MIPI User Guide

Issue 04

Date 2019-09-12

HiSilicon (Shanghai) Technologies Co., Ltd.

Address: New R&D Center, 49 Wuhe Road, Bantian,
Longgang District,
Shenzhen 518129 P. R. China

Website: http://www.hisilicon.com/en/

Email: support@hisilicon.com

About This Document

Related Versions

The following table lists the product versions related to this document.

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

NOTE

- Unless otherwise stated, Hi3559C V100 and Hi3559A V100 contents are consistent.
- Unless otherwise stated, Hi3556A V100 and Hi3519A V100 contents are consistent.
- Unless otherwise stated, Hi3516D V300, Hi3559 V200, Hi3556 V200, Hi3516A V300, and Hi3516C V500 contents are consistent.

Intended Audience

This document is intended for:

- Technical support engineers
- Software development engineers

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
DANGER	Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
CAUTION	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.
NOTICE	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal injury.
NOTE	Calls attention to important information, best practices and tips. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.

Change History

Changes between document issues are cumulative. The latest document issue contains all changes made in previous issues.

Issue 04 (2019-09-12)

This issue is the fourth official release, which incorporates the following changes:

In section 1.3, Table 1-4 and Table 1-5 are modified.

Section 1.4 is modified.

In section 1.5, the **Note** field of mipi_dev_attr_t is modified.

Issue 03 (2019-08-01)

This issue is the third official release, which incorporates the following changes:

In section 1.4, the **Note** field of HI_MIPI_SET_HS_MODE is modified.

Section 1.6 "Parameter of the MIPI TX Module" is added.

Issue 02 (2019-06-20)

This issue is the second official release, which incorporates the following changes:

In section 1.4, the **Note** fields of HI_MIPI_TX_ENABLE, HI_MIPI_SET_PHY_CMVMODE, and HI_MIPI_TX_ENABLE are updated.

In section 1.5, the **Syntax** field of data_type_t is updated. The **Member** fields of sync_info_t, combo_dev_cfg_t, and cmd_info_t are updated.

Section 1.7 is updated.

Issue 01 (2019-05-20)

This issue is the first official release, which incorporates the following changes:

In section 1.5, the description in the **Note** field of input_mode_t is modified.

Section 1.6 is modified

Issue 00B16 (2019-04-15)

This issue is the sixteenth draft release, which incorporates the following changes:

The support for the Hi3516D V200 chip is added.

In section 1.4, HI_MIPI_TX_GET_CMD is added.

In section 1.5, get_cmd_info_t, slvs_err_check_mode_t, mipi_clk_mode_t, lvds_lane_work_mode_t, lvds_lane_work_attr_t, and lvds_dev_attr_ex_t are added. input_mode_t, data_type_t, data_type_t, slvs_dev_attr_t, and combo_dev_attr_t are updated.

Section 1.6 is updated.

Issue 00B15 (2019-02-28)

This issue is the fifteenth draft release, which incorporates the following changes:

The support for the Hi3516A V300 chip is added.

Section 1.6 is modified

Issue 00B14 (2019-01-15)

This issue is the fourteenth draft release, which incorporates the following changes:

In section 1.3, Table 1-2 is modified.

In section 1.5, the descriptions in the **Chip Difference** field of mipi_data_rate_t and the **Note** field of mipi_dev_attr_t are modified.

Issue 00B13 (2018-11-13)

This issue is the thirteenth draft release, which incorporates the following changes:

The content related to Hi3518E V300 is added.

In section 1.5, WDR_VC_NUM, SYNC_CODE_NUM, wdr_mode_t, video_mode_t, cmd_info_t, and mipi_dev_attr_t are updated.

In section 1.6, the **Parameter Description** field below Figure 1-5 is updated.

Issue 00B12 (2018-10-30)

This issue is the twelfth draft release, which incorporates the following changes:

The description of Hi3516E V200/Hi3516E V300 is added.

Issue 00B11 (2018-10-15)

This issue is the eleventh draft release, which incorporates the following changes:

In section 1.5, WDR_VC_NUM and wdr_mode_t are modified.

Section 1.6 is modified.

Issue 00B10 (2018-09-29)

This issue is the tenth draft release, which incorporates the following changes:

The description of Hi3559 V200/Hi3556 V200 is added.

Issue 00B09 (2018-09-04)

This issue is the ninth draft release, which incorporates the following changes:

The description of Hi3516C V500/Hi3516D V300 is added.

Issue 00B08 (2018-08-08)

This issue is the eighth draft release, which incorporates the following changes:

In section 1.6, the description of Hi3519A V100 proc information is added.

Issue 00B07 (2018-07-06)

This issue is the seventh draft release, which incorporates the following changes:

In section 1.5, the description in the **Syntax** field of WDR_VC_NUM is modified.

Section 1.6 is modified.

Issue 00B06 (2018-05-15)

This issue is the sixth draft release, which incorporates the following changes:

In section 1.5, the descriptions in the **Note** field of img_rect_t and the **Member** field of slvs_dev_attr_t are modified.

In section 1.4, HI_MIPI_CLEAR is added. The description in the **Note** field of HI_MIPI_SET_DEV_ATTR is modified.

In section 1.5, COMS_MAX_DEV_NUM is added. The descriptions in the **Definition** and **Note** fields of input_mode_t are modified.

Issue 00B05 (2018-04-28)

This issue is the fifth draft release, which incorporates the following changes:

The descriptions about the Hi3519A V100 are added.

In section 1.4, HI_MIPI_TX_SET_DEV_CFG, HI_MIPI_TX_SET_CMD, and HI_MIPI_TX_ENABLE are added.

In section 1.5, HI_MIPI_TX_IOC_MAGIC to cmd_info_t are added.

Issue 00B04 (2018-01-15)

This issue is the fourth draft release, which incorporates the following changes:

The descriptions about the Hi3559A V100 and Hi3559C V100 are added.

Issue 00B03 (2017-09-20)

This issue is the third draft release, which incorporates the following changes:

In section 1.4, HI_MIPI_SET_OUTPUT_MSB is deleted. The description in the **Definition** field of HI_MIPI_DISABLE_SENSOR_CLOCK is modified.

In section 1.5, SYNC_CODE_NUM, wdr_mode_t, lvds_sync_mode_t, lvds_bit_endian_t, lvds_vsync_type_t, lvds_vsync_attr_t, lvds_fid_type_t, lvds_fid_attr_t, lvds_dev_attr_t and slvs_dev_attr_t are added. data_rate_t is renamed to mipi_data_rate_t and the descriptions in its **Definition** field are modified. The contents related to phy_cmv_mode_t and combo_dev_attr_t are modified. output_msb_t is deleted.

combo_dev_t, sns_rst_source_t, and sns_clk_source_t are modified.

In section 1.6, the descriptions in the **Debugging Information** and **Parameter Description** fields are modified.

Issue 00B02 (2017-07-20)

This issue is the second draft release, which incorporates the following changes:

In section 1.3, table 1-2 is modified.

Issue 00B01 (2017-06-14)

This issue is the first draft release.

Contents

Figures

Tables

1 Introduction

NOTICE

- Hi3516E V300, Hi3518E V300, Hi3516D V200, and Hi3516E V200 do not support MIPI TX.
- Unless otherwise specified, the descriptions of Hi3516D V200 and Hi3516E V300 are the same, and the descriptions of Hi3518E V300 and Hi3516E V200 are the same.

1.1 Overview

The mobile industry processor interface (MIPI) RX module receives raw video data by using low-voltage differential signals, converts the received serial differential signals into digital camera (DC) timings, and then transmit the timings to the downstream video capture (VICAP) module.

The MIPI RX supports the MIPI D-PHY, LVDS, and high-speed serial pixel interface (HiSPi) serial video signal inputs and is compatible with the DC video interface.

The SLVS-EC interface is defined by SONY and used for capturing images with high frame rate and resolution. The interface can convert the high-speed serial data into the digital camera timing and transfers it to the downstream VICAP module.

The SLVS-EC serial video interface provides higher transmission bandwidth, lower power consumption, and lower data redundancy in packaging mode. The SLVS-EC interface provides more reliable and stable transmission in application.

1.2 Important Concepts

- MIPI

  The MIPI described in this document refers to the communications interface which uses the D-PHY transmission specifications at the physical layer and CSI-2 at the protocol layer.
- LVDS

The low-voltage differential signaling (LVDS) technology differentiates blanking regions and valid data by using the sync code.

- SLVS-EC

  The scalable low voltage signaling embedded clock (SLVS_EC) is an interface paralleled to the MIPI interface and used for capturing images with high frame rate and resolution.

- Lane

  A lane is a high-speed differential pair for connecting the TX end and RX end. It can be a clock lane or a data lane.

- Link

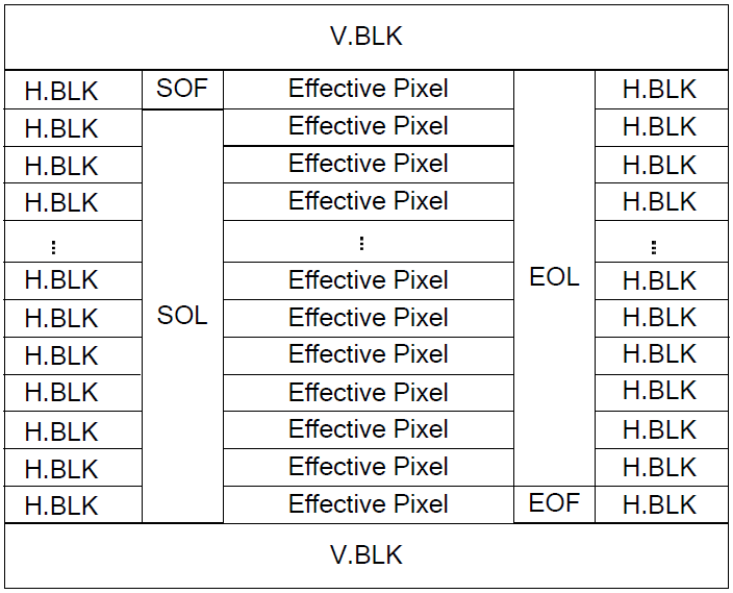
  A link consists of a clock lane and at least one data lane between the TX end and RX end. In this document, link is a software concept, and each link contains two data lanes.

  Sync code

  The MIPI interface uses the short packet in CSI-2 for synchronization, and the LVDS uses the sync code to differentiate valid data and blanking regions. There are two sync modes for the LVDS:

  – SOF and EOF indicate the start and end of a frame respectively, and SOL and EOL indicate the start and end of a line respectively. Figure 1-1 shows the sync mode.

Figure 1-1 SOF/EOF/SOL/EOL sync mode

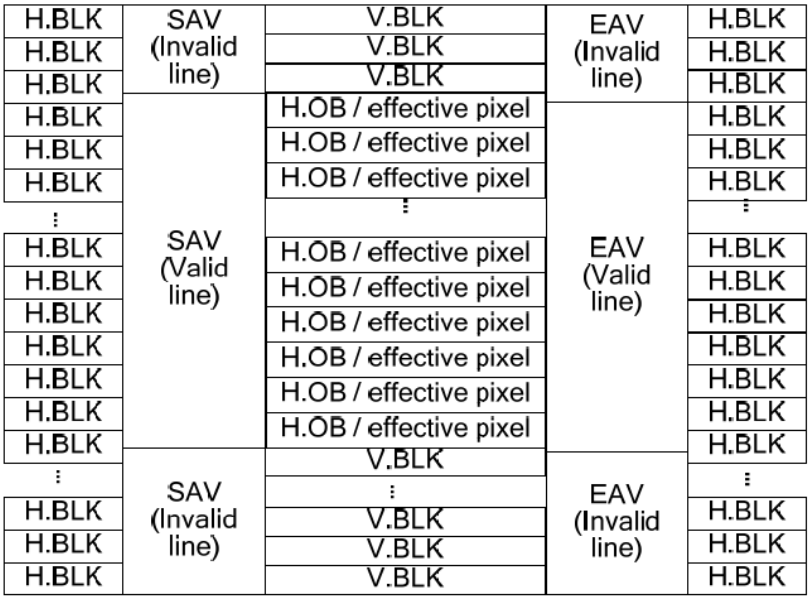


  – SAV (invalid) and EAV (invalid) indicate the start and end of invalid data in the blanking region respectively, and SAV (valid) and EAV (valid) indicate the start and end of valid pixel data respectively.

    Each sync code consists of four fields. The bit width of each field is consistent with that of pixel data. The first three fields are fixed reference codewords, and the fourth field is defined by the sensor vendor.

    Because the sync code differs according to the sensor, it needs to be configured based on the sensor. Figure 1-2 shows the sync mode.

Figure 1-2 SAV/EAV sync mode



- DOL

  Digital overlap (DOL) indicates the WDR function of Sony.

1.3 Function Description

The MIPI RX module is a collection unit that supports multiple differential video input interfaces. It uses the combo-PHY to receive data through the MIPI, LVDS, sub-LVDS, HiSPi, or DC interface. Depending on the functional mode configuration, the MIPI RX allows data transmission at different speeds and resolutions and supports multiple external input devices. Table 1-1 shows the maximum number of lanes supported by each chip.

Table 1-1 Maximum number of lanes

Chip	Maximum Number of Lanes
Hi3559A V100ES	8-lane MIPI input or 16-lane LVDS input
Hi3559A V100	8-lane MIPI input or 16-lane LVDS input
Hi3519A V100	8-lane MIPI input or 12-lane LVDS input
Hi3516C V500/Hi3516E V300/Hi3516D V200	4-lane MIPI input or 4-lane LVDS input
Hi3516E V200/Hi3518E V300	2-lane MIPI input or 2-lane LVDS input

The MIPI RX can interwork with multiple sensors at a time. Table 1-2 describes the maximum sensors supported by each chip.

Table 1-2 Maximum number of sensors

Chip	Maximum Number of Sensors
Hi3559A V100ES	6
Hi3559A V100	8
Hi3519A V100	5
Hi3516C V5000 Hi3516E V200 Hi3516E V300 Hi3518E V300	1
Hi3516D V300 Hi3559 V200 Hi3556 V200 Hi3516A V300	2

Connecting to different number of sensors requires different number of lanes. Therefore, you need to determine the lane distribution mode of the MIPI RX. For details about the lane distribution mode, see Table 1-3.

Table 1-3 MIPI RX lane distribution mode

Chip	Mode	DEV0	DEV1	DEV2	DEV3	DEV4	DEV5	DEV6	DEV7
Hi3559A V100ES	0	L0–L15	*N*	*N*	*N*	*N*	*N*	*N*	*N*
	1	L0–L11	*N*	*N*	*N*	L12–L15	*N*	*N*	*N*
	2	L0–L11	*N*	*N*	*N*	L12 and L14	L13 and L15	*N*	*N*
	3	L0–L7	*N*	L8–L15	*N*	*N*	*N*	*N*	*N*
	4	L0–L7	*N*	L8–L11	*N*	L12–L15	*N*	*N*	*N*
	5	L0–L7	*N*	L8–L11	*N*	L12 and L14	L13 and L15	*N*	*N*
	6	L0–L7	*N*	L8 and L10	L9 L11	L12 and L14	L13 and L15	*N*	*N*
	7	L0–L3	L4–L7	L8–L11	*N*	L12–L15	*N*	*N*	*N*
	8	L0–L3	L4–L7	L8–L11	*N*	L12 and L14	L13 and	*N*	*N*

Chip	Mode	DEV0	DEV1	DEV2	DEV3	DEV4	DEV5	DEV6	DEV7
							L15		
	9	L0–L3	L4–L7	L8 and L10	L9 L11	L12 and L14	L13 and L15	*N*	*N*
Hi3559A V100	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 and L14	L13 and 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 and L14	L13 and L15
	6	L0–L7	*N*	*N*	*N*	L8 and L10	L9 L11	L12 and L14	L13 and L15
	7	L0–L3	*N*	L4–L7	*N*	L8–L11	*N*	L12–L15	*N*
	8	L0–L3	*N*	L4–L7	*N*	L8–L11	*N*	L12 and L14	L13 and L15
	9	L0–L3	*N*	L4–L7	*N*	L8 and L10	L9 and L11	L12 and L14	L13 and L15
	A	L0–L3	*N*	L4 and L6	L5 and L7	L8 and L10	L9 and L11	L12 and L14	L13 and L15
	B	L0 and L2	L1 and L3	L4 and L6	L5 and L7	L8 and L10	L9 and L11	L12 and L14	L13 and L15
Hi3519A V100	0	L0–L11	N	N	N	N	N	N	N
	1	L0–L7	N	N	L8–L11	N	N	N	N
	2	L0–L7	N	N	L8 and L10	L9 and L11	N	N	N
	3	L0–L3	L4–L11	N	N	N	N	N	N
	4	L0–L3	L4–L7	N	L8–L11	N	N	N	N
	5	L0–L3	L4–L7	N	L8 and	L9 and	N	N	N

Chip	Mode	DEV0	DEV1	DEV2	DEV3	DEV4	DEV5	DEV6	DEV7
					L10	L11			
	6	L0–L3	L4 and L6	L5 L7	L8 and L10	L9 and L11	N	N	N
Hi3516C V500	0	L0–L3	N	N	N	N	N	N	N
	1	L0 and L2	L1 and L3	N	N	N	N	N	N
Hi3516E V200	0	L0 and L2	N	N	N	N	N	N	N
Hi3516E V300	0	L0–L3	N	N	N	N	N	N	N

The SLVS-EC interface supports image capture with higher frame rate and resolution. By receiving high-speed serial data through PHY, converting data into the DC timing, and configuring different functional modes, the SLVS-EC can support multiple external input devices and the transmission of data with multiple speeds and resolutions. Table 1-4 shows the maximum number of supported lanes.

