

Congothy only For sheathen From charting Industrial Technology Co. Ltd. MIPI 使用指南

文档版本 05

2019-09-12 发布日期

版权所有 © 上海海思技术有限公司 2019。保留一切权利。

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

商标声明

(A) THISILICON 、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。除非另有领导,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。
Rednord

Technolog

Te 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

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 cthole
Hi3559C	V100
Hi3519A	V100 WEIT
Hi3556A	产品版本 V100ES V100 V100 V100 V100 V100 V100 V100 V10
Hi3516D	V300c.)
Hi3516C	
Hi3516A Hi3559 Hi3556 Hi3556	V300
Hi3559	V200
Hi3556 60	V200
Hi3516E 0114	V300
Hi35 BOD	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	一列标志,它们所代表的含义如天 (2005) 说明 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
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 态 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分份害。
注意 kol si	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。
	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。

Charling Industrial Technology 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 Hi3516DV206 芯片支持

1.4 小节,新增 HI_MIPI_TX_GET_CMD。

1.5 小节分新增 get_cmd_info_t、slvs_err_check_mode_t、mipi_clk_mode_t、 lvds_tare_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)

aning Industrial Technology co.、Ltd.
相主 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/Hi3516DV.360 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**说**3519AV100 MIPI Rx proc 信息。

文档版本 00B08 (2018-07-06)

第8%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_MPI_DISABLE_SENSOR_CLOCK 【定义】涉及更新。HI_MIPI_RESET_SENSOR 至何_MIPI_UNRESET_SENSOR【定义】和【叁数】涉及修改

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

combo_dev_t, sns_rst_source_t, \$\pi\$ sns_clk_source_t 涉及修改

1.6 小节,【调试信息】【参数说明】涉及更新。

SNS_RESET_DEV 和 SNS_CLOCK_DEV

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 (2007-06-30)

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、

文档版本 00B01 (2017-05-28)

第1次临时版本发布。



前 L N			
	1.2 重要概念	7,9.	1
	1.3 功能描述	<i>.</i> 60.	3
	1.4 API 参考	1007	
	1.5 数据类型	Then in the control of the control o	27
	1.6 MIPI Tx 模块参数		71
	1.7 Proc 信息	ust'	71
	1.7.1 MIPI_RX Proc 信息	, Ind	71
	1.7.2 MIPI_TX Proc 信息	aling.	101
	1.8 FAQ		103
	1.8.1 Lane id 如何配置		103
	1.8.2 MIPI 频率说明	, and the second	104
	and the second		
	60,3		
	, Kin		
	ro Kon		
	codop		



插图目录

图 1-1 SOF/EOF/SOL/EOL 同步方式	2
图 1-2 SAV/EAV 同步方式	3
图 1-3 MIPI 数据流	75
	84
图 1-5 MIPI 数据流	92
图 1-6 MIPI 数据流	9298

cogday only for sheathen fushi chanting industrial T



表格目录

	3
	4
<i>*9</i> ,	4
\sim	
	6
etho!	51
	103
	Lexicology Co., Tray

cogodiny only for sheathen fushi chanling Industr



MIPI 使用指南

注意

- Hi3516EV300、Hi3518EV300、Hi3516DV200 与、163516EV200 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 通过低电压差分信贷接收原始视频数据,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。



MIPI 使用指南 1 MIPI 使用指南

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**征**的起始和结束。同步方式如图 1-1 所示。

图I-1 SOF/EOF/SOL/EOL 同步万式							
V.BLK H.BLK SOF Effective Pixel H.BLK H.BLK Effective Pixel H.BLK H.BLK Effective Pixel H.BLK							
H.BLK	SOF	Effective Pixel	Kilo	H.BLK			
H.BLK		Effective Pixel		H.BLK			
H.BLK		Effective Pixel		H.BLK			
H.BLK		Effective Pixel		H.BLK			
!		: Cha		:			
H.BLK		Effective Pixel	EOL	H.BLK			
H.BLK	SOL	Effective Pixel		H.BLK			
H.BLK		Effective Pixel		H.BLK			
H.BLK	d	Effective Pixel		H.BLK			
H.BLK	, ¢ot	Effective Pixel		H.BLK			
H.BLK	14	Effective Pixel		H.BLK			
H.BLK ₄		Effective Pixel	EOF	H.BLK			
Codopy		V.BLK					

图1-1 SOF/FOF/SOL/FOL 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。

1 MIPI 使用指南

图1-2 SAV/EAV 同步方式

H.BLK	SAV	V.BLK	EAV	H.BLK
H.BLK	(Invalid	V.BLK	(Invalid	H.BLK
H.BLK	line)	V.BLK	line)	H.BLK
H.BLK		H.OB / effective pixel		H.BLK
H.BLK		H.OB / effective pixel		H.BLK
H.BLK		H.OB / effective pixel		H.BLK
:		!		'
H.BLK	SAV	H.OB / effective pixel	EAV	H.BLK
H.BLK	(Valid line)	H.OB / effective pixel	(Valid	H.BLK
H.BLK		H.OB / effective pixel	line)	H.BLK
H.BLK		H.OB / effective pixel		H.BLK
H.BLK		H.OB / effective pixel		H.BLK
H.BLK		H.OB / effective pixel		H.BLK
H.BLK		V.BLK		H.BLK
	SAV	, DLK	E 43.7	170.
H.BLK	(Invalid	v.BLK	EAV (Invalid	H.BLK
H.BLK	line)	V.BLK V.BLK	line)	H.BLK
H.BLK	,	V.BLK	1003	H.BLK

DOL

DOL 的全称是 Digital Overlap,指 SQL 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/Pi/DC 接口的数据,通过不同的功能模式配置,MIPI Rx 可以 支持多种速度和分款率的数据传输需求,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 使用指南 1 MIPI 使用指南

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	1
Hi3516EV200	
Hi3516EV300	
Hi3518EV300	٠6.
Hi3516DV300	2
Hi3559V200	70.
Hi3556V200	700g
Hi3516AV300	2 Technology Co.,

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 . ox

				100					
芯片类型	Mode	DEV0	DEVK	DEV2	DEV3	DEV4	DEV5	DEV6	DEV7
Hi3559AV100ES	0	L0~L15	Miles	N	N	N	N	N	N
	1	L0~L11	N	N	N	L12~L15	N	N	N
	2	L0-111	N	N	N	L12 L14	L13 L15	N	N
	3 0000	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~L 7	L8~L11	N	L12~L15	N	N	N
	8	L0~L3	L4~L 7	L8~L11	N	L12 L14	L13 L15	N	N
	9	L0~L3	L4~L 7	L8 L10	L9 L11	L12 L14	L13 L15	N	N



1 MIPI 使用指南

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 , 6.	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 L8~L11 L8~L10	5Z	L12 L14	L13 L15
	9	L0~L3	N	L4~L7	N	L88£10	L9 L11	L12 L14	L13 L15
	A	L0~L3	N	L4 L6	L5 JET	L8 L10	L9 L11	L12 L14	L13 L15
	В	L0 L2	L1 L3	L4 L6Jin	L5 L7	L8 L10	L9 L11	L12 L14	L13 L15
Hi3519AV100	0	L0~L11	N	3 N	N	N	N	N	N
	1	L0~L7	Menthen	N	L8~L 11	N	N	N	N
	2	L0~L76	N	N	L8 L10	L9 L11	N	N	N
	3 3000	L0~L3	L4~L 11	N	N	N	N	N	N
	4	L0~L3	L4~L 7	N	L8~L 11	N	N	N	N
	5	L0~L3	L4~L 7	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 复用类系图

芯片类 型	LANE	MIPI0	MIPI1	MIPI2	MIPI3	MIPI4	MIPI5	MIPI6	MIPI7	SL VS0	SLV S1	SL VS2	SLV S3
Hi3559A	Lane0	√	-	-	FUSH	-	-	-	1	√	√	-	-
V100ES	Lane1	√	-	- entre	-	-	-	-	-	√	√	-	-
	Lane2	√	-	- Stenzh	-	-	-	-	-	√	√	-	-
	Lane3	√	20/2	-	-	-	-	-	-	√	√	-	-
	Lane4	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	γ	-	-	-	-	-	-	√	√	-	-
	Lane5		√	-	ı	ı	ı	ı	ı	√	√	ı	-
	Lane6	√	√	ı	ı	ı	ı	ı	ı	√	√	ı	-
	Lane7	√	√	-	ı	ı	ı	ı	i	√	√	ı	-
	Lane8	√	ı	√			ı	ı	ı	ı	-	√	√
	Lane9	√	-	√	√		-	-	1	-	-	√	√
	Lane10	√	-	√			-	-	1	-	-	√	√
	Lane11	√	1	√	√		1	1	1	ı	-	√	√
	Lane12	√	1	-	1	√		1	ı	1	-	√	√
	Lane13	√	-	-	-	√	√	-	-	-	-	√	√



MIPI 使用指南 1 MIPI 使用指南

芯片类 型	LANE	MIPI0	MIPI1	MIPI2	MIPI3	MIPI4	MIPI5	MIPI6	MIPI7	SL VS0	SLV S1	SL VS2	SLV S3
	Lane14	√	-	-	-	√		-	-	-	-	√	√
	Lane15	√	-	-	-	√	√	-	-	-	-	√	√
Hi3559 AV100	Lane0	√								√	√	-	-
A V 100	Lane1	√	√							√	√	-	-
	Lane2	√								√	√	-	-
	Lane3	√	√							√	√	-	-
	Lane4	√		√						√	√	-	-
	Lane5	√		√	√					, Ø.	√	-	-
	Lane6	√		√				Technol	, ₀ ,	√	√	-	-
	Lane7	√		√	√			100	90	√	√	-	-
	Lane8	√				√		Lect.		-	-	√	√
	Lane9	√				√	ndustria			-	-	√	√
	Lane10	√				√ °	de			-	-	√	√
	Lane11	√				nathling	√			-	-	√	√
	Lane12	√			Si	` √		√		-	-	√	√
	Lane13	√		470	\$ %	√		√	√	-	-	√	√
	Lane14	√		Shent		√		√		-	-	√	√
	Lane15	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	14	5		√		√	√	-	-	√	√
Hi3519 AV100	Lane0	√	y or i							√			
AVIOO	Lane1	4000r								√			
	Lane2	√								√			
	Lane3	√								√			
	Lane4	√	√							√			
	Lane5	√	√	√						√			
	Lane6	√	√							√			
	Lane7	√	√	√						√			
	Lane8	√	√		√								
	Lane9	√	√		√	√							



