MIPI 使用指南

文档版本 05

发布日期 2019-09-12

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：深圳市龙岗区坂田华为总部办公楼　　邮编：518129

网址：http://www.hisilicon.com/cn/

客户服务邮箱：support@hisilicon.com

前 言

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
Hi3559A	V100ES
Hi3559A	V100
Hi3559C	V100
Hi3519A	V100
Hi3556A	V100
Hi3516D	V300
Hi3516C	V500
Hi3516A	V300
Hi3559	V200
Hi3556	V200
Hi3516E	V300
Hi3516D	V200
Hi3516E	V200
Hi3518E	V300

说明

- 未有特殊说明，Hi3559CV100 与 Hi3559AV100 内容一致。
- 未有特殊说明，Hi3556AV100 与 Hi3519AV100 内容一致。
- 未有特殊说明，Hi3516DV300、Hi3516AV300、Hi3559V200、Hi3556V200 与 Hi3516CV500 内容一致。

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
危险	用于警示紧急的危险情形，若不避免，将会导致人员死亡或严重的人身伤害。
警告	用于警示潜在的危险情形，若不避免，可能会导致人员死亡或严重的人身伤害。
注意	用于警示潜在的危险情形，若不避免，可能会导致中度或轻微的人身伤害。
注意	用于传递设备或环境安全警示信息，若不避免，可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 不带安全警示符号的“注意”不涉及人身伤害。
说明	用于突出重要/关键信息、最佳实践和小窍门等。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

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。

文档版本 05 (2019-09-12)

第 5 次正式版本发布

1.3 小节，修改表 1-4 和表 1-5

1.4 小节涉及修改

1.5 小节，mipi_dev_attr_t 的【注意事项】涉及修改

文档版本 04 (2019-08-01)

第 4 次正式版本发布

1.4 小节，HI_MIPI_SET_HS_MODE【注意】涉及修改

新增 1.6 小节“MIPI Tx 模块参数”

文档版本 03 (2019-06-20)

1.4 小节，新增 HI_MIPI_TX_ENABLE，HI_MIPI_SET_PHY_CMVMODE 和 HI_MIPI_TX_ENABLE【注意】涉及更新。

1.5 小节，data_type_t【定义】涉及更新。sync_info_t、combo_dev_cfg_t、cmd_info_t【成员】涉及更新。

1.7 小节，涉及更新。

文档版本 02 (2019-05-30)

1.6 小节，涉及更新。

文档版本 01 (2019-05-15)

1.6 小节，input_mode_t【注意】涉及更新。

文档版本 00B17 (2019-04-15)

添加 Hi3516DV200 芯片支持

1.4 小节，新增 HI_MIPI_TX_GET_CMD。

1.5 小节，新增 get_cmd_info_t、slvs_err_check_mode_t、mipi_clk_mode_t、lvds_lane_work_mode_t、lvds_lane_work_attr_t、lvds_dev_attr_ex_t。input_mode_t、data_type_t、data_type_t、slvs_dev_attr_t、combo_dev_attr_t 涉及更新。

1.6 小节，涉及更新。

文档版本 00B16 (2019-02-28)

添加 Hi3516AV300 芯片支持

1.6 小节涉及修改

文档版本 00B15 (2019-01-15)

1.3 小节，表 1-2 涉及修改

1.5 小节，mipi_data_rate_t【芯片差异】和 mipi_dev_attr_t【注意事项】涉及修改

文档版本 00B14 (2018-11-13)

新增 Hi3518EV300 相关内容。

1.5 小节，WDR_VC_NUM、SYNC_CODE_NUM、wdr_mode_t、video_mode_t、cmd_info_t、mipi_dev_attr_t 涉及更新

1.6 小节，图 1-5 下【参数说明】涉及更新。

文档版本 00B13 (2018-10-30)

新增 Hi3516EV200/Hi3516EV300 相关内容。

文档版本 00B12 (2018-10-15)

1.5 小节，WDR_VC_NUM、wdr_mode_t 涉及修改

1.6 小节涉及修改

文档版本 00B11 (2018-09-29)

第 11 次临时版本发布。

新增 Hi3559V200/Hi3556V200 相关内容

文档版本 00B10 (2018-09-04)

第 10 次临时版本发布。

新增 Hi3516CV500/Hi3516DV300 相关内容。

文档版本 00B09 (2018-08-08)

第 9 次临时版本发布。

1.6 小节，新增 Hi3519AV100 MIPI Rx proc 信息。

文档版本 00B08 (2018-07-06)

第 8 次临时版本发布。

1.5 小节，WDR_VC_NUM 的【定义】涉及修改

1.6 小节涉及修改

文档版本 00B07 (2018-05-15)

第 7 次临时版本发布。

1.4 小节，新增 HI_MIPI_CLEAR；HI_MIPI_SET_DEV_ATTR【注意】涉及修改

1.5 小节，新增 COMS_MAX_DEV_NUM；input_mode_t【定义】和【注意事项】涉及修改；img_rect_t 的【注意】涉及修改，slvs_dev_attr_t 的【成员】涉及修改。

文档版本 00B06 (2018-04-13)

第 6 次临时版本发布。

添加 Hi3519AV100 的相关内容

1.4 小节，新增 HI_MIPI_TX_SET_DEV_CFG、HI_MIPI_TX_SET_CMD 和 HI_MIPI_TX_ENABLE

1.5 小节，新增 HI_MIPI_TX_IOC_MAGIC ~ cmd_info_t

文档版本 00B05 (2018-01-15)

第 5 次临时版本发布。

添加 Hi3559AV100 和 Hi3559CV100 的相关内容

文档版本 00B04 (2017-09-20)

第 4 次临时版本发布。

1.4 小节，删除 HI_MIPI_SET_OUTPUT_MSB。HI_MIPI_DISABLE_SENSOR_CLOCK【定义】涉及更新。HI_MIPI_RESET_SENSOR 至 HI_MIPI_UNRESET_SENSOR【定义】和【叁数】涉及修改

1.5 小节，新增 SYNC_CODE_NUM、wdr_mode_t 到 slvs_dev_attr_t，data_rate_t 改名为 mipi_data_rate_t 并更新【定义】。phy_cmv_mode_t 和 combo_dev_attr_t 涉及更新。删除 output_msb_t。

combo_dev_t, sns_rst_source_t, 和 sns_clk_source_t 涉及修改

1.6 小节，【调试信息】【参数说明】涉及更新。

文档版本 00B03 (2017-07-20)

第 3 次临时版本发布。

1.3 小节，修改表 1-2

文档版本 00B02 (2017-06-30)

第 2 次临时版本发布。

1.4 小节，新增 HI_MIPI_ENABLE_MIPI_CLOCK~ HI_MIPI_DISEABLE_SENSOR_CLOCK；HI_MIPI_SET_DEV_ATTR【注意】涉及修改

1.5 小节，新增 SENSOR_MAX_RESET_DEV、SENSOR_MAX_CLOCK_DEV、SNS_RESET_DEV 和 SNS_CLOCK_DEV

文档版本 00B01 (2017-05-28)

第 1 次临时版本发布。

目 录

插图目录

表格目录

1 MIPI 使用指南

注意

- Hi3516EV300、Hi3518EV300、Hi3516DV200 与 Hi3516EV200 均不支持 MIPI TX。
- 如无特殊说明，Hi3516DV200 与 Hi3516EV300 使用一致，Hi3518EV300 与 Hi3516EV200 使用一致。

1.1 概述

MIPI Rx 通过低电压差分信号接收原始视频数据，将接收到的串行差分信号（serial differential signal）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，同时兼容 DC 视频接口。

SLVS-EC 接口由 SONY 公司定义，用于高帧率和高分辨率图像采集，它可以将高速串行的数据转化为 DC（Digital Camera）时序后传递给下一级模块 VICAP（Video Capture）。

SLVS-EC 串行视频接口可以提供更高的传输带宽，更低的功耗，在组包方式上，数据的冗余度也更低。在应用中 SLVS-EC 接口提供了更加可靠和稳定的传输。

1.2 重要概念

- MIPI

  MIPI 的全称是 Mobile Industry Processor Interface(移动行业处理器接口)，本文描述的 MIPI 接口特指物理层使用 D-PHY 传输规范，协议层使用 CSI-2 的通信接口。

- LVDS

  LVDS 的全称是 Low Voltage differential Signaling(低压差分信号)，通过同步码区分消隐区和有效数据。

- SLVS-EC

  SLVS-EC 的全称是 Scalable Low Voltage Signaling Embedded Clock，是与 MIPI 并列的接口，用于高帧率和高分辨率图像采集。

- Lane

  用于连接发送端和接收端的一对高速差分线，即可以是时钟 Lane，也可以是数据 Lane。

- Link

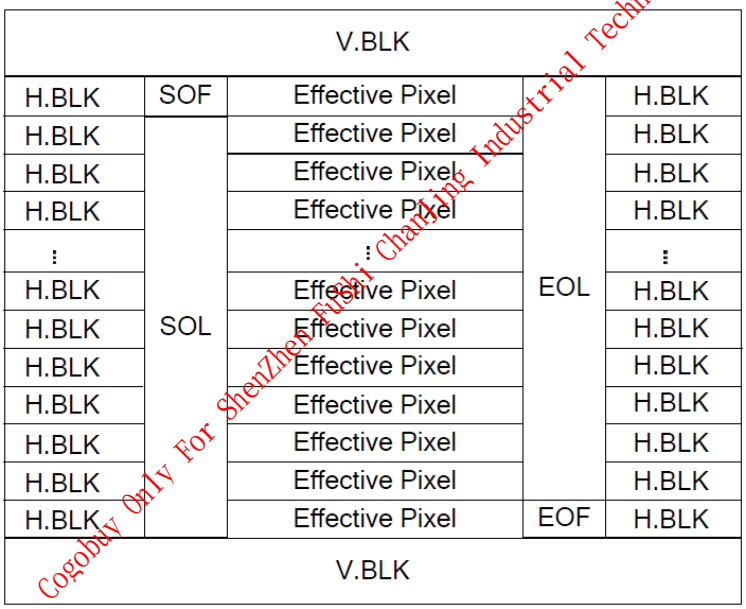
  发送端和接收端之间的时钟 Lane 和至少一个数据 Lane 组成一个 Link，本文中的 link 是一个软件概念，每一个 link 包括两个数据 lane。

- 同步码

  MIPI 接口使用 CSI-2 里面的短包进行同步，LVDS 使用同步码区分有效数据和消隐区。LVDS 有两种同步方式：

  – 使用 SOF/EOF 表示帧起始和结束，使用 SOL/EOL 表示行的起始和结束。同步方式如图 1-1 所示。

图1-1 SOF/EOF/SOL/EOL 同步方式

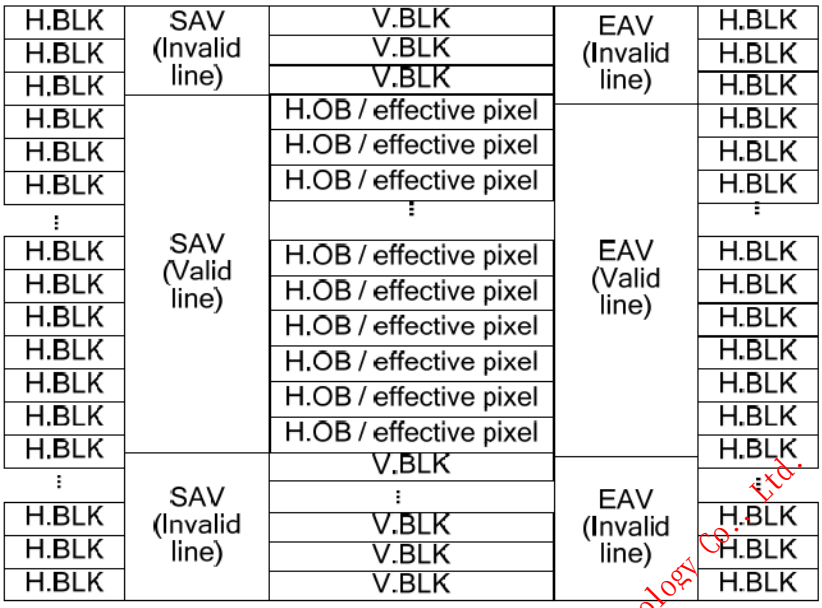


  – 使用 SAV(invalid) EAV(invalid)表示消隐区的无效数据开始和结束，使用 SAV(valid) EAV(valid)表示有效像素数据的开始和结束。

    每个同步码由 4 个字段组成，每个字段的位宽与像素数据位宽保持一致。前 3 个字段为固定基准码字，第 4 个字段由 sensor 厂家确定。

    由于不同的 sensor 可能会有不同的同步码，所以需要根据 sensor 配置同步码。同步方式如图 1-2 所示。

图1-2 SAV/EAV 同步方式



- DOL

  DOL 的全称是 Digital Overlap，指 SONY 的 WDR 功能。

1.3 功能描述

MIPI Rx 是一个支持多种差分视频输入接口的采集单元，通过 combo-PHY 接收 MIPI/LVDS/sub-LVDS/HiSPi/DC 接口的数据，通过不同的功能模式配置，MIPI Rx 可以支持多种速度和分辨率的数据传输需求，支持多种外部输入设备。最大支持 Lane 个数如表 1-1 所示。

表1-1 最大支持 Lane 的个数

芯片类型	最大支持 lane 数
Hi3559AV100ES	MIPI Rx 最大支持 8Lane MIPI 输入或 16Lane LVDS 输入。
Hi3559AV100	MIPI Rx 最大支持 8Lane MIPI 输入或 16Lane LVDS 输入。
Hi3519AV100	MIPI Rx 最大支持 8Lane MIPI 输入或 12Lane LVDS 输入。
Hi3516CV500 /Hi3516EV300 /Hi3516DV200	MIPI Rx 最大支持 4Lane MIPI 输入或 4Lane LVDS 输入。
Hi3516EV200 /Hi3518EV300	MIPI Rx 最大支持 2Lane MIPI 输入或 2Lane LVDS 输入。

MIPI Rx 能同时对接多个 sensor，最多对接 sensor 的数目如表 1-2 所示。

表1-2 最大对接 sensor 数目

芯片类型	对接 sensor 数目
Hi3559AV100ES	6
Hi3559AV100	8
Hi3519AV100	5
Hi3516CV500 Hi3516EV200 Hi3516EV300 Hi3518EV300	1
Hi3516DV300 Hi3559V200 Hi3556V200 Hi3516AV300	2

MIPI Rx 最大能同时对接不同数量的 sensor，每个 sensor 需要的 Lane 也不尽相同。因此用户需要确定 MIPI Rx 的 LANE 分布模式。具体的 Lane 分布模式请参见表 1-3。

表1-3 MIPI Rx Lane 分布模式

芯片类型	Mode	DEV0	DEV1	DEV2	DEV3	DEV4	DEV5	DEV6	DEV7
Hi3559AV100ES	0	L0~L15	N	N	N	N	N	N	N
	1	L0~L11	N	N	N	L12~L15	N	N	N
	2	L0~L11	N	N	N	L12 L14	L13 L15	N	N
	3	L0~L7	N	L8~L15	N	N	N	N	N
	4	L0~L7	N	L8~L11	N	L12~L15	N	N	N
	5	L0~L7	N	L8~L11	N	L12 L14	L13 L15	N	N
	6	L0~L7	N	L8 L10	L9 L11	L12 L14	L13 L15	N	N
	7	L0~L3	L4~L7	L8~L11	N	L12~L15	N	N	N
	8	L0~L3	L4~L7	L8~L11	N	L12 L14	L13 L15	N	N
	9	L0~L3	L4~L7	L8 L10	L9 L11	L12 L14	L13 L15	N	N

芯片类型	Mode	DEV0	DEV1	DEV2	DEV3	DEV4	DEV5	DEV6	DEV7
Hi3559AV100	0	L0~L15	N	N	N	N	N	N	N
	1	L0~L11	N	N	N	N	N	L12~L15	N
	2	L0~L11	N	N	N	N	N	L12 L14	L13 L15
	3	L0~L7	N	N	N	L8~L15	N	N	N
	4	L0~L7	N	N	N	L8~L11	N	L12~L15	N
	5	L0~L7	N	N	N	L8~L11	N	L12 L14	L13 L15
	6	L0~L7	N	N	N	L8 L10	L9 L11	L12 L14	L13 L15
	7	L0~L3	N	L4~L7	N	L8~L11	N	L12~L15	N
	8	L0~L3	N	L4~L7	N	L8~L11	N	L12 L14	L13 L15
	9	L0~L3	N	L4~L7	N	L8 L10	L9 L11	L12 L14	L13 L15
	A	L0~L3	N	L4 L6	L5 L7	L8 L10	L9 L11	L12 L14	L13 L15
	B	L0 L2	L1 L3	L4 L6	L5 L7	L8 L10	L9 L11	L12 L14	L13 L15
Hi3519AV100	0	L0~L11	N	N	N	N	N	N	N
	1	L0~L7	N	N	L8~L11	N	N	N	N
	2	L0~L7	N	N	L8 L10	L9 L11	N	N	N
	3	L0~L3	L4~L11	N	N	N	N	N	N
	4	L0~L3	L4~L7	N	L8~L11	N	N	N	N
	5	L0~L3	L4~L7	N	L8 L10	L9 L11	N	N	N
	6	L0~L3	L4 L6	L5 L7	L8 L10	L9 L11	N	N	N
Hi3516CV500	0	L0~L3	N	N	N	N	N	N	N
	1	L0L2	L1L3	N	N	N	N	N	N
Hi3516EV200	0	L0L2	N	N	N	N	N	N	N
Hi3516EV300	0	L0~L3	N	N	N	N	N	N	N

SLVS-EC 接口支持更高帧率更大分辨率图像的采集，通过 SLVS-EC 的 PHY 接收高速串行的数据转化为 DC（Digital Camera）时序，通过不同的功能模式配置，SLVS-EC 可以支持多种速度和分辨率的数据传输需求，支持多种外部输入设备。最大支持 Lane 个数如表 1-4 所示。

表1-4 最大支持 lane 的个数

芯片类型	定义
Hi3559AV100ES	SLVS-EC 最大支持 8Lane SLVS 输入。
Hi3559AV100	SLVS-EC 最大支持 8Lane SLVS 输入。
Hi3519AV100	SLVS-EC 最大支持 8Lane SLVS 输入。
Hi3516CV500/ Hi3516EV200	不支持 SLVS-EC 输入。

MIPI Rx 与 SLVS-EC 的 Lane 复用管脚，同一时刻同一个 Lane 只能被 MIPI Rx 和 SLVS-EC 中的一个使用。具体的 Lane 管脚连接请参见表 1-5。