Table 1-4 Maximum number of lanes

Chip	Definition
Hi3559A V100ES	The SLVS-EC supports a maximum of 8-lane SLVS input.
Hi3559A V100	The SLVS-EC supports a maximum of 8-lane SLVS input.
Hi3519A V100	The SLVS-EC supports a maximum of 8-lane SLVS input.
Hi3516C V500/ Hi3516E V200	SLVS-EC input is not supported.

The lane pins of MIPI RX are multiplexed with that of the SLVS-EC. A lane can be used by the MIPI RX or SLVS-EC at a time. For details about the lane pin connection, see Table 1-5.

Table 1-5 Lane multiplexing relationship between MIPI RX and SLVS-EC

Chip	LANE	MIPI 0	MIPI 1	MIPI 2	MIPI 3	MIPI 4	MIPI 5	MIPI 6	MIPI 7	SLVS 0	SLVS1	SLVS 2	SLVS3
Hi3559A V100ES	Lane0	√	-	-	-	-	-	-	-	√	√	-	-
	Lane1	√	-	-	-	-	-	-	-	√	√	-	-
	Lane2	√	-	-	-	-	-	-	-	√	√	-	-
	Lane3	√	-	-	-	-	-	-	-	√	√	-	-
	Lane4	√	√	-	-	-	-	-	-	√	√	-	-

Chip	LANE	MIPI 0	MIPI 1	MIPI 2	MIPI 3	MIPI 4	MIPI 5	MIPI 6	MIPI 7	SLVS 0	SLVS1	SLVS 2	SLVS3
	Lane5	√	√	-	-	-	-	-	-	√	√	-	-
	Lane6	√	√	-	-	-	-	-	-	√	√	-	-
	Lane7	√	√	-	-	-	-	-	-	√	√	-	-
	Lane8	√	-	√			-	-	-	-	-	√	√
	Lane9	√	-	√	√		-	-	-	-	-	√	√
	Lane10	√	-	√			-	-	-	-	-	√	√
	Lane11	√	-	√	√		-	-	-	-	-	√	√
	Lane12	√	-	-	-	√		-	-	-	-	√	√
	Lane13	√	-	-	-	√	√	-	-	-	-	√	√
	Lane14	√	-	-	-	√		-	-	-	-	√	√
	Lane15	√	-	-	-	√	√	-	-	-	-	√	√
Hi3559A V100	Lane0	√								√	√	-	-
	Lane1	√	√							√	√	-	-
	Lane2	√								√	√	-	-
	Lane3	√	√							√	√	-	-
	Lane4	√		√						√	√	-	-
	Lane5	√		√	√					√	√	-	-
	Lane6	√		√						√	√	-	-
	Lane7	√		√	√					√	√	-	-
	Lane8	√				√				-	-	√	√
	Lane9	√				√	√			-	-	√	√
	Lane10	√				√				-	-	√	√
	Lane11	√				√	√			-	-	√	√
	Lane12	√				√		√		-	-	√	√
	Lane13	√				√		√	√	-	-	√	√
	Lane14	√				√		√		-	-	√	√
	Lane15	√				√		√	√	-	-	√	√
Hi3519A V100	Lane0	√								√			
	Lane1	√								√			
	Lane2	√								√			

Chip	LANE	MIPI 0	MIPI 1	MIPI 2	MIPI 3	MIPI 4	MIPI 5	MIPI 6	MIPI 7	SLVS 0	SLVS1	SLVS 2	SLVS3
	Lane3	√								√			
	Lane4	√	√							√			
	Lane5	√	√	√						√			
	Lane6	√	√							√			
	Lane7	√	√	√						√			
	Lane8	√	√		√								
	Lane9	√	√		√	√							
	Lane10	√	√		√								
	Lane11	√	√		√	√							

1.4 API Reference

The MIPI RX provides the function of interworking with sensor timings. It provides the ioctl interface and the following APIs:

- HI_MIPI_SET_DEV_ATTR: Sets the attributes of MIPI, SLVS, and parallel devices.
- HI_MIPI_SET_HS_MODE: Sets the lane distribution of the MIPI RX.
- HI_MIPI_SET_PHY_CMVMODE: Configures the common-mode voltage mode.
- HI_MIPI_RESET_SENSOR: Resets the sensor.
- HI_MIPI_UNRESET_SENSOR: Deasserts the reset on the sensor.
- HI_MIPI_RESET_MIPI: Resets the MIPI RX.
- HI_MIPI_UNRESET_MIPI: Deasserts the reset on the MIPI RX.
- HI_MIPI_RESET_SLVS: Resets the SLVS.
- HI_MIPI_UNRESET_SLVS: Deasserts the reset on the SLVS.
- HI_MIPI_ENABLE_MIPI_CLOCK: Enables the clock of the MIPI device.
- HI_MIPI_DISABLE_MIPI_CLOCK: Disables the clock of the MIPI device.
- HI_MIPI_ENABLE_SLVS_CLOCK: Enables the clock of the SLVS device.
- HI_MIPI_DISABLE_SLVS_CLOCK: Disables the clock of the SLVS device.
- HI_MIPI_ENABLE_SENSOR_CLOCK: Enables the sensor clock.
- HI_MIPI_DISABLE_SENSOR_CLOCK: Disables the sensor clock.
- HI_MIPI_CLEAR: Clears the configurations of the device.

The MIPI TX provides the display and cascading functions. It provides the ioctl interface and the following APIs:

- HI_MIPI_TX_SET_DEV_CFG: Sets the attributes of an MIPI RX device.
- HI_MIPI_TX_SET_CMD: Sets the command data to be sent to an MIPI TX device.

- HI_MIPI_TX_ENABLE: Enables the MIPI TX device.
- HI_MIPI_TX_DISABLE: Disables the MIPI TX device.
- HI_MIPI_TX_GET_CMD: Reads information from peripherals.

NOTICE

If the VO does not use MIPI_TX for output, it is forbidden to use the MIPI_TX interface and use MIPI_TX output for the boot screen. After the interface is called, the MIPI_TX will be enabled. When the supply voltage of AVDD3318_MIPITX exceeds 1.8 V, MIPI_TX will be damaged.

HI_MIPI_SET_DEV_ATTR

[Description]

Sets the attributes of MIPI and SLVS devices.

[Definition]

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

[Parameter]

combo_dev_attr_t pointer

[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

None

[Requirement]

Header file: **hi_mipi.h**

[Note]

- The following APIs also need to be configured besides HI_MIPI_SET_DEV_ATTR.
- Setting the mode: The API is HI_MIPI_SET_HS_MODE.
- Enabling the MIPI/SLVS clock: The API is HI_MIPI_ENABLE_MIPI_CLOCK/HI_MIPI_ENABLE_SLVS_CLOCK.
- Resetting the MIPI/SLVS: The API is HI_MIPI_RESET_MIPI/HI_MIPI_RESET_SLVS.
- Enabling the sensor clock: The API is HI_MIPI_ENABLE_SENSOR_CLOCK.
- Resetting the sensor: The API is HI_MIPI_RESET_SENSOR.

- Deasserting the reset on the MIPI/SLVS: The API is HI_MIPI_UNRESET_MIPI/HI_MIPI_UNRESET_SLVS.
- Deasserting the reset on the sensor: The API is HI_MIPI_UNRESET_SENSOR.
- The recommended configuration process is as follows:
  1. Set the mode.
  2. Enable the clocks of multiple MIPI/SLVS.
  3. Reset the MIPI RX/SLVS connected to multiple sensors.
  4. Enable the clocks connected to multiple sensors.
  5. Reset all connected sensors.
  6. Set the attributes of the MIPI RX/SLVS.
  7. Deassert the reset on the MIPI RX/SLVS connected to multiple sensors.
  8. Deassert the reset on all connected sensors.
- The recommended exiting process is as follows:
  1. Reset all connected sensors.
  2. Disable the clocks connected to multiple sensors.
  3. Reset the MIPI RX/SLVS connected to multiple sensors.
  4. Clear the configurations of the MIPI RX/SLVS devices connected to multiple sensors.
  5. Disable the clocks of multiple MIPI/SLVS devices.
- Operating sensor reset signal cables or clock signal cables affects all sensors connected to the cables.

[See Also]

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

HI_MIPI_SET_HS_MODE

[Description]

Sets the lane distribution of the MIPI RX (invalid for the SLVS).

[Definition]

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

[Parameter]

lane_divide_mode_t pointer

[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Supported
Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Supported
Hi3516E V200	Supported

[Requirement]

Header file: **hi_mipi.h**

[Note]

When multiple sensor inputs are connected, you are advised to arrange the global lane distribution based on the hardware connection during initialization. This API cannot be called during multi-sensor data collection. Otherwise, the data collection of other sensors may be affected.

HI_MIPI_SET_PHY_CMVMODE

[Description]

Configures the common-mode voltage mode.

[Definition]

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

[Parameter]

phy_cmv_t, indicating the pointer

[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Supported
Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Supported
Hi3516E V200	Supported

[Requirement]

Header file: **hi_mipi.h**

[Note]

The SLVS mode is not supported.

HI_MIPI_RESET_SENSOR

[Description]

Resets the sensor.

[Definition]

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

[Parameter]

sns_rst_source_t, indicating the sensor reset signal cable number

[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Supported
Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Supported

Chip	Supported or Not
Hi3516E V200	Supported

[Requirement]

Header file: **hi_mipi.h**

[Note]

None

HI_MIPI_UNRESET_SENSOR

[Description]

Deasserts the reset on the sensor.

[Definition]

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

[Parameter]

sns_rst_source_t, indicating the sensor reset signal cable number

[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Supported
Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Supported
Hi3516E V200	Supported

[Requirement]

Header file: **hi_mipi.h**

[Note]

None

HI_MIPI_RESET_MIPI

[Description]

Resets the MIPI RX.

[Definition]

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

[Parameter]

combo_dev_t, indicating the device ID

[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Supported
Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Supported
Hi3516E V200	Supported

[Requirement]

Header file: **hi_mipi.h**

[Note]

None

HI_MIPI_UNRESET_MIPI

[Description]

Deasserts the reset on the MIPI RX.

[Definition]

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

[Parameter]

combo_dev_t, indicating the device ID

[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Supported
Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Supported
Hi3516E V200	Supported

[Requirement]

Header file: **hi_mipi.h**

[Note]

None

HI_MIPI_RESET_SLVS

[Description]

Resets the SLVS.

[Definition]

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

[Parameter]

combo_dev_t, indicating the device ID

[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Supported
Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Not supported
Hi3516E V200	Not supported

[Requirement]

Header file: **hi_mipi.h**

[Note]

None

HI_MIPI_UNRESET_SLVS

[Description]

Deasserts the reset on the SLVS.

[Definition]

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

[Parameter]

combo_dev_t, indicating the device ID

[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Supported
Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Not supported

Chip	Supported or Not
Hi3516E V200	Not supported

[Requirement]

Header file: **hi_mipi.h**

[Note]

None

HI_MIPI_ENABLE_MIPI_CLOCK

[Description]

Enables the clock of the MIPI device.

[Definition]

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

[Parameter]

combo_dev_t, indicating the device ID

[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Supported
Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Supported
Hi3516E V200	Supported

[Requirement]

Header file: **hi_mipi.h**

[Note]

None

HI_MIPI_DISABLE_MIPI_CLOCK

[Description]

Disables the clock of the MIPI device.

[Definition]

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

[Parameter]

combo_dev_t, indicating the device ID

[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Supported
Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Supported
Hi3516E V200	Supported

[Requirement]

Header file: **hi_mipi.h**

[Note]

None

HI_MIPI_ENABLE_SLVS_CLOCK

[Description]

Enables the clock of the SLVS device.

[Definition]

```
#define HI_MIPI_ENABLE_SLVS_CLOCK             _IOW(HI_MIPI_IOC_MAGIC,
```



```
0x0e, combo_dev_t)
```

[Parameter]

combo_dev_t, indicating the device ID

[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Supported
Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Not supported
Hi3516E V200	Not supported

[Requirement]

Header file: **hi_mipi.h**

[Note]

None

HI_MIPI_DISABLE_SLVS_CLOCK

[Description]

Disables the clock of the SLVS device.

[Definition]

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

[Parameter]

combo_dev_t, indicating the device ID

[Return Value]

Return Value	Description
0	Success

Return Value	Description
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Supported
Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Not supported
Hi3516E V200	Not supported

[Requirement]

Header file: **hi_mipi.h**

[Note]

None

HI_MIPI_ENABLE_SENSOR_CLOCK

[Description]

Enables the sensor clock.

[Definition]

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

[Parameter]

Source ID of the sensor clock device

[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Supported

Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Supported
Hi3516E V200	Supported

[Requirement]

Header file: **hi_mipi.h**

[Note]

None

HI_MIPI_DISABLE_SENSOR_CLOCK

[Description]

Disables the sensor clock.

[Definition]

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

[Parameter]

Source ID of the sensor clock device

[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Supported
Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Supported

[Requirement]

Header file: **hi_mipi.h**

[Note]

None

HI_MIPI_CLEAR

[Description]

Clears the configurations of the device.

[Syntax]

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

[Parameter]

Device ID

[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Not supported
Hi3559A V100	Supported
Hi3519A V100	Not supported
Hi3516C V500	Not supported
Hi3516E V200	Not supported

[Requirement]

Header file: **hi_mipi.h**

[Note]

This MIPI is called when a service exits and is used to clear the configurations of the device.

HI_MIPI_TX_SET_DEV_CFG

[Description]

Sets the attributes of an MIPI RX device.

[Definition]


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

[Parameter]

Attributes of an MIPI device

[[Return Value]

Return Value	Description
0	Success
-1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Not supported
Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Supported
Hi3516E V200	Not supported

[Requirement]

Header file: hi_mipi_tx.h

[Note]

None

HI_MIPI_TX_SET_CMD

[Description]

Sets the command data to be sent to an MIPI TX device.

[Definition]

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

[Parameter]

Command data sent to an MIPI TX device

[[Return Value]

Return Value	Description
0	Success
-1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559AV100ES	Not supported
Hi3559AV100	Supported
Hi3519AV100	Supported
Hi3516C V500	Supported
Hi3516E V200	Not supported

[Requirement]

Header file: **hi_mipi_tx.h**

[Note]

None

HI_MIPI_TX_GET_CMD

[Description]

Reads information from peripherals.

[Syntax]

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

[Parameter]

For details, see get_cmd_info_t.