MIPI 使用指南 1 MIPI 使用指南

芯片类 型	LANE	MIPI0	MIPI1	MIPI2	MIPI3	MIPI4	MIPI5	MIPI6	MIPI7	SL VS0	SLV S1	SL VS2	SLV S3
	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 SLW。
- HI_MIPI_UNRESET_SLVS: 撤销复位 SLVS。
- HI_MIPI_ENABLE_MIPI_COCK: 打开 MIPI 设备的时钟。
- HI_MIPI_DISABLE_MEN_CLOCK: 关闭 MIPI 设备的时钟。
- HI_MIPI_ENABLE SLVS_CLOCK: 打开 SLVS 设备的时钟。
- HI_MIPI_DISANCE_SLVS_CLOCK: 关闭 SLVS 设备的时钟。
- HI_MIPI_ENABLE_SENSOR_CLOCK: 打开 SENSOR 的时钟。
- HI_MIRCDISABLE_SENSOR_CLOCK: 关闭 SENSOR 的时钟。
- HI MIPI_CLEAR: 清除设备相关的配置。

MIRPTx 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_JOC_MAGIC, 0x01, Tial Technology Co. combo dev attr t)

【参数】

combo_dev_attr_t 类型的指针。

【返回值】

返回值,成战	描述
0 iino	成功。
-1 Chard	失败,并设置 errno
【芯片差异】 无。 【需求】 Jony	
【需求】 on Y 头文件 o'hi_mipi.h	

【芯片差异】

【需求】

头文件。whi_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

MIPI 使用指南 1 MIPI 使用指南

- ▶ 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_SIXS
- HI_MIPI_RESET_SENSOR
- HI_MIPI_UNRESET_SENSOR
- HI_MIPI_RESET_MIPI
- HI_MIPI_\(\forall \)\(\text{RESET_MIPI}\)

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 2009				
	支持 支持 支持 本紹介的 文持 建议 本紹介的 表示 表示 表示 表示 表示 表示 表示 表示 表示 表				
【需求】	ctrial				
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					

【需求】

【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 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】

A Fushi Chaning Industrial Technology Co., Ltd. #define HI_MIPI_RECET_SENSOR __IOW(HI_MIPI_IOC_MAGIC, 0x05, sns_rst_source_

【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

1 MIPI 使用指南

Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500	支持
Hi3516EV200	支持

【需求】

头文件: hi_mipi.h

【注意】

HI_MIPI_UNRESET_SENSOR

【注意】	
无。	_Iow(HI_MIPI_IOC_MAGIC, 0x06,
RESET_SENSOR	<i>\oldoy</i> ., <i>\\</i>
【描述】	oology
撤销复位 sensor。	Lectur
【定义】	xial
#define HI_MIPI_UNRESET_SENSOR	IOW(HI_MIPI_IOC_MAGIC, 0x06,
sns_rst_source_t) ind	
【参数】 chand	
sns_rst_source_t SENSOR 复选信号线编号。	
I 返回值 Lot Sensor And The Transport of t	
返回值	描述
0 ONT	成功。
-1 addish	失败,并设置 errno
- 693	

【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】

MIP. CO. _IOW(HI_MIP1\sqrt{IOC_MAGIC, 0x07,} #define HI_MIPI_RESET_MIPI combo_dev_t)

【参数】

combo_dev_t 设备号。

【返回值】

返回值	in ⁰	描述
0	chan	成功。
-1	EUSNI	失败,并设置 errno

【芯片差异】

芯片类型の分	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。

MIPI 使用指南 1 MIPI 使用指南

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 mind	支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500 sperther	支持
Hi3516EV200,0	支持

【需求

头文件: hi_mipi.h

【注意】

无。

HI_MIPI_RESET_SLVS

【描述】

复位 SLVS。

【定义】



【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 z持ednadd
Hi3519AV100	支持echno
Hi3516CV500	水 支持
Hi3516EV200 Indué	不支持
ming	
【需求】	
头文件: hi_mipi.h	
【注意】 charthe	
Hi3516EV200 【需求】 头文件: hi_mipi.h 【注意】 无。 PESET SPLYS	
RESET_SEVS	

【需求】

【注意】

HI_MIPI_UNRESET_SEVS

【描述》

撤销复位 SLVS。

【定义】

【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。 ABLE_MIPI_CLOCK transit chanting and the chanting th	. al Technolos
头文件: hi_mipi.h 【注意】	stri
无。	
ABLE_MIPI_CLOCK	
【描述】	
打开 MIPI 设备的时钟。	
- · · · - · · · · · · · · · · · · · · ·	

【需求】

【注意】

HI_MIPI_ENABLE_MIPI_CLOCK

【描述】

【定义】

#define_HPI_MIPI_ENABLE_MIPI_CLOCK
comboolev_t) _IOW(HI_MIPI_IOC_MAGIC, 0x0c,

【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】

1 MIPI 使用指南

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】

HI_MIPI_DISABLE_MIPI_CLOCK

【需求】	
头文件: hi_mipi.h	*9·
【注意】	20°,
无。	100 CC
DISABLE_MIPI_CLOCK	Tochhi
【描述】	rial .
关闭 MIPI 设备的时钟。	
【定义】 anling	
头文件: hi_mipi.h 【注意】 无。 DISABLE_MIPI_CLOCK 【描述】	_IOW(HI_MIPI_IOC_MAGIC,
【参数】	
combo_dev_t 设备号。	
【返回值】の対	
返回傳	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

1 MIPI 使用指南

Hi3516CV500	支持
Hi3516EV200	支持

【需求】

头文件: hi_mipi.h

【注意】

无。

HI_MIPI_ENABLE_SLVS_CLOCK

【描述】

【定义】

【参数】

【返回值】

ABLE_SLVS_CLOCK	
【描述】	\xd·
打开 SLVS 设备的时钟。	<i>\$</i> 0.,
【定义】	allogy
【描述】 打开 SLVS 设备的时钟。 【定义】 #define HI_MIPI_ENABLE_SLVS_CLOCK 0x0e, combo_dev_t) 【参数】 combo_dev_t 设备号。 【返回值】 返回值 0 -1 constanting	.al
【参数】 Indu ^s	Kr.
combo_dev_t 设备号。	
【返回值】	
返回值	描述
0 renther	成功。
-1 601	失败,并设置 errno

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】

【返回值】

<pre>0x0f, combo_dev_t)</pre>	
【参数】	rg.
combo_dev_t 设备号。	~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
【返回值】	描述 edino logy Co., Ltd.
返回值	
0	成功。
-1 , in	失败,并设置 errno
chan	
-1 【芯片差异】 Then Fushi Chanting Ind	
+++**********************************	日不士柱

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	支持
Hi3559A X 100	支持
Hi3519AV100	支持
Hi3516CV500	不支持
Hi3516EV200	不支持

【需求】

头文件: hi_mipi.h

【注意】

无。

MIPI 使用指南 1 MIPI 使用指南

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。 20.
-1	失败,兼设置 errno

【芯片差异】

芯片类型 ,,,,,,,,,	是否支持
Hi3559AV100ES	支持
Hi3559AV100	支持
Hi3519AV100 ather	支持
Hi3516CV500 Sheet	支持
Hi3516EV200	支持

了。 【宋常】

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】

	【心片差异】	, 48.
	芯片类型	是否支持
	Hi3559AV100ES	支持
	Hi3559AV100	支持。
	Hi3519AV100	文持
	Hi3516CV500 rdus	支持
	Hi3516EV200 min ¹⁰	支持
	chi Chia	
	【需求】	
	头文件: hi_mipi.h_merthe	
	【注意】	
	无。	
HI_MIPI_CLE	Hi3519AV100 Hi3516CV500 Hi3516EV200 【需求】 头文件: hi_mipi.h sperthen tushi 【注意】 tot 无。 only ARcotony 【描述】	
	【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 2009

【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。

#def 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

HI_MIPI_TX_SET_CMD

IOW(HI MIPI TX IOC MAGIC, 0x02,

【参数】

发送给 MIPLOX 设备的命令信息。

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

尤 。	
_GET_CMD	rg.
【描述】	0.,
用于从外围设备读取信息。	9/0g/
【定义】	Lectur
<pre>#define HI_MIPI_TX_GET_CMD get_cmd_info_t)</pre>	IOWR(HI_MIPI_TX_IOC_MAGIC, 0x04
【描述】 用于从外围设备读取信息。 【定义】 #define HI_MIPI_TX_GET_CMD get_cmd_info_t) 【参数】 详见 get_cmd_info_t 结构体说明なればない。 【返回值】 返回值 0 LOT LOT LOT LOT LOT LOT LOT	
返回值	描述
0 601	成功。
-1 only	失败,并设置 errno

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

MIPI 使用指南 1 MIPI 使用指南

【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。	' Ø _l
【返回值】	rectificated to the second of
1	111

返回值		描述
0	Indus	成功。
-1	ding	失败,并设置 errno
	thi Char.	
【芯片差异】	men kushi	
サレ米刑	anti	日不士仕

【芯片差异】

芯片类型	是否支持
Hi3559AV100ES	不支持
Hi3559A 100	支持
Hi3599AV100	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 mi ^{no}	不支持
Hi3559AV100	支持
Hi3519AV100	支持
Hi3516CV500 spenther!	支持
Hi3516EV200	不支持

【需求的

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 Chnpa 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 分析。
- input_mode_t: MIPI Rx 输入接口类型。
- mipi_data_rate_t: MIPI Rx, SLV 输入速率。
- img_rect_t: crop 属性。
- slvs_lane_rate_t: SLVS_kane 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 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

【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

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 数。 【定义】

Hi3559AV100ES:

#define SNS_MAX_RST_SOURCE_NUM Hi3559AV100:

#define SNS MAX RST SOURCE NUM

Hi3519AV100:

#define SNS MAX RST SOURCE NUM

Hi3516CV500:

#define SNS_MAX_RST_SOURCE_NUM

Hi3516EV200:

#define SNS_MAX_RST_SOURCE_NUM

【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

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

Hi3519AV100

#definerSNS_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

Chanling Industrial Technology Co., Ltd.





