PERI_CRG10785

PERI_CRG10785 为 AIAO DMIC 时钟复位配置寄存器。

Offset Address: 0xA884 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	aiao_dmic_cken	AIAO_DMIC 时钟门控。 0: 时钟关闭; 1: 时钟打开。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	dmic_srst_req	AIAO_DMIC 软复位请求。 0: 不复位; 1: 复位。	0x1

PERI_CRG10912

PERI_CRG10912 为 AUDIO CODEC 时钟复位控制寄存器。

Offset Address: 0xAA80 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	codec_ana_rst_req	AUDIO COEDC 模拟部分复位请求(复位时间持续至少 0.5us)。 0: 撤消复位; 1: 复位。	0x1
[0]	RW	codec_dig_rst_req	AUDIO COEDC 数字逻辑复位请求(复位时间持续至少 0.5us)。 0: 撤消复位; 1: 复位。	0x1



3.3 处理器子系统

该芯片采用 ARM Cortex-A55 双核处理器，具有以下特点：

- 处理器工作频率 1000MHz。
- 集成多媒体加速引擎 NEON 和硬件浮点协处理器。
- L1 Cache 包含 32KB Instruction Cache 和 32KB Data Cache。
- 不含 L2 Cache，双核共享 256KB L3 Cache
- 支持 MMU (Memory Management Unit) 。
- 支持 JTAG 调试接口。
- 集成中断控制器 GIC (Generic Interrupt Controller)，支持 320 个中断源。

3.4 中断系统

芯片主 CPU 支持 170 个外部中断源，对应的中断映射如表 3-6 所示。

表3-6 中断源分配表

中断向量位	中断源	中断向量位	中断源
0~31	A55 内部使用	101	KLAD_REE
32	SOFTWARE	102	SEC_ALARM
33	SDK_SOFTWARE	103	PRES_ALARM
34	ALL_PLL_LOCK	104	MIPI_RX
35	WDG	105	VICAP_INT0
36	TSSENSOR	106	VICAP_INT1
37	UART0	107	VIPROC
38	UART1	108	保留
39	UART2	109	VPSS_INT0
40	UART3	110	VPSS_INT1



中断向量位	中断源	中断向量位	中断源
41	UART4	111	保留
42	UART5	112	VDP_INT0
43	TIMER01	113	VDP_INT1
44	TIMER23	114	AIAO
45	TIMER45	115	MIPI_TX
46	TIMER67	116	保留
47	SEC_TIMER01	117	NPU_NS
48	SEC_TIMER23	118	NPU_S
49	nclusterpmuirq_n	119	NPU_INT0
50	I2C0	120	NPU_INT1
51	I2C1	121	保留
52	I2C2	122	保留
53	I2C3	123	DPU_RECT_INT0
54	I2C4	124	DPU_MATCH_INT0
55	I2C5	125	DPU_RECT_INT1
56	I2C6	126	DPU_MATCH_INT1
57	I2C7	127	IVE_INT0
58	SPI0	128	IVE_INT1
59	SPI1	129	保留
60	SPI2	130	保留
61	SPI3	131	保留
62	SPI_TFT	132	SCD
63	SPI_3WIRE	133	VEDU_INT0
64	GPIO0	134	VEDU_INT1
65	GPIO1	135	JPGE
66	GPIO2	136	JPGD_INT0



中断向量位	中断源	中断向量位	中断源
67	GPIO3	137	JPGD_INT1
68	GPIO4	138	VGS
69	GPIO5	139	TDE
70	GPIO6	140	GZIP
71	GPIO7	141	保留
72	GPIO8	142	DDRT
73	GPIO9	143	DDRC_ERR
74	GPIO10	144	DDRC_SEC
75	GPIO11	145	保留
76	GPIO12	146	保留
77	GPIO13	147	SOFTWARE1
78	GPIO14	148	SDIO0_PWR_SW_OCP_INT
79	LSADC	149	保留
80	CRC	150	保留
81	RTC	151	保留
82	eMMC	152	保留
83	SDIO0	153	保留
84	SDIO1	154	保留
85	FMC	155	保留
86	GSF_INT0	156	保留
87	GSF_INT1	157	保留
88	GSF_INT2	158	保留
89	GSF_INT3	159	保留
90	DMA	160	保留
91	DMA_NS	161	保留



中断向量位	中断源	中断向量位	中断源
92	USB3_CTRL	162	保留
93	USB3_TRACE_INT	163	保留
94	SPACC_TEE	164	保留
95	SPACC_REE	165	保留
96	PKE_TEE	166	保留
97	PKE_REE	167	保留
98	RKP_TEE	168	保留
99	RKP_REE	169	保留
100	KLAD_TEE		

3.5 系统控制器

3.5.1 概述

系统控制器管理系统中的重要功能，完成对外设的某些功能的配置。

3.5.2 特点

系统控制器具有以下特点：

- 提供对系统地址重映射的控制和状态监控
- 提供通用外设寄存器
- 提供对关键寄存器的写保护功能
- 提供芯片的标识寄存器



3.5.3 功能描述

3.5.3.1 软复位控制

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，系统控制器将给片内复位模块发起请求，芯片将被复位。

3.5.3.2 系统地址重映射控制

请参见“1.5 地址空间映射”章节。

3.5.3.3 对关键寄存器的写保护

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，系统控制器提供了对系统软复位寄存器 (SC_SYSRES) 的写保护功能。对该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，保护这些关键寄存器不会被软件随意改写。

说明

系统默认为复位后不对这些关键寄存器进行写保护处理。为启用此功能，建议在系统启动时利用 SC_LOCKEN 对这些关键寄存器进行写保护处理。

3.5.3.4 芯片的标识寄存器

系统控制器提供了芯片标识 (ID) 寄存器 CHIP_ID，软件可以通过 CHIP_ID 来识别芯片型号。

3.5.4 系统控制器寄存器

3.5.4.1 系统控制器寄存器概览

系统控制器寄存器概览如表 3-1 所示。

表3-1 系统控制器(SYS CTRL)寄存器概览 (基址: 0x0_1102_0000)

偏移地址	名称	描述	页码
0x0004	SC_SYSRES	系统软复位寄存器	3-109
0x0014	SELF_BOOT_FLAG	系统自举控制寄存器	3-109
0x0018	SYSSTAT	系统状态寄存器	3-110



偏移地址	名称	描述	页码
0x001C	SOFTINT	软中断寄存器	3-111
0x0020	SOFTTYPE	软中断向量寄存器	3-111
0x0024	SOFTINT1	软中断寄存器 1	3-112
0x0028	SOFTTYPE1	软中断向量寄存器 1	3-112
0x0044	SC_LOCKEN	关键系统控制寄存器的锁定寄存器	3-112
0x02C0	TEE_REGISTER0	TEE 寄存器 0	3-113
0x02C4	TEE_REGISTER1	TEE 寄存器 1	3-113
0x02C8	TEE_REGISTER2	TEE 寄存器 2	3-113
0x02CC	TEE_REGISTER3	TEE 寄存器 3	3-114
0x02D0	TEE_REGISTER4	TEE 寄存器 4	3-114
0x02D4	TEE_REGISTER5	TEE 寄存器 5	3-114
0x02D8	TEE_REGISTER6	TEE 寄存器 6	3-114
0x02DC	TEE_REGISTER7	TEE 寄存器 7	3-115
0x02E0	TEE_REGISTER8	TEE 寄存器 8	3-115
0x02E4	TEE_REGISTER9	TEE 寄存器 9	3-115
0x02E8	TEE_REGISTER10	TEE 寄存器 10	3-115
0x02EC	TEE_REGISTER11	TEE 寄存器 11	3-116
0x02F0	TEE_REGISTER12	TEE 寄存器 12	3-116
0x02F4	TEE_REGISTER13	TEE 寄存器 13	3-116
0x02F8	TEE_REGISTER14	TEE 寄存器 14	3-116
0x02FC	TEE_REGISTER15	TEE 寄存器 15	3-117
0x0360	WDG_RST_CNT	WATCH DOG 复位统计寄存器	3-117
0x0364	SOFTRST_CNT	软复位统计寄存器	3-117



偏移地址	名称	描述	页码
0x0EE0	CHIP_ID	芯片 ID 寄存器	3-117
0x0EEC	VENDOR_ID	厂商 ID 寄存器	3-118
0x1030	SDK_SOFTINT	软中断寄存器	3-118
0x1034	SDK_SOFTTYPE	软中断向量寄存器	3-118
0x1100	CUSTOMER_ID0	CUSTOMER_ID0 寄存器	3-118
0x1104	CUSTOMER_ID1	CUSTOMER_ID1 寄存器	3-119
0x1108	CUSTOMER_ID2	CUSTOMER_ID2 寄存器	3-119
0x110C	CUSTOMER_ID3	CUSTOMER_ID3 寄存器	3-119
0x1200	CHIP_UNIQUE_ID0	CHIP_UNIQUE_ID0 寄存器	3-120
0x1204	CHIP_UNIQUE_ID1	CHIP_UNIQUE_ID1 寄存器	3-120
0x1208	CHIP_UNIQUE_ID2	CHIP_UNIQUE_ID2 寄存器	3-120
0x120C	CHIP_UNIQUE_ID3	CHIP_UNIQUE_ID3 寄存器	3-120
0x1210	CHIP_UNIQUE_ID4	CHIP_UNIQUE_ID4 寄存器	3-121
0x1214	CHIP_UNIQUE_ID5	CHIP_UNIQUE_ID5 寄存器	3-121
0x1300	USERREG0	USER 专用寄存器 0	3-121
0x1304	USERREG1	USER 专用寄存器 1	3-122
0x1308	USERREG2	USER 专用寄存器 2	3-122
0x130C	USERREG3	USER 专用寄存器 3	3-122
0x1310	USERREG4	USER 专用寄存器 4	3-122
0x1314	USERREG5	USER 专用寄存器 5	3-123
0x1318	USERREG6	USER 专用寄存器 6	3-123
0x131C	USERREG7	USER 专用寄存器 7	3-123



3.5.4.2 系统控制器寄存器描述

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，使复位模块进行系统软复位。

须知

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

SELF_BOOT_FLAG

SELF_BOOT_FLAG 为系统自举控制寄存器。

Offset Address: 0x0014 Total Reset Value: 0X0000_0000

Bits	Access	Name	Description	Reset
[31:8]	RO	softresreq	保留。	0x00000
[7:0]	RO	self_boot_flag	系统自举方式选择。 0x00: 正常 FLASH 启动; 0x01: UART 非裸烧; 0x04: SD Card 非裸烧; 0x10: UART 裸烧; 0x11: UART 裸烧 to DDR; 0x20: USB Device 裸烧; 0x40: SD Card 裸烧; 其它: 保留。	0x00



SYSSTAT

SYSSTAT 为系统状态寄存器。

Offset Address: 0x0018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30]	RO	boot_para_sel3	SFC 器件工作电压和速率选择。 0: SFC 3.3V, 最高支持 99MHz; 1: SFC1.8V, 最高支持 75MHz。	0x0
[29:27]	RO	boot_para_sel	BOOT 参数表格选择。 0x0~0x7: 分别代表 8 种 BOOT 参数表格 0-7。	0x0
[26:16]	RO	reserved	保留。	0x0
[15]	RO	por_enable	复位方案选择。 0: 外部复位; 1: 内部 POR 复位。	0x0
[14:12]	-	reserved	保留。	0x0
[11]	RO	sfc_emmc_boot_mode	当从 SPI NOR FLASH 启动时, 表示 SPI NOR FLASH 启动地址模式选择。 0: 3Byte 地址模式; 1: 4Byte 地址模式。 当从 SPI NAND FLASH 启动时, 表示 SPI NAND FLASH 启动模式选择。 0: 1 线启动模式; 1: 4 线启动模式。 当从 eMMC 启动时, 表示 eMMC 启动模式选择。 0: 4Bit 启动模式;	0x0



Bits	Access	Name	Description	Reset
			1: 8Bit 启动模式。	
[10:5]	-	reserved	保留。	0x00
[4]	RO	fast_boot_mode	BOOT 模式选择。 0: Normal Flash 启动模式; 1: 串口烧写模式; Normal Flash 启动模式和串口烧写模式下 根据 boot_sel[1:0]选择启动/烧写介质。	0x0
[3:2]	RO	boot_sel	启动介质选择。 00: 从 SPI Nor Flash 启动; 01: 从 SPI Nand Flash 启动; 11: 从 eMMC 启动; 其他: 保留。	0x0
[1:0]	-	reserved	保留。	0x0

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]	RW	software_int_vector	软件中断向量。	0x00000000

SOFTINT1

SOFTINT1 为软中断寄存器 1。

Offset Address: 0x0024 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留	0x00000000
[0]	RW	software1_int	向此位写入 1 产生软件中断。	0x0

SOFTTYPE1

SOFTTYPE1 为软中断向量寄存器 1。

Offset Address: 0x0028 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	software1_int_vector	软件中断向量 1。	0x00000000

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	系统软复位寄存器 (SC_SYSRES) 的锁定寄存器。向该寄存器写入 0x1ACC_E551, 可打开系统软复位寄存器的写权限, 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	0x00000000



Bits	Access	Name	Description	Reset
			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)。	

TEE_REGISTER0

TEE_REGISTER0 为 TEE 寄存器 0。(安全寄存器, 仅支持安全访问)

Offset Address: 0x02C0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tee_register0	TEE 寄存器 0。	0x00000000

TEE_REGISTER1

TEE_REGISTER1 为 TEE 寄存器 1。(安全寄存器, 仅支持安全访问)

Offset Address: 0x02C4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tee_register1	TEE 寄存器 1。	0x00000000

TEE_REGISTER2

TEE_REGISTER2 为 TEE 寄存器 2。(安全寄存器, 仅支持安全访问)

Offset Address: 0x02C8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tee_register2	TEE 寄存器 2。	0x00000000



TEE_REGISTER3

TEE_REGISTER3 为 TEE 寄存器 3。(安全寄存器, 仅支持安全访问)

Offset Address: 0x02CC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tee_register3	TEE 寄存器 3。	0x00000000

TEE_REGISTER4

TEE_REGISTER4 为 TEE 寄存器 4。(安全寄存器, 仅支持安全访问)

Offset Address: 0x02D0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tee_register4	TEE 寄存器 4。	0x00000000

TEE_REGISTER5

TEE_REGISTER5 为 TEE 寄存器 5。(安全寄存器, 仅支持安全访问)

Offset Address: 0x02D4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW_SEC	tee_register5	TEE 寄存器 5。	0x00000000

TEE_REGISTER6

TEE_REGISTER6 为 TEE 寄存器 6。(安全寄存器, 仅支持安全访问)

Offset Address: 0x02D8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tee_register6	TEE 寄存器 6。	0x00000000



TEE_REGISTER7

TEE_REGISTER7 为 TEE 寄存器 7。(安全寄存器, 仅支持安全访问)

Offset Address: 0x02DC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tee_register7	TEE 寄存器 7。	0x00000000

TEE_REGISTER8

TEE_REGISTER8 为 TEE 寄存器 8。(安全寄存器, 仅支持安全访问)

Offset Address: 0x02E0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tee_register8	TEE 寄存器 8。	0x00000000

TEE_REGISTER9

TEE_REGISTER9 为 TEE 寄存器 9。(安全寄存器, 仅支持安全访问)

Offset Address: 0x02E4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tee_register9	TEE 寄存器 9。	0x00000000

TEE_REGISTER10

TEE_REGISTER10 为 TEE 寄存器 10。(安全寄存器, 仅支持安全访问)

Offset Address: 0x02E8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tee_register10	TEE 寄存器 10。	0x00000000



TEE_REGISTER11

TEE_REGISTER11 为 TEE 寄存器 11。(安全寄存器, 仅支持安全访问)

Offset Address: 0x02EC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tee_register11	TEE 寄存器 11。	0x00000000

TEE_REGISTER12

TEE_REGISTER12 为 TEE 寄存器 12。(安全寄存器, 仅支持安全访问)

Offset Address: 0x02F0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tee_register12	TEE 寄存器 12。	0x00000000

TEE_REGISTER13

TEE_REGISTER13 为 TEE 寄存器 13。(安全寄存器, 仅支持安全访问)

Offset Address: 0x02F4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tee_register13	TEE 寄存器 13。	0x00000000

TEE_REGISTER14

TEE_REGISTER14 为 TEE 寄存器 14。(安全寄存器, 仅支持安全访问)

Offset Address: 0x02F8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tee_register14	TEE 寄存器 14。	0x00000000



TEE_REGISTER15

TEE_REGISTER15 为 TEE 寄存器 15。(安全寄存器, 仅支持安全访问)

Offset Address: 0x02FC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tee_register15	TEE 寄存器 15。	0x00000000

WDG_RST_CNT

WDG_RST_CNT 为 WATCHDOG 复位统计寄存器。

Offset Address: 0x0360 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	wdg_rst_cnt	WATCH DOG 复位统计寄存器,该寄存器值仅被上电或者外部输入复位信号清零。	0x00000000

SOFRST_CNT

SOFRST_CNT 为软复位统计寄存器。

Offset Address: 0x0364 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	sofrst_cnt	软复位统计寄存器,该寄存器值仅被上电或者外部输入复位信号清零。	0x00000000

CHIP_ID

CHIP_ID 为芯片 ID 寄存器。

Offset Address: 0x0EE0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	chip_id	芯片 ID 寄存器, 为 OTP 回读值。	0x00000000



VENDOR_ID

VENDOR_ID 为厂商 ID 寄存器。

Offset Address: 0x0EEC Total Reset Value: 0x0000_0035

Bits	Access	Name	Description	Reset
[31:0]	RO	vendor_id	厂商 ID 寄存器, 固定为 0x35。	0x00000035

SDK_SOFTINT

SDK_SOFTINT 为软中断寄存器。

Offset Address: 0x1030 Total Reset Value: 0x0000_0000

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

SDK_SOFTTYPE

SDK_SOFTTYPE 为软中断向量寄存器。

Offset Address: 0x1034 Total Reset Value: 0x0000_0000

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

CUSTOMER_ID0

CUSTOMER_ID0 为 CUSTOMER_ID0 寄存器。

Offset Address: 0x1100 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	customer_id0	CUSTOMER_ID0 至 CUSTOMER_ID3 拼接, bit[127:0]组合。	0x00000000



Bits	Access	Name	Description	Reset
			CUSTOMER_ID[31:0]	

CUSTOMER_ID1

CUSTOMER_ID1 为 CUSTOMER_ID1 寄存器。

Offset Address: 0x1104 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	customer_id1	CUSTOMER_ID0 至 CUSTOMER_ID3 拼接, bit[127:0]组合。 CUSTOMER_ID[63:32]	0x00000000

CUSTOMER_ID2

CUSTOMER_ID2 为 CUSTOMER_ID2 寄存器。

Offset Address: 0x1108 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	customer_id2	CUSTOMER_ID0 至 CUSTOMER_ID3 拼接, bit[127:0]组合。 CUSTOMER_ID[95:64]	0x00000000

CUSTOMER_ID3

CUSTOMER_ID3 为 CUSTOMER_ID3 寄存器。

Offset Address: 0x110C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	customer_id3	CUSTOMER_ID0 至 CUSTOMER_ID3 拼接, bit[127:0]组合。 CUSTOMER_ID[127:96]	0x00000000



CHIP_UNIQUE_ID0

CHIP_UNIQUE_ID0 为 CHIP_UNIQUE_ID0 寄存器。

Offset Address: 0x1200 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	chip_unique_id0	CHIP_UNIQUE_ID0 至 CHIP_UNIQUE_ID5 拼接, bit[191:0]组合。 CHIP_UNIQUE_ID[31:0]	0x00000000

CHIP_UNIQUE_ID1

CHIP_UNIQUE_ID1 为 CHIP_UNIQUE_ID1 寄存器。

Offset Address: 0x1204 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	chip_unique_id1	CHIP_UNIQUE_ID0 至 CHIP_UNIQUE_ID5 拼接, bit[191:0]组合。 CHIP_UNIQUE_ID[63:32]	0x00000000

CHIP_UNIQUE_ID2

CHIP_UNIQUE_ID2 为 CHIP_UNIQUE_ID2 寄存器。

Offset Address: 0x1208 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	chip_unique_id2	CHIP_UNIQUE_ID0 至 CHIP_UNIQUE_ID5 拼接, bit[191:0]组合。 CHIP_UNIQUE_ID[95:64]	0x00000000

CHIP_UNIQUE_ID3

CHIP_UNIQUE_ID3 为 CHIP_UNIQUE_ID3 寄存器。

Offset Address: 0x120C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RO	chip_unique_id3	CHIP_UNIQUE_ID0 至 CHIP_UNIQUE_ID5 拼接, bit[191:0]组合。 CHIP_UNIQUE_ID[127:96]	0x00000000

CHIP_UNIQUE_ID4

CHIP_UNIQUE_ID4 为 CHIP_UNIQUE_ID4 寄存器。

Offset Address: 0x1210 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	chip_unique_id4	CHIP_UNIQUE_ID0 至 CHIP_UNIQUE_ID5 拼接, bit[191:0]组合。 CHIP_UNIQUE_ID[159:128]	0x00000000

CHIP_UNIQUE_ID5

CHIP_UNIQUE_ID5 为 CHIP_UNIQUE_ID5 寄存器。

Offset Address: 0x1214 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	chip_unique_id5	CHIP_UNIQUE_ID0 至 CHIP_UNIQUE_ID5 拼接, bit[191:0]组合。 CHIP_UNIQUE_ID[191:160]	0x00000000

USERREG0

USERREG0 为 USER 专用寄存器 0。该寄存器不会被系统软复位所复位。

Offset Address: 0x1300 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg0	用户专用寄存器 0。	0x00000000



USERREG1

USERREG1 为 USER 专用寄存器 1。该寄存器不会被系统软复位所复位。

Offset Address: 0x1304 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg1	用户专用寄存器 1。	0x00000000

USERREG2

USERREG2 为 USER 专用寄存器 2。该寄存器不会被系统软复位所复位。

Offset Address: 0x1308 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg2	用户专用寄存器 2。	0x00000000

USERREG3

USERREG3 为 USER 专用寄存器 3。该寄存器不会被系统软复位所复位。

Offset Address: 0x130C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg3	用户专用寄存器 3。	0x00000000

USERREG4

USERREG4 为 USER 专用寄存器 4。

Offset Address: 0x1310 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg4	用户专用寄存器 4。	0x00000000



USERREG5

USERREG5 为 USER 专用寄存器 5。

Offset Address: 0x1314 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg5	用户专用寄存器 5。	0x00000000

USERREG6

USERREG6 为 USER 专用寄存器 6。

Offset Address: 0x1318 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg6	用户专用寄存器 6。	0x00000000

USERREG7

USERREG7 为 USER 专用寄存器 7。该寄存器不会被系统软复位所复位。

Offset Address: 0x131C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user	用户专用寄存器 7。	0x00000000

3.5.5 SOC MISC 寄存器

3.5.5.1 SOC MISC 寄存器概览

SOC MISC 寄存器概览如表 3-4 所示。

图3-5 SOC MISC 寄存器 (基址是 0x0_1102_4000)

偏移地址	名称	描述	页码
0x4204	MISC_CTRL1	CPU 64bit 启动地址寄存器	3-125



偏移地址	名称	描述	页码
0x4218	DDRCA_REE_RANDOM_L	DDR CA 功能 REE 随机值低 32bit 值	3-125
0x421C	DDRCA_REE_RANDOM_H	DDR CA 功能 REE 随机值高 32bit 值	3-126
0x4220	DDRCA_EN_REE	DDR_CA 功能模式控制寄存器	3-126
0x4224	DDRCA_REE_UPDATE	DDR_CA 功能 REE 使能寄存器	3-126
0x4228	DDRCA_LOCK_REE	DDR_CA LOCK 寄存器	3-127
0x4230	DDRC_LOCK_CTRL1	DDRC LOCK 寄存器 1	3-127
0x4234	DDRC_LOCK_CTRL2	DDRC LOCK 寄存器 2	3-127
0x4238	DDRC_LOCK_CTRL3	DDRC LOCK 寄存器 3	3-128
0x423C	DDRC_LOCK_CTRL4	DDRC LOCK 寄存器 4	3-128
0x4240	DDRC_LOCK_CTRL5	DDRC LOCK 寄存器 5	3-128
0x4610	SPI_CFG3	SPI2/3 片选信号 (CS) 极性配置	3-129
0x4614	I2C_MONTAGE_CFG0	I2C 蒙太奇功能使能配置寄存器	3-129
0x461C	I2C_MONTAGE_CFG2	I2C 蒙太奇读写选择寄存器	3-130
0x4620	I2C0_I2C6_DMA_SELECT	I2C0/I2C6 DMA 请求线选择控制寄存器	3-130
0x4624	I2C1_I2C7_DMA_SELECT	I2C1/I2C7 DMA 请求线选择控制寄存器	3-131
0x4700	PWR_SWITCH	power switch 配置	3-131
0x5000	SEC_CTRL0	SLAVE 系统安全属性控制寄存器 0	3-132
0x5004	SEC_CTRL1	SLAVE 系统安全属性控制寄存器 1	3-135
0x5008	SEC_CTRL2	SLAVE 系统安全属性控制寄存器 2	3-137
0x500C	SEC_CTRL3	SLAVE 系统安全属性控制寄存器 3	3-140
0x5014	SEC_CTRL5	SLAVE 系统安全属性控制寄存器 5	3-143



偏移地址	名称	描述	页码
0x5018	SEC_CTRL6	SLAVE 系统安全属性控制寄存器 6	3-145
0x501C	SEC_CTRL7	MASTER 系统安全属性控制寄存器 0	3-147
0x5020	SEC_CTRL8	MASTER 系统安全属性控制寄存器 1	3-149
0x5028	SEC_BOOTRAM_SE C_CFG	BOOT SRAM 安全属性配置寄存器	3-151
0x502C	SEC_BOOTRAM_RO _CFG	BOOT SRAM 只读属性配置寄存器	3-152
0x5030	SEC_BOOTRAM_SE C_CFG_LOCK1	BOOT SRAM 安全属性配置 LOCK 寄存器	3-152
0x5034	SEC_BOOTRAM_RO _CFG_LOCK2	BOOT SRAM 只读属性配置 LOCK 寄存器	3-153

3.5.5.2 MISC 寄存器描述

MISC_CTRL1

MISC_CTRL1 为 CPU 64bit 启动地址寄存器。(安全寄存器，仅安全模块可以访问)

Offset Address: 0x4204 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW_SEC	cpu_rvbaraddr	CPU 启动地址寄存器单位为 16Byte。	0x00000000

DDRCA_REE_RANDOM_L

DDRCA_REE_RANDOM_L 为 DDR CA 功能 REE 随机值低 32bit 值。

Offset Address: 0x4218 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW_CLR	ddrca_ree_random _l	DDR CA 功能 REE 随机值 Bit31~0。 ddrca_lock 写 1 后清除。仅硬复位可复位。	0x00000000



DDRCA_REE_RANDOM_H

DDRCA_REE_RANDOM_H 为 DDR CA 功能 REE 随机值高 32bit 值。

Offset Address: 0x421C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW_CLR	ddrca_ree_random_h	DDR CA 功能 REE 随机值 Bit63~32。 ddrca_lock 写 1 后清除。仅硬复位可复位。	0x00000000

DDRCA_EN_REE

DDRCA_EN_REE 为 DDR_CA 功能模式控制寄存器。

Offset Address: 0x4220 Total Reset Value: 0x0000_000A

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留	0x00000000
[3:0]	RW_CLR	ddrca_en	DDR 加扰功能模式。ddrca_lock 写 1 后置为复位值。 0xA: 旁路掉 MDDRCA 功能; 其他: 开启 DDRCA 功能。	0xA

DDRCA_REE_UPDATE

DDRCA_REE_UPDATE 为 DDR_CA 功能 REE 使能寄存器。

Offset Address: 0x4224 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留	0x00000000
[0]	RW_LOCK	ddrca_ree_update	DDR REE 随机值加扰功能使能。写 1 生效。ddrca_ree_lock 写 1 后锁定。	0x0



DDRCA_LOCK_REE

DDRCA_LOCK_REE 为 DDR_CA LOCK 寄存器。

Offset Address: 0x4228 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW_LOCK	ddrca_lock_ree	DDRC CA LOCK 寄存器。写 1 后锁定。并且同时清除掉 DDRCA_EN_REE [ddrca_en]/ DDRCA_REE_RANDOM_L [ddrca_ree_random_l]/ DDRC_A_REE_RANDOM_H [ddrca_ree_random_h]。	0x0

DDRC_LOCK_CTRL1

DDRC_LOCK_CTRL1 为 DDRC LOCK 寄存器 1。

Offset Address: 0x4230 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留	0x00000000
[0]	RW_LOCK	ddrc_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

DDRC_LOCK_CTRL2

DDRC_LOCK_CTRL2 为 DDRC LOCK 寄存器 2。

Offset Address: 0x4234 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留	0x00000000
[0]	RW_LOCK	ddrc_dmc_write_lock	DDR DMC 寄存器锁定控制。	0x0



Bits	Access	Name	Description	Reset
			0: DDR DMC 寄存器可以被改写; 1: DDR DMC 寄存器不能被改写。	

DDRC_LOCK_CTRL3

DDRC_LOCK_CTRL3 为 DDRC LOCK 寄存器 3。

Offset Address: 0x4238 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留	0x00000000
[0]	RW_LOCK	ddrc_qosbuf_write_lock	DDR QOSBUS 寄存器锁定控制。 0: DDR QOSBUS 寄存器可以被改写; 1: DDR QOSBUS 寄存器不能被改写。	0x0

DDRC_LOCK_CTRL4

DDRC_LOCK_CTRL4 为 DDRC LOCK 寄存器 4。

Offset Address: 0x423C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留	0x00000000
[0]	RW_LOCK	ddrc_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

DDRC_LOCK_CTRL5

DDRC_LOCK_CTRL5 为 DDRC LOCK 寄存器 5。

Offset Address: 0x4240 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留	0x00000000
[0]	RW_LOCK	ddrc_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

SPI_CFG3

SPI_CFG3 为 SPI2/3 片选信号 (CS) 极性配置寄存器。

Offset Address: 0x4610 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	RO	reserved	保留	0x00000000
[2]	RW	spi3_cs_mux	SPI3 片选信号选择。 0: SPI3_CSN0 输出有效片选; 1: SPI3_CSN1 输出有效片选。	0x0
[1]	RW	spi3_cs_pctrl	SPI3 片选信号极性控制。 0: SPI3_CSN 低有效; 1: SPI3_CSN 高有效。	0x0
[0]	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

I2C_MONTAGE_CFG0

I2C_MONTAGE_CFG0 为 I2C 蒙太奇功能使能配置寄存器。

Offset Address: 0x4614 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	RO	reserved	保留	0x00000000



Bits	Access	Name	Description	Reset
[3:0]	RW	i2c_montage_en	每路 I2C 蒙太奇功能使能。 bit[0]: I2C3 蒙太奇功能控制; bit[1]: I2C4 蒙太奇控制; bit[2]: I2C5 蒙太奇控制; bit[3]: I2C6 蒙太奇控制。 0: 不使能 1: 使能 使用 I2C3 控制器同时控制 I2C3/4/5/6。	0x0

I2C_MONTAGE_CFG2

I2C_MONTAGE_CFG2 为 I2C 蒙太奇读写选择寄存器。

Offset Address: 0x461C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:12]	RO	reserved	保留。	0x00000
[11:8]	RW	fixed_sel	调试使用。	0x0
[7:4]	RW	i2c_write_slave	回写 I2C 选择。	0x0
[3:0]	RW	i2c_read_slave	回读 I2C 选择。	0x0

I2C0_I2C6_DMA_SEL

I2C0_I2C6_DMA_SEL 为 I2C0/I2C6 DMA 请求线选择控制寄存器。

Offset Address: 0x4620 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留	0x00000000
[0]	RW	i2c0_i2c6_dma_req_sel	I2C0/I2C6 DMA 请求选择信号。 0: I2C0	0x0



Bits	Access	Name	Description	Reset
			1: I2C6	

I2C1_I2C7_DMA_SEL

I2C1_I2C7_DMA_SEL 为 I2C1/I2C7 DMA 请求线选择控制寄存器。

Offset Address: 0x4624 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留	0x00000000
[0]	RW	i2c1_i2c7_dma_req_sel	I2C1/I2C7 DMA 请求选择信号。 0: I2C1 1: I2C7	0x0

PWR_SWITCH

PWR_SWITCH 为 power switch 配置寄存器。

Offset Address: 0x4700 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:6]	RO	reserved	保留	0x00000000
[5]	RW	sdio0_pwr_sw_sel_by_misc	SDIO0 LDO 供电输出电压选择。 0: 3.3V 1: 1.8V	0x0
[4]	RW	sdio0_pwr_en_by_misc	SDIO0_CCLK_OUT/SDIO0_CCMD/SDIO0_CDATA*管脚供电使能。 0: SDIO0_CCLK_OUT/SDIO0_CCMD/SDIO0_CDATA*供电不使能; 1: SDIO0_CCLK_OUT/SDIO0_CCMD/SDIO0_CDATA*供电使能。	0x0



Bits	Access	Name	Description	Reset
[3:2]	-	reserved	保留。	0x0
[1]	RW	sdio0_io_mode_sel 1_from_misc	SDIO0_CCLK_OUT/SDIO0_CCMD/SDIO0_CDATA*管脚工作模式选择。 0: 3.3V 模式; 1: 1.8V 模式。	0x0
[0]	RW	sdio0_pwr_ctrl_by_ user	SDIO0_CCLK_OUT/SDIO0_CCMD/SDIO0_CDATA*管脚供电控制选择。 0: SDIO0_CCLK_OUT/SDIO0_CCMD/SDIO0_CDATA*供电由 SDIO 控制器控制; 1: SDIO0_CCLK_OUT/SDIO0_CCMD/SDIO0_CDATA*供电由 MISC 寄存器控制。	0x1

SEC_CTRL0

SEC_CTRL0 为 SLAVE 系统安全属性控制寄存器 0。(安全寄存器, 仅安全模块可以访问)

Offset Address: 0x5000 Total Reset Value: 0x1111_1111

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30:28]	RW_SEC	{i2c7_slv_sec_acc_ctrl, i2c7_slv_sec_acc_ctrl_high, i2c7_slv_sec_acc_disable}	I2C7 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1



Bits	Access	Name	Description	Reset
[27]	-	reserved	保留。	0x0
[26:24]	RW_SEC	{i2c6_slv_sec_acc_ctrl, i2c6_slv_sec_acc_ctrl_high, i2c6_slv_sec_acc_disable}	I2C6 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1
[23]	-	reserved	保留。	0x0
[22:20]	RW_SEC	{i2c5_slv_sec_acc_ctrl, i2c5_slv_sec_acc_ctrl_high, i2c5_slv_sec_acc_disable}	I2C5 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1
[19]	-	reserved	保留。	0x0
[18:16]	RW_SEC	{i2c4_slv_sec_acc_ctrl, i2c4_slv_sec_acc_ctrl_high, i2c4_slv_sec_acc_disable}	I2C4 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。	0x1



Bits	Access	Name	Description	Reset
			注：X 表示 0 和 1。	
[15]	-	reserved	保留。	0x0
[14:12]	RW_SEC	{i2c3_slv_sec_acc_ctrl, i2c3_slv_sec_acc_ctrl_high, i2c3_slv_sec_acc_disable}	I2C3 安全访问模式控制。 0X0：开启安全模式，读写访问均由模块自行鉴权； 100：开启安全模式，写仅支持安全访问，读由模块自行鉴权； 110：开启安全模式，读写都仅支持安全访问； XX1：关闭安全模式。 注：X 表示 0 和 1。	0x1
[11]	-	reserved	保留。	0x0
[10:8]	RW_SEC	{i2c2_slv_sec_acc_ctrl, i2c2_slv_sec_acc_ctrl_high, i2c2_slv_sec_acc_disable}	I2C2 安全访问模式控制。 0x0：开启安全模式，读写访问均由模块自行鉴权； 100：开启安全模式，写仅支持安全访问，读由模块自行鉴权； 110：开启安全模式，读写都仅支持安全访问； xx1：关闭安全模式。 注：X 表示 0 和 1。	0x1
[7]	-	reserved	保留。	0x0
[6:4]	RW_SEC	{i2c1_slv_sec_acc_ctrl, i2c1_slv_sec_acc_ctrl_high, i2c1_slv_sec_acc_disable}	I2C1 安全访问模式控制。 0X0：开启安全模式，读写访问均由模块自行鉴权； 100：开启安全模式，写仅支持安全访问，读由模块自行鉴权； 110：开启安全模式，读写都仅支持安全访问；	0x1



Bits	Access	Name	Description	Reset
			XX1: 关闭安全模式。 注: X 表示 0 和 1。	
[3]	-	reserved	保留。	0x0
[2:0]	RW_SEC	{i2c0_slv_sec_acc_ctrl, i2c0_slv_sec_acc_ctrl_high, i2c0_slv_sec_acc_disable}	I2C0 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1

SEC_CTRL1

SEC_CTRL1 为 SLAVE 系统安全属性控制寄存器 1。(安全寄存器, 仅安全模块可以访问)

Offset Address: 0x5004 Total Reset Value: 0x1101_1011

Bits	Access	Name	Description	Reset
[31]	RO	reserved	保留。	0x0
[30:28]	RO	reserved	保留。	0x1
[27]	-	reserved	保留。	0x0
[26:24]	RW_SEC	{edma_slv_sec_acc_ctrl, edma_slv_sec_acc_ctrl_high, edma_slv_sec_acc_disable}	DMA 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访	0x1



Bits	Access	Name	Description	Reset
			问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	
[23:19]	-	reserved	保留。	0x00
[18:16]	RW_SEC	{hpm_ctrl_slv_sec_acc_ctrl, hpm_ctrl_slv_sec_acc_ctrl_high, hpm_ctrl_slv_sec_acc_disable}	HPM CTRL 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1
[15]	-	reserved	保留。	0x0
[14:12]	-	reserved	保留。	0x1
[11:7]	-	reserved	保留。	0x00
[6:4]	RW_SEC	{sec_timer23_slv_sec_acc_ctrl, sec_timer23_slv_sec_acc_ctrl_high, sec_timer23_slv_sec_acc_disable }	sec_timer23 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1
[3]	-	reserved	保留。	0x0
[2:0]	RW_SEC	{sec_timer01_slv_sec_acc_ctrl, sec_timer01_slv_sec_acc_disable }	sec_timer01 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自	0x1



Bits	Access	Name	Description	Reset
		c_acc_ctrl_high, sec_timer01_slv_sec_acc_disable}	行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	

SEC_CTRL2

SEC_CTRL2 为 SLAVE 系统安全属性控制寄存器 2。(安全寄存器, 仅安全模块可以访问)

Offset Address: 0x5008 Total Reset Value: 0x1111_1111

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x2
[26:24]	RW_SEC	{dpu_slv_sec_acc_ctrl, dpu_slv_sec_acc_ctrl_high, dpu_slv_sec_acc_disable }	DPU 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1
[23]	-	reserved	保留。	0x0
[22:20]	RW_SEC	{vdp_slv_sec_acc_ctrl, vdp_slv_sec_acc_ctrl_high, vdp_slv_sec_acc_disable }	VDP 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问,	0x1



Bits	Access	Name	Description	Reset
			读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	
[19]	-	reserved	保留。	0x0
[18:16]	RW_SEC	{mipitx_ctrl_slv_sec_acc_ctrl, mipitx_ctrl_slv_sec_acc_ctrl_high, mipitx_ctrl_slv_sec_acc_disable}	MIPITX_CTRL 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1
[15]	-	reserved	保留。	0x0
[14:12]	RW_SEC	{mipirx_ctrl_slv_sec_acc_ctrl, mipirx_ctrl_slv_sec_acc_ctrl_high, mipirx_ctrl_slv_sec_acc_disable}	MIPIRX_CTRL 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1
[11]	-	reserved	保留。	0x0
[10:8]	RW_SEC	{viproc_slv_sec_acc_ctrl, viproc_slv_sec_acc_ctrl_high,	VIPROC 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权;	0x1



Bits	Access	Name	Description	Reset
		viproc_slv_sec_acc_disable}	100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	
[7]	-	reserved	保留。	0x0
[6:4]	RW_SEC	{vpss_slv_sec_acc_ctrl, vpss_slv_sec_acc_ctrl_high, vpss_slv_sec_acc_disable}	VPSS 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式 注: X 表示 0 和 1。	0x1
[3]	-	reserved	保留。	0x0
[2:0]	RW_SEC	{vicap_slv_sec_acc_ctrl, vicap_slv_sec_acc_ctrl_high, vicap_slv_sec_acc_disable}	VICAP 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1



SEC_CTRL3

SEC_CTRL3 为 SLAVE 系统安全属性控制寄存器 3。(安全寄存器, 仅安全模块可以访问)

Offset Address: 0x500C Total Reset Value: 0x1111_1111

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30:28]	RW_SEC	{scd_slv_sec_acc_ctrl,scd_slv_sec_acc_ctrl_high,scd_slv_sec_acc_disable }	SCD 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1
[27]	-	reserved	保留。	0x0
[26:24]	RW_SEC	{vedu_slv_sec_acc_ctrl,vedu_slv_sec_acc_ctrl_high,vedu_slv_sec_acc_disable }	VEDU 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1	0x1
[23]	-	reserved	保留。	0x0
[22:20]	RW_SEC	{fmu1_slv_sec_acc_ctrl,fmu1_slv_sec_acc_ctrl_high,fmu1_slv_sec_acc_	FMU1 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问,	0x1



Bits	Access	Name	Description	Reset
		disable }	读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	
[19]	-	reserved	保留。	0x0
[18:16]	RW_SEC	{fmu0_slv_sec_acc_ctrl, fmu0_slv_sec_acc_ctrl_high, fmu0_slv_sec_acc_disable}	FMU0 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1
[15]	-	reserved	保留。	0x0
[14:12]	RW_SEC	{vgs_slv_sec_acc_ctrl, vgs_slv_sec_acc_ctrl_high, vgs_slv_sec_acc_disable }	VGS 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1
[11]	-	reserved	保留。	0x0
[10:8]	RW_SEC	{jpgd_slv_sec_acc_ctrl, jpgd_slv_sec_acc_ctrl_high,	JPGD 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权;	0x1



Bits	Access	Name	Description	Reset
		jpgd_slv_sec_acc_disable }	100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	
[7]	-	reserved	保留。	0x0
[6:4]	RW_SEC	{jpge_slv_sec_acc_ctrl, jpge_slv_sec_acc_ctrl_high, jpge_slv_sec_acc_disable }	JPGE 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1
[3]	-	reserved	保留。	0x0
[2:0]	RW_SEC	{thermo_slv_sec_acc_ctrl, thermo_slv_sec_acc_ctrl_high, thermo_slv_sec_acc_disable }	THERMO 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1



SEC_CTRL5

SEC_CTRL5 为 SLAVE 系统安全属性控制寄存器 5。(安全寄存器, 仅安全模块可以访问)

Offset Address: 0x5014 Total Reset Value: 0x0011_1111

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23]	-	reserved	保留。	0x0
[22:20]	RW_SEC	{spi_3wire_slv_sec_acc_ctrl, spi_3wire_slv_sec_acc_ctrl_high, spi_3wire_slv_sec_acc_disable }	SPI_3WIRE 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式 注: X 表示 0 和 1。	0x1
[21]	-	reserved	保留。	0x0
[20]	-	reserved	保留。	0x1
[19]	-	reserved	保留。	0x0
[18:16]	RW_SEC	{spi_3tft_slv_sec_acc_ctrl, spi_3tft_slv_sec_acc_ctrl_high, spi_3tft_slv_sec_acc_disable }	SPI_TFT, 新 3 线安全访问模式控制(小屏)。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1



Bits	Access	Name	Description	Reset
[15]	-	reserved	保留。	0x0
[14:12]	RW_SEC	{spi3_slv_sec_acc_ctrl, spi3_slv_sec_acc_ctrl_high, spi3_slv_sec_acc_disable }	SPI3 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1
[11]	-	reserved	保留。	0x0
[10:8]	RW_SEC	{spi2_slv_sec_acc_ctrl, spi2_slv_sec_acc_ctrl_high, spi2_slv_sec_acc_disable }	SPI2 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1
[7]	-	reserved	保留。	0x0
[6:4]	RW_SEC	{spi1_slv_sec_acc_ctrl, spi1_slv_sec_acc_ctrl_high, spi1_slv_sec_acc_disable }	SPI1 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。	0x1



Bits	Access	Name	Description	Reset
			注：X 表示 0 和 1。	
[3]	-	reserved	保留。	0x0
[2:0]	RW_SEC	{spi0_slv_sec_acc_ctrl, spi0_slv_sec_acc_ctrl_high, spi0_slv_sec_acc_disable }	SPI0 安全访问模式控制。 0X0：开启安全模式，读写访问均由模块自行鉴权； 100：开启安全模式，写仅支持安全访问，读由模块自行鉴权； 110：开启安全模式，读写都仅支持安全访问； XX1：关闭安全模式。 注：X 表示 0 和 1。	0x1

SEC_CTRL6

SEC_CTRL5 为 SLAVE 系统安全属性控制寄存器 6。（安全寄存器，仅安全模块可以访问）

Offset Address: 0x5018 Total Reset Value: 0x0011_1111

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23]	-	reserved	保留。	0x0
[22:20]	RW_SEC	{uart5_slv_sec_acc_ctrl, uart5_slv_sec_acc_ctrl_high, uart5_slv_sec_acc_disable }	UART5 线安全访问模式控制。 0X0：开启安全模式，读写访问均由模块自行鉴权； 100：开启安全模式，写仅支持安全访问，读由模块自行鉴权； 110：开启安全模式，读写都仅支持安全访问； XX1：关闭安全模式。 注：X 表示 0 和 1。	0x1



Bits	Access	Name	Description	Reset
[19]	-	reserved	保留。	0x0
[18:16]	RW_SEC	{uart4_slv_sec_acc_ctrl, uart4_slv_sec_acc_ctrl_high, uart4_slv_sec_acc_disable }	UART4 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1
[15]	-	reserved	保留。	0x0
[14:12]	RW_SEC	{uart3_slv_sec_acc_ctrl, uart3_slv_sec_acc_ctrl_high, uart3_slv_sec_acc_disable }	UART3 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。 注: X 表示 0 和 1。	0x1
[11]	-	reserved	保留。	0x0
[10:8]	RW_SEC	{uart2_slv_sec_acc_ctrl, uart2_slv_sec_acc_ctrl_high, uart2_slv_sec_acc_disable }	UART2 安全访问模式控制。 0X0: 开启安全模式, 读写访问均由模块自行鉴权; 100: 开启安全模式, 写仅支持安全访问, 读由模块自行鉴权; 110: 开启安全模式, 读写都仅支持安全访问; XX1: 关闭安全模式。	0x1



Bits	Access	Name	Description	Reset
			注：X 表示 0 和 1。	
[7]	-	reserved	保留。	0x0
[6:4]	RW_SEC	{uart1_slv_sec_acc_ctrl, uart1_slv_sec_acc_ctrl_high, uart1_slv_sec_acc_disable}	UART1 安全访问模式控制。 0X0：开启安全模式，读写访问均由模块自行鉴权； 100：开启安全模式，写仅支持安全访问，读由模块自行鉴权； 110：开启安全模式，读写都仅支持安全访问； XX1：关闭安全模式。 注：X 表示 0 和 1。	0x1
[3]	-	reserved	保留。	0x0
[2:0]	RW_SEC	{uart0_slv_sec_acc_ctrl, uart0_slv_sec_acc_ctrl_high, uart0_slv_sec_acc_disable}	UART0 安全访问模式控制。 0X0：开启安全模式，读写访问均由模块自行鉴权； 100：开启安全模式，写仅支持安全访问，读由模块自行鉴权； 110：开启安全模式，读写都仅支持安全访问； XX1：关闭安全模式。 注：X 表示 0 和 1。	0x1

SEC_CTRL7

SEC_CTRL7 为 MASTER 系统安全属性控制寄存器 0。（安全寄存器，仅安全模块可以访问）

Offset Address: 0x501C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	RW_SEC	dpu_mst_ar_sec_m	DPU MASTER 读口安全访问控制模式。	0x0



Bits	Access	Name	Description	Reset
		ode	00: 非安全模式; 01: 安全模式; 其它: 保留。	
[29:28]	RW_SEC	dpu_mst_aw_sec_mode	DPU MASTER 写口安全访问控制模式。 00: 非安全模式; 01: 安全模式; 其它: 保留。	0x0
[27:26]	RW_SEC	scd_mst_ar_sec_mode	SCD MASTER 读口安全访问控制模式。 00: 非安全模式; 01: 安全模式; 其它: 保留。	0x0
[25:24]	RW_SEC	scd_mst_aw_sec_mode	SCD MASTER 写口安全访问控制模式。 00: 非安全模式; 01: 安全模式; 其它: 保留。	0x0
[23:22]	RW_SEC	vedu_mst_ar_sec_mode	VEDU MASTER 读口安全访问控制模式。 00: 非安全模式; 01: 安全模式; 其它: 保留。	0x0
[21:20]	RW_SEC	vedu_mst_aw_sec_mode	VEDU MASTER 写口安全访问控制模式。 00: 非安全模式; 01: 安全模式; 其它: 保留。	0x0
[19:18]	RW_SEC	vdp_mst_ar_sec_mode	VDP MASTER 读口安全访问控制模式。 00: 非安全模式; 01: 安全模式; 其它: 保留。	0x0



Bits	Access	Name	Description	Reset
[17:16]	RW_SEC	vdp_mst_aw_sec_mode	VDP MASTER 写口安全访问控制模式。 00: 非安全模式; 01: 安全模式; 其它: 保留。	0x0
[15:14]	RW_SEC	aiao_mst_ar_sec_mode	AIAO MASTER 读口安全访问控制模式。 00: 非安全模式; 01: 安全模式。 其它: 保留	0x0
[13:12]	RW_SEC	aiao_mst_aw_sec_mode	AIAO MASTER 写口安全访问控制模式。 00: 非安全模式; 01: 安全模式; 其它: 保留。	0x0
[11:10]	RW_SEC	vgs_mst_ar_sec_mode	VGS MASTER 读口安全访问控制模式。 00: 非安全模式; 01: 安全模式; 其它: 保留。	0x0
[9:8]	RW_SEC	vgs_mst_aw_sec_mode	VGS MASTER 写口安全访问控制模式。 00: 非安全模式; 01: 安全模式; 其它: 保留。	0x0
[7:0]	-	reserved	保留。	0x0

SEC_CTRL8

SEC_CTRL7 为 MASTER 系统安全属性控制寄存器 1。(安全寄存器, 仅安全模块可以访问)

Offset Address: 0x5020 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:28]	RO	reserved	保留	0x0
[27:26]	RW_SEC	vicap_mst_ar_sec_mode	VICAP MASTER 读口安全访问控制模式。 00: 非安全模式; 01: 安全模式。 其它: 保留	0x0
[25:24]	RW_SEC	vicap_mst_aw_sec_mode	VICAP MASTER 写口安全访问控制模式。 00: 非安全模式; 01: 安全模式。 其它: 保留	0x0
[23:22]	RW_SEC	viproc_mst_ar_sec_mode	VIPROC MASTER 读口安全访问控制模式。 00: 非安全模式; 01: 安全模式。 其它: 保留	0x0
[21:20]	RW_SEC	viproc_mst_aw_sec_mode	VIPROC MASTER 写口安全访问控制模式。 00: 非安全模式; 01: 安全模式。 其它: 保留	0x0
[19:18]	RW_SEC	vpss_mst_ar_sec_mode	VPSS MASTER 读口安全访问控制模式。 00: 非安全模式; 01: 安全模式。 其它: 保留	0x0
[17:16]	RW_SEC	vpss_mst_aw_sec_mode	VPSS MASTER 写口安全访问控制模式。 00: 非安全模式; 01: 安全模式。 其它: 保留	0x0



Bits	Access	Name	Description	Reset
[15:14]	-	reserved	保留	0x0
[13:12]	-	reserved	保留	0x0
[11:10]	-	reserved	保留	0x0
[9:8]	-	reserved	保留	0x0
[7:6]	RW_SEC	jpgd_mst_ar_sec_mode	JPGD MASTER 读口安全访问控制模式。 00: 非安全模式; 01: 安全模式。 其它: 保留	0x0
[5:4]	RW_SEC	jpgd_mst_aw_sec_mode	JPGD MASTER 写口安全访问控制模式。 00: 非安全模式; 01: 安全模式。 其它: 保留	0x0
[3:2]	RW_SEC	jpge_mst_ar_sec_mode	JPGE MASTER 读口安全访问控制模式。 00: 非安全模式; 01: 安全模式。 其它: 保留	0x0
[1:0]	RW_SEC	jpge_mst_aw_sec_mode	JPGE MASTER 写口安全访问控制模式。 00: 非安全模式; 01: 安全模式。 其它: 保留	0x0

SEC_BOOTRAM_SEC_CFG

SEC_BOOTRAM_SEC_CFG 为 BOOT SRAM 安全属性配置寄存器。(安全寄存器, 仅安全模块可以访问)

Offset Address: 0x5028 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW_SEC_LOCK	bootram_sec_cfg	bootram 安全读写区间, 从 0 地址开始, 单位为 1KB。 bootram_sec_cfg_lock 写 1 后锁定, 不能修改。	0x00

SEC_BOOTRAM_RO_CFG

SEC_BOOTRAM_RO_CFG 为 BOOT SRAM 只读属性配置寄存器。(安全寄存器, 仅安全模块可以访问)

Offset Address: 0x502C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW_SEC_LOCK	bootram_ro_cfg	bootram 只读区间, 从 0 地址开始, 单位为 1KB。 bootram_ro_cfg_lock 写 1 后锁定, 不能修改。	0x00

SEC_BOOTRAM_SEC_CFG_LOCK1

SEC_BOOTRAM_SEC_CFG_LOCK1 为 BOOT SRAM 安全属性配置 LOCK 寄存器。(安全寄存器, 仅安全模块可以访问)

Offset Address: 0x5030 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW_SEC_LOCK	bootram_sec_lock	bootram 安全读区间配置 LOCK 寄存器。 写 1 后锁定。	0x0



SEC_BOOTRAM_RO_CFG_LOCK2

SEC_BOOTRAM_RO_CFG_LOCK2 为 BOOT SRAM 只读属性配置 LOCK 寄存器。(安全寄存器, 仅安全模块可以访问)

Offset Address: 0x5034 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW_SEC_LOCK	bootram_ro_lock	bootram 只读区间配置 LOCK 寄存器。写 1 后锁定。	0x0

3.5.6 VI_MIPI 及 SPI0/1 寄存器

3.5.6.1 VI_MIPI 及 SPI0/1 寄存器概览

VI_MIPI 及 SPI0/1 寄存器概览如表 3-2 所示。

表3-2 VI_MIPI 及 SPI0/1 寄存器概览 (基址是 0x0_17A4_0000)

偏移地址	名称	描述	页码
0x0220	SPI_CFG10	SPI0/1 片选信号 (CS) 极性配置寄存器	3-153
0x0300	VI_MIPI_CFG	VI_MIPI 配置寄存器	3-154

3.5.6.2 VI_MIPI 及 SPI0/1 寄存器描述

SPI_CFG10

SPI_CFG10 为 SPI0/1 片选信号 (CS) 极性配置寄存器

Offset Address: 0x0220 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	RO	reserved	保留。	0x00000000
[2]	RW	spi0_cs_mux	SPI0 片选信号。 0: SPI0_CSN0 输出有效片选;	0x0



Bits	Access	Name	Description	Reset
			1: SPI0_CSN1 输出有效片选。	
[1]	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
[0]	RW	spi1_cs_pctrl	SPI1 片选信号极性控制。 0: SPI1_CSN 低有效; 1: SPI1_CSN 高有效。	0x0

VI_MIPI_CFG

VI_MIPI_CFG 为 VI_MIPI 配置寄存器。

Offset Address: 0x0300 Total Reset Value: 0x0000_0600

Bits	Access	Name	Description	Reset
[31]	RW	viproc_vpss_vnr_sel	VNR 模块静态选择合入模块的控制信号。 0: VNR 静态合入 VIPROC 中; 1: VNR 静态合入 VPSS 中。	0x0
[30]	-	reserved	保留。	0x0
[29]	RW	viproc_vpss_online_mode	VIPROC&VPSS 工作模式选择。 0: 离线模式; 1: 在线模式。	0x0
[28]	RW	vicap_viproc_online_mode	VICAP&VIPROC_工作模式选择。 0: 离线模式; 1: 在线模式。	0x0
[27:12]	-	reserved	保留。	0x00
[11]	RW	vi_port3_input_sel	VI Port3 视频源选择。 0: MIPI 通道 3; 1: CMOS0(BT656/BT1120/RAW)。	0x0



Bits	Access	Name	Description	Reset
[10]	-	reserved	保留。	0x1
[9:8]	RW	vi_port1_input_sel	VI Port1 视频源选择。 00: CMOS0(BT656/BT1120/RAW); 01: 热成像通道; 1x: MIPI 通道 1。	0x2
[7:6]	RO	reserved	保留。	0x0
[5:4]	RW	mipi2_work_mode	MIPI 通道 2 模式选择。 00: MIPI 模式; 01: LVDS 模式; 11: CMOS 模式; 其它: 保留。	0x0
[3:2]	RW	mipi1_work_mode	MIPI 通道 1 模式选择。 00: MIPI 模式; 01: LVDS 模式; 11: CMOS 模式; 其它: 保留。	0x0
[1:0]	RW	mipi0_work_mode	MIPI 通道 0 模式选择。 00: MIPI 模式; 01: LVDS 模式; 11: CMOS 模式; 其它: 保留。	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。

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。
- 在外设传输时，支持 DMAC 流控和外设流控方式。
- 支持 32 路外设请求，每路请求包括 Single、Burst 和 Last Single 三种请求。
- 支持 DMA 链表传输。
- 支持通道优先级可配。
- 支持 DMA BURST 长度软件可配。
- 支持 8bit、16bit、32bit、64bit 数据位宽方式传输。
- 只支持小端模式。

3.6.3 功能描述

3.6.3.1 外设请求线

DMAC 的 Single 和 Burst 硬件请求和相应设备的对应关系如表 3-3 所示。其中编号 0~3 对应的请求线可通过寄存器配置。

表3-3 DMAC 外设 Single 和 Burst 硬件请求线编号说明

外设硬件请求线编号	对应设备
0	I2C0 或 I2C6 RX 通道 DMA 请求
1	I2C0 或 I2C6 TX 通道 DMA 请求



外设硬件请求线编号	对应设备
2	I2C1 或 I2C7 RX 通道 DMA 请求
3	I2C1 或 I2C7 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	SPI0 RX 通道 DMA 请求
13	SPI0 TX 通道 DMA 请求
14	SPI1 RX 通道 DMA 请求
15	SPI1 TX 通道 DMA 请求
16	SPI2 RX 通道 DMA 请求
17	SPI2 TX 通道 DMA 请求
18	SPI3 RX 通道 DMA 请求
19	SPI3 TX 通道 DMA 请求
20	UART0 RX 通道 DMA 请求
21	UART0 TX 通道 DMA 请求
22	UART1 RX 通道 DMA 请求
23	UART1 TX 通道 DMA 请求
24	UART2 RX 通道 DMA 请求



外设硬件请求线编号	对应设备
25	UART2 TX 通道 DMA 请求
26	UART3 RX 通道 DMA 请求
27	UART3 TX 通道 DMA 请求
28	UART4 RX 通道 DMA 请求
29	UART4 TX 通道 DMA 请求
30	UART5 RX 通道 DMA 请求
31	UART5 TX 通道 DMA 请求

DMAC 的 Last Single 硬件请求和相应设备的对应关系如表 3-4 所示，仅在外设流控模式时生效。当外设发出接收 Last Single 请求时，表示外设请求 DMAC 触发最后一次数据传输。

表3-4 DMAC 外设接收 Last Single 硬件请求线编号说明

外设硬件请求线编号	对应设备
0~19	保留
20	UART0 RX 通道 DMA 请求
21	保留
22	UART1 RX 通道 DMA 请求
23	保留
24	UART2 RX 通道 DMA 请求
25	保留
26	UART3 RX 通道 DMA 请求
27	保留
28	UART4 RX 通道 DMA 请求
29	保留



外设硬件请求线编号	对应设备
30	UART5 RX 通道 DMA 请求
31	保留

3.6.3.2 访问空间

DMAC 可以访问的地址空间如表 3-5 所示。

表3-5 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	I2C0
	I2C1
	I2C2
	I2C3
	I2C4
	I2C5
	I2C6
	I2C7
	SPI0
	SPI1
	SPI2
	SPI3
	UART0
	UART1
	UART2



访问空间类型	描述
	UART3
	UART4
	UART5

须知

- 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]为 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 65535Bytes 的数据。

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



- 如果目的地址配置为不递增，目的地址需要和目的端的传输位宽对齐，否则会报配置错误中断；如果目的端为外设时，目的端传输位宽的大小应该与外设 TX FIFO 位宽相等。

须知

- 对于内存到外设以及外设到内存的传输，当一次传输的数据量大于一次 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 会在传完一次基本传输后从 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 65535 Bytes。

三种基本传输均支持 DMA 链表功能。如需使能该功能，必须配置寄存器 $C(n)_{LLI_L}[chain_en]$ 为 0x2，且除链尾节点外的各节点链表信息中的 $C(n)_{LLI_L}[chain_en]$ 也必须为 0x2。对于链表尾，应当配置该节点的 $C(n)_{LLI_L}$ 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 $C(n)_{LLI_L}[lli_l]$ 和 $C(n)_{LLI_H}[lli_h]$ ），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 $C(n)_{SRC_ADDR_L}$ 、 $C(n)_{SRC_ADDR_H}$ 、 $C(n)_{DES_ADDR_L}$ 、 $C(n)_{DES_ADDR_H}$ 、 $C(n)_{LLI_L}$ 、 $C(n)_{LLI_H}$ 、 $C(n)_{CNT0}$ 、 $C(n)_{CONFIG}[31:1]$ 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。

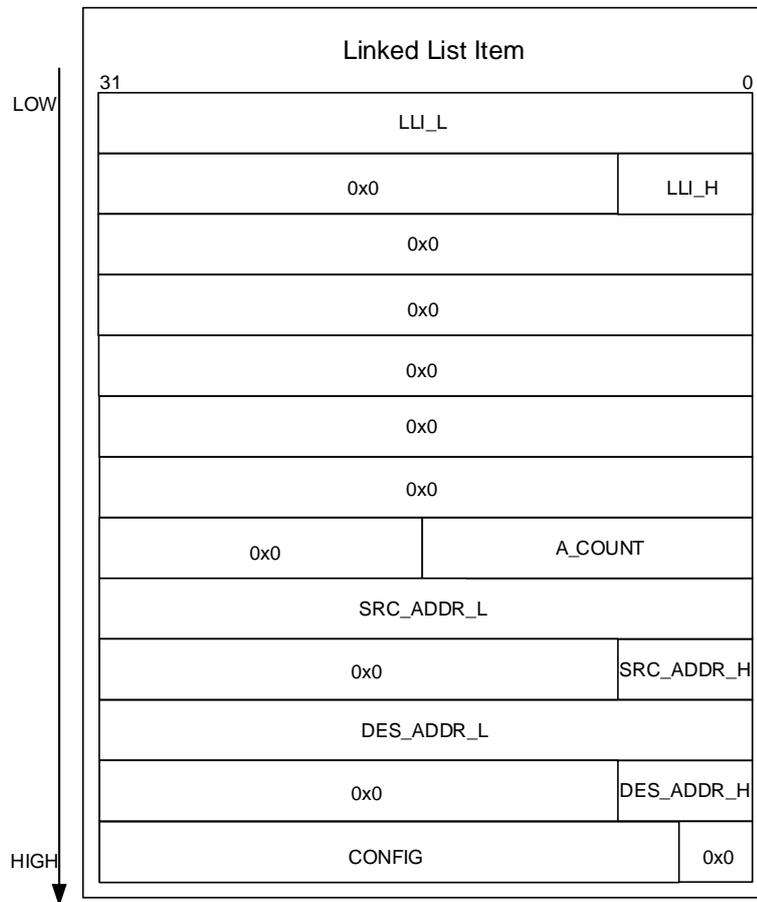


说明

- 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 3-6 所示。

图3-6 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, 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_L 和 C(n)_CURR_SRC_ADDR_H 用于查询当前传输的源地址, 寄存器 C(n)_CURR_DEST_ADDR_L 和 C(n)_CURR_DEST_ADDR_H 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_CRG2720[edma_axi_cken]写 0x1, 打开 DMAC 的 AXI 时钟门控, 向 CRG 寄存器 PERI_CRG2720[edma_apb_cken]写 0x1, 打开 DMAC 的 APB 时钟门控。

步骤 2 向 CRG 寄存器 PERI_CRG2720[edma_srst_req]写 0x1, 对 DMAC 进行复位。

步骤 3 向 CRG 寄存器 PERI_CRG2720[edma_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”、“3.6.4.8 链表传输”、“3.6.4.9 中断处理”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 0x010, 将通道的安全属性转换为非安全。

步骤 2 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”、“3.6.4.8 链表传输”、“3.6.4.9 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。

---结束

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_RAW`、`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 `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 写通道寄存器 `C(n)_SRC_ADDR_L` 和 `C(n)_SRC_ADDR_H`，设置源端的首物理地址。

步骤 4 写通道寄存器 `C(n)_DES_ADDR_L` 和 `C(n)_DES_ADDR_H`，设置目的端的首物理地址。

步骤 5 写通道寄存器 `C(n)_LLI_L[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 内存之间的传输

内存到内存的传输示例步骤如下：

步骤 1 读寄存器 CH_STAT，发现通道 0 空闲。

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。

步骤 3 向寄存器 C(0)_SRC_ADDR_L 和 C(0)_SRC_ADDR_H 写 0x08400000。

步骤 4 向寄存器 C(0)_DES_ADDR_L 和 C(0)_DES_ADDR_H 写 0x08500000。

步骤 5 向寄存器 C(0)_LLI_L[1:0]为 0x0。

步骤 6 向寄存器 C(0)_CNT0 写 0x1000。

步骤 7 向寄存器 C(0)_CONFIG 写 0xcff33000。

步骤 8 向寄存器 C(0)_CONFIG[ch_en]为 0x1。

----结束

3.6.4.6 内存到外设的传输

内存到 I2C0 的传输示例步骤如下：

步骤 1 读寄存器 CH_STAT，发现通道 1 空闲。

步骤 2 向寄存器 INT_TC1_RAW、INT_TC2_RAW、INT_ERR1_RAW、INT_ERR2_RAW 和 INT_ERR3_RAW 写 0x2。

步骤 3 向寄存器 C(1)_SRC_ADDR_L 和 C(1)_SRC_ADDR_H 写 0x08400000。

步骤 4 向寄存器 C(1)_DES_ADDR_L 和 C(1)_DES_ADDR_H 写 0x011060000。



- 步骤 5 向寄存器 C(1)_LLI_L[1:0]为 0x0。
- 步骤 6 向寄存器 C(1)_CNT0 写 0x10。
- 步骤 7 向寄存器 C(1)_CONFIG 写 0x80000014。
- 步骤 8 向寄存器 I2C0_I2C6_DMA_SEL 写 0x0，选择 I2C0 请求线。
- 步骤 9 向寄存器 C(1)_CONFIG[ch_en]为 0x1。
- 步骤 10 完成 I2C0 配置并打开 DMA TX 功能。

----**结束**

内存到 UART0 的传输示例步骤如下：

- 步骤 1 读寄存器 CH_STAT，发现通道 1 空闲。
- 步骤 2 向寄存器 INT_TC1_RAW、INT_TC2_RAW、INT_ERR1_RAW、INT_ERR2_RAW 和 INT_ERR3_RAW 写 0x2。
- 步骤 3 向寄存器 C(1)_SRC_ADDR_L 和 C(1)_SRC_ADDR_H 写 0x084000000。
- 步骤 4 向寄存器 C(1)_DES_ADDR_L 和 C(1)_DES_ADDR_H 写 0x012100000。
- 步骤 5 向寄存器 C(1)_LLI_L[1:0]为 0x0。
- 步骤 6 向寄存器 C(1)_CNT0 写 0x10。
- 步骤 7 向寄存器 C(1)_CONFIG 写 0x80000154。
- 步骤 8 向寄存器 C(1)_CONFIG[ch_en]为 0x1。
- 步骤 9 完成 UART0 配置并打开 DMA TX 功能。

----**结束**

内存到 SSP0 的传输示例步骤如下：

- 步骤 1 读寄存器 CH_STAT，发现通道 1 空闲。
- 步骤 2 向寄存器 INT_TC1_RAW、INT_TC2_RAW、INT_ERR1_RAW、INT_ERR2_RAW 和 INT_ERR3_RAW 写 0x2。
- 步骤 3 向寄存器 C(1)_SRC_ADDR_L 和 C(1)_SRC_ADDR_H 写 0x084000000。
- 步骤 4 向寄存器 C(1)_DES_ADDR_L 和 C(1)_DES_ADDR_H 写 0x012120008。



- 步骤 5 向寄存器 C(1)_LLI_L[1:0]为 0x0。
- 步骤 6 向寄存器 C(1)_CNT0 写 0x10。
- 步骤 7 向寄存器 C(1)_CONFIG 写 0x800110D4。
- 步骤 8 向寄存器 C(1)_CONFIG[ch_en]为 0x1。
- 步骤 9 完成 SSP0 配置并打开 DMA TX 功能。

----结束

3.6.4.7 外设到内存的传输

I2C0 到内存的传输示例步骤如下：

- 步骤 1 读寄存器 CH_STAT，发现通道 2 空闲。
- 步骤 2 向寄存器 INT_TC1_RAW、INT_TC2_RAW、INT_ERR1_RAW、INT_ERR2_RAW 和 INT_ERR3_RAW 写 0x4。
- 步骤 3 向寄存器 C(2)_SRC_ADDR_L 和 C(2)_SRC_ADDR_H 写 0x011060000。
- 步骤 4 向寄存器 C(2)_DES_ADDR_L 和 C(2)_DES_ADDR_H 写 0x085000000。
- 步骤 5 向寄存器 C(2)_LLI_L[1:0]为 0x0。
- 步骤 6 向寄存器 C(2)_CNT0 写 0x10。
- 步骤 7 向寄存器 C(2)_CONFIG 写 0x40000008。
- 步骤 8 向寄存器 I2C0_I2C6_DMA_SEL 写 0x0，选择 I2C0 请求线。
- 步骤 9 向寄存器 C(2)_CONFIG[ch_en]为 0x1。
- 步骤 10 完成 I2C0 配置并打开 DMA RX 功能。

----结束

UART0 到内存的传输示例步骤如下：

- 步骤 1 读寄存器 CH_STAT，发现通道 2 空闲。
- 步骤 2 向寄存器 INT_TC1_RAW、INT_TC2_RAW、INT_ERR1_RAW、INT_ERR2_RAW 和 INT_ERR3_RAW 写 0x4。
- 步骤 3 向寄存器 C(2)_SRC_ADDR_L 和 C(2)_SRC_ADDR_H 写 0x011040000。



步骤 4 向寄存器 C(2)_DES_ADDR_L 和 C(2)_DES_ADDR_H 写 0x085000000。

步骤 5 向寄存器 C(2)_LLI_L[1:0]为 0x0。

步骤 6 向寄存器 C(2)_CNT0 写 0x10。

步骤 7 向寄存器 C(2)_CONFIG 写 0x40000148。

步骤 8 向寄存器 C(2)_CONFIG[ch_en]为 0x1。

步骤 9 完成 UART0 配置并打开 DMA RX 功能。

----结束

SSP0 到内存的传输示例步骤如下：

步骤 1 读寄存器 CH_STAT，发现通道 2 空闲。

步骤 2 向寄存器 INT_TC1_RAW、INT_TC2_RAW、INT_ERR1_RAW、INT_ERR2_RAW 和 INT_ERR3_RAW 写 0x4。

步骤 3 向寄存器 C(2)_SRC_ADDR_L 和 C(2)_SRC_ADDR_H 写 0x011070000。

步骤 4 向寄存器 C(2)_DES_ADDR_L 和 C(2)_DES_ADDR_H 写 0x085000000。

步骤 5 向寄存器 C(2)_LLI_L[1:0]为 0x0。

步骤 6 向寄存器 C(2)_CNT0 写 0x10。

步骤 7 向寄存器 C(2)_CONFIG 写 0x400110C8。

步骤 8 向寄存器 C(2)_CONFIG[ch_en]为 0x1。

步骤 9 完成 SSP0 配置并打开 DMA RX 功能。

----结束

3.6.4.8 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 $0x1 \ll n$ ，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_L` 和 `C(n)_SRC_ADDR_H`，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_L` 和 `C(n)_DES_ADDR_H`，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_L` [`lli_l`] 和 `C(n)_LLI_H` [`lli_h`]，设置链表空间的首地址。
- 步骤 7 写通道寄存器 `C(n)_LLI_L` [`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。

3.6.4.9 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：
- 步骤 5 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。
- 步骤 6 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。



步骤 7 退出中断处理。

步骤 8 错误中断处理。可以分为以下几个子步骤：

1. 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
2. 给出错误信息，有必要的话重新配置并启动该通道。
3. 退出中断处理。

步骤 9 链表节点传输完成中断处理。可以分为以下几个子步骤：

1. 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。
2. 取走或使用掉内存中的数据。
3. 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 3-6 所示。

表3-6 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 3-7 所示。

表3-7 DMAC 寄存器概览(基地址: 0x0_1028_0000)

偏移地址	名称	描述	页码
0x0000	INT_STAT	中断状态寄存器	3-173
0x0004	INT_TC1	通道传输完成中断状态寄存器	3-174
0x0008	INT_TC2	链表节点传输完成中断状态寄存器	3-174
0x000C	INT_ERR1	配置错误中断状态寄存器	3-175



偏移地址	名称	描述	页码
0x0010	INT_ERR2	数据传输错误中断状态寄存器	3-175
0x0014	INT_ERR3	读链表错误中断状态寄存器	3-175
0x0018	INT_TC1_MASK	通道传输完成中断屏蔽寄存器	3-176
0x001C	INT_TC2_MASK	链表节点传输完成中断屏蔽寄存器	3-176
0x0020	INT_ERR1_MASK	配置错误中断屏蔽寄存器	3-177
0x0024	INT_ERR2_MASK	数据传输错误中断屏蔽寄存器	3-177
0x0028	INT_ERR3_MASK	链表读取错误中断屏蔽寄存器	3-177
0x0404 + 0x20 × n	C(n)_CURR_CNT0	通道 n 的剩余数据量寄存器	3-178
0x0408 + 0x20 × n	C(n)_CURR_SRC_ADDR_L	通道 n 的源地址低位寄存器	3-178
0x040C + 0x20 × n	C(n)_CURR_SRC_ADDR_H	通道 n 的源地址高位寄存器	3-178
0x0410 + 0x20 × n	C(n)_CURR_DES_ADDR_L	通道 n 的目的地址低位寄存器	3-179
0x0414 + 0x20 × n	C(n)_CURR_DES_ADDR_H	通道 n 的目的地址高位寄存器	3-179
0x0600	INT_TC1_RAW	原始通道传输完成中断状态寄存器	3-179
0x0608	INT_TC2_RAW	原始链表节点传输完成中断状态寄存器	3-180
0x0610	INT_ERR1_RAW	原始配置错误中断状态寄存器	3-180
0x0618	INT_ERR2_RAW	原始数据传输错误中断状态寄存器	3-181
0x0620	INT_ERR3_RAW	原始链表读取错误中断状态寄存器	3-181
0x0688	CH_PRI	优先级控制寄存器	3-182
0x0690	CH_STAT	DMA 状态寄存器	3-184
0x0694	SEC_CTRL	全局安全控制寄存器	3-184



偏移地址	名称	描述	页码
0x0800 + 0x40 × n	C(n)_LLI_L	通道 n 的链表地址低位配置寄存器	3-185
0x0804 + 0x40 × n	C(n)_LLI_H	通道 n 的链表地址高位配置寄存器	3-185
0x081C + 0x40 × n	C(n)_CNT0	通道 n 的传输长度配置寄存器	3-186
0x0820 + 0x40 × n	C(n)_SRC_ADDR_ L	通道 n 的源地址低位配置寄存器	3-186
0x0824 + 0x40 × n	C(n)_SRC_ADDR_ H	通道 n 的源地址高位配置寄存器	3-186
0x0828 + 0x40 × n	C(n)_DES_ADDR_ L	通道 n 的目的地址低位配置寄存器	3-187
0x082C + 0x40 × n	C(n)_DES_ADDR_ H	通道 n 的目的地址高位配置寄存器	3-187
0x0830 + 0x40 × n	C(n)_CONFIG	通道 n 的配置寄存器	3-187
0x0834 + 0x40 × n	C(n)_AXI_CONF	通道 n 的 AXI 特殊操作配置寄存器	3-191
0x0f00	DMAC_VERSION	版本寄存器	3-191

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



Bits	Access	Name	Description	Reset
[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。 0: 未产生中断； 1: 已产生中断。	0x00



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



Bits	Access	Name	Description	Reset
[7:0]	RO	int_err3	DMAC 各通道经过屏蔽后的读链表错误中断状态, 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_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
[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,	0x00



			<p>bit[7:0]分别对应通道 7 ~ 0。</p> <p>0: 屏蔽中断;</p> <p>1: 不屏蔽中断。</p>	
--	--	--	---	--

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_L

C(n)_CURR_SRC_ADDR_L 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_l	通道 n 当前正在传输的源地址的低 32 位。	0x00000000

C(n)_CURR_SRC_ADDR_H

C(n)_CURR_SRC_ADDR_H 为通道 n 的源地址高位寄存器。

Offset Address: $0x040C + 0x20 \times n$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1:0]	RO	curr_src_addr_h	通道 n 当前正在传输的源地址的高 2 位。	0x0



C(n)_CURR_DES_ADDR_L

C(n)_CURR_DES_ADDR_L 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_l	通道 n 当前正在传输的目的地址低 32 位。	0x00000000

C(n)_CURR_DES_ADDR_H

C(n)_CURR_DES_ADDR_H 为通道 n 的目的地址高位寄存器。

Offset Address: $0x0414 + 0x20 \times n$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1:0]	RO	curr_des_addr_h	通道 n 当前正在传输的目的地址的高 2 位。	0x0

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。	0x00



Bits	Access	Name	Description	Reset
			写： 0：保持之前的值； 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，	0x00



Bits	Access	Name	Description	Reset
			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。	

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。	0x0



Bits	Access	Name	Description	Reset
			00: 0 优先级(最高); 01: 1 优先级; 10: 2 优先级; 11: 3 优先级(最低)。	
[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; 10: 2 优先级; 11: 3 优先级(最低)。	0x0
[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;	0x0



Bits	Access	Name	Description	Reset
			10: 2 优先级; 11: 3 优先级(最低)。	

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

SEC_CTRL

SEC_CTRL 为 DMA 全局安全控制寄存器。

Offset Address: 0x0694 Total Reset Value: 0x0000_0002

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[1]	RW	intr_sec_ctrl	中断安全控制寄存器(只能通过安全操作访问)。 0: 安全操作只能访问安全通道的中断信息, 安全中断通过 DMAC 上报, 非安全中断通过 DMAC_NS 上报; 1: 安全操作能访问所有通道的信息, 安全中断和非安全中断都会通过 DMAC 上报, DMAC_NS 只上报非安全中断	0x1
[0]	RW	global_sec	全局寄存器的安全控制寄存器(只能通过安	0x0



Bits	Access	Name	Description	Reset
			全操作访问)。 0: CH_PRI 只有安全操作才能访问; 1: CH_PRI 可以受安全和非安全操作访问。	

C(n)_LLI_L

C(n)_LLI_L 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: $0x0800 + 0x40 \times n$ Total Reset Value: $0x0000_0000$

Bits	Access	Name	Description	Reset
[31:6]	RW	lli_l	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)_LLI_H

C(n)_LLI_H 为通道 n 的链表地址高位配置寄存器。

Offset Address: $0x0804 + 0x40 \times n$ Total Reset Value: $0x0000_0000$

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1:0]	RW	lli_h	通道 n 的下一个链表节点地址的[33:32]位, 链表地址始终是 512-bit 对齐。	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； Acount 配置为 0 会触发配置错误中断。	0x0000

C(n)_SRC_ADDR_L

C(n)_SRC_ADDR_L 为通道 n 的源地址低位配置寄存器。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_l	通道 n 的源地址低 32 位。	0x00000000

C(n)_SRC_ADDR_H

C(n)_SRC_ADDR_H 为通道 n 的源地址高位配置寄存器。

Offset Address: $0x0824 + 0x40 \times n$ Total Reset Value: $0x0000_0000$

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1:0]	RW	src_addr_h	通道 n 的源地址高 2 位。	0x0



C(n)_DES_ADDR_L

C(n)_DES_ADDR_L 为通道 n 的目的地址低位配置寄存器。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_l	通道 n 的目的地址低 32 位。	0x00000000

C(n)_DES_ADDR_H

C(n)_DES_ADDR_H 为通道 n 的目的地址高位配置寄存器。

Offset Address: $0x082C + 0x40 \times n$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1:0]	RW	des_addr_h	通道 n 的目的地址高 2 位。	0x0

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。 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。	0x0
[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。	0x0



Bits	Access	Name	Description	Reset
			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。	
[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;	0x0



Bits	Access	Name	Description	Reset
			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。	
[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



Bits	Access	Name	Description	Reset
[3:2]	RW	flow_ctrl	<p>流控制和传输类型。</p> <p>00: 存储器和存储器之间传输, DMAC 流控;</p> <p>01: 存储器和外设之间传输, DMAC 流控;</p> <p>10: 存储器和外设之间传输, 外设流控; 。</p> <p>11: 保留</p>	0x0
[1]	RW	itc_en	<p>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。</p> <p>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;</p> <p>1: 当前链表节点传输完成以后触发链表节点传输完成中断, 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。</p> <p>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。</p>	0x0
[0]	RW	ch_en	<p>当前通道使能控制。</p> <p>当写该寄存器, 控制当前通道的运行:</p> <p>0: 通道禁能;</p> <p>1: 通道使能。</p> <p>当读该寄存器, 显示当前通道的状态:</p> <p>0: 通道停止中;</p> <p>1: 通道运行中(链表传输使能时, 通道一直使能)。</p>	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_C310

Bits	Access	Name	Description	Reset
[31:0]	RO	dmac_version	控制器版本信息。	0x0000C310



3.7 CRC 循环冗余校验

3.7.1 概述

CRC 用来校验数据传输或存储后是否出现错误。

3.7.2 特点

CRC 具有以下特点：

- 支持 APB2.0 接口。
- 支持 CRC 计算输入数据可由 FIFO 缓存，缓存大小为 16×32。
- 支持 FIFO 满时可反压 APB 总线；
- 支持产生 CRC 校验码，并可由寄存器回读；
- 支持 5 种标准多项式，分别为 CRC8-SAE、CRC8_8H2F、CRC16-CCITT、CRC32P4、CRC32 Ethernet。

表3-8 CRC 多项式规格

CRC definition	CRC8-SAE	CRC8_8H2F	CRC16-CCITT	CRC32P4	CRC32 Ethernet
CRC 宽度 (bits)	8	8	16	32	32
CRC 多项式	0x1D	0x2F	0x1021	0xF4ACFB13	0x04C11DB7
初始化值	0x0	0x0	0xFFFF	0x0	0x0
输入数据高低互换	NO	NO	No	Yes	Yes
输出数据高低互换	NO	NO	No	Yes	Yes
输出结果按位取反	YES	YES	NO	Yes	Yes

- CRC 的写入数据支持 8bit/16bit/32bit 有效可配置；
- CRC 每个工作时钟计算数据量为 1Byte。
- CRC 仅支持输入数据小端对齐；
- 支持对 CRC 校验码校验，并通过寄存器上报校验是否成功；
- 支持寄存器读取计算结束标志位。
- APB 总线异常反压可上报中断。



3.7.3 功能描述

CRC 支持如下功能：

- 支持计算输出 CRC 值；
- 支持 CRC 初始值可配置；
- 支持 CRC 值校验。

3.7.4 工作方式

3.7.4.1 Generate CRC

CRC 可根据输入的源数据产生校验码，具体步骤如下：

步骤 1 配置 CRC_CTRL_CFG0 寄存器，选择 CRC 多项式和 APB 写入数据有效 bit 位。

步骤 2 配置 CRC_CTRL_CFG1 寄存器，初始化 CRC 模块。

注意：若无需更改初始化值，则跳过步骤 3 和步骤 4，直接进行步骤 5。

步骤 3 配置 CRC_CTRL_DATA0 寄存器，更改初始化值。

步骤 4 配置 CRC_CTRL_CFG2 寄存器，使能初始化值修改。

步骤 5 向 CRC_CTRL_DATA1 寄存器中写入需要校验的源数据。

步骤 6 反复读取 CRC_CTRL_STATUS1 寄存器，直到该寄存器[0]值为 1，代表当前计算结束。

步骤 7 读取 CRC_CTRL_DATA2 寄存器，获取校验码。

----结束

3.7.4.2 CRC Check

CRC 模块可校验传输或存储后数据是否正确，在 CRC_CTRL_STATUS0[0]显示校验是否成功。

步骤 1 **输入源数据（先参考 Generate CRC 章节步骤 1~步骤 6）。**

步骤 2 数据接收完成后，继续向 [CRC_CTRL_DATA1](#) 中写入 CRC 校验码。

步骤 3 读取 [CRC_CTRL_STATUS0](#)，确认此寄存器[0]的标志位是否拉高。

----结束



不同多项式对 CRC 校验码的输入有不同要求，具体请参考表 3-4 所示。

表3-9 CRC 校验码输入要求

CRC 多项式	CRC8-SAE	CRC8_8H2F	CRC16-CCITT	CRC32P4	CRC32 Ethernet
CRC 校验值输入要求	crc_value [7:0]	crc_value [7:0]	crc_value [15:0]	{crc_value[7:0], crc_value[15:8], crc_value[23:16], crc_value[31:24]}	{crc_value[7:0], crc_value[15:8], crc_value[23:16], crc_value[31:24]}

注：crc_value 为 CRC 校验码。

3.7.5 CRC 寄存器概览

CRC 寄存器概览如表 3-10 所示。

表3-10 CRC 寄存器概览 (基址是 0x0_1100_8000)

偏移地址	名称	描述	页码
0x0000	CRC_CTRL_CFG0	CRC_CTRL 控制寄存器 0	3-195
0x0004	CRC_CTRL_CFG1	CRC_CTRL 控制寄存器 1	3-195
0x0008	CRC_CTRL_CFG2	CRC_CTRL 控制寄存器 2	3-196
0x0010	CRC_CTRL_DATA0	CRC_CTRL 数据寄存器 0	3-196
0x0014	CRC_CTRL_DATA1	CRC_CTRL 数据寄存器 1	3-196
0x0018	CRC_CTRL_DATA2	CRC_CTRL 数据寄存器 2	3-197
0x001C	CRC_CTRL_STATUS0	CRC_CTRL 状态寄存器 0	3-197
0x0020	CRC_CTRL_INT	APB 总线状态中断寄存器	3-197
0x0024	CRC_CTRL_INTMASK0	APB 总线状态中断屏蔽寄存器	3-198
0x0030	CRC_CTRL_INT_ERR_INJ	APB 总线状态错误注入寄存器	3-198
0x0034	CRC_CTRL_STATUS1	CRC_CTRL 状态寄存器 1	3-198



3.7.6 CRC 寄存器描述

CRC_CTRL_CFG0

CRC_CTRL_CFG0 为 CRC_CTRL 控制寄存器 0。

此寄存器可配置 CRC 多项式和数据有效 bit 位。

Offset Address: 0x0000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x00000000
[4:2]	RW	crc_mode	CRC 多项式选择。 000: CRC8-SAE 001: CRC8_8H2F 01x: CRC16-CCITT 10x: CRC32P4 11x: CRC32 Ethernet 上述 x 代表 0 或者 1 都可以。	0x0
[1:0]	RW	crc_byte_mode	总线写入 CRC_CTRL_DATA1 寄存器数据 bit 有效控制位。 00: 低 8bit 有效; 01: 低 16bit 有效; 11: 32bit 有效; 其他: 保留。	0x0

CRC_CTRL_CFG1

CRC_CTRL_CFG1 为 CRC_CTRL 控制寄存器 1。

此寄存器可配置 CRC 初始化值。

Offset Address: 0x0004 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	WO	crc_init	CRC 初始化信号，高有效，单周期脉冲信号。	0x0

CRC_CTRL_CFG2

CRC_CTRL_CFG2 为 CRC_CTRL 控制寄存器 2。

此寄存器可配置 CRC_CTRL_DATA0 的有效信号。

Offset Address: 0x0008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	WO	crc_ld	CRC 的 crc_ld 信号有效时，使能初始化值 CRC_CTRL_DATA0 修改(多项式不同 CRC_CTRL_DATA2 读出的初始化值可能经过按位取反或高低位互换)，高有效，单周期脉冲信号。	0x0

CRC_CTRL_DATA0

CRC_CTRL_DATA0 为 CRC_CTRL 数据寄存器 0。

此寄存器可配置 CRC 初始化值。

Offset Address: 0x0010 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	crc_check_in	初始化值配置寄存器。	0x00000000

CRC_CTRL_DATA1

CRC_CTRL_DATA1 为 CRC_CTRL 数据寄存器 1。



Offset Address: 0x0014 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	crc_data_in	需要计算的 CRC 源数据，不足位补零。	0x00000000

CRC_CTRL_DATA2

CRC_CTRL_DATA2 为 CRC_CTRL 数据寄存器 2。

Offset Address: 0x0018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	crc_out	CRC 计算结果寄存器，不足位补零。	0x00000000

CRC_CTRL_STATUS0

CRC_CTRL_STATUS0 为 CRC_CTRL 状态寄存器 0。

此寄存器输出 CRC 内部 FIFO 状态和 CRC 校验结果标志位。

Offset Address: 0x001C Total Reset Value: 0x0000_0004

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2]	RO	crc_fifo_empty	FIFO 空标志位，高有效。	0x1
[1]	RO	crc_fifo_full	FIFO 满标志位，高有效。	0x0
[0]	RO	crc_ok	CRC 校验正确标志位，高有效。	0x0

CRC_CTRL_INT

CRC_CTRL_INT 为 APB 总线状态中断寄存器。

Offset Address: 0x0020 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000



Bits	Access	Name	Description	Reset
[0]	INT_WC	time_out_err	APB 总线状态中断寄存器。	0x0

CRC_CTRL_INTMASK0

CRC_CTRL_INTMASK0 为 APB 总线状态中断屏蔽寄存器。

Offset Address: 0x0024 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	int_ras_safety_mask	APB 总线状态中断屏蔽寄存器。 0: 不屏蔽; 1: 屏蔽。	0x1

CRC_CTRL_INT_ERR_INJ

CRC_CTRL_INT_ERR_INJ 为 APB 总线状态错误注入寄存器。

Offset Address: 0x0030 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	time_out_err_inj	CRC_CTRL_INT 源中断错误注入。 0: 无错误注入; 1: 错误注入。	0x0

CRC_CTRL_STATUS1

CRC_CTRL_STATUS1 为 CRC_CTRL 状态寄存器 1。

Offset Address: 0x0034 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	cal_done	CRC 计算结束标志位，拉高时代表计算结束。	0x0

3.8 定时器

3.8.1 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 12 个 Timer，其中 4 个仅支持安全 OS 访问。

3.8.2 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。
- Timer 的计数时钟为 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。

3.8.3 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。

3.8.4 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 `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

3.8.5 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 基址：0x0_1100_0000
- TIMER1 基址：0x0_1100_0020
- TIMER2 基址：0x0_1100_1000
- TIMER3 基址：0x0_1100_1020
- TIMER4 基址：0x0_1100_2000
- TIMER5 基址：0x0_1100_2020
- TIMER6 基址：0x0_1100_3000
- TIMER7 基址：0x0_1100_3020
- TIMER8(Sec_Timer0)基址：0x0_1100_4000
- TIMER9(Sec_Timer1)基址：0x0_1100_4020
- TIMER10(Sec_Timer2)基址：0x0_1100_5000
- TIMER11(Sec_Timer3)基址：0x0_1100_5020

说明

TIMERx 中的“x”取值为 0~11，各控制器的寄存器相同，寄存器描述均以 Timer0 为例进行描述。Timer8~Timer11 可配置为仅安全系统访问。



表3-11 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	3-202
0x004	TIMERx_VALUE	当前计数值寄存器	3-203
0x008	TIMERx_CONTROL	Timer 控制寄存器	3-203
0x00C	TIMERx_INTCLR	中断清除寄存器	3-204
0x010	TIMERx_RIS	原始中断寄存器	3-205
0x014	TIMERx_MIS	屏蔽后中断寄存器	3-205
0x018	TIMERx_BGLOAD	周期模式计数初值寄存器	3-205

3.8.6 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：

向 TIMERx_BGLOAD 寄存器和 TIMERx_LOAD 寄存器都写入值，则在 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



Bits	Access	Name	Description	Reset
[5]	RW	intenable	TIMERx_RIS 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。

须知

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

3.9 WatchDog

3.9.1 概述

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 1 个 WatchDog 模块。

3.9.2 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。

3.9.3 功能描述

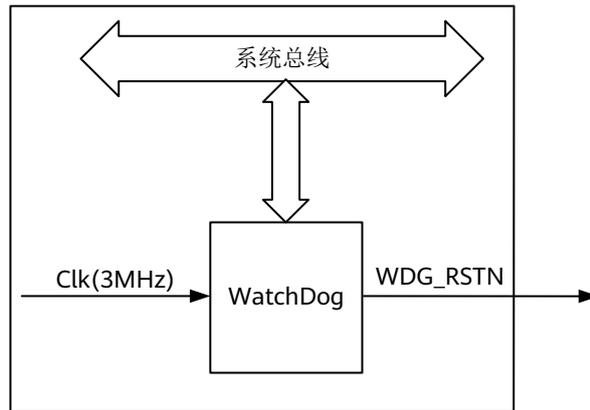
应用框图

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（当

芯片使用内置 POR 时，WatchDog 产生的复位信号会复位芯片同时通过 WDG_RSTN 管脚输出。当芯片使用外置 POR 时，WatchDog 产生的复位信号通过 WDG_RSTN 管脚输出给外置复位电路，然后该复位电路产生的复位信号通过 RSTN 输入管脚复位整芯片，使系统复位，达到监控系统运行的目的。

WatchDog 应用框图如图 3-7 所示。

图3-7 WatchDog 应用框图



功能原理

WatchDog 的运行基于 1 个 32bit 减法计数器，计数初值由寄存器 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，处理器还没有清除 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。

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。

3.9.4 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} ：

$$T_{WDG} = \text{Value}_{WDG_LOAD} \times \left(\frac{1}{f_{clk}} \right)$$

说明

其中上式中，各参数代表的意思分别是：

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

WatchDog 的计数时间范围值为 0s ~ 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。

3.9.5 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。

WatchDog 寄存器概览如表 3-12 所示。

表3-12 WatchDog 寄存器概览（基址是 `0x0_1103_0000`）

偏移地址	名称	描述	页码
0x0000	WDG_LOAD	计数初值寄存器	3-210
0x0004	WDG_VALUE	计数器当前值寄存器	3-210
0x0008	WDG_CONTROL	控制寄存器	3-210
0x000C	WDG_INTCLR	中断清除寄存器	3-211
0x0010	WDG_RIS	原始中断寄存器	3-211
0x0014	WDG_MIS	屏蔽后中断寄存器	3-212



偏移地址	名称	描述	页码
0x0C00	WDG_LOCK	LOCK 寄存器	3-212

3.9.6 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，计数最小初始值应大于 0。

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。	0x0



			0: 禁止; 1: 使能。	
[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

3.10 实时时钟

3.10.1 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。



3.10.2 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
- 支持固定分频模式
- 64bit 用户寄存器提供用户保存数据

3.10.3 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。



3.10.4 工作方式

3.10.4.1 计数时钟频率

RTC 采用 100Hz 时钟进行计数，计数最大时间为：

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

说明

T_{RTC} 表示 RTC 计数时间。

3.10.4.2 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

3.10.4.3 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_MR_D_H，设置 RTC 计数初始值。

步骤 6 配置 RTC_LOAD 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

3.10.4.4 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_LOAD` 为 1，让 RTC 计数初始值加载 RTC 计数器中。

步骤 3 等待 10ms。

----结束

3.10.4.5 RTC 时间读取

RTC 时间读取过程如下：

步骤 1 配置 `RTC_LOAD` 为 0x2。

步骤 2 回读 `RTC_LOAD` 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

3.10.4.6 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_H、RTC_MR_D_L、RTC_MR_D_H 写入新的比较值。

----结束

3.10.5 RTC 寄存器概览

RTC 寄存器概览如表 3-13 所示。

表3-13 RTC 内部寄存器概览 (基址是 0x0_1111_0000)

偏移地址	名称	描述	页码
0x0000	RTC_10MS_COUNT	RTC 10ms 计数值寄存器	3-218
0x0004	RTC_S_COUNT	RTC 秒计数值寄存器	3-218
0x0008	RTC_M_COUNT	RTC 分计数值寄存器	3-219
0x000C	RTC_H_COUNT	RTC 时计数值寄存器	3-219
0x0010	RTC_D_COUNT_L	RTC 天计数值低 8 位寄存器	3-219
0x0014	RTC_D_COUNT_H	RTC 天计数值高 8 位寄存器	3-220
0x0018	RTC_MR_10MS	RTC 10ms 定时值寄存器	3-220
0x001C	RTC_MR_S	RTC 秒定时值寄存器	3-220
0x0020	RTC_MR_M	RTC 分定时值寄存器	3-221
0x0024	RTC_MR_H	RTC 时定时值寄存器	3-221
0x0028	RTC_MR_D_L	RTC 天定时值低 8 位寄存器	3-221
0x002C	RTC_MR_D_H	RTC 天定时值高 8 位寄存器	3-222
0x0030	RTC_LR_10MS	RTC 10ms 定时值寄存器	3-222
0x0034	RTC_LR_S	RTC 秒定时值寄存器	3-222
0x0038	RTC_LR_M	RTC 分定时值寄存器	3-223
0x003C	RTC_LR_H	RTC 时定时值寄存器	3-223
0x0040	RTC_LR_D_L	RTC 天定时值低 8 位寄存器	3-223



偏移地址	名称	描述	页码
0x0044	RTC_LR_D_H	RTC 天定时值高 8 位寄存器	3-224
0x0048	RTC_LOAD	RTC 设置值使能加载寄存器	3-224
0x004C	RTC_IMSC	RTC 中断使能寄存器	3-225
0x0050	RTC_INT_CLR	RTC 中断清除寄存器	3-225
0x0054	RTC_MSC_INT	RTC mask 中断状态寄存器	3-225
0x0058	RTC_RAW_INT	RTC 原始中断状态寄存器	3-226
0x005C	RTC_CLK	RTC 输出时钟选择寄存器	3-226
0x0060	RTC_POR_N	RTC 复位控制寄存器	3-227
0x0068	RTC_UV_CTRL	RTC 内部低压检测控制寄存器	3-227
0x006C	RTC_ANA_CTRL	RTC 振荡电流控制寄存器	3-228
0x0144	SDM_COEF_OUSID E_H	外部分频系数高 4 位寄存器	3-229
0x0148	SDM_COEF_OUSID E_L	外部分频系数低 8 位寄存器	3-229
0x014C	USER_REGISTER1	64bit 用户寄存器 1	3-229
0x0150	USER_REGISTER2	64bit 用户寄存器 2	3-230
0x0154	USER_REGISTER3	64bit 用户寄存器 3	3-230
0x0158	USER_REGISTER4	64bit 用户寄存器 4	3-230
0x015C	USER_REGISTER5	64bit 用户寄存器 5	3-231
0x0160	USER_REGISTER6	64bit 用户寄存器 6	3-231
0x0164	USER_REGISTER7	64bit 用户寄存器 7	3-231
0x0168	USER_REGISTER8	64bit 用户寄存器 8	3-231
0x0180	USER_DEFINE_REGI STER1	用户自定义寄存器 1	3-232
0x0184	USER_DEFINE_REGI STER2	用户自定义寄存器 2	3-232



偏移地址	名称	描述	页码
0x0188	USER_DEFINE_REGISTER3	用户自定义寄存器 3	3-232
0x018C	USER_DEFINE_REGISTER4	用户自定义寄存器 4	3-233
0x0190	RTC_REG_LOCK1	RTC 锁定寄存器 1	3-233
0x0194	RTC_REG_LOCK2	RTC 锁定寄存器 2	3-233
0x0198	RTC_REG_LOCK3	RTC 锁定寄存器 3	3-234
0x019C	RTC_REG_LOCK4	RTC 锁定寄存器 4	3-234

3.10.6 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: 0x0000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	保留。	0x0000000
[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：[0, 99]	0x00

RTC_S_COUNT

RTC_S_COUNT 为 RTC 秒计数值寄存器。

Offset Address: 0x0004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[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：[0, 59]	0x00



RTC_M_COUNT

RTC_M_COUNT 为 RTC 分计数值寄存器。

Offset Address: 0x0008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5:0]	RO	rtc_m_count	RTC 分计数器值，表示当前计时的分时间个数。取值范围：[0, 59]	0x00

RTC_H_COUNT

RTC_H_COUNT 为 RTC 时计数值寄存器。

Offset Address: 0x000C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4:0]	RO	rtc_h_count	RTC 时计数器值，表示当前计时的小时时间个数。取值范围：[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: 0x0010 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x0000000
[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：[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: 0x0014 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[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: [0, 65535]	0x00

RTC_MR_10MS

RTC_MR_10MS 为 RTC 10ms 定时值寄存器。

Offset Address: 0x0018 Total Reset Value: 0x0000_007f

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	保留。	0x0000000
[6:0]	RW	rtc_mr_10ms	RTC 10ms 定时值, 表示定时的 10ms 时间个数。取值范围: [0, 99]	0x7f

RTC_MR_S

RTC_MR_S 为 RTC 秒定时值寄存器。

Offset Address: 0x001C Total Reset Value: 0x0000_003f

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5:0]	RW	rtc_mr_s	RTC 秒定时值, 表示定时的秒时间个数。取值范围: [0, 59]	0x3f



RTC_MR_M

RTC_MR_M 为 RTC 分定时值寄存器。

Offset Address: 0x0020 Total Reset Value: 0x0000_003f

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5:0]	RW	rtc_mr_m	RTC 分定时值值，表示定时的分时间个数。取值范围：[0, 59]	0x3f

RTC_MR_H

RTC_MR_H 为 RTC 时定时值寄存器。

Offset Address: 0x0024 Total Reset Value: 0x0000_001f

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4:0]	RW	rtc_mr_h	RTC 时定时值值，表示定时的小时时间个数。取值范围：[0, 23]	0x1f

RTC_MR_D_L

RTC_MR_D_L 为 RTC 天定时值低 8 位寄存器。

Offset Address: 0x0028 Total Reset Value: 0x0000_00ff

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x0000000
[7:0]	RW	rtc_mr_d_l	RTC 天定时值值的低 8 位，与 RTC_MR_D_H 配合，表示定时的天时间个数，取值范围：[0, 65535]	0xff



RTC_MR_D_H

RTC_MR_D_H 为 RTC 天定时值高 8 位寄存器。

Offset Address: 0x002C Total Reset Value: 0x0000_00ff

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	rtc_mr_d_h	RTC 天定时值的高 8 位, 与 RTC_MR_D_L 配合, 表示定时的天时间个数, 取值范围: [0, 65535]	0xff

RTC_LR_10MS

RTC_LR_10MS 为 RTC 10ms 定时值寄存器。

Offset Address: 0x0030 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	保留。	0x0000000
[6:0]	RW	rtc_lr_10ms	RTC 10ms 定时值, 表示设置的 10ms 时间个数。取值范围: [0, 99]	0x00

RTC_LR_S

RTC_LR_S 为 RTC 秒定时值寄存器。

Offset Address: 0x0034 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5:0]	RW	rtc_lr_s	RTC 秒定时值, 表示设置的秒时间个数。取值范围: [0, 59]	0x00



RTC_LR_M

RTC_LR_M 为 RTC 分定时值寄存器。

Offset Address: 0x0038 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5:0]	RW	rtc_lr_m	RTC 分定时值，表示设置的分时间个数。 取值范围：[0, 59]	0x00

RTC_LR_H

RTC_LR_H 为 RTC 时定时值寄存器。

Offset Address: 0x003C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4:0]	RW	rtc_lr_h	RTC 时定时值，表示设置的小时时间个数。 取值范围：[0, 23]	0x00

RTC_LR_D_L

RTC_LR_D_L 为 RTC 天定时值低 8 位寄存器。

Offset Address: 0x0040 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x0000000
[7:0]	RW	rtc_lr_d_l	RTC 天设置值的低 8 位，与 RTC_LR_D_H 配合，表示设置的天时间个数，取值范 围：[0, 65535]	0x00



RTC_LR_D_H

RTC_LR_D_H 为 RTC 天定时值高 8 位寄存器。

Offset Address: 0x0044 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	rtc_lr_d_h	RTC 天设置值的高 8 位, 与 RTC_LR_D_L 配合, 表示设置的天时间个数, 取值范围: [0, 65535]	0x00

RTC_LOAD

RTC_LOAD 为 RTC 设置值使能加载寄存器。

Offset Address: 0x0048 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[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: 0x004C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[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; 1: 输出中断。	0x0

RTC_INT_CLR

RTC_INT_CLR 为 RTC 中断清除寄存器。

Offset Address: 0x0050 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[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，回读无意义。	0x0

RTC_MSC_INT

RTC_MSC_INT 为 RTC mask 中断状态寄存器。

Offset Address: 0x0054 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[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: 0x0058 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[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: 0x005C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000



[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: 0x0060 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[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。 1: 非复位, 正常工作。	0x1

RTC_UV_CTRL

RTC_UV_CTRL 为 RTC 内部低压检测控制寄存器。

Offset Address: 0x0068 Total Reset Value: 0x0000_0004

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x00000000
[5]	RW	bat_uv_ctrl_en	低压检测使能控制。 0: 关闭; 1: 打开。	0x0
[4]	RW	bat_uv_ctrl_sel	低压检测来源选择。 0: 不经过滤波处理; 1: 经过滤波处理。	0x0
[3:2]	RW	bat_uv_sel	电池欠压检测的阈值电压设置 (建议默认	0x1



Bits	Access	Name	Description	Reset
			为 01)。 00: 1.6V; 01: 1.8V; 10: 2V; 11: 2.2V。	
[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

RTC_ANA_CTRL

RTC_ANA_CTRL 为 RTC 振荡电流控制寄存器。

受 RTC 锁定寄存器锁定, 只有当{RTC_LOCK3, RTC_LOCK2, RTC_LOCK1, RTC_LOCK0}=0x5A5AABCD 时才可以被修改。

Offset Address: 0x006C Total Reset Value: 0x05

Bits	Access	Name	Description	Reset
[7:0]	RW	rtc_ana_ctrl	RTC 振荡电流控制。 0x00: 档位 0; 0x01: 档位 1; 0x02: 档位 2; 0x03: 档位 3; 其他: 保留。 备注: 档位值越大振荡电源越大。	0x5



SDM_COEF_OUSIDE_H

SDM_COEF_OUSIDE_H 为外部分频系数高 4 位寄存器。

Offset Address: 0x0144 Total Reset Value: 0x0000_0008

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[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: 0x0148 Total Reset Value: 0x0000_001B

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x0000000
[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 用户寄存器 1。

Offset Address: 0x014C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x0000000
[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 用户寄存器 2。

Offset Address: 0x0150 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[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 用户寄存器 3。

Offset Address: 0x0154 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[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 用户寄存器 4。

Offset Address: 0x0158 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[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 用户寄存器 5。

Offset Address: 0x015C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[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 用户寄存器 6。

Offset Address: 0x0160 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[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 用户寄存器 7。

Offset Address: 0x0164 Total Reset Value: 0x0000_0000

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



Offset Address: 0x0168 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	user_register8	64bit 用户使用寄存器 8, 对应 bit[63:56]。	0x00

USER_DEFINE_REGISTER1

USER_DEFINE_REGISTER1 为用户自定义寄存器 1。

Offset Address: 0x0180 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	user_define_register1	用户自定义寄存器 1。	0x00

USER_DEFINE_REGISTER2

USER_DEFINE_REGISTER2 为用户自定义寄存器 2。

Offset Address: 0x0184 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	user_define_register2	用户自定义寄存器 2。	0x00

USER_DEFINE_REGISTER3

USER_DEFINE_REGISTER3 为用户自定义寄存器 3。

Offset Address: 0x0188 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	user_define_register3	用户自定义寄存器 3。	0x00

USER_DEFINE_REGISTER4

USER_DEFINE_REGISTER4 为用户自定义寄存器 4。

Offset Address: 0x018C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	user_define_register4	用户自定义寄存器 4。	0x00

RTC_REG_LOCK1

RTC_REG_LOCK1 为 RTC 锁定寄存器 1。

Offset Address: 0x0190 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	rtc_reg_lock1	RTC 寄存器保护锁定寄存器 1。	0x00

RTC_REG_LOCK2

RTC_REG_LOCK2 为 RTC 锁定寄存器 2。

Offset Address: 0x0194 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	rtc_reg_lock2	RTC 寄存器保护锁定寄存器 2。	0x00



RTC_REG_LOCK3

RTC_REG_LOCK3 为 RTC 锁定寄存器 3。

Offset Address: 0x0198 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	rtc_reg_lock3	RTC 寄存器保护锁定寄存器 3。	0x00

RTC_REG_LOCK4

RTC_REG_LOCK4 为 RTC 锁定寄存器 4。

Offset Address: 0x019C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	rtc_reg_lock4	RTC 寄存器保护锁定寄存器 4。	0x00

3.11 电源管理与低功耗模式控制

3.11.1 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。
- 芯片的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。
- DDR 低功耗控制
DDR 的控制器进行了动态的功耗控制功能，可以选择启动该功能，降低芯片功耗。
- SVB (Selective Voltage Binning) 功能
- 芯片内部温度检测
芯片集成温度传感器(T-Sensor)，用于获取芯片内部温度。

3.11.2 PMC

3.11.2.1 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：

- 支持开关机按键及行车点火按键
- 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
- 支持管脚复用为 GPIO、I2C、UART 以及 PWM
- 支持启动参数、启动 eMMC 值以及 HASH 值存储

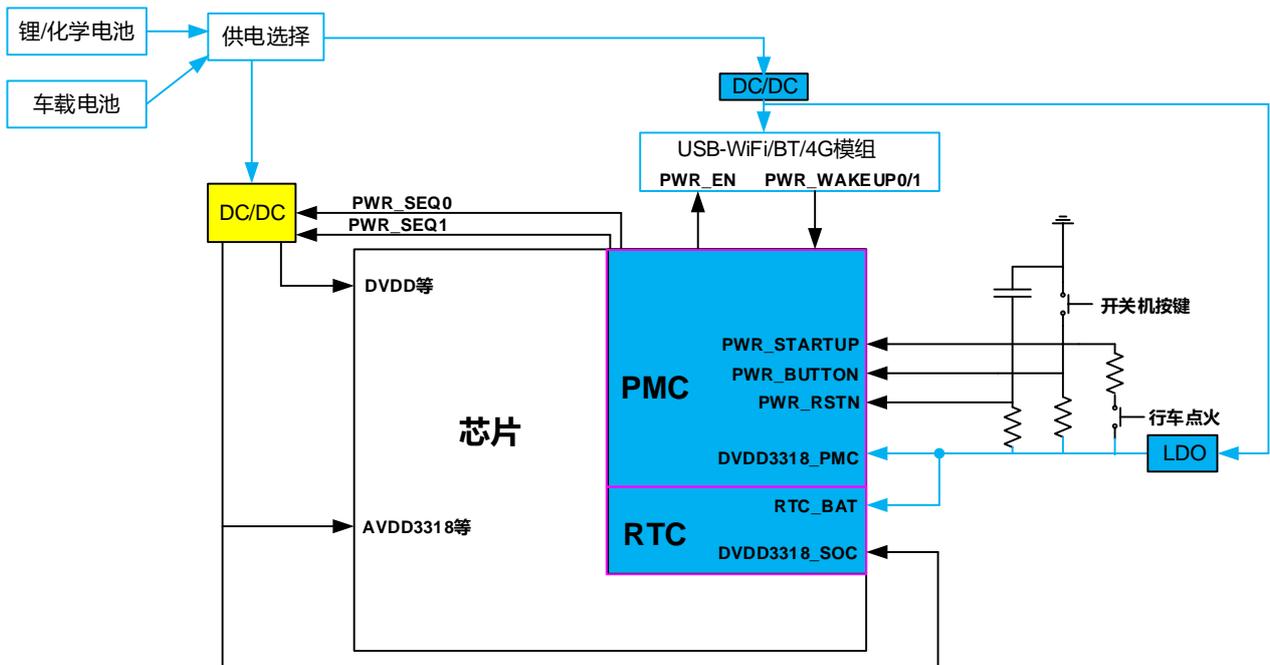
须知

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

PWR_X 管脚(X 代表管脚选项，具体请参考《XXXX_PINOUT_CN》)配置为 PMC 功能时，对应管脚的 GPIO、I2C、UART 以及 PWM 功能不可用。

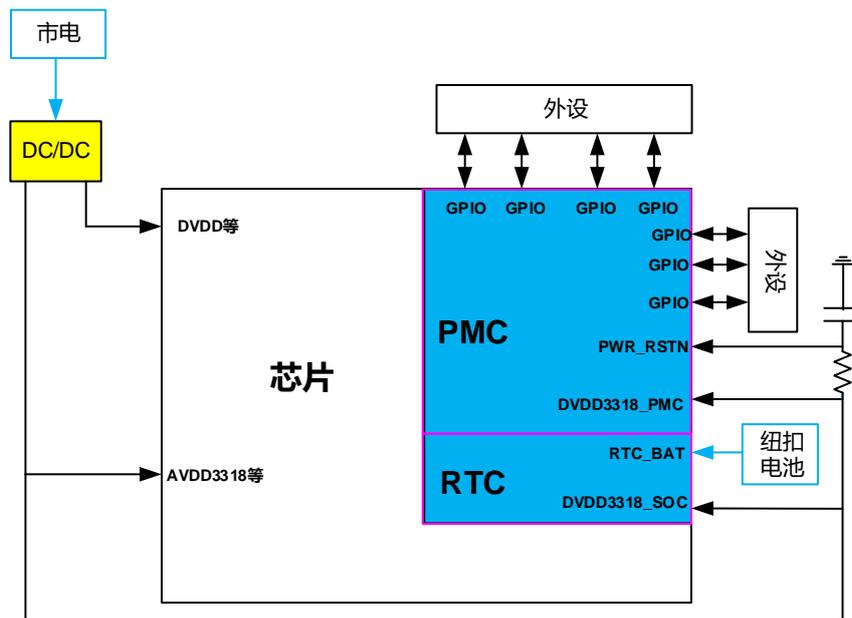
使用 PMC 模块进行芯片的上下电时序控制时，电源网络粗略框图如图 3-8 所示。其中 PMC 为恒电区，PWR_SEQ0 和 PWR_SEQ1 控制的芯片其它电源为可下电区，当处于关机状态下，芯片除了 PMC 模块，其它区域都下电，这时整个芯片处于最小的功耗状态。

图3-8 使用 PMC 时的电源网络粗略框图



当不使用 PMC 功能，但使用 PMC 管脚作为其他外设功能时的处理如图 3-9 所示。其中 PWR_RSTN 接 RC 复位，其他管脚配置外设功能。

图3-9 不使用 PMC 功能但使用 PMC 管脚

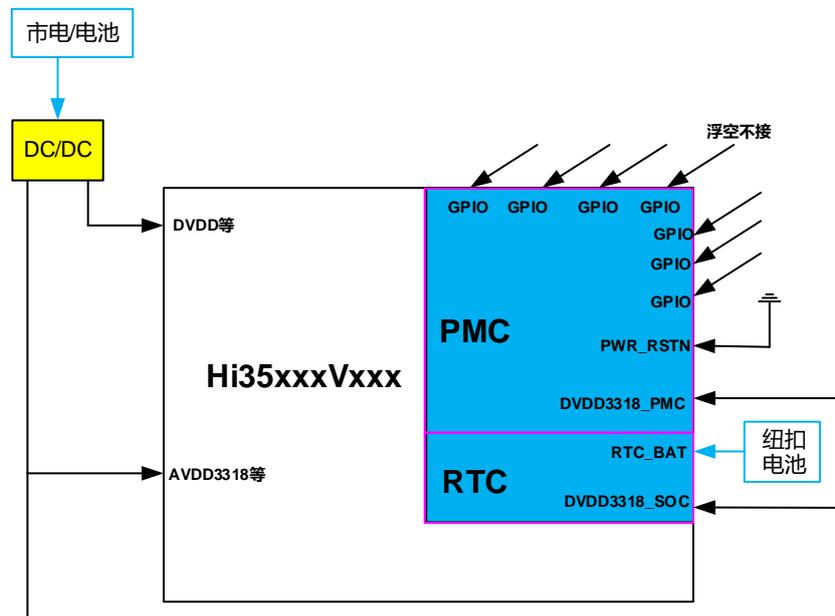


须知

PMC 默认管脚配置为 PMC 功能，在不使用 PMC 功能作为外设使用时，建议先将管脚默认配置为 GPIO 功能，防止管脚电平扰动触发 PMC 控制逻辑造成管脚间互扰。