[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Difference]

Chip Type	Supported or Not
Hi3559A V100ES	Not supported
Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Supported
Hi3516E V200	Not supported

[Requirement]

Header file: **hi_mipi_tx.h**

[Note]

None

HI_MIPI_TX_ENABLE

[Description]

Enables an MIPI TX device.

[Definition]

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

[Parameter]

None

[[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559AV100ES	Not supported
Hi3559AV100	Supported
Hi3519AV100	Supported
Hi3516C V500	Supported
Hi3516E V200	Not supported

[Requirement]

Header file: hi_mipi_tx.h

[Note]

After this API is called, the MIPI works in HS mode.

HI_MIPI_TX_DISABLE

[Description]

Disables an MIPI TX device.

[Definition]

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

[Parameter]

None

[[Return Value]

Return Value	Description
0	Success
–1	Failure. errno is configured.

[Chip Difference]

Chip	Supported or Not
Hi3559A V100ES	Not supported
Hi3559A V100	Supported
Hi3519A V100	Supported
Hi3516C V500	Supported
Hi3516E V200	Not supported

[Requirement]

Header file: hi_mipi_tx.h

[Note]

After this API is called, the MIPI works in LP mode.

1.5 Data Structures

The MIPI RX provides the following data structures:

- HI_MIPI_IOC_MAGIC: Defines the magic number of the MIPI RX ioctl command.
- combo_dev_t: Defines the type of MIPI RX and SLVS devices.
- SNS_MAX_RST_SOURCE_NUM: Defines the number of reset signal cables of the sensor.
- SNS_MAX_CLK_SOURCE_NUM: Defines the number of clock signal cables of the sensor.
- sns_rst_source_t: Defines the ID of a sensor reset signal cable. The cable is called the sensor reset source in terms of software.
- sns_clk_source_t: Defines the ID of a sensor clock signal cable. The cable is called the sensor clock source in terms of software.
- MIPI_RX_MAX_DEV_NUM: Defines the number of devices supported by the MIPI RX.
- SLVS_MAX_DEV_NUM: Defines the number of devices supported by the SLVS.
- SLVS_DEV_NUM_START: Defines the start device ID of the SLVS.
- COMBO_MAX_LANE_NUM: Specifies the maximum number of lanes supported by the device.
- MAX_LANE_NUM_PER_LINK: Specifies the number of lanes of an MIPI RX link.
- MIPI_LANE_NUM: Specifies the maximum number of lanes supported by the MIPI device of the MIPI RX.
- LVDS_LANE_NUM: Specifies the number of lanes supported by the LVDS/HiSPi interface.
- SLVS_LANE_NUM: Specifies the maximum number of lanes supported by the SLVS devices.
- COMS_MAX_DEV_NUM: Specifies the maximum number of supported parallel devices.
- COMBO_MAX_LINK_NUM: Specifies the maximum number of links supported by the MIPI RX.
- COMBO_MAX_DEV_NUM: Specifies the number of the MIPI RX devices.
- WDR_VC_NUM: Specifies the maximum number of supported virtual channels.
- SYNC_CODE_NUM: Defines the number of sync codes of per virtual channel of the LVDS.
- lane_divide_mode_t: Specifies the lane distribution of the MIPI RX.
- input_mode_t: Specifies the MIPI RX input modes.
- mipi_data_rate_t: Defines the input rate of the MIPI RX and SLVS.
- img_rect_t: Defines crop attributes.
- slvs_lane_rate_t: Defines the input rate of the SLVS lane.
- slvs_err_check_mode_t: Defines the SLVS error check mode.
- data_type_t: Defines the data type for transmission.
- mipi_wdr_mode_t: Defines the MIPI WDR mode.
- mipi_dev_attr_t: Defines the attributes of the MIPI device.
- wdr_mode_t: Defines the LVDS WDR mode.
- lvds_sync_mode_t: Defines the LVDS sync mode.
- lvds_bit_endian_t: Defines the bit endian mode.
- lvds_vsync_type_t: Defines the LVDS vsync type.
- lvds_vsync_attr_t: Defines the LVDS vsync parameter.

- lvds_fid_type_t: Defines the frame ID type.
- lvds_fid_attr_t: Defines the frame ID configuration information.
- mipi_clk_mode_t: Defines the clock sharing mode.
- lvds_lane_work_mode_t: Defines the arrival mode of the synchronization code of the LVDS lane.
- lvds_lane_work_attr_t: Defines the working attributes of the synchronization code of the LVDS lane.
- lvds_dev_attr_t: Defines the device attributes supported by the LVDS, SubLVDS, and HiSPi interfaces.
- lvds_dev_attr_ex_t: Defines the advanced device attributes supported by the LVDS, sub-LVDS, and HiSPi interfaces.
- slvs_dev_attr_t: Defines the attributes of the SLVS device.
- phy_cmv_mode_t: Defines the PHY common-mode voltage mode.
- phy_cmv_t: Defines the configuration information of the PHY common-mode voltage.
- combo_dev_attr_t: Defines the combo device attributes.
- HI_MIPI_TX_IOC_MAGIC: Defines the magic number of **MIPI RX ioctl** in a command.
- LANE_MAX_NUM: Defines the maximum number of lanes supported by the MIPI TX.
- output_mode_t: Defines the MIPI TX output mode.
- video_mode_t: Defines the MIPI TX video mode.
- output_format_t: Defines the MIPI TX output format.
- sync_info_t: Defines the synchronization information of an MIPI TX device.
- combo_dev_cfg_t: Defines the attributes of an MIPI TX device.
- cmd_info_t: Defines the command data to be sent to an MIPI TX device.
- get_cmd_info_t: Defines the command information sent to the MIPI TX device.

HI_MIPI_IOC_MAGIC

[Description]

Defines the magic number of the **MIPI RX ioctl** in a command.

[Definition]

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

[Member]

None

[Chip Difference]

None

[Note]

None

[See Also]

None

combo_dev_t

[Description]

Defines the type of MIPI RX and SLVS devices.

[Definition]

```
typedef unsigned int combo_dev_t;
```

[Chip Difference]

Chip	MIPI Device Range	SLVS Device Range
Hi3559A V100ES	[0, MIPI_RX_MAX_DEV_NUM)	[SLVS_DEV_NUM_START, COMBO_MAX_LANE_NUM)
Hi3559A V100	[0, MIPI_RX_MAX_DEV_NUM)	[SLVS_DEV_NUM_START, SLVS_MAX_DEV_NUM)
Hi3519A V100	[0, MIPI_RX_MAX_DEV_NUM)	[0, SLVS_MAX_DEV_NUM)
Hi3516C V500	[0, MIPI_RX_MAX_DEV_NUM)	0

[Note]

None

[See Also]

- combo_dev_attr_t
- HI_MIPI_SET_DEV_ATTR
- HI_MIPI_RESET_SENSOR
- HI_MIPI_UNRESET_SENSOR
- HI_MIPI_RESET_MIPI
- HI_MIPI_UNRESET_MIPI

SNS_MAX_RST_SOURCE_NUM

[Description]

Defines the number of reset signal cables of the sensor.

[Definition]

Hi3559A V100ES:

```
#define SNS_MAX_RST_SOURCE_NUM    3
```

Hi3559A V100:

```
#define SNS_MAX_RST_SOURCE_NUM    4
```

Hi3519AV100:


```
#define SNS_MAX_RST_SOURCE_NUM    3
```

Hi3516C V500:

```
#define SNS_MAX_RST_SOURCE_NUM    2
```

Hi3516E V200:

```
#define SNS_MAX_RST_SOURCE_NUM    1
```

[Chip Difference]

Chip	Number of Reset Signal Cables of the Sensor
Hi3559A V100ES	3
Hi3559A V100	4
Hi3519A V100	3
Hi3516C V500	2
Hi3516E V200	1

[Note]

None

[See Also]

None

SNS_MAX_CLK_SOURCE_NUM

[Description]

Defines the number of clock signal cables of the sensor.

[Definition]

Hi3559A V100ES:

```
#define SNS_MAX_CLK_SOURCE_NUM    3
```

Hi3559A V100:

```
#define SNS_MAX_CLK_SOURCE_NUM    4
```

Hi3519AV100:

```
#define SNS_MAX_CLK_SOURCE_NUM    3
```

Hi3516C V500:

```
#define SNS_MAX_CLK_SOURCE_NUM    2
```

Hi3516E V200:

```
#define SNS_MAX_CLK_SOURCE_NUM    1
```


[Chip Difference]

Chip	Number of Sensor Clock Signal Cables
Hi3559A V100ES	3
Hi3559A V100	4
Hi3519A V100	3
Hi3516C V500	2
Hi3516E V200	1

[Note]

None

[See Also]

None

sns_rst_source_t

[Description]

Defines the ID of a sensor reset signal cable. The cable is called the sensor reset source in terms of software.

[Definition]

```
typedef unsigned int sns_rst_source_t;
```

[Chip Difference]

Chip	Sensor Reset Device Range
Hi3559A V100ES	[0, SNS_MAX_RST_SOURCE_NUM)
Hi3559A V100	[0, SNS_MAX_RST_SOURCE_NUM)
Hi3519A V100	[0, SNS_MAX_RST_SOURCE_NUM)
Hi3516C V500	[0, SNS_MAX_RST_SOURCE_NUM)
Hi3516E V200	[0, SNS_MAX_RST_SOURCE_NUM)

[Note]

Each sensor reset signal cable can connect to two sensors. You need to confirm the number of each sensor reset signal cable according to the signal cable connections on board.

[See Also]

- HI_MIPI_RESET_SLVS
- HI_MIPI_UNRESET_SLVS

sns_clk_source_t

[Description]

Defines the ID of a sensor clock signal cable. The cable is called the sensor clock source in terms of software.

[Definition]

```
typedef unsigned int sns_clk_source_t;
```

[Chip Difference]

Chip	Sensor Reset Device Range
Hi3559A V100ES	[0, SNS_MAX_CLK_SOURCE_NUM)
Hi3559A V100	[0, SNS_MAX_CLK_SOURCE_NUM)
Hi3519A V100	[0, SNS_MAX_CLK_SOURCE_NUM)
Hi3516C V500	[0, SNS_MAX_CLK_SOURCE_NUM)
Hi3516E V200	[0, SNS_MAX_CLK_SOURCE_NUM)

[Note]

Each sensor reset signal cable can connect to two sensors. You need to confirm the number of each sensor reset signal cable according to the signal cable connections on board.

[See Also]

- HI_MIPI_ENABLE_SENSOR_CLOCK
- HI_MIPI_DISABLE_SENSOR_CLOCK

MIPI_RX_MAX_DEV_NUM

[Description]

Defines the number of devices supported by the MIPI RX.

[Definition]

Hi3559A V100ES:

```
#define MIPI_RX_MAX_DEV_NUM       6
```

Hi3559A V100:

```
#define MIPI_RX_MAX_DEV_NUM       8
```

Hi3519A V100:

```
#define MIPI_RX_MAX_DEV_NUM       5
```

Hi3516C V500:

```
#define MIPI_RX_MAX_DEV_NUM       2
```


Hi3516E V200:

```
#define MIPI_RX_MAX_DEV_NUM       1
```

[Chip Difference]

Chip	Number of Devices Supported by the MIPI RX
Hi3559A V100ES	6
Hi3559A V100	8
Hi3519A V100	5
Hi3516C V500	2
Hi3516E V200	1

[Note]

- For Hi3516C V500, only one MIPI RX device is available at a time.
- For Hi3516D V300/Hi3559 V200/Hi3556 V200/Hi3516A V300, two MIPI RX devices are available at a time.

[See Also]

None

SLVS_MAX_DEV_NUM

[Description]

Defines the number of devices supported by the SLVS.

[Definition]

Hi3559A V100ES/Hi3559A V100:

```
#define SLVS_MAX_DEV_NUM       4
```

Hi3519A V100:

```
#define SLVS_MAX_DEV_NUM       1
```

[Chip Difference]

None

[Note]

None

[See Also]

None

SLVS_DEV_NUM_START

[Description]

Defines the start device ID of the SLVS.

[Definition]

Hi3559A V100ES:

```
#define SLVS_DEV_NUM_START      MIPI_RX_MAX_DEV_NUM
```

Hi3559A V100:

```
#define SLVS_DEV_NUM_START      0
```

[Chip Difference]

Chip	Start Device ID of the SLVS
Hi3559A V100ES	MIPI_RX_MAX_DEV_NUM
Hi3559A V100	0

[Note]

None

[See Also]

None

COMBO_MAX_LANE_NUM

[Description]

Defines the total number of MIPI RX and SLVS lanes.

[Definition]

Hi3559A V100ES/Hi3559A V100:

```
#define COMBO_MAX_LANE_NUM      16
```

Hi3519A V100:

```
#define COMBO_MAX_LANE_NUM      12
```

Hi3516C V500:

```
#define COMBO_MAX_LANE_NUM      4
```

Hi3516E V200:

```
#define COMBO_MAX_LANE_NUM      2
```

Hi3516E V300:

```
#define COMBO_MAX_LANE_NUM      4
```


[Chip Difference]

None

[Note]

None

[See Also]

None

MAX_LANE_NUM_PER_LINK

[Description]

Specifies the number of lanes of an MIPI RX link.

[Definition]

```
#define MAX_LANE_NUM_PER_LINK   2
```

[Chip Difference]

None

[Note]

The link is a software concept. A logical link is split into two links of the software.

[See Also]

None

MIPI_LANE_NUM

[Description]

Specifies the maximum number of lanes supported by the MIPI device of the MIPI RX.

[Definition]

Hi3559A V100ES/Hi3559A V100:

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

Hi3519A V100:

```
#define MIPI_LANE_NUM          8
```

Hi3516C V500:

```
#define MIPI_LANE_NUM          4
```

Hi3516E V200:

```
#define MIPI_LANE_NUM          2
```

Hi3516E V300:

```
#define MIPI_LANE_NUM          4
```


[Chip Difference]

None

[Note]

None

[See Also]

None

LVDS_LANE_NUM

[Description]

Specifies the maximum number of lanes supported by the LVDS device of the MIPI RX.

[Definition]

Hi3559A V100ES/Hi3559A V100:

```
#define LVDS_LANE_NUM      COMBO_MAX_LANE_NUM
```

Hi3519A V100:

```
#define LVDS_LANE_NUM          12
```

Hi3516C V500:

```
#define LVDS_LANE_NUM          4
```

Hi3516E V200:

```
#define LVDS_LANE_NUM          2
```

Hi3516E V300:

```
#define LVDS_LANE_NUM          4
```

[Chip Difference]

None

[Note]

None

[See Also]

None

SLVS_LANE_NUM

[Description]

Specifies the maximum number of lanes supported by the SLVS devices.

[Definition]

Hi3559A V100ES/Hi3559A V100:

```
#define SLVS_LANE_NUM          8
```


Hi3519A V100:

```
#define SLVS_LANE_NUM          4
```

[Chip Difference]

None

[Note]

None

[See Also]

None

COMS_MAX_DEV_NUM

[Description]

Specifies the maximum number of supported parallel devices.

[Definition]

Hi3559A V100:

```
#define COMS_MAX_DEV_NUM       3
```

[Chip Difference]

None

[Note]

None

[See Also]

None

COMBO_MAX_LINK_NUM

[Description]

Specifies the maximum number of links supported by the MIPI RX.

[Definition]

Hi3559A V100ES/Hi3559A V100:

```
#define COMBO_MAX_LINK_NUM     8
```

[Chip Difference]

None

[Note]

None

[See Also]

None

COMBO_MAX_DEV_NUM

[Description]

Specifies the number of the MIPI RX devices.

[Definition]

Hi3559A V100ES:

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

Hi3559A V100/Hi3519A V100/Hi3516C V500/Hi3516E V200:

```
#define COMBO_MAX_DEV_NUM     MIPI_RX_MAX_DEV_NUM
```

[Chip Difference]

Chip	Number of MIPI RX Devices
Hi3559A V100ES	(MIPI_RX_MAX_DEV_NUM + SLVS_MAX_DEV_NUM)
Hi3559A V100/Hi3519A V100/ Hi3516C V500/H3516E V200	MIPI_RX_MAX_DEV_NUM

[Note]

- For Hi3559A V100ES, the IDs of MIPI and SLVS devices are unified.
- For Hi3559A V100/Hi3519A V100, the IDs of MIPI and SLVS devices are independent.