表1-5 MIPI Rx 与 SLVS-EC 的 Lane 复用关系图

芯片类型	LANE	MIPI0	MIPI1	MIPI2	MIPI3	MIPI4	MIPI5	MIPI6	MIPI7	SLVS0	SLVS1	SLVS2	SLVS3
Hi3559AV100ES	Lane0	√	-	-	-	-	-	-	-	√	√	-	-
	Lane1	√	-	-	-	-	-	-	-	√	√	-	-
	Lane2	√	-	-	-	-	-	-	-	√	√	-	-
	Lane3	√	-	-	-	-	-	-	-	√	√	-	-
	Lane4	√	√	-	-	-	-	-	-	√	√	-	-
	Lane5	√	√	-	-	-	-	-	-	√	√	-	-
	Lane6	√	√	-	-	-	-	-	-	√	√	-	-
	Lane7	√	√	-	-	-	-	-	-	√	√	-	-
	Lane8	√	-	√			-	-	-	-	-	√	√
	Lane9	√	-	√	√		-	-	-	-	-	√	√
	Lane10	√	-	√			-	-	-	-	-	√	√
	Lane11	√	-	√	√		-	-	-	-	-	√	√
	Lane12	√	-	-	-	√		-	-	-	-	√	√
	Lane13	√	-	-	-	√	√	-	-	-	-	√	√

芯片类型	LANE	MIPI0	MIPI1	MIPI2	MIPI3	MIPI4	MIPI5	MIPI6	MIPI7	SLVS0	SLVS1	SLVS2	SLVS3
	Lane14	√	-	-	-	√		-	-	-	-	√	√
	Lane15	√	-	-	-	√	√	-	-	-	-	√	√
Hi3559 AV100	Lane0	√								√	√	-	-
	Lane1	√	√							√	√	-	-
	Lane2	√								√	√	-	-
	Lane3	√	√							√	√	-	-
	Lane4	√		√						√	√	-	-
	Lane5	√		√	√					√	√	-	-
	Lane6	√		√						√	√	-	-
	Lane7	√		√	√					√	√	-	-
	Lane8	√				√				-	-	√	√
	Lane9	√				√	√			-	-	√	√
	Lane10	√				√				-	-	√	√
	Lane11	√					√			-	-	√	√
	Lane12	√				√		√		-	-	√	√
	Lane13	√				√		√	√	-	-	√	√
	Lane14	√				√		√		-	-	√	√
	Lane15	√				√		√	√	-	-	√	√
Hi3519 AV100	Lane0	√								√			
	Lane1	√								√			
	Lane2	√								√			
	Lane3	√								√			
	Lane4	√	√							√			
	Lane5	√	√	√						√			
	Lane6	√	√							√			
	Lane7	√	√	√						√			
	Lane8	√	√		√								
	Lane9	√	√		√	√							

芯片类型	LANE	MIPI0	MIPI1	MIPI2	MIPI3	MIPI4	MIPI5	MIPI6	MIPI7	SLVS0	SLVS1	SLVS2	SLVS3
	Lane10	√	√		√								
	Lane11	√	√		√	√							

1.4 API 参考

MIPI Rx 提供对接 sensor 时序的功能。提供 ioctl 接口，可用的命令如下：

- HI_MIPI_SET_DEV_ATTR：设置 MIPI、SLVS 和并口设备属性。
- HI_MIPI_SET_HS_MODE：设置 MIPI Rx 的 Lane 分布。
- HI_MIPI_SET_PHY_CMVMODE：设置共模电压模式。
- HI_MIPI_RESET_SENSOR：复位 sensor。
- HI_MIPI_UNRESET_SENSOR：撤销复位 sensor。
- HI_MIPI_RESET_MIPI：复位 MIPI Rx。
- HI_MIPI_UNRESET_MIPI：撤销复位 MIPI Rx。
- HI_MIPI_RESET_SLVS：复位 SLVS。
- HI_MIPI_UNRESET_SLVS：撤销复位 SLVS。
- HI_MIPI_ENABLE_MIPI_CLOCK：打开 MIPI 设备的时钟。
- HI_MIPI_DISABLE_MIPI_CLOCK：关闭 MIPI 设备的时钟。
- HI_MIPI_ENABLE_SLVS_CLOCK：打开 SLVS 设备的时钟。
- HI_MIPI_DISABLE_SLVS_CLOCK：关闭 SLVS 设备的时钟。
- HI_MIPI_ENABLE_SENSOR_CLOCK：打开 SENSOR 的时钟。
- HI_MIPI_DISABLE_SENSOR_CLOCK：关闭 SENSOR 的时钟。
- HI_MIPI_CLEAR：清除设备相关的配置。

MIPI Tx 提供对接显示屏、级联的功能。提供 ioctl 接口，可用的命令如下：

- HI_MIPI_TX_SET_DEV_CFG：设置 MIPI Tx 设备的属性。
- HI_MIPI_TX_SET_CMD：设置发送给 MIPI Tx 设备的命令数据。
- HI_MIPI_TX_ENABLE：使能 MIPI Tx 设备。
- HI_MIPI_TX_DISABLE：禁用 MIPI Tx 设备。
- HI_MIPI_TX_GET_CMD：用于从外围设备读取信息。

注意

如果 VO 不使用 MIPI_TX 进行输出时，禁止调用任何 MIPI_TX 接口，且禁止开机画面使用 MIPI_TX 输出。调用接口后会使能 MIPI_TX，当 AVDD3318_MIPITX 供电电压超过 1.8v 时，会造成 MIPI_TX 损坏。

HI_MIPI_SET_DEV_ATTR

【描述】

设置 MIPI Rx、SLVS 和并口设备属性。

【定义】

```
#define HI_MIPI_SET_DEV_ATTR                _IOW(HI_MIPI_IOC_MAGIC, 0x01, combo_dev_attr_t)
```

【参数】

combo_dev_attr_t 类型的指针。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

无。

【需求】

头文件：hi_mipi.h

【注意】

- 除了配置 HI_MIPI_SET_DEV_ATTR 之外，还需要配置以下接口。
- 设置模式：接口为 HI_MIPI_SET_HS_MODE
- 打开 MIPI/SLVS 时钟：接口为 HI_MIPI_ENABLE_MIPI_CLOCK/HI_MIPI_ENABLE_SLVS_CLOCK。
- 复位 MIPI/SLVS：接口为 HI_MIPI_RESET_MIPI/HI_MIPI_RESET_SLVS。
- 打开 SENSOR 的时钟：接口为 HI_MIPI_ENABLE_SENSOR_CLOCK。
- 复位 SENSOR：接口为 HI_MIPI_RESET_SENSOR
- 撤销复位 MIPI/SLVS：接口为 HI_MIPI_UNRESET_MIPI/HI_MIPI_UNRESET_SLVS
- 撤销复位 SENSOR：接口为 HI_MIPI_UNRESET_SENSOR

- 推荐的配置流程如下：
  1. 设置模式
  2. 打开多路 MIPI/SLVS 时钟
  3. 复位多路 SENSOR 所对接的 MIPI Rx/SLVS
  4. 打开多路 SENSOR 所连接的时钟。
  5. 复位对接的所有 SENSOR
  6. 配置 MIPI Rx/SLVS 设备属性
  7. 撤销复位多路 SENSOR 所对接的 MIPI Rx/SLVS
  8. 撤销复位对接的所有 SENSOR
- 推荐的退出流程如下：
  1. 复位多路对接的 SENSOR。
  2. 关闭多路 SENSOR 所连接的时钟。
  3. 复位多路 SENSOR 所对接的 MIPI Rx/SLVS。
  4. 清除多路 SENSOR 所对接的 MIPI Rx/SLVS 设备的配置。
  5. 关闭多路 MIPI/SLVS 时钟。
- 操作 SENSOR 复位信号线和时钟信号线会对所连接到该信号线的所有 SENSOR 都产生效果。

【相关数据类型及接口】

- HI_MIPI_SET_HS_MODE
- HI_MIPI_RESET_SLVS
- HI_MIPI_UNRESET_SLVS
- HI_MIPI_RESET_SENSOR
- HI_MIPI_UNRESET_SENSOR
- HI_MIPI_RESET_MIPI
- HI_MIPI_UNRESET_MIPI

HI_MIPI_SET_HS_MODE

【描述】

设置 MIPI Rx 的 Lane 分布模式，对 SLVS 无作用。

【定义】

```
#define HI_MIPI_SET_HS_MODE              _IOW(HI_MIPI_IOC_MAGIC, 0x0b,
lane_divide_mode_t)
```

【参数】

lane_divide_mode_t 类型的指针。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	支持
Hi3516EV200	支持

【需求】

头文件：hi_mipi.h

【注意】

在接多路 sensor 输入时，建议在初始时根据硬件连接对整个 lane 分布进行全局的 lane 分布模式的设定，在之后的多路 sensor 采集过程中不能再调用此接口，否则可能对其他 sensor 采集有影响。

HI_MIPI_SET_PHY_CMVMODE

【描述】

设置共模电压模式。

【定义】

```
#define HI_MIPI_SET_PHY_CMVMODE        _IOW(HI_MIPI_IOC_MAGIC, 0x04,
phy_cmv_t)
```

【参数】

phy_cmv_t 类型的指针。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	支持
Hi3516EV200	支持

【需求】

头文件：hi_mipi.h

【注意】

SLVS 模式不支持。

HI_MIPI_RESET_SENSOR

【描述】

复位 sensor。

【定义】

```
#define HI_MIPI_RESET_SENSOR    _IOW(HI_MIPI_IOC_MAGIC, 0x05,
sns_rst_source_t)
```

【参数】

sns_rst_source_t SENSOR 复位信号线编号。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	支持

Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	支持
Hi3516EV200	支持

【需求】

头文件：hi_mipi.h

【注意】

无。

HI_MIPI_UNRESET_SENSOR

【描述】

撤销复位 sensor。

【定义】

```
#define HI_MIPI_UNRESET_SENSOR          _IOW(HI_MIPI_IOC_MAGIC, 0x06,
sns_rst_source_t)
```

【参数】

sns_rst_source_t SENSOR 复位信号线编号。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	支持
Hi3516EV200	支持

【需求】

头文件：hi_mipi.h

【注意】

无。

HI_MIPI_RESET_MIPI

【描述】

复位 MIPI_Rx。

【定义】

```
#define HI_MIPI_RESET_MIPI                _IOW(HI_MIPI_IOC_MAGIC, 0x07, combo_dev_t)
```

【参数】

combo_dev_t 设备号。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	支持
Hi3516EV200	支持

【需求】

头文件：hi_mipi.h

【注意】

无。

HI_MIPI_UNRESET_MIPI

【描述】

撤销复位 MIPI_Rx。

【定义】

```
#define HI_MIPI_UNRESET_MIPI   _IOW(HI_MIPI_IOC_MAGIC, 0x08, combo_dev_t)
```

【参数】

combo_dev_t 设备号。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	支持
Hi3516EV200	支持

【需求】

头文件：hi_mipi.h

【注意】

无。

HI_MIPI_RESET_SLVS

【描述】

复位 SLVS。

【定义】

```
#define HI_MIPI_RESET_SLVS     _IOW(HI_MIPI_IOC_MAGIC, 0x09, combo_dev_t)
```

【参数】

combo_dev_t 设备号。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	不支持
Hi3516EV200	不支持

【需求】

头文件：hi_mipi.h

【注意】

无。

HI_MIPI_UNRESET_SLVS

【描述】

撤销复位 SLVS。

【定义】

```
#define HI_MIPI_UNRESET_SLVS   _IOW(HI_MIPI_IOC_MAGIC, 0x0a, combo_dev_t)
```

【参数】

combo_dev_t 设备号。

【返回值】

返回值	描述
0	成功。

返回值	描述
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	不支持
Hi3516EV200	不支持

【需求】

头文件：hi_mipi.h

【注意】

无。

HI_MIPI_ENABLE_MIPI_CLOCK

【描述】

打开 MIPI 设备的时钟。

【定义】

```
#define HI_MIPI_ENABLE_MIPI_CLOCK       _IOW(HI_MIPI_IOC_MAGIC, 0x0c,
combo_dev_t)
```

【参数】

combo_dev_t 设备号。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	支持
Hi3516EV200	支持

【需求】

头文件：hi_mipi.h

【注意】

无。

HI_MIPI_DISABLE_MIPI_CLOCK

【描述】

关闭 MIPI 设备的时钟。

【定义】

```
#define HI_MIPI_DISABLE_MIPI_CLOCK              _IOW(HI_MIPI_IOC_MAGIC, 0x0d, combo_dev_t)
```

【参数】

combo_dev_t 设备号。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	支持
Hi3559AV100	支持
Hi3519AV100	支持

Hi3516CV500	支持
Hi3516EV200	支持

【需求】

头文件：hi_mipi.h

【注意】

无。

HI_MIPI_ENABLE_SLVS_CLOCK

【描述】

打开 SLVS 设备的时钟。

【定义】

```
#define HI_MIPI_ENABLE_SLVS_CLOCK           _IOW(HI_MIPI_IOC_MAGIC, 0x0e, combo_dev_t)
```

【参数】

combo_dev_t 设备号。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	不支持
Hi3516EV200	不支持

【需求】

头文件：hi_mipi.h

【注意】

无。

HI_MIPI_DISABLE_SLVS_CLOCK

【描述】

关闭 SLVS 设备的时钟。

【定义】

```
#define HI_MIPI_DISABLE_SLVS_CLOCK              _IOW(HI_MIPI_IOC_MAGIC,
0x0f, combo_dev_t)
```

【参数】

combo_dev_t 设备号。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	不支持
Hi3516EV200	不支持

【需求】

头文件：hi_mipi.h

【注意】

无。

HI_MIPI_ENABLE_SENSOR_CLOCK

【描述】

打开 SENSOR 的时钟。

【定义】

```
#define HI_MIPI_ENABLE_SENSOR_CLOCK        _IOW(HI_MIPI_IOC_MAGIC, 0x10,
sns_clk_source_t)
```

【参数】

SENSOR 的时钟设备源编号。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	支持
Hi3516EV200	支持

【需求】

头文件：hi_mipi.h

【注意】

无。

HI_MIPI_DISABLE_SENSOR_CLOCK

【描述】

关闭 SENSOR 的时钟。

【定义】

```
#define HI_MIPI_DISABLE_SENSOR_CLOCK       _IOW(HI_MIPI_IOC_MAGIC, 0x11,
```



```
sns_clk_source_t)
```

【参数】

SENSOR 的时钟设备源编号。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	支持
Hi3516EV200	支持

【需求】

头文件：hi_mipi.h

【注意】

无。

HI_MIPI_CLEAR

【描述】

清除设备相关的配置。

【定义】

```
#define HI_MIPI_CLEAR           _IOW(HI_MIPI_IOC_MAGIC, 0x12, combo_dev_t)
```

【参数】

设备号。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	不支持
Hi3559AV100	支持
Hi3519AV100	不支持
Hi3516CV500	不支持
Hi3516EV200	不支持

【需求】

头文件：hi_mipi.h

【注意】

该接口在业务退出时调用，用于清除设备的相关配置。

HI_MIPI_TX_SET_DEV_CFG

【描述】

设置 MIPI Tx 设备的属性。

【定义】

```
#define HI_MIPI_TX_SET_DEV_CFG              _IOW(HI_MIPI_TX_IOC_MAGIC, 0x01, combo_dev_cfg_t)
```

【参数】

MIPI Tx 设备属性。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	不支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	支持
Hi3516EV200	不支持

【需求】

头文件：hi_mipi_tx.h

【注意】

无。

HI_MIPI_TX_SET_CMD

【描述】

设置发送给 MIPI Tx 设备的命令数据。

【定义】

```
#define HI_MIPI_TX_SET_CMD                _IOW(HI_MIPI_TX_IOC_MAGIC, 0x02,
cmd_info_t)
```

【参数】

发送给 MIPI Tx 设备的命令信息。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	不支持
Hi3559AV100	支持

Hi3519AV100	支持
Hi3516CV500	支持
Hi3516EV200	不支持

【需求】

头文件：hi_mipi_tx.h

【注意】

无。

HI_MIPI_TX_GET_CMD

【描述】

用于从外围设备读取信息。

【定义】

```
#define HI_MIPI_TX_GET_CMD                  _IOWR(HI_MIPI_TX_IOC_MAGIC, 0x04,
get_cmd_info_t)
```

【参数】

详见 get_cmd_info_t 结构体说明。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	不支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	支持
Hi3516EV200	不支持

【需求】

头文件：hi_mipi_tx.h

【注意】

无。

HI_MIPI_TX_ENABLE

【描述】

使能 MIPI Tx 设备。

【定义】

```
#define HI_MIPI_TX_ENABLE                    _IO(HI_MIPI_TX_IOC_MAGIC, 0x03)
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	不支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	支持
Hi3516EV200	不支持

【需求】

头文件：hi_mipi_tx.h

【注意】

此接口调用后 mipi 将工作于 HS 模式。

HI_MIPI_TX_DISABLE

【描述】

禁用 MIPI Tx 设备。

【定义】

```
#define HI_MIPI_TX_DISABLE                  _IO(HI_MIPI_TX_IOC_MAGIC, 0x05)
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
-1	失败，并设置 errno

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	不支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	支持
Hi3516EV200	不支持

【需求】

头文件：hi_mipi_tx.h

【注意】

此接口调用后 mipi 将工作于 LP 模式。

1.5 数据类型

MIPI Rx 相关数据类型定义如下：

- HI_MIPI_IOC_MAGIC：MIPI Rx ioctl 命令的幻数。
- combo_dev_t：MIPI Rx、SLVS 设备类型。
- SNS_MAX_RST_SOURCE_NUM：SENSOR 的复位信号线个数。

- SNS_MAX_CLK_SOURCE_NUM：SENSOR 的时钟信号线个数。
- sns_rst_source_t：SENSOR 的复位信号线编号，软件上称为 SENSOR 的复位源。
- sns_clk_source_t：SENSOR 的时钟信号线编号，软件上称为 SENSOR 的时钟源。
- MIPI_RX_MAX_DEV_NUM：MIPI Rx 支持的设备数。
- SLVS_MAX_DEV_NUM：SLVS 支持的设备数。
- SLVS_DEV_NUM_START：SLVS 起始设备号。
- COMBO_MAX_LANE_NUM：设备最大支持的 Lane 数量。
- MAX_LANE_NUM_PER_LINK：MIPI Rx 一个 link 的 Lane 数。
- MIPI_LANE_NUM：MIPI Rx 的 MIPI 设备支持的最大 Lane 数。
- LVDS_LANE_NUM：LVDS/HiSPi 接口支持的 Lane 数量。
- SLVS_LANE_NUM：SLVS 设备支持的最大 Lane 数。
- COMS_MAX_DEV_NUM：支持的并口设备数。
- COMBO_MAX_LINK_NUM：MIPI Rx 最大的支持的 Link 数量
- COMBO_MAX_DEV_NUM：MIPI Rx 设备的数量。
- WDR_VC_NUM：定义最多支持的 Virtual Channel 数量。
- SYNC_CODE_NUM：定义 LVDS 每个 Virtual Channel 的同步码数量。
- lane_divide_mode_t：MIPI Rx 的 Lane 分布。
- input_mode_t：MIPI Rx 输入接口类型。
- mipi_data_rate_t：MIPI Rx，SLVS 输入速率。
- img_rect_t：crop 属性。
- slvs_lane_rate_t：SLVS Lane 的输入速率。
- slvs_err_check_mode_t：SLVS 的 CRC 和 ECC 模式。
- data_type_t：传输的数据类型。
- mipi_wdr_mode_t：MIPI WDR 模式。
- mipi_dev_attr_t：MIPI 设备属性。
- wdr_mode_t：LVDS WDR 模式。
- lvds_sync_mode_t：LVDS 同步方式。
- lvds_bit_endian_t：比特位大小端模式。
- lvds_vsync_type_t：LVDS vsync 类型。
- lvds_vsync_attr_t：LVDS vsync 参数。
- lvds_fid_type_t：Frame identification Id 类型。
- lvds_fid_attr_t：Frame indentification Id 配置信息。
- mipi_clk_mode_t：时钟共享模式。
- lvds_lane_work_mode_t：LVDS LANE 的同步码是否同步到达。
- lvds_lane_work_attr_t：LVDS LANE 的 LANE 的同步码工作属性。
- lvds_dev_attr_t：LVDS/SubLVDS/HiSPi 设备属性。
- lvds_dev_attr_ex_t：LVDS/SubLVDS/HiSPi 设备高级属性。
- slvs_dev_attr_t：SLVS 设备属性。