当不使用 PMC 功能且不使用管脚时，相关管脚的处理如图 3-10 所示，其中 PWR_RSTN 接地，其它管脚浮空处理。但此时 DVDD3318_PMC 仍然需要接 SOC 侧的 3V3 电源。

图3-10 不使用 PMC 功能且不使用管脚时外围管脚的处理



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_WAKEUP0/1、PWR_STARTUP、和 PWR_BUTTON)或者 RTC 定时中断触发。

说明

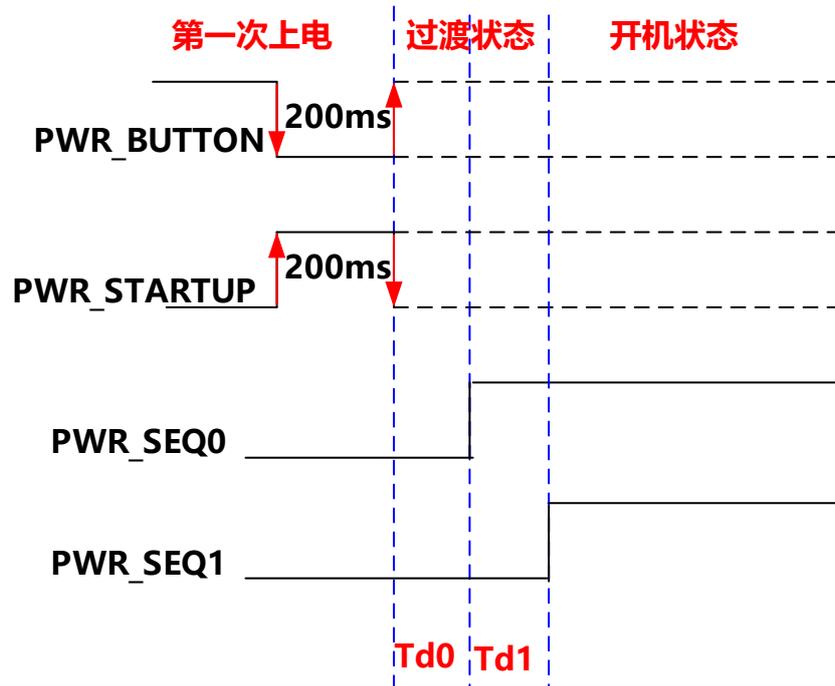
本章节中第一次上电状态、开机状态、关机状态名词说明:

- 第一次上电状态: 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 为 0, 在此状态下可以由 PWR_BUTTON、PWR_STARTUP、PWR_WAKEUP0/1、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 3-11 所示。



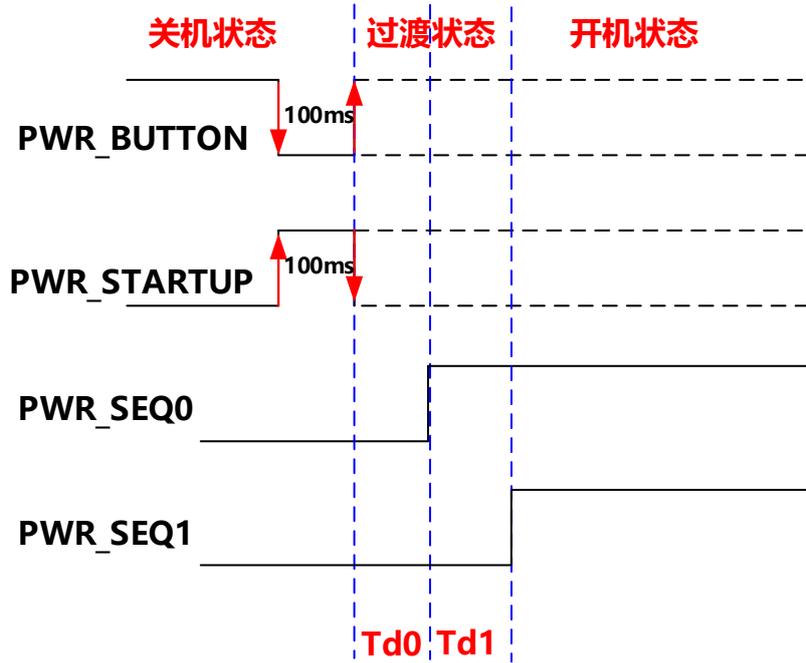
图3-11 PWR_BUTTON、PWR_STARTUP 第一次上电时序



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 0~25s；PWR_BUTTON、PWR_STARTUP 的上电时序如图 3-12 所示（以 100ms 为例）。



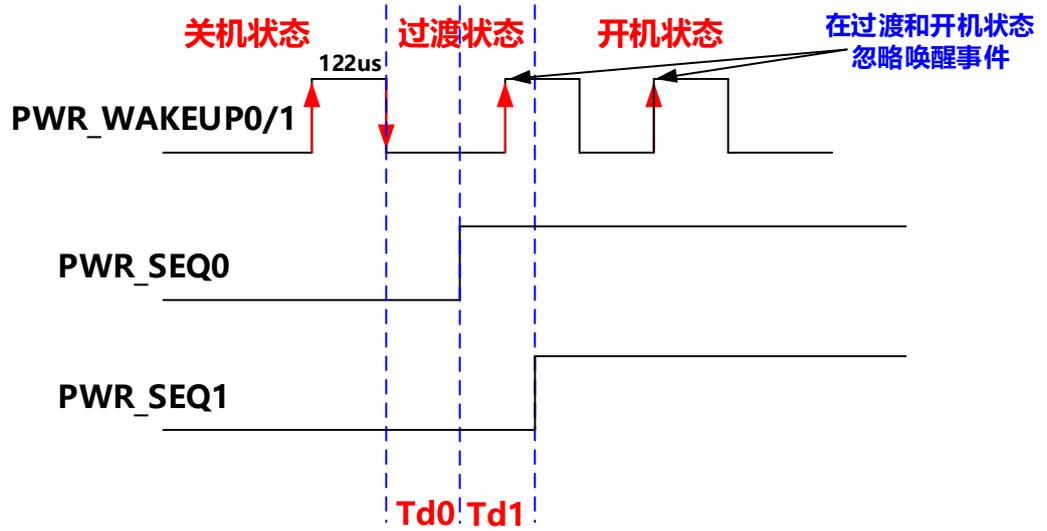
图3-12 PWR_BUTTON、PWR_STARTUP 上电时序



PWR_WAKEUP0/1 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 3-13 所示（以上升沿触发为例）。

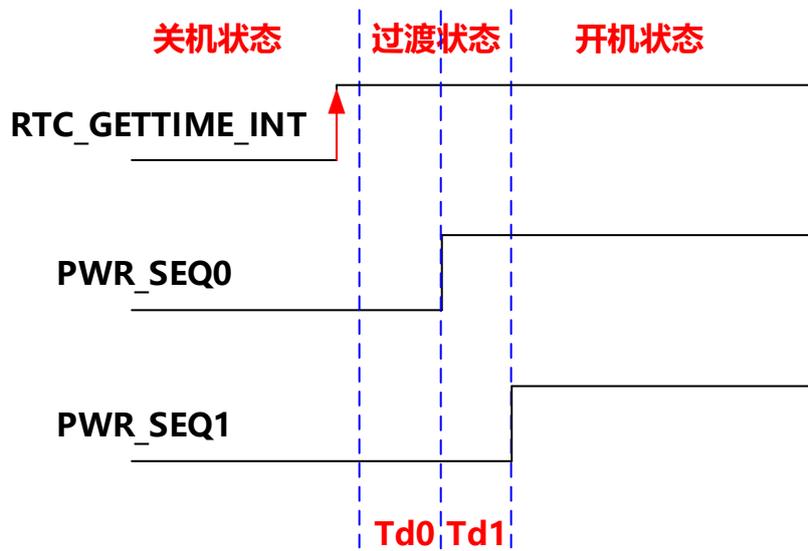


图3-13 PWR_WAKEUP 上电时序 (以上升沿触发为例)



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 3-14 所示。

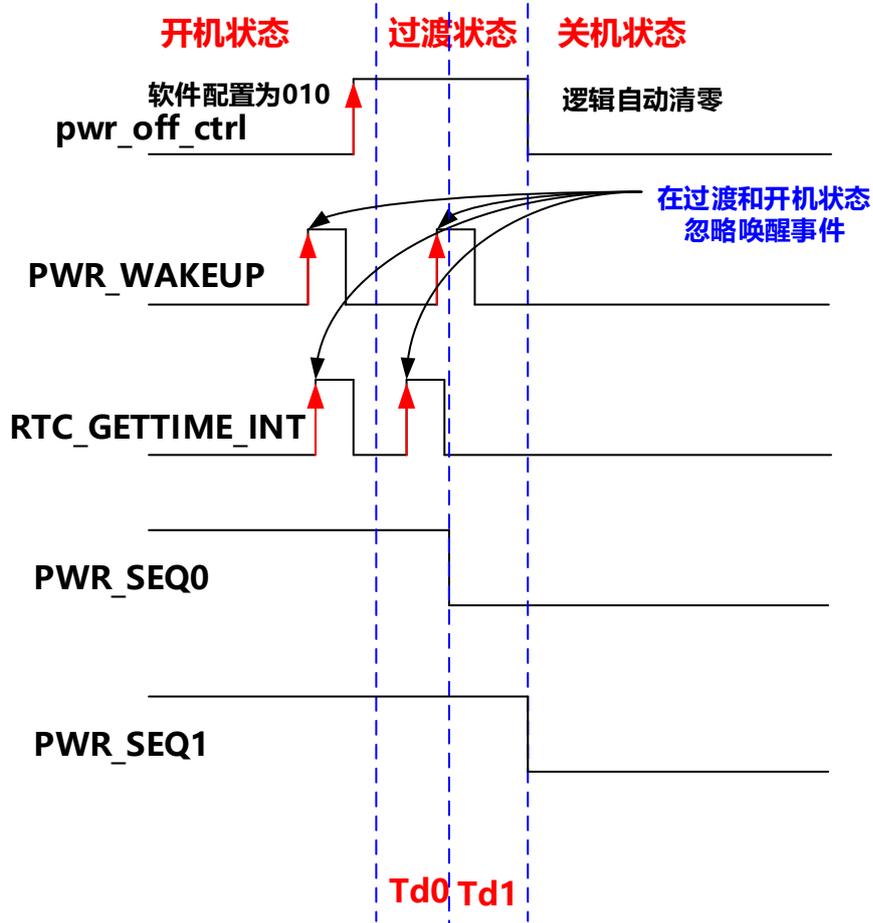
图3-14 RTC 定时中断上电时序





在开机状态下，通过配置寄存器 PWR_CTRL0[start_powerdown]为 1，即可触发下电时序。如图 3-15 所示(以下电顺序模式 1 为例)。

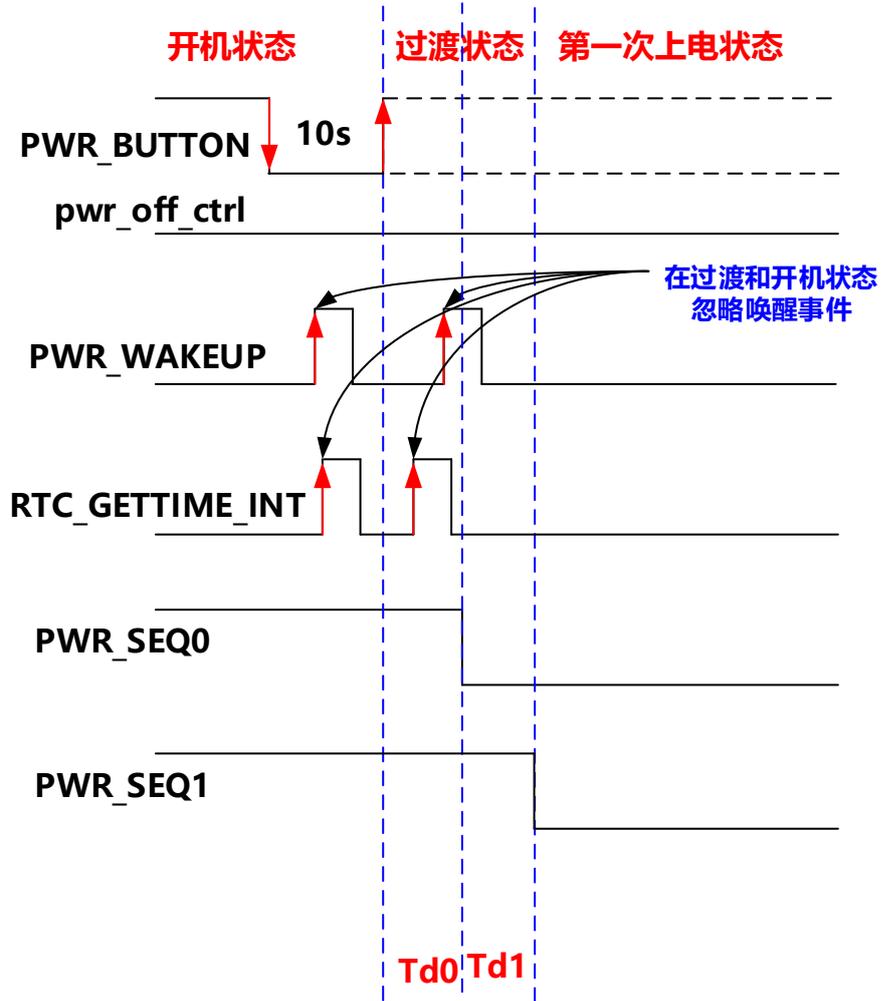
图3-15 软件配置下电时序(以下电模式 1 为例)



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 3-16 所示(以下电顺序模式 1 为例)。



图3-16 异常下电时序(以下电顺序模式 1 为例)



Td0/Td1 是可以编程的值，对应的寄存器为 PWR_CTRL1[pwr_td0/1]，取值范围为 1~255，计时单位为 ms，复位默认为 1。

PMC 对输入信号(PWR_WAKEUP0/1、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。



须知

- 唤醒源 PWR_WAKEUP0/1、PWR_STARTUP、PWR_BUTTON 只有在开机状态或者关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 寄存器中的 wakeup 和 rtc 唤醒屏蔽位，其余寄存器不复位。
- PWR_STARTUP 除无长按 10s 强制关机、以及在关机状态下检测上升沿开机外，其余功能与 PWR_BUTTON 相同，即 PWR_STARTUP 可以在关机下启动，在关机状态下产生中断。

上下电时序的电源的分组，根据“硬件”章节中的“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 1)。

3.11.2.2 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 0x0_1112_0000)

偏移地址	名称	描述	页码
0x000	PWR_CTRL0	PWR 控制寄存器 0	3-245
0x004	PWR_CTRL1	PWR 控制寄存器 1	3-247
0x008	PWR_STATUS	PWR 状态寄存器	3-247
0x00C	PWR_PD_SEL	PWR 下电时序控制寄存器	3-248
0x010	PWR_PUPD_CNT	PWR 掉电次数统计寄存器	3-248
0x014	PWR_USER_REG0	PWR 用户寄存器 0	3-249
0x018	PWR_USER_REG1	PWR 用户寄存器 1	3-249
0x01C	PWR_USER_REG2	PWR 用户寄存器 2	3-249
0x020	EMMC_PARAMS_CONFIG0	启动 eMMC 值存阵列寄存器 0	3-250
0x024	EMMC_PARAMS_CONFIG1	启动 eMMC 值存阵列寄存器 1	3-250
0x028	EMMC_PARAMS_CONFIG2	启动 eMMC 值存阵列寄存器 2	3-250



偏移地址	名称	描述	页码
0x02C	PWR_HASH_VALUE_LOCK_REG	PWR 存储启动 HASH 值锁存使能寄存器	3-251
(n×0x4) + 0x30	PWR_HASH_VALUE_REG	PWR 存储启动 HASH 值寄存器	3-251

寄存器偏移地址中变量的取值范围和含义如表 3-15 所示。

表3-15 寄存器偏移地址变量表

变量名称	取值范围	描述
n	0 ~ 63	存储启动 HASH 值的 64 个寄存器

3.11.2.3 PMC 寄存器描述

PWR_CTRL0

PWR_CTRL0 为 PWR 控制寄存器 0。

Offset Address: 0x000 Total Reset Value: 0x0010_0780

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	保留。	0x0
[28]	RW	high_adj_startup_en	STARTUP 开机时间配置使能。 0: 可配置; 1: 不可配置, 默认 200ms。	0x0
[27:20]	RW	high_adj_startup_time	STARTUP 开机时间配置, 单位为 ms, 范围为 100~25600, 以 100ms 的倍数可配。	0x01
[19]	-	reserved	保留。	0x0
[18]	RW	high_adj_button0_en	按键开机时间配置使能。 0: 可配置;	0x0



Bits	Access	Name	Description	Reset
			1: 不可配置, 默认 200ms。	
[17:10]	RW	high_adj_button0_time	按键开机时间配置, 单位为 ms, 范围为 100~25600, 以 100ms 的倍数可配。	0x01
[9]	RW	wakeup1_mask	是否屏蔽 WAKEUP1 唤醒。异常按键也可清除为默认值。 0: 不屏蔽; 1: 屏蔽。	0x1
[8]	RW	wakeup0_mask	是否屏蔽 WAKEUP0 唤醒。异常按键也可清除为默认值。 0: 不屏蔽; 1: 屏蔽。	0x1
[7]	RW	rtc_int_mask	是否屏蔽 RTC 中断唤醒。异常按键也可清除为默认值。 0: 不屏蔽; 1: 屏蔽。	0x1
[6:5]	RW	wakeup1_act_mode	WAKEUP1 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
[4:3]	RW	wakeup0_act_mode	WAKEUP0 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_pen 输出控制。 0: 输出 0; 1: 输出 1。	0x0



Bits	Access	Name	Description	Reset
[1]	WO	start_powerdown	开始关机流程, 写 1 关机。	0x0
[0]	RO	start_powerdown_state	关机状态。 0: 开机; 1: 关机。	0x0

PWR_CTRL1

PWR_CTRL1 为 PWR 控制寄存器 1。

Offset Address: 0x004 Total Reset Value: 0x0000_0101

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x0000
[16]	RW	pwr_cfg_rst	0: 不复位; 1: 复位。	0x0
[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。	0x01
[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。	0x01

PWR_STATUS

PWR_STATUS 为 PWR 状态寄存器。

Offset Address: 0x008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	保留。	0x000000
[10]	RO	startup_in	STRATUP 管脚输入值。	0x0
[9]	RO	wakeup1_in	WAKEUP1 管脚输入值。	0x0
[8]	RO	wakeup0_in	WAKEUP0 管脚输入值。	0x0



Bits	Access	Name	Description	Reset
[7]	RO	button_in	BUTTON 管脚输入值。	0x0
[6:2]	RO	wakeup_source	唤醒源。 0x0: WAKEUP0; 0x1: WAKEUP1; 0x2: STARTUP; 0x3: BUTTON; 0x4: RTC。	0x00
[1]	RO	poweron_from_poweroff	从关机进入开机状态的指示。 0: 不是; 1: 是。	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。 000: 模式 1, 下电时序为 SEQ0->SEQ1; 001: 模式 2, 下电时序为 SEQ1->SEQ0; 其它: 模式 1, 下电时序为 SEQ0->SEQ1。	0x0

PWR_PUPD_CNT

PWR_PUPD_CNT 为 PWR 掉电次数统计寄存器。

Offset Address: 0x010 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	RO	pwr_pd_cnt	下电次数统计。	0x0000
[15:0]	RO	pwr_pu_cnt	上电次数统计。	0x0000

PWR_USER_REG0

PWR_USER_REG0 为 PWR 用户寄存器 0。

Offset Address: 0x014 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	pwr_user_reg0	Flash 中的启动镜像的起始地址。当值为 0x0 时选择从原始镜像启动。	0x00000000

PWR_USER_REG1

PWR_USER_REG1 为 PWR 用户寄存器 1。

Offset Address: 0x018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	pwr_user_reg1	启动镜像大小, 包括 Flash 公钥, GSL, 启动参数和 Uboot。	0x00000000

PWR_USER_REG2

PWR_USER_REG2 为 PWR 用户寄存器 2。

Offset Address: 0x01C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	RW	pwr_user_reg2	PWR 用户寄存器 2。	0x000000
[9:8]	RW	verify_backup_image	表示从原始镜像或备份镜像启动的标志位。 0x0: 原始镜像;	0x0



Bits	Access	Name	Description	Reset
			0x1: 备份镜像; 0x2/0x3: 原始镜像和备份镜像启动失败。	
[7:0]	RW	low_power_mode	安全子系统的低功耗模式。 0x5A: 待机到 LPDS; 其它: 正常启动。	0x00

EMMC_PARAMS_CONFIG0

EMMC_PARAMS_CONFIG0 为启动 eMMC 值存阵列寄存器 0。

Offset Address: 0x020 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	emmc_params_config0	启动 eMMC 值存阵列。	0x00000000

EMMC_PARAMS_CONFIG1

EMMC_PARAMS_CONFIG1 为启动 eMMC 值存阵列寄存器 1。

Offset Address: 0x024 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	emmc_params_config1	启动 eMMC 值存阵列。	0x00000000

EMMC_PARAMS_CONFIG2

EMMC_PARAMS_CONFIG2 为启动 eMMC 值存阵列寄存器 2。

Offset Address: 0x028 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	emmc_params_config2	启动 eMMC 值存阵列。	0x00000000



PWR_HASH_VALUE_LOCK_REG

PWR_HASH_VALUE_LOCK_REG 为 PWR 存储启动 HASH 值锁存使能寄存器，受硬复位和掉电区系统软复位影响

Offset Address: 0x02C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	lock_pwr_hash_value	写 1 锁存寄存器。	0x0

PWR_HASH_VALUE_REG

PWR_HASH_VALUE_REG 为 PWR 存储启动 HASH 值寄存器，仅受硬复位影响。

Offset Address: (n×0x4) + 0x30 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	pwr_hash_value	启动 HASH 值存储，lock 寄存器。	0x00000000

3.11.3 时钟门控和时钟频率调整

关闭不需要的时钟

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。

调整模块工作频率

除了提供系统工作频率的调整外，部分模块的工作频率也可以单独进行调整。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。



3.11.4 模块级低功耗控制

芯片各个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。

3.11.5 DDR 低功耗控制

关于 DDRC 的低功耗控制的详细内容请参见“DDR 控制器”章节中 4.3.2 的“低功耗管理”部分。

3.11.6 SVB 功能描述

SVB (Selective Voltage Binning) 是根据芯片工艺、温度、电路时序情况，静态调整电压来进一步降低功耗。系统可由 HPM 模块获取芯片的工艺相关信息，并通过 SVB PWM 输出信号实现电压的自动调整。

- HPM 控制
通过 HPM 获取当前芯片的工艺相关信息，根据这些信息产生芯片供电电压调节。
- SVB PWM
芯片中包含可编程的 SVB PWM 模块，外围调压电路可通过芯片的 SVB PWM 输出，调节芯片的供电电压。芯片提供 3 个 SVB_PWM 输出，分别控制对应的电源电压调节。

3.11.6.1 芯片内部温度检测

芯片内部集成 2 个温度传感器(T-Sensor)，温度的检测范围为-40~125°C。使能温度传感器数据采集的流程如下：

步骤 1 设置 T-Sensor 采集模式 TSENSOR_CTRL0 [30]。

步骤 2 如果设置为循环采集模式，需设置循环采集周期 TSENSOR_CTRL0 [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 $N = TSENSOR_CTRL0 [27:20]$ 。

步骤 3 使能 T-Sensor TSENSOR_CTRL0 [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 TSENSOR_CTRL2[9:0]中记录的温度记录码 0 有效。

循环采集模式下，TSENSOR_CTRL2[31:0]~ TSENSOR_CTRL5 [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 = $[(\text{tsensor_result}-132)/808]*165-40(^{\circ}\text{C})$ ，tsensor_result 取值范围[132, 940]

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

---结束

3.11.6.2 SVB PWM 电压调节

芯片中有 3 个可编程的 SVB PWM 输出，分别对应管脚 SVB_PWM0/SVB_PWM1/SVB_PWM2。

对于每组 SVB_PWM 输出：

- 3MHz、24MHz 和 50MHz 工作时钟源可选，默认为 50MHz。
- 内部有 10bit 计数器，输出周期可配置。支持最高 25MHz (50MHz/2) 输出，最低约 3KHz (3MHz/1000) 输出。
- 内部有 10bit 计数器，输出高电平拍数可配置。
- 周期和占空比计数器为 10bits。通过配置 SVB PWM 相关的寄存器，可实现指定周期和占空比的 PWM 输出。

SVB_PWM 内部工作于 3MHz 或者 24MHz 或者 50MHz，SVB_PWM 输出配置流程如下：

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：

$\text{svb_pwn_period}=(50000000/\text{Freq})-1$ (50000000 为工作时钟，此时钟可选范围请参考 PERI_CRG4448)

高电平周期数计算公式为：

$\text{svb_pwn_duty}=(50000000/\text{Freq})*\text{duty}-1$ (50000000 为工作时钟，此时钟可选范围请参考 PERI_CRG4448)



步骤 2 配置 PWM 循环周期数 (svb_pwm_period) 和高电平周期数寄存器 (svb_pwm_duty) 。

步骤 3 配置 SVB_PWM(n)_CTRL[2]为 1, 加载配置的参数, 使配置的参数生效。(n=0,1,2)

步骤 4 配置 SVB_PWM(n)_CTRL[0]为 1, 使能 PWM 输出。(n=0,1,2)

----结束

例如: SVB_PWM0 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 (周期数) \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 PERI_CRG4448[13:12]写 0x0, 选择 SVB_PWM 的时钟源为 3MHz, 再向 PERI_CRG4448[4]写 0x1, 打开 SVB_PWM 时钟。

步骤 2 向 SVB_PWM0_CTRL[13:4](svb_pwm0_period)写入 0x0000_03E7。

步骤 3 向 SVB_PWM0_CTRL[25:16](svb_pwm0_duty)写入 0x0000_02D4。

步骤 4 向 SVB_PWM0_CTRL[2](svb_pwm0_load)写入 0x0000_0001。

步骤 5 向 SVB_PWM0_CTRL[0](svb_pwm0_enable)写入 0x0000_0001。

----结束

3.11.7 SVB_PWM 寄存器概览

SVB_PWM 寄存器概览如表 3-16 所示。

表3-16 SVB_PWM 寄存器概览 (基址是 0x0_1102_9000)

偏移地址	名称	描述	页码
0x0000	SVB_PWM0_CTRL	SVB_PWM0 控制寄存器	3-255
0x0004	SVB_PWM1_CTRL	SVB_PWM1 控制寄存器	3-255
0x0008	SVB_PWM2_CTRL	SVB_PWM2 控制寄存器	3-256



3.11.8 SVB_PWM 寄存器描述

SVB_PWM0_CTRL

SVB_PWM0_CTRL 为 SVB_PWM0 控制寄存器。

Offset Address: 0x0000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:16]	RW	svb_pwm0_duty	SVB_PWM0 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_pwm0_period	SVB_PWM0 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]	W1_PULSE	svb_pwm0_load	SVB_PWM0 输出参数加载控制，向此值写 1 使参数生效。	0x0
[1]	RW	svb_pwm0_inv	SVB_PWM0 输出正反相位控制。 0: 正相; 1: 反相。	0x0
[0]	RW	svb_pwm0_enable	SVB_PWM0 使能控制。 0: 不使能; 1: 使能。	0x0

SVB_PWM1_CTRL

SVB_PWM1_CTRL 为 SVB_PWM1 控制寄存器。

Offset Address: 0x0004 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:16]	RW	svb_pwm1_duty	SVB PWM1 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_pwm1_period	SVB PWM1 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]	W1_PULSE	svb_pwm1_load	SVB PWM1 输出参数加载控制，向此值写 1 使参数生效。	0x0
[1]	RW	svb_pwm1_inv	SVB PWM1 输出正反相位控制。 0: 正相; 1: 反相。	0x0
[0]	RW	svb_pwm1_enable	SVB PWM1 使能控制。 0: 不使能; 1: 使能。	0x0

SVB_PWM2_CTRL

SVB_PWM2_CTRL 为 SVB_PWM2 控制寄存器。

Offset Address: 0x0008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:16]	RW	svb_pwm2_duty	SVB PWM2 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_pwm2_period	SVB PWM2 输出的循环周期数，实际周期数为此数值加 1。	0x000



Bits	Access	Name	Description	Reset
[3]	-	reserved	保留。	0x0
[2]	W1_PULSE	svb_pwm2_load	SVB PWM2 输出参数加载控制，向此值写 1 使参数生效。	0x0
[1]	RW	svb_pwm2_inv	SVB PWM2 输出正反相位控制。 0: 正相; 1: 反相。	0x0
[0]	RW	svb_pwm2_enable	SVB PWM2 使能控制。 0: 不使能; 1: 使能。	0x0

3.11.9 Tsensor_CTRL 内部寄存器概览

Tsensor_CTRL 寄存器概览如表 3-17 所示。

表3-17 Tsensor_CTRL 寄存器概览 (基址是 0x0_1102_A000)

偏移地址	名称	描述	页码
0x0	TSENSOR_CTRL0	Tsensor 控制寄存器	3-258
0x4	TSENSOR_CTRL1	Tsensor 状态寄存器	3-258
0x8	TSENSOR_CTRL2	Tsensor 温度记录值寄存器 0	3-259
0xC	TSENSOR_CTRL3	Tsensor 温度记录值寄存器 1	3-259
0x10	TSENSOR_CTRL4	Tsensor 温度记录值寄存器 2	3-260
0x14	TSENSOR_CTRL5	Tsensor 温度记录值寄存器 3	3-260
0x20	TSENSOR_INT_MASK	Tsensor 中断屏蔽寄存器	3-260
0x24	TSENSOR_INT_CLR	Tsensor 中断清除寄存器	3-261
0x28	TSENSOR_INT_RAW	Tsensor 原始中断状态寄存器	3-261
0x2C	TSENSOR_INT_STAT	Tsensor 屏蔽后中断状态寄存器	3-262



3.11.10 Tsensor_CTRL 寄存器描述

TSENSOR_CTRL0

TSENSOR_CTRL0 为 Tsensor 控制寄存器。

Offset Address: 0x0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	tsensor_en	Tsensor 使能控制。 0: 关闭; 1: 使能。	0x0
[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

TSENSOR_CTRL1

TSENSOR_CTRL1 为 Tsensor 状态寄存器。

Offset Address: 0x4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[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;	0x0



Bits	Access	Name	Description	Reset
			10: 0.256ms; 11: 2.048ms。 注意：循环时间配置需要满足 $T_{\text{温循环时}} > 16 * T_{\text{转码时间}}$ 。	
[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

TSENSOR_CTRL2

TSENSOR_CTRL2 为 Tsensor 温度记录值寄存器 0。

Offset Address: 0x8 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

TSENSOR_CTRL3

TSENSOR_CTRL3 为 Tsensor 温度记录值寄存器 1。

Offset Address: 0xC 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



TSENSOR_CTRL4

TSENSOR_CTRL4 为 Tsensor 温度记录值寄存器 2。

Offset Address: 0x10 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

TSENSOR_CTRL5

TSENSOR_CTRL5 为 Tsensor 温度记录值寄存器 3。

Offset Address: 0x14 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

TSENSOR_INT_MASK

TSENSOR_CTRL5 为 Tsensor 中断屏蔽寄存器。

Offset Address: 0x20 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	tsensor0_low_war	温度下溢告警原始中断屏蔽控制。	0x0



Bits	Access	Name	Description	Reset
		ning_int_mask	0: 屏蔽中断; 1: 使能中断。	
[0]	RW	tsensor0_up_warning_int_mask	温度上溢告警原始中断屏蔽控制。 0: 屏蔽中断; 1: 使能中断。	0x0

TSENSOR_INT_CLR

TSENSOR_CTRL5 为 Tsensor 中断清除寄存器。

Offset Address: 0x24 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	WO	tsensor0_low_warning_int_clr	温度下溢告警中断清除, 写 1 生效。	0x0
[0]	WO	tsensor0_up_warning_int_clr	温度上溢告警中断清除, 写 1 生效。	0x0

TSENSOR_INT_RAW

TSENSOR_INT_RAW 为 Tsensor 原始中断状态寄存器。

Offset Address: 0x28 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RO	tsensor0_low_warning_int_raw	温度下溢告警原始中断。 0: 无中断; 1: 有中断。	0x0
[0]	RO	tsensor0_up_warning_int_raw	温度上溢告警原始中断。 0: 无中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	

TSENSOR_INT_STAT

TSENSOR_INT_STAT 为 Tsensor 屏蔽后中断状态寄存器。

Offset Address: 0x2C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RO	tsensor0_low_warning_int	屏蔽后温度下溢告警中断状态。 0: 无中断; 1: 有中断;	0x0
[0]	RO	tsensor0_up_warning_int	屏蔽后温度上溢告警中断状态。 0: 无中断; 1: 有中断;	0x0



目 录

4 DDR 控制器	4-1
4.1 概述	4-1
4.2 特点	4-1
4.3 功能描述	4-2
4.3.1 应用框图	4-2
4.3.2 功能原理	4-4
4.4 工作方式	4-7
4.4.1 软复位	4-7
4.4.2 DDR 初始化配置流程	4-8



插图目录

图 4-1 DDRC 单通道模式下, 与两片 16bit 位宽 DDR4 SDRAM 的互联示意图.....	4-2
图 4-2 DDRC 双通道模式下, 两个通道与一片 LPDDR4 SDRAM 的互联示意图.....	4-3
图 4-3 LPDDR4 SDRAM 双通道交织示意图.....	4-7



表格目录

表 4-1 DDR4 命令真值表.....	4-4
表 4-2 LPDDR4 命令真值表.....	4-5



4 DDR 控制器

4.1 概述

DDRC (DDR SDRAM Controller) 实现对动态存储器 DDR4/LPDDR4/LPDDR4X SDRAM 的存取控制。

说明

LPDDR4X 相关描述与 LPDDR4 一致。

4.2 特点

DDRC 的功能特点：

- 支持最大存储空间为 4GB。
- 仅支持小端存储模式。
- 支持 DDR4 SDRAM 接口最高工作频率为 1333MHz，即数据速率为 2.666Gbps，此时 DDRC 工作在 1:2 模式下，频率为 667MHz。
- 支持 LPDDR4 SDRAM 接口最高工作频率为 1333MHz，即数据速率为 2.666Gbps，此时 DDRC 工作在 1:2 模式下，频率为 667MHz。
- 支持 DDR4/LPDDR4 SDRAM 的 Power Down 低功耗模式。
- DDRC 最大支持 DDR4 单通道单 rank，LPDDR4 双通道双 rank，DDR SDRAM 数据总线位宽最大为 32bit。

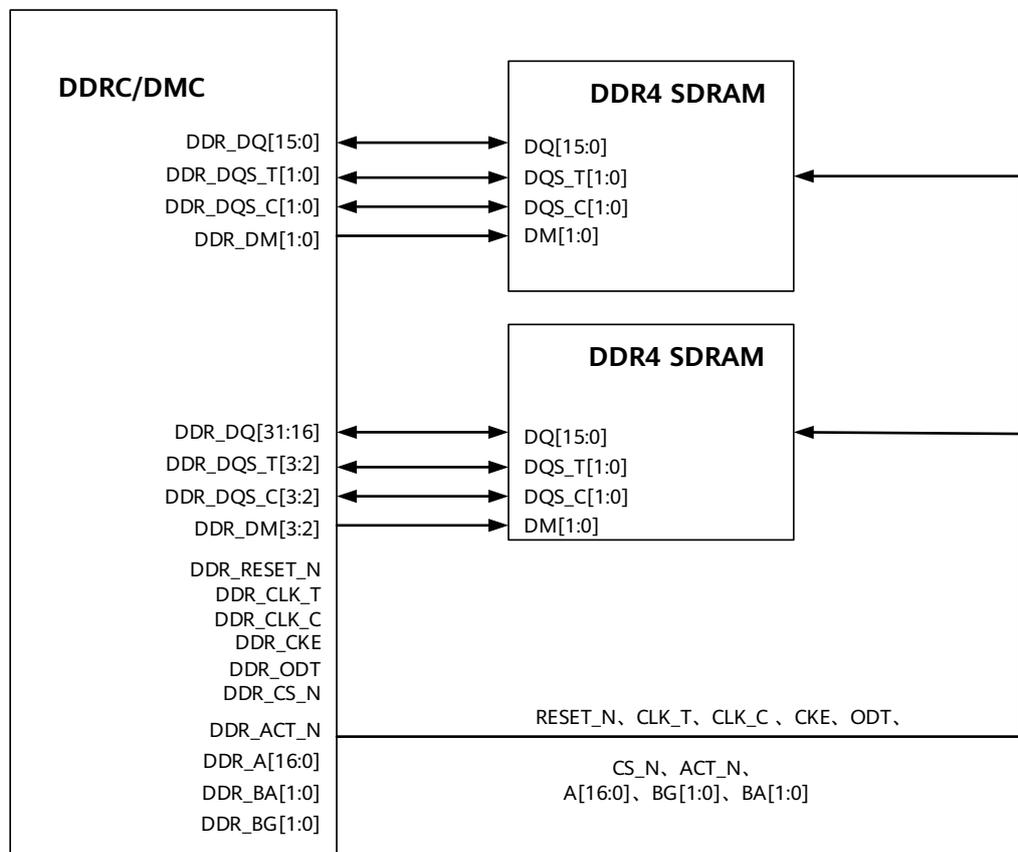
4.3 功能描述

4.3.1 应用框图

DDRC 实现了 SoC 系统中的 CPU 等主设备对 DDR4/LPDDR4 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 DDR4/LPDDR4 SDRAM 器件。

DDRC 在 DDR4 应用时为单通道 32bit，连接示意图如图 4-1 所示。

图4-1 DDRC 单通道模式下，与两片 16bit 位宽 DDR4 SDRAM 的互联示意图

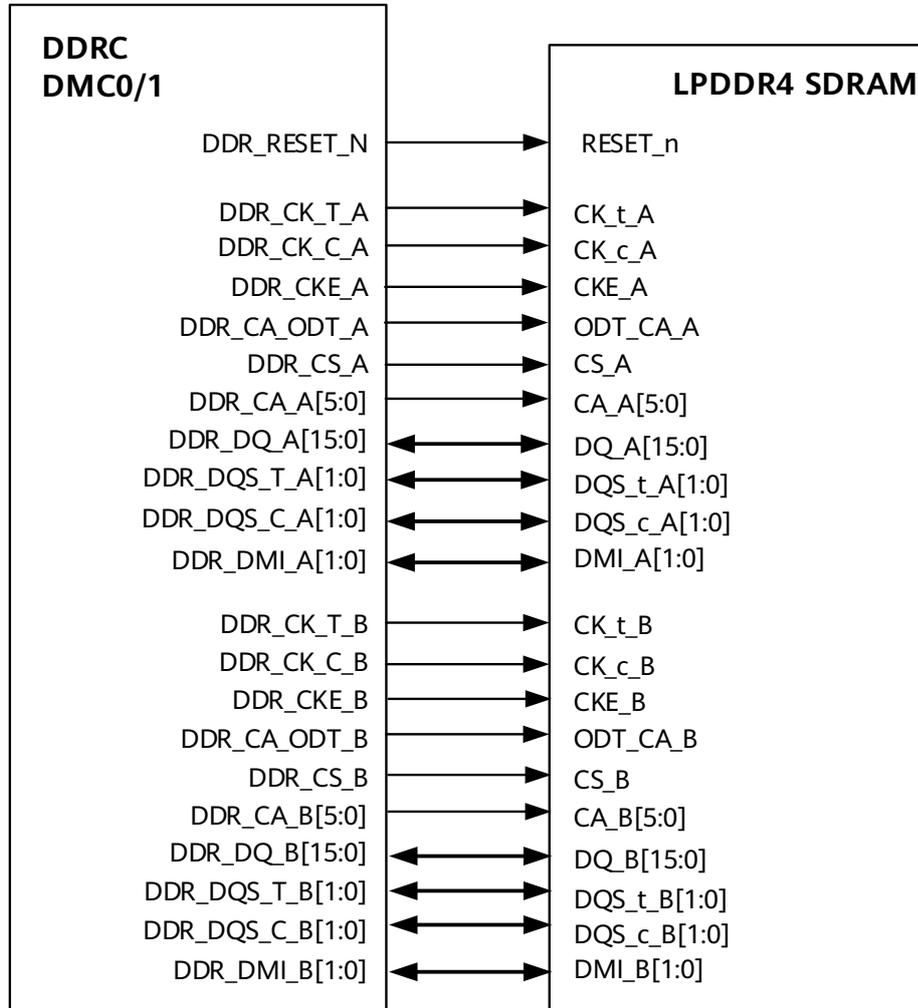


连接说明：

单通道模式下，32bit 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_CLK_T、DDR_CLK_C、DDR_CKE、DDR_RESET_N、DDR_ODT、DDR_ACT_N、DDR_BG[1:0]、DDR_A[16:0]、DDR_BA[1:0]，与 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。

对接 LPDDR4 时 DDRC 为双通道模式，每个通道支持 16bit 互联，每两个通道对接 1 片 LPDDR4。连接示意图如图 4-2 所示。

图4-2 DDRC 双通道模式下，两个通道与一片 LPDDR4 SDRAM 的互联示意图



连接说明:

通道 0 和通道 1 对接一片 LPDDR4 SDRAM。其中通道 0 与 LPDDR4 SDRAM 的 A 通道对接，通道 1 与 LPDDR4 SDRAM 的 B 通道对接。

注意：图 4-1 与图 4-2 中的管脚连接仅为示意图，与 DDR4/LPDDR4 颗粒的详细管脚连接关系请参见《Hi3519DV500 DDR SWAP 场景详细说明》。



4.3.2 功能原理

DDRC 接口时序满足 JEDEC 标准，通过发送 DDR4/LPDDR4 SDRAM 的命令字，完成对 DDR4/LPDDR4 SDRAM 的数据访问和状态控制。包括 DDR4/LPDDR4 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 DDR4/LPDDR4 SDRAM 的读写和控制命令字。

DDR4 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 DDR4 命令真值表

FUNCTION	CKE	CSN	ACTN	RASN/ ADR1 6	CASN/ ADR1 5	WEN/ ADR1 4	ADDR			BG	BA
							13:1 1	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
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

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



LPDDR4 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 LPDDR4 命令真值表

FUNCTION	CS	CA0	CA1	CA2	CA3	CA4	CA5	CK_t edge
DESELECT	L	X	X	X	X	X	X	R1
MPC	H	L	L	L	L	L	OP6	R1
	L	OP0	OP1	OP2	OP3	OP4	OP5	R2
PRECHARGE	H	L	L	L	L	H	AB	R1
	L	BA0	BA1	BA2	V	V	V	R2
REFRESH	H	L	L	L	H	L	AB	R1
	L	BA0	BA1	BA2	V	V	V	R2
SELF REFRESH ENTRY	H	L	L	L	H	H	V	R1
	L	V	V	V	V	V	V	R2
SELF REFRESH EXIT	H	L	L	H	L	H	V	R1
	L	V	V	V	V	V	V	R2
WRITE-1	H	L	L	H	L	L	BL	R1
	L	BA0	BA1	BA2	V	C9	AP	R2
MASK WRITE-1	H	L	L	H	H	L	L	R1
	L	BA0	BA1	BA2	V	C9	AP	R2
READ-1	H	L	H	L	L	L	BL	R1
	L	BA0	BA1	BA2	V	C9	AP	R2
CAS-2 (WRITE-2, MASK WRITE-2, READ-2, MRR2,MPC)	H	L	H	L	L	H	C8	R1
	L	C2	C3	C4	C5	C6	C7	R2
MODE REGISTER WRITE -1	H	L	H	H	L	L	OP7	R1
	L	MA0	MA1	MA2	MA3	MA4	MA5	R2
MODE REGISTER WRITE -2	H	L	H	H	L	H	OP6	R1
	L	OP0	OP1	OP2	OP3	OP4	OP5	R2



FUNCTION	CS	CA0	CA1	CA2	CA3	CA4	CA5	CK_t edge
MODE REGISTER READ-1	H	L	H	H	H	L	V	R1
	L	MA0	MA1	MA2	MA3	MA4	MA5	R2
ACTIVATE-1	H	H	L	R12	R13	R14	R15	R1
	L	BA0	BA1	BA2	R16	R10	R11	R2
ACTIVATE-2	H	H	H	R6	R7	R8	R9	R1
	L	R0	R1	R2	R3	R4	R5	R2

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

自动刷新

DDRC 自动产生周期性 AUTO REFRESH 命令，完成对 DDR4/LPDDR4 SDRAM 的刷新操作。常温下，DDR4 SDRAM 自动刷新操作的周期为 7.8us，LPDDR4 SDRAM 自动刷新操作的周期为 3.9us。

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。

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 DDR4/LPDDR4 SDRAM 进入到普通低功耗模式。

当系统需要进入到待机模式时，可控制 DDR4/LPDDR4 SDRAM 进入到自刷新低功耗模式。该模式下可以将 DDR4/LPDDR4 SDRAM 的功耗降至最低，同时保持 DDR4/LPDDR4 SDRAM 中的数据，但是此时系统不能访问 DDR4/LPDDR4 SDRAM。

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。

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。



详见软件文档《安全子系统使用说明》。

流量统计和命令 latency 统计功能

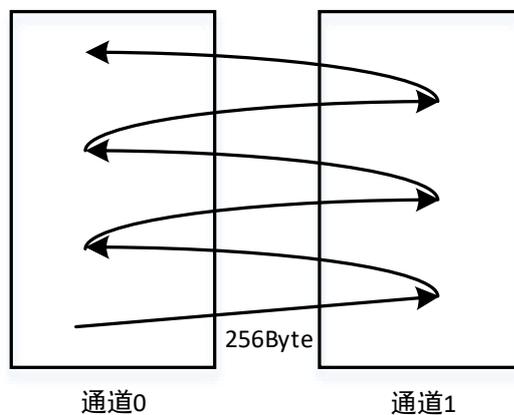
DDRC 支持流量统计功能：可以统计接口读写流量。可以对指定 ID 访问读/写流量单独统计。可以对总体读/写流量统计。支持 DDR 接口利用率统计，统计计数器支持连续统计和单次统计，当为连续统计时计数器为非饱和计数器，计数器记到最大值后会卷绕，方便系统进行连续统计，所以需要系统在计数器未卷绕之前将计数器值读出。当为单次统计时，统计计数器在进行统计时间到达后停止统计，系统可以用此功能可以统计瞬时的流量以及 latency。

通道交织

对接 DDR4 SDRAM 时仅使用 DDRC 单通道，无须关注通道交织。

对接 LPDDR4 SDRAM 时使用 DDRC 双通道，访问地址跨越通道交织粒度时会切换通道，通道交织粒度可配置，通常使用 256Byte 通道交织粒度。

图4-3 LPDDR4 SDRAM 双通道交织示意图



4.4 工作方式

4.4.1 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 DDR4/LPDDR4 SDRAM。



4.4.2 DDR 初始化配置流程

系统上电之后，必须先完成 DDR4/LPDDR4 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 DDR4/LPDDR4 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，DDR4/LPDDR4 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。



目 录

5 FMC	5-1
5.1 概述.....	5-1
5.2 特点.....	5-1
5.3 功能描述.....	5-2
5.3.1 接口描述.....	5-2
5.3.2 SPI 接口时序说明.....	5-7
5.3.3 SPI NAND FLASH 地址说明.....	5-7
5.3.4 Boot 功能.....	5-8
5.3.5 寄存器方式操作.....	5-9
5.3.6 内置 DMA 操作方式.....	5-9
5.3.7 ECC 校验功能.....	5-10
5.3.8 TIMEOUT 功能.....	5-10
5.4 工作流程.....	5-11
5.4.1 初始化流程.....	5-11
5.4.2 FMC_OP 操作流程（寄存器操作方式）.....	5-11
5.4.3 读器件状态寄存器操作.....	5-12
5.4.4 SPI NOR Flash 地址模式切换流程.....	5-12
5.4.5 擦除操作流程（SPI NAND Flash 和 SPI NOR Flash）.....	5-12
5.4.6 内置 DMA 读操作流程（FMC_OP_CTRL 读操作）.....	5-13
5.4.7 内置 DMA 写操作流程（FMC_OP_CTRL 写操作）.....	5-15
5.4.8 其它注意事项.....	5-16
5.5 数据结构(SPI NAND Flash).....	5-16
5.5.2 4bit ECC 模式(纠错 8bit/1KB).....	5-18
5.5.3 8bit ECC 模式(纠错 16bit/1KB).....	5-18



5.5.4 24bit ECC 模式(纠错 24bit/1KB)	5-19
5.6 ECC 模式选择说明	5-19
5.7 FMC 寄存器概览	5-20
5.8 FMC 寄存器描述	5-22



插图目录

图 5-1 Standard SPI 接口模式写操作时序图.....	5-3
图 5-2 Standard SPI 接口模式读操作时序图.....	5-3
图 5-3 Dual-Output/Dual-Input SPI 接口时序图.....	5-4
图 5-4 Dual I/O SPI 模式 STR 时序图.....	5-4
图 5-5 Dual I/O SPI 模式 DTR 时序图.....	5-5
图 5-6 Quad-Output/Quad-Input SPI 模式时序图.....	5-5
图 5-7 Quad-I/O SPI 模式接口时序图.....	5-6
图 5-8 Quad-I/O SPI 模式 DTR 接口时序图.....	5-6
图 5-9 SPI 输出时序图.....	5-7
图 5-10 擦除操作流程图中.....	5-13
图 5-11 4bit ECC 模式 2K page_size 的数据结构.....	5-18
图 5-12 4bit ECC 模式 4K page_size 的数据结构.....	5-18
图 5-13 8bit ECC 模式 2K page_size 的数据结构.....	5-18
图 5-14 8bit ECC 模式 4K page_size 的数据结构.....	5-19
图 5-15 24bit ECC 模式 2K page-size 的数据结构.....	5-19
图 5-16 24bit ECC 模式 4K page_size 的数据结构.....	5-19



表格目录

表 5-1 FLASH 地址分配表	5-7
表 5-2 非 ECC0 模式数据结构长度	5-17
表 5-3 FMC 寄存器概览 (基址是 0x0_1000_0000)	5-20



5 FMC

5.1 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。

5.2 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。
- 支持外接双片选 (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。
- 支持多种规格的 SPI NOR Flash 器件
 - 支持 SPI NOR DTR /STR 模式的读数据。
 - 支持器件最大容量 256MB。
- 支持多种规格的 SPI NAND Flash 器件。
 - 支持 SPI NAND DTR /STR 模式的读数据。
 - 支持页大小 2K、4K 的器件。
 - 支持块大小 64 Pages/Block、128 Pages/Block 的器件。
 - 支持器件最大容量 2GB。
- 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 FAST READ 命令，最高可提频至 99MHz 进行读操作。
- 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。

5.3 功能描述

5.3.1 接口描述

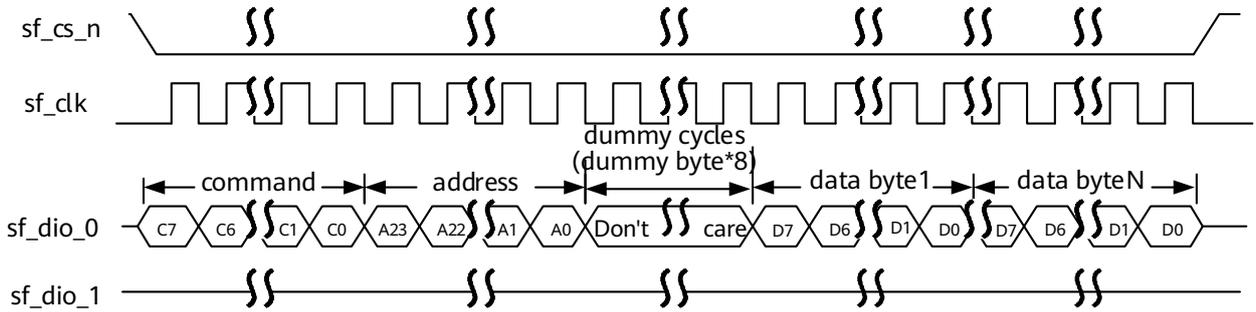
FMC 可以支持五种 SPI 接口类型，五种类型分别为 Standard SPI、Dual-Output/Dual-Input SPI、Quad-Output/Quad-Input SPI、Dual I/O SPI、Quad I/O SPI 类型，Standard SPI、Dual I/O SPI、Quad I/O SPI 类型支持 STR、DTR 模式。

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 5-1 为 Standard SPI 接口模式写操作时序图。



图5-1 Standard SPI 接口模式写操作时序图

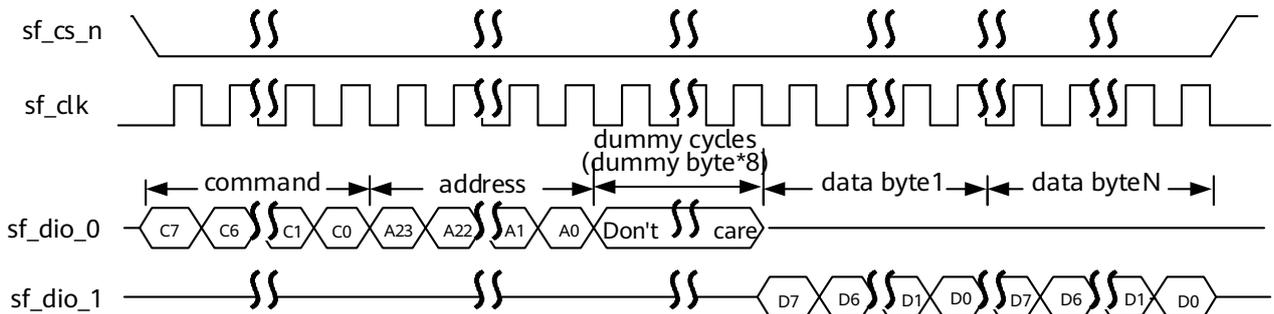


时序说明:

- 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 5-2 为 Standard SPI 接口模式读操作时序图。

图5-2 Standard SPI 接口模式读操作时序图



时序说明:

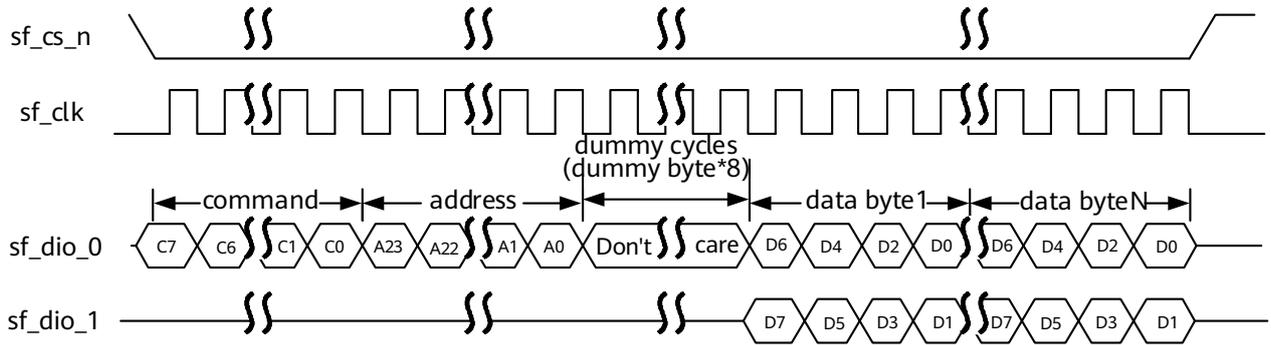
- command/address/dummy cycles 以单 bit 串行方式在 `sf_dio_0` 线上输出。
- Data 以单 bit 串行方式在 `sf_dio_1` 线上输入。

Dual-Output /Dual-Input SPI 模式

Dual-Output /Dual-Input SPI 接口时序图 5-3 所示。



图5-3 Dual-Output/Dual-Input SPI 接口时序图



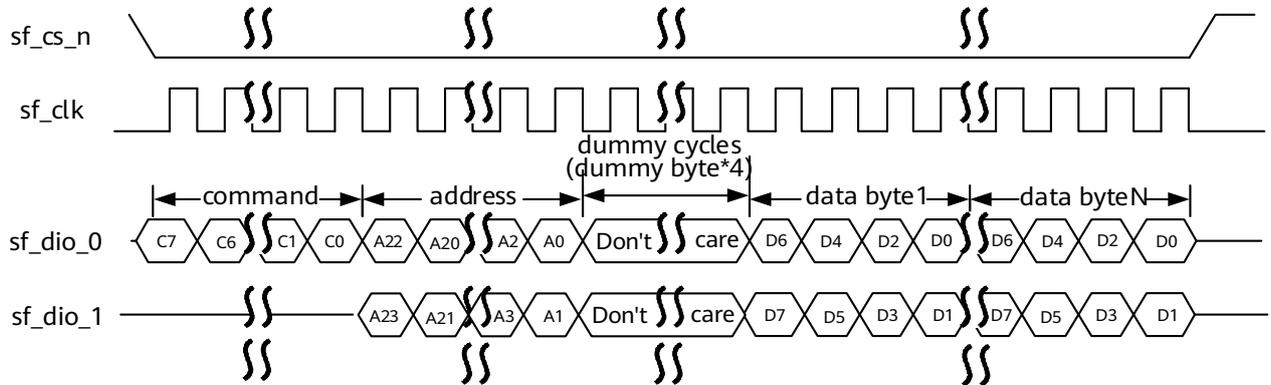
时序说明:

- 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 时序图为图 5-4 和图 5-5 所示。

图5-4 Dual I/O SPI 模式 STR 时序图

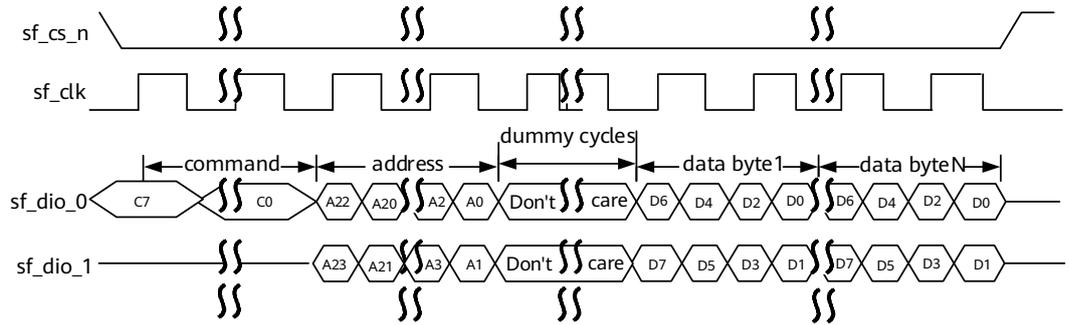


时序说明:

- 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5-5 Dual I/O SPI 模式 DTR 时序图



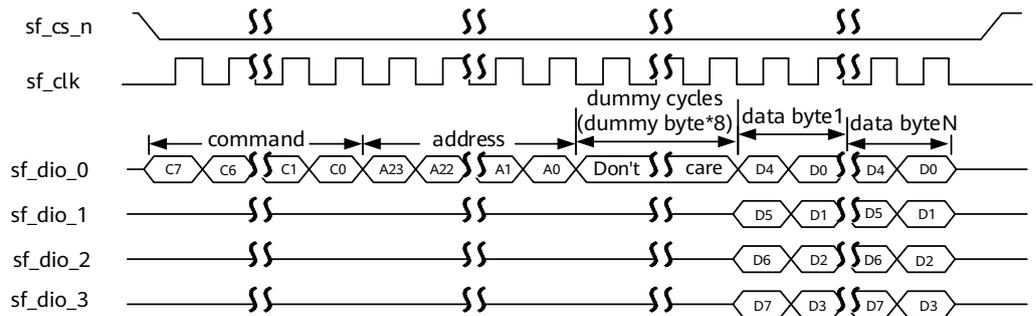
时序说明:

- 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 5-6 所示。

图5-6 Quad-Output/Quad-Input SPI 模式时序图



时序说明:

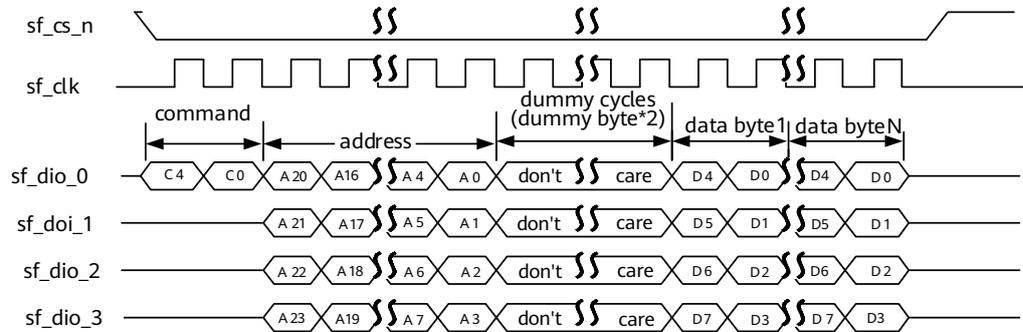
- command/address/dummy cycles 以单 Bit 串行方式在 sf_dio_0 线上输出。
- Data 以 4 Bits 方式在 sf_dio_0/sf_dio_1/sf_dio_2/sf_dio_3 线上输出 (写) 或输入 (读)。



Quad I/O SPI 模式

Quad I/O SPI 模式时序图如图 5-7 和图 5-8 所示。

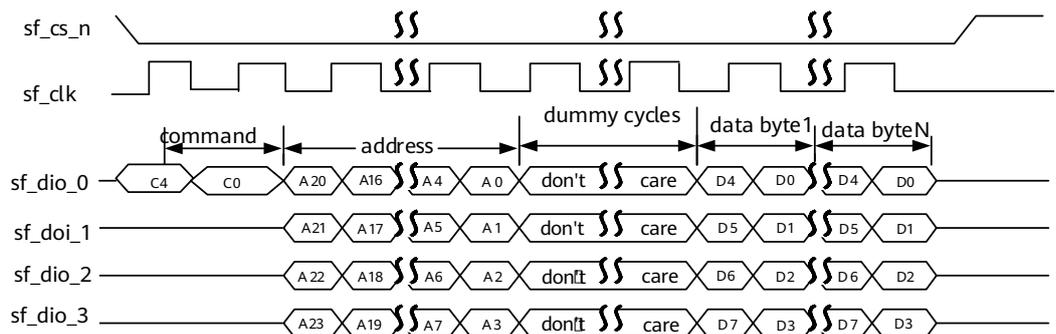
图5-7 Quad-I/O SPI 模式接口时序图



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5-8 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。

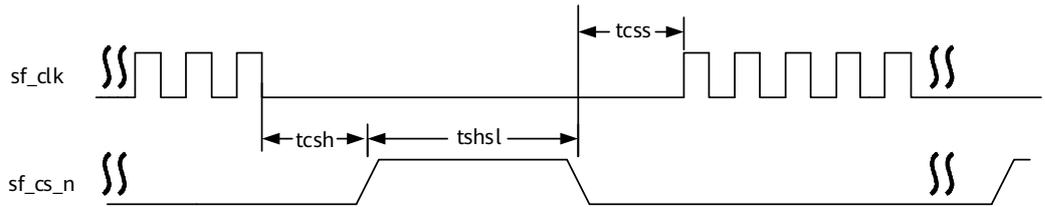


- Data 以 4 Bits 双沿采数方式在 sf_dio_0/sf_dio_1/sf_dio_2/sf_dio_3 线上输入。

5.3.2 SPI 接口时序说明

SPI 接口时序以及参数说明如图 5-9 所示。

图5-9 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。

5.3.3 SPI NAND FLASH 地址说明

对于 SPI NAND Flash 器件地址，如表 5-1。其中 1st Byte 和 2nd Byte 为列地址，3rd Byte，4th Byte 和 5th Byte 为行地址。

表5-1 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。

5.3.4 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为 0xff。
- 当前物理块的第一个页和最后一个页空块标记为 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。

5.3.5 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。

5.3.6 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。

5.3.7 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 3 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。

5.3.8 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。

5.4 工作流程

5.4.1 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“5.4.4 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

5.4.2 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

5.4.3 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

5.4.4 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]为 4B 模式，完成 3Byte 模式到 4Byte 模式切换。

----结束

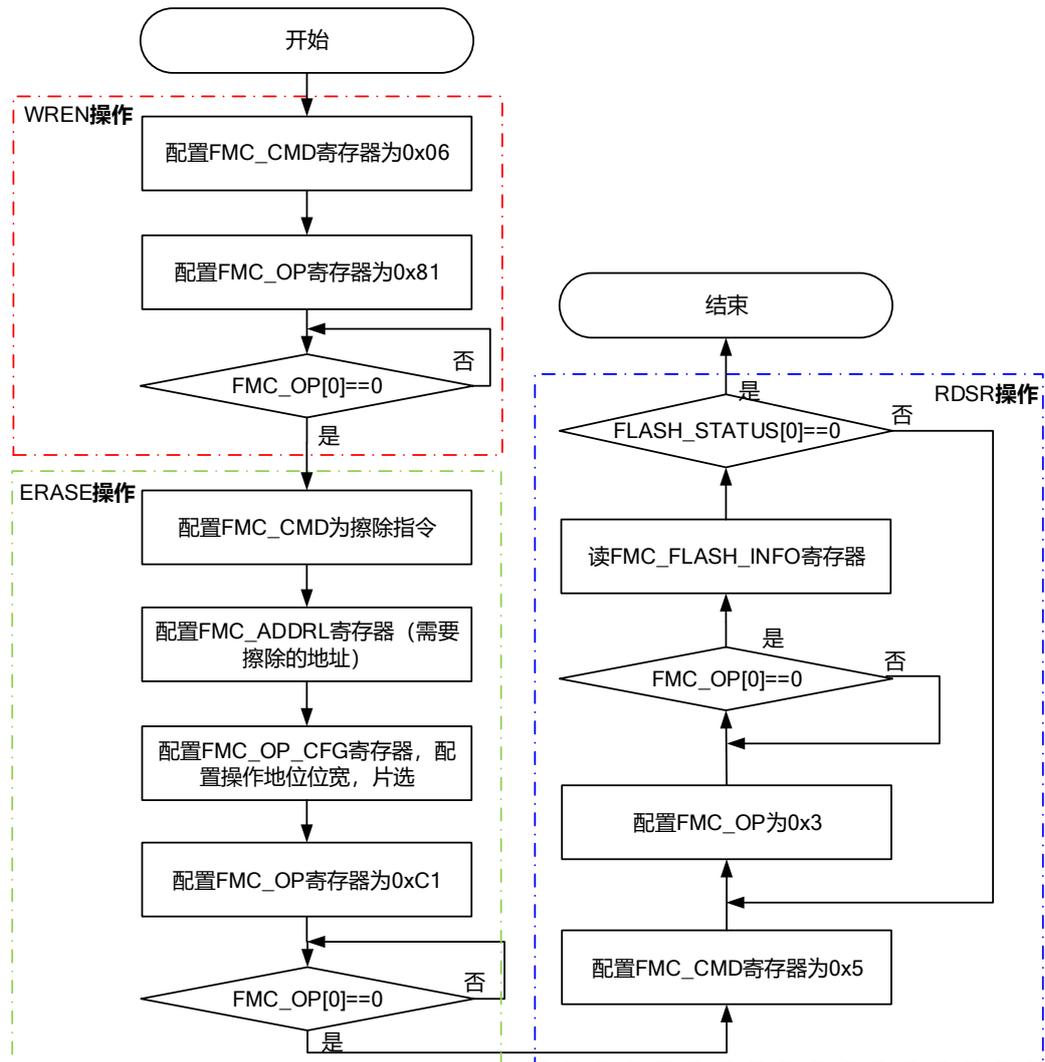
说明

具体 SPI NOR Flash 地址模式切换的命令，请查找相关器件手册。

5.4.5 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 5-10 所示。

图5-10 擦除操作流程



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。

5.4.6 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。



须知

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_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。

须知

- 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。

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

5.4.7 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。

须知

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。

须知

- 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。

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

5.4.8 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。

5.5 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 5-2。

表5-2 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

说明

- 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。

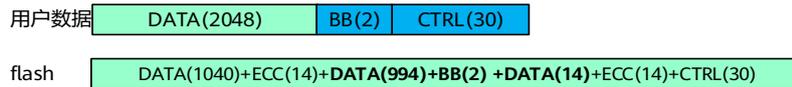


5.5.2 4bit ECC 模式(纠错 8bit/1KB)

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 5-11 所示。

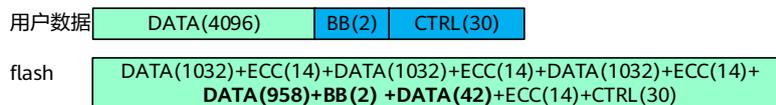
图5-11 4bit ECC 模式 2K page_size 的数据结构



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 5-12 所示。

图5-12 4bit ECC 模式 4K page_size 的数据结构

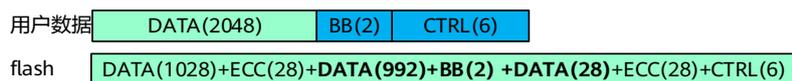


5.5.3 8bit ECC 模式(纠错 16bit/1KB)

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 5-13 所示。

图5-13 8bit ECC 模式 2K page_size 的数据结构

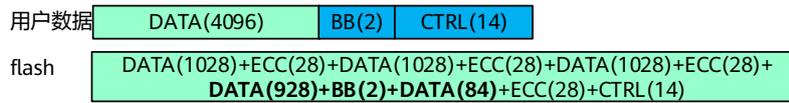


4KB page_size

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 5-14 所示。



图5-14 8bit ECC 模式 4K page_size 的数据结构

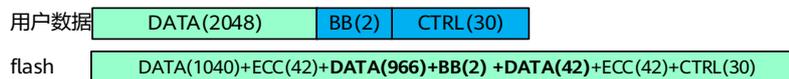


5.5.4 24bit ECC 模式(纠错 24bit/1KB)

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 5-15 所示。

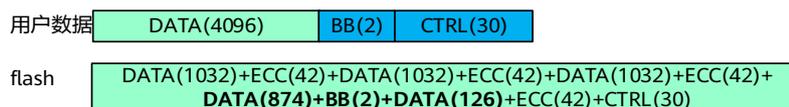
图5-15 24bit ECC 模式 2K page-size 的数据结构



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 5-16 所示。

图5-16 24bit ECC 模式 4K page_size 的数据结构



5.6 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 >= 控制器 ECC 模式所需的存储大小

比如：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“5.5 数据结构(SPI NAND Flash)”这一章节。

5.7 FMC 寄存器概览

FMC 寄存器概览如表 5-3 所示。

表5-3 FMC 寄存器概览 (基址是 0x0_1000_0000)

偏移地址	名称	描述	页码
0x0000	FMC_CFG	器件配置寄存器	5-22
0x0004	GLOBAL_CFG	全局配置寄存器	5-23
0x0008	TIMING_SPI_CFG	SPI Timing 配置寄存器	5-25
0x0018	FMC_INT	中断状态寄存器	5-25
0x001C	FMC_INT_EN	中断使能寄存器	5-27
0x0020	FMC_INT_CLR	中断清除寄存器	5-28
0x0024	FMC_CMD	命令字配置寄存器	5-28
0x0028	FMC_ADDRH	器件地址配置寄存器高 byte	5-29
0x002C	FMC_ADDRL	器件地址配置寄存器低 4byte	5-29
0x0030	FMC_OP_CFG	操作配置寄存器	5-29
0x0034	SPI_OP_ADDR	操作地址配置寄存器	5-30
0x0038	FMC_DATA_NUM	数据长度寄存器	5-31
0x003C	FMC_OP	操作寄存器	5-31
0x0040	FMC_DMA_LEN	DMA 操作长度寄存器	5-33
0x0048	FMC_DMA_AHB_CTR	DMA AHB 总线操作控制寄存器	5-33



偏移地址	名称	描述	页码
	L		
0x004C	FMC_DMA_SADDR_D0	DMA 操作 DDR 起始地址寄存器	5-34
0x005C	FMC_DMA_SADDR_OOB	DMA 操作 DDR 存放 OOB 信息起始地址寄存器	5-34
0x0068	FMC_OP_CTRL	操作控制寄存器	5-34
0x006C	FMC_TIMEOUT_WR	写操作 TIMEOUT 时间寄存器	5-35
0x0070	FMC_OP_PARA	OP 操作参数选择寄存器	5-35
0x0074	FMC_BOOT_SET	Boot 设定寄存器	5-36
0x0078	FMC_LP_CTRL	低功耗控制寄存器	5-37
0x00A8	FMC_ERR_THD	ECC 告警阈值寄存器	5-37
0x00AC	FMC_FLASH_INFO	器件状态寄存器值	5-38
0x00BC	FMC_VERSION	版本寄存器	5-38
0x00C0	FMC_ERR_NUM0_BUFFER0	SPI NAND Flash 第一次 buffer 纠错信息 0 统计寄存器	5-38
0x00D0	FMC_ERR_ALARM_ADDRH	ECC 告警 Flash 地址高 byte 寄存器	5-39
0x00D4	FMC_ERR_ALARM_ADDRL	ECC 告警 Flash 地址低 byte 寄存器	5-39
0x00D8	FMC_ECC_INVALID_ADDRH	ECC 不可纠错地址高 byte 寄存器	5-39
0x00DC	FMC_ECC_INVALID_ADDRL	ECC 不可纠错地址低 4byte 寄存器	5-40
0x0037C	FMC_BOOT_CMD	BOOT 启动读操作命令配置寄存器	5-40



5.8 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); 1: 4Bytes address mode。 复位值由管脚决定。	0x0
[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;	0x0



Bits	Access	Name	Description	Reset
			01: 4KB page_size; 其他: 保留。	
[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: 0x0001_00C4

Bits	Access	Name	Description	Reset
[31:21]	-	reserved	保留。	0x0000
[20]	RW	ddr_mode_byp	SPI NAND DDR 功能 bypass, 为了保证之前所有功能的正确性。 1: 使能 bypass 功能; 0: 不使能 bypass 功能。 注: 该位域只针对 SPI NAND DDR 模式。	0x0
[19:16]	-	reserved	保留。	0x1
[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;	0x0



Bits	Access	Name	Description	Reset
			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。	
[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; 111: 读数据延迟 3.5 个器件工作时钟周期。	0x0



Bits	Access	Name	Description	Reset
[2]	-	reserved	保留。	0x1
[1:0]	-	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



Bits	Access	Name	Description	Reset
[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
[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



Bits	Access	Name	Description	Reset
[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。	0x0



Bits	Access	Name	Description	Reset
			0: 禁止; 1: 使能。	
[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
[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
[12:11]	RW	fm_cs	选择操作的 Flash 器件。 00: cs0; 01: cs1。 其他: 保留。	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;	0x0



Bits	Access	Name	Description	Reset
			1: 使能。	
[6]	RW	addr_en	向 FLASH 写操作地址使能。 0: 禁止; 1: 使能。	0x0
[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 flash_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。

Offset Address: 0x004C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	dma_mem_saddr_d0	FMC 的 DMA 操作访问 DDR 的起始地址。	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;	0x0



Bits	Access	Name	Description	Reset
			01: 只读 OOB 操作; 其他: 保留。	
[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。	0x0000_0000



Bits	Access	Name	Description	Reset
			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。	

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_a ddrh	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_a ddrl	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

FMC_BOOT_CMD

FMC_BOOT_CMD 为 BOOT 启动读操作命令配置寄存器

Offset Address: 0x0037C Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	rd_cmd	BOOT 启动读操作命令。 0x03: 读取(默认); 0x0B: 快速读取, 最高可提频至 99MHz, 需同步配置 dummy_num; 其他: 保留。	0x03



目 录

6 GMAC	6-1
6.1 概述	6-1
6.2 功能描述	6-1
6.3 总体数据流	6-2
6.4 单网口功能配置描述	6-2
6.4.1 以太网收发帧管理功能	6-2
6.4.2 以太网收包中断管理功能	6-3
6.4.3 配置 PHY 芯片工作状态	6-3
6.4.4 工作模式切换	6-4
6.5 典型应用	6-4
6.5.1 以太网限速功能	6-4
6.5.2 寄存器偏移地址说明	6-5
6.5.3 GMAC 寄存器概览	6-5
6.5.4 GMAC 寄存器描述	6-8



插图目录

图 6-1 GMAC 总体数据流.....	6-2
-----------------------	-----



表格目录

表 6-1 GMAC 寄存器概览 (基址是 0x0_1029_0000)	6-6
--	-----



6 GMAC

6.1 概述

千兆以太网模块实现网络接口数据的接收和发送，支持 10/100/1000Mbit/s 工作模式可配置，支持全双工、半双工工作模式。可实现和 CPU 端口间的数据通信。

芯片支持 1 个 GMAC 模块。

6.2 功能描述

千兆以太网模块有如下功能特点：

- 支持 10/100/1000Mbit/s 速率。
- 支持全双工或半双工工作模式。
- 支持 RMII/RGMII (Reduced Gigabit Media Independent Interface) 接口。
- 提供 MDIO 接口。
- 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。
- 支持对接收和发送帧进行统计计数。
- 支持广播帧、组播帧和单播帧过滤。
- 支持 IP 报文、广播或多播报文限速处理功能可配置。
- 支持非法包过滤功能。

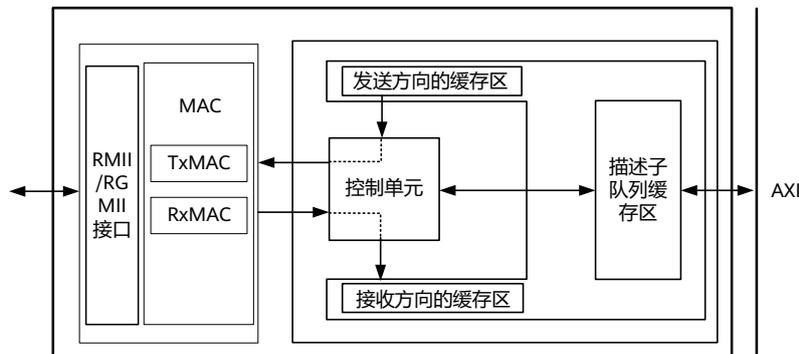


- 支持入队中断和超时中断两种中断方式。
- 支持收发包缓存。
- 支持 TSO 功能 (TCP segmentation offload, TCP 分段卸载)。
- 支持 UFO 功能 (UDP fragmentation offload, UDP 分片卸载)。
- 支持多核多中断 RSS 功能。

6.3 总体数据流

GMAC 千兆以太网交换接口的总体数据流如图 6-1 所示。

图6-1 GMAC 总体数据流



6.4 单网口功能配置描述

6.4.1 以太网收发帧管理功能

CPU 通过配置描述子队列缓存区，对 Ethernet 进行收发帧地址管理：

- 接收时，Ethernet 分辨从外网收到的各种数据包，并根据 CPU 配置的报文缓存信息，包括报文缓存起始地址、报文缓存深度等，将收到的合法包通过总线存放于 DDR 中。
- 发送时，Ethernet 根据 CPU 配置的报文缓存信息，包括报文缓存起始地址、报文长度以及其他的报文信息等，通过总线，将存于 DDR 的报文搬运过来，自行组装成包，然后发送到网络接口上。



6.4.2 以太网收包中断管理功能

中断产生

配置接收方向入队中断使能以及上报入队中断的水线，当逻辑回写到 DDR 中的描述子个数达到配置水线时，则产生一个接收入队中断：

步骤 1 配置 ENA_PMU_INT bit[17] rx_bq 入队中断使能；

步骤 2 配置 ENA_PMU_INT bit[28] rx_bq 入队超时中断使能；

----**结束**

中断清除

CPU 收到接收入队中断或者接收超时中断后，写 1 清除接收入队中断和接收超时中断。

6.4.3 配置 PHY 芯片工作状态

以太网交换模块提供 MDIO 接口实现对 PHY 芯片的管理。MDIO 分为读操作和写操作，下面分别介绍两者的操作步骤。

读操作的配置步骤如下：

步骤 1 CPU 向 MDIO_SINGLE_CMD bit[12:8]写入 PHY 芯片地址，向 bit[4:0]写入 PHY 内部寄存器地址，同时配置寄存器 MDIO_SINGLE_CMD bit[20]为 1，配置 bit[17:16]为 2'b10，启动 MDIO 读操作；

步骤 2 MDIO 将从外部 PHY 芯片读回的数据写到 MDIO_SINGLE_DATA bit[31:16]，并将 MDIO_SINGLE_CMD bit[20]配置为 0；

步骤 3 CPU 查询 MDIO_SINGLE_DATA bit[31:16]得到 MDIO 从外部 PHY 芯片读回的数据。

----**结束**

写操作的配置步骤如下：

步骤 1 CPU 将发送给外部 PHY 芯片的数据写入 MDIO_SINGLE_DATA bit[15:0]；

步骤 2 向 MDIO_SINGLE_CMD bit[9:8]写入 PHY 芯片地址，向 bit[4:0]写入 PHY 内部寄存器地址，同时配置 bit[20]为 1，配置 bit[17:16]为 1，启动 MDIO 写操作；



步骤 3 MDIO 将 MDIO_SINGLE_DATA bit[15:0]的值写入相应的 PHY 内部寄存器中，完成 MDIO 的写操作，并自动将 MDIO_SINGLE_CMD bit[20]配置为 0x0。

----结束

6.4.4 工作模式切换

千兆以太网的工作模式：

- RMII (10M/100M)
- RGMII (10M/100M/1000M)

上述模式根据单板硬件连接确认。

下面介绍速率双工切换步骤。

步骤 1 配置 MAC_IF_STAT_CTRL bit[8:0](0 网口 MAC_IF 状态控制寄存器)；

步骤 2 配置 PERI_CRG3568/PERI_CRG3569 bit[0] (GMAC CRG 控制寄存器) 为 1 进行复位，然后再配置为 0 撤销复位，使配置的工作模式生效；

步骤 3 配置 PORT_MODE (MAC 端口速率模式寄存器)。

📖 说明

芯片正常工作时不可进行此项配置，建议在初始化时进行配置。

----结束

6.5 典型应用

6.5.1 以太网限速功能

以太网交换模块具有对接收报文进行限速的功能，即在某段时间内，当通过的报文数超过设定的最大数量时，后续报文将被丢弃。

以太网交换模块对报文限速分为以下两种：

- 对 IP 报文的限速
- 对广播或多播报文的限速



对 IP 报文的限速

IP 报文的限速以 $1\mu\text{s}$ 为单位进行计数，在限速时间内，当通过的报文达到限速个数时，后续报文被丢弃。

对 IP 报文进行限速时，需要做如下配置：

步骤 1 配置 CONTROL_WORD bit[21]为 1。

步骤 2 配置 FLOW_CTRL_PKG_THRSLD。

步骤 3 配置 CRF_FLOW_TIME_THRSLD。

----结束

对广播或多播报文的限速

广播或多播报文的限速时间以 $1\mu\text{s}$ 为单位进行计数，在限速时间内，当通过的报文达到限速个数时，后续报文被丢弃。

对广播或者多播报文进行限速时，需要做如下配置：

步骤 1 配置 CONTROL_WORD bit[16]为 1。

步骤 2 配置 CRF_BM_PKT_THRSLD。

步骤 3 配置 CRF_BM_PKT_THRSLD。

----结束

6.5.2 寄存器偏移地址说明

GMAC 偏移地址共占用 16bit 地址空间，其中：

- 0x0000~0x0fff：用于 GMAC 以太网口；
- 0x300c：MAC_IF 状态控制寄存器。

6.5.3 GMAC 寄存器概览

GMAC 寄存器概览如表 6-1 所示。



表6-1 GMAC 寄存器概览 (基址是 0x0_1029_0000)

偏移地址	名称	描述	页码
0x0000	STATION_ADDR_LOW	本机 MAC 地址寄存器	6-8
0x0004	STATION_ADDR_HIGH	本机 MAC 地址寄存器	6-8
0x0008	DUPLEX_SEL_RGMI	半双工选择寄存器	6-9
0x000C	FD_FC_TYPE	流控帧类型域寄存器	6-9
0x0014	COL_DISTANCE	单次重传包长水线寄存器	6-9
0x0038	PAUSE_THR	发送流控帧间隙寄存器	6-9
0x003C	MAX_FRM_SIZE	最大帧长寄存器	6-10
0x0040	PORT_MODE	端口状态寄存器	6-10
0x0044	PORT_EN	通道使能寄存器	6-11
0x0048	PAUSE_EN	流控使能寄存器	6-11
0x0050	SHORT_RUNTS_THR	超短帧界限寄存器	6-12
0x0054	DROP_UNK_CTL_FRM	未知控制帧丢弃使能寄存器	6-12
0x0060	TRANSMIT_CONTROL	常用配置寄存器	6-12
0x0064	REC_FILT_CONTROL	接收帧过滤控制寄存器	6-13
0x0080	RX_OCTETS_OK_COUNT	接收有效帧的字节统计寄存器	6-14
0x0084	RX_OCTETS_BAD_COUNT	接收错误帧字节统计寄存器	6-14
0x0088	RX_UC_PKTS	MAC 接收单播帧数统计寄存器	6-14
0x008C	RX_MC_PKTS	接收多播帧数统计寄存器	6-15
0x0090	RX_BC_PKTS	接收的广播帧数统计寄存器	6-15



偏移地址	名称	描述	页码
0x00B0	RX_FCS_ERRORS	接收 CRC 检验错误的帧数统计寄存器	6-15
0x0100	OCTETS_TRANSMITTED_OK	发送成功的好包字节数统计寄存器	6-15
0x0104	OCTETS_TRANSMITTED_BAD	发送成功的错包字节数统计寄存器	6-16
0x0108	TX_UC_PKTS	发送的单播帧数统计寄存器	6-16
0x010C	TX_MC_PKTS	发送的多播帧数统计寄存器	6-16
0x0110	TX_BC_PKTS	发送的广播帧数统计寄存器	6-16
0x0158	TX_CRC_ERROR	发送帧长正确 CRC 错误的帧数统计寄存器	6-17
0x01B0	CF_CRC_STRIP	CRC 剥离使能寄存器	6-17
0x01B4	MODE_CHANGE_ENABLE	端口模式改变使能寄存器	6-17
0x01C0	COL_SLOT_TIME	半双工冲突重传间隔时间计数器寄存器	6-18
0x01E0	RECV_CONTROL	接收控制寄存器	6-18
0x01E8	VLAN_CODE	VLAN Code 寄存器	6-19
0x020C	CRF_MAX_PACKET	最大过滤包长寄存器	6-19
0x0210	CRF_MIN_PACKET	最小过滤包长寄存器	6-19
0x0214	CONTROL_WORD	控制寄存器	6-20
0x0218	FLOW_CTRL_PKG_THRSLD	限速包个数寄存器	6-21
0x021C	CRF_FLOW_TIME_THRSLD	限速时间寄存器	6-21
0x0358	CRF_BM_PKT_THRSLD	广播和多播报文的限速处理包个数寄存器	6-22
0x02E8	TSO_COE_CTRL	TSO 开关和 COE 过滤开关寄存器	6-22
0x035C	CRF_BM_TIME_THRSLD	广播和多播报文的限速时间计数器寄存器	6-23



偏移地址	名称	描述	页码
0x03C0	MDIO_SINGLE_CM D	MDIO 单次操作寄存器	6-23
0x03C4	MDIO_SINGLE_DAT A	MDIO 读写数据寄存器	6-24
0x05C4	ENA_PMU_INT	PMU 模块原始中断使能寄存器	6-24
0x05C8	STATUS_PMU_INT	PMU 模块中断状态寄存器	6-29
0x05CC	DESC_WR_RD_ENA	CFF 读写描述子使能寄存器	6-33
0x300C	MAC_IF_STAT_CTR L	MAC_IF 模式控制寄存器	6-34

6.5.4 GMAC 寄存器描述

STATION_ADDR_LOW

STATION_ADDR_LOW 为本机 MAC 地址寄存器。

Offset Address: 0x0000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	station_addr_low	MAC_CORE 的 MAC 源地址低 32bit。	0x00000000

STATION_ADDR_HIGH

STATION_ADDR_HIGH 为本机 MAC 地址寄存器。

Offset Address: 0x0004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	station_addr_high	MAC_CORE 的 MAC 源地址高 16bit, 可默认。	0x0000



DUPLEX_SEL_RGMII

DUPLEX_SEL_RGMII 为半双工选择寄存器。

Offset Address: 0x0008 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	duplex_sel_rgmii	半双工选择信号。 0: 半双工; 1: 全双工。	0x1

FD_FC_TYPE

FD_FC_TYPE 为流控帧类型域寄存器。

Offset Address: 0x000C Total Reset Value: 0x0000_8808

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	fd_fc_type	全双工模式下流控帧 TYPE 域。	0x8808

COL_DISTANCE

COL_DISTANCE 为单次重传包长流水线寄存器。

Offset Address: 0x0014 Total Reset Value: 0x0000_0043

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x000000
[9:0]	RW	col_distance	单次重传包长流水线。	0x043

PAUSE_THR

PAUSE_THR 为发送流控帧间隙寄存器。



Offset Address: 0x0038 Total Reset Value: 0x0000_002F

Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	保留。	0x0000
[15:0]	RW	pause_thr	流控帧间隔时间，若流控时间大于间隔时间，MAC 自动发送流控帧。以 512bit 为时间单位。	0x002F

MAX_FRM_SIZE

MAX_FRM_SIZE 为最大帧长寄存器。

Offset Address: 0x003C Total Reset Value: 0x0000_05EE

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13:0]	RW	max_frm_size	MAC 部分允许的最大帧长。 当接收到大于该帧长的帧时，将该帧认为超长错误帧；当发送帧超过该帧长的帧时，将发送帧截断后做为错帧发送。	0x05EE

PORT_MODE

PORT_MODE 为端口状态寄存器。

Offset Address: 0x0040 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	port_mode	指示当前 MAC 端口工作的模式。 000: 10Mbps; 001: 100Mbps; 101: 1000Mbps;	0x1



Bits	Access	Name	Description	Reset
			其他：保留。	

PORT_EN

PORT_EN 为通道使能寄存器。

Offset Address: 0x0044 Total Reset Value: 0x0000_0006

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2]	RW	tx_en	发送通道使能位。 0：不使能； 1：使能。	0x1
[1]	RW	rx_en	接收通道使能位。 0：不使能； 1：使能。	0x1
[0]	-	reserved	保留。	0x0

PAUSE_EN

PAUSE_EN 为流控使能寄存器。

Offset Address: 0x0048 Total Reset Value: 0x0000_0007

Bits	Access	Name	Description	Reset
[31:2]	RO	reserved	保留。	0x00000001
[1]	RW	tx_fdfc	全双工模式下发送流控帧使能。 0：不使能； 1：使能。	0x1
[0]	RW	rx_fdfc	全双工模式下响应流控帧使能。 0：不使能；	0x1



Bits	Access	Name	Description	Reset
			1: 使能。	

SHORT_RUNTS_THR

SHORT_RUNTS_THR 为超短帧界限寄存器。

Offset Address: 0x0050 Total Reset Value: 0x0000_000C

Bits	Access	Name	Description	Reset
[31:5]	RO	reserved	保留。	0x00000000
[4:0]	RW	short_runs_thr	短帧、超短帧界限(只用于统计短帧、超短帧计数)。	0x0C

DROP_UNK_CTL_FRM

DROP_UNK_CTL_FRM 为未知控制帧丢弃使能寄存器。

Offset Address: 0x0054 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留。	0x00000000
[0]	RW	drop_unk_ctl_frm	未知控制帧处理位。 0: 正常转发未知控制帧; 1: 丢弃未知控制帧。	0x1

TRANSMIT_CONTROL

TRANSMIT_CONTROL 为常用配置寄存器。

Offset Address: 0x0060 Total Reset Value: 0x0000_00C0

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x00000000



Bits	Access	Name	Description	Reset
[7]	RW	pad_enable	发送添加 PAD 使能。 0: 不使能; 1: 使能。	0x1
[6]	RW	crc_add	发送添加 FCS 使能。 0: 不使能; 1: 使能。	0x1
[5:0]	-	reserved	保留。	0x00

REC_FILT_CONTROL

REC_FILT_CONTROL 为接收帧过滤控制寄存器。

Offset Address: 0x0064 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x00000000
[5]	RW	crc_err_pass	过滤 CRC 错误帧使能。 0: 不使能; 1: 使能, 表示把 CRC 错误帧丢弃。	0x0
[4]	RW	pause_frm_pass	过滤流控帧使能。 0: 不使能, 流控使能有效才起作用, 要上传至软件; 1: 使能, 流控使能有效才起作用, 不上传至软件。	0x0
[3]	RW	vlan_drop_en	过滤 VLAN 帧使能。 0: 不使能; 1: 使能。	0x0
[2]	RW	bc_drop_en	过滤广播帧使能。 0: 不使能;	0x0



Bits	Access	Name	Description	Reset
			1: 使能。	
[1]	RW	mc_match_en	过滤 DA 不匹配的多播帧使能。 0: 不使能; 1: 使能。	0x0
[0]	RW	uc_match_en	过滤 DA 不匹配的单播帧使能。 0: 不使能; 1: 使能。	0x0

RX_OCTETS_OK_CNT

RX_OCTETS_OK_CNT 为接收有效帧的字节统计寄存器。

Offset Address: 0x0080 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RC	rx_octets_ok_cnt	接收有效帧字节统计, 范围包括 DA ~ FCS。	0x00000000

RX_OCTETS_BAD_CNT

RX_OCTETS_BAD_CNT 为接收错误帧字节统计寄存器。

Offset Address: 0x0084 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RC	rx_octets_bad_cnt	接收错帧字节统计, 包括 CRC 错误, 对齐错误等。	0x00000000

RX_UC_PKTS

RX_UC_PKTS 为 MAC 接收单播帧数统计寄存器。

Offset Address: 0x0088 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RC	rx_uc_pkts_cnt	接收单播帧数统计(不包括错帧)。	0x00000000

RX_MC_PKTS

RX_MC_PKTS 为接收多播帧数统计寄存器。

Offset Address: 0x008C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RC	rx_mc_pkts_cnt	接收多播帧数统计(不包括错帧)。	0x00000000

RX_BC_PKTS

RX_BC_PKTS 为接收的广播帧数统计寄存器。

Offset Address: 0x0090 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RC	rx_bc_pkts_cnt	接收广播帧数统计(不包括错帧)。	0x00000000

RX_FCS_ERRORS

RX_FCS_ERRORS 为接收 CRC 检验错误的帧数统计寄存器。

Offset Address: 0x00B0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RC	rx_fcs_errors	CRC 检测错误的帧数统计(不包括短帧)。	0x00000000

OCTETS_TRANSMITTED_OK

OCTETS_TRANSMITTED_OK 为发送成功的好包字节数统计寄存器。

Offset Address: 0x0100 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RC	octets_transmitted_ok	发送成功的好包字节数(不包括前导码和 SFD)。	0x00000000

OCTETS_TRANSMITTED_BAD

OCTETS_TRANSMITTED_BAD 为发送成功的错包字节数统计寄存器。

Offset Address: 0x0104 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RC	octets_transmitted_bad	发送成功的错包的字节数。	0x00000000

TX_UC_PKTS

TX_UC_PKTS 为发送的单播帧数统计寄存器。

Offset Address: 0x0108 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RC	tx_uc_pkts	发送的单播帧数统计(不包括错包)。	0x00000000

TX_MC_PKTS

TX_MC_PKTS 为发送的多播帧数统计寄存器。

Offset Address: 0x010C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RC	tx_mc_pkts	发送的多播帧数统计(不包括错包)。	0x00000000

TX_BC_PKTS

TX_BC_PKTS 为发送的广播帧数统计寄存器。



Offset Address: 0x0110 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RC	tx_bc_pkts	发送的广播帧数(不包括错包)。	0x00000000

TX_CRC_ERROR

TX_CRC_ERROR 为发送帧长正确 CRC 错误的帧数统计寄存器。

Offset Address: 0x0158 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RC	tx_crc_error	发送帧长正确而 CRC 错误的帧数目统计。	0x00000000

CF_CRC_STRIP

CF_CRC_STRIP 为 CRC 剥离使能寄存器。

Offset Address: 0x01B0 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	cf_crc_strip	MAC 剥离接收方向 CRC 使能。 0: 不使能, 上报包长包括 CRC 的 4 字节; 1: 使能, 剥离后上报包长不包括 CRC 的 4 字节。	0x1

MODE_CHANGE_EN

MODE_CHANGE_EN 为端口模式改变使能寄存器。

Offset Address: 0x01B4 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	mode_change_en	port_mode 改变生效使能。 0: 不使能; 1: 使能。	0x0

COL_SLOT_TIME

COL_SLOT_TIME 为半双工冲突重传间隔时间计数器寄存器。

Offset Address: 0x01C0 Total Reset Value: 0x0000_40FF

Bits	Access	Name	Description	Reset
[31:24]	RO	reserved	保留。	0x00
[23:8]	RW	cf2bc_slottime	半双工冲突重传单位间隔时间。	0x0040
[7:0]	RW	cf2bc_random_seed	半双工冲突重传随机倍数基数。	0xFF

RECV_CONTROL

RECV_CONTROL 为接收控制寄存器。

Offset Address: 0x01E0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x00000000
[4]	RW	runt_pkt_en	接收超短帧透穿功能。 0: 丢弃, 不上传给软件; 1: 上传给软件。	0x0
[3]	RW	strip_pad_en	剥离接收帧的 PAD 使能。 0: 不使能; 1: 使能。	0x0



[2:0]	-	reserved	保留。	0x0
-------	---	----------	-----	-----

VLAN_CODE

VLAN_CODE 为 VLAN Code 寄存器。

Offset Address: 0x01E8 Total Reset Value: 0x0000_8100

Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	保留。	0x00000
[15:0]	RW	cf_vlan_code	Ethernet Type 域配置。	0x8100

CRF_MAX_PACKET

CRF_MAX_PACKET 为最大过滤包长寄存器。

Offset Address: 0x020C Total Reset Value: 0x05EE_0000

Bits	Access	Name	Description	Reset
[31:27]	RO	reserved	保留。	0x0
[26:16]	RW	crf_tx_max_packet	PMU 中允许的 normal 包和 SG 包最大长度。	0x5EE
[15:0]	RO	reserved	保留。	0x0000

CRF_MIN_PACKET

CRF_MIN_PACKET 为最小过滤包长寄存器。

Offset Address: 0x0210 Total Reset Value: 0x0000_0F2A

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13:8]	RW	crf_tx_min_packet	发送方向配置的最小发送包长度, 默	0x0F



Bits	Access	Name	Description	Reset
			认为 15byte。	
[7:6]	-	reserved	保留。	0x0
[5:0]	RW	crf_rx_min_packet	接收方向配置的最小发送包长度，默认为 42byte。	0x2A

CONTROL_WORD

CONTROL_WORD 为控制寄存器。

Offset Address: 0x0214 Total Reset Value: 0x0000_0640

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25]	RW	crf_tx_standard	发送 FIFO 的发送水线设置标准。 0: 按包和将空水线设置。当发送 FIFO 中有一个完整的包，或者发送 FIFO 中的有效数据个数大于等于 4 倍的发送水线时，即向 MAC 发送读请求； 1: 按包设置；当发送 FIFO 中有一个完整的包时，才向 MAC 发送读请求。	0x0
[24:22]	-	reserved	保留。	0x0
[21]	RW	crf_ip_flow_ctrl	IP 报文限速使能。 0: 不限速； 1: 限速。	0x0
[20]	-	reserved	保留。	0x0
[19:18]	-	reserved	保留。	0x0
[17]	RW	crf_filt_unused_pk g	过滤非法报文控制。 0 不过滤；	0x0



Bits	Access	Name	Description	Reset
			1: 过滤。	
[16]	RW	crf_bm_flow_ctrl	对广播或多播报文流控控制。 0: 不流控; 1: 流控。	0x0
[15:14]	-	reserved	保留。	0x0
[13:0]	RW	crf_large_packet	配置的最大包长度, 默认为 1600byte(PMU 使用的最大包长)。	0x0640

FLOW_CTRL_PKG_THRSLD

FLOW_CTRL_PKG_THRSLD 为限速包个数寄存器。

Offset Address: 0x0218 Total Reset Value: 0xFFFF_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	crf_ip_pkg_thrsld	IP 报文的包上限, 当在 T 时间内接收的 IP 报文超过该数, 则进行限速, 否则, 不限速。	0xFFFF
[15:0]	-	reserved	保留。	0x0000

CRF_FLOW_TIME_THRSLD

CRF_FLOW_TIME_THRSLD 为限速时间寄存器。

Offset Address: 0x021C Total Reset Value: 0x0000_00FF

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	crf_flow_time_thrsld	限速处理的时间, 以 125μs 为单位。 限速时间 $T=(crf_flow_time_thrsld+1)(125\mu s)$	0xFF



CRF_BM_PKT_THRSLD

CRF_BM_PKT_THRSLD 为广播和多播报文的限速处理包个数寄存器。

Offset Address: 0x0358 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	crf_bm_pkt_thrslld	广播和多播报文的包上限，当在限速单位时间内接收的广播或多播报文超过该数，则进行限速，否则不限速。	0x0001

TSO_COE_CTRL

TSO_COE_CTRL 为 TSO 开关和 COE 过滤开关寄存器。

Offset Address: 0x02E8 Total Reset Value: 0x8000_0081

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x8000
[15:8]	-	reserved	保留。	0x00
[7]	RW	coe_drop_cnt_en	接收报文统计计数使能。 0: 关闭; 1: 打开。	0x1
[6]	RW	coe_ipv6udp_zero_drop	接收报文 IPv6 UDP checksum 为 0 的包是否配置丢弃。 0: 关闭; 1: 打开。	0x0
[5]	RW	coe_payload_drop	接收报文 TCP/UDP checksum 错误的包是否配置丢弃。 0: 关闭; 1: 打开。	0x0
[4]	RW	coe_iphead_drop	接收报文 IPv4 头 checksum 错误的包	0x0



Bits	Access	Name	Description	Reset
			是否配置丢弃。 0: 关闭; 1: 打开。	
[3:0]	-	reserved	保留。	0x1

CRF_BM_TIME_THRSLD

CRF_BM_TIME_THRSLD 为广播和多播报文的限速时间计数寄存器。

Offset Address: 0x035C Total Reset Value: 0x0000_2710

Bits	Access	Name	Description	Reset
[31:20]	RO	reserved	保留。	0x000
[19:0]	RW	crf_bm_time_thrsl d	广播和多播报文的限速时间上限, 以 1us 为单位进行计数, 当等于该计数 值时, 为一个限速单位时间。	0x02710

MDIO_SINGLE_CMD

MDIO_SINGLE_CMD 为 MDIO 单次操作寄存器。

Offset Address: 0x03C0 Total Reset Value: 0x0001_0000

Bits	Access	Name	Description	Reset
[31:21]	-	reserved	保留。	0x000
[20]	RW	mdio_cmd	MDIO 操作完成指示。 0: MDIO 操作完成; 1: 启动 MDIO 操作。	0x0
[19:18]	-	reserved	保留。	0x0
[17:16]	RW	op_code	MDIO 操作类型。 00: 保留;	0x1



Bits	Access	Name	Description	Reset
			01: 写操作; 10: 读操作; 11: 保留。	
[15:13]	-	reserved	保留。	0x0
[12:8]	RW	phy_addr	配置外部 PHY 地址的 5bit。	0x00
[7:5]	-	reserved	保留。	0x0
[4:0]	RW	reg_addr	PHY 器件内部的寄存器地址。	0x00

MDIO_SINGLE_DATA

MDIO_SINGLE_DATA 为 MDIO 读写数据寄存器。

Offset Address: 0x03C4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	mdio_rd_data	MDIO 从外部 PHY 器件回读的数据。	0x0000
[15:0]	RW	mdio_wr_data	MDIO 写数据。	0x0000

ENA_PMU_INT

ENA_PMU_INT 为 PMU 模块原始中断使能寄存器。

Offset Address: 0x05C4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RO	reserved	保留。	0x0
[30]	RW	ena_mac_fifo_err_int	MAC 内部 FIFO 错误中断使能。 0: 不使能; 1: 使能。	0x0
[29]	RW	ena_tx_rq_in_timeout_int	发送方向的 RQ 队列描述子入队超时中断使能。	0x0



Bits	Access	Name	Description	Reset
			0: 不使能; 1: 使能。	
[28]	RW	ena_rx_bq_in_time out_int	接收方向的 BQ 队列描述子入队超时 中断使能。 0: 不使能; 1: 使能。	0x0
[27]	RW	ena_txoutcff_full_i nt	发送方向 DESC_OUTCFF 满中断使 能。 0: 不使能; 1: 使能。	0x0
[26]	RW	ena_txoutcff_empt y_int	发送方向 DESC_OUTCFF 空中断使 能。 0: 不使能; 1: 使能。	0x0
[25]	RW	ena_txcff_full_int	发送方向 DESC_FIFO 满中断使能。 0: 不使能; 1: 使能。	0x0
[24]	RW	ena_txcff_empty_i nt	发送方向 DESC_FIFO 空中断使能。 0: 不使能; 1: 使能。	0x0
[23]	RW	ena_rxoutcff_full_i nt	接收方向 DESC_OUTCFF 满中断使 能。 0: 不使能; 1: 使能。	0x0
[22]	RW	ena_rxoutcff_empt y_int	接收方向 DESC_OUTCFF 空中断使 能。 0: 不使能; 1: 使能。	0x0



Bits	Access	Name	Description	Reset
[21]	RW	ena_rxcff_full_int	接收方向 DESC_FIFO 满中断使能。 0: 不使能; 1: 使能。	0x0
[20]	RW	ena_rxcff_empty_int	接收方向 DESC_FIFO 空中断使能。 0: 不使能; 1: 使能。	0x0
[19]	RW	ena_tx_rq_in_int	发送方向 tx_rq 队列的描述子入队(多个或者单个描述子入队)中断使能。 0: 不使能; 1: 使能。	0x0
[18]	RW	ena_tx_bq_out_int	发送方向 tx_bq 队列的描述子出队(多个或者单个描述子出队)中断使能。 0: 不使能; 1: 使能。	0x0
[17]	RW	ena_rx_bq_in_int	接收方向 rx_bq 队列的描述子入队(多个或者单个描述子入队)中断使能。 0: 不使能; 1: 使能。	0x0
[16]	RW	ena_rx_fq_out_int	接收方向 rx_fq 队列的描述子出队(多个或者单个描述子出队)中断使能。 0: 不使能; 1: 使能。	0x0
[15]	RW	ena_tx_rq_empty_int	发送方向的回收描述子队列空中断使能。 0: 不使能; 1: 使能。	0x0
[14]	RW	ena_tx_rq_full_int	发送方向的回收描述子队列满中断使能。	0x0



Bits	Access	Name	Description	Reset
			0: 不使能; 1: 使能。	
[13]	RW	ena_tx_rq_alempty_int	发送方向的回收描述子队列几乎空中断使能。 0: 不使能; 1: 使能。	0x0
[12]	RW	ena_tx_rq_alfull_int	发送方向的回收描述子队列几乎满中断使能。 0: 不使能; 1: 使能。	0x0
[11]	RW	ena_tx_bq_empty_int	发送方向的 buff 描述子队列空中断使能。 0: 不使能; 1: 使能。	0x0
[10]	RW	ena_tx_bq_full_int	发送方向的 buff 描述子队列满中断使能。 0: 不使能。 1: 使能	0x0
[9]	RW	ena_tx_bq_alempty_int	发送方向的 buff 描述子队列几乎空中断使能。 0: 不使能; 1: 使能。	0x0
[8]	RW	ena_tx_bq_alfull_int	发送方向的 buff 描述子队列几乎满中断使能。 0: 不使能; 1: 使能。	0x0
[7]	RW	ena_rx_bq_empty_int	接收方向的 buff 描述子队列空中断使能。	0x0



Bits	Access	Name	Description	Reset
			0: 不使能; 1: 使能。	
[6]	RW	ena_rx_bq_full_int	接收方向的 buff 描述子队列满中断使能。 0: 不使能; 1: 使能。	0x0
[5]	RW	ena_rx_bq_alempy_int	接收方向的 buff 描述子队列几乎空中断使能。 0: 不使能; 1: 使能。	0x0
[4]	RW	ena_rx_bq_alfull_int	接收方向的 buff 描述子队列几乎满中断使能。 0: 不使能; 1: 使能。	0x0
[3]	RW	ena_rx_fq_empty_int	接收方向的空闲描述子队列空中断使能。 0: 不使能; 1: 使能。	0x0
[2]	RW	ena_rx_fq_full_int	接收方向的空闲描述子队列满中断使能。 0: 不使能; 1: 使能。	0x0
[1]	RW	ena_rx_fq_alempy_int	接收方向的空闲描述子队列几乎空中断使能。 0: 不使能; 1: 使能。	0x0
[0]	RW	ena_rx_fq_alfull_int	接收方向的空闲描述子队列几乎满中断使能。	0x0



Bits	Access	Name	Description	Reset
			0: 不使能; 1: 使能。	

STATUS_PMU_INT

STATUS_PMU_INT 为 PMU 模块中断状态寄存器。

Offset Address: 0x05C8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30]	RW	status_mac_fifo_err_int	MAC 内部 FIFO 又空又满错误中断状态。 0: 无中断; 1: 有中断。	0x0
[29]	RW	status_tx_rq_in_timeout_int	发送方向的 RQ 队列描述子入队超时中断状态。 0: 无中断; 1: 有中断。	0x0
[28]	RW	status_rx_bq_in_timeout_int	接收方向的 BQ 队列描述子入队超时中断状态。 0: 无中断; 1: 有中断。	0x0
[27]	RW	status_txoutcfull_int	发送方向 DESC_OUTCFF 满中断状态。 0: 无中断; 1: 有中断。	0x0
[26]	RW	status_txoutcempty_int	发送方向 DESC_OUTCFF 空中断状态。 0: 无中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	
[25]	RW	status_txcff_full_int	发送方向 DESC_FIFO 满中断状态。 0: 无中断; 1: 有中断。	0x0
[24]	RW	status_txcff_empty_int	发送方向 DESC_FIFO 空中断状态。 0: 无中断; 1: 有中断。	0x0
[23]	RW	status_rxoutcff_full_int	接收方向 DESC_OUTCFF 满中断状态。 0: 无中断; 1: 有中断。	0x0
[22]	RW	status_rxoutcff_empty_int	接收方向 DESC_OUTCFF 空中断状态。 0: 无中断; 1: 有中断。	0x0
[21]	RW	status_rxcff_full_int	接收方向 DESC_FIFO 满中断状态。 0: 无中断; 1: 有中断。	0x0
[20]	RW	status_rxcff_empty_int	接收方向 DESC_FIFO 空中断状态。 0: 无中断; 1: 有中断。	0x0
[19]	RW	status_tx_rq_in_int	发送方向 tx_rq 队列的描述子入队(多个或者单个描述子入队)中断状态。 0: 无中断; 1: 有中断。	0x0
[18]	RW	status_tx_bq_out_int	发送方向 tx_bq 队列的描述子出队(多个或者单个描述子出队)中断状态。 0: 无中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	
[17]	RW	status_rx_bq_in_int	接收方向 rx_bq 队列的描述子入队(多个或者单个描述子入队)中断状态。 0: 无中断; 1: 有中断。	0x0
[16]	RW	status_rx_fq_out_int	接收方向 rx_fq 队列的描述子出队(多个或者单个描述子出队)中断状态。 0: 无中断; 1: 有中断。	0x0
[15]	RW	status_tx_rq_empty_int	发送方向的回收描述子队列空中断状态。 0: 无中断; 1: 有中断。	0x0
[14]	RW	status_tx_rq_full_int	发送方向的回收描述子队列满中断状态。 0: 无中断; 1: 有中断。	0x0
[13]	RW	status_tx_rq_alempy_int	发送方向的回收描述子队列几乎空中断状态。 0: 无中断; 1: 有中断。	0x0
[12]	RW	status_tx_rq_alfull_int	发送方向的回收描述子队列几乎满中断状态。 0: 无中断; 1: 有中断。	0x0
[11]	RW	status_tx_bq_empty_int	发送方向的 buff 描述子队列空中断状态。 0: 无中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	
[10]	RW	status_tx_bq_full_int	发送方向的 buff 描述子队列满中断状态。 0: 无中断; 1: 有中断。	0x0
[9]	RW	status_tx_bq_alempty_int	发送方向的 buff 描述子队列几乎空中断状态。 0: 无中断; 1: 有中断。	0x0
[8]	RW	status_tx_bq_alfull_int	发送方向的 buff 描述子队列几乎满中断状态。 0: 无中断; 1: 有中断。	0x0
[7]	RW	status_rx_bq_empty_int	接收方向的 buff 描述子队列空中断状态。 0: 无中断; 1: 有中断。	0x0
[6]	RW	status_rx_bq_full_int	接收方向的 buff 描述子队列满中断状态。 0: 无中断; 1: 有中断。	0x0
[5]	RW	status_rx_bq_alempty_int	接收方向的 buff 描述子队列几乎空中断状态。 0: 无中断; 1: 有中断。	0x0
[4]	RW	status_rx_bq_alfull_int	接收方向的 buff 描述子队列几乎满中断状态。 0: 无中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	
[3]	RW	status_rx_fq_empty_int	接收方向的空闲描述子队列空中断状态。 0: 无中断; 1: 有中断。	0x0
[2]	RW	status_rx_fq_full_int	接收方向的空闲描述子队列满中断状态。 0: 无中断; 1: 有中断。	0x0
[1]	RW	status_rx_fq_alempty_int	接收方向的空闲描述子队列几乎空中断状态。 0: 无中断; 1: 有中断。	0x0
[0]	RW	status_rx_fq_alfull_int	接收方向的空闲描述子队列几乎满中断状态。 0: 无中断; 1: 有中断。	0x0

DESC_WR_RD_ENA

DESC_WR_RD_ENA 为 cff 读写描述子使能寄存器。

Offset Address: 0x05CC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	RO	reserved	保留。	0x00000000
[3]	RW	rx_outcff_wr_desc_ena	接收方向的 RX_OUTCFF 向 RX_BQ 中写入 desc 使能。 0: 不使能; 1: 使能。	0x0



Bits	Access	Name	Description	Reset
[2]	RW	rx_cff_rd_desc_ena	接收方向的 RX_CFF 从空闲描述子队列中读取 desc 使能。 0: 不使能; 1: 使能。	0x0
[1]	RW	tx_outcff_wr_desc_ena	发送方向的 TX_OUTCFF 向 TX_RQ 中写入 desc 使能。 0: 不使能; 1: 使能。	0x0
[0]	RW	tx_cff_rd_desc_ena	发送方向的 TX_CFF 从 TX_BQ 中读取 desc 使能。 0: 不使能; 1: 使能。	0x0

MAC_IF_STAT_CTRL

MAC_IF_STAT_CTRL 为 MAC_IF 模式控制寄存器寄存器。

Offset Address: 0x300C Total Reset Value: 0x0000_003F

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:5]	RW	phy_select	PHY 接口模式。 001: RGMII mode; 100: RMII mode; 其他: 保留。	0x1
[4]	RW	duplex_mode	PHY 双工模式。 0: 半双工模式; 1: 全双工模式。	0x1
[3]	RW	tx_config	发送配置使能信号。 0: Tx Config Disable;	0x1



Bits	Access	Name	Description	Reset
			1: Tx Config Enable。	
[2]	RW	link_status	PHY 连接状态控制。 0: Link Down; 1: Link Up。	0x1
[1]	RW	mac_speed	10/100Mbps 模式。 0: 10Mbps; 1: 100Mbps。	0x1
[0]	RW	port_select	网口模式选择。 0: 1000Mbps; 1: 10/100Mbps。	0x1