[See Also]

None

WDR_VC_NUM

[Description]

Specifies the maximum number of supported virtual channels.

[Definition]

Hi3559A V100ES/Hi3559A V100/Hi3519A V100/Hi3516C V500:

```
#define WDR_VC_NUM        4
```

Hi3516E V300:

```
#define WDR_VC_NUM        2
```

Hi3516E V200/Hi3518E V300:

```
#define WDR_VC_NUM        2
```

[Chip Difference]

None

[Note]

None

[See Also]

None

SYNC_CODE_NUM

[Description]

Defines the number of sync codes of per virtual channel of the LVDS.

[Definition]

Hi3559A V100ES/Hi3559A V100/Hi3519A V100/Hi3516C V500:

```
#define SYNC_CODE_NUM          4
```

Hi3516EV300:

```
#define SYNC_CODE_NUM          2
```

Hi3516EV200/Hi3518E V300:

```
#define SYNC_CODE_NUM          2
```

[Chip Difference]

None

[Note]

None

[See Also]

None

lane_divide_mode_t

[Description]

Defines the lane distribution of the MIPI RX.

[Definition]

Hi3559A V100ES:

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



```
    LANE_DIVIDE_MODE_6    = 6,
    LANE_DIVIDE_MODE_7    = 7,
    LANE_DIVIDE_MODE_8    = 8,
    LANE_DIVIDE_MODE_9    = 9,
    LANE_DIVIDE_MODE_BUTT
} lane_divide_mode_t;
```

Hi3559A V100:

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

Hi3519A V100:

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

Hi3516C V500:

```
typedef enum
{
    LANE_DIVIDE_MODE_0    = 0,
    LANE_DIVIDE_MODE_1    = 1,
    LANE_DIVIDE_MODE_BUTT
```



```
} lane_divide_mode_t;
```

Hi3516E V200:

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

[Chip Difference]

Chip	Lane Distribution of the MIPI RX
Hi3559A V100ES	[LANE_DIVIDE_MODE_0, LANE_DIVIDE_MODE_9]
Hi3559A V100	[LANE_DIVIDE_MODE_0, LANE_DIVIDE_MODE_B]
Hi3519A V100	[LANE_DIVIDE_MODE_0, LANE_DIVIDE_MODE_6]
Hi3516C V500	[LANE_DIVIDE_MODE_0, LANE_DIVIDE_MODE_1]
Hi3516E V200	LANE_DIVIDE_MODE_0

[Note]

Only the MIPI requires the lane distribution setting.

[See Also]

HI_MIPI_SET_HS_MODE

input_mode_t

[Description]

Specifies the MIPI RX input modes.

[Definition]

```
typedef enum
{
    INPUT_MODE_MIPI        = 0x0,           /* mipi * /
    INPUT_MODE_SUBLVDS     = 0x1,           /* SUB_LVDS * /
    INPUT_MODE_LVDS        = 0x2,           /* LVDS * /
    INPUT_MODE_HISPI       = 0x3,           /* HISPI * /
    INPUT_MODE_SLVS        = 0x4,           /* SLVS * /
    INPUT_MODE_CMOS        = 0x5,           /* CMOS * /
    INPUT_MODE_BT601       = 0x6,           /* BT601 * /
    INPUT_MODE_BT656       = 0x7,           /* BT656 * /
    INPUT_MODE_BT1120      = 0x8,           /* BT1120 * /
    INPUT_MODE_BYPASS      = 0x9,           /* MIPI Bypass * /
    INPUT_MODE_LVDS_EX     = 0xA,           /* LVDS EX * /
```



```
    INPUT_MODE_BUTT
} input_mode_t;
```

[Chip Difference]

None

[Note]

- For Hi3559A V100: When the input interface is a parallel device (such as INPUT_MODE_CMOS, INPUT_MODE_BT601, INPUT_MODE_BT656, or INPUT_MODE_BT1120), only a number of COMS_MAX_DEV_NUM channels are supported. Channels 0 and 1 can be set as required, while channel 2 must be set.
- For Hi3559A V100: When channel 2 is set to INPUT_MODE_BT601 or INPUT_MODE_BT656, lane 8–lane 11 cannot be used as other interfaces. When channel 2 is set to INPUT_MODE_CMOS or INPUT_MODE_BT1120, lane 8–lane 15 cannot be used as other interfaces.
- For Hi3519A V100: When the input mode is INPUT_MODE_BT601 or INPUT_MODE_BT656 and MIPI device 0 or 1 is used, the pins of channel 0 are multiplexed with lane 8–lane 11, and the pins of channel 1 are multiplexed with lane 4–lane 7.
- For Hi3519A V100: When the input mode is INPUT_MODE_BT1120, only MIPI device 0 can be used, and the pins of channel 0 are multiplexed with lane 4–lane 11.

[See Also]

None

mipi_data_rate_t

[Description]

Defines the input rate of the MIPI RX and SLVS.

[Definition]

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

    MIPI_DATA_RATE_BUTT
} mipi_data_rate_t;
```

[Chip Difference]

Chip	MIPI_DATA_RATE_X2 Supported or Not
Hi3559A V100ES	Supported by MIPI0 and SLVS6
Hi3559A V100 Hi3519A V100	Supported by MIPI0 and SLVS6
Hi3516C V500 Hi3516E V200	Not supported

[Note]

None

[See Also]

None

img_rect_t

[Description]

Defines MIPI crop attributes.

[Definition]

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

[Member]

Member	Description
x	Horizontal coordinate of the crop start position
y	Vertical coordinate of the crop start position
width	Crop width (in pixels)
height	Crop height (in pixels)

[Chip Difference]

None

[Note]

The vertical coordinate of the SLVS-EC cropping must be greater than or equal to the number of valid lines of the sensor output.

[See Also]

None

slvs_lane_rate_t

[Description]

Defines the input rate of the SLVS lane.

[Definition]

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

[Chip Difference]

None

[Note]

None

[See Also]

None

slvs_err_check_mode_t

[Description]

Defines the SLVS error check mode.

[Syntax]

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

    SLVS_ERR_CHECK_MODE_BUTT
} slvs_err_check_mode_t;
```

[Member]

Member	Description
SLVS_ERR_CHECK_MODE_NONE	ECC and CRC disabled
SLVS_ERR_CHECK_MODE_CRC	CRC enabled
SLVS_ERR_CHECK_MODE_ECC_2BYTE	2-byte ECC enabled
SLVS_ERR_CHECK_MODE_ECC_4BYTE	4-byte ECC enabled

[Difference]

None

[Note]

The ECC and CRC functions of the SLVS are mutually exclusive and cannot be enabled at the same time.

[See Also]

slvs_dev_attr_t

data_type_t

[Description]

Defines the data type for transmission.

[Definition]

Hi3519A V100/Hi3516C V500:

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

Other chips:

```
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
```

[Member]

Member	Description
DATA_TYPE_RAW_8BIT	8-bit raw data
DATA_TYPE_RAW_10BIT	10-bit raw data
DATA_TYPE_RAW_12BIT	12-bit raw data
DATA_TYPE_RAW_14BIT	14-bit raw data
DATA_TYPE_RAW_16BIT	16-bit raw data
DATA_TYPE_YUV420_8BIT_NORMAL	8-bit YUV420 data, normal format
DATA_TYPE_YUV420_8BIT_LEGACY	8-bit YUV420 data, legacy format
DATA_TYPE_YUV422_8BIT	8-bit YUV422 data
DATA_TYPE_YUV422_PACKED	YUV422 data is packed and received in 16-bit raw format over the MIPI.

[Chip Difference]

Member	Data type supported
Hi3559A V100ES	• DATA_TYPE_RAW_8BIT • DATA_TYPE_RAW_10BIT • DATA_TYPE_RAW_12BIT • DATA_TYPE_RAW_14BIT • DATA_TYPE_RAW_16BIT
Hi3559A V100	All, except **DATA_TYPE_YUV422 PACKED**
Hi3519A V100/Hi3516C V500	All
Hi3516E V200	• DATA_TYPE_RAW_8BIT • DATA_TYPE_RAW_10BIT • DATA_TYPE_RAW_12BIT • DATA_TYPE_RAW_14BIT

[Note]

None

[See Also]

None

mipi_wdr_mode_t

[Description]

Defines the MIPI WDR mode.

[Definition]

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

[Member]

Member	Description
HI_MIPI_WDR_MODE_NONE	Linear mode
HI_MIPI_WDR_MODE_VC	The virtual channel in the packet header is used to differentiate long and short exposure frames.
HI_MIPI_WDR_MODE_DT	The self-defined data type in the packet header is used to differentiate long and short exposure frames.
HI_MIPI_WDR_MODE_DOL	DOL-mode WDR. A pixel after the packet header is used to differentiate long and short exposure frames.

[Chip Difference]

None

[Note]

None

[See Also]

None

mipi_dev_attr_t

[Description]

Defines the attributes of the MIPI device.

[Definition]

```
typedef struct
{
    data_type_t        input_data_type;
    mipi_wdr_mode_t      wdr_mode;
    short                lane_id[MIPI_LANE_NUM];
   union
   {
```



```
        short data_type[WDR_VC_NUM];
    };
}mipi_dev_attr_t;
```

[Member]

Member	Description
input_data_type	Data type for transmission
lane_id	Mapping between the TX end (sensor) and RX end (MIPI RX) lanes This member is set to **–1** for unused lanes.
wdr_mode	MIPI WDR mode
data_type	Data type corresponding to different exposure length data. When **wdr_mode** is **HI_MIPI_WDR_MODE_DT**, **data_type** needs to be configured.

[Chip Difference]

None

[Note]

- In 2L+2L mode of the Hi3516C V500, MIPI CH1 has only two valid lanes, which are connected to lane 1 and lane 3 of the PHY, respectively. Therefore, **lane_id** must be set to {0, 1, –1, –1} or {1, 3, –1, –1}. If they are cross-connected, **lane_id** must be set to {1, 0, –1, –1} or {3, 1, –1, –1}.
- Hi3516E V200 and Hi3518E V300 have only two lanes, namely, lane 0 and lane 2. Therefore, **lane_id** must be set to {0, 2, –1, –1}.
- Hi3516E V300 has four lanes. When a 2-lane sensor is connected, lane 0 and lane 1 are used and **lane_id** must be set to {0, 1, –1, –1}. When a 4-lane sensor is connected, **lane_id** must be set to {0, 1, 2, 3}.

[See Also]

- data_type_t
- mipi_wdr_mode_t
- HI_MIPI_SET_DEV_ATTR

wdr_mode_t

[Description]

Defines the LVDS WDR mode.

[Definition]

```
typedef enum
{
    HI_WDR_MODE_NONE    = 0x0,
    HI_WDR_MODE_2F      = 0x1,
```



```
    HI_WDR_MODE_3F      = 0x2,
    HI_WDR_MODE_4F      = 0x3,
    HI_WDR_MODE_DOL_2F  = 0x4,
    HI_WDR_MODE_DOL_3F  = 0x5,
    HI_WDR_MODE_DOL_4F  = 0x6,
    HI_WDR_MODE_BUTT
} wdr_mode_t;
```

[Member]

Member	Description
HI_WDR_MODE_NONE	Linear mode
HI_WDR_MODE_2F	2-to-1 WDR mode
HI_WDR_MODE_3F	3-to-1 WDR mode
HI_WDR_MODE_4F	4-to-1 WDR mode
HI_WDR_MODE_DOL_2F	DOL mode with 2-to-1 WDR
HI_WDR_MODE_DOL_3F	DOL mode with 3-to-1 WDR
HI_WDR_MODE_DOL_4F	DOL mode with 4-to-1 WDR

[Chip Difference]

Chip	WDR Mode
Hi3559A V100ES/ Hi3559A V100/ Hi3519A V100/ Hi3516C V500	All modes are supported.
Hi3516E V200/Hi3518E V300	The 2-to-1 WDR frame mode is supported.
Hi3516E V300	All modes are supported.

[Note]

- To enable the DOL WDR mode, select HI_WDR_MODE_DOL_2F, HI_WDR_MODE_DOL_3F, or HI_WDR_MODE_DOL_4F.
- To enable the built-in WDR mode or the multi-frame combination WDR mode, select HI_WDR_MODE_NONE.

[See Also]

None

lvds_sync_mode_t

[Description]

Defines the LVDS sync mode.

[Definition]

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

Table 1-6 LVDS sync mode

sync_mode	Sync Mode
LVDS_SYNC_MODE_SOF	SOF, EOF, SOL, and EOL See Figure 1-1.
LVDS_SYNC_MODE_SAV	invalid SAV, invalid EAV, valid SAV, and valid EAV See Figure 1-2.

[Chip Difference]

None

[Note]

None

[See Also]

None

lvds_bit_endian_t

[Description]

Defines the bit endian mode.

[Definition]

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

[Chip Difference]

None

[Note]

None

[See Also]

None

lvds_vsync_type_t

[Description]

Defines the LVDS vsync type.

[Definition]

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

[Member]

Member	Description
LVDS_VSYNC_NORMAL	Long and short exposure frames have independent SOF-EOF, independent SOL-EOL, invalid SAV-invalid EAV, or valid SAV-valid EAV.
LVDS_VSYNC_SHARE	Long and short exposure frames share one pair of SOF-EOF identifiers. The first several rows of short exposure frames are filled with fixed values.
LVDS_VSYNC_HCONNECT	Long and short exposure frames share one pair of SAV-EAV identifiers. There is a fixed period of blanking between the long and short exposure frames.

- LVDS_VSYNC_SHARE sync modes:

<table>
<tr><td>SOF</td><td rowspan="2">Long Exposure</td><td rowspan="2">EOL</td><td rowspan="2">Horizontal Blanking</td><td rowspan="2">SOL</td><td>Padding</td><td rowspan="2">EOL</td><td rowspan="2">Horizontal Blanking</td></tr>
<tr><td>SOL</td><td>Short Exposure</td></tr>
</table>

	Padding						
						EOF	
SOV	V.BLK	EOV	-	SOV	V.BLK	EOV	-

- LVDS_VSYNC_HCONNECT sync modes:

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

[Chip Difference]

None

[Note]

None

[See Also]

lvds_vsync_attr_t

lvds_vsync_attr_t

[Description]

Defines the LVDS vsync parameter.