- phy_cmv_mode_t：PHY 共模电压模式。
- phy_cmv_t：PHY 共模电压配置信息。
- combo_dev_attr_t：combo 设备属性。
- HI_MIPI_TX_IOC_MAGIC：MIPI Rx ioctl 命令的幻数。
- LANE_MAX_NUM：定义 MIPI Tx 支持的最大 Lane 数。
- output_mode_t：MIPI Tx 输出模式。
- video_mode_t：MIPI Tx 视频模式。
- output_format_t：MIPI Tx 输出格式。
- sync_info_t：MIPI Tx 设备同步信息。
- combo_dev_cfg_t：MIPI Tx 设备属性。
- cmd_info_t：发送给 MIPI Tx 设备的命令信息。
- get_cmd_info_t：发送给 MIPI Tx 设备的命令信息。

HI_MIPI_IOC_MAGIC

【说明】

MIPI Rx ioctl 命令的幻数。

【定义】

```
#define HI_MIPI_IOC_MAGIC   'm'
```

【成员】

无

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无

combo_dev_t

【说明】

MIPI Rx、SLVS 设备类型。

【定义】

```
typedef unsigned int combo_dev_t;
```

【芯片差异】

芯片类型	MIPI 设备范围	SLVS 设备范围
Hi3559AV100ES	[0, MIPI_RX_MAX_DEV_NUM)	[SLVS_DEV_NUM_START, COMBO_MAX_DEV_NUM)
Hi3559AV100	[0, MIPI_RX_MAX_DEV_NUM)	[SLVS_DEV_NUM_START, SLVS_MAX_DEV_NUM)
Hi3519AV100	[0, MIPI_RX_MAX_DEV_NUM)	[0, SLVS_MAX_DEV_NUM)
Hi3516CV500	[0, MIPI_RX_MAX_DEV_NUM)	0

【注意事项】

无。

【相关数据类型及接口】

- combo_dev_attr_t
- HI_MIPI_SET_DEV_ATTR
- HI_MIPI_RESET_SLVS
- HI_MIPI_UNRESET_SLVS
- HI_MIPI_RESET_MIPI
- HI_MIPI_UNRESET_MIPI

SNS_MAX_RST_SOURCE_NUM

【说明】

SENSOR 的复位信号线个数。

【定义】

Hi3559AV100ES：

```
#define SNS_MAX_RST_SOURCE_NUM    3
```

Hi3559AV100：

```
#define SNS_MAX_RST_SOURCE_NUM    4
```

Hi3519AV100：

```
#define SNS_MAX_RST_SOURCE_NUM    3
```

Hi3516CV500：

```
#define SNS_MAX_RST_SOURCE_NUM    2
```

Hi3516EV200：

```
#define SNS_MAX_RST_SOURCE_NUM    1
```

【芯片差异】

芯片类型	SENSOR 复位信号线数目
Hi3559AV100ES	3
Hi3559AV100	4
Hi3519AV100	3
Hi3516CV500	2
Hi3516EV200	1

【注意事项】

无。

【相关数据类型及接口】

无

SNS_MAX_CLK_SOURCE_NUM

【说明】

SENSOR 的时钟信号线个数。

【定义】

Hi3559AV100ES：

```
#define SNS_MAX_CLK_SOURCE_NUM    3
```

Hi3559AV100：

```
#define SNS_MAX_CLK_SOURCE_NUM    4
```

Hi3519AV100：

```
#define SNS_MAX_CLK_SOURCE_NUM    3
```

Hi3516CV500：

```
#define SNS_MAX_CLK_SOURCE_NUM    2
```

Hi3516EV200：

```
#define SNS_MAX_CLK_SOURCE_NUM    1
```

【芯片差异】

芯片类型	SENSOR 时钟信号线数目
Hi3559AV100ES	3
Hi3559AV100	4
Hi3519AV100	3

Hi3516CV500	2
Hi3516EV200	1

【注意事项】

无

【相关数据类型及接口】

无

sns_rst_source_t

【说明】

SENSOR 的复位信号线编号，软件上称为 SENSOR 的复位源。

【定义】

```
typedef unsigned int sns_rst_source_t;
```

【芯片差异】

芯片类型	SENSOR 复位设备范围
Hi3559AV100ES	[0, SNS_MAX_RST_SOURCE_NUM)
Hi3559AV100	[0, SNS_MAX_RST_SOURCE_NUM)
Hi3519AV100	[0, SNS_MAX_RST_SOURCE_NUM)
Hi3516CV500	[0, SNS_MAX_RST_SOURCE_NUM)
Hi3516EV200	[0, SNS_MAX_RST_SOURCE_NUM)

【注意事项】

每条 SENSOR 复位信号线可以接两个 SENSOR，用户需要根据板子的连线确认 SENSOR 复位信号线编号。不同的芯片的 SENSOR 复位信号线数目请参考取值范围。

【相关数据类型及接口】

- HI_MIPI_RESET_SLVS
- HI_MIPI_UNRESET_SLVS

sns_clk_source_t

【说明】

SENSOR 的时钟信号线编号，软件上称为 SENSOR 的时钟源。

【定义】


```
typedef unsigned int sns_clk_source_t;
```

【芯片差异】

芯片类型	SENSOR 时钟设备范围
Hi3559AV100ES	[0, SNS_MAX_CLK_SOURCE_NUM)
Hi3559AV100	[0, SNS_MAX_CLK_SOURCE_NUM)
Hi3519AV100	[0, SNS_MAX_CLK_SOURCE_NUM)
Hi3516CV500	[0, SNS_MAX_CLK_SOURCE_NUM)
Hi3516EV200	[0, SNS_MAX_CLK_SOURCE_NUM)

【注意事项】

每条 SENSOR 时钟信号线可以接两个 SENSOR，用户需要根据板子的连线确认 SENSOR 时钟信号线编号。不同的芯片的 SENSOR 复位信号线数目请参考取值范围。

【相关数据类型及接口】

- HI_MIPI_ENABLE_SENSOR_CLOCK
- HI_MIPI_DISABLE_SENSOR_CLOCK

MIPI_RX_MAX_DEV_NUM

【说明】

MIPI Rx 支持的设备数。

【定义】

Hi3559AV100ES:

```
#define MIPI_RX_MAX_DEV_NUM       6
```

Hi3559AV100:

```
#define MIPI_RX_MAX_DEV_NUM       8
```

Hi3519AV100:

```
#define MIPI_RX_MAX_DEV_NUM       5
```

Hi3516CV500:

```
#define MIPI_RX_MAX_DEV_NUM       2
```

Hi3516EV200:

```
#define MIPI_RX_MAX_DEV_NUM       1
```

【芯片差异】

芯片类型	MIPI Rx 支持的设备数
Hi3559AV100ES	6
Hi3559AV100	8
Hi3519AV100	5
Hi3516CV500	2
Hi3516EV200	1

【注意事项】

- Hi3516CV500 在同一时刻只有 1 个 MIPI Rx 设备可用。
- Hi3516DV300/Hi3559V200/Hi3556V200/Hi3516AV300 的两个 MIPI Rx 设备可以同时使用。

【相关数据类型及接口】

无

SLVS_MAX_DEV_NUM

【说明】

SLVS 支持的设备数。

【定义】

Hi3559AV100ES/Hi3559AV100

```
#define SLVS_MAX_DEV_NUM        4
```

Hi3519AV100

```
#define SLVS_MAX_DEV_NUM        1
```

【芯片差异】

无

【注意事项】

无

【相关数据类型及接口】

无

SLVS_DEV_NUM_START

【说明】

SLVS 起始设备号。

【定义】

Hi3559AV100ES:

```
#define SLVS_DEV_NUM_START      MIPI_RX_MAX_DEV_NUM
```

Hi3559AV100:

```
#define SLVS_DEV_NUM_START      0
```

【芯片差异】

芯片类型	SLVS 起始设备号
Hi3559AV100ES	MIPI_RX_MAX_DEV_NUM
Hi3559AV100	0

【注意事项】

无

【相关数据类型及接口】

无

COMBO_MAX_LANE_NUM

【说明】

MIPI Rx/SLVS 的 Lane 总数目。

【定义】

Hi3559AV100ES/Hi3559AV100:

```
#define COMBO_MAX_LANE_NUM      16
```

Hi3519AV100:

```
#define COMBO_MAX_LANE_NUM      12
```

Hi3516CV500:

```
#define COMBO_MAX_LANE_NUM      4
```

Hi3516EV200:

```
#define COMBO_MAX_LANE_NUM      2
```

Hi3516EV300:

```
#define COMBO_MAX_LANE_NUM      4
```

【芯片差异】

无

【注意事项】

无

【相关数据类型及接口】

无

MAX_LANE_NUM_PER_LINK

【说明】

MIPI Rx 一个 link 的 Lane 数。

【定义】

```
#define MAX_LANE_NUM_PER_LINK   2
```

【芯片差异】

无

【注意事项】

这里的 link 是软件概念，软件上把一个逻辑的 link 拆分成了 2 个软件的 link。

【相关数据类型及接口】

无

MIPI_LANE_NUM

【说明】

MIPI Rx 的 MIPI 设备支持的最大 Lane 数。

【定义】

Hi3559AV100ES/Hi3559AV100：

```
#define MIPI_LANE_NUM          (MAX_LANE_NUM_PER_LINK * 4)
```

Hi3519AV100：

```
#define MIPI_LANE_NUM          8
```

Hi3516CV500：

```
#define MIPI_LANE_NUM          4
```

Hi3516EV200：

```
#define MIPI_LANE_NUM          2
```

Hi3516EV300：

```
#define MIPI_LANE_NUM          4
```

【芯片差异】

无

【注意事项】

无

【相关数据类型及接口】

无

LVDS_LANE_NUM

【说明】

MIPI Rx 的 LVDS 设备支持的最大 Lane 数。

【定义】

Hi3559AV100ES/Hi3559AV100：

```
#define LVDS_LANE_NUM          COMBO_MAX_LANE_NUM
```

Hi3519AV100：

```
#define LVDS_LANE_NUM          12
```

Hi3516CV500：

```
#define LVDS_LANE_NUM          4
```

Hi3516EV200：

```
#define LVDS_LANE_NUM          2
```

Hi3516EV300：

```
#define LVDS_LANE_NUM          4
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

SLVS_LANE_NUM

【说明】

SLVS 设备支持的最大 Lane 数。

【定义】

Hi3559AV100ES/Hi3559AV100：

```
#define SLVS_LANE_NUM          8
```

Hi3519AV100：

```
#define SLVS_LANE_NUM          4
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

COMS_MAX_DEV_NUM

【说明】

支持的并口设备数。

【定义】

Hi3559AV100：

```
#define COMS_MAX_DEV_NUM       3
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

COMBO_MAX_LINK_NUM

【说明】

MIPI Rx 最大的支持的 Link 数量。

【定义】

Hi3559AV100ES/Hi3559AV100：

```
#define COMBO_MAX_LINK_NUM     8
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

COMBO_MAX_DEV_NUM

【说明】

MIPI Rx 设备的数量。

【定义】

Hi3559AV100ES：

```
#define COMBO_MAX_DEV_NUM     (MIPI_RX_MAX_DEV_NUM + SLVS_MAX_DEV_NUM)
```

Hi3559AV100/ Hi3519AV100/Hi3516CV500/Hi3516EV200：

```
#define COMBO_MAX_DEV_NUM     MIPI_RX_MAX_DEV_NUM
```

【芯片差异】

芯片类型	MIPI Rx 设备的数量
Hi3559AV100ES	(MIPI_RX_MAX_DEV_NUM + SLVS_MAX_DEV_NUM)
Hi3559AV100/ Hi3519AV100/ Hi3516CV500/H3516EV200	MIPI_RX_MAX_DEV_NUM

【注意事项】

Hi3559AV100ES 的 MIPI 设备号与 SLVS 的设备号是统一编号的，
Hi3559AV100/Hi3519AV100 的 MIPI 设备号与 SLVS 的设备号是独立编号的。

【相关数据类型及接口】

无。

WDR_VC_NUM

【说明】

定义最多支持的 Virtual Chnnael 数量。

【定义】

Hi3559AV100ES/Hi3559AV100/Hi3519AV100/Hi3516CV500：

```
#define WDR_VC_NUM          4
```

Hi3516EV300:

```
#define WDR_VC_NUM          2
```

Hi3516EV200/Hi3518EV300:

```
#define WDR_VC_NUM          2
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

SYNC_CODE_NUM

【说明】

定义 LVDS 每个 Virtual Channel 的同步码数量

【定义】

Hi3559AV100ES/Hi3559AV100/Hi3519AV100/Hi3516CV500：

```
#define SYNC_CODE_NUM          4
```

Hi3516EV300:

```
#define SYNC_CODE_NUM          2
```

Hi3516EV200/Hi3518EV300:

```
#define SYNC_CODE_NUM          2
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

lane_divide_mode_t

【说明】

MIPI Rx 的 LANE 分布。

【定义】

Hi3559AV100ES：

```
typedef enum
{
    LANE_DIVIDE_MODE_0    = 0,
    LANE_DIVIDE_MODE_1    = 1,
    LANE_DIVIDE_MODE_2    = 2,
    LANE_DIVIDE_MODE_3    = 3,
    LANE_DIVIDE_MODE_4    = 4,
    LANE_DIVIDE_MODE_5    = 5,
    LANE_DIVIDE_MODE_6    = 6,
    LANE_DIVIDE_MODE_7    = 7,
    LANE_DIVIDE_MODE_8    = 8,
    LANE_DIVIDE_MODE_9    = 9,
    LANE_DIVIDE_MODE_BUTT
} lane_divide_mode_t;
```

Hi3559AV100：

```
typedef enum
{
    LANE_DIVIDE_MODE_0    = 0,
    LANE_DIVIDE_MODE_1    = 1,
    LANE_DIVIDE_MODE_2    = 2,
    LANE_DIVIDE_MODE_3    = 3,
    LANE_DIVIDE_MODE_4    = 4,
    LANE_DIVIDE_MODE_5    = 5,
    LANE_DIVIDE_MODE_6    = 6,
    LANE_DIVIDE_MODE_7    = 7,
    LANE_DIVIDE_MODE_8    = 8,
    LANE_DIVIDE_MODE_9    = 9,
    LANE_DIVIDE_MODE_A    = 0xA,
    LANE_DIVIDE_MODE_B    = 0xB,
    LANE_DIVIDE_MODE_BUTT
} lane_divide_mode_t;
```

Hi3519AV100：

```
typedef enum
{
    LANE_DIVIDE_MODE_0    = 0,
    LANE_DIVIDE_MODE_1    = 1,
    LANE_DIVIDE_MODE_2    = 2,
    LANE_DIVIDE_MODE_3    = 3,
    LANE_DIVIDE_MODE_4    = 4,
    LANE_DIVIDE_MODE_5    = 5,
    LANE_DIVIDE_MODE_6    = 6,
    LANE_DIVIDE_MODE_BUTT
```

```
} lane_divide_mode_t;
```

Hi3516CV500：

```
typedef enum
{
    LANE_DIVIDE_MODE_0    = 0,
    LANE_DIVIDE_MODE_1    = 1,
    LANE_DIVIDE_MODE_BUTT
} lane_divide_mode_t;
```

Hi3516EV200：

```
typedef enum
{
    LANE_DIVIDE_MODE_0    = 0,
    LANE_DIVIDE_MODE_BUTT
} lane_divide_mode_t;
```

【芯片差异】

芯片类型	MIPI Rx 的 LANE 分布
Hi3559AV100ES	[LANE_DIVIDE_MODE_0, LANE_DIVIDE_MODE_9]
Hi3559AV100	[LANE_DIVIDE_MODE_0, LANE_DIVIDE_MODE_B]
Hi3519AV100	[LANE_DIVIDE_MODE_0, LANE_DIVIDE_MODE_6]
Hi3516CV500	[LANE_DIVIDE_MODE_0, LANE_DIVIDE_MODE_1]
Hi3516EV200	仅能取值 LANE_DIVIDE_MODE_0

【注意事项】

只有 MIPI 需要设置 LANE 的分布。

【相关数据类型及接口】

HI_MIPI_SET_HS_MODE

input_mode_t

【说明】

MIPI Rx 输入接口类型

【定义】

```
typedef enum
{
    INPUT_MODE_MIPI         = 0x0,            /* mipi */
    INPUT_MODE_SUBLVDS      = 0x1,            /* SUB_LVDS */
```



```
    INPUT_MODE_LVDS        = 0x2,          /* LVDS */
    INPUT_MODE_HISPI       = 0x3,          /* HISPI */
    INPUT_MODE_SLVS        = 0x4,          /* SLVS */
    INPUT_MODE_CMOS        = 0x5,          /* CMOS */
    INPUT_MODE_BT601       = 0x6,          /* BT601 */
    INPUT_MODE_BT656       = 0x7,          /* BT656 */
    INPUT_MODE_BT1120      = 0x8,          /* BT1120 */
    INPUT_MODE_BYPASS      = 0x9,          /* MIPI Bypass */
    INPUT_MODE_LVDS_EX     = 0xA,          /* LVDS EX */
    INPUT_MODE_BUTT
} input_mode_t;
```

【芯片差异】

无。

【注意事项】

- Hi3559AV100 输入接口类型为并口设备(INPUT_MODE_CMOS，INPUT_MODE_BT601，INPUT_MODE_BT656，INPUT_MODE_BT1120)时只支持 COMS_MAX_DEV_NUM 路，其中第 0 和第 1 路可以不设置，第 2 路必须设置。
- Hi3559AV100 当第 2 路设置时，当为 INPUT_MODE_BT601 和 INPUT_MODE_BT656 时，LANE8-LANE11 不能作为其他接口使用，当为 INPUT_MODE_CMOS 和 INPUT_MODE_BT1120 时，LANE8-LANE15 不能作为其他接口使用。
- Hi3519AV100 输入接口类型为 INPUT_MODE_BT601 或者 INPUT_MODE_BT656 时，MIPI 设备设置成 0 或者 1，第 0 路的管脚复用了 LANE8~LANE11，第 1 路的管脚复用了 LANE4~LANE7。
- Hi3519AV100 输入接口类型为 INPUT_MODE_BT1120 时，MIPI 设备只能设置成 0，第 0 路的管脚复用了 LANE4~LANE11。

