CV182x

MIPI 使用指南

Version: 1.1.1
Release date: 2021-04-20

修订记录

版本	日期	修订说明	修订人
1.0.0	2021/04/20	RD update by review	廖尹辰
1.1.1	2021/06/21	更正 MIPI_TX ioctl	黃立銘

法律声明

本数据手册包含北京晶视智能科技有限公司（下称“晶视智能”）的保密信息。未经授权，禁止使用或披露本数据手册中包含的信息。如您未经授权披露全部或部分保密信息，导致晶视智能遭受任何损失或损害，您应对因之产生的损失/损害承担责任。

本文件内信息如有更改，恕不另行通知。晶视智能不对使用或依赖本文件所含信息承担任何责任。

本数据手册和本文件所含的所有信息均按“原样”提供，无任何明示、暗示、法定或其他形式的保证。晶视智能特别声明未做任何适销性、非侵权性和特定用途适用性的默示保证，亦对本数据手册所使用、包含或提供的任何第三方的软件不提供任何保证；用户同意仅向该第三方寻求与此相关的任何保证索赔。此外，晶视智能亦不对任何其根据用户规格或符合特定标准或公开讨论而制作的可交付成果承担责任。

目录

1. 功能概述 ..... 错误!未定义书签。
1.1. 目的 ..... 错误!未定义书签。
1.2. 定义及缩写 ..... 错误!未定义书签。
1.3. 参考指引 ..... 错误!未定义书签。
2. 设计概述 ..... 错误!未定义书签。
2.1. 系统架构 ..... 错误!未定义书签。
2.2. 注意事项 ..... 错误!未定义书签。
3. API 参考 ..... 错误!未定义书签。
4. 数据类型 ..... 错误!未定义书签。
5. 错误码 ..... 错误!未定义书签。
6. 相关测试 ..... 错误!未定义书签。
6.1. 单元测试 ..... 错误!未定义书签。
6.2. 功能测试 ..... 错误!未定义书签。
6.3. 性能测试 ..... 错误!未定义书签。

1. MIPI 使用指南

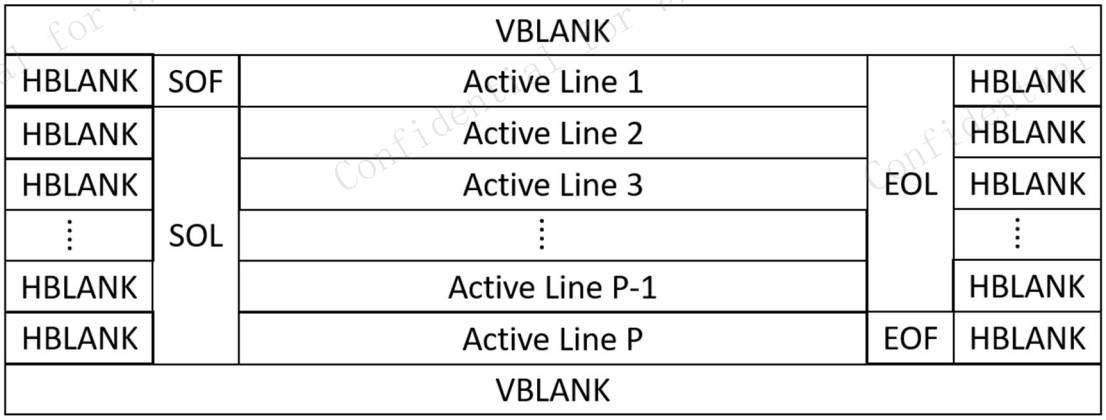
1.1. 概述

MIPI Rx 可接收差分與 DC(TTL)介面數據，並將數據轉換成 pixel 數據後傳給下一級的 ISP 模塊。 差分訊號支援 SubLVDS(Sub Low-Voltage Differential Signal), MIPI-CSI 與 HiSPi(High-Speed Serial Pixel Interface)等視頻輸入。DC 訊號支援 Sensor RAW12, BT1120, BT656 與 BT601。