[Definition]

```
typedef struct
{
    lvds_vsync_type_t sync_type;

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


[Chip Difference]

None

[Note]

When **sync_type** is set to LVDS_VSYNC_HCONNECT, **hblank1** and **hblank2** must be configured to indicate the length of the blanking region of Hconnect.

[See Also]

lvds_vsync_type_t

lvds_fid_type_t

[Description]

Defines the frame ID type.

[Definition]

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

[Member]

Member	Description
LVDS_FID_NONE	Frame ID is not used.
LVDS_FID_IN_SAV	Frame ID will be inserted at the fourth field of the SAV. For the sync codes of the four fields of the DOL, fid_type should be set to LVDS_FID_IN_SAV.
LVDS_FID_IN_DATA	As the frame information column, the frame ID will be inserted before the first pixel after the sync code. For the sync codes of the five fields of the DOL, fid_type should be set to LVDS_FID_IN_DATA

[Chip Difference]

None

[Note]

None

[See Also]

None

lvds_fid_attr_t

[Description]

Defines the frame ID configuration information.

[Definition]

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

[Member]

Member	Description
fid_type	Frame ID type in LVDS DOL mode
output_fil	The frame information line will be outputted right after V-Blanking in DOL mode. The frame ID is the first pixel value of the frame information line. The frame information line does not contain valid video data: • If output_fil is set to **HI_TRUE**, the frame information line will be outputted to the back-end device. • If output_fil is set to **HI_FALSE,** MIPI RX will abandon this frame information line.

[Chip Difference]

None

[Note]

None

[See Also]

lvds_fid_type_t

mipi_clk_mode_t

[Description]

Defines the clock sharing mode when the LVDS mode involves multiple PHYs. The clock of the PHY with the minimum number where the lane is located can be used. Alternatively, each PHY can use its own clock.

[Syntax]

```
typedef enum
{
    MIPI_CLK_MODE_SHARE  = 0x0,
    MIPI_CLK_MODE_SEPARATE = 0x1,
```



```
    MIPI_CLK_MODE_BUTT
} mipi_clk_mode_t;
```

[Member]

Member	Description
MIPI_CLK_MODE_SHARE	The PHYs share the clock of a PHY.
MIPI_CLK_MODE_SEPARATE	Each PHY uses its own clock.

[Difference]

None

[Note]

None

[See Also]

lvds_dev_attr_ex_t

lvds_lane_work_mode_t

[Description]

Defines the arrival mode of the synchronization code of the LVDS lane.

[Syntax]

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

[Member]

Member	Description
LVDS_LANE_WORK_MODE_SYNC	The synchronization code of the LVDS lane arrives synchronously.
LVDS_LANE_WORK_MODE_ASYNC	The synchronization code of the LVDS lane arrives asynchronously.

[Difference]

None

[Note]

None

[See Also]

lvds_lane_work_attr_t

lvds_lane_work_attr_t

[Description]

Defines the working attributes of the synchronization code of the LVDS lane.

[Syntax]

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

[Member]

Member	Description
lane_work_mode	Working mode of the lane synchronization code
sensor_valid_width	Number of pixels in a line of data packets (RAW H)

[Difference]

None

[Note]

None

[See Also]

lvds_dev_attr_ex_t

lvds_dev_attr_t

[Description]

Defines the device attributes supported by the LVDS, SubLVDS, and HiSPi interfaces.

[Definition]

```
typedef struct
{
    data_type_t       input_data_type;
    wdr_mode_t        wdr_mode;
    lvds_sync_mode_t   sync_mode;
    lvds_vsync_attr_t  vsync_attr;
    lvds_fid_attr_t    fid_attr;
    lvds_bit_endian_t  data_endian;
    lvds_bit_endian_t  sync_code_endian;
```



```
    short               lane_id[LVDS_LANE_NUM];
    unsigned short      sync_code[LVDS_LANE_NUM][WDR_VC_NUM][SYNC_CODE_NUM];
} lvds_dev_attr_t;
```

[Member]

Member	Description
input_data_type	Data type for transmission
wdr_mode	WDR mode
sync_mode	LVDS sync mode
vsync_ attr	vsync type. When **wdr_mod** is set to DOL mode and **sync_mode** is set to **LVDS_SYNC_MODE_SAV**, the vsync type must be configured.
fid_attr	Frame ID type. When **wdr_mod** is set to DOL mode and **sync_mode** is set to **LVDS_SYNC_MODE_SAV**, the frame ID type must be configured.
data_endian	Endian mode of the data
sync_code_endian	Endian mode of the sync codes
lane_id	Lane mapping between the TX end (sensor) and the RX end (MIPI RX) The ID of each unused lane is set to **–1**. See section 1.8.1 "How Do I Configure a Lane ID?" for details about the configuration of the lane ID.
sync_code	Each virtual channel has four sync codes. According to the sync mode, they stands for the sync codes of SOF, EOF, SOL, and EOL or sync codes of invalid SAV, invalid EAV, valid SAV, and valid EAV.

[Chip Difference]

None

[Note]

When this structure is used, the SLVS clock is shared, and the LVDS lane synchronization code arrives synchronously.

[See Also]

- 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

[Description]

Defines the advanced device attributes supported by the LVDS, sub-LVDS, and HiSPi interfaces.

[Syntax]

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

[Member]

Member	Description
clk_mode	Data type for transmission
lvds_attr	WDR mode
lane_work_attr	LVDS sync mode

[Difference]

None

[Note]

None

[See Also]

- 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

[Description]

Defines the attributes of the SLVS device.

[Definition]

- Other chips:

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

- Hi3559A V100:

```
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;
```

[Member]

Member	Description
input_data_type	Data type for transmission
wdr_mode	WDR mode
lane_rate	SLVS lane rate
sensor_valid_width	Number of pixels in a line of data packets (RAW H)
lane_id	Mapping between the TX-end (sensor) lane and the RX-end (SLVS) lane The lane_id of the unused lane is set to **–1**.
err_check_mode	SLVS CRC/ECC mode. The setting must match that for the sensor.

[Chip Difference]

Chip	slvs_dev_attr_t
Hi3559A V100ES/ Hi3559A V100/ Hi3519A V100	Supported
Hi3516C V500/ Hi3616E V200	SLVS is not supported.

[Note]

The SLVS device supports only the linear mode and the 2-to-1 WDR mode.

[See Also]

- data_type_t
- slvs_lane_rate_t
- HI_MIPI_SET_DEV_ATTR

phy_cmv_mode_t

[Description]

Defines the PHY common-mode voltage mode.

[Definition]

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

[Member]

Member	Description
PHY_CMV_GE1200MV	The PHY common-mode voltage is greater than or equal to 1200 mV.
PHY_CMV_LT1200MV	The PHY common-mode voltage is less than 1200 mV.

[Chip Difference]

None

[Note]

None

[See Also]

None

phy_cmv_t

[Description]

Defines the configuration information of the PHY common-mode voltage.

[Definition]

```
typedef struct
```



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

[Member]

Member	Description
devno	ID of the MIPI RX device
cmv_mode	Voltage mode of the PHY function

[Chip Difference]

None

[Note]

None

[See Also]

- phy_cmv_mode_t
- HI_MIPI_SET_PHY_CMVMODE

combo_dev_attr_t

[Description]

Defines the combo device attributes. The MIPI RX device is called the combo device because the MIPI RX can interwork with the CSI-2, LVDS, and HiSPi timings.

[Definition]

- Other chips:

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

- Hi3559A V100:

```
typedef struct
```

```
{
    combo_dev_t             devno;
    input_mode_t          input_mode;
    mipi_data_rate_t      data_rate;
    img_rect_t            img_rect;
    union
    {
        mipi_dev_attr_t     mipi_attr;
        lvds_dev_attr_t     lvds_attr;
        Defines the LVDS WDR mode.
     slvs_attr;
        lvds_dev_attr_ex_t  lvds_attr_ex;
    };
} combo_dev_attr_t;
```

[Member]

Member	Description
devno	Numbers of MIPI RX and SLVS devices
input_mode	Input interface type
data_rate	Interface transmission rate
img_rect	Crop region of the image
mipi_attr	If **input_mode** is set to **INPUT_MODE_MIPI**, **mipi_attr** must be configured.
lvds_attr	If **input_mode** is set to **INPUT_MODE_SUBLVDS**, **INPUT_MODE_LVDS**, or **INPUT_MODE_HISPI**, **lvds_attr** must be configured.
slvs_attr	If **input_mode** is set to **INPUT_MODE_SLVS**, **slvs_attr** must be configured.
lvds_attr_ex	If **input_mode** is set to **INPUT_MODE_LVDS_EX**, **lvds_attr_ex** must be configured.

[Chip Difference]

Chip	combo_dev_attr_t
Hi3559A V100ES/ Hi3559A V100/ Hi3519A V100/	Supported
Hi3516C V500/ Hi3516E V200	SLVS is not supported.

[Note]

None

[See Also]

None

HI_MIPI_TX_IOC_MAGIC

[Description]

Defines the magic number of **MIPI RX ioctl** in a command.

[Definition]

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

[Member]

None

[Chip Difference]

None

[Note]

None

[See Also]

None

LANE_MAX_NUM

[Description]

Defines the maximum number of lanes supported by the MIPI TX.

[Definition]

```
#define LANE_MAX_NUM          4
```

[Chip Difference]

None

[Note]

None

[See Also]

None

output_mode_t

[Description]

Defines the MIPI TX output mode.

[Definition]

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

[Chip Difference]

None

[Note]

None

[See Also]

None

video_mode_t

[Description]

Defines the MIPI TX video mode.

[Definition]

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

[Chip Difference]

None

[Note]

None

[See Also]

None

output_format_t

[Description]

Defines the MIPI TX output format.

[Definition]

```
typedef enum
```



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

[Chip Difference]

None

[Note]

None

[See Also]

None

sync_info_t

[Description]

Defines the synchronization information of an MIPI TX device.

[Definition]

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

[Member]

Member	Description
vid_pkt_size	Size of an RX packet
vid_hsa_pixels	Number of pixels in the line sync pulse region
vid_hbp_pixels	Number of pixels in the horizontal back porch (HBP)

Member	Description
vid_hline_pixels	Number of detected pixels in each line
vid_vsa_lines	Number of detected frame sync pulse lines
vid_vbp_lines	Number of lines in the frame sync pulse vertical back porch (VBP)
vid_vfp_lines	Number of lines in the frame sync pulse vertical front porch (VFP)
vid_active_lines	Number of lines in the active area
edpi_cmd_size	Number of bytes in the write memory command This parameter is invalid in Video mode, and is set to the **HACT** in Command mode.

[Chip Difference]

None

[Note]

None

[See Also]

None

combo_dev_cfg_t

[Description]

Defines the attributes of an MIPI TX device.

[Definition]

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

[Member]

Member	Description
devno	MIPI TX device ID
lane_id	Lane mapping between the TX end (sensor) and RX end (MIPI RX). The value of unused lanes is set to **–1**.
output_mode	Output mode of an MIPI TX device
video_mode	Video mode of an MIPI TX device
output_format	Output format of an MIPI TX device
sync_info	Synchronization information of an MIPI TX device
phy_data_rate	MIPI TX output rate. For details about the range, see the description of the lane rate range of each lane in MIPI TX high-speed mode in section "TX D-PHY" in the *Hi35xx xx Camera SoC Data Sheet*.
pixel_clk	Pixel clock, in kHz

[Chip Difference]

None

[Note]

None

[See Also]

None

cmd_info_t

[Description]

Defines the command data to be sent to an MIPI TX device.

[Definition]

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

[Member]

Member	Description
devno	MIPI TX device number

Member	Description
devno	MIPI TX device number
data_type	Command data type
cmd_size	Command data size Value range: (0, 200] For a single command, when **cmd** is set to **NULL**, the lower eight bits correspond to data 1, and the upper eight bits correspond to data 2.
cmd	Pointer to command data When the length of the command to be sent is within four bytes, this parameter can be set to NULL. When the length of the command to be sent exceeds four bytes, this parameter can be configured as required.

[Chip Difference]

None

[Note]

None

[See Also]

None

get_cmd_info_t

[Description]

Defines the command information sent to the MIPI TX device.

[Syntax]

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

[Member]

Member	Description
devno	MIPI TX device ID
data_type	Command data type

Member	Description
devno	MIPI TX device ID
data_param	Data parameter. The lower eight bits define the first parameter and the upper eight bits define the second parameter. Set a bit to **0** if the bit is not used.
get_data_size	Expected number of data bytes to be obtained Value range: (0, 200]
get_data	Pointer to the obtained data storage address. The address needs to be allocated by the user.

[Difference]

None

[Note]

None

[See Also]

None

1.6 Parameter of the MIPI TX Module

NOTICE

This section applies only to Hi3516C V500.

smooth

The **smooth** parameter implements smooth transition. Use this module as follows:

- On Linux, enable or disable the logic reset of the MIPI TX module when loading the MIPI TX driver.

  To disable the logic reset: **insmod hi_mipi_tx.ko smooth=1**

  To enable the logic reset: **insmod hi_mipi_tx.ko smooth=0**
- On Huawei LiteOS, enable or disable the logic reset of the MIPI TX module by transferring the **smooth** argument.

  To disable the logic reset: **int smooth=1; MIPI_TX_init(smooth)**

  To enable the logic reset: **int smooth=0; MIPI_TX_init(smooth)**

1.7 Proc Information

1.7.1 MIPI_RX Proc Information

When MIPI RX works normally, the width and height in the proc information are stable and matched with those of the sensor output timing. In addition, the error interrupt count of MIPI RX is 0. If the value is not 0, check whether the attributes of MIPI RX are configured correctly.

Hi3559A V100

[Debugging Information]

```
Module: [MIPI], Build Time: [May 23 2017, 10:04:19]
-----MIPI LANE DIVIDE MODE-------------------------------------------------
 MODE        LANE DIVIDE
 7          4+4+4+4
-----MIPI DEV ATTR---------------------------------------------------------
Devno  WorkMode  DataRate  DataType   WDRMode         LinkId    ImgX    ImgY
ImgW    ImgH
      0     MIPI       X1     RAW12     None           0, 1      0       0
3840    2160

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

-----MIPI LINK INFO--------------------------------------------------------
 LinkIdx LaneCount  LaneId    PhyData0    PhyData1  AlignedData0
AlignedData1   ValidLane
      0        2    0, 1        0x0        0x0         0x0         0x0
Invalid
      1        2    2, 3        0x0        0x0         0x0         0x0
Invalid
-----MIPI DETECT INFO------------------------------------------------------
 Devno VC   width  height
    0  0    3840    2160
    0  1       0       0
    0  2       0       0
    0  3       0       0

-----LVDS DETECT INFO-------------------------------------------------------
 Devno VC   width  height
    0  0    5480    3648
    0  1       0       0
    0  2       0       0
```



```
    0  3      0      0

-----LVDS LANE DETECT INFO----------------------------------------------
 Devno  Lane   width  height
    0     0     548    3689
    0     1     548    3689
    0     2     548    3689
    0     3     548    3689
    0     4     548    3689
    0     5     548    3689
    0     6     548    3689
    0     7     548    3689
    0     8     548    3689
    0    10     548    3689

-----FSM TIMEOUT AND ESCAPE INFO----------------------------------------
  phy clkTOutCnt d0TOutCnt   d1TOutCnt   d2TOutCnt   d3TOutCnt clkEscCnt
d0EscCnt  d1EscCnt  d2EscCnt  d3EscCnt
    0     0          0          0          0          0        0        0
0        0        0
    1     0          0          0          0          0        0        0
0        0        0
    2     0          0          0          0          0        0        0
0        0        0
    3     0          0          0          0          0        0        0
0        0        0
-----MIPI INT ERROR INFO-----------------------------------------------------
 Devno vc0CRC vc1CRC vc2CRC vc3CRC vc0OrderErr vc1OrderErr vc2OrderErr
vc3OrderErr vc0NMatCnt vc1NMatCnt vc2NMatCnt vc3NMatCnt
    0     0     0     0     0          0          0          0          0
0         0         0         0

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

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



```
 Devno FIFO_FullErr  Lane0Err  Lane1Err  Lane2Err  Lane3Err  Lane4Err
Lane5Err  Lane6Err  Lane7Err  Lane8Err  Lane9Err Lane10Err Lane11Err
Lane12Err Lane13Err Lane14Err Lane15Err
    0          0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0
0

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

-----SLVS LANE INFO----------------------------------------------------
   Devno   LaneCnt               LaneID
      6        8        0,  1,  2,  3,  4,  5,  6,  7
-----SLVS DATA INFO----------------------------------------------------
   Devno    LaneId     PhyData   AlignedData   ValidLane
      0        0      0x38b       0x72         0,
      0        1      0x2d4      0x2a4         1,
      0        2      0x12d      0x38a         2,
      0        3      0x23a      0x32a         3,
      0        4       0x4d       0xe4         4,
      0        5      0x1a3      0x18b         5,
      0        6      0x109      0x2e4         6,
      0        7      0x21f      0x1b4         7,
-----SLVS DETECT INFO--------------------------------------------------
   Devno  VC   width  height
      6   0    3840    2160
      6   1       0       0
      6   2       0       0
      6   3       0       0

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

-----SLVS PHY ERROR INFO-----------------------------------------------
  PhyIdx   LaneIdx  AFifoAlign   CodeErr   DispErr
      0        0         1        3        3
      0        1         1        3        3
      0        2         1        3        3
```

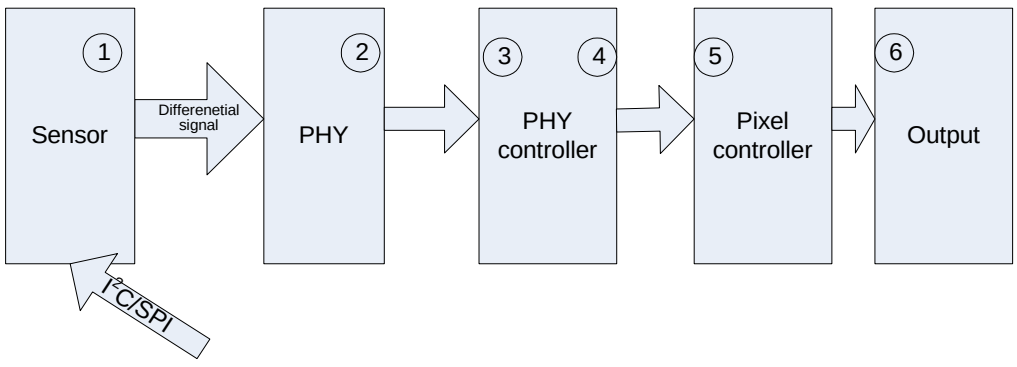


```
0       3       1       3       3
0       4       0       3       3
0       5       0       3       3
0       6       0       3       3
0       7       0       3       3
1       8       0       0       0
1       9       0       0       0
1       10      0       0       0
1       11      0       0       0
1       12      0       0       0
1       13      0       0       0
1       14      0       0       0
1       15      0       0       0
```

[Analysis]

- The MIPI RX receives differential data of the sensor by using the PHY. After detecting the sync header, the PHY controller aligns data in each lane.
- The pixel controller parses sync information and merges data in the lane into pixel data based on the bit width of raw data. It transmits pixel data to the downstream module in output mode.
- The clocks of the PHY, PHY controller, and pixel controller are provided by the pixel clock of the sensor. The clock of the output module is the associated clock, which is the same as the working clock of the downstream module. The crop function of the MIPI RX is implemented at the end of the pixel controller. Therefore, the required associated clock can be reduced after cropping.