【相关数据类型及接口】

无。

mipi_data_rate_t

【说明】

MIPI Rx，SLVS 输入速率。

【定义】

```
typedef enum
{
    MIPI_DATA_RATE_X1 = 0,        /* output 1 pixel per clock */
    MIPI_DATA_RATE_X2 = 1,        /* output 2 pixel per clock */
    MIPI_DATA_RATE_BUTT
} mipi_data_rate_t;
```


【芯片差异】

芯片类型	是否支持 MIPI_DATA_RATE_X2
Hi3559AV100ES	MIPI 的设备 0 和 SLVS 的设备 6 支持。
Hi3559AV100 Hi3519AV100	MIPI 的设备 0 和 SLVS 的设备 0 支持。
Hi3516CV500 Hi3516EV200	不支持。

【注意事项】

无。

【相关数据类型及接口】

无。

img_rect_t

【说明】

Mipi crop 属性。

【定义】

```
typedef struct
{
    int x;
    int y;
    unsigned int width;
    unsigned int height;
} img_rect_t;
```


【成员】

成员名称	描述
x	Crop 起始位置 x。
y	Crop 起始位置 y。
width	Crop 宽度，单位：像素。
height	Crop 高度单位：像素。

【芯片差异】

无。

【注意事项】

SLV-EC 的裁剪 Y 坐标必须要大于等于 sensor 输出有效行的行号。

【相关数据类型及接口】

无。

slvs_lane_rate_t

【说明】

SLVS LANE 的输入速率。

【定义】

```
typedef enum
{
    SLVS_LANE_RATE_LOW = 0,        /* 1152Mbps */
    SLVS_LANE_RATE_HIGH = 1,       /* 2304Mbps */
    SLVS_LANE_RATE_BUTT
} slvs_lane_rate_t;
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

slvs_err_check_mode_t

【说明】

SLVS 的 CRC 和 ECC 模式。

【定义】

```
typedef enum
{
    SLVS_ERR_CHECK_MODE_NONE = 0x0,
    SLVS_ERR_CHECK_MODE_CRC = 0x1,
    SLVS_ERR_CHECK_MODE_ECC_2BYTE = 0x2,
    SLVS_ERR_CHECK_MODE_ECC_4BYTE = 0x3,

    SLVS_ERR_CHECK_MODE_BUTT
} slvs_err_check_mode_t;
```

【成员】

成员名称	描述
SLVS_ERR_CHECK_MODE_NONE	ECC 与 CRC 都不使能。
SLVS_ERR_CHECK_MODE_CRC	CRC 使能。
SLVS_ERR_CHECK_MODE_ECC_2BYTE	2 Byte ECC 使能。
SLVS_ERR_CHECK_MODE_ECC_4BYTE	4 Byte ECC 使能。

【芯片差异】

无。

【注意事项】

SLVS ECC 与 CRC 功能是互斥的，不能同时打开。

【相关数据类型及接口】

slvs_dev_attr_t

data_type_t

【说明】

传输的数据类型。

【定义】

Hi3519AV100/Hi3516CV500：

```
typedef enum
{
    DATA_TYPE_RAW_8BIT = 0,
    DATA_TYPE_RAW_10BIT,
    DATA_TYPE_RAW_12BIT,
    DATA_TYPE_RAW_14BIT,
    DATA_TYPE_RAW_16BIT,
    DATA_TYPE_YUV420_8BIT_NORMAL,
    DATA_TYPE_YUV420_8BIT_LEGACY,
    DATA_TYPE_YUV422_8BIT,
    DATA_TYPE_YUV422_PACKED,
    DATA_TYPE_BUTT
} data_type_t;
```

其他芯片：

```
typedef enum
{
    DATA_TYPE_RAW_8BIT = 0,
    DATA_TYPE_RAW_10BIT,
```

```
    DATA_TYPE_RAW_12BIT,
    DATA_TYPE_RAW_14BIT,
    DATA_TYPE_RAW_16BIT,
    DATA_TYPE_YUV420_8BIT_NORMAL,
    DATA_TYPE_YUV420_8BIT_LEGACY,
    DATA_TYPE_YUV422_8BIT,
    DATA_TYPE_BUTT
} data_type_t
```

【成员】

成员名称	描述
DATA_TYPE_RAW_8BIT	8BIT 的 RAW 数据。
DATA_TYPE_RAW_10BIT	10BIT 的 RAW 数据。
DATA_TYPE_RAW_12BIT	12BIT 的 RAW 数据。
DATA_TYPE_RAW_14BIT	14BIT 的 RAW 数据。
DATA_TYPE_RAW_16BIT	16BIT 的 RAW 数据。
DATA_TYPE_YUV420_8BIT_NORMAL	8BIT 的 YUV420 数据，NORMAL 模式。
DATA_TYPE_YUV420_8BIT_LEGACY	8BIT 的 YUV420 数据，LEGACY 模式。
DATA_TYPE_YUV422_8BIT	8BIT 的 YUV422 数据。
DATA_TYPE_YUV422_PACKED	YUV422 数据在 MIPI 按 16bit raw 打包接收。

【芯片差异】

芯片类型	支持的数据类型
Hi3559AV100ES	DATA_TYPE_RAW_8BIT DATA_TYPE_RAW_10BIT DATA_TYPE_RAW_12BIT DATA_TYPE_RAW_14BIT DATA_TYPE_RAW_16BIT
Hi3559AV100	除 DATA_TYPE_YUV422 PACKED 外都支持
Hi3519AV100/Hi3516CV500	全部支持
Hi3516EV200	DATA_TYPE_RAW_8BIT DATA_TYPE_RAW_10BIT DATA_TYPE_RAW_12BIT DATA_TYPE_RAW_14BIT

【注意事项】

无。

【相关数据类型及接口】

无。

mipi_wdr_mode_t

【说明】

MIPI WDR 模式。

【定义】

```
typedef enum
{
    HI_MIPI_WDR_MODE_NONE = 0x0,
    HI_MIPI_WDR_MODE_VC   = 0x1,    /* Virtual Channel */
    HI_MIPI_WDR_MODE_DT   = 0x2,    /* Data Type */
    HI_MIPI_WDR_MODE_DOL  = 0x3,    /* DOL Mode */
    HI_MIPI_WDR_MODE_BUTT
} mipi_wdr_mode_t;
```

【成员】

成员名称	描述
HI_MIPI_WDR_MODE_NONE	线性模式
HI_MIPI_WDR_MODE_VC	使用 Packet header 中的 Virtual Channel 区分长短曝光帧
HI_MIPI_WDR_MODE_DT	使用 Packet header 中的自定义 Data type 区分长短曝光帧
HI_MIPI_WDR_MODE_DOL	表示 DOL 模式 WDR，使用 Packet header 之后的一个 pixel 识别长短曝光帧

【芯片差异】

无

【注意事项】

无

【相关数据类型及接口】

无

mipi_dev_attr_t

【说明】

mipi 设备属性。

【定义】

```
typedef struct
{
    data_type_t          input_data_type;
    mipi_wdr_mode_t      wdr_mode;
    short                lane_id[MIPI_LANE_NUM];
    union
    {
        short data_type[WDR_VC_NUM];
    };
} mipi_dev_attr_t;
```

【成员】

成员名称	描述
input_data_type	传输的数据类型。
lane_id	发送端(sensor)和接收端(MIPI Rx) Lane 的对应关系 未使用的 Lane 设置为-1。
wdr_mode	MIPI WDR 模式
data_type	当 wdr_mode 为 HI_MIPI_WDR_MODE_DT 时，需要设置 data_type，表示不同曝光长度数据对应的 Data Type。

【芯片差异】

无。

【注意事项】

- Hi3516CV500 的 2L+2L 模式时，因 MIPI CH1 只有两条有效的 LANE，分别连接到了 PHY 的 lane1 和 lane3，所以 lane_id 需配置为{0, 1, -1, -1}或{1,3,-1,-1}，若连接交叉了，需配置为{1, 0, -1, -1}或{3,1,-1,-1}。
- Hi3516EV200、Hi3518EV300 只有两条 LANE，分别为 lane0 和 lane2，所以 lane_id 配置为{0, 2, -1, -1}。
- Hi3516EV300 有 4 条 LANE，接 2lane sensor 时用 lane0 和 lane1， lane_id 配置为{0, 1, -1, -1}；接 4lane sensor 时 lane_id 配置为{0, 1, 2, 3}

【相关数据类型及接口】

- data_type_t
- mipi_wdr_mode_t

- HI_MIPI_SET_DEV_ATTR

wdr_mode_t

【说明】

LVDS WDR 模式。

【定义】

```
typedef enum
{
    HI_WDR_MODE_NONE    = 0x0,
    HI_WDR_MODE_2F      = 0x1,
    HI_WDR_MODE_3F      = 0x2,
    HI_WDR_MODE_4F      = 0x3,
    HI_WDR_MODE_DOL_2F  = 0x4,
    HI_WDR_MODE_DOL_3F  = 0x5,
    HI_WDR_MODE_DOL_4F  = 0x6,
    HI_WDR_MODE_BUTT
} wdr_mode_t;
```

【成员】

成员名称	描述
HI_WDR_MODE_NONE	线性模式
HI_WDR_MODE_2F	2 合一 WDR
HI_WDR_MODE_3F	3 合一 WDR
HI_WDR_MODE_4F	4 合一 WDR
HI_WDR_MODE_DOL_2F	DOL 模式 2 合一 WDR
HI_WDR_MODE_DOL_3F	DOL 模式 3 合一 WDR
HI_WDR_MODE_DOL_4F	DOL 模式 4 合一 WDR

【芯片差异】

芯片类型	WDR 模式
Hi3559AV100ES/Hi3559AV100/Hi3519AV100/Hi3516CV500	都支持
Hi3516EV200/Hi3518EV300	支持 2 合一帧模式 WDR
Hi3516EV300	都支持

【注意事项】

- DOL WDR 模式需要配置为 HI_WDR_MODE_DOL_2F/ HI_WDR_MODE_DOL_3F/ HI_WDR_MODE_DOL_4F。
- Built-in WDR 模式和帧合成 WDR 模式都需要配置为 HI_WDR_MODE_NONE。

【相关数据类型及接口】

无。

lvds_sync_mode_t

【说明】

LVDS 同步方式。

【定义】

```
typedef enum
{
    LVDS_SYNC_MODE_SOF = 0,        /* sensor SOL, EOL, SOF, EOF */
    LVDS_SYNC_MODE_SAV,            /* SAV, EAV */
    LVDS_SYNC_MODE_BUTT
} lvds_sync_mode_t;
```

表1-6 LVDS 同步方式

sync_mode	同步方式
LVDS_SYNC_MODE_SOF	SOF、EOF、SOL、EOL 请参考图 1-1
LVDS_SYNC_MODE_SAV	invalid SAV、invalid EAV、valid SAV、valid EAV 请参考图 1-2

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

lvds_bit_endian_t

【说明】

比特位大小端模式

【定义】

```
typedef enum
{
    LVDS_ENDIAN_LITTLE  = 0x0,
    LVDS_ENDIAN_BIG     = 0x1,
    LVDS_ENDIAN_BUTT
} lvds_bit_endian_t;
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

lvds_vsync_type_t

【说明】

LVDS vsync 类型。

【定义】

```
typedef enum
{
    LVDS_VSYNC_NORMAL   = 0x00,
    LVDS_VSYNC_SHARE    = 0x01,
    LVDS_VSYNC_HCONNECT = 0x02,
    LVDS_VSYNC_BUTT
} lvds_vsync_type_t;
```

【成员】

成员名称	描述
LVDS_VSYNC_NORMAL	长短曝光帧有独立的 SOF-EOF、SOL-EOL 或者 invalid SAV-invalid EAV, valid SAV-valid EAV。
LVDS_VSYNC_SHARE	长短曝光帧共用一对 SOF-EOF 标识，短曝光的起始几行用固定值填充。
LVDS_VSYNC_HCONNECT	长短曝光帧共用一对 SAV-EAV 标识，长短曝光帧之间是固定周期的消隐。

- LVDS_VSYNC_SHARE 同步方式：

SOF	Long Exposure	EOL	Horizontal Blanking	SOL	Padding	EOL	Horizontal Blanking
SOL					Short Exposure		
	Padding					EOF	
SOV	V.BLK	EOV	-	SOV	V.BLK	EOV	-

- LVDS_VSYNC_HCONNECT 同步方式：

SAV	Long Exposure frame	Horizontal Blanking(fix period)	V.BLK	Horizontal Blanking(fix period)	V.BLK	EAV	Horizontal Blanking
	V.BLK		Short ExposureFrame1		Short Exposure Frame2		
			V.BLK				
	V.BLK						-

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

lvds_vsync_attr_t

lvds_vsync_attr_t

【说明】

LVDS vsync 参数

【定义】

```
typedef struct
{
    lvds_vsync_type_t sync_type;

    unsigned short hblank1;
    unsigned short hblank2;
} lvds_vsync_attr_t;
```

【芯片差异】

无。

【注意事项】

当 sync_type 为 LVDS_VSYNC_HCONNECT 时，需要配置 hblank1 和 hblank2，表示 Hconnect 的消隐区长度。

【相关数据类型及接口】

lvds_vsync_type_t

lvds_fid_type_t

【说明】

Frame identification Id 类型

【定义】

```
typedef enum
{
    LVDS_FID_NONE    = 0x00,
    LVDS_FID_IN_SAV   = 0x01,    /* frame identification id in SAV 4th */
    LVDS_FID_IN_DATA  = 0x02,    /* frame identification id in first data
*/
    LVDS_FID_BUTT
} lvds_fid_type_t;
```

【成员】

成员名称	描述
LVDS_FID_NONE	不使用 frame identification id。
LVDS_FID_IN_SAV	FID 插入在 SAV 第 4 个字段中，DOL 4 个字段的同步码需要将 fid_type 配置为 LVDS_FID_IN_SAV。

成员名称	描述
LVDS_FID_IN_DATA	FID 作为 Frame information column 插入在同步码之后的第一个像素之前，DOL 5 个字段的同步码需要将 fid_type 配置为 LVDS_FID_IN_DATA。

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

lvds_fid_attr_t

【说明】

Frame indentification Id 配置信息。

【定义】

```
typedef struct
{
    lvds_fid_type_t fid_type;
    HI_BOOL output_fil;
} lvds_fid_attr_t;
```

【成员】

成员名称	描述
fid_type	LVDS DOL 模式下的 Frame identification Id 类型
output_fil	DOL 模式中的 Frame information line 紧接在 V-Blanking 之后输出，Frame ID 是 Frame information line 中的第一个像素值。 Frame information line 中并不包含有效的视频数据： • 如果 output_fil 设置为 HI_TRUE，Frame information line 会输出到后端设备。 • 如果 output_fil 设置为 HI_FALSE，MIPI Rx 将丢弃这一行数据。

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

lvds_fid_type_t

mipi_clk_mode_t

【说明】

时钟共享模式，LVDS 模式跨 PHY 的时候时钟的选择，可以选择使用 LANE 所在的最小编号 PHY 的时钟，也可以选择每个 PHY 使用自己独立的时钟。

【定义】

```
typedef enum
{
    MIPI_CLK_MODE_SHARE  = 0x0,
    MIPI_CLK_MODE_SEPARATE = 0x1,
    MIPI_CLK_MODE_BUTT
} mipi_clk_mode_t;
```

【成员】

成员名称	描述
MIPI_CLK_MODE_SHARE	PHY 共享某一个 PHY 的时钟。
MIPI_CLK_MODE_SEPARATE	PHY 使用自己独立的时钟。

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

lvds_dev_attr_ex_t

lvds_lane_work_mode_t

【说明】

LVDS LANE 的同步码是否同步到达。

【定义】

```
typedef enum
{
```

```
    LVDS_LANE_WORK_MODE_SYNC   = 0x0,
    LVDS_LANE_WORK_MODE_ASYNC  = 0x1,
    LVDS_LANE_WORK_MODE_BUTT
} lvds_lane_work_mode_t;
```

【成员】

成员名称	描述
LVDS_LANE_WORK_MODE_SYNC	LVDS LANE 的同步码同步到达。
LVDS_LANE_WORK_MODE_ASYNC	LVDS LANE 的同步码异步到达。

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

lvds_lane_work_attr_t

lvds_lane_work_attr_t

【说明】

LVDS LANE 的 LANE 的同步码工作属性。

【定义】

```
typedef struct
{
    lvds_lane_work_mode_t lane_work_mode;
    unsigned int          sensor_valid_width;
} lvds_lane_work_attr_t;
```

【成员】

成员名称	描述
lane_work_mode	LANE 同步码工作模式。
sensor_valid_width	一行数据包的像素个数（RAW H）。

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

lvds_dev_attr_ex_t

lvds_dev_attr_t

【说明】

LVDS/SubLVDS/HiSPi 设备属性。

【定义】

```
typedef struct
{
    data_type_t         input_data_type;
    wdr_mode_t          wdr_mode;
    lvds_sync_mode_t    sync_mode;
    lvds_vsync_attr_t   vsync_attr;
    lvds_fid_attr_t     fid_attr;
    lvds_bit_endian_t   data_endian;
    lvds_bit_endian_t   sync_code_endian;
    short               lane_id[LVDS_LANE_NUM];
    unsigned short      sync_code[LVDS_LANE_NUM][WDR_VC_NUM][SYNC_CODE_NUM];
} lvds_dev_attr_t;
```


【成员】

成员名称	描述
input_data_type	传输的数据类型。
wdr_mode	WDR 模式。
sync_mode	LVDS 同步模式。
vsync_ attr	vsync 类型，当 wdr_mod 为 DOL 模式并且 sync_mode 为 LVDS_SYNC_MODE_SAV 时，需要配置 vsync 的类型。
fid_attr	frame identification 类型，当 wdr_mode 为 DOL 模式，并且 sync_mode 为 LVDS_SYNC_MODE_SAV 时，需要配置。
data_endian	数据大小端模式。
sync_code_endian	同步码大小端模式。
lane_id	发送端(sensor)和接收端(MIPI Rx) lane 的对应关系 未使用的 lane 设置为-1 lane id 的配置方式请参考“Lane id 如何配置”。

成员名称	描述
sync_code	每个 Virtual Channel 有 4 个同步码，根据同步模式不同，分别表示 SOF/EOF/ SOL/EOL 的同步码或者 invalid SAV/invalid EAV/ valid SAV/valid EAV 的同步码。