1.2. 重要概念

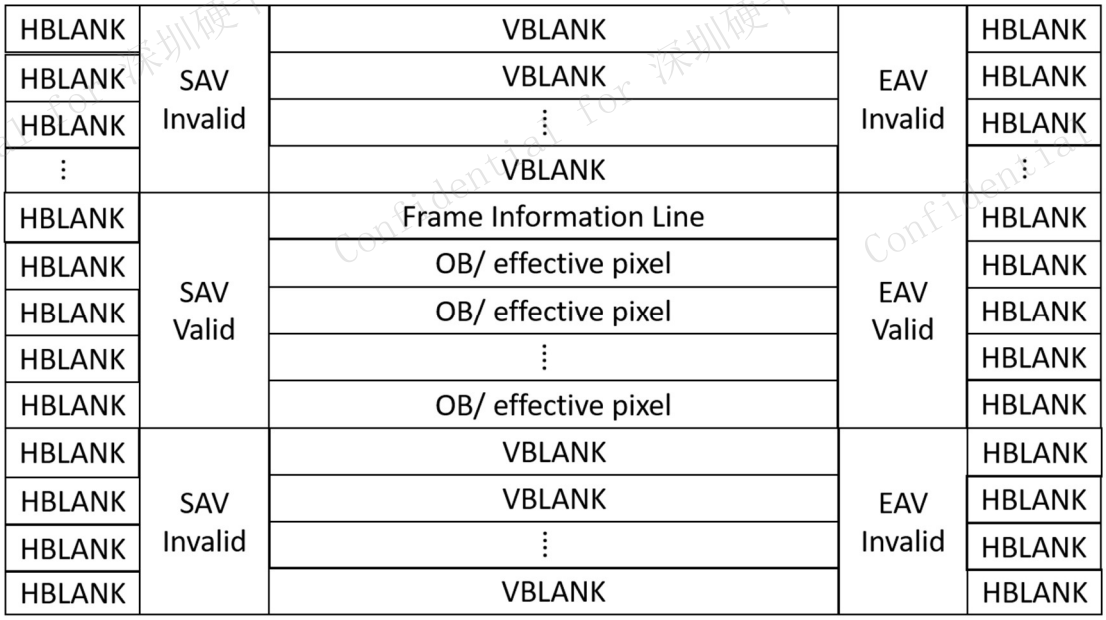
- **MIPI**
  移动行业处理器接口-Mobile Industry Processor Interface，MIPI 特指物理层使用 D-PHY 传输规范並使用 CSI-2 為协议层的通信接口。
- **Lane**
  物理層用于连接发送端和接收端的一对高速差分线。一個 Lane 可傳送時鐘或資料。1C4D 指一個時鐘 Lane 及 4 個資料 Lane。
- **LVDS**
  低压差分信号(Low Voltage differential Signaling)，這裡的 LVDS 泛指 LVDS 發展的 sub-LVDS 與 HiSPi，通过同步码区分消隐区和有效数据。
- **同步碼**
  MIPI-CSI 利用標準的短包(Short Packet)當做同步訊號。 LVDS 訊號利用同包碼(Sync Code)作為同步訊號。 LVDS 有兩種同步模式:
  - 使用 SOF/EOF 表示一幀的開始與結束。
  - 使用 SOL/EOL 表示行的開始與結束。

圖 1-1 SOF/EOF/SOL/EOL 同步方式



- 使用 SAV invalid 與 EAV invalid 表示 VBLANK 的開始與結束。 使用 SAV valid 與 EAV valid 表示有效資料(information line, H.OB 與 pixel data)的開始與結束。

圖 1-2 SAV/EAV 同步方式



- **DOL**
  SONY 的交錯式 WDR 模式，全稱為 Digital Overlap。

1.3. 功能描述

MIPI RX 支援一路最大 4 lanes 或兩路最大 2 lanes 的 MIPI-CSI，sub-LVDS 與 HiSPi 差分視頻輸入接口，接口支持 lane 互換與差分訊號的 PN 互換。注意若使用兩路插分訊號輸入，腳位 MIPIRX#必須 0 到 2 同一組輸入，3 到 5 同一組輸

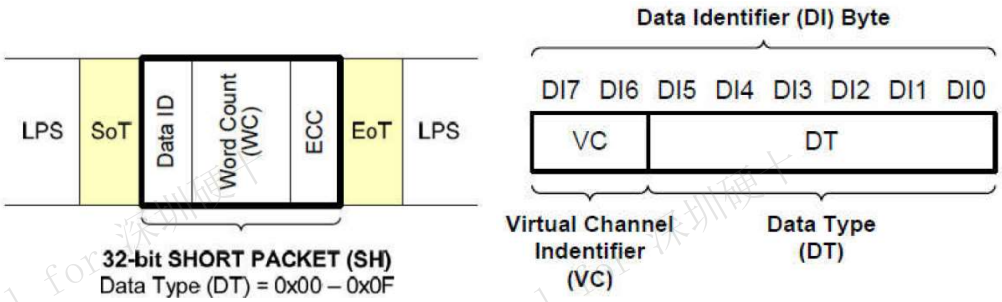
入。MIPI RX 也支援 sensor TTL，BT1120，BT656 與 BT601 等平行輸入接口，支持除了 Clock 以外的 lane 功能互換。 具體的 Lane 分佈如下表:

表 1-1 支持 BT 接口分佈表. BGA 封裝支持 VI0~VI2, QFN 封裝支持 VI0.

VI0	PAD NAME	VI1	PAD NAME	VI2	PAD NAME
VI0_CLK	MIPIRX4N	VI1_CLK	VIVO_CLK	VI2_CLK	VIDO_CLK
VI0_D[0]	MIPIRX4P	VI1_D[0]	VIVO_D0	VI2_D[0]	VIVO_D0
VI0_D[1]	MIPIRX3N	VI1_D[1]	VIVO_D1	VI2_D[1]	VIVO_D1
VI0_D[2]	MIPIRX3P	VI1_D[2]	VIVO_D2	VI2_D[2]	VIVO_D2
VI0_D[3]	MIPIRX2N	VI1_D[3]	VIVO_D3	VI2_D[3]	VIVO_D3
VI0_D[4]	MIPIRX2P	VI1_D[4]	VIVO_D4	VI2_D[4]	VIVO_D4
VI0_D[5]	MIPIRX1N	VI1_D[5]	VIVO_D5	VI2_D[5]	VIVO_D5
VI0_D[6]	MIPIRX1P	VI1_D[6]	VIVO_D6	VI2_D[6]	VIVO_D6
VI0_D[7]	MIPIRX0N	VI1_D[7]	VIVO_D7	VI2_D[7]	VIVO_D7
VI0_D[8]	MIPIRX0P	VI1_D[8]	VIVO_D8		
VI0_D[9]	MIPI_TXM0	VI1_D[9]	VIVO_D9		
VI0_D[10]	MIPI_TXP0	VI1_D[10]	VIVO_D10		
VI0_D[11]	MIPI_TXM1	VI1_D[11]	MIPIRX5N		
VI0_D[12]	MIPI_TXP1	VI1_D[12]	MIPIRX5P		
VI0_D[13]	MIPI_TXM2	VI1_D[13]	MIPIRX4N		
VI0_D[14]	MIPI_TXP2	VI1_D[14]	MIPIRX4P		
		VI1_D[15]	MIPIRX3N		
		VI1_D[16]	MIPIRX3P		
		VI1_D[17]	MIPIRX2N		
		VI1_D[18]	MIPIRX2P		