目 录

7 视频编解码	7-1
7.1 VEDU.....	7-1
7.1.1 概述.....	7-1
7.1.2 特点.....	7-1
7.1.3 功能描述.....	7-4
7.2 JPGE/JPGD.....	7-6
7.2.1 概述.....	7-6
7.2.2 特点.....	7-7
7.2.3 功能描述.....	7-8
7.2.4 工作方式.....	7-11



插图目录

图 7-1 VEDU 编码功能框图	7-5
图 7-2 VEDU 解码功能框图	7-6
图 7-3 JPGE 功能框图	7-9
图 7-4 JPGD 总体结构图	7-10
图 7-5 JPEG 码流结构图	7-11



7 视频编解码

7.1 VEDU

7.1.1 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。

7.1.2 特点

VEDU 编码具有以下特点：

- 支持 ITU-T H.265Main Profile @Level 5.1 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 32x32、16x16、8x8 三种 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 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：
 - 3840x2160@30fps+1920x1080@30fps+720x480@30fps 编码
 - 3840x2160@30fps(编码)+720x480@30fps(编码)+1920x1080@30fps(解码)
- 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：4096x4096
 - 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 256 级的 alpha 叠加
 - OSD 叠加功能可使能/禁止
 - 支持 OSD ARGB1555、ARGB4444 格式、CLUT2、CLUT4 格式



- 支持彩转灰编码功能
- 支持输入低延时功能
- 支持视频前端叠加 mosaic 编码
- 支持数字水印编码
- 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 80Mbps，H.264 最高 80Mbps。
- 支持 slice 输出中断

VEDU 解码具有以下特点：

- 支持 ITU-T H.265Main Profile @Level 5.1 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 CTB32x32、CTB64x64 解码
 - 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
 - 不支持 TILE 解码
- 支持 ITU-T H.264 High Profile/Main Profile/Baseline Profile@Level 5.1 解码
 - 支持 1/2、1/4 像素精度运动补偿
 - 支持帧间预测 16x16、8x8、8x4、4x8、4x4 五种子块类型
 - 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 3840x2160@30fps 解码
- 支持图像分辨率
 - 最小图像分辨率: 114x114
 - 最大图像分辨率: 6144x6144

须知

VEDU 的区域编码保护功能只对 VEDU 叠加的 OSD 起作用。

7.1.3 功能描述

VEDU 功能框图如图 7-1 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。

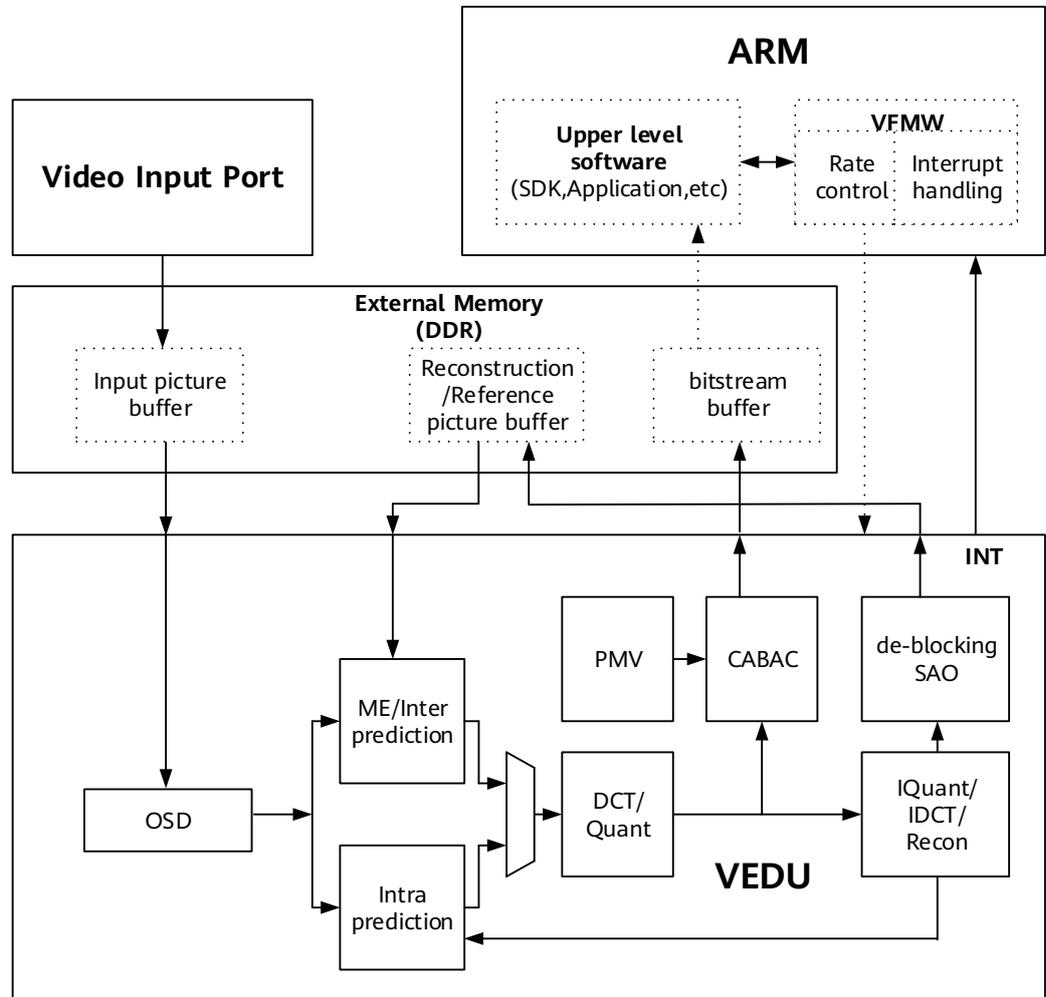
- 输入图像缓冲区

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7-1 VEDU 编码功能框图



VEDU 解码功能框图如图 7-2 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。

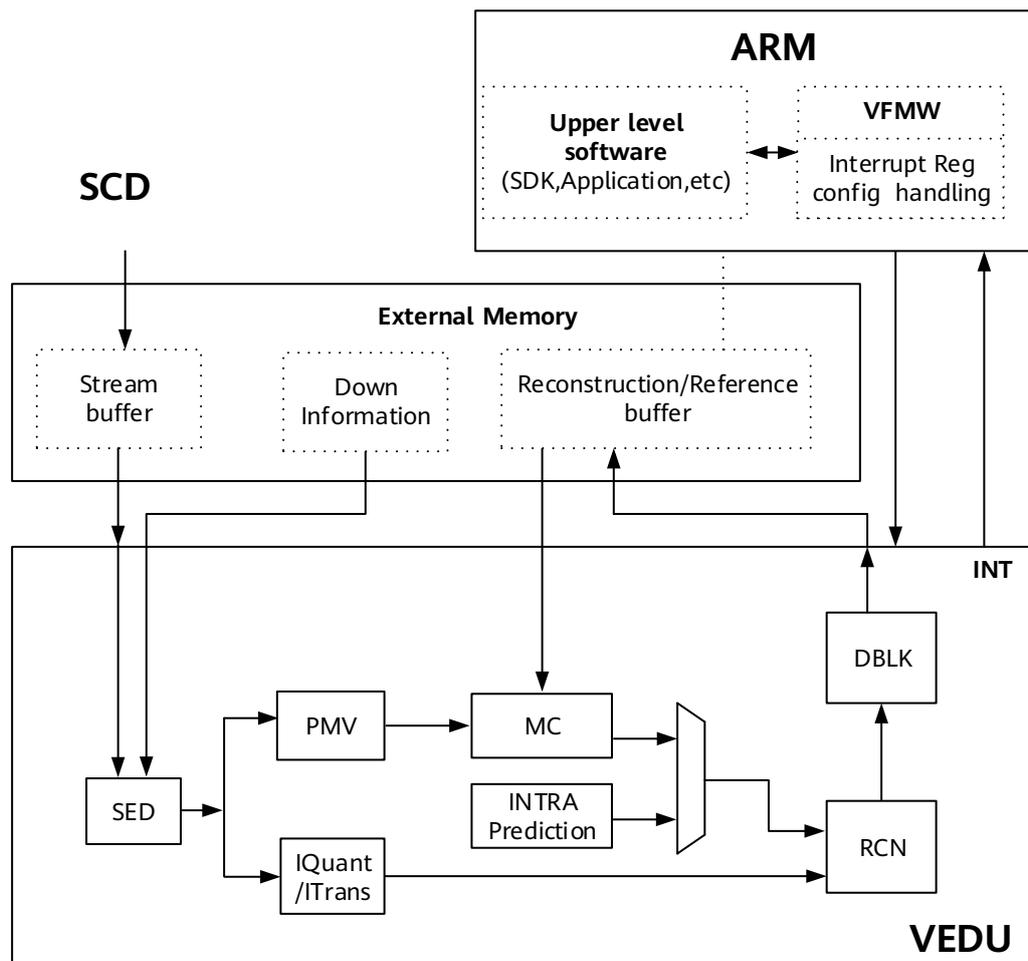
在启动 VEDU 进行视频解码前，软件需要为其在外部存储器（DDR SDRAM）中分配以下三种类型的缓冲区。

- 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7-2 VEDU 解码功能框图



7.2 JPGE/JPGD

7.2.1 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。

7.2.2 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 3840x2160@60fps (YUV420)
 - 支持的图像分辨率
 - 最小图像分辨率: 32x32
 - 最大图像分辨率: 16384x16384
 - 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 256 级的 alpha 叠加
 - OSD 叠加功能可使能/禁止
 - 支持 OSD ARGB1555、ARGB4444 格式、CLUT2、CLUT4 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 16384x16384 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。
- 支持输出低延时功能
- 解码性能：3840x2160@30fps (YUV420 格式)。

7.2.3 功能描述

JPGE 功能如图 7-3 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：

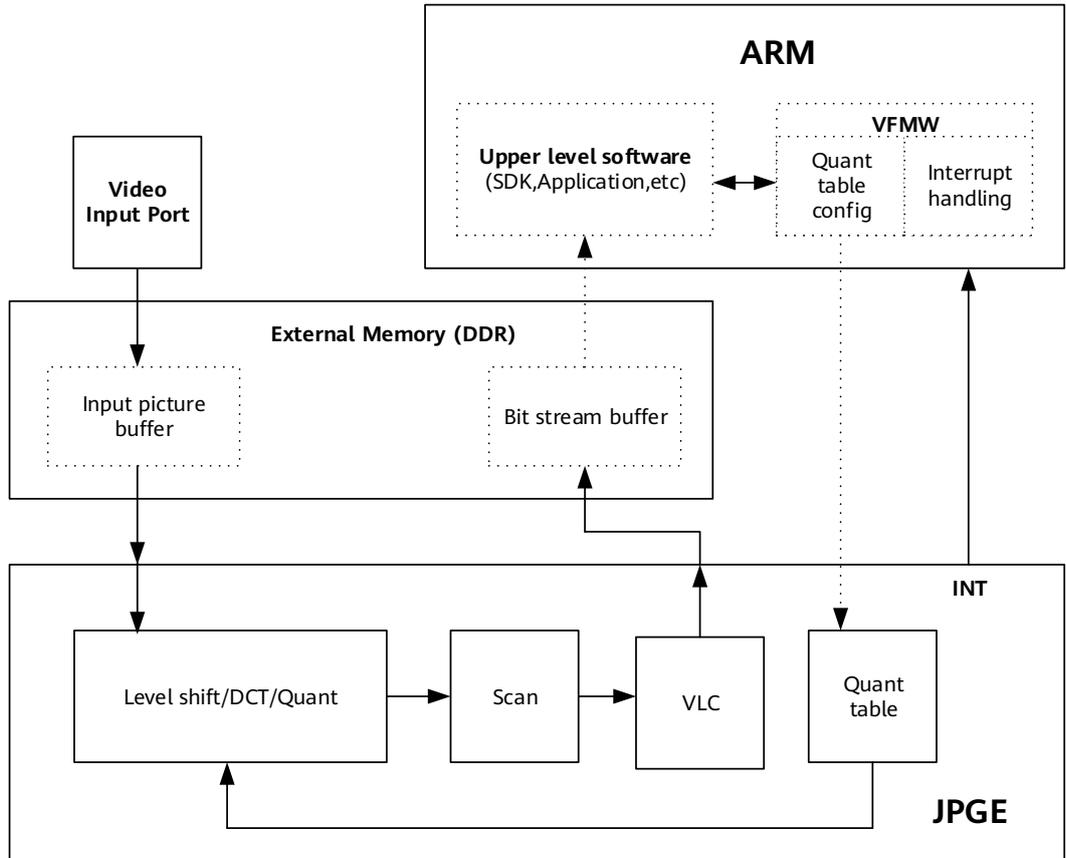
- 输入图像缓冲区

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

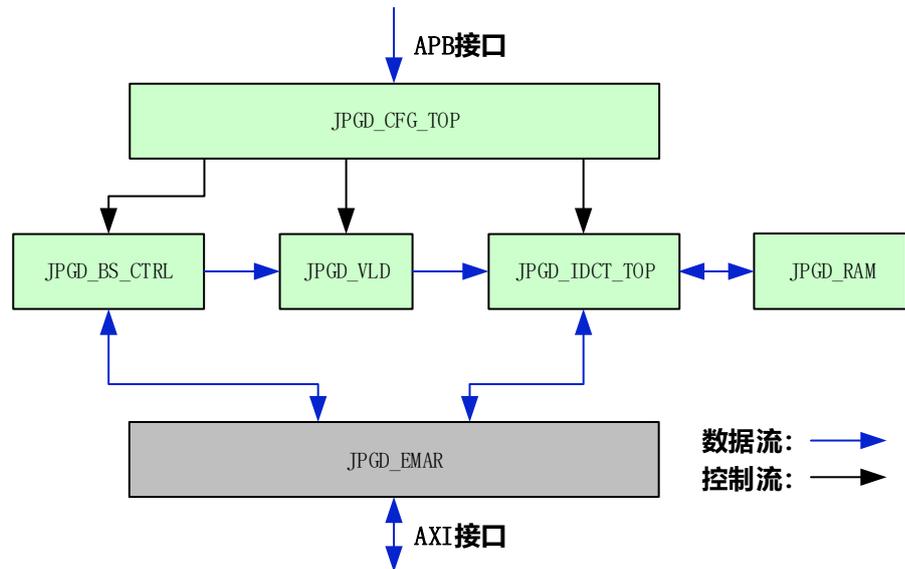
该缓冲区用于存放编码输出的码流。JPGE 在编码过程中会将码流写入该缓冲区。该缓冲区通常由软件读取。

图7-3 JPGE 功能框图



JPGD 总体结构如图 7-4 所示。

图7-4 JPGD 总体结构图



JPGD 内部模块说明如表 7-1 所示。

表7-1 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_TOP	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。

7.2.4 工作方式

7.2.4.1 软硬件划分

JPEG 码流结构如图 7-5 所示。

图7-5 JPEG 码流结构图

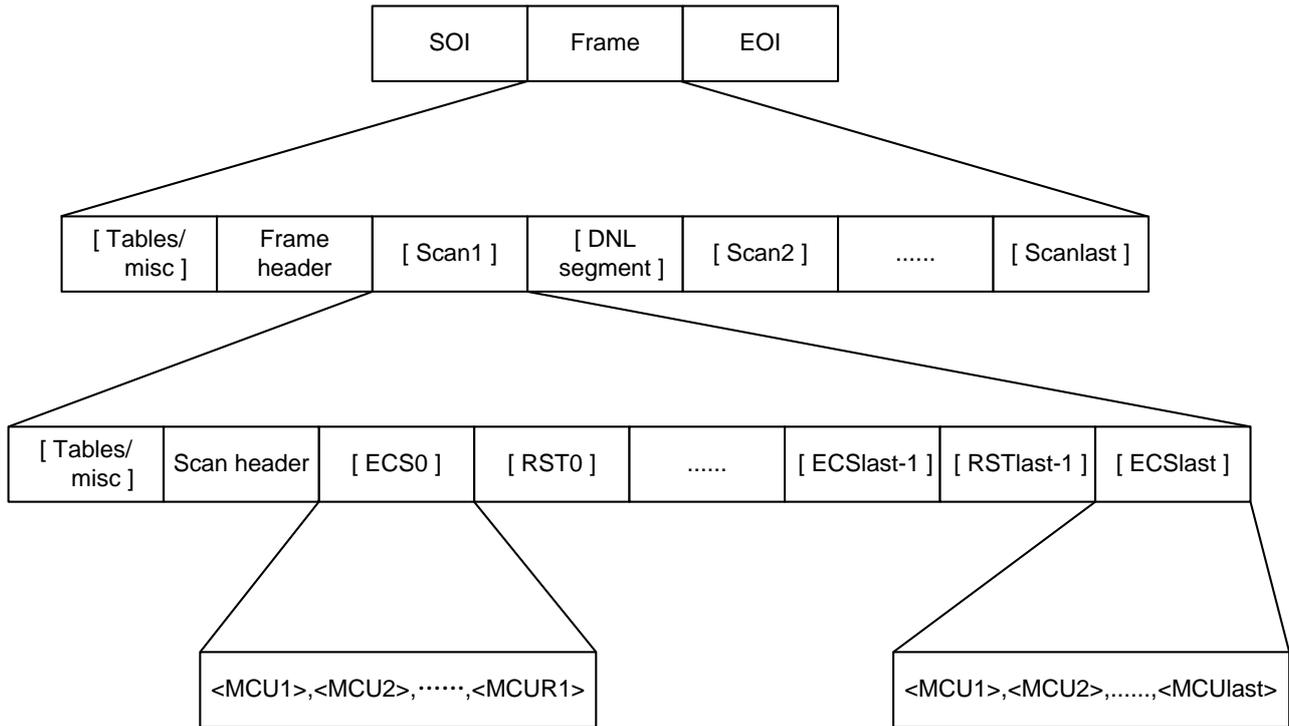


图 7-5 为广义结构图，对于 JPEG 码流，由软件解析 Scan header 及其以上部分，硬件解析 ECS 层和 RSTn 标志。

7.2.4.2 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。
- 低延时行号中断，表示当前图像解码达到了软件配置的行号的水线，JPGD 继续解码。



目 录

目 录.....	i
8 视频及图形处理.....	8-1
8.1 TDE.....	8-1
8.1.1 概述.....	8-1
8.1.2 功能描述.....	8-1
8.2 VPSS.....	8-2
8.2.1 概述.....	8-2
8.2.2 功能描述.....	8-3
8.3 VGS.....	8-4
8.3.1 概述.....	8-4
8.4 GDC.....	8-5
8.4.1 概述.....	8-5



8 视频及图形处理

8.1 TDE

8.1.1 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。

8.1.2 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。
- 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、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。
- 支持角框。
- 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 4096x4096。

8.2 VPSS

8.2.1 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 功能。

VPSS 特性如下:

- 最大分辨率 8192x8192 (水平分辨率超过 4096 需工作在 stripping 模式)。
- 支持最大宽度 4096 的视频源。
- 支持最多 4 个通道视频输出。
- 支持 3D 降噪功能。
- 输出支持 12 个区域的马赛克处理。
- 输出支持 12 个区域的视频矩形实心遮挡。
- 输出支持 12 个区域的视频矩形角框。
- 支持 LetterBox 功能。
- 支持 4 个输出通道的视频裁剪和 mirror/flip 功能。

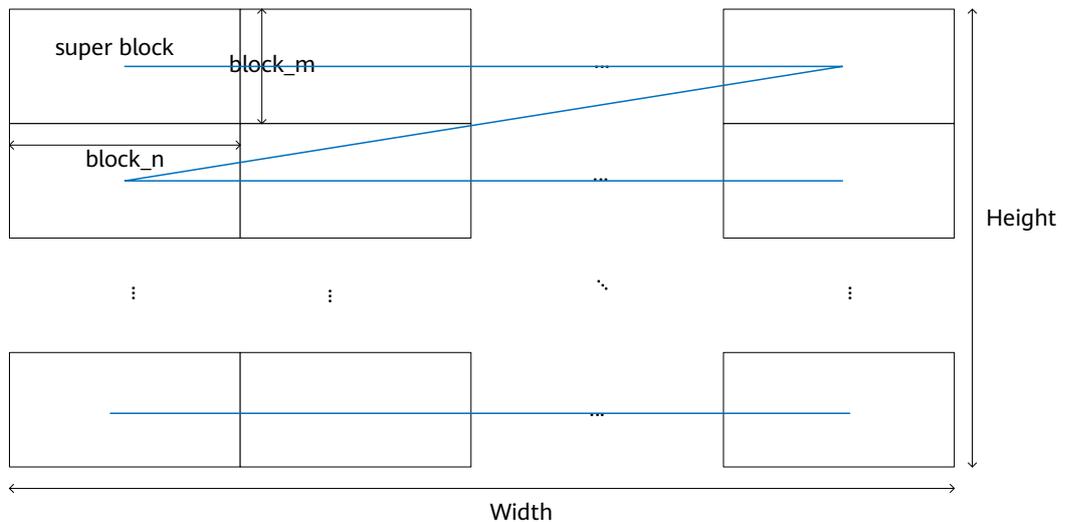


- 支持视频数据压缩/解压缩。
- 输入数据为 YUV420 semi-planar/ YUV422 semi-planar/单分量。
- 输入数据支持 linear 格式。
- 输出数据为 YUV420 semi-planar/ YUV422 semi-planar/单分量。
- 输出数据支持 linear 格式。
- 支持亮度单分量处理。
- 支持输入、输出通道低延时模式。

注：马赛克/矩形遮挡/角框一共 12 个。

8.2.2 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：CH(Channel)0-CH3 同时支持 1-1/15 倍缩小。CH0-CH3 支持任意 1 个通道 16 倍放大。
 - 在线模式下， CH0~CH3 支持 1~1/15 倍缩小。





8.3 VGS

8.3.1 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、角框、画点画线功能。

VGS 特性如下：

- 最大分辨率 8192x8192 (水平分辨率超过 3072 需工作在 stripping 模式)。
- VGS 支持三个通道视频输出。
- 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/CLUT2/CLUT4
- 输入数据支持 YUV420 semi-planar/ YUV422 semi-planar/单分量/YUV422 package 格式。
- 输出数据支持 YUV420 semi-planar/ YUV422 semi-planar/单分量/YUV422 package 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。
- 输出支持 4 个区域的马赛克处理。
- 输出支持 8 个区域的视频矩形实心遮挡。支持视频裁剪。
- 输出支持 8 个区域的角框和实线框。
- 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。
- 支持输入、输出通道低延时模式。
- 支持 LetterBox 功能。
- 支持水平拼接。



8.4 GDC

8.4.1 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：

- 支持最大输入宽度 8192 的视频源。
- 支持 YUV420 semi-planar 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。
- 支持鱼眼矫正
 - 支持 360、180、normal 三种模型
 - 支持顶装，壁装，桌装三种安装方式
 - 支持鱼眼三种矫正方式下的 PTZ 功能
 - 支持 180/normal 壁装模式下的俯仰角矫正功能
 - 支持 180 模式下的扇形矫正功能
- 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。
- 支持输入、输出通道低延时模式。



目 录

9 智能加速引擎	9-1
9.1 NPU 加速引擎.....	9-1
9.1.1 概述.....	9-1
9.1.2 特点.....	9-2
9.2 DPU.....	9-3
9.2.1 概述.....	9-3
9.2.2 特点.....	9-3
9.2.3 功能描述.....	9-4
9.3 IVE.....	9-4
9.3.1 概述.....	9-4
9.3.2 功能描述.....	9-5
9.3.3 工作方式.....	9-7



插图目录

图 9-1 NPU 加速引擎在系统中的位置.....	9-2
图 9-2 DPU 在系统中的位置.....	9-4
图 9-3 IVE 在系统中的位置.....	9-5
图 9-4 SP422 存储格式.....	9-7
图 9-5 SP420 存储格式.....	9-7
图 9-6 8bit 单分量数据在 Memory 中的存储.....	9-8
图 9-7 Package 存储格式.....	9-8
图 9-8 Plannar 存储格式.....	9-9
图 9-9 16bit 单分量数据在 Memory 中的存储.....	9-9
图 9-10 32bit 单分量数据在 Memory 中的存储.....	9-10
图 9-11 64bit 单分量数据在 Memory 中的存储.....	9-10
图 9-12 NCC 输出数据在 Memory 中的存储.....	9-10
图 9-13 CCL 统计信息在 Memory 中的存储（顺序存放）.....	9-10
图 9-14 16bit 水平和垂直方向的间插数据在 Memory 中的存储.....	9-11
图 9-15 HOG 输出格式 1.....	9-11
图 9-16 HOG 输出格式 2.....	9-12



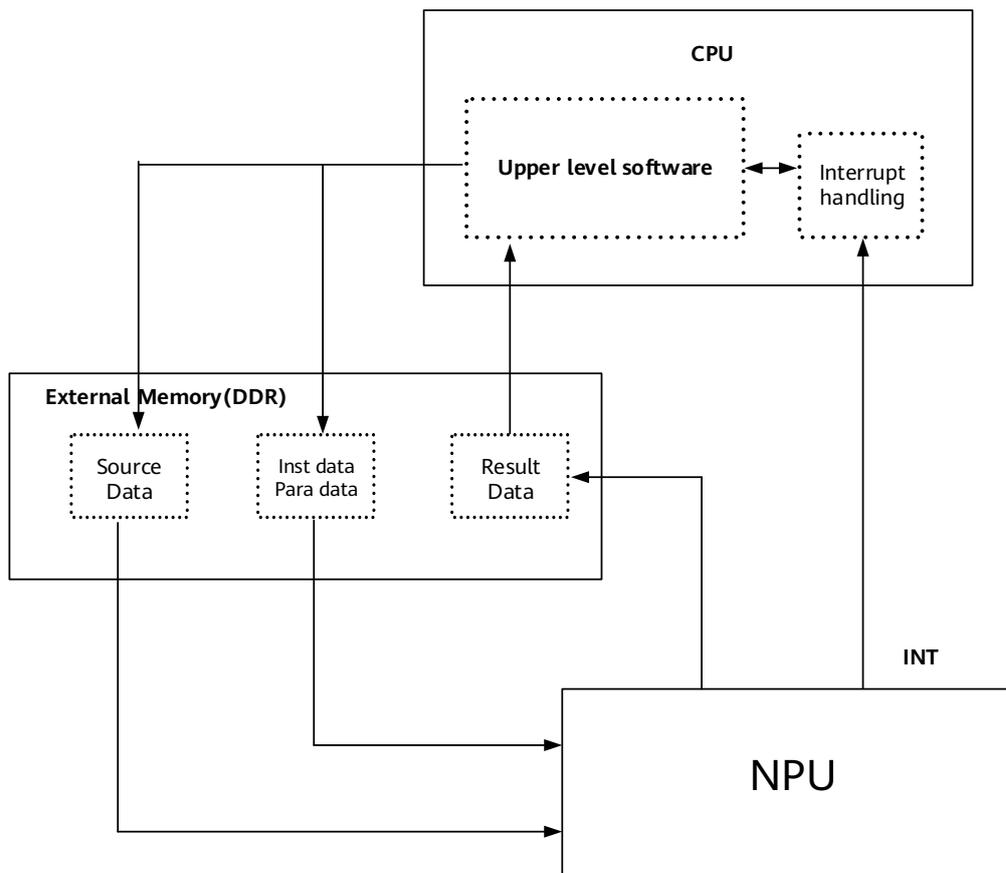
9 智能加速引擎

9.1 NPU 加速引擎

9.1.1 概述

智能加速引擎 NPU(Neural-Network Processing Unit)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。NPU 加速引擎在系统中的位置如[图 9-1](#) 所示。

图9-1 NPU 加速引擎在系统中的位置



9.1.2 特点

NPU 加速引擎特性如下：

- 支持 int4 * int4 下 4Tops 算力
- 支持 int8 * int8 下 2.5Tops 算力
- 支持 int8 * int16 下 1Tops 算力
- 支持 FP16 矢量运算
- 支持 N * M 卷积
- 支持 Pooling (Max 和 Average)
- 支持 Stride
- 支持 Pad
- 支持激活函数 (Relu、Sigmoid 和 TanH 等)
- 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 4bit/8bit/16bit 的数据与参数模式
- 支持 4bit/8bit 的参数
- 支持 featuremap 压缩
- 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
- 支持图像批处理
- 支持中间层结果上报

9.2 DPU

9.2.1 概述

DPU(Depth Process Unit)对输入的左图像和右图像经过校正和匹配计算得出深度图。

9.2.2 特点

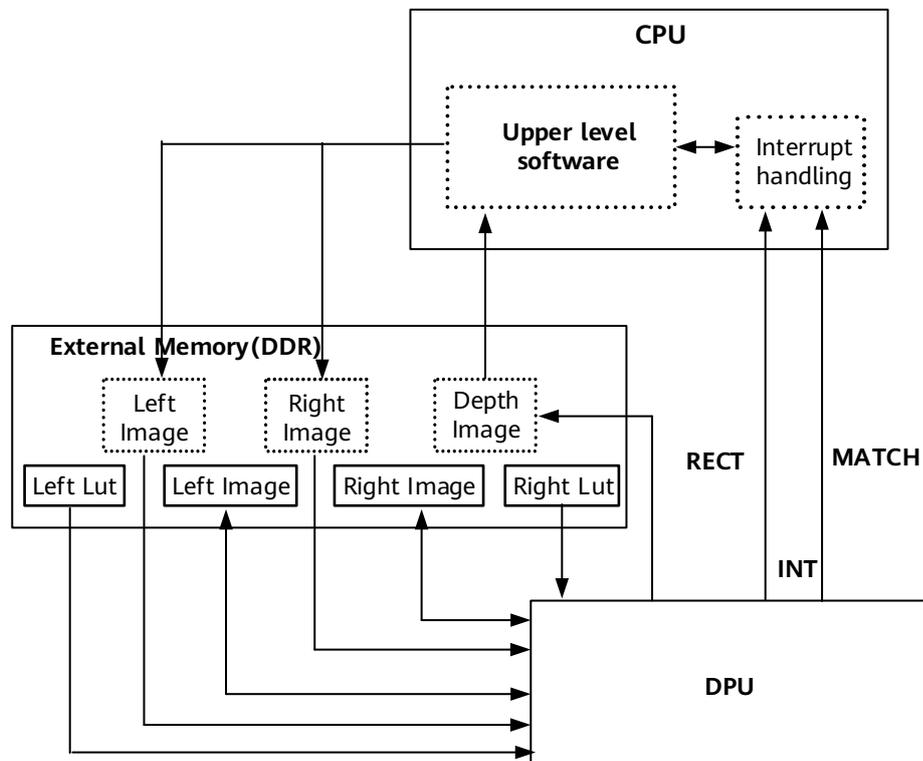
DPU 有如下主要规格点：

- 支持校正和匹配（支持单独或同时使用）；
- 支持最大分辨率 2048x2048；
- 支持最大搜索视差数目 224；
- 支持起始视差可配置，范围：[-224, 224]；
- 只支持 Y 单分量输入；
- 支持亚像素深度图输出；
- 只支持以右图为参考图像，左图作为搜索图像；
- 匹配支持左图宽度大于右图宽度；
- 校正支持左图和右图分辨率不同，输入输出分辨率不同。

9.2.3 功能描述

DPU 在系统中的位置如图 9-2 所示。

图9-2 DPU 在系统中的位置



DPU 的功能简要描述如下：

- 使用校正的功能时，输入的是待校正图和对应的查找表，输出校正后的图，上报的是 RECT 中断。
- 使用匹配的功能时，输入的是校正后的左图和右图，输出的是深度图，上报的是 MATCH 中断。

9.3 IVE

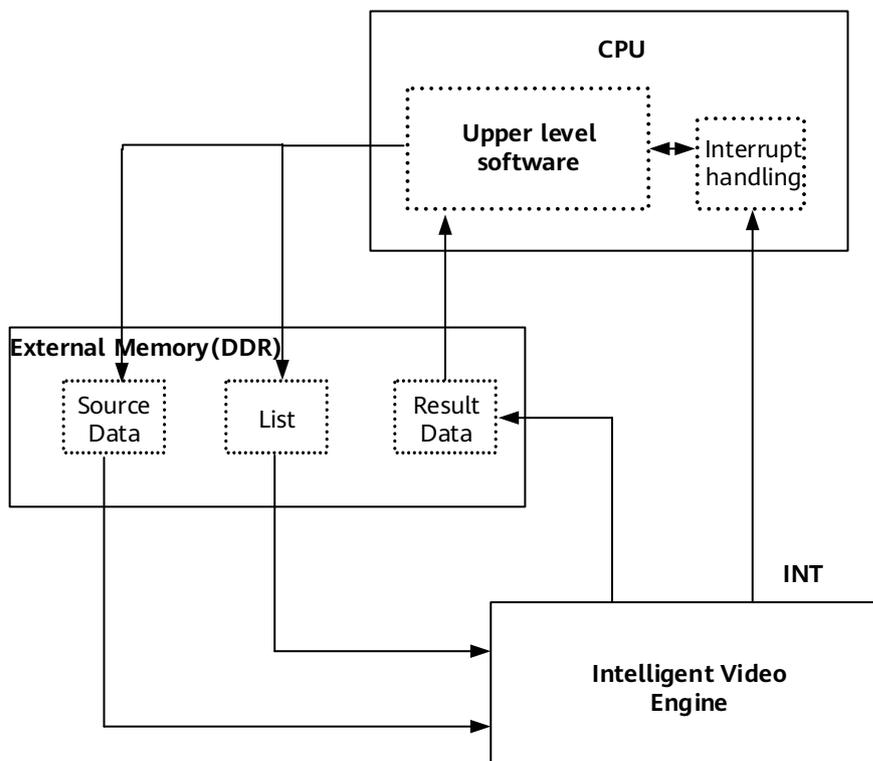
9.3.1 概述

说明

支持 IVE2.4。

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 9-3 所示。

图9-3 IVE 在系统中的位置



9.3.2 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、mask 拷贝。
- Filter: 支持 5x5、7x7、11x11 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。
- Threshold\Threshold_s16\Threshold_u16: 支持图像阈值化处理。



- And\Or\Xor: 支持两幅图像相与、或、异或。
- Add\Sub: 支持两幅图像相加权加、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。
- LKOpticalFlowPyr: 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。



9.3.3 工作方式

9.3.3.1 输入、输出数据格式

图 9-4 ~ 图 9-14 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。9.3.3.2 支持的功能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9-4 SP422 存储格式

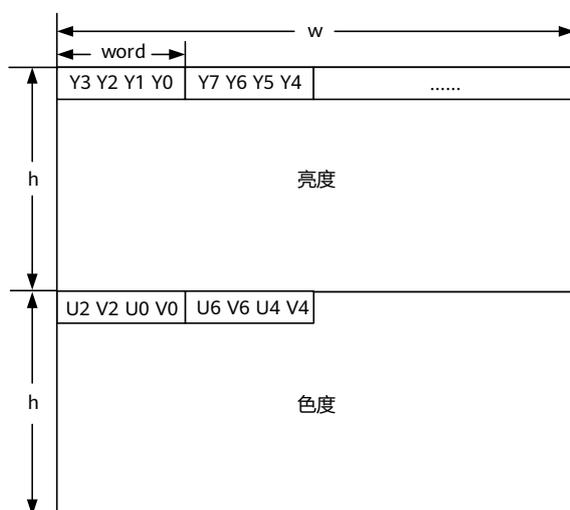


图9-5 SP420 存储格式

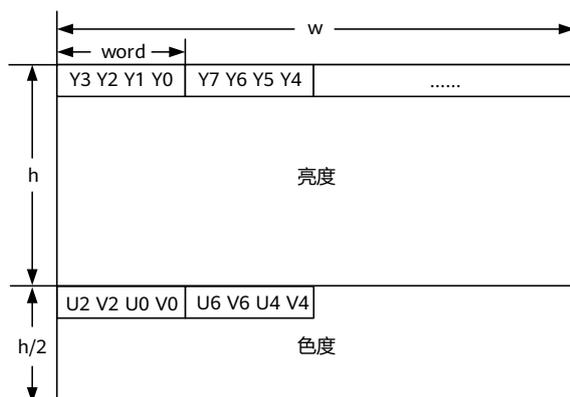




图9-6 8bit 单分量数据在 Memory 中的存储

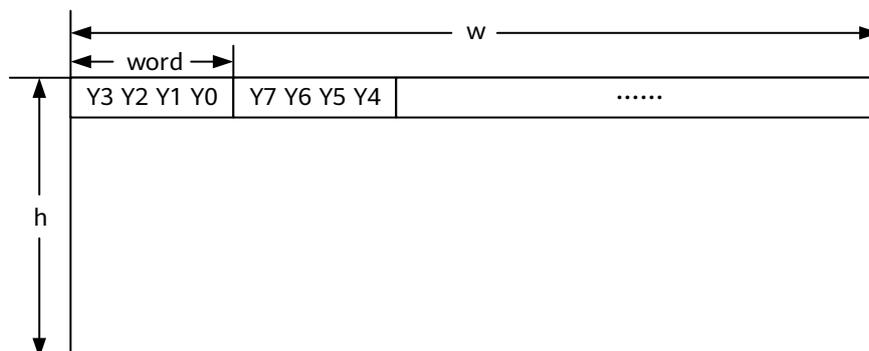


图9-7 Package 存储格式

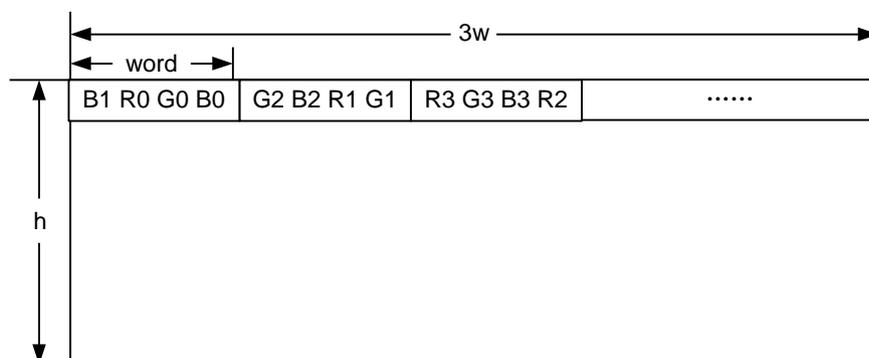




图9-8 Plannar 存储格式

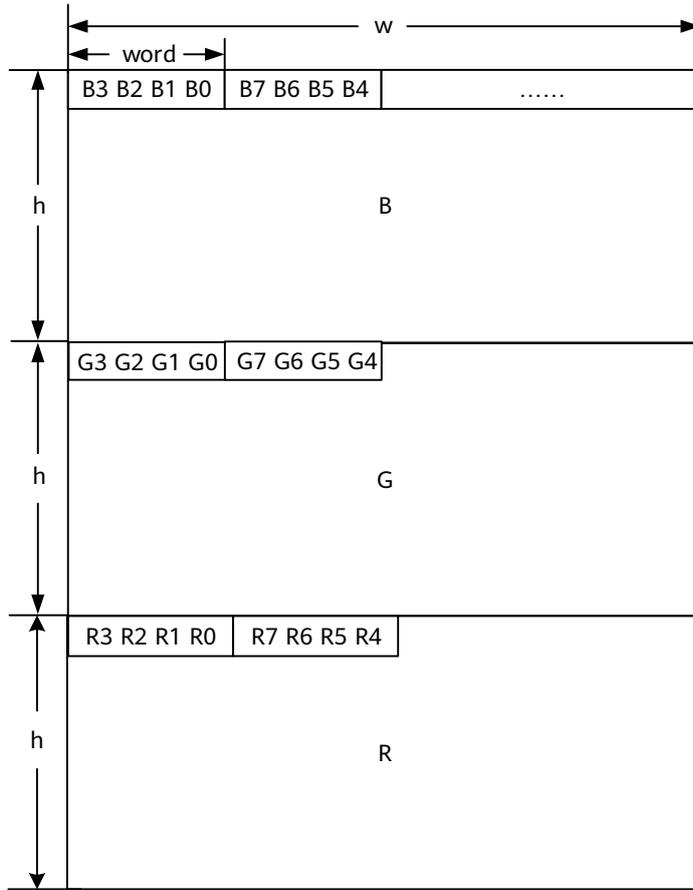


图9-9 16bit 单分量数据在 Memory 中的存储

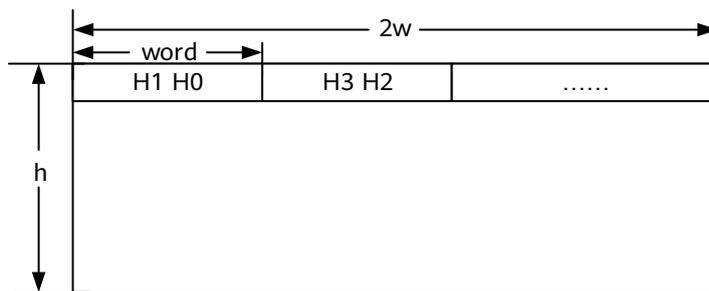




图9-10 32bit 单分量数据在 Memory 中的存储

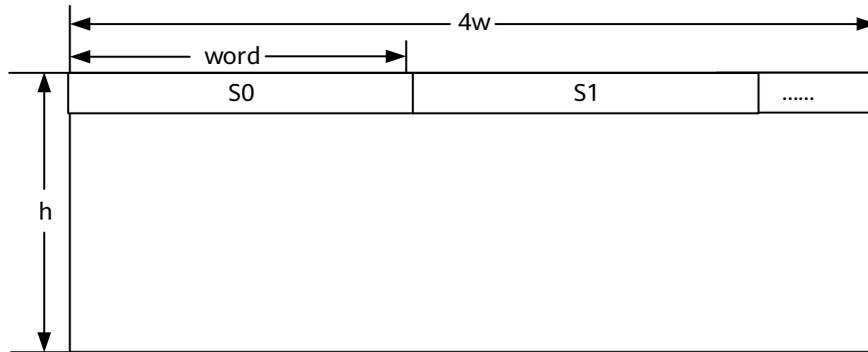


图9-11 64bit 单分量数据在 Memory 中的存储

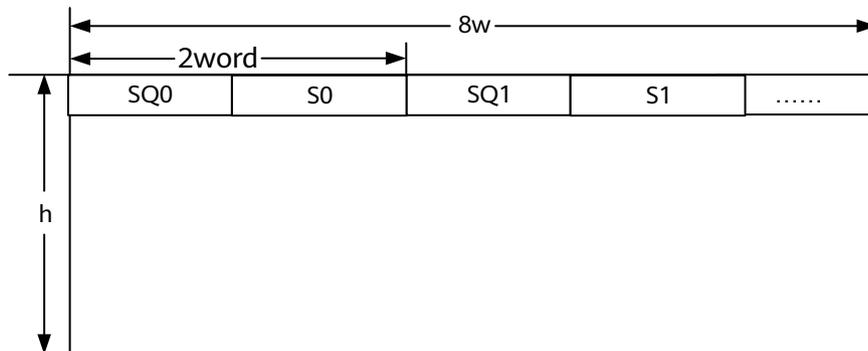


图9-12 NCC 输出数据在 Memory 中的存储

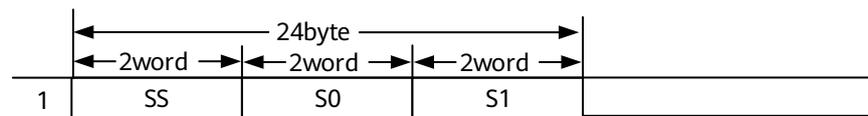


图9-13 CCL 统计信息在 Memory 中的存储 (顺序存放)

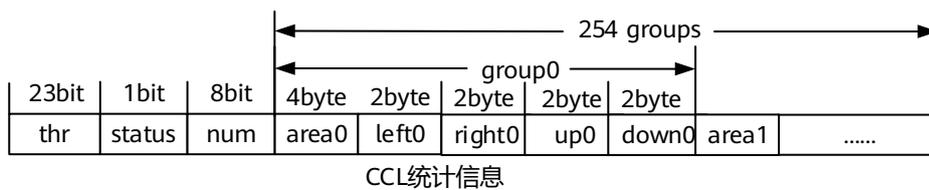




图9-14 16bit 水平和垂直方向的间插数据在 Memory 中的存储

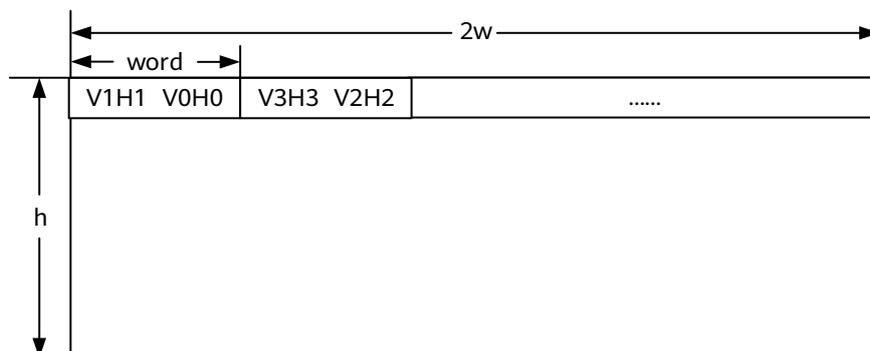


图9-15 HOG 输出格式 1

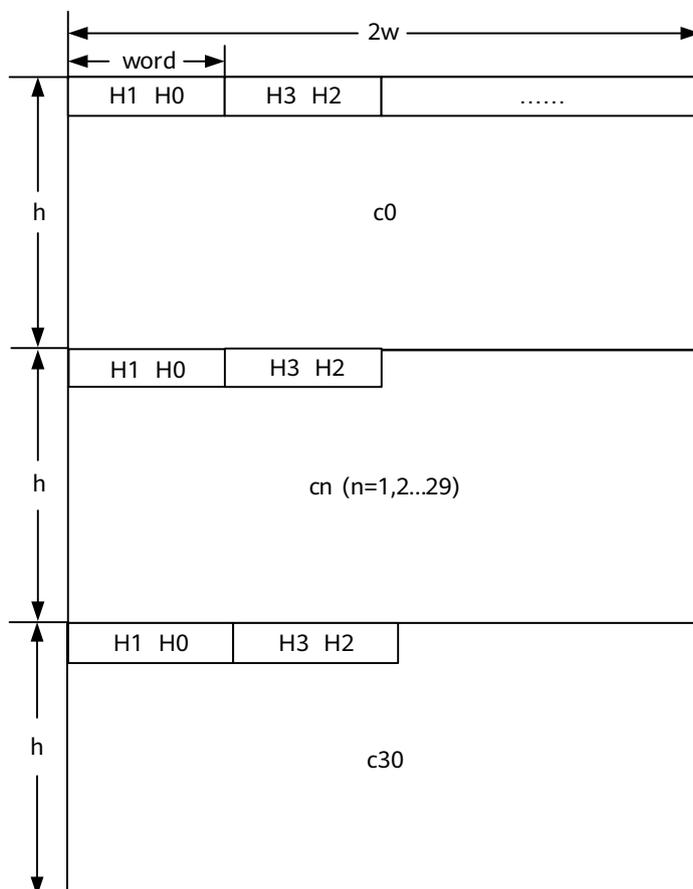
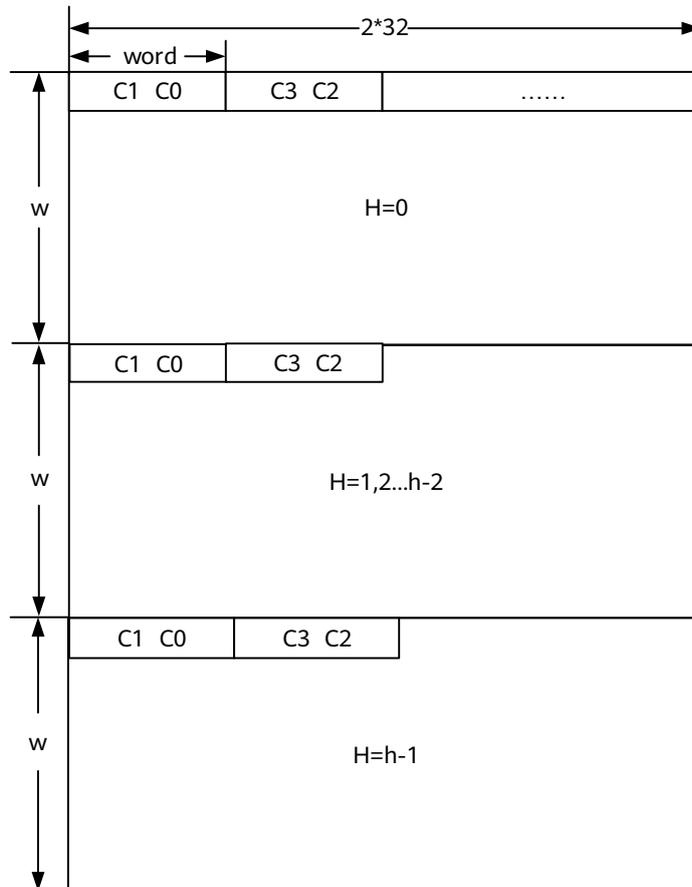


图9-16 HOG 输出格式 2



9.3.3.2 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 9-6 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 9-6 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 9-6 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 9-6 所示。

5. 支持 mask 拷贝模式

实现矩形数据区域的 mask 拷贝功能，支持水平 mask 和垂直 mask 拷贝，均支持 32bit mask 循环拷贝模式和 24bit mask 循环拷贝模式。

- 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：输入和输出都是 8bit 单分量数据，如图 9-6 所示。

Filter

将源图像以 5x5、7x7、11x11 模板滤波后输出。

- 地址对齐方式：输入输出地址要求 byte 对齐（8bit byte 对齐、16bit 2byte 对齐），输入输出 stride 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、16bit 单分量数据，如图 9-6、图 9-9 所示
 - 输入和输出都是 SP420 数据，如图 9-5 所示
 - 输入和输出都是 SP422 数据，如图 9-4 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-->SP422

SP420 或 SP422 格式时，如图 9-5 和图 9-4 所示。

package 格式时，如图 9-7 所示。

plannar 格式时，如图 9-8 所示。

FilterAndCSC

将 YUV SP420/SP422 图像以 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 9-4 和图 9-5 所示；package 格式时，如图 9-7 所示；plannar 格式时，如图 9-8 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 9-6 所示
 - 只输出 H 或 V 时，如图 9-9 所示



- H 和 V 都输出时，如图 9-9 所示

MagAndAng

计算图像梯度的幅值、幅角，支持对幅值图 TO_ZERO threshold 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 9-7 所示
 - 输出为 16bit 单分量幅值图，如图 9-9 所示
 - 输出为 8bit 单分量幅角图，如图 9-6 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 9-6 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 9-6 所示

Threshold

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 9-6 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。

- 图像分辨率：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 9-6 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。

- 图像分辨率：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 9-6 所示

Or

源图 2 数据与源图 1 数据进行或运算，并将结果输出到目的区域。

- 图像分辨率：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 9-6 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，分量平方累加和占高 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 9-6 所示



- 和积分图单独输出时，输入为 32bit 单分量图，如图 9-10 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 9-11 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 9-6 所示
 - 输出统计结果 32bit 单分量数据，如图 9-10 所示

Threshold_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 9-9
 - 输出转换后的 8bit 单分量数据，如图 9-6

Threshold_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 9-9 所示
 - 输出转换后的 8bit 单分量数据，如图 9-6 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 9-9 所示
 - 输出转换后的 8bit 数据，如图 9-6 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 9-6 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 9-6 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 9-6 和图 9-9 所示

Add

两幅灰度图像的加权和，可分别配置各自的权重。

- 图像分辨率：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 9-6 所示
 - 输出相加后的 8bit 单分量图像数据。如图 9-6 所示

Xor

两幅二值图像对应数据相异或。

- 图像分辨率：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 9-6 所示
 - 输出异或后的 8bit 单分量图像数据，如图 9-6 所示



NCC

求取两相同分辨率灰度图像的归一化互相关系数。

- 图像分辨率：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 9-6 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 9-12 所示

CCL

二值图的 8-连通/4-连通区域标记。

- 图像分辨率：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 9-6 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 9-13 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 9-6 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 9-6 所示
 - 输入 16bit 单分量幅值图，如图 9-9 所示
 - 输出 8bit 单分量边缘标记图，如图 9-6 所示
 - 输出 32bit 单分量数据表示的栈，如图 9-10 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 9-6 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 9-6 所示
 - 当 out_fmt 为 0x00、0x01 或 0x02 时，输出为 8bit 单分量数据，如图 9-6 所示；当 out_fmt 为 0x03 时，输出为 16bit H 和 V 间插数据，如图 9-14 所示

LKOpticalFlowPyr

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 9-6 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 9-6 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 9-14 所示
 - 输出 16bit 单分量数据，如图 9-9 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 9-6 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：所有输入输出地址与 stride 都要求 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 9-6 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 9-9 所示，dst2 为二值化图，8bit 单分量，如图 9-6 所示。
 - b. 当 out_fmt 配置为 0x01 时，dst1 为 sad 值，8bit 单分量，如图 9-6 所示，dst2 为二值化图，8bit 单分量，如图 9-6 所示。
 - c. 当 out_fmt 配置为 0x02 时，dst1 为 sad 值，16bit 单分量，如图 9-9 所示。
 - d. 当 out_fmt 配置为 0x03 时，dst1 为 sad 值，8bit 单分量，如图 9-6 所示。
 - e. 当 out_fmt 配置为 0x04 时，dst1 为二值化结果，8bit 单分量，如图 9-6 所示。



Resize

支持双线性插值(bilinear)缩放, 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: 输入输出地址要求 byte 对齐 (8bit byte 对齐、16bit 2byte 对齐), 输入输出 stride 要求 16byte 对齐。
- 输入输出数据格式:
 - 输入 8bit、16bit 单分量图像, 如图 9-6 和图 9-9 所示。
 - 输出缩放后的图像, 对应于输入图像格式。
- 缩放模式: 支持双线性缩放模式

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 9-6 和图 9-7 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 9-9 所示
 - 输入模型数据
 - 输出 8bit 单分量前景二值图, 如图 9-6 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 9-6 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: 20x20~1024x1024; 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 9-6](#)，YUV SP420 如[图 9-5](#)，RGB Package 如[图 9-7](#) 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：40x40~1024x1024
- 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 9-15](#)，HOG 输出格式 2 如[图 9-16](#) 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。



目 录

10 视频接口	10-1
10.1 VI	10-1
10.1.1 概述	10-1
10.1.2 特点	10-2
10.1.3 模式功能描述	10-3
10.1.4 工作方式	10-12
10.1.5 VICAP 寄存器概览	10-16
10.1.6 VICAP 寄存器描述	10-24
10.2 VDP	10-104
10.2.1 概述	10-104
10.2.2 架构描述	10-104
10.2.3 工作方式	10-105
10.2.4 功能描述	10-106
10.2.5 VDP 寄存器概览	10-109
10.2.6 VDP 寄存器描述	10-115
10.3 MIPI Rx	10-168
10.3.1 概述	10-168
10.3.2 特点	10-168
10.3.3 功能描述	10-169
10.3.4 MIPI Rx 控制器工作方式	10-184
10.3.5 MIPI Rx 寄存器概览	10-186
10.3.6 MIPI Rx 寄存器描述	10-194
10.4 MIPI Tx	10-272
10.4.1 概述	10-272



10.4.2 特点	10-272
10.4.3 功能描述	10-273
10.4.4 配置流程	10-282
10.4.5 MIPI Tx 寄存器概览.....	10-283
10.4.6 MIPI Tx 寄存器描述.....	10-286
10.4.7 MIPI Tx PHY 寄存器概览.....	10-324
10.4.8 MIPI Tx PHY 寄存器描述.....	10-325
10.5 热成像接口	10-329
10.5.1 概述	10-329
10.5.2 特点	10-329
10.5.3 功能描述	10-330
10.5.4 工作方式	10-331
10.5.5 热成像控制器寄存器概览.....	10-331
10.5.6 热成像控制器寄存器描述.....	10-332



插图目录

图 10-1 VI 功能框图	10-1
图 10-2 高清接口输入时序水平时序	10-4
图 10-3 高清接口输入时序垂直时序	10-4
图 10-4 720 行视频系统垂直时序	10-6
图 10-5 ITU-R BT.601 水平时序图	10-6
图 10-6 NTSC 制式垂直同步时序图	10-7
图 10-7 PAL 制式垂直同步时序图	10-8
图 10-8 数字摄像头水平时序	10-8
图 10-9 数字摄像头垂直时序脉冲方式	10-9
图 10-10 数字摄像头垂直时序行有效方式	10-9
图 10-11 有效图像区域与水平垂直消隐关系图	10-10
图 10-12 YUV4:2:2 的存储模式	10-11
图 10-13 RAW DATA 的存储模式	10-12
图 10-14 little endian 图像存储模式	10-12
图 10-15 VICAP 的硬件工作流程	10-13
图 10-16 软件操作流程	10-14
图 10-17 VIPROC 离线硬件工作流程	10-15
图 10-18 VIPROC 离线模式软件工作流程	10-15
图 10-19 VDP 总体框图	10-104
图 10-20 MIPI Rx 功能框图及在系统中的位置	10-168



图 10-21 CSI-2 数据包的传输机制.....	10-172
图 10-22 CSI-2 长包的格式.....	10-173
图 10-23 CSI-2 短包的格式.....	10-173
图 10-24 MIPI 接口图像格式.....	10-174
图 10-25 MIPI 接口宽动态数据传输 (使用 VC)	10-175
图 10-26 MIPI 接口宽动态数据传输 (使用 DT)	10-175
图 10-27 MIPI DOL 数据传输.....	10-176
图 10-28 LVDS 同步码和图像传输模式.....	10-178
图 10-29 LVDS 单个像素点时序.....	10-179
图 10-30 LVDS 同步方式 1.....	10-179
图 10-31 LVDS 同步方式 2.....	10-180
图 10-32 LVDS 宽动态模式 1 (2 帧)	10-181
图 10-33 LVDS 宽动态模式 2 (2 帧)	10-181
图 10-34 HiSPi Packetized-SP 数据格式	10-182
图 10-35 HiSPi 宽动态模式.....	10-184
图 10-36 MIPI Tx 功能框图及在系统中的位置.....	10-272
图 10-37 lane id 配置方法举例, 左图{lane0_id, lane1_id, lane2_id, lane3_id}={2,3,0,1},右图为{0,1,2,3}	10-274
图 10-38 RGB 16-bit 格式.....	10-274
图 10-39 RGB 18-bit non-loosely 格式.....	10-275
图 10-40 RGB 18-bit loosely 格式.....	10-275
图 10-41 RGB 24-bit 格式.....	10-276
图 10-42 YUV4:2:0 12-bit 格式.....	10-276
图 10-43 YUV4:2:2 16-bit 格式.....	10-276
图 10-44 YUV4:2:2 20-bit 格式.....	10-277
图 10-45 Legacy YUV420 8-bit 格式.....	10-277
图 10-46 YUV420 8-bit 格式.....	10-278



图 10-47 YUV422 8-bit 格式.....	10-278
图 10-48 CSI RAW8 格式.....	10-279
图 10-49 CSI RAW10 格式.....	10-279
图 10-50 CSI RAW12 格式.....	10-279
图 10-51 CSI RAW14 格式.....	10-279
图 10-52 CSI RAW16 格式.....	10-279
图 10-53 CSI RGB888 格式.....	10-280
图 10-54 Burst Mode.....	10-280
图 10-55 Non-Burst Mode with Sync Pulses.....	10-281
图 10-56 Non-Burst Mode with Sync Events.....	10-281
图 10-57 CSI-2 帧时序.....	10-282
图 10-58 热成像接口功能框图及在系统中的位置.....	10-329
图 10-59 SmartData 配置数据流.....	10-330
图 10-60 OOC 校正数据流.....	10-330
图 10-61 热成像时序解析.....	10-331



表格目录

表 10-1 ITU-R BT.656 YUV4:2:2 行数据格式.....	10-5
表 10-2 SAV/EAV 格式	10-5
表 10-3 VICAP 寄存器地址范围及分组.....	10-16
表 10-4 VICAP 寄存器概览 (基址是 0x0_1740_0000)	10-16
表 10-5 VDP 寄存器概览 (基址是 0x0_17A0_0000)	10-109
表 10-6 MIPI Rx 对接场景分类.....	10-169
表 10-7 MIPI Rx 支持的接口类型.....	10-171
表 10-8 LVDS 同步码格式.....	10-177
表 10-9 LVDS 同步码第 4 个字段举例	10-178
表 10-10 HiSPi 传输模式	10-183
表 10-11 MIPI Rx 寄存器偏移地址变量表.....	10-186
表 10-12 MIPI Rx 寄存器概览 (基址是 0x0_173C_0000)	10-186
表 10-13 MIPI Tx 寄存器概览 (基址是 0x0_1027_0000)	10-283
表 10-14 MIPI Tx PHY 寄存器概览 (基址是 0x0_1027_2000)	10-324
表 10-15 热成像控制器寄存器概览 (基址是 0x0_173D_0000)	10-331



10 视频接口

10.1 VI

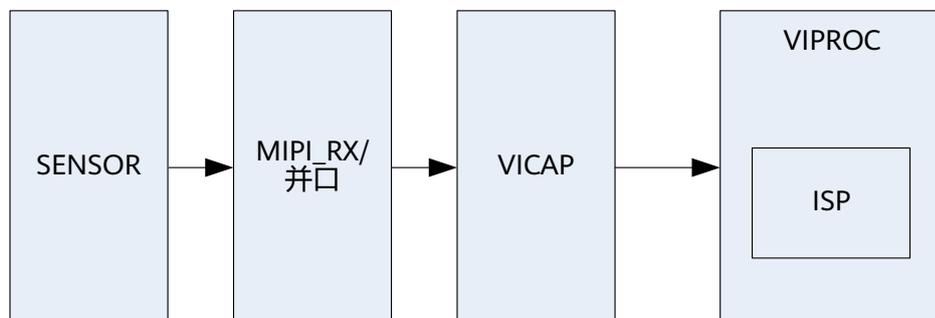
10.1.1 概述

视频输入单元 VI (Video Input), 可以通过 MIPI Rx (包含 MIPI、Sub-LVDS、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 10-1 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10-1 VI 功能框图





10.1.2 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。

- 输入最大宽度为 8192，最大分辨率为 8192 x 8192。
- 最大支持 4 路 sensor 数据输入。
 - 接口 0 最大支持 4096 x2160@30fps 的 2F-WDR RAW 输入
 - 接口 1/2/3 最大支持 4096x2160@30fps 的 MIPI RAW 时序输入
- 输入支持最大数据位宽为 14bit。
- 支持逐行输入模式。
- 支持 BT.656 (2mux)、BT.601、BT.1120 和 DC 接口。
最高支持 148.5MHz 单沿采样
- 支持 MIPI、Sub-LVDS、LVDS、HiSPi 接口。
- 支持 MIPI 接口输入 YUV 格式
 - YUV422 semi-panner、YUV420 semi-panner 格式
 - 支持 embed data 采集
- 支持 flash trigger
- 支持 shutter trigger
- 支持 sensor 主模式和从模式
- 支持主流热成像 sensor 时序
- 支持 RAW 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 7 个写出通道
 - CH0(channel0)~CH3 支持 RAW 格式写出
 - CH3~CH6 支持 YUV 格式写出
 - CH0~CH3 (RAW) 最大宽高为 8192x8192
 - CH3~CH6 (YUV) 最大宽高为 4096x4096

注：VCDS 最大支持宽度为 1920



- 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

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 4096 x 2160@30fps（WDR 合成后）
- 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

10.1.3 模式功能描述

10.1.3.1 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 4 路视频输入信号。

VI 的典型输入有以下 3 种：

- 1 路 4096x2160@30fps 2F-WDR 输入
- 2 路 3072x1728@30fps 2F-WDR 输入
- 1 路 5M(3072x1728)@30fps RAW 输入、1 路 2M(1920x1080)@30fps RAW 输入和 3 路 1M(1280x720)@30fps YUV 输入



10.1.3.2 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 10-2 和图 10-3 所示。

图10-2 高清接口输入时序水平时序

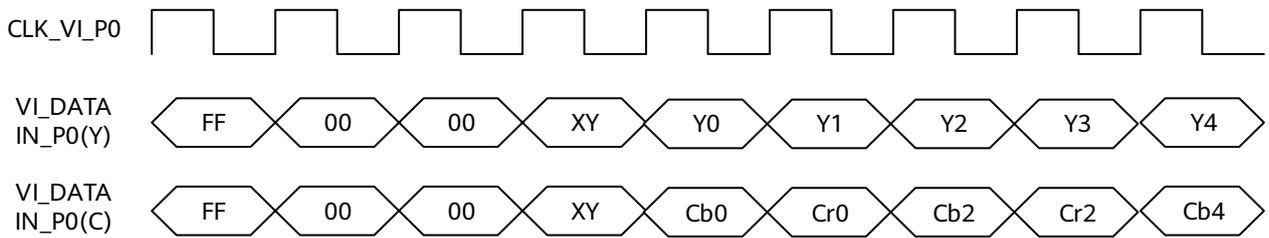
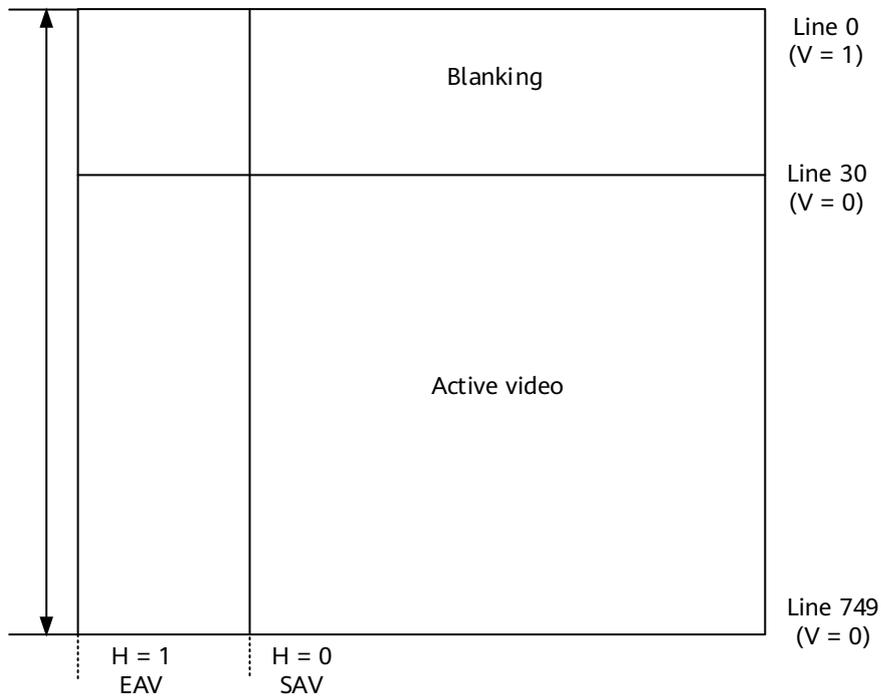


图10-3 高清接口输入时序垂直时序





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 10-1 所示。

表10-1 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	0	EA	8	1	...	8	1	FF	0	0	SAV	U0	Y0	V0	Y1	...	V71	Y71
		0	V	0	0		0	0		0	0							8	9

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 10-2 所示。

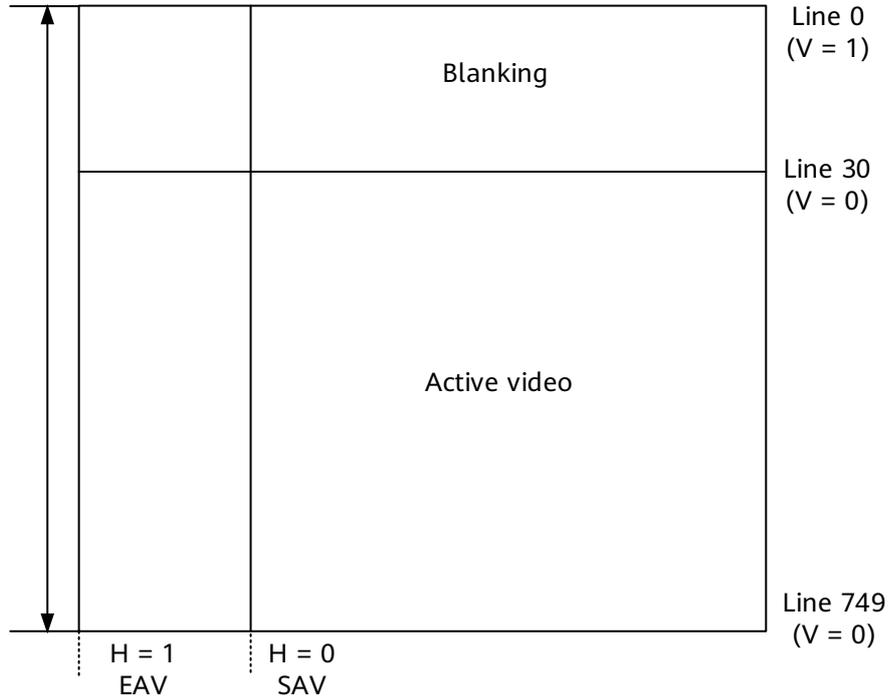
表10-2 SAV/EAV 格式

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 10-4 所示。

图10-4 720 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。

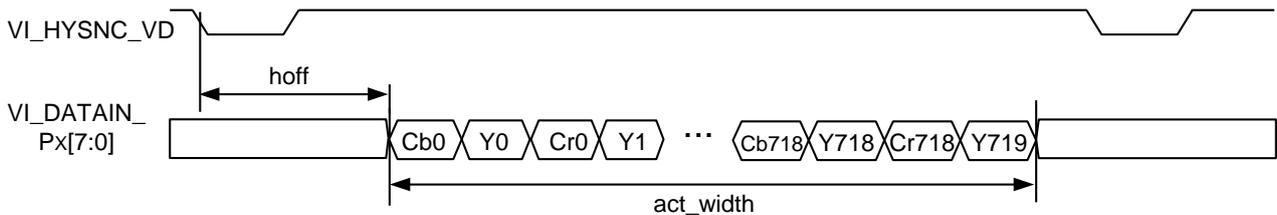
BT.601 接口时序

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 10-5 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10-5 ITU-R BT.601 水平时序图



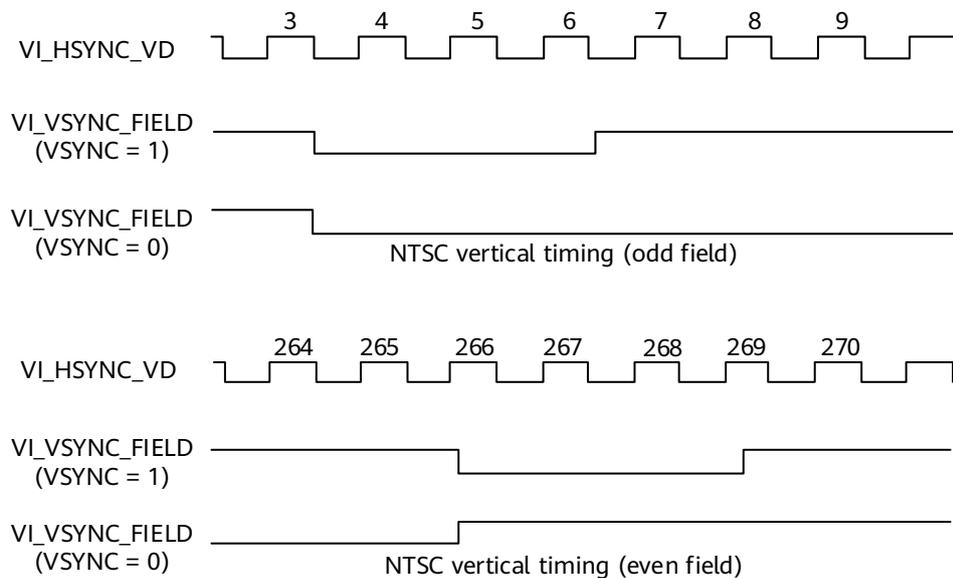


- 垂直时序

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。

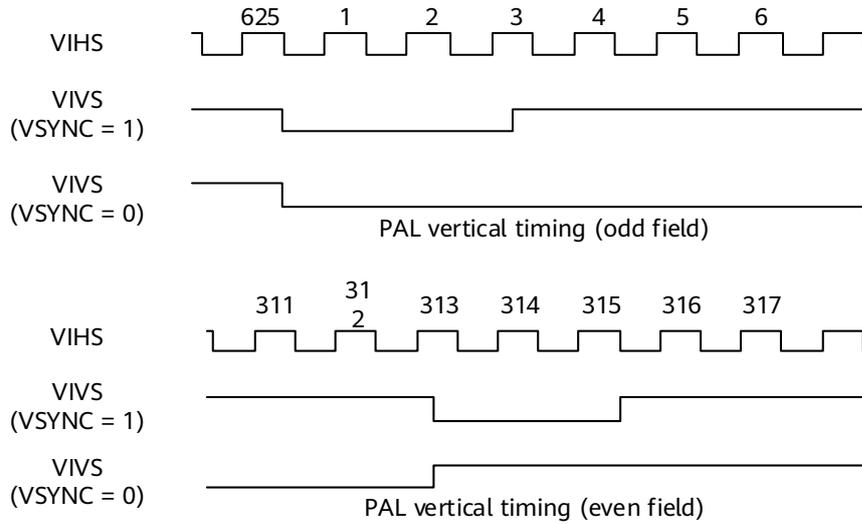
VICAP 在 NTSC 制式（525 行）和 PAL 制式（625 行）的垂直时序关系图如图 10-6 和图 10-7 所示，其中 VI_HSYNC_VD 为水平同步脉冲，VI_VSYNC_FIELD 在 VSYNC=1 时为垂直同步脉冲；在 VSYNC=0 时为场同步信号。

图10-6 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10-7 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。

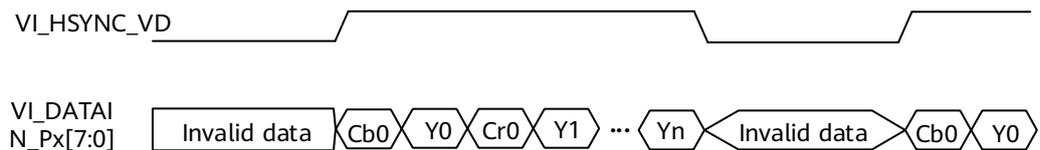
上面两种为典型的 BT.601 垂直时序，VICAP 内部支持场开始到场有效行开始之间的行数可配，场有效行的行数可配，垂直同步的极性可配。

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 10-8 所示。

图10-8 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 10-9 和图 10-10 所示。垂直同步极性可配。

图10-9 数字摄像头垂直时序脉冲方式

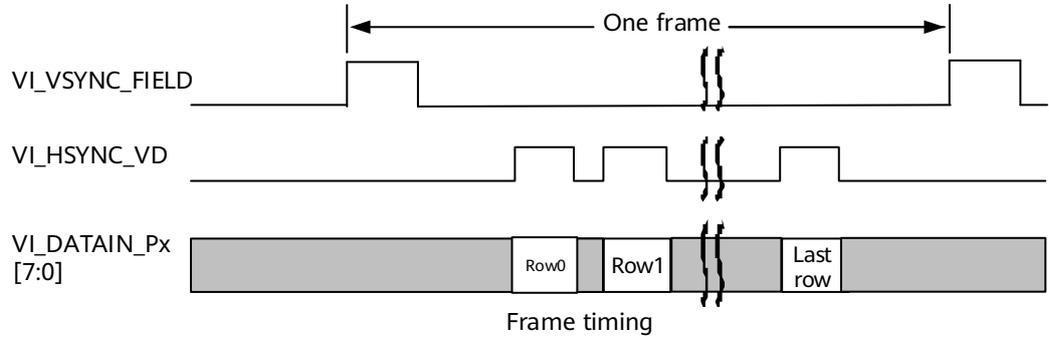
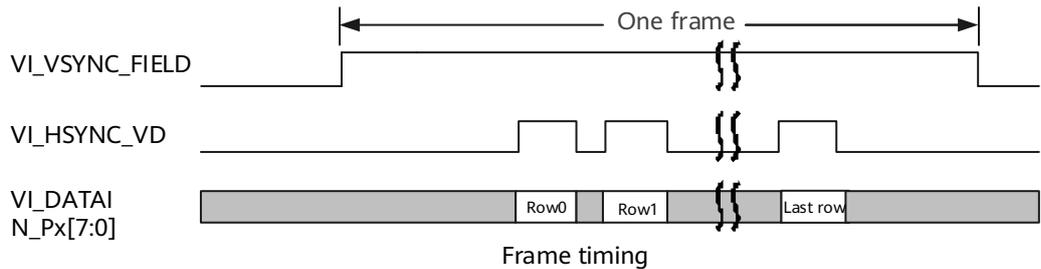


图10-10 数字摄像头垂直时序行有效方式

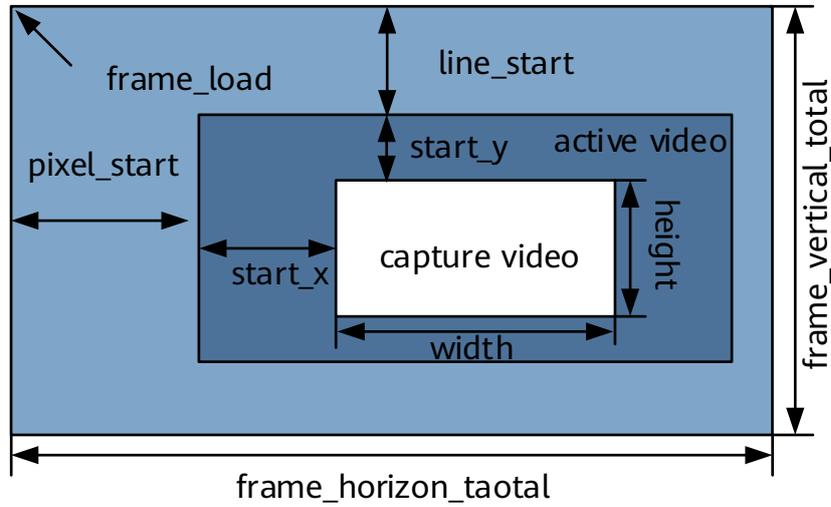


对于 VICAP 内部处理来说，这两种时序相同，VICAP 内部检测到一个上升沿或下降沿，这认为是一帧的开始，然后检测数据有效信号，来判断当前数据是否有效。

10.1.3.3 图像 CROP

有效视频范围如图 10-11 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10-11 有效图像区域与水平垂直消隐关系图



10.1.3.4 图像存储模式

图像存储模式包括：

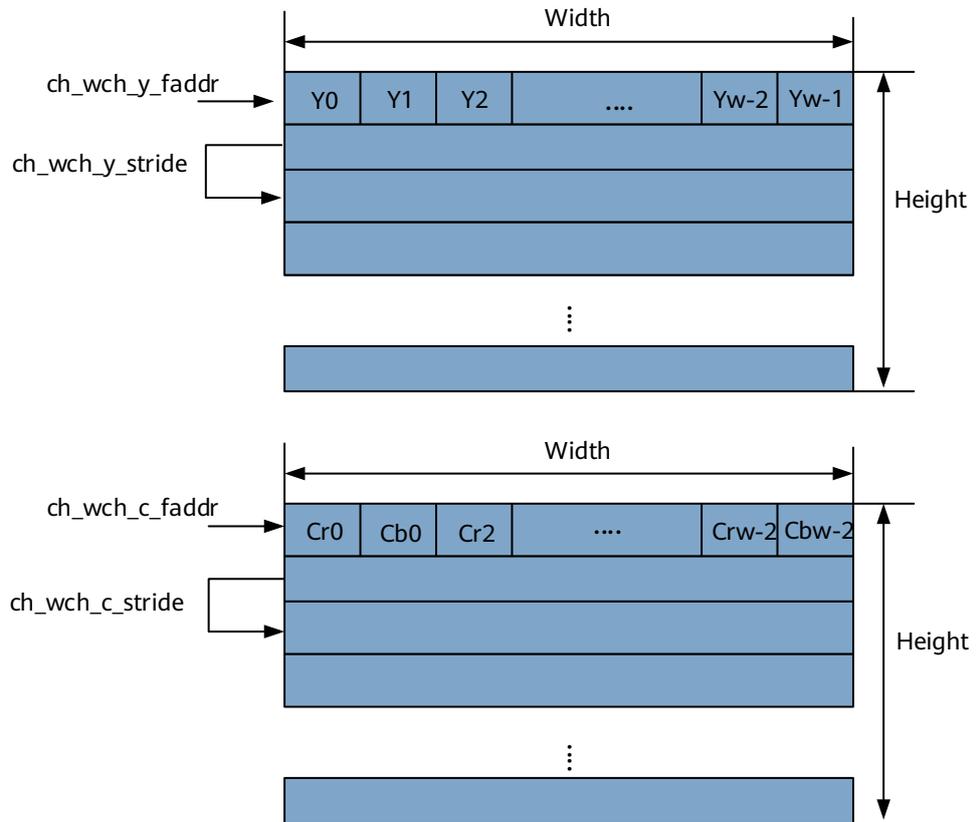
1. YUV 数据存储

- 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 10-12 所示。

图10-12 YUV4:2:2 的存储模式

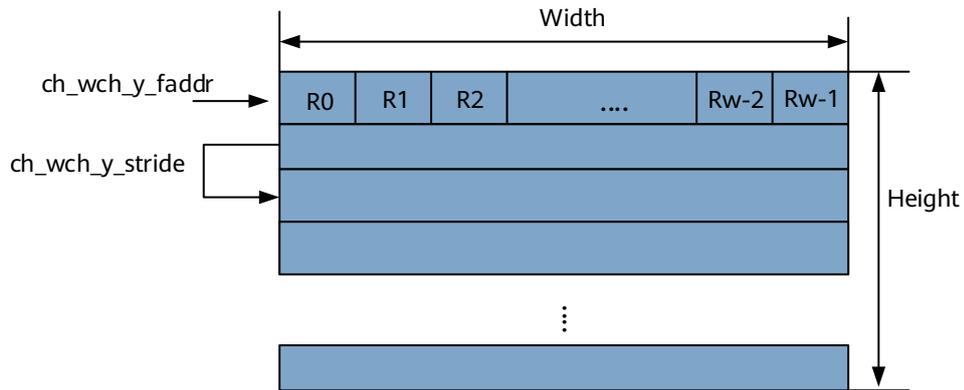


2. RAW 数据存储

- 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。

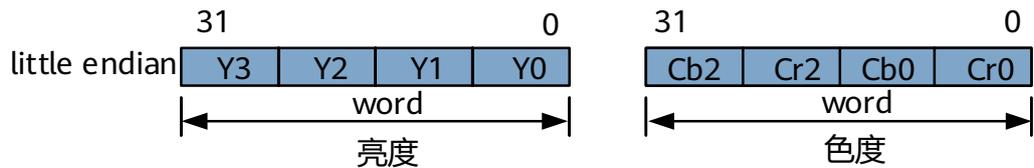
VI 捕获的 RAW 数据的存储结构如图 10-12 所示。

图10-13 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 10-14。

图10-14 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。

10.1.4 工作方式

10.1.4.1 VICAP 的 reg_newer 功能

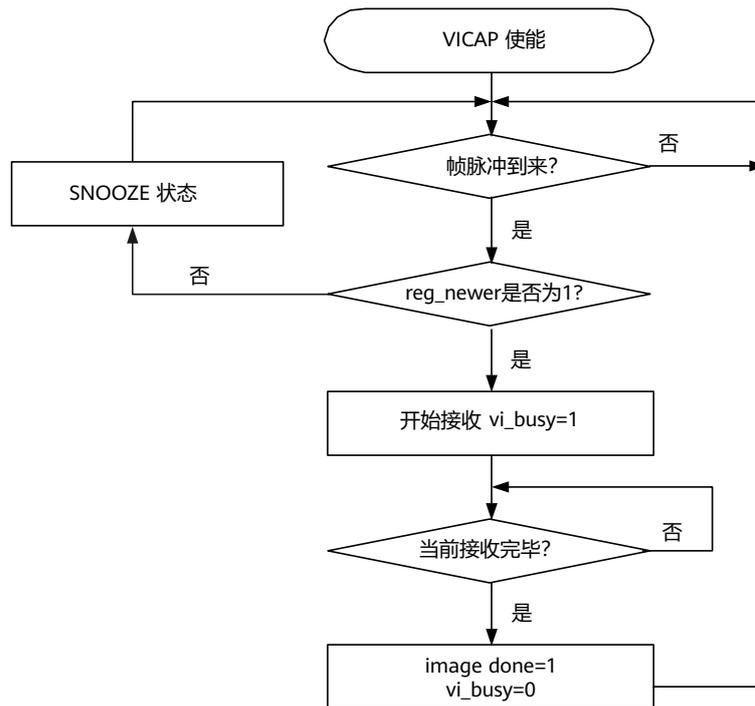
- 在软件使能 VICAP 通道之前，软件应该完成以下操作：
 - 完成对 VICAP 的属性寄存器的配置操作。
 - 写 reg_newer 位为 “1”，通知 VICAP 模块当前的寄存器已经准备就绪。
- 使能 VICAP 后，VICAP 逻辑开始工作，当一帧到来的时候，则有：
 - 如果 reg_newer 为 0，则 VICAP 将不会接收数据，置硬件状态为 SNOOZE（以下各硬件状态为图 10-15 所用），等待下一帧的数据的到来。
 - 如果 reg_newer 为 1，则开始接收数据，同时给出寄存器更新中断 (reg_update_int)，并设置硬件状态为 busy。
- 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。

10.1.4.2 在线模式硬件工作流程

VICAP 的硬件工作流程如图 10-15 所示。

图10-15 VICAP 的硬件工作流程



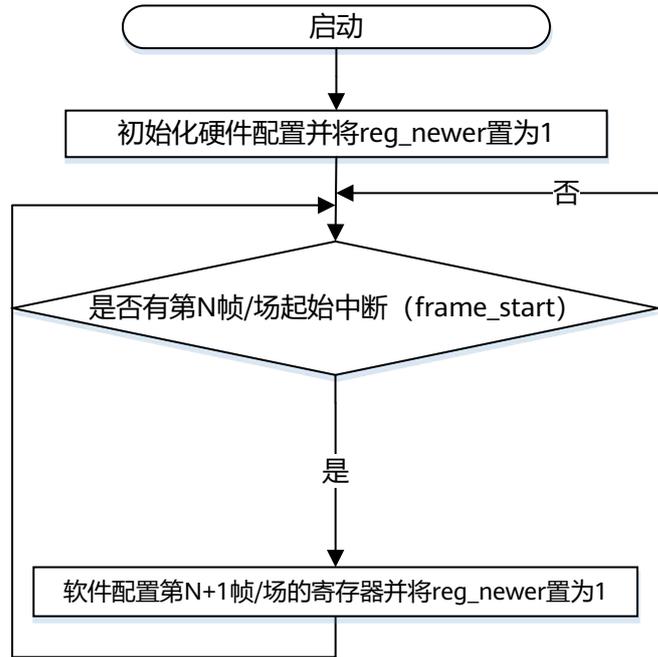
在 VICAP 工作时，每接收完一帧数据，在下一帧的到来时，VICAP 将检测 reg_newer 位。如果 reg_newer 位为 1（表示软件已经更新或者确认 VICAP 的寄存器），VICAP 将自动 load 软件所配置的寄存器值到工作寄存器（工作寄存器软件不可访问），然后将 reg_newer 位清 0，并开始接收下一帧数据。否则，只有等到 reg_newer 为 1 且新的一帧到来时开始接收数据。

10.1.4.3 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 10-16 所示。



图10-16 软件操作流程

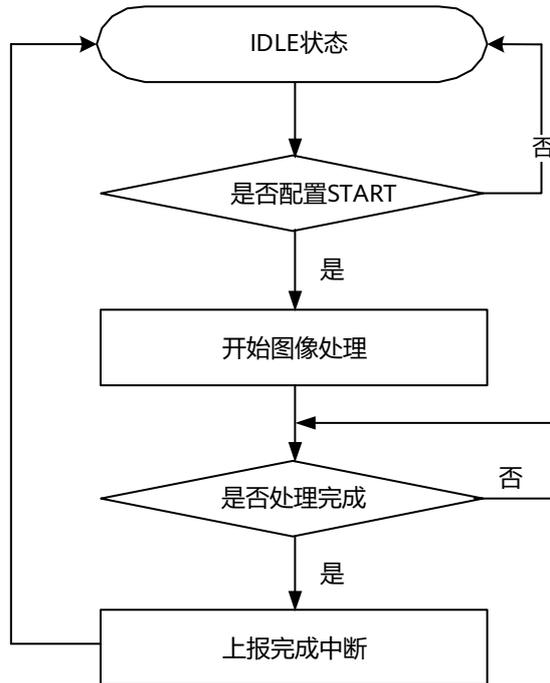


10.1.4.4 VIPROC 离线模式硬件工作流程

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 10-17 所示。



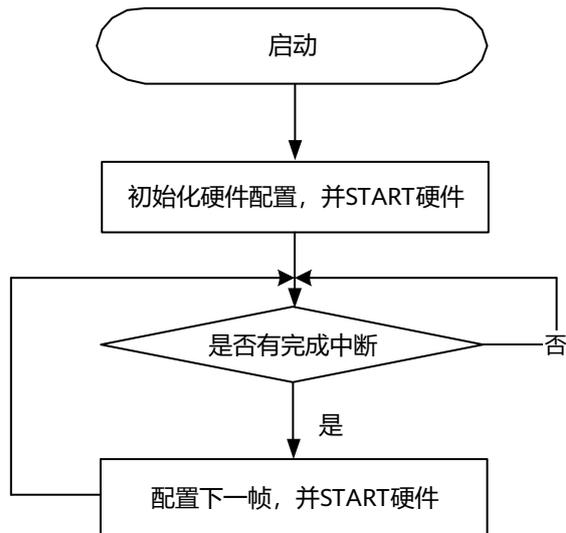
图10-17 VIPROC 离线硬件工作流程



10.1.4.5 VIPROC 离线模式软件配置流程

软件配置流程和如下图 10-18 所示。

图10-18 VIPROC 离线模式软件工作流程





10.1.5 VICAP 寄存器概览

VICAP 寄存器地址范围及分组如表 10-3 所示。

表10-3 VICAP 寄存器地址范围及分组

地址范围	寄存器分组
0x00000-0x000FC	VICAP 全局寄存器
0x00000-0x001FC	VICAP SENSOR 从模式配置寄存器
0x00200-0x002FC	VICAP 在线输出配置寄存器
0x01000-0x010FC	PROT0 寄存器
0x01100-0x011FC	PROT1 寄存器
0x01200-0x011FC	PROT2 寄存器
0x01300-0x013FC	PROT3 寄存器
0x02000-0x21FC	SRC 寄存器
0x04000-0x04FFC	BAS 寄存器
0x10000-0x10FFC	CH0(channel0)寄存器
0x11000-0x11FFC	CH1 寄存器
0x12000-0x12FFC	CH2 寄存器
0x13000-0x13FFC	CH3 寄存器

VICAP 寄存器概览如表 10-4 所示。

表10-4 VICAP 寄存器概览 (基址是 0x0_1740_0000)

偏移地址	名称	描述	页码
0x0000	WK_MODE	全局工作配置寄存器	10-24
0x0010	AXI_CFG	总线配置寄存器	10-25
0x0050	CMUX_SEL	CMUX 通路选择寄存器	10-25



偏移地址	名称	描述	页码
0x0054	CMUX_ID_CFG	CMUX 帧 ID 选择寄存器	10-27
0x0058	ISP_MODE	ISP 模式选择寄存器	10-30
0x0060	AD_SEL	AD 通路选择寄存器	10-31
0x00E0	VICAP_INT1	中断指示寄存器	10-32
0x00E8	VICAP_INT1_MASK	中断 MASK 配置寄存器	10-35
0x00F0	VICAP_INT0	中断指示寄存器	10-37
0x00F8	VICAP_INT0_MASK	中断 MASK 配置寄存器	10-40
0x0100	SLAVE0_MODE_CFG	SENSOR0 从模式配置寄存器	10-42
0x0104	SLAVE0_MODE_VS_TIME_H	SENSOR0 从模式输出 VSYNC 的周期配置寄存器 (高位)	10-43
0x0108	SLAVE0_MODE_VS_TIME_L	SENSOR0 从模式输出 VSYNC 的周期配置寄存器 (低位)	10-44
0x010C	SLAVE0_MODE_HS_TIME	SENSOR0 从模式输出 HSYNC 的周期配置寄存器	10-44
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 (高位)	10-44
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 (低位)	10-44
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	10-45
0x012C	SLAVE0_MODE_HS_DLY_CYC	SENSOR0 从模式输出 HSYNC 脉冲输出相对 VSYNC 脉冲的延迟周期配置	10-45
0x0140	SLAVE1_MODE_CFG	SENSOR1 从模式配置寄存器	10-45
0x0144	SLAVE1_MODE_VS_TIME_H	SENSOR1 从模式输出 VSYNC 的周期配置寄存器 (高位)	10-47



偏移地址	名称	描述	页码
0x0148	SLAVE1_MODE_VS_TIME_L	SENSOR1 从模式输出 VSYNC 的周期配置寄存器 (低位)	10-47
0x014C	SLAVE1_MODE_HS_TIME	SENSOR1 从模式输出 HSYNC 的周期配置寄存器	10-47
0x0158	SLAVE1_MODE_VS_CYC_H	SENSOR1 从模式输出 VSYNC 脉冲有效信号持续时间配置寄存器 (高位)	10-47
0x015C	SLAVE1_MODE_VS_CYC_L	SENSOR1 从模式输出 VSYNC 脉冲有效信号持续时间配置寄存器 (低位)	10-48
0x0160	SLAVE1_MODE_HS_CYC	SENSOR1 从模式输出 HSYNC 脉冲有效信号持续时间配置寄存器	10-48
0x016C	SLAVE1_MODE_HS_DLY_CYC	SENSOR1 从模式输出 HSYNC 脉冲输出相对 VSYNC 脉冲的延迟周期配置	10-48
0x0180	SLAVE2_MODE_CFG	SENSOR2 从模式配置寄存器	10-49
0x0184	SLAVE2_MODE_VS_TIME_H	SENSOR2 从模式输出 VSYNC 的周期配置寄存器 (高位)	10-50
0x0188	SLAVE2_MODE_VS_TIME_L	SENSOR2 从模式输出 VSYNC 的周期配置寄存器 (低位)	10-50
0x018C	SLAVE2_MODE_HS_TIME	SENSOR2 从模式输出 HSYNC 的周期配置寄存器	10-50
0x0198	SLAVE2_MODE_VS_CYC_H	SENSOR2 从模式输出 VSYNC 脉冲有效信号持续时间配置寄存器 (高位)	10-51
0x019C	SLAVE2_MODE_VS_CYC_L	SENSOR2 从模式输出 VSYNC 脉冲有效信号持续时间配置寄存器 (低位)	10-51
0x01A0	SLAVE2_MODE_HS_CYC	SENSOR2 从模式输出 HSYNC 脉冲有效信号持续时间配置寄存器	10-51
0x01AC	SLAVE2_MODE_HS_DLY_CYC	SENSOR2 从模式输出 HSYNC 脉冲输出相对 VSYNC 脉冲的延迟周期配置	10-52



偏移地址	名称	描述	页码
0x01C0	SLAVE3_MODE_CFG	SENSOR3 从模式配置寄存器	10-52
0x01C4	SLAVE3_MODE_VS_TIME_H	SENSOR3 从模式输出 VSYNC 的周期配置寄存器 (高位)	10-53
0x01C8	SLAVE3_MODE_VS_TIME_L	SENSOR3 从模式输出 VSYNC 的周期配置寄存器 (低位)	10-53
0x01CC	SLAVE3_MODE_HS_TIME	SENSOR3 从模式输出 HSYNC 的周期配置寄存器	10-54
0x01D8	SLAVE3_MODE_VS_CYC_H	SENSOR3 从模式输出 VSYNC 脉冲有效信号持续时间配置寄存器 (高位)	10-54
0x01DC	SLAVE3_MODE_VS_CYC_L	SENSOR3 从模式输出 VSYNC 脉冲有效信号持续时间配置寄存器 (低位)	10-54
0x01E0	SLAVE3_MODE_HS_CYC	SENSOR3 从模式输出 HSYNC 脉冲有效信号持续时间配置寄存器	10-55
0x01EC	SLAVE3_MODE_HS_DLY_CYC	SENSOR3 从模式输出 HSYNC 脉冲输出相对 VSYNC 脉冲的延迟周期配置	10-55
0x01F8	SLAVE_MODE_TIME_CFG_SEL	SENSOR 从模式 TIME 配置选择寄存器	10-55
0x01FC	SLAVE_MODE_SYN	SENSOR 同步信号, 上升沿有效	10-56
0x0200	ONLINE0_CFG	在线通道 0 控制寄存器	10-57
0x02AC	ONLINE0_SIZE	在线通道 0 输出有效区大小寄存器	10-57
0x030c	VICAP_AFIFO0_SIZE	AFIFO 大小配置为寄存器 0	10-57
0x031c	VICAP_AFIFO1_SIZE	AFIFO 大小配置为寄存器 1	10-58
0x032c	VICAP_AFIFO2_SIZE	AFIFO 大小配置为寄存器 2	10-58
0x033c	VICAP_AFIFO3_SIZE	AFIFO 大小配置为寄存器 3	10-58
0x0fe0	VICAP_AXI0_RD_ERROR	AXI0 读错误状态寄存器	10-59



偏移地址	名称	描述	页码
0x0fe4	VICAP_AXIO_WR_ER ROR	AXIO 写错误状态寄存器	10-59
0x0ff0	VICAP_DBG_INT	系统异常中断状态寄存器	10-60
0x0ff8	VICAP_DBG_INT_M ASK	系统异常状态屏蔽寄存器	10-61
0x1000	PT_INTF_MOD	接口模式寄存器	10-61
0x1010	PT_OFFSET0	0 分量 offset 寄存器	10-62
0x1014	PT_OFFSET1	1 分量 offset 寄存器	10-62
0x1018	PT_OFFSET2	2 分量 offset 寄存器	10-63
0x1020	PT_BT656_CFG	BT.656 配置寄存器	10-63
0x1038	PT_UNIFY_TIMING_ CFG	时序配置寄存器	10-65
0x103C	PT_GEN_TIMING_C FG	时序恢复模块配置寄存器	10-67
0x1040	PT_UNIFY_DATA_C FG	数据配置寄存器	10-68
0x1044	PT_GEN_DATA_CFG	数据产生模块配置寄存器	10-69
0x1048	PT_GEN_DATA_CO EF	数据产生模块系数寄存器	10-70
0x104C	PT_GEN_DATA_INIT	数据产生初始值配置寄存器	10-71
0x1050	PT_YUV444_CFG	YUV444 配置寄存器	10-71
0x1060	PT_FSTART_DLY	端口 fstart 中断延时寄存器	10-71
0x1064	PT_FSTART_H_DLY	端口 fstart 中断行延时寄存器	10-72
0x1080	PT_INTF_HFB	水平前消隐区宽度寄存器	10-72
0x1084	PT_INTF_HACT	水平有效区宽度寄存器	10-72
0x1088	PT_INTF_HBB	水平后消隐区宽度寄存器	10-73
0x108C	PT_INTF_VFB	垂直前消隐区宽度寄存器	10-73



偏移地址	名称	描述	页码
0x1090	PT_INTF_VACT	垂直有效区宽度寄存器	10-73
0x1094	PT_INTF_VBB	垂直后消隐区宽度寄存器	10-73
0x1098	PT_INTF_VBFB	垂直底场前消隐区宽度寄存器	10-74
0x109C	PT_INTF_VBACT	垂直底场有效区宽度寄存器	10-74
0x10A0	PT_INTF_VBBB	垂直底场后消隐区宽度寄存器	10-74
0x10A4	PT_ID_CFG	ID 配置寄存器	10-75
0x10C0	PT_FLASH_CFG	闪光灯配置寄存器	10-75
0x10C4	PT_FLASH_CYC0	闪光灯时序 0 宽度寄存器	10-76
0x10C8	PT_FLASH_CYC1	闪光灯时序 1 宽度寄存器	10-77
0x10D0	PT_SHUTTER_CYC0	快门时序 0 宽度寄存器	10-77
0x10D4	PT_SHUTTER_CYC1	快门时序 1 宽度寄存器	10-77
0x10D8	PT_SHUTTER_CYC2	快门时序 2 宽度寄存器	10-77
0x10DC	PT_SHUTTER_CYC3	快门时序 3 宽度寄存器	10-78
0x10E0	PT_STATUS	端口状态寄存器	10-78
0x10E4	PT_BT656_STATUS	BT.656 状态寄存器	10-78
0x10EC	PT_SIZE	输入图像有效区大小指示寄存器	10-79
0x10F0	PT_INT	端口中断指示寄存器	10-79
0x10F4	PT_INT_STATE	端口中断状态寄存器	10-80
0x10F8	PT_INT_MASK	端口中断屏蔽寄存器	10-80
0x1800	PT3_MUX_MOD	接口 3 复合模式寄存器	10-81
0x1810	PT3_MUX_OFFSET	接口 offset 寄存器	10-82
0x1850	PT3_ID_STATUS	接口 ID 状态寄存器	10-83
0x10000	CH_CTRL	通道控制寄存器	10-83
0x10004	CH_REG_NEWER	采集控制寄存器	10-84



偏移地址	名称	描述	页码
0x10034	CH_DLY_CFG	通道输入图像起始中断延迟配置寄存器	10-85
0x10038	CH_FRM_CNT	延时中断锁存帧号寄存器	10-85
0x10040	CH_CROP_CFG	通道 CROP 使能寄存器	10-85
0x10044	CH_CROP0_Y_CFG	通道垂直 CROP 配置寄存器	10-86
0x10048	CH_CROP0_X0_CFG	通道 0 区域水平 CROP 配置寄存器	10-86
0x1004C	CH_CROP0_X1_CFG	通道 1 区域水平 CROP 配置寄存器	10-86
0x10050	CH_SKIP_Y_CFG	通道 Y SKIP 配置寄存器	10-87
0x10058	CH_SKIP_C_CFG	通道 C SKIP 配置寄存器	10-87
0x10060	CH_VCDS_CFG	主码流色度垂直下采样配置寄存器	10-87
0x10064	CH_VCDS_COEF	主码流色度垂直下采样系数寄存器	10-88
0x10080	CH_WCH_Y_CFG	Y 分量配置寄存器	10-88
0x10084	CH_WCH_Y_SIZE	Y 分量存储大小寄存器	10-89
0x10090	CH_WCH_Y_FADDR_H	Y 分量存储基地址高 32bit 寄存器	10-90
0x10094	CH_WCH_Y_FADDR_L	Y 分量存储基地址低 32bit 寄存器	10-90
0x100A0	CH_WCH_Y_STRIDE	Y 分量行偏移寄存器	10-90
0x100B0	CH_WCH_C_CFG	C 分量配置寄存器	10-91
0x100B4	CH_WCH_C_SIZE	C 分量存储大小寄存器	10-92
0x100C0	CH_WCH_C_FADDR_H	C 分量存储基地址高 32bit 寄存器	10-92
0x100C4	CH_WCH_C_FADDR_L	C 分量存储基地址低 32bit 寄存器	10-92
0x100C8	CH_WCH_TUNL_H_ADDR_H	低延时信息基地址高 32bit 寄存器	10-93
0x100CC	CH_WCH_TUNL_H_ADDR_L	低延时信息基地址低 32bit 寄存器	10-93



偏移地址	名称	描述	页码
0x100D0	CH_WCH_C_STRIDE	C 分量行偏移寄存器	10-93
0x100D8	CH_RCH_PARA_ADDR_H	低延时信息基地址高 32bit 寄存器	10-93
0x100DC	CH_RCH_PARA_ADDR_L	低延时信息基地址低 32bit 寄存器	10-94
0x100E8	CH_Y_ACT_SIZE	亮度通道输出图像有效区宽高指示寄存器	10-94
0x100EC	CH_C_ACT_SIZE	色度通道输出图像有效区宽高指示寄存器	10-94
0x100F0	CH_INT	通道原始中断寄存器	10-95
0x100F4	CH_INT_STATE	通道原始中断状态寄存器	10-96
0x100F8	CH_INT_MASK	通道中断屏蔽寄存器	10-98
0x10204	CH_WCH_STT_AE_HIST_ADDR_H	ISP AE_HIST 统计信息存储基地址高 32bit 寄存器	10-99
0x10208	CH_WCH_STT_AE_HIST_ADDR_L	ISP AE_HIST 统计信息存储基地址低 32bit 寄存器	10-100
0x1020C	CH_WCH_STT_AE_AVER_R_GR_ADDR_H	ISP AE_AVER_R_GR 统计信息存储基地址高 32bit 寄存器	10-100
0x10210	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 寄存器	10-100
0x10214	CH_WCH_STT_AE_AVER_GB_B_ADDR_H	ISP AE_AVER_GB_B 统计信息存储基地址高 32bit 寄存器	10-101
0x10218	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 寄存器	10-101
0x1023C	CH_WCH_STT_AF_STAT_Y_ADDR_H	ISP AF_STAT_Y 统计信息存储基地址高 32bit 寄存器	10-101



偏移地址	名称	描述	页码
0x10240	CH_WCH_STT_AF_STAT_Y_ADDR_L	ISP AF_STAT_Y 统计信息存储基地址低 32bit 寄存器	10-102
0x10280	CH_WCH_EM_CFG	Embedded Data 写出配置寄存器	10-102
0x10284	CH_WCH_EM_SIZE	Embedded Data 信息写出大小寄存器	10-103
0x10288	CH_WCH_EM_FADDR_H	WCH 模块 Y 分量存储基地址高 32bit 寄存器	10-103
0x1028C	CH_WCH_EM_FADDR_L	Embedded Data 存储基地址低 32bit 寄存器	10-103

10.1.6 VICAP 寄存器描述

寄存器描述中的 CH 全称 Channel; SRC 全称 Source; VS 全称 Vsync; HS 全称 Hsync; TUNL 全称 Tunnel; STT 全称 Statistics; EM 全称 Embedded

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]	RO	reserved	保留。	0x00000000
[1]	RW	axi_clk_en	AXI 总线模块时钟开关控制(全在线模式可关闭总线模块时钟)。 0: 总线模块时钟关闭; 1: 总线模式时钟打开。	0x1
[0]	RO	reserved	保留。	0x1



AXI_CFG

AXI_CFG 为总线配置寄存器。

Offset Address: 0x0010 Total Reset Value: 0x0000_0770

Bits	Access	Name	Description	Reset
[31:12]	-	reserved	保留。	0x000000
[11:8]	RW	r_outstanding	读请求 outstanding 个数。 outstanding 的取值范围为[0,15]。	0x7
[7:4]	RW	w_outstanding	写请求 outstanding 个数。 outstanding 的取值范围为[0, 7]。	0x7
[3:0]	-	reserved	保留。	0x0

CMUX_SEL

CMUX_SEL 为 CMUX 通路选择寄存器。

Offset Address: 0x0050 Total Reset Value: 0x0000_3210

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27:24]	RW	chn6_sel	CHN6 选择配置寄存器。 0x0: pt0 输入数据; 0x1: pt1 输入数据; 0x2: pt2 输入数据; 0x3: pt3 输入数据(bt656 复合 id0); 0x4: pt3 输入数据(bt656 复合 id1); 0x5: pt3 输入数据(bt656 复合 id2); 0x6: pt3 输入数据(bt656 复合 id3); 其他: 保留。	0x0
[23:20]	RW	chn5_sel	CHN5 选择配置寄存器。 0x0: pt0 输入数据;	0x0



Bits	Access	Name	Description	Reset
			0x1: pt1 输入数据; 0x2: pt2 输入数据; 0x3: pt3 输入数据(bt656 复合 id0); 0x4: pt3 输入数据(bt656 复合 id1); 0x5: pt3 输入数据(bt656 复合 id2); 0x6: pt3 输入数据(bt656 复合 id3); 其余保留。	
[19:16]	RW	chn4_sel	CHN4 选择配置寄存器。 0x0: pt0 输入数据; 0x1: pt1 输入数据; 0x2: pt2 输入数据; 0x3: pt3 输入数据(bt656 复合 id0); 0x4: pt3 输入数据(bt656 复合 id1); 0x5: pt3 输入数据(bt656 复合 id2); 0x6: pt3 输入数据(bt656 复合 id3); 其他: 保留。	0x0
[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; 0x2: pt2 输入数据; 0x3: pt3 输入数据(bt656 复合 id0); 0x4: pt3 输入数据(bt656 复合 id1); 0x5: pt3 输入数据(bt656 复合 id2); 0x6: pt3 输入数据(bt656 复合 id3); 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;	0x2



Bits	Access	Name	Description	Reset
			0x2: pt2 输入数据; 0x3: pt3 输入数据(bt656 复合 id0); 0x4: pt3 输入数据(bt656 复合 id1); 0x5: pt3 输入数据(bt656 复合 id2); 0x6: pt3 输入数据(bt656 复合 id3); 其他: 保留。	
[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; 0x2: pt2 输入数据; 0x3: pt3 输入数据(bt656 复合 id0); 0x4: pt3 输入数据(bt656 复合 id1); 0x5: pt3 输入数据(bt656 复合 id2); 0x6: pt3 输入数据(bt656 复合 id3); 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; 0x2: pt2 输入数据; 0x3: pt3 输入数据(bt656 复合 id0); 0x4: pt3 输入数据(bt656 复合 id1); 0x5: pt3 输入数据(bt656 复合 id2); 0x6: pt3 输入数据(bt656 复合 id3); 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: 0x0FFF_FFFF



Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27:24]	RW	chn6_id	<p>CHN6 ID 选择寄存器。</p> <p>0: 过滤该 ID 的数据; 1: 接收该 ID 的数据。</p> <p>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。</p> <p>备注: bt656 复合时序, 配 ID 为 0</p>	0xF
[23:20]	RW	chn5_id	<p>CHN5 ID 选择寄存器。</p> <p>0: 过滤该 ID 的数据; 1: 接收该 ID 的数据。</p> <p>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。</p> <p>备注: bt656 复合时序, 配 ID 为 0。</p>	0xF
[19:16]	RW	chn4_id	<p>CHN4 ID 选择寄存器。</p> <p>0: 过滤该 ID 的数据; 1: 接收该 ID 的数据。</p> <p>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。</p> <p>备注: bt656 复合时序, 配 ID 为 0。</p>	0xF
[15:12]	RW	chn3_id	<p>CHN3 ID 选择寄存器。</p> <p>0: 过滤该 ID 的数据; 1: 接收该 ID 的数据。</p>	0xF



Bits	Access	Name	Description	Reset
			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。 备注: bt656 复合时序, 配 ID 为 0	
[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: 0x0F00_0000

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27]	RW	isp3_clk_en	ISP3 门控开关寄存器。 0: 关; 1: 开。	0x1
[26]	RW	isp2_clk_en	ISP2 门控开关寄存器。 0: 关; 1: 开。	0x1
[25]	RW	isp1_clk_en	ISP1 门控开关寄存器。 0: 关; 1: 开。	0x1
[24]	RW	isp0_clk_en	ISP0 门控开关寄存器。 0: 关; 1: 开。	0x1
[23:20]	-	reserved	保留。	0x0
[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



Bits	Access	Name	Description	Reset
[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

AD_SEL

AD_SEL 为 AD 通路选择寄存器。

Offset Address: 0x0060 Total Reset Value: 0x0000_0008

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5:3]	RW	ad1_sel	AD1 选择配置寄存器。 0x0: CH0 AD 数据; 0x1: CH1 AD 数据; 0x2: CH2 AD 数据;	0x1



Bits	Access	Name	Description	Reset
			0x3: CH3 AD 数据; 0x4: CH4 AD 数据; 0x5: CH5 AD 数据; 0x6: CH6 AD 数据; 其他: 保留。	
[2:0]	RW	ad0_sel	AD0 选择配置寄存器。 0x0: CH0 AD 数据; 0x1: CH1 AD 数据; 0x2: CH2 AD 数据; 0x3: CH3 AD 数据; 0x4: CH4 AD 数据; 0x5: CH5 AD 数据; 0x6: CH6 AD 数据; 其他: 保留。	0x0

VICAP_INT1

VICAP_INT1 为中断指示寄存器。

Offset Address: 0x00E0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27]	RO	int1_fpn0	FPN0 中断指示。 0: 无中断; 1: 有中断。	0x0
[26]	RO	int1_dbg	系统总线异常中断指示。 0: 无中断; 1: 有中断。	0x0
[25]	RO	int1_wdr0	在线 WDR0 中断指示。	0x0



Bits	Access	Name	Description	Reset
			0: 无中断; 1: 有中断。	
[24]	RO	int1_src0	SRC0 中断指示。 0: 无中断; 1: 有中断。	0x0
[23:20]	-	reserved	保留。	0x0
[19]	RO	int1_isp3	ISP3 中断指示。 0: 无中断; 1: 有中断。	0x0
[18]	RO	int1_isp2	ISP2 中断指示。 0: 无中断; 1: 有中断。	0x0
[17]	RO	int1_isp1	ISP1 中断指示。 0: 无中断; 1: 有中断。	0x0
[16]	RO	int1_isp0	ISP0 中断指示。 0: 无中断; 1: 有中断。	0x0
[15]	-	reserved	保留。	0x0
[14]	RO	int1_ch6	通道 6 中断指示。 0: 无中断; 1: 有中断。	0x0
[13]	RO	int1_ch5	通道 5 中断指示。 0: 无中断; 1: 有中断。	0x0
[12]	RO	int1_ch4	通道 4 中断指示。 0: 无中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	
[11]	RO	int1_ch3	通道 3 中断指示。 0: 无中断; 1: 有中断。	0x0
[10]	RO	int1_ch2	通道 2 中断指示。 0: 无中断; 1: 有中断。	0x0
[9]	RO	int1_ch1	通道 1 中断指示。 0: 无中断; 1: 有中断。	0x0
[8]	RO	int1_ch0	通道 0 中断指示。 0: 无中断; 1: 有中断。	0x0
[7:4]	-	reserved	保留。	0x0
[3]	RO	int1_pt3	端口 3 中断指示。 0: 无中断; 1: 有中断。	0x0
[2]	RO	int1_pt2	端口 2 中断指示。 0: 无中断; 1: 有中断。	0x0
[1]	RO	int1_pt1	端口 1 中断指示。 0: 无中断; 1: 有中断。	0x0
[0]	RO	int1_pt0	端口 0 中断指示。 0: 无中断; 1: 有中断。	0x0



VICAP_INT1_MASK

VICAP_INT1_MASK 为中断 MASK 配置寄存器。

Offset Address: 0x00E8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27]	RW	mask1_fpn0	FPN0 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[26]	RW	mask1_dbg	系统总线异常中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[25]	RW	mask1_wdr0	在线 WDR0 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[24]	RW	mask1_src0	SRC0 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[23:20]	-	reserved	保留。	0x0
[19]	RW	mask1_isp3	ISP3 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[18]	RW	mask1_isp2	ISP2 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[17]	RW	mask1_isp1	ISP1 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0



Bits	Access	Name	Description	Reset
[16]	RW	mask1_isp0	ISP0 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[15]	-	reserved	保留。	0x0
[14]	RW	mask1_ch6	通道 6 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[13]	RW	mask1_ch5	通道 5 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[12]	RW	mask1_ch4	通道 4 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[11]	RW	mask1_ch3	通道 3 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[10]	RW	mask1_ch2	通道 2 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[9]	RW	mask1_ch1	通道 1 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[8]	RW	mask1_ch0	通道 0 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[7:4]	-	reserved	保留。	0x0
[3]	RW	mask1_pt3	端口 3 中断使能。	0x0



Bits	Access	Name	Description	Reset
			0: 屏蔽中断; 1: 使能中断。	
[2]	RW	mask1_pt2	端口 2 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	RW	mask1_pt1	端口 1 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	mask1_pt0	端口 0 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0