【芯片差异】

无。

【注意事项】

使用该结构体，SLVS 时钟共享，LVDS 的 LANE 同步码同步到达。

【相关数据类型及接口】

- wdr_mode_t
- lvds_sync_mode_t
- data_type_t
- lvds_bit_endian_t
- lvds_vsync_type_t
- lvds_fid_type_t
- HI_MIPI_SET_DEV_ATTR

lvds_dev_attr_ex_t

【说明】

LVDS/SubLVDS/HiSPi 设备高级属性。

【定义】

```
typedef struct
{
    mipi_clk_mode_t         clk_mode;
    lvds_dev_attr_t         lvds_attr;
    lvds_lane_work_attr_t   lane_work_attr;
} lvds_dev_attr_ex_t;
```

【成员】

成员名称	描述
clk_mode	传输的数据类型。
lvds_attr	WDR 模式。
lane_work_attr	LVDS 同步模式。

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

- wdr_mode_t
- lvds_sync_mode_t
- data_type_t
- lvds_bit_endian_t
- lvds_vsync_type_t
- lvds_fid_type_t
- HI_MIPI_SET_DEV_ATTR

slvs_dev_attr_t

【说明】

SLVS 设备属性。

【定义】

- 其他芯片：

```
typedef struct
{
    data_type_t          input_data_type;
    wdr_mode_t           wdr_mode;
    slvs_lane_rate_t      lane_rate
    int                  sensor_valid_width;
    short                lane_id[LVDS_LANE_NUM];
} slvs_dev_attr_t;
```

- Hi3559AV100：

```
typedef struct
{
    data_type_t         input_data_type;
    wdr_mode_t           wdr_mode;
    slvs_lane_rate_t     lane_rate
    int                  sensor_valid_width;
    short                lane_id[LVDS_LANE_NUM];
    slvs_err_check_mode_t err_check_mode;
} slvs_dev_attr_t;
```

【成员】

成员名称	描述
input_data_type	传输的数据类型。
wdr_mode	WDR 模式。
lane_rate	SLVS Lane 速率。
sensor_valid_width	一行数据包的像素个数（RAW H）。
lane_id	发送端(sensor)和接收端(SLVS) Lane 的对应关系。 未使用的 lane 设置为-1。
err_check_mode	SLVS 的 CRC，ECC 模式，必须与 sensor 端匹配。

【芯片差异】

芯片类型	本结构体
Hi3559AV100ES/ Hi3559AV100/ Hi3519AV100/	支持
Hi3516CV500/ Hi3616EV200	不支持 SLVS

【注意事项】

SLVS 只能支持线性模式和 WDR2to1。

【相关数据类型及接口】

- data_type_t
- slvs_lane_rate_t
- HI_MIPI_SET_DEV_ATTR

phy_cmv_mode_t

【说明】

PHY 共模电压模式。

【定义】

```
typedef enum
{
    PHY_CMV_GE1200MV    = 0x00,
    PHY_CMV_LT1200MV    = 0x01,
    PHY_CMV_BUTT
} phy_cmv_mode_t;
```


【成员】

成员名称	描述
PHY_CMV_GE1200MV	PHY 共模电压大于等于 1200mv。
PHY_CMV_LT1200MV	PHY 共模电压小于 1200 mv。

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

phy_cmv_t

【说明】

PHY 共模电压配置信息。

【定义】

```
typedef struct
{
    combo_dev_t      devno;
    phy_cmv_mode_t   cmv_mode;
} phy_cmv_t;
```

【成员】

成员名称	描述
devno	MIPI Rx 设备号。
cmv_mode	PHY 功能电压模式。

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

- phy_cmv_mode_t

- HI_MIPI_SET_PHY_CMVMODE

combo_dev_attr_t

【说明】

combo 设备属性，由于 MIPI Rx 能够对接 CSI-2、LVDS、HiSPi 等时序，所以将 MIPI Rx 称为 combo 设备。

【定义】

- 其他芯片：

```
typedef struct
{
    combo_dev_t             devno;
    input_mode_t          input_mode;
    mipi_data_rate_t      data_rate;
    img_rect_t            img_rect;
    union
    {
        mipi_dev_attr_t     mipi_attr;
        lvds_dev_attr_t     lvds_attr;
        slvs_dev_attr_t     slvs_attr;
    };
} combo_dev_attr_t;
```

Hi3559AV100：

```
typedef struct
{
    combo_dev_t             devno;
    input_mode_t          input_mode;
    mipi_data_rate_t      data_rate;
    img_rect_t            img_rect;
    union
    {
        mipi_dev_attr_t     mipi_attr;
        lvds_dev_attr_t     lvds_attr;
        slvs_dev_attr_t     slvs_attr;
        lvds_dev_attr_ex_t  lvds_attr_ex;
    };
} combo_dev_attr_t;
```


【成员】

成员名称	描述
devno	MIPI Rx，SLVS 设备号

成员名称	描述
devno	MIPI Rx，SLVS 设备号
input_mode	输入接口类型。
data_rate	接口传输速率。
img_rect	图像 crop 区域。
mipi_attr	如果 input_mode 配置为 INPUT_MODE_MIPI，则必须配置 mipi_attr。
lvds_attr	如果 input_mode 配置为 INPUT_MODE_SUBLVDS/ INPUT_MODE_LVDS/ INPUT_MODE_HISPI，则必须配置 lvds_attr
slvs_attr	如果 input_mode 配置为 INPUT_MODE_SLVS，则必须配置 slvs_attr。
lvds_attr_ex	如果 input_mode 配置为 INPUT_MODE_LVDS_EX，则必须配置 lvds_attr_ex

【芯片差异】

芯片类型	本结构体
Hi3559AV100ES/ Hi3559AV100/ Hi3519AV100/	支持
Hi3516CV500/ Hi3516EV200	不支持 SLVS

【注意事项】

无。

【相关数据类型及接口】

无。

HI_MIPI_TX_IOC_MAGIC

【说明】

MIPI Rx ioctl 命令的幻数。

【定义】

```
#define HI_MIPI_TX_IOC_MAGIC   't'
```


【成员】

无

【芯片差异】

无

【注意事项】

无

【相关数据类型及接口】

无

LANE_MAX_NUM

【说明】

定义 MIPI Tx 支持的最大 Lane 数。

【定义】

```
#define LANE_MAX_NUM          4
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

output_mode_t

【说明】

MIPI Tx 输出模式。

【定义】

```
typedef enum
{
    OUTPUT_MODE_CSI           = 0x0,            /* csi mode */
    OUTPUT_MODE_DSI_VIDEO      = 0x1,           /* dsi video mode */
    OUTPUT_MODE_DSI_CMD        = 0x2,           /* dsi command mode */
    OUTPUT_MODE_BUTT
} output_mode_t;
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

video_mode_t

【说明】

MIPI Tx 视频模式。

【定义】

```
typedef enum
{
    BURST_MODE                      = 0x0,
    NON_BURST_MODE_SYNC_PULSES      = 0x1,
    NON_BURST_MODE_SYNC_EVENTS      = 0x2,
} video_mode_t;
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

output_format_t

【说明】

MIPI Tx 输出格式。

【定义】

```
typedef enum
{
    OUT_FORMAT_RGB_16_BIT          = 0x0,
    OUT_FORMAT_RGB_18_BIT          = 0x1,
    OUT_FORMAT_RGB_24_BIT          = 0x2,
    OUT_FORMAT_YUV420_8_BIT_NORMAL = 0x3,
    OUT_FORMAT_YUV420_8_BIT_LEGACY = 0x4,
    OUT_FORMAT_YUV422_8_BIT         = 0x5,
    OUT_FORMAT_BUTT
} output_format_t;
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

sync_info_t

【说明】

MIPI Tx 设备同步信息。

【定义】

```
typedef struct
{
    unsigned short  vid_pkt_size;
    unsigned short  vid_hsa_pixels;
    unsigned short  vid_hbp_pixels;
    unsigned short  vid_hline_pixels;
    unsigned short  vid_vsa_lines;
    unsigned short  vid_vbp_lines;
    unsigned short  vid_vfp_lines;
    unsigned short  vid_active_lines;
    unsigned short  edpi_cmd_size;
} sync_info_t;
```


【成员】

成员名称	描述
vid_pkt_size	接收包大小。
vid_hsa_pixels	输入行同步脉冲区像素个数。
vid_hbp_pixels	输入后消隐区像素个数。
vid_hline_pixels	检测到的每行总像素个数。
vid_vsa_lines	检测到的帧同步脉冲行数。
vid_vbp_lines	帧同步脉冲后消隐区行数。
vid_vfp_lines	帧同步脉冲前消隐区行数。
vid_active_lines	VACTIVE 行数。

成员名称	描述
edpi_cmd_size	写内存命令字节数。 video mode 时该值无效，command mode 时该值设为 hact。

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

combo_dev_cfg_t

【说明】

MIPI Tx 设备属性。

【定义】

```
typedef struct
{
    unsigned int        devno;
    short               lane_id[LANE_MAX_NUM];
    output_mode_t       output_mode;
    video_mode_t        video_mode;
    output_format_t     output_format;
    sync_info_t         sync_info;
    unsigned int        phy_data_rate;
    unsigned int        pixel_clk;
} combo_dev_cfg_t;
```


【成员】

成员名称	描述
devno	MIPI Tx 设备号
lane_id	发送端(sensor)和接收端(MIPI Rx) Lane 的对应关系 未使用的 Lane 设置为-1。
output_mode	MIPI Tx 输出模式。
video_mode	MIPI Tx 视频模式。
output_format	MIPI Tx 输出格式。

成员名称	描述
devno	MIPI Tx 设备号
sync_info	MIPI Tx 设备的同步信息。
phy_data_rate	MIPI Tx 输出速率，输出范围请参考《Hi35xx xx Camera SoC 用户指南》“Tx D-PHY”章节中，MIPI Tx 高速模式每个通道（lane）的速率范围的描述。
pixel_clk	像素时钟。单位为 KHz

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

cmd_info_t

【说明】

发送给 MIPI Tx 设备的命令信息。

【定义】

```
typedef struct
{
    unsigned int       devno;
    unsigned short     data_type;
    unsigned short     cmd_size;
    unsigned char      *cmd;
} cmd_info_t;
```

【成员】

成员名称	描述
devno	MIPI Tx 设备号
data_type	命令数据类型。
cmd_size	命令数据大小，范围：(0,200]。 单条指令时，cmd 置为 NULL 时、低八位对应数据 1、高八位对应数据 2。

成员名称	描述
devno	MIPI Tx 设备号
cmd	命令数据指针。 单条指令时，可置为 NULL；长指令时，可进行赋值。

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

get_cmd_info_t

【说明】

发送给 MIPI Tx 设备的命令信息。

【定义】

```
typedef struct
{
    unsigned int       devno;
    unsigned short     data_type;
    unsigned short     data_param;
    unsigned short     get_data_size;
unsigned char      *get_data;
} get_cmd_info_t;
```

【成员】

成员名称	描述
devno	MIPI Tx 设备号
data_type	命令数据类型。
data_param	数据参数，低八比特为第一个参数，高八比特为第二个参数、不用时填 0
get_data_size	预期获取的数据字节数，范围：(0,200]。
get_data	获取到的数据存放地址指针，需要用户分配。

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

1.6 MIPI Tx 模块参数

注意

本小节只适用于 Hi3516CV500 芯片。

smooth

smooth 模块参数用于实现平滑过渡，使用方法如下：

- 在 Linux 操作系统下，加载 MIPI Tx 驱动的时候可以控制是否复位 MIPI Tx 的逻辑。

  关掉 MIPI Tx 逻辑复位的操作方法为 insmod hi_mipi_tx.ko smooth=1。

  打开 MIPI Tx 逻辑复位操作方法为 insmod hi_mipi_tx.ko smooth=0。
- 在 Huawei LiteOS 系统下，传入 smooth 参数来控制 MIPI Tx 驱动是否复位逻辑。

  关掉逻辑复位的操作方法为 int smooth=1；MIPI_TX_init(smooth)。

  打开逻辑复位的操作方法为 int smooth=0；MIPI_TX_init(smooth)。

1.7 Proc 信息

1.7.1 MIPI_RX Proc 信息

MIPI_RX 正常工作状态下 proc 信息中宽高应该是稳定不变且和 sensor 输出时序的宽高匹配，并且 MIPI_RX 各种错误中断计数为 0。如果错误中断计数不为 0，请检查 MIPI_RX 相关属性是否配置正确。

Hi3559AV100

【调试信息】

```
Module: [MIPI], Build Time: [May 23 2017, 10:04:19]
-----MIPI LANE DIVIDE MODE-----------------------------------------------
  MODE        LANE DIVIDE
```

```
  7          4+4+4+4
-----MIPI DEV ATTR------------------------------------------------------
Devno  WorkMode  DataRate  DataType   WDRMode         LinkId    ImgX    ImgY
ImgW    ImgH
0      MIPI       X1     RAW12      None           0, 1      0       0    3840
2160

-----MIPI LANE INFO-----------------------------------------------------
   Devno   LaneCnt               LaneID
      0        4        0,  1,  2,  3, -1, -1, -1, -1

-----MIPI LINK INFO-----------------------------------------------------
 LinkIdx LaneCount  LaneId    PhyData0    PhyData1  AlignedData0
AlignedData1   ValidLane
      0        2    0, 1        0x0        0x0          0x0          0x0
Invalid
      1        2    2, 3        0x0        0x0          0x0          0x0
Invalid
-----MIPI DETECT INFO--------------------------------------------------
 Devno VC   width  height
    0  0    3840    2160
    0  1       0       0
    0  2       0       0
    0  3       0       0
-----LVDS DETECT INFO---------------------------------------------------
 Devno VC   width  height
    0  0    5480    3648
    0  1       0       0
    0  2       0       0
    0  3       0       0
-----LVDS LANE DETECT INFO----------------------------------------------
 Devno  Lane   width  height
    0     0     548    3689
    0     1     548    3689
    0     2     548    3689
    0     3     548    3689
    0     4     548    3689
    0     5     548    3689
    0     6     548    3689
    0     7     548    3689
    0     8     548    3689
    0    10     548    3689
-----FSM TIMEOUT AND ESCAPE INFO---------------------------------------
phy clkTOutCnt d0TOutCnt   d1TOutCnt   d2TOutCnt   d3TOutCnt clkEscCnt
```

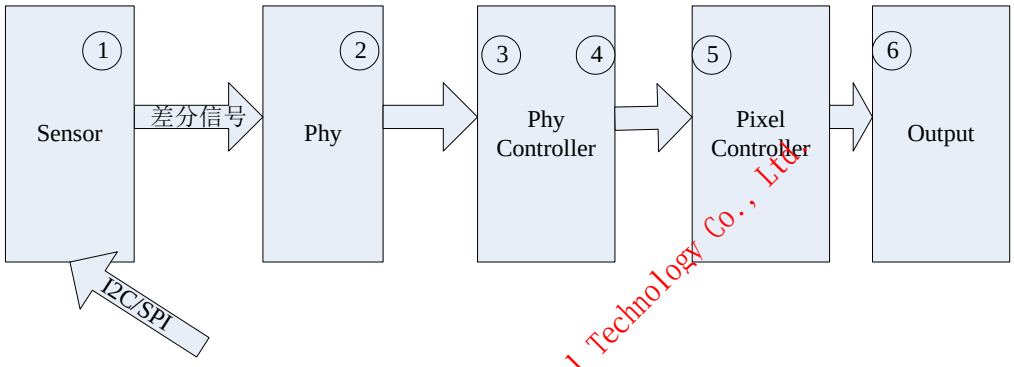
```
d0EscCnt  d1EscCnt  d2EscCnt  d3EscCnt
    0     0           0           0           0           0         0         0
0         0         0
    1     0           0           0           0           0         0         0
0         0         0
    2     0           0           0           0           0         0         0
0         0         0
    3     0           0           0           0           0         0         0
0         0         0
-----MIPI INT ERROR INFO-------------------------------------------------
Devno vc0CRC vc1CRC vc2CRC vc3CRC vc0OrderErr vc1OrderErr vc2OrderErr
vc3OrderErr vc0NMatCnt vc1NMatCnt vc2NMatCnt vc3NMatCnt
0      0      0      0      0           0           0           0           0
0      0      0      0

Devno HCntErr vc0HECC vc1HECC vc2HECC vc3HECC vc0DtErr vc1DtErr vc2DtErr
vc3DtErr
  0       0       0       0       0       0       0       0       0       0

Devno CMD_FIFO_RERR DATA_FIFO_RERR CMD_FIFO_WERR DATA_FIFO_WERR
     0            0            0            0            0
-----SLVS DEV ERROR INFO-------------------------------------------------
Devno  HeaderCRC  PayloadCRC  EccErr  DataFifoWrite   DataFifoRead
CmdFifoFull   SkewErr
       6         0         0       0             0             0
0         1

-----ALING ERROR INFO-----------------------------------
Devno FIFO_FullErr  Lane0Err  Lane1Err  Lane2Err  Lane3Err  Lane4Err
Lane5Err  Lane6Err  Lane7Err  Lane8Err  Lane9Err Lane10Err Lane11Err
Lane12Err Lane13Err Lane14Err Lane15Err
    0           0         0         0         0         0         0         0
0         0         0         0         0         0         0         0         0
0

-----SLVS DEV ATTR-------------------------------------------------------
   Devno  WorkMode  DataRate  DataType   WDRMode         LinkId    ImgX
ImgY    ImgW    ImgH  LaneRate    ValidW  ErrMode
       6      SLVS        X2     RAW10      None      0, 1, 2, 3    164      41
3840    2160       LOW      4144  NONE

-----SLVS LANE INFO------------------------------------------------------
   Devno   LaneCnt               LaneID
       6         8         0,  1,  2,  3,  4,  5,  6,  7
```

```
-----SLVS DATA INFO----------------------------------------------------
   Devno    LaneId     PhyData   AlignedData   ValidLane
      0         0       0x38b        0x72         0,
      0         1       0x2d4       0x2a4         1,
      0         2       0x12d       0x38a         2,
      0         3       0x23a       0x32a         3,
      0         4        0x4d        0xe4         4,
      0         5       0x1a3       0x18b         5,
      0         6       0x109       0x2e4         6,
      0         7       0x21f       0x1b4         7,
-----SLVS DETECT INFO--------------------------------------------------
   Devno  VC   width  height
      6   0    3840    2160
      6   1       0       0
      6   2       0       0
      6   3       0       0

-----SLVS DEV ERROR INFO-----------------------------------------------
   Devno  HeaderCRC  PayloadCRC  EccErr  DataFifoWrite    DataFifoRead
CmdFifoFull   SkewErr
      6         0          0       0              0              0
0         1

-----SLVS PHY ERROR INFO-----------------------------------------------
  PhyIdx   LaneIdx  AFifoAlign   CodeErr   DispErr
      0         0           1         3         3
      0         1           1         3         3
      0         2           1         3         3
      0         3           1         3         3
      0         4           0         3         3
      0         5           0         3         3
      0         6           0         3         3
      0         7           0         3         3
      1         8           0         0         0
      1         9           0         0         0
      1        10           0         0         0
      1        11           0         0         0
      1        12           0         0         0
      1        13           0         0         0
      1        14           0         0         0
      1        15           0         0         0
```