MIPI RX 支援 MIPI-CSI 的解多工 (CSI Demux)，可接收不同 channel ID 的通道資料。

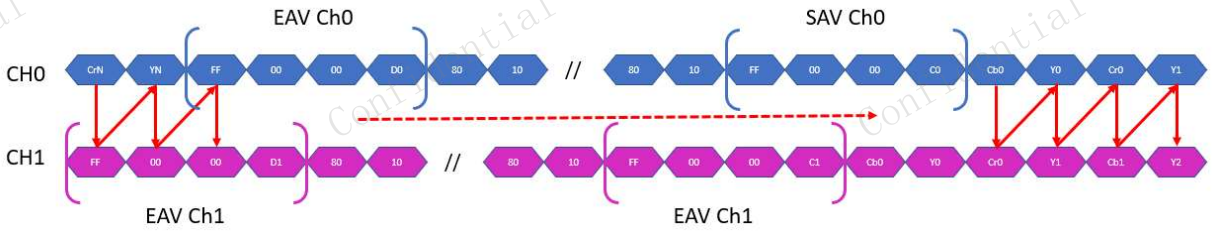
圖 13 MIPI-CSI 的 channel ID



MIPI RX 支援 BT 接口的解多工 (BT Demux)，輸入端將不同通道的資料以 BT656 字節交錯的格式打包，MIPI RX 可還原各通道再以對應的同步碼解出有效資料。注意此模式只支

援單沿取樣(SDR)。

圖 1-4 BT demux 模式支援的輸入



1.4. API 參考

MIPI Rx 提供對接 sensor 時序的功能。 提供 ioctl 接口，可用的命令如下:

- CVI_MIPI_SET_HS_MODE: 設置 MIPI 的 Lane 分布模式。
- CVI_MIPI_SET_DEV_ATTR: 設置 MIPI 和並口設備屬性。
- CVI_MIPI_RESET_SENSOR: 復位 sensor。
- CVI_MIPI_UNRESET_SENSOR: 撤銷復位 sensor。
- CVI_MIPI_RESET_MIPI: 復位 MIPI Rx。
- CVI_MIPI_ENABLE_SENSOR_CLOCK: 打開 SENSOR 的時鐘。
- CVI_MIPI_DISABLE_SENSOR_CLOCK: 關閉 SENSOR 的時鐘。
- CVI_MIPI_SET_CROP_TOP: 捨棄每幀中的首 N 條資料。
- CVI_MIPI_SET_WDR_MANUAL: 打開 WDR 手動模式。
- CVI_MIPI_SET_LVDS_FP_VS: 設定 LVDS 中 VSYNC 生成的時間點。

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

- CVI_VIP_MIPI_TX_SET_DEV_CFG：设置 MIPI Tx 设备的属性。
- CVI_VIP_MIPI_TX_GET_CMD：从 MIPI Tx 设备读取信息。
- CVI_VIP_MIPI_TX_SET_CMD：设备发送命令给 MIPI Tx 设备
- CVI_VIP_MIPI_TX_ENABLE：启用 MIPI Tx 设备。
- CVI_VIP_MIPI_TX_DISABLE：禁用 MIPI Tx 设备。
- CVI_VIP_MIPI_TX_SET_HS_SETTLE：設定 MIPI Tx 在 HS mode 下的 settle timing。
- CVI_VIP_MIPI_TX_GET_HS_SETTLE：取得 MIPI Tx 在 HS mode 下的 settle timing。

CVI_MIPI_SET_HS_MODE

【描述】

相關功能被 CVI_MIPI_SET_DEV_ATTR 取代。

CVI_MIPI_SET_DEV_ATTR

【描述】

設置 MIPI 和並口設備屬性。

【参數】

#define CVI_MIPI_SET_DEV_ATTR _IOW(CVI_MIPI_IOC_MAGIC, 0x01, struct combo_dev_attr_t)

【定義】

struct combo_dev_attr_t 類型的指針。

【返回值】

返回值	描述
0	成功
-1	失敗，並設置 errno

【芯片差異】

無

【需求】

頭文件: cvi_vip_cif.h

【注意】

- 配置 CVI_MIPI_SET_DEV_ATTR 之前，需要使用 ISP 接口打開 MIPI_RX 時鐘. 詳情請見 ISP 相關文件。
- 除了配置 CVI_MIPI_SET_DEV_ATTR 之前，還需要配置以下接口。
- 復位 MIPI: 接口為 CVI_MIPI_RESET_MIPI。
- 打開 Sensor 的時鐘: 接口為 CVI_MIPI_ENABLE_SENSOR_CLOCK
- 復位 Sensor: 接口為 CVI_MIPI_RESET_SENSOR
- 撤銷復位 Sensor: 接口為 CVI_MIPI_UNRESET_SENSOR
- 推薦的配置流程如下:
  1. 打開 ISP 時鐘。
  2. 復位對接的 Sensor。
  3. 復位 MIPI Rx。
  4. 配置 MIPI Rx 設備屬性。

4. 打開 Sensor 所連接的時鐘。
5. 撤銷復位對接的 Sensor。

- 推薦的退出程序如下:
  1. 復位對接的 Sensor。
  2. 關閉 Sensor 所連接的時鐘。
  3. 復位 MIPI Rx。
  4. 關閉 ISP 時鐘。
- 操作 Sensor 復位信號線和時鐘信號線會對所連接到該信號線的所有 Sensor 都產生效果。