VICAP_INT0

VICAP_INT0 为中断指示寄存器。

Offset Address: 0x00F0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	保留。	0x0
[28]	-	reserved	保留。	0x0
[27]	RO	int0_fpn0	FPN0 中断指示。 0: 无中断; 1: 有中断。	0x0
[26]	RO	int0_dbg	系统总线异常中断指示。 0: 无中断; 1: 有中断。	0x0
[25]	RO	int0_wdr0	在线 WDR0 中断指示。 0: 无中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	
[24]	RO	int0_src0	SRC0 中断指示。 0: 无中断; 1: 有中断。	0x0
[23:20]	-	reserved	保留。	0x0
[19]	RO	int0_isp3	ISP3 中断指示。 0: 无中断; 1: 有中断。	0x0
[18]	RO	int0_isp2	ISP2 中断指示。 0: 无中断; 1: 有中断。	0x0
[17]	RO	int0_isp1	ISP1 中断指示。 0: 无中断; 1: 有中断。	0x0
[16]	RO	int0_isp0	ISP0 中断指示。 0: 无中断; 1: 有中断。	0x0
[15]	-	reserved	保留。	0x0
[14]	RO	int0_ch6	通道 6 中断指示。 0: 无中断; 1: 有中断。	0x0
[13]	RO	int0_ch5	通道 5 中断指示。 0: 无中断; 1: 有中断。	0x0
[12]	RO	int0_ch4	通道 4 中断指示。 0: 无中断; 1: 有中断。	0x0



Bits	Access	Name	Description	Reset
[11]	RO	int0_ch3	通道 3 中断指示。 0: 无中断; 1: 有中断。	0x0
[10]	RO	int0_ch2	通道 2 中断指示。 0: 无中断; 1: 有中断。	0x0
[9]	RO	int0_ch1	通道 1 中断指示。 0: 无中断; 1: 有中断。	0x0
[8]	RO	int0_ch0	通道 0 中断指示。 0: 无中断; 1: 有中断。	0x0
[7:4]	-	reserved	保留。	0x0
[3]	RO	int0_pt3	端口 3 中断指示。 0: 无中断; 1: 有中断。	0x0
[2]	RO	int0_pt2	端口 2 中断指示。 0: 无中断; 1: 有中断。	0x0
[1]	RO	int0_pt1	端口 1 中断指示。 0: 无中断; 1: 有中断。	0x0
[0]	RO	int0_pt0	端口 0 中断指示。 0: 无中断; 1: 有中断。	0x0



VICAP_INT0_MASK

VICAP_INT0_MASK 为中断 MASK 配置寄存器。

Offset Address: 0x00F8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	保留。	0x0
[28]	-	reserved	保留。	0x0
[27]	RW	mask0_fpn0	FPN0 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[26]	RW	mask0_dbg	系统总线异常中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[25]	RW	mask0_wdr0	在线 WDR0 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[24]	RW	mask0_src0	SRC0 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[23:20]	-	reserved	保留。	0x0
[19]	RW	mask0_isp3	ISP3 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[18]	RW	mask0_isp2	ISP2 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[17]	RW	mask0_isp1	ISP1 中断使能。 0: 屏蔽中断;	0x0



Bits	Access	Name	Description	Reset
			1: 使能中断。	
[16]	RW	mask0_isp0	ISP0 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[15]	-	reserved	保留。	0x0
[14]	RW	mask0_ch6	通道 6 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[13]	RW	mask0_ch5	通道 5 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[12]	RW	mask0_ch4	通道 4 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[11]	RW	mask0_ch3	通道 3 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[10]	RW	mask0_ch2	通道 2 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[9]	RW	mask0_ch1	通道 1 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[8]	RW	mask0_ch0	通道 0 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[7:4]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[3]	RW	mask0_pt3	端口 3 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RW	mask0_pt2	端口 2 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	RW	mask0_pt1	端口 1 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	mask0_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(Vsync)输出使能。 0: 关闭; 1: 使能。	0x0
[30]	RW	slave0_hs_enable	HS(Hsync)输出使能。 0: 关闭; 1: 使能。	0x0
[29]	RW	slave0_vs_out_enable	VS_OUT 输出使能。 0: 关闭; 1: 使能。	0x0



Bits	Access	Name	Description	Reset
[28:18]	-	reserved	保留。	0x000
[17]	RW	slave0_vs_inv	VS 输出极性。 0: 高有效; 1: 低有效。	0x0
[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/V _{S_OUT} 输出信号切换控制。 0: VS/V _{S_OUT} 输出保持不变; 1: VS/V _{S_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（高位）。

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（低位）。

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_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（高位）。

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（低位）。

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_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

SLAVE1_MODE_CFG

SLAVE1_MODE_CFG 为 SENSOR1 从模式配置寄存器。

Offset Address: 0x0140 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	slave1_vs_enable	VS 输出使能。 0: 关闭;	0x0



Bits	Access	Name	Description	Reset
			1: 使能。	
[30]	RW	slave1_hs_enable	HS 输出使能。 0: 关闭; 1: 使能。	0x0
[29]	RW	slave1_vs_out_enable	VS_OUT 输出使能。 0: 关闭; 1: 使能。	0x0
[28:18]	-	reserved	保留。	0x000
[17]	RW	slave1_vs_inv	VS 输出极性。 0: 高有效; 1: 低有效。	0x0
[16]	RW	slave1_hs_inv	HS 输出极性。 0: 高有效; 1: 低有效。	0x0
[15]	RW	slave1_vs_out_inv	VS_OUT 输出极性。 0: 高有效; 1: 低有效。	0x0
[14]	RW	slave1_vs_mux	VS/V _{S_OUT} 输出信号切换控制。 0: VS/V _{S_OUT} 输出保持不变; 1: VS/V _{S_OUT} 输出切换。	0x0
[13:1]	-	reserved	保留。	0x0000
[0]	RW	slave1_update_mode	更新模式选择。 0: 立即更新模式; 1: 帧更新模式。	0x0



SLAVE1_MODE_VS_TIME_H

SLAVE1_MODE_VS_TIME_H 为 SENSOR1 从模式输出 VSYNC 的周期配置寄存器（高位）。

Offset Address: 0x0144 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave1_vs_time_h	VS 输出周期配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE1_MODE_VS_TIME_L

SLAVE1_MODE_VS_TIME_L 为 SENSOR1 从模式输出 VSYNC 的周期配置寄存器（低位）。

Offset Address: 0x0148 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave1_vs_time_l	VS 输出周期配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE1_MODE_HS_TIME

SLAVE1_MODE_HS_TIME 为 SENSOR1 从模式输出 HSYNC 的周期配置寄存器。

Offset Address: 0x014C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave1_hs_time	HS 输出周期配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE1_MODE_VS_CYC_H

SLAVE1_MODE_VS_CYC_H 为 SENSOR1 从模式输出 VSYNC 脉冲有效信号持续时间配置寄存器（高位）。

Offset Address: 0x0158 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	slave1_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

SLAVE1_MODE_VS_CYC_L

SLAVE1_MODE_VS_CYC_L 为 SENSOR1 从模式输出 VSYNC 脉冲有效信号持续时间配置寄存器 (低位)。

Offset Address: 0x015C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave1_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

SLAVE1_MODE_HS_CYC

SLAVE1_MODE_HS_CYC 为 SENSOR1 从模式输出 HSYNC 脉冲有效信号持续时间配置寄存器。

Offset Address: 0x0160 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave1_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

SLAVE1_MODE_HS_DLY_CYC

SLAVE1_MODE_HS_DLY_CYC 为 SENSOR1 从模式输出 HSYNC 脉冲输出相对 VSYNC 脉冲的延迟周期配置

Offset Address: 0x016C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave1_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	0x00000000



Bits	Access	Name	Description	Reset
			期)。	

SLAVE2_MODE_CFG

SLAVE2_MODE_CFG 为 SENSOR2 从模式配置寄存器。

Offset Address: 0x0180 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	slave2_vs_enable	VS 输出使能。 0: 关闭; 1: 使能。	0x0
[30]	RW	slave2_hs_enable	HS 输出使能。 0: 关闭; 1: 使能。	0x0
[29]	RW	slave2_vs_out_enable	VS_OUT 输出使能。 0: 关闭; 1: 使能。	0x0
[28:18]	-	reserved	保留。	0x000
[17]	RW	slave2_vs_inv	VS 输出极性。 0: 高有效; 1: 低有效。	0x0
[16]	RW	slave2_hs_inv	HS 输出极性。 0: 高有效; 1: 低有效。	0x0
[15]	RW	slave2_vs_out_inv	VS_OUT 输出极性。 0: 高有效; 1: 低有效。	0x0
[14]	RW	slave2_vs_mux	VS/V _{S_OUT} 输出信号切换控制。	0x0



Bits	Access	Name	Description	Reset
			0: VS/V _{S_OUT} 输出保持不变; 1: VS/V _{S_OUT} 输出切换。	
[13:1]	-	reserved	保留。	0x0000
[0]	RW	slave2_update_mode	更新模式选择。 0: 立即更新模式; 1: 帧更新模式。	0x0

SLAVE2_MODE_VS_TIME_H

SLAVE2_MODE_VS_TIME_H 为 SENSOR2 从模式输出 VSYNC 的周期配置寄存器（高位）。

Offset Address: 0x0184 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave2_vs_time_h	VS 输出周期配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE2_MODE_VS_TIME_L

SLAVE2_MODE_VS_TIME_L 为 SENSOR2 从模式输出 VSYNC 的周期配置寄存器（低位）。

Offset Address: 0x0188 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave2_vs_time_l	VS 输出周期配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE2_MODE_HS_TIME

SLAVE2_MODE_HS_TIME 为 SENSOR2 从模式输出 HSYNC 的周期配置寄存器。

Offset Address: 0x018C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	slave2_hs_time	HS 输出周期配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE2_MODE_VS_CYC_H

SLAVE2_MODE_VS_CYC_H 为 SENSOR2 从模式输出 VSYNC 脉冲有效信号持续时间配置寄存器。

Offset Address: 0x0198 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave2_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

SLAVE2_MODE_VS_CYC_L

SLAVE2_MODE_VS_CYC_L 为 SENSOR2 从模式输出 VSYNC 脉冲有效信号持续时间配置寄存器。

Offset Address: 0x019C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave2_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

SLAVE2_MODE_HS_CYC

SLAVE2_MODE_HS_CYC 为 SENSOR2 从模式输出 HSYNC 脉冲有效信号持续时间配置寄存器。

Offset Address: 0x01A0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave2_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



SLAVE2_MODE_HS_DLY_CYC

SLAVE2_MODE_HS_DLY_CYC 为 SENSOR2 从模式输出 HSYNC 脉冲输出相对 VSYNC 脉冲的延迟周期配置

Offset Address: 0x01AC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave2_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

SLAVE3_MODE_CFG

SLAVE3_MODE_CFG 为 SENSOR3 从模式配置寄存器。

Offset Address: 0x01C0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	slave3_vs_enable	VS 输出使能。 0: 关闭; 1: 使能。	0x0
[30]	RW	slave3_hs_enable	HS 输出使能。 0: 关闭; 1: 使能。	0x0
[29]	RW	slave3_vs_out_enable	VS_OUT 输出使能。 0: 关闭; 1: 使能。	0x0
[28:18]	-	reserved	保留。	0x000
[17]	RW	slave3_vs_inv	VS 输出极性。 0: 高有效; 1: 低有效。	0x0



Bits	Access	Name	Description	Reset
[16]	RW	slave3_hs_inv	HS 输出极性。 0: 高有效; 1: 低有效。	0x0
[15]	RW	slave3_vs_out_inv	VS_OUT 输出极性。 0: 高有效; 1: 低有效。	0x0
[14]	RW	slave3_vs_mux	VS/V _{S_OUT} 输出信号切换控制。 0: VS/V _{S_OUT} 输出保持不变; 1: VS/V _{S_OUT} 输出切换。	0x0
[13:1]	-	reserved	保留。	0x0000
[0]	RW	slave3_update_mode	更新模式选择。 0: 立即更新模式; 1: 帧更新模式。	0x0

SLAVE3_MODE_VS_TIME_H

SLAVE3_MODE_VS_TIME_H 为 SENSOR3 从模式输出 VSYNC 的周期配置寄存器（高位）。

Offset Address: 0x01C4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave3_vs_time_h	VS 输出周期配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE3_MODE_VS_TIME_L

SLAVE3_MODE_VS_TIME_L 为 SENSOR3 从模式输出 VSYNC 的周期配置寄存器（低位）。

Offset Address: 0x01C8 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	slave3_vs_time_l	VS 输出周期配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE3_MODE_HS_TIME

SLAVE3_MODE_HS_TIME 为 SENSOR3 从模式输出 HSYNC 的周期配置寄存器。

Offset Address: 0x01CC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave3_hs_time	HS 输出周期配置(单位: SENSOR 工作时钟周期)。	0x00000000

SLAVE3_MODE_VS_CYC_H

SLAVE3_MODE_VS_CYC_H 为 SENSOR3 从模式输出 VSYNC 脉冲有效信号持续时间配置寄存器 (高位)。

Offset Address: 0x01D8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave3_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

SLAVE3_MODE_VS_CYC_L

SLAVE3_MODE_VS_CYC_L 为 SENSOR3 从模式输出 VSYNC 脉冲有效信号持续时间配置寄存器 (低位)。

Offset Address: 0x01DC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave3_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



SLAVE3_MODE_HS_CYC

SLAVE3_MODE_HS_CYC 为 SENSOR3 从模式输出 HSYNC 脉冲有效信号持续时间配置寄存器。

Offset Address: 0x01E0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave3_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

SLAVE3_MODE_HS_DLY_CYC

SLAVE3_MODE_HS_DLY_CYC 为 SENSOR3 从模式输出 HSYNC 脉冲输出相对 VSYNC 脉冲的延迟周期配置

Offset Address: 0x01EC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	slave3_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

SLAVE_MODE_TIME_CFG_SEL

SLAVE_MODE_TIME_CFG_SEL 为 SENSOR 从模式 TIME 配置选择寄存器。

Offset Address: 0x01F8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:6]	RW	sensor3_time_sel	SENSOR3 TIME 选择。 00: 选择 SENSOR0 timing 配置; 01: 选择 SENSOR1 timing 配置; 10: 选择 SENSOR2 timing 配置; 11: 选择 SENSOR3 timing 配置。	0x0



Bits	Access	Name	Description	Reset
[5:4]	RW	sensor2_time_sel	SENR2 TIME 选择。 00: 选择 SENSOR0 timing 配置; 01: 选择 SENSOR1 timing 配置; 10: 选择 SENSOR2 timing 配置; 11: 选择 SENSOR3 timing 配置。	0x0
[3:2]	RW	sensor1_time_sel	SENR1 TIME 选择。 00: 选择 SENSOR0 timing 配置; 01: 选择 SENSOR1 timing 配置; 10: 选择 SENSOR2 timing 配置; 11: 选择 SENSOR3 timing 配置。	0x0
[1:0]	RW	sensor0_time_sel	SENR0 TIME 选择。 00: 选择 SENSOR0 timing 配置; 01: 选择 SENSOR1 timing 配置; 10: 选择 SENSOR2 timing 配置; 11: 选择 SENSOR3 timing 配置。	0x0

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
[31:4]	-	reserved	保留。	0x0000000
[3]	RW	sensor3_sync	SENSOR3 同步信号。	0x0
[2]	RW	sensor2_sync	SENSOR2 同步信号。	0x0
[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 为在线通道 0 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_AFIFO0_SIZE

VICAP_AFIFO0_SIZE 为 AFIFO 大小配置寄存器 0。

Offset Address: 0x030c Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	RW	afifo0_height	获取图像的高度(以像素为单位), 配置值为实际值减 1。	0x0000
[15:0]	RW	afifo0_width	获取图像的宽度(以像素为单位), 配置值为实际值减 1。	0x0000

VICAP_AFIFO1_SIZE

VICAP_AFIFO1_SIZE 为 AFIFO 大小配置为寄存器 1。

Offset Address: 0x031c Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	afifo1_height	获取图像的高度(以像素为单位), 配置值为实际值减 1。	0x0000
[15:0]	RW	afifo1_width	获取图像的宽度(以像素为单位), 配置值为实际值减 1。	0x0000

VICAP_AFIFO2_SIZE

VICAP_AFIFO2_SIZE 为 AFIFO 大小配置为寄存器 2。

Offset Address: 0x032c Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	afifo2_height	获取图像的高度(以像素为单位), 配置值为实际值减 1。	0x0000
[15:0]	RW	afifo2_width	获取图像的宽度(以像素为单位), 配置值为实际值减 1。	0x0000

VICAP_AFIFO3_SIZE

VICAP_AFIFO3_SIZE 为 AFIFO 大小配置为寄存器 3。

Offset Address: 0x033c Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	RW	afifo3_height	获取图像的高度(以像素为单位), 配置值为实际值减 1。	0x0000
[15:0]	RW	afifo3_width	获取图像的宽度(以像素为单位), 配置值为实际值减 1。	0x0000

VICAP_AXI0_RD_ERROR

VICAP_AXI0_RD_ERROR 为 AXI0 读错误状态寄存器。

Offset Address: 0x0fe0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	WC	src0_bus_err	SRC0 通道非法访问错误, 写 1 清 0。	0x0

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]	-	reserved	保留。	0x0
[30]	WC	ch3_tunl_bus_err	CH3 TUNL 通道非法访问错误, 写 1 清 0。	0x0
[29]	WC	ch2_tunl_bus_err	CH2 TUNL 通道非法访问错误, 写 1 清 0。	0x0
[28]	WC	ch1_tunl_bus_err	CH1 TUNL 通道非法访问错误, 写 1 清 0。	0x0
[27]	WC	ch0_tunl_bus_err	CH0 TUNL 通道非法访问错误, 写 1 清 0。	0x0
[26]	-	reserved	保留。	0x0
[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



Bits	Access	Name	Description	Reset
[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
[21:20]	-	reserved	保留。	0x0
[19]	WC	ch3_em_bus_err	CH3 EM 通道非法访问错误, 写 1 清 0。	0x0
[18]	WC	ch2_em_bus_err	CH2 EM 通道非法访问错误, 写 1 清 0。	0x0
[17]	WC	ch1_em_bus_err	CH1 EM 通道非法访问错误, 写 1 清 0。	0x0
[16]	WC	ch0_em_bus_err	CH0 EM 通道非法访问错误, 写 1 清 0。	0x0
[15:8]	-	reserved	保留。	0x00
[7]	WC	ch3_c_bus_err	CH3_C 通道非法访问错误, 写 1 清 0。	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]	WC	ch1_c_bus_err	CH1_C 通道非法访问错误, 写 1 清 0。	0x0
[2]	WC	ch1_y_bus_err	CH1_Y 通道非法访问错误, 写 1 清 0。	0x0
[1]	WC	ch0_c_bus_err	CH0_C 通道非法访问错误, 写 1 清 0。	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;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	
[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 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	pt_enable	0x0_1000 ~ 0x0_10FF: PT0 寄存器地址空间。 0x0_1100 ~ 0x0_11FF: PT1 寄存器地址空间。 0x0_1200 ~ 0x0_12FF: PT2 寄存器地址空间。	0x0



Bits	Access	Name	Description	Reset
			间。 0x0_1300 ~ 0x0_13FF: PT3 寄存器地址空间。(BT.656 复合 ID0) 0x0_1400 ~ 0x0_14FF: PT3 寄存器地址空间。(BT.656 复合 ID1) 端口使能。 0: 关闭; 1: 使能。	
[30:1]	-	reserved	保留。	0x0000000
[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 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 Total Reset Value: 0xFFFF0_0010

Bits	Access	Name	Description	Reset
[31:16]	RW	offset1_mask	1 分量 mask。	0xFFFF0
[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 Total Reset Value: 0xFFFF0_0020

Bits	Access	Name	Description	Reset
[31:16]	RW	offset2_mask	2 分量 mask。	0xFFFF0
[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 Total Reset Value: 0x0000_0303

Bits	Access	Name	Description	Reset
[31]	RW	bt656_enable	BT.656 使能寄存器。 0: 关闭;	0x0



Bits	Access	Name	Description	Reset
			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_MIPI_YUV_CFG

PT_MIPI_YUV_CFG 为 MIPI YUV 输入配置转换开关寄存器。

Offset Address: 0x102C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2]	RW	yuv_vu2uv	MIPI YUV420 UV 奇偶行顺序。 0: U 在前, V 在后; 1: V 在前, U 在后。	0x0
[1:0]	RW	yuv_cfg	MIPI 数据输入格式配置寄存器。 00: MIPI RAW/YUV422 格式(MIPI YUV 时序解析功能关闭); 01: MIPI YUV420, 非 legacy 格式; 10: MIPI YUV420, legacy 格式; 其他: 保留。	0x0

PT_MIPI_YUV_SIZE

PT_MIPI_YUV_SIZE 为 MIPI YUV 输入图像宽高寄存器。

Offset Address: 0x1034 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	mipi_yuv_height	MIPI YUV 实际输入高度减 1。	0x0000
[15:0]	RW	mipi_yuv_width	MIPI YUV 实际宽度减 1。	0x0000

PT_UNIFY_TIMING_CFG

PT_UNIFY_TIMING_CFG 为时序配置寄存器。

Offset Address: 0x1038 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)。 0: 不反相;	0x0



Bits	Access	Name	Description	Reset
			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;	0x0



Bits	Access	Name	Description	Reset
			01: 相与; 10: 异或; 11: 保留。	
[10]	RW	unify_hsync_inv	hsync 是否反相(hsync 第 1 级处理)。 0: 不反相; 1: 反相。	0x0
[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 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;	0x0



Bits	Access	Name	Description	Reset
			1: 使能。	
[30]	RW	gen_mode	时序恢复模式, 根据时序参数恢复时序。 0: 依赖端口输入的 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 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。	0x0000000
[3]	RW	unify_data_uv_seq	CbCr 顺序。 0: CbCr; 1: CrCb。	0x0
[2]	RW	unify_data_yc_seq	YC 顺序。	0x0



Bits	Access	Name	Description	Reset
			0: CY; 1: YC。	
[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 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。 0: 关闭; 1: 使能。	0x0
[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_re	数据是否根据 vsync 信号复位。	0x0



Bits	Access	Name	Description	Reset
		set	0: 不复位; 1: 复位。	
[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 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 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 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 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 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	fstart_h_dly	fstart 中断行延时时间，以为有效区行号为单位。	0x0000
[15:4]	-	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 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 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 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 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 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 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 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 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 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 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
[28:6]	-	reserved	保留。	0x000000
[5:4]	RO	reserved	保留。	0x0
[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 Total Reset Value: 0x1000_0000

Bits	Access	Name	Description	Reset
[31:28]	RW	id_sel	触发帧 ID 选择寄存器。 0x1: 通过 ID 为 0 的帧起始中断触发; 0x2: 通过 ID 为 1 的帧起始中断触发; 0x4: 通过 ID 为 2 的帧起始中断触发; 0x8: 通过 ID 为 3 的帧起始中断触发。	0x1



Bits	Access	Name	Description	Reset
			其他：保留。	
[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 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 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 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 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 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，	0x00000000



			配置值为实际值减 1。	
--	--	--	-------------	--

PT_SHUTTER_CYC3

PT_SHUTTER_CYC3 为快门时序 3 宽度寄存器。

Offset Address: 0x10DC 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 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x00000000
[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 Total Reset Value: 0x0000_0000

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



Bits	Access	Name	Description	Reset
[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 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 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。	0x0



Bits	Access	Name	Description	Reset
			0: 无中断; 1: 有中断。	

PT_INT_STATE

PT_INT_STATE 为端口中断状态寄存器。

Offset Address: 0x10F4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000
[3]	RO	pt_fstart_h_dly_masked	pt_fstart_h_dly 中断 mask 后的中断状态。	0x0
[2]	RO	pt_height_err_masked	pt_height_err 中断 mask 后的中断状态。	0x0
[1]	RO	pt_width_err_masked	pt_width_err 中断 mask 后的中断状态。	0x0
[0]	RO	pt_fstart_masked	pt_fstart 中断 mask 后的中断状态。	0x0

PT_INT_MASK

PT_INT_MASK 为端口中断屏蔽寄存器。

Offset Address: 0x10F8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000
[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;	0x0



Bits	Access	Name	Description	Reset
			1: 使能中断。	
[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

PT3_MUX_MOD

PT3_MUX_MOD 为接口 3 复合模式寄存器。

Offset Address: 0x1800 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	pt3_enable	端口使能。 0: 关闭; 1: 使能。	0x0
[30:25]	-	reserved	保留。	0x00
[24]	RW	pt3_interleave_seq	interleave 时的 YC 顺序。 0: C 在前; 1: Y 在前。	0x0
[23]	RW	pt3_id3_en	3 通道 ID 使能。 0: 不检测 ID; 1: 通过 ID 选择数据通路。	0x0
[22]	RW	pt3_id2_en	2 通道 ID 使能。 0: 不检测 ID; 1: 通过 ID 选择数据通路。	0x0
[21]	RW	pt3_id1_en	1 通道 ID 使能。	0x0



Bits	Access	Name	Description	Reset
			0: 不检测 ID; 1: 通过 ID 选择数据通路。	
[20]	RW	pt3_id0_en	0 通道 ID 使能。 0: 不检测 ID; 1: 通过 ID 选择数据通路。	0x0
[19:10]	-	reserved	保留	0x000
[9:8]	RW	pt3_port_mux_mode	BT.656 复合时序模式。 00: 非复合时序; 01: 2 路复合模式; 10: 3 路复合模式; 11: 4 路复合模式。	0x0
[7:5]	-	reserved	保留。	0x0
[4]	RW	pt3_interleave_mode	interleave 模式选择。 0: 不是 BT.1120 interleave 模式; 1: 是 BT.1120 interleave 模式。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	pt3_mode	时序模式配置。 0: 外同步; 1: BT.656。	0x0

PT3_MUX_OFFSET

PT3_MUX_OFFSET 为接口 offset 寄存器。

Offset Address: 0x1810 Total Reset Value: 0xFFFF_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	mask	0 分量 mask。	0xFFFF
[15]	RW	pt3_rev	数据线是否翻转。	0x0



Bits	Access	Name	Description	Reset
			0: 不反转; 1: 反转。	
[14:6]	-	reserved	保留。	0x000
[5:0]	RW	pt3_offset	0 分量偏移。	0x00

PT3_ID_STATUS

PT3_ID_STATUS 为接口 ID 状态寄存器。

Offset Address: 0x1850 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RO	pt3_id3	通道 3 检测到数据(id=3)。	0x0
[2]	RO	pt3_id2	通道 2 检测到数据(id=2)。	0x0
[1]	RO	pt3_id1	通道 1 检测到数据(id=1)。	0x0
[0]	RO	pt3_id0	通道 0 检测到数据(id=0)。	0x0

CH_CTRL

CH_CTRL 为通道控制寄存器。

Offset Address: 0x10000 Total Reset Value: 0x0002_0000

Bits	Access	Name	Description	Reset
[31]	RW	ch_enable	0x1_0000 ~ 0x1_0FFF: CH0 寄存器地址空间。(单分量 RAW) 0x1_1000 ~ 0x1_1FFF: CH1 寄存器地址空间。(单分量 RAW) 0x1_2000 ~ 0x1_2FFF: CH2 寄存器地址空间。(单分量 RAW) 0x1_3000 ~ 0x1_3FFF: CH3 寄存器地址空	0x0



Bits	Access	Name	Description	Reset
			间。(单分量 RAW/YUV) 0x1_4000 ~ 0x1_4FFF: CH4 寄存器地址空间。(单分量 RAW/YUV) 0x1_5000 ~ 0x1_5FFF: CH5 寄存器地址空间。(单分量 RAW/YUV) 0x1_6000 ~ 0x1_6FFF: CH6 寄存器地址空间。(单分量 RAW/YUV) 通道使能寄存器。 0: 不使能; 1: 使能。	
[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。	0x0000

CH_REG_NEWER

CH_REG_NEWER 为采集控制寄存器。

Offset Address: 0x10004 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 Total Reset Value: 0x0010_0010

Bits	Access	Name	Description	Reset
[31:16]	RW	ch_v_dly_cfg	垂直行数 delay 寄存器。	0x0010
[15:0]	RW	ch_v_dly1_cfg	垂直行数 delay1 寄存器。	0x0010

CH_FRM_CNT

CH_FRM_CNT 为延时中断锁存帧号寄存器。

Offset Address: 0x10038 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	保留。	0x0000
[15:0]	RO	ch_frm_cnt1	延时中断 1 锁存帧号寄存器(0-255)。	0x0000

CH_CROP_CFG

CH_CROP_CFG 为通道 CROP 使能寄存器。

Offset Address: 0x10040 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2]	RW	ch_crop_mode	CROP 模式。 0: 水平区域内 CROP; 1: 水平区域外 CROP。	0x0
[1]	RW	ch_crop_n1_en	1 区域使能。 0: 禁止; 1: 使能。	0x0
[0]	RW	ch_crop_n0_en	0 区域使能。	0x0



Bits	Access	Name	Description	Reset
			0: 禁止; 1: 使能。	

CH_CROP0_Y_CFG

CH_CROP0_Y_CFG 为通道垂直 CROP 配置寄存器。

Offset Address: 0x10044 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_height	获取图像的高度(以行为单位), 配置值为实际值减 1。	0x0000

CH_CROP0_X0_CFG

CH_CROP0_X0_CFG 为通道 0 区域水平 CROP 配置寄存器。

Offset Address: 0x10048 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	ch_crop_x_n0_start	开始获取图像的像素号。	0x0000
[15:0]	RW	ch_crop_n0_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_CROP0_X1_CFG

CH_CROP0_X1_CFG 为通道 1 区域水平 CROP 配置寄存器。

Offset Address: 0x1004C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	ch_crop_x_n1_start	开始获取图像的像素号。	0x0000
[15:0]	RW	ch_crop_n1_width	获取图像的宽度(以像素为单位), 配置值为实际值减 1。	0x0000



Bits	Access	Name	Description	Reset
			为实际值减 1。	

CH_SKIP_Y_CFG

CH_SKIP_Y_CFG 为通道 Y SKIP 配置寄存器。

Offset Address: 0x10050 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 Total Reset Value: 0xFFFF_FFFF

Bits	Access	Name	Description	Reset
[31:0]	RW	ch_skipc_cfg	SKIP 配置。	0xFFFFFFFF

CH_VCDS_CFG

CH_VCDS_CFG 为主码流色度垂直下采样配置寄存器。

Offset Address: 0x10060 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	vcds_enable	下采样使能。 0: 下采样关闭; 1: 下采样使能。	0x0
[30:0]	-	reserved	保留。	0x00000000



CH_VCDS_COEF

CH_VCDS_COEF 为主码流色度垂直下采样系数寄存器。

Offset Address: 0x10064 Total Reset Value: 0x0000_001F

Bits	Access	Name	Description	Reset
[31:21]	-	reserved	保留。	0x000
[20:16]	RW	vcds_coef1	下采样系数 1。	0x00
[15:5]	-	reserved	保留。	0x000
[4:0]	RW	vcds_coef0	下采样系数 0。	0x1F

CH_WCH_Y_CFG

CH_WCH_Y_CFG 为 Y 分量配置寄存器。

Offset Address: 0x10080 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]	RO	reserved	保留。	0x0
[29]	RW	wch_y_tunl_en	低延时使能。 0: 不使能; 1: 使能。	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;	0x0



Bits	Access	Name	Description	Reset
			1: 行压缩。	
[12]	RW	wch_y_cmp_en	写通道压缩使能寄存器。 0: 不使能; 1: 使能。	0x0
[11:8]	RW	wch_y_bit_sft	16bit 数据低对齐配置。 Wch_y_bit_width 配置 16, 如果输入数据 10bit, 需要移位 6bit。	0x0
[7]	RW	wch_y_bit_sft_mod e	shift 模式配置。 0: 无符号数移位; 1: 符号数移位。	0x0
[6]	-	reserved	保留。	0x0
[5:0]	RW	wch_y_bit_width	数据位宽。 0x08: 8bit; 0x0A: 10bit; 0x0C: 12bit; 0x0E: 14bit; 0x10: 16bit; 0x14: 20bit; 0x18: 24bit ; 0x1c: 28bit ; 0x20: 32bit; 其他: 保留。	0x00

CH_WCH_Y_SIZE

CH_WCH_Y_SIZE 为 Y 分量存储大小寄存器。

Offset Address: 0x10084 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_H

CH_WCH_Y_FADDR_H 为 Y 分量存储基地址高 32bit 寄存器。

Offset Address: 0x10090 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	wch_y_faddr_h	Y 分量存储基地址高 32bit。	0x00000000

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 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 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 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]	RO	reserved	保留。	0x0
[29]	RW	wch_c_tunl_en	低延时使能。	0x0
[28:27]	RW	wch_tunl_mode	OUT0 通道 TUNNEL 信息写间隔。 00: 每 2 行写 1 次; 01: 每 4 行写 1 次; 10: 每 8 行写 1 次; 11: 每 16 行写 1 次。	0x0
[26:13]	RW	wch_tunl_finish_line	OUT0 通道 TUNNEL 水线。	0x0000
[12]	-	reserved	保留。	0x0
[11:8]	RW	wch_c_bit_sft	16bit 数据低对齐配置。 Wch_c_bit_width 配置 16, 如果输入数据 10bit, 需要移位 6bit。	0x0
[7]	RW	wch_c_bit_sft_mode	shift 模式配置。 0: 无符号数移位; 1: 符号数移位。	0x0
[6]	-	reserved	保留。	0x0
[5:0]	RW	wch_c_bit_width	写通道数据位宽。 0x08: 8bit; 0x0A: 10bit; 0x0C: 12bit;	0x00



Bits	Access	Name	Description	Reset
			0x0E: 14bit; 0x10: 16bit; 0x14: 20bit; 0x18: 24bit ; 0x1c: 28bit ; 0x20: 32bit; 其他: 保留。	

CH_WCH_C_SIZE

CH_WCH_C_SIZE 为 C 分量存储大小寄存器。

Offset Address: 0x100B4 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_H

CH_WCH_C_FADDR_H 为 C 分量存储基地址高 32bit 寄存器。

Offset Address: 0x100C0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	wch_c_faddr_h	C 分量存储基地址高 32bit。	0x00000000

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 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_TUNL_HADDR_H

CH_WCH_TUNL_HADDR_H 为低延时信息基地址高 32bit 寄存器。

Offset Address: 0x100C8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	wch_tunl_addr_h	低延时基地址高 32bit。	0x00000000

CH_WCH_TUNL_HADDR_L

CH_WCH_TUNL_HADDR_L 为低延时信息基地址低 32bit 寄存器。

Offset Address: 0x100CC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	wch_tunl_addr_l	低延时基地址低 32bit。	0x00000000

CH_WCH_C_STRIDE

CH_WCH_C_STRIDE 为 C 分量行偏移寄存器。

Offset Address: 0x100D0 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_RCH_PARA_ADDR_H

CH_RCH_PARA_ADDR_H 为低延时信息基地址高 32bit 寄存器。



Offset Address: 0x100D8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	rch_para_addr_h	LBCL 系数基地址高 32bit。	0x00000000

CH_RCH_PARA_ADDR_L

CH_RCH_PARA_ADDR_L 为低延时信息基地址低 32bit 寄存器。

Offset Address: 0x100DC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	rch_para_addr_l	LBCL 系数基地址低 32bit。	0x00000000

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 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 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 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:23]	-	reserved	保留。	0x000
[22]	WC	ch_lblc_para_int	lblc para 载入完成中断状态, 写 1 清零。 0: 无中断; 1: 有中断。	0x0
[21]	WC	ch_tunl_line_int	低延时水线中断状态, 写 1 清零。 0: 无中断; 1: 有中断。	0x0
[20]	WC	ch_em_int_buf_ovf	EM 写出内部 FIFO 溢出错误中断状态, 写 1 清零。 0: 无中断; 1: 有中断。	0x0
[19]	WC	ch_stt_int_buf_ovf	STT 写出内部 FIFO 溢出错误中断状态, 写 1 清零。 0: 无中断; 1: 有中断。	0x0
[18]	WC	ch_stt_em_finish	EM 信息完成中断状态, 写 1 清零。 0: 无中断; 1: 有中断。	0x0
[17]	WC	ch_stt_af_finish	AF 统计信息完成中断状态, 写 1 清零。 0: 无中断; 1: 有中断。	0x0
[16]	WC	ch_stt_ae_finish	AE 统计信息完成中断状态, 写 1 清零。 0: 无中断; 1: 有中断。	0x0



Bits	Access	Name	Description	Reset
[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]	WC	ch_int_fstart_dly1	delay1 后的场/帧起始中断状态, 写 1 清零。 0: 无中断; 1: 有中断。	0x0
[13:5]	-	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_STATE

CH_INT_STATE 为通道原始中断状态寄存器。



Offset Address: 0x100F4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:23]	-	reserved	保留。	0x000
[22]	RO	ch_lblc_para_int_masked	ch_lblc_para_int 中断 mask 后的中断状态。	0x0
[21]	RO	ch_tunl_line_int_masked	ch_tunl_line_int 中断 mask 后的中断状态。	0x0
[20]	RO	ch_em_int_buf_ovf_masked	ch_em_int_buf_ovf 中断 mask 后的中断状态。	0x0
[19]	RO	ch_stt_int_buf_ovf_masked	ch_stt_int_buf_ovf 中断 mask 后的中断状态。	0x0
[18]	RO	ch_stt_em_finish_masked	ch_stt_em_finish 中断 mask 后的中断状态。	0x0
[17]	RO	ch_stt_af_finish_masked	ch_stt_af_finish 中断 mask 后的中断状态。	0x0
[16]	RO	ch_stt_ae_finish_masked	ch_stt_ae_finish 中断 mask 后的中断状态。	0x0
[15]	RO	ch_int_fstart_dly_masked	ch_int_fstart_dly 中断 mask 后的中断状态。	0x0
[14]	RO	ch_int_fstart_dly1_masked	ch_int_fstart_dly1 中断 mask 后的中断状态。	0x0
[13:5]	-	reserved	保留。	0x000
[4]	RO	ch_int_update_cfg_masked	ch_int_update_cfg 中断 mask 后的中断状态。	0x0
[3]	RO	ch_int_field_throw_masked	ch_int_field_throw 中断 mask 后的中断状态。	0x0
[2]	RO	ch_int_buf_ovf_masked	ch_int_buf_ovf 中断 mask 后的中断状态。	0x0
[1]	RO	ch_int_cc_int_masked	ch_int_cc_int 中断 mask 后的中断状态。	0x0
[0]	RO	ch_int_fstart_masked	ch_int_fstart 中断 mask 后的中断状态。	0x0



CH_INT_MASK

CH_INT_MASK 为通道中断屏蔽寄存器。

Offset Address: 0x100F8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:23]	-	reserved	保留。	0x000
[22]	RW	mask_lblc_para_int	LBLC para 载入完成中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[21]	RW	mask_tunl_line_int	低延时水线中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[20]	RW	mask_em_buf_ovf	EM 写出内部 FIFO 溢出错误中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[19]	RW	mask_stt_buf_ovf	STT 写出内部 FIFO 溢出错误中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[18]	RW	mask_stt_em_finish	EM 信息完成中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[17]	RW	mask_stt_af_finish	AF 统计信息完成中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[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]	RW	mask_fstart_dly1	delay1 后的场/帧起始中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[13:5]	-	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_AE_HIST_ADDR_H

CH_WCH_STT_AE_HIST_ADDR_H 为 ISP AE_HIST 统计信息存储基地址高 32bit 寄存器。

Offset Address: 0x10204 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	ch_wch_stt_ae_hist_addr_h	ISP AE_HIST 统计信息存储基地址高 32bit。 (仅 CH0/CH1/CH2/CH3 支持)	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 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_H

CH_WCH_STT_AE_AVER_R_GR_ADDR_H 为 ISP AE_AVER_R_GR 统计信息存储基地址高 32bit 寄存器。

Offset Address: 0x1020C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	ch_wch_stt_ae_ave_r_r_gr_addr_h	ISP AE_AVER_R_GR 统计信息存储基地址高 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 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	ch_wch_stt_ae_ave_r_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_H

CH_WCH_STT_AE_AVER_GB_B_ADDR_H 为 ISP AE_AVER_GB_B 统计信息存储基地址高 32bit 寄存器。

Offset Address: 0x10214 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	ch_wch_stt_ae_ave_r_gb_b_addr_h	ISP AE_AVER_GB_B 统计信息存储基地址高 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 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	ch_wch_stt_ae_ave_r_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_Y_ADDR_H

CH_WCH_STT_AF_STAT_Y_ADDR_H 为 ISP AF_STAT_Y 统计信息存储基地址高 32bit 寄存器。

Offset Address: 0x1023C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	ch_wch_stt_af_stat_y_addr_h	ISP AF_STAT_Y 统计信息存储基地址高 32bit。 (仅 CH0 支持)	0x00000000

CH_WCH_STT_AF_STAT_Y_ADDR_L

CH_WCH_STT_AF_STAT_Y_ADDR_L 为 ISP AF_STAT_Y 统计信息存储基地址低 32bit 寄存器。

Offset Address: 0x10240 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	ch_wch_stt_af_stat_y_addr_l	ISP AF_STAT_Y 统计信息存储基地址低 32bit。 (仅 CH0 支持)	0x00000000

CH_WCH_EM_CFG

CH_WCH_EM_CFG 为 Embedded Data 写出配置寄存器。

Offset Address: 0x10280 Total Reset Value: 0x0002_0010

Bits	Access	Name	Description	Reset
[31]	RW	ch_wch_enable	Embedded Data 写出使能。 (仅 CH0/CH1/CH2/CH3 支持) 0: 关闭; 1: 使能。	0x0
[30]	RO	reserved	保留。	0x0
[29]	RW	ch_wch_em_mode	Embedded Data 模式。 0: 在后消隐区; 1: 在前消隐区。	0x0
[28:18]	-	reserved	保留。	0x000



[17]	RO	reserved	保留。	0x1
[16:6]	-	reserved	保留	0x000
[5:0]	RW	ch_wch_em_bit_width	数据位宽。 0x10: 16bit; 其他: 保留。	0x10

CH_WCH_EM_SIZE

CH_WCH_EM_SIZE 为 Embedded Data 信息写出大小寄存器。

Offset Address: 0x10284 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	ch_wch_em_size	Embedded Data 写出大小。减一配置。	0x0000

CH_WCH_EM_FADDR_H

CH_WCH_EM_FADDR_H 为 WCH 模块 Y 分量存储基地址高 32bit 寄存器。

Offset Address: 0x10288 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	ch_wch_em_faddr_h	EM 信息存储基地址高 32bit。	0x00000000

CH_WCH_EM_FADDR_L

CH_WCH_EM_FADDR_L 为 Embedded Data 存储基地址低 32bit 寄存器。

Offset Address: 0x1028C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	ch_wch_em_faddr_l	Embedded Data 存储基地址低 32bit。	0x00000000



10.2 VDP

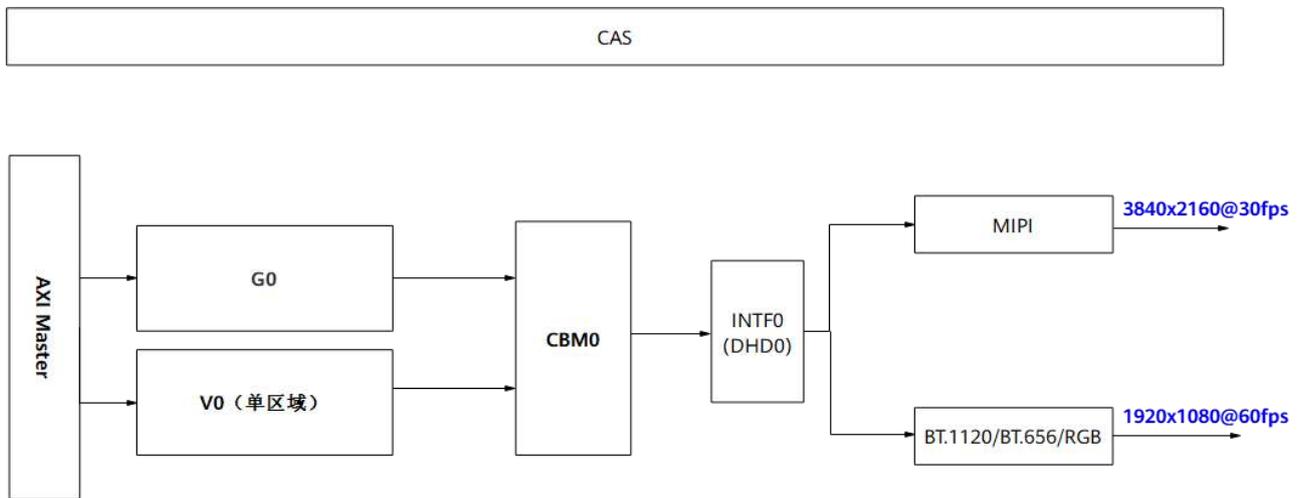
10.2.1 概述

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。

10.2.2 架构描述

VDP 的总体框图如图 10-19 所示。

图10-19 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 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：

- 4-lane MIPI-Tx 输出接口（兼容数字 1.8V LVCMOS），@2.5Gbps/lane。
- 数字输出接口：支持 ITU-R BT.1120/BT.656/RGB 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，每个通道有 4 个垂直时序中断以及 1 个低带宽中断。

10.2.3 工作方式

10.2.3.1 时钟配置

VDP 的可配时钟寄存器有 1 个，具体参见 3.2.6 章节 CRG 寄存器：
PERI_CRG8336~PERI_CRG8348。

10.2.3.2 复位

VDP 的复位包括 1 个硬件复位、1 个软件复位。

须知

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。
-

10.2.3.3 输出接口

VDP 支持以下两种接口输出。

- MIPI
- ITU-R BT.1120/BT.656/RGB



10.2.3.4 中断

VDP 中断分以下两类。

- 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，每个显示通道可生成 4 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，每个显示通道可生成 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。

10.2.4 功能描述

10.2.4.1 视频层功能

视频层 V0 特性

- 支持非压缩数据源。
- 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 3840x2160。
- 最小输出分辨率 32x32，最大输出分辨率为 3840x2160。
- 支持拆分功能。
- 支持 Hsharpen 处理。
- 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。
- 支持垂直色度上采样功能 CVFIR，输入分辨率宽度超过 1920 时，CVFIR 只支持复制模式。
- 支持水平色度上采样。
- 支持色彩空间转换，支持亮度/对比度/色调/饱和度调节。
- 支持视频层 BT.601、BT.709 色域转换。

10.2.4.2 图形层功能

图形层 G0 特性

- 支持非压缩数据源。
- 支持的输入像素格式：ARGB1555、ARGB4444、ARGB8888
- 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
- 输入水平垂直分辨率为 2 的倍数。
- 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。

10.2.4.3 叠加处理

- 支持 V0、G0 叠加。
- 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。

10.2.4.4 显示通道

显示通道特性

- DHD0 可作为高清 0 的输出通道。



- DHD0 支持以下典型输出时序：2160P@30fps、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。

须知

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。

10.2.4.5 输出接口

MIPI 特性

- 最大支持 3840x2160@30fps。
- 时钟频率最高 297MHz。
- 支持 gamma 处理。
- 支持 Raw8/10/12/14/16 数据透传。
- 通过配置接口 CSC 实现 RGB888 输出。
- 支持数据选择来自 DHD0。

BT.1120/BT.656 显示接口特性

- 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@60fps、1080p@60fps、PAL/NTSC
- 支持 dither 处理。
- BT.1120 支持 16bit 数据，默认 Y 在高位，C 在低位，YC 位置可互换，支持 bit 倒序。
- BT.656 支持 8bit 数据，不支持 YC 位置互换，支持 bit 倒序。
- 单沿时钟频率最大支持 148.5MHz。
- 支持输出随路时钟反相。
- 支持数据选择来自 DHD0。



RGB 显示接口特性

- 支持 RGB 接口输出。
- 支持并行 RGB565 (16bit)、RGB666 (18bit)、RGB888 (24bit) 并行输出，RGB/BGR 顺序可选。
- 支持串行 6bit、8bit 输出，不支持分量互换。支持输出随路时钟反相。
- 支持 gamma 处理。
- 串行 4 周期支持 dummy 位置可配。
- 支持数据选择来自 DHD0。

10.2.5 VDP 寄存器概览

VDP 寄存器概览如表 10-5 所示。

表10-5 VDP 寄存器概览 (基址是 0x0_17A0_0000)

偏移地址	名称	描述	页码
0x0000	VOCTRL	VDP 控制寄存器	10-115
0x0004	VOINTSTA	VDP 中断状态, 只读寄存器	10-116
0x0008	VOMSKINTSTA	VDP 经过 Mask 的中断状态寄存器	10-117
0x000C	VOINTMSK	VDP 中断屏蔽寄存器	10-118
0x0014	VOINTSTA1	VDP 中断状态 1, 只读寄存器	10-119
0x0018	VOMSKINTSTA1	VDP 经过 Mask 的离线中断状态寄存器 1	10-120
0x001C	VOINTMSK1	VDP 中断屏蔽寄存器 1	10-121
0x0300	CBM_BKG1	CBM Mixer1 叠加背景色寄存器	10-122
0x0308	CBM_MIX1	CBM Mixer1 优先级配置寄存器	10-122
0x1000	V0_CTRL	V0 控制寄存器, 为非即时寄存器	10-124
0x1004	V0_UPD	V0 通道更新使能寄存器	10-124
0x1010	V0_IRESO	V0 输入分辨率读取寄存器	10-125



偏移地址	名称	描述	页码
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	10-125
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	10-125
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	10-126
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	10-126
0x1090	V0_BK	surface 的背景色寄存器	10-126
0x1094	V0_ALPHA	surface 的背景填充色 ALPHA 寄存器	10-127
0x10A0	V0_RIMWIDTH	视频层 RIM 宽度寄存器, 为非即时寄存器	10-127
0x10A4	V0_RIMCOL0	视频层高 32 分区 RIM 颜色 0 寄存器, 为非即时寄存器	10-127
0x10A8	V0_RIMCOL1	视频层高 32 分区 RIM 颜色 1 寄存器, 为非即时寄存器	10-128
0x1200	V0_HIPP_CSC_CTRL	HIPP_CSC 控制寄存器	10-128
0x1400	V0_CVFIR_VINFO	亮度垂直位置偏移寄存器	10-128
0x1404	V0_CVFIR_VSP	垂直缩放参数配置 (Vertical Scaling Parameter) 寄存器	10-129
0x1500	V0_HFIR_CTRL	hfir 控制寄存器	10-130
0x1624	V0_HSPCFG10	V0 水平锐化配置寄存器 10	10-130
0x7000	G0_CTRL	G0 控制寄存器, 为非即时寄存器	10-131



偏移地址	名称	描述	页码
0x7004	G0_UPD	图形层更新使能寄存器	10-131
0x7010	G0_IRESO	输入分辨率寄存器	10-132
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	10-132
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	10-132
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	10-133
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	10-133
0x7090	G0_BK	图形层的背景色寄存器	10-133
0x7094	G0_ALPHA	图形层的背景填充色 ALPHA 寄存器	10-134
0x7100	G0_HIPP_CSC_CTRL	HIPP_CSC 控制寄存器	10-134
0xD000	DHD0_CTRL	显示通道的总体控制寄存器	10-134
0xD004	DHD0_VSYNC1	显示通道的垂直时序控制 1 寄存器	10-136
0xD008	DHD0_VSYNC2	显示通道的垂直时序控制 2 寄存器	10-136
0xD00C	DHD0_HSYNC1	显示通道的水平时序控制 1 寄存器	10-137
0xD010	DHD0_HSYNC2	显示通道的水平时序控制 2 寄存器	10-137
0xD014	DHD0_VPLUS1	显示通道隔行输出的底场垂直时序控制寄存器	10-137
0xD018	DHD0_VPLUS2	显示通道隔行输出的底场垂直时序控制寄存器	10-138



偏移地址	名称	描述	页码
0xD01C	DHD0_PWR	同步信号脉冲宽度寄存器, 非即时寄存器	10-138
0xD020	DHD0_VTTHD3	垂直时序门限值 3 (Vertical Timing Threshold), 为即时寄存器	10-139
0xD024	DHD0_VTTHD	垂直时序门限值 (Vertical Timing Threshold), 为即时寄存器	10-139
0xD02C	DHD0_PRECHARGE_THD	TCON 预充电阈值寄存器	10-140
0xD030	DHD0_START_POS	DHD0 通道 start 信号起始位置寄存器	10-141
0xD044	DHD0_RGB_FIX_CTRL	INTF 输出颜色固定寄存器	10-141
0xD090	VO_MUX	VDP 输出接口复选寄存器	10-141
0xD200	INTF_BT_CTRL	接口通道配置寄存器	10-142
0xD204	INTF_BT_UPD	INTF_BT 更新寄存器	10-143
0xD210	BT_CLIP0_L	BT.1120 Clip 处理最低门限值寄存器, 为即时寄存器	10-143
0xD214	BT_CLIP0_H	BT.1120 Clip 处理最高门限值寄存器, 为即时寄存器	10-144
0xD280	BT_DITHER_CTRL	Dither 控制寄存器	10-144
0xD2C4	BT_DITHER_THR	Dither 门限值寄存器	10-146
0xD300	INTF_RGB_CTRL	接口通道配置寄存器	10-146
0xD304	INTF_RGB_UPD	INTF_RGB 更新寄存器	10-147
0xD320	RGB_CSC_IDC	色彩空间转换输入直流分量寄存器, 为即时寄存器	10-148
0xD380	RGB_DITHER_CTRL	Dither 控制寄存器	10-148
0xD3C4	RGB_DITHER_THR	Dither 门限值寄存器	10-150



偏移地址	名称	描述	页码
0xD844	MIPI_DITHER_THR	Dither 门限值寄存器	10-150
0xD900	MIPI_GAMMA_CTRL	Gamma 控制寄存器	10-150
0x10004	MAC_CTRL	MAC&MASTER 控制寄存器	10-150
0x10200	VID_READ_CTRL	VID FDR 读取控制寄存器, 为非即时寄存器	10-151
0x10204	VID_MAC_CTRL	VID FDR 申请控制寄存器, 为非即时寄存器	10-152
0x10210	VID_OUT_CTRL	VID FDR 输出控制寄存器, 为非即时寄存器	10-153
0x10214	VID_MUTE_ALPHA	视频层 FDR MUTE 颜色 alpha 寄存器, 为非即时寄存器	10-153
0x1021C	VID_MUTE_BK	视频层 FDR MUTE 颜色寄存器, 为非即时寄存器	10-154
0x10240	VID_SRC_INFO	VID FDR 输入图像属性寄存器, 为非即时寄存器	10-154
0x10244	VID_SRC_RESO	原始图像分辨率, 非及时寄存器	10-155
0x10248	VID_SRC_CROP	源图的 CROP 寄存器, 非及时寄存器	10-155
0x1024C	VID_IN_RESO	输入图像分辨率寄存器, 为非即时寄存器	10-156
0x10250	VID_ADDR_H	亮度地址高位, 为非即时寄存器	10-156
0x10254	VID_ADDR_L	亮度地址低位, 为非即时寄存器	10-156
0x10258	VID_CADDR_H	色度地址高位, 为非即时寄存器	10-156
0x1025C	VID_CADDR_L	色度地址低位, 为非即时寄存器	10-157
0x10270	VID_STRIDE	stride 配置寄存器	10-157
0x10278	VID_HEAD_STRIDE	视频层头信息 stride 寄存器	10-157



偏移地址	名称	描述	页码
0x10290	VID_HEAD_ADDR_H	亮度头地址高位, 为非即时寄存器	10-158
0x10294	VID_HEAD_ADDR_L	亮度头地址低位, 为非即时寄存器	10-158
0x10298	VID_HEAD_CADDR_H	色度头地址高位, 为非即时寄存器	10-158
0x1029C	VID_HEAD_CADDR_L	色度头地址低位, 为非即时寄存器	10-159
0x102C0	VID_TUNL_CTRL	TUNL 低延时控制寄存器	10-159
0x102C4	VID_TUNL_CROP	视频层低延时 crop 掉的行	10-159
0x102D0	VID_TUNL_ADDR_H	TUNL 低延时行号地址高 bit	10-160
0x102D4	VID_TUNL_ADDR_L	TUNL 低延时行号地址低 bit	10-160
0x10E00	GFX_READ_CTRL	GFX FDR 读取控制寄存器, 为非即时寄存器	10-160
0x10E04	GFX_MAC_CTRL	VID FDR 申请控制寄存器, 为非即时寄存器	10-161
0x10E08	GFX_OUT_CTRL	GFX FDR 输出控制寄存器, 为非即时寄存器	10-161
0x10E10	GFX_MUTE_ALPHA	图形层 FDR MUTE 颜色 alpha 寄存器, 为非即时寄存器	10-163
0x10E14	GFX_MUTE_BK	图形层 FDR MUTE 颜色寄存器, 为非即时寄存器	10-163
0x10E28	GFX_1555_ALPHA	ARGB1555 格式 alpha 值, 为非即时寄存器	10-163
0x10E40	GFX_SRC_INFO	GFX FDR 源图像属性寄存器, 为非即时寄存器	10-164
0x10E44	GFX_SRC_RESO	原始图像分辨率, 非及时寄存器	10-164
0x10E48	GFX_SRC_CROP	源图的 CROP 寄存器, 非及时寄存器	10-164



偏移地址	名称	描述	页码
0x10E4C	GFX_IRESO	输入分辨率寄存器, 为非即时寄存器	10-165
0x10E50	GFX_ADDR_H	图形层高 bit 地址寄存器	10-165
0x10E54	GFX_ADDR_L	图形层低 bit 地址寄存器	10-165
0x10E60	GFX_STRIDE	图形层的 stride 寄存器	10-166
0x10E70	GFX_DCMP_ADDR_H	图形层左眼压缩高位地址寄存器	10-166
0x10E74	GFX_DCMP_ADDR_L	图形层左眼压缩低位地址寄存器	10-166
0x10F00	GFX_CKEY_MAX	color key 最大值, 为非即时寄存器	10-166
0x10F04	GFX_CKEY_MIN	color key 最小值, 为非即时寄存器	10-167
0x10F08	GFX_CKEY_MASK	color key mask 值, 为非即时寄存器	10-167

10.2.6 VDP 寄存器描述

VOCTRL

VOCTRL 为 VDP 控制寄存器。

Offset Address: 0x0000 Total Reset Value: 0x8000_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。	0x1
[30]	-	reserved	保留。	0x0
[29]	RW	v0_ck_gt_en	V0 层时钟门控使能,既低功耗模式寄存器。 0: V0 层的时钟门控关闭; 1: V0 层的时钟门控打开。	0x0
[28]	-	reserved	保留。	0x0



[27]	RW	g0_ck_gt_en	G0 层时钟门控使能,既低功耗模式寄存器。 0: G0 层的时钟门控关闭; 1: G0 层的时钟门控打开。	0x0
[26:0]	-	reserved	保留。	0x0

VOINTSTA

VOINTSTA 为 VDP 中断状态, 只读寄存器。

Offset Address: 0x0004 Total Reset Value: 0x0000_0444

Bits	Access	Name	Description	Reset
[31:23]	-	reserved	保留。	0x000
[22]	RO	dhd0vtthd4_int	HD0 通道垂直时序中断 4 使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[21]	-	reserved	保留。	0x0
[20]	RO	v0_tunl_int	V0 低延时错误中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[19:4]	-	reserved	保留。	0x0
[3]	RO	dhd0uf_int	HD0 通道低带宽告警中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RO	dhd0vtthd3_int	HD0 通道垂直时序中断 3 使能。 0: 屏蔽中断; 1: 使能中断。	0x1
[1]	RO	dhd0vtthd2_int	HD0 通道垂直时序中断 2 使能。 0: 屏蔽中断; 1: 使能中断。	0x0



Bits	Access	Name	Description	Reset
[0]	RO	dhd0vtthd1_int	HD0 通道垂直时序中断 1 使能。 0: 屏蔽中断; 1: 使能中断。	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
[31:23]	-	reserved	保留。	0x00
[22]	WC	dhd0vtthd4_clr	HD0 通道垂直时序中断 4 使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[21]	-	reserved	保留。	0x0
[20]	WC	v0_tunl_clr	V0 低延时错误中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[19:4]	-	reserved	保留。	0x0
[3]	WC	dhd0uf_clr	HD0 通道低带宽告警中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	WC	dhd0vtthd3_clr	HD0 通道垂直时序中断 3 使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	WC	dhd0vtthd2_clr	HD0 通道垂直时序中断 2 使能。 0: 屏蔽中断; 1: 使能中断。	0x0



Bits	Access	Name	Description	Reset
[0]	WC	dhd0vtthd1_clr	HD0 通道垂直时序中断 1 使能。 0: 屏蔽中断; 1: 使能中断。	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
[31:23]	-	reserved	保留。	0x00
[22]	RW	dhd0vtthd4_intmask	HD0 通道垂直时序中断 4 使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[21]	-	reserved	保留。	0x0
[20]	RW	v0_tunl_intmask	V0 低延时错误中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[19:4]	-	reserved	保留。	0x0
[3]	RW	dhd0uf_intmask	HD0 通道低带宽告警中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RW	dhd0vtthd3_intmask	HD0 通道垂直时序中断 3 使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	RW	dhd0vtthd2_intmask	HD0 通道垂直时序中断 2 使能。 0: 屏蔽中断;	0x0



Bits	Access	Name	Description	Reset
			1: 使能中断。	
[0]	RW	dhd0vtthd1_intmask	HD0 通道垂直时序中断 1 使能。 0: 屏蔽中断; 1: 使能中断。	0x0

VOINTSTA1

VOINTSTA1 为 VDP 中断状态 1, 只读寄存器。

Offset Address: 0x0014 Total Reset Value: 0x0000_0444

Bits	Access	Name	Description	Reset
[31:23]	-	reserved	保留。	0x00
[22]	RO	dhd0vtthd4_int	HD0 通道垂直时序中断 4 使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[21]	-	reserved	保留。	0x0
[20]	RO	v0_tunl_int	V0 低延时错误中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[19:4]	-	reserved	保留。	0x0
[3]	RO	dhd0uf_int	HD0 通道低带宽告警中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RO	dhd0vtthd3_int	HD0 通道垂直时序中断 3 使能。 0: 屏蔽中断; 1: 使能中断。	0x1
[1]	RO	dhd0vtthd2_int	HD0 通道垂直时序中断 2 使能。 0: 屏蔽中断;	0x0



Bits	Access	Name	Description	Reset
			1: 使能中断。	
[0]	RO	dhd0vtthd1_int	HD0 通道垂直时序中断 1 使能。 0: 屏蔽中断; 1: 使能中断。	0x0

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
[31:23]	-	reserved	保留。	0x00
[22]	WC	dhd0vtthd4_clr	HD0 通道垂直时序中断 4 使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[21]	-	reserved	保留。	0x0
[20]	WC	v0_tunl_clr	V0 低延时错误中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[19:4]	-	reserved	保留。	0x0
[3]	WC	dhd0uf_clr	HD0 通道低带宽告警中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	WC	dhd0vtthd3_clr	HD0 通道垂直时序中断 3 使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	WC	dhd0vtthd2_clr	HD0 通道垂直时序中断 2 使能。 0: 屏蔽中断;	0x0



Bits	Access	Name	Description	Reset
			1: 使能中断。	
[0]	WC	dhd0vtthd1_clr	HD0 通道垂直时序中断 1 使能。 0: 屏蔽中断; 1: 使能中断。	0x0

VOINTMSK1

VOINTMSK1 为 VDP 中断屏蔽寄存器 1。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: 0x001C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:23]	-	reserved	保留。	0x00
[22]	RW	dhd0vtthd4_intmask	HD0 通道垂直时序中断 4 使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[21]	-	reserved	保留。	0x0
[20]	RW	v0_tunl_intmask	V0 低延时错误中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[19:4]	-	reserved	保留。	0x0000
[3]	RW	dhd0uf_intmask	HD0 通道低带宽告警中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RW	dhd0vtthd3_intmask	HD0 通道垂直时序中断 3 使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	RW	dhd0vtthd2_intmask	HD0 通道垂直时序中断 2 使能。	0x0



Bits	Access	Name	Description	Reset
			0: 屏蔽中断; 1: 使能中断。	
[0]	RW	dhd0vtthd1_intmask	HD0 通道垂直时序中断 1 使能。 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
[31:20]	-	reserved	保留。	0x000
[19:16]	RW	mixer_prio4	CBM Mixer1 的叠加层优先级配置, 表示优先级 4 的驱动层。 0x0: 表示没有层驱动;	0x0



Bits	Access	Name	Description	Reset
			0x1: V0; 0x2: G0; 其他: 保留。	
[15:12]	RW	mixer_prio3	CBM Mixer1 的叠加层优先级配置, 表示优先级 3 的驱动层。 0x0: 表示没有层驱动; 0x1: V0; 0x2: G0; 其他: 保留。	0x0
[11:8]	RW	mixer_prio2	CBM Mixer1 的叠加层优先级配置, 表示优先级 2 的驱动层。 0x0: 表示没有层驱动; 0x1: V0; 0x2: G0; 其他: 保留。	0x0
[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: 0x6000_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
[29:11]	-	reserved	保留。	0x0
[10]	RW	symmetraic_splitti ng_en	虚拟通道左右交织模式使能(给 HFIR 使用)。 0: 禁止; 1: 使能。	0x0
[9]	RW	hsp_en	V0 中 hsharpen 模块开关信号。 0: 关闭 hsharpen 模块; 1: 打开 hsharpen 模块。	0x0
[8]	RW	mir_en	mirror 使能开关。 0: 关闭; 1: 开启 mirror。	0x0
[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	0x0



			动清零。	
--	--	--	------	--

V0_IRESO

V0_IRESO 为 V0 输入分辨率读取寄存器，

Offset Address: 0x1010 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	ih	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	iw	宽度，像素单位。实际宽度减 1。	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。	0x0000



			以帧高度为参考，以行为单位。	
[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。	0x0000
[15:0]	RW	video_xfpos	视频内容列起始坐标。	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。	0x0000
[15:0]	RW	video_xlpos	视频内容列结束坐标。	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_RIMWIDTH

V0_RIMWIDTH 为视频层 RIM 宽度寄存器，为非即时寄存器。

Offset Address: 0x10A0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4:0]	RW	v0_rim_width	所有区域边框的厚度。	0x00

V0_RIMCOLO

V0_RIMCOLO 为视频层高 32 分区 RIM 颜色 0 寄存器，为非即时寄存器。

Offset Address: 0x10A4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29:20]	RW	v0_rim_y0	RIM 颜色 0,R 分量。	0x000
[19:10]	RW	v0_rim_u0	RIM 颜色 0,G 分量。	0x000
[9:0]	RW	v0_rim_v0	RIM 颜色 0,B 分量。	0x000



V0_RIMCOL1

V0_RIMCOL1 为视频层高 32 分区 RIM 颜色 1 寄存器，为非即时寄存器。

Offset Address: 0x10A8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29:20]	RW	v0_rim_y1	RIM 颜色 1,R 分量。	0x000
[19:10]	RW	v0_rim_u1	RIM 颜色 1,G 分量。	0x000
[9:0]	RW	v0_rim_v1	RIM 颜色 1,B 分量。	0x000

V0_HIPP_CSC_CTRL

V0_HIPP_CSC_CTRL 为 HIPP_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_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: 0x0008_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19]	-	reserved	保留。	0x1



Bits	Access	Name	Description	Reset
[18]	RW	out_pro	输出逐隔行。 0: 隔行; 1: 逐行;	0x0
[17:16]	RW	out_fmt	输出数据格式。 00: YUV420; 01: YUV422; 10: YUV444; 其他: 保留。	0x0
[15:0]	RW	out_height	垂直亮度输出高度。	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。

缩放比率 = 输入宽度/输出宽度。

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
[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_0012

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000010
[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。	0x1
[0]	-	reserved	保留。	0x0

V0_HSPCFG10

V0_HSPCFG10 为 V0 水平锐化配置寄存器 10。

Offset Address: 0x1624 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27:20]	RW	hsp_peak_ratio	亮度增强比率, u(1.7)数据格式, 无符号数, 1bit 整数位, 7bit 小数位。	0x00



Bits	Access	Name	Description	Reset
[19:0]	-	reserved	保留。	0x0000

G0_CTRL

G0_CTRL 为 G0 控制寄存器，为非即时寄存器。

Offset Address: 0x7000 Total Reset Value: 0x6000_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]	RW	nosec_flag	总线接口标示操作安全属性信号，安全 CPU 下配置。 0: 安全属性; 1: 非安全属性。	0x1
[29:8]	-	reserved	保留。	0x200000
[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_IRESO

G0_IRESO 为输入分辨率寄存器。

Offset Address: 0x7010 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	ih	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	iw	宽度，像素单位。实际宽度减 1。	0x0000

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 HIPP_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; 1: 使能。	0x0

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
[28:19]	-	reserved	保留。	0x000
[18]	RW	cbar_mode	color bar 模式选择。 0: 水平 color bar; 1: 垂直 color bar。	0x0
[17]	-	reserved	保留。	0x0
[16]	RW	twochn_en	双通道处理使能寄存器。 0: 双通道处理不使能; 1: 双通道处理使能。	0x0
[15:14]	RO	reserved	保留。	0x0
[13:5]	-	reserved	保留。	0x000
[4]	RW	iop	逐行或者隔行显示, 非即时寄存器。 0: 隔行显示; 1: 逐行显示。	0x0
[3:1]	RO	reserved	保留。	0x0
[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	0x0



Bits	Access	Name	Description	Reset
			动清零。	

DHD0_VSYNC1

DHD0_VSYNC1 为显示通道的垂直时序控制 1 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	vo_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。 vo 接口时序中 $v_{total} = v_{act} + v_{bb} + v_{fb}$ ， vpw 不计入 v_{total}	0x0028
[15:0]	RW	vo_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。 VO 接口时序中 $v_{total} = v_{act} + v_{bb} + v_{fb}$ ， vpw 不计入 v_{total}	0x0437

DHD0_VSYNC2

DHD0_VSYNC2 为显示通道的垂直时序控制 2 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	vo_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。 VO 接口时序中 $v_{total} = v_{act} + v_{bb} +$	0x0003



			vfb, vpw 不计入 vtotal	
--	--	--	---------------------	--

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	vo_hbb	水平消隐后肩，单位为像素。 VO 接口时序中 $htotal = hact + hbb + hfb$, hpw 不计入 htotal	0x00BF
[15:0]	RW	vo_hact	活动区域水平像素数。 VO 接口时序中 $htotal = hact + hbb + hfb$, hpw 不计入 htotal	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	vo_hfb	水平消隐前肩，单位为像素。 VO 接口时序中 $htotal = hact + hbb + hfb$, hpw 不计入 htotal	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	vo_bvbb	隔行输出时：底场垂直消隐后肩垂直脉冲的宽度。 已包含 VPW 宽度。	0x0000
[15:0]	RW	vo_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	vo_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
[31:24]	-	reserved	保留。	0x08
[23:16]	RW	vo_vpw	VO 垂直脉冲宽度减 1；独立配置，不参与 vttotal 计算。 vo 接口时序中 vttotal = vact + vbb + vfb, vpw 不计入 vttotal	0x04
[15:0]	RW	vo_hpw	VO 水平脉冲宽度减 1。独立配置，不参与	0x002B



			htotal 计算。 vo 接口时序中 htotal = hact + hbb + hfb, hpw 不计入 htotal	
--	--	--	--	--

DHD0_VTTHD3

DHD0_VTTHD3 为垂直时序门限值 3 (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: 0xD020 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	thd4_mode	门限 4 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	vtmgthd4	垂直时序门限值 4, 用于 DHD0 的触发。	0x0001
[15]	RW	thd3_mode	门限 3 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	vtmgthd3	垂直时序门限值 3, 当垂直时序计数器到达该门限时触发 VOINTSTA[dhdvtthd3_int]中断。	0x0001

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[dhdvtthd_int1]中断。	0x0001

DHD0_PRECHARGE_THD

DHD0_PRECHARGE_THD 为 TCON 预充电阈值寄存器。

Offset Address: 0xD02C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:21]	-	reserved	保留。	0x000
[20]	RW	vsync_te_mode	垂直时序从模式使能寄存器。 0: VFB(vertical front blanking)到 VBB(vertical back blanking)的切换由时序配置控制; 1: VFB 到 VBB 的切换由外部 vsync 信号控制。	0x0



[19:0]	-	reserved	保留。	0x00000
--------	---	----------	-----	---------

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
[31:8]	-	reserved	保留。	0x00A303
[7:0]	RW	start_pos	表示从消隐区开始的的第几行产生该 start 信号。从 0 开始计数。	0x0A

DHD0_RGB_FIX_CTRL

DHD0_RGB_FIX_CTRL 为 INTF 输出颜色固定寄存器。

Offset Address: 0xD044 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30]	RW	rgb_fix_mux	输出颜色固定选择寄存器。 0: 不固定; 1: 固定颜色, 可配。	0x0
[29:20]	RW	fix_r	输出颜色 R 分量固定, 可配。	0x000
[19:10]	RW	fix_g	输出颜色 G 分量固定, 可配。	0x000
[9:0]	RW	fix_b	输出颜色 B 分量固定, 可配。	0x000

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。 0x0: BT.1120; 0x1: BT.656; 0x2: RGB; 其他: 保留。	0x0
[27:12]	-	reserved	保留。	0x00
[11:8]	RW	bt_sel	BT.1120/656 数据选择(默认 0)。 0x0: DHD0 BT.1120/656; 其他: 保留。	0x0
[7:4]	RW	rgb_sel	RGB 数据选择(默认 0)。 0x0: DHD0 RGB; 其他: 保留。	0x0
[3:0]	RW	mipi_sel	MIPI 数据选择(默认 0)。 0x0: DHD0 MIPI; 其他: 保留。	0x0

INTF_BT_CTRL

INTF_BT_CTRL 为接口通道配置寄存器。

Offset Address: 0xD200 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	-	reserved	保留。	0x0
[30]	RW	dfir_en	dfir 使能信号。 0: 禁止; 1: 使能。	0x0
[29:20]	-	reserved	保留。	0x000
[19]	RW	yc_mode	BT.1120 YC 输出顺序。	0x0



Bits	Access	Name	Description	Reset
			0: YU/YV; 1: UY/VY。	
[18]	RW	uv_mode	BT.1120 C 分量输出顺序。 0: UVUV 顺序; 1: VUVU 顺序。	0x0
[17]	RW	bit_inv	BT.1120/BT.656 输出 bit 反序。 0: 正序; 1: 反序。	0x0
[16]	RW	data_width	BT.1120 输出精度。 0: 10bit; 1: 8bit。	0x0
[15:0]	-	reserved	保留。	0x0000

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

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

BT_DITHER_CTRL

BT_DITHER_CTRL 为 Dither 控制寄存器。

Offset Address: 0xD280 Total Reset Value: 0x0000_1600

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。 000: 5;	0x5



Bits	Access	Name	Description	Reset
			001: 6; 010: 7; 011: 8; 100: 9; 101: 10。 其他: 保留。	
[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。 其他: 保留。	0x4
[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: 关闭; 1: 使能。	0x0
[5]	RW	dither_en	Dither 使能选择。 0: 模块不使能, 直接 bypass。 1: 模块使能。	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: 关闭; 1: 使能。	0x0
[2]	RW	dither_domain_mode	Dither 时域和空域模式寄存器。 0: 空域;	0x0



Bits	Access	Name	Description	Reset
			1: 时域和空域的组合。	
[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

BT_DITHER_THR

BT_DITHER_THR 为 Dither 门限值寄存器。

Offset Address: 0xD2C4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	dither_thr_max	做 dither 处理的像素上限值。	0x0000
[15:0]	RW	dither_thr_min	做 dither 处理的像素下限值。	0x0000

INTF_RGB_CTRL

INTF_RGB_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]	RO	reserved	保留。	0x0
[30]	RW	dfir_en	dfir 使能信号。 0: 禁止; 1: 使能。	0x0
[29:28]	-	reserved	保留。	0x0
[27]	RW	dummy_position_sel	串行 4 周期 dummy 位置配置。 0: dummy_RGB	0x0



Bits	Access	Name	Description	Reset
			1: RGB_dummy	
[26]	RW	rgb_serial_perd	RGB 串行输出单像素时钟周期数。 0: 3 个周期; 1: 4 个周期。	0x0
[25]	RW	rgb_comp_order	RGB 并行输出顺序。 0: RGB; 1: BGR。	0x0
[24]	RW	rgb_bit_inv	RGB 输出 bit 反序。 0: 正序; 1: 反序。	0x0
[23:20]	RW	rgb_format	RGB 输出格式。 0x0: 并行 RGB888 输出; 0x1: 并行 RGB565 输出; 0x2: 并行 RGB666 输出; 0x3: 串行 8bit 输出; 0x4: 串行 6bit 输出。 其他: 保留。	0x0
[19:0]	-	reserved	保留。	0x00000

INTF_RGB_UPD

INTF_RGB_UPD 为 INTF_RGB 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



RGB_CSC_IDC

RGB_CSC_IDC 为色彩空间转换输入直流分量寄存器，为即时寄存器。

Offset Address: 0xD320 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:23]	-	reserved	保留。	0x000
[22]	RW	csc_en	CSC 使能控制信号。 0: CSC 不使能; 1: CSC 使能。	0x0
[21:11]	RW	cscidc1	输入分量 1 直流参数, MSB 为符号位。补码表示。	0x000
[10:0]	RW	cscidc0	输入分量 0 直流参数, MSB 为符号位。补码表示。	0x000

RGB_DITHER_CTRL

RGB_DITHER_CTRL 为 Dither 控制寄存器。

Offset Address: 0xD380 Total Reset Value: 0x0000_1600

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。 000: 5; 001: 6; 010: 7; 011: 8; 100: 9; 101: 10; 其他: 保留。	0x5
[9:7]	RW	i_data_width_dither	输入像素数据位宽。	0x4



Bits	Access	Name	Description	Reset
			000: 8; 001: 9; 010: 10; 011: 11; 100: 12; 其他: 保留。	
[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: 使能; 0: 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



RGB_DITHER_THR

RGB_DITHER_THR 为 Dither 门限值寄存器。

Offset Address: 0xD3C4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	dither_thr_max	做 dither 处理的像素上限值。	0x0000
[15:0]	RW	dither_thr_min	做 dither 处理的像素下限值。	0x0000

MIPI_DITHER_THR

MIPI_DITHER_THR 为 Dither 门限值寄存器。

Offset Address: 0xD844 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	dither_thr_max	做 dither 处理的像素上限值。	0x0000
[15:0]	RW	dither_thr_min	做 dither 处理的像素下限值。	0x0000

MIPI_GAMMA_CTRL

MIPI_GAMMA_CTRL 为 Gamma 控制寄存器。

Offset Address: 0xD900 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	gamma_en	Gamma 使能。 0: 禁止; 1: 使能。	0x0

MAC_CTRL

MAC_CTRL 为 MAC&MASTER 控制寄存器。

Offset Address: 0x10004 Total Reset Value: 0x0000_0101



Bits	Access	Name	Description	Reset
[31:9]	-	reserved	保留。	0x000000
[8]	RW	mid_enable	多 ID 模式。 0: 单 ID 模式; 1: 多 ID 模式。	0x1
[7:4]	RW	arb_mode	ARB 模式。 0x0: 轮询模式; 0x1: 仲裁连提模式; 其他: 保留。	0x0
[3:0]	RW	split_mode	拆分模式。 0x0: 128B 拆分; 0x1: 256B 拆分; 0x2: 1K 拆分; 0x3: 2K 拆分; 0x4: 4K 拆分; 其他: 4K 拆分。	0x1

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: 0x0022_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x002
[19]	RW	mrg_enable	多区域使能。 0: 不使能; 1: 使能。	0x0
[18:17]	-	reserved	保留。	0x1
[16]	RW	mute_en	fdr mute 使能寄存器。	0x0



Bits	Access	Name	Description	Reset
			0: 禁止; 1: 使能。	
[15:14]	-	reserved	保留。	0x0
[13]	RW	chm_copy_en	色度重复读使能寄存器。 0: 禁止; 1: 使能。	0x0
[12:7]	-	reserved	保留。	0x00
[6:4]	RW	lm_rmode	亮度逐隔行读取模式寄存器。 000: 读取模式跟接口绑定; 001: 读取逐行; 010: 读取顶场; 011: 读取底场; 100~111: 保留, 等同于 000。	0x0
[3]	-	reserved	保留。	0x0
[2:0]	RW	chm_rmode	色度逐隔行读取模式寄存器。 000: 读取模式跟接口绑定; 001: 读取逐行; 010: 读取顶场; 011: 读取底场; 100~111: 保留, 等同于 000。	0x0

VID_MAC_CTRL

VID_MAC_CTRL 为 VID FDR 申请控制寄存器, 为非即时寄存器。

Offset Address: 0x10204 Total Reset Value: 0x8000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x20000000



Bits	Access	Name	Description	Reset
[1:0]	RW	req_ctrl	连续申请 16 个 burst 个数控制寄存器。 00: 一次连续申请 1 个 16burst; 01: 一次连续申请 2 个 16burst; 10: 一次连续申请 4 个 16burst。 其他: 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]	-	reserved	保留。	0x0
[2:0]	-	reserved	保留。	0x0

VID_MUTE_ALPHA

VID_MUTE_ALPHA 为视频层 FDR MUTE 颜色 alpha 寄存器，为非即时寄存器。

Offset Address: 0x10214 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	mute_alpha	A 分量。	0x00



VID_MUTE_BK

VID_MUTE_BK 为视频层 FDR MUTE 颜色寄存器，为非即时寄存器。

Offset Address: 0x1021C Total Reset Value: 0x0008_0200

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29:20]	RW	mute_y	Y 分量。	0x000
[19:10]	RW	mute_cb	Cb 分量。	0x200
[9:0]	RW	mute_cr	Cr 分量。	0x200

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
[31:24]	-	reserved	保留。	0x00
[23]	RW	compact_req_mode	紧凑段压缩的数据申请方式。 0: 按段申请; 1: 按行申请。	0x0
[22]	RW	compact_en	输入段压缩类型。 0: 非紧凑排布; 1: 紧凑排布。	0x0
[21:20]	RW	dcmp_en	输入图像压缩类型。 00: 非压缩; 01: 段压缩; 10: 行压缩; 11: 帧压缩。	0x0
[19:3]	-	reserved	保留。	0x00000



Bits	Access	Name	Description	Reset
[2:0]	RW	data_type	输入数据格式类型。 001: YUV400; 010: YUV420; 011: YUV422; 100: YUV444; 101: RGB888; 110: ARGB8888; 其他: 保留, 默认 YUV422。	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。	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

VID_ADDR_H

VID_ADDR_H 为亮度地址高位，为非即时寄存器。

Offset Address: 0x10250 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	vhdaddr_h	在 3D 模式下为左眼亮度地址寄存器。TILE 10bit 时，为高 8bit 亮度度地址。2D 时为亮度申请地址。	0x00000000

VID_ADDR_L

VID_ADDR_L 为亮度地址低位，为非即时寄存器。

Offset Address: 0x10254 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	vhdaddr_l	在 3D 模式下为左眼亮度地址寄存器。TILE 10bit 时，为高 8bit 亮度度地址。2D 时为亮度申请地址。	0x00000000

VID_CADDR_H

VID_CADDR_H 为色度地址高位，为非即时寄存器。

Offset Address: 0x10258 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	vhdcaddr_h	在 3D 模式下为左眼色度地址寄存器。TILE 10bit 时, 为高 8bit 色度度地址。2D 时为亮度申请地址。	0x00000000

VID_CADDR_L

VID_CADDR_L 为色度地址低位, 为非即时寄存器。

Offset Address: 0x1025C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	vhdcaddr_l	在 3D 模式下为左眼色度地址寄存器。TILE 10bit 时, 为高 8bit 色度度地址。2D 时为亮度申请地址。	0x00000000

VID_STRIDE

VID_STRIDE 为 stride 配置寄存器。TILE 10bit 时, 为 8bit stride。为非即时寄存器。

Offset Address: 0x10270 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	chm_stride	色度 stride 配置寄存器。TILE 10bit 时, 为 8bit stride。为非即时寄存器。	0x0000
[15:0]	RW	lm_stride	亮度 stride 配置寄存器。TILE 10bit 时, 为 8bit stride。为非即时寄存器。	0x0000

VID_HEAD_STRIDE

VID_HEAD_STRIDE 为视频层头信息 stride 寄存器。

为非即时寄存器。

Offset Address: 0x10278 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	RW	chm_head_stride	视频层色度头信息 stride, 16bit 对齐, 非即时寄存器。	0x0000
[15:0]	RW	lm_head_stride	视频层亮度头信息 stride, 16bit 对齐, 非即时寄存器。	0x0000

VID_HEAD_ADDR_H

VID_HEAD_ADDR_H 为亮度头地址高位, 为非即时寄存器。

Offset Address: 0x10290 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	h_vhdaddr_h	亮度头地址高位信息。	0x00000000

VID_HEAD_ADDR_L

VID_HEAD_ADDR_L 为亮度头地址低位, 为非即时寄存器。

Offset Address: 0x10294 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	h_vhdaddr_l	亮度头地址低位信息。	0x00000000

VID_HEAD_CADDR_H

VID_HEAD_CADDR_H 为色度头地址高位, 为非即时寄存器。

Offset Address: 0x10298 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	h_vhdcaddr_h	色度头地址高位信息。	0x00000000



VID_HEAD_CADDR_L

VID_HEAD_CADDR_L 为色度头地址低位，为非即时寄存器。

Offset Address: 0x1029C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	h_vhdcaddr_l	色度头地址低位信息。	0x00000000

VID_TUNL_CTRL

VID_TUNL_CTRL 为 TUNL 低延时控制寄存器。

Offset Address: 0x102C0 Total Reset Value: 0x0001_0000

Bits	Access	Name	Description	Reset
[31]	RW	rtunl_en	读通道正向 tunl 使能。 0: 禁止; 1: 使能。	0x0
[30]	RW	tunl_uf	TUNL 低带宽保护使能。 0: 使能保护; 1: 解除保护。	0x0
[29:24]	-	reserved	保留。	0x00
[23:8]	RW	tunl_thd	VDP 的启动行号。	0x0100
[7:0]	RW	tunl_interval	行号申请间隔，单位是 1024 个周期，即 (tunl_interval+1)*1024	0x00

VID_TUNL_CROP

VID_TUNL_CROP 为视频层低延时 crop 掉的行。

Offset Address: 0x102C4 Total Reset Value: 0x0000_0000

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



Bits	Access	Name	Description	Reset
[15:0]	RW	tunl_crop_line	低延时 crop 的 y 的起点。	0x0000

VID_TUNL_ADDR_H

VID_TUNL_ADDR_H 为 TUNL 低延时行号地址高 bit。

Offset Address: 0x102D0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tunl_addr_h	TUNL 低延时行号地址高 bit。	0x00000000

VID_TUNL_ADDR_L

VID_TUNL_ADDR_L 为 TUNL 低延时行号地址低 bit。

Offset Address: 0x102D4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tunl_addr_l	TUNL 低延时行号地址低 bit。	0x00000000

GFX_READ_CTRL

GFX_READ_CTRL 为 GFX FDR 读取控制寄存器，为非即时寄存器。

Offset Address: 0x10E00 Total Reset Value: 0x0000_1000

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	保留。	0x00000
[12]	RW	fdr_ck_gt_en	门控使能。 0: 禁止; 1: 使能。	0x1
[11]	-	reserved	保留。	0x0
[10]	RW	mute_en	fdr mute 使能寄存器。	0x0



Bits	Access	Name	Description	Reset
			0: 禁止; 1: 使能。	
[9:2]	-	reserved	保留。	0x000
[1:0]	RW	read_mode	数据读取模式。 0: 根据接口读取方式自动选择(逐行显示时逐行读, 隔行显示时隔行读); 1: 强制按照逐行读取。	0x0

GFX_MAC_CTRL

GFX_MAC_CTRL 为 VID FDR 申请控制寄存器, 为非即时寄存器。

Offset Address: 0x10E04 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25:24]	RW	req_ld_mode	低延时模式寄存器。 0: 非低延时; 1: 图形低延时。	0x0
[23:2]	-	reserved	保留。	0x000000
[1:0]	RW	req_ctrl	连续申请 16 个 burst 个数控制寄存器。 00: 一次连续申请 1 个 16burst; 01: 一次连续申请 2 个 16burst; 10: 一次连续申请 4 个 16burst。 其他: 保留。	0x0

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_0001



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。 其他: 保留。	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 $Keymin \leq Pixel \leq Keymax$ 时, 处理为关键色; 1: 满足 $Pixel \leq Keymin$ 或者 $Pixel \geq Keymax$ 时处理为关键色。	0x0
[3:2]	-	reserved	保留。	0x0
[1]	RW	palpha_en	像素 alpha 使能。 0: 禁止; 1: 使能。	0x0
[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。	0x1



GFX_MUTE_ALPHA

GFX_MUTE_ALPHA 为图形层 FDR MUTE 颜色 alpha 寄存器，为非即时寄存器。

Offset Address: 0x10E10 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	mute_alpha	A 分量。	0x00

GFX_MUTE_BK

GFX_MUTE_BK 为图形层 FDR MUTE 颜色寄存器，为非即时寄存器。

Offset Address: 0x10E14 Total Reset Value: 0x0008_0200

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29:20]	RW	mute_y	Y 分量。	0x000
[19:10]	RW	mute_cb	Cb 分量。	0x200
[9:0]	RW	mute_cr	Cr 分量。	0x200

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
[31:20]	-	reserved	保留。	0x000
[19:8]	-	reserved	保留。	0x000
[7:0]	RW	ifmt	输入数据格式。 0x00: CLUT 1bpp; 0x10: CLUT 2bpp; 0x20: CLUT 4bpp; 0x42: RGB565; 0x48: ARGB4444; 0x49: ARGB1555; 0x50: RGB888(24bpp); 其他: 保留。	0x00

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_ADDR_H

GFX_ADDR_H 为图形层高 bit 地址寄存器。在有水平像素偏移的情况下，地址计算参考 G0SFPOS 说明。

Offset Address: 0x10E50 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	gfx_addr_h	surface 帧 buffer 地址。	0x00000000

GFX_ADDR_L

GFX_ADDR_L 为图形层低 bit 地址寄存器。在有水平像素偏移的情况下，地址计算参考 G0SFPOS 说明。

Offset Address: 0x10E54 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	gfx_addr_l	surface 帧 buffer 地址。	0x00000000

GFX_STRIDE

GFX_STRIDE 为图形层的 stride 寄存器。

Offset Address: 0x10E60 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	surface_stride	帧 buffer 的 stride。	0x0000

GFX_DCMP_ADDR_H

GFX_DCMP_ADDR_H 为图形层左眼压缩高位地址寄存器。

Offset Address: 0x10E70 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	dcmp_addr_h	surface 帧 buffer 地址。	0x00000000

GFX_DCMP_ADDR_L

GFX_DCMP_ADDR_L 为图形层左眼压缩低位地址寄存器。

Offset Address: 0x10E74 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	dcmp_addr_l	surface 帧 buffer 地址。	0x00000000

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	colry 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	colry 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	colry key mask 的 R 分量。	0xFF
[15:8]	RW	key_g_msk	colry key mask 的 G 分量。	0xFF
[7:0]	RW	key_b_msk	colry key mask 的 B 分量。	0xFF



10.3 MIPI Rx

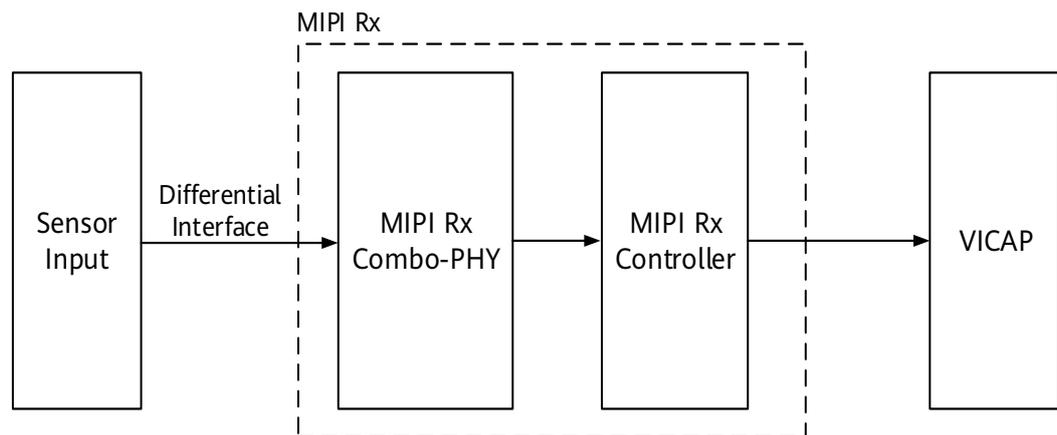
10.3.1 概述

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 10-20 所示。

图10-20 MIPI Rx 功能框图及在系统中的位置



10.3.2 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 8 条 Lane。
- Link 指 Lane 的分组, 每个分组中包含 4 条 Lane, MIPI_Rx 有 2 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-ver1.2
- 可同时支持 4 路 sensor 输入
- 单路最多支持 8-Lane MIPI D-PHY 接口，最大支持 2.5Gbps/Lane
- 单路最多支持 8-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 legacy/ YUV420 8-bit non-legacy/ YUV422 8-bit/ 数据类型的解析,支持最大 4 路 YUV virtual channel
- 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

10.3.3 功能描述

10.3.3.1 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 2 个 D-PHY，每个 PHY 都有 2 对差分随路时钟，4 对数据，每对时钟对应 2 对数据。因此 MIPI Rx 可以同时支持 1~4 路 sensor 输入。连接关系如表 10-6 所示。

表10-6 MIPI Rx 对接场景分类

Mode	PHY0		PHY1	
0	Sensor0(phy0clk0+lane0~lane7)			
7	Sensor0(phy0clk0+lane0~3)		Sensor1(phy1clk0+lane4~7)	
10	Sensor0(phy0clk0+lane0~3)		Sensor1(phy1clk0+lane4/6)	Sensor2(phy1clk1+lane5/7)
11	Sensor0(phy0clk0+lane0/2)	Sensor1(phy0clk1+lane1/3)	Sensor2(phy1clk0+lane4/6)	Sensor3(phy1clk1+lane5/7)
12	Sensor0(phy0clk0+lane0/2)	Sensor1(phy0clk1+lane1/3)	Sensor2(phy1clk0+lane4~7)	



- 当采用 Mode0 时，最大支持 8Lane 输入。
Sensor0 的时钟对接到 PHY0，该时钟会共享给 PHY1。
- 当采用 Mode7 时，最大支持 4Lane + 4Lane 输入。
 - Sensor0 的时钟对接到 PHY0 的 CLK0。
 - Sensor1 的时钟对接到 PHY1 的 CLK0。
- 当采用 Mode10 时，最大支持 4Lane + 2Lane + 2Lane 输入。
 - Sensor0 的时钟对接到 PHY0 的 CLK0。
 - Sensor1 的时钟对接到 PHY1 的 CLK0。
 - Sensor2 的时钟对接到 PHY1 的 CLK1。
- 当采用 Mode11 时，最大支持 2Lane + 2Lane + 2Lane + 2Lane 输入。
 - Sensor0 的时钟对接到 PHY0 的 CLK0。
 - Sensor1 的时钟对接到 PHY0 的 CLK1。
 - Sensor2 的时钟对接到 PHY1 的 CLK0。
 - Sensor3 的时钟对接到 PHY1 的 CLK1。
- 当采用 Mode12 时，最大支持 2Lane + 2Lane + 4Lane 输入。
 - Sensor0 的时钟对接到 PHY0 的 CLK0。
 - Sensor1 的时钟对接到 PHY0 的 CLK1。
 - Sensor2 的时钟对接到 PHY1 的 CLK0。

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 [10.3.4.1 MIPI 模式配置流程](#)和 [10.3.4.2 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 的同一个 PHY 中，即不可在 sensor 之间乱序，也不可在同一个 sensor 的不同 PHY 之间乱序。

例如：对于 Mode7 中 4Lane+4Lane 的输入模式，sensor0 与 sensor1 之间不可乱序，也不可在 sensor0 的两个 PHY 之间乱序（同 sensor1）。乱序仅仅可发生在 PHY0 的 Lane 之间，或者 PHY1 的 Lane 之间（同 sensor1）；对于 Mode11 中 2Lane + 2Lane + 2Lane + 2Lane 的输入模式，乱序仅仅可发生在同一个 sensor 所涉及的 Lane 中，即 PHY1 的第 0 和 2 个 Lane 之间可以乱序，不可与其他的 Lane 之间乱序。

MIPI Rx 支持的接口类型如[表 10-7](#)所示。



表10-7 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	1250MHz	2.5Gbps
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 输入都是接口最大支持 2.5Gbps/Lane，内部处理速度最大 600M*2pixels/s。

说明

Combo-PHY 支持的最大速率为 2.5Gbps/Lane，总共支持 8 Lane 同时传输。对于 mipi_rx 控制器的来说，通道的最大工作时钟为 600MHz。当配置为一拍一像素时最大通道带宽为 600Mpixels/s。

10.3.3.2 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 80~2500Mbps。D-PHY 可以工作在低功耗 (Low Power, LP) 和高速 (High Speed, HS) 两种模式下。当传输速率介于 1500Mbps~2500Mbps 之间时，D-PHY 支持相位自动校准功能。
- CSI-2

CSI-2 是针对摄像头的的数据协议，规定了主机与外设通信的数据包格式。

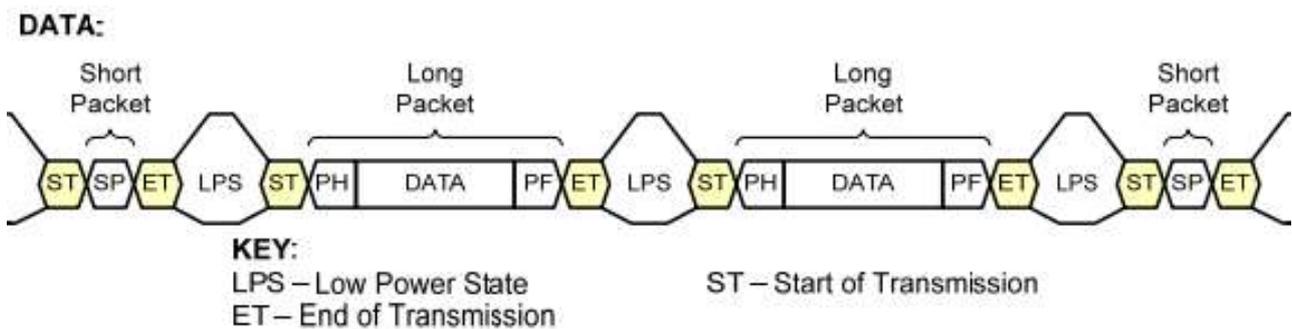
CSI-2 可以支持不同像素格式的图像应用，数据传输的最小粒度是字节。为增加 CSI-2 性能，可以选择数据 Lane 的数量，CSI-2 协议规定了发送端将像素数据打包成字节的机制，并指明多个数据 Lane 分配和管理的方式。字节数据以数据包的形式组织，数据包在 SoT 和 EoT 之间传输。接收端根据协议解析相应的数据包，恢复出原始的像素数据。

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 10-21 所示。PH 和 PF 分别表示 Packet Header 和 Packet Footer。

图10-21 CSI-2 数据包的传输机制



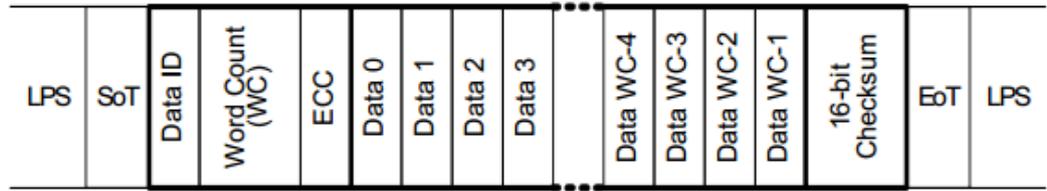
长包用于传输有效像素数据，分为五部分：Data ID, Word Count, ECC, PAYLOAD, Checksum。

- 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 10-22 所示。

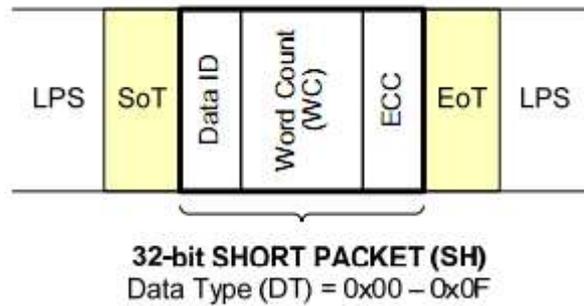


图10-22 CSI-2 长包的格式



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 10-23 所示。

图10-23 CSI-2 短包的格式

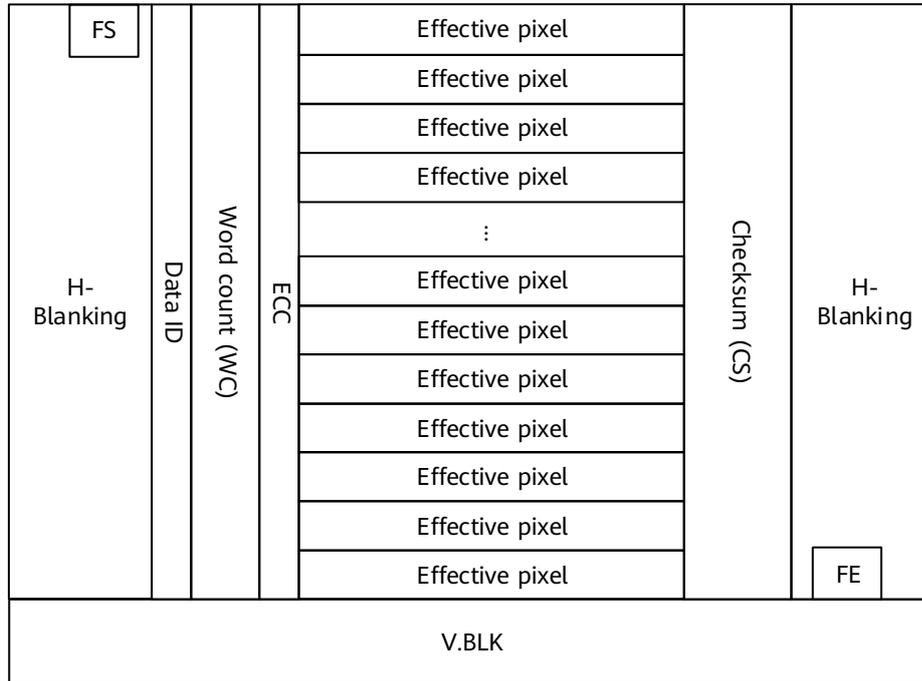


MIPI 接口线性模式

MIPI 接口的线性模式视频传输格式如图 10-24 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10-24 MIPI 接口图像格式



MIPI 接口宽动态模式

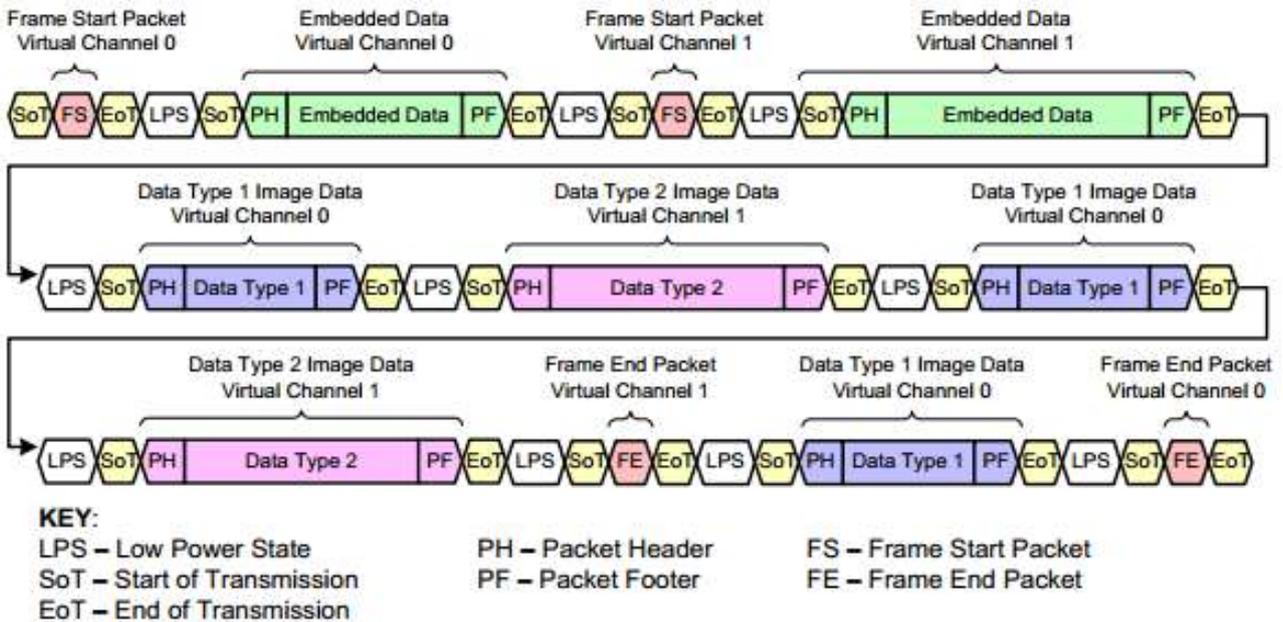
MIPI Rx 支持 3 种 MIPI 接口的宽动态 (WDR) 模式, 分别为:

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 10-25 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。

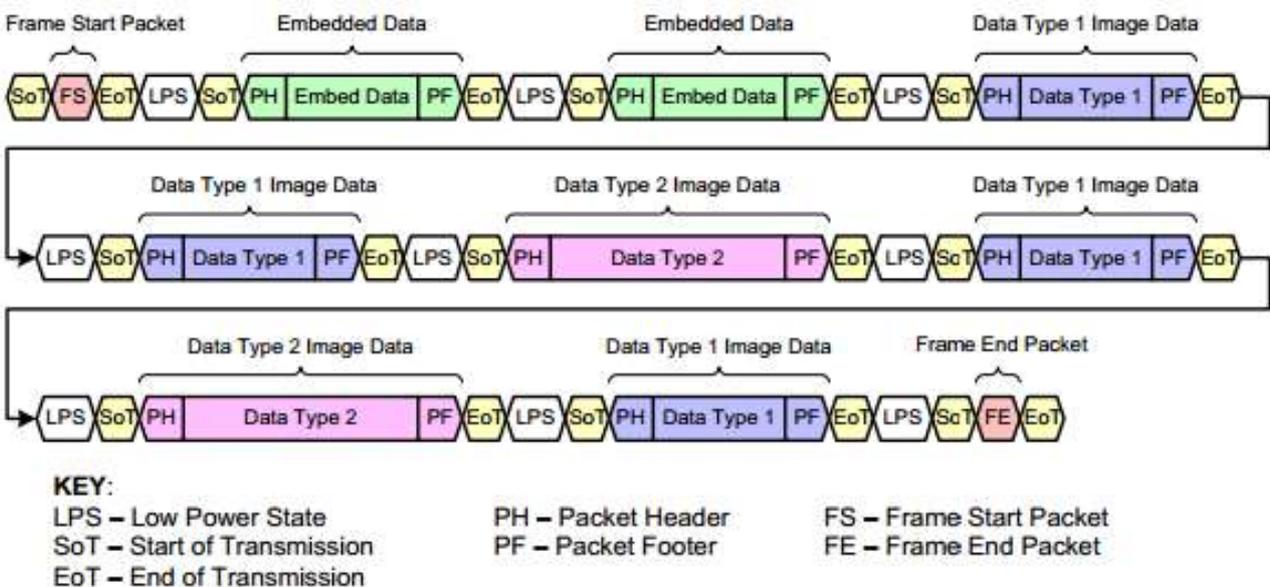


图10-25 MIPI 接口宽动态数据传输 (使用 VC)



使用 DT 的 WDR 传输方式如图 10-26 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。

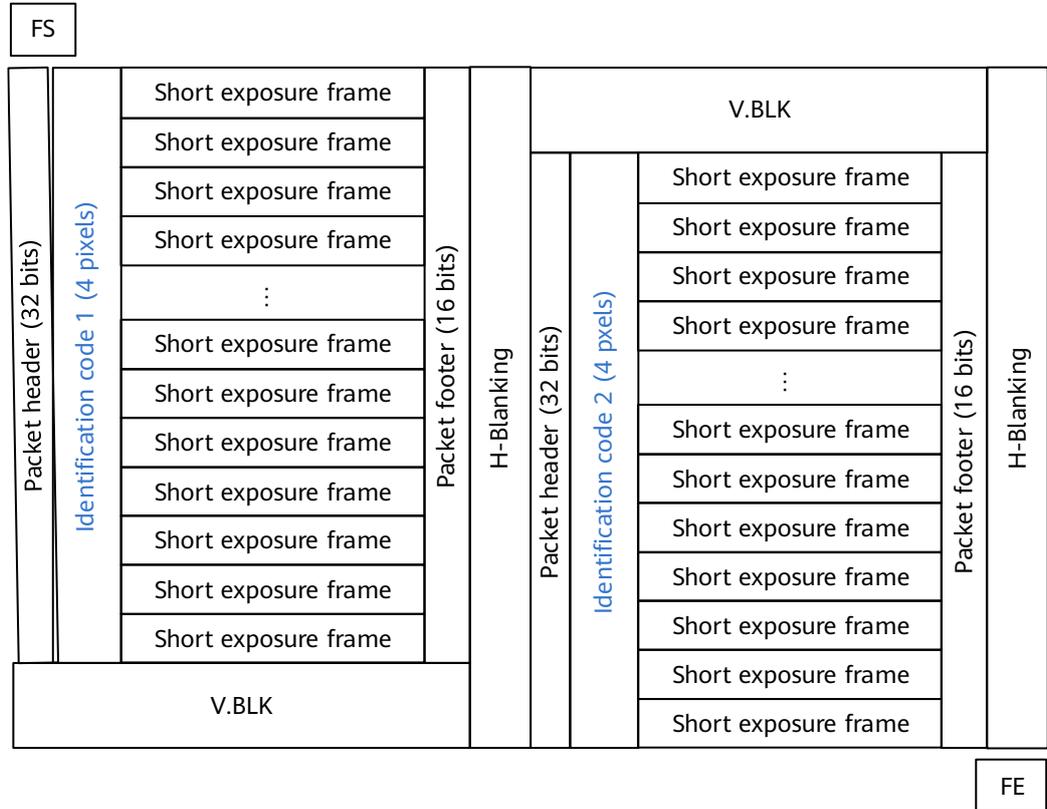
图10-26 MIPI 接口宽动态数据传输 (使用 DT)





MIPI DOL 宽动态传输格式如图 10-27 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10-27 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 10-8 所示。

表10-8 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

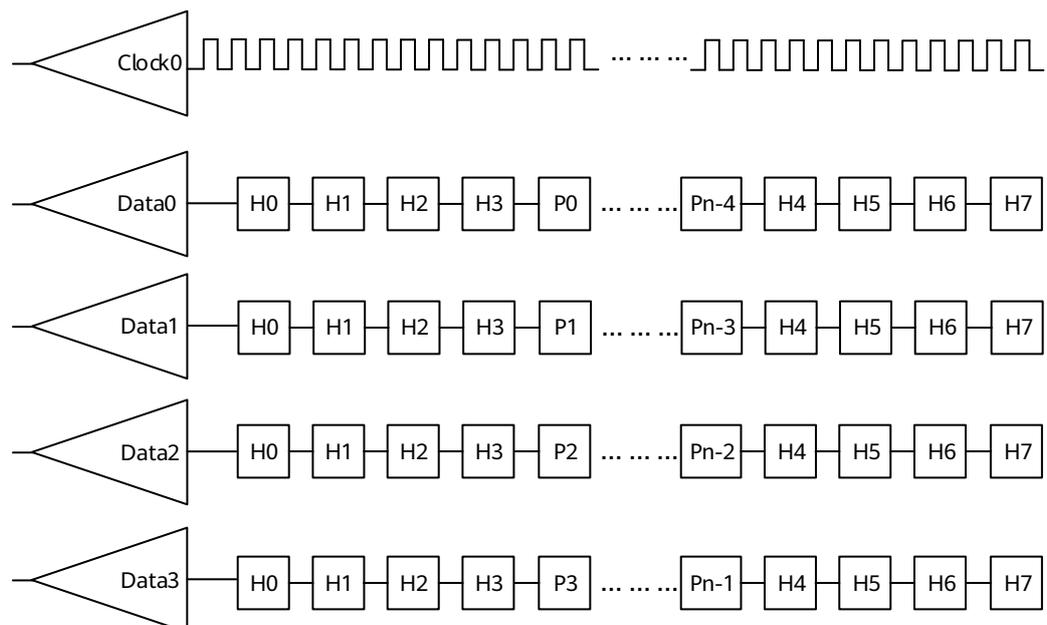
同步码前三个字段固定，第 4 个字段标识行场的起始或结束。第 4 个字段的值由图像传感器厂商确定，不同的厂商会使用不同的数值。表 10-9 是其中一种实现方式。

表10-9 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 code	8bit	80h	9Dh	ABh	B6h
	10bit	200h	274h	2ACh	2D8h
	12bit	800h	9D0h	AB0h	B60h
	14bit	2000h	2740h	2AC0h	2D80h
	16bit	8000h	9D00h	AB00h	B600h

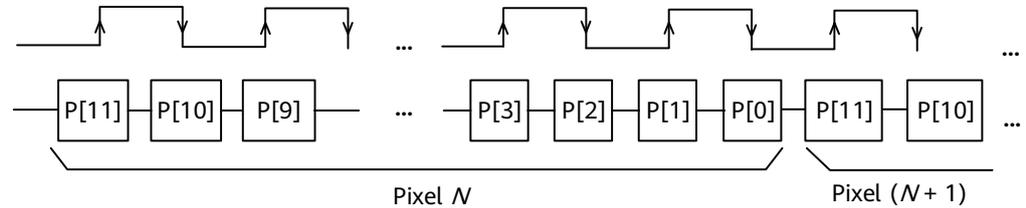
以 4 个 Lane 为例，LVDS 同步码和像素数据在各个 Lane 上传输方式如图 10-28 所示，图中 H 表示同步码，P 表示像素，H 和 P 的位宽与图像传感器输出单个像素的位宽一致。各个数据通道首先传输 4 个像素位宽的同步码，紧接着是像素数据，像素数据的分布与通道数有关。

图10-28 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 10-29 所示。

图10-29 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 10-30 所示。

图10-30 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		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 10-31 所示。

图10-31 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
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：

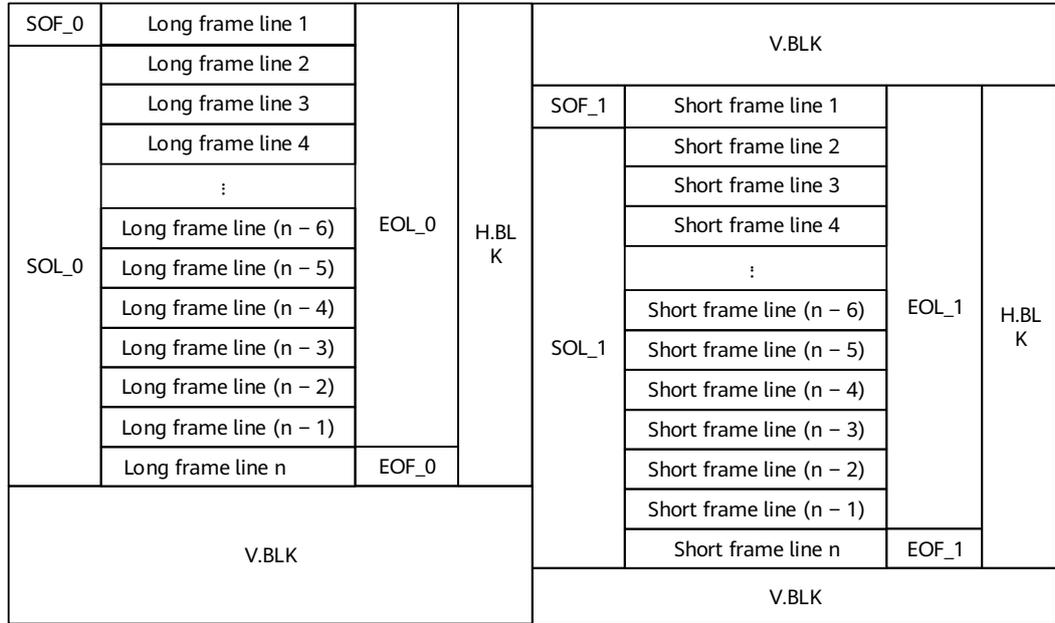
1. LVDS 宽动态模式 1：SOF-EOF 标识，长短曝光帧有独立的同步码。
2. LVDS 宽动态模式 2：SAV-EAV 标识，DOL 模式，四个字段的同步码，长短曝光帧有独立的同步码。
3. LVDS 宽动态模式 3：SAV-EAV 标识，DOL 模式，五个字段的同步码，长短曝光帧有独立的同步码。