【调试信息分析】

- MIPI_Rx 通过 Phy 接收 sensor 的差分数据，Phy Controller 检测到同步头后，将每条 lane 上的数据对齐；
- Pixel Controller 解析同步信息并按照 raw data 的位宽将 lane 上面的数据合并为 Pixel 数据；Output 模式将 Pixel 数据发送给后级模块。
- Phy PhyController PixelController 由 sensor 的 pixel clk 提供时钟，output 模块的时钟为称为随路时钟，与后级模块的工作时钟相同。MIPI_Rx 的 crop 功能在 Pixel Controller 的末端实现，所以 Crop 后可以降低需要的随路时钟。

图1-3 MIPI 数据流



【参数说明】

参数		描述
MIPI LANE DIVIDE MODE	MODE	MIPI Rx 的 lane 分布模式
	LANE DIVIDE	MIPI_Rx 详细的 LANE 分布。
MIPI DEV ATTR	Devno	MIPI 设备号
	WorkMode	MIPI 设备工作模式：LVDS/MIPI/CMOS/SLVS 等模式
	DataRate	MIPI Rx 的速率。
	DataType	数据类型：RAW8 / RAW10/ RAW12/ RAW14/ RAW16 bit 等类型

参数		描述
	WDRMode	WDR 模式： • None：非 WDR 模式 • 2To1：2 合 1 WDR • 3To1：3 合 1 WDR • 4To1：4 合 1 WDR • DOL2To1：DOL2 合 1WDR • DOL3To1：DOL3 合 1WDR • DOL4To1：DOL4 合 1WDR
	LinkId	此设备使用了哪几个 Link，对应 Link ID。 一个物理 Link 对应 4 个 Lane。
	ImgX	Crop 图像起始 X
	ImgY	Crop 图像起始 Y
	ImgW	Crop 图像宽度
	ImgH	Crop 图像高度
MIPI LANE INFO	Devno	MIPI 设备号
	LaneCnt	Lane 数目
	LaneID	Lane ID
MIPI LINK INFO	LinkIdx	Link ID 序号
	LaneCount	该 Link 中使用了几条 Lane
	LaneId	对应的 Lane Id
	PhyData0	PHY 接收到的实时数据 0
	PhyData1	PHY 接收到的实时数据 1
	AlignedData0	检测到帧同步信号后的实时数据 0
	AlignedData1	检测到帧同步信号后的实时数据 1
	ValidLane	Link 内部有效 Lane ID，对于 MIPI 模式来说这个值是动态变化的，有时候有值有时候为 Invalid。
MIPI DETECT INFO (仅 MIPI 模式下可见)	Devno	MIPI_Rx 设备号
	VC	Virtual Channel
	width	MIPI 控制器检测到的图像总宽度
	height	MIPI 控制器检测到的图像总高度

参数		描述
LVDS DETECT INFO	Devno	MIPI_Rx 设备号
	VC	Virtual Channel
	width	MIPI 控制器检测到的图像总宽度
	height	MIPI 控制器检测到的图像总高度
LVDS LANE DETECT INFO	Devno	MIPI_Rx 设备号
	Lane	Lane ID
	width	该 lane 上检测到的宽度。
	height	该 lane 上检测到的高度。
FSM TIMEOUT AND ESCAPE INFO (仅 MIPI 模式下可见)	phy	PHY ID
	clkTOutCnt	时钟 Lane 从 LP 切换到 HS 超时
	d0TOutCnt	数据 Lane 0 从 LP 切换到 HS 超时
	d1TOutCnt	数据 Lane 1 从 LP 切换到 HS 超时
	d2TOutCnt	数据 Lane 2 从 LP 切换到 HS 超时
	d3TOutCnt	数据 Lane 3 从 LP 切换到 HS 超时
	clkEscCnt	时钟 Lane 切换到 escape 模式超时
	d0EscCnt	数据 Lane 0 切换到 escape 模式超时
	d1EscCnt	数据 Lane 1 切换到 escape 模式超时
	d2EscCnt	数据 Lane 2 切换到 escape 模式超时
	d3EscCnt	数据 Lane 3 切换到 escape 模式超时
MIPI INT ERR INFO(仅 MIPI 模式下可见)	Devno	MIPI_Rx 设备号。
	vc0CRC	VC0 通道数据的 CRC 错误计数。
	vc1CRC	VC1 通道数据的 CRC 错误计数。
	vc2CRC	VC2 通道数据的 CRC 错误计数。
	vc3CRC	VC3 通道数据的 CRC 错误计数。
	vc0OrderErr	VC0 的帧序错误计数。
	vc1OrderErr	VC1 的帧序错误计数。
	vc2OrderErr	VC2 的帧序错误计数。
	vc3OrderErr	VC3 的帧序错误计数。
	vc0NMatCnt	VC0 通道的帧起始与帧结束短包不匹配计数。

参数		描述
	vc1NMatCnt	VC1 通道的帧起始与帧结束短包不匹配计数。
	vc2NMatCnt	VC2 通道的帧起始与帧结束短包不匹配计数。
	vc3NMatCnt	VC3 通道的帧起始与帧结束短包不匹配计数。
	HCntErr	Header 的 ECC 无法纠错的错误计数。
	vc0HECC	VC0 通道 Header 的 ECC 已纠正的错误计数。
	vc1HECC	VC1 通道 Header 的 ECC 已纠正的错误计数。
	vc2HECC	VC2 通道 Header 的 ECC 已纠正的错误计数。
	vc3HECC	VC3 通道 Header 的 ECC 已纠正的错误计数。
	vc0DtErr	VC0 通道不支持的数据类型计数。
	vc1DtErr	VC1 通道不支持的数据类型计数。
	vc2DtErr	VC2 通道不支持的数据类型计数。
	vc3DtErr	VC3 通道不支持的数据类型计数。
	CMD_FIFO_RERR	MIPI 读命令 FIFO 原始中断计数。
	DATA_FIFO_RERR	MIPI 读数据 FIFO 原始中断计数。
	CMD_FIFO_WERR	MIPI 写命令 FIFO 原始中断计数。
	DATA_FIFO_WERR	MIPI 写数据 FIFO 原始中断计数。
ALING ERROR INFO	Devno	MIPI 设备号
	FIFO_FullErr	FIFO 溢出
	Lane0Err	Lane0 FIFO 溢出
	Lane1Err	Lane1 FIFO 溢出
	Lane2Err	Lane2 FIFO 溢出
	Lane3Err	Lane3 FIFO 溢出
	Lane4Err	Lane4 FIFO 溢出
	Lane5Err	Lane5 FIFO 溢出
	Lane6Err	Lane6 FIFO 溢出
	Lane7Err	Lane7 FIFO 溢出
	Lane8Err	Lane8 FIFO 溢出
	Lane9Err	Lane9 FIFO 溢出
	Lane10Err	Lane10 FIFO 溢出

参数		描述
	Lane11Err	Lane11 FIFO 溢出
	Lane12Err	Lane12 FIFO 溢出
	Lane13Err	Lane13 FIFO 溢出
	Lane14Err	Lane14 FIFO 溢出
	Lane15Err	Lane15 FIFO 溢出
SLVS DEV ATTR	Devno	SLVS 设备号
	WorkMode	SLVS 设备工作模式： LVDS/MIPI/CMOS/SLVS 等模式
	DataRate	SLVS 的速率。
	DataType	数据类型： RAW8 / RAW10/ RAW12/ RAW14/ RAW16 bit 等类型
	WDRMode	WDR 模式： • None：非 WDR 模式 • 2To1：2 合 1 WDR • 3To1：3 合 1 WDR • 4To1：4 合 1 WDR • DOL2To1：DOL2 合 1WDR • DOL3To1：DOL3 合 1WDR DOL4To1：DOL4 合 1WDR
	LinkId	此设备使用了哪几个 Link，对应 Link ID。 一个物理 Link 对应 4 个 lane。
	ImgX	Crop 图像起始 X
	ImgY	Crop 图像起始 Y
	ImgW	Crop 图像宽度
	ImgH	Crop 图像高度
	LaneRate	Lane 的速率
	ValidW	SENSOR 的实际有效宽度
	ErrMode	SLVS 的 CRC ECC 模式。
SLVS LANE	Devno	SLVS 设备号
	LaneCnt	Lane 数目

参数		描述
INFO	LaneID	Lane ID
SLVS DATA INFO	Devno	SLVS 设备号
	LaneID	Lane ID
	PhyData	PHY 对应的 LANE 接收到的实时数据
	AlignedData	PHY 对应的 LANE 检测到同步码后接收到的实时数据
	ValidLane	PHY 内部有效 Lane ID，对于 LVDS 模式来说这个值是动态变化的，有时候有值有时候为 Invalid。
SLVS DETECT INFO	Devno	SLVS 设备号
	VC	Virtual Channel
	width	SLVS 控制器检测到的图像总宽度
	height	SLVS 控制器检测到的图像总高度
SLVS DEV ERROR INFO	Devno	SLVS 设备号
	HeaderCRC	数据头 CRC 错误统计
	PayloadCRC	数据 CRC 错误统计
	EccErr	ECC 错误统计
	DataFifoWrite	数据 FIFO 写错误统计
	DataFifoRead	数据 FIFO 读错误统计
	CmdFifoFull	命令 FIFO 满统计
	SkewErr	SKEW 错误统计
SLVS PHY ERROR INFO	PhyIdx	PHY ID
	LaneIdx	Lane ID
	AFifoAlign	FIFO 对齐错误统计
	CodeErr	10B8B 编码错误统计
	DispErr	DISPARITY 错误统计

Hi3519AV100

【调试信息】

```
Module: [MIPI_RX], Build Time[Jul 30 2018, 10:02:04]
```

```
-----MIPI LANE DIVIDE MODE-------------------------------------------------
 MODE        LANE DIVIDE
 5           4+4+2+2
-----MIPI DEV ATTR---------------------------------------------------------
----------------------------------------------
   Devno  WorkMode  DataRate    DataType   WDRMode    ImgX    ImgY    ImgW
ImgH
      0     MIPI       X1       RAW12      None       0       0    3840    2160

-----MIPI LANE INFO--------------------------------------------------------
----------------------------------------------
   Devno                LaneID
      0        0,  1,  2,  3, -1, -1, -1, -1

-----MIPI PHY DATA INFO----------------------------------------------------
----
   PhyId        LaneId          PhyData             MipiData
LvdsData
      0      0, 1, 2, 3    0x00,0x00,0xff,0xff    0xb5,0x9f,0x40,0x13
0xcc,0x88,0x38,0x43
      1      4, 5, 6, 7    0x00,0x00,0x00,0x00    0x00,0x00,0x00,0x00
0x00,0x00,0x00,0x00
      2      8, 9,10,11    0x00,0x00,0x00,0x00    0x00,0x00,0x00,0x00
0x00,0x00,0x00,0x00

-----MIPI DETECT INFO------------------------------------------------------
 Devno VC   width  height
    0  0    3840    2160
    0  1       0       0
    0  2       0       0
    0  3       0       0
-----LVDS DETECT INFO------------------------------------------------------
 Devno VC   width  height
    0  0    3840    2160
    0  1       0       0
    0  2       0       0
    0  3       0       0

-----LVDS LANE DETECT INFO-------------------------------------------------
 Devno  Lane   width  height
    0     2     960    2179
    0     4     960    2179
    0     5     960    2179
    0     7     960    2179
```

```
-----PHY CIL ERR INT INFO-----------------------------------------
   PhyId  Clk2TmOut  ClkTmOut  Lane0TmOut  Lane1TmOut  Lane2TmOut
Lane3TmOut  Clk2Esc  ClkEsc  Lane0Esc  Lane1Esc  Lane2Esc  Lane3Esc
      0         0        0          0          0          0          0       0
0        0        0        0        0
      1         0        0          0          0          0          0       0
0        0        0        0        0
      2         0        0          0          0          0          0       0
0        0        0        0        0
-----MIPI ERROR INT INFO 1----------------------------------------------
------------
   Devno  Ecc2  Vc0CRC  Vc1CRC  Vc2CRC  Vc3CRC  Vc0EccCorrct  Vc1EccCorrct
Vc2EccCorrct  Vc3EccCorrct
      0    0      0      0      0      0           0          0
0           0

-----MIPI ERROR INT INFO 2----------------------------------------------
   Devno  Vc0Dt  Vc1Dt  Vc2Dt  Vc3Dt  Vc0FrmCrc  Vc1FrmCrc  Vc2FrmCrc
Vc3FrmCrc
      0     0     0     0     0          0          0          0          0

-----MIPI ERROR INT INFO 3----------------------------------------------
   Devno  Vc0FrmSeq  Vc1FrmSeq  Vc2FrmSeq  Vc3FrmSeq  Vc0BndryMt
Vc1BndryMt  Vc2BndryMt  Vc3BndryMt
      0         0         0          0          0          0          0
0          0

-----MIPI ERROR INT INFO 4----------------------------------------------
-----------
   Devno  DataFifoRdErr  CmdFifoRdErr  Vsync  CmdFifoWrErr  DataFifoWrErr
      0              0           0     0           0           0
-----LVDS ERROR INT INFO 1----------------------------------------------
------------
   Devno  Vsync  CmdRdErr  CmdWrErr  PopErr  StatErr
      0     0        0         0       0       0

-----LVDS ERROR INT INFO 2----------------------------------------------
   Devno  Link0WrErr  Link1WrErr  Link2WrErr  Link0RdErr  Link1RdErr
Link2RdErr
      0          0          0          0          0          0          0

-----LVDS ERROR INT INFO 3----------------------------------------------
   Devno  Lane0Err  Lane1Err  Lane2Err  Lane3Err  Lane4Err  Lane5Err
Lane6Err  Lane7Err  Lane8Err  Lane9Err Lane10Err Lane11Err
```

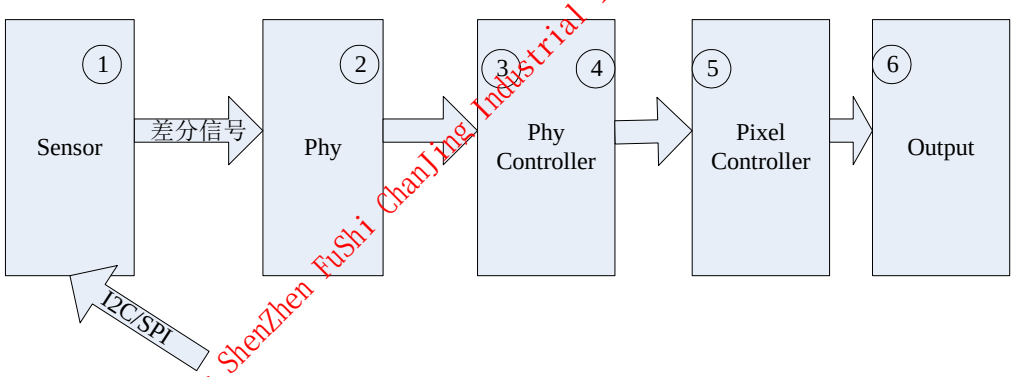
```
      0        0        0        0        0        0        0        0
0        0        0        0        0
-----ALIGN ERROR INT INFO--------------------------------------
   Devno  FIFO_FullErr  Lane0Err  Lane1Err  Lane2Err  Lane3Err  Lane4Err
Lane5Err  Lane6Err  Lane7Err  Lane8Err  Lane9Err Lane10Err Lane11Err
      0           0        0        0        0        0        0        0
0        0        0        0        0        0
-----SLVS DEV ATTR----------------------------------------------------------
   Devno  WorkMode  DataRate  DataType   WDRMode    ImgX    ImgY    ImgW
ImgH  LaneRate    ValidW
      0     SLVS       X2     RAW12      None      48      24    3840    2160
HIGHT     3936

-----SLVS LANE INFO---------------------------------------------------------
   Devno                LaneID
      0        6,  4,  5,  0,  7,  2, -1, -1

-----SLVS PHY DATA INFO-----------------------------------------------------
   PhyId    LaneID     PhyData   AlignedData
      0         0       0x227         0x345
      0         1         0x0           0x0
      0         2       0x30d         0x1c6
      0         3         0x0           0x0
      0         4       0x2b4         0x3a9
      0         5       0x368         0x18b
      0         6       0x25c         0x274
      0         7       0x1c4         0x134

-----SLVS DETECT INFO-------------------------------------------------------
   Devno  VC   width  height
      0   0    3840    2160
      0   1       0       0
      0   2       0       0
      0   3       0       0

-----SLVS DEV ERROR INFO----------------------------------------------------
Devno  HeaderCRC  PayloadCRC  EccErr   DataFifoWrite    DataFifoRead
CmdFifoFull   SkewErr
      0         0         0      0             0             0
0        0

-----SLVS PHY ERROR INFO----------------------------------------------------
  PhyIdx   LaneIdx  AFifoAlign   CodeErr   DispErr
      0         0         0         0         0
```

```
0    1    0    0    0
0    2    0    0    0
0    3    0    0    0
0    4    0    0    0
0    5    0    0    0
0    6    0    0    0
0    7    0    0    0
```

【调试信息分析】

- MIPI_Rx 通过 Phy 接收 sensor 的差分数据，Phy Controller 检测到同步头后，将每条 lane 上的数据对齐；
- Pixel Controller 解析同步信息并按照 raw data 的位宽将 lane 上面的数据合并为 Pixel 数据；Output 模式将 Pixel 数据发送给后级模块。
- Phy PhyController PixelController 由 sensor 的 pixel clk 提供时钟，output 模块的时钟为称为随路时钟，与后级模块的工作时钟相同。MIPI_Rx 的 crop 功能在 Pixel Controller 的末端实现，所以 Crop 后可以降低需要的随路时钟。

图1-4 MIPI 数据流



【参数说明】

参数		描述
MIPI LANE DIVIDE MODE	MODE	MIPI Rx 的 lane 分布模式
	LANE DIVIDE	MIPI_Rx 详细的 LANE 分布。
MIPI DEV ATTR	Devno	MIPI 设备号
	WorkMode	MIPI 设备工作模式：LVDS/MIPI/CMOS/SLVS 等模式
	DataRate	MIPI Rx 的速率。
	DataType	数据类型：RAW8 / RAW10/ RAW12/ RAW14/ RAW16 bit 等类型