【相關數據類型及接口】

- CVI_MIPI_RESET_MIPI
- CVI_MIPI_ENABLE_SENSOR_CLOCK
- CVI_MIPI_DISABLE_SENSOR_CLOCK
- CVI_MIPI_RESET_SENSOR
- CVI_MIPI_UNRESET_SENSOR

CVI_MIPI_RESET_SENSOR

【描述】

復位 sensor

【定義】

```
#define CVI_MIPI_RESET_SENSOR
_IOW(CVI_MIPI_IOC_MAGIC, 0x05, unsigned int)
```

【參數】

unsigned int。 Sensor 復位信號線編號。

【返回值】

返回值	描述
0	成功
-1	失敗，並設置 errno

【芯片差異】

無

【需求】

頭文件: cvi_vip_cif.h

【注意】

- Sensor 復位信號綁定在對應的 dts。

```
mipi_rx: cif {
        compatible = "cvitek,cif";
        reg = <0x0 0x0a0c2000 0x0 0x2000>, <0x0 0x0a0d0000 0x0
        0x1000>,
        <0x0 0x0a0c4000 0x0 0x2000>;
        reg-names = "csi_mac0", "csi_wrap0", "csi_mac1";
        interrupts = <GIC_SPI 155 IRQ_TYPE_LEVEL_HIGH>,
                     <GIC_SPI 156 IRQ_TYPE_LEVEL_HIGH>;
        interrupt-names = "csi0", "csi1";
        snsr-reset = <&porta 2 GPIO_ACTIVE_LOW>, <&porta 2
        GPIO_ACTIVE_LOW>;
        resets = <&rst RST_CSIPHY0>, <&rst RST_CSIPHY1>,
        <&rst RST_CSIPHY0RST_APB>, <&rst RST_CSIPHY1RST_APB>;
        reset-names = "phy0", "phy1", "phy-apb0", "phy-apb1";
        clocks = <&clk CV182X_CLK_CAM0>, <&clk
        CV182X_CLK_CAM1>, <&clk CV182X_CLK_SRC_VIP_SYS_2>;
        clock-names = "clk_cam0", "clk_cam1", "clk_sys_2";
};
```

CVI_MIPI_UNRESET_SENSOR

【描述】

撤銷復位 sensor。

【定義】

```
#define CVI_MIPI_UNRESET_SENSOR
_IOW(CVI_MIPI_IOC_MAGIC, 0x06, unsigned int)
```

【參數】

unsigned int。 Sensor 復位信號線編號。

【返回值】

返回值	描述
0	成功
-1	失敗，並設置 errno

【芯片差異】

無

【需求】

頭文件: cvi_vip_cif.h

【注意】

- Sensor 復位信號綁定在對應的 dts。

```
mipi_rx: cif {
        compatible = "cvitek,cif";
        reg = <0x0 0x0a0c2000 0x0 0x2000>, <0x0 0x0a0d0000 0x0
        0x1000>,
        <0x0 0x0a0c4000 0x0 0x2000>;
        reg-names = "csi_mac0", "csi_wrap0", "csi_mac1";
        interrupts = <GIC_SPI 155 IRQ_TYPE_LEVEL_HIGH>,
                     <GIC_SPI 156 IRQ_TYPE_LEVEL_HIGH>;
        interrupt-names = "csi0", "csi1";
        snsr-reset = <&porta 2 GPIO_ACTIVE_LOW>, <&porta 2
        GPIO_ACTIVE_LOW>;
        resets = <&rst RST_CSIPHY0>, <&rst RST_CSIPHY1>,
        <&rst RST_CSIPHY0RST_APB>, <&rst RST_CSIPHY1RST_APB>;
        reset-names = "phy0", "phy1", "phy-apb0", "phy-apb1";
        clocks = <&clk CV182X_CLK_CAM0>, <&clk
        CV182X_CLK_CAM1>, <&clk CV182X_CLK_SRC_VIP_SYS_2>;
        clock-names = "clk_cam0", "clk_cam1", "clk_sys_2";
};
```

CVI_MIPI_RESET_MIPI

【描述】

復位 MIPI_Rx。

【定義】

```
#define CVI_MIPI_RESET_MIPI                _IOW(CVI_MIPI_IOC_MAGIC,
0x07, unsigned int)
```

【參數】

unsigned int。 MIPI_Rx 設備號。

【返回值】

返回值	描述

0	成功
-1	失敗，並設置 errno

【芯片差異】

無

【需求】

頭文件: cvi_vip_cif.h

【注意】

無

CVI_MIPI_ENABLE_SENSOR_CLOCK

【描述】

打開 Sensor 的時鐘。

【定義】

```
#define CVI_MIPI_ENABLE_SENSOR_CLOCK
_IOW(CVI_MIPI_IOC_MAGIC, 0x10, unsigned int)
```

【參數】

unsigned int。 MIPI_Rx 設備號。

【返回值】

返回值	描述
0	成功
-1	失敗，並設置 errno

【芯片差異】

無

【需求】

頭文件: cvi_vip_cif.h

【注意】

無

CVI_MIPI_DISABLE_SENSOR_CLOCK

【描述】

關閉 Sensor 的時鐘。

【定義】

```
#define CVI_MIPI_DISABLE_SENSOR_CLOCK      _IOW(CVI_MIPI_IOC_MAGIC, 0x11,
unsigned int)
```