Hi3516CV500	2
Hi3516EV200	1

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】

【芯片差异】

SENSOR 的复位信号线编号,软件上称为 SENSOR 的复位源。		
【定义】		
【定义】 typedef unsigned int sns_rst_source_t; technology		
【芯片差异】		
芯片类型	SENSOR复位设备范围	
Hi3559AV100ES	[0, SNS_MAX_RST_SOURCE_NUM)	
Hi3559AV100	(N), 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

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

MIPI_RX_MAX_DEV_NUM

Hi3559AV100ES

#define MIAT_RX_MAX_DEV_NUM	6
Hi35584V100:	
#define MIPI_RX_MAX_DEV_NUM	8
Hi3519AV100:	
#define MIPI_RX_MAX_DEV_NUM	5
Hi3516CV500:	
#define MIPI_RX_MAX_DEV_NUM	2

【芯片差异】

Hi3516EV200:

#define MIPI_RX_MAX_DEV_NUM

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。
- 的,Charling Industrial Technology Co. Hi3516DV300/Hi3559V200/Hi3556V200/Hi3516AV300 的两**於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

Hi3519AV100

#define SLN3_MAX_DEV_NUM

【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

【芯片差异】

	芯片类型	SLVS 起始设备号
	Hi3559AV100ES	MIPI_RX_MAX_DEV_NUM
	Hi3559AV100	0
		rg.
	【注意事项】	9, Co.,
	无	amo los
	【相关数据类型及接口】	1 2/ N
	无	O Chanting Industrial Technology Co., Ltd. Chanting Industrial Technology Co., Ltd. Chanting Industrial Technology Co., Ltd.
4 <i>X</i>	(_LANE_NUM	Jing In
	【说明】	Char.
	MIPI Rx/SLVS 的 Lane 总数管	
	【定义】 anther	
	Hi3559AV100ES/Hi3559AV100):
	#define COMBO_MAX_LANE_NUM	16
	Hi3519AX 100:	

COMBO_MAX_LANE_NUM

Hi3519AV 100:

#detine COMBO MAX LANE NUM 12

Hi3516CV500:

#define COMBO_MAX_LANE_NUM

Hi3516EV200:

#define COMBO MAX LANE NUM

Hi3516EV300:

#define COMBO_MAX_LANE_NUM

【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】

一个逻辑的**, l**ink 拆分成了 2 个软件的 link。 Fushi Chanling Industri 这里的 link 是软件概念,软件上把-

【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】

Hi3559AV160ES/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

【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

【芯片差异】

无。

【注意事项】

无。

(1)。 【相关数据类型及接口】

无。

COMBO_MAX_LĬNK_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】

文档版本 05 (2019-09-12)



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:

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_RXOMAX_DEV_NUM + SLVS_MAX_DEV_NUM)
Hi3559AV100/	MET_RX_MAX_DEV_NUM
Hi3519AV100/	
Hi3516CV500/H3516EV200	

【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

Hi3516EV300:

#define WDR_VC_NUM

Hi3516EV200/Hi3518EV300:

#define WDR_VC_NUM

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

SYNC_CODE_NUM

定义】
Hi3559AV100ES/Hi3559AV100/Hi3519AV166/Hi3516CV500:
#define SYNC_CODE_NUM 40
Hi3516EV300:
define SYNC_CODE_NUM 2
i3516EV200/Hi3518EV300
efine SYNC 7

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
   LANE DIVIDE MODE BUTT
} lane_divide_mode_t;
```

Hi3559AV100:

```
= 2, Industrial Technology co.,

= 2, Industrial Technology co.,

= 2, Industrial Technology co.,

= 4, Industrial Technology co.,

= 4, Industrial Technology co.,

= 4, Industrial Technology co.,

= 6, Industrial Technology co.,

= 7, Industrial Technology co.,

typedef enum
                     LANE DIVIDE MODE 0
                    LANE DIVIDE MODE 1
                     LANE DIVIDE MODE 2
                     LANE DIVIDE MODE 3
                     LANE_DIVIDE_MODE_4
                      LANE DIVIDE MODE 5
                      LANE DIVIDE MODE 6
                     LANE_DIVIDE_MODE
                                                                                                                                                          = 7,
                      LANE_DIVIDE_MODE_8
                                                                                                                                                                  = 8,
                        LANE_DIVIDE MODE 9
                                                                                                                                                                     = 9,
                        LANE_DIXIDE_MODE_A
                                                                                                                                                                    = 0xA,
                        LANE, DIVIDE MODE B
                                                                                                                                                                     = 0xB
                      LAGE_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
   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
                                       Technology co., Ltd.
   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_DIXTOE_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**人**」」 「注意事项**人**」」 只有 MAPI 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
                                           /* mipi */
   INPUT MODE MIPI
                    = 0 \times 0
   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 */
                                            /* BT601 */
   INPUT MODE BT601
                          = 0x6,
   INPUT MODE BT656
                          = 0x7
                                           /* BT656 */
   INPUT MODE BT1120
                                           /* BT1120 */
                          = 0x8,
   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】



【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】

```
Steathen Fishi Chanting Industrial Technology Co., Ltd.

* wid+

* **
typedef struct
   int x;
   int y;
   unsigned int width;
   unsigned int height;
```

【成绩

成员名称	描述
x	Crop 起始位置 x。
у	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。

```
/* 1152Mbps */ co.`

1, /* 2304Mbps */ co.`

1, /* 2304Mbps */ co.`

Lasting industrial pathology

E.

【相关数据类型及接口】 chaning industrial

The short should be should be
```

slvs_err_check_mode_t

SLVs 的 CRC 和 ECC 模式。

【定义】

```
typedef enum
   SLVS_ERR_CHECK_MODE_NONE = 0 \times 0,
   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。

data_type_t

```
一致据类型。
【定义】
Hi3519AV100/Hi3516CV560shi
typedef enum

DATA_TYPEKel
DAT
                                          DATATYPE_RAW_12BIT,
                                    A_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 CYANT	8BIT 的 YUV422 数据。
DATA_TYPE_YUV422_PACKED	YUV422 数据在 MIPI 按 16bit raw 打包接收。

~70°,		
【芯片差异】 yed stentile 支持的数据类型		
芯片类製	支持的数据类型	
Hi3939AV100ES	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	



无。

【相关数据类型及接口】

无。

mipi_wdr_mode_t

【说明】

MIPI WDR 模式。

【定义】

```
/* Virtual Channel */
typedef enum
                                                                                                                                                                                                                                               virtual no 
                                  HI MIPI WDR MODE NONE = 0 \times 0,
                                  HI\_MIPI\_WDR\_MODE\_VC = 0x1,
                                  HI\_MIPI\_WDR\_MODE\_DT = 0x2,
                                  HI\_MIPI\_WDR\_MODE\_DOL = 0x3,
                                  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
                                         Technology Co., Ltd.
       short data_type[WDR_VC_NUM];
   } ;
} mipi_dev_attr_t;
```

【成员】

成员名称	描述
input_data_type	传输的数据类型的影
lane_id	发送端(sensor)和接收端(MIPI Rx) Lane 的对应关系 未使用的 Cane 设置为-1。
wdr_mode	MIREWOR 模式
data_type	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。

【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,
                              Industrial Technology Co., Ltd.
   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_WOR_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】

```
/* sensor SoL, EOL, SOF, EOF */
/* SAV, F&V */
typedef enum
   LVDS_SYNC_MODE SOF = 0,
                               Aling Industrial
   LVDS_SYNC_MODE_SAV,
   LVDS SYNC MODE BUTT
} lvds_sync_mode_t;
```

表1-6 LVDS 同步方式

sync_mode	尚 步方式
LVDS_SYNC_MODE_SOME	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 = 0 \times 0,
   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】

```
LVDS_VSYNC_NORMALN FLORING LODGE CO., Ltd.

LVDS_VSYNC_NORMALN FLORING LVDS_VSYNC_SHARTH VDS_VSYNC_HOONN

VDS_VSYNC_P.

s_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsync_vsyn
typedef enum
      } lvds_vaync_type_t;
```

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	Horizo ntal Blanki ng
SOL	Dodding				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 of Blanking (fix period)	V.BLK	EAV	Horizontal Blanking
			Short ExposureFrame1	l			
			ChanJi		Short Exposure		
	V.BLK		Exposurer raine in		Frame2		
		చ	E.BLK				
		ko _t 2					
	V.BLK	only					-

。。。 【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】

lvds_fid_type_t

```
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
_t
【说明】
Frame identification Id 类型 thanking thanking the first translation in the strength of th
                                    LVDS_FID NONE
                                      LVDS FID IN SAV
                                                                                                                                                                                                                                                                              /* frame identification id in SAV 4th */
                                                                                                                                                                                                = 0 \times 01,
                                    LVOS_FID_IN_DATA = 0 \times 02,
                                                                                                                                                                                                                                                                             /* frame identification id in first data
                                      LVDS FID BUTT
        } lvds_fid_type_t;
```

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】

```
. id 配置信息。

Lid 配置信息。

Lid 配置信息。

Lid 配置信息。

Chanting Industrial Technology Co.

Lid.

Chanting Industrial Technology Co.

Lid.

Lid.

Chanting Industrial Technology Co.

Lid.

Lid.

Chanting Industrial Technology Co.

Lid.

Li
typedef struct
   } lvds fid attr t
```