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 10-32 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。

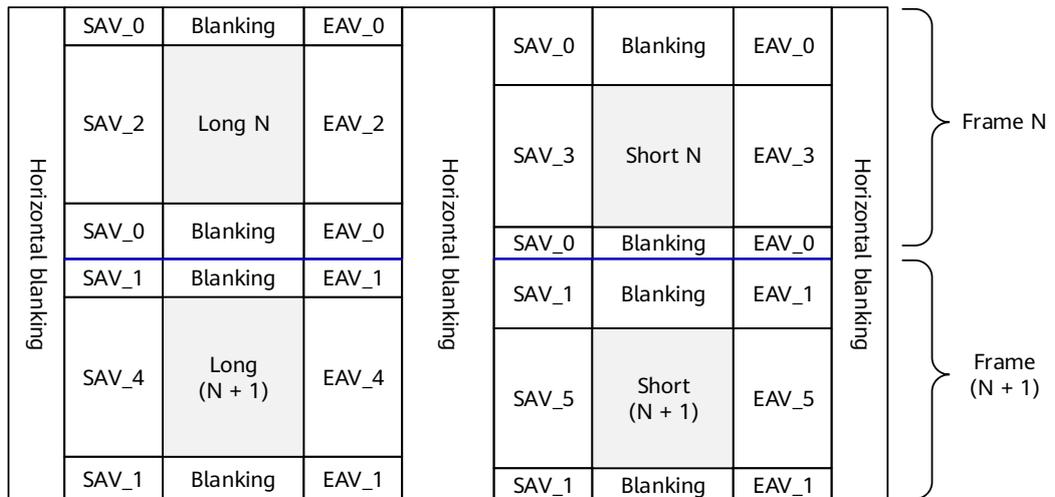
图10-32 LVDS 宽动态模式 1 (2 帧)



LVDS 宽动态模式 2 如图 10-33 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10-33 LVDS 宽动态模式 2 (2 帧)





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 10-31 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 10-34 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10-34 HiSPi Packetized-SP 数据格式

SOF		Effective pixel			
		Effective pixel			
		Effective pixel			
		Effective pixel			
		⋮			
	FLR	Effective pixel	EOL		
SOL		Effective pixel		CRC	H.BLK
		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 10-10 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10-10 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
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 10-28)。

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10-35 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										

10.3.4 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。

10.3.4.1 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 根据使用场景配置为 MIPI 模式。

步骤 3 配置 CRG 寄存器中的 PERI_CRG8528，打开 mipi_bus_cken、cil_cken 以及对应通道 PERI_CRG8536~PERI_CRG8560 的 mipi_pix_cken。配置 mipi 总线软复位，撤销复位；配置对应通道 mipi_pix_core_srst_req 复位，撤销复位。

步骤 4 配置 CRG 寄存器中的 PERI_CRG8464~PERI_CRG8488，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_CRG9305~PERI_CRG9329，选择 MIPI_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 均衡调节 (PHY_EQ_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

10.3.4.2 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 根据使用场景配置为 LVDS 模式。

步骤 3 配置 CRG 寄存器中的 PERI_CRG8528, 打开 mipi_bus_cken、cil_cken 以及对应通道 PERI_CRG8536~PERI_CRG8560 的 mipi_pix_clken。配置 mipi 总线软复位, 撤销复位; 配置对应通道 mipi_pix_core_srst_req 复位, 撤销复位。

步骤 4 配置 CRG 寄存器中的 PERI_CRG8464~PERI_CRG8488, 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_CRG9305~PERI_CRG9329, 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 均衡调节 (PHY_EQ_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

10.3.5 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 10-11 所示。

表10-11 MIPI Rx 寄存器偏移地址变量表

变量名称	取值范围	描述
P	0 ~ 1	PHY 序号
N	0 ~ 3	通道号
M	0 ~ 7	Lane 号 当 N=0 时, M 取值: [0:7] 当 N=1 时, M 取值: [0:1] 当 N=2 时, M 取值: [0:3] 当 N=3 时, M 取值: [0:1]

MIPI Rx 寄存器概览如表 10-12 所示。

表10-12 MIPI Rx 寄存器概览 (基址是 0x0_173C_0000)

偏移地址	名称	描述	页码
0x0000 + P × 0x0200	PHY_MODE_LINK(P)	Link(P) PHY 工作模式寄存器	10-194
0x0004 + P × 0x0200	PHY_SKEW_LINK(P)	Link(P) PHY 通道延迟调节寄存器	10-196
0x0008 + P × 0x0200	PHY_LANE_SI_LINK(P)	Link(P) PHY 通道 LANE SI 调节寄存器	10-197



偏移地址	名称	描述	页码
0x000C + P× 0x0200	PHY_EN_LINK(P)	Link(P) PHY 通道使能寄存器	10-197
0x0018 + P× 0x0200	PHY_DATA_LINK(P)	Link(P) PHY 输出的并行数据寄存器	10-199
0x001C + P× 0x0200	PHY_PH_MIPI_LINK(P)	Link(P) MIPI 数据包头寄存器	10-199
0x0020 + P× 0x0200	PHY_DATA_MIPI_LINK(P)	Link(P) MIPI 模式数据寄存器	10-199
0x0024 + P× 0x0200	PHY_SYNC_DCT_LINK(P)	Link(P) PHY LVDS 模式同步头检测控制寄存器	10-200
0x0030 + P× 0x0200	PHY_SYNC_SOF0_LINK(P)	Link(P) PHY LVDS 模式 Lane0 同步头寄存器	10-202
0x0034 + P× 0x0200	PHY_SYNC_SOF1_LINK(P)	Link(P) PHY LVDS 模式 Lane1 同步头寄存器	10-202
0x0038 + P× 0x0200	PHY_SYNC_SOF2_LINK(P)	Link(P) PHY LVDS 模式 Lane2 同步头寄存器	10-202
0x003C + P× 0x0200	PHY_SYNC_SOF3_LINK(P)	Link(P) PHY LVDS 模式 Lane3 同步头寄存器	10-202
0x0050 + P× 0x0200	PHY_DESKEW_CAL_LINK(P)	Link(P) AUTO DESKEW 控制寄存器	10-203
0x0150 + P× 0x0200	FREQ_MEASURE(P)	Link(P) 时钟频率测量寄存器	10-203
0x0160 + P× 0x0200	PHY_EXT_REG0(P)	PHY 通道配置的补充配置寄存器	10-204
0x01F0 + P× 0x0200	MIPI_CIL_INT_RAW_LINK(P)	Link(P) MIPI CIL 原始中断状态寄存器	10-204
0x01F4 + P× 0x0200	MIPI_CIL_INT_LINK(P)	Link(P) MIPI CIL 中断状态寄存器	10-206
0x01F8 + P× 0x0200	MIPI_CIL_INT_MASK_LINK(P)	Link(P) MIPI CIL 中断屏蔽寄存器	10-208



偏移地址	名称	描述	页码
0x0800	HS_MODE_SELECT	HS ID 选择寄存器	10-210
0x0804	D_P_SWAP	差分信号的 P/N 端互换选择寄存器	10-211
0x0808	PHY_EN	PHY 使能寄存器	10-212
0x080C	LANE_EN	LANE_EN 使能寄存器	10-212
0x0810	PHY_CIL_CTRL	PHY CIL 控制寄存器	10-213
0x0818	PHYCFG_MODE	PHY 配置模式寄存器	10-214
0x081C	PHYCFG_EN	PHY 配置使能寄存器	10-216
0x0820	CHN0_MEM_CTRL	CHN0 MEMORY 控制寄存器	10-217
0x0824	CHN0_CLR_EN	CHN0 强制置位使能寄存器	10-217
0x0828	CHN1_MEM_CTRL	CHN1 MEMORY 控制寄存器	10-217
0x082C	CHN1_CLR_EN	CHN1 强制置位使能寄存器	10-218
0x0830	CHN2_MEM_CTRL	CHN2 MEMORY 控制寄存器	10-218
0x0834	CHN2_CLR_EN	CHN2 强制置位使能寄存器	10-219
0x0838	CHN3_MEM_CTRL	CHN3 MEMORY 控制寄存器	10-219
0x083C	CHN3_CLR_EN	CHN3 强制置位使能寄存器	10-219
0x0880	MIPI_TEST_PHY	BIST_TEST_PHY 控制寄存器	10-220
0x0FF0	MIPI_INT_RAW	MIPI 系统原始中断状态寄存器	10-220
0x0FF4	MIPI_INT_ST	MIPI 系统中断状态寄存器	10-221
0x0FF8	MIPI_INT_MSK	MIPI 系统中断屏蔽寄存器	10-222
0x1020 + N ×0x1000	MIPI(N)_CRC_INT R_RAW	MIPI(N) CSI 原始中断状态寄存器	10-223
0x1024 + N ×0x1000	MIPI(N)_CRC_INT R_ST	MIPI(N) CSI 屏蔽后中断状态寄存器	10-226



偏移地址	名称	描述	页码
0x1028 + N ×0x1000	MIPI(N)_CRC_INT R_MSK	MIPI(N) CSI 中断屏蔽寄存器	10-229
0x1100 + N ×0x1000	MIPI(N)_USERDEF _DT	MIPI(N)用户自定义 data type 对应的像素位宽配置寄存器	10-231
0x1104 + N ×0x1000	MIPI(N)_USER_DE F	MIPI(N)用户自定义 data type 的类 型使能配置寄存器	10-233
0x1108 + N ×0x1000	MIPI(N)_CTRL_M ODE_HS	MIPI(N)工作模式使能寄存器	10-234
0x1200 + N ×0x1000	MIPI(N)_DOL_ID_ CODE0	MIPI(N) DOL 模式帧识别 0 寄存器	10-235
0x1204 + N ×0x1000	MIPI(N)_DOL_ID_ CODE1	MIPI(N) DOL 模式帧识别 1 寄存器	10-235
0x1208 + N ×0x1000	MIPI(N)_DOL_ID_ CODE2	MIPI(N) DOL 模式帧识别 2 寄存器	10-235
0x1210 + N ×0x1000	MIPI(N)_CROP_ST ART_CHN0	MIPI(N) CHN0 CROP 宽高起始点寄 存器	10-235
0x1214 + N ×0x1000	MIPI(N)_CROP_ST ART_CHN1	MIPI(N) CHN1 CROP 宽高起始点寄 存器	10-236
0x1218 + N ×0x1000	MIPI(N)_CROP_ST ART_CHN2	MIPI(N) CHN2 CROP 宽高起始点寄 存器	10-236
0x121C + N ×0x1000	MIPI(N)_CROP_ST ART_CHN3	MIPI(N) CHN3 CROP 宽高起始点寄 存器	10-236
0x1224 + N ×0x1000	MIPI(N)_IMGSIZE	MIPI(N)图像宽高寄存器	10-237
0x1230 + N ×0x1000	MIPI(N)_CTRL_M ODE_PIXEL	MIPI(N)输出工作模式使能寄存器	10-237
0x1240 + N ×0x1000	MIPI(N)_DUMMY _PIX_REG	MIPI(N) dummy 行像素值寄存器	10-238



偏移地址	名称	描述	页码
0x1250 + N ×0x1000	MIPI(N)_IMGSIZE 0_STATIS	MIPI(N) VC0 传输的图像宽高寄存器	10-239
0x1254 + N ×0x1000	MIPI(N)_IMGSIZE 1_STATIS	MIPI(N) VC1 传输的图像宽高寄存器	10-239
0x1258 + N ×0x1000	MIPI(N)_IMGSIZE 2_STATIS	MIPI(N) VC2 传输的图像宽高寄存器	10-239
0x125C + N ×0x1000	MIPI(N)_IMGSIZE 3_STATIS	MIPI(N) VC3 传输的图像宽高寄存器	10-240
0x12F0 + N ×0x1000	MIPI(N)_CTRL_IN T_RAW	MIPI(N)读数据错误原始中断状态寄存器	10-240
0x12F4 + N ×0x1000	MIPI(N)_CTRL_IN T	MIPI(N)读数据错误中断状态寄存器	10-241
0x12F8 + N ×0x1000	MIPI(N)_CTRL_IN T_MSK	MIPI(N)读数据错误中断屏蔽寄存器	10-242
0x1300 + N ×0x1000	LVDS(N)_WDR	LVDS(N) WDR 控制寄存器	10-243
0x1304 + N ×0x1000	LVDS(N)_DOLSCD _HBLK	LVDS(N) SCD 控制寄存器	10-244
0x1308 + N ×0x1000	LVDS(N)_CTRL	LVDS(N)控制寄存器	10-244
0x130C + N ×0x1000	LVDS(N)_IMGSIZE	LVDS(N)图像宽高寄存器	10-246
0x1310 + N ×0x1000	LVDS(N)_CROP_S TART0	LVDS(N)第 0 帧裁剪寄存器	10-246
0x1314 + N ×0x1000	LVDS(N)_CROP_S TART1	LVDS(N)第 1 帧裁剪寄存器	10-247
0x1318 + N ×0x1000	LVDS(N)_CROP_S TART2	LVDS(N)第 2 帧裁剪寄存器	10-247



偏移地址	名称	描述	页码
0x131C + N ×0x1000	LVDS(N)_CROP_S TART3	LVDS(N)第3帧裁剪寄存器	10-247
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	10-248
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	10-248
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	10-249
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	10-249
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	10-250
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	10-250
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	10-251
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	10-251
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	10-252
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	10-252



偏移地址	名称	描述	页码
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	10-253
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	10-253
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	10-254
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	10-254
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	10-255
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	10-255
0x1720 + N ×0x1000	LVDS(N)_LI_WOR D0	LVDS DOL 模式第 0 帧 LI 寄存器	10-256
0x1724 + N ×0x1000	LVDS(N)_LI_WOR D1	LVDS DOL 模式第 1 帧 LI 寄存器	10-256
0x1728 + N ×0x1000	LVDS(N)_LI_WOR D2	LVDS DOL 模式第 2 帧 LI 寄存器	10-256
0x172C + N ×0x1000	LVDS(N)_LI_WOR D3	LVDS DOL 模式第 3 帧 LI 寄存器	10-257
0x1780 + N ×0x1000	LVDS(N)_IMGSIZE 0_STATIS	LVDS LEF 图像宽高统计寄存器	10-257
0x1784 + N ×0x1000	LVDS(N)_IMGSIZE 1_STATIS	LVDS SEF1 图像宽高统计寄存器	10-257



偏移地址	名称	描述	页码
0x1788 + N ×0x1000	LVDS(N)_IMGSIZE 2_STATIS	LVDS SEF2 图像宽高统计寄存器	10-258
0x178C + N ×0x1000	LVDS(N)_IMGSIZE 3_STATIS	LVDS SEF3 图像宽高统计寄存器	10-258
0x17A0 + N ×0x1000	LVDS(N)_OUTPUT _PIX_NUM	LVDS 输出像素位宽选择寄存器	10-259
0x17A4 + N ×0x1000	LVDS(N)_16to14_ EN	LVDS 16bit 转 14bit 时序使能寄存器	10-259
0x17F0 + N ×0x1000	LVDS(N)_CTRL_IN T_RAW	LVDS 读数据原始中断状态寄存器	10-259
0x17F4 + N ×0x1000	LVDS(N)_CTRL_IN T	LVDS 读数据中断状态寄存器	10-261
0x17F8 + N ×0x1000	LVDS(N)_CTRL_IN T_MSK	LVDS 读数据中断屏蔽寄存器	10-263
0x1800 + N ×0x1000	LANE_ID0_CHN(N)	LANE 优先级配置寄存器 0	10-265
0x1804 + N ×0x1000	LANE_ID1_CHN(N)	LANE 优先级配置寄存器 1	10-265
0x18F0 + N ×0x1000	ALIGN(N)_INT_RA W	MIPI_ALIGN 原始中断状态寄存器	10-266
0x18F4 + N ×0x1000	ALIGN(N)_INT	MIPI_ALIGN 中断状态寄存器	10-267
0x18F8 + N ×0x1000	ALIGN(N)_INT_M SK	MIPI_ALIGN 中断屏蔽寄存器	10-268
0x1FF0 + N ×0x1000	CHN(N)_INT_RA W	通道(N)原始中断状态寄存器	10-270
0x1FF4 + N ×0x1000	CHN(N)_INT	通道(N)中断状态寄存器	10-270



偏移地址	名称	描述	页码
0x1FF8 + N ×0x1000	CHN(N)_INT_MAS K	通道(N)中断屏蔽寄存器	10-271

10.3.6 MIPI Rx 寄存器描述

PHY_MODE_LINK(P)

PHY_MODE_LINK(P)为 Link(P) PHY 工作模式寄存器。

Offset Address: 0x0000 + P×0x0200 Total Reset Value: 0x0003_0000

Bits	Access	Name	Description	Reset
[31:25]	-	reserved	保留。	0x00
[25:24]	RW	phy_rg_mipi_mode	输入数据模式选择。 00: MIPIRX/HISPI 01: sub_LVDS 10: LVDS 11: 保留	0x0
[23:18]	-	reserved	保留。	0x00
[17]	RW	phy_rg_drveclk2_enz	外部时钟 cloclk2 源。 0: 使用外部时钟源; 1: 不使用外部时钟源。	0x1
[16]	RW	phy_rg_drveclk_enz	外部时钟 cloclk 源。 0: 使用外部时钟源; 1: 不使用外部时钟源。	0x1
[15]	RO	reserved	保留。	0x0
[14]	RW	phy_rg_ext_en	MIPI 8L 应用场景使能, CLKP/N 源于 LINK0。 0: 禁止;	0x0



Bits	Access	Name	Description	Reset
			1: 使能。	
[13]	RW	phy_rg_ext2_en	MIPI 4L 应用场景使能, CLKP/N 源于 LINK0/LINK2。 0: 禁止; 1: 使能。	0x0
[12]	RW	phy_rg_int_en	MIPI 2L 应用场景使能, CLKP/N 源于每个 LINK 自身。 0: 禁止; 1: 使能。	0x0
[11]	RW	phy_rg_da_en_cmos_tx	CMOS 模式输出使能。 0: 禁止; 1: 使能。	0x0
[10]	RW	phy_rg_en_mipi2lcmos1_swap	clk0/d0/d2 工作在 CMOS 模式使能。 0: 禁止; 1: 使能。	0x0
[9]	RW	phy_rg_en_mipi2lcmos	clk2/d1/d3 工作在 CMOS 模式使能。 0: 禁止; 1: 使能。	0x0
[8]	RW	phy_rg_en_cmos	各个 lane 都工作在 CMOS 模式使能。 0: 禁止; 1: 使能。	0x0
[7:6]	-	reserved	保留。	0x0
[5]	RW	phy_rg_en_clk2	Clock1 Lane 使能。 0: 禁止; 1: 使能。	0x0
[4]	RW	phy_rg_en_clk	Clock0 Lane 使能。 0: 禁止;	0x0



Bits	Access	Name	Description	Reset
			1: 使能。	
[3:0]	RW	phy_rg_en_d	Data Lane 使能。 0: 禁止; 1: 使能。	0x0

PHY_SKEW_LINK(P)

PHY_SKEW_LINK(P)为 Link(P) PHY 通道延迟调节寄存器。

Offset Address: 0x0004 + P×0x0200 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:20]	RW	phy_rg_ah_ds	Lane3~0 延迟自动设置, 手动调整 clock 与 data skew 控制, 搭配 RG_AADH_DS 使用。 <ul style="list-style-type: none"> 设置 RG_AH_DS=1 且 RG_AADH_DS=1 时, 进入手动调整 deskew 模式; 设置 RG_AH_DS=1 且 RG_AADH_DS=0 时, 进入工程测试模式。 	0x0
[19:16]	RW	phy_rg_adv12	当进入工程测试模式使用。 0: clock 向右移动一阶相位差; 1: clock 向右移动二阶相位差。	0x0
[15:12]	RW	phy_rg_aadh_ds	搭配 RG_AH_DS 使用, 设定参考 RG_AH_DS 说明。 [3:0]分别表示四个 data lane。	0x0
[11:8]	RW	phy_rg_dirbi	Lane3~0 用户设定延迟方向。 0: 时钟相位向右移动; 1: 数据相位向右移动。	0x0
[7:4]	RW	phy_rg_deskew	Lane3~0 自动调整相位机制模式, 采样窗	0x0



Bits	Access	Name	Description	Reset
			口控制 0: 采样窗口为 3 级延迟(约 80ps); 1: 采样窗口为 4 级延迟(约 108ps)。	
[3:0]	RO	reserved	保留。	0x0

PHY_LANE_SI_LINK(P)

PHY_LANE_SI_LINK(P)为 Link(P) PHY 通道 LANE SI 调节寄存器。

Offset Address: 0x0008 + P×0x0200 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RW	phy_rg_lane3_si	Lane3 低 7Bit 的数据延迟设置。	0x00
[23:16]	RW	phy_rg_lane2_si	Lane2 低 7Bit 的数据延迟设置。	0x00
[15:8]	RW	phy_rg_lane1_si	Lane1 低 7Bit 的数据延迟设置。	0x00
[7:0]	RW	phy_rg_lane0_si	Lane0 低 7Bit 的数据延迟设置。	0x00

PHY_EN_LINK(P)

PHY_EN_LINK(P)为 Link(P) PHY 通道使能寄存器。

Offset Address: 0x000C + P×0x0200 Total Reset Value: 0x0000_00F0

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x0000
[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



Bits	Access	Name	Description	Reset
[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
[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]	RW	phy_da_d_valid_in v	Data valid 控制使能, 与 phy_da_dx_valid(x=0~3)配合使用。 0: 关闭; 1: 使能。	0xF
[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;	0x0



Bits	Access	Name	Description	Reset
			1: 高速模式使能。	

PHY_DATA_LINK(P)

PHY_DATA_LINK(P)为 Link(P) PHY 输出的并行数据寄存器。

Offset Address: 0x0018 + P×0x0200 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(P)

PHY_PH_MIPI_LINK(P)为 Link(P) MIPI 数据包头寄存器。

Offset Address: 0x001C + P×0x0200 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(P)

PHY_DATA_MIPI_LINK(P)为 Link(P) MIPI 模式数据寄存器。

Offset Address: 0x0020 + P×0x0200 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(P)

PHY_SYNC_DCT_LINK(P)为 Link(P) PHY LVDS 模式同步头检测控制寄存器。

Offset Address: 0x0024 + P×0x0200 Total Reset Value: 0x0010_1101

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	保留。	0x00
[25]	RW	ci_l_sync_det_mode	LVDS/HiSPi 同步模式选择。 0: 检测到 SOF 后一直有效; 1: 同时检查 SOF/EOF/SOL/EOL。	0x0
[24]	-	reserved	保留。	0x0
[23:21]	-	reserved	保留。	0x0
[20]	RW	ci_l_code_big_endian1	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
[19]	-	reserved	保留。	0x0
[18:16]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[15]	-	reserved	保留。	0x0
[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_endian0	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, bit10, ...bit0。	0x1
[7]	-	reserved	保留。	0x0
[6:4]	-	reserved	保留。	0x0
[3]	-	reserved	保留。	0x0
[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。 其他: 保留。	0x1



PHY_SYNC_SOF0_LINK(P)

PHY_SYNC_SOF0_LINK(P)为 Link(P) PHY LVDS 模式 Lane0 同步头寄存器。

Offset Address: $0x0030 + P \times 0x0200$ Total Reset Value: $0x0000_0000$

Bits	Access	Name	Description	Reset
[31:16]	RW	ci_l_sof1_word4_0	Lane0 帧起始同步码(N+1 Frame)。	0x0000
[15:0]	RW	ci_l_sof0_word4_0	Lane0 帧起始同步码(N Frame)。	0x0000

PHY_SYNC_SOF1_LINK(P)

PHY_SYNC_SOF1_LINK(P)为 Link(P) PHY LVDS 模式 Lane1 同步头寄存器。

Offset Address: $0x0034 + P \times 0x0200$ Total Reset Value: $0x0000_0000$

Bits	Access	Name	Description	Reset
[31:16]	RW	ci_l_sof1_word4_1	Lane1 帧起始同步码(N+1 Frame)。	0x0000
[15:0]	RW	ci_l_sof0_word4_1	Lane1 帧起始同步码(N Frame)。	0x0000

PHY_SYNC_SOF2_LINK(P)

PHY_SYNC_SOF2_LINK(P)为 Link(P) PHY LVDS 模式 Lane2 同步头寄存器。

Offset Address: $0x0038 + P \times 0x0200$ Total Reset Value: $0x0000_0000$

Bits	Access	Name	Description	Reset
[31:16]	RW	ci_l_sof1_word4_2	Lane2 帧起始同步码(N+1 Frame)。	0x0000
[15:0]	RW	ci_l_sof0_word4_2	Lane2 帧起始同步码(N Frame)。	0x0000

PHY_SYNC_SOF3_LINK(P)

PHY_SYNC_SOF3_LINK(P)为 Link(P) PHY LVDS 模式 Lane3 同步头寄存器。

Offset Address: $0x003C + P \times 0x0200$ Total Reset Value: $0x0000_0000$



Bits	Access	Name	Description	Reset
[31:16]	RW	ci_l_sof1_word4_3	Lane3 帧起始同步码(N+1 Frame)。	0x0000
[15:0]	RW	ci_l_sof0_word4_3	Lane3 帧起始同步码(N Frame)。	0x0000

PHY_DESKEW_CAL_LINK(P)

PHY_DESKEW_CAL_LINK(P)为 Link(P) AUTO DESKEW 控制寄存器。

Offset Address: 0x0050 + P×0x0200 Total Reset Value: 0x0000_FFFF

Bits	Access	Name	Description	Reset
[31:28]	RW	deskew_cal_8b1	tranning 检测 8 个 1 后就自动打开 phy_da_en_deskew。	0x0
[27:24]	RW	deskew_cal_sot_by pass	tranning 时跳过 SOT 检测开关, [3:0]对应 lane3~lane0。	0x0
[23:20]	RW	deskew_cal_rstn_en	自动 tranning 每帧清 0 开关, [3:0]对应 lane3~lane0。	0x0
[19:16]	RW	deskew_cal_en	自动 tranning 使能开关, [3:0]对应 lane3~lane0。	0x0
[15:0]	RW	deskew_cal_cycle	自动 tranning 校正时间, 单位(cycle)。	0xFFFF

FREQ_MEASURE(P)

FREQ_MEASURE(P)为 Link(P) 时钟频率测量寄存器。

Offset Address: 0x0150 + P×0x0200 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	hs_cnt	hs 时钟域计数值。	0x0000
[15:0]	RO	ci_l_cnt	ci_l 时钟域计数值。	0x0000



PHY_EXT_REG0(P)

PHY_EXT_REG0(P)为 PHY 通道配置的补充配置寄存器。

Offset Address: 0x0160 + P×0x0200 Total Reset Value: 0x1249_2000

Bits	Access	Name	Description	Reset
[31:15]	RO	reserved	保留。	0x1249
[14:12]	RO	reserved	保留。	0x2
[11:10]	RW	phy_rg_lane3_si	Lane3 高 2bit 的数据延迟设置。	0x0
[9:8]	RW	phy_rg_lane2_si	Lane2 高 2bit 的数据延迟设置。	0x0
[7:6]	RW	phy_rg_lane1_si	Lane1 高 2bit 的数据延迟设置。	0x0
[5:4]	RW	phy_rg_lane0_si	Lane0 高 2bit 的数据延迟设置。	0x0
[3:0]	RO	reserved	保留。	0x0

MIPI_CIL_INT_RAW_LINK(P)

MIPI_CIL_INT_RAW_LINK(P)为 Link(P) MIPI CIL 原始中断状态寄存器。

Offset Address: 0x01F0 + P×0x0200 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19]	WC	sot_detc_d3_raw	Data Lane3 在 deskew pattern 中检测到 sot 序列原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[18]	WC	sot_detc_d2_raw	Data Lane2 在 deskew pattern 中检测到 sot 序列原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[17]	WC	sot_detc_d1_raw	Data Lane1 在 deskew pattern 中检测到	0x0



Bits	Access	Name	Description	Reset
			sot 序列原始中断状态。 0: 无原始中断; 1: 有原始中断。	
[16]	WC	sot_detc_d0_raw	Data Lane0 在 deskew pattern 中检测到 sot 序列原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[15:13]	-	reserved	保留。	0x0
[12]	WC	err_timeout_ck_raw	Clock Lane FSM timeout 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[11]	WC	err_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
[10]	WC	err_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	err_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	err_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
[7:5]	-	reserved	保留。	0x0
[4]	WC	err_escape_ck_raw	Clock Lane escape 序列原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0



Bits	Access	Name	Description	Reset
[3]	WC	err_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	err_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	err_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	err_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(P)

MIPI_CIL_INT_LINK(P)为 Link(P) MIPI CIL 中断状态寄存器。

Offset Address: 0x01F4 + P×0x0200 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19]	RO	sot_detc_d3_st	Data Lane3 在 deskew pattern 中检测到 sot 序列中断状态。 0: 无中断; 1: 有中断。	0x0
[18]	RO	sot_detc_d2_st	Data Lane2 在 deskew pattern 中检测到 sot 序列中断状态。 0: 无中断; 1: 有中断。	0x0
[17]	RO	sot_detc_d1_st	Data Lane1 在 deskew pattern 中检测到	0x0



Bits	Access	Name	Description	Reset
			sot 序列中断状态。 0: 无中断; 1: 有中断。	
[16]	RO	sot_detc_d0_st	Data Lane0 在 deskew pattern 中检测到 sot 序列中断状态。 0: 无中断; 1: 有中断。	0x0
[15:13]	-	reserved	保留。	0x0
[12]	RO	err_timeout_ck_st	Clock Lane FSM timeout 中断状态。 0: 无中断; 1: 有中断。	0x0
[11]	RO	err_timeout_d3_st	Data Lane3 FSM timeout 中断状态。 0: 无中断; 1: 有中断。	0x0
[10]	RO	err_timeout_d2_st	Data Lane2 FSM timeout 中断状态。 0: 无中断; 1: 有中断。	0x0
[9]	RO	err_timeout_d1_st	Data Lane1 FSM timeout 中断状态。 0: 无中断; 1: 有中断。	0x0
[8]	RO	err_timeout_d0_st	Data Lane0 FSM timeout 中断状态。 0: 无中断; 1: 有中断。	0x0
[7:5]	-	reserved	保留。	0x0
[4]	RO	err_escape_ck_st	Clock Lane escape 序列中断状态。 0: 无中断; 1: 有中断。	0x0



Bits	Access	Name	Description	Reset
[3]	RO	err_escape_d3_st	Data Lane3 escape 序列中断状态。 0: 无中断; 1: 有中断。	0x0
[2]	RO	err_escape_d2_st	Data Lane2 escape 序列中断状态。 0: 无中断; 1: 有中断。	0x0
[1]	RO	err_escape_d1_st	Data Lane1 escape 序列中断状态。 0: 无中断; 1: 有中断。	0x0
[0]	RO	err_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(P)

MIPI_CIL_INT_MSK_LINK(P)为 Link(P) MIPI CIL 中断屏蔽寄存器。

Offset Address: 0x01F8 + P×0x0200 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	保留。	0x000
[19]	RO	sot_detc_d3_msk	Data Lane3 在 deskew pattern 中检测到 sot 序列中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[18]	RO	sot_detc_d2_msk	Data Lane2 在 deskew pattern 中检测到 sot 序列中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[17]	RO	sot_detc_d1_msk	Data Lane1 在 deskew pattern 中检测到	0x0



Bits	Access	Name	Description	Reset
			sot 序列中断状态使能。 0: 屏蔽中断; 1: 使能中断。	
[16]	RO	sot_detc_d0_msk	Data Lane0 在 deskew pattern 中检测到 sot 序列中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[15:13]	-	reserved	保留。	0x0
[12]	RW	err_timeout_ck_msk	Clock Lane FSM timeout 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[11]	RW	err_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	err_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	err_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	err_timeout_d0_msk	Data Lane0 FSM timeout 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[7:5]	-	reserved	保留。	0x0
[4]	RW	err_escape_ck_msk	Clock Lane escape 序列中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0



Bits	Access	Name	Description	Reset
[3]	RW	err_escape_d3_mask	Data Lane3 escape 序列中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RW	err_escape_d2_mask	Data Lane2 escape 序列中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	RW	err_escape_d1_mask	Data Lane1 escape 序列中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	err_escape_d0_mask	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
[31:4]	-	reserved	保留。	0x0000000
[3:0]	RW	hs_mode	场景模式选择。 0x0: 8Lane; 0x7: 4Lane+4Lane; 0xA: 4Lane+2Lane+2Lane; 0xB: 2Lane+2Lane+2Lane+2Lane; 0xC: 2Lane+2Lane+4lane; 其它: 保留。	0x0



D_P_SWAP

D_P_SWAP 为差分信号的 P/N 端互换选择寄存器。

Offset Address: 0x0804 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:12]	-	reserved	保留。	0x00000
[11]	RW	phy1_d3_swap	PHY1 的 DATA3 P/N 互换使能。 0: 禁止; 1: 使能。	0x0
[10]	RW	phy1_d2_swap	PHY1 的 DATA2 P/N 互换使能。 0: 禁止; 1: 使能。	0x0
[9]	RW	phy1_d1_swap	PHY1 的 DATA1 P/N 互换使能。 0: 禁止; 1: 使能。	0x0
[8]	RW	phy1_d0_swap	PHY1 的 DATA0 P/N 互换使能。 0: 禁止; 1: 使能。	0x0
[7:4]	-	reserved	保留。	0x0
[3]	RW	phy0_d3_swap	PHY0 的 DATA3 P/N 互换使能。 0: 禁止; 1: 使能。	0x0
[2]	RW	phy0_d2_swap	PHY0 的 DATA2 P/N 互换使能。 0: 禁止; 1: 使能。	0x0
[1]	RW	phy0_d1_swap	PHY0 的 DATA1 P/N 互换使能。 0: 禁止; 1: 使能。	0x0



Bits	Access	Name	Description	Reset
[0]	RW	phy0_d0_swap	PHY0 的 DATA0 P/N 互换使能。 0: 禁止; 1: 使能。	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
[31:2]	-	reserved	保留。	0x0000000
[1]	RW	phy1_en	PHY1 使能控制。 0: 禁止; 1: 使能。	0x0
[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
[31:8]	-	reserved	保留。	0x0000
[7]	RW	lane7_en	Lane7 工作使能。 0: 禁止; 1: 使能。	0x0
[6]	RW	lane6_en	Lane6 工作使能。	0x0



Bits	Access	Name	Description	Reset
			0: 禁止; 1: 使能。	
[5]	RW	lane5_en	Lane5 工作使能。 0: 禁止; 1: 使能。	0x0
[4]	RW	lane4_en	Lane4 工作使能。 0: 禁止; 1: 使能。	0x0
[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。	0x00000



Bits	Access	Name	Description	Reset
[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
[7:2]	-	reserved	保留。	0x0
[1]	RW	phycil1_cken	PHYCIL1 时钟门控。 0: 关闭时钟; 1: 打开时钟。	0x0
[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
[31:15]	-	reserved	保留。	0x0
[14:12]	RW	phycil1_1_cfg_mod e	PHY1 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	0x0



Bits	Access	Name	Description	Reset
			FSM, 时钟通道的使能受控于 PHYCFG_EN。 其他: 保留。	
[11]	RW	phycil1_cfg_mode_sel	PHY1 配置模式。 0: 受控于 PHYCFG_EN; 1: 受控于寄存器的值。(cmos)	0x0
[10:8]	RW	phycil1_0_cfg_mode	PHY1 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。 其他: 保留。	0x0
[7]	-	reserved	保留。	0x0
[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。 其他: 保留。	0x0
[3]	RW	phycil0_cfg_mode_sel	PHY0 配置模式。	0x0



Bits	Access	Name	Description	Reset
			0: 受控于 PHYCFG_EN; 1: 受控于寄存器的值。(CMOS)	
[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。 其他: 保留。	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
[31:2]	-	reserved	保留。	0x0000000
[1]	WO	phycil1_cfg_en	PHY1 配置使能寄存器。 0: 不使能 PHY 的配置; 1: 使能 PHY 的配置。	0x0
[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_2020

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000010
[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_2020

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000010



[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

CHN2_MEM_CTRL

CHN2_MEM_CTRL 为 CHN2 MEMORY 控制寄存器。

Offset Address: 0x0830 Total Reset Value: 0x0000_2020

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000010
[0]	RW	chn2_mem_ck_gt	CHN2 MEMORY 时钟门控端口。 0: 关闭 memory 时钟 1: 打开 memory 时钟	0x0



CHN2_CLR_EN

CHN2_CLR_EN 为 CHN2 强制置位使能寄存器。

Offset Address: 0x0834 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	chn2_clr_en_align	CHN2 ALIGN 模块的强制置位控制。 0: 关闭 CHN2 ALIGN 的置位请求; 1: 使能 CHN2 ALIGN 的置位请求。	0x0
[0]	RW	chn2_clr_en_lvds	CHN2 LVDS_CTRL 模块的强制置位控制。 0: 关闭 CHN2 LVDS_CTRL 的置位请求; 1: 使能 CHN2 LVDS_CTRL 的置位请求。	0x0

CHN3_MEM_CTRL

CHN3_MEM_CTRL 为 CHN3 MEMORY 控制寄存器。

Offset Address: 0x0838 Total Reset Value: 0x0000_2020

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000010
[0]	RW	chn3_mem_ck_gt	CHN3 MEMORY 时钟门控端口。 0: 关闭 memory 时钟; 1: 打开 memory 时钟。	0x0

CHN3_CLR_EN

CHN3_CLR_EN 为 CHN3 强制置位使能寄存器。

Offset Address: 0x083C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000



[1]	RW	chn3_clr_en_align	CHN3 ALIGN 模块的强制置位控制。 0: 关闭 CHN3 ALIGN 的置位请求; 1: 使能 CHN3 ALIGN 的置位请求。	0x0
[0]	RW	chn3_clr_en_lvds	CHN3 LVDS_CTRL 模块的强制置位控制。 0: 关闭 CHN3 LVDS_CTRL 的置位请求; 1: 使能 CHN3 LVDS_CTRL 的置位请求。	0x0

MIPI_TEST_PHY

MIPI_TEST_PHY 为 BIST_TEST_PHY 控制寄存器。

Offset Address: 0x0880 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	test_phy_srst_req	bist 模块复位请求。 0: 撤销复位; 1: 复位。	0x0
[3:1]	-	reserved	保留。	0x0
[0]	RW	test_phy_en	test_phy 使能信号。 0: 关闭; 1: 使能。	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
[31:8]	-	reserved	保留。	0x000000
[7]	WC	int_chn3_raw	通道 3 原始中断状态。	0x0



Bits	Access	Name	Description	Reset
			0: 无原始中断; 1: 有原始中断。	
[6]	WC	int_chn2_raw	通道 2 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[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
[3:2]	-	reserved	保留。	0x0
[1]	WC	int_phycil1_raw	PHYCIL1 原始中断状态。 0: 无原始中断; 1: 有原始中断。	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
[31:8]	-	reserved	保留。	0x000000
[7]	RO	int_chn3_st	通道 3 中断状态。 0: 无中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	
[6]	RO	int_chn2_st	通道 2 中断状态。 0: 无中断; 1: 有中断。	0x0
[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
[3:2]	-	reserved	保留。	0x0
[1]	RO	int_phycil1_st	PHYCIL1 中断状态。 0: 无中断; 1: 有中断。	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
[31:8]	-	reserved	保留。	0x000000
[7]	RW	int_chn3_mask	通道 3 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0



Bits	Access	Name	Description	Reset
[6]	RW	int_chn2_mask	通道 2 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[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
[3:2]	-	reserved	保留。	0x0
[1]	RW	int_phycil1_mask	PHYCIL1 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	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)_CRC_INTR_RAW

MIPI(N)_CRC_INTR_RAW 为 MIPI(N) CSI 原始中断状态寄存器。

Offset Address: $0x1020 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23]	RO	frame_s_e_num_mismatch_vc3_raw	virtual channel3 的帧起始序号和帧结束序号不匹配原始中断状态。写 1 清零。 0: 无中断; 1: 有中断。	0x0



Bits	Access	Name	Description	Reset
[22]	RO	frame_s_e_num_mismatch_vc2_raw	virtual channel2 的帧起始序号和帧结束序号不匹配原始中断状态。写 1 清零。 0: 无中断; 1: 有中断。	0x0
[21]	RO	frame_s_e_num_mismatch_vc1_raw	virtual channel1 的帧起始序号和帧结束序号不匹配原始中断状态。写 1 清零。 0: 无中断; 1: 有中断。	0x0
[20]	RO	frame_s_e_num_mismatch_vc0_raw	virtual channel0 的帧起始序号和帧结束序号不匹配原始中断状态。写 1 清零。 0: 无中断; 1: 有中断。	0x0
[19]	RO	frame_num_err_vc3_raw	virtual channel3 的帧序号出错原始中断状态, 可能发生了丢帧。写 1 清零。 0: 无中断; 1: 有中断。	0x0
[18]	RO	frame_num_err_vc2_raw	virtual channel2 的帧序号出错原始中断状态, 可能发生了丢帧。写 1 清零。 0: 无中断; 1: 有中断。	0x0
[17]	RO	frame_num_err_vc1_raw	virtual channel1 的帧序号出错原始中断状态, 可能发生了丢帧。写 1 清零。 0: 无中断; 1: 有中断。	0x0
[16]	RO	frame_num_err_vc0_raw	virtual channel0 的帧序号出错原始中断状态, 可能发生了丢帧。写 1 清零。 0: 无中断; 1: 有中断。	0x0



Bits	Access	Name	Description	Reset
[15:9]	-	reserved	保留。	0x00
[8]	RO	ecc_err_mult_raw	包 ECC 校验原始中断状态, 至少 2 个错误, ECC 无法纠正。写 1 清零。 0: 无中断; 1: 有中断。	0x0
[7]	RO	ecc_err_vc3_raw	virtual channel3 有 1 位 ECC 校验错误原始中断状态, , ECC 已纠正。写 1 清零。 0: 无中断; 1: 有中断。	0x0
[6]	RO	ecc_err_vc2_raw	virtual channel2 有 1 位 ECC 校验错误原始中断状态, , ECC 已纠正。写 1 清零。 0: 无中断; 1: 有中断。	0x0
[5]	RO	ecc_err_vc1_raw	virtual channel1 有 1 位 ECC 校验错误原始中断状态, , ECC 已纠正。写 1 清零。 0: 无中断; 1: 有中断。	0x0
[4]	RO	ecc_err_vc0_raw	virtual channel0 有 1 位 ECC 校验错误原始中断状态, , ECC 已纠正。写 1 清零。 0: 无中断; 1: 有中断。	0x0
[3]	RO	crc_err_vc3_raw	virtual channel3 数据 CRC 错误原始中断状态。写 1 清零。 0: 无中断; 1: 有中断。	0x0
[2]	RO	crc_err_vc2_raw	virtual channel2 数据 CRC 错误原始中断状态。写 1 清零。 0: 无中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	
[1]	RO	crc_err_vc1_raw	virtual channel1 数据 CRC 错误原始中断状态。写 1 清零。 0: 无中断; 1: 有中断。	0x0
[0]	RO	crc_err_vc0_raw	virtual channel0 数据 CRC 错误原始中断状态。写 1 清零。 0: 无中断; 1: 有中断。	0x0

MIPI(N)_CRC_INTR_ST

MIPI(N)_CRC_INTR_ST 为 MIPI(N) CSI 屏蔽后中断状态寄存器。

Offset Address: $0x1024 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23]	RO	frame_s_e_num_mismatch_vc3_st	virtual channel3 的帧起始序号和帧结束序号不匹配, 经中断屏蔽寄存器处理后的中断状态。 0: 无中断; 1: 有中断。	0x0
[22]	RO	frame_s_e_num_mismatch_vc2_st	virtual channel2 的帧起始序号和帧结束序号不匹配, 经中断屏蔽寄存器处理后的中断状态。 0: 无中断; 1: 有中断。	0x0
[21]	RO	frame_s_e_num_mismatch_vc1_st	virtual channel1 的帧起始序号和帧结束序号不匹配, 经中断屏蔽寄存器处理后的中断状态。	0x0



Bits	Access	Name	Description	Reset
			0: 无中断; 1: 有中断。	
[20]	RO	frame_s_e_num_mismatch_vc0_st	virtual channel0 的帧起始序号和帧结束序号不匹配, 经中断屏蔽寄存器处理后的中断状态。 0: 无中断; 1: 有中断。	0x0
[19]	RO	frame_num_err_vc3_st	virtual channel3 的帧序号出错, 经中断屏蔽寄存器处理后的中断状态, 可能发生了丢帧。 0: 无中断; 1: 有中断。	0x0
[18]	RO	frame_num_err_vc2_st	virtual channel2 的帧序号出错, 经中断屏蔽寄存器处理后的中断状态, 可能发生了丢帧。 0: 无中断; 1: 有中断。	0x0
[17]	RO	frame_num_err_vc1_st	virtual channel1 的帧序号出错, 经中断屏蔽寄存器处理后的中断状态, 可能发生了丢帧。 0: 无中断; 1: 有中断。	0x0
[16]	RO	frame_num_err_vc0_st	virtual channel0 的帧序号出错, 经中断屏蔽寄存器处理后的中断状态, 可能发生了丢帧。 0: 无中断; 1: 有中断。	0x0
[15:9]	-	reserved	保留。	0x00
[8]	RO	ecc_err_mult_st	包 ECC 校验至少 2 个错误, ECC 无法纠正, 经中断屏蔽寄存器处理后的中断状	0x0



Bits	Access	Name	Description	Reset
			态。 0: 无中断; 1: 有中断。	
[7]	RO	ecc_err_vc3_st	virtual channel3 有 1 位 ECC 校验错误, ECC 已纠正, 经中断屏蔽寄存器处理后的中断状态。 0: 无中断; 1: 有中断。	0x0
[6]	RO	ecc_err_vc2_st	virtual channel2 有 1 位 ECC 校验错误, ECC 已纠正, 经中断屏蔽寄存器处理后的中断状态。 0: 无中断; 1: 有中断。	0x0
[5]	RO	ecc_err_vc1_st	virtual channel1 有 1 位 ECC 校验错误, ECC 已纠正, 经中断屏蔽寄存器处理后的中断状态。 0: 无中断; 1: 有中断。	0x0
[4]	RO	ecc_err_vc0_st	virtual channel0 有 1 位 ECC 校验错误, ECC 已纠正, 经中断屏蔽寄存器处理后的中断状态。 0: 无中断; 1: 有中断。	0x0
[3]	RO	crc_err_vc3_st	virtual channel3 数据 CRC 错误, 经中断屏蔽寄存器处理后的中断状态。 0: 无中断; 1: 有中断。	0x0
[2]	RO	crc_err_vc2_st	virtual channel2 数据 CRC 错误, 经中断屏蔽寄存器处理后的中断状态。	0x0



Bits	Access	Name	Description	Reset
			0: 无中断; 1: 有中断。	
[1]	RO	crc_err_vc1_st	virtual channel1 数据 CRC 错误, 经中断屏蔽寄存器处理后的中断状态。 0: 无中断; 1: 有中断。	0x0
[0]	RO	crc_err_vc0_st	virtual channel0 数据 CRC 错误, 经中断屏蔽寄存器处理后的中断状态。 0: 无中断; 1: 有中断。	0x0

MIPI(N)_CRC_INTR_MSK

MIPI(N)_CRC_INTR_MSK 为 MIPI(N) CSI 中断屏蔽寄存器。

Offset Address: $0x1028 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23]	RW	frame_s_e_num_mismatch_vc3_msk	frame_s_e_num_mismatch_vc3_raw 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[22]	RW	frame_s_e_num_mismatch_vc2_msk	frame_s_e_num_mismatch_vc2_raw 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[21]	RW	frame_s_e_num_mismatch_vc1_msk	frame_s_e_num_mismatch_vc1_raw 中断屏蔽寄存器。 0: 屏蔽中断;	0x0



Bits	Access	Name	Description	Reset
			1: 使能中断。	
[20]	RW	frame_s_e_num_mismatch_vc0_msk	frame_s_e_num_mismatch_vc0_raw 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[19]	RW	frame_num_err_vc3_msk	frame_num_err_vc3_raw 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[18]	RW	frame_num_err_vc2_msk	frame_num_err_vc2_raw 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[17]	RW	frame_num_err_vc1_msk	frame_num_err_vc1_raw 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[16]	RW	frame_num_err_vc0_msk	frame_num_err_vc0_raw 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[15:9]	-	reserved	保留。	0x00
[8]	RW	ecc_err_mult_msk	ecc_err_mult_raw 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[7]	RW	ecc_err_vc3_msk	ecc_err_vc3_raw 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0



Bits	Access	Name	Description	Reset
[6]	RW	ecc_err_vc2_msk	ecc_err_vc2_raw 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[5]	RW	ecc_err_vc1_msk	ecc_err_vc1_raw 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[4]	RW	ecc_err_vc0_msk	ecc_err_vc0_raw 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[3]	RW	crc_err_vc3_msk	crc_err_vc3_raw 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[2]	RW	crc_err_vc2_msk	crc_err_vc2_raw 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[1]	RW	crc_err_vc1_msk	crc_err_vc1_raw 中断屏蔽寄存器。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	crc_err_vc0_msk	crc_err_vc0_raw 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; 其他: embedded 格式。	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; 001:10bit; 010:12bit; 011:14bit; 100:16bit; 其他: embedded 格式。	0x0
[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 其他: embedded 格式。	0x0
[3]	-	reserved	保留	0x0
[2:0]	RW	user_def0_dt	data type 为 user_def0 时传输的像素位	0x0



Bits	Access	Name	Description	Reset
			宽。 000:8bit; 001:10bit; 010:12bit; 011:14bit; 100:16bit; 其他: embedded 格式。	

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 MIPI(N)_USERDEF_DT.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 MIPI(N)_USERDEF_DT.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	0x35



Bits	Access	Name	Description	Reset
			MIPI(N)_USERDEF_DT.user_def1_dt 配合使用。	
[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 MIPI(N)_USERDEF_DT.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×0x1000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:9]	-	reserved	保留。	0x00000
[8]	RW	user_def_en	user define 模式使能。 0: 禁止; 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: 禁止; 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: 禁止; 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	virtual chn0 裁剪起始点高度。	0x0000
[15:0]	RW	mipi_start_x_chn0	virtual 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	virtual chn1 裁剪起始点高度。	0x0000
[15:0]	RW	mipi_start_x_chn1	virtual 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	virtual chn2 裁剪起始点高度。	0x0000
[15:0]	RW	mipi_start_x_chn2	virtual 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	virtual chn3 裁剪起始点高度。	0x0000
[15:0]	RW	mipi_start_x_chn3	virtual 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
[31:21]	-	reserved	保留。	0x000
[20]	RW	mipi_yuv_420_legacy_en	YUV420 Legacy 使能。 0: 禁止; 1: 使能。	0x0
[19]	RW	mipi_yuv_420_nolegacy_en	YUV420 NO Legacy 使能。 0: 禁止; 1: 使能。	0x0
[18]	RW	mipi_yuv_422_en	YUV422 使能。 0: 禁止; 1: 使能。	0x0
[17]	RW	mipi_double_yuv_en	MIPI 模式输出 YUV 位宽选择。 0: 一拍一像素; 1: 保留。	0x0



Bits	Access	Name	Description	Reset
[16]	RW	mipi_double_pix_en	MIPI 模式输出像素位宽选择。 0: 一拍一像素; 1: 保留。	0x0
[15]	RO	reserved	保留。	0x0
[14:13]	RW	stagger_frm_num	stagger hdr mode 帧数。 01: 两帧 hdr; 10: 三帧 hdr; 其他: 保留。	0x0
[12]	RW	stagger_hdr_mode	omnivision stagger hdr mode 模式使能。 0: 禁止; 1: 使能。	0x0
[11]	RW	sync_clear_en	帧结束自清使能。 0: 禁止; 1: 使能。	0x0
[10:5]	-	reserved	保留。	0x0
[4]	RW	mipi_dol_mode	MIPI DOL 模式使能。 0: 禁止; 1: 使能。	0x0
[3]	-	reserved	保留。	0x0
[2:1]	-	reserved	保留。	0x0
[0]	RW	crop_en	裁剪使能。 0: 禁止; 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×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)_IMGSIZE0_STATIS

MIPI(N)_IMGSIZE0_STATIS 为 MIPI(N) VC0 传输的图像宽高寄存器。

Offset Address: 0x1250 + N×0x1000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	imgheight_statis_vc0	MIPI 模式下, virtual channel0 传输的上 一帧图像高度。	0x0000
[15:0]	RO	imgwidth_statis_vc0	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×0x1000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	imgheight_statis_vc1	MIPI 模式下, virtual channel1 传输的上 一帧图像高度。	0x0000
[15:0]	RO	imgwidth_statis_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×0x1000 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	RO	imgheight_statis_vc2	MIPI 模式下, virtual channel2 传输的上 一帧图像高度。	0x0000
[15:0]	RO	imgwidth_statis_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_statis_vc3	MIPI 模式下, virtual channel3 传输的上 一帧图像高度。	0x0000
[15:0]	RO	imgwidth_statis_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



Bits	Access	Name	Description	Reset
[15:5]	-	reserved	保留。	0x00
[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
[15:5]	-	reserved	保留。	0x000
[4]	RO	int_vsync_st	MIPI CTR vsync 中断状态。 0: 无中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	
[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



Bits	Access	Name	Description	Reset
[1]	RW	int_dfifo_wrerr_mask	MIPI CTRL 写数据 FIFO 中断使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[0]	RW	int_cfifo_wrerr_mask	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。 0000: SOF-EOF 标识的 WDR, 长短曝光都有独立的同步码; 0010: 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; 0100: 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; 0101: 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; 0110: SONY DOL 模式, 5 个字段的同步码, 第 5 个字段标识长短帧;	0x0



Bits	Access	Name	Description	Reset
			1000: SONY IMX136 模式, 帧 WDR 模式, 不要打开 WDR_EN; 1001: SHARP 长短曝光两行交错方式, 长帧在前短帧在后; 1010: SHARP 长短曝光两行交错方式, 短帧在前长帧在后; 其它: 保留。	
[7:6]	-	reserved	保留。	0x0
[5:4]	RW	lvds_wdr_num	WDR 模式配置信号。 01: 2 帧 WDR; 10: 3 帧 WDR; 其他: 保留。	0x0
[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×0x1000 Total Reset Value: 0x0000_0330

Bits	Access	Name	Description	Reset
[31:19]	-	reserved	保留。	0x0000
[18:16]	-	reserved	保留。	0x0
[15:13]	-	reserved	保留。	0x0
[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, bit10, ...bit0。	0x1
[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;	0x3



Bits	Access	Name	Description	Reset
			011: Raw 12bit; 100: Raw 14bit; 101: Raw 16bit; 其它: 保留。	
[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×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 0 帧裁剪寄存器。

Offset Address: 0x1310 + N×0x1000 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	RW	lvds_start_y0	第 0 帧裁剪起始点高度。	0x0000
[15:0]	RW	lvds_start_x0_lane	第 0 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 1 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 1 帧裁剪起始点高度。	0x0000
[15:0]	RW	lvds_start_x1_lane	第 1 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 2 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 2 帧裁剪起始点高度。	0x0000
[15:0]	RW	lvds_start_x2_lane	第 2 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 3 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 3 帧裁剪起始点高度。	0x0000
[15:0]	RW	lvds_start_x3_lane	第 3 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; 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; 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; 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; 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; 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; 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; 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; 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; 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; 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; 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; 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; 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; 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; 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; 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; 仅在 WDR 模式下可配置, linear 模式下配置无效。	0x0000



Bits	Access	Name	Description	Reset
[15:0]	RW	lane(m)_nxt_sof_2	LVDS、HiSPi 模式下帧 2 的帧起始同步码; 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; 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; 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; 仅在 WDR 模式下可配置, linear 模式下配	0x0000



Bits	Access	Name	Description	Reset
			置无效。	
[15:0]	RW	lane(m)_nxt_eof_2	LVDS、HiSPi 模式下帧 2 的帧结束同步码; 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; 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; 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;	0x0000



			仅在 WDR 模式下可配置，linear 模式下配置无效。	
[15:0]	RW	lane(m)_nxt_sol_2	LVDS、HiSPi 模式下帧 2 的行起始同步码； 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； 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； 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； 仅在 WDR 模式下可配置，linear 模式下配	0x0000



			置无效。	
[15:0]	RW	lane(m)_nxt_eol_2	LVDS、HiSPi 模式下帧 2 的行结束同步码； 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×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×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×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)_OUTPUT_PIX_NUM

LVDS(N)_OUTPUT_PIX_NUM 为 LVDS 输出像素位宽选择寄存器。

Offset Address: $0x17A0 + N \times 0x1000$ Total Reset Value: $0x0000_0000$

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	lvds_double_pix_en	LVDS 模式输出像素位宽选择。 0: 一拍一像素; 1: 保留。	0x0

LVDS(N)_16to14_EN

LVDS(N)_16to14_EN 为 LVDS 16 转 14 时序使能寄存器。

Offset Address: $0x17A4 + N \times 0x1000$ Total Reset Value: $0x0000_0000$

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	lvds_16to14_en	LVDS 16 转 14 时序使能。 0: 不使能; 1: 使能。	0x0

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



Bits	Access	Name	Description	Reset
[27:25]	-	reserved	保留。	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
[23:21]	-	reserved	保留。	0x0
[20]	WC	data_rd_err_raw	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	data_wr_err_raw	写 FIFO 错误原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[15:8]	-	reserved	保留。	0x00
[7]	WC	lane7_sync_err_raw	Lane7 同步错误原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[6]	WC	lane6_sync_err_raw	Lane6 同步错误原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[5]	WC	lane5_sync_err_raw	Lane5 同步错误原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[4]	WC	lane4_sync_err_raw	Lane4 同步错误原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[3]	WC	lane3_sync_err_raw	Lane3 同步错误原始中断状态。	0x0



Bits	Access	Name	Description	Reset
			0: 无原始中断; 1: 有原始中断。	
[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×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
[27:25]	-	reserved	保留。	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	data_rd_err_st	读 FIFO 错误中断状态。	0x0



Bits	Access	Name	Description	Reset
			0: 无中断; 1: 有中断。	
[19:17]	-	reserved	保留。	0x0
[16]	RO	data_wr_err_st	写 FIFO 错误中断状态。 0: 无中断; 1: 有中断。	0x0
[15:8]	-	reserved	保留。	0x0
[7]	RO	lane7_sync_err_st	Lane7 同步错误中断状态。 0: 无中断; 1: 有中断。	0x0
[6]	RO	lane6_sync_err_st	Lane6 同步错误中断状态。 0: 无中断; 1: 有中断。	0x0
[5]	RO	lane5_sync_err_st	Lane5 同步错误中断状态。 0: 无中断; 1: 有中断。	0x0
[4]	RO	lane4_sync_err_st	Lane4 同步错误中断状态。 0: 无中断; 1: 有中断。	0x0
[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;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	
[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
[27:25]	-	reserved	保留。	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	data_rd_err_msk	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	data_wr_err_msk	写 FIFO 错误中断状态使能。 0: 无原始中断; 1: 有原始中断。	0x0



Bits	Access	Name	Description	Reset
[15:8]	-	reserved	保留。	0x0
[7]	RW	lane7_sync_err_msk	Lane7 同步错误中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[6]	RW	lane6_sync_err_msk	Lane6 同步错误中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[5]	RW	lane5_sync_err_msk	Lane5 同步错误中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[4]	RW	lane4_sync_err_msk	Lane4 同步错误中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[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。 0: 屏蔽中断; 1: 使能中断。	0x0
[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, 7]。 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, 7]。 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, 7]。 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, 7]。 取值表示为连接到 Lane0 的图像传感器的实际通道号。	0x0

LANE_ID1_CHN(N)

LANE_ID1_CHN(N)为 LANE 优先级配置寄存器 1。

Offset Address: $0x1804 + N \times 0x1000$ Total Reset Value: $0x0000_7654$

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:12]	RW	lane7_id	指示 Lane7 的通道号，取值范围为[0, 7]。 取值表示为连接到 Lane7 的图像传感器的实际通道号。	0x7
[11:8]	RW	lane6_id	指示 Lane6 的通道号，取值范围为[0, 7]。 取值表示为连接到 Lane6 的图像传感器的	0x6



			实际通道号。	
[7:4]	RW	lane5_id	指示 Lane5 的通道号，取值范围为[0, 7]。 取值表示为连接到 Lane5 的图像传感器的实际通道号。	0x5
[3:0]	RW	lane4_id	指示 Lane4 的通道号，取值范围为[0, 7]。 取值表示为连接到 Lane4 的图像传感器的实际通道号。	0x4

ALIGN(N)_INT_RAW

ALIGN(N)_INT_RAW 为 MIPI_ALIGN 原始中断状态寄存器。

Offset Address: 0x18F0 + N×0x1000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:9]	-	reserved	保留。	0x000000
[8]	WC	err_lane7_raw	ALIGN lane7 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[7]	WC	err_lane6_raw	ALIGN lane6 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[6]	WC	err_lane5_raw	ALIGN lane5 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[5]	WC	err_lane4_raw	ALIGN lane4 原始中断状态。 0: 无原始中断; 1: 有原始中断。	0x0
[4]	WC	err_lane3_raw	ALIGN lane3 原始中断状态。 0: 无原始中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有原始中断。	
[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
[31:9]	-	reserved	保留。	0x000000
[8]	RO	err_lane7_st	ALIGN lane7 中断状态。 0: 无中断; 1: 有中断。	0x0
[7]	RO	err_lane6_st	ALIGN lane6 中断状态。 0: 无中断; 1: 有中断。	0x0
[6]	RO	err_lane5_st	ALIGN lane5 中断状态。	0x0



Bits	Access	Name	Description	Reset
			0: 无中断; 1: 有中断。	
[5]	RO	err_lane4_st	ALIGN lane4 中断状态。 0: 无中断; 1: 有中断。	0x0
[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
[31:9]	-	reserved	保留。	0x000000



Bits	Access	Name	Description	Reset
[8]	RW	err_lane7_mask	ALIGN lane7 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[7]	RW	err_lane6_mask	ALIGN lane6 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[6]	RW	err_lane5_mask	ALIGN lane5 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[5]	RW	err_lane4_mask	ALIGN lane4 中断状态使能。 0: 屏蔽中断; 1: 使能中断。	0x0
[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;	0x0



			1: 有中断。	
[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(N)_INT_MASK

CHN(N)_INT_MASK 为通道(N)中断屏蔽寄存器。

Offset Address: $0x1FF8 + N \times 0x1000$ Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000
[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;	0x0



			1: 使能中断。	
--	--	--	----------	--

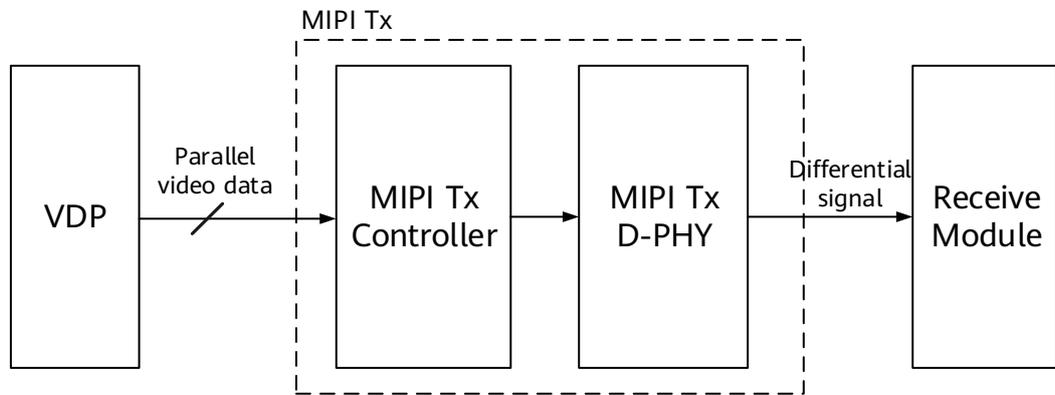
10.4 MIPI Tx

10.4.1 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 接口的应用层支持 Mipi DSI、Mipi CSI-2 协议的发送, 物理层支持 Mipi D-PHY-V1.2 串行输出。

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 10-36 所示。

图10-36 MIPI Tx 功能框图及在系统中的位置



10.4.2 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、顺序可配置、差分线 PN 互换
- 高速模式最大支持 1800Mbps/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 video mode 和 command mode，video mode 支持 Burst mode、Non-burst Mode with Sync Pulses 和 Non-burst Mode with Sync Events，command mode 支持 TE 信号输入
- Video mode 支持 DSI RGB 16-bit/18-bit/loosely18-bit/24-bit、YUV4:2:0 12-bit/YUV 4:2:2 16-bit 数据类型的输出
- command mode 支持 DSI RGB 16-bit/18-bit/24-bit 等数据类型的输出。
- 支持 CSI-2 Legacy YUV420 8-bit/YUV420 8-bit/YUV422 8-bit/RAW8/RAW10/RAW12/RAW14/RAW16/RGB888 数据类型的输出

10.4.3 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 ver1.2 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。

10.4.3.1 Tx D-PHY

Tx D-PHY 可以工作在低速和高速（High Speed, HS）两种模式下：

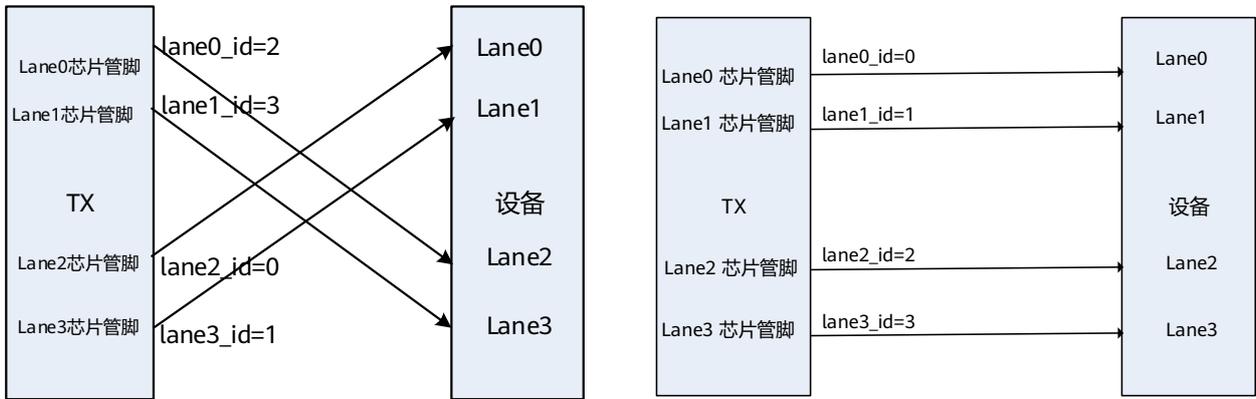
- video mode/CSI-2 的数据通过高速模式传输。
- command mode 的数据可以通过高速或者低速模式来传输。

高速模式每个通道（lane）的数据速率范围为 200~1800Mbps，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，data lane 顺序可以配置，默认情况下当仅使用一个 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。当需要改变 data lane 顺序时，需要配置 lane id 寄存器，lane id 的值表示连接到该 lane 的设备端的 lane 号，用法参考图 10-37；lane id 只能取值 0/1/2/3，而且各个 lane 的 id 值不能配置一样，不管当前使能了多少 lane。仅 id 号为 0 的 lane 支持低速的发送、接收和 Bus Turn-Around。不管使用几条 lane，都必须从 id 号为 0 开始连续配置使用，每条通道支持 initial 和 periodic calibration sequence 的发送。



图10-37 lane id 配置方法举例，左图{lane0_id,lane1_id,lane2_id,lane3_id}={2,3,0,1},右图为{0,1,2,3}



10.4.3.2 Tx Controller

数据包的发送

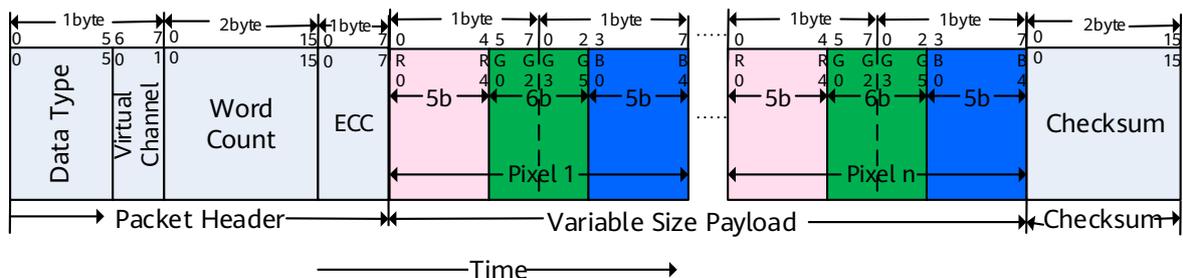
当多个数据包需要发送时，控制器会根据间隔时间自动进行 HS 和 LP 模式的切换。控制器支持 EoT 包 (End of Transmission, EoTp) 发送的使能和关闭，具体配置寄存器 PCKHDL_CFG bit[0]，可以在每个 HS 传输结束时发送一个 EoT 包，也可以不发送。

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/RAW8/RAW10/RAW12/RAW14/RAW16/RGB888 的发送，各种数据类型的数据组成格式如图 10-38~图 10-53 所示。

DSI RGB 16-bit，每两个字节包含一个像素。

图10-38 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 10-39 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 10-40 所示。

图10-39 RGB 18-bit non-loosely 格式

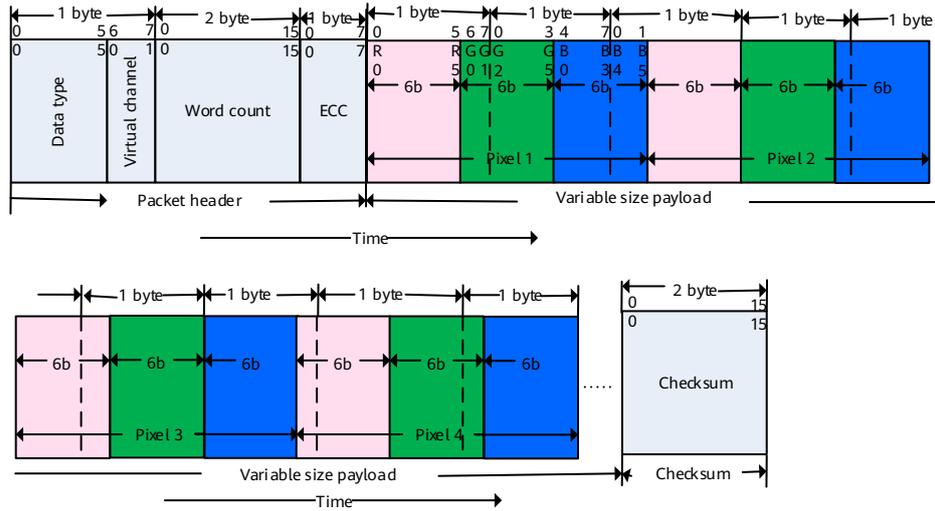
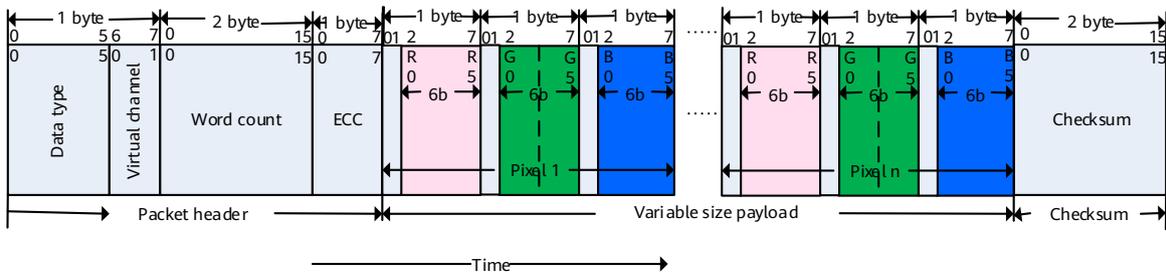


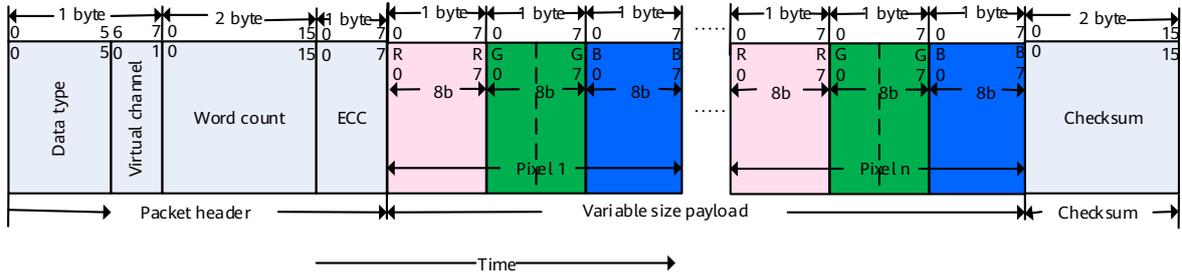
图10-40 RGB 18-bit loosely 格式



DSI RGB 24-bit, 一个分量占用一个字节, 一个 pixel 占用 3 个字节。

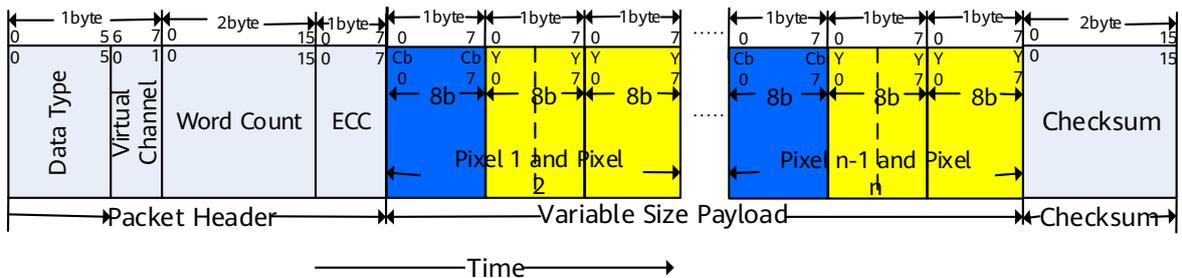


图10-41 RGB 24-bit 格式



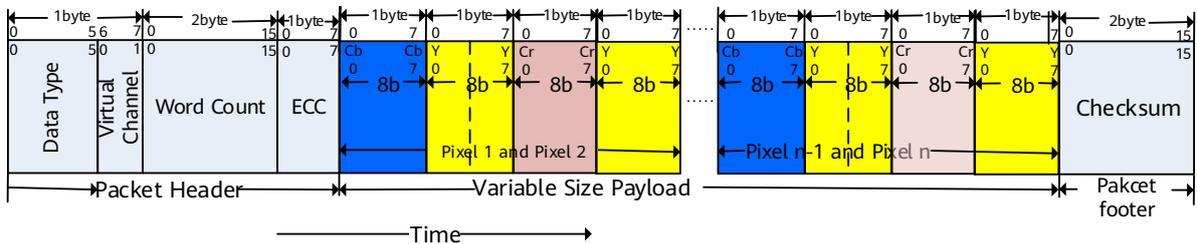
DSI YUV4:2:0 12-bit, 奇数行发 Cb 和 Y 分量, 偶数行发 Cr 和 Y 分量。每个数据包的像素个数是 2 的倍数。

图10-42 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。

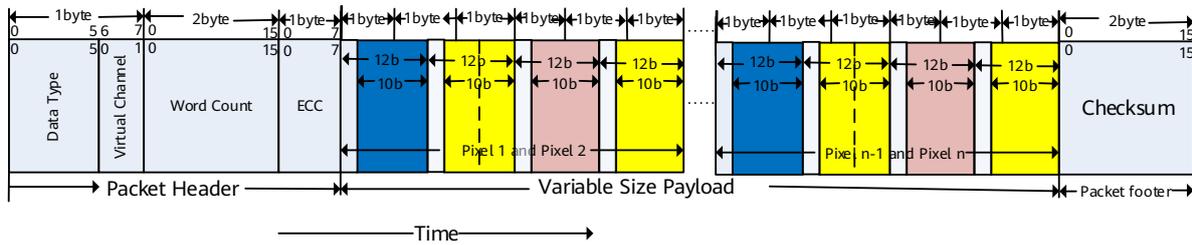
图10-43 YUV4:2:2 16-bit 格式



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。

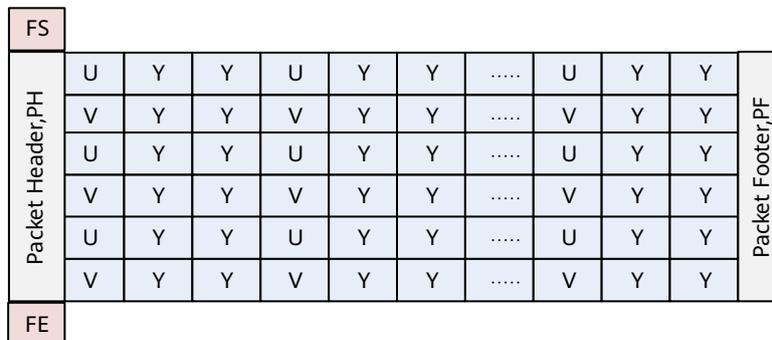
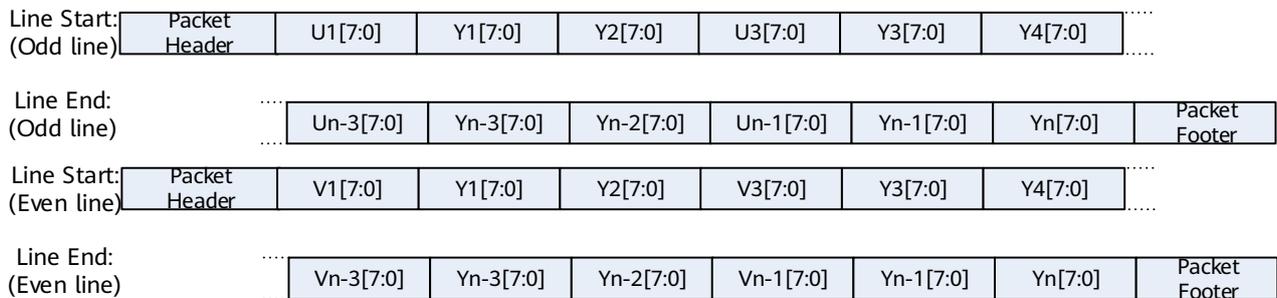


图10-44 YUV4:2:2 20-bit 格式



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。

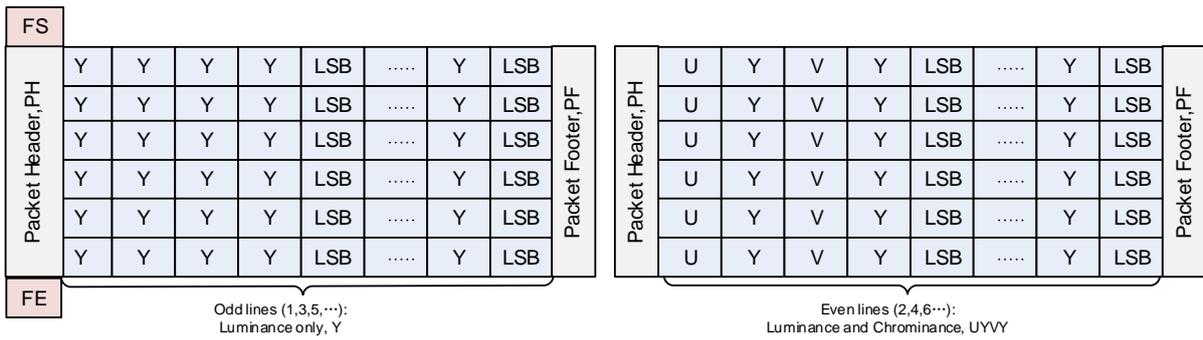
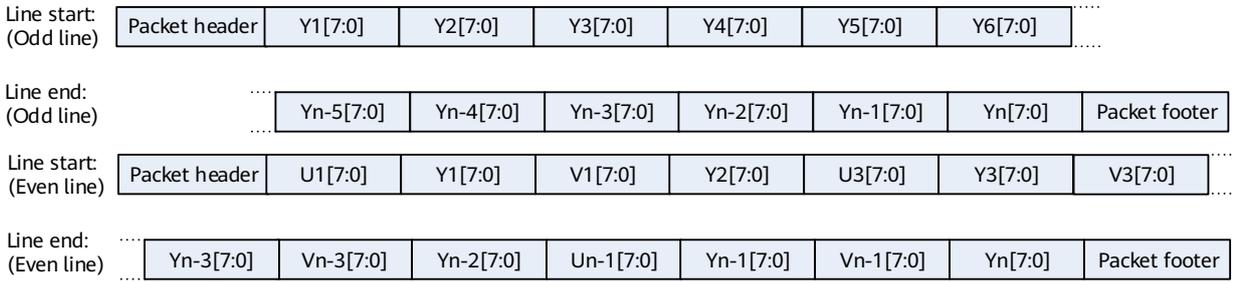
图10-45 Legacy YUV420 8-bit 格式



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10-46 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10-47 YUV422 8-bit 格式

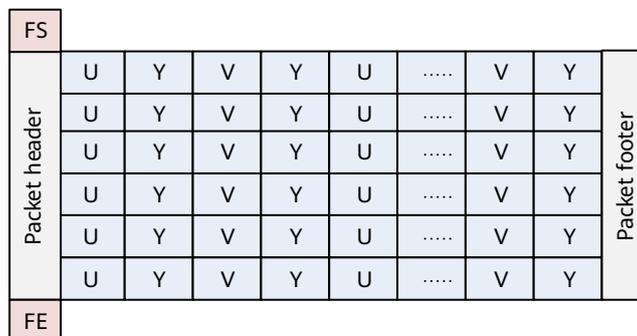
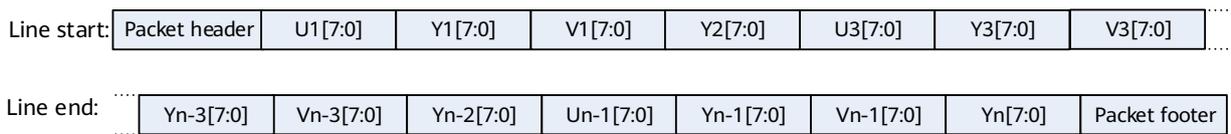




图10-48 CSI RAW8 格式

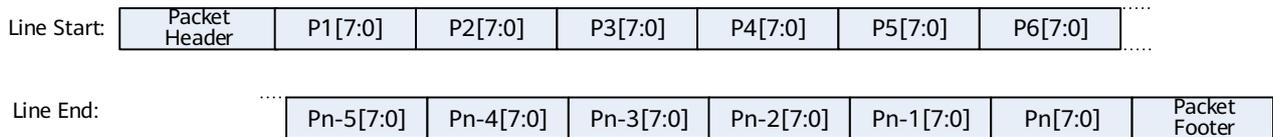


图10-49 CSI RAW10 格式

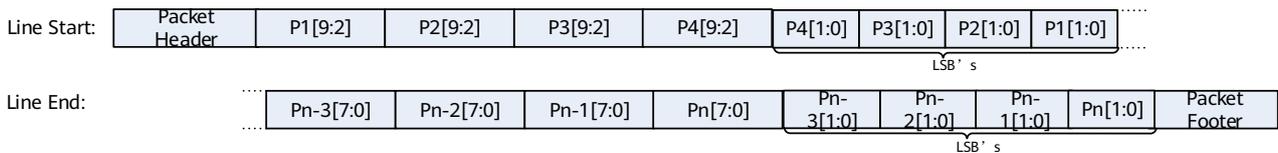


图10-50 CSI RAW12 格式

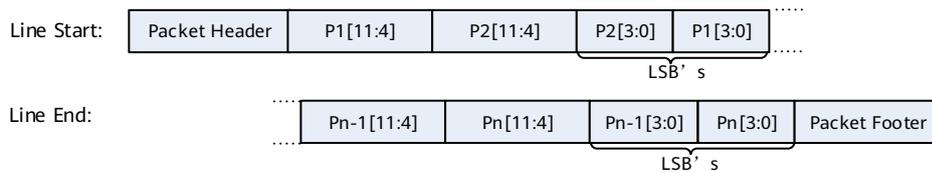


图10-51 CSI RAW14 格式

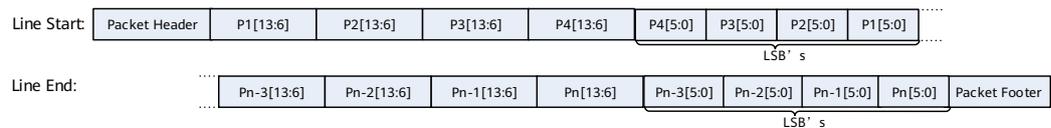


图10-52 CSI RAW16 格式

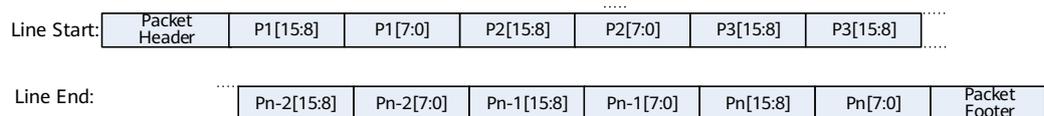
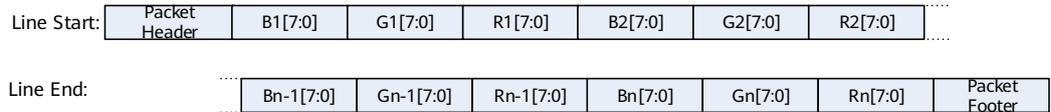


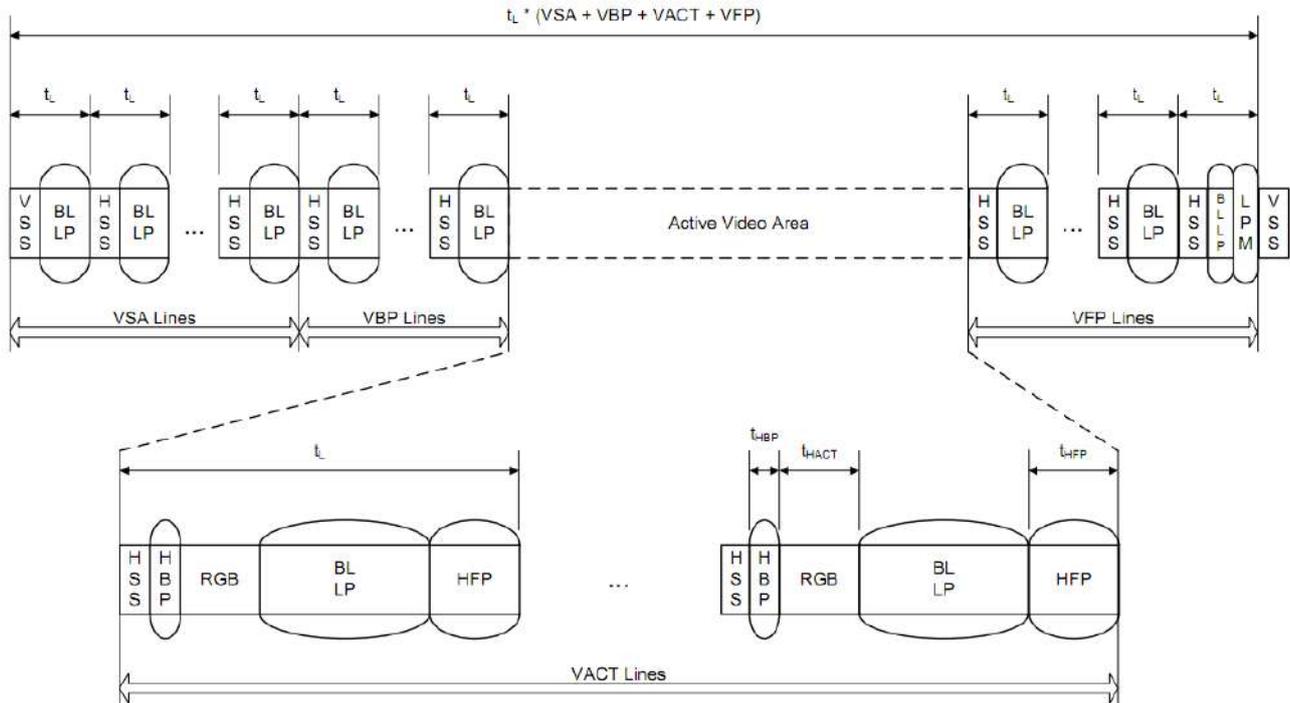
图10-53 CSI RGB888 格式



接口时序

Video mode 的 Burst Mode 时序如图 10-54 所示，一帧图像包含 VSA (vertical sync active), VBP (vertical back porch), VACT (vertical active lines) 和 VFP (vertical front porch) 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。

图10-54 Burst Mode

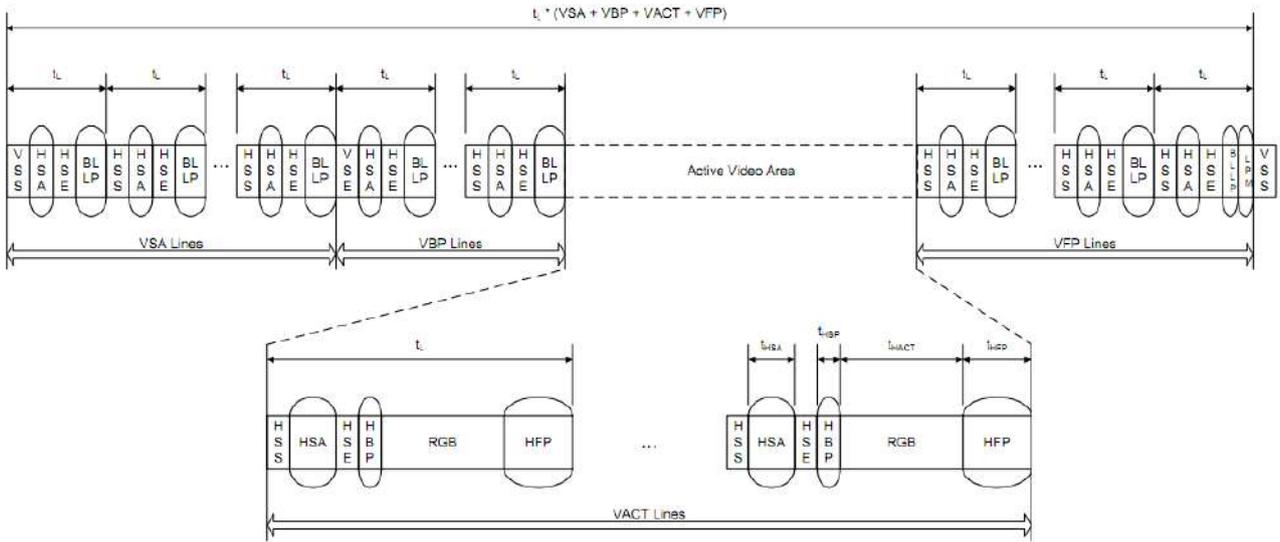


Video mode 的 Non-Burst Mode with Sync Pulses 时序如图 10-55 所示，一帧图像包含 VSA, VBP, VACT 和 VFP 四部分。有效数据行包括行 sync start, 行 start active



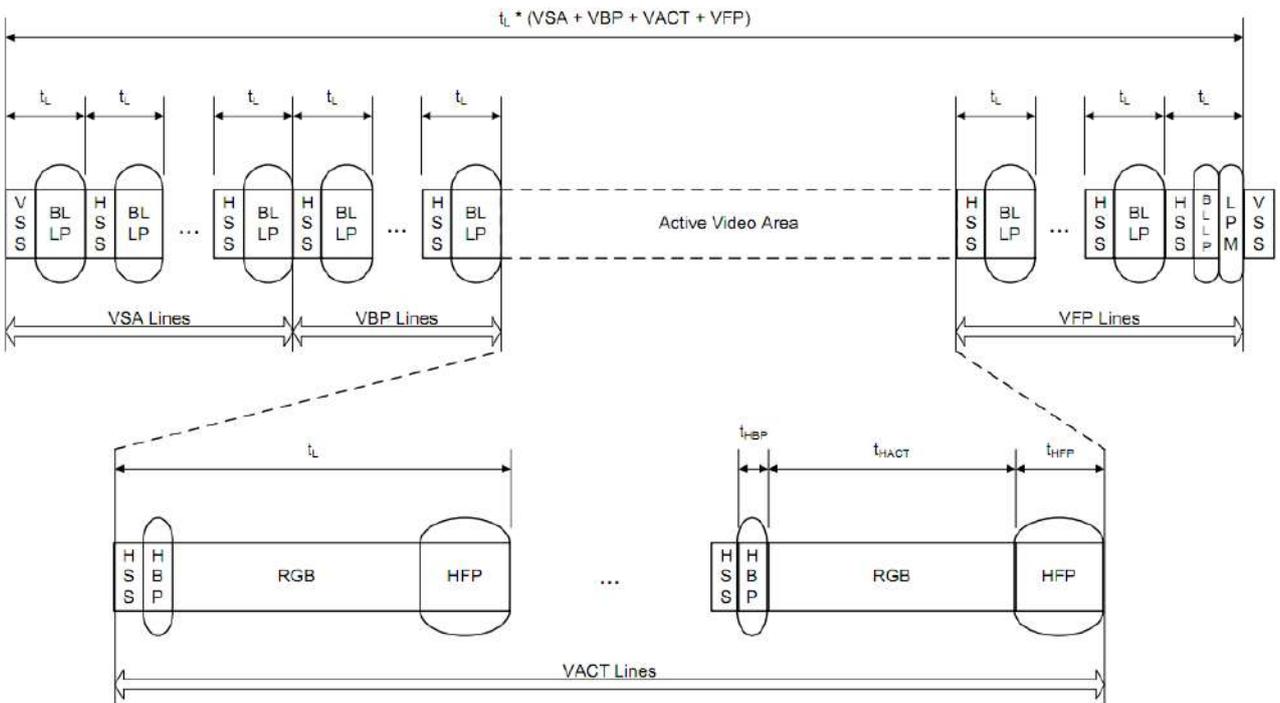
区, 行 sync end, 行后消隐区, 行有效区, 行前消隐区六部分, 无效数据行包含行 sync start, 行 start active 区, 行 sync end 和 BLLP。

图10-55 Non-Burst Mode with Sync Pulses



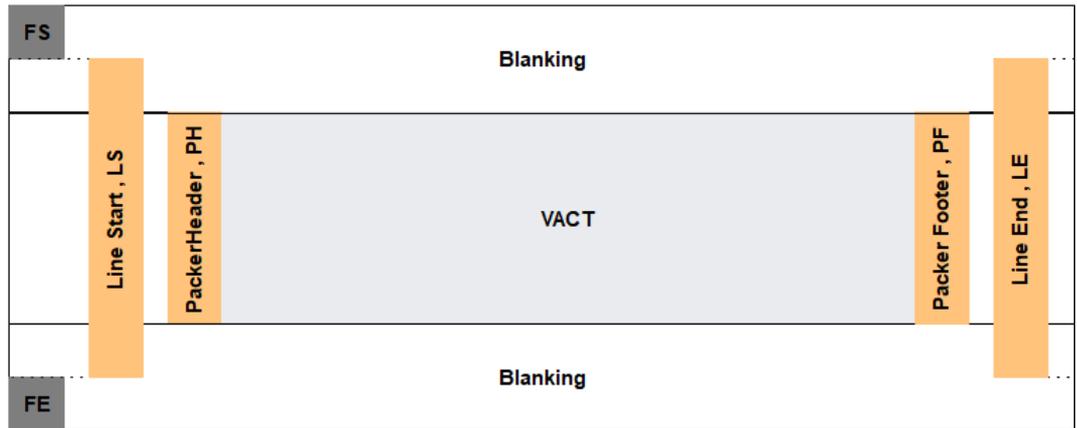
Video mode 的 Non-Burst Mode with Sync Events 时序如图 10-56 所示, 一帧图像包含 VSA, VBP, VACT 和 VFP 四部分。有效数据行包括行 sync start, 行后消隐区, 行有效区, 行前消隐区四部分, 无效数据行只包含行 sync start 和 BLLP。

图10-56 Non-Burst Mode with Sync Events



CSI-2 发送的帧时序如图 10-57 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。FS 包可在帧起始发送，也可在 VBP 第一行或最后一行发送。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。同时支持每行发送 LS、LE 包。

图10-57 CSI-2 帧时序



10.4.4 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。

应该根据不同的场景来配置，寄存器具体描述请参考“10.4.6 MIPI Tx 寄存器描述”章节。

步骤 2 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~1800Mbps。

步骤 3 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。此时要把控制器配置成 command 模式。

配置方法：配置命令写入寄存器 `COMMAND_HEADER`，长包的配置数据写入寄存器 `COMMAND_PAYLOAD`。先写配置数据，再写配置命令。

举例：用 generic long write (data type: 29h) 写 6 个字节数据：



COMMAND_PAYLOAD =32'h04030201; COMMAND_PAYLOAD =32'h00000605;
COMMAND_HEADER =32'h00000629;

用 DCS Short WRITE, 1parameter (data type: 15h) 写数据 35h:
COMMAND_HEADER =32'h00003515;

当用读命令获取设备信息时, 读操作完成后, 通过 COMMAND_PAYLOAD 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 COMMAND_PAYLOAD 返回 4 字节。

步骤 4 配置控制器工作模式, 分辨率。

应注意各参数的单位, 详细参考寄存器描述。

步骤 5 配置前面模块 (VDP), 给 TX 输入视频内容。

----结束

10.4.5 MIPI Tx 寄存器概览

MIPI Tx 寄存器概览如表 10-13 所示。

表10-13 MIPI Tx 寄存器概览 (基址是 0x0_1027_0000)

偏移地址	名称	描述	页码
0x0000	CTRL_RESET	复位寄存器	10-286
0x0008	CRG_CFG	时钟分频寄存器	10-286
0x000C	VIDEO_VC	视频数据 virtual channel ID 配置寄存器	10-286
0x0010	DATA_TYPE	数据格式配置寄存器	10-287
0x0018	LP_CMD_BYTE	video 模式 LP 命令可用时间配置寄存器	10-288
0x002C	PCK_EN	特定包使能寄存器	10-289
0x0030	GEN_VC	Generic 接口 virtual channel ID 配置寄存器	10-290
0x0034	MODE_CFG	模式配置寄存器	10-290



偏移地址	名称	描述	页码
0x0038	VIDEO_LP_EN	video 模式 LP 使能寄存器	10-291
0x003C	VIDEOM_PKT_SIZE	VIDEO 模式 packet 大小配置寄存器	10-292
0x0040	VIDEOM_NUM_CHUNKS	一行 chunks 个数配置寄存器	10-292
0x0044	VIDEOM_NULL_SIZE	null packet 大小配置寄存器	10-293
0x0048	VIDEOM_HSA_SIZE	HSA 长度配置寄存器	10-293
0x004C	VIDEOM_HBP_SIZE	HBP 长度配置寄存器	10-293
0x0050	VIDEOM_HLINE_SIZE	HLINE 长度配置寄存器	10-294
0x0054	VIDEOM_VSA_LINES	VSA 行数配置寄存器	10-294
0x0058	VIDEOM_VBP_LINES	VBP 行数配置寄存器	10-294
0x005C	VIDEOM_VFP_LINES	VFP 行数配置寄存器	10-295
0x0060	VIDEOM_VACTIVE_LINES	VACTIVE 行数配置寄存器	10-295
0x0064	COMMAND_PKT_SIZE	COMMAND 模式 packet 大小配置寄存器	10-295
0x0068	COMMAND_TRAN_MODE	命令传输模式配置寄存器	10-295
0x006C	COMMAND_HEADER	命令 header 设置寄存器	10-297
0x0070	COMMAND_PAYLOAD	命令发送的数据寄存器	10-298
0x0074	COMMAND_STATUS	命令状态寄存器	10-298
0x0088	LPWR_TO_SET	LP 写超时配置寄存器	10-299
0x008C	BTA_TO_SET	BTA 回复超时配置寄存器	10-300
0x0094	CLKLANE_CTRL	clklane 控制寄存器	10-300
0x0098	CLKLANE_TIME	clklane HS 和 LP 切换时间配置寄存器	10-300



偏移地址	名称	描述	页码
0x009C	DATALANE_TIME	datalane HS 和 LP 切换时间配置寄存器	10-301
0x00A0	PHY_CTRL	PHY PLL 和复位配置寄存器	10-301
0x00A4	LANE_NUM	PHY 接口配置寄存器	10-302
0x00A8	ULPS_CTRL	PHY 的 ULPS 状态配置寄存器	10-302
0x00B0	PPI_STATUS	PHY 状态寄存器	10-303
0x00B4	PHY_REG_CFG0	PHY 配置控制 0 寄存器	10-305
0x00B8	PHY_REG_CFG1	PHY 配置接口 1 寄存器	10-305
0x00BC	INT0_STATUS	中断 0 状态寄存器	10-306
0x00C0	INT1_STATUS	中断 1 状态寄存器	10-309
0x00C4	INT0_MASK	中断 0 屏蔽寄存器	10-312
0x00C8	INT1_MASK	中断 1 屏蔽寄存器	10-315
0x00CC	PHY_CAL	PHY calibration 使能寄存器	10-318
0x01B0	POLARITY_CTRL	前级模块输入控制信号极性控制寄存器	10-318
0x024C	GINT_MSK	总中断屏蔽配置寄存器	10-319
0x0308	OPERATION_MODE	工作模式控制寄存器	10-319
0x0330	DATATYPE0	datatype 配置寄存器 0	10-321
0x0338	CSI_CTRL	CSI 时序配置寄存器	10-322
0x033C	SKEW_BEGIN	period skew 开始发送位置寄存器	10-323
0x0340	SKEW_END	period skew 结束发送位置寄存器	10-323
0x0350	LANE_ID	lane id 配置寄存器	10-324



10.4.6 MIPI Tx 寄存器描述

说明

寄存器描述中的 hclk period=1/(Data Rate/8)。

CTRL_RESET

CTRL_RESET 为复位寄存器。

Offset Address: 0x0000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	ctrl_reset	控制逻辑复位使能，不影响寄存器配置值。 0: 复位; 1: 撤销复位。	0x0

CRG_CFG

CRG_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。	0x0000
[7:0]	RW	txescclk_div	TX escape clock 分频倍数，时钟源为 hclk，配 0 和 1 将不会产生时钟。(本模块中 hclk 频率为 Data Rate/8)	0x00

VIDEO_VC

VIDEO_VC 为视频数据 virtual channel ID 配置寄存器。

Offset Address: 0x000C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x0000000
[4]	RW	vcid_sel	视频数据 virtual channel ID 来源选择。 0: 来自寄存器 vcid; 1: 来自前级模块。	0x0
[3:2]	-	reserved	保留。	0x0
[1:0]	RW	vcid	视频数据 virtual channel ID 配置值。	0x0

DATA_TYPE

DATA_TYPE 为数据格式配置寄存器。

Offset Address: 0x0010 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x0000000
[7:0]	RW	data_type	video 模式: 0x00: DSI RGB16-bit; 0x03: DSI RGB18-bit; 0x04: DSI RGB18-bit loosely; 0x05: DSI RGB24-bit; 0x06: DSI YCbCr422 20-bit; 0x08: DSI YCbCr422 16-bit; 0x09: DSI RGB30-bit; 0x0A: DSI RGB36-bit; 0x0B: DSI YCbCr420 12-bit; 0x0F: DSC24 compressed Data; 0x10: CSI YUV422 8-bit; 0x11: CSI YUV422 10-bit; 0x12: CSI legacy YUV420 8-bit; 0x13: CSI YUV420 8-bit;	0x00



Bits	Access	Name	Description	Reset
			0x14: CSI YUV420 10-bit; 0x15: CSI RGB888; 0x16: CSI RGB666; 0x17: CSI RGB565; 0x18: CSI RGB555; 0x19: CSI RGB444; 0x1A: CSI RAW8; 0x1B: CSI RAW10; 0x1C: CSI RAW12; 0x1D: CSI RAW14; 0x1E: CSI RAW16; 其他: 保留。 command 模式: 0x00: DSI RGB16-bit; 0x03: DSI RGB18-bit; 0x04: DSI RGB18-bit loosely; 0x05: DSI RGB24-bit; 0x09: DSI RGB30-bit; 0x0A: DSI RGB36-bit; 0x0F: DSC24 compressed Data; 其他: 保留。	

LP_CMD_BYTE

LP_CMD_BYTE 为 video 模式 LP(Low Power)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_byt	VSA(vertical sync active),VBP(vertical	0x00



		e	back porch)和 VFP(vertical front porch)区域可以发送的 LP 命令的最大字节数。	
[15:8]	-	reserved	保留。	0x00
[7:0]	RW	invact_lpcmd_byte	VACT(vertical active lines)区域可以发送的 LP 命令的最大字节数。	0x00

PCK_EN

PCK_EN 为特定包使能寄存器。

Offset Address: 0x002C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RW	frame_ack_en	帧结束 ACK 回复请求使能。 0: 不使能; 1: 使能。	0x0
[6]	-	reserved	保留。	0x0
[5]	-	reserved	保留。	0x0
[4]	RW	bta_en	Bus Turn-Around 请求。 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	crc_rx_en	CRC 接收使能。 0: 不使能; 1: 使能。	0x0
[1]	RW	eotp_rx_en	EoTp 接收使能。 0: 不使能;	0x0



Bits	Access	Name	Description	Reset
			1: 使能。	
[0]	RW	eotp_tx_en	EoTp 发送使能。 0: 不使能; 1: 使能。	0x0

GEN_VC

GEN_VC 为 Generic 接口 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	Generic 接口 virtual channel ID 值。	0x0

MODE_CFG

MODE_CFG 为模式配置寄存器。

Offset Address: 0x0034 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x000
[9:8]	RW	video_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
[7:5]	-	reserved	保留。	0x0
[4]	RW	cmd_mode_tran_type	在 video 模式时, 命令发送的方式: 0: 高速模式; 1: 低速模式。	0x0



Bits	Access	Name	Description	Reset
[3:1]	-	reserved	保留。	0x0
[0]	RW	cmd_video_mode	video 和 command 模式选择。 0: video 模式; 1: command 模式。	0x0

VIDEO_LP_EN

VIDEO_LP_EN 为 video 模式 LP 使能寄存器。

Offset Address: 0x0038 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x00000000
[5]	RW	hfp_lp_en	如果时间足够, HFP 区域返回 low-power 状态使能。如果不使能, 该区域将总是处于 High speed 状态。 0: 不使能; 1: 使能。	0x0
[4]	RW	hbp_lp_en	如果时间足够, HBP(horizontal back porch)区域返回 low-power 状态使能。如果不使能, 该区域将总是处于 High speed 状态。 0: 不使能; 1: 使能。	0x0
[3]	RW	vact_lp_en	如果时间足够, VACT 区域返回 low-power 状态使能。如果不使能, 该区域将总是处于 High speed 状态。 0: 不使能; 1: 使能。	0x0
[2]	RW	vfp_lp_en	如果时间足够, VFP 区域返回 low-power 状态使能。如果不使能, 该区域将总是处	0x0



Bits	Access	Name	Description	Reset
			于 High speed 状态。 0: 不使能; 1: 使能。	
[1]	RW	vbp_lp_en	如果时间足够, VBP 区域返回 low-power 状态使能。如果不使能, 该区域将总是处于 High speed 状态。 0: 不使能; 1: 使能。	0x0
[0]	RW	vsa_lp_en	如果时间足够, VSA 区域返回 low-power 状态使能。如果不使能, 该区域将总是处于 High speed 状态。 0: 不使能; 1: 使能。	0x0

VIDEOM_PKT_SIZE

VIDEOM_PKT_SIZE 为 VIDEO 模式 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	videom_pkt_size	video 模式 packet 大小, 单位 pixel。	0x0000

VIDEOM_NUM_CHUNKS

VIDEOM_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	videom_num_chunks	video 模式一行的 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
--------	----	-------------------	--	--------

VIDEOM_NULL_SIZE

VIDEOM_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	videom_null_size	video 模式 null packet 大小, 单位字节。如果为 0, 不发送 null packet。	0x0000

VIDEOM_HSA_SIZE

VIDEOM_HSA_SIZE 为 HSA 长度配置寄存器。

Offset Address: 0x0048 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:12]	-	reserved	保留。	0x00000
[11:0]	RW	videom_hsa_size	HSA(horizontal sync active)长度, 单位 hsclock 周期。	0x000

VIDEOM_HBP_SIZE

VIDEOM_HBP_SIZE 为 HBP 长度配置寄存器。

Offset Address: 0x004C Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:12]	-	reserved	保留。	0x00000
[11:0]	RW	videom_hbp_size	HBP(horizontal back porch)长度, 单位 hclk 周期。	0x000

VIDEOM_HLINE_SIZE

VIDEOM_HLINE_SIZE 为 HLINE(Horizontal Line)长度配置寄存器。

Offset Address: 0x0050 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	保留。	0x00000
[14:0]	RW	video_hline_size	一行总长度(HSA+HBP+HACT+HFP), 单位 hclk 周期。	0x0000

VIDEOM_VSA_LINES

VIDEOM_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

VIDEOM_VBP_LINES

VIDEOM_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
-------	----	-----------	---------	-------

VIDEOM_VFP_LINES

VIDEOM_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

VIDEOM_VACTIVE_LINES

VIDEOM_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

COMMAND_PKT_SIZE

COMMAND_PKT_SIZE 为 COMMAND 模式 packet 大小配置寄存器。

Offset Address: 0x0064 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	cmd_pkt_size	command 模式 packet 大小, 单位 pixel。	0x0000

COMMAND_TRAN_MODE

COMMAND_TRAN_MODE 为命令传输模式配置寄存器。



Offset Address: 0x0068 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:12]	-	reserved	保留。	0x00000
[11]	RW	dcs_lw	DCS(Display Command Set) long write packet command 传输方式。 0: high-speed; 1: low-power.	0x0
[10]	RW	dcs_sr_0p	DCS short read packet with zero parameter command 传输方式。 0: high-speed; 1: low-power.	0x0
[9]	RW	dcs_sw_1p	DCS short write packet with one parameter command 传输方式。 0: high-speed; 1: low-power.	0x0
[8]	RW	dcs_sw_0p	DCS short write packet with zero parameter command 传输方式。 0: high-speed; 1: low-power.	0x0
[7]	RW	gen_lw	Generic long write packet command 传输方式。 0: high-speed; 1: low-power.	0x0
[6]	RW	gen_sr_2p	Generic short read packet with two parameters command 传输方式。 0: high-speed; 1: low-power.	0x0
[5]	RW	gen_sr_1p	Generic short read packet with one parameters command 传输方式。 0: high-speed; 1: low-power.	0x0



Bits	Access	Name	Description	Reset
[4]	RW	gen_sr_0p	Generic short read packet with zero parameters command 传输方式。 0: high-speed; 1: low-power.	0x0
[3]	RW	gen_sw_2p	Generic short write packet with two parameters command 传输方式。 0: high-speed; 1: low-power.	0x0
[2]	RW	gen_sw_1p	Generic short write packet with one parameters command 传输方式。 0: high-speed; 1: low-power.	0x0
[1]	RW	gen_sw_0p	Generic short write packet with zero parameters command 传输方式。 0: high-speed; 1: low-power.	0x0
[0]	RW	max_rd_pkt_size_tran	maximun read packet size command 传输方式。 0: high-speed; 1: low-power.	0x0

COMMAND_HEADER

COMMAND_HEADER 为命令 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	command_wordcount_hi	长包长度高字节, 或者短包的 data 1。	0x00



[15:8]	RW	command_wordcount_low	长包长度低字节, 或者短包的 data 0。	0x00
[7:6]	RW	command_virtualchannel	命令包的 virtual channel。	0x0
[5:0]	RW	command_datatype	命令包的 data type。	0x00

COMMAND_PAYLOAD

COMMAND_PAYLOAD 为命令发送的数据寄存器。

Offset Address: 0x0070 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RW	command_pld_b4	命令包数据的第 4 字节。	0x00
[23:16]	RW	command_pld_b3	命令包数据的第 3 字节。	0x00
[15:8]	RW	command_pld_b2	命令包数据的第 2 字节。	0x00
[7:0]	RW	command_pld_b1	命令包数据的第 1 字节。	0x00

COMMAND_STATUS

COMMAND_STATUS 为命令状态寄存器。

Offset Address: 0x0074 Total Reset Value: 0x0000_0095

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RO	command_idle	命令 idle 状态指示。 0: 不处于 idle 状态; 1: 处于 idle 状态。	0x1
[6]	RW	rd_cmd_busy	读命令忙标志。 0: 空闲; 1: 忙。	0x0



[5]	RW	pld_read_full	存储读命令返回数据的 FIFO 满标志。 0: 不满; 1: 满。	0x0
[4]	RW	pld_read_empty	存储读命令返回数据的 FIFO 空标志。 0: 不空; 1: 空。	0x1
[3]	RW	pld_write_full	存储写命令发送数据的 FIFO 满标志。 0: 不满; 1: 满。	0x0
[2]	RW	pld_write_empty	存储写命令发送数据的 FIFO 空标志。 0: 不空; 1: 空。	0x1
[1]	RW	command_full	存储命令的 FIFO 满标志。 0: 不满; 1: 满。	0x0
[0]	RW	command_empty	存储命令的 FIFO 空标志。 0: 不空; 1: 空。	0x1

LPWR_TO_SET

LPWR_TO_SET 为 LP(Low Power)写超时配置寄存器。

Offset Address: 0x0088 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	lpwr_to_set	LP 发送写命令后等待时间, 单位为 hclk。	0x0000



BTA_TO_SET

BTA_TO_SET 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_set	Bus Turn-Around 后 host 等待时间, 单位为 hsclk。	0x0000

CLKLANE_CTRL

CLKLANE_CTRL 为 clklane 控制寄存器。

Offset Address: 0x0094 Total Reset Value: 0x0000_0002

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	clklane_continue	clklane 空闲时间时钟自动关断使能。 0: 不使能; 1: 使能。	0x1
[0]	RW	txrequesthsclk	hsclk 请求使能。 0: 不使能; 1: 使能。	0x0

CLKLANE_TIME

CLKLANE_TIME 为 clklane HS(High Speed)和 LP(Low Power)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	clklane_hs2lp_time	clklane hs 转换到 lp 模式需要的时间, 单	0x000



Bits	Access	Name	Description	Reset
			位 hsclk。	
[15:10]	-	reserved	保留。	0x00
[9:0]	RW	cklane_lp2hs_time	cklane lp 转换到 hs 模式需要的时间, 单位 hsclk。	0x000

DATALANE_TIME

DATALANE_TIME 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	datalane_hs2lp_time	datalane hs 转换到 lp 模式需要的时间, 单位 hsclk。	0x000
[15:10]	-	reserved	保留。	0x00
[9:0]	RW	datalane_lp2hs_time	datalane lp 转换到 hs 模式需要的时间, 单位 hsclk。	0x000

PHY_CTRL

PHY_CTRL 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	forcepll	PHY PLL 处于有效状态使能。 0: 不使能; 1: 使能。	0x0
[2]	RW	enableclk	PHY cklane 使能。	0x0



Bits	Access	Name	Description	Reset
			0: 不使能; 1: 使能。	
[1]	RW	rstz	PHY 数字电路复位使能。 0: 复位; 1: 撤销复位。	0x0
[0]	RW	shutdownz	PHY 整体 powerdown 使能。 0: power down; 1: power up。	0x0

LANE_NUM

LANE_NUM 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	stopstate_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	lane_num	datalane 个数配置。 00:1lane; 01:2lanes; 10:3lanes; 11:4lanes。	0x0

ULPS_CTRL

ULPS_CTRL 为 PHY 的 ULPS(Ultra Low Power State)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。	0x0000000
[3]	RW	txulpsexitdatalane	datalanes 退出 ULPS 模式使能。 0: 不使能; 1: 使能。	0x0
[2]	RW	txulpsdatalane	datalanes 进入 ULPS 模式请求使能。 0: 不使能; 1: 使能。	0x0
[1]	RW	txulpsexitclkane	clkane 退出 ULPS 模式使能。 0: 不使能; 1: 使能。	0x0
[0]	RW	txulpsclkane	clkane 进入 ULPS 模式请求使能。 0: 不使能; 1: 使能。	0x0