参数		描述
	WDRMode	WDR 模式： • None：非 WDR 模式 • 2To1：2 合 1 WDR • 3To1：3 合 1 WDR • 4To1：4 合 1 WDR • DOL2To1：DOL2 合 1WDR • DOL3To1：DOL3 合 1WDR • DOL4To1：DOL4 合 1WDR
	ImgX	Crop 图像起始 X
	ImgY	Crop 图像起始 Y
	ImgW	Crop 图像宽度
	ImgH	Crop 图像高度
MIPI LANE INFO	Devno	MIPI 设备号
	LaneCnt	Lane 数目
	LaneID	Lane ID
MIPI PHY DATA INFO	PhyId	PHY ID 序号
	LaneId	对应的 Lane Id
	PhyData	PHY 对应的 4 条 Lane 接收到的实时数据
	MipiData	PHY 对应的 4 条 Lane 检测到 MIPI 帧同步信号后的实时数据
	LvdsData	PHY 对应的 4 条 Lane 检测到 LVDS 帧同步信号后的实时数据
MIPI DETECT INFO (仅 MIPI 模式下可见)	Devno	MIPI_Rx 设备号
	VC	Virtual Channel
	width	MIPI 控制器检测到的图像总宽度
	height	MIPI 控制器检测到的图像总高度
LVDS DETECT INFO (仅 LVDS 模式下可见)	Devno	MIPI_Rx 设备号
	VC	Virtual Channel
	width	MIPI 控制器检测到的图像总宽度
	height	MIPI 控制器检测到的图像总高度
LVDS	Devno	MIPI_Rx 设备号

参数		描述
LANE DETECT INFO (仅 LVDS 模式下可见)	Lane	Lane ID
	width	该 lane 上检测到的宽度。
	height	该 lane 上检测到的高度。
PHY CIL ERR INT INFO	phy	PHY ID
	Clk2TmOut	Clock2 Lane 从 LP 切换到 HS 超时
	ClkTmOut	Clock1 Lane 从 LP 切换到 HS 超时
	Lane0TmOut	DataLane 0 从 LP 切换到 HS 超时
	Lane1TmOut	Data Lane 1 从 LP 切换到 HS 超时
	Lane2TmOut	Data Lane 2 从 LP 切换到 HS 超时
	Lane3TmOut	Data Lane 3 从 LP 切换到 HS 超时
	Clk2Esc	Clock2 Lane 切换到 escape 模式超时
	ClkEsc	Clock Lane 切换到 escape 模式超时
	Lane0Esc	Data Lane 0 切换到 escape 模式超时
	Lane1Esc	Data Lane 1 切换到 escape 模式超时
	Lane2Esc	Data Lane 2 切换到 escape 模式超时
	Lane3Esc	Data Lane 3 切换到 escape 模式超时
MIPI ERROR INT INFO 1 (仅 MIPI 模式下可见)	Devno	MIPI_Rx 设备号。
	Ecc2	Header 至少 2 个错误，ECC 无法纠错
	Vc0CRC	VC0 通道数据的 CRC 错误计数。
	Vc1CRC	VC1 通道数据的 CRC 错误计数。
	Vc2CRC	VC2 通道数据的 CRC 错误计数。
	Vc3CRC	VC3 通道数据的 CRC 错误计数。
	Vc0EccCorrct	VC0 通道 Header 的 ECC 已纠正的错误计数。
	Vc1EccCorrct	VC1 通道 Header 的 ECC 已纠正的错误计数。
	Vc2EccCorrct	VC2 通道 Header 的 ECC 已纠正的错误计数。
	Vc3EccCorrct	VC3 通道 Header 的 ECC 已纠正的错误计数。
MIPI ERROR INT INFO	Devno	MIPI_Rx 设备号。
	Vc0Dt	VC0 通道不支持的数据类型计数

参数		描述
2 (仅 MIPI 模式下可见)	Vc1Dt	VC1 通道不支持的数据类型计数
	Vc2Dt	VC2 通道不支持的数据类型计数
	Vc3Dt	VC3 通道不支持的数据类型计数
	Vc0FrmCrc	VC0 通道帧数据错误计数
	Vc1FrmCrc	VC1 通道帧数据错误计数
	Vc2FrmCrc	VC2 通道帧数据错误计数
	Vc3FrmCrc	VC3 通道帧数据错误计数
MIPI ERROR INT INFO 3 (仅 MIPI 模式下可见)	Devno	MIPI_Rx 设备号。
	Vc0FrmSeq	VC0 的帧序错误计数
	Vc1FrmSeq	VC1 的帧序错误计数
	Vc2FrmSeq	VC2 的帧序错误计数
	Vc3FrmSeq	VC3 的帧序错误计数
	Vc0BndryMt	VC0 通道的帧起始与帧结束短包不匹配计数
	Vc1BndryMt	VC1 通道的帧起始与帧结束短包不匹配计数
	Vc2BndryMt	VC2 通道的帧起始与帧结束短包不匹配计数
	Vc3BndryMt	VC3 通道的帧起始与帧结束短包不匹配计数
MIPI ERROR INT INFO 4 (仅 MIPI 模式下可见)	Devno	MIPI_Rx 设备号。
	DataFifoRdErr	MIPI CTRL 读数据 FIFO 中断计数
	CmdFifoRdErr	MIPI CTRL 读命令 FIFO 中断计数
	Vsync	MIPI CTR vsync 中断计数
	CmdFifoWrErr	MIPI CTRL 写命令 FIFO 中断计数
	DataFifoWrErr	MIPI CTRL 写数据 FIFO 中断计数
LVDS ERROR INT INFO 1(仅 LVDS 模式下可见)	Devno	MIPI 设备号
	Vsync	lvds vsync 中断计数
	CmdRdErr	lvds CMD_FIFO 读错误中断计数
	CmdWrErr	lvds CMD_FIFO 写错误中断计数
	PopErr	lvds 读取 line_buf 错误中断计数
	StatErr	lvds 各 lane 同步错误中断计数
LVDS	Devno	MIPI 设备号

参数		描述
ERROR INT INFO 2(仅 LVDS 模式下可见)	Link0WrErr	link0 读 FIFO 错误中断计数
	Link1WrErr	link1 读 FIFO 错误中断计数
	Link2WrErr	link2 读 FIFO 错误中断计数
	Link0RdErr	link0 写 FIFO 错误中断计数
	Link1RdErr	link1 写 FIFO 错误中断计数
	Link2RdErr	link2 写 FIFO 错误中断计数
LVDS ERROR INT INFO 3(仅 LVDS 模式下可见)	Devno	MIPI 设备号
	Lane0Err	Lane0 同步错误中断计数
	Lane1Err	Lane1 同步错误中断计数
	Lane2Err	Lane2 同步错误中断计数
	Lane3Err	Lane3 同步错误中断计数
	Lane4Err	Lane4 同步错误中断计数
	Lane5Err	Lane5 同步错误中断计数
	Lane6Err	Lane6 同步错误中断计数
	Lane7Err	Lane7 同步错误中断计数
	Lane8Err	Lane8 同步错误中断计数
	Lane9Err	Lane9 同步错误中断计数
	Lane10Err	Lane10 同步错误中断计数
	Lane11Err	Lane11 同步错误中断计数
ALING ERROR INFO	Devno	MIPI 设备号
	FIFO_FullErr	FIFO 溢出
	Lane0Err	Lane0 FIFO 溢出
	Lane1Err	Lane1 FIFO 溢出
	Lane2Err	Lane2 FIFO 溢出
	Lane3Err	Lane3 FIFO 溢出
	Lane4Err	Lane4 FIFO 溢出
	Lane5Err	Lane5 FIFO 溢出
	Lane6Err	Lane6 FIFO 溢出
	Lane7Err	Lane7 FIFO 溢出

参数		描述
	Lane8Err	Lane8 FIFO 溢出
	Lane9Err	Lane9 FIFO 溢出
	Lane10Err	Lane10 FIFO 溢出
	Lane11Err	Lane11 FIFO 溢出
SLVS DEV ATTR	Devno	SLVS 设备号
	WorkMode	SLVS 设备工作模式： LVDS/MIPI/CMOS/SLVS 等模式
	DataRate	SLVS 的速率。
	DataType	数据类型： RAW8 / RAW10/ RAW12/ RAW14/ RAW16 bit 等类型
	WDRMode	WDR 模式： • None：非 WDR 模式 • 2To1：2 合 1 WDR • 3To1：3 合 1 WDR • 4To1：4 合 1 WDR • DOL2To1：DOL2 合 1WDR • DOL3To1：DOL3 合 1WDR DOL4To1：DOL4 合 1WDR
	ImgX	Crop 图像起始 X
	ImgY	Crop 图像起始 Y
	ImgW	Crop 图像宽度
	ImgH	Crop 图像高度
	LaneRate	Lane 的速率
	ValidW	SENSOR 的实际有效宽度
SLVS LANE INFO	Devno	SLVS 设备号
	LaneCnt	Lane 数目
	LaneID	Lane ID
SLVS PHY DATA INFO	PhyId	PHY ID
	LaneID	Lane ID
	PhyData	PHY 对应的 LANE 接收到的实时数据

参数		描述
	AlignedData	PHY 对应的 LANE 检测到同步码后接收到的实时数据
SLVS DETECT INFO	Devno	SLVS 设备号
	VC	Virtual Channel
	width	SLVS 控制器检测到的图像总宽度
	height	SLVS 控制器检测到的图像总高度
SLVS DEV ERROR INFO	Devno	SLVS 设备号
	HeaderCRC	数据头 CRC 错误统计
	PayloadCRC	数据 CRC 错误统计
	EccErr	ECC 错误统计
	DataFifoWrite	数据 FIFO 写错误统计
	DataFifoRead	数据 FIFO 读错误统计
	CmdFifoFull	命令 FIFO 满统计
	SkewErr	SKEW 错误统计
SLVS PHY ERROR INFO	PhyIdx	PHY ID
	LaneIdx	Lane ID
	AFifoAlign	FIFO 对齐错误统计
	CodeErr	10B8B 编码错误统计
	DispErr	DISPARITY 错误统计

Hi3516CV500

【调试信息】

```
Module: [MIPI_RX], Build Time[Oct  8 2018, 09:27:31]
-----MIPI LANE DIVIDE MODE-------------------------------------------------
  MODE        LANE DIVIDE
    0                4


-----MIPI DEV ATTR---------------------------------------------------------
   Devno  WorkMode  DataRate          DataType   WDRMode    ImgX    ImgY
ImgW    ImgH
      0     MIPI       X1             RAW12      None       0     204    2592
1944
```



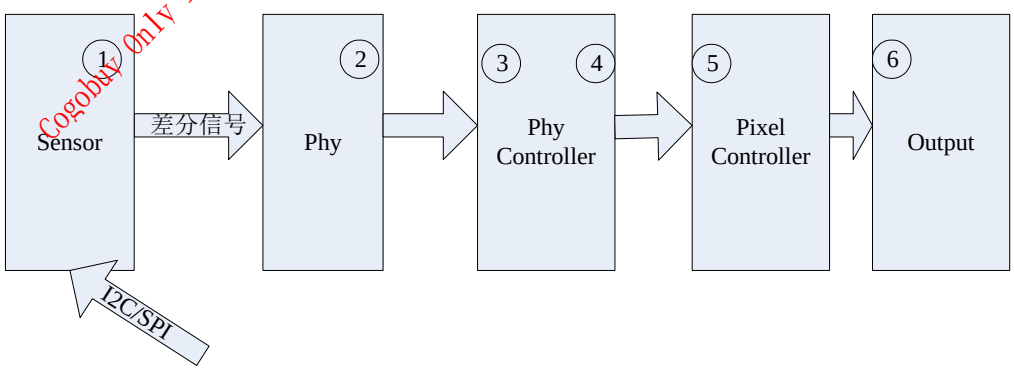
```
-----MIPI LANE INFO-----------------------------------------------------
   Devno               LaneID
      0        0,  1,  2,  3

-----MIPI PHY DATA INFO-------------------------------------------------
   PhyId        LaneId          PhyData             MipiData
LvdsData
      0      0, 1, 2, 3    0x00,0x00,0x00,0x49    0x36,0x2b,0x34,0x45
0x9b,0x1ad275,0x00,0x00

-----MIPI DETECT INFO---------------------------------------------------
 Devno VC   width  height
    0  0    2592    1760
    0  1       0       0
    0  2       0       0
    0  3       0       0
-----LVDS DETECT INFO---------------------------------------------------
 Devno VC   width  height
    0  0    2592    1760
    0  1       0       0
    0  2       0       0
    0  3       0       0
-----LVDS LANE DETECT INFO----------------------------------------------
 Devno  Lane   width  height
    0     0     648    1759
    0     1     648    1759
    0     2     648    1759
    0     3     648    1759
-----PHY CIL ERR INT INFO----------------------------------------------
   PhyId  Clk2TmOut  ClkTmOut  Lane0TmOut  Lane1TmOut  Lane2TmOut
Lane3TmOut  Clk2Esc  ClkEsc  Lane0Esc  Lane1Esc  Lane2Esc  Lane3Esc
      0         0        0          0          0          0          0       0
0         0        0        0        0
-----MIPI ERROR INT INFO 1---------------------------------------------
   Devno  Ecc2  Vc0CRC  Vc1CRC  Vc2CRC  Vc3CRC  Vc0EccCorrct  Vc1EccCorrct
Vc2EccCorrct  Vc3EccCorrct
      0     0      0       0       0       0           0           0
0           0

-----MIPI ERROR INT INFO 2---------------------------------------------
   Devno  Vc0Dt  Vc1Dt  Vc2Dt  Vc3Dt  Vc0FrmCrc  Vc1FrmCrc  Vc2FrmCrc
Vc3FrmCrc
      0     0      0      0      0         0          0          0         0
```

```
-----MIPI ERROR INT INFO 3-----------------------------------------------
Devno  Vc0FrmSeq  Vc1FrmSeq  Vc2FrmSeq  Vc3FrmSeq  Vc0BndryMt  Vc1BndryMt
Vc2BndryMt  Vc3BndryMt
      0         0         0         0         0          0          0
0          0

-----MIPI ERROR INT INFO 4-----------------------------------------------
   Devno  DataFifoRdErr  CmdFifoRdErr  CmdFifoWrErr  DataFifoWrErr
      0            0           0           0            0
-----LVDS ERROR INT INFO -------------------------------------------------
   Devno  CmdRdErr  CmdWrErr  PopErr  StatErr  Link0WrErr  Link0RdErr
      0       0        0      0       0          0          0

-----ALIGN ERROR INT INFO-------------------------------------
   Devno  FIFO_FullErr  Lane0Err  Lane1Err  Lane2Err  Lane3Err
      0           0        0         0         0         0
```

【调试信息分析】

- MIPI_Rx 通过 Phy 接收 sensor 的差分数据，Phy Controller 检测到同步头后，将每条 lane 上的数据对齐；
- Pixel Controller 解析同步信息并按照 raw data 的位宽将 lane 上面的数据合并为 Pixel 数据；Output 模式将 Pixel 数据发送给后级模块。
- Phy PhyController PixelController 由 sensor 的 pixel clk 提供时钟，output 模块的时钟为称为随路时钟，与后级模块的工作时钟相同。MIPI_Rx 的 crop 功能在 Pixel Controller 的末端实现，所以 Crop 后可以降低需要的随路时钟。

图1-5 MIPI 数据流



【参数说明】

参数		描述
MIPI	MODE	MIPI Rx 的 lane 分布模式

参数		描述
LANE DIVIDE MODE	LANE DIVIDE	MIPI_Rx 详细的 LANE 分布。
MIPI DEV ATTR	Devno	MIPI 设备号
	WorkMode	MIPI 设备工作模式：LVDS/MIPI/CMOS 等模式
	DataRate	MIPI Rx 的速率。
	DataType	数据类型：RAW8 / RAW10/ RAW12/ RAW14/ RAW16 bit 等类型
	WDRMode	WDR 模式： • None：非 WDR 模式 • 2To1：2 合 1 WDR • 3To1：3 合 1 WDR • 4To1：4 合 1WDR • DOL2To1：DOL2 合 1WDR • DOL3To1：DOL3 合 1WDR • DOL4To1：DOL4 合 1WDR
	ImgX	Crop 图像起始 X
	ImgY	Crop 图像起始 Y
	ImgW	Crop 图像宽度
	ImgH	Crop 图像高度
MIPI LANE INFO	Devno	MIPI 设备号
	LaneID	Lane ID
MIPI PHY DATA INFO	PhyId	PHY ID 序号
	LaneId	对应的 Lane Id
	PhyData	PHY 对应的 4 条 Lane 接收到的实时数据
	MipiData	PHY 对应的 4 条 Lane 检测到 MIPI 帧同步信号后的实时数据
	LvdsData	PHY 对应的 4 条 Lane 检测到 LVDS 帧同步信号后的实时数据
MIPI DETECT INFO (仅 MIPI 模	Devno	MIPI_Rx 设备号
	VC	Virtual Channel
	width	MIPI 控制器检测到的图像总宽度

参数		描述
式下可见)	height	MIPI 控制器检测到的图像总高度
LVDS DETECT INFO (仅 LVDS 模式下可见)	Devno	MIPI_Rx 设备号
	VC	Virtual Channel
	width	MIPI 控制器检测到的图像总宽度
	height	MIPI 控制器检测到的图像总高度
LVDS LANE DETECT INFO (仅 LVDS 模式下可见)	Devno	MIPI_Rx 设备号
	Lane	Lane ID
	width	该 lane 上检测到的宽度。
	height	该 lane 上检测到的高度。
PHY CIL ERR INT INFO	PhyId	PHY ID
	Clk2TmOut	Clock2 Lane 从 LP 切换到 HS 超时
	ClkTmOut	Clock1 Lane 从 LP 切换到 HS 超时
	Lane0TmOut	DataLane 0 从 LP 切换到 HS 超时
	Lane1TmOut	Data Lane 1 从 LP 切换到 HS 超时
	Lane2TmOut	Data Lane 2 从 LP 切换到 HS 超时
	Lane3TmOut	Data Lane 3 从 LP 切换到 HS 超时
	Clk2Esc	Clock2 Lane 切换到 escape 模式超时
	ClkEsc	Clock Lane 切换到 escape 模式超时
	Lane0Esc	Data Lane 0 切换到 escape 模式超时
	Lane1Esc	Data Lane 1 切换到 escape 模式超时
	Lane2Esc	Data Lane 2 切换到 escape 模式超时
	Lane3Esc	Data Lane 3 切换到 escape 模式超时
MIPI ERROR INT INFO 1 (仅 MIPI 模式下可见)	Devno	MIPI_Rx 设备号。
	Ecc2	Header 至少 2 个错误，ECC 无法纠错
	Vc0CRC	VC0 通道数据的 CRC 错误计数。
	Vc1CRC	VC1 通道数据的 CRC 错误计数。
	Vc2CRC	VC2 通道数据的 CRC 错误计数。
	Vc3CRC	VC3 通道数据的 CRC 错误计数。
	Vc0EccCorrct	VC0 通道 Header 的 ECC 已纠正的错误计数。