【成员】

成员名称	描述
fid ppe	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】

```
Industrial Technology Co., Ltd.
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。
Sheathe	
【芯片差异】	
无。	
【注意事项】	

【注意事项

【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 = 0 \times 0,
   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。

lvds_lane_work_attr_t

```
__attr_t
__ar_t
【说明】
LVDS LANE 的 LANE 的简步码工作属性。
【定义】
ypedef struct sheat
lvds_Che_work_mode
uncouned int
was_lane_w
```

【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
                                      Technology Co., Ltd.
   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;
                     sync_code(LVDS_LANE_NUM)[WDR_VC_NUM][SYNC_CODE_NUM];
                      lane_id[LVDSWANE_NUM];
   short
   unsigned short
} lvds_dev_attr_t;
```

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。
vs vac_ 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/HiSPinie 备高级属性。 【定义】

```
typedef struct
        clk mode t
                        clk mode;
  coods_dev_attr_t
                        lvds_attr;
   lvds_lane_work_attr_t lane_work_attr;
} lvds_dev_attr_ex_t;
```

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:

```
Shi Chanling Industrial Technology Co., Ltd.
typedef struct
   data_type_t
   wdr_mode_t
                        wdr_mode;
   slvs lane race
                         lane rate
                          sensor_valid_width;
                          lane id[LVDS LANE NUM];
   short
} slvs_dev_attr_t;
```

• (\$\mathbf{9}\mathbf{3}559AV100:

```
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】

	<u>cQ</u>
芯片类型	本结构体
Hi3559AV100ES/	支持
Hi3559AV100/	. 8 7 8
Hi3519AV100/	ETILO
Hi3516CV500/	不支持 SLVS
Hi3616EV200	aling .

【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
- slystane_rate_t
- _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】

```
struct

combo_dev_t

phy_cmv_mode_tattcmv_mode;

hy_cmv_t;

and

i)

only

only

iii

only

o
typedef struct
   } phy_cmv_t;
```

【成员】

成员备称	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
       mipi_dev_attr_t mipi_attr;
lvds_dev_attr_t lvds_attr;
slvs_dev_attr_t slvs_attrdus
_dev_attr_t;
_dev_attr_t;
                                  devno;
    input mode t
    mipi_data_rate_t data_rate;
    img rect t
    union
    {
    };
} combo_dev_attr_t;
```

Hi3559AV100:

```
sharthan fushi chanjing
typedef struct
  combo dev t
  input_mode_
                    input_mode;
  mipi ta rate t
                    data rate;
  imgyPect t
                    img_rect;
  chion
     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/	支持
Hi3559AV100/	
Hi3519AV100/	
Hi3516CV500/ 5781	不支持 SLVS
Hi3516EV2000	

【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

本語

「記事項】

无。

【相关数据类型及接口Men

た。

「Lidy

「

output_mode_t

【定义】

```
typedef enum
   OUTPUT_MODE_CSI
                            = 0x0,
                                              /* csi mode */
                                              /* dsi video mode */
   OUTPUT MODE DSI VIDEO
                            = 0x1,
   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】

```
このx1,
= 0x2,
= 0x2,
= 0x2,
Technology co. Ltd.

このx2,
Technology co. Ltd.

このx2,
Technology co. Ltd.

このx2,
Technology co. Ltd.

には、Cranting Industrial

を表現を表現を表現を表現を

たいない。

というない。

といるない。

といるない
```

output_format_t

MIPPYx 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
   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】

```
vid_pkt_size;

Jued short vid_hsa_pixels;

unsigned short vid_hbp_pixels;

unsigned short vid_hline_pixels

unsigned short vid_vsa_line

unsigned short vid_vbp_line

nsigned short vid

nsigned short vid
typedef struct
       unsigned short repi_cmd_size;
ync_info_t; started
} sync_info_t;
```

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】

```
devnotanting industrial rechnology co., trd.
typedef struct
   unsigned int
   short
                    output_mode;
   output_mode_t
                      video_mode;
                      output_format;
   sync_info_
                      sync_info;
   unsigned int
                      phy_data_rate;
   unagned int
                      pixel_clk;
} combo_dev_cfg_t;
```

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】

cmd_info_t

```
【说明】
发送给 MIPI Tx 设备的命令信息。
【定义】

ypedef struct spentral unsigned int unsigned int unsigned un
                                                                               unagned short
                                                                                                                                                                                                                                                                                                                                                                                                                       cmd_size;
                                                                     Unsigned char
                                                                                                                                                                                                                                                                                                                                                                                                                            *cmd;
                          } cmd_info_t;
```

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】

```
Chanjing Industrial Technology Co., Ltd.

Geno,

data_tv.

lata
typedef struct
   unsigned int
   unsigned short The
   unsigned show
                         data_param;
   unsigned cort
                        get_data_size;
unsigned char
                     *get data;
```

【成策】

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,使成为法如下:

● 在 Linux 操作系统下,加载 MIRY 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 LiteOsxx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



MIPI DEV ATTR
Devno WorkMode DataRate DataType WDRMode LinkId ImgX Im
ImgW ImgH
0 MIPI X1 RAW12 None 0, 1 0 0 384
2160
MIPI LANE INFO
Devno LaneCnt LaneID
0 4 0, 1, 2, 3, -1, -1, -1
MIPI LINK INFO
LinkIdx LaneCount LaneId PhyData0 PhyData1 AlignedData0
AlignedData1 ValidLane
0 2 0, 1 0x0 0x0 0x0 0x0
AlignedData1 ValidLane 0 2 0, 1 0x0 0x0 0x0 0x0 0x0 Invalid 1 2 2, 3 0x0 0x0 0x0 0x0 0x0 Invalid MIPI DETECT INFO
1 2 2, 3 0x0 0x0 0x0 0x0
Invalid
MIPI DETECT INFO
Devno VC width height
0 0 3840 2160 (rd)
0 1 0 0 1
0 2 0 0 man
030000
LVDS DETECT INFO
Devno VC width keeght
0 0 5480 s(e^{r)}3 648
0 1 4 0
0 2 4 0 0
0 3 0 0
DO DS 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	•		Ü	ŭ	Ü	Ü
2			0	0	0	0	0
			U	U	U	U	U
0	0 0		0	0	0	0	0
3	•	0	0	0	0	0	0
0	0 0)					
		R INFO					
Devno vo	OCRC vc1CF	RC vc2CRC v	c3CRC vc0	OrderErr v	c10rderE	rr vc201	derErr
vc30rder	Err vc0NMa	tCnt vc1NM	latCnt vc2	NMatCnt vo	:3NMatCnt		
0 0	0	0 0	0	0	0	•	0
0 0	0	0			7,0	•	
					ري.		
Devno HC	ntErr vc0H	HECC vc1HEC	C vc2HECC	vc3HECC v	DtErr	vc1DtErr	vc2DtErr
vc3DtErr				201	,,		
0			0 0	. al Technoli	0	0	0
				. 2			
Devino CM	ID FIFO REF	RR DATA_FIF	'O RERR CM	P FIFO WER	R DATA F	TFO WERE	2
0	0		0 Indits	0	0	_,,_,,	
O	0	OR INFO	× ,	O	O		
Dorran	ondowCDC 1	DaviloadCDC	71,	DotoEifoW	mita Da	+ a E i fa E	a a d
Devno H	eaderCRC	PayloadCRC	OFCCFLL	Datafilow.	rice Da	alarilor	eau
CmdFifoF		Err (SI)	_	_			
6	0	% 0	0	0		0	
0	1	The.					
	3/1/5)					
ALI	NG ERROR I	NFO					
Devno FI	FONULLErr	Lane0Err	Lane1Err	Lane2Err	Lane3Er	r Lane	4Err
Lane5Err	Lane6Err	Lane7Err	Lane8Err	Lane9Err	Lane10Er	r Lane1	1Err
ري.	,	r Lane14Er					
\mathcal{C}°_{o}	0			0	0	0	0
)	0 0	0					
)	0	, 0	· ·	O .	Ŭ	O	O
,							
~=-	10 DETT -==						
		₹					
		DataRate				inkId	ImgX
		H LaneRate					
6	SLVS	X2	RAW10	None	0, 1, 2,	3 1	64 41
3840 2	2160 I	LOW 414	4 NONE				
SLV	S LANE INE	70					
Devno	LaneCnt		LaneI	D			
6	8	0, 1	2 3	4 5 6	7		