PPI_STATUS

PPI_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	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	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	ulpsactivenot2lane	PHY lane2 ulps 状态指示信号。	0x0



Bits	Access	Name	Description	Reset
			0: 处于 ulps 状态; 1: 不处于 ulps 状态。	
[9]	RO	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	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	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	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	ulpsactivenot0lane	PHY lane0 ulps 状态指示信号。 0: 处于 ulps 状态; 1: 不处于 ulps 状态。	0x0
[4]	RO	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	ulpsactivenotclk	PHY clk lane ulps 状态指示信号。 0: 处于 ulps 状态; 1: 不处于 ulps 状态。	0x0
[2]	RO	stopstatecklane	PHY clk lane stopstate 状态指示信号。 0: 不处于 stopstate 状态; 1: 处于 stopstate 状态。	0x0
[1]	RO	phy_tran_direction	PHY direction 信号状态。	0x0



Bits	Access	Name	Description	Reset
			0: 发送状态; 1: 接收状态。	
[0]	RO	phy_pll_lock	PHY PLL lock 信号状态。 0: PHY PLL lock 无效; 1: PHY PLL lock 有效。	0x0

PHY_REG_CFG0

PHY_REG_CFG0 为 PHY 配置控制 0 寄存器。

Offset Address: 0x00B4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	保留。	0x00000000
[1]	RW	testclk	PHY 配置接口时钟控制位。 0: 低电平; 1: 高电平。	0x0
[0]	RW	testclr	配置清除。 0: 不清除; 1: 清除。	0x0

PHY_REG_CFG1

PHY_REG_CFG1 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	testen	testen 信号控制。 0: 在 testclk 的上升沿采集数据;	0x0



			1: 在 testclk 的下降沿采集地址。	
[15:8]	RO	testdout	PHY 配置接口输出数据。	0x00
[7:0]	RW	testdin	PHY 配置接口输入数据。	0x00

INT0_STATUS

INT0_STATUS 为中断 0 状态寄存器。

Offset Address: 0x00BC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:21]	-	reserved	保留。	0x000
[20]	RC	errcontentionlp1	Lane0 LP1 contention 错误中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[19]	RC	errcontentionlp0	Lane0 LP0 contention 错误, 读清零。 0: 无中断; 1: 有中断。	0x0
[18]	RC	errcontrol	错误状态序列检测中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[17]	RC	errsyncesc	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	errescentry	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; 1: 有中断。	0x0
[15]	RC	error_report_15	设备错误报告(Error Reporting)中的	0x0



Bits	Access	Name	Description	Reset
			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。	
[14]	RC	error_report_14	设备错误报告(Error Reporting)中的 Reserved 位, 读清零。	0x0
[13]	RC	error_report_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	error_report_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	error_report_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	error_report_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	error_report_9	设备错误报告(Error Reporting)中的 ECC	0x0



Bits	Access	Name	Description	Reset
			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。	
[8]	RC	error_report_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有 1 位 ECC 错误时报错误, 读清零。 0: 无中断; 1: 有中断。	0x0
[7]	RC	error_report_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	error_report_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	error_report_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	error_report_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



Bits	Access	Name	Description	Reset
[3]	RC	error_report_3	设备错误报告(Error Reporting)中的 Escape Mode Entry Commnad 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	error_report_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	error_report_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; 1: 有中断。	0x0
[0]	RC	error_report_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

INT1_STATUS

INT1_STATUS 为中断 1 状态寄存器。

Offset Address: 0x00C0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[27]	RW	rxtrigger3	01100010 trigger 中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[26]	RW	rxtrigger2	01011101 trigger 中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[25]	RW	rxtrigger1	00100001 trigger 中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[24]	RW	rxtrigger0	10100000 trigger 中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[23:21]	-	reserved	保留。	0x0
[20]	RC	vss	帧起始中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[19]	-	reserved	保留。	0x0
[18]	RC	fifo_nempty_when_vsync	vsync 有效时数据 fifo 不空。 0: 无中断; 1: 有中断。	0x0
[17]	RC	hss_abnormal	hss 包紧接着 hs 模式发送, 读清零。 0: 无中断; 1: 有中断。	0x0
[16]	RC	line_length_change	行长度变化中断: 0: 没有中断; 1: 有中断。	0x0
[15]	RC	cmd_tran_end	命令发送完成标志:	0x0



Bits	Access	Name	Description	Reset
			0: 发送中; 1: 发送完成。	
[14]	RC	phy_pll_lock_err	phy pll 在设定的时间内无法 lock 中断, 时间由 phy_pll_start_time_qst 设定。 0: 无中断; 1: 有中断。	0x0
[13]	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
[12]	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
[11]	RC	rxpld_wr_err	接收数据 fifo 满写错误中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[10]	RC	rxpld_rd_err	接收数据 fifo 空读错误中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[9]	RC	cmd_pld_rd_err	命令的发送数据 fifo 空读错误中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[8]	RC	cmd_pld_wr_err	命令的发送数据 fifo 满写错误中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[7]	RC	cmd_wr_err	命令 fifo 满写错误中断, 读清零。 0: 无中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	
[6]	RC	pld_rd_err	视频数据 fifo 空读错误中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[5]	RC	pld_wr_err	视频数据 fifo 满写错误中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[4]	RC	rxecc_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
[3]	RC	rxecc_single_err	接收数据 ECC 单 bit 错误中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[2]	RC	rxcrc_err	接收数据 CRC 错误中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[1]	RC	rxpkt_size_err	包大小接收错误中断, 读清零。 0: 无中断; 1: 有中断。	0x0
[0]	RC	eotp_rx_err	EoTp 接收错误中断, 读清零。 0: 无中断; 1: 有中断。	0x0

INT0_MASK

INT0_MASK 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_errcontentionlp1	errcontentionlp1 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[19]	RW	mask_errcontentionlp0	errcontentionlp0 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[18]	RW	mask_errcontrol	errcontrol 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[17]	RW	mask_errsyncesc	errsyncesc 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[16]	RW	mask_errescentry	errescentry 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[15]	RW	mask_error_report_15	error_report_15 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[14]	RW	mask_error_report_14	error_report_14 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[13]	RW	mask_error_report_13	error_report_13 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[12]	RW	mask_error_report_12	error_report_12 中断屏蔽使能。	0x0



Bits	Access	Name	Description	Reset
			0: 屏蔽; 1: 使能。	
[11]	RW	mask_error_report_11	error_report_11 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[10]	RW	mask_error_report_10	error_report_10 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[9]	RW	mask_error_report_9	error_report_9 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[8]	RW	mask_error_report_8	error_report_8 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[7]	RW	mask_error_report_7	error_report_7 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[6]	RW	mask_error_report_6	error_report_6 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[5]	RW	mask_error_report_5	error_report_5 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[4]	RW	mask_error_report_4	error_report_4 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[3]	RW	mask_error_report_3	error_report_3 中断屏蔽使能。	0x0



Bits	Access	Name	Description	Reset
			0: 屏蔽; 1: 使能。	
[2]	RW	mask_error_report_2	error_report_2 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[1]	RW	mask_error_report_1	error_report_1 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[0]	RW	mask_error_report_0	error_report_0 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0

INT1_MASK

INT1_MASK 为中断 1 屏蔽寄存器。

Offset Address: 0x00C8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27]	RW	mask_rxtrigger3	01100010 trigger 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[26]	RW	mask_rxtrigger2	01011101 trigger 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[25]	RW	mask_rxtrigger1	00100001 trigger 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0



Bits	Access	Name	Description	Reset
[24]	RW	mask_rxtrigger0	10100000 trigger 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[23:21]	-	reserved	保留。	0x0
[20]	RW	mask_vss	帧起始中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[19]	-	reserved	保留。	0x0
[18]	RW	mask_fifo_nempty_when_vsync	fifo_nempty_when_vsync 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[17]	RW	mask_hss_abnormal	hss_abnormal 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[16]	RW	mask_line_length_change	line_length_change 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[15]	RW	mask_cmd_tran_end	cmd_tran_end 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[14]	RW	mask_phy_pll_lock_err	phy_pll_lock_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[13]	RW	mask_to_hs_tx	to_hs_tx 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[12]	RW	mask_to_lp_rx	to_lp_rx 中断屏蔽使能。	0x0



Bits	Access	Name	Description	Reset
			0: 屏蔽; 1: 使能。	
[11]	RW	mask_rxpld_wr_err	rxpld_wr_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[10]	RW	mask_rxpld_rd_err	rxpld_rd_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[9]	RW	mask_cmd_pld_rd_err	cmd_pld_rd_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[8]	RW	mask_cmd_pld_wr_err	cmd_pld_wr_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[7]	RW	mask_cmd_wr_err	cmd_wr_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[6]	RW	mask_pld_rd_err	pld_rd_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[5]	RW	mask_pld_wr_err	pld_wr_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[4]	RW	mask_rxecc_multi_err	rxecc_multi_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[3]	RW	mask_rxecc_single_err	rxecc_single_err 中断屏蔽使能。	0x0



Bits	Access	Name	Description	Reset
			0: 屏蔽; 1: 使能。	
[2]	RW	mask_rxcrc_err	rxcrc_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[1]	RW	mask_rxpkt_size_err	rxpkt_size_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0
[0]	RW	mask_eotp_rx_err	eotp_rx_err 中断屏蔽使能。 0: 屏蔽; 1: 使能。	0x0

PHY_CAL

PHY_CAL 为 PHY calibration 使能寄存器。

Offset Address: 0x00CC Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	txskewcalhs	HS skew calibration 使能。 0: 屏蔽; 1: 使能。	0x0

POLARITY_CTRL

POLARITY_CTRL 为前级模块输入控制信号极性控制寄存器。

Offset Address: 0x01B0 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000000
[3]	RW	data_en_plr	data valid 极性选择。 0: 高有效; 1: 低有效。	0x0
[2]	-	reserved	保留。	0x0
[1]	RW	hsync_plr	行同步信号极性选择。 0: 高有效; 1: 低有效。	0x0
[0]	RW	vsync_plr	帧同步信号极性选择。 0: 高有效; 1: 低有效。	0x0

GINT_MSK

GINT_MSK 为总中断屏蔽配置寄存器。

Offset Address: 0x024C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	int_msk	总中断屏蔽配置寄存器。固定配置成 0x1FFFF。	0x00000000

OPERATION_MODE

OPERATION_MODE 为工作模式控制寄存器。

Offset Address: 0x0308 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;	0x0



Bits	Access	Name	Description	Reset
			1: 使能。	
[30:28]	-	reserved	保留。	0x00
[27]	RW	video_te_en	Cmd mode 发视频通过 35h 命令获取 TE trigger 使能。 0: 不使能; 1: 使能。	0x0
[26]	RW	forcetxstopmode	强制 lane 处于发送模式使能。 0: 不使能; 1: 使能。	0x0
[25]	RW	forceroxmode	强制 lane 属于接收模式。 0: 不使能; 1: 使能。	0x0
[24]	RW	turndisable	Turnaround 关闭使能。 0: 不使能; 1: 使能。	0x0
[23]	RW	generic_hdr_wr_err_rrst_en	generic 命令 header 写失败自复位使能。 0: 不使能; 1: 使能。	0x0
[22]	RW	generic_pld_wr_err_rrst_en	generic 命令 payload 写失败自复位使能。 0: 不使能; 1: 使能。	0x0
[21]	RW	rxesc_buf_en	rxesc 数据打拍使能。 0: 不使能; 1: 使能。	0x0
[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;	0x0



Bits	Access	Name	Description	Reset
			1: 使能。	
[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
[15:7]	-	reserved	保留。	0x0
[6]	RW	period_skew_en	period skew 使能。 0: 不使能; 1: 使能。	0x0
[5]	RW	init_skew_en	initial skew 使能。 0: 不使能; 1: 使能。	0x0
[4:2]	-	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]	-	reserved	保留。	0x0

DATATYPE0

DATYPE0 为 datatype 配置寄存器 0。

Offset Address: 0x0330 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]	-	reserved	保留。	0x3D

CSI_CTRL

CSI_CTRL 为 CSI(Camera Serial Interface)时序配置寄存器。

Offset Address: 0x0338 Total Reset Value: 0x0001_0100

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	保留。	0x0001
[16]		csi_hss_en	LS/LE 包发送使能。 0: 不使能; 1: 使能。	0x1
[15:13]	-	reserved	保留。	0x0
[12]		dt_config	数据长包 datatype 来自寄存器值使能。 0: 不使能; 1: 使能。	0x0
[11:9]	-	reserved	保留	0x0
[8]	RW	frame_num_inc	FS/FE 帧号配置。 0: 固定为 0; 1: 每帧加 1。	0x1



[7:6]	-	reserved	保留	0x0
[5]	RW	csi_vss_position	FS 发送位置配置。 0: 在 VBP 的第一行; 1: 在 VBP 的最后一行。	0x0
[4]	RW	csi_vse_position	FE 发送位置配置。 0: 在 VFP 的第一行; 1: 在 VFP 的最后一行。	0x0
[3:2]	-	reserved	保留	0x0
[1]	RW	csi_mode	CSI 工作模式 0: 不支持 LS/LE 包 CSI 方式 1: 支持 LS/LE 及 FS 位置可配的 CSI 方式	0x0
[0]	RW	csi_en	CSI 时序使能。 0: 不使能; 1: 使能。	0x0

SKEW_BEGIN

SKEW_BEGIN 为 period skew 开始发送位置寄存器。

Offset Address: 0x033C Total Reset Value: 0x0000_0030

Bits	Access	Name	Description	Reset
[31:0]	RW	skew_begin	period skew 开始发送的位置, 从最后一行的 header 发送开始计算。	0x00000030

SKEW_END

SKEW_END 为 period skew 结束发送位置寄存器。

Offset Address: 0x0340 Total Reset Value: 0x0000_00B0



Bits	Access	Name	Description	Reset
[31:0]	RW	skew_end	period skew 结束发送的位置，从最后一行的 header 发送开始计算。	0x000000B0

LANE_ID

LANE_ID 为 lane id 配置寄存器。

Offset Address: 0x0350 Total Reset Value: 0x0000_3210

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	保留。	0x00000
[13:12]	RW	lane3_id	lane3 的 id 号，表示 phy 的 lane3 实际对接的设备的 lane 号。	0x3
[11:10]	-	reserved	保留。	0x0
[9:8]	RW	lane2_id	lane2 的 id 号，表示 phy 的 lane2 实际对接的设备的 lane 号。	0x2
[7:6]	-	reserved	保留。	0x0
[5:4]	RW	lane1_id	lane1 的 id 号，表示 phy 的 lane1 实际对接的设备的 lane 号。	0x1
[3:2]	-	reserved	保留。	0x0
[1:0]	RW	lane0_id	lane0 的 id 号，表示 phy 的 lane0 实际对接的设备的 lane 号。	0x0

10.4.7 MIPI Tx PHY 寄存器概览

MIPI Tx PHY 寄存器概览如表 10-14 所示。

表10-14 MIPI Tx PHY 寄存器概览 (基址是 0x0_1027_2000)

偏移地址	名称	描述	页码
0x0D20	COMMON_REG_48	PLL 配置 posdiv 参数配置寄存器	10-325



偏移地址	名称	描述	页码
0x0D24	COMMON_REG_49	PLL 配置 prediv 和 phy_div_int[10:8]参数配置寄存器	10-326
0x0D28	COMMON_REG_4A	PLL 配置 phy_div_int[7:0]参数配置寄存器	10-326
0x0D2C	COMMON_REG_4B	PLL 配置 update 参数配置寄存器	10-326
0x0D98+ 0x80*i	LANE_PN_SWAP	Lane 的 PN 互换使能寄存器	10-327
0x0DC0+ 0x80*i	PRE_REQ_DY	HS 时钟有效延时配置寄存器	10-327
0x0DC4+ 0x80*i	POS_REQ_DY	HS 时钟无效延时配置寄存器	10-327
0x0DC8+ 0x80*i	TLPX	lane Tlpx 时间配置寄存器	10-328
0x0DCC+ 0x80*i	TPREPARE	lane Tck-prepare 时间配置寄存器	10-328
0x0DD0+ 0x80*i	TZERO	lane Tclk-zero 时间配置寄存器	10-328
0x0DD4+ 0x80*i	THS_TRAIL	lane Ths-trail 时间配置寄存器	10-328
0XDF4+0 x80*i	TTAGET_TAGO	Lane taget、tago 时间	10-329

10.4.8 MIPI Tx PHY 寄存器描述

说明

寄存器描述中的 hsclock period=1/(Data Rate/8)。

寄存器描述中 i=0,1,2,3,4，其中时钟 lane 对应 i=2，数据 lane 分别对应 i=0,1,3,4。

COMMON_REG_48

COMMON_REG_48 是 PLL 配置 posdiv 参数配置寄存器。

Offset Address: 0XD20 Total Reset Value: 0x01



Bits	Access	Name	Description	Reset
[7:3]	-	Reserved	保留。	0x0
[2:0]	RW	reg_pll_phy_posdiv	PHY PLL POSDIV 参数配置。	0x01

COMMON_REG_49

COMMON_REG_49 是 PLL 配置 prediv 和 div_int[10:8]参数配置寄存器。

Offset Address: 0XD24 Total Reset Value: 0x01

Bits	Access	Name	Description	Reset
[7]	-	Reserved	保留。	0x0
[6:4]	RW	reg_pll_phy_prediv	PHY PLL PREDIV 参数配置。	0x0
[3]	-	Reserved	保留。	0x0
[2:0]	RW	reg_pll_phy_div_int_bit10to8	PHY PLL DIV_INT[10:8]参数配置。	0x01

COMMON_REG_4A

COMMON_REG_4A 是 PLL 配置 div_int[7:0]参数配置寄存器。

Offset Address: 0XD28 Total Reset Value: 0x04

Bits	Access	Name	Description	Reset
[7:0]	RW	reg_pll_phy_posdiv	PHY PLL POSDIV 参数配置。	0x04

COMMON_REG_4B

COMMON_REG_4A 是 PLL 配置 div_int 参数配置寄存器。

Offset Address: 0XD2C Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:1]	-	Reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[0]	RW	reg_pll_phy_div_update	PHY 参数配置更新寄存器。	0x00

LANE_PN_SWAP

LANE_PN_SWAP 是 Lane 的 PN 互换使能寄存器。

Offset Address: 0XD98+0x80*i Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:1]	-	Reserved	保留。	0x0
[0]	RW	reg_wire_swap	Lane 的 PN 互换使能配置。	0x00

PRE_REQ_DY

PRE_ERQ_DY 为 HS 时钟有效延时配置寄存器。

Offset Address: 0XDCC0+0x80*i Total Reset Value: 0x60

Bits	Access	Name	Description	Reset
[7:0]	RW	pre_req_dy	clock lane 从 txrequesthsclk 有效到进入 HS 模式的延时时间, 单位 hsclock 周期。	0x60

POS_REQ_DY

POS_REQ_DY 为 HS 时钟无效延时配置寄存器。

Offset Address: 0XDCC4+x080*i Total Reset Value: 0x01

Bits	Access	Name	Description	Reset
[7:0]	RW	post_req_dy	clock lane 从 txrequesthsclk 无效到退出 HS 模式的延时时间, 单位 hsclock 周期。	0x01



TLPX

TLPX 为 lane Tlpx 时间配置寄存器。

Offset Address: 0Xdc8+0x80*i Total Reset Value: 0x25

Bits	Access	Name	Description	Reset
[7:0]	RW	Tlpx	lane Tlpx 时间, 单位 hsclk 周期。	0x25

TPREPREARE

TCK_PREPARE 为 lane Tprepare 时间配置寄存器。

Offset Address: 0Xdcc+0x80*i Total Reset Value: 0x14

Bits	Access	Name	Description	Reset
[7:0]	RW	Tprepare	lane Tprepare 时间, 单位 hsclk 周期。	0x14

TZERO

TCLK_ZERO 为 lane Tzero 时间配置寄存器。

Offset Address: 0XDD0+0x80*i Total Reset Value: 0x1C

Bits	Access	Name	Description	Reset
[7:0]	RW	Tzero	lane Tzero 时间, 单位 hsclk 周期。	0x1c

THS_TRAIL

TTRAIL 为 lane Ths_trail 时间配置寄存器。

Offset Address: 0xDD4+0x80*i Total Reset Value: 0x16

Bits	Access	Name	Description	Reset
[7:0]	RW	Ths_trail	lane Ths_trail 时间, 单位 hsclk 周期。	0x16



TTAGET_TAGO

TTAGET_TAGO 是 PLL 配置 ttaget、tago 参数配置寄存器。

Offset Address: 0XDF4+0x80*i Total Reset Value: 0x33

Bits	Access	Name	Description	Reset
[7:4]	RW	reg_ttaget	lane Tta_get 时间, 单位 txclkesc 周期。	0x03
[3:0]	RW	reg_ttago	lane Tta_go 时间, 单位 txclkesc 周期。	0x03

10.5 热成像接口

10.5.1 概述

热成像接口可以将热成像探测器的 CMOS 电平数据输出时序转化为 DC (Digital Camera) 时序后传递给到下一级模块 VICAP (Video Capture), 热成像接口也支持将 VICAP 提供的 SmartData 和 OOC(On-the-chip Output Correction)校正数据按照特定时序发送给热成像探测器。

图10-58 热成像接口功能框图及在系统中的位置



10.5.2 特点

热成像接口有以下特点：

- 仅支持 CMOS 电平输入输出。
- 最多同时支持 1 路视频输入。



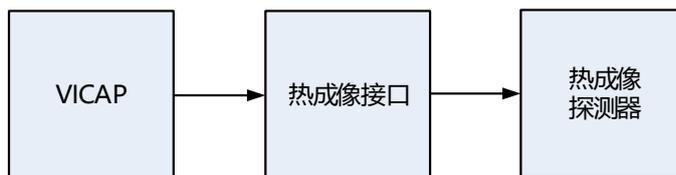
- 支持 SmartData 配置和 OOC 校正功能。
- 支持 14bit 数据类型的解析。
- 支持热成像接口输出数据 8+6、6+8 和 7+7 的模式。
- 热成像接口输出管脚支持乱序。
- 热成像接口数据输入管脚支持高低位反序。

10.5.3 功能描述

10.5.3.1 SmartData 配置

SmartData 的配置值按照一定格式写入 VICAP 的寄存器，配置 SmartData 时 VICAP 把配置值发送给热成像接口，热成像接口按照特定时序发送给热成像探测器。

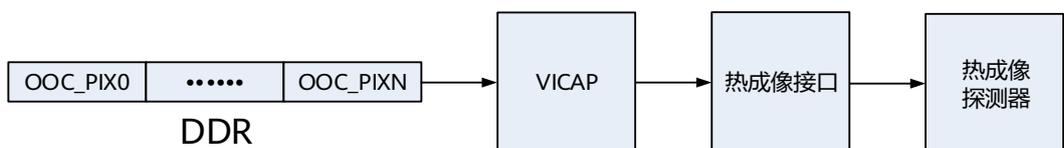
图10-59 SmartData 配置数据流



10.5.3.2 OOC 校正

每个像素对应的 OOC 值先按照一定格式存储在 DDR，VICAP 从 DDR 读取后发送给热成像接口，热成像接口按照特定时序发送给热成像探测器。

图10-60 OOC 校正数据流



10.5.3.3 热成像时序解析

热成像接口将特定时序转换成 DC 时序发送给 VICAP。



图10-61 热成像时序解析



10.5.4 工作方式

10.5.4.1 配置流程

- 步骤 1 上电启动。
- 步骤 2 根据使用场景配置相应的管脚复用。
- 步骤 3 配置 CRG 寄存器中的 PERI_CRG4498，打开热成像时钟门控。配置热成像接口软复位，撤销复位。
- 步骤 4 配置 CRG 寄存器中的 PERI_CRG4498，配置热成像主时钟频率。配置热成像 APB 总线，VI 侧热成像控制器软复位，撤销复位。
- 步骤 5 配置 VI 侧热成像控制器相关寄存器。
- 步骤 6 VI Port1 的视频源配置为热成像。
- 步骤 7 根据使用场景配置热成像接口模块相关寄存器。
- 步骤 8 配置热成像探测器序列。

---结束

10.5.5 热成像控制器寄存器概览

热成像寄存器控制器概览如表 10-15 所示。

表10-15 热成像控制器寄存器概览（基址是 0x0_173D_0000）

偏移地址	名称	描述	页码
0x0100	THERMO_WIDTH_CFG	宽度配置寄存器。	10-332
0x0170	THERMO_IMGSIZE	图像宽高统计寄存器。	10-332



偏移地址	名称	描述	页码
0x0300	THERMO_OUTPUT_MUX	热成像控制器 4 个输出管脚功能选择寄存器。	10-332
0x0214	DATA_RECV	T2 模式或 T3 模式探测器数据输入方式。	10-334
0x0220	OOC_MODE	输出 OOC 数据的串行方式。	10-335

10.5.6 热成像控制器寄存器描述

THERMO_WIDTH_CFG

THERMO_WIDTH_CFG 为宽度配置寄存器。

Offset Address: 0x0100 Total Reset Value: 0x0000_0100

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	T0_cfg_width	对接 T0 模式探测器时配置的宽度(配置值为实际宽度的一半)。	0x0100

THERMO_IMGSIZE

THERMO_IMGSIZE 为 T0 模式探测器图像宽高统计寄存器。

Offset Address: 0x0170 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	保留。	0x0000
[15:0]	RO	imgwidth_statis	图像宽度统计。	0x0000

THERMO_OUTPUT_MUX

THERMO_OUTPUT_MUX 为热成像控制器 4 个输出管脚功能选择寄存器。



Offset Address: 0x0300 Total Reset Value: 0x0000_8810

Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	保留。	0x0000
[15:12]	RW	sd3_mux	SD3 管脚功能选择。 0x0: T0_sd0 0x1: T0_sd1 0x2: T1_sda 0x3: T1_fs 0x4: T2_sda0/T3_sda0 0x5: T2_sda1/T3_sda1 0x6: T2_sda2/T3_sda2 0x7: T2_fs 0x8: 输出固定为 0 其他: 保留。	0x8
[11:8]	RW	sd2_mux	SD2 管脚功能选择。 0x0: T0_sd0 0x1: T0_sd1 0x2: T1_sda 0x3: T1_fs 0x4: T2_sda0/T3_sda0 0x5: T2_sda1/T3_sda1 0x6: T2_sda2/T3_sda2 0x7: T2_fs 0x8: 输出固定为 0 其他: 保留。	0x8
[7:4]	RW	sd1_mux	SD1 管脚功能选择。 0x0: T0_sd0 0x1: T0_sd1 0x2: T1_sda	0x1



Bits	Access	Name	Description	Reset
			0x3: T1_fs 0x4: T2_sda0/T3_sda0 0x5: T2_sda1/T3_sda1 0x6: T2_sda2/T3_sda2 0x7: T2_fs 0x8: 输出固定为 0 其他: 保留。	
[3:0]	RW	sd0_mux	SD0 管脚功能选择。 0x0: T0_sd0 0x1: T0_sd1 0x2: T1_sda 0x3: T1_fs 0x4: T2_sda0/T3_sda0 0x5: T2_sda1/T3_sda1 0x6: T2_sda2/T3_sda2 0x7: T2_fs 0x8: 输出固定为 0 其他: 保留。	0x0

DATA_RECV

DATA_RECV 为 T2 模式或 T3 模式探测器数据输入方式。

Offset Address: 0x0214 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:1]	RW	data_merge_mode	数据拼接方式。 00: 7+7 模式(bit7 接 0); 01: 6+8 模式;	0x0



Bits	Access	Name	Description	Reset
			10: 8+6 模式。	
[0]	-	reserved	保留。	0x0

OOC_MODE

OOC_MODE 为 OOC 数据的串行输出模式。

Offset Address: 0x0220 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	保留。	0x00000000
[0]	RW	OOC_mode	输出 OOC 数据的串行方式。 0: T2 模式探测器; 1: T3 模式探测器。	0x0



目 录

11 ISP	11-1
11.1 功能简介.....	11-1
11.2 总体概要.....	11-2
11.3 工作模式.....	11-5
11.4 ISP 中断系统	11-7
11.5 模块功能.....	11-9



插图目录

图 11-1 ISP 整体结构图.....	11-3
图 11-2 ISP_FE 结构图.....	11-3
图 11-3 ISP_BE 结构图.....	11-5
图 11-4 中断时序示意图.....	11-9
图 11-5 有效图像区域与水平垂直消隐关系图.....	11-10
图 11-6 AWB 有效像素颜色空间.....	11-14
图 11-7 Gamma 曲线.....	11-16
图 11-8 锐化、过冲说明.....	11-17



表格目录

表 11-1 外接 ISP 关键参数	11-6
表 11-2 位置可调模块关键参数	11-6
表 11-3 中断指示寄存器	11-7
表 11-4 中断时序.....	11-9
表 11-5 AE 区域统计信息	11-12
表 11-6 AE 直方图统计信息.....	11-13
表 11-7 AWB 区域统计方式 A 信息.....	11-14



11 ISP

11.1 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、坏点校正及镜头阴影校正等基本功能，也支持 WDR(Wide Dynamic Range)、DRC(Dynamic Range Compression)、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 BLC(Black Level Correction)
- 支持静态以及动态坏点校正，坏点簇矫正
- 支持固定噪声消除
- 支持 Bayer 降噪
- 支持 Demosaic 处理
- 支持紫边校正 CAC (Chromatic aberration correction)
- 支持 Gamma 校正
- 支持动态范围压缩 DRC (Dynamic Range Compression)
- 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 AWB(Automatic White Balance)
- 支持自动曝光 AE(Automatic Exposure)
- 支持 3A 相关统计信息输出
- 支持镜头阴影校正 LSC(Lens shading correction)
- 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 (Chroma Adjust)
- 支持 AIBNR、AIDRC、AIDM、AI3DNR
- 支持 3D 降噪

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 4096x8192 (不分块), 8192x8192 (分块)
- 支持最小图像分辨率 120x88
- 最小水平消隐区 64 像素
- 最小垂直消隐区 40 行(DRC 打开时最小 64 行)
- 最大性能：3840×2160@45fps

11.2 总体概要

功能框图

ISP 的功能结构图如图 11-1、图 11-2、图 11-3 所示。此图与本文中提到的 ISP_FE(ISP Front End)均代指 ISP pipeline 中 FPN (不包含) 之前的部分, ISP_BE(ISP Back End)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11-1 ISP 整体结构图

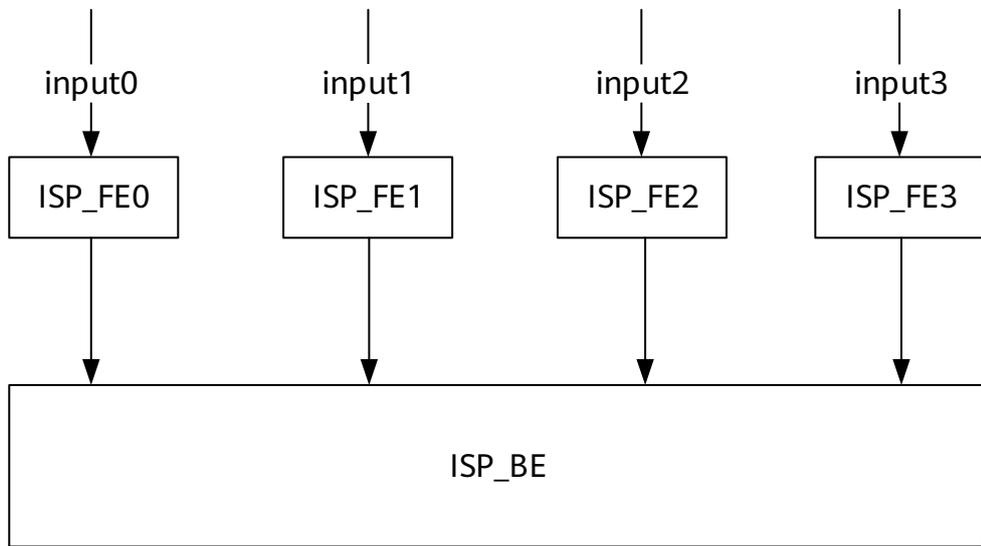
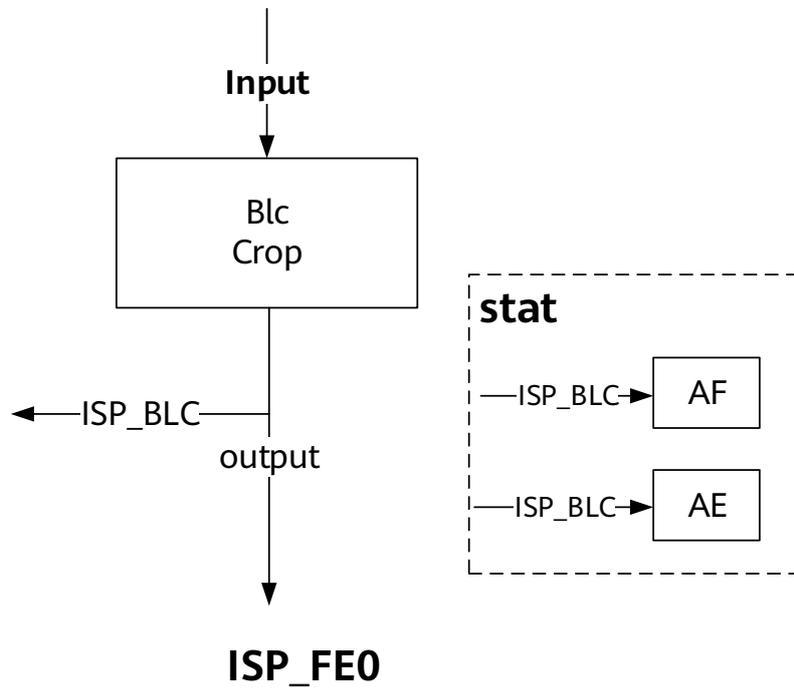


图11-2 ISP_FE 结构图



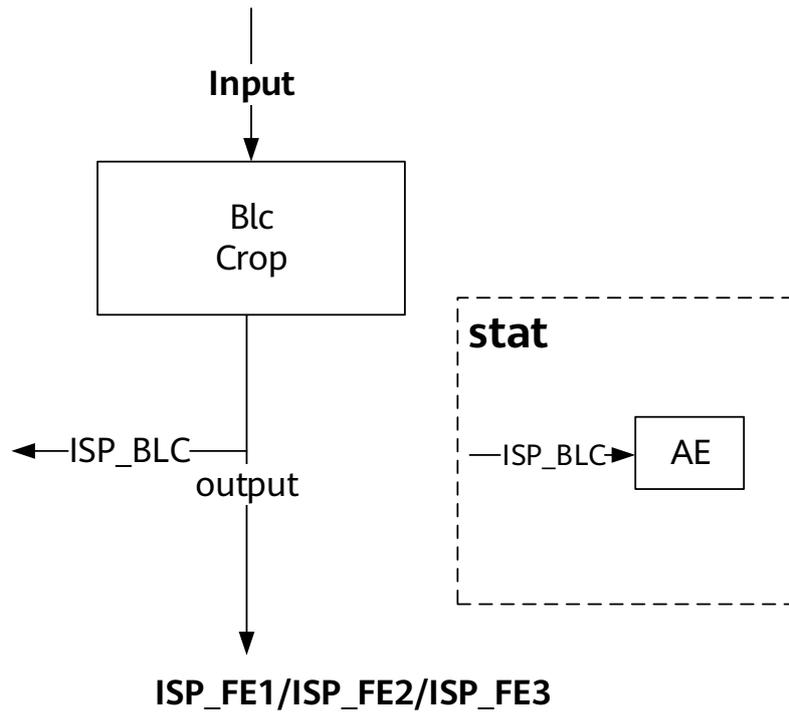
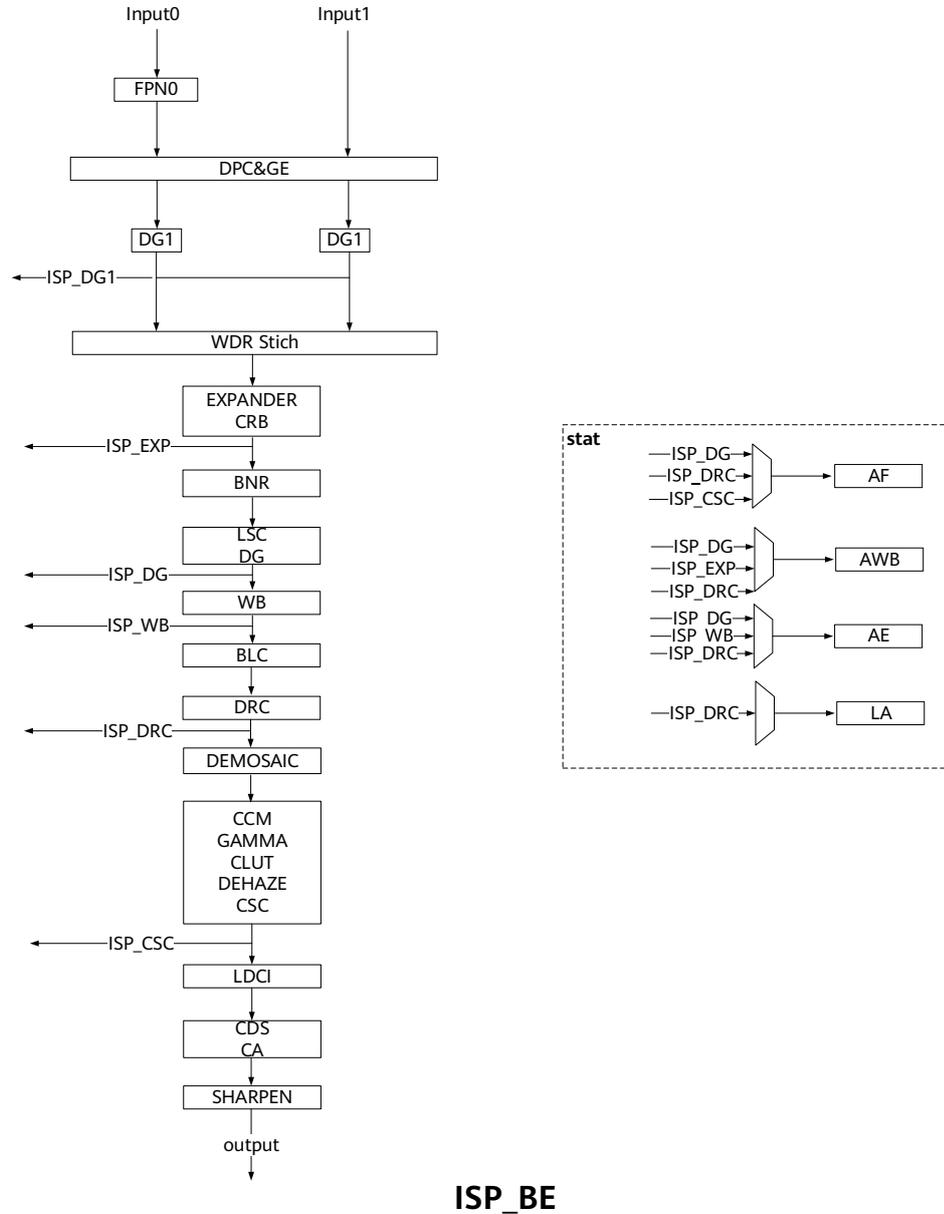




图11-3 ISP_BE 结构图



ISP_BE

11.3 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/ISP_FE1/ISP_FE2/ISP_FE3 时钟频率 300MHz。

ISP_BE 可以分时处理 ISP_FE0 - 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11-1 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: RAW 输入; 1: YUV 输入。

- **支持模块位置可调**

AE、AWB、AF、CLUT 支持位置可调。

表11-2 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 Dither 后面; 10: AF 在 CSC 之后; 其他: 保留。



参数名称	参数说明
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 Dither 后面; 其他: 保留。
ISP_BE_MODULE_POS[14:12]	AE 位置调节寄存器。 000: AE 在 DG 后面; 001: AE 在 WB 后面; 010: AE 在 DRC Dither 后面。 其他: 保留。
ISP_BE_MODULE_POS[7]	CLUT 位置调节寄存器。 0: CLUT 在 SQRT1 后面; 1: CLUT 在 Demosaic 后面。

11.4 ISP 中断系统

功能描述

现 ISP 有 38 个硬件中断事件, 详细见表 11-3 所示。

说明

* FE_N 对应 ISP 整体架构图中 FE0~FE3, 分别对应配置 FE0~FE3 的寄存器, 基址是 0x1740_0000。

表11-3 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)
0x200F0+FE_N*0x20000	bit[6]	AF 统计完成中断。(仅 FE0 支持)
	bit[5]	动态 BLC 统计完成中断。



偏移地址	32bit	描述 (写 1 清零, 0: 无中断 1: 有中断)
	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。
0x00364	bit[28]	LA 统计完成中断。
	bit[27]	FLICK 统计中断
	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。

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 11-4 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 11-4 为图 11-4 中断序号与中断事件的对应表格。

图11-4 中断时序示意图

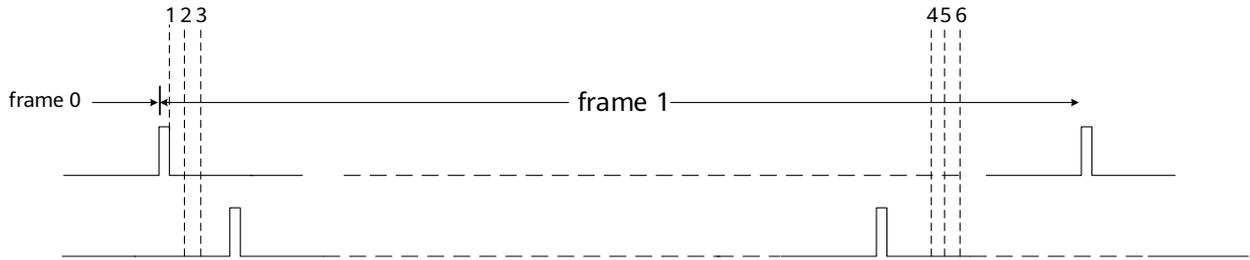


表11-4 中断时序

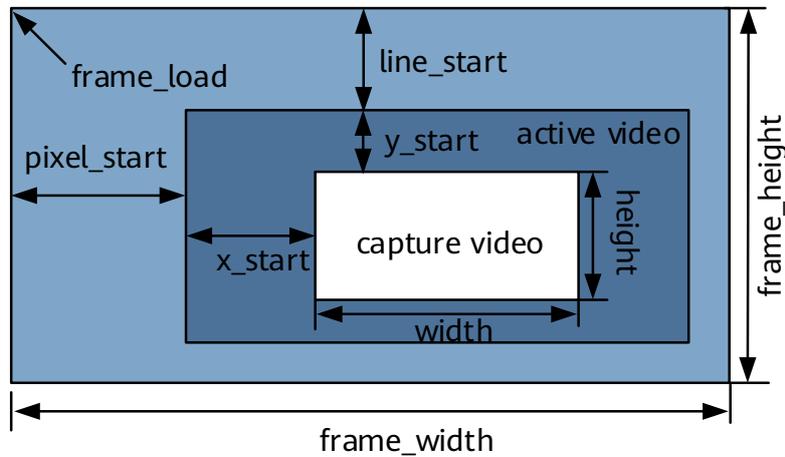
序号	中断事件
1	fstart(ISP_FE)/fstart(ISP_BE)
2	cfg_loss(ISP_FE)/cfg_loss(ISP_BE) update_cfg(ISP_FE) /update_cfg(ISP_BE)
3	fstart_delay(ISP_FE)
4	ae_int
5	af_int
6	LA 统计完成中断 DEHAZE 统计完成中断 LDCI 统计完成中断

11.5 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 11-5 所示。

图11-5 有效图像区域与水平垂直消隐关系图



FPN

FPN(Fixed Pattern Noise)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

BLC(Black Level Correction)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

DPC(Defect Pixel Correction)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

WDR(Wide Dynamic Range)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。

CRB

CRB(Color Rebalance)在 WDR 模式下改善暗区偏红的问题。

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

LSC(Lens shading correction)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

DG(Digital Gain)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(Luma Average)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

AE(Automatic Exposure)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 11-5 所示。m 表示水平分区个数，n 表示垂直分区个数。

表11-5 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
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 11-6 所示。



表11-6 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

AF(Auto Focus)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。

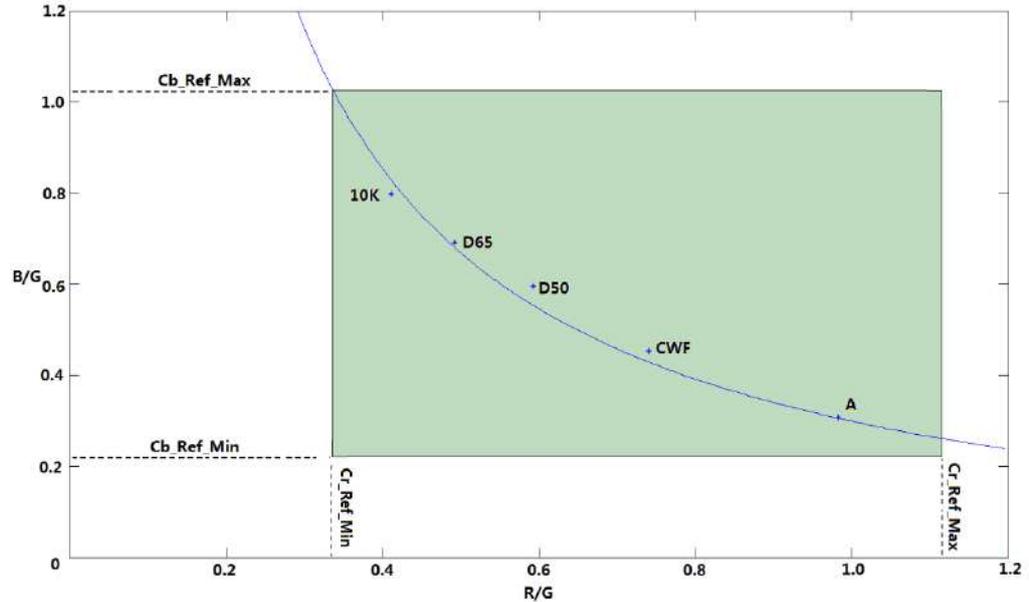
AWB

AWB(Auto White Balance Correction)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11-6 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 11-7 所示。

表11-7 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

WB(White Balance)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

DRC(Dynamic Range Compression)用于图像动态范围进行压缩。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(Chromatic aberration correction)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。

BAYERSHARP

BayerSharp 模块在 bayer 域实现图像细节增强，提高图像的整体清晰度。

CCM

CCM(Color Correction Matrix)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。

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} coef00 & coef01 & coef02 \\ coef10 & coef11 & coef12 \\ coef20 & coef21 & coef22 \end{pmatrix} \times \begin{pmatrix} R+in_dc0 \\ G+in_dc1 \\ B+in_dc2 \end{pmatrix} + \begin{pmatrix} out_dc0 \\ out_dc1 \\ out_dc2 \end{pmatrix}$$

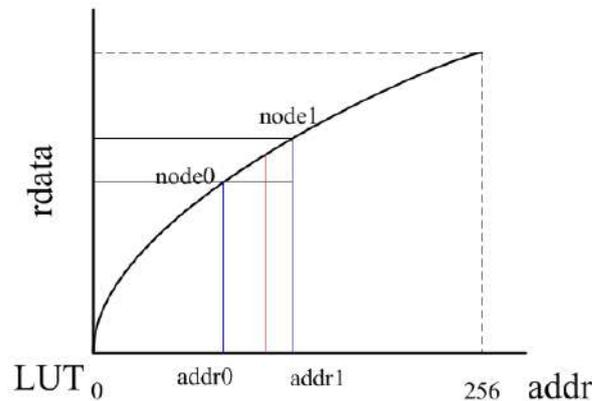


R、G、B 是输入数据，coef 是一个 3×3 的可配置矩阵系数。系数均为 15bit，格式 S5.10。

GAMMA

Gamma/preGamma 模块应用于 rgb/bayer 阈，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11-7 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]= 0xFFFF。

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(Color Space Conversion)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}。

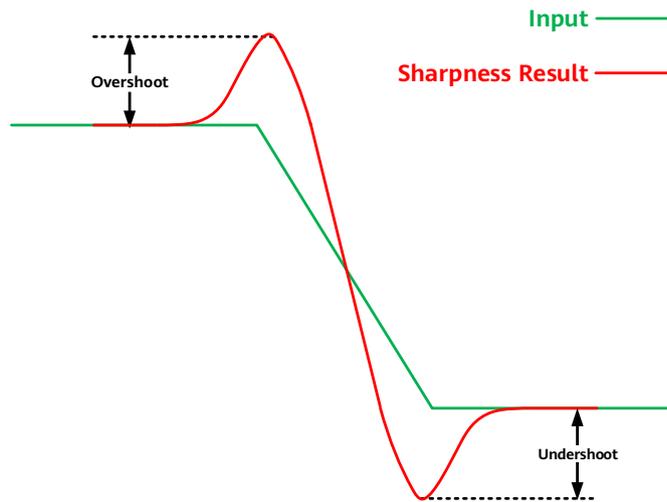
$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} coef00 & coef01 & coef02 \\ coef10 & coef11 & coef12 \\ coef20 & coef21 & coef22 \end{pmatrix} \times \begin{pmatrix} R + in_dc0 \\ G + in_dc1 \\ B + in_dc2 \end{pmatrix} + \begin{pmatrix} out_dc0 \\ out_dc1 \\ out_dc2 \end{pmatrix}$$

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11-8 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 11-8。通过调整 `overshoot` 和 `undershoot` 控制参数，能够抑制白边、黑边现象。

CDS

CDS(Chroma Down Sample)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

CA(Chroma Adjust)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(Local Dynamic Contrast Improvement)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。



目 录

12 音频接口	12-1
12.1 AIAO.....	12-1
12.1.1 概述.....	12-1
12.1.2 特点.....	12-2
12.1.3 功能描述.....	12-3
12.1.4 工作方式.....	12-7
12.1.5 AIAO 寄存器概览.....	12-9
12.1.6 AIAO 寄存器描述.....	12-13
12.2 Audio Codec.....	12-51
12.2.1 概述.....	12-51
12.2.2 特点.....	12-51
12.2.3 Audio Codec 寄存器概览.....	12-52
12.2.4 Audio Codec 寄存器描述.....	12-53



插图目录

图 12-1 芯片 AIAO 框图.....	12-1
图 12-2 AIP0 与 AOP0/AOP1 通过 I ² S/PCM 接口主/从模式对接外置 ADC/DAC 示意图.....	12-4
图 12-3 I ² S 接口时序.....	12-4
图 12-4 PCM 接口标准模式时序 (PCM_OFFSET =1)	12-5
图 12-5 PCM 接口自定义模式时序 (PCM_OFFSET =0)	12-5
图 12-6 I ² S 1/2/4/8/16 路接收.....	12-6
图 12-7 PCM 1/2/4/8/16 路接收, 在 256fs, offset=1 时的示例.....	12-6



表格目录

表 12-1 各模块的寄存器偏移地址变量表.....	12-10
表 12-2 AIAO 寄存器概览 (基址是 0x0_17C0_0000)	12-10
表 12-3 Audio Codec 寄存器概览 (基址是 0x0_17C4_0000)	12-52

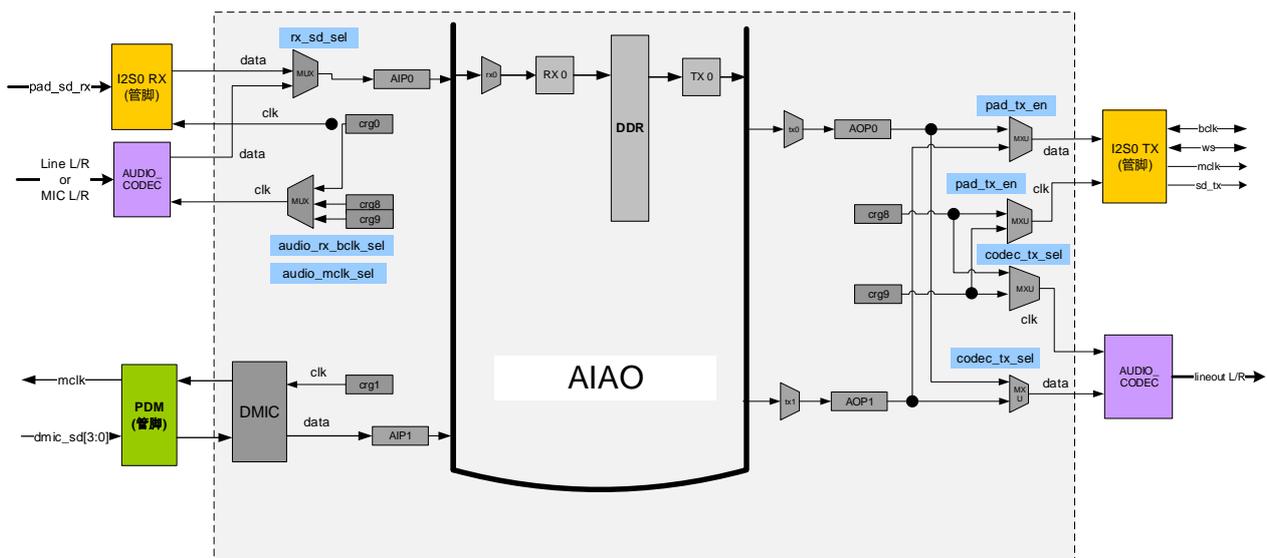
12 音频接口

12.1 AIAO

12.1.1 概述

音频输入输出接口 AIAO (Audio Input/Audio Output), 用于和片内或片外 Audio Codec 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。芯片内置 1 个 AIAO, 包含 2 个 AIP (Audio Input Port) 和 2 个 AOP (Audio Output Port), 支持立体声输入输出。AOP0 和 AOP1 通过 MUX 选择输出实现与 I2S 管脚、内置 Codec 在芯片内部对接, 基本模块框图如图 12-1 所示。

图12-1 芯片 AIAO 框图



注：两路 AIP 只能选其中一路数据送至 DDR，如果需要两路 AOP 只能同源输出



说明

- 图中黄色表示外置 CODEC 相关通路部分
- 图中紫色表示内置 CODEC 相关通路
- 图中绿色表示数字 MIC 相关通路时钟选择
- 图中灰色表示内部逻辑
- 图中蓝色表示 AIAO_MUX_SEL 寄存器

12.1.2 特点

AIAO 接口支持 I²S 和 PCM (Pulse Code Modulation) 两种模式和外置 Codec 对接, 并支持 PDM 接口与数字 MIC 对接。

- AIAO 接口支持 I²S 模式和内置 Codec 对接; 内置 Codec 的寄存器配置参考 12.2 Audio Codec 章节。
- AIAO 支持 PDM 接口, 对接数字 MIC。

接收 (AIP0) 和发送 (AOP0/AOP1) 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。

PCM 模式

PCM 模式有如下特点:

- 支持主模式和从模式。
- 支持单声道 8bit/16bit PCM 数据的发送和接收。
- 支持 2/4/8/16 路 8/16 bit 数据的多路 PCM 接收。
- 支持 8kHz ~ 96kHz 采样率。
- PCM 模式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), 支持标准和自定义两种模式。
- 接收 (AIP0) 和发送 (AOP0/AOP1) 相互独立, 可以单独使能或关闭。
- 接收 (AIP0/AIP1) 和发送 (AOP0/AOP1) 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 模式

I²S 模式有如下特点:

- 支持主模式和从模式。



- AIP0 既支持非时分复用模式下 1 路 16bit/24bit 数据的接收，也支持时分复用模式下 2/4/8/16 路 8/16bit 数据的接收。
- AIP1 只支持非时分复用模式下 2/4/8 路 16/24bit 数据的接收。
- AOP0 和 AOP1 支持非时分复用模式下的 1/2 路 16/24bit 数据的发送，也支持时分复用下的 8 路 16/24bit 数据的发送。
- 支持 8kHz ~ 96kHz 采样率。
- 接收 (AIP0/AIP1) 和发送 (AOP0/AOP1) 相互独立，可以单独使能或关闭。
- 接收 (AIP0/AIP1) 和发送 (AOP0/AOP1) 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。

DMIC

DMIC 有如下特点：

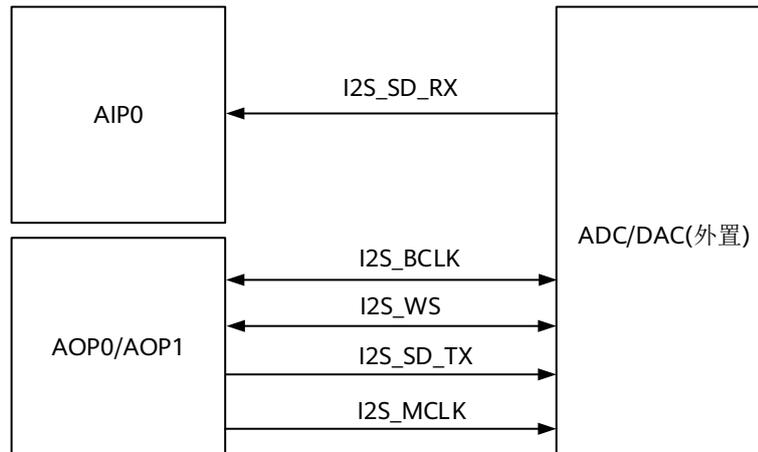
- 固定主模式。
- 支持 2/4/8 路 MIC 输入。
- 只支持非时分复用 I²S 模式，数据精度 16/24bit。
- 支持 8/12/16/24/32/48kHz 采样率。

12.1.3 功能描述

典型应用

芯片内置 2 个 AIP 和 2 个 AOP，其中：

- AIP0/AOP 0/AOP 1 对接内置 Audio Codec 时，仅支持 I²S 主模式；
- AIP0 支持 I²S/PCM 主/从模式对接外置 Audio Codec；
- AOP0/AOP1 支持 I²S/PCM 主/从模式对接外置 Audio Codec；
- AIP1 支持 I²S 主模式与内置 DMIC 模块对接。

图12-2 AIP0 与 AOP0/AOP1 通过 I²S/PCM 接口主/从模式对接外置 ADC/DAC 示意图

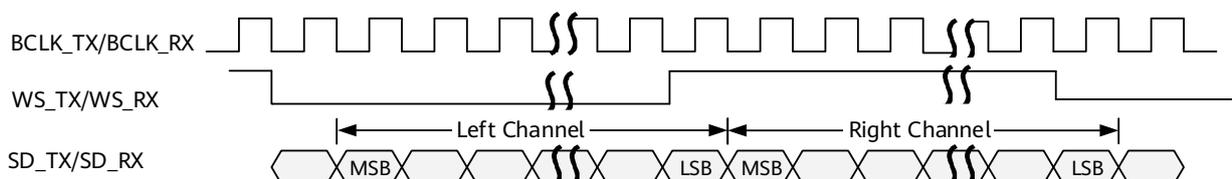
功能原理

AIP0 通过 I²S 接收对接外置 ADC 或内置的 Audio Codec 进行 AD (Analog-to-Digital) 转换后的音频数据, 存入为 AIP0 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。

AIP1 通过 I²S 接收对接内置 DMIC 进行降采样转换后的音频 PCM 数据, 存入为 AIP1 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 接口发送给外置 DAC, 或内置的 Audio Codec 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 12-3。按照协议, 用 BCLK_TX/BCLK_RX 时钟的下降沿发送数据, 上升沿进行数据采样, MSB 在 WS_TX/WS_RX 变化的下一个时钟周期有效。总是先传送 MSB, 后传送 LSB。

图12-3 I²S 接口时序

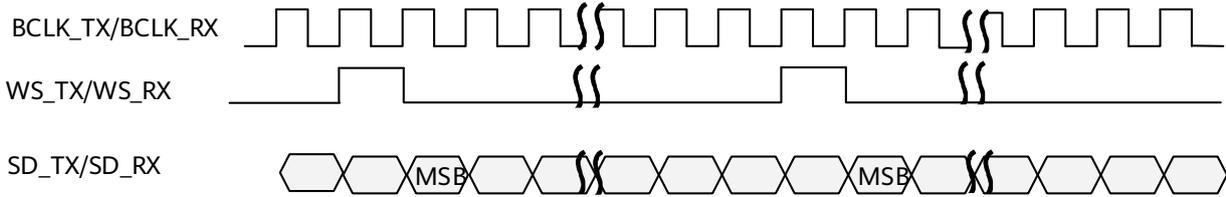
PCM 接口传输的数据是单声道数据, WS_TX/WS_RX 标识数据的起始位置, 先发送 (接收) 最高有效位 MSB, 使用时钟的上升沿发送数据, 下降沿采样数据 (可反向



BCLK 调整)。标准模式时序中，MSB 数据在 WS_TX/WS_RX 高电平脉冲之后一个周期有效；自定义模式时序中，MSB 的位置根据 PCM_OFFSET 不同可调，当 PCM_OFFSET 为 0 时与 WS_TX/WS_RX 的高电平脉冲对齐的。

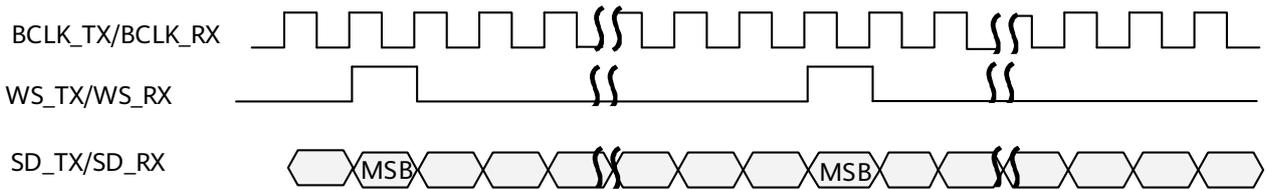
PCM 接口标准模式下的时序如图 12-4。

图12-4 PCM 接口标准模式时序 (PCM_OFFSET =1)



PCM 自定义模式时序如图 12-5 所示。

图12-5 PCM 接口自定义模式时序 (PCM_OFFSET =0)

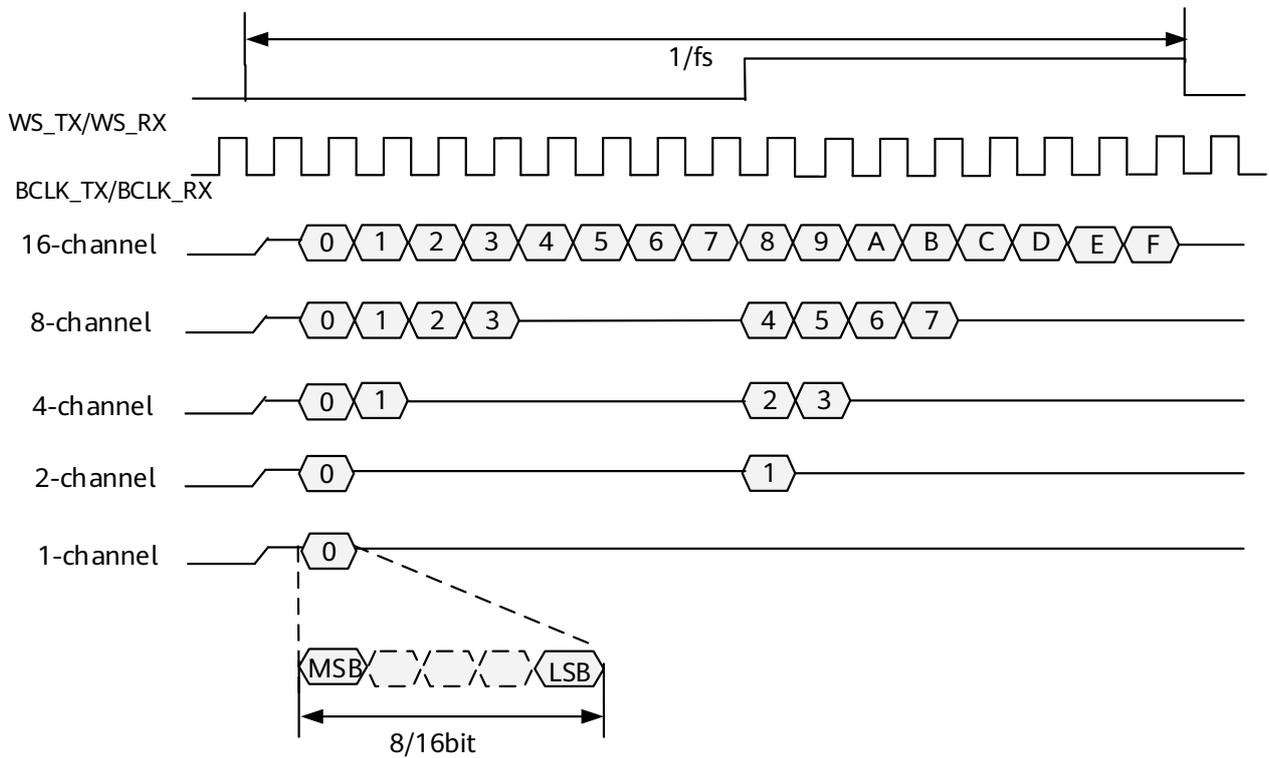


自定义模式时，数据与 WS 脉冲同拍开始发出。

I²S 进行多路 (2/4/8/16 路 8/16bit) 接收时，数据分别放于 I²S 时序的左右声道，如图 12-6 所示。



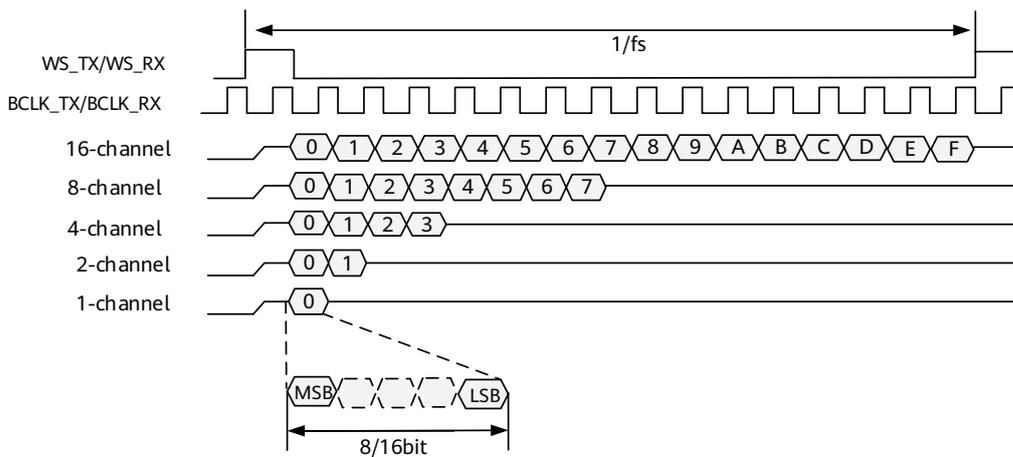
图12-6 I²S 1/2/4/8/16 路接收



注：1-channel 模式仅在非时分复用支持。

PCM 模式下的多路接收，如图 12-7 所示。支持 PCM 标准和自定义两种模式，AIAO 可以选择数据采样时刻（上升沿或下降沿）接收。图 12-7 中以上升沿为例。

图12-7 PCM 1/2/4/8/16 路接收，在 256fs, offset=1 时的示例



注：1-channel 模式仅在非时分复用支持。



12.1.4 工作方式

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/AOP) 的时钟门控。具体步骤如下：

- 步骤 1 在系统 PERI_CRG10784 寄存器中 aiao_pll_cksel 配置为 1，选择 786.432MHz 时钟源。
- 步骤 2 在系统 PERI_CRG10784 寄存器中，配置 aiao_srst_req 寄存器为 0，解除 AIAO 复位，配置 aiao_pll_cken 寄存器为 1，打开时钟门控。
- 步骤 3 在 AIAO 中 AIAO_MUX_SEL 寄存器中，配置时钟和数据的 MUX 关系。
- 步骤 4 配置分频系数。

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。CRG 和 AIP、AOP 的对应关系见 AIAO 框图。

----结束

软复位

AIAO 内部的 4 个通道 (AIP0、AIP1、AOP0、AOP1) 支持独立的软复位，当复位 AIAO 模块时，4 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 48kHz，2 声道 16bit 精度，假设 AIAO pll 源头时钟为 786.432M，系统控制器 AIAO 时钟已经使能)。

- 步骤 1 配置寄存器 I2S_CRG_CFG0_08(m=0)为 0x00200000，此时第 8 路时钟输出 mclk 频率为 12.288MHz。
- 步骤 2 配置寄存器 I2S_CRG_CFG1_08(m=0)为 0x00000133，此时使能第 8 路时钟，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 AIAO_SWITCH_RX_BCLK 寄存器为 0x00000008，此时设置接收通路选择工作在第 8 路时钟上面，也即上面配置好的时钟。



- 步骤 4 配置 RX_IF_ATTRI 为 0xE480014，此时设置接收通道工作在 I²S 模式下，2 声道，采样精度为 16bit。
- 步骤 5 配置 RX_BUFF_SADDR_LOW 寄存器为分配 DDR 的起始地址低 32bit 地址，比如 0x80000000，配置 RX_BUFF_SADDR_HI 寄存器为分配 DDR 的起始地址高 32bit 地址，比如 0x00000000，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 6 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，只使能 trans_int 中断。
- 步骤 7 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 8 通过读取 RX_BUFF_WPTR 和 RX_BUFF_RPTR 的值判断缓冲区的空/满状态以及有效数据量；要保证在缓冲区满之前将数据取走，并将更新后的缓冲区读地址写入 RX_BUFF_RPTR，否则可能会造成缓冲区溢出，声音不连续。
- 步骤 9 录音完成后，写寄存器 RX_DSP_CTRL 为 0x00000000，查询 RX_DSP_CTRL 寄存器，直到其值变为 0x20000000（说明接收通道已停止工作），即可退出。

----结束

须知

启动 AIP0 之前，必须完成 AIP0 时钟的配置，以保证 AIAO_BCLK_RX 和 AIAO_WS_RX 正常。

播放工作流程

AOP0、AOP1 的工作流程相同，下面以 AOP0 为例进行说明。

播放步骤如下：（假设场景为 I²S 模式 48kHz，2 声道 16bit 精度，假设 AIAO pll 源头时钟为 786.432M，系统控制器 AIAO 时钟已经使能）：

- 步骤 1 配置寄存器 I2S_CRG_CFG0_08 为 0x00200000，此时第 8 路时钟输出 mclk 频率为 12.288MHz。
- 步骤 2 配置寄存器 I2S_CRG_CFG1_08 为 0x00000133，此时使能第 8 路时钟，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 AIAO_SWITCH_TX_BCLK 寄存器为 0x00000008，此时设置接收通路选择工作在第 8 路时钟上面，也即上面配置好的时钟。
- 步骤 4 配置 TX_IF_ATTRI 寄存器为 0xE4000014，即配置发送接口工作于 I²S 模式下，2 声道，16bit 采样精度。
- 步骤 5 配置 TX_BUFF_SADDR_LOW 为分配 BUF 的起始低 32bit 地址，比如 0x80000000，配置 TX_BUFF_SADDR_HI。为分配 BUF 的高 32bit 起始地址，比如 0x00000000，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 6 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 7 配置寄存器 TX_DSP_CTRL 为 0x10000000，使能播放通道。
- 步骤 8 通过读取 TX_BUFF_WPTR 和 TX_BUFF_RPTR 的值判断缓冲区的空/满状态以及有效数据量；要保证在缓冲区空之前将新的音频数据填入，并将更新后的缓冲区写地址写入 TX_BUFF_WPTR，否则可能会造成缓冲区下溢出，声音不连续。
- 步骤 9 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，当其值变为 0x20000000 时（说明播放通道已停止工作），即可退出。

---结束

须知

启动 AOP0 前，必须完成 AOP0 时钟的配置，以保证 AIAO_BCLK_TX 和 AIAO_WS_TX 正常，对 AOP1、AOP2 有相同要求。

向 AOP0 的缓冲区写入数据以及更新 TX_BUFF_WPTR 时，必须保证 AOP0 的缓冲区空闲空间不小于 32 字节，对 AOP1 有相同要求。

12.1.5 AIAO 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 12-1 所示。



表12-1 各模块的寄存器偏移地址变量表

变量名称	取值范围	描述
m	0	发送通道号。
n	0	接收通道号

AIAO 寄存器概览如表 12-2 所示。

表12-2 AIAO 寄存器概览 (基址是 0x0_17C0_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	12-13
0x0004	AIAO_INT_STATUS	AIAO 模块中断状态寄存器	12-14
0x0008	AIAO_INT_RAW	AIAO 模块原始中断寄存器	12-14
0x0028	AIAO_SWITCH_RX_BCLK	AIAO I2S RX BCLK SWITCH 配置寄存器	12-15
0x002C	AIAO_SWITCH_TX_BCLK	AIAO I2S TX BCLK SWITCH 配置寄存器	12-15
0x0030	AIAO_STATUS	AIAO 状态寄存器	12-16
0x0034	VHB_OUTSTANDING	VHB_OUTSTANDING 配置寄存器	12-16
0x006C	AIAO_MUX_SEL	I2S 通路选择控制寄存器	12-16
0x0100	I2S_CRG_CFG0_00	I2S00 CRG 配置 0 号寄存器	12-18
0x0104	I2S_CRG_CFG1_00	I2S00 CRG 配置 1 号寄存器	12-19
0x0108	I2S_CRG_CFG0_01	I2S01 CRG 配置 0 号寄存器	12-21
0x010C	I2S_CRG_CFG1_01	I2S01 CRG 配置 1 号寄存器	12-21
0x0140	I2S_CRG_CFG0_08	I2S08 CRG 配置 0 号寄存器	12-23
0x0144	I2S_CRG_CFG1_08	I2S08 CRG 配置 1 号寄存器	12-23



偏移地址	名称	描述	页码
0x0148	I2S_CRG_CFG0_09	I2S09 CRG 配置 0 号寄存器	12-25
0x014C	I2S_CRG_CFG1_09	I2S09 CRG 配置 1 号寄存器	12-26
0x1000 + 0x100×n	RX_IF_ATTRI	接收通道的接口属性设置寄存器	12-28
0x1004 + 0x100×n	RX_DSP_CTRL	接收处理通道的控制寄存器	12-30
0x107C + 0x100×n	RX_BUFF_SADDR_HI	为接收通道的 DDR 缓存起始地址高 32bit 寄存器。	12-31
0x1080 + 0x100×n	RX_BUFF_SADDR_LOW	接收通道的 DDR 缓存起始地址低 32bit 寄存器	12-31
0x1084 + 0x100×n	RX_BUFF_SIZE	接收通道的 DDR 缓存大小寄存器	12-32
0x1088 + 0x100×n	RX_BUFF_WPTR	接收通道的 DDR 缓存写地址寄存器	12-32
0x108C + 0x100×n	RX_BUFF_RPTR	接收通道的 DDR 缓存读地址寄存器	12-33
0x1090 + 0x100×n	RX_BUFF_ALFULL_TH	接收通道的 DDR 缓存几乎满水线寄存器	12-33
0x1094 + 0x100×n	RX_TRANS_SIZE	接收通道的数据传输长度寄存器	12-33
0x1098 + 0x100×n	RX_WPTR_TMP	上报传输完成中断时，保存接收通道的写地址寄存器	12-34
0x10A0 + 0x100×n	RX_INT_ENA	接收通道的中断使能寄存器	12-34
0x10A4 + 0x100×n	RX_INT_RAW	接收通道的原始中断寄存器	12-35
0x10A8 + 0x100×n	RX_INT_STATUS	接收通道的中断状态寄存器	12-36



偏移地址	名称	描述	页码
0x10AC + 0x100×n	RX_INT_CLR	接收通道的中断清除寄存器	12-37
0x2000 + 0x100×m	TX_IF_ATTRI	发送通道的接口属性设置寄存器	12-38
0x2004 + 0x100×m	TX_DSP_CTRL	发送处理通道的控制寄存器	12-41
0x2020 + 0x100×m	TX_WS_CNT	接收处理通道的 WS 循环计数状态寄存器	12-43
0x2024 + 0x100×n	TX_BCLK_CNT	接收处理通道的 BCLK 循环计数状态寄存器	12-43
0x207C + 0x100×n	TX_BUFF_SADDR_HI	为发送通道的 DDR 缓存起始地址高 32bit 寄存器	12-44
0x2080 + 0x100×m	TX_BUFF_SADDR_LOW	发送通道的 DDR 缓存起始地址低 32bit 寄存器	12-44
0x2084 + 0x100×m	TX_BUFF_SIZE	发送通道的 DDR 缓存大小寄存器	12-44
0x2088 + 0x100×m	TX_BUFF_WPTR	发送通道的 DDR 缓存写地址寄存器	12-45
0x208C + 0x100×m	TX_BUFF_RPTR	发送通道的 DDR 缓存读地址寄存器	12-45
0x2090 + 0x100×m	TX_BUFF_ALEMPY_TH	发送通道的 DDR 缓存几乎空流水线寄存器	12-45
0x2094 + 0x100×m	TX_TRANS_SIZE	发送通道的数据传输长度寄存器	12-46
0x2098 + 0x100×m	TX_RPTR_TMP	上报传输完成中断时，保存发送通道的读地址寄存器	12-46
0x20A0 + 0x100×m	TX_INT_ENA	发送通道的中断使能寄存器	12-47



偏移地址	名称	描述	页码
0x20A4 + 0x100×m	TX_INT_RAW	发送通道的原始中断寄存器	12-48
0x20A8 + 0x100×m	TX_INT_STATUS	发送通道的中断状态寄存器	12-49
0x20AC + 0x100×m	TX_INT_CLR	发送通道的中断清除寄存器	12-50

12.1.6 AIAO 寄存器描述

说明

寄存器描述中的 BCLK 是位时钟；MCLK 是主时钟；FSCLK 是采样时钟。

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
[31:27]	-	reserved	保留。	0x0
[26:17]	-	reserved	保留。	0x000
[16]	RW	tx_ch0_int_ena	发送通道 0 的中断使能。 0: 不使能; 1: 使能。	0x0
[15:1]	-	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
[31:27]	-	reserved	保留。	0x0
[26:17]	-	reserved	保留。	0x000
[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]	-	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
[31:27]	-	reserved	保留。	0x0
[26:17]	-	reserved	保留。	0x000
[16]	RO	tx_ch0_int_raw	发送通道 0 的原始中断。 0: 无原始中断; 1: 有原始中断。	0x0
[15:1]	-	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;	0x0



Bits	Access	Name	Description	Reset
			1: 有原始中断。	

AIAO_SWITCH_RX_BCLK

AIAO_SWITCH_RX_BCLK 为 AIAO I2S RX BCLK SWITCH 配置寄存器。

Offset Address: 0x0028 Total Reset Value: 0x0000_0070

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x0000007
[3:0]	RW	inner_bclk_ws_sel_rx_00	内部 BCLK 选择。 0x0: 选择 BCLK0; 0x1: 选择 BCLK1; 0x8: 选择 BCLK8; 0x9: 选择 BCLK9; 其他: 保留。	0x0

AIAO_SWITCH_TX_BCLK

AIAO_SWITCH_TX_BCLK 为 AIAO I2S TX BCLK SWITCH 配置寄存器。

Offset Address: 0x002C Total Reset Value: 0x0000_00F8

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x000000F
[3:0]	RW	inner_bclk_ws_sel_tx_00	发送通道内部 BCLK 选择。 0x0: 选择 BCLK0; 0x1: 选择 BCLK1; 0x8: 选择 BCLK8; 0x9: 选择 BCLK9; 其他: 保留。	0x8



AIAO_STATUS

AIAO_STATUS 为 AIAO 状态寄存器。

Offset Address: 0x0030 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RO	srst_rdy	APB 复位撤销指示。 0: 正在复位; 1: 复位完成。	0x1

VHB_OUTSTANDING

VHB_OUTSTANDING 为 VHB_OUTSTANDING 配置寄存器。

Offset Address: 0x0034 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x00000000
[2:0]	RW	vhb_outst_num	Vhb Outstanding 的个数。	0x0

AIAO_MUX_SEL

AIAO_MUX_SEL 为 I2S 通路选择控制寄存器。

Offset Address: 0x006C Total Reset Value: 0x0003_9000

Bits	Access	Name	Description	Reset
[31:23]	-	reserved	保留。	0x000
[22]	RW	crg9_mode_sel	CRG9 主模式下 fsclk 产生模式控制位。 0: I2S 模式; 1: PCM 模式。	0x0
[21:18]	RW	i2s1_tx_sd_src_sel	tx1_sd 的 source 选择。 0x0: I2S TX0;	0x0



Bits	Access	Name	Description	Reset
			0x8: I2S RX0; 0x9: I2S RX1; 其他: 保留。	
[17:16]	RW	i2s1_tx_sd3_sel	tx1_sd 的 SD3 选择。 00: SD0; 01: SD1; 10: SD2; 11: SD3。 注意: 先选择通道, 然后选择数据线。	0x3
[15:14]	RW	i2s1_tx_sd2_sel	tx1_sd 的 SD2 选择。 00: SD0; 01: SD1; 10: SD2; 11: SD3。 注意: 先选择通道, 然后选择数据线。	0x2
[13:12]	RW	i2s1_tx_sd1_sel	tx1_sd 的 SD1 选择。 00: SD0; 01: SD1; 10: SD2; 11: SD3。 注意: 先选择通道, 然后选择数据线。	0x1
[11:10]	RW	i2s1_tx_sd0_sel	tx1_sd 的 SD0 选择。 00: SD0; 01: SD1; 10: SD2; 11: SD3。 注意: 先选择通道, 然后选择数据线。	0x0
[9:5]	-	reserved	保留。	0x00



Bits	Access	Name	Description	Reset
[4]	RW	codec_tx_sel	DAC 输入数据和时钟选择控制。 0: tx0_sd、CRG8; 1: tx1_sd、CRG9。	0x0
[3]	RW	audio_mclk_sel	芯片 I2S 接口的 mclk 选择控制。 0: aiao_i2s_rx_mclk, 即 CRG0; 1: aiao_i2s_tx_mclk, 即 CRG8 或 CRG9, 具体选 CRG8 或者 CRG9 要通过 codec_tx_sel 来配合选择。	0x0
[2]	RW	rx_sd_sel	rx0_sd 数据源选择。 0: I2S 管脚; 1: 内置 codec_ADC 的 iis_sd 输出。	0x0
[1]	RW	audio_rx_bclk_sel	内置 codec_ADC 的 bclk/ws 选择。 0: 选择 CRG0 的时钟; 1: 选择 CRG8 或者 CRG9 的时钟, 这种情况下, ADC 和 DAC 共时钟, 具体选 CRG8 或 CRG9 要通过 codec_tx_sel 来配合选择。	0x0
[0]	RW	pad_tx_en	I2S TX0 输出管脚数据和时钟选择控制。 0: tx1_sd、CRG9; 1: tx0_sd、CRG8。	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。	0x00
[26:0]	RW	aiao_mclk_div	MCLK 的分频时钟配置值, 配置值为	0x0AAAAAA



Bits	Access	Name	Description	Reset
			(MCLK/SIO 时钟源头频率) × 2 ²⁷ 。	

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
[31:16]	-	reserved	保留。	0x0000
[15]	RW	aiao_ws_en	WS 使能。 0: 不使能; 1: 使能。	0x0
[14]	RW	aiao_bclk_en	BCLK 使能。 0: 不使能; 1: 使能。	0x0
[13]	RW	aiao_bclkout_pctrl	BCLKOUT 的极性控制。 0: 正向; 1: 反向。	0x0
[12]	RW	aiao_bclk_in_pctrl	BCLKIN 的极性控制。 0: 正向; 1: 反向。	0x0
[11]	RW	aiao_bclk_sel	BCLK/FSCLK 选择。 0: 时钟从内部产生; 1: 时钟从外部输入。	0x0
[10]	RW	aiao_bclk_oen	BCLK/FSCLK IO oen 控制。 0 : BCLK/FSCLK IO 为输出; 1 : BCLK/FSCLK IO 为输入。 注意: 需要和 aiao_bclk_sel 配对使用, 实现 I2S 接口主从模式选择。	0x0



Bits	Access	Name	Description	Reset
[9]	RW	aiao_srst_req	软复位请求。 0: 撤消复位; 1: 复位。	0x0
[8]	RW	aiao_cken	时钟状态。 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 FSCLK 的分频关系。 000: FSCLK 是 BCLK 的 16 分频; 001: FSCLK 是 BCLK 的 32 分频; 010: FSCLK 是 BCLK 的 48 分频; 011: FSCLK 是 BCLK 的 64 分频; 100: FSCLK 是 BCLK 的 128 分频; 101: FSCLK 是 BCLK 的 256 分频; 其他: FSCLK 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。 0x0: BCLK 是 MCLK 的 1 分频; 0x1: BCLK 是 MCLK 的 3 分频; 0x2: BCLK 是 MCLK 的 2 分频; 0x3: BCLK 是 MCLK 的 4 分频; 0x4: BCLK 是 MCLK 的 6 分频; 0x5: BCLK 是 MCLK 的 8 分频; 0x6: BCLK 是 MCLK 的 12 分频; 0x7: BCLK 是 MCLK 的 16 分频; 0x8: BCLK 是 MCLK 的 24 分频; 0x9: BCLK 是 MCLK 的 32 分频; 0xA: BCLK 是 MCLK 的 48 分频;	0x1



Bits	Access	Name	Description	Reset
			0xB: BCLK 是 MCLK 的 64 分频; 其他: BCLK 是 MCLK 的 8 分频。	

I2S_CRG_CFG0_01

I2S_CRG_CFG0_01 为 I2S01 CRG 配置 0 号寄存器。

Offset Address: 0x0108 Total Reset Value: 0x00AA_AAAA

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x00
[26:0]	RW	aiao_mclk_div	MCLK 的分频时钟配置值, 配置值为 (MCLK/SIO 时钟源头频率) × 2 ²⁷ 。	0x0AAAAAA

I2S_CRG_CFG1_01

I2S_CRG_CFG1_01 为 I2S01 CRG 配置 1 号寄存器。

Offset Address: 0x010C Total Reset Value: 0x0000_0131

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x00000
[15]	-	aiao_ws_en	WS 使能。 0: 不使能; 1: 使能。	0x0
[14]	-	aiao_bclk_en	BCLK 使能。 0: 不使能; 1: 使能。	0x0
[13]	RW	aiao_bclkout_ctrl	BCLKOUT 的极性控制。 0: 正向; 1: 反向。	0x0



Bits	Access	Name	Description	Reset
[12]	RW	aiao_bclkin_pctrl	BCLKIN 的极性控制。 0: 正向; 1: 反向。	0x0
[11]	RW	aiao_bclk_sel	BCLK/FSCLK 选择。 0: 时钟从内部产生; 1: 时钟从外部输入。	0x0
[10]	RW	aiao_bclk_oen	BCLK/FSCLK IO oen 控制。 0: BCLK/FSCLK IO 为输出; 1: BCLK/FSCLK IO 为输入。 注意: 需要和 aiao_bclk_sel 配对使用, 实现 I2S 接口主从模式选择。	0x0
[9]	RW	aiao_srst_req	软复位请求。 0: 撤消复位; 1: 复位。	0x0
[8]	RW	aiao_cken	时钟状态。 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 FSCLK 的分频关系。 000: FSCLK 是 BCLK 的 16 分频; 001: FSCLK 是 BCLK 的 32 分频; 010: FSCLK 是 BCLK 的 48 分频; 011: FSCLK 是 BCLK 的 64 分频; 100: FSCLK 是 BCLK 的 128 分频; 101: FSCLK 是 BCLK 的 256 分频; 其他: FSCLK 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。	0x1



Bits	Access	Name	Description	Reset
			0x0: BCLK 是 MCLK 的 1 分频; 0x1: BCLK 是 MCLK 的 3 分频; 0x2: BCLK 是 MCLK 的 2 分频; 0x3: BCLK 是 MCLK 的 4 分频; 0x4: BCLK 是 MCLK 的 6 分频; 0x5: BCLK 是 MCLK 的 8 分频; 0x6: BCLK 是 MCLK 的 12 分频; 0x7: BCLK 是 MCLK 的 16 分频; 0x8: BCLK 是 MCLK 的 24 分频; 0x9: BCLK 是 MCLK 的 32 分频; 0xA: BCLK 是 MCLK 的 48 分频; 0xB: BCLK 是 MCLK 的 64 分频; 其他: BCLK 是 MCLK 的 8 分频。	

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。	0x00
[26:0]	RW	aiao_mclk_div	MCLK 的分频时钟配置值，配置值为 (MCLK/SIO 时钟源头频率) × 2 ²⁷ 。	0x0AAAAAA

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
[31:16]	-	reserved	保留。	0x0000
[15]	RW	aiao_ws_en	WS 使能。 0: 不使能; 1: 使能。	0x0
[14]	RW	aiao_bclk_en	BCLK 使能。 0: 不使能; 1: 使能。	0x0
[13]	RW	aiao_bclkout_pctrl	BCLKOUT 的极性控制。 0: 正向; 1: 反向。	0x0
[12]	RW	aiao_bclkin_pctrl	BCLKIN 的极性控制。 0: 正向; 1: 反向。	0x0
[11]	RW	aiao_bclk_sel	BCLK/FSCLK 选择。 0: 时钟从内部产生; 1: 时钟从外部输入。	0x0
[10]	RW	aiao_bclk_oen	BCLK/FSCLK IO oen 控制。 0: BCLK/FSCLK IO 为输出; 1: BCLK/FSCLK IO 为输入。 注意: 需要和 aiao_bclk_sel 配对使用, 实现 I2S 接口主从模式选择。	0x0
[9]	RW	aiao_srst_req	软复位请求。 0: 撤消复位; 1: 复位。	0x0
[8]	RW	aiao_cken	时钟状态。 0: 关闭; 1: 打开。	0x1



Bits	Access	Name	Description	Reset
[7]	-	reserved	保留。	0x0
[6:4]	RW	aiao_fsclk_div	位时钟 BCLK 与采样时钟 FSCLK 的分频关系。 000: FSCLK 是 BCLK 的 16 分频; 001: FSCLK 是 BCLK 的 32 分频; 010: FSCLK 是 BCLK 的 48 分频; 011: FSCLK 是 BCLK 的 64 分频; 100: FSCLK 是 BCLK 的 128 分频; 101: FSCLK 是 BCLK 的 256 分频; 其他: FSCLK 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。 0x0: BCLK 是 MCLK 的 1 分频; 0x1: BCLK 是 MCLK 的 3 分频; 0x2: BCLK 是 MCLK 的 2 分频; 0x3: BCLK 是 MCLK 的 4 分频; 0x4: BCLK 是 MCLK 的 6 分频; 0x5: BCLK 是 MCLK 的 8 分频; 0x6: BCLK 是 MCLK 的 12 分频; 0x7: BCLK 是 MCLK 的 16 分频; 0x8: BCLK 是 MCLK 的 24 分频; 0x9: BCLK 是 MCLK 的 32 分频; 0xA: BCLK 是 MCLK 的 48 分频; 0xB: 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。	0x00
[26:0]	RW	aiao_mclk_div	MCLK 的分频时钟配置值，配置值为 (MCLK/SIO 时钟源头频率) × 2 ²⁷ 。	0x0AAAAA

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
[31:16]	-	reserved	保留。	0x0000
[15]	RW	aiao_ws_en	WS 使能。 0: 不使能; 1: 使能。	0x0
[14]	RW	aiao_bclk_en	BCLK 使能。 0: 不使能; 1: 使能。	0x0
[13]	RW	aiao_bclkout_pctrl	BCLKOUT 的极性控制。 0: 正向; 1: 反向。	0x0
[12]	RW	aiao_bclk_in_pctrl	BCLKIN 的极性控制。 0: 正向; 1: 反向。	0x0
[11]	RW	aiao_bclk_sel	BCLK/FSCLK 选择。 0: 时钟从内部产生; 1: 时钟从外部输入。	0x0
[10]	RW	aiao_bclk_oen	BCLK/FSCLK IO 方向控制。 0: BCLK/FSCLK IO 为输出;	0x0



Bits	Access	Name	Description	Reset
			1: BCLK/FSCLK IO 为输入。 注意: 需要和 aiao_bclk_sel 配对使用, 实现 I2S 接口主从模式选择。	
[9]	RW	aiao_srst_req	软复位请求。 0: 撤消复位; 1: 复位。	0x0
[8]	RW	aiao_cken	时钟状态。 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 FSCLK 的分频关系。 000: FSCLK 是 BCLK 的 16 分频; 001: FSCLK 是 BCLK 的 32 分频; 010: FSCLK 是 BCLK 的 48 分频; 011: FSCLK 是 BCLK 的 64 分频; 100: FSCLK 是 BCLK 的 128 分频; 101: FSCLK 是 BCLK 的 256 分频; 其他: FSCLK 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。 0x0: BCLK 是 MCLK 的 1 分频; 0x1: BCLK 是 MCLK 的 3 分频; 0x2: BCLK 是 MCLK 的 2 分频; 0x3: BCLK 是 MCLK 的 4 分频; 0x4: BCLK 是 MCLK 的 6 分频; 0x5: BCLK 是 MCLK 的 8 分频; 0x6: BCLK 是 MCLK 的 12 分频; 0x7: BCLK 是 MCLK 的 16 分频;	0x1



Bits	Access	Name	Description	Reset
			0x8: BCLK 是 MCLK 的 24 分频; 0x9: BCLK 是 MCLK 的 32 分频; 0xA: BCLK 是 MCLK 的 48 分频; 0xB: BCLK 是 MCLK 的 64 分频; 其他: BCLK 是 MCLK 的 8 分频。	

RX_IF_ATTRI

RX_IF_ATTRI 为接收通道的接口属性设置寄存器。

Offset Address: 0x1000 + 0x100×n Total Reset Value: 0xE408_0004

Bits	Access	Name	Description	Reset
[31:24]	RO	reserved	保留。	0xE4
[23:20]	RW	rx_sd_source_sel	SD0, SD1, SD2, SD3 接口源选择。 0x0: I2S TX0 接口; 0x1: I2S TX1 接口; ... 0x7: I2S TX7 接口; 0x8: I2S RX0 接口; 0x9: I2S RX1 接口; ... 0xF: I2S RX7 接口。	0x0
[19]	RO	reserved	保留。	0x1
[18:16]	RW	rx_trackmode	I2S 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;	0x0



Bits	Access	Name	Description	Reset
			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。 注意: 单声道接收时, trackmode 仍然有效。	
[15:8]	RW	rx_sd_offset	PCM 模式下, 数据相对帧同步信号延迟 n 个 BCLK 周期。 0x00: 0 bit clocks 0x01: 1 bit clocks 0x02: 2 bit clocks 0xFE: 254 bit clocks 0xFF: 255 bit clocks	0x00
[7]	RW	rx_multislot_en	时分复用有效指示位。 0: 时分复用模式无效, 属于正常模式; 1: 时分复用模式有效。	0x0
[6]	-	reserved	保留。	0x0
[5:4]	RW	rx_ch_num	接收的路数选择。 <ul style="list-style-type: none"> • rx_multislot_en=0 <ul style="list-style-type: none"> 00: 1 路(ch)接收; (SD0 数据线) 01: 2 路(ch)接收; (SD0 数据线) 11: 8 路(ch)接收。(SD0,SD1,SD2,SD3 数据线) 其他: 保留 • rx_multislot_en=1 <ul style="list-style-type: none"> 时分复用方式下, 接收的路数选择。 00: 2 路接收; (SD0 数据线) 	0x0



Bits	Access	Name	Description	Reset
			01: 4 路接收; (SD0 数据线) 10: 8 路接收; (SD0 数据线) 11: 16 路接收。(SD0 数据线) 注意: 多路接收时, 只用第 0 根数据线 SD0。	
[3:2]	RW	rx_i2s_precision	数据采样精度配置位。 <ul style="list-style-type: none"> • rx_multislot_en=0 <ul style="list-style-type: none"> ◆ I2S 正常模式。 <ul style="list-style-type: none"> 00: 保留; 01: 16bit; 10: 24bit; 11: 保留。 ◆ PCM 正常模式。 <ul style="list-style-type: none"> 00: 8bit; 01: 16bit; 其他: 保留。 • rx_multislot_en=1 I2S/PCM 模式。 <ul style="list-style-type: none"> 00: 8bit; 01: 16bit; 其他: 保留。 	0x1
[1:0]	RW	rx_mode	接收通道的接口模式选择。 00: I2S 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 + 0x100×n 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、停止指示位。 0: 停止; 1: 启动。	0x0
[27]	RW	bypass_en	运算处理 bypass, 控制功能仍然生效制。 0: bypass 功能关闭; 1: bypass 功能打开; trackmode 功能被 bypass。	0x0
[26:0]	-	reserved	保留。	0x0000000

RX_BUFF_SADDR_HI

RX_BUFF_SADDR_HI 为接收通道的 DDR 缓存起始地址高 32bit 寄存器。

Offset Address: 0x107C + 0x100×n Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	rx_buff_saddr_hi	接收通道的 DDR 缓存起始地址的高 32bit, 以字节单位。 注意: DDR 缓存起始地址要求 32Byte 对齐。	0x00000000

RX_BUFF_SADDR_LOW

RX_BUFF_SADDR_LOW 为接收通道的 DDR 缓存起始地址低 32bit 寄存器。

Offset Address: 0x1080 + 0x100×n Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	rx_buff_saddr_low	起始地址的低 32bit。	0x00000000

RX_BUFF_SIZE

RX_BUFF_SIZE 为接收通道的 DDR 缓存大小寄存器。

Offset Address: 0x1084 + 0x100×n Total Reset Value: 0x0000_0032

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。 注意: 要求 rx_buff_size 是 32 字节的整数倍。	0x000032

RX_BUFF_WPTR

RX_BUFF_WPTR 为接收通道的 DDR 缓存写地址寄存器。

Offset Address: 0x1088 + 0x100×n 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。 注意: 1.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; 2.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: $0x108C + 0x100 \times n$ 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。 注意： 1.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； 2.软件按照字节为单位，硬件内部按照 32 字节对齐操作。	0x000000

RX_BUFF_ALFULL_TH

RX_BUFF_ALFULL_TH 为接收通道的 DDR 缓存几乎满流水线寄存器。

Offset Address: $0x1090 + 0x100 \times n$ 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 缓存几乎满流水线，以字节为单位。当 DDR 缓存可写空间小于几乎满流水线时，产生几乎满原始中断。 注意：如果使用 RX_INT_RAW[rx_alfull_int_raw]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 + 0x100×n 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_WPTR_TMP

RX_WPTR_TMP 为上报传输完成中断时, 保存接收通道的写地址寄存器。

Offset Address: 0x1098 + 0x100×n Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RO	rx_wptr_tmp	上报传输完成中断时, 保存接收通道的写地址, 以字节为单位。	0x000000

RX_INT_ENA

RX_INT_ENA 为接收通道的中断使能寄存器。

Offset Address: 0x10A0 + 0x100×n Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RW	rx_if_full_lost_int_ena	接收通道的接口数据满, 导致数据丢失的原始中断使能。 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。	0x0



Bits	Access	Name	Description	Reset
			0: 不使能; 1: 使能。	
[4]	RW	rx_ififo_full_int_ena	接收通道的接口 FIFO 上溢中断使能。 0: 不使能; 1: 使能。	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 + 0x100×n Total Reset Value: 0x0000_0020

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, 导致数据丢失的原始中断。 0: 无原始中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有原始中断。	
[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。	0x1
[4]	-	reserved	保留。	0x0
[3]	RO	rx_fifo_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 + 0x100×n Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RO	tx_if_full_lost_int_status	接收通道的接口数据满,导致数据丢失的中断状态。	0x0



Bits	Access	Name	Description	Reset
			0: 无中断状态; 1: 有中断状态。	
[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]	RO	rx_ififo_full_int_status	接收通道的接口 FIFO 上溢中断状态。 0: 无中断状态; 1: 有中断状态。	0x0
[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: $0x10AC + 0x100 \times n$ Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RW	rx_if_full_lost_int_clear	接收通道的接口数据满丢失中断清除。 0: 不清除; 1: 清除。	0x0
[6]	-	reserved	保留。	0x0
[5]	RW	rx_stop_int_clear	接收通道的停止中断清除位。 0: 不清除; 1: 清除。	0x0
[4]	RW	rx_ififo_full_int_clear	接收通道的接口 fifo 上溢中断清除位。 0: 不清除; 1: 清除。	0x0
[3]	RW	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。	0x0
[2]	RW	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。	0x0
[1]	RW	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。	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。	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 + 0x100×m Total Reset Value: 0xE400_0004

Bits	Access	Name	Description	Reset
[31:24]	RO	reserved	保留。	0xE4
[23:20]	RW	tx_sd_source_sel	SD0, SD1, SD2, SD3 接口源选择。 0x0: I2S TX0; 0x1: I2S TX1; 0x7: I2S TX7; 0x8: I2S RX0; 0x9: I2S RX1; 0xF: I2S RX7。	0x0
[19]	-	reserved	保留。	0x0
[18:16]	RW	tx_trackmode	I2S 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。 注意: 单声道接收时, trackmode 仍然有效。	0x0
[15:8]	RW	tx_sd_offset	PCM 模式下, 数据相对帧同步信号延迟 n 个 BCLK 周期。 0x00: 0 bit clocks;	0x00



Bits	Access	Name	Description	Reset
			0x01: 1 bit clocks; 0x02: 2 bit clocks; ... 0xFE: 254 bit clocks; 0xFF: 255 bit clocks.	
[7]	RW	tx_multislot_en	时分复用使能控制位。 0: 正常模式 (非时分复用) ; 1: 时分复用模式。	0x0
[6]	RW	tx_underflow_ctrl	欠载时 aiao 的输出值。 0: 输出 0 值; 1: 输出最后一个数据值。	0x0
[5:4]	RW	tx_ch_num	发送路数选择。 <ul style="list-style-type: none"> tx_multislot_en=0 <ul style="list-style-type: none"> 00: 1 路发送; 01: 2 路发送; 10: 保留; 11: 保留。 tx_multislot_en=1 <ul style="list-style-type: none"> 分时复用方式下, 发送的路数选择。 11: 8 路发送。 其他: 保留。 注意: 多路发送时, 只用第 0 根数据线 sd0。	0x0
[3:2]	RW	tx_i2s_precision	数据采样精度配置位。 <ul style="list-style-type: none"> I2S 正常模式: <ul style="list-style-type: none"> 00: 保留; 01: 16bit; 10: 24bit; 11: 保留。 	0x1



Bits	Access	Name	Description	Reset
			<ul style="list-style-type: none"> PCM 正常模式。 <ul style="list-style-type: none"> 00: 8bit; 01: 16bit; 其他: 保留。 时分复用下(I2S)。 <ul style="list-style-type: none"> 00: 保留; 01: 16bit; 10: 24bit; 11: 保留。 	
[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 + 0x100×m 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。 0: 未完成; 1: 完成。	0x1
[28]	RW	tx_enable	发送通道的启动、停止指示位。 0: 停止; 1: 启动。	0x0
[27]	RW	bypass_en	运算处理 bypass, 控制功能仍然生效制。 0: bypass 功能关闭;	0x0



Bits	Access	Name	Description	Reset
			1: bypass 功能打开; 音量控制, trackmode, 淡入淡出等功能被 bypass.	
[26:24]	-	reserved	保留。	0x0
[23:20]	RW	fade_out_rate	淡出速度。 0x0: 1 个采样点改变一次; 0x1: 2 个采样点改变一次; 0x2: 4 个采样点改变一次; 0x3: 8 个采样点改变一次; 0x4: 16 个采样点改变一次; 0x5: 32 个采样点改变一次; 0x6: 64 个采样点改变一次; 0x7: 128 个采样点改变一次; 其他: 保留。	0x0
[19:16]	RW	fade_in_rate	淡入速度。 0x0: 1 个采样点改变一次; 0x1: 2 个采样点改变一次; 0x2: 4 个采样点改变一次; 0x3: 8 个采样点改变一次; 0x4: 16 个采样点改变一次; 0x5: 32 个采样点改变一次; 0x6: 64 个采样点改变一次; 0x7: 128 个采样点改变一次; 其他: 保留。	0x0
[15]	-	reserved	保留。	0x0
[14:8]	RW	volume	音量控制。 0x00 ~ 0x28: 静音; 0x29: -80dB;	0x00



Bits	Access	Name	Description	Reset
			0x7E: +5dB; 0x7F: +6dB。	
[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_WS_CNT

TX_WS_CNT 为接收处理通道的 ws 循环计数状态寄存器。

Offset Address: 0x2020 + 0x100×m Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RO	ws_count	fsclk 循环计数寄存器。 计数单位: fsclk。	0x000000

TX_BCLK_CNT

TX_BCLK_CNT 为接收处理通道的 bclk 循环计数状态寄存器。

Offset Address: 0x2024 + 0x100×n Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RO	bclk_count	bclk 循环计数寄存器。 计数单位: bclk。	0x000000



TX_BUFF_SADDR_HI

TX_BUFF_SADDR_HI 为发送通道的 DDR 缓存起始地址高 32bit 寄存器。

Offset Address: 0x207C + 0x100×n Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tx_buff_saddr_hi	发送通道的 DDR 缓存起始地址的高 32bit, 以字节单位。 注意: DDR 缓存起始地址要求 32Byte 对齐。	0x00000000

TX_BUFF_SADDR_LOW

TX_BUFF_SADDR_LOW 为发送通道的 DDR 缓存起始地址低 32bit 寄存器。

Offset Address: 0x2080 + 0x100×m Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tx_buff_saddr_low	发送通道的 DDR 缓存起始低 32bit, 以字节为单位。 注意: DDR 缓存起始地址要求 128×2Byte 对齐。	0x00000000

TX_BUFF_SIZE

TX_BUFF_SIZE 为发送通道的 DDR 缓存大小寄存器。

Offset Address: 0x2084 + 0x100×m 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。 注意: 要求 tx_buff_size 是 32 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: $0x2088 + 0x100 \times m$ 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。 注意：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。 1. 软件必须保证 TX_BUF 空闲空间不小于 32 字节； 2. 软件按照字节为单位，硬件内部按照 32 字节对齐操作。	0x000000

TX_BUFF_RPTR

TX_BUFF_RPTR 为发送通道的 DDR 缓存读地址寄存器。

Offset Address: $0x208C + 0x100 \times m$ 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。 注意：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 $128 \times 2\text{Byte}$ 对齐。	0x000000

TX_BUFF_ALEMPY_TH

TX_BUFF_ALEMPY_TH 为发送通道的 DDR 缓存几乎空水线寄存器。

Offset Address: $0x2090 + 0x100 \times m$ 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 缓存几乎空水线，以字节为单位。当 DDR 缓存可读空间小于几乎空水线时，产生几乎空原始中断。 注意：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。	0x000000

TX_TRANS_SIZE

TX_TRANS_SIZE 为发送通道的数据传输长度寄存器。

Offset Address: 0x2094 + 0x100×m 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_RPTR_TMP

TX_RPTR_TMP 为上报传输完成中断时，保存发送通道的读地址寄存器。

Offset Address: 0x2098 + 0x100×m Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:0]	RO	tx_rptr_tmp	上报传输完成中断时，保存发送通道的读地址，以字节为单位。	0x000000



TX_INT_ENA

TX_INT_ENA 为发送通道的中断使能寄存器。

Offset Address: 0x20A0 + 0x100×m 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: 不使能; 1: 使能。	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
[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;	0x0



Bits	Access	Name	Description	Reset
			1: 使能。	

TX_INT_RAW

TX_INT_RAW 为发送通道的原始中断寄存器。

Offset Address: 0x20A4 + 0x100×m 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; 1: 有原始中断。	0x0
[2]	RO	tx_aleempty_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。	0x0



Bits	Access	Name	Description	Reset
			0: 无原始中断; 1: 有原始中断。	
[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 + 0x100×m 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: 无中断状态; 1: 有中断状态。	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;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断状态。	
[2]	RO	tx_alempty_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 + 0x100×m Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7]	RW	tx_dat_break_int_clear	发送通道的接口数据断流中断清除。 0: 不清除; 1: 清除。	0x0
[6]	RW	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。	0x0
[5]	RW	tx_stop_int_clear	发送通道的停止中断清除位。 0: 不清除; 1: 清除。	0x0



Bits	Access	Name	Description	Reset
[4]	RW	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。	0x0
[3]	RW	tx_bfifo_empty_int_clear	发送通道的总线 FIFO 下溢中断清除位。 TX_DSP 模块使用。 0: 不清除; 1: 清除。	0x0
[2]	RW	tx_alempty_int_clear	发送通道的 DDR 缓存几乎空中断清除位。 0: 不清除; 1: 清除。	0x0
[1]	RW	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。	0x0
[0]	RW	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。	0x0

12.2 Audio Codec

12.2.1 概述

芯片集成高性能的 Audio Codec，包括高品质回放 DAC (90dB DR A-Weighted)，支持 1 路单端立体声 lineout 输出；高品质录音 ADC (90dB DR A-Weighted)，支持 1 路立体声差分输入（两对 LINE IN 或 MIC IN 差分输入），麦克风输入支持 0~30dB，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 96kHz 的标准采样率。

12.2.2 特点

Audio Codec 模块有如下特点：



- 90dBA DR DAC, 支持 1 对立体声 lineout 输出
- DAC 数字音量控制范围: -121dB ~ 6dB, 1dB 步长
- 90dBA DR ADC, 支持 1 路立体声差分输入 (两对 LINE IN 或 MIC IN 差分输入)
- ADC 通路模拟音量控制范围: 0~30dB, 2dB 步长, 另外一档 Boost gain 为 20dB
- 提供内部麦克风偏置
- 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;
 - 44.1kHz 系列采样率包括 11.025kHz、22.05kHz、44.1kHz;
 - 48kHz 系列采样率包括 12kHz、24kHz、48kHz、96kHz。

12.2.3 Audio Codec 寄存器概览

Audio Codec 寄存器概览如表 12-3 所示。

表12-3 Audio Codec 寄存器概览 (基址是 0x0_17C4_0000)

偏移地址	名称	描述	页码
0x0000	AUDIO_ANA_CTRL_0	Audio Codec ANA 控制寄存器 0	12-53
0x0004	AUDIO_ANA_CTRL_1	Audio Codec ANA 控制寄存器 1	12-54
0x0008	AUDIO_ANA_CTRL_2	Audio Codec ANA 控制寄存器 2	12-55
0x000C	AUDIO_ANA_CTRL_3	Audio Codec ANA 控制寄存器 3	12-55
0x00CC	AUDIO_CTRL_REG_1	Audio Codec DIG 控制寄存器 0	12-56
0x00D0	AUDIO_DAC_REG_0	Audio Codec DIG 控制寄存器 1	12-58
0x00D4	AUDIO_DAC_REG_1	Audio Codec DIG 控制寄存器 2	12-59
0x00D8	AUDIO_ADC_REG_0	Audio Codec DIG 控制寄存器 3	12-60



12.2.4 Audio Codec 寄存器描述

AUDIO_ANA_CTRL_0

AUDIO_ANA_CTRL_0 为 Audio Codec ANA 控制寄存器 0。

Offset Address: 0x0000 Total Reset Value: 0x1C1C_DFFF

Bits	Access	Name	Description	Reset
[31:27]	RW	LINEIN_R_GAIN	右通路输入 PGA 增益控制。 0x0: 0dB; 0x1: 2dB; 0xF: 30dB。	0x03
[26:24]	RW	LINEIN_R_SEL	右通路输入 PGA 输入通道选择。 000: MIC 差分输入; 001: MIC_R 单端输入; 010: MIC_L 单端输入; 011: LINEIN_L 单端输入; 100: LINEIN_R 单端输入; 101: LINEIN 差分输入; 其他: 保留。	0x4
[23:19]	RW	LINEIN_L_GAIN	左通路输入 PGA 增益控制。 0x00: 0dB; 0x01: 2dB; ... 0x0F: 30dB。	0x03
[18:16]	RW	LINEIN_L_SEL	左通路输入 PGA 输入通道选择。 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;	0x4



Bits	Access	Name	Description	Reset
			100: LINEIN_L 单端输入; 101: LINEIN 差分输入; 其他: 保留。	
[15]	RW	PD_MICBIAS1	MICBIAS1 上下电控制。 0: 正常工作; 1: 下电。	0x1
[14:2]	-	reserved	保留。	0x17FF
[1]	RW	MUTE_LINEIN_L	左通路输入 PGA 静音控制。 0: 正常工作; 1: 静音。	0x1
[0]	RW	MUTE_LINEIN_R	右通路输入 PGA 静音控制。 0: 正常工作; 1: 静音。	0x1

AUDIO_ANA_CTRL_1

AUDIO_ANA_CTRL_1 为 Audio Codec ANA 控制寄存器 1。

Offset Address: 0x0004 Total Reset Value: 0xDF60_5E65

Bits	Access	Name	Description	Reset
[31:20]	RW	reserved	保留。	0xDF6
[19]	RW	MODE_ADCL	左通路 ADC 模式选择信号。 0: 6.144MHz 采样率(48kHz 系列); 1: 4.096MHz 采样率(32kHz 系列)。	0x0
[18]	RW	MODE_ADCR	右通路 ADC 模式选择信号。 0: 6.144MHz 采样率(48kHz 系列); 1: 4.096MHz 采样率(32kHz 系列)。	0x0
[17]	RW	BOOST_ADCL	左通路 ADC Boost 开关控制。	0x0



Bits	Access	Name	Description	Reset
			0: 不开 Boost; 1: 打开 Boost=20dB。	
[16]	RW	BOOST_ADCR	右通路 ADC Boost 开关控制。 0: 不开 Boost; 1: 打开 Boost=20dB。	0x0
[15:0]	RW	reserved	保留。	0x5E65

AUDIO_ANA_CTRL_2

AUDIO_ANA_CTRL_2 为 AudioCodec 模控制寄存器 2。

Offset Address: 0x0008 Total Reset Value: 0x0025_5548

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	保留。	0x0
[29:27]	RW	MICBIAS_ADJ	MICBIAS 的输出电压选择。 000: 2.1V; 001: 2.2V; 111: 2.8V。	0x0
[26:1]	-	reserved	保留。	0x12AAA4
[0]	RW	ANA_LOOP	模拟环回使能控制。 0: DAC 的输入由数字部分提供; 1: ADC 到 DAC 模拟内部环回。	0x0

AUDIO_ANA_CTRL_3

AUDIO_ANA_CTRL_3 为 AudioCodec 模控制寄存器 3。

Offset Address: 0x000C Total Reset Value: 0x0429_3B50



Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27:10]	RW	reserved	保留。	0x10A4E
[9]	RW	mute_dacl	左通路 DAC 静音控制。 0: 正常工作; 1: 通路静音。	0x1
[8]	RW	mute_dacr	右通路 DAC 静音控制。 0: 正常工作; 1: 静音。	0x1
[7:0]	RW	reserved	保留。	0x50

AUDIO_CTRL_REG_1

AUDIO_CTRL_REG_1 为 Audio Codec DIG 控制寄存器 0。

Offset Address: 0x00CC Total Reset Value: 0x00F3_5A4A

Bits	Access	Name	Description	Reset
[31]	RW	dacl_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



Bits	Access	Name	Description	Reset
[27]	RW	dacl_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]	RO	reserved	保留。	0x3
[21:20]	RO	reserved	保留。	0x3
[19]	RO	reserved	保留。	0x0
[18]	RO	reserved	保留。	0x0
[17:13]	RW	i2s1_fs_sel	I2S 通道采样率选择。 0x18: mclk/512/2; 0x19: mclk/256/2; 0x1A: mclk/128/2; 0x1B: mclk/64/2; 0x1C~0x1F: mclk/32/2。 其他: 保留。	0x1A
[12:8]	RO	reserved	保留。	0x1A
[7:0]	-	reserved	保留。	0x4A



AUDIO_DAC_REG_0

AUDIO_DAC_REG_0 为 Audio Codec DIG 控制寄存器 1。

Offset Address: 0x00D0 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31]	RW	smutel	DACL soft-mute 控制位。 0: 关闭 soft-mute; 1: 开启 soft-mute。	0x0
[30]	RW	smuter	DACR soft-mute 控制位。 0: 关闭 soft-mute; 1: 开启 soft-mute。	0x0
[29]	RW	sunmutel	DACL soft-unmute 控制位。 0: 关闭 soft-unmute; 1: 开启 soft-unmute。	0x0
[28]	RW	sunmuter	DACR soft-unmute 控制位。 0: 关闭 soft-unmute; 1: 开启 soft-unmute。	0x0
[27]	RW	dacvu	DAC 音量更新控制位。 0: 不更新音量; 1: 更新音量。	0x0
[26:25]	RW	mutel_rate	DACL soft-mute 速率控制位。 00: fs/2; 01: fs/8; 10: fs/32; 11: fs/64。	0x0
[24:23]	RW	muter_rate	DACR soft-mute 速率控制位。 00: fs/2; 01: fs/8; 10: fs/32;	0x0