Figure 1-3 MIPI data stream



[Parameter Description]

Parameter		Description
MIPI LANE DIVIDE MODE	MODE	Lane distribution mode of the MIPI RX
	LANE DIVIDE	Detailed lane distribution of the MIPI RX
MIPI DEV	Devno	MIPI device ID

Parameter		Description
ATTR	WorkMode	Working mode of the MIPI device • LVDS • MIPI • CMOS • SLVS
	DataRate	MIPI RX rate
	DataType	Data type • 8-bit raw data • 10-bit raw data • 12-bit raw data • 14-bit raw data • 16-bit raw data
	WDRMode	WDR mode • None: non-WDR mode • 2To1: 2-to-1WDR • 3To1: 3-to-1 WDR • 4To1: 4-to-1 WDR • DOL2To1: DOL 2-to-1WDR • DOL3To1: DOL 3-to-1 WDR • DOL4To1: DOL 4-to-1 WDR
	LinkId	IDs of links used by the device A physical link corresponds to four lanes.
	ImgX	Horizontal coordinate of the cropped image
	ImgY	Vertical coordinate of the cropped image
	ImgW	Width of the cropped image
	ImgH	Height of the cropped image
MIPI LANE INFO	Devno	MIPI device ID
	LaneCnt	Number of lanes
	LaneID	Lane ID
MIPI LINK INFO	LinkIdx	Link ID
	LaneCount	Number of lanes in the link
	LaneId	Lane ID
	PhyData0	Real-time data 0 received by the PHY
	PhyData1	Real-time data 1 received by the PHY

Parameter		Description
	AlignedData0	Real-time data 0 after the frame sync signal is detected
	AlignedData1	Real-time data 1 after the frame sync signal is detected
	ValidLane	Valid lane ID in the link In MIPI mode, the value is dynamically changed. Sometimes, the value is **Invalid**.
MIPI DETECT INFO (visible only in MIPI mode)	Devno	MIPI RX device ID
	VC	Virtual channel
	width	Total width of images detected by the MIPI controller
	height	Total height of images detected by the MIPI controller
LVDS DETECT INFO	Devno	MIPI RX device ID
	VC	Virtual channel
	width	Total width of images detected by the MIPI controller
	height	Total height of images detected by the MIPI controller
LVDS LANE DETECT INFO	Devno	MIPI RX device ID
	Lane	Lane ID
	width	Width detected by the lane
	height	Height detected by the lane
FSM TIMEOUT AND ESCAPE INFO (visible only in MIPI mode)	phy	PHY ID
	clkTOutCnt	Timeout when the clock lane is switched from the LP to HS
	d0TOutCnt	Timeout when data lane 0 is switched from the LP to HS
	d1TOutCnt	Timeout when data lane 1 is switched from the LP to HS
	d2TOutCnt	Timeout when data lane 2 is switched from the LP to HS
	d3TOutCnt	Timeout when data lane 3 is switched from the LP to HS
	clkEscCnt	Timeout when the clock lane is switched to escape mode

Parameter		Description
	d0EscCnt	Timeout when data lane 0 is switched to escape mode
	d1EscCnt	Timeout when data lane 1 is switched to escape mode
	d2EscCnt	Timeout when data lane 2 is switched to escape mode
	d3EscCnt	Timeout when data lane 3 is switched to escape mode
MIPI INT ERR INFO (Available only in MIPI mode)	Devno	Device ID of the MIPI RX
	vc0CRC	CRC error count of the VC0 channel data
	vc1CRC	CRC error count of the VC1 channel data
	vc2CRC	CRC error count of the VC2 channel data
	vc3CRC	CRC error count of the VC3 channel data
	vc0OrderErr	Error count of VC0 frame sequence
	vc1OrderErr	Error count of VC1 frame sequence
	vc2OrderErr	Error count of VC2 frame sequence
	vc3OrderErr	Error count of VC3 frame sequence
	vc0NMatCnt	Count of the short packet mismatches between the SOF and EOF of the VC 0 channel
	vc1NMatCnt	Count of the short packet mismatches between the SOF and EOF of the VC 1 channel
	vc2NMatCnt	Count of the short packet mismatches between the SOF and EOF of the VC 2 channel
	vc3NMatCnt	Count of the short packet mismatches between the SOF and EOF of the VC 3 channel
	HCntErr	EEC failure count for header errors
	vc0HECC	EEC success count for header errors of VC 0 channel
	vc1HECC	EEC success count for header errors of VC 1 channel
	vc2HECC	EEC success count for header errors of VC 2 channel

Parameter		Description
	vc3HECC	EEC success count for header errors of VC 3 channel
	vc0DtErr	Count of data types that are not supported by the VC0 channel
	vc1DtErr	Count of data types that are not supported by the VC1 channel
	vc2DtErr	Count of data types that are not supported by the VC2 channel
	vc3DtErr	Count of data types that are not supported by the VC3 channel
	CMD_FIFO_RERR	Count of raw FIFO interrupts for MIPI read commands
	DATA_FIFO_RERR	Count of raw FIFO interrupts for MIPI read data
	CMD_FIFO_WERR	Count of raw FIFO interrupts for MIPI write commands
	DATA_FIFO_WERR	Count of raw FIFO interrupts for MIPI write data
ALING ERROR INFO	Devno	MIPI device ID
	FIFO_FullErr	FIFO overflow
	Lane0Err	Lane 0 FIFO overflow
	Lane1Err	Lane 1 FIFO overflow
	Lane2Err	Lane 2 FIFO overflow
	Lane3Err	Lane 3 FIFO overflow
	Lane4Err	Lane 4 FIFO overflow
	Lane5Err	Lane 5 FIFO overflow
	Lane6Err	Lane 6 FIFO overflow
	Lane7Err	Lane 7 FIFO overflow
	Lane8Err	Lane 8 FIFO overflow
	Lane9Err	Lane 9 FIFO overflow
	Lane10Err	Lane 10 FIFO overflow
	Lane11Err	Lane 11 FIFO overflow
	Lane12Err	Lane 12 FIFO overflow
	Lane13Err	Lane 13 FIFO overflow

Parameter		Description
	Lane14Err	Lane 14 FIFO overflow
	Lane15Err	Lane 15 FIFO overflow
SLVS DEV ATTR	Devno	SLVS device number
	WorkMode	SLVS device working mode • LVDS • MIPI • CMOS • SLVS
	DataRate	SLVS data rate
	DataType	Data type • 8-bit raw data • 10-bit raw data • 12-bit raw data • 14-bit raw data • 16-bit raw
	WDRMode	WDR mode • None: non-WDR mode • 2To1: 2-to-1WDR • 3To1: 3-to-1 WDR • 4To1: 4-to-1 WDR • DOL2To1: DOL 2-to-1WDR • DOL3To1: DOL 3-to-1 WDR • DOL4To1: DOL 4-to-1 WDR
	LinkId	IDs of links used by the device. A physical link corresponds to four lanes.
	ImgX	Horizontal coordinate of the cropped image
	ImgY	Vertical coordinate of the cropped image
	ImgW	Width of the cropped image
	ImgH	Height of the cropped image
	LaneRate	Lane rate
	ValidW	Actual valid width of the sensor
	ErrMode	SLVS CRC/ECC mode
SLVS LANE INFO	Devno	SLVS device number
	LaneCnt	Number of lanes
	LaneID	Lane ID

Parameter		Description
SLVS DATA INFO	Devno	SLVS device ID
	LaneID	Lane ID
	PhyData	Real-time data received by the lane corresponding to the PHY
	AlignedData	Real-time data received by the lane corresponding to the PHY after the synchronization code is detected
	ValidLane	Valid lane ID in the PHY In LVDS mode, the value is dynamically changed. Sometimes, the value is **Invalid**.
SLVS DETECT INFO	Devno	SLVS device number
	VC	Virtual Channel
	width	Total width of images detected by the SLVS controller
	height	Total height of images detected by the SLVS controller
SLVS DEV ERROR INFO	Devno	SLVS device number
	HeaderCRC	Number of data header CRC errors
	PayloadCRC	Number of data CRC errors
	EccErr	Number of ECC errors
	DataFifoWrite	Number of data FIFO write errors
	DataFifoRead	Number of data FIFO read errors
	CmdFifoFull	Counts of command FIFO full
	SkewErr	Number of skew errors
SLVS PHY ERROR INFO	PhyIdx	PHY ID
	LaneIdx	Lane ID
	AFifoAlign	Number of FIFO alignment errors
	CodeErr	Number of 10B8B coding errors
	DispErr	Number of disparity errors

Hi3519A V100

[Debugging Information]

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



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

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

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

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

-----LVDS LANE DETECT INFO-------------------------------------------------
 Devno  Lane   width  height
    0     2     960    2179
    0     4     960    2179
    0     5     960    2179
    0     7     960    2179
-----PHY CIL ERR INT INFO--------------------------------------------------
   PhyId  Clk2TmOut  ClkTmOut  Lane0TmOut  Lane1TmOut  Lane2TmOut
Lane3TmOut  Clk2Esc  ClkEsc  Lane0Esc  Lane1Esc  Lane2Esc  Lane3Esc
```



```
      0         0         0         0         0         0         0         0
0         0         0         0         0
      1         0         0         0         0         0         0         0
0         0         0         0         0
      2         0         0         0         0         0         0         0
0         0         0         0         0
-----MIPI ERROR INT INFO 1-------------------------------------------------
   Devno  Ecc2  Vc0CRC  Vc1CRC  Vc2CRC  Vc3CRC  Vc0EccCorrct  Vc1EccCorrct
Vc2EccCorrct  Vc3EccCorrct
      0    0      0      0      0      0           0           0
0           0

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

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

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

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

-----LVDS ERROR INT INFO 3-------------------------------------------------
   Devno  Lane0Err  Lane1Err  Lane2Err  Lane3Err  Lane4Err  Lane5Err
Lane6Err  Lane7Err  Lane8Err  Lane9Err Lane10Err Lane11Err
      0        0        0        0        0        0        0        0
0        0        0        0        0
-----ALIGN ERROR INT INFO---------------------------------------
   Devno  FIFO_FullErr  Lane0Err  Lane1Err  Lane2Err  Lane3Err  Lane4Err
Lane5Err  Lane6Err  Lane7Err  Lane8Err  Lane9Err Lane10Err Lane11Err
      0            0        0        0        0        0        0        0
```



```
0        0        0        0        0        0
-----SLVS DEV ATTR-----------------------------------------------------
   Devno  WorkMode  DataRate  DataType   WDRMode    ImgX    ImgY    ImgW
ImgH  LaneRate    ValidW
      0     SLVS       X2     RAW12      None      48      24    3840    2160
HIGHT     3936

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

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

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

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

-----SLVS PHY ERROR INFO-----------------------------------------------
  PhyIdx   LaneIdx  AFifoAlign   CodeErr   DispErr
      0        0         0        0        0
      0        1         0        0        0
      0        2         0        0        0
      0        3         0        0        0
      0        4         0        0        0
      0        5         0        0        0
      0        6         0        0        0
```

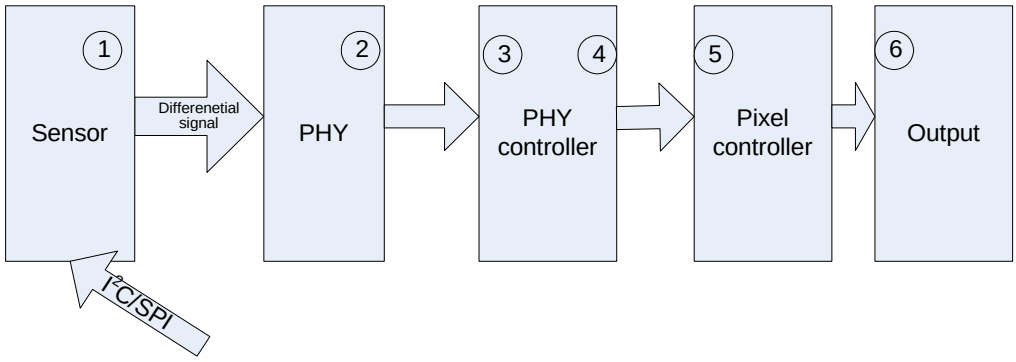