Devno	LaneId	PhyData	AlignedDat	ta ValidLane	3
0		0x38b		0,	
0	1	0x2d4	0x2a4		
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,	
SLV	S DETECT	INFO			
Devno	VC widt	th height			
6	0 3840	2160			Δ.
6	1 0	0			Yo.
6	2 0	0		<i>c</i> o. ·	
6	3 0	0		loo,	
Devno lFifoF 6	HeaderCR ull Skew 0	C PayloadCR0 vErr 0	C Eccerrial	co 	e DataFifoRead 0
Devno lFifoF 6	HeaderCR ull Skew 0	C PayloadCR0	C Eccerrial	DataFifoWrite	e DataFifoRead 0
2TA	HeaderCR ull Skew 0 1 S PHY ERR	C PayloadCRO		0	e DataFifoRead 0
hyIdx	HeaderCR ull Skew 0 1 S PHY ERR LaneIdx	C PayloadCRO VErr 0 OR INFO AFI VAlign	CodeErr	0 DispErr	e DataFifoRead 0
hyIdx 0	HeaderCR ull Skew 0 1 S PHY ERR LaneIdx 0	C PayloadCRO VErr O OR INFORMATION AFIRMALIGN 1	CodeErr 3	0 DispErr 3	e DataFifoRead 0
hyIdx	HeaderCR ull Skew 0 1 S PHY ERR LaneIdx 0 1 5	C PayloadCRO VErr O OR INFORMATION AFICALIGN 1 1	CodeErr 3	0 DispErr	e DataFifoRead 0
hyIdx 0 0	LaneIdx 0 1cm	C PayloadCRO VErr O OR INFORMATION AFIGNALIGN 1 1 1	CodeErr 3 3	DispErr 3	e DataFifoRead 0
hyIdx 0 0	LaneIdx 0 1cm	C PayloadCRO VErr O OR INFORMATION AFICOALIGN 1 1 1 0	CodeErr 3 3 3	0 DispErr 3 3	e DataFifoRead 0
hyIdx 0 0	LaneIdx 0 1cm	AFICOALIGN AFICOALIGN 1 1 1	CodeErr 3 3 3 3	DispErr 3 3 3	e DataFifoRead 0
hyIdx 0 0	LaneIdx 0 1cm	AFit dalign AFit dalign 1 1 1 0	CodeErr 3 3 3 3 3 3	0 	e DataFifoRead 0
hyIdx 0 0	LaneIdx 0 1cm	AFICOALIGN AFICOALIGN 1 1 1 0 0	CodeErr 3 3 3 3 3 3	DispErr 3 3 3 3 3 3 3 3 3 3 3	e DataFifoRead 0
hyldx 0 0 0 0	LaneIdx 0 1st vo2 4 5 6	AFirdAlign AFirdAlign 1 1 0 0 0	CodeErr 3 3 3 3 3 3 3 3 3	0 DispErr 3 3 3 3 3 3 3 3	e DataFifoRead 0
hyldx 0 0 0 0 0	LaneIdx O Service LaneIdx O Service Service O Service O Service Service O Service Service	AFirdAlign 1 1 1 0 0 0	CodeErr 3 3 3 3 3 3 3 3 3	DispErr 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	e DataFifoRead 0
hyldx 0 0 0 0 0 0 0 0	LaneIdx 0 150 7 6 7 8	AFirdAlign AFirdAlign 1 1 0 0 0 0	CodeErr 3 3 3 3 3 3 3 3 0	DispErr 3 3 3 3 3 3 3 0	e DataFifoRead 0
hyldx 0 0 0 0 0 0 0 0 0	LaneIdx O LoneY For For For For For For For Fo	AFicoAlign AFicoAlign 1 1 0 0 0 0 0	CodeErr 3 3 3 3 3 3 0 0	DispErr 3 3 3 3 3 3 3 3 0 0 0	e DataFifoRead 0
hyldx 0 0 0 0 0 0 0 1 1	LaneIdx 0 150 7 8 9 10	AFirdAlign AFirdAlign 1 1 0 0 0 0 0 0	CodeErr 3 3 3 3 3 3 3 0 0 0	DispErr 3 3 3 3 3 3 0 0 0	e DataFifoRead 0
hyldx 0 0 0 0 0 0 0 1 1 1	LaneIdx 0 150 150 7 8 9 10 11	AFICALIGN AFICALIGN 1 1 1 0 0 0 0 0 0 0	CodeErr 3 3 3 3 3 3 3 0 0 0	DispErr 3 3 3 3 3 3 3 3 0 0 0 0 0 0 0	e DataFifoRead 0

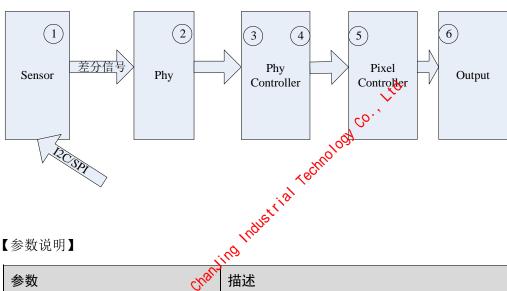
【调试信息分析】



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	. Chan	描述		
MIPI LANE	MODE (15th)	MIPI Rx 的 lane 分布模式		
DIVIDE MODE	LANE DIN TOE	MIPI_Rx 详细的 LANE 分布。		
MIPI	Devno	MIPI 设备号		
DEV OF ATTR	WorkMode	MIPI 设备工作模式: LVDS/MIPI/CMOS/SLVS 等模式		
COS	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 Tol: 2 合 1 WDR Tol: 3 合 1 WDR Tol: 4 合 1 WDR DOL2Tol: DOL2 合 1 WDR DOL3Tol: DOL3 合 1 WDR DOL4Tol: DOL4 合 1 WDR				
	LinkId	此设备使用了哪几个 Link, 对应 Link ID。 一个物理 Link 对应 4 个 Lanc。				
	ImgX	Crop 图像起始 X Co.				
	ImgY	Crop 图像起始 You				
	ImgW	Crop 图像宽要 ⁿ				
	ImgH	Crop 图像高度				
MIPI LANE	Devno	MIRY设备号				
INFO	LaneCnt LaneID CYan	Lane 数目				
	LaneID Chi	Lane ID				
MIPI LINK	LinkIdx (F)	Link ID 序号				
INFO	LaneCount	该 Link 中使用了几条 Lane				
	Lameld	对应的 Lane Id				
700	PhyData0	PHY 接收到的实时数据 0				
CORDINACE	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	Devno	MIPI_Rx 设备号				
INFO (仅	VC	Virtual Channel				
MIPI 模 式下可见)	width	MIPI 控制器检测到的图像总宽度				
- (1 (70)	height	MIPI 控制器检测到的图像总高度				



参数		描述				
LVDS	Devno	MIPI_Rx 设备号				
DETECT INFO	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	Devno	MIPI_Rx 设备号				
DETECT	Lane	Lane ID				
INFO	width	该 lane 上检测到的宽度。				
	height	该 lane 上检测到的高度。				
FSM	phy	PHY ID				
TIMEOU T AND	clkTOutCnt	时钟 Lane 从 LP 切换到 HS 超时				
ESCAPE INFO (仅	d0TOutCnt	数据 Lane 0 从 切换到 HS 超时				
MIPI 模 式下可见)	d1TOutCnt	数据 Lane 1 从 LP 切换到 HS 超时				
八下可见)	d2TOutCnt	数据 Fane 2 从 LP 切换到 HS 超时				
	d3TOutCnt	数据 Lane 3 从 LP 切换到 HS 超时				
	clkEscCnt craft	时钟 Lane 切换到 escape 模式超时				
	d0EscCnt (157)	数据 Lane 0 切换到 escape 模式超时				
	d1EscCnte ⁿ	数据 Lane 1 切换到 escape 模式超时				
	d2EseCnt	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	Devno	MIPI_Rx 设备号。				
ERRO(仅	vc0CRC	VC0 通道数据的 CRC 错误计数。				
MIPI 模 式下可见)	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。			
	ve1HECC	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。			
	ve3HECC	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	MIPA 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_WERE	MIPI 写命令 FIFO 原始中断计数。			
	DATA_FIFO	MIPI 写数据 FIFO 原始中断计数。			
ALING	Devno atter	MIPI 设备号			
ERROR INFO	FIFO_FullErr	FIFO 溢出			
00	Lane0Err	Lane0 FIFO 溢出			
codopn _A or	Lane1Err	Lane1 FIFO 溢出			
Cos	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	Devno	SLVS 设备号			
DEV ATTR	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 / RAW106 RAW12/ RAW14/ RAW16 bit 等 类型			
	WDRMode	WDR 模式:			
		• None: 非 WDR 模式			
		② 2To1: 2 合 1 WDR			
	Char	• 3To1: 3 合 1 WDR			
	<i>sush</i> i	 4To1: 4 合 1 WDR DOL2To1: DOL2 合 1WDR 			
	Then ,	• DOL3To1: DOL3 合 1WDR			
	cot sheathen Fushi chank	DOL4To1: DOL4 合 1WDR			
	_ Y	此设备使用了哪几个 Link, 对应 Link ID。			
ro Vi		一个物理 Link 对应 4 个 lane。			
cogobuyor	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	Devno	SLVS 设备号			
LANE	LaneCnt	Lane 数目			



参数		描述			
INFO	LaneID	Lane ID			
SLVS	Devno	SLVS 设备号			
DATA INFO	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	Devno	SLVS 设备号			
DETECT INFO	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	Devno	SLVS-设备号			
ERROR	HeaderCRC	数据头 CRC 错误统计			
INFO	PayloadCRC Charl	数据 CRC 错误统计			
	EccErr (1571)	ECC 错误统计			
	DataFifoWeite	数据 FIFO 写错误统计			
	DataBifoRead	数据 FIFO 读错误统计			
	CmdFifoFull	命令 FIFO 满统计			
SIGNS PHY	SkewErr	SKEW 错误统计			
SIAS	PhyIdx	PHY ID			
ERROR	LaneIdx	Lane ID			
INFO	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]

	LANE						
	4+4+2						
MIP	I DEV ATT	R					
Devno	WorkMode	DataRat	e DataType	WDRMode	ImgX	ImgY I	mgW
ImgH							
0	MIPI	X1	RAW12	None	0	0 3840	216
MIP	I LANE IN	FO					
Devno		Lan	eID				
0	0,	1, 2,	3, -1, -1, -1	, -1	ه.		
MTP	T PHY DATE	A TNFO					
		3		~~~~~~ √echnologi (ech) 0×ff	4		
PhyId	La	neId	PhyData	20100	Mipil	Data	
LvdsData				Lechi.			
0	0, 1,	2, 3	0x00,0x00,0xf	0xff	0xb5,0x9	9f,0x40,0x13	3
)xcc,0x8	8,0x38,0x	43	ustr	•			
1	4, 5,	6, 7	0x00,0x00,0x0	00,0x00	0x00,0x0	00,0x00,0x00)
0x00,0x0	0,0x00,0x	00	1100				
2	8, 9,	10,11	0x00,0x00,0x0	00,0x00	0x00,0x	00,0x00,0x00)
0x00,0x0	0,0x00,0x	00	•				
	0,0x00,0x I DETECT C widtes 3840	40,					
MIP	I DETECT	ING					
Devno V	C width	height					
	80						
0 1	W.	0					
0 2	0	0					
606.3	0	0					
Devno V							
0 0 0 1		2160					
	0	0					
0 2		0					
0 3	U	U					
LVD	S LANE DE'	TECT INFO)				
	Lane wid						
0	2 960						
0	4 960	2179					
0	5 960						
0	3 900	21/3					