【參數】

unsigned int。 MIPI_Rx 設備號。

【返回值】

返回值	描述
0	成功
-1	失敗，並設置 errno

【芯片差異】

無

【需求】

頭文件: cvi_vip_cif.h

【注意】

無

CVI_MIPI_SET_CROP_TOP

【描述】

捨棄每幀中的首 N 條資料

【定義】

```
#define CVI_MIPI_SET_WDR_MANUAL
_IOW(CVI_MIPI_IOC_MAGIC, 0x21, struct crop_top_s)
```

【參數】

struct crop_top_s

【返回值】

返回值	描述
0	成功
-1	失敗，並設置 errno

【芯片差異】

無

【需求】

頭文件: cvi_vip_cif.h

【注意】

無

CVI_MIPI_ SET_WDR_MANUAL

【描述】

打開 WDR 手動模式

【定義】

```
#define CVI_MIPI _SET_WDR_MANUAL          _IOW(CVI_MIPI_IOC_MAGIC, 0x21, struct manual_wdr_s)
```

【參數】

struct manual_wdr_s

【返回值】

返回值	描述
0	成功
-1	失敗，並設置 errno

【芯片差異】

無

【需求】

頭文件: cvi_vip_cif.h

【注意】

無

CVI_MIPI_SET_LVDS_FP_VS

【描述】

設定 sub-LVDS 中 VSYNC 生成的時間點

【定義】

```
#define CVI_MIPI _SET_LVDS_FP_VS          _IOW(CVI_MIPI_IOC_MAGIC, 0x22, struct vsync_gen_s)
```

【參數】

struct vsync_gen_s

【返回值】

返回值	描述
0	成功
-1	失敗，並設置 errno

【芯片差異】

無

【需求】

頭文件: cvi_vip_cif.h

【注意】

無

CVI_VIP_MIPI_TX_SET_DEV_CFG

【描述】

设置 MIPI Tx 设备的属性。

【定義】

```
#define CVI_VIP_MIPI_TX_SET_DEV_CFG _IOW( CVI_MIPI _TX_IOC_MAGIC , 0x01,
combo_dev_cfg_t)
```

【參數】

MIPI Tx 设备属性。

【返回值】

返回值	描述
0	成功
-1	失敗，並設置 errno

【芯片差異】

無

【需求】

頭文件: cvi_mipi_tx.h

【注意】

無

CVI_VIP_MIPI_TX_GET_CMD

【描述】

从 MIPI Tx 设备读取信息。

【定義】

```
#define CVI_VIP_MIPI_TX_GET_CMD _IOW( CVI_MIPI _TX_IOC_MAGIC , 0x04,
Get_cmd_info_t)
```

【參數】

详见 get_ cmd_info_t 结构体 说明。

【返回值】

返回值	描述
0	成功
-1	失敗，並設置 errno

【芯片差異】

無

【需求】

頭文件: cvi_mipi_tx.h

【注意】

無

CVI_VIP_MIPI_TX_SET_CMD

【描述】

设备发送命令给 MIPI Tx 设备。

【定義】

```
#define CVI_VIP_MIPI_TX_SET_CMD _IOW( CVI_MIPI _TX_IOC_MAGIC , 0x02, cmd_info_t)
```

【參數】

详见 cmd_info_t 结构体说明。

【返回值】

返回值	描述
0	成功
-1	失敗, 並設置 errno

【芯片差異】

無

【需求】

頭文件: cvi_mipi_tx.h

【注意】

CVI_VIP_MIPI_TX_ENABLE

【描述】

启用 MIPI Tx 设备。

【定義】

```
#define CVI_VIP_MIPI_TX_ENABLE _IOW( CVI_MIPI _TX_IOC_MAGIC , 0x03, cmd_info_t)
```

【參數】

無

【返回值】

返回值	描述
0	成功
-1	失敗，並設置 errno

【芯片差異】

無

【需求】

頭文件: cvi_mipi_tx.h

【注意】

此 API 呼叫后 mipi 将切换至 HS 模式。

CVI_VIP_MIPI_TX_DISABLE

【描述】

禁用 MIPI Tx 设备。

【定義】

```
#define CVI_VIP_MIPI_TX_DISABLE _IOW( CVI_MIPI _TX_IOC_MAGIC , 0x05, cmd_info_t)
```

【參數】

無

【返回值】

返回值	描述
0	成功
-1	失敗，並設置 errno

【芯片差異】

無

【需求】

頭文件: cvi_mipi_tx.h

【注意】

此 API 呼叫后 mipi 将切换至 LP 模式。

CVI_VIP_MIPI_TX_SET_HS_SETTLE

【描述】

設定 MIPI Tx 在 HS mode 下的 settle timing。

【定義】

```
#define CVI_VIP_MIPI_TX_SET_HS_SETTLE IOW(CVI_MIPI_TX_IOC_MAGIC, 0x06, struct hs_settle_s)
```

【參數】

無

【返回值】

返回值	描述
0	成功
-1	失敗，並設置 errno

【芯片差異】

無

【需求】

頭文件: cvi_mipi_tx.h

【注意】

CVI_VIP_MIPI_TX_GET_HS_SETTLE

【描述】

取得 MIPI Tx 在 HS mode 下的 settle timing。

【定義】

```
#define CVI_VIP_MIPI_TX_GET_HS_SETTLE _IOWR(CVI_MIPI_TX_IOC_MAGIC, 0x06, struct hs_settle_s)
```

【參數】

無

【返回值】

返回值	描述
0	成功
-1	失敗，並設置 errno

【芯片差異】

無

【需求】

頭文件: cvi_mipi_tx.h

【注意】

1.5. 數據類型

MIPI RX 相關數據類型定義如下:

- CVI_MIPI_IOC_MAGIC: MIPI Rx ioctl 命令的幻數。
- MIPI_LANE_NUM: MIPI Rx 的 MIPI 設備支持的最大 Lane 數。
- WDR_VC_NUM: 定義最多支持的 Virtual Channel 數量。
- SYNC_CODE_NUM: 定義 LVDS 每個 Virtual Channel 的同步碼數量。
- input_mode_e: MIPI Rx 輸入接口類型。
- img_size_s: MIPI Rx 輸入資料每幀的大小。
- rx_mac_clk_e: MAC 的支持工作时鐘。
- cam_pll_freq_e: MIPI-RX 輸出的 Sensor 參考時鐘。
- mclk_pll_s: MIPI-RX 輸出的 Sensor 參考時鐘設定。
- raw_data_type_e: MIPI Rx 輸入資料格式。
- mipi_wdr_mode_e: MIPI-CSI WDR 模式。
- wdr_mode_e: LVDS/HISPI WDR 模式。
- lvds_sync_mode_e: LVDS 同步模式。
- lvds_bit_endian: 比特位大小端模式。
- lvds_vsync_type_e: LVDS 在 WDR 模式的同步方式。
- lvds_fid_type_e: LVDS frame identification ID 類型
- lvds_fid_type_s: LVDS frame identification ID 類型
- lvds_vsync_type_s: LVDS WDR 同步參數
- lvds_dev_attr_s: SubLVDS/HiSPi 設備屬性
- dphy_s: MIPI DPHY 屬性
- mipi_demux_info_s: MIPI 解多工模式屬性
- mipi_dev_attr_s: MIPI-CSI 設備屬性
- manual_wdr_attr_s: 手動 WDR 模式參數
- ttl_pin_func_e: TTL/BT 接口的配置功能
- ttl_src_e: TTL/BT 接口輸入來源
- bt_demux_mode_e: BT 解多工模式的通道數量
- bt_demux_sync_s: BT 解多工模式的同步碼配置
- bt_demux_attr_s: BT 解多工模式的設備屬性
- ttl_dev_attr_s: TTL/BT 設備屬性
- combo_dev_attr_s: combo 設備屬性
- crop_top_s: 捨棄開頭的行資料屬性

- manual_wdr_s: 手動 WDR 模式設定
- vsync_gen_s: SUBLVDS 垂直同步訊號屬性

MIPI TX 相關數據類型定義如下:

- LANE_MAX_NUM: 一個 MIPI Tx 設備支援的最大 Lane 數
- output_mode_e: MIPI Tx 輸出模式
- video_mode_e: MIPI Tx 视频模式
- output_format_e: MIPI Tx 輸出格式
- sync_info_s: MIPI Tx 設備同步信息
- combo_dev_cfg_s: MIPI Tx 設備屬性
- cmd_info_s: 從 MIPI Tx 設備取回信息
- get_cmd_info_s: 從 MIPI Tx 設備取回信息
- hs_settle_s: MIPI Tx 設備 HS mode 下的 settle 信息

CVI_MIPI_IOC_MAGIC

【说明】

MIPI Rx ioctl 命令的幻數

【定义】

```
#define CVI_MIPI_IOC_MAGIC                'm'
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

無

MIPI_LANE_NUM

【说明】

一個 MIPI Rx 設備支援的最大 Lane 數。

【定义】

```
#define MIPI_LANE_NUM     4
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

無

WDR_VC_NUM

【说明】

定義最多支持的 Virtual Channel 數量

【定义】

```
#define WDR_VC_NUM     2
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

無

SYNC_CODE_NUM

【说明】

定義 LVDS 每個 Virtual Channel 的同步碼數量。

【定义】

```
#define SYNC_CODE_NUM     4
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

無

BT_DEMUX_NUM

【说明】

定義使用 bt demux 功能時支持最大的通道數量。

【定义】

```
#define BT_DEMUX_NUM     4
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

無

MIPI_DEMUX_NUM

【说明】

定義使用 mipi demux 功能時支持最大的 virtual channel 數量。

【定义】

```
#define MIPI_DEMUX_NUM     4
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

無

input_mode_e

【说明】

MIPI Rx 輸入接口類型。

【定义】

```
enum input_mode_e {
        INPUT_MODE_MIPI = 0,
        INPUT_MODE_SUBLVDS,
        INPUT_MODE_HISPI,
        INPUT_MODE_CMOS,
        INPUT_MODE_BT1120,
        INPUT_MODE_BT601_19B_VHS,
        INPUT_MODE_BT656_9B,
        INPUT_MODE_CUSTOM_0,
        INPUT_MODE_BT_DEMUX,
        INPUT_MODE_BUTT
};
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

無

img_size_s

【说明】

MIPI Rx 輸入資料每幀的大小。

【定义】

```
struct img_size_s {
        unsigned int    width;
        unsigned int    height;
};
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

無

rx_mac_clk_e

【说明】

MAC 的支持工作时鐘。

【定义】

```
enum rx_mac_clk_e {
        RX_MAC_CLK_200M = 0,
    RX_MAC_CLK_400M,
        RX_MAC_CLK_500M,
    RX_MAC_CLK_600M,
    RX_MAC_CLK_BUTT,
};
```

【芯片差异】

无。

【注意事项】

MAC 時鐘與支持的 MIPI 時鐘關係請參考 1.7.2。

【相关数据类型及接口】

無

cam_pll_freq_e

【说明】

MIPI-RX 輸出的 Sensor 參考時鐘。

【定义】

```
enum cam_pll_freq_e {
        CAMPLL_FREQ_NONE = 0,
        CAMPLL_FREQ_37P125M,
        CAMPLL_FREQ_25M,
        CAMPLL_FREQ_27M,
        CAMPLL_FREQ_24M,
        CAMPLL_FREQ_26M,
        CAMPLL_FREQ_NUM
};
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

無

mclk_pll_s

【说明】

MIPI-RX 輸出的 Sensor 參考時鐘設定。

【定义】