Bits	Access	Name	Description	Reset
			11: fs/64。	
[22:0]	RO	reserved	保留。	0x0000001

AUDIO_DAC_REG_1

AUDIO_DAC_REG_1 为 Audio Codec DIG 控制寄存器 2。

Offset Address: 0x00D4 Total Reset Value: 0x0606_2424

Bits	Access	Name	Description	Reset
[31]	RW	dacl_mute	DACL 数字静音控制。 0: 正常工作; 1: 静音。	0x0
[30:24]	RW	dacl_vol	DACL 数字音量控制。 计算公式为: $(6-dacl_vol*1)db$ 。 当 dacl_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。 计算公式为: $(6-dacr_vol*1)db$ 。 当 dacr_vol 为 0x7F 时, DACR 数字静音。 0x00: 6dB; 0x01: 5dB;	0x06



Bits	Access	Name	Description	Reset
			0x02: 4dB; ... 0x7E: -120dB; 0x7F: mute。	
[15:0]	RO	reserved	保留。	0x2424

AUDIO_ADC_REG_0

AUDIO_ADC_REG_0 为 Audio Codec DIG 控制寄存器 3。

Offset Address: 0x00D8 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 计算公式: $(30 - \text{adcl_vol} * 1) \text{db}$ 。 0x00: 30dB 0x01: 29dB 0x02: 28dB ... 0x7E: -96dB 0x7F: -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。 计算公式: $(30 - \text{adcr_vol} * 1) \text{db}$ 。 0x00: 30dB	0x1E



Bits	Access	Name	Description	Reset
			0x01: 29dB 0x02: 28dB ... 0x7E: -96dB 0x7F: -97dB	
[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
[13:0]	-	reserved	保留。	0x0001



目 录

13 外围设备	13-1
13.1 I ² C	13-1
13.1.1 概述	13-1
13.1.2 功能描述	13-1
13.1.3 功能框图	13-2
13.1.4 时序描述原理	13-2
13.1.5 工作方式	13-12
13.1.6 I ² C 寄存器概览	13-15
13.1.7 I ² C 寄存器描述	13-17
13.2 UART	13-32
13.2.1 概述	13-32
13.2.2 特点	13-33
13.2.3 功能描述	13-33
13.2.4 工作方式	13-35
13.2.5 UART 寄存器概览	13-38
13.2.6 UART 寄存器描述	13-39
13.3 SPI	13-53
13.3.1 概述	13-53
13.3.2 特点	13-53
13.3.3 功能描述	13-54
13.3.4 三种外设总线时序	13-55
13.3.5 工作方式	13-62
13.3.6 SPI 寄存器概览	13-65
13.3.7 SPI 寄存器描述	13-66



13.4 SPI_3WIRE	13-75
13.4.1 概述	13-75
13.4.2 功能描述	13-75
13.4.3 工作方式	13-75
13.4.4 SPI_3WIRE 寄存器概览	13-76
13.4.5 SPI_3WIRE 寄存器描述	13-76
13.5 SPI_TFT	13-78
13.5.1 概述	13-78
13.5.2 功能描述	13-78
13.5.3 接口总线时序	13-79
13.5.4 工作方式	13-80
13.5.5 SPI_TFT 寄存器概览	13-81
13.5.6 SPI_TFT 寄存器描述	13-82
13.6 eMMC/SDIO/SD Card 控制器	13-88
13.6.1 概述	13-88
13.6.2 特点	13-88
13.6.3 功能描述	13-89
13.6.4 寄存器概览	13-101
13.6.5 MMC 寄存器描述	13-103
13.7 GPIO	13-137
13.7.1 概述	13-137
13.7.2 特点	13-138
13.7.3 工作方式	13-138
13.7.4 GPIO 寄存器概览	13-139
13.7.5 GPIO 寄存器描述	13-141
13.8 USB3.0	13-145
13.8.1 概述	13-145
13.8.2 功能描述	13-145
13.8.3 工作方式	13-147
13.8.4 USB3.0 寄存器概览	13-148
13.8.5 USB3.0 寄存器描述	13-151
13.9 LSADC	13-192



13.9.1 概述.....	13-192
13.9.2 特点.....	13-192
13.9.3 工作方式.....	13-193
13.9.4 LSADC 寄存器概览.....	13-196
13.9.5 LSADC 寄存器描述.....	13-197
13.10 PWM.....	13-207
13.10.1 概述.....	13-207
13.10.2 特点.....	13-207
13.10.3 工作方式.....	13-208
13.10.4 配置流程.....	13-214
13.10.5 寄存器偏移地址变量表寄存器概览.....	13-217
13.10.6 PWM 寄存器概览.....	13-218
13.10.7 PWM 寄存器描述.....	13-219



插图目录

图 13-1 控制器的功能框图.....	13-2
图 13-2 7bit 寻址, 写操作时序图.....	13-5
图 13-3 7bit 寻址, 直接读时序图.....	13-6
图 13-4 10bit 寻址, 写操作时序图.....	13-7
图 13-5 10bit 寻址, 组合读时序图.....	13-9
图 13-6 非标准时序图.....	13-11
图 13-7 UART 的典型应用框图.....	13-34
图 13-8 UART 帧格式.....	13-34
图 13-9 SPI 接 Slave 时的应用.....	13-55
图 13-10 SPI 单帧帧格式 (SPO=0、SPH=0).....	13-55
图 13-11 SPI 连续帧帧格式 (SPO=0、SPH=0).....	13-56
图 13-12 SPI 单帧帧格式 (SPO=0、SPH=1).....	13-57
图 13-13 SPI 连续帧帧格式 (SPO=0、SPH=1).....	13-57
图 13-14 SPI 单帧帧格式 (SPO=1、SPH=0).....	13-58
图 13-15 SPI 连续帧帧格式 (SPO=1、SPH=0).....	13-58
图 13-16 SPI 单帧帧格式 (SPO=1、SPH=1).....	13-59
图 13-17 SPI 连续帧帧格式 (SPO=1、SPH=1).....	13-59
图 13-18 SPI 接口时序图.....	13-60
图 13-19 TI 同步串行单帧帧格式.....	13-60
图 13-20 TI 同步串行连续帧帧格式.....	13-60



图 13-21 National Semiconductor Microwire 单帧帧格式.....	13-61
图 13-22 National Semiconductor Microwire 连续帧帧格式.....	13-61
图 13-23 SPI_TFT 帧格式.....	13-79
图 13-24 SPI_TFT 帧格式.....	13-79
图 13-25 SPI_TFT 帧格式.....	13-79
图 13-26 SPI_TFT 帧格式.....	13-80
图 13-27 MMC 功能框图.....	13-90
图 13-28 MMC 典型应用图.....	13-91
图 13-29 MMC 指令格式.....	13-92
图 13-30 MMC 指令响应格式.....	13-92
图 13-31 MMC 非数据指令操作.....	13-93
图 13-32 单块与多块读操作.....	13-94
图 13-33 单块与多块写操作.....	13-95
图 13-34 1bit 数据线传输模式下的块数据格式.....	13-95
图 13-35 4bit 数据线传输模式下的块数据格式.....	13-96
图 13-36 8bit 数据线传输模式下的块数据格式.....	13-96
图 13-37 USB 3.0 逻辑框图.....	13-146
图 13-38 单次扫描处理流程.....	13-193
图 13-39 连续扫描模式下通道轮询扫描示意图.....	13-194
图 13-40 连续扫描处理流程.....	13-194
图 13-41 16 次平均算法处理流程.....	13-195
图 13-42 左对齐输出波形.....	13-208
图 13-43 右对齐输出波形.....	13-209
图 13-44 中间对齐输出波形.....	13-210
图 13-45 三对互补信号输出波形.....	13-211
图 13-46 三对互补信号中间对齐时输出波形.....	13-212



图 13-47 死区控制输出波形.....	13-213
图 13-48 同步模式输出波形.....	13-214



表格目录

表 13-1 时序命令.....	13-3
表 13-2 7bit 寻址, 写操作对应的时序描述配置.....	13-5
表 13-3 7bit 寻址, 直接读操作的时序描述配置.....	13-6
表 13-4 10bit 寻址, 写操作对应的时序描述配置.....	13-7
表 13-5 10bit 寻址, 组合读对应的时序描述配置.....	13-9
表 13-6 非标准时序对应时序描述配置.....	13-11
表 13-7 I ² C 寄存器偏移地址变量表.....	13-16
表 13-8 I ² C 寄存器概览.....	13-16
表 13-9 UART 寄存器概览.....	13-38
表 13-10 SPI 寄存器概览.....	13-66
表 13-11 SPI_3WIRE 寄存器概览 (基地址 0x0_1107_5000)	13-76
表 13-12 SPI_TFT 寄存器概览 (基地址 0x0_1107_4000)	13-81
表 13-13 芯片中 3 个 MMC 控制器对应的功能信号和管脚名.....	13-89
表 13-14 传输模式表.....	13-97
表 13-15 传输模式表.....	13-97
表 13-16 传输模式表.....	13-97
表 13-17 MMC 寄存器概览 (eMMC/SDIO0/SDIO1 基地址分别是 0x0_1002_0000/0x0_1003_0000/0x0_1004_0000)	13-101
表 13-18 芯片 GPIO 寄存器对应的基地址.....	13-139
表 13-19 GPIO 寄存器概览.....	13-140
表 13-20 USB3.0 寄存器概览 (USB3.0 DRD 基地址是 0x0_1032_0000)	13-148



表 13-21 LSADC 寄存器概览 (基地址: 0x0_1110_0000)	13-196
表 13-22 各模块的寄存器偏移地址变量表	13-218
表 13-23 PWM 寄存器概览.....	13-218



13 外围设备

13.1 I²C

13.1.1 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。8 个控制器的工作时钟由相应 CRG 寄存器的 i2c_cksel 选择。

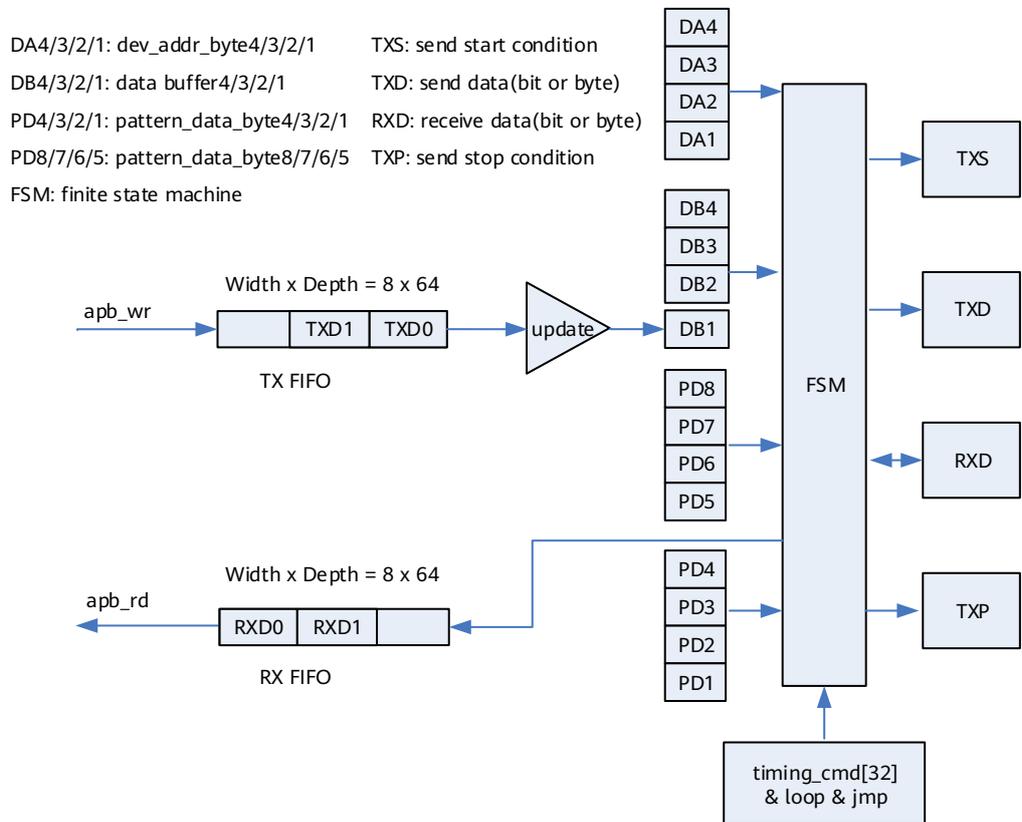
13.1.2 功能描述

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。

13.1.3 功能框图

图13-1 控制器的功能框图



13.1.4 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 13-1 所示。



表13-1 时序命令

时序命令编号	时序命令代号	时序命令含义
0x00	EXIT	结束，用于控制器退出
0x01	S	总线 START
0x02	SDA4	发送 dev_addr_byte4
0x03	SDA3	发送 dev_addr_byte3
0x04	SDA2	发送 dev_addr_byte2
0x05	SDA1	发送 dev_addr_byte1
0x06	SDB4	发送 data_buf_byte4
0x07	SDB3	发送 data_buf_byte3
0x08	SDB2	发送 data_buf_byte2
0x09	SDB1	发送 data_buf_byte1
0x0A	SPD8	发送 pattern_data_byte8
0x0B	SPD7	发送 pattern_data_byte7
0x0C	SPD6	发送 pattern_data_byte6
0x0D	SPD5	发送 pattern_data_byte5
0x0E	SPD4	发送 pattern_data_byte4
0x0F	SPD3	发送 pattern_data_byte3
0x10	SPD2	发送 pattern_data_byte2
0x11	SPD1	发送 pattern_data_byte1
0x12	RD	接收 1 字节数据 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 I ² C 总线的状态。
0x13	RACK	接收低电平应答
0x14	RNACK	接收高电平非应答



时序命令编号	时序命令代号	时序命令含义
0x15	RNC	接收应答, 高电平低电平都无所谓
0x16	SACK	发送低电平应答
0x17	SNACK	发送高电平非应答
0x18	JMPN1	有限次数跳转, 目的由 DST1 寄存器指出,次数由 LOOP1 寄存器指出
0x19	JMPN2	有限次数跳转, 目的由 DST2 寄存器指出,次数由 LOOP2 寄存器指出
0x1A	JMPN3	有限次数跳转, 目的由 DST3 寄存器指出,次数由 LOOP3 寄存器指出
0x1B	UNDEF	未定义
0x1C	UNDEF	未定义
0x1D	UDB1	从 TX FIFO 更新数据到 data_buf_byte1 注意: 执行 UDB1 命令时, 若 TX FIFO 为空, 则控制器会等待直到 TX FIFO 有数据; 等待期间控制器不会改变 I ² C 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13-2 7bit 寻址，写操作时序图

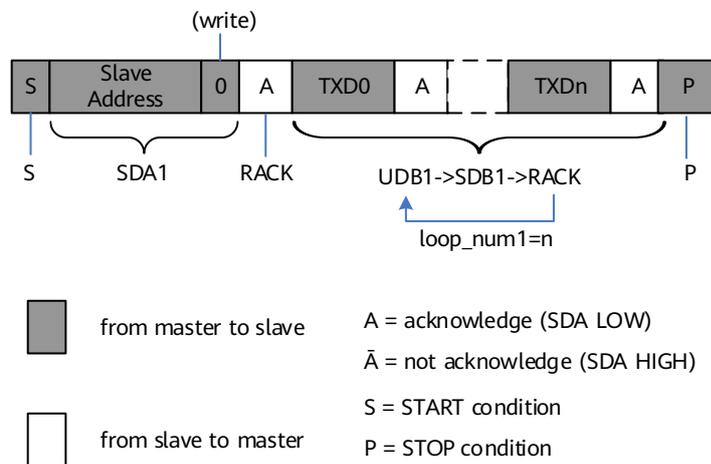


图 13-2 对应的时序描述配置如表 13-2 所示。

表13-2 7bit 寻址，写操作对应的时序描述配置

类型	寄存器	配置
时序格式描述	I2C_TIMING_CMD[0]	S
	I2C_TIMING_CMD[1]	SDA1
	I2C_TIMING_CMD[2]	RACK
	I2C_TIMING_CMD[3]	UDB1
	I2C_TIMING_CMD[4]	SDB1
	I2C_TIMING_CMD[5]	RACK
	I2C_TIMING_CMD[6]	JMPN1
	I2C_TIMING_CMD[7]	P
	I2C_TIMING_CMD[8]	EXIT
	I2C_LOOP1	n
	I2C_DST1	3
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13-3 7bit 寻址，直接读时序图

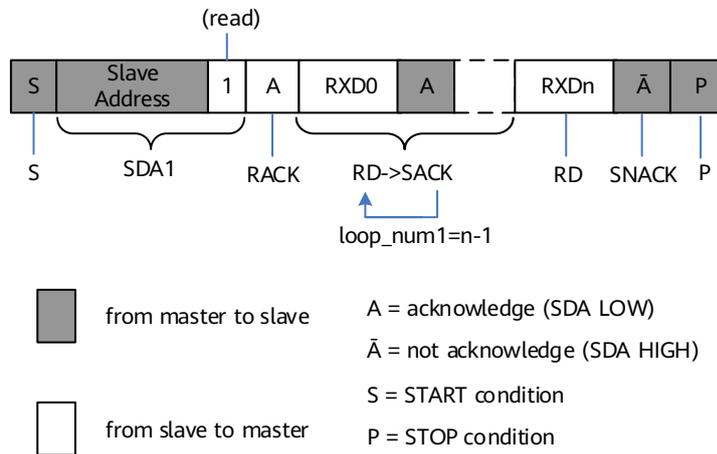


图 13-3 对应的时序描述配置如表 13-3 所示。

表13-3 7bit 寻址，直接读操作的时序描述配置

类型	寄存器	配置
时序格式描述	I2C_TIMING_CMD[0]	S
	I2C_TIMING_CMD[1]	SDA1
	I2C_TIMING_CMD[2]	RACK
	I2C_TIMING_CMD[3]	RD
	I2C_TIMING_CMD[4]	SACK
	I2C_TIMING_CMD[5]	JMPN1
	I2C_TIMING_CMD[6]	RD
	I2C_TIMING_CMD[7]	SNACK
	I2C_TIMING_CMD[8]	P
	I2C_TIMING_CMD[9]	EXIT
	I2C_LOOP1	n-1
	I2C_DST1	3



类型	寄存器	配置
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13-4 10bit 寻址，写操作时序图

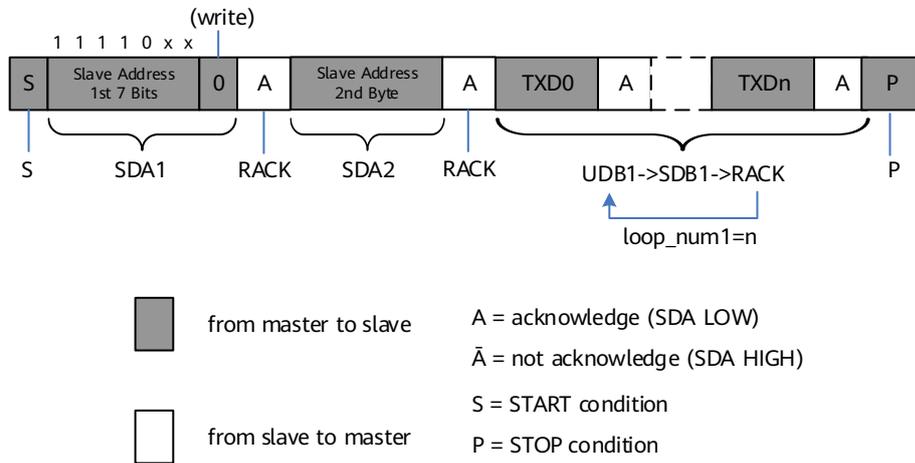


图 13-4 对应的时序描述配置如表 13-4 所示。

表13-4 10bit 寻址，写操作对应的时序描述配置

类型	寄存器	配置
时序格式描述	I2C_TIMING_CMD[0]	S
	I2C_TIMING_CMD[1]	SDA1
	I2C_TIMING_CMD[2]	RACK
	I2C_TIMING_CMD[3]	SDA2
	I2C_TIMING_CMD[4]	RACK
	I2C_TIMING_CMD[5]	UDB1



类型	寄存器	配置
	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13-5 10bit 寻址，组合读时序图

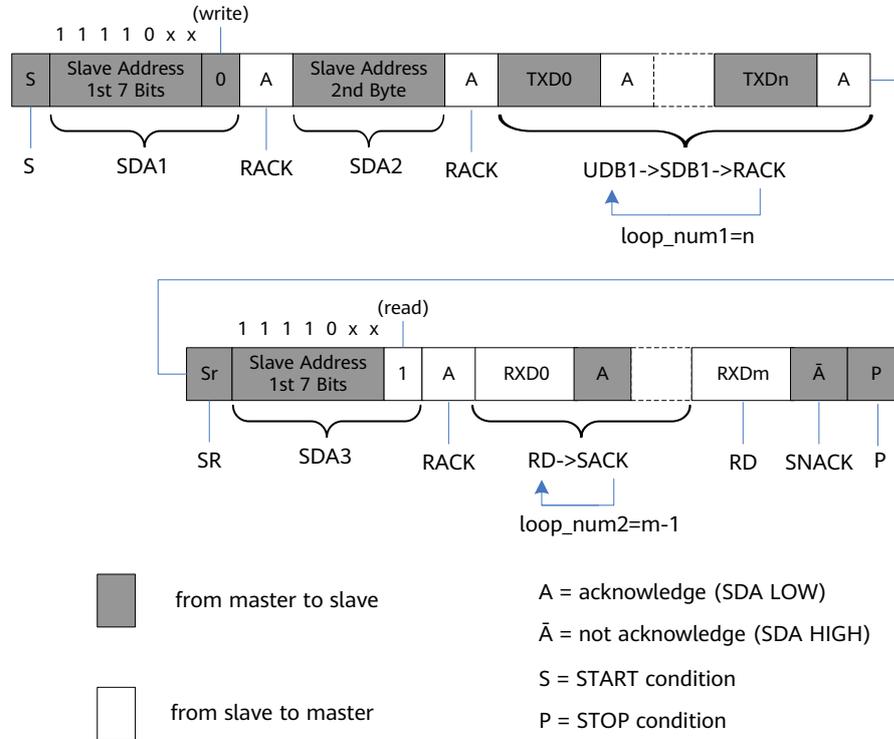


图 13-5 对应的时序描述配置如表 13-5 所示。

表13-5 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



类型	寄存器	配置
	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 13-6 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13-6 非标准时序图

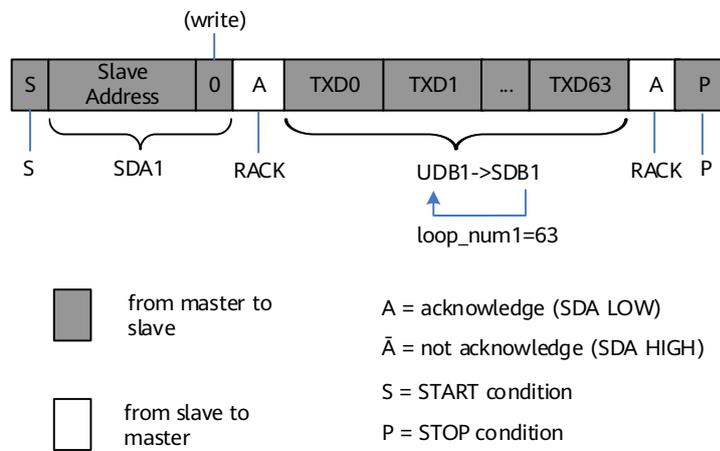


图 13-6 对应的时序描述配置如表 13-6 所示。

表13-6 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。



13.1.5 工作方式

13.1.5.1 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 13.1.4 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

13.1.5.2 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 13.1.4 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

13.1.5.3 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 13.1.4 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 则认为是异常，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 rx_gt_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

13.1.5.4 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 13.1.4 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**

13.1.5.5 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 13.1.4 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**



13.1.5.6 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 13.1.4 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

13.1.5.7 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

13.1.6 I²C 寄存器概览

I²C 基地址如下：

- I²C0 寄存器基地址：0x0_1106_0000。



- I²C1 寄存器基地址：0x0_1106_1000。
- I²C2 寄存器基地址：0x0_1106_2000。
- I²C3 寄存器基地址：0x0_1106_3000。
- I²C4 寄存器基地址：0x0_1106_4000。
- I²C5 寄存器基地址：0x0_1106_5000。
- I²C6 寄存器基地址：0x0_1106_6000。
- I²C7 寄存器基地址：0x0_1106_7000。

I²C 寄存器偏移地址中变量的取值范围和含义如表 13-7 所示。

表13-7 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13-8 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	13-17
0x0004	I2C_HCNT	I ² C 高电平时长寄存器	13-18
0x0008	I2C_LCNT	I ² C 低电平时长寄存器	13-18
0x0010	I2C_DEV_ADDR	I ² C 器件地址寄存器	13-19
0x0014	I2C_DATA_BUF	I ² C 数据 buf 寄存器	13-19
0x0018	I2C_PATTERN_DATA1	I ² C PATTERN 数据 1 寄存器	13-20
0x001C	I2C_PATTERN_DATA2	I ² C PATTERN 数据 2 寄存器	13-20
0x0020	I2C_TX_FIFO	I ² C TX FIFO 数据寄存器	13-21
0x0024	I2C_RX_FIFO	I ² C RX FIFO 数据寄存器	13-21
0x0030 + n x 4	I2C_TIMING_CMD	I ² C 时序命令寄存器	13-21
0x00B0	I2C_LOOP1	I ² C 循环次数 1 寄存器	13-23



偏移地址	名称	描述	页码
0x00B4	I2C_DST1	I ² C 跳转目的 1 寄存器	13-24
0x00B8	I2C_LOOP2	I ² C 循环次数 2 寄存器	13-24
0x00BC	I2C_DST2	I ² C 跳转目的 2 寄存器	13-24
0x00C0	I2C_LOOP3	I ² C 循环次数 3 寄存器	13-24
0x00C4	I2C_DST3	I ² C 跳转目的 3 寄存器	13-25
0x00C8	I2C_TX_WATERMARK	I ² C TX FIFO 水线寄存器	13-25
0x00CC	I2C_RX_WATERMARK	I ² C RX FIFO 水线寄存器	13-25
0x00D0	I2C_CTRL1	I ² C 控制寄存器 1	13-26
0x00D4	I2C_CTRL2	I ² C 控制寄存器 2	13-26
0x00D8	I2C_FIFO_STAT	I ² C FIFO 状态寄存器	13-28
0x00E0	I2C_INTR_RAW	I ² C 原始中断寄存器	13-28
0x00E4	I2C_INTR_EN	I ² C 中断使能寄存器	13-30
0x00E8	I2C_INTR_STAT	I ² C 中断状态寄存器	13-31

13.1.7 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]	-	reserved	保留。	0x00
[23:8]	RW	sda_hold_duration	SDA hold time 时长。建议配置为 0xa。 注意：只能控制从主机发往从设备的数据的 hold time 时长。	0x0000



Bits	Access	Name	Description	Reset
[7:1]	-	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]	-	reserved	保留。	0x0000
[15:0]	RW	i2c_high_duration	SCL 高电平时长。 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: $i2c_high_duration = (F_{I2C_CLK}/F_{SCL}) \times 0.5$ 快速模式下: $i2c_high_duration = (F_{I2C_CLK}/F_{SCL}) \times 0.36$ (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]	-	reserved	保留。	0x0000
[15:0]	RW	i2c_low_duration	<p>SCL 低电平时长。</p> <p>建议在标准模式下为 SCL 周期的 1/2，快速模式下为 SCL 周期的 0.64。</p> <p>标准模式下： $i2c_low_duration = (F_{I2C_CLK}/F_{SCL}) \times 0.5$</p> <p>快速模式下： $i2c_low_duration = (F_{I2C_CLK}/F_{SCL}) \times 0.64$ (F_{SCL} 为 SCL 总线的频率)</p> <p>以参考时钟为 50MHz 且工作在快速模式下 ($F_{SCL} = 400kHz$)为例， $i2c_low_duration = (50MHz / 400kHz) \times 0.64 = 80$。</p>	0x0000

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]	-	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]	-	reserved	保留。	0x000000
[7:0]	RO	rx_fifo	RX FIFO 出口。 读出的数据是 I ² C 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]	-	reserved	保留。	0x0000000
[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);	0x00



Bits	Access	Name	Description	Reset
			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 DST1 寄存器指出, 次数由 LOOP1 寄存器指	



Bits	Access	Name	Description	Reset
			<p>出);</p> <p>0x19: JMPN2(有限次数跳转, 目的由 DST2 寄存器指出,次数由 LOOP2 寄存器指出);</p> <p>0x1A: JMPN3(有限次数跳转, 目的由 DST3 寄存器指出,次数由 LOOP3 寄存器指出);</p> <p>0x1B: UNDEF (未定义);</p> <p>0x1C: UNDEF (未定义);</p> <p>0x1D: UDB1 (从 TX FIFO 更新数据到 data_buf_byte1);</p> <p>0x1E: SR (总线 repeated START);</p> <p>0x1F: P(总线 STOP)。</p> <p>注意:</p> <p>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。</p> <p>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。</p>	

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。	0x0000000
[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。	0x0000000

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。	0x0000000
[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。	0x0000000



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。	0x0000000
[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]	-	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]	-	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]	-	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]	-	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]	-	reserved	保留。	0x0
[24]	RO	i2c_sda_oen	监测内部 I2C 的 SCL 输出电平。	0x1



Bits	Access	Name	Description	Reset
			0: SCL 输出低电平; 1: SCL 输出高电平。	
[23:21]	-	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]	-	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]	-	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]	-	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]	-	reserved	保留。	0x0
[0]	RW	force_sda_oen	在 gpio_mode 有效时控制 SDA 管脚电平。 0: SDA 管脚为低电平; 1: SDA 管脚为高电平。	0x1



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]	-	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]	-	reserved	保留。	0x0
[14:8]	RO	tx_fifo_vld_num	TX FIFO 里的有效数目。	0x00
[7]	-	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]	-	reserved	保留。	0x00000
[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]	-	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]	-	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]	-	reserved	保留。	0x0
[0]	RWC	ack_bit_unmatch_raw	应答位不符合预期。 0: 无中断;	0x0



Bits	Access	Name	Description	Reset
			1: 有中断。	

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]	-	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]	-	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]	-	reserved	保留。	0x0
[2]	RW	rx_gt_watermark_en	rx_gt_watermark 中断使能。 0: 不使能;	0x0



Bits	Access	Name	Description	Reset
			1: 使能。	
[1]	-	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]	-	reserved	保留。	0x00000
[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]	-	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。	0x0



Bits	Access	Name	Description	Reset
			0: 无中断; 1: 有中断。	
[3]	-	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]	-	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

13.2 UART

13.2.1 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。

芯片共 6 个 UART 控制器:

- UART0: 2 线 UART, 主要用于调试。
- UART1/2: 4 线 UART。
- UART3/4/5: 2 线 UART。



须知

当使用 UART 功能时，需要在 UART 初始化前对 RXD 管脚配置为上拉使能。具体的配置寄存器请参考 Hi35xxVxx_PINOUT_CN.xlsx (Hi35xxVxx 表示具体项目名称)。

例如：当使用系统的 UART0 时，需要配置 iocfg_reg121 的 bit[8]为高。

13.2.2 特点

UART 模块有以下特点：

- 支持 256 x 8bit 的发送 FIFO 和 256 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。

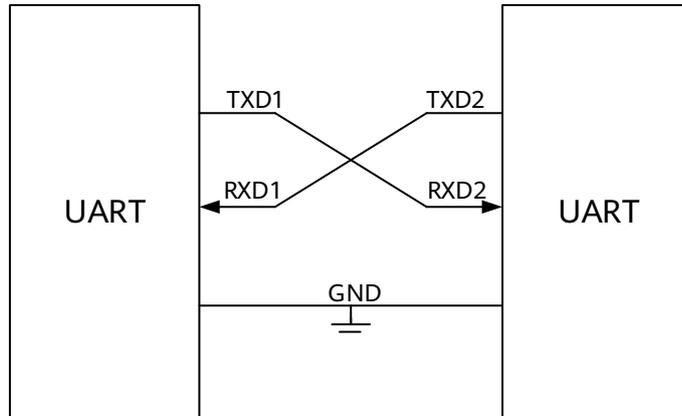
13.2.3 功能描述

应用框图

UART 的典型应用框图如图 13-7 所示。



图13-7 UART 的典型应用框图

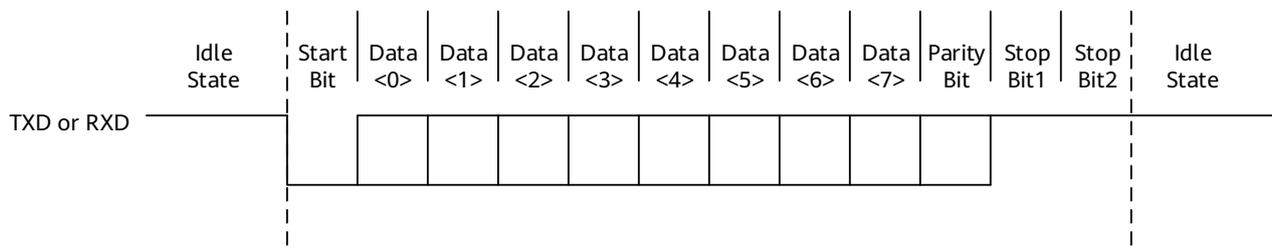


UART 是一种异步双向串行总线，它提供了一种简单有效的数据传输方式，只需要两根数据线互相对接。

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 13-8 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13-8 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。

13.2.4 工作方式

13.2.4.1 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 (100MHz、50MHz、24MHz 或 3MHz) / (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 $24 / (16 \times 30) = 0.05\text{Mbit/s}$ 。

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 $m = \text{integer} (FBRD \times 2^n + 0.5)$ ($n = \text{UART_FBRD}$ 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。



13.2.4.2 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。

13.2.4.3 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。

13.2.4.4 DMA 方式下的数据传输

初始化

初始化步骤如下：

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

步骤 2 写相应的配置值到 UART_IBRD、UART_FBRD 寄存器，配置传输速率。

步骤 3 配置 UART_CR、UART_LCR_H，设定相应的 UART 工作模式。

步骤 4 配置 UART_IFLS 设定相应的发送及接收 FIFO 阈值。

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

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

----**结束**

数据发送

数据发送（以 DMA 模式为例）步骤如下：

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

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

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

----**结束**



数据接收

数据接收（以 DMA 模式为例）步骤如下：

- 步骤 1 配置 DMA 数据通道，包括数据传输源和目的地址、数据接收区地址、数据传输个数、传输类型等参数。
- 步骤 2 配置 UART_DMCR 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

13.2.5 UART 寄存器概览

地址空间内：

- UART0 寄存器基地址：0x0_1104_0000。
- UART1 寄存器基地址：0x0_1104_1000。
- UART2 寄存器基地址：0x0_1104_2000。
- UART3 寄存器基地址：0x0_1104_3000。
- UART4 寄存器基地址：0x0_1104_4000。
- UART5 寄存器基地址：0x0_1104_5000。

UART 寄存器概览如表 13-9 所示。

表13-9 UART 寄存器概览

偏移地址	名称	描述	页码
0x000	UART_DR	数据寄存器	13-39
0x004	UART_RSR	接收状态寄存器/错误清除寄存器	13-40
0x018	UART_FR	标志寄存器	13-41
0x024	UART_IBRD	整数波特率寄存器	13-42
0x028	UART_FBRD	小数波特率寄存器	13-43
0x02C	UART_LCR_H	线控寄存器	13-43



偏移地址	名称	描述	页码
0x030	UART_CR	控制寄存器	13-45
0x034	UART_IFLS	中断 FIFO 阈值选择寄存器	13-47
0x038	UART_IMSC	中断屏蔽寄存器	13-48
0x03C	UART_RIS	原始中断状态寄存器	13-49
0x040	UART_MIS	屏蔽后中断状态寄存器	13-50
0x044	UART_ICR	中断清除寄存器	13-51
0x048	UART_DMACR	DMA 控制寄存器	13-52

13.2.6 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]	-	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。	0x0



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

UART_RSR

UART_RSR 为接收状态寄存器/错误清除寄存器。

- 寄存器读时作为接收状态寄存器。
- 寄存器写时作为错误清除寄存器。

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

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

Offset Address: 0x004 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:4]	-	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;	0x0



Bits	Access	Name	Description	Reset
			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。	
[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(有效的停止位为 1)。	0x0

UART_FR

UART_FR 为 UART 标志寄存器。

Offset Address: 0x018 Total Reset Value: 0x0096

Bits	Access	Name	Description	Reset
[15:8]	-	reserved	保留。	0x00
[7]	RO	txfe	该位的含义由 UART_LCR_H[fen]的状态决定。 <ul style="list-style-type: none"> 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。 <ul style="list-style-type: none"> 如果 UART_LCR_H[fen]为 0, 则当接收 	0x0



Bits	Access	Name	Description	Reset
			holding register 满时该位置 1; <ul style="list-style-type: none"> 如果 UART_LCR_H[fen]为 1, 则当接收 FIFO 为满时该位置 1。 	
[5]	RO	txff	该位的含义由 UART_LCR_H[fen]的状态决定。 <ul style="list-style-type: none"> 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。 <ul style="list-style-type: none"> 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。 一旦发送 FIFO 非空该位就置位, 不管 UART 使能与否。	0x0
[2:0]	-	reserved	保留。	0x6

UART_IBRD

UART_IBRD 为整数波特率寄存器。

Offset Address: 0x024 Total Reset Value: 0x0000



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

UART_FBRD

UART_FBRD 为小数波特率寄存器。

须知

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

Offset Address: 0x028 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:6]	-	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]	-	reserved	保留。	0x00
[7]	RW	sps	stick parity 选择。 <ul style="list-style-type: none"> • 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 	0x0



Bits	Access	Name	Description	Reset
			<p>测;</p> <ul style="list-style-type: none"> • 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。 <p>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。。</p>	
[6:5]	RW	wlen	<p>指示发送和接收一个帧里数据比特的数目。</p> <p>00: 5bit; 01: 6bit; 10: 7bit; 11: 8bit。</p>	0x0
[4]	RW	fen	<p>发送和接收 FIFO 使能控制。</p> <p>0: 不使能发送和接收 FIFO; 1: 发送和接收 FIFO 使能。</p>	0x0
[3]	RW	stp2	<p>发送帧尾 2bit 停止位判断。</p> <p>0: 发送的帧尾没有 2bit 停止位; 1: 发送的帧尾有 2bit 停止位。</p> <p>接收逻辑在接收时不检查 2bit 的停止位。</p>	0x0
[2]	RW	eps	<p>发送和接收过程中的奇偶校验选择。</p> <p>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。</p> <p>当 UART_LCR_H[pen]为 0 时, 该位不起作用。</p>	0x0



Bits	Access	Name	Description	Reset
[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。 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。	0x0

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;	0x0



Bits	Access	Name	Description	Reset
			1: 使能 CTS 硬件流控, 只有当 nUARTCTS 信号有效时才发送数据。	
[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]	-	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; 1: 即该 bit 配置为 1, 则输出信号为 0。	0x0
[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	rxen	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	txen	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。	0x0



Bits	Access	Name	Description	Reset
			0: 禁止; 1: UARTTXD 输出环回到 UARTRXD。	
[6:1]	-	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]	-	reserved	保留。	0x000
[5:3]	RW	rxiflssel	接收中断 FIFO 的阈值选择, 接收中断的触发点如下。 000: 接收 FIFO ≥ 1/8full; 001: 接收 FIFO ≥ 1/4full; 010: 接收 FIFO ≥ 1/2full; 011: 接收 FIFO ≥ 3/4full; 100: 接收 FIFO ≥ 7/8full; 101: 接收 FIFO ≥ 1/16full 110: 接收 FIFO ≥ 1/32full 111: 保留。	0x2
[2:0]	RW	txiflssel	发送中断 FIFO 的阈值选择, 发送中断的触	0x2



Bits	Access	Name	Description	Reset
			发点如下。 000: 发送 FIFO≤1/8full; 001: 发送 FIFO≤1/4full; 010: 发送 FIFO≤1/2full; 011: 发送 FIFO≤3/4full; 100: 发送 FIFO≤7/8full; 101: 发送 FIFO≤15/16full; 110: 发送 FIFO≤31/32full; 111: 保留。	

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]	-	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;	0x0



Bits	Access	Name	Description	Reset
			1: 不屏蔽该中断。	
[6]	RW	rtim	接收超时中断的屏蔽状态。 0: 屏蔽该中断; 1: 不屏蔽该中断。	0x0
[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]	-	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]	-	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



Bits	Access	Name	Description	Reset
[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; 1: 已产生中断。	0x0
[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]	-	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]	-	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



Bits	Access	Name	Description	Reset
[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; 1: 已产生中断。	0x0
[3:0]	-	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]	-	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。	0x0



Bits	Access	Name	Description	Reset
			0: 无效; 1: 清除中断。	
[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]	-	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
[15:4]	-	reserved	保留。	0x0000
[3]	RW	rxlastsreq_en	UART RX DMA 支持最后一笔数据的 REQ 使能。使能打开后，传输最后一笔数据时	0x0



Bits	Access	Name	Description	Reset
			会发出 LASTREQ 给 DMA 控制器。 0: 禁止; 1: 使能。	
[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

13.3 SPI

13.3.1 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 SPI 接口、TI 串行同步接口和 National Semiconductor MicroWire 接口三种外设接口协议。

13.3.2 特点



须知

- 芯片总共有 4 组 SPI 接口。
- 工作参考时钟为 100MHz。其中，SPI0 主模式输出的 SPI_CLK 最大支持 50MHz，SPI1/2/3 主模式输出的 SPI_CLK 最大支持 25MHz，SPI0/1/2/3 从模式输入的 SPI_CLK 最大支持 8.25MHz。

SPI 的功能特点有：

- 接口时钟频率可编程。
- 4 组 SPI 接口都支持主模式和从模式。
- SPI0/1/2/3 只支持单片选，片选信号支持极性可配。
- 接收 FIFO 的宽度为 16bit、深度为 256。
- 发送 FIFO 的宽度为 16bit、深度为 256。
- 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。

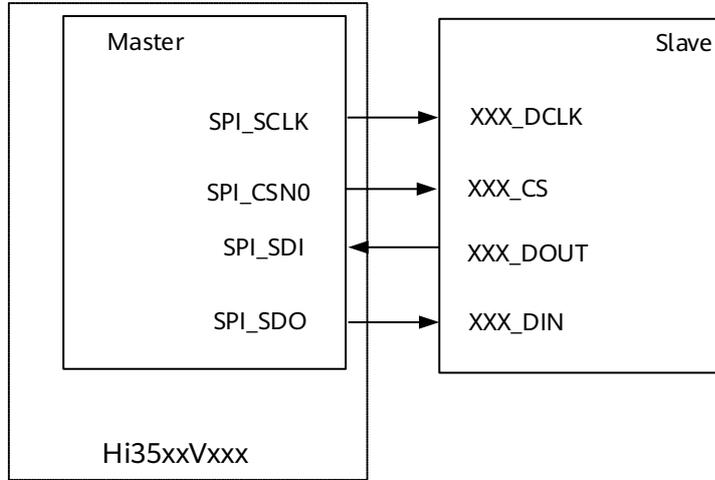
13.3.3 功能描述

典型应用

SPI 接 Slave 时的应用框图如图 13-9 所示。



图13-9 SPI 接 Slave 时的应用



13.3.4 三种外设总线时序

图 13-10 ~ 图 13-17 中的缩略语含义：

- MSB: Most Significant Bit
- LSB: Least Significant Bit
- Q: Q is an undefined signal

Motorola SPI 接口

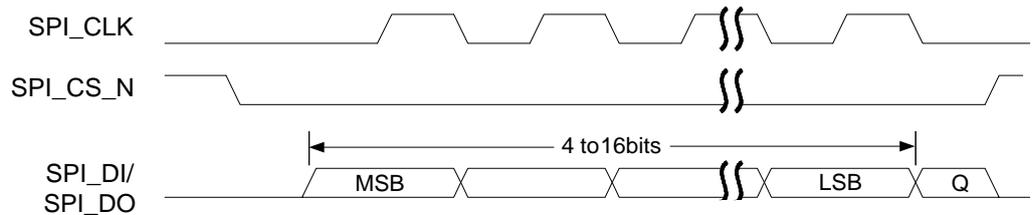
说明

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

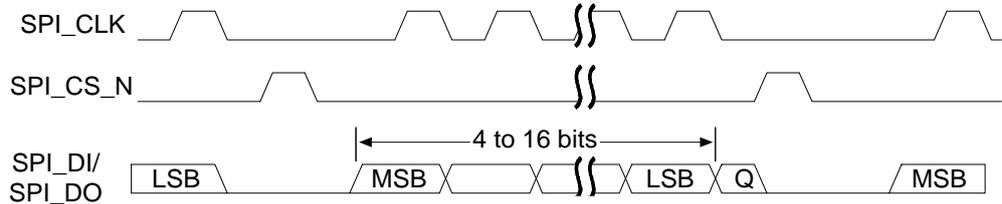
SPI 单帧格式如图 13-10 所示。

图13-10 SPI 单帧格式 (SPO=0、SPH=0)



SPI 连续帧格式如图 13-11 所示。

图13-11 SPI 连续帧格式 (SPO=0、SPH=0)



在该模式下，当 SPI 处于空闲状态时：

- SPI_CLK 信号设置为低
- SPI_CS_N 信号设置为高
- 发送数据线 SPI_DO 强制为低

当 SPI 处于使能状态，而且发送 FIFO 内存在有效数据时，设置 SPI_CS_N 信号为低，表示开始传输数据。来自 Slave 的数据立刻发送到 Master 的接收数据线 SPI_DI。半个 SPI_CLK 时钟周期之后，有效的 Master 数据传输到 SPI_DO。此时 Master 和 Slave 数据都已经有效，SPI_CLK 管脚在接下来的半个 SPI_CLK 时钟周期之后变为高电平。数据在 SPI_CLK 时钟的上升沿被捕获，在时钟的下降沿被传送。

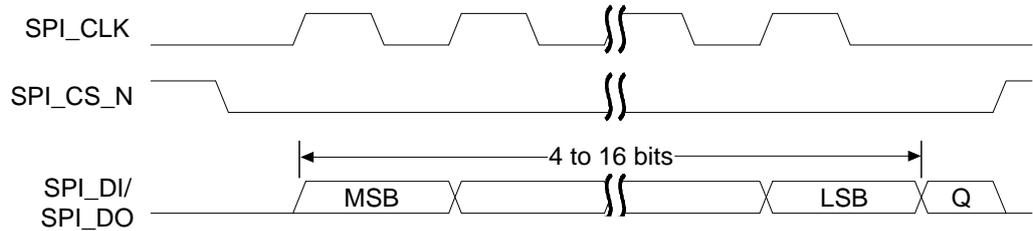
如果传输单个 word，当捕捉到最后 1bit 数据时，SPI_CS_N 在接下来的 1 个 SPI_CLK 时钟周期之后恢复为高电平。

如果是连续的传输，SPI_CS_N 信号在每个 word 传输之间必须将 SPI_CLK 时钟拉高一个时钟周期。这是因为 SPH 为 0 时，Slave 选择管脚会固定其内部串行设备寄存器的数据，使它不会变化。因此在连续传输时，主设备必须在每个 word 传输之间将 SPI_CS_N 信号拉高。连续传输结束时，SPI_CS_N 在捕捉到最后 1bit 之后的 1 个 SPI_CLK 时钟周期之后恢复为高电平。

2. SPO=0、SPH=1

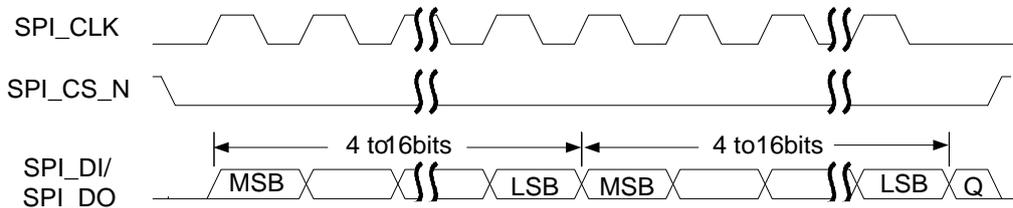
SPI 单帧格式如图 13-12 所示。

图13-12 SPI 单帧格式 (SPO=0、SPH=1)



SPI 连续帧格式如图 13-13 所示。

图13-13 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。

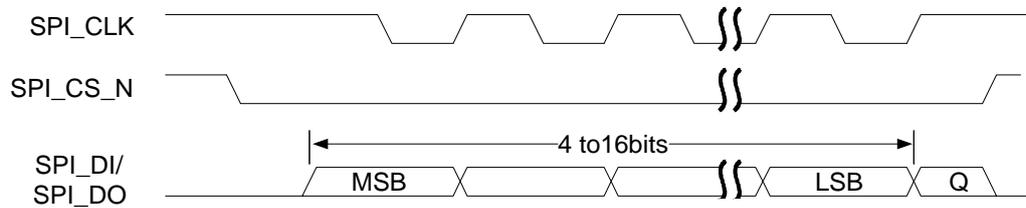
如果传输单个 word，当捕捉到最后 1bit 数据时，SPI_CS_N 在接下来的 1 个 SPI_CLK 时钟之后恢复为高电平。

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

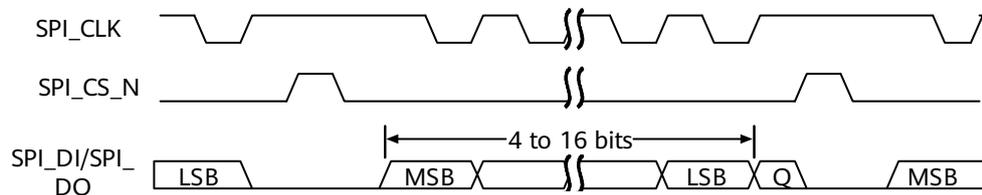
SPI 单帧格式如图 13-14 所示。

图13-14 SPI 单帧格式 (SPO=1、SPH=0)



SPI 连续帧格式如图 13-15 所示。

图13-15 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。

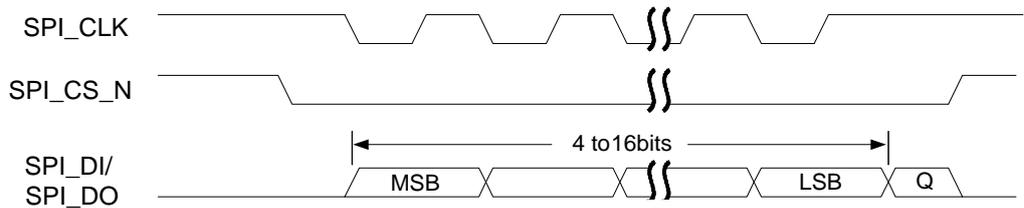
如果是连续的传输，SPI_CS_N 信号在每个 word 传输之间必须拉高。这是因为当 SPH 为 0 时，Slave 选择管脚固定其内部串行设备寄存器的数据，使它不会变化。

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

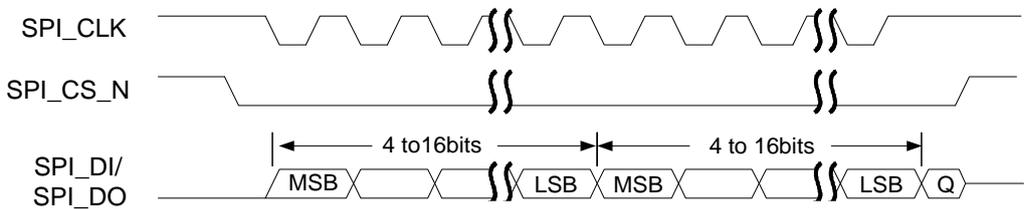
SPI 单帧格式如图 13-16 所示。

图13-16 SPI 单帧格式 (SPO=1、SPH=1)



SPI 连续帧格式如图 13-17 所示。

图13-17 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

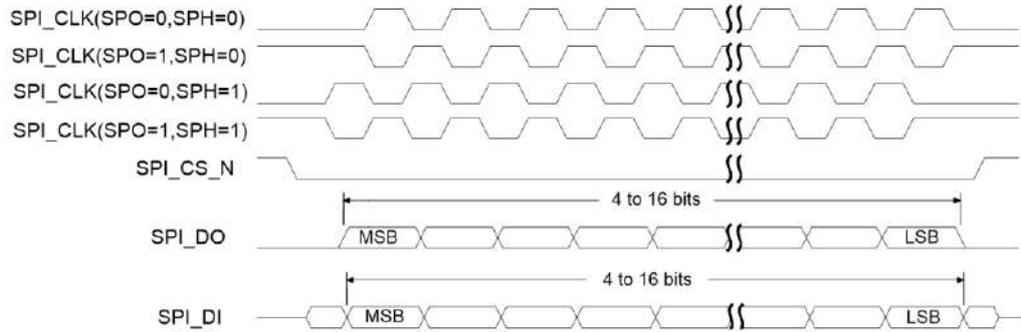
当 SPI 为使能状态，而且发送 FIFO 内有有效数据时，设置 SPI_CS_N Master 信号为低表示开始传输数据。半个 SPI_CLK 时钟周期后，Master 和 Slave 数据在各自的传输线上有效。同时，时钟 SPI_CLK 从 1 个下降沿开始有效。数据在 SPI_CLK 时钟的上升沿被捕获，在时钟的下降沿被传送。当传输单个 word 时，SPI_CS_N 在传输的最后 1bit 捕获之后的 1 个 SPI_CLK 时钟周期之后恢复为高电平。

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



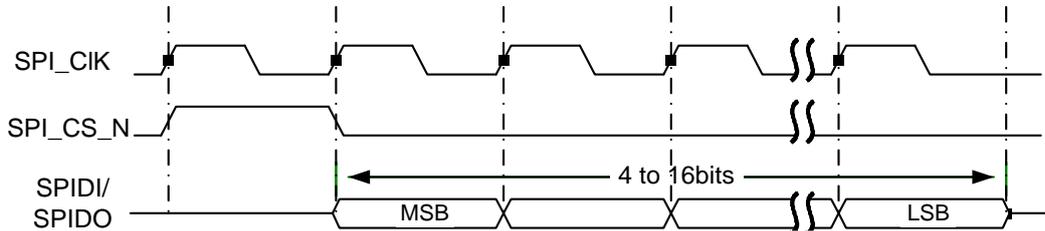
图13-18 SPI 接口时序图



TI 同步串行接口

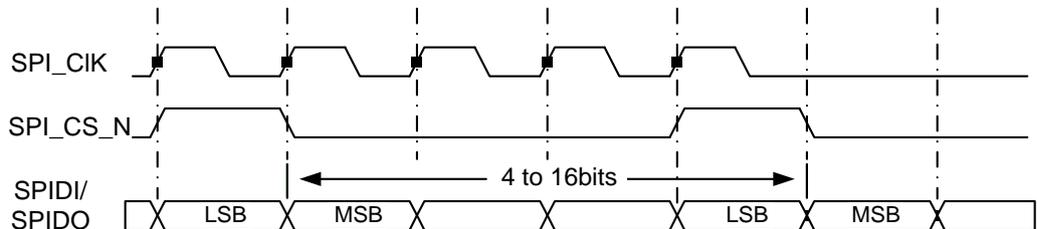
TI 同步串行单帧格式如图 13-19 所示。

图13-19 TI 同步串行单帧格式



TI 同步串行连续帧格式如图 13-20 所示。

图13-20 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_CLK 为低电平。
- SPI_CS_N 为低电平。
- 传输数据线 SPIDO 保持为高阻。

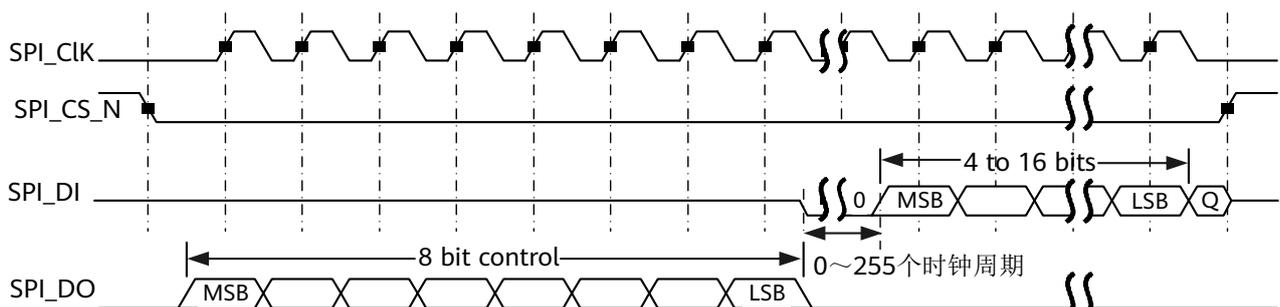
一旦发送 FIFO 有数据，SPICSN 就会产生一个 SPICK 时钟周期的高电平脉冲，将被发送的数据就会从发送 FIFO 传送到发送逻辑串行移位寄存器。在 SPICK 时钟的下一个上升沿，4bit ~ 16bit 数据帧的 MSB 就会从 SPIDO 移位输出。同样，从外部串行 slave 设备接收数据的 MSB 会从 SPIDI 管脚移位输入。

SPI 和片外串行设备在 SPICK 时钟的下降沿将数据存入串行移位寄存器。接收串行寄存器在接收到 LSB 之后的第一个 SPICK 时钟上升沿将数据送给接收 FIFO。

National Semiconductor Microwire 接口

National Semiconductor Microwire 单帧格式如图 13-21 所示。

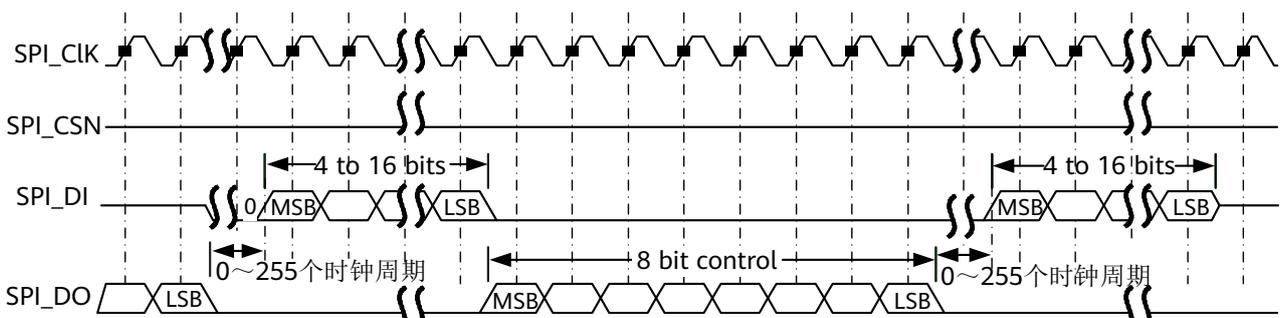
图13-21 National Semiconductor Microwire 单帧格式



在 SPIDO LSB 结束和 SPIDI MSB 开始之间可以延迟 0 ~ 255 个时钟周期。

National Semiconductor Microwire 连续帧格式如图 13-22 所示。

图13-22 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。

13.3.5 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：

$$F_{\text{sspclkout}} = F_{\text{sspclk}} / (\text{CPSDVSR} \times (1 + \text{SCR}))$$

F_{sspclk} : SPI 的工作参考时钟 100MHz。

CPSDVSR、SCR 请查询 SPICPSR、SPICR0。

支持独立软复位和时钟门控，具体请参见 3.2.6 CRG 寄存器描述章节。

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 SPITXFIFO CR 和 SPIRXFIFO CR。

步骤 6 DMA 方式下，配置寄存器 SPIDMACR，使能 SPI 的 DMA 功能。

----结束

查询方式下的数据传输

具体步骤如下：

步骤 1 向寄存器 SPICR1 [sse]写“1”，使能 SPI。

步骤 2 将需发送的数据连续写到寄存器 SPIDR。

步骤 3 轮询寄存器 SPISR，直到[BSY]=0（表示总线不忙）、[TFE]=1（表示发送 FIFO 已空）、[RNE]=1（表示接收 FIFO 非空）。

步骤 4 读出数据，需保证读空接收 FIFO（可通过查询 SPISR [RNE]得到）。

须知

Motorola SPI/TI 同步串行接口的全双工特性，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 6 向寄存器 SPICR1 [sse]写“1”，使能 SPI。

步骤 7 将需发送的数据连续写到寄存器 SPIDR。

步骤 8 等待中断 SPIRXINTR，读出数据。循环直到读出所有数据。



须知

Motorola SPI/TI 同步串行接口的全双工特性，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 9 向寄存器 SPICR1 [sse]写“0”，禁止 SPI。

----结束

DMA 方式下的数据传输

具体步骤如下：

步骤 1 获取一个 DMAC 通道。

步骤 2 向寄存器 SPICR1 [sse]写“1”，使能 SPI。

步骤 3 发送数据

1. 配置该 DMAC 通道的配置寄存器和控制寄存器中的相关参数。
2. 启动 DMAC，响应 SPI 发送 FIFO 的 DMA 请求进行数据传输。
3. 通过 DMA 中断上报，判断数据是否发送完成，如果完成则关闭 SPI 的 DMA 功能。

步骤 4 接收数据

1. 配置该 DMAC 通道的配置寄存器和控制寄存器中的相关参数。
2. 启动 DMAC，响应 SPI 接收 FIFO 的 DMA 请求进行数据传输。
3. 通过 DMA 中断上报，判断数据是否接收完成，如果完成则关闭 SPI 的 DMA 功能。

步骤 5 向寄存器 SPICR1 [sse]写“0”，禁止 SPI。

----结束

13.3.6 SPI 寄存器概览

SPI 寄存器概览如表 13-10 所示。

- SPI0 寄存器基地址：0x0_1107_0000。



- SPI1 寄存器基地址: 0x0_1107_1000。
- SPI2 寄存器基地址: 0x0_1107_2000。
- SPI3 寄存器基地址: 0x0_1107_3000。

表13-10 SPI 寄存器概览

偏移地址	名称	描述	页码
0x000	SPICR0	控制寄存器 0	13-66
0x004	SPICR1	控制寄存器 1	13-68
0x008	SPIDR	数据寄存器	13-69
0x00C	SPISR	状态寄存器	13-69
0x010	SPICPSR	时钟分频寄存器	13-70
0x014	SPIIMSC	中断屏蔽寄存器	13-70
0x018	SPIRIS	原始中断状态寄存器	13-71
0x01C	SPIMIS	屏蔽后中断状态寄存器	13-72
0x020	SPIICR	中断清除寄存器	13-72
0x024	SPIDMACR	DMA 控制寄存器	13-72
0x028	SPITXFIFO CR	发送 FIFO 控制寄存器	13-73
0x02C	SPIRXFIFO CR	接收 FIFO 控制寄存器	13-74

13.3.7 SPI 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	0x00



Bits	Access	Name	Description	Reset
			式为 $Fsspclk / (CPSDVSR \times (1+SCR))$ 。 CPSDVSR 是一个 2 ~ 254 之间的偶数，由寄存器 SPICPSR 配置。	
[7]	RW	SPH	SPICLKOUT 相位，具体含义请参见“13.3.4 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“13.3.4 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; 0100: 5bit; 0101: 6bit; 0110: 7bit; 0111: 8bit; 1000: 9bit; 1001: 10bit; 1010: 11bit; 1011: 12bit; 1100: 13bit; 1101: 14bit; 1110: 15bit; 1111: 16bit; 其他: 保留。	0x0



SPICR1

SPICR1 为控制寄存器 1。

Offset Address: 0x004 Total Reset Value: 0x7F00

Bits	Access	Name	Description	Reset
[15]	RW	WaitEn	等待使能，当 SPICR0 寄存器的 FRF 配置为 National Microwire 帧格式时有效。 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]	-	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]	-	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]	-	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; 1: Slave 模式。	0x0
[1]	RW	SSE	设置 SPI 使能。	0x0



Bits	Access	Name	Description	Reset
			0: 不使能; 1: 使能。	
[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

SPI SR

SPI SR 为状态寄存器。

Offset Address: 0x00C Total Reset Value: 0x0003

Bits	Access	Name	Description	Reset
[15:5]	-	reserved	保留。	0x000
[4]	RO	BSY	SPI 忙标记。 0: 空闲; 1: 忙。	0x0



Bits	Access	Name	Description	Reset
[3]	RO	RFF	接收 FIFO 是否已满。 0: 未满; 1: 已满。	0x0
[2]	RO	RNE	接收 FIFO 是否未空。 0: 已空; 1: 未空。	0x0
[1]	RO	TNF	发送 FIFO 是否未空。 0: 已空; 1: 未空。	0x1
[0]	RO	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]	-	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]	-	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
[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]	-	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]	-	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]	-	reserved	保留。	0x0000
[1]	WC	RTIC	清除接收超时中断。	0x0
[0]	WC	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]	-	reserved	保留位。	0x0000
[1]	RW	TXDMAE	DMA 发送 FIFO 使能寄存器。 0: 禁止;	0x0



Bits	Access	Name	Description	Reset
			1: 使能。	
[0]	RW	RXDMAE	DMA 接收 FIFO 使能寄存器。 0: 禁止; 1: 使能。	0x0

SPITXFIFOCR

SPITXFIFOCR 为发送 FIFO 控制寄存器。

Offset Address: 0x028 Total Reset Value: 0x0009

Bits	Access	Name	Description	Reset
[15:6]	-	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; 011: 16; 100: 32; 101: 64; 110: 128; 111: 128。	0x1
[2:0]	RW	DMATXBRSIZE	配置发送 FIFO 请求 DMA 进行 burst 传输的水线。即，发送 FIFO 中数据数目小于或等于 (256-DMATXBRSIZE) 所配置的字数时，DMATXBREQ 有效，此处字长是 16 位。 000: 1; 001: 4;	0x1



Bits	Access	Name	Description	Reset
			010: 8; 011: 16; 100: 32; 101: 64; 110: 64; 111: 64。	

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]	-	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; 111: 192。	0x1
[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。	0x1



Bits	Access	Name	Description	Reset
			000: 1; 001: 4; 010: 8; 011: 16; 100: 32; 101: 64; 110: 96; 111: 128。	

13.4 SPI_3WIRE

13.4.1 概述

SPI_3WIRE 控制器提供 1 个三线 SPI 接口，作为 Master 与外部设备进行同步串行通信。

13.4.2 功能描述

SPI_3WIRE 控制器具有以下功能特点：

- 最大可支持 10MHz。
- 仅支持 Master 模式，即片选信号和接口时钟只输出。

13.4.3 工作方式

SPI_3WIRE 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 $(100\text{MHz}/1\text{MHz})/2 - 1 = 49$ ，所以配置 spi_clk_div 为 0x31。

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

13.4.4 SPI_3WIRE 寄存器概览

SPI_3WIRE 寄存器概览如表 13-11 所示。

表13-11 SPI_3WIRE 寄存器概览 (基地址 0x0_1107_5000)

偏移地址	名称	描述	页码
0x0000	SPI_3WIRE_COEF0	SPI_3WIRE 配置寄存器 0	13-76
0x0004	SPI_3WIRE_COEF1	SPI_3WIRE 配置寄存器 1	13-77
0x0008	SPI_3WIRE_COEF2	SPI_3WIRE 配置寄存器 2	13-77

13.4.5 SPI_3WIRE 寄存器描述

SPI_3WIRE_COEF0

SPI_3WIRE_COEF0 为 SPI_3WIRE 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 SPI_3WIRE 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 SPI_3WIRE 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;	0x0



Bits	Access	Name	Description	Reset
			0: SPI 总线处于 IDLE。	
[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

13.5 SPI_TFT

13.5.1 概述

SPI_TFT 控制器提供 1 个三线 SPI 接口，作为 Master 实现与对接的 LCD 小屏芯片的同步串行通信。

13.5.2 功能描述

SPI_TFT 控制器具有以下功能特点：

- 工作参考时钟有 24MHz 和 100MHz 两档可选，默认选择 100MHz。接口时钟 SPI_TFT_SCL 由工作参考时钟分频产生，分频系数可编程，范围 10~4000，最大可支持输出 10MHz。
- 仅支持 Master 模式，即片选信号和接口时钟只输出。
- 接口时钟极性、相位可配。
- 接口时钟的空闲态可配。
- 仅支持单片选，片选信号支持极性可配。
- 接收 FIFO 的位宽为 32bit、深度为 32。
- 发送 FIFO 的位宽为 32bit、深度为 32。
- 串行数据帧长度可编程，读数据位宽：1-32bit，写命令或数据位宽：1-19bit。
- 读操作的 dummy clock cycle 长度可配。



13.5.3 接口总线时序

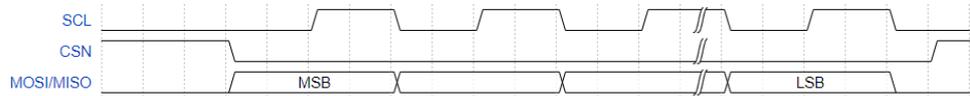
说明

SPO 表示 SPI_TFT_SCL 极性，SPH 表示 SPI_TFT_SCL 相位。SPO、SPH 是寄存器 SPICR bit[2:1]。

1. SPO=0、SPH=0

此处仅给出该模式下，cs_po=0，scl_static_state=0 配置下的帧格式。此时，SCL 空闲态为低电平，且 master 在 SCL 的上升沿采样，在 SCL 的下降沿发送。

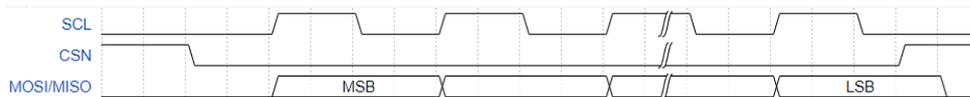
图13-23 SPI_TFT 帧格式



2. SPO=0、SPH=1

此处仅给出该模式下，cs_po=0，scl_static_state=0 配置下的帧格式。此时，SCL 空闲态为低电平，且 master 在 SCL 的上升沿发送，在 SCL 的下降沿采样。

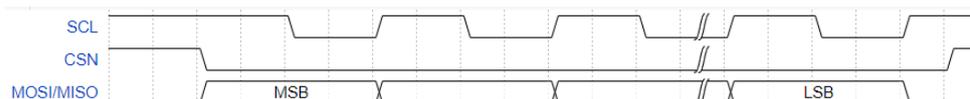
图13-24 SPI_TFT 帧格式



3. SPO=1、SPH=0

此处仅给出该模式下，cs_po=0，scl_static_state=1 配置下的帧格式。此时，SCL 空闲态为低电平，且 master 在 SCL 的下降沿采样，在 SCL 的上升沿发送。

图13-25 SPI_TFT 帧格式

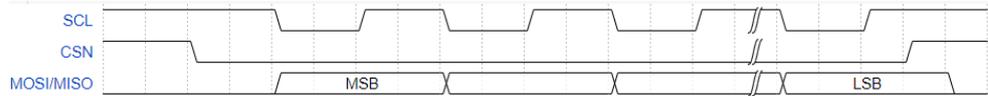


4. SPO=1、SPH=1



此处仅给出该模式下， $cs_po=0$ ， $scl_static_state=1$ 配置下的帧格式。此时，SCL 空闲态为低电平，且 master 在 SCL 的下降沿发送，在 SCL 的上升沿采样。

图13-26 SPI_TFT 帧格式



13.5.4 工作方式

13.5.4.1 初始化

- 步骤 1 向寄存器 SPICR[0]写“0”，禁止 SPI；
- 步骤 2 写寄存器 SPICR[1]、[2]、[3]、[15]和[31:24]，配置 SPI 接口参数；
- 步骤 3 写寄存器 SPICR[7:4]和 SPICPSR[7:0]，配置 SPI 接口时钟分频系数；
- 步骤 4 配置寄存器 SPIIMSC，根据需要屏蔽对应的中断请求。

----结束

13.5.4.2 写操作

- 步骤 1 向寄存器 SPICR[0]写“1”，使能 SPI；
- 步骤 2 写寄存器 SPICR[13:8]和[14]，配置写数据位宽和读写类型；
- 步骤 3 将需要发送的写命令按 slave 要求的数据包格式拼接好写入寄存器 SPIDR；
- 步骤 4 发送 FIFO 中的数据传输完成后，若要继续传输数据，可根据需求重复 13.5.4.1 “初始化”小节的步骤 2 来配置 SPI 接口，然后重复**写操作**的步骤 2、步骤 3。

----结束

13.5.4.3 读操作

- 步骤 1 向寄存器 SPICR[0]写“1”，使能 SPI；
- 步骤 2 写寄存器 SPICR[13:8]、[21:16]和[14]，配置读/写数据位宽(读/写操作数据包的位宽)和读写类型(0 表示写，1 表示读)；
- 步骤 3 将需要发送的读命令按 slave 要求的数据包格式拼接好，写入寄存器 SPIDR；



步骤 4 待 SPI 接口数据传输完成后，即当 SPI_TFT 状态寄存器 SPIDR[0]=1 且 SPIDR[4]=0，CPU 可通过访问寄存器 SPIDR，回读 slave 返回的读数据；

步骤 5 若需要继续传输数据，可根据需求重复 13.5.4.1 “初始化”小节的步骤 2 来配置 SPI 接口，然后重复读操作的步骤 2、步骤 3、步骤 4。

----结束

📖 说明

需要待写入 SPIDR 寄存器中的数据传输完成后才可更改 SPI 接口配置。

13.5.5 SPI_TFT 寄存器概览

SPI_TFT 寄存器概览如表 13-12 所示。

表13-12 SPI_TFT 寄存器概览 (基地址 0x0_1107_4000)

偏移地址	名称	描述	页码
0x0000	SPICR	SPI_TFT 控制寄存器	13-82
0x0004	SPIDR	SPI_TFT 数据寄存器	13-69
0x0008	SPISR	SPI_TFT 状态寄存器	13-69
0x000C	SPICPSR	SPI_TFT 时钟分频寄存器	13-70
0x0010	SPIIMSC	SPI_TFT 中断屏蔽寄存器	13-70
0x0014	SPIRIS	SPI_TFT 原始中断状态寄存器	13-71
0x0018	SPIMIS	SPI_TFT 屏蔽后中断状态寄存器	13-72
0x001C	SPIICR	SPI_TFT 中断清除寄存器	13-72
0x0020	SPITXFIFO CR	SPI_TFT 发送 FIFO 控制寄存器	13-73
0x0024	SPIRXFIFO CR	SPI_TFT 接收 FIFO 控制寄存器	13-74



13.5.6 SPI_TFT 寄存器描述

SPICR

SPICR 为 SPI_TFT 控制寄存器。

Offset Address: 0x0000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	RW	dummy_clock_cycle	Dummy clock cycle 长度, 以 SCL 时钟周期为单位。	0x00
[23:22]	-	reserved	保留。	0x0
[21:16]	RW	rd_width	读数据位宽。 0x01: 1bit; 0x02: 2bit; 0x03: 3bit; 0x04: 4bit; 0x05: 5bit; 0x1c: 28bit; 0x1d: 29bit; 0x1e: 30bit; 0x1f: 31bit; 0x20: 32bit。 其他: 保留。	0x00
[15]	RW	scl_static_state	接口时钟 SPI_TFT_SCL 空闲状态。 0: 表示 SPI 接口空闲时, SPI_TFT_SCL 固定为低电平; 1: 表示 SPI 接口空闲时, SPI_TFT_SCL 固定为高电平。	0x0
[14]	RW	write	SPI 写使能。 0: 写操作使能; 1: 读操作使能。	0x0
[13:8]	RW	wr_width	SPI 写命令或数据位宽(包含地址、读/写或数据/命令标	0x00



Bits	Access	Name	Description	Reset
			<p>志、写数据)。 0x01: 1bit; 0x02: 2bit; 0x03: 3bit; 0x04: 4bit; 0x10: 16bit; 0x11: 17bit; 0x12: 18bit; 0x13: 19bit。 其他: 保留。</p>	
[7:4]	RW	scr	<p>串行时钟率, SCR 的值用来产生 SPI 发送和接收的比特率。 公式为 $FSSPCLK/(2 \times (CPSDVSr+1) \times (1+SCR))$。 CPSDVSr 由寄存器 SPICPSR 配置。</p>	0x0
[3]	RW	cs_po	<p>SPI 片选极性。 0: 低电平有效; 1: 高电平有效。</p>	0x0
[2]	RW	sph	<p>接口时钟 SPI_TFT_SCL 相位。 0: 在 SPI_TFT_SCL 的第一个时钟边沿采样数据, 第二个边沿发送数据; 1: 在 SPI_TFT_SCL 的第二个时钟边沿采样数据, 第一个边沿发送数据。</p>	0x0
[1]	RW	spo	<p>接口时钟 SPI_TFT_SCL 极性。 0: SPI_TFT_SCL 高电平有效; 1: SPI_TFT_SCL 低电平有效。</p>	0x0
[0]	RW	sse	<p>设置 SPI 使能。 0: 禁止 SPI; 1: 使能 SPI。</p>	0x0



SPI DR

SPI DR 为 SPI_TFT 数据寄存器。

Offset Address: 0x0004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	data	数据寄存器。 如果数据比特数少于 32 则必须右对齐。对该寄存器读时将忽略高位未使用的比特位，写时则自动将数据右对齐。	0x0000 0000

SPI SR

SPI SR 为 SPI_TFT 状态寄存器。

Offset Address: 0x0008 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:16]	RO	rx fifodata num	表示接收 FIFO 中数据个数，接收 FIFO 深度为 32。	0x00
[15:8]	RO	tx fifodata num	表示发送 FIFO 中数据个数，发送 FIFO 深度为 32。	0x00
[7:5]	-	reserved	保留。	0x0
[4]	RO	bsy	SPI 忙标记。 0: 空闲; 1: 忙。	0x0
[3]	RO	rff	接收 FIFO 是否已满。 0: 未滿; 1: 已滿。	0x0
[2]	RO	rne	接收 FIFO 是否未空。 0: 已空; 1: 未空。	0x0



Bits	Access	Name	Description	Reset
[1]	RO	tnf	发送 FIFO 是否未。 0: 已。 1: 未。	0x1
[0]	RO	tfe	发送 FIFO 是否已。 0: 未。 1: 已。	0x1

SPICPSR

SPICPSR 为 SPI_TFT 时钟分频寄存器。

Offset Address: 0x000c Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x0000
[7:0]	RW	cpsdvsr	时钟分频因子, 取值范围: [0,255]	0x00

SPIIMSC

SPIIMSC 为 SPI_TFT 中断屏蔽寄存器。

Offset Address: 0x0010 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x000000
[2]	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



Bits	Access	Name	Description	Reset
[1]	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
[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 SPI_TFT 原始中断状态寄存器。

Offset Address: 0x0014 Total Reset Value: 0x0000_0004

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x000000
[2]	RO	txris	发送 FIFO 中断的原始中断状态。	0x1
[1]	RO	rxris	接收 FIFO 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 SPI_TFT 屏蔽后中断状态寄存器。

Offset Address: 0x0018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x000000
[2]	RO	txmis	发送 FIFO 中断屏蔽后的状态。	0x0



Bits	Access	Name	Description	Reset
[1]	RO	rxmis	接收 FIFO 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 SPI_TFT 中断清除寄存器。

Offset Address: 0x001c Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	roric	清除接收溢出中断。	0x0

SPI TX FIFO CR

SPI TX FIFO CR 为 SPI_TFT 发送 FIFO 控制寄存器。

Offset Address: 0x0020 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x000000
[2:0]	RW	txinsize	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 TXINTSize 所配置的字数时, TXRIS 有效。 000: 1; 001: 4; 010: 8; 011: 16; 其他: 保留。	0x1

SPI RX FIFO CR

SPI RX FIFO CR 为 SPI_TFT 接收 FIFO 控制寄存器。



Offset Address: 0x0024 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	保留。	0x000000
[2:0]	RW	rxinsize	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(32-RXINTSize)所配置的字数时, RXRIS 有效。 000: 1; 001: 4; 010: 8; 011: 16; 其他: 保留。	0x1

13.6 eMMC/SDIO/SD Card 控制器

13.6.1 概述

eMMC/SDIO 控制器 (以下简称 MMC) 用于处理对 eMMC、SD Card 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。

13.6.2 特点

MMC 的功能特点有:

- 最大可支持 2TB 容量的 eMMC 器件。
- 支持 non-DMA、SDMA、ADMA2、ADMA3 的数据传输。
- 支持命令与数据的 CRC 生成与校验。
- 数据位宽支持 1bit、4bit、8bit, 可根据对接器件选择。
- 支持 1byte ~ 65535bytes 的块数据读写操作。
- SD 支持 Default Speed(DS)/High Speed(HS)/SDR12/SDR25/SDR50/SDR104 模式。
- SDIO 支持 Default Speed(DS)/High Speed(HS)/SDR12/SDR25/SDR50/SDR104 模式模式。



- eMMC 支持 DS/HS/HS200/HS400 模式，支持 boot 功能。
- 支持 Enhanced data strobe 功能。

13.6.3 功能描述

功能框图

芯片中 3 个 MMC 控制器对应的功能信号和管脚名，如表 13-13 所示。

表13-13 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
eMMC	EMMC_CLK	EMMC_CLK
	EMMC_CMD	EMMC_CMD
	EMMC_DATA0	EMMC_DATA0
	EMMC_DATA1	EMMC_DATA1
	EMMC_DATA2	EMMC_DATA2
	EMMC_DATA3	EMMC_DATA3
	EMMC_DATA4	SPI1_SCLK
	EMMC_DATA5	SPI1_SDO
	EMMC_DATA6	SPI1_SDI
	EMMC_DATA7	SPI1_CSN
	EMMC_DS	EMMC_DS
	EMMC_RST_N	EMMC_RST_N
SDIO0	SDIO0_CARD_DETECT	SDIO0_CARD_DETECT
	SDIO0_CARD_POWER_EN_N	SDIO0_CARD_POWER_EN_N
	SDIO0_CCMD	SDIO0_CCMD
	SDIO0_CDATA0	SDIO0_CDATA0
	SDIO0_CDATA1	SDIO0_CDATA1
	SDIO0_CDATA2	SDIO0_CDATA2
	SDIO0_CDATA3	SDIO0_CDATA3
	SDIO0_CCLK_OUT	SDIO0_CCLK_OUT

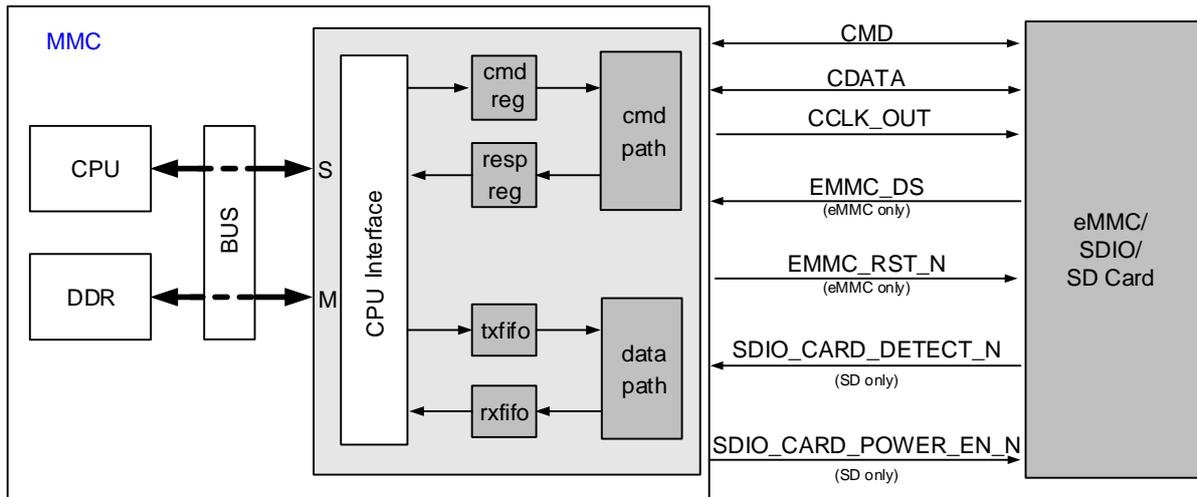


MMC 控制器	功能信号	管脚名
SDIO1	SDIO1_CARD_DETECT	SDIO1_CARD_DETECT
	SDIO1_CARD_POWER_EN_N	SDIO1_CARD_POWER_EN_N
	SDIO1_CCMD	SDIO1_CCMD
	SDIO1_CDATA0	SDIO1_CDATA0
	SDIO1_CDATA1	SDIO1_CDATA1
	SDIO1_CDATA2	SDIO1_CDATA2
	SDIO1_CDATA3	SDIO1_CDATA3
	SDIO1_CCLK_OUT	SDIO1_CCLK_OUT

- eMMC/SDIO/SD 支持符合以下协议的设备：
 - MultiMediaCard (eMMC-version 5.1)
 - Secure Digital I/O (SDIO version 3.0)
 - Secure Digital Memory (SD mem-version 3.0)

MMC 的功能框图如图 13-27 所示。

图13-27 MMC 功能框图



注：S：slave 接口；M：master 接口。



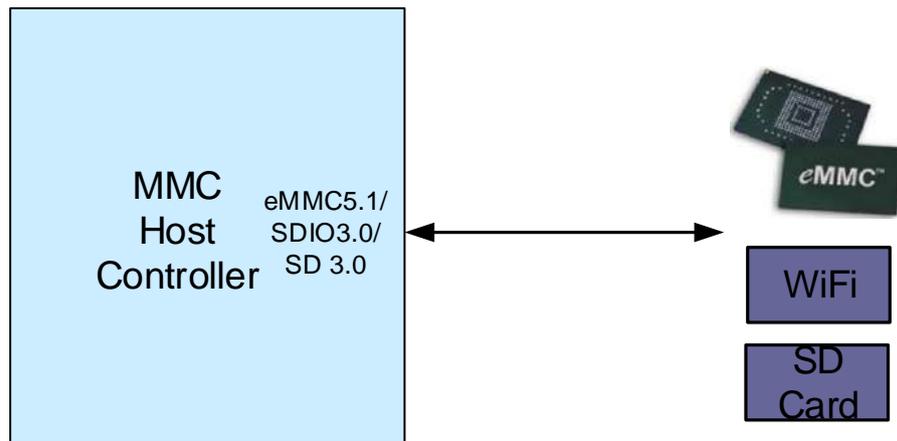
MMC 通过内部总线与系统连接，由以下单元构成：

- 命令通道
完成指令的发送与响应的接收。
- 数据通道
配合命令通道完成数据读写操作。
- 接口时钟控制单元
控制接口时钟的关闭与开启,以及接口方向时钟相位调节。

典型应用

MMC 的典型应用如图 13-28 所示。

图13-28 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13-29 MMC 指令格式

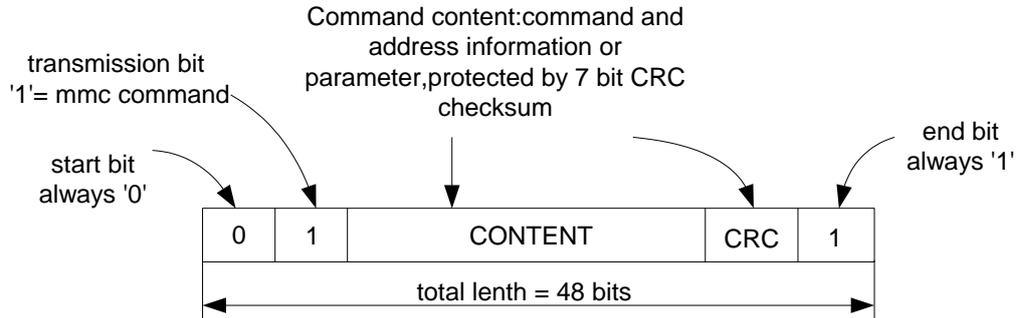
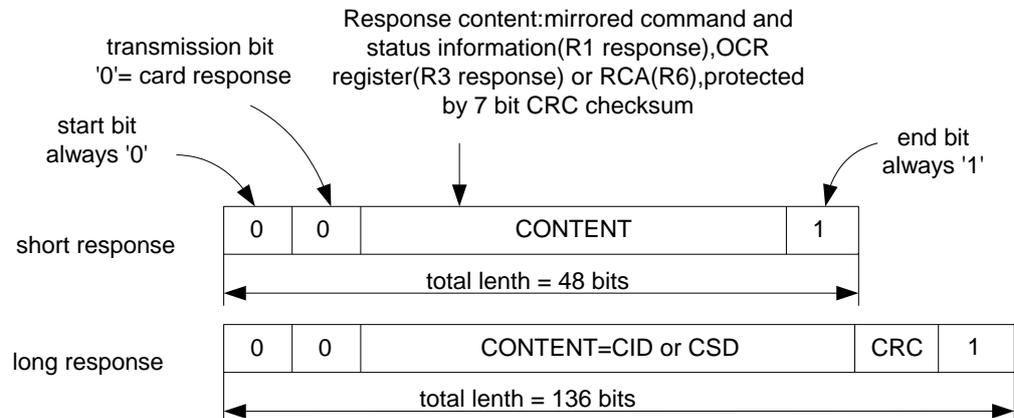


图13-30 MMC 指令响应格式



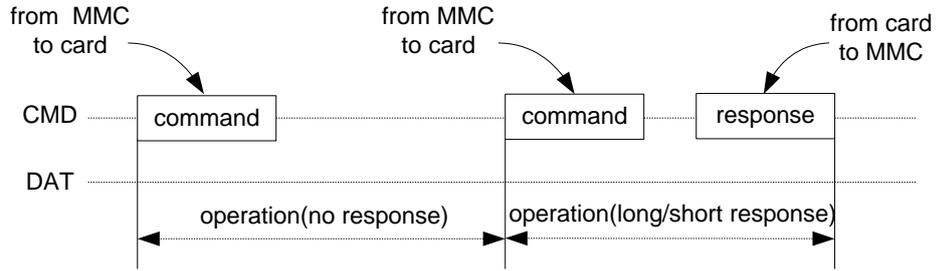
根据是否有数据传输，指令分为以下 2 种：

- 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 (eMMC DAT0~DAT7) 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 13-31 所示。

图13-31 MMC 非数据指令操作



(2) 数据传输指令

MMC 卡支持以下数据传输指令：

- 流数据读写指令
仅 MMC 卡支持，只使用 1 根数据线（即 DAT0）进行数据传输，无 CRC 校验。
 - 单块数据读写指令
一次传输完成一个块大小的数据，不需要使用停止命令结束一次数据传输。
 - 多块数据读写指令
 - predefined block count 方式
在多块读写指令前，发送块数量指令指定待传输的数据量。
 - open ended 方式
发送读写指令后，在数据传输末尾，需使用停止指令来结束一次数据传输。
- SDIO 设备的多块读写指令，不同于上述 2 种方式，在发送读写指令时，指令参数中包含待传输的数据量。

根据响应的类型，指令分为以下 3 种：

- 无响应指令
如卡复位指令。
- 短响应指令
数据传输指令、卡状态查询等均属于这类指令。
- 长响应指令
仅用于读取卡的寄存器 CID 和 CSD 信息。



数据传输

单块读写和多块读写为较常用的数据传输方式。通常 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。

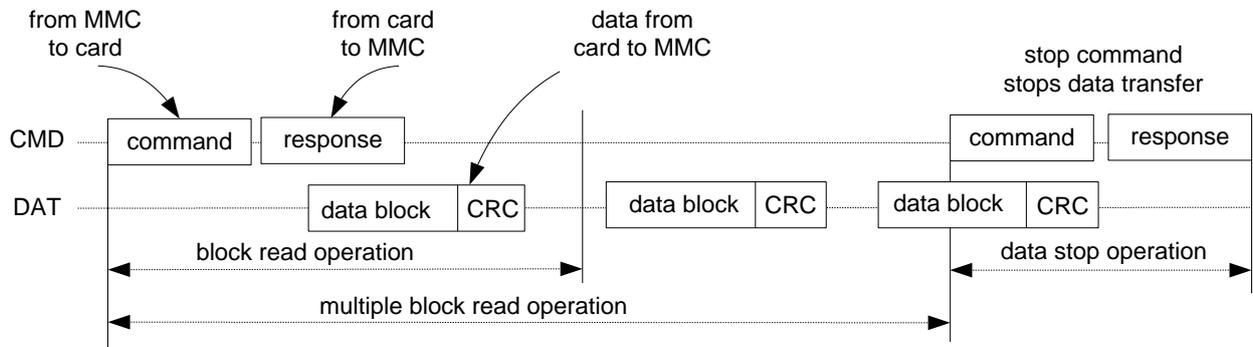
说明

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 13-32 和图 13-33 所示。

(1) 单块与多块读操作

图13-32 单块与多块读操作



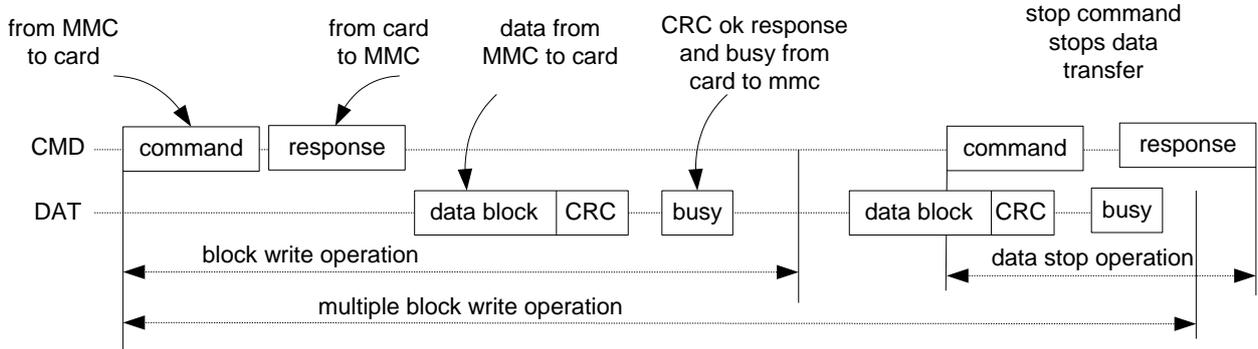
MMC 向卡发送单块或多块读指令。在接收响应的过程中，接收以块为单位的数据，其中每块数据中均包含有 CRC 校验位，以保证数据传输的完整性。

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13-33 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、4bit 或 8bit (eMMC only) 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、4bit 或 8bit (eMMC only)，使它们保持一致。MMC 的数据位宽通过寄存器 HOST_CTRL1_R 设置，卡的数据位宽则通过发送相应的指令进行设置。

1bit、4bit 和 8bit (eMMC only) 模式下的数据传输格式如图 13-34、图 13-35 和图 13-36 所示。

图13-34 1bit 数据线传输模式下的块数据格式

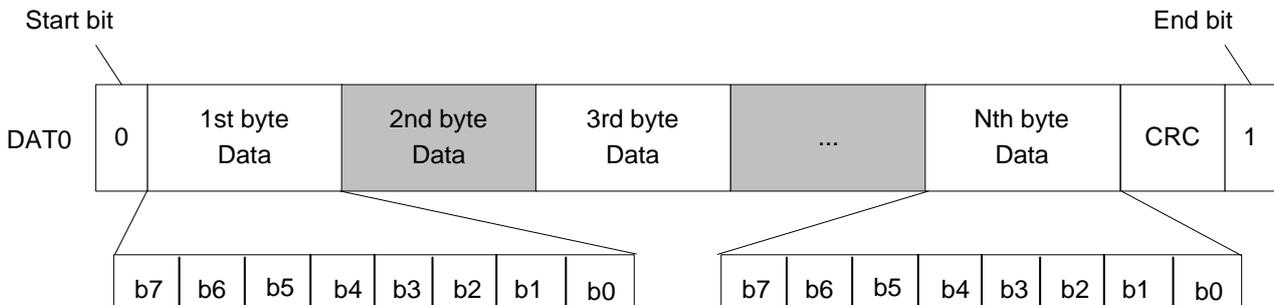
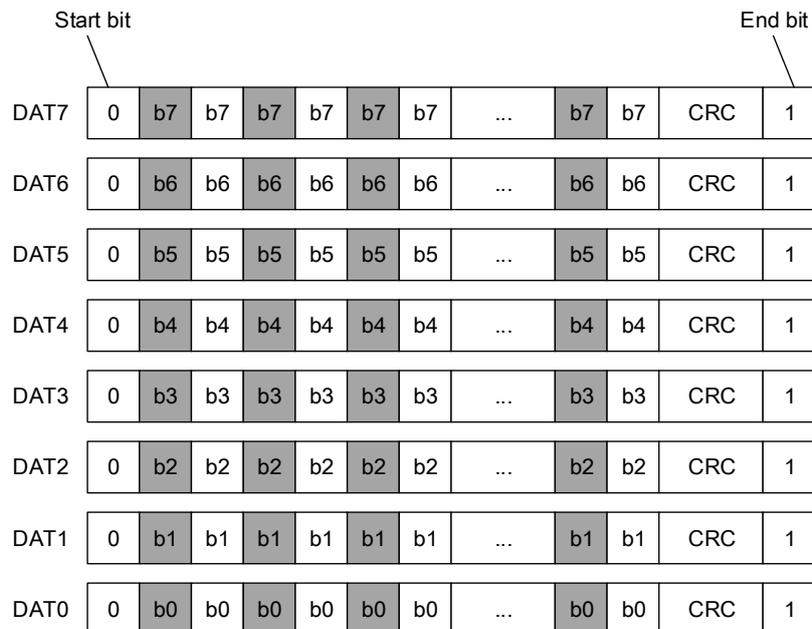




图13-35 4bit 数据线传输模式下的块数据格式



图13-36 8bit 数据线传输模式下的块数据格式



SD3.0 支持的速度模式

控制器支持 SD3.0, 各传输模式如表 13-14 所示。



表13-14 传输模式表

模式	支持的输入时钟 (MHz)	卡侧时钟	最大数据位宽	电压
SDR104	148.5MHz	148.5MHz	4bit	1.8V
SDR50	100MHz	100MHz	4bit	1.8V
SDR25	50MHz	50MHz	4bit	1.8V
SDR12	25MHz	25MHz	4bit	1.8V
HS	50MHz	50MHz	4bit	3.3V
DS	25MHz	25MHz	4bit	3.3V

SDIO3.0 支持的速度模式

控制器支持 SDIO3.0, 在 SD-mode 模式下各传输模式如表 13-15 所示。

表13-15 传输模式表

模式	接口时钟	最大数据位宽	电压
SDR104	148.5MHz	4bit	1.8V
SDR50	100MHz	4bit	1.8V
SDR25	50MHz	4bit	1.8V
SDR12	25MHz	4bit	1.8V
HS	50MHz	4bit	3.3V
DS	25MHz	4bit	3.3V

eMMC 支持的速度模式及电压

控制器在 eMMC-mode 模式下支的各传输模式如表 13-16 所示。

表13-16 传输模式表

模式	接口时钟	最大数据位宽	电压
HS400	148.5MHz	8bit	1.8V



模式	接口时钟	最大数据位宽	电压
HS200	148.5MHz	4bit	1.8V
HS_SDR	50MHz	4bit	1.8V/3.3V
Backwards Compatibility with legacy MMC card	25MHz	4bit	1.8V/3.3V

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"。

软复位

在数据传输出现异常而导致 MMC 无法回到空闲状态时：

CRG 寄存器 PERI_CRG3376/PERI_CRG3440/PERI_CRG3504 bit[16]写“1”分别对 eMMC/SDIO0/SDIO1 模块进行软复位。

建议在使用 MMC 之前，软复位 MMC。

接口时钟配置

遵从不同协议版本的 eMMC/SD/SDIO 器件，以及 eMMC/SD/SDIO 器件处于不同的状态时，均使用不同的时钟频率。

eMMC/SDIO0/SDIO1 分别通过 CRG 寄存器
PERI_CRG3376/PERI_CRG3440/PERI_CRG3504 bit[26:24]配置；

在改变 eMMC/SD/SDIO 器件的时钟频率之前，必须保证没有数据或指令正在传输。
为了避免输出到 eMMC/SD/SDIO 器件的时钟产生毛刺，在改变 eMMC/SD/SDIO 器件的时钟频率时应该遵照以下步骤：

步骤 1 关闭接口时钟。

将寄存器 CLK_CTRL_R bit[2]置“0”。

步骤 2 设置接口时钟频率。



配置 CRG 寄存器 PERI_CRG3376/PERI_CRG3440/PERI_CRG3504 bit[26:24]，选择接口时钟频率。

步骤 3 重新使能接口时钟。

将寄存器 CLK_CTRL_R bit[2]置“1”。

----结束

MMC 初始化

与卡进行命令和数据的交互前，需要初始化控制器，步骤如下：

步骤 1 清中断。将寄存器 NORMAL_INT_STAT_R bit[15:0] 和 ERROR_INT_STAT_R bit[15:0] 所有位置“1”，清除原始中断状态位。

步骤 2 关闭中断信号使能。将寄存器 NORMAL_INT_SIGNAL_EN_R bit[15:0] 和 ERROR_INT_SIGNAL_EN_R bit[15:0]所有位置“0”。

步骤 3 设置超时计数。配置寄存器 TOUT_CTRL_R bit[3:0]。

步骤 4 如果为 eMMC，将寄存器 EMMC_CTRL_R bit[0] 置“1”，否则置“0”。

步骤 5 开启 power_on。将寄存器 PWR_CTRL_R bit[0] 置“1”。

步骤 6 打开时钟。配置寄存器 CLK_CTRL_R bit[3]、[2]和[0] 为“1”。

步骤 7 打开中断状态使能。配置寄存器 NORMAL_INT_STAT_EN_R bit[15:0]和 ERROR_INT_STAT_R bit[15:0]。

步骤 8 打开中断信号使能。配置寄存器 NORMAL_INT_SIGNAL_EN_R bit[15:0]和 ERROR_INT_SIGNAL_EN_R bit[15:0]。

----结束

完成以上步骤后，配置接口时钟以后，就可以往卡发送指令了。

非数据传输指令

发送非数据传输指令的步骤如下：

步骤 1 在寄存器 ARGUMENT_R 中设置相应的指令内容。

步骤 2 在寄存器 XFER_MODE_R 和 CMD_R 中设置相应指令参数。



步骤 3 等待指令被 MMC 执行。

步骤 4 检查寄存器 NORMAL_INT_STAT_R bit[0]是否产生指令完成中断。

步骤 5 检查寄存器 ERROR_INT_STAT_R bit[15:0]是否有异常响应中断，必要时可读取响应值。

----结束

单块或多块读数据

读取单块或多块数据的步骤如下：

步骤 1 清除中断寄存器 NORMAL_INT_STAT_R 和 ERROR_INT_STAT_R 中的中断。

步骤 2 向寄存器 BLOCKSIZE_R 和 BLOCKCOUNT_R bit[11:0]写入块的大小。

步骤 3 向寄存器 SDMASA_R 写入块的个数。

步骤 4 向寄存器 ARGUMENT_R 写入读取数据的起始地址。

步骤 5 配置寄存器 XFER_MODE_R 和 CMD_R，设置数据传输参数。

对于 eMMC/SD 器件，使用指令 CMD17/CMD18 进行单块/多块读操作；对于 SDIO 卡，使用指令 CMD53 进行单块/多块读操作。

一旦寄存器 CMD_R 被写入，MMC 就执行指令；当指令被送到总线上以后，产生 cmd_done 中断。

步骤 6 Non_DMA 方式读取数据。检查寄存器 NORMAL_INT_STAT_R bit[5]，如果为 1，则从寄存器 BUF_DATA_R 读取 FIFO 中的数据，以便 MMC 接收后面的数据；同时检查数据错误中断，即寄存器 ERROR_INT_STAT_R bit[4]、bit[5]、bit[6]。此时，程序可以发送一个停止指令中止数据的传输。

步骤 7 当寄存器 NORMAL_INT_STAT_R bit[1]为 1 时，数据传输完成，从寄存器 BUF_DATA_R 中读取残留在 FIFO 中的数据。

----结束

单块与多块写数据

写入单块或多块数据的步骤如下：

步骤 1 清除中断寄存器 NORMAL_INT_STAT_R 和 ERROR_INT_STAT_R 中的中断。



步骤 2 向寄存器 BLOCKSIZE_R 和 BLOCKCOUNT_R bit[11:0]写入块的大小。

步骤 3 向寄存器 SDMASA_R 写入块的个数。

步骤 4 向寄存器 ARGUMENT_R 写入要写数据的起始地址。

步骤 5 配置寄存器 XFER_MODE_R 和 CMD_R，设置数据传输参数。

对于 eMMC 器件，使用指令 CMD24/CMD25 进行单块/多块写操作；对于 SDIO 卡，使用指令 CMD53 进行单块/多块写操作。

步骤 6 Non_DMA 方式写数据。将数据写入 FIFO，即写寄存器 BUF_DATA_R。

步骤 7 检查寄存器 NORMAL_INT_STAT_R bit[4]，如果为 1，则向寄存器 BUF_DATA_R 写入要填充到 FIFO 中的数据，以便 MMC 发送后面的数据；同时检查数据错误中断，即寄存器 ERROR_INT_STAT_R bit[4]、bit[5]、bit[6]。此时，程序可以发送一个停止指令中止数据的传输。

步骤 8 当寄存器 NORMAL_INT_STAT_R bit[1]为 1 时，数据传输完成。

----结束

13.6.4 寄存器概览

寄存器概览如表 13-17 所示。

表13-17 MMC 寄存器概览 (eMMC/SDIO0/SDIO1 基地址分别是 0x0_1002_0000/0x0_1003_0000/0x0_1004_0000)

偏移地址	名称	描述	页码
0x0000	SDMASA_R	SDMA 系统地址寄存器	13-103
0x0004	BLOCKSIZE_R	BLOCK SIZE 寄存器	13-104
0x0006	BLOCKCOUNT_R	BLOCK CNT 寄存器	13-104
0x0008	ARGUMENT_R	SD/eMMC 命令内容寄存器	13-104
0x000c	XFER_MODE_R	Transfer 模式寄存器	13-105
0x000e	CMD_R	命令寄存器	13-106
0x0010	RESP01_R	RESP01 寄存器	13-107



偏移地址	名称	描述	页码
0x0014	RESP23_R	RESP23 寄存器	13-107
0x0018	RESP45_R	RESP45 寄存器	13-108
0x001c	RESP67_R	RESP67 寄存器	13-108
0x0020	BUF_DATA_R	Buffer 数据寄存器	13-108
0x0024	PSTATE_R	当前状态寄存器	13-108
0x0028	HOST_CTRL1_R	HOST 控制 1 寄存器	13-110
0x0029	PWR_CTRL_R	上电控制寄存器	13-111
0x002c	CLK_CTRL_R	时钟控制寄存器	13-112
0x002e	TOUT_CTRL_R	超时控制寄存器	13-112
0x002f	SW_RST_R	软复位控制寄存器	13-113
0x0030	NORMAL_INT_STAT_R	普通中断状态寄存器	13-113
0x0032	ERROR_INT_STAT_R	错误中断状态寄存器	13-115
0x0034	NORMAL_INT_STAT_EN_R	普通中断状态使能寄存器	13-117
0x0036	ERROR_INT_STAT_EN_R	错误中断状态使能寄存器	13-119
0x0038	NORMAL_INT_SIGNAL_EN_R	普通中断信号使能寄存器	13-120
0x003a	ERROR_INT_SIGNAL_EN_R	错误中断信号使能寄存器	13-122
0x003c	AUTO_CMD_STAT_R	Auto 命令状态寄存器	13-124
0x003e	HOST_CTRL2_R	HOST 控制 2 寄存器	13-125
0x0040	CAPABILITIES1_R	能力 1 寄存器	13-127
0x0044	CAPABILITIES2_R	能力 2 寄存器	13-129
0x0048	CURR_CAPBILITIES1_R	当前能力 1 寄存器	13-130
0x0054	ADMA_ERR_STAT_R	ADMA 错误状态寄存器	13-131



偏移地址	名称	描述	页码
0x0058	ADMA_SA_LOW_R	ADMA 系统地址低 32bit 寄存器	13-131
0x005c	ADMA_SA_HIGH_R	ADMA 系统地址高 32bit 寄存器	13-132
0x0078	ADMA_ID_LOW_R	ADMA Intergrated 描述子地址低 32bit 寄存器	13-132
0x007c	ADMA_ID_HIGH_R	ADMA Intergrated 描述子地址高 32bit 寄存器	13-132
0x0500	MSHC_VER_ID_R	控制器版本 ID 寄存器	13-132
0x0504	MSHC_VER_TYPE_R	控制器版本类型寄存器	13-133
0x0510	MBIU_CTRL_R	MBIU 控制寄存器	13-133
0x052c	EMMC_CTRL_R	eMMC 控制寄存器	13-134
0x052e	BOOT_CTRL_R	BOOT 控制寄存器	13-135
0x0534	GP_OUT_R	eMMC 复位和 GPIO 寄存器	13-136
0x0540	AT_CTRL_R	Tuning 控制寄存器	13-136
0x0544	AT_STAT_R	Tuning 状态寄存器	13-137

13.6.5 MMC 寄存器描述

SDMASA_R

SDMASA_R 为 SDMA 系统地址寄存器。

Offset Address: 0x0000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	blockcnt_sdmasa	Host Version Enable=0, 表示 SDMA 系统地址。 Host Version Enable=1, 表示 Block	0x00000000



Bits	Access	Name	Description	Reset
			Count。 1 表示一个 block。	

BLOCKSIZE_R

BLOCKSIZE_R 为 BLOCK SIZE 寄存器。

Offset Address: 0x0004 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15]	-	reserved	保留。	0x0
[14:12]	RW	sdma_buf_bdary	SDMA 数据块边界值。 000: 4K bytes; 001: 8K bytes; 111: 512K bytes。	0x0
[11:0]	RW	xfer_block_size	传输 block size。 (单位为 byte, 最大值: 0x800)	0x000

BLOCKCOUNT_R

BLOCKCOUNT_R 为 BLOCK COUNT 寄存器。

Offset Address: 0x0006 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	block_cnt	Block 个数。	0x0000

ARGUMENT_R

ARGUMENT_R 为 SD/eMMC 命令内容寄存器。

Offset Address: 0x0008 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	argument	32bit 命令内容值。	0x00000000

XFER_MODE_R

XFER_MODE_R 为 Transfer 模式命令寄存器。

Offset Address: 0x000c Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:9]	-	reserved	保留。	0x00
[8]	RW	resp_int_disable	response 中断关闭。 0: 使能; 1: 关闭。	0x0
[7]	RW	resp_err_chk_enable	response 错误检查使能。 0: 不使能; 1: 使能。	0x0
[6]	RW	resp_type	response R1/R5 类型。 0: R1; 1: R5。	0x0
[5]	RW	multi_blk_sel	multiple/single block 选择。 0: single block; 1: multiple block。	0x0
[4]	RW	data_xfer_dir	Data Transfer 方向。 0: 控制器到卡; 1: 卡到控制器。	0x0
[3:2]	RW	auto_cmd_enable	Auto Command Enable。 00: 关闭; 01: Auto CMD12 Enable; 10: Auto CMD23 Enable;	0x0



Bits	Access	Name	Description	Reset
			11: Auto CMD Auto Select. (SDIO 中固定为 00)	
[1]	RW	block_count_enabl e	Block count 使能。 0: 关闭; 1: 使能。	0x0
[0]	RW	dma_enable	DMA 使能。 0: 无数据传输或者非 DMA 数据传输; 1: DMA 数据传输。	0x0

CMD_R

CMD_R 为 Transfer 模式命令寄存器。

Offset Address: 0x000e Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:14]	-	reserved	保留。	0x0
[13:8]	RW	cmd_index	命令编号。	0x00
[7:6]	RW	cmd_type	命令类型。 00: Normal 命令; 01: Suspend 命令; 10: Resume 命令; 11: Abort 命令。	0x0
[5]	RW	data_present_sel	当前是否为数据传输。 0: 无数据传输; 1: 有数据传输。	0x0
[4]	RW	cmd_idx_chk_enab le	命令编号检查使能。 0: 关闭; 1: 使能。	0x0



Bits	Access	Name	Description	Reset
[3]	RW	cmd_crc_chk_enable	命令 CRC 检查使能。 0: 关闭; 1: 使能。	0x0
[2]	RW	sub_cmd_flag	子命令标识。 0: Main; 1: Sub Command。	0x0
[1:0]	RW	resp_type_select	response 类型。 00: 无 response; 01: response Length 136; 10: response length 48; 11: response length 48 check。	0x0

RESP01_R

RESP01_R 为 RESP01 寄存器。

Offset Address: 0x0010 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	resp01	RESP01 寄存器。存放控制器发送命令的 response bit[39:8]bits 内容。	0x00000000

RESP23_R

RESP23_R 为 RESP23 寄存器。

Offset Address: 0x0014 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	resp23	RESP23 寄存器。存放控制器发送命令的 response bit[71:40]内容。	0x00000000



RESP45_R

RESP45_R 为 RESP45 寄存器。

Offset Address: 0x0018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	resp45	RESP45 寄存器。存放控制器发送命令的 response bit[103:72]内容。	0x00000000

RESP67_R

RESP67_R 为 RESP67 寄存器。

Offset Address: 0x001c Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	resp67	RESP67 寄存器。存放控制器发送命令的 response bit[135:104]内容。	0x00000000

BUF_DATA_R

BUF_DATA_R 为 Buffer 数据寄存器。

Offset Address: 0x0020 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	buf_data	控制器中存放接收发送数据的 Buffer，在 Slave 方式下通过不断写这个寄存器来写数据到卡中，通过不断读这个寄存器来获取卡中的数据。	0x00000000

PSTATE_R

PSTATE_R 为当前状态寄存器。



Offset Address: 0x0024 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27]	RO	cmd_issue_err	命令发送出错状态。 0: 发送命令无错; 1: 命令不能被发送出去。	0x0
[26:25]	-	reserved	保留。	0x0
[24]	RO	cmd_line_lvl	Cmd 管脚状态。 0: cmd 管脚为低电平; 1: cmd 管脚为高电平。	0x0
[23:20]	RO	sd_dat_in[3:0]	Data[3:0]管脚状态, 每个 bit 含义。 0: Data 管脚为低电平; 1: Data 管脚为高电平。	0x0
[19:12]	-	reserved	保留。	0x0
[11]	RO	buf_rd_enable	Buffer 读使能。 0: 不使能; 1: 使能。	0x0
[10]	RO	buf_wr_enable	Buffer 写使能。 0: 不使能; 1: 使能。	0x0
[9]	RO	Rd_xfer_active	读 transfer 有效。 0: 空闲状态; 1: 正在读数据。	0x0
[8]	RO	Wr_xfer_active	写 transfer 有效。 0: 空闲状态; 1: 正在写数据。	0x0
[7:4]	RO	sd_dat_in[7:4]	Data[7:4]管脚状态, 每个 bit 含义。	0x0



Bits	Access	Name	Description	Reset
			0: Data 管脚为低电平; 1: Data 管脚为高电平。	
[3]	-	reserved	保留。	0x0
[2]	RO	Dat_line_active	数据线有效。 0: 空闲状态; 1: 正在数据传输。	0x0
[1]	RO	cmd_inhibit_dat	带数据的命令有效。 0: 空闲状态, 可以发送带数据的命令; 1: 数据线正在传输或者读操作有效中。	0x0
[0]	RO	cmd_inhibit	数据线有效。 0: 空闲状态, 控制器可以发送命令; 1: 忙状态, 控制器不可以发送命令。	0x0

HOST_CTRL1_R

HOST_CTRL1_R 为 HOST 控制 1 寄存器。

Offset Address: 0x0028 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:6]	-	reserved	保留。	0x0
[5]	RW	ext_data_xfer	扩展数据位宽。 0: 参见 dat_xfer_width 寄存器描述; 1: 8bit Bus width。	0x0
[4:3]	RW	dma_sel	DMA 选择。 当 HOST_CTRL2_R 寄存器中 Host Version 4 Enable 为 1 时: 00: SDMA; 01: 保留;	0x0



Bits	Access	Name	Description	Reset
			10: ADMA2; 11: ADMA2 或者 ADMA3。 当 HOST_CTRL2_R 寄存器中 Host Version 4 Enable 为 0 时: 00: SDMA; 01: 保留; 10: 32bit 地址 ADMA2; 11: 64bit 地址 ADMA2。	
[2]	RW	high_speed_en	高速使能。 0: Normal Speed; 1: High Speed。	0x0
[1]	RW	dat_xfer_width	传输数据位宽。 0: 1bit; 1: 4bit。	0x0
[0]	-	reserved	保留。	0x0