	Jimout C	Clk2Esc (JIKESC .	Бансовьс		тпос	папесьз	о да	nesEsc	
	0	0	0	0		0		0	0	
0	0	0	0	0						
	1	0	0	0		0		0	0	
0	0	0	0	0						
	2	0	0	0		0		0	0	
)	0	0	0	0						
	-MIPI ERI	ROR INT I	INFO 1							
	evno Ecc			C Vc2CRC	Vc3C	RC V	c0EccCor	rct '	Vc1EccC	orrct
/c2E	ccCorrct			0	0		0 *	9.	0	
		0	0	Ü	0		0 💸	•	0	
)	()					00., 00.,			
	-MIPI ERI	OOD TNITT T	NEO 2				000			
D/	evno Vc0	D+ Va1D+	.NEO 2	7/c3D+ 7	Zc O E rm	chil	Val ErmCr		?FrmCrc	
	rmCrc	DC VCIDC	. VCZDC		•	OI C	VCITIMCI	.c vc.	ZIIMCIC	
CJI) 0	Λ	O Indu	XXIO		0	0		0
	0 (0	U	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	9 0		U	U		U
D	-MIPI ERI evno Vc0: ndryMt V	rimseq v /c2BndryM	t V¢3Bn	Vc2Frm dryMt	 \Seq \	/c3Fri		0Bndry		
/c1B	evilo vco.	rimseq v /c2BndryM	t V¢3Bn	Vc2Frm dryMt		/c3Fri		0Bndry		
/c1B	andryMt V	rimseq v vc2BndryM 0	t Vi3Bn	Vc2Frm dryMt 0	 \Seq \	/c3Fri	mSeq Vc	0Bndry	γMt	
/c1B	endryMt V	rimseq v vc2BndryM 0	t Vi3Bn	Vc2Frm dryMt 0	 \Seq \	/c3Fri	mSeq Vc	0Bndry	γMt	
/c1B	o o o o o o o o o o o o o o o o o o o	C2BndryM 0 SNEWTHE	t VASBn	Vc2Frm dryMt 0	nSeq (0 0	mSeq Vc	OBndry	7Mt 0	
7c1B	andryMt V	chenthe	t YasBn	Vc2Frm dryMt 0	vsyn	0	mSeq Vc	OBndry	vMt 0 	 VrErr
rc1B	ondryMt V O O -MIPI ERI	C2BndryM 0 CSREATHER ROW INT I	t VASBn	Vc2Frm dryMt 0 ifoRdErr	Vsyn	0 0 	mSeq Vc0 0 dFifoWrE 0	OBndry	yMt 0 ataFifoW 0	
rc1B	o o o o o o o o o o o o o o o o o o o	C2BndryM 0 CSREATHER ROW INT I	t VASBn	Vc2Frm dryMt 0 ifoRdErr	Vsyn	0 0 	mSeq Vc0 0 dFifoWrE 0	OBndry	yMt 0 ataFifoW 0	
De C	ondryMt V O -MIPI ERI evnoy Data -LVDS ERI	chenther chenth	t YasBn	Vc2Frm dryMt 0	Vsyn	0 0 Cm	mSeq Vc0 0 dFifoWrE 0	OBndry	yMt 0 ataFifoW 0	
De C	ondryMt V O O -MIPI ERI evnoy Data -LVDS ERI evno Vsy	chenther chenth	t Vi3Bn LNFO 4	Vc2Frm dryMt 0 ifoRdErr 0	Vsynd	0 C Cm Stat	mSeq Vc0 0 dFifoWrE 0	OBndry	yMt 0 ataFifoW 0	
De C	ondryMt V O O -MIPI ERI evnoy Data -LVDS ERI evno Vsy	C2BndryM C3BndryM C3Bndr	t Vi3Bn LNFO 4	Vc2Frm dryMt 0 ifoRdErr 0	Vsynd	0 C Cm Stat	mSeq Vc0 0 dFifoWrE 0	OBndry	yMt 0 ataFifoW 0	
De C	ondryMt V 0 -MIPI ERI evnoy Data -LVDS ERI evno Vsy: 0 0	C2BndryM C2BndryM CREATIVE CHEATIVE CHEATI	t YasBn LUS 0 ENFO 4 ENFO 1 EErr Cmd	Vc2Frm dryMt 0 ifoRdErr 0	Vsyndo Operr	0 0 Cmc Cmc Stat	mSeq Vc0 0 dFifoWrE 0 	OBndry	7Mt 0 ataFifoN 0	
De Control De	ondryMt V ondryMt V	COENTRO COE	t Ya3Bn LY0 INFO 4 INFO 1 Err Cmd INFO 2	Vc2Frm dryMt 0 ifoRdErr 0 WrErr P 0 (Vsyn	0 Cmc	mSeq Vc0	OBndry	7Mt 0 ataFifoT 0	
De De	ondryMt V 0 -MIPI ERI evnoy Data -LVDS ERI evno Vsy: 0 0	COENTRO COE	t Ya3Bn LY0 INFO 4 INFO 1 Err Cmd INFO 2	Vc2Frm dryMt 0 ifoRdErr 0 WrErr P 0 (Vsyn	0 Cmc	mSeq Vc0	OBndry	7Mt 0 ataFifoT 0	
De De	ondryMt V o o -MIPI ERI evnoy Data -LVDS ERI evno Vsy: o -LVDS ERI evno Lin :2RdErr	COENTRO COE	t Ya3Bn LY 0 ENFO 4 ETT Cmd ETT Cmd	Vc2Frm dryMt 0 ifoRdErr 0 iWrErr P 0 (Vsyndo OopErr	0 Cmc Cmc	mSeq Vc0 0 dFifoWrE 0 EErr	OBndry	7Mt 0 ataFifoT 0	
De De	ondryMt V o o -MIPI ERI evnoy Data -LVDS ERI evno Vsy: o -LVDS ERI evno Lin :2RdErr	COENTRO CHERTNE CHE	t Ya3Bn LY 0 ENFO 4 ETT Cmd ETT Cmd	Vc2Frm dryMt 0 ifoRdErr 0 iWrErr P 0 (Vsyndo OopErr	0 Cmc Cmc	mSeq Vc0 0 dFifoWrE 0 EErr	OBndry	7Mt 0 ataFifoT 0	

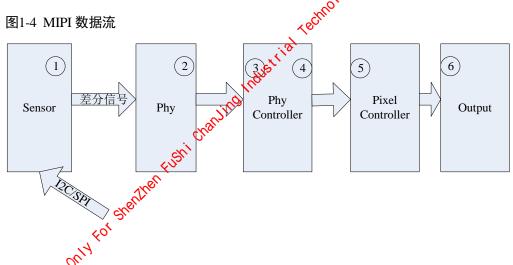
0	0	0	0	Λ	0	0	0
) 0		0	U	U	U	U
	N ERROR IN	•					
				Zww Iono	DEnn In	.no2Enn Io	n o AE non
	FIFO_FullE						
	Lane6Err						
0	0			0	U	0	0
		0					
	DEV ATTR-						
	WorkMode I		ata'I'ype	WDRMode	ImgX	ImgY	ImgW
_	eRate Val						
	SLVS	X2 R	RAW12	None	48	24 3840	2160
HIGHT	3936						
						۰9.	
	LANE INFO					,	
Devno		LaneID			ં. જ [ે]		
0	6, 4	5, 0,	7, 2, -1	, -1	² 04		
				chho,	²⁰ 00.,		
	PHY DATA			<			
	LaneID		_ (<	Jac a			
0		0x227	45x0				
0	1		100×0				
0	2	0x30d	<i>jin^{lo}</i> 0x1c				
0	3	0×0 C/s	0x0				
0	4	0x2b4	0x3a				
0	5	% 368	0x18				
0	6	€ 0x25c	0x27				
0	3che.	0x1c4	0x13	4			
	, 40°						
SLVS	PATECT IN						
Devnoy	VC width	height					
co _{log}	0 3840	2160					
0	0 3840 1 0 2 0	0					
0	3 0	0					
	DEV ERROR						
	aderCRC Pa		EccErr 1	DataFifoW	rite	DataFifoRe	ad
	ll SkewEr						
	0	0	0	0		0	
0 ()						
	PHY ERROR						
	LaneIdx A				r		
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 提供的钟, 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。MIPL-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。



【参数说明】

参数		描述		
MIPI LANE	MODE	MIPI Rx 的 lane 分布模式		
DIVIDE MODE	LANE DIVIDE	MIPI_Rx 详细的 LANE 分布。		
MIPI DEV	Devno	MIPI 设备号		
ATTR	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 Tol: 2 合 1 WDR Tol: 3 合 1 WDR Tol: 4 合 1 WDR DOL2Tol: DOL2 合 1 WDR DOL3Tol: DOL3 合 1 WDR DOL4Tol: DOL4 合 1 WDR
	ImgX	Crop 图像起始 X
	ImgY	Crop 图像起始 Y Vo.
	ImgW	Crop 图像宽度 Co.`
	ImgH	Crop 图像高度 Crop 图像 Crop
MIPI LANE	Devno	MIPI 设备号 ^{®C}
INFO	LaneCnt	Lane 数目
	LaneID	Lane ID
MIPI PHY DATA	PhyId	PHY ID 序号
INFO	LaneId	对应的 Lane Id
	PhyData Risk	PHY 对应的 4 条 Lane 接收到的实时数据
	Mipi B ata	PHY 对应的 4 条 Lane 检测到 MIPI 帧同步信号 后的实时数据
or Vulgo	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	Devno	MIPI_Rx 设备号
INFO (仅	VC	Virtual Channel
MIPI 模 式下可见)	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	Devno	MIPI_Rx 设备号
INFO (仅	VC	Virtual Channel
LVDS 模 式下可见)	width	MIPI 控制器检测到的图像总宽度
241.476)	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	Lane	Lane ID
INFO (仅	width	该 lane 上检测到的宽度。
LVDS 模 式下可见)	height	该 lane 上检测到的高度。
PHY CIL	phy	PHY ID
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
	Lane2TmOut	Data Lane 2 从 LP 切换到 HS 超时
	Lane3TmOut	Data Lane 3 从 LP 数换到 HS 超时
	Clk2Esc	Clock2 Lane 英典到 escape 模式超时
	ClkEsc	Clock Lane 切换到 escape 模式超时
	Lane0Esc	Data ane 0 切换到 escape 模式超时
	Lane1Esc	Data Lane 1 切换到 escape 模式超时
	Lane2Esc Chart	Data Lane 2 切换到 escape 模式超时
	Lane3Esc (4)5	Data Lane 3 切换到 escape 模式超时
MIPI ERROR	Devno ather Ecg2	MIPI_Rx 设备号。
INT INFO	Ecg2	Header 至少 2 个错误,ECC 无法纠错
1 (仅 MIPI 模 ぐ	Vc0CRC	VC0 通道数据的 CRC 错误计数。
式下可见)	Vc1CRC	VC1 通道数据的 CRC 错误计数。
0,	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。
	Vc1EccCorret	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。
	Vc3EccCorret	VC3 通道 Header 的 ECC 已纠正的错误计数。
MIPI	Devno	MIPI_Rx 设备号。
ERROR INT INFO	Vc0Dt	VC0 通道不支持的数据类型计数