```
struct mclk_pll_s {
        unsigned int          cam;
        enum cam_pll_freq_e    freq;
};
```

【成員】

成員名稱	描述

cam	0: 輸出為 CAM_MCLK0 1: 輸出為 CAM_MCLK1
freq	輸出的 Sensor 參考時鐘

【芯片差异】

無

【注意事项】

無

【相关数据类型及接口】

無

raw_data_type_e

【说明】

MIPI Rx 輸入資料格式。

【定义】

```
enum raw_data_type_e {
        RAW_DATA_8BIT = 0,
        RAW_DATA_10BIT,
        RAW_DATA_12BIT,
        YUV422_8BIT,      /* MIPI-CSI only */
        YUV422_10BIT,     /* MIPI-CSI only*/
        RAW_DATA_BUTT
};
```

【芯片差异】

无。

【注意事项】

YUV422_8BIT 與 YUV422_10BIT 只支援 MIPI-CSI 格式。

【相关数据类型及接口】

無

mipi_wdr_mode_e

【说明】

MIPI-CSI WDR 模式。

【定义】

```
enum mipi_wdr_mode_e {
```

```
        CVI_MIPI_WDR_MODE_NONE = 0,
        CVI_MIPI_WDR_MODE_VC,
        CVI_MIPI_WDR_MODE_DT,
        CVI_MIPI_WDR_MODE_DOL,
        CVI_MIPI_WDR_MODE_MANUAL,   /* SOI case */
        CVI_MIPI_WDR_MODE_BUTT
};
```

【成員】

成員名稱	描述
CVI_MIPI_WDR_MODE_NONE	線性模式
CVI_MIPI_WDR_MODE_VC	MIPI-CSI Virtual Channel 模式
CVI_MIPI_WDR_MODE_DT	MIPI-CSI Data Type 模式
CVI_MIPI_WDR_MODE_DOL	Sony DOL LI 模式
CVI_MIPI_WDR_MODE_MANUAL	WDR 手動模式

【芯片差异】

无。

【注意事项】

- CVI_MIPI_WDR_MODE_VC 適用於使用 MIPI-CSI Virtual Channel ID 分辨長曝線與短曝線的 Sensor。
- CVI_MIPI_WDR_MODE_DT 適用於使用 MIPI-CSI Data Type ID 分辨長曝線與短曝線的 Sensor。 注意注意長曝與短曝必須在同一幀開始與結束。
- CVI_MIPI_WDR_MODE_DOL 適用於使用 SONY MIPI-CSI Line Information 模式。
- CVI_MIPI_WDR_MODE_MANUAL 使用自訂的規則決定長曝線與短曝線。

【相关数据类型及接口】

無

wdr_mode_e

【说明】

Sub-LVDS/HISPI WDR 模式。

【定义】

```
enum wdr_mode_e {
        CVI_WDR_MODE_NONE = 0,
        CVI_WDR_MODE_2F,
```

```
        CVI_WDR_MODE_3F,
        CVI_WDR_MODE_DOL_2F,
        CVI_WDR_MODE_DOL_3F,
        CVI_WDR_MODE_DOL_BUTT
};
```

【成員】

成員名稱	描述
CVI_WDR_MODE_NONE	線性模式
CVI_WDR_MODE_2F	一般雙曝 WDR
CVI_WDR_MODE_3F	一般三曝 WDR
CVI_WDR_MODE_DOL_2F	Sony DOL-2 WDR
CVI_WDR_MODE_DOL_3F	Sony DOL-3 WDR

【芯片差异】

无。

【注意事项】

- CVI_WDR_MODE_2F 適用於一般 MIPI-CSI/HiSPi 的交錯式 WDR。
- CVI_WDR_MODE_DOL_2F 適合用 Sony Sub-LVDS DOL-2 WDR。
- CV182x 不支援 CVI_WDR_MODE_3F 與 CVI_WDR_MODE_DOL_3F
- CV182x 不支援平行訊號的 WDR 模式。

【相关数据类型及接口】

無

lvds_sync_mode_e

【说明】

LVDS 同步模式。

【定义】

```
enum lvds_sync_mode_e {
        LVDS_SYNC_MODE_SOF = 0,
        LVDS_SYNC_MODE_SAV,
        LVDS_SYNC_MODE_BUTT
};
```

【成員】

成員名稱	描述
LVDS_SYNC_MODE_SOF	SOF, EOF, SOL, EOL。 請參考圖 1-1

LVDS_SYNC_MODE_SAV	Invalid SAV, invalid EAV, valid SAV, valid EAV。 請參考圖 1-2

【芯片差异】

无。

【注意事项】

- LVDS_SYNC_MODE_SOF 適用於 HiSPi Packetize-SP 模式。
- LVDS_SYNC_MODE_SAV 適用於 sub-LVDS 與 HiSPi Streaming-SP 模式
- 當輸入為 INPUT_MODE_HISPI 並且同步模式為 LVDS_SYNC_MODE_SAV。 MIPI-Rx 切換到 HiSPi Streaming-SP 模式。

【相关数据类型及接口】

無

lvds_bit_endian

【说明】

比特位大小端模式。

【定义】