参数		描述
	Vc1EccCorrct	VC1 通道 Header 的 ECC 已纠正的错误计数。
	Vc2EccCorrct	VC2 通道 Header 的 ECC 已纠正的错误计数。
	Vc3EccCorrct	VC3 通道 Header 的 ECC 已纠正的错误计数。
MIPI ERROR INT INFO 2 (仅 MIPI 模式下可见)	Devno	MIPI_Rx 设备号。
	Vc0Dt	VC0 通道不支持的数据类型计数
	Vc1Dt	VC1 通道不支持的数据类型计数
	Vc2Dt	VC2 通道不支持的数据类型计数
	Vc3Dt	VC3 通道不支持的数据类型计数
	Vc0FrmCrc	VC0 通道帧数据错误计数
	Vc1FrmCrc	VC1 通道帧数据错误计数
	Vc2FrmCrc	VC2 通道帧数据错误计数
	Vc3FrmCrc	VC3 通道帧数据错误计数
MIPI ERROR INT INFO 3 (仅 MIPI 模式下可见)	Devno	MIPI_Rx 设备号。
	Vc0FrmSeq	VC0 的帧序错误计数
	Vc1FrmSeq	VC1 的帧序错误计数
	Vc2FrmSeq	VC2 的帧序错误计数
	Vc3FrmSeq	VC3 的帧序错误计数
	Vc0BndryMt	VC0 通道的帧起始与帧结束短包不匹配计数
	Vc1BndryMt	VC1 通道的帧起始与帧结束短包不匹配计数
	Vc2BndryMt	VC2 通道的帧起始与帧结束短包不匹配计数
	Vc3BndryMt	VC3 通道的帧起始与帧结束短包不匹配计数
MIPI ERROR INT INFO 4 (仅 MIPI 模式下可见)	Devno	MIPI_Rx 设备号
	DataFifoRdErr	MIPI CTRL 读数据 FIFO 中断计数
	CmdFifoRdErr	MIPI CTRL 读命令 FIFO 中断计数
	CmdFifoWrErr	MIPI CTRL 写命令 FIFO 中断计数
	DataFifoWrErr	MIPI CTRL 写数据 FIFO 中断计数
LVDS ERROR INT INFO (仅 LVDS	Devno	MIPI 设备号
	CmdRdErr	lvds CMD_FIFO 读错误中断计数
	CmdWrErr	lvds CMD_FIFO 写错误中断计数

参数		描述
模式下可见)	PopErr	lvds 读取 line_buf 错误中断计数
	StatErr	lvds 各 lane 同步错误中断计数
	Link0WrErr	link0 读 FIFO 错误中断计数
	Link0RdErr	link0 写 FIFO 错误中断计数
ALING ERROR INFO	Devno	MIPI 设备号
	FIFO_FullErr	FIFO 溢出
	Lane0Err	Lane0 FIFO 溢出
	Lane1Err	Lane1 FIFO 溢出
	Lane2Err	Lane2 FIFO 溢出
	Lane3Err	Lane3 FIFO 溢出

Hi3516EV200

【调试信息】

```
Module: [MIPI_RX], Build Time[Nov  24 2018, 14:50:01]
-----MIPI LANE DIVIDE MODE----------------------------------------------
  MODE        LANE DIVIDE
    0

-----MIPI DEV ATTR------------------------------------------------------
   Devno  WorkMode  DataRate          DataType   WDRMode    ImgX    ImgY
ImgW    ImgH
      0     MIPI       X1             RAW12      None       0     204    2304
1296

-----MIPI LANE INFO-----------------------------------------------------
   Devno               LaneID
      0                0,  1,

-----MIPI PHY DATA INFO-------------------------------------------------
   PhyId        LaneId          PhyData             MipiData
LvdsData
      0       0, 1   0x00,0x00    0x36,0x2b   0x9b,0x1ad275
-----MIPI DETECT INFO--------------------------------------------------
 Devno VC   width  height
    0  0    2304    1296
```

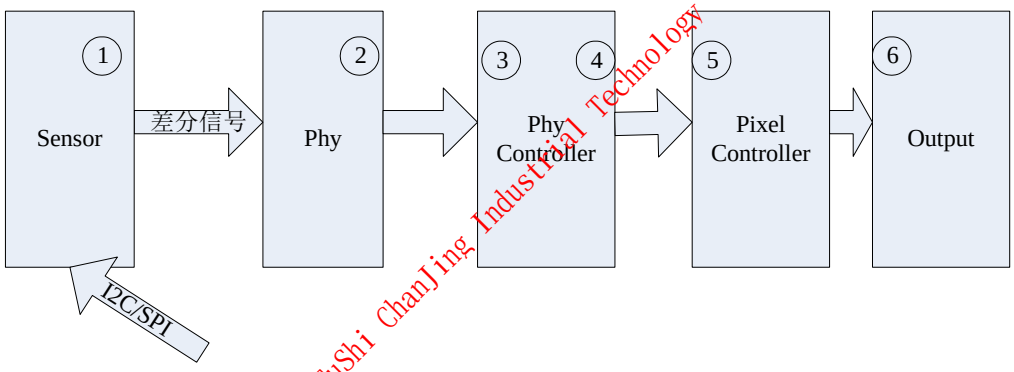
```
    0  1      0      0
-----LVDS DETECT INFO----------------------------------------------------
 Devno VC   width  height
    0  0    2304   1296
    0  1      0      0
-----LVDS LANE DETECT INFO-----------------------------------------------
-----
 Devno  Lane   width  height
    0    0    0      0
    0    1    0      0
-----PHY CIL ERR INT INFO------------------------------------------------
   PhyId  Clk2TmOut  ClkTmOut  Lane0TmOut  Lane1TmOut  Lane2TmOut
Lane3TmOut  Clk2Esc  ClkEsc  Lane0Esc  Lane1Esc  Lane2Esc  Lane3Esc
      0         0        0          0          0          0          0       0
0        0        0        0        0
-----MIPI ERROR INT INFO 1-----------------------------------------------
------------
   Devno  Ecc2  Vc0CRC  Vc1CRC  Vc2CRC  Vc3CRC  Vc0EccCorrct  Vc1EccCorrct
Vc2EccCorrct  Vc3EccCorrct
      0    0      0      0      0      0           0           0
0           0

-----MIPI ERROR INT INFO 2-----------------------------------------------
   Devno  Vc0Dt  Vc1Dt  Vc2Dt  Vc3Dt  Vc0FrmCrc  Vc1FrmCrc  Vc2FrmCrc
Vc3FrmCrc
      0     0      0     0     0         0         0         0         0

-----MIPI ERROR INT INFO 3-----------------------------------------------
   Devno  Vc0FrmSeq  Vc1FrmSeq  Vc2FrmSeq  Vc3FrmSeq  Vc0BndryMt
Vc1BndryMt  Vc2BndryMt  Vc3BndryMt
      0          0         0         0         0          0          0
0         0

-----MIPI ERROR INT INFO 4-----------------------------------------------
   Devno  DataFifoRdErr  CmdFifoRdErr  Vsync  CmdFifoWrErr  DataFifoWrErr
      0           0           0     0           0           0
-----LVDS ERROR INT INFO 1-----------------------------------------------
   Devno  Vsync  CmdRdErr  CmdWrErr  PopErr  StatErr
      0     0        0        0      0       0

-----LVDS ERROR INT INFO 2-----------------------------------------------
   Devno  Link0WrErr  Link1WrErr  Link2WrErr  Link0RdErr  Link1RdErr
Link2RdErr
      0          0          0          0          0          0          0
```

```
-----ALIGN ERROR INT INFO-------------------------------------
   Devno  FIFO_FullErr  Lane0Err  Lane1Err  Lane2Err  Lane3Err
      0           0        0         0         0         0
```

【调试信息分析】

- MIPI_Rx 通过 Phy 接收 sensor 的差分数据，Phy Controller 检测到同步头后，将每条 lane 上的数据对齐；
- Pixel Controller 解析同步信息并按照 raw data 的位宽将 lane 上面的数据合并为 Pixel 数据；Output 模式将 Pixel 数据发送给后级模块。
- Phy PhyController PixelController 由 sensor 的 pixel clk 提供时钟，output 模块的时钟为称为随路时钟，与后级模块的工作时钟相同。MIPI_Rx 的 crop 功能在 Pixel Controller 的末端实现，所以 Crop 后可以降低需要的随路时钟。

图1-6 MIPI 数据流



【参数说明】

参数		描述
MIPI LANE DIVIDE MODE	MODE	MIPI Rx 的 lane 分布模式
	LANE DIVIDE	MIPI_Rx 详细的 LANE 分布。
MIPI DEV ATTR	Devno	MIPI 设备号
	WorkMode	MIPI 设备工作模式：LVDS/MIPI/CMOS 等模式
	DataRate	MIPI Rx 的速率。
	DataType	数据类型：RAW8 / RAW10/ RAW12/ RAW14/ RAW16 bit 等类型

参数		描述
	WDRMode	WDR 模式： • None：非 WDR 模式 • 2To1：2 合 1 WDR • DOL2To1：DOL2 合 1WDR
	ImgX	Crop 图像起始 X
	ImgY	Crop 图像起始 Y
	ImgW	Crop 图像宽度
	ImgH	Crop 图像高度
MIPI LANE INFO	Devno	MIPI 设备号
	LaneID	Lane ID
MIPI PHY DATA INFO	PhyId	PHY ID 序号
	LaneId	对应的 Lane Id
	PhyData	PHY 对应的 2 条 Lane 接收到的实时数据
	MipiData	PHY 对应的 2 条 Lane 检测到 MIPI 帧同步信号后的实时数据
	LvdsData	PHY 对应的 2 条 Lane 检测到 LVDS 帧同步信号后的实时数据
MIPI DETECT INFO (仅 MIPI 模式下可见)	Devno	MIPI_Rx 设备号
	VC	Virtual Channel
	width	MIPI 控制器检测到的图像总宽度
	height	MIPI 控制器检测到的图像总高度
LVDS DETECT INFO (仅 LVDS 模式下可见)	Devno	MIPI_Rx 设备号
	VC	Virtual Channel
	width	MIPI 控制器检测到的图像总宽度
	height	MIPI 控制器检测到的图像总高度
LVDS LANE DETECT INFO (仅 LVDS 模式下可见)	Devno	MIPI_Rx 设备号
	Lane	Lane ID
	width	该 lane 上检测到的宽度。
	height	该 lane 上检测到的高度。
PHY CIL	PhyId	PHY ID

参数		描述
ERR INT INFO	Clk2TmOut	Clock2 Lane 从 LP 切换到 HS 超时
	ClkTmOut	Clock1 Lane 从 LP 切换到 HS 超时
	Lane0TmOut	DataLane 0 从 LP 切换到 HS 超时
	Lane1TmOut	Data Lane 1 从 LP 切换到 HS 超时
	Clk2Esc	Clock2 Lane 切换到 escape 模式超时
	ClkEsc	Clock Lane 切换到 escape 模式超时
	Lane0Esc	Data Lane 0 切换到 escape 模式超时
	Lane1Esc	Data Lane 1 切换到 escape 模式超时
MIPI ERROR INT INFO 1 (仅 MIPI 模式下可见)	Devno	MIPI_Rx 设备号。
	Ecc2	Header 至少 2 个错误，ECC 无法纠错
	Vc0CRC	VC0 通道数据的 CRC 错误计数。
	Vc1CRC	VC1 通道数据的 CRC 错误计数。
	Vc0EccCorrct	VC0 通道 Header 的 ECC 已纠正的错误计数。
	Vc1EccCorrct	VC1 通道 Header 的 ECC 已纠正的错误计数。
MIPI ERROR INT INFO 2 (仅 MIPI 模式下可见)	Devno	MIPI_Rx 设备号。
	Vc0Dt	VC0 通道不支持的数据类型计数
	Vc1Dt	VC1 通道不支持的数据类型计数
	Vc0FrmCrc	VC0 通道帧数据错误计数
	Vc1FrmCrc	VC1 通道帧数据错误计数
MIPI ERROR INT INFO 3 (仅 MIPI 模式下可见)	Devno	MIPI_Rx 设备号。
	Vc0FrmSeq	VC0 的帧序错误计数
	Vc1FrmSeq	VC1 的帧序错误计数
	Vc0BndryMt	VC0 通道的帧起始与帧结束短包不匹配计数
	Vc1BndryMt	VC1 通道的帧起始与帧结束短包不匹配计数
MIPI ERROR INT INFO 4 (仅 MIPI 模式下可见)	Devno	MIPI_Rx 设备号。
	DataFifoRdErr	MIPI CTRL 读数据 FIFO 中断计数
	CmdFifoRdErr	MIPI CTRL 读命令 FIFO 中断计数
	Vsync	MIPI CTR vsync 中断计数
	CmdFifoWrErr	MIPI CTRL 写命令 FIFO 中断计数

参数		描述
	DataFifoWrErr	MIPI CTRL 写数据 FIFO 中断计数
LVDS ERROR INT INFO 1(仅 LVDS 模式下可见)	Devno	MIPI 设备号
	Vsync	lvds vsync 中断计数
	CmdRdErr	lvds CMD_FIFO 读错误中断计数
	CmdWrErr	lvds CMD_FIFO 写错误中断计数
	PopErr	lvds 读取 line_buf 错误中断计数
	StatErr	lvds 各 lane 同步错误中断计数
LVDS ERROR INT INFO 2(仅 LVDS 模式下可见)	Devno	MIPI 设备号
	Link0WrErr	link0 读 FIFO 错误中断计数
	Link1WrErr	link1 读 FIFO 错误中断计数
	Link0RdErr	link0 写 FIFO 错误中断计数
	Link1RdErr	link1 写 FIFO 错误中断计数
ALING ERROR INFO	Devno	MIPI 设备号
	FIFO_FullErr	FIFO 溢出
	Lane0Err	Lane0 FIFO 溢出
	Lane1Err	Lane1 FIFO 溢出

1.7.2 MIPI_TX Proc 信息

注意

MIPI_TX Proc 信息适用于 Hi3559AV100、Hi3519AV100 和 Hi3516CV500。

MIPI_Tx 的 proc 信息主要有：MIPI_Tx 设备配置信息、MIPI_Tx 时序配置信息、MIPI_Tx 设备状态信息。

【调试信息】

```
Module: [MIPI_TX], Build Time[Jun 21 2018, 15:23:45]
----------MIPI_Tx DEV CONFIG----------------------------
   devno   lane0   lane1   lane2   lane3    output_mode  phy_data_rate
pixel_clk(KHz)     video_mode     output_fmt
0       0       1       2       3            1           945         148500
0         2
```



```
----------MIPI_Tx SYNC CONFIG--------------------------
     pkt_size    hsa_pixels    hbp_pixels  hline_pixels     vsa_lines
vbp_lines     vfp_lines  active_lines edpi_cmd_size
         1080           8          20         1238          10          26
16         1920           0

----------MIPI_Tx DEV STATUS---------------------------
width  height HoriAll VertAll hbp hsa vsa
1080    1920    1237    1972    20   8   10
```

【调试信息分析】

MIPI_Tx 设备配置信息、MIPI_Tx 时序配置信息等为设备启动前通过接口配置的信息；MIPI_Tx 设备状态信息是设备运行时检测到的部分时序信息：有效宽高、水平总宽度、垂直总高度。

【参数说明】

参数		描述
MIPI_Tx DEV CONFIG	devno	设备号。
	lane0 lane1 lane2 lane3	>=0：lane 号码。 -1：禁用。
	output_mode	0: csi mode 1: dsi video mode 2: dsi command mode
	phy_data_rate	Phy 数据率。
	pixel_clk(KHz)	像素时钟(单位 KHz)。
	video_mode	0: BURST_MODE 1: NON_BURST_MODE_SYNC_PULSES 2: NON_BURST_MODE_SYNC_EVENTS
	output_fmt	0:OUT_FORMAT_RGB_16_BIT 1:OUT_FORMAT_RGB_18_BIT 2:OUT_FORMAT_RGB_24_BIT 3:OUT_FORMAT_YUV420_8_BIT_NORMAL 4:OUT_FORMAT_YUV420_8_BIT_LEGACY 5:OUT_FORMAT_YUV422_8_BIT
MIPI_Tx SYNC	pkt_size	hact
	hsa_pixels	hsync

参数		描述
CONFIG	hbp_pixels	hbp
	hline_pixels	hact+hsa+hbp+hfp
	vsa_lines	vsa
	vbp_lines	vbp
	vfp_lines	vfp
	active_lines	vact
	edpi_cmd_size	edpi cmd size
MIPI_Tx DEV STATUS	width	检测到的宽。
	height	检测到的高。
	HoriAll	水平总宽。
	VertAll	垂直总高。
	hbp	水平前肩。
	hsa	水平同步脉冲个数。
	vsa	垂直同步脉冲个数。

1.8 FAQ

MIPI 具体规格请参考芯片手册《Hi35xxVxxx ultra-HD Mobile Camera SoC 用户指南》和《Features of the Video Interfaces of HiSilicon IP Cameras.pdf》

1.8.1 Lane id 如何配置

Lane id 的配置对应 mipi_dev_attr_t 中的 short lane_id[MIPI_LANE_NUM]或者 slvs_dev_attr_t 中的 short lane_id[SLVS_LANE_NUM]，其中 lane_id 数组的索引号表示的是 SENSOR 的 LANE ID，lane_id 数组的值表示的是 MIPI 的 LANE ID。

对接 sensor 时，未使用的 lane 将其对应的 lane_id 配置为-1。配置 lane_id 还可以调整数据通道顺序，**根据硬件单板与实际 sensor 输出通道的对应关系调整 lane_id 的配置。**

下面举例进行说明，例如 MIPI 与 SENSOR 的引脚硬件连接如表 1-7 所示。

表1-7 SENSOR 与 MIPI_Rx 管脚关系

MIPI Lane 管脚	SENSOR 管脚
MIPI_RX1_D0	Lane 0

MIPI Lane 管脚	SENSOR 管脚
MIPI_RX1_D1	Lane 1
MIPI_RX1_D2	Lane 2
MIPI_RX1_D3	Lane 3

MIPI 的最大 Lane 数为 8，我们认为 SENSOR 的 Lane 数目最多 8 个，由于 sensor 实际只有 4 个 Lane，只输出数据到 MIPI 的 4 个 Lane，需要将 SENSOR 未连接的或者不存在的 Lane 的 lane_id 配置为-1，所以所以 lane_id 配置如下：

```
// 索引   sensor_lane0      sensor_lane1      sensor_lane2      sensor_lane3 ……
//
lane_id ={ MIPI_RX1_D0 ,  MIPI_RX1_D1 , MIPI_RX1_D2 ,  MIPI_RX1_D3,-1,-1,-1,-1}
```

1.8.2 MIPI 频率说明

MIPI Lane 频率与 VI 频率关系

使用 MIPI 多个 Lanes 进行数据传输，MIPI Lane 的传输频率与 VI 处理频率如何对应，每一 Lane 可传输的最高速率如何计算？

- MIPI_Rx 使用多 Lane 接收数据，会转成内部时序，送给 VI 模块进行处理，多 Lane 传输的数据总量不变，有这样的计算公式：

  VI_Freq * Pix_Width = Lane_Num * MIPI_Freq
- 其中，VI_Freq 为 VI 的工作时钟， Pix_Width 为像素位宽，Lane_Num 为传输 lane 个数，MIPI_Freq 为一个 lane 能接收的最大频率。
- 下面以 VI 工作频率为 250M，MIPI 数据为 RAW 12, 4Lane 传输为例进行说明：

  MIPI_Freq = (250 * 12) / 4 = 750

  即每个 Lane 最高频率为 750MHz