参数		描述
2 (仅 MIPI 模	Vc1Dt	VC1 通道不支持的数据类型计数
式下可见)	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	Devno	MIPI_Rx 设备号。
ERROR INT INFO	Vc0FrmSeq	VC0 的帧序错误计数
3 (仅 MIPI 模	Vc1FrmSeq	VC1 的帧序错误计数
式下可见)	Vc2FrmSeq	VC2 的帧序错误计数
	Vc3FrmSeq	VC3 的帧序错误计数
	Vc0BndryMt	VCQ通道的帧起始与帧结束短包不匹配计数
	Vc1BndryMt	◆C1 通道的帧起始与帧结束短包不匹配计数
	Vc2BndryMt Charl	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	Devno atter	MIPI_Rx 设备号。
INT INFO	DataFifoRdErr	MIPI CTRL 读数据 FIFO 中断计数
4 (仅 MIPI 模 ♂	CmdFifoRdErr	MIPI CTRL 读命令 FIFO 中断计数
式下可观)	Vsync	MIPI CTR vsync 中断计数
COS	CmdFifoWrErr	MIPI CTRL 写命令 FIFO 中断计数
	DataFifoWrErr	MIPI CTRL 写数据 FIFO 中断计数
LVDS ERROR	Devno	MIPI 设备号
INT INFO	Vsync	lvds vsync 中断计数
1(仅 LVDS 模	CmdRdErr	lvds CMD_FIFO 读错误中断计数
式下可见)	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	Link0WrErr	link0 读 FIFO 错误中断计数
2(仅	Link1WrErr	link1 读 FIFO 错误中断计数
LVDS 模 式下可见)	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	Devno	MIPI 设备号
INT INFO	Lane0Err	LaneO 同步错误中断计数
3(仅 LVDS 模	Lane1Err	Lane1 同步错误中断计数
式下可见)	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	Lances。同步错误中断计数
	Lane6Err	Cane6 同步错误中断计数
	Lane7Err Chair	Lane7 同步错误中断计数
	Lane8Err & ST	Lane8 同步错误中断计数
	Lane9Ent	Lane9 同步错误中断计数
	Lane 10Err	Lane10 同步错误中断计数
9.	Lane11Err	Lane11 同步错误中断计数
ALING ^N ER R OR	Devno	MIPI 设备号
INFO	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	Devno	SLVS 设备号
DEV ATTR	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: o非 WDR 模式
		• 2Te1: 2 合 1 WDR
		• 3To1: 3 合 1 WDR • 4To1: 4 合 1 WDR
	Chan	• DOL2To1: DOL2 合 1WDR
	<i>tu</i> shi	• DOL3To1: DOL3 合 1WDR
	other !	DOL4To1: DOL4 台 1WDR
	Img & ten fushi chani	Crop 图像起始 X
		Crop 图像起始 Y
hio.	ImgW	Crop 图像宽度
CORDIN	ImgH	Crop 图像高度
	LaneRate	Lane 的速率
	ValidW	SENSOR 的实际有效宽度
SLVS	Devno	SLVS 设备号
LANE INFO	LaneCnt	Lane 数目
	LaneID	Lane ID
SLVS	PhyId	PHY ID
PHY DATA	LaneID	Lane ID
INFO	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	Devno	SLVS 设备号
INFO	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	Devno	SLVS 设备号
DEV ERROR	HeaderCRC	数据头 CRC 错误统计
INFO	PayloadCRC	数据 CRC 错误统计 💛 💍
	EccErr	ECC 错误统计 (O).
	DataFifoWrite	数据 FIFO 写错误统计
	DataFifoRead	数据 FIFO 读错误统计
	CmdFifoFull	命令FFO满统计
	SkewErr	SKEW 错误统计
SLVS	Phyldx chart	PHY ID
PHY ERROR	LaneIdx AFifoAlign	Lane ID
INFO	AFifoAlign	FIFO 对齐错误统计
	CodeFor	10B8B 编码错误统计
	BiSpErr	DISPARITY 错误统计

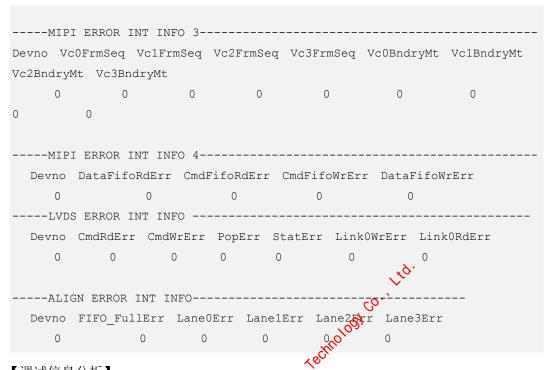
Hi3516CV500

【调试信息】

Module: [MIPI_RX],	Build Time[Oct 8	2018, 09:	27:31]			
MIPI LANE DIV	TIDE MODE					
MODE LANE	DIVIDE					
0	4					
MIPI DEV ATTE	₹					
Devno WorkMode	DataRate	DataType	WDRMode	Img	K Im	gY
ImgW ImgH						
0 MIPI	X1	RAW12	None	0	204	2592
1944						



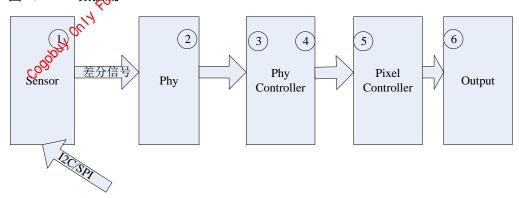
MIPI	LANE IN	NFO					
Devno		Lan	neID				
0	0,	1, 2,	3				
MIDI	. DIII DA						
PhyId	Lá	aneId	PhyDa	ıta	MipiD	ata	
LvdsData	0 1	0 0		0 00 0 40	0.06.0.0		_
			0x00,0x00,	0x00,0x49	0x36,0x2	b,0x34,0x4	5
0x9b,0x1a	.a2/5 , 0x0	00,0x00					
MIPI	DETECT	INFO					
Devno VC	width	height					
0 0	2592	1760			19.		
0 1	0	0			دن.		
0 2	0	0			4		
0 3	0	0		~	2/03		
LVDS	DETECT	INFO		-			
Devno VC	width	height		(8)			
0 0	2592	1760		eti,			
0 1	0	0	Indi	35			
0 2	0	0	On;				
0 3	0	0	chans.		ologico., rq.		
LVDS	LANE DE	ETECT INE	»				
Devno L	ane wio	dth heigh	nt				
0	0 64	8 10 1759					
0	1 64	1759					
0	2 4064	8 1759					
0	3,4 64						
PHX	CIL ERR	INT INFO					
					1TmOut Lane	e2TmOut	
Lane3TmOu	t Clk2E	sc ClkEs	c Lane0Esc	Lane1Es	c Lane2Esc	Lane3Esc	
0	0	0	0	0	0	0	0
0	0	0	0 0				
MIPI	ERROR 1	INT INFO	1				
Devno	Ecc2 Vo	OCRC Vc1	.CRC Vc2CR	C Vc3CRC	Vc0EccCorro	t Vc1EccC	orrct
Vc2EccCor	rct Vc3	EccCorrct	-				
0	0	0 (0	0	0	0	
0	0						
MIPI	ERROR 1	INT INFO	2				
Devno	Vc0Dt V	vc1Dt Vc2	Dt Vc3Dt	Vc0FrmCrc	Vc1FrmCrc	Vc2FrmCrc	
Vc3FrmCrc	:						
0	0	0 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 Pixel Controller 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 的末端交貌,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	Devno	MIPI 设备号
DEV ATTR	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 1 WDR DOL2To1 DOL2 合 1 WDR DOL3To1: DOL3 合 1 WDR DOL4To1: DOL4 合 1 WDR
	ImgX	Crop 图像起始 X
	ImgY Chan	Crop 图像起始 Y
	ImgY Chi	Crop 图像宽度
	ImgH 1	Crop 图像高度
MIPI	Devno	MIPI 设备号
LANE INFO	LaneID	Lane ID
MIPI RPY DATA	PhyId	PHY ID 序号
INFO	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	Devno	MIPI_Rx 设备号
DETECT INFO (仅	VC	Virtual Channel
MIPI 模	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	Devno	MIPI_Rx 设备号
DETECT INFO (仅	VC	Virtual Channel
LVDS 模 式下可见)	width	MIPI 控制器检测到的图像总宽度
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	height	MIPI 控制器检测到的图像总高度
LVDS LANE	Devno	MIPI_Rx 设备号
DETECT	Lane	Lane ID
INFO (仅 LVDS 模	width	该 lane 上检测到的宽度。
式下可见)	height	该 lane 上检测到的高度。
PHY CIL ERR INT	PhyId	PHY ID CO
INFO	Clk2TmOut	Clock2 Lane 从CP 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 cran	Data Lane 2 从 LP 切换到 HS 超时
	Lane3TmOut	Data Lane 3 从 LP 切换到 HS 超时
	Clk2Escatter	Clock2 Lane 切换到 escape 模式超时
	CIKES	Clock Lane 切换到 escape 模式超时
9	Lane0Esc	Data Lane 0 切换到 escape 模式超时
Codopny	Lane1Esc	Data Lane 1 切换到 escape 模式超时
coor	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	Devno	MIPI_Rx 设备号。
ERROR INT INFO	Ecc2	Header 至少 2 个错误,ECC 无法纠错
1 (仅 MIPI 模	Vc0CRC	VC0 通道数据的 CRC 错误计数。
式下可见)	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。
	Vc0EccCorret	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	Devno	MIPI_Rx 设备号。
ERROR INT INFO	Vc0Dt	VC0 通道不支持的数据类型计数
2(仅 MIPI 模	Vc1Dt	VC1 通道不支持的数据类型计数
式下可见)	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 通道放数据错误计数
MIPI ERROR	Devno	MIRE Rx 设备号。
INT INFO	Vc0FrmSeq	℃ C0 的帧序错误计数
3 (仅 MIPI 模	Vc1FrmSeq Chair	VC1 的帧序错误计数
式下可见)	Vc2FrmSeq & ST	VC2 的帧序错误计数
	Vc3Frm3eq	VC3 的帧序错误计数
	VcQBndryMt	VC0 通道的帧起始与帧结束短包不匹配计数
Q.	♥c1BndryMt	VC1 通道的帧起始与帧结束短包不匹配计数
codopin or	Vc2BndryMt	VC2 通道的帧起始与帧结束短包不匹配计数
CO3	Vc3BndryMt	VC3 通道的帧起始与帧结束短包不匹配计数
MIPI ERROR	Devno	MIPI_Rx 设备号
INT INFO	DataFifoRdErr	MIPI CTRL 读数据 FIFO 中断计数
4 (仅 MIPI 模	CmdFifoRdErr	MIPI CTRL 读命令 FIFO 中断计数
式下可见)	CmdFifoWrErr	MIPI CTRL 写命令 FIFO 中断计数
	DataFifoWrErr	MIPI CTRL 写数据 FIFO 中断计数
LVDS ERROR	Devno	MIPI 设备号
INT INFO	CmdRdErr	lvds CMD_FIFO 读错误中断计数
(仅 LVDS	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
رال (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 の対