```
0       7       0       0       0
```

[Analysis]

- The MIPI RX receives differential data of the sensor by using the PHY. After detecting the sync header, the PHY controller aligns data in each lane.
- The pixel controller parses sync information and merges data in the lane into pixel data based on the bit width of raw data. It transmits pixel data to the downstream module in output mode.
- The clocks of the PHY, PHY controller, and pixel controller are provided by the pixel clock of the sensor. The clock of the output module is the associated clock, which is the same as the working clock of the downstream module. The crop function of the MIPI RX is implemented at the end of the pixel controller. Therefore, the required associated clock can be reduced after cropping.

Figure 1-4 MIPI data stream



[Parameter Description]

Parameter		Description
MIPI LANE DIVIDE MODE	MODE	Lane distribution mode of the MIPI RX
	LANE DIVIDE	Detailed lane distribution of the MIPI RX
MIPI DEV ATTR	Devno	MIPI device ID
	WorkMode	Working mode of the MIPI device • LVDS • MIPI • CMOS • SLVS
	DataRate	MIPI RX rate

Parameter		Description
	DataType	Data type • 8-bit raw data • 10-bit raw data • 12-bit raw data • 14-bit raw data • 16-bit raw data
	WDRMode	WDR mode • None: non-WDR mode • 2To1: 2-to-1WDR • 3To1: 3-to-1 WDR • 4To1: 4-to-1 WDR • DOL2To1: DOL 2-to-1WDR • DOL3To1: DOL 3-to-1 WDR • DOL4To1: DOL 4-to-1 WDR
	ImgX	Horizontal coordinate of the cropped image
	ImgY	Vertical coordinate of the cropped image
	ImgW	Width of the cropped image
	ImgH	Height of the cropped image
MIPI LANE INFO	Devno	MIPI device ID
	LaneCnt	Number of lanes
	LaneID	Lane ID
MIPI PHY DATA INFO	PhyId	PHY ID
	LaneId	Lane ID
	PhyData	Real-time data received by the four lanes corresponding to the PHY
	MipiData	Real-time data received after the four lanes corresponding to the PHY detect the MIPI frame synchronization signal
	LvdsData	Real-time data received after the four lanes corresponding to the PHY detect the LVDS frame synchronization signal
MIPI DETECT INFO (available only in	Devno	MIPI RX device ID
	VC	Virtual channel
	width	Total width of images detected by the MIPI controller

Parameter		Description
MIPI mode)	height	Total height of images detected by the MIPI controller
LVDS DETECT INFO (available only in LVDS mode)	Devno	MIPI RX device ID
	VC	Virtual channel
	width	Total width of images detected by the MIPI controller
	height	Total height of images detected by the MIPI controller
LVDS LANE DETECT INFO (available only in LVDS mode)	Devno	MIPI RX device ID
	Lane	Lane ID
	width	Image width detected on the lane
	height	Image height detected on the lane
PHY CIL ERR INT INFO	phy	PHY ID
	Clk2TmOut	Timeout when the lane of clock 2 is switched from LP mode to HS mode
	ClkTmOut	Timeout when the lane of clock 1 is switched from LP mode to HS mode
	Lane0TmOut	Timeout when data lane 0 is switched from LP mode to HS mode
	Lane1TmOut	Timeout when data lane 1 is switched from LP mode to HS mode
	Lane2TmOut	Timeout when data lane 2 is switched from LP mode to HS mode
	Lane3TmOut	Timeout when data lane 3 is switched from LP mode to HS mode
	Clk2Esc	Timeout when the lane of clock 2 is switched to escape mode
	ClkEsc	Timeout when the lane of clock 1 is switched to escape mode
	Lane0Esc	Timeout when data lane 0 is switched to escape mode
	Lane1Esc	Timeout when data lane 1 is switched to escape mode
	Lane2Esc	Timeout when data lane 2 is switched to escape mode

Parameter		Description
	Lane3Esc	Timeout when data lane 3 is switched to escape mode
MIPI ERROR INT INFO 1 (available only in MIPI mode)	Devno	MIPI RX device ID
	Ecc2	Whether the header has at least two uncorrectable ECC errors
	Vc0CRC	Count of CRC errors for VC0 data
	Vc1CRC	Count of CRC errors for VC1 data
	Vc2CRC	Count of CRC errors for VC2 data
	Vc3CRC	Count of CRC errors for VC3 data
	Vc0EccCorrct	Count of the corrected ECC errors for the VC0 header
	Vc1EccCorrct	Count of the corrected ECC errors for the VC1 header
	Vc2EccCorrct	Count of the corrected ECC errors for the VC2 header
	Vc3EccCorrct	Count of the corrected ECC errors for the VC3 header
MIPI ERROR INT INFO 2 (available only in MIPI mode)	Devno	MIPI RX device ID
	Vc0Dt	Count of data types unsupported by VC0
	Vc1Dt	Count of data types unsupported by VC1
	Vc2Dt	Count of data types unsupported by VC2
	Vc3Dt	Count of data types unsupported by VC3
	Vc0FrmCrc	Count of frame data errors for VC0
	Vc1FrmCrc	Count of frame data errors for VC1
	Vc2FrmCrc	Count of frame data errors for VC2
	Vc3FrmCrc	Count of frame data errors for VC3
MIPI ERROR INT INFO 3 (available only in MIPI mode)	Devno	MIPI RX device ID
	Vc0FrmSeq	Count of frame sequence errors for VC0
	Vc1FrmSeq	Count of frame sequence errors for VC1
	Vc2FrmSeq	Count of frame sequence errors for VC2
	Vc3FrmSeq	Count of frame sequence errors for VC3
	Vc0BndryMt	Count of short packets with SOF and EOF mismatching for VC0

Parameter		Description
	Vc1BndryMt	Count of short packets with SOF and EOF mismatching for VC1
	Vc2BndryMt	Count of short packets with SOF and EOF mismatching for VC2
	Vc3BndryMt	Count of short packets with SOF and EOF mismatching for VC3
MIPI ERROR INT INFO 4 (available only in MIPI mode)	Devno	MIPI RX device ID
	DataFifoRdErr	Count of MIPI CTRL read data FIFO interrupts
	CmdFifoRdErr	Count of MIPI CTRL read command FIFO interrupts
	Vsync	Count of MIPI CTR VSync interrupts
	CmdFifoWrErr	Count of MIPI CTRL write command FIFO interrupts
	DataFifoWrErr	Count of MIPI CTRL write data FIFO interrupts
LVDS ERROR INT INFO 1 (available only in LVDS mode)	Devno	MIPI device ID
	Vsync	Count of VSync interrupts for LVDS
	CmdRdErr	Count of CMD_FIFO read error interrupts for LVDS
	CmdWrErr	Count of CMD_FIFO write error interrupts for LVDS
	PopErr	Count of line_buf read error interrupts for LVDS
	StatErr	Count of lane synchronization error interrupts for LVDS
LVDS ERROR INT INFO 2 (available only in LVDS mode)	Devno	MIPI device ID
	Link0WrErr	Count of FIFO read error interrupts for link 0
	Link1WrErr	Count of FIFO read error interrupts for link 1
	Link2WrErr	Count of FIFO read error interrupts for link 2
	Link0RdErr	Count of FIFO write error interrupts for link 0
	Link1RdErr	Count of FIFO write error interrupts for link 1
	Link2RdErr	Count of FIFO write error interrupts for link 2
LVDS ERROR INT INFO 3 (available only in LVDS	Devno	MIPI device ID
	Lane0Err	Count of synchronization error interrupts for lane 0
	Lane1Err	Count of synchronization error interrupts for lane 1
	Lane2Err	Count of synchronization error interrupts for lane 2

Parameter		Description
mode)	Lane3Err	Count of synchronization error interrupts for lane 3
	Lane4Err	Count of synchronization error interrupts for lane 4
	Lane5Err	Count of synchronization error interrupts for lane 5
	Lane6Err	Count of synchronization error interrupts for lane 6
	Lane7Err	Count of synchronization error interrupts for lane 7
	Lane8Err	Count of synchronization error interrupts for lane 8
	Lane9Err	Count of synchronization error interrupts for lane 9
	Lane10Err	Count of synchronization error interrupts for lane 10
	Lane11Err	Count of synchronization error interrupts for lane 11
ALING ERROR INFO	Devno	MIPI device ID
	FIFO_FullErr	FIFO overflow
	Lane0Err	Lane 0 FIFO overflow
	Lane1Err	Lane 1 FIFO overflow
	Lane2Err	Lane 2 FIFO overflow
	Lane3Err	Lane 3 FIFO overflow
	Lane4Err	Lane 4 FIFO overflow
	Lane5Err	Lane 5 FIFO overflow
	Lane6Err	Lane 6 FIFO overflow
	Lane7Err	Lane 7 FIFO overflow
	Lane8Err	Lane 8 FIFO overflow
	Lane9Err	Lane 9 FIFO overflow
	Lane10Err	Lane 10 FIFO overflow
	Lane11Err	Lane 11 FIFO overflow
SLVS DEV ATTR	Devno	SLVS device ID
	WorkMode	Working mode of the SLVS device, such as **LVDS**, **MIPI**, **CMOS**, or **SLVS**
	DataRate	SLVS data rate
	DataType	Data type,which can be **RAW8 bit**, **RAW10 bit**, **RAW12 bit**, **RAW14 bit**, or **RAW16 bit**

Parameter		Description
	WDRMode	WDR mode • None: non-WDR mode • 2To1: 2-to-1 WDR • 3To1: 3-to-1 WDR • 4To1: 4-to-1 WDR • DOL2To1: DOL 2-to-1 WDR • DOL3To1: DOL 3-to-1 WDR • DOL4To1: DOL 4-to-1 WDR
	ImgX	X coordinate of the start position of the cropped image
	ImgY	Y coordinate of the start position of the cropped image
	ImgW	Width of the cropped image
	ImgH	Height of the cropped image
	LaneRate	Lane rate
	ValidW	Actual valid width of the sensor
SLVS LANE INFO	Devno	SLVS device ID
	LaneCnt	Number of lanes
	LaneID	Lane ID
SLVS PHY DATA INFO	PhyId	PHY ID
	LaneID	Lane ID
	PhyData	Real-time data received by the lane corresponding to the PHY
	AlignedData	Real-time data received after the lane corresponding to the PHY detects the synchronization code
SLVS DETECT INFO	Devno	SLVS device ID
	VC	Virtual channel
	width	Total width of images detected by the SLVS controller
	height	Total height of images detected by the SLVS controller
SLVS DEV ERROR INFO	Devno	SLVS device ID
	HeaderCRC	Count of data header CRC errors
	PayloadCRC	Count of data CRC errors
	EccErr	Count of ECC errors

Parameter		Description
	DataFifoWrite	Count of data FIFO write errors
	DataFifoRead	Count of data FIFO read errors
	CmdFifoFull	Count of the full status of the command FIFO
	SkewErr	Count of skew errors
SLVS PHY ERROR INFO	PhyIdx	PHY ID
	LaneIdx	Lane ID
	AFifoAlign	Count of FIFO alignment errors
	CodeErr	Count of 10B8B encoding errors
	DispErr	Count of disparity errors

Hi3516C V500

[Debugging Information]

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

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

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

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

-----MIPI DETECT INFO----------------------------------------------------
 Devno VC   width  height
    0  0    2592    1760
    0  1       0       0
    0  2       0       0
```



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

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

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

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

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

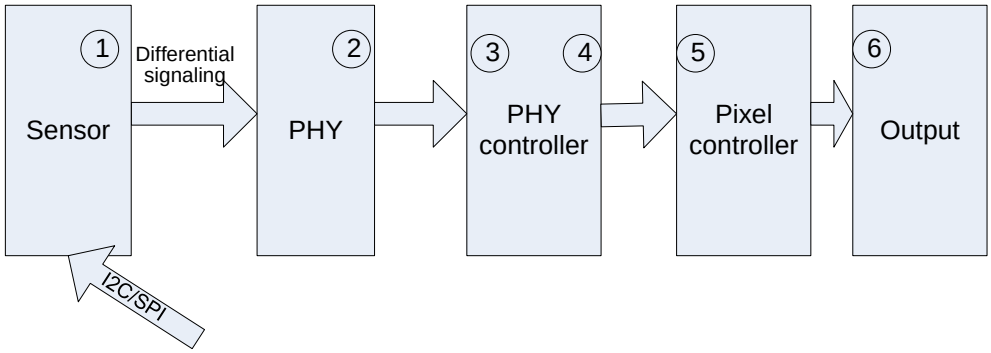


```
0          0          0          0          0          0
```

[Analysis]

- The MIPI RX receives differential data of the sensor by using the PHY. After detecting the sync header, the PHY controller aligns data in each lane.
- The pixel controller parses sync information and merges data in the lane into pixel data based on the bit width of raw data. It transmits pixel data to the downstream module in output mode.
- The PHY, PHY controller, and pixel controller use the pixel clock of the sensor. The clock of the output module is the associated clock, which is the same as the working clock of the downstream module. The crop function of the MIPI RX is implemented at the end of the pixel controller. Therefore, the required associated clock can be reduced after cropping.

Figure 1-5 MIPI data stream



[Parameter Description]

Parameter		Description
MIPI LANE DIVIDE MODE	MODE	Lane distribution mode of the MIPI RX
	LANE DIVIDE	Detailed lane distribution of the MIPI RX
MIPI DEV ATTR	Devno	MIPI device ID
	WorkMode	Working mode of the MIPI device • LVDS • MIPI • CMOS
	DataRate	MIPI RX rate

Parameter		Description
	DataType	Data type • 8-bit raw data • 10-bit raw data • 12-bit raw data • 14-bit raw data • 16-bit raw data
	WDRMode	WDR mode • None: non-WDR mode • 2To1: 2-to-1WDR • 3To1: 3-to-1 WDR • 4To1: 4-to-1 WDR • DOL2To1: DOL 2-to-1WDR • DOL3To1: DOL 3-to-1 WDR • DOL4To1: DOL 4-to-1 WDR
	ImgX	Horizontal coordinate of the cropped image
	ImgY	Vertical coordinate of the cropped image
	ImgW	Width of the cropped image
	ImgH	Height of the cropped image
MIPI LANE INFO	Devno	MIPI device ID
	LaneID	Lane ID
MIPI PHY DATA INFO	PhyId	PHY ID
	LaneId	Lane ID
	PhyData	Real-time data received by the four lanes corresponding to the PHY
	MipiData	Real-time data received after the four lanes corresponding to the PHY detect the MIPI frame synchronization signal
	LvdsData	Real-time data received after the four lanes corresponding to the PHY detect the LVDS frame synchronization signal
MIPI DETECT INFO (available only in MIPI mode)	Devno	MIPI RX device ID
	VC	Virtual channel
	width	Total width of images detected by the MIPI controller
	height	Total height of images detected by the MIPI controller

Parameter		Description
LVDS DETECT INFO (available only in LVDS mode)	Devno	MIPI RX device ID
	VC	Virtual channel
	width	Total width of images detected by the MIPI controller
	height	Total height of images detected by the MIPI controller
LVDS LANE DETECT INFO (available only in LVDS mode)	Devno	MIPI RX device ID
	Lane	Lane ID
	width	Image width detected on the lane
	height	Image height detected on the lane
PHY CIL ERR INT INFO	PhyId	PHY ID
	Clk2TmOut	Timeout when the lane of clock 2 is switched from LP mode to HS mode
	ClkTmOut	Timeout when the lane of clock 1 is switched from LP mode to HS mode
	Lane0TmOut	Timeout when data lane 0 is switched from LP mode to HS mode
	Lane1TmOut	Timeout when data lane 1 is switched from LP mode to HS mode
	Lane2TmOut	Timeout when data lane 2 is switched from LP mode to HS mode
	Lane3TmOut	Timeout when data lane 3 is switched from LP mode to HS mode
	Clk2Esc	Timeout when the lane of clock 2 is switched to escape mode
	ClkEsc	Timeout when the lane of clock 1 is switched to escape mode
	Lane0Esc	Timeout when data lane 0 is switched to escape mode
	Lane1Esc	Timeout when data lane 1 is switched to escape mode
	Lane2Esc	Timeout when data lane 2 is switched to escape mode
	Lane3Esc	Timeout when data lane 3 is switched to escape mode
MIPI	Devno	MIPI RX device ID

Parameter		Description
ERROR INT INFO 1 (available only in MIPI mode)	Ecc2	Whether the header has at least two uncorrectable ECC errors
	Vc0CRC	Count of CRC errors for VC0 data
	Vc1CRC	Count of CRC errors for VC1 data
	Vc2CRC	Count of CRC errors for VC2 data
	Vc3CRC	Count of CRC errors for VC3 data
	Vc0EccCorrct	Count of the corrected ECC errors for the VC0 header
	Vc1EccCorrct	Count of the corrected ECC errors for the VC1 header
	Vc2EccCorrct	Count of the corrected ECC errors for the VC2 header
	Vc3EccCorrct	Count of the corrected ECC errors for the VC3 header
MIPI ERROR INT INFO 2 (available only in MIPI mode)	Devno	MIPI RX device ID
	Vc0Dt	Count of data types unsupported by VC0
	Vc1Dt	Count of data types unsupported by VC1
	Vc2Dt	Count of data types unsupported by VC2
	Vc3Dt	Count of data types unsupported by VC3
	Vc0FrmCrc	Count of frame data errors for VC0
	Vc1FrmCrc	Count of frame data errors for VC1
	Vc2FrmCrc	Count of frame data errors for VC2
	Vc3FrmCrc	Count of frame data errors for VC3
MIPI ERROR INT INFO 3 (available only in MIPI mode)	Devno	MIPI RX device ID
	Vc0FrmSeq	Count of frame sequence errors for VC0
	Vc1FrmSeq	Count of frame sequence errors for VC1
	Vc2FrmSeq	Count of frame sequence errors for VC2
	Vc3FrmSeq	Count of frame sequence errors for VC3
	Vc0BndryMt	Count of short packets with SOF and EOF mismatching for VC0
	Vc1BndryMt	Count of short packets with SOF and EOF mismatching for VC1
	Vc2BndryMt	Count of short packets with SOF and EOF mismatching for VC2