PWR_CTRL_R

PWR_CTRL_R 为 HOST 基础配置控制寄存器。

Offset Address: 0x0029 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:1]	-	reserved	保留。	0x00
[0]	RW	sd_bus_pwr_vdd1	VDD2 Power 使能。 0: Power off; 1: Power on。	0x0



CLK_CTRL_R

CLK_CTRL_R 为时钟控制寄存器。

Offset Address: 0x002c Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:4]	-	reserved	保留。	0x00
[3]	RW	pll_enable	PLL 使能开关。 0: PLL 处于 low power 模式; 1: PLL 使能。	0x0
[2]	RW	sd_clk_en	SDIO/eMMC 时钟使能。 0: 关闭; 1: 打开。	0x0
[1]	RO	internal_clk_stable	内部时钟状态。 0: 不稳定; 1: 稳定。	0x0
[0]	RW	internal_clk_en	内部时钟使能。 0: 不使能; 1: 使能。	0x1

TOUT_CTRL_R

TOUT_CTRL_R 为超时控制寄存器。

Offset Address: 0x002e Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:4]	-	reserved	保留。	0x0
[3:0]	RW	tout_cnt	数据超時計数值。 0x0: $TMCLK \times 2^{13}$;	0x0



Bits	Access	Name	Description	Reset
			0xe: TMCLKx2^27; 其他: 保留。	

SW_RST_R

SW_RST_R 为软复位控制寄存器。

Offset Address: 0x002f Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:3]	-	reserved	保留。	0x00
[2]	RW	sw_rst_dat	软复位数据线请求。 0: 不复位; 1: 复位。	0x0
[1]	RW	sw_rst_cmd	软复位命令线请求。 0: 不复位; 1: 复位。	0x0
[0]	RW	sw_rst_all	软复位控制器请求。 0: 不复位; 1: 复位。	0x0

NORMAL_INT_STAT_R

NORMAL_INT_STAT_R 为普通中断状态寄存器。

Offset Address: 0x0030 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15]	RO	err_interrupt	汇总错误中断状态。 0: 无中断; 1: 有中断。	0x0



Bits	Access	Name	Description	Reset
[14]	WC	cqe_event	CMDQ 事件中断状态。 0: 无中断; 1: 有中断。	0x0
[13]	RO	fx_event	FX 事件中断状态。 0: 无中断; 1: 有中断。	0x0
[12]	-	reserved	保留。	0x0
[11]	RO	int_c	INT_C 中断状态。 0: 无中断; 1: 有中断。	0x0
[10]	RO	int_b	INT_B 中断状态。 0: 无中断; 1: 有中断。	0x0
[9]	RO	int_a	INT_A 中断状态。 0: 无中断; 1: 有中断。	0x0
[8]	RO	card_interrupt	卡中断中断状态。 0: 无中断; 1: 有中断。	0x0
[7:6]	-	reserved	保留。	0x0
[5]	WC	buf_rd_ready	Buffer 读 ready 中断状态。 0: 无中断; 1: 有中断。	0x0
[4]	WC	buf_wr_ready	Buffer 写 ready 中断状态。 0: 无中断; 1: 有中断。	0x0



Bits	Access	Name	Description	Reset
[3]	WC	dma_interrupt	DMA 中断状态。 0: 无中断; 1: 有中断。	0x0
[2]	WC	bgap_event	因停止请求发生 Block Gap 事件中断状态。 0: 无中断; 1: 有中断。	0x0
[1]	WC	xfer_comlete	读/写 transfer 完成中断状态。 0: 无中断; 1: 有中断。	0x0
[0]	WC	cmd_complete	命令完成中断状态。 0: 无中断; 1: 有中断。	0x0

ERROR_INT_STAT_R

ERROR_INT_STAT_R 为错误中断状态寄存器。

Offset Address: 0x0032 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:13]	-	reserved	保留。	0x0
[12]	WC	boot_ack_err	Boot Ack 错误中断, 仅用于 eMMC。 0: 无中断; 1: 有中断。	0x0
[11]	WC	resp_err	response 错误中断。 0: 无中断; 1: 有中断。	0x0
[10]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[9]	WC	adma_err	ADMA 错误中断，系统总线错误 response，或者 ADMA2/3 描述子无效。或者 CQE 描述子无效。 0：无中断； 1：有中断。	0x0
[8]	WC	auto_cmd_err	Auto CMD 错误中断状态。 0：无中断； 1：有中断。	0x0
[7]	WC	cur_lmt_err	Current limit 错误中断状态。 0：无中断； 1：有中断。	0x0
[6]	WC	data_end_bit_err	数据 endbit 错误中断状态。 0：无中断； 1：有中断。	0x0
[5]	WC	data_crc_err	数据 CRC 错误中断状态。 0：无中断； 1：有中断。	0x0
[4]	WC	data_tout_err	数据超时错误中断状态。 0：无中断； 1：有中断。	0x0
[3]	WC	cmd_idx_err	命令 Index 错误中断状态。 0：无中断； 1：有中断。	0x0
[2]	WC	cmd_end_bit_err	命令 endbit 错误中断状态。 0：无中断； 1：有中断。	0x0
[1]	WC	cmd_crc_err	命令 CRC 错误中断状态。	0x0



Bits	Access	Name	Description	Reset
			0: 无中断; 1: 有中断。	
[0]	WC	cmd_tout_err	命令超时错误中断状态。 0: 无中断; 1: 有中断。	0x0

NORMAL_INT_STAT_EN_R

NORMAL_INT_STAT_EN_R 为普通中断状态使能寄存器。

Offset Address: 0x0034 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15]	-	reserved	保留。	0x0
[14]	RW	cqe_event_stat_en	CMDQ 事件中断状态使能。 0: 不使能; 1: 使能。	0x0
[13]	RW	fx_event_stat_en	FX 事件中断状态使能。 0: 不使能; 1: 使能。	0x0
[12]	-	reserved	保留。	0x0
[11]	RW	int_c_stat_en	INT_C 中断状态使能。 0: 不使能; 1: 使能。	0x0
[10]	RW	int_b_stat_en	INT_B 中断状态使能。 0: 不使能; 1: 使能。	0x0
[9]	RW	int_a_stat_en	INT_A 中断状态使能。	0x0



Bits	Access	Name	Description	Reset
			0: 不使能; 1: 使能。	
[8]	RW	card_interrupt_stat_en	卡中断状态使能。 0: 不使能; 1: 使能。	0x0
[7:6]	-	reserved	保留。	0x0
[5]	RW	buf_rd_ready_stat_en	Buffer 读 ready 中断使能。 0: 不使能; 1: 使能。	0x0
[4]	RW	buf_wr_ready_stat_en	Buffer 写 ready 中断使能。 0: 不使能; 1: 使能。	0x0
[3]	RW	dma_interrupt_stat_en	DMA 中断使能。 0: 不使能; 1: 使能。	0x0
[2]	RW	bgap_event_stat_en	因停止请求发生 Block Gap 事件中断使能。 0: 不使能; 1: 使能。	0x0
[1]	RW	xfer_comlete_stat_en	读/写 transfer 完成中断状态使能。 0: 不使能; 1: 使能。	0x0
[0]	RW	cmd_complete_stat_en	命令完成中断状态使能。 0: 不使能; 1: 使能。	0x0



ERROR_INT_STAT_EN_R

ERROR_INT_STAT_EN_R 为错误中断状态使能寄存器。

Offset Address: 0x0036 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:13]	-	reserved	保留。	0x0
[12]	RW	boot_ack_err_stat_en	Boot Ack 错误中断状态使能, 仅用于 eMMC。 0: 不使能; 1: 使能。	0x0
[11]	RW	resp_err_stat_en	response 错误中断状态使能。 0: 不使能; 1: 使能。	0x0
[10]	-	reserved	保留。	0x0
[9]	RW	adma_err_stat_en	ADMA 错误中断状态使能。 系统总线错误 response, 或者 ADMA2/3 描述子无效, 或者 CQE 描述子无效。 0: 不使能; 1: 使能。	0x0
[8]	RW	auto_cmd_err_stat_en	Auto CMD 错误中断状态使能。 0: 不使能; 1: 使能。	0x0
[7]	RW	cur_lmt_err_stat_en	Current limit 错误中断状态使能。 0: 不使能; 1: 使能。	0x0
[6]	RW	data_end_bit_err_stat_en	数据 endbit 错误中断状态使能。 0: 不使能; 1: 使能。	0x0
[5]	RW	data_crc_err_stat_	数据 CRC 错误中断状态使能。	0x0



Bits	Access	Name	Description	Reset
		en	0: 不使能; 1: 使能。	
[4]	RW	data_tout_err_stat_en	数据超时错误中断状态使能。 0: 不使能; 1: 使能。	0x0
[3]	RW	cmd_idx_err_stat_en	命令 Index 错误中断状态使能。 0: 不使能; 1: 使能。	0x0
[2]	RW	cmd_end_bit_err_stat_en	命令 endbit 错误中断状态使能。 0: 不使能; 1: 使能。	0x0
[1]	RW	cmd_crc_err_stat_en	命令 CRC 错误中断状态使能。 0: 不使能; 1: 使能。	0x0
[0]	RW	cmd_tout_err_stat_en	命令超时错误中断状态使能。 0: 不使能; 1: 使能。	0x0

NORMAL_INT_SIGNAL_EN_R

NORMAL_INT_SIGNAL_EN_R 为普通中断信号使能寄存器。

Offset Address: 0x0038 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15]	-	reserved	保留	0x0
[14]	RW	cqe_event_signal_en	CMDQ 事件中断信号使能。 0: 不使能; 1: 使能。	0x0



Bits	Access	Name	Description	Reset
[13]	RW	fx_event_signal_en	FX 事件中断信号使能。 0: 不使能; 1: 使能。	0x0
[12]	-	reserved	保留	0x0
[11]	RW	int_c_signal_en	INT_C 中断信号使能。 0: 不使能; 1: 使能。	0x0
[10]	RW	int_b_signal_en	INT_B 中断信号使能。 0: 不使能; 1: 使能。	0x0
[9]	RW	int_a_signal_en	INT_A 中断信号使能。 0: 不使能; 1: 使能。	0x0
[8]	RW	card_interrupt_signal_en	卡中断信号使能。 0: 不使能; 1: 使能。	0x0
[7:6]	-	reserved	保留	0x0
[5]	RW	buf_rd_ready_signal_en	Buffer 读 ready 信号使能。 0: 不使能; 1: 使能。	0x0
[4]	RW	buf_wr_ready_signal_en	Buffer 写 ready 信号使能。 0: 不使能; 1: 使能。	0x0
[3]	RW	dma_interrupt_signal_en	DMA 中断信号使能。 0: 不使能; 1: 使能。	0x0



Bits	Access	Name	Description	Reset
[2]	RW	bgap_event_signal_en	因停止请求发生 Block Gap 事件中断信号使能。 0: 不使能; 1: 使能。	0x0
[1]	RW	xfer_complete_signal_en	读/写 transfer 完成中断信号使能。 0: 不使能; 1: 使能。	0x0
[0]	RW	cmd_complete_signal_en	命令完成中断信号使能。 0: 不使能; 1: 使能。	0x0

ERROR_INT_SIGNAL_EN_R

ERROR_INT_SIGNAL_EN_R 为错误中断信号使能寄存器。

Offset Address: 0x003a Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:13]	-	reserved	保留。	0x0
[12]	RW	boot_ack_err_signal_en	Boot Ack 错误中断信号使能, 仅用于 eMMC。 0: 不使能; 1: 使能。	0x0
[11]	RW	resp_err_signal_en	response 错误中断信号使能。 0: 不使能; 1: 使能。	0x0
[10]	-	reserved	保留。	0x0
[9]	RW	adma_err_signal_en	ADMA 错误中断信号使能, 系统总线错误 response, 或者 ADMA2/3 描述子无效,	0x0



Bits	Access	Name	Description	Reset
			或者 CQE 描述子无效。 0: 不使能; 1: 使能。	
[8]	RW	auto_cmd_err_signal_en	Auto CMD 错误中断信号使能。 0: 不使能; 1: 使能。	0x0
[7]	RW	cur_lmt_err_signal_en	Current limit 错误中断信号使能。 0: 不使能; 1: 使能。	0x0
[6]	RW	data_end_bit_err_signal_en	数据 endbit 错误中断信号使能。 0: 不使能; 1: 使能。	0x0
[5]	RW	data_crc_err_signal_en	数据 CRC 错误中断信号使能。 0: 不使能; 1: 使能。	0x0
[4]	RW	data_tout_err_signal_en	数据超时错误中断信号使能。 0: 不使能; 1: 使能。	0x0
[3]	RW	cmd_idx_err_signal_en	命令 Index 错误中断信号使能。 0: 不使能; 1: 使能。	0x0
[2]	RW	cmd_end_bit_err_signal_en	命令 endbit 错误中断信号使能。 0: 不使能; 1: 使能。	0x0
[1]	RW	cmd_crc_err_signal_en	命令 CRC 错误中断信号使能。 0: 不使能; 1: 使能。	0x0



Bits	Access	Name	Description	Reset
[0]	RW	cmd_tout_err_signal_en	命令超时错误中断信号使能。 0: 不使能; 1: 使能。	0x0

AUTO_CMD_STAT_R

AUTO_CMD_STAT_R 为 Auto 命令状态寄存器。

Offset Address: 0x003c Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:8]	-	reserved	保留。	0x00
[7]	RO	cmd_not_issued_auto_cmd12	有 Auto CMD12 错误时发送后续命令状态。 0: 无错误; 1: 还没有通过数据线发送 CMD。	0x0
[6]	-	reserved	保留。	0x0
[5]	RO	auto_cmd_resp_err	Auto CMD response 状态。 0: 无错; 1: 错。	0x0
[4]	RO	auto_cmd_idx_err	Auto CMD Index 状态。 0: 无错; 1: 错。	0x0
[3]	RO	auto_cmd_endbit_err	Auto CMD Endbit 状态。 0: 无错; 1: 错。	0x0
[2]	RO	auto_cmd_crc_err	Auto CMD CRC 状态。 0: 无错; 1: 错。	0x0



Bits	Access	Name	Description	Reset
[1]	RO	auto_cmd_tout_err	Auto CMD 超时状态。 0: 无错; 1: 错。	0x0
[0]	RO	auto_cmd12_not_exec	Auto CMD12 执行状态。 0: 执行; 1: 没有执行。	0x0

HOST_CTRL2_R

HOST_CTRL2_R 为控制器 2 寄存器。

Offset Address: 0x003e Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15]	RW	preset_val_enable	预设值自动选择使能。 0: 不使能, SDCLK 和驱动能力由驱动控制; 1: 使能。	0x0
[14]	RW	async_int_enable	异步中断使能。 0: 关闭; 1: 打开。	0x0
[13]	-	reserved	保留。	0x0
[12]	RW	host_ver4_enable	控制器版本 4 使能。 0: 3.0 版本; 1: 4.0 版本。	0x0
[11]	RW	cmd23_enable	CMD23 使能。 0: Auto CMD23 关闭; 1: Auto CMD23 使能。	0x0
[10]	RW	adma2_len_mode	ADMA2 长度。	0x0



Bits	Access	Name	Description	Reset
			0: 16bit 数据长度; 1: 26bit 数据长度。	
[9:8]	-	reserved	保留。	0x0
[7]	RW	sample_clk_sel	sample 时钟选择。 0: 选择 fixed 时钟采数据; 1: 选 tuned 时钟采数据。	0x0
[6]	RW	exec_tuning	执行 tuning,完成后自动清 0。 0: 没做 tuning 或者 tuning 完成; 1: 执行 tuning。	0x0
[5:4]	RW	drv_strength_sel	驱动能力选择。 00: 选择 TYPEB; 01: 选择 TYPEA; 10: 选择 TYPEC; 11: 选择 TYPED。	0x0
[3]	-	reserved	保留。	0x0
[2:0]	RW	uhs_mode_sel	UHS 模式: 000: SDR12; 001: SDR25; 010: SDR50; 011: SDR104; 100: reserved; 101: reserved; 110: reserved; 111: reserved。 eMMC 模式: 000: Legacy; 001: High Speed SDR;	0x0



Bits	Access	Name	Description	Reset
			010: reserved; 011: HS200; 100: reserved; 101: reserved; 110: reserved; 111: HS400。	

CAPABILITIES1_R

CAPABILITIES1_R 为能力 1 寄存器。

Offset Address: 0x0040 Total Reset Value: 0x3F6E_0181

Bits	Access	Name	Description	Reset
[31:30]	RO	slot_type_r	Slot 类型。 00: 可移除卡 Slot; 01: Embedded Slot; 10: 共享总线 Slot(SD 模式下使用); 11: 保留。	0x0
[29]	RO	async_int_support	同步中断是否支持。 0: 不支持; 1: 支持。	0x1
[28]	-	reserved	保留。	0x1
[27]	-	reserved	保留。	0x1
[26:24]	-	reserved	保留。	0x7
[23]	RO	sus_res_support	暂停、恢复支持。 0: 不支持; 1: 支持。	0x0
[22]	RO	sdma_support	SDMA 支持。	0x1



Bits	Access	Name	Description	Reset
			0: 不支持; 1: 支持。	
[21]	RO	high_speed_support	是否支持 High Speed。 0: 不支持; 1: 支持。	0x1
[20]	-	reserved	保留。	0x0
[19]	RO	adma2_support	是否支持 ADMA2。 0: 不支持; 1: 支持。	0x1
[18]	RO	embedded_8_bit	是否支持 Embedded 器件 8bit。 0: 不支持; 1: 支持。	0x1
[17:16]	RO	max_blk_len	最大 block 长度。 00: 512Byte; 01: 1024Byte; 10: 2048Byte; 11: 保留。	0x2
[15:8]	RO	base_clk_freq	SD 时钟基础频率。 0x0: 1MHz; 0x3f: 63MHz; 0x40~0xff: 不支持。	0x01
[7]	RO	tout_clk_unit	超时时钟单位。 0: KHz; 1: MHz。	0x1
[6]	-	reserved	保留。	0x0
[5:0]	RO	tout_clk_freq	超时频率。	0x01



Bits	Access	Name	Description	Reset
			0x1: 1KHz/1MHz; 0x2: 2KHz/2MHz; 0x3f: 63KHz/63MHz。	

CAPABILITIES2_R

CAPABILITIES2_R 为能力 2 寄存器。

Offset Address: 0x0044 Total Reset Value: 0x0800_2077

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	保留。	0x0
[27]	RO	adma3_support	是否支持 ADMA3。 0: 不支持; 1: 支持。	0x1
[26:16]	-	reserved	保留。	0x000
[15:14]	RO	re_tuning_modes	retuning 模式。 00: MODE1,Timer; 01: MODE2,Timer 和 ReTuning 请求; 10: MODE3,Auto retuning Timer 和 Retuning 请求; 11: 保留。	0x0
[13]	RO	use_tuning_ser50	SDR50 使用 Tuning。 0: 不使用; 1: 使用。	0x1
[12]	-	reserved	保留。	0x0
[11:8]	RO	retune_cnt	Retuning 计数。 0x1: 1 秒;	0x0



Bits	Access	Name	Description	Reset
			0x3: 4 秒; 其他: 保留。	
[7]	-	reserved	保留。	0x0
[6]	RO	drv_typed	是否支持 TYPED。 0: 不支持; 1: 支持。	0x1
[5]	RO	drv_typec	是否支持 TYPEC。 0: 不支持; 1: 支持。	0x1
[4]	RO	drv_typea	是否支持 TYPEA。 0: 不支持; 1: 支持。	0x1
[3]	RO	uhs2_support	是否支持 UHS2。 0: 不支持; 1: 支持。	0x0
[2]	-	reserved	保留。	0x1
[1]	RO	sdr104_support	是否支持 SDR104。 0: 不支持; 1: 支持。	0x1
[0]	RO	sdr50_support	是否支持 SDR50。 0: 不支持; 1: 支持。	0x1

CURR_CAPBILITIES1_R

CURR_CAPBILITIES1_R 为当前能力 1 寄存器。

Offset Address: 0x0048 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:24]	-	reserved	保留。	0x00
[23:16]	RO	max_cur_18v	1.8V 下电流值。	0x00
[15:8]	RO	max_cur_30v	3.0V 下电流值。	0x00
[7:0]	RO	mac_cur_33v	3.3V 下电流值。	0x00

ADMA_ERR_STAT_R

ADMA_ERR_STAT_R 为 ADMA 错误状态寄存器。

Offset Address: 0x0054 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:3]	-	reserved	保留。	0x00
[2]	RO	adma_len_err	ADMA 长度不匹配错误状态。 0: 无错; 1: 有错。	0x0
[1:0]	RO	adma_err_states	ADMA 错误状态。 00: ST_STOP; 01: ST_FDS; 10: ST_UNUSED; 11: ST_TFR。	0x0

ADMA_SA_LOW_R

ADMA_SA_LOW_R 为 ADMA 系统地址低 32bit 寄存器。

Offset Address: 0x0058 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	adma_sa_low	ADMA 系统地址低 32bit。	0x00000000



ADMA_SA_HIGH_R

ADMA_SA_HIGH_R 为 ADMA 系统地址高 32bit 寄存器。

Offset Address: 0x005c Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	adma_sa_high	ADMA 系统地址高 32bit。	0x00000000

ADMA_ID_LOW_R

ADMA_ID_LOW_R 为 ADMA Intergrated 描述子地址低 32bit 寄存器。

Offset Address: 0x0078 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	adma_id_low	ADMA Intergrated 描述子地址低 32bit。	0x00000000

ADMA_ID_HIGH_R

ADMA_ID_HIGH_R 为 ADMA Intergrated 描述子地址高 32bit 寄存器。

Offset Address: 0x007c Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	adma_id_high	ADMA Intergrated 描述子地址高 32bit。	0x00000000

MSHC_VER_ID_R

MSHC_VER_ID_R 为控制器版本 ID 寄存器。

Offset Address: 0x0500 Total Reset Value: 0x3137_302A

Bits	Access	Name	Description	Reset
[31:0]	RO	mshc_ver_id	控制器版本 ID 寄存器。	0x3137302A



MSHC_VER_TYPE_R

MSHC_VER_TYPE_R 为控制器版本类型寄存器。

Offset Address: 0x0504 Total Reset Value: 0x6761_2A2A

Bits	Access	Name	Description	Reset
[31:0]	RO	mshc_ver_id	控制器版本类型寄存器。	0x67612A2A

MBIU_CTRL_R

MBIU_CTRL_R 为 MBIU 控制寄存器。

Offset Address: 0x0510 Total Reset Value: 0x0707_000F

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	保留。	0x00
[26:24]	RW	gm_write_osrc_lmt	写总线操作 outstanding 值。 等于此寄存器值加 1。	0x7
[23:19]	-	reserved	保留。	0x00
[18:16]	RW	gm_read_osrc_lmt	读总线操作 outstanding 值。 等于此寄存器值加 1。	0x7
[15:10]	-	reserved	保留。	0x00
[9]	RW	gm_onerequestonly	outstanding 配置生效使能。 0: outstanding 按 gm_write_osrc_lmt /read_osrc_lmt 生效; 1: outstanding 固定为 1。	0x0
[8]	RW	gm_axi_1k_en	AXI 总线跨 1K 边界读写功能使能。 0: 不使能; 1: 使能。	0x0
[7:4]	-	reserved	保留。	0x0



Bits	Access	Name	Description	Reset
[3]	RW	gm_enburst16	16burst 使能。 0: 关闭 16burst; 1; 使能 16burst。	0x1
[2]	RW	gm_enburst8	8burst 使能。 0: 关闭 8burst; 1; 打开 8burst。	0x1
[1]	RW	gm_enburst4	4burst 使能。 0: 关闭 4burst; 1; 打开 4burst。	0x1
[0]	RW	gm_enburst_undef	burst 配置生效使能。 0: 按下面三种配置 gm_enburst4/8/16 产生固定 burst; 1: 按实际数据长度产生对应 burst。	0x1

EMMC_CTRL_R

EMMC_CTRL_R 为 eMMC 控制寄存器。

Offset Address: 0x052c Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:9]	-	reserved	保留。	0x00
[8]	RW	enh_strobe_enable	Enhanced Strobe 使能。 此寄存器域指示在 HS400 模式下控制器使用 data Strobe 来采集 CMD 线。 0: HS400 模式下, 用 cclk_rx 采样 CMD 线; 1: HS400 模式下, 用 data strobe 采样 CMD 线。	0x0



Bits	Access	Name	Description	Reset
[7:2]	-	reserved	保留。	0x00
[1]	RW	disable_data_crc_chk	关闭数据 CRC 检查。 0: 数据 CRC 检查使能; 1: 数据 CRC 检查关闭。	0x00
[0]	RW	card_is_emmc	连接的卡类型。 0: 非 eMMC 卡; 1: eMMC 卡。	0x0

BOOT_CTRL_R

BOOT_CTRL_R 为 BOOT 控制寄存器。

Offset Address: 0x052e Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:12]	RW	boot_tout_cnt	boot ack 超时计数。 0x0: $TMCLKx2^{13}$; 0x1: $TMCLKx2^{14}$; 0xE: $TMCLKx2^{27}$; 其他: 保留。	0x00
[11:9]	-	reserved	保留。	0x0
[8]	RW	boot_ack_enable	boot ack 使能。 0: 关闭; 1: 使能。	0x0
[7]	WO	validate_boot	强制 boot 有效使能。 0: 无影响; 1: boot 使能有效。	0x0
[6:1]	-	reserved	保留。	0x00



Bits	Access	Name	Description	Reset
[0]	RW	man_boot_en	强制 boot 使能, 要和 validate_boot 一起配置, boot 传输完成后, 这个 bit 会自清 0: 无影响; 1: boot 使能有效。	0x0

GP_OUT_R

GP_OUT_R 为 eMMC 复位和 GPIO 寄存器。

Offset Address: 0x0534 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x000000
[0]	RW	emmc_rst_n	eMMC 卡复位信号。 0: 复位; 1: 不复位。	0x1

AT_CTRL_R

AT_CTRL_R 为 tuning 控制寄存器。

Offset Address: 0x0540 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	保留。	0x000000
[4]	RW	sw_tuning_en	软件配置 tuning 使能。 0: 不能使; 1: 使能。	0x00
[3:0]	-	reserved	保留。	0x0



AT_STAT_R

AT_STAT_R 为 tuning 状态寄存器。

Offset Address: 0x0544 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	保留。	0x000000
[7:0]	RW	sw_ph_sel_code	软件配置相位值。 0x0: 0°; 0x1: 11.25°; 0x2: 22.5°; 0x1f: 348.75°; 其他: 保留。	0x0

13.7 GPIO

13.7.1 概述

芯片支持 15 组 GPIO (General Purpose Input/Output), 即 GPIO0 ~ GPIO14。每组 GPIO 提供 8 个可编程的输入输出管脚。

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。

须知

- 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 Hi35xxVxx_PINOUT_CN.xlsx (Hi35xxVxx 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。

13.7.2 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。

13.7.3 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

须知

当 GPIO 用作输出时，应不使能 GPIO 的中断功能，否则当输出信号符合中断产生条件时，会产生 GPIO 中断。

中断操作

GPIO 的中断通过 GPIO_IS 等 7 个寄存器进行控制。通过这些寄存器可以选择中断源、极性以及边沿特性。GPIO 对应的中断号请参见“3.3 中断系统”。



当有多个中断同时发生时，将会统一汇集成一个中断进行上报，GPIO 的中断映射关系请参见“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_IIE 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_IIE 为 1，使能中断。

----结束

须知

初始化过程中 GPIO 管脚上数据需保持稳定，以免产生假中断。

GPIO 的中断设置由 7 个寄存器控制。当有一个或多个 GPIO 管脚产生中断，一个组合中断会输送到中断控制器。边沿触发和电平触发有以下不同：

- 边沿触发的中断：软件必需清除该中断以使能更深的中断。
- 电平触发的中断：外部中断源应该保持该电平直到处理器识别到该中断。

13.7.4 GPIO 寄存器概览

芯片的 GPIO 基地址如表 13-18 所示。

表13-18 芯片 GPIO 寄存器对应的基地址

基地址	GPIO 控制器
0x0_1109_E000	GPIO14



基地址	GPIO 控制器
0x0_1109_D000	GPIO13
0x0_1109_C000	GPIO12
0x0_1109_B000	GPIO11
0x0_1109_A000	GPIO10
0x0_1109_9000	GPIO9
0x0_1109_8000	GPIO8
0x0_1109_7000	GPIO7
0x0_1109_6000	GPIO6
0x0_1109_5000	GPIO5
0x0_1109_4000	GPIO4
0x0_1109_3000	GPIO3
0x0_1109_2000	GPIO2
0x0_1109_1000	GPIO1
0x0_1109_0000	GPIO0

说明

GPIO_n 对应的寄存器地址：GPIO_n 基地址+该寄存器偏移地址。

表 13-19 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13-19 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	13-141
0x400	GPIO_DIR	GPIO 方向控制寄存器	13-142
0x404	GPIO_IS	GPIO 中断触发寄存器	13-142
0x408	GPIO_IBE	GPIO 双沿触发中断寄存器	13-142



偏移地址	名称	描述	页码
0x40C	GPIO_IEV	GPIO 触发中断条件寄存器	13-143
0x410	GPIO_IE	GPIO 中断屏蔽寄存器	13-143
0x414	GPIO_RIS	GPIO 原始中断状态寄存器	13-144
0x418	GPIO_MIS	GPIO 屏蔽状态中断寄存器	13-144
0x41C	GPIO_IC	GPIO 中断清除寄存器	13-144

13.7.5 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。

须知

当 GPIO_DIR 相应的比特配置为输入时，有效读取的结果将返回管脚的值；当配置为输出的时候，有效读取的结果将返回写入的值。

GPIO_DATA 寄存器利用 PADDR[9:2]实现了读写寄存器比特的屏蔽操作。该寄存器对应 256 个地址空间。PADDR[9:2]分别对应 GPIO_DATA[7:0]，当相应的 bit 为高时，则可以对相应的位进行读写操作；反之，若对应 bit 为低则不能进行操作。例如：

- 若地址为 0x3FC (0b11_1111_1100)，则对 GPIO_DATA[7:0]这 8bit 操作全部有效。
- 若地址为 0x200 (0b10_0000_0000)，则仅对 GPIO_DATA[7]的操作有效。

Offset Address: 0x000 ~ 0x3FC Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	gpio_data	当 GPIO 配置为输入模式时，为 GPIO 输入	0x00



Bits	Access	Name	Description	Reset
			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。	

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_I_EV 控制; 1: 双边触发中断。	0x00

GPIO_I_EV

GPIO_I_EV 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_I_E

GPIO_I_E 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。 0: 屏蔽中断; 1: 不屏蔽中断。	0x00



GPIO_RIS

GPIO_RIS 为 GPIO 原始中断状态寄存器。用来查询 GPIO 管脚原始中断状态。

Offset Address: 414 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RO	gpio_ris	GPIO 原始中断寄存器, bit[7:0]分别对应 GPIO_DATA[7:0], 指示未屏蔽的中断状态。该状态不受 GPIO_IE 寄存器屏蔽控制。 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	0x00



Bits	Access	Name	Description	Reset
			GPIO_DATA [7:0], 各比特可独立控制。 0: 无影响; 1: 清除中断。	

13.8 USB3.0

13.8.1 概述

该系统支持 1 个 USB3.0 DRD(Dual Role Device)模块, USB3.0 DRD 支持 USB3.0 host 和 USB3.0 device 的静态切换, 即只需要在控制器启动前, 配置相关寄存器就能使该模块工作在 USB3.0 host 或者 USB3.0 device 模式;

两个模块支持 USB3.0 协议规定的 5Gbit/s 传输速率以及向后兼容 USB2.0 协议规定的 480Mbit/s 传输速率; 完全支持 XHCI 1.0 协议; 支持超速传输的 PIPE 接口协议以及兼容高速传输的 UTMI 接口协议; 两个 USB3.0 控制器均可以完成:

- 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

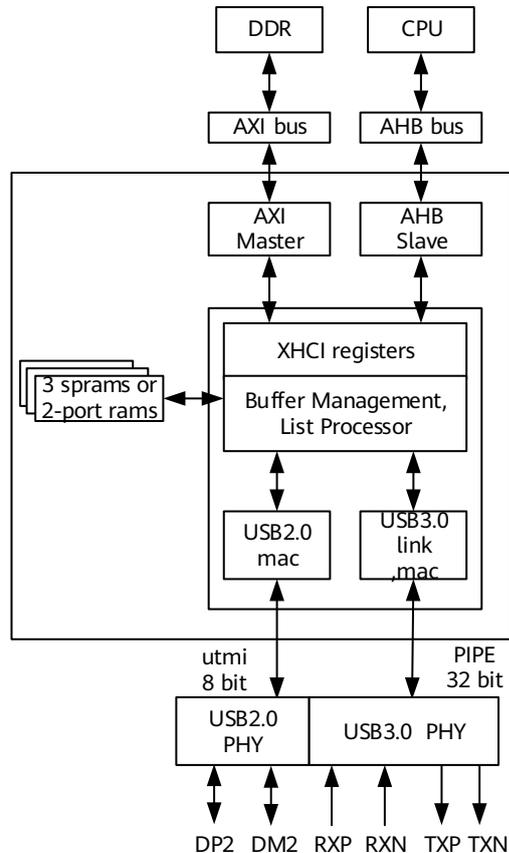
13.8.2 功能描述

逻辑框图

USB 3.0 逻辑框图如图 13-37 所示。



图13-37 USB 3.0 逻辑框图



UTMI: USB2.0 Transceiver Macrocell Interface
XHCI: eXtensible Host Controller Interface

功能特点

USB 3.0 DRD 具有以下功能特点:

- 完全兼容 USB3.0 以及向下兼容 USB 2.0。
- 完全符合 XHCI 1.0。
- USB3.0 DRD 可单独工作在 Host 或者 Device 模式。
- 可以支持 Super-speed、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 和 USB3.0 的 U0、U1、U2、U3 四种功耗状态。
- 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。



- USB3.0 DRD 支持 Device 工作模式下的 Control Transfer、Bulk Transfer、Isochronous Transfer、Interrupt Transfer 四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 64 个设备。

工作原理

USB 3.0 DRD 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。

13.8.3 工作方式

USB3.0 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。

关断时钟的步骤如下：

步骤 1 向 PERI_CRG3632[6]、[4]写入 0x1，打开 U2PHY 的参考时钟门控。



步骤 2 向 PERI_CRG3632[0]、[1]、[2]写入 0x0，撤离 U2PHY 的配置复位，模拟复位和数字复位。

步骤 3 向 PERI_CRG3665[4]、[2]写入 0x1，打开 COMPHY 参考时钟门控、配置时钟门控。

步骤 4 向 PERI_CRG3665[1]、[0]写入 0x0，撤离 COMPHY 的配置复位和软复位。

步骤 5 向 PERI_CRG3664 [12]、[8]、[6]、[5]、[4]写入 0x1，打开控制器的 PIPE、UTMI、SUSPEND、REF 和总线等工作时钟门控。

步骤 6 向 PERI_CRG3664[0]写入 0x0，撤离 U3 DRD 控制器的复位。

----结束

USB3.0 DRD Host/device 工作模式切换

切换操作如下：

- 向 PERI_USB3_GCTL [prtcapdir] 写入 2'b01，DRD 控制器将工作于 Host 模式。
- 向 PERI_USB3_GCTL [prtcapdir] 写入 2'b10，DRD 控制器将工作于 Device 模式。

须知

Host/Device 工作模式切换只支持静态切换，不支持动态切换。

13.8.4 USB3.0 寄存器概览

USB3.0 寄存器概览如表 13-20 所示。

表13-20 USB3.0 寄存器概览 (USB3.0 DRD 基地址是 0x0_1032_0000)

偏移地址	名称	描述	页码
0xC100	PERI_USB3_GSBUS_CFG0	全局 SOC 总线配置寄存器 0	13-151
0xC104	PERI_USB3_GSBUS_CFG1	全局 SOC 总线配置寄存器 1	13-153
0xC108	PERI_USB3_GTXTH_RCFG	全局发送门限控制寄存器	13-154



偏移地址	名称	描述	页码
0xC10C	PERI_USB3_GRXTHRCFG	全局接收门限控制寄存器	13-154
0xC110	PERI_USB3_GCTL	全局 core 控制寄存器	13-155
0xC118	PERI_USB3_GSTS	全局状态寄存器	13-160
0xC11C	PERI_USB3_GUCTL1	全局用户控制寄存器 1	13-161
0xC124	PERI_USB3_GGPIO	全局 GPIO 寄存器	13-162
0xC128	PERI_USB3_GUID	全局用户 ID 寄存器	13-162
0xC12C	PERI_USB3_GUCTL	全局用户控制寄存器	13-162
0xC130	PERI_USB3_GBUSERRADDR_HI	全局总线错误地址高 32 位寄存器	13-164
0xC134	PERI_USB3_GBUSERRADDR_LO	全局总线错误地址低 32 位寄存器	13-164
0xC138	PERI_USB3_GPRTBIMAP_HI	超速端口-总线映射高 32 位寄存器	13-165
0xC13C	PERI_USB3_GPRTBIMAP_LO	超速端口-总线映射低 32 位寄存器	13-165
0xC180	PERI_USB3_GPRTBIMAP_HS_HI	高速端口-总线映射高 32 位寄存器	13-165
0xC184	PERI_USB3_GPRTBIMAP_HS_LO	高速端口-总线映射低 32 位寄存器	13-166
0xC188	PERI_USB3_GPRTBIMAP_FS_HI	全速端口-总线映射高 32 位寄存器	13-166
0xC18C	PERI_USB3_GPRTBIMAP_FS_LO	全速端口-总线映射低 32 位寄存器	13-166
0xC200	PERI_USB3_GUSB2PHYCFG	全局 USB2.0 PHY 配置寄存器	13-166
0xC2C0	PERI_USB3_GUSB3PIPECTLN	全局 USB3.0 PIPE 控制寄存器	13-169
0xC304	PERI_USB3_GTXFIFOSIZN	全局发送 FIFO 大小寄存器	13-173



偏移地址	名称	描述	页码
0xC384	PERI_USB3_GRXFIFOSIZN	全局接收 FIFO 大小寄存器	13-173
0xC410	PERI_USB3_GEVNTADR_HI	全局 event buffer 的地址高 32 位寄存器	13-174
0xC414	PERI_USB3_GEVNTADR_LO	全局 event buffer 的地址低 32 位寄存器	13-174
0xC418	PERI_USB3_GEVNTSIZN	全局 event buffer 的大小寄存器	13-174
0xC41C	PERI_USB3_GEVNTCOUNTN	全局 event buffer 的计数寄存器	13-175
0xC610	PERI_USB3_GTXFIFOPRIDEV	外设的全局 TX FIFO DMA 优先寄存器	13-175
0xC618	PERI_USB3_GTXFIFOPRIHST	host 的全局 TX FIFO DMA 优先寄存器	13-175
0xC61C	PERI_USB3_GRXFIFOPRIHST	host 的全局 RX FIFO DMA 优先寄存器	13-176
0xC620	PERI_USB3_GFIFOPRIDBC	host 的全局 Debug 性能时 DMA 优先寄存器	13-176
0xC624	PERI_USB3_GDMAHLRATIO	host 的全局 FIFO DMA 高、低优先权比例寄存器	13-176
0xC630	PERI_USB3_GFLADJ	GFLADJ 为全局帧长度调整寄存器	13-177
0xC700	PERI_USB3_DCFG	外设配置寄存器	13-178
0xC704	PERI_USB3_DCTL	外设控制寄存器	13-179
0xC708	PERI_USB3_DEVTEEN	外设事件使能寄存器	13-183
0xC70C	PERI_USB3_DSTS	外设状态寄存器	13-185
0xC710	PERI_USB3_DGCMDDPAR	外设类命令参数寄存器	13-187



偏移地址	名称	描述	页码
0xC714	PERI_USB3_DGCM D	外设类命令寄存器	13-187
0xC718	PERI_USB3_DALEPE NA	外设 USB 端点使能寄存器	13-189
0xC810	PERI_USB3_DEPCM DPAR2N	外设端点命令参数寄存器 2	13-189
0xC814	PERI_USB3_DEPCM DPAR1N	外设端点命令参数寄存器 1	13-190
0xC818	PERI_USB3_DEPCM DPAR0N	外设端点命令参数寄存器 0	13-190
0xC81C	PERI_USB3_DEPCM DN	外设物理端点命令寄存器	13-190

13.8.5 USB3.0 寄存器描述

PERI_USB3_GSBUSCFG0

PERI_USB3_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。	0x0



Bits	Access	Name	Description	Reset
			0: 小端; 1: 大端。	
[10]	RW	descbigend	链表存取大小端选择。 0: 小端; 1: 大端。	0x0
[9:8]	RO	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。	0x0



Bits	Access	Name	Description	Reset
			0: 不使能; 1: 使能。	
[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_USB3_GSBUSCFG1

PERI_USB3_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]	RO	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]	RO	reserved	保留。	0x00



PERI_USB3_GTXTHRCFG

PERI_USB3_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]	RO	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]	RO	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, 有效值在 1-16 之间。	0x00
[15:0]	RO	reserved	保留。	0x0000

PERI_USB3_GRXTHRCFG

PERI_USB3_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]	RO	reserved	保留。	0x0
[29]	RW	usbtxpktcntsel	USB RXFIFO 门限选择, 仅在 SuperSpeed 时有效。	0x0



Bits	Access	Name	Description	Reset
			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。	
[28]	RO	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]	RO	reserved	保留。	0x00000

PERI_USB3_GCTL

PERI_USB3_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	pwrDNScale	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_USB3_EN_BUS_FILTERS 值为 1 时, 选择滤波功能; 1: 无论当 DWC_USB3_EN_BUS_FILTERS 值为多少, 关闭滤波功能。	0x0
[17]	RW	bypsssetaddr	Device 模式下 SetAddress 命令选择。 0: Host 正常发送 SetAddress 命令给 Device;	0x0



Bits	Access	Name	Description	Reset
			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。	
[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;	0x0



Bits	Access	Name	Description	Reset
			10: Device 配置; 11: 保留。	
[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, 将清除中断及复位除去以下寄存器以外的所有寄存器: -GCTL -GUCTL -GSTS -GSNPSID -GGPIO -GUID -GUSB2PHYCFGn -GUSB3PIPECTLn -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_USB3_GCTL[5:4]	0x0



Bits	Access	Name	Description	Reset
			(ScaleDown) = X1, 则关闭 U1/U2 的反应时间的 scaledown。	
[8]	RW	debugattach	<p>Debug Attach 信号</p> <p>当此位被置 1 时</p> <ul style="list-style-type: none"> -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	ramclkset	<p>RAM Clock 选择。</p> <p>00: bus clock;</p> <p>01: pipe clock;</p> <p>10: pipe/2 clock;</p> <p>11: reserved。</p> <p>说明: 当处于 host 模式时, 硬件将置此 2 位为 00, 即将 ram_clk 接 bus_clk, 因为当</p> <p>SS port 处于 P3 状态时, pipe_clk 会被关闭, USB2.0 port 将不会工作。</p>	0x0
[5:4]	RW	scaledown	<p>scale-down timing 选择。</p> <p>HS/FS/LS 模式下:</p> <p>00: 关闭掉所有的 scale-down timing, 使用实际的 timing 进行仿真;</p> <p>01: 启用除以下功能外地所有的 scale-down timing:</p> <ul style="list-style-type: none"> -speed enumeration -HNP/SRP -Host 模式 的 suspend 和 resume; <p>10: 仅仅开启 device 模式时 suspend 和 resume 功能时的 scale-down timing;</p>	0x0



Bits	Access	Name	Description	Reset
			<p>11: 打开所有的 scale-down timing。 SS 模式下 HS/FS/LS 模式下</p> <p>00: 关闭掉所有的 scale-down timing, 使用实际的 timing 进行仿真; 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。</p>	
[3]	RW	disssramble	<p>关闭加扰功能。</p> <p>1: 关闭; 0: 不关闭。</p>	0x0
[2]	RW	u2exit_lfps	<p>U2 状态退出信号。</p> <p>0: link 将把 248ns 的 LFPS 信号当做有效的 U2 退出状态信号; 1: link 层在检测到有效的 U2 退出信号前等待 8us。</p>	0x0
[1]	RW	gblhibernationen	<p>休眠使能。</p> <p>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。</p>	0x0
[0]	RW	dsblclkgtnng	<p>内部 Clock Gating 选择。</p> <p>0: 选择内部 clock gating; 1: 当 core 处于 lpm 模式时, 关闭掉内部 clock gating。</p>	0x0



Bits	Access	Name	Description	Reset
			说明：上电复位后可将此 bit 置为 1	

PERI_USB3_GSTS

PERI_USB3_GSTS 为全局状态寄存器。

Offset Address: 0xC118 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:20]	RO	cbelt	指示在 Host 模式下，所有接收到的 device BELT value 的最小值，以及 Set Latency Tolerance Value 命令设置的 BELT 值。	0x000
[19:10]	-	reserved	保留。	0x000
[9]	RO	bc_ip	指示 BCEVT 寄存器中有一个与 BC 有关的中断等待处理。 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; 1: 有指示。	0x0
[6]	RO	device_ip	指示 Device event queue 中有一个与 xHCI 有关的中断等待处理。 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	0x0



Bits	Access	Name	Description	Reset
			DWC_USB3_CSR_ACCESS_TIMEOUT 定义的时间。 0: 无指示; 1: 有指示。	
[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]	RO	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_USB3_GUCTL1

PERI_USB3_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]	RO	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_USB3_GGPIO

PERI_USB3_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_USB3_GUID

PERI_USB3_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_USB3_GUCTL

PERI_USB3_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 一个包。	0x0



Bits	Access	Name	Description	Reset
[20:18]	RW	psqextrressp	保留。	0x0
[17]	RW	sprctrltransen	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 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。	0x0
[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]	RW	extcapsupten	保留。	0x0



Bits	Access	Name	Description	Reset
[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, $T=DTFT*256*8\text{ us}$ 。	0x000

PERI_USB3_GBUSERRADDR_HI

PERI_USB3_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_USB3_GBUSERRADDR_LO

PERI_USB3_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_USB3_GPRTBIMAP_HI

PERI_USB3_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]	RW	reserved	保留	0x00000000

PERI_USB3_GPRTBIMAP_LO

PERI_USB3_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]	RO	reserved	保留。	0x00000000
[3:0]	RW	binumn	指示当前连接的 superspeed 的总线号。	0x0

PERI_USB3_GPRTBIMAP_HS_HI

PERI_USB3_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_USB3_GPRTBIMAP_HS_LO

PERI_USB3_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]	RO	reserved	保留。	0x0000000
[3:0]	RW	binumn	指示当前连接的 highspeed 的总线号。	0x0

PERI_USB3_GPRTBIMAP_FS_HI

PERI_USB3_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]	RW	reserved	保留。	0x00000000

PERI_USB3_GPRTBIMAP_FS

PERI_USB3_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]	RO	reserved	保留。	0x0000000
[3:0]	RW	binumn	指示当前连接的 fullspeed 的总线号。	0x0

PERI_USB3_GUSB2PHYCFG

PERI_USB3_GUSB2PHYCFG 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	reserved	保留	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]	RO	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]	RO	reserved	保留。	0x0
[13:10]	RW	usbtrdtim	USB2 周转时间(Turnround Time), 指 MAC 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。	0x0
[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。	0x0



Bits	Access	Name	Description	Reset
			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。	
[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 High-speed 模式: One 30-MHz PHY clock = 16 bit times	0x0



Bits	Access	Name	Description	Reset
			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_USB3_GUSB3PIPECTLN

PERI_USB3_GUSB3PIPECTLN 为全局 USB3.0 PIPE 控制寄存器

Offset Address: 0xC2C0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31]	RW	phy_soft_rst	USB3.0 软复位。 0: 不复位; 1: 复位。	0x0
[30]	RW	hstprtcmpl	保留。	0x0
[29]	RW	u2ssinactp3ok	U2/SSInactive 状态时 PHY 的状态选择。 0: PHY 进入 P2 状态; 1: PHY 进入 P3 状态。	0x0
[28]	RW	disrxdetp3	p3 状态接收检测使能。 0: 如果 PHY 处于 P3 状态, 并且 core 需要接收检测, 则 core 会在 P3 状态下进行接收检测; 1: 如果 PHY 处于 P3 状态, 并且 core 需要接收检测, 则 Core 会将 PHY 的状态转变为 P2, 再进行接收检测, 在检测完毕后, core 再将 PHY 的状态转换回 P2。	0x0
[27]	RW	ux_exit_in_px	core 状态切换时, PHY 的状态选择。 0: core 退出 U1/U2/U3 状态时处于 PHY 的 P0 状态; 1: core 退出 U1/u2/u3 状态时处于 PHY	0x0



Bits	Access	Name	Description	Reset
			的相对应的 P1/P2/P3 状态。	
[26]	RW	ping_enhancement_en	下行端口 U1 的 ping 命令的 timeout 从 500ms 变为 300ms。 0: 不变化; 1: 变化。	0x0
[25]	RW	u1u2exitfail_to_recover	P2 状态下退出 P3 的信号选择。 当此位被置 1 时, core 会在发出退出 U3 的握手信号之前, 将 PHY 的状态置为 P2。	0x0
[24]	RW	request_p1p2p3	当 core 从状态 U0 切换到 U1/U2/U3 时, core 总是会请求 PHY 从状态 P0 切换到 P1/P2/P3 0: 不切换; 1: 切换。	0x0
[23]	RW	startrxdetu3rxdet	需固定为 0。	0x0
[22]	RW	disrxdetu3rxdet	需固定为 0。	0x0
[21:19]	RW	delayp1p2p3	P0 到 P1/P2/P3 状态的延时 当 core 进入 U1/U2/U3 状态时, 延长 P0 进入 P1/P2/P3 的时间, 直到 Pipe3_RxValid 被拉为 0, 或者发生 8B10B 的错误。 说明: 第 18bit 必须被置为 1 时才有效。	0x0
[18]	RW	delay_phy_powerchange	PHY 状态切换延时寄存器。 0: 当 core 从 U0 进入 U1/U2/U3 状态时, PHY 直接进入 P1/P2/P3 状态, 不需要查看 Pipe3_RxElecIldc 和 pipe3_RxValid 的值; 1: 当 core 从 U0 进入 U1/U2/U3 状态时, PHY 到 P1/P2/P3 的状态将会被延时, 直到 Pipe3_RxElecIldc 为 1 和 pipe3_RxValid 为 0。	0x0



Bits	Access	Name	Description	Reset
[17]	RW	suspend_en	USB3.0 PHY 挂起使能。 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
[16:15]	RW	datwidth	PIPE 接口的数据位宽。 00: 32 bits; 01: 16 bits; 10: 8 bits; 其他: 保留。	0x0
[14]	RW	abortrxdetinu2	取消 U2 状态下的 Rx Detect。 0: 不取消; 1: 取消。 当此位被置 1 并且连接状态时 U2 时, core 接收到远端连接设备发送的 U2 退出信号时, 将不会采取接收检测。	0x0
[13]	RW	skiprxdet	跳过 Rx Detect 状态。 0: 不跳过; 1: 跳过。 如果此位被置 1, 当 pipe3_RxEleclidle 被拉低时, 将跳过 Rx Detection。	0x0
[12]	RW	lfps_p0_align	控制器退出 U1/U2/U3 状态时在请求 PHY P0 的信号的时钟边沿终止发送 LFPS, 否则 LFPS 信号将早一个 cycle 之前发送 当 PHY 从 P1 或 P2 状态切换到 P0 状态时, 在 PHY 设置 PhyStatus 以后 2 个时钟周期, 控制器请求传输。	0x0
[11]	RW	p3p2_tran_ok	P3/P2 状态直接切换。 0: PHY 每次进行 P2P3 质检的切换都需要	0x0



Bits	Access	Name	Description	Reset
			经过中间状态 P0; 1: PHY 的双胎直接从 P2 切换到 P3, 或者直接从 P3 切换到 P2, 不需要经过中间状态 P0。	
[10]	RW	p3exsigp2	P3 退出状态选择。 当此位为 1 时, 当 core 从 U3 退出时, PHY 的状态一定是从 P3 退出到 P2, 否则可从 P3 退出到 p1 或者 P0。	0x0
[9]	RW	lfps_filter	LFPS 过滤。 0: 不过滤; 1: 过滤。 当此位置 1 时, 控制器将过滤来自 PHY 的 LFPS 信号除非 pipe3_Rxelecidle 和 pipe3_RxValid 被撤销。	0x0
[8]	RW	polling_lfps_control	RX_DETECT 到 Polling.LFPS 控制。 0: (默认值)RX_DETECT 后延时 400us 开始 Polling LFPS; 1: RX_DETECT 后不加延时直接开始 Polling LFPS。	0x0
[7]	RO	reserved	保留。	0x0
[6]	RW	txswing	PIPE 接口发送摆幅。 0: Full swing; 1: Low swing。	0x0
[5:3]	RW	txmargin	PIPE 接口发送端余量。 000: Normal; 001: 800-1200mV Full Swing/400-700mV Half Swing; 101-111: 200-400mV Full Swing/100-200mV Half Swing。	0x0



Bits	Access	Name	Description	Reset
[2:1]	RW	txdeemphasis	PIPE 接口发送端预加重。 00: -6Db; 01: -3.5Db; 10: No Deemphasis; 11: 保留。	0x0
[0]	RW	elastic_buffer_mode	弹性 buffer 模式选择使能。 0: 禁止; 1: 使能。	0x0

PERI_USB3_GTXFIFOSIZN

PERI_USB3_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_USB3_GRXFIFOSIZN

PERI_USB3_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。	0x0000



Bits	Access	Name	Description	Reset
			最小: 32 MDWIDTH-bit words; 最大: 32768 MDWIDTH-bit words。	

PERI_USB3_GEVNTADRN_HI

PERI_USB3_GEVNTADRN_HI 为全局 event buffer 的地址高 32 位寄存器。

Offset Address: 0xC410 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RWSC	evntadrhi	Event Buffer 高 32 位地址。	0x00000000

PERI_USB3_GEVNTADRN_LO

PERI_USB3_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_USB3_GEVNTSIZN

PERI_USB3_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]	RO	reserved	保留。	0x0000
[15:0]	RW	evntsiz	Event Buffer 大小(byte)。	0x0000



PERI_USB3_GEVNTCOUNTN

PERI_USB3_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]	RO	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_USB3_GTXFIFOPRIDEV

PERI_USB3_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]	RO	reserved	保留。	0x00000000
[0]	RW	device_txfifo_priority	Device TXFIFO 优先级。 0: low; 1: high。	0x0

PERI_USB3_GTXFIFOPRIHST

PERI_USB3_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]	RO	reserved	保留。	0x00000000
[0]	RW	host_txfifo_priority	Host TXFIFO 优先级。 0: low; 1: high。	0x0



PERI_USB3_GRXFIFOPRIHST

PERI_USB3_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]	RO	reserved	保留。	0x00000000
[0]	RW	host_rxfifo_priority	Host RXFIFO 优先级。 0: low; 1: high。	0x0

PERI_USB3_GFIFOPRIDBC

PERI_USB3_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]	RO	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_USB3_GDMAHLRATIO

PERI_USB3_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]	RO	reserved	保留。	0x00000



Bits	Access	Name	Description	Reset
[12:8]	RW	hst_rxfifo_dma_hilo_priority_ratio	Host RXFIFO DMA 高-低 优先级 比例。	0x00
[7:5]	RO	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_USB3_GFLADJ

PERI_USB3_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_240mhzdecr_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]	RO	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	0x0



Bits	Access	Name	Description	Reset
			SOF/ITP 进行校准; 1: 控制器用 GFLADJ.GFLADJ_30MHZ 的值来对 SOF/ITP 进行校准。	
[6]	RO	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_USB3_DCFG

PERI_USB3_DCFG 为外设配置寄存器。

Offset Address: 0xC700 Total Reset Value: 0x0008_0004

Bits	Access	Name	Description	Reset
[31:24]	RO	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]	RO	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	0x00



Bits	Access	Name	Description	Reset
			处清 0。	
[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_USB3_DCTL

PERI_USB3_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	0x0



Bits	Access	Name	Description	Reset
			<p>作:</p> <p>1: 清除所有中断和复位除以下寄存器外的所有寄存器;</p> <ul style="list-style-type: none"> - GCTL - GUCTL - GSTS - GSNPSID - GGPIO - GUID - GUSB2PHYCFGn registers - GUSB3PIPECTLn registers - DCFG - DCTL - DEVTEN - DSTS <p>2: 所有状态机将回归 IDLE 状态, 所有的 FIFO 将清 0;</p> <p>3: SoC 总线上的所有传输将立刻终止。</p>	
[29]	RO	reserved	保留。	0x0
[28:24]	RW	hird_threshold	<p>HIRD 门限值</p> <p>core 将在满足以下的条件下产生 utmi_l1_suspend_n 和 utmi_sleep_n 信号。</p> <p>1.当满足下面的条件时, core 在 L1 状态产生 utmi_l1_suspend_n , 将 PHY 切换到深度 Low-Power:</p> <ul style="list-style-type: none"> -HIRD 的值大于或者等于 DCTL.HIRD_Thres[3:0] -HIRD_Thres[4]的值为 1 <p>2.当满足以下条件时, core 在 L1 状态将产生 utmi_sleep_n:</p> <ul style="list-style-type: none"> -HIRD 的值小于 HIRD_Thres[3:0] 	0x00



Bits	Access	Name	Description	Reset
			-HIRD_Thres[4]的值为 0	
[23]	RW	appl1res	LPM 响应配置。 当 PERI_USB3_DCFG[lpm_capable]为 0: core 没有响应; 当 PERI_USB3_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]	RO	reserved	保留。	0x0
[19]	RW	keepconnect	保持连接状态。 如果此位被置 1 时, 当 runStop =0 时, 所 有涉及模型将得到保存或被恢复。同时, 当 link 进入 U3 或者 L2 时, 将使能 Hibernation Request Event。	0x0
[18]	RW	l1_hibernation_en	当 keepconnect 为 1, 并且此位为 1, 如 果 L1 被使能 并且 HIRD 的值大于 PERI_USB3_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,	0x0



Bits	Access	Name	Description	Reset
			将 PERI_USB3_DSTS[rss]置 0。 说明：当读此比特的时候，返回 0。	
[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_USB3_DSTS[sss]置 0。 说明：当读此比特的时候，返回 0。	0x0
[15:13]	RO	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。 说明：USB 复位后，硬件自动清除此比特位。	0x0
[8:5]	WO	ulstchngraq	USB/Link 状态切换请求。	0x0



Bits	Access	Name	Description	Reset
			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。	
[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]	RO	reserved	保留。	0x0

PERI_USB3_DEVTEN

PERI_USB3_DEVTEN 为外设事件使能寄存器。

Offset Address: 0xC708 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x000000



Bits	Access	Name	Description	Reset
[9]	RW	erratic_error_event_enable	不规则错误事件使能。 0: 禁止; 1: 使能。	0x0
[8]	RO	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。	0x0



Bits	Access	Name	Description	Reset
			0: 禁止; 1: 使能。	

PERI_USB3_DSTS

PERI_USB3_DSTS 为外设状态寄存器。

Offset Address: 0xC70C Total Reset Value: 0x0012_0004

Bits	Access	Name	Description	Reset
[31:30]	RO	reserved	保留。	0x0
[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]	RO	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_USB3_DCTL[run_stop]为 1 的时候, 此位置 0; 当软件向 PERI_USB3_DCTL[run_stop]写 0 后, 并且 core 处于空闲状态以及更下一层	0x0



Bits	Access	Name	Description	Reset
			完成断开的操作, core 将对此位写 1。	
[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; 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。	0x4
[17]	RO	reserved	保留。	0x1



Bits	Access	Name	Description	Reset
[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); 100: 超速(PHY clock 工作在 125 或者 250MHz) ; 其他: 保留。	0x4

PERI_USB3_DGCMDDPAR

PERI_USB3_DGCMDDPAR 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_USB3_DGCMDD

PERI_USB3_DGCMDD 为外设类命令寄存器。

Offset Address: 0xC714 Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:16]	RO	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]	RO	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]	RO	reserved	保留。	0x0
[8]	WO	cmdloc	命令完成中断。 说明: PERI_USB3_DCTL[run_stop] 为 0 时, 此位为 1。	0x0
[7:4]	RO	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_USB3_DALEPENA

PERI_USB3_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	<p>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：</p> <p>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。</p> <p>对该寄存器操作时必须设置 bit0 和 bit1，这样 USBReset 后才会使能物理端点中的控制端点。</p> <p>当探测到 USB reset 时，应用软件会清除了输出端点 0/输入端点 0 其他所有端点对应的位。当收到设置配置和设置接口要求时，应用程序相应的设置端点寄存器以及本寄存器的这些位。</p>	0x00000000

PERI_USB3_DEPCMDPAR2N

PERI_USB3_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	<p>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。</p>	0x00000000



PERI_USB3_DEPCMDPAR1N

PERI_USB3_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_USB3_DEPCMDPAR0N

PERI_USB3_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_USB3_DEPCMDN

PERI_USB3_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]: 表示为本次传输声明的 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	0x0000



Bits	Access	Name	Description	Reset
			(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(XferRscldx)。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。	
[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。但是这并不能表明先前发送的命令都已实现。	0x0
[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_USB3_DCTL[run_stop]=0 时，该 bit 不能设置为 1。	0x0
[7:4]	RO	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)；	0x0



Bits	Access	Name	Description	Reset
			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。	

13.9 LSADC

13.9.1 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, 4 个独立通道;

13.9.2 特点

LSADC 具有以下特点:

- 电源电压 3.3V;
- 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;
- 支持 16 次平均算法模式和 32 次平均算法模式;
- 扫描完成自动上报中断。
- 支持多通道扫描防串扰

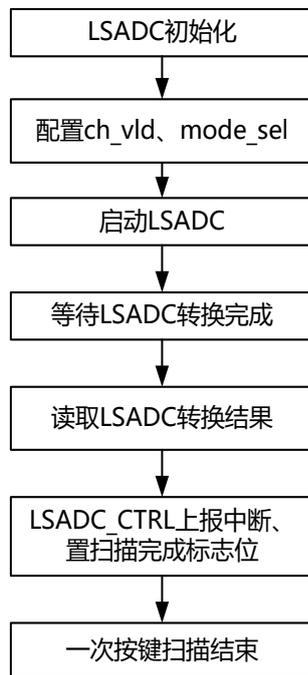


13.9.3 工作方式

单次扫描处理流程

在单次读取模式 (LSADC_CTRL0 [mode_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13-38 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 [mode_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。

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 13-39 所示。



图13-39 连续扫描模式下通道轮询扫描示意图

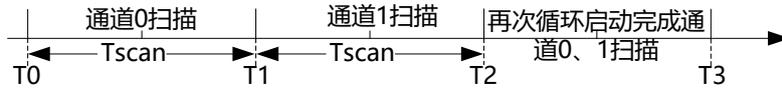
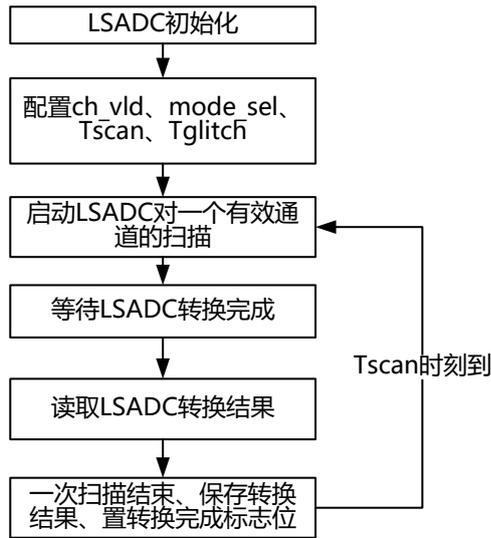


图13-40 连续扫描处理流程



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。

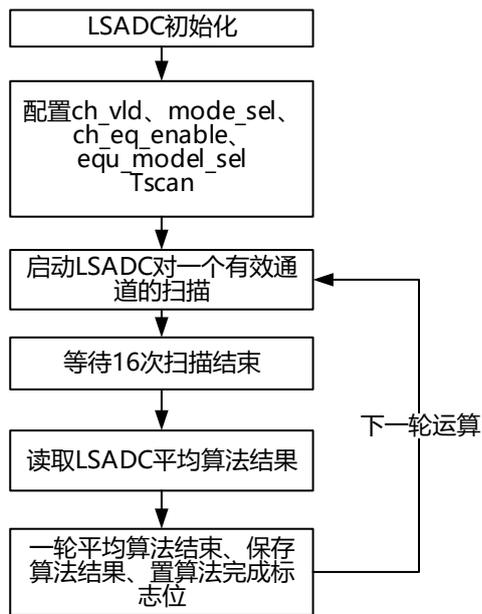
平均算法设置

通过 LSADC_CTRL9 [12]可以选择通道采样的平均算法模式，可以根据应用需要设置对应的平均算法模式。请参考 LSADC_CTRL0 寄存器的描述。

- 单次读取模式，不支持采样平均算法操作。
- 连续扫描模式，可选择 16 次或 32 次平均算法。

以使能 16 次平均算法模式为例，连续扫描模式下通道平均算法示意图如图 13-41 所示。

图13-41 16 次平均算法处理流程



防串扰模式配置

连续扫描配置基础上，可通过配置如下寄存器防止通道间串扰。

- 配置使能防串扰模式：LSADC_CTRL0[24]和 LSADC_CTRL0[16]为 1；
- 配置防串扰周期：LSADC_CTRL0[28:25]和 LSADC_CTRL0[7:4]。(注：LSADC_CTRL0[7:4]值需要大于等于 LSADC_CTRL0[28:25])。



13.9.4 LSADC 寄存器概览

LSADC 寄存器概览如表 13-21 所示。

表13-21 LSADC 寄存器概览 (基地址: 0x0_1110_0000)

偏移地址	名称	描述	页码
0x0000	LSADC_CTRL0	LSADC 配置寄存器	13-197
0x0004	LSADC_CTRL1	滤毛刺配置寄存器	13-200
0x0008	LSADC_CTRL2	扫描间隔配置寄存器	13-200
0x0010	LSADC_CTRL4	中断使能寄存器	13-200
0x0014	LSADC_CTRL5	中断状态寄存器	13-201
0x0018	LSADC_CTRL6	中断清除寄存器	13-202
0x001C	LSADC_CTRL7	Start 配置寄存器	13-202
0x0020	LSADC_CTRL8	Stop 配置寄存器	13-203
0x0024	LSADC_CTRL9	转换结果精度寄存器	13-203
0x0028	LSADC_CTRL10	LSADC_ZERO 寄存器	13-203
0x002C	LSADC_CTRL11	LSADC 通道 0 数据寄存器	13-204
0x0030	LSADC_CTRL12	LSADC 通道 1 数据寄存器	13-204
0x0034	LSADC_CTRL13	LSADC 通道 2 数据寄存器	13-204
0x0038	LSADC_CTRL14	LSADC 通道 3 数据寄存器	13-205
0x0050	LSADC_CTRL20	LSADC 通道 0 平均算法数据寄存器	13-205
0x0054	LSADC_CTRL21	LSADC 通道 1 平均算法数据寄存器	13-205
0x0058	LSADC_CTRL22	LSADC 通道 2 平均算法数据寄存器	13-206
0x005C	LSADC_CTRL23	LSADC 通道 3 平均算法数据寄存器	13-206



13.9.5 LSADC 寄存器描述

LSADC_CTRL0

LSADC_CTRL0 为 LSADC 配置寄存器。

Offset Address: 0x0000 Total Reset Value: 0x0000_800F

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	保留	0x00
[28:25]	RW	channel_scan_time	防串扰周期选择。 单个时钟周期为：333.3ns。 0x0：15 个时钟周期； 0x1：20 个时钟周期； 0x2：25 个时钟周期； 0x3：30 个时钟周期； 0x4：35 个时钟周期； 0x5：40 个时钟周期； 0x6：45 个时钟周期； 0x7：50 个时钟周期； 0x8：60 个时钟周期； 0x9：70 个时钟周期； 0xa：80 个时钟周期； 0xb：100 个时钟周期； 0xc：150 个时钟周期； 其他：15 个时钟周期。	0x0
[24]	RW	channel_scan_sel	防串扰使能配置 bit[1]。 1：使能 0：不使能	0x0
[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	0x0



Bits	Access	Name	Description	Reset
			果相同)。	
[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: 不使能滤毛刺功能。	0x0
[16]	RW	channel_scan_mode_sel	防串扰使能配置 bit[0]。 1: 使能 0: 不使能	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]	RW	equ_model_sel	LSADC 平均算法模式选择。 0: 16 次平均算法模式; 1: 32 次平均算法模式。	0x0
[11]	RW	ch_3_vld	LSADC 通道 3 是否有效。 0: 无效; 1: 有效。	0x0
[10]	RW	ch_2_vld	LSADC 通道 2 是否有效。 0: 无效; 1: 有效。	0x0
[9]	RW	ch_1_vld	LSADC 通道 1 是否有效。 0: 无效;	0x0



Bits	Access	Name	Description	Reset
			1: 有效。	
[8]	RW	ch_0_vld	LSADC 通道 0 是否有效。 0: 无效; 1: 有效。	0x0
[7:4]	RW	cnt_cs_time	防串扰模式下系统连续扫描周期选择。 单个时钟周期为: 333.3ns。 0x0: 20 个时钟周期; 0x1: 25 个时钟周期; 0x2: 30 个时钟周期; 0x3: 35 个时钟周期; 0x4: 40 个时钟周期; 0x5: 45 个时钟周期; 0x6: 50 个时钟周期; 0x7: 60 个时钟周期; 0x8: 70 个时钟周期; 0x9: 80 个时钟周期; 0xa: 100 个时钟周期; 0xb: 150 个时钟周期; 0xc: 200 个时钟周期; 其他: 20 个时钟周期。	0x0
[3]	RW	ch_3_eq_enable	LSADC 通道 3 采样平均算法是否有效。 0: 无效; 1: 有效。	0x1
[2]	RW	ch_2_eq_enable	LSADC 通道 2 采样平均算法是否有效。 0: 无效; 1: 有效。	0x1
[1]	RW	ch_1_eq_enable	LSADC 通道 1 采样平均算法是否有效。 0: 无效;	0x1



Bits	Access	Name	Description	Reset
			1: 有效。	
[0]	RW	ch_0_eq_enable	LSADC 通道 0 采样平均算法是否有效。 0: 无效; 1: 有效。	0x1

LSADC_CTRL1

LSADC_CTRL1 为滤毛刺配置寄存器。

Offset Address: 0x0004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	glitch_sample	滤毛刺时间窗口(当 LSADC 的转换结果在此时间窗口内保持不变时, 则认为此转换结果为有效值, 否则则认为是毛刺, 此窗口值一般设为 ms 级, 在连续扫描模式下不能配置为 0)。	0x00000000

LSADC_CTRL2

LSADC_CTRL2 为扫描间隔配置寄存器。

Offset Address: 0x0008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	time_scan	在连续扫描模式下, 两个扫描通道之间连续扫描的时间间隔。时间间隔为 $Tscan=time_scan*0.333us$ (连续扫描时间间隔应该大于 LSADC 的转换时间, 即配置值不能少于 0x20)。	0x00000000

LSADC_CTRL4

LSADC_CTRL4 为中断使能寄存器。



Offset Address: 0x0010 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	保留。	0x00000000
[0]	RW	int_enable	扫描值有效中断使能位。 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。	0x00000000
[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]	RO	int_flag_in3	通道 3 扫描值有效中断标志位。 0: 无中断; 1: 有中断。	0x0
[2]	RO	int_flag_in2	通道 2 扫描值有效中断标志位。 0: 无中断; 1: 有中断。	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。	0x0



Bits	Access	Name	Description	Reset
			0: 无中断; 1: 有中断。	

LSADC_CTRL6

LSADC_CTRL6 为中断清除寄存器。

Offset Address: 0x0018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	保留。	0x00000000-
[3]	WO	clr_int_flag_in3	通道 3 中断清除寄存器。 0: 不清除; 1: 清除中断。	0x0
[2]	WO	clr_int_flag_in2	通道 2 中断清除寄存器。 0: 不清除; 1: 清除中断。	0x0
[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]	WO	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)。	0x00000000

LSADC_CTRL8

LSADC_CTRL8 为 Stop 配置寄存器。

Offset Address: 0x0020 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	WO	stop	停止自动扫描(在自动扫描模式下, 向此寄存器写任意值都可以停止 LSADC 的自动扫描功能, 需再写使能 start 才可以重新启动自动扫描)。	0x00000000

LSADC_CTRL9

LSADC_CTRL9 为转换结果精度寄存器。

Offset Address: 0x0024 Total Reset Value: 0x0000_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 通道 0 数据寄存器。

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]	RO	lsadc_data_in1	LSADC 通道 1 扫描值。	0x000

LSADC_CTRL13

LSADC_CTRL13 为 LSADC 通道 2 数据寄存器。

Offset Address: 0x0034 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	保留。	0x000000
[9:0]	RO	lsadc_data_in0	LSADC 通道 2 扫描值。	0x000



LSADC_CTRL14

LSADC_CTRL14 为 LSADC 通道 3 数据寄存器。

Offset Address: 0x0038 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 3 扫描值。	0x000

LSADC_CTRL20

LSADC_CTRL20 为 LSADC 通道 0 平均算法数据寄存器。

Offset Address: 0x0050 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15]	RO	lsadc_equ_value_valid_0	LSADC 通道 0 采样平均算法结果有效指示。 0: 无效; 1: 有效。	0x0
[11:10]	-	reserved	保留。	0x0
[9:0]	RO	lsadc_equ_value_0	LSADC 通道 0 采样平均算法结果。	0x000

LSADC_CTRL21

LSADC_CTRL21 为 LSADC 通道 1 平均算法数据。

Offset Address: 0x0054 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15]	RO	lsadc_equ_value_valid_1	LSADC 通道 1 采样平均算法结果有效指示。	0x0



Bits	Access	Name	Description	Reset
			0: 无效; 1: 有效。	
[11:10]	-	reserved	保留。	0x0
[9:0]	RO	lsadc_equ_value_1	LSADC 通道 1 采样平均算法结果。	0x000

LSADC_CTRL22

LSADC_CTRL22 为 LSADC 通道 2 平均算法数据寄存器。

Offset Address: 0x0058 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15]	RO	lsadc_equ_value_v alid_2	LSADC 通道 2 采样平均算法结果有效指 示。 0: 无效; 1: 有效。	0x0
[11:10]	-	reserved	保留。	0x0
[9:0]	RO	lsadc_equ_value_2	LSADC 通道 2 采样平均算法结果。	0x000

LSADC_CTRL23

LSADC_CTRL23 为 LSADC 通道 3 平均算法数据寄存器。

Offset Address: 0x005C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15]	RO	lsadc_equ_value_v alid_3	LSADC 通道 3 采样平均算法结果有效指 示。 0: 无效;	0x0



Bits	Access	Name	Description	Reset
			1: 有效。	
[11:10]	-	reserved	保留。	0x0
[9:0]	RO	lsadc_equ_value_3	LSADC 通道 3 采样平均算法结果。	0x000

13.10 PWM

13.10.1 概述

PWM 为脉冲宽度调节信号模块，用来输出周期性脉冲信号。

芯片提供 3 组 PWM 控制器：

- PWM0：最多提供 6 路脉冲宽度调制信号输出。其中 PWM_OUT3 支持三对互补信号输出，PWM_OUT0/PWM_OUT4/PWM_OUT5 支持一对互补信号输出。
(注：PWM0 对应的 6 路输出为 PWM_OUT0~PWM_OUT5)。
- PWM1：最多提供 6 路脉冲宽度调制信号输出。其中 PWM1_OUT0/PWM1_OUT3/PWM1_OUT5 支持一对互补信号输出。
- PWM2：最多提供 5 路脉冲宽度调制信号输出。

13.10.2 特点

对于每组 PWM 控制器：

- 工作时钟
 - PWM0/PWM1 支持 198MHz/Sensor clk0~3(详见系统章节 PERI_CRG4450/PERI_CRG4452)切换。
 - PWM2 固定 198MHz。
- 支持预分频，分频比可配，最大支持 256 分频。
- 每路 PWM 输出支持独立使能。如：PWM_OUT3_xxx 为一路输出。
- 每路 PWM 输出中，每对信号仅占空比支持独立配置；如：
PWM_OUT3_0_P/PWM_OUT3_0_N 为 PWM_OUT3_xxx 该路输出的一对信号，它们互为反相输出，相位差为 180°。
- 支持周期方波输出模式，周期可配。



- 支持固定数目方波输出模式，方波数目可配。
- 支持脉冲信号左对齐、右对齐、中间对齐。
- 支持脉冲信号极性可控。
- 支持周期、占空比、对齐模式、极性等配置动态更新。
- 支持多路 PWM 信号同步输出，支持单次同步输出和循环输出两种同步模式。
- 支持占空比配置范围：0%~100%（占空比 0%一直是低电平，占空比 100%一直是高电平，配置流程如 13.10.4 配置流程章节所示）。
- 支持周期配置范围：
 - Pad name 为 MIPI_RX_xxx，管脚复用为 PWM 功能，PWM 周期配置范围：1Hz~9.9MHz。
 - Pad name 为非 MIPI_RX_xxx 的情况下，管脚复用为 PWM 功能，PWM 周期配置范围：1Hz~66MHz。

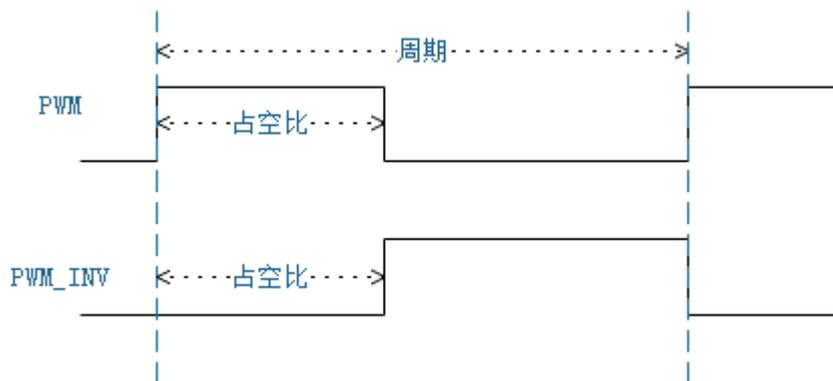
13.10.3 工作方式

13.10.3.1 对齐模式

PWM 输出的脉冲信号支持三种模式：左对齐，右对齐和中间对齐。可以通过配置 PWM(x)_ctrl. pwm(x)_align_mode_cfg 来设置每路 PWM 输出的脉冲对齐模式。

左对齐模式

图13-42 左对齐输出波形



将 PWM(x)_ctrl. pwm(x)_align_mode_cfg 配置为 1。左对齐模式。

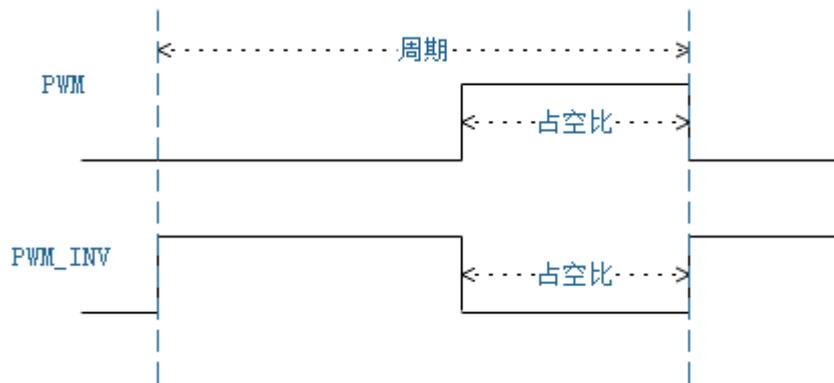


该模式下，PWM 先根据占空比配置输出有效信号电平，再输出无效信号电平直到配置的周期输出完成。

有效信号电平默认为高，并可以通过配置 `PWM(x)_ctrl.pwm(x)_inv_cfg` 来控制。当 `PWM(x)_ctrl.pwm(x)_inv_cfg` 配置为 1 时，有效信号电平为低，对应的波形为 `PWM_INV`。

右对齐模式

图13-43 右对齐输出波形



将 `PWM(x)_ctrl.pwm(x)_align_mode_cfg` 配置为 0，PWM 输出为右对齐模式。该模式下，PWM 先输出无效信号电平，再根据占空比配置输出有效信号电平。

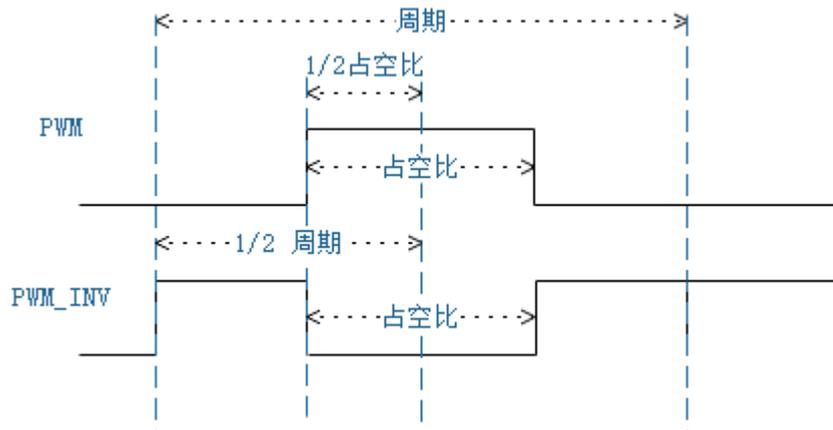
有效信号电平默认为高，并可以通过配置 `PWM(x)_ctrl.pwm(x)_inv_cfg` 来控制。当 `PWM(x)_ctrl.pwm(x)_inv_cfg` 配置为 1 时，有效信号电平为低，对应的波形为 `PWM_INV`。

中间对齐模式

须知

中间对齐模式下仅支持占空比和周期数为偶数。

图13-44 中间对齐输出波形



将 `PWM(x)_ctrl.pwm(x)_align_mode_cfg` 配置为 2，PWM 输出为中间对齐模式。

PWM 先输出无效信号电平，再按占空比配置输出有效信号电平，最后再输出无效信号电平。整个周期中，有效信号电平位于中央。

有效信号电平默认为高，并可以通过配置 `PWM(x)_ctrl.pwm(x)_inv_cfg` 来控制。当 `PWM(x)_ctrl.pwm(x)_inv_cfg` 配置为 1 时，有效信号电平为低，对应的波形为 PWM_INV。

13.10.3.2 互补输出

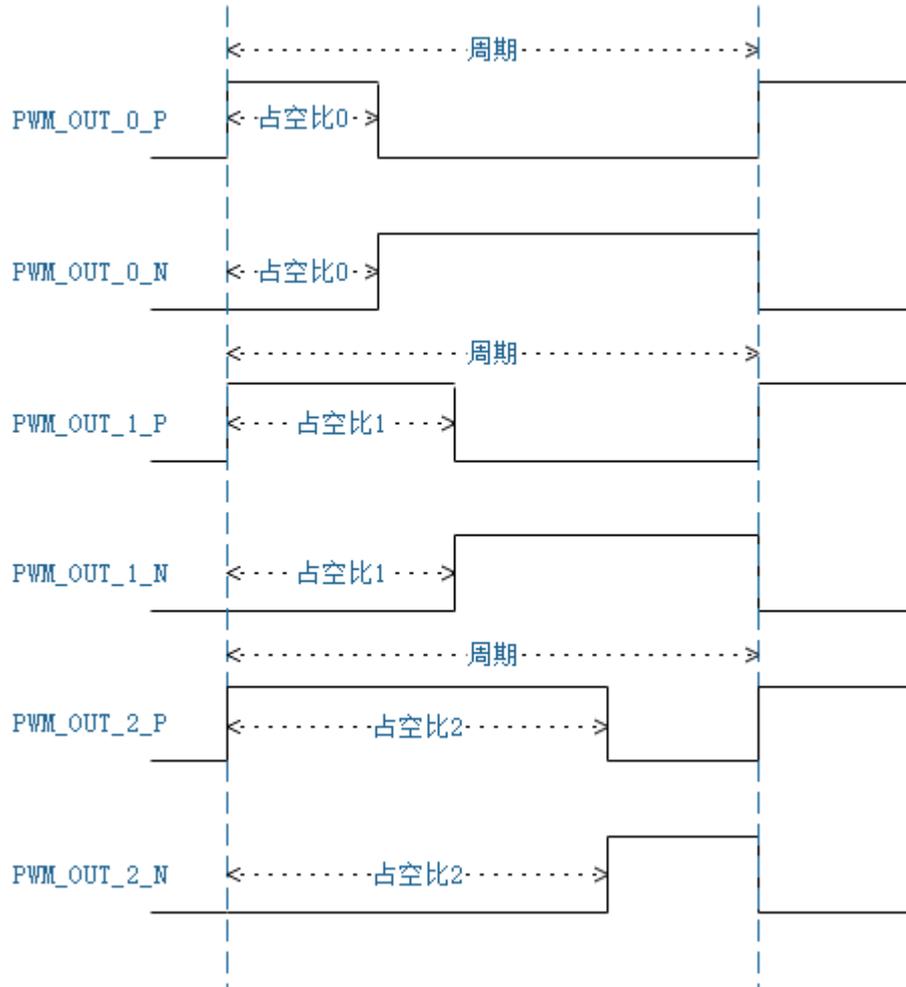
每路 PWM 最多可以支持三对互补信号输出，信号为 `PWM_OUT(x)_0_P/ PWM_OUT(x)_0_N/ PWM_OUT(x)_1_P/ PWM_OUT(x)_1_N/ PWM_OUT(x)_2_P/ PWM_OUT(x)_2_N`。其中 P/N 为互补信号，互为反相输出。

三对互补信号共用一个脉冲周期配置，通过 `PWM(x)_period_cfg` 来配置。每对信号的占空比支持各自独立配置，分别通过 `PWM(x)_duty0_cfg`、`PWM(x)_duty1_cfg`、`PWM(x)_duty2_cfg` 来控制。

其输出波形如图 13-45：

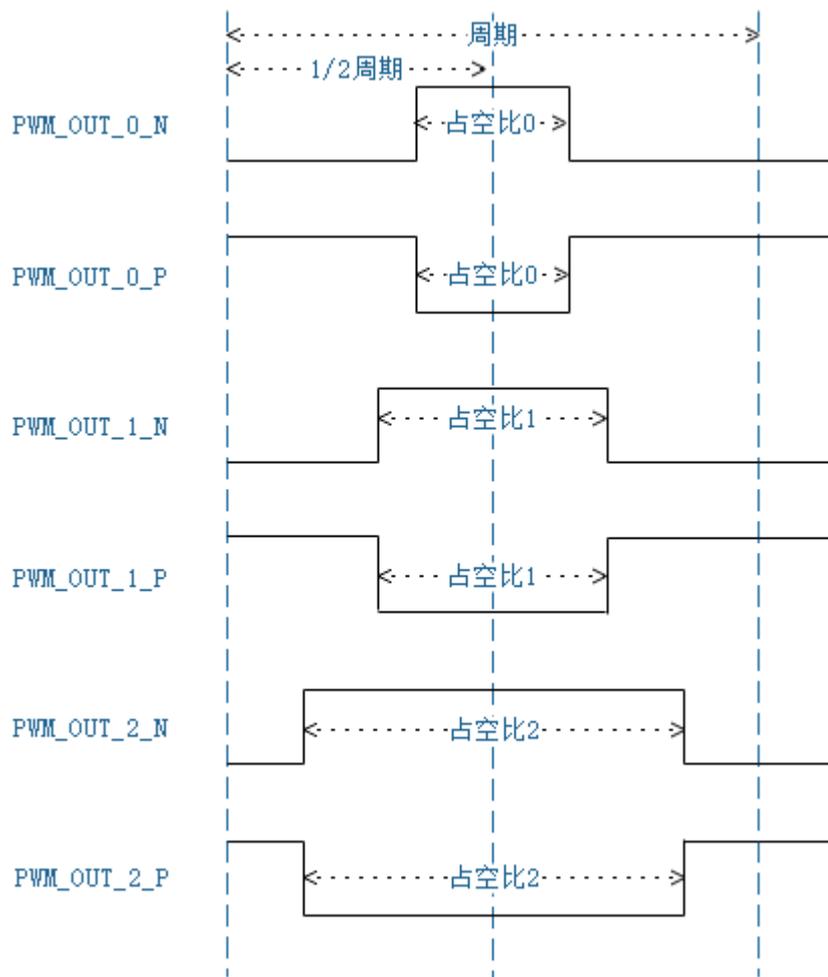


图13-45 三对互补信号输出波形



同时也可以支持与对齐模式的配置组合，如配置成中间对齐模式的时候，输出波形如图 13-46:

图13-46 三对互补信号中间对齐时输出波形



13.10.3.3 死区控制

互补信号输出模式还支持死区控制。即当输出信号从无效状态转变成有效状态时，延时若干个时钟周期后生效。

每路 PWM 提供两组死区延时时间配置寄存器

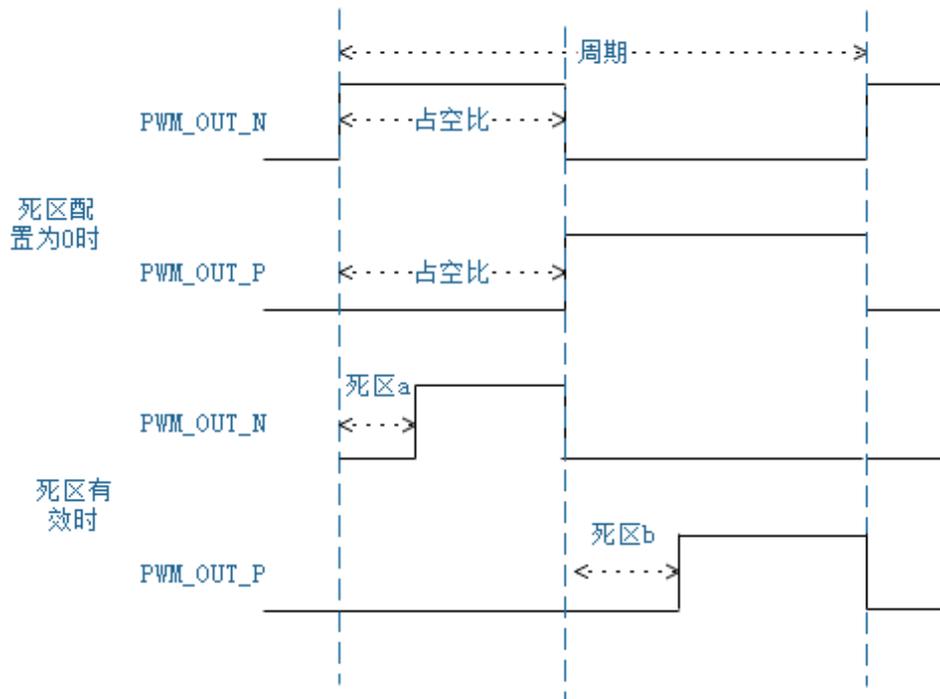
(PWM(x)_dt_value_cfg.PWM(x)_dt_a_cfg/PWM(x)_dt_value_cfg.PWM(x)_dt_b_cfg)

。当死区时间配置为 0 时，即无死区延时。每一个 PWM 信号都可以通过寄存器 PWM(x)_dt_ctrl_cfg 配置来选择各自的死区时间。

死区控制打开后的互补信号输出的波形请参考图 13-47。



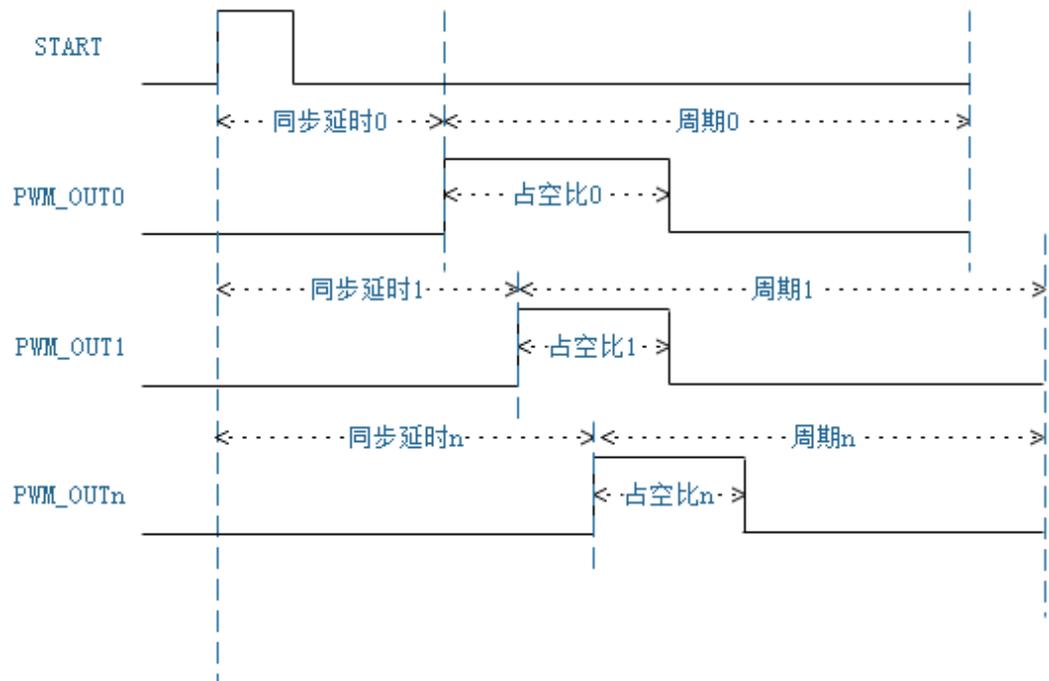
图13-47 死区控制输出波形



13.10.3.4 同步模式

每个 PWM 控制器提供的多路 PWM 信号进行同步输出，各路同步延时独立可配。
如图 13-48。

图13-48 同步模式输出波形



从起始点 START 开始，各路 PWM 根据 PWM(x)_sync_delay_cfg 寄存器配置来延时启动各自的 PWM。每路 PWM 的周期，占空比，对齐方式等可以独立配置。

PWM 同步模式下支持触发模式和循环输出两种模式。

- 触发模式（适用于输出若干个脉冲信号）

每路 PWM 输出信号完成配置脉冲输出信号数目后，则停止输出。直到下一次 START 启动为止。

- 循环模式（周期性输出脉冲信号）

每路 PWM 通过 PWM(x)_sync_cfg.pwm(x)_sync_mux 来配置同步源头，当同步源头对应的 PWM 信号完成所配置 PWM 输出后，相关同步每一路 PWM 都会进行一次重载，重新进行 START 状态，以此实现周期性的输出。

13.10.4 配置流程

13.10.4.1 通用模式

配置流程如下：



- 步骤 1 配置对应的 CRG 寄存器 (PERI_CRG4450/PERI_CRG4452/PERI_CRG4454) , 打开时钟门控。
- 步骤 2 选择合适的预分频比, 并写入寄存器 `PWM(x)_ctrl[11:8]`。
- 步骤 3 通过计算得到需要的周期数和高电平拍数, 并写入寄存器 `PWM(x)_duty0_cfg/PWM(x)_duty1_cfg/PWM(x)_duty2_cfg`、`PWM(x)_period`。
- 步骤 4 如果需要周期性输出脉冲信号, 将寄存器 `PWM(x)_ctrl[2]`位域配置为 1。如果需要输出固定个数脉冲信号, 将寄存器 `PWM(x)_ctrl[2]`配置为 0, 并将脉冲个数写入寄存器 `PWM(x)_num_cfg`。
- 步骤 5 根据是否需要死区控制, 配置寄存器 `PWM(x)_dt_value_cfg` 和 `PWM(x)_dt_ctrl_cfg`。
- 步骤 6 根据输出脉冲信号对齐方式配置寄存器 `PWM(x)_ctrl[5:4]`。
- 步骤 7 根据输出脉冲信号极性需求配置寄存器 `PWM(x)_ctrl[1]`。
- 步骤 8 配置脉冲使能寄存器 `PWM(x)_ctrl[0]`, 写入 1, 使能 PWM 输出。

---结束

例如: 需要输出 1 个频率为 4KHz, 高电平占 80% (即占空比), 脉冲个数为 10 的波形, 脉冲右对齐。

选择预分频系数为 2 分频, 周期数配置为 $198\text{MHz}/2*1/4\text{KHz}=24750$, 四舍五入后为 24750, 十六进制为 0x00060AE。高电平数配置为 24750 (周期数) \times 80% (占空比) =19800, 四舍五入后为 19800, 十六进制为 0x0004D58。

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 配置对应的 CRG 寄存器 (如 PERI_CRG4450[4]) 写入 0x1, 打开 PWM 时钟。
- 步骤 2 配置 `PWM(x)_ctrl` 寄存器为 0x100。
- 步骤 3 向 `PWM(x)_period_cfg` 写入 0x00060AD。
- 步骤 4 向 `PWM(x)_duty0_cfg` 写入 0x0004D57。
- 步骤 5 向 `PWM(x)_num_cfg` 写入 0x9 (仅更新使能 bit)。(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 6 向 `PWM(x)_ctrl` 写入 0x101 (仅更新 bit[0])。(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 `PWM(x)_ctrl_st` bit[0], 等待 bit0 位为 1 (表示 PWM 正在输出方波)



步骤 8 读取 `PWM(x)_period` 和 `0x00060AD` 进行校验。

步骤 9 读取 `PWM(x)_duty0` 和 `0x0004D57` 进行校验。

步骤 10 读取 `PWM(x)_num` bit[15:0]和 `0x0A` 进行校验(当 `PWM(x)_ctrl_st` bit[0]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

13.10.4.2 占空比 0%和 100%配置方法

占空比 100%配置流程

例如: 需要输出 1 个高电平。

选择预分频系数为 2 分频, 周期数配置为 $198\text{MHz}/2*1/4\text{KHz}=24750$, 四舍五入后为 24750, 十六进制为 `0x00060AE`。高电平数配置为 24750 (周期数) $\times 100\%$ (占空比) $=24750$, 四舍五入后为 24750, 十六进制为 `0x00060AE`。

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 配置对应的 CRG 寄存器 (如 `PERI_CRG4450 [4]`) 写入 `0x1`, 打开 PWM 时钟。

步骤 2 配置 `PWM(x)_ctrl` 寄存器为 `0x100`。

步骤 3 向 `PWM(x)_period_cfg` 写入 `0x00060AD`。

步骤 4 向 `PWM(x)_duty0_cfg` 写入 `0x00060AD`。

步骤 5 向 `PWM(x)_num_cfg` 写入 `0x9` (仅更新使能 bit) 。

步骤 6 向 `PWM(x)_ctrl` 写入 `0x101` (仅更新 bit[0]) 。

----结束

占空比 0%配置流程

例如: 需要输出 1 个低电平。

选择预分频系数为 2 分频, 周期数配置为 $198\text{MHz}/2*1/4\text{KHz}=24750$, 四舍五入后为 24750, 十六进制为 `0x00060AE`。低电平数配置为 24750 (周期数) $\times 100\%$ (占空比) $=24750$, 四舍五入后为 24750, 十六进制为 `0x00060AE`。

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 配置对应的 CRG 寄存器（如 PERI_CRG4450[4]）写入 0x1，打开 PWM 时钟。
- 步骤 2 配置 PWM(x)_ctrl 寄存器为 0x102。（配置 PWM 正相输出占空比为 100%，即 PWM 反相输出占空比为 0%。这里配置 pwm_x_inv_cfg 寄存器为 1，PWM 输出反相）
- 步骤 3 向 PWM(x)_period_cfg 写入 0x00060AD。
- 步骤 4 向 PWM(x)_duty0_cfg 写入 0x00060AD。
- 步骤 5 向 PWM(x)_num_cfg 写入 0x9（仅更新使能 bit）。
- 步骤 6 向 PWM(x)_ctrl 写入 0x103（仅更新 bit[0]）。

----结束

13.10.4.3 同步模式

同步模式配置流程如下：

- 步骤 1 配置 PWM 同步模式，PWM(x)_sync_cfgbit[5]配置为 0x1。
- 步骤 2 根据触发模式或循环模式配置寄存器。

触发模式：PWM(x)_sync_cfg bit[4]配置为 0。

循环模式：PWM(x)_sync_cfg bit[5]配置为 1。并根据需要配置 PWM 重载信号源头，配置 PPWM(x)_sync_cfg bit[3:0]

- 步骤 3 配置 PWM 信号的模式（同通用模式的步骤 1~7）
- 步骤 4 配置 PWM_sync_start 寄存器的对应 bit 启动 PWM。

例如：需要同步启动 PWM0、PMW1、PMW5 的话，则在 PWM_sync_start 寄存器写入 0x23。

----结束

13.10.5 寄存器偏移地址变量表寄存器概览

PWM0 寄存器基地址：0x0_1108_0000。

PWM1 寄存器基地址：0x0_1108_1000。

PWM2 寄存器基地址：0x0_1108_2000。

各模块的寄存器偏移地址中变量的取值范围和含义如表 13-22 所示。



表13-22 各模块的寄存器偏移地址变量表

变量名称	取值范围	描述
x	0 ~ 5	PWM 模块的 6 个通道

13.10.6 PWM 寄存器概览

PWM 寄存器概览如表 13-23 所示。

表13-23 PWM 寄存器概览

偏移地址	名称	描述	页码
0x0000 + 0x100×x	PWM(x)_period_cfg	PWM(x)周期配置寄存器	13-219
0x0004 + 0x100×x	PWM(x)_duty0_cfg	PWM(x)占空比 0 配置寄存器	13-220
0x0008 + 0x100×x	PWM(x)_duty1_cfg	PWM(x)占空比 1 配置寄存器	13-220
0x000C + 0x100×x	PWM(x)_duty2_cfg	PWM(x)占空比 2 配置寄存器	13-220
0x0010 + 0x100×x	PWM(x)_num_cfg	PWM(x)输出方波数目配置寄存器	13-221
0x0014 + 0x100×x	PWM(x)_ctrl	PWM(x)工作模式配置寄存器	13-221
0x0020 + 0x100×x	PWM(x)_dt_value_ cfg	PWM(x)死区时间配置寄存器	13-222
0x0024 + 0x100×x	PWM(x) _dt_ctrl_cfg	PWM(x)死区时间选择配置寄存器	13-222
0x0030 + 0x100×x	PWM(x)_sync_cfg	PWM(x)同步模式配置寄存器	13-223



偏移地址	名称	描述	页码
0x0034 + 0x100×x	PWM(x)_sync_delay_cfg	PWM(x)同步延时配置寄存器	13-224
0x0040 + 0x100×x	PWM(x)_period	PWM(x)实时周期回读寄存器	13-224
0x0044 + 0x100×x	PWM(x)_duty0	PWM(x)实时占空比 0 回读寄存器	13-225
0x0048 + 0x100×x	PWM(x)_duty1	PWM(x)实时占空比 1 回读寄存器	13-225
0x004C + 0x100×x	PWM(x)_duty2	PWM(x)实时占空比 2 回读寄存器	13-225
0x0050 + 0x100×x	PWM(x)_num	PWM(x)方波数量实时回读寄存器。	13-225
0x0054 + 0x100×x	PWM(x)_ctrl_st	PWM(x)实时工作状态回读寄存器	13-226
0x0060 + 0x100×x	PWM(x)_dt_value	PWM(x)实时死区时间回读寄存器	13-227
0x0064 + 0x100×x	PWM(x)_dt_ctrl	PWM(x)实时死区时间选择回读寄存器	13-227
0x0074 + 0x100×x	PWM(x)_sync_delay	PWM(x)实时同步延时配置回读寄存器	13-228
0xFF0	PWM_sync_start	PWM 同步启动使能寄存器	13-229

13.10.7 PWM 寄存器描述

PWM(x)_period_cfg

PWM(x)_period_cfg 为 PWM(x)周期配置寄存器。

Offset Address: 0x0000 + 0x100×x Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	pwm(x)_period_cfg	PWM x 周期配置。单位：时钟周期，实际值减一配置。	0x00000000

PWM(x)_duty0_cfg

PWM(x)_duty0_cfg 为 PWM(x)占空比 0 配置寄存器。

Offset Address: 0x0004 + 0x100×x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	pwm(x)_duty0_cfg	PWM x 有效信号占空比 0 配置。单位：时钟周期，实际值减一配置。	0x00000000

PWM(x)_duty1_cfg

PWM(x)_duty1_cfg 为 PWM(x)占空比 1 配置寄存器。

Offset Address: 0x0008 + 0x100×x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	pwm(x)_duty1_cfg	PWM x 有效信号占空比 1 配置。单位：时钟周期，实际值减一配置。	0x00000000

PWM(x)_duty2_cfg

PWM(x)_duty2_cfg 为 PWM(x)占空比 2 配置寄存器。

Offset Address: 0x000C + 0x100×x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	pwm(x)_duty2_cfg	PWM x 有效信号占空比 2 配置。单位：时钟周期，实际值减一配置。	0x00000000



PWM(x)_num_cfg

PWM(x)_num_cfg 为 PWM(x)输出方波数目配置寄存器。

Offset Address: 0x0010 + 0x100×x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	保留。	0x0000
[15:0]	RW	pwm(x)_num_cfg	PWM x 输出方波的数目配置。实际值减一配置。 只在 pwm(x)_keep 配置为 0 时才生效。	0x0000

PWM(x)_ctrl

PWM(x)_ctrl 为 PWM(x)工作模式配置寄存器。

Offset Address: 0x0014 + 0x100×x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:12]	-	reserved	保留。	0x00000
[11:8]	RW	pwm(x)_pre_div_sel_cfg	PWM x 预分频控制。 0x0: 不分频; 0x1: 2 分频; 0x2: 4 分频; 0x3: 8 分频; 0x4: 16 分频; 0x5: 32 分频; 0x6: 64 分频; 0x7: 128 分频; 0x8: 256 分频; 其他: 不分频。	0x0
[7:6]	-	reserved	保留。	0x0
[5:4]	RW	pwm(x)_align_mo	PWM x 对齐模式控制。	0x0



Bits	Access	Name	Description	Reset
		de_cfg	00: 右对齐模式; 01: 左对齐模式; 10: 中间对齐模式; 其他: 保留。	
[3]	-	reserved	保留。	0x0
[2]	RW	pwm(x)_keep_cfg	PWM x 输出模式控制。 0: 输出 pwm(x)_num 个方波; 1: 一直输出方波。	0x0
[1]	RW	pwm(x)_inv_cfg	PWM x 输出正反相控制。 0: 正常; 1: 反相。	0x0
[0]	RW	pwm(x)_enable	PWM x 使能控制。 0: 关闭; 1: 使能。	0x0

PWM(x)_dt_value_cfg

PWM(x)_dt_value_cfg 为 PWM(x) 死区时间配置寄存器。

Offset Address: 0x0020 + 0x100×x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	pwm(x)_dt_b_cfg	PWM x 死区时间 B, 单位周期。0 代表无死区。	0x0000
[15:0]	RW	pwm(x)_dt_a_cfg	PWM x 死区时间 A, 单位周期。0 代表无死区。	0x0000

PWM(x)_dt_ctrl_cfg

PWM(x)_dt_ctrl_cfg 为 PWM(x) 死区时间选择配置寄存器。



Offset Address: 0x0024 + 0x100×x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x00000000
[5]	RW	pwm(x)_dts_out_2_n_cfg	PWM_OUT(x)_2_N 死区时间选择控制： 0: 选择死区时间 A； 1: 选择死区时间 B。	0x0
[4]	RW	pwm(x)_dts_out_2_p_cfg	PWM_OUT(x)_2_P 死区时间选择控制： 0: 选择死区时间 A； 1: 选择死区时间 B。	0x0
[3]	RW	pwm(x)_dts_out_1_n_cfg	PWM_OUT(x)_1_N 死区时间选择控制： 0: 选择死区时间 A； 1: 选择死区时间 B。	0x0
[2]	RW	pwm(x)_dts_out_1_p_cfg	PWM_OUT(x)_1_P 死区时间选择控制： 0: 选择死区时间 A； 1: 选择死区时间 B。	0x0
[1]	RW	pwm(x)_dts_out_0_n_cfg	PWM_OUT(x)_0_N 死区时间选择控制： 0: 选择死区时间 A； 1: 选择死区时间 B。	0x0
[0]	RW	pwm(x)_dts_out_0_p_cfg	PWM_OUT(x)_0_P 死区时间选择控制： 0: 选择死区时间 A； 1: 选择死区时间 B。	0x0

PWM(x)_sync_cfg

PWM(x)_sync_cfg 为 PWM(x)同步模式配置寄存器。

Offset Address: 0x0030 + 0x100×x Total Reset Value: 0x0000_0000



Bits	Access	Name	Description	Reset
[31:6]	-	reserved	保留。	0x0000000
[5]	RW	pwm(x)_sync_enable	PWM 同步工作模式使能。 0: 普通模式; 1: 开启同步工作模式, 由 PWM_SYNC_START 启动 PWM 功能。	0x0
[4]	RW	pwm(x)_sync_mode	PWM 同步模式选择。 0: 触发模式; 1: 循环模式。	0x0
[3:0]	RW	pwm(x)_sync_mux	循环模式的重载信号源头选择。 0x0: 来自 PWM0 完成信号; 0x1: 来自 PWM1 完成信号; 0x2: 来自 PWM2 完成信号; 0x3: 来自 PWM3 完成信号; 0x4: 来自 PWM4 完成信号; 0x5: 来自 PWM5 完成信号; 其他: 保留。	0x0

PWM(x)_sync_delay_cfg

PWM(x)_sync_delay_cfg 为 PWM(x)同步延时配置寄存器。

Offset Address: 0x0034 + 0x100×x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	pwm(x)_sync_delay_cfg	PWM x 同步模式启动延时配置寄存器。单位: 周期, 实际减一配置。	0x00000000

PWM(x)_period

PWM(x)_period 为 PWM(x)实时周期回读寄存器。



Offset Address: 0x0040 + 0x100×x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	pwm(x)_period	PWM x 周期生效值。	0x00000000

PWM(x)_duty0

PWM(x)_duty0 为 PWM(x)实时占空比 0 回读寄存器。

Offset Address: 0x0044 + 0x100×x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	pwm(x)_duty0	PWM x 占空比 0 生效值。	0x00000000

PWM(x)_duty1

PWM(x)_duty1 为 PWM(x)实时占空比 1 回读寄存器。

Offset Address: 0x0048 + 0x100×x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	pwm(x)_duty1	PWM x 占空比 1 生效值。	0x00000000

PWM(x)_duty2

PWM(x)_duty2 为 PWM(x)实时占空比 2 回读寄存器。

Offset Address: 0x004C + 0x100×x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	pwm(x)_duty2	PWM x 占空比 2 生效值。	0x00000000

PWM(x)_num

PWM(x)_num 为 PWM(x)方波数量实时回读寄存器。



Offset Address: 0x0050 + 0x100×x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	pwm(x)_num_cnt	PWM x 还需要输出的方波数目。 只有当 pwm(x)_busy 为 1 且 pwm(x)_keep 为 0 时才有意义。	0x0000
[15:0]	RO	pwm(x)_num	PWM x 输出方波的数目生效值。	0x0000

PWM(x)_ctrl_st

PWM(x)_ctrl_st 为 PWM(x)实时工作状态回读寄存器。

Offset Address: 0x0054 + 0x100×x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:12]	-	reserved	保留。	0x00000
[11:8]	RO	pwm(x)_pre_div_sel	PWM x 预分频生效值。 0x0: 不分频; 0x1: 2 分频; 0x2: 4 分频; 0x3: 8 分频; 0x4: 16 分频; 0x5: 32 分频; 0x6: 64 分频; 0x7: 128 分频; 0x8: 256 分频; 其它: 不分频。	0x0
[7:6]	-	reserved	保留。	0x0
[5:4]	RO	pwm(x)_align_mode	PWM x 对齐模式生效值。 00: 右对齐模式; 01: 左对齐模式;	0x0



Bits	Access	Name	Description	Reset
			10: 中间对齐模式; 其他: 保留	
[3]	-	reserved	保留。	0x0
[2]	RO	pwm(x)_keep	PWM x 输出模式生效值。 0: 输出 PWM(x)_num 个方波; 1: 一直输出方波。	0x0
[1]	RO	pwm(x)_inv	PWM x 输出正反相生效值。 0: 正常; 1: 反相。	0x0
[0]	RO	pwm(x)_busy	PWM x 工作状态。 0: 空闲; 1: 工作。	0x0

PWM(x)_dt_value

PWM(x)_dt_value 为 PWM(x) 实时死区时间回读寄存器。

Offset Address: 0x0060 + 0x100×x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	pwm(x)_dt_b	PWM x 死区时间 B 生效值。	0x0000
[15:0]	RO	pwm(x)_dt_a	PWM x 死区时间 A 生效值。	0x0000

PWM(x)_dt_ctrl

PWM(x)_dt_ctrl 为 PWM(x) 实时死区时间选择回读寄存器。

Offset Address: 0x0064 + 0x100×x Total Reset Value: 0x0000_0000

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



Bits	Access	Name	Description	Reset
[5]	RO	pwm(x)_dts_out_2 n	PWM_OUT(x)_2_N 死区时间生效状态: 0: 选择死区时间 A; 1: 选择死区时间 B。	0x0
[4]	RO	pwm(x)_dts_out_2 p	PWM_OUT(x)_2_P 死区时间生效状态: 0: 选择死区时间 A; 1: 选择死区时间 B。	0x0
[3]	RO	pwm(x)_dts_out_1 n	PWM_out(x)_1_n 死区时间生效状态: 0: 选择死区时间 A; 1: 选择死区时间 B。	0x0
[2]	RO	pwm(x)_dts_out_1 p	PWM_OUT(x)_1_P 死区时间生效状态: 0: 选择死区时间 A; 1: 选择死区时间 B。	0x0
[1]	RO	pwm(x)_dts_out_0 n	PWM_OUT (x)_0_N 死区时间生效状态: 0: 选择死区时间 A; 1: 选择死区时间 B。	0x0
[0]	RO	pwm(x)_dts_out_0 p	PWM_OUT(x)_0_P 死区时间生效状态: 0: 选择死区时间 A; 1: 选择死区时间 B。	0x0

PWM(x)_sync_delay

PWM(x)_sync_delay 为 PWM(x)实时同步延时配置回读寄存器。

Offset Address: 0x0074 + 0x100×x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	pwm(x)_sync_dela y	PWM x 同步模式启动延时配置生效值。	0x00000000



PWM_sync_start

PWM_sync_start 为 PWM 同步启动使能寄存器。

Offset Address: 0xFF0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	resverd	保留。	0x0000
[5]	WO	pwm5_sync_start	PWM5 启动配置寄存器, 写 1 启动。	0x0
[4]	WO	pwm4_sync_start	PWM4 启动配置寄存器, 写 1 启动。	0x0
[3]	WO	pwm3_sync_start	PWM3 启动配置寄存器, 写 1 启动。	0x0
[2]	WO	pwm2_sync_start	PWM2 启动配置寄存器, 写 1 启动。	0x0
[1]	WO	pwm1_sync_start	PWM1 启动配置寄存器, 写 1 启动。	0x0
[0]	WO	pwm0_sync_start	PWM0 启动配置寄存器, 写 1 启动。	0x0



目 录

14 安全子系统	14-1
14.1 概述	14-1



14 安全子系统

14.1 概述

安全子系统支持以下安全特性：

- 支持硬件产生真随机数，产生的真随机数符合 SP800-22 的随机测试标准。
- 支持 OTP (One Time Programmable) ，用户可用空间达 28Kbit。
- 支持硬件对称加解密算法和 HASH 算法：
 - 对称加解密算法
 - 支持 AES128、AES256，AES 算法的实现符合 FIPS 197 标准。AES 的工作模式符合 NIST special800-38a 标准；
 - 支持国密 SM4 算法。
 - HASH 算法
 - 支持 SHA256、SHA384、SHA512、HMAC-SHA256、HMAC-SHA384、HMAC-SHA512；
 - 支持国密 SM3 算法。
- 支持硬件非对称加解密算法：
 - 支持 RSA 3072/4096，符合 PKCS#1 V1.5/2.1 标准；
 - 支持国密 SM2 算法；
 - 支持 ECC256/384/512 椭圆曲线算法。
- 支持 JTAG 保护。
- 支持 KeyLadder，用于密钥加解密管理，增加安全强度。
- 支持 ARM TrustZone。



- 支持安全内存隔离。
- 支持安全启动。支持 bootloader 的签名校验和加解密。