Hi3516EV200

```
【调试信息】

Module: [MIPI_RX], Build Time Wov 24 2018, 14:50:01]
----MIPI LANE DIVIDE MODE
 MODE
----MIPI DEV ABTR--
 Devno WorkMode DataRate
                      DataType WDRMode ImgX ImgY
         MIPI X1
                     RAW12 None 0 204 2304
----MIPI LANE INFO-----
               LaneID
  Devno
                0, 1,
----MIPI PHY DATA INFO-----
 PhyId LaneId
                      PhyData
LvdsData
        0, 1 0x00,0x00 0x36,0x2b 0x9b,0x1ad275
----MIPI DETECT INFO-----
 Devno VC width height
 0 0 2304 1296
```

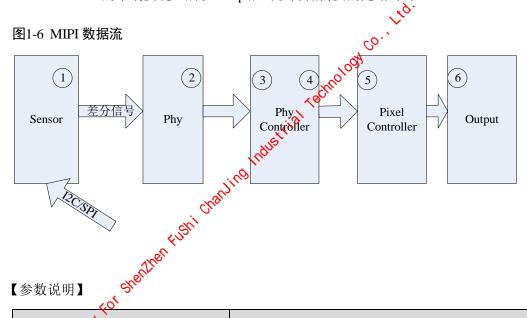


TANKS DEFINED TANKS	
LVDS DETECT INFO	
Devno VC width height	
0 0 2304 1296	
0 1 0 0	
LVDS LANE DETECT INFO	
Devno Lane width height	
0 0 0 0	
0 1 0 0	
PHY CIL ERR INT INFO	_
PhyId Clk2TmOut ClkTmOut Lane0TmOut Lane1TmOut Lane2TmOut	
Lane3TmOut Clk2Esc ClkEsc Lane0Esc Lane1Esc Lane2Esc Lane3Esc	
	0
~	O
MIPI ERROR INT INFO 1	c+
Vc2EccCorret Vc3EccCorret	
Vc2EccCorret Vc3EccCorret 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
o o tribus	
MIDT EDDOR THE THEO 2 and 1.	
Devno VcODt Vc1Dt Vc2Dt Vc3Dt Vc0FrmCrc Vc1FrmCrc Vc2FrmCrc	
W-2F	
Vc3FrmCrc (vis)	
Vc3FrmCrc	
o o trento o o o o o	
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
O O O O O O O O O O O O O O O O O O O	
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
O O O O O O O O O O O O O O O O O O O	 cr
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 cr
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 cr
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 rr
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 rr
0 0 the 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 rr
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 rr
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 rr
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

----ALIGN ERROR INT INFO---Devno FIFO FullErr LaneOErr Lane1Err Lane2Err Lane3Err 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。



参数	14	描述
MIPI OUT	MODE	MIPI Rx 的 lane 分布模式
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



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	Devno	MIPI 设备号
INFO	LaneID	Lane ID
MIPI PHY DATA	PhyId	PHY ID 序号 2005
INFO	LaneId	对应的 Lane of
	PhyData	PHY 对应的 2 条 Lane 接收到的实时数据
	MipiData	PKY 对应的 2 条 Lane 检测到 MIPI 帧同步信号 治的实时数据
	LvdsData Change	PHY 对应的 2 条 Lane 检测到 LVDS 帧同步信号 后的实时数据
MIPI DETECT	Devno Aren	MIPI_Rx 设备号
INFO (仅	VC S	Virtual Channel
MIPI 模 式下可见)	width	MIPI 控制器检测到的图像总宽度
,o kuo	height	MIPI 控制器检测到的图像总高度
LVPS DETECT	Devno	MIPI_Rx 设备号
INFO (仅	VC	Virtual Channel
LVDS 模 式下可见)	width	MIPI 控制器检测到的图像总宽度
2(1.176)	height	MIPI 控制器检测到的图像总高度
LVDS LANE	Devno	MIPI_Rx 设备号
DETECT	Lane	Lane ID
INFO (仅 LVDS 模	width	该 lane 上检测到的宽度。
式下可见)	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	
11.10	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	Devno	MIPI_Rx 设备号。	
INT INFO	Ecc2	Header 至少 2 个错误,ECC 无法纠错	
1 (仅 MIPI 模	Vc0CRC	VC0 通道数据的 CRC 错误计数。	
式下可见)	Vc1CRC	VC1 通道数据的 CRC 错误计数。	
	Vc0EccCorrct	VCCC通道 Header 的 ECC 已纠正的错误计数。	
	Vc1EccCorret	♥C1 通道 Header 的 ECC 已纠正的错误计数。	
MIPI ERROR	Devno cram	MIPI_Rx 设备号。	
INT INFO	Vc0Dt Kishi	VC0 通道不支持的数据类型计数	
2 (仅 MIPI 模	Vc1Dt the	VC1 通道不支持的数据类型计数	
式下可见)	VcQFrmCrc	VC0 通道帧数据错误计数	
, or	Vc1FrmCrc	VC1 通道帧数据错误计数	
MIPION ERROR	Devno	MIPI_Rx 设备号。	
INT INFO	Vc0FrmSeq	VC0 的帧序错误计数	
3 (仅 MIPI 模	Vc1FrmSeq	VC1 的帧序错误计数	
式下可见)	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	Devno	MIPI_Rx 设备号。	
INT INFO	DataFifoRdErr	MIPI CTRL 读数据 FIFO 中断计数	
4 (仅 MIPI 模	CmdFifoRdErr	MIPI CTRL 读命令 FIFO 中断计数	
式下可见)	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	Devno	MIPI 设备号
	Vsync	lvds vsync 中断计数
1(仅 LVDS 模	CmdRdErr	lvds CMD_FIFO 读错误中断计数
式下可见)	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	Devno	MIPI 设备号
INT INFO	Link0WrErr	link0 读 FIFO 错误中断计数
2(仅 LVDS 模	Link1WrErr	link1 读 FIFO 错误的断计数
式下可见)	Link0RdErr	link0 写 FIFO 替误中断计数
	Link1RdErr	link1 写,FiFO 错误中断计数
ALING ERROR	Devno	MIPS设备号
INFO	FIFO_FullErr	PIFO 溢出
	Lane0Err Char.	Lane0 FIFO 溢出
	Lane1Err (1587)	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
-----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 【setmo"
MIPI_Tx DEV	devno	设备号点
CONFIG	lane0	>= 0 d lane 号码。
	lane1	9 :禁用。
	lane2	
	lane2 change	
	output_mode	0: csi mode
	benthe	1: dsi video mode
	SIL.	2: dsi command mode
	phy_data_rate	Phy 数据率。
codoppina	pixel_clk(KHz)	像素时钟(单位 KHz)。
Cool	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
SINC	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	width	检测到的宽。
STATUS	height	检测到的高。
	HoriAll	检测到的局。 水平总宽。 垂直总高。 水平前肩。
	VertAll	垂直总高。 2000
	hbp	水平前肩。人名
	hsa	水平同步脉冲个数。
	vsa	垂直同步脉冲个数。
vsa 垂直同步脉冲个数。 MIPI 具体规格请参考芯片手册《Hi35xxVxxx ultra-HD Mobile Camera SoC 用户指南》		

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。

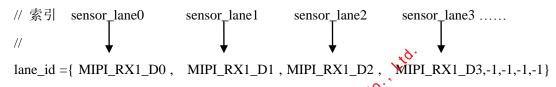
表1-7 SENSOR 与 MIPI_Rx 管脚关系

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:



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,MPI_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_Greq = (250 * 12) / 4 = 750 即每个 Lane 最高频率为 750MHz