Parameter		Description
	Vc3BndryMt	Count of short packets with SOF and EOF mismatching for VC3
MIPI ERROR INT INFO 4 (available only in MIPI mode)	Devno	MIPI RX device ID
	DataFifoRdErr	Count of MIPI CTRL read data FIFO interrupts
	CmdFifoRdErr	Count of MIPI CTRL read command FIFO interrupts
	CmdFifoWrErr	Count of MIPI CTRL write command FIFO interrupts
	DataFifoWrErr	Count of MIPI CTRL write data FIFO interrupts
LVDS ERROR INT INFO (available only in LVDS mode)	Devno	MIPI device ID
	CmdRdErr	Count of CMD_FIFO read error interrupts for LVDS
	CmdWrErr	Count of CMD_FIFO write error interrupts for LVDS
	PopErr	Count of line_buf read error interrupts for LVDS
	StatErr	Count of lane synchronization error interrupts for LVDS
	Link0WrErr	Count of FIFO read error interrupts for link 0
	Link0RdErr	Count of FIFO write error interrupts for link 0
ALING ERROR INFO	Devno	MIPI device ID
	FIFO_FullErr	FIFO overflow
	Lane0Err	Lane 0 FIFO overflow
	Lane1Err	Lane 1 FIFO overflow
	Lane2Err	Lane 2 FIFO overflow
	Lane3Err	Lane 3 FIFO overflow

Hi3516E V200

[Debugging Information]

```
Module: [MIPI_RX], Build Time[Nov  24 2018, 14:50:01]
-----MIPI LANE DIVIDE MODE-----------------------------------------------
  MODE        LANE DIVIDE
     0                2


-----MIPI DEV ATTR-------------------------------------------------------
   Devno  WorkMode  DataRate          DataType   WDRMode    ImgX    ImgY
```



```
ImgW    ImgH
      0     MIPI       X1             RAW12      None      0     204    2304
1296

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

-----MIPI PHY DATA INFO-------------------------------------------------
   PhyId        LaneId          PhyData              MipiData
LvdsData
      0      0, 1   0x00,0x00    0x36,0x2b   0x9b,0x1ad275
-----MIPI DETECT INFO--------------------------------------------------
 Devno VC   width  height
    0  0    2304    1296
    0  1       0       0
-----LVDS DETECT INFO--------------------------------------------------
 Devno VC   width  height
    0  0    2304    1296
    0  1       0       0
-----LVDS LANE DETECT INFO---------------------------------------------
-----
 Devno  Lane   width  height
    0     0     0       0
    0     1     0       0
-----PHY CIL ERR INT INFO-----------------------------------------------
   PhyId  Clk2TmOut  ClkTmOut  Lane0TmOut  Lane1TmOut  Lane2TmOut
Lane3TmOut  Clk2Esc  ClkEsc  Lane0Esc  Lane1Esc  Lane2Esc  Lane3Esc
      0         0        0          0          0          0          0       0
0        0        0        0        0
-----MIPI ERROR INT INFO 1---------------------------------------------
   Devno  Ecc2  Vc0CRC  Vc1CRC  Vc2CRC  Vc3CRC  Vc0EccCorrct  Vc1EccCorrct
Vc2EccCorrct  Vc3EccCorrct
      0     0      0       0       0       0            0            0
0           0

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

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

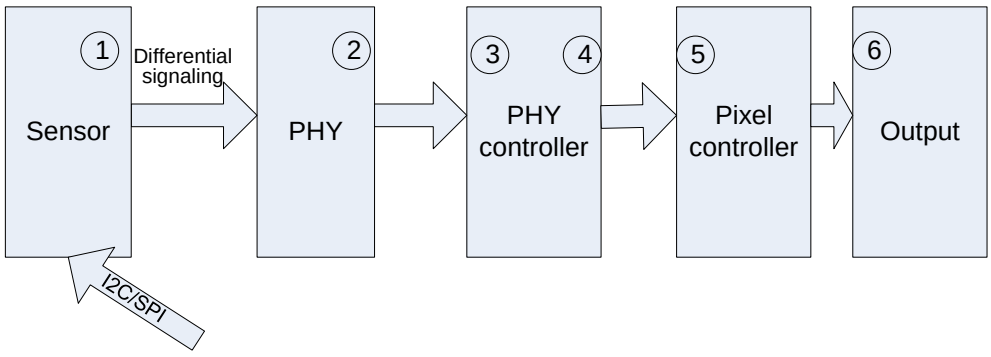


```
      0         0         0         0         0         0         0
0         0

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

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

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

[Analysis]

- The MIPI RX receives differential data of the sensor by using the PHY. After detecting the sync header, the PHY controller aligns data in each lane.
- The pixel controller parses sync information and merges data in the lane into pixel data based on the bit width of raw data. It transmits pixel data to the downstream module in output mode.
- The PHY, PHY controller, and pixel controller use the pixel clock of the sensor. The clock of the output module is the associated clock, which is the same as the working clock of the downstream module. The crop function of the MIPI RX is implemented at the end of the pixel controller. Therefore, the required associated clock can be reduced after cropping.

Figure 1-6 MIPI data stream



[Parameter Description]

Parameter		Description
MIPI LANE DIVIDE MODE	MODE	Lane distribution mode of the MIPI RX
	LANE DIVIDE	Detailed lane distribution of the MIPI RX
MIPI DEV ATTR	Devno	MIPI device ID
	WorkMode	Working mode of the MIPI device • LVDS • MIPI • CMOS
	DataRate	MIPI RX rate
	DataType	Data type • 8-bit raw data • 10-bit raw data • 12-bit raw data • 14-bit raw data • 16-bit raw data
	WDRMode	WDR mode • None: non-WDR mode • 2To1: 2-to-1WDR • DOL2To1: DOL 2-to-1WDR
	ImgX	Horizontal coordinate of the cropped image
	ImgY	Vertical coordinate of the cropped image
	ImgW	Width of the cropped image
	ImgH	Height of the cropped image
MIPI LANE INFO	Devno	MIPI device ID
	LaneID	Lane ID
MIPI PHY DATA INFO	PhyId	PHY ID
	LaneId	Lane ID
	PhyData	Real-time data received by the two lanes corresponding to the PHY
	MipiData	Real-time data received after the two lanes corresponding to the PHY detect the MIPI frame synchronization signal
	LvdsData	Real-time data received after the two lanes corresponding to the PHY detect the LVDS frame synchronization signal

Parameter		Description
MIPI DETECT INFO (available only in MIPI mode)	Devno	MIPI RX device ID
	VC	Virtual channel
	width	Total width of images detected by the MIPI controller
	height	Total height of images detected by the MIPI controller
LVDS DETECT INFO (available only in LVDS mode)	Devno	MIPI RX device ID
	VC	Virtual channel
	width	Total width of images detected by the MIPI controller
	height	Total height of images detected by the MIPI controller
LVDS LANE DETECT INFO (available only in LVDS mode)	Devno	MIPI RX device ID
	Lane	Lane ID
	width	Image width detected on the lane
	height	Image height detected on the lane
PHY CIL ERR INT INFO	PhyId	PHY ID
	Clk2TmOut	Timeout when the lane of clock 2 is switched from LP mode to HS mode
	ClkTmOut	Timeout when the lane of clock 1 is switched from LP mode to HS mode
	Lane0TmOut	Timeout when data lane 0 is switched from LP mode to HS mode
	Lane1TmOut	Timeout when data lane 1 is switched from LP mode to HS mode
	Clk2Esc	Timeout when the lane of clock 2 is switched to escape mode
	ClkEsc	Timeout when the lane of clock 1 is switched to escape mode
	Lane0Esc	Timeout when data lane 0 is switched to escape mode
	Lane1Esc	Timeout when data lane 1 is switched to escape mode
MIPI	Devno	MIPI RX device ID

Parameter		Description
ERROR INT INFO 1 (available only in MIPI mode)	Ecc2	Whether the header has at least two uncorrectable ECC errors
	Vc0CRC	Count of CRC errors for VC0 data
	Vc1CRC	Count of CRC errors for VC1 data
	Vc0EccCorrct	Count of the corrected ECC errors for the VC0 header
	Vc1EccCorrct	Count of the corrected ECC errors for the VC1 header
MIPI ERROR INT INFO 2 (available only in MIPI mode)	Devno	MIPI RX device ID
	Vc0Dt	Count of data types unsupported by VC0
	Vc1Dt	Count of data types unsupported by VC1
	Vc0FrmCrc	Count of frame data errors for VC0
	Vc1FrmCrc	Count of frame data errors for VC1
MIPI ERROR INT INFO 3 (available only in MIPI mode)	Devno	MIPI RX device ID
	Vc0FrmSeq	Count of frame sequence errors for VC0
	Vc1FrmSeq	Count of frame sequence errors for VC1
	Vc0BndryMt	Count of short packets with SOF and EOF mismatching for VC0
	Vc1BndryMt	Count of short packets with SOF and EOF mismatching for VC1
MIPI ERROR INT INFO 4 (available only in MIPI mode)	Devno	MIPI RX device ID
	DataFifoRdErr	Count of MIPI CTRL read data FIFO interrupts
	CmdFifoRdErr	Count of MIPI CTRL read command FIFO interrupts
	Vsync	Count of MIPI CTR VSync interrupts
	CmdFifoWrErr	Count of MIPI CTRL write command FIFO interrupts
	DataFifoWrErr	Count of MIPI CTRL write data FIFO interrupts
LVDS ERROR INT INFO 1 (available only in LVDS mode)	Devno	MIPI device ID
	Vsync	Count of VSync interrupts for LVDS
	CmdRdErr	Count of CMD_FIFO read error interrupts for LVDS
	CmdWrErr	Count of CMD_FIFO write error interrupts for LVDS

Parameter		Description
	PopErr	Count of line_buf read error interrupts for LVDS
	StatErr	Count of lane synchronization error interrupts for LVDS
LVDS ERROR INT INFO 2 (available only in LVDS mode)	Devno	MIPI device ID
	Link0WrErr	Count of FIFO read error interrupts for link 0
	Link1WrErr	Count of FIFO read error interrupts for link 1
	Link0RdErr	Count of FIFO write error interrupts for link 0
	Link1RdErr	Count of FIFO write error interrupts for link 1
ALING ERROR INFO	Devno	MIPI device ID
	FIFO_FullErr	FIFO overflow
	Lane0Err	Lane 0 FIFO overflow
	Lane1Err	Lane 1 FIFO overflow

1.7.2 MIPI_TX Proc Information

NOTICE

The MIPI_TX proc information is applicable to Hi3559A V100, Hi3519A V100, and Hi3516C V500.

The proc information of the MIPI TX includes the configuration information, timing configuration information, and status information of the MIPI TX.

[Debugging Information]

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

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

----------MIPI_Tx DEV STATUS-------------------------
```



```
width  height HoriAll VertAll hbp hsa vsa
1080   1920   1237   1972    20  8   10
```

[Analysis]

Proc information such as MIPI TX device configuration information and MIPI TX timing configuration information is configured through the interface before the device starts. The MIPI TX device status information is part of the timing information detected during device running, which includes the valid width and height, total horizontal width, and total vertical height.

[Parameter Description]

Parameter		Description
MIPI_Tx DEV CONFIG	devno	Device ID
	lane0 lane1 lane2 lane3	≥ 0: lane ID –1: disabled
	output_mode	0: CSI mode 1: DSI video mode 2: DSI command mode
	phy_data_rate	PHY data rate
	pixel_clk(KHz)	Pixel clock, in KHz
	video_mode	0: BURST_MODE 1: NON_BURST_MODE_SYNC_PULSES 2: NON_BURST_MODE_SYNC_EVENTS
	output_fmt	0:OUT_FORMAT_RGB_16_BIT 1:OUT_FORMAT_RGB_18_BIT 2:OUT_FORMAT_RGB_24_BIT 3:OUT_FORMAT_YUV420_8_BIT_NORMAL 4:OUT_FORMAT_YUV420_8_BIT_LEGACY 5:OUT_FORMAT_YUV422_8_BIT
MIPI_Tx SYNC CONFIG	pkt_size	hact
	hsa_pixels	hsync
	hbp_pixels	hbp
	hline_pixels	hact+hsa+hbp+hfp
	vsa_lines	vsa
	vbp_lines	vbp
	vfp_lines	vfp

Parameter		Description
	active_lines	vact
	edpi_cmd_size	edpi cmd size
MIPI_Tx DEV STATUS	width	Detected width
	height	Detected width
	HoriAll	Total horizontal width
	VertAll	Total vertical height
	hbp	Horizontal front porch
	hsa	Number of horizontal sync pulses
	vsa	Number of vertical sync pulses

1.8 FAQs

For details about the Hi3559A V100ES MIPI specifications, see the *Hi35xx Vxxx ultra-HD Mobile Camera SoC Data Sheet* and *Features of the Video Interfaces of HiSilicon IP Cameras*.

1.8.1 How Do I Configure a Lane ID?

The lane ID corresponds to **short lane_id[**MIPI_LANE_NUM**]** in mipi_dev_attr_t or **short lane_id[**SLVS_LANE_NUM] in slvs_dev_attr_t. The index number of the lane_id array indicates the sensor lane ID, and the value of the lane_id array the MIPI lane ID.

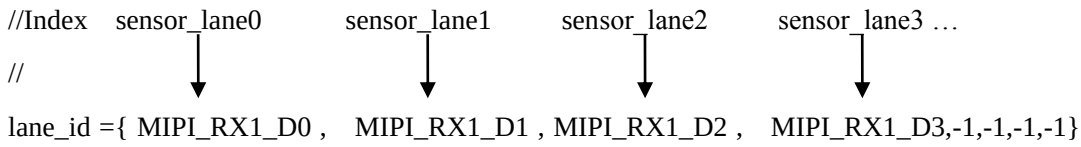
Set **lane_id** of unused lanes to **–1** when the MIPI connects to the sensor. **You can also adjust the data channel sequence by configuring lane-id based on the hardware board and actual sensor output channels.**

Table 1-7 shows an example for pin hardware connection between the MIPI and the sensor.

Table 1-7 Mapping between the sensor and MIPI RX pins

MIPI Lane Pins	Sensor Lane Pins
MIPI_RX1_D0	Lane 0
MIPI_RX1_D1	Lane 1
MIPI_RX1_D2	Lane 2
MIPI_RX1_D3	Lane 3

The MIPI has a maximum number of eight lanes. It is supposed that the sensor also has up to eight lanes. Because there are actually four sensor lanes that transmit data to four MIPI lanes, it requires that the IDs of sensor unconnected lanes or nonexistent lanes are set to **–1**. **lane_id** is configured as follows:



1.8.2 What Is the Relationship Between the MIPI Lane Frequency and VI Frequency?

When multiple MIPI lanes are used for data transfer, what is the relationship between the transfer frequency of MIPI lanes and VI processing frequency, and how do I calculate the maximum transfer frequency of each lane?

- The MIPI RX receives data from multiple lanes, converts data into the internal timing, and transmits the timing to the video input unit (VIU) for processing. The total amount of data transferred by multiple lanes remains unchanged, as described in the following equation:

  VI_Freq x Pix_Width = Lane_Num x MIPI_Freq

- Where **VI_Freq** indicates the frequency of the VI working clock, **Pix_Width** indicates the pixel bit width, **Lane_Num** indicates the number of lanes used for data transfer, and **MIPI_Freq** indicates the maximum RX frequency of each lane.
- **MIPI_Freq** is calculated as follows: MIPI_Freq = (VI_Freq x Pix_Width)/Lane_Num. For example, if the frequency of the VI working clock is 250 MHz, the MIPI data is in RAW12 format, and four lanes are used for data transfer, **MIPI_Freq** is calculated as follows:

  MIPI_Freq = (250 x 12)/4 = 750

  That is, the maximum transfer frequency of each lane is 750 MHz.