```
enum lvds_bit_endian {
        LVDS_ENDIAN_LITTLE = 0,
        LVDS_ENDIAN_BIG,
        LVDS_ENDIAN_BUTT
};
```

【芯片差异】

無

【注意事项】

無

【相关数据类型及接口】

無

lvds_vsync_type_e

【说明】

LVDS 在 WDR 模式的同步方式。

【定义】

```
enum lvds_vsync_type_e {
        LVDS_VSYNC_NORMAL = 0,
```

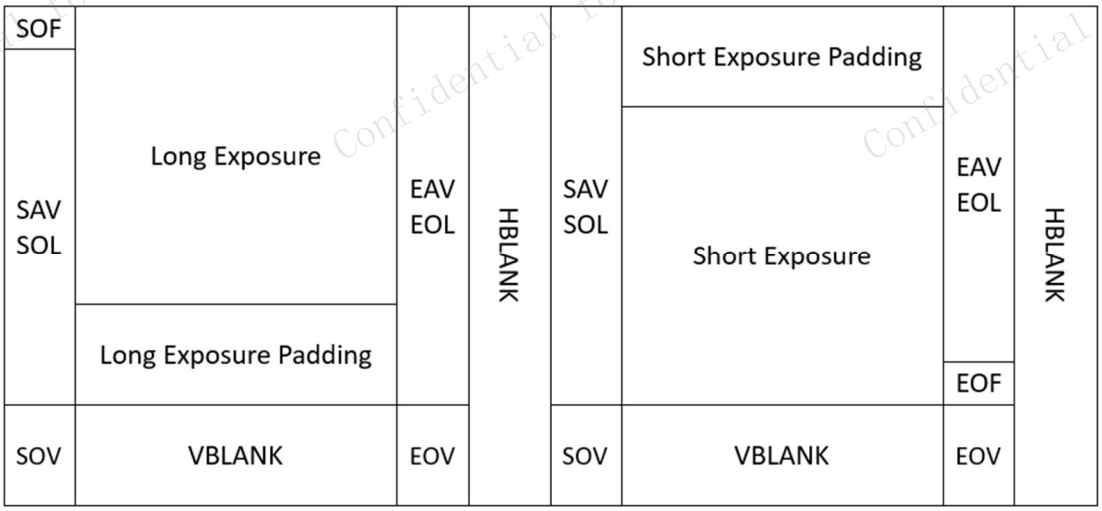
```
        LVDS_VSYNC_SHARE,
        LVDS_VSYNC_HCONNECT,
        LVDS_VSYNC_BUTT
};
```

【成員】

成員名稱	描述
LVDS_VSYNC_NORMAL	長短曝光幀有獨立的 SOF-EOF, SOL-EOL 或 invalid-SAV-invalid-EAV, valid SAV-valid EAV
LVDS_VSYNC_SHARE	長短曝光帧共用一对 SOF-EOF 标识，短曝光的起始几行用固定值填充
LVDS_VSYNC_HCONNECT	長短曝光帧共用一对 SAV-EAV 标识，长短曝光帧之间是固定周期的消隐。

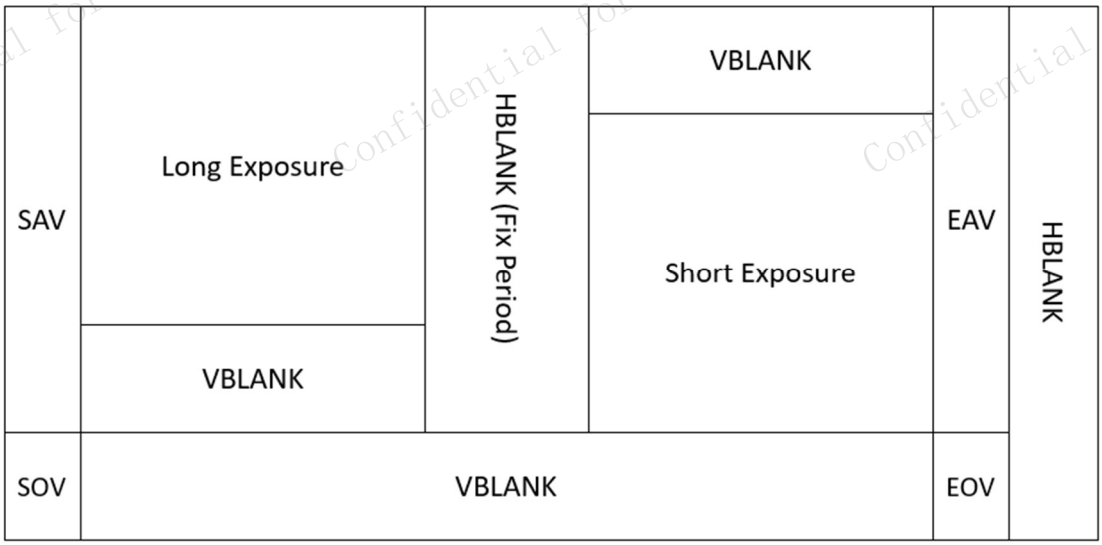
- LVDS_VSYNC_NORMAL 同步方式:

圖 1-3 LVDS_VSYNC_NORMAL 同步方式.



- LVDS_VSYNC_HCONNECT 同步方式:

圖 1-4 LVDS_VSYNC_HCONNECT 同步方式.



【芯片差异】

無

【注意事项】

- LVDS_VSYNC_NORMAL 適用於 SONY sub-LVDS DOL-2 pattern 1 與 HiSPi Packtized-SP WDR 模式。
- LVDS_VSYNC_SHARE 適用於 HiSPi Streaming-SP WDR 模式。
- LVDS_VSYNC_HCONNECT 適用於 SONY sub-LVDS DOL-2 pattern 2。

【相关数据类型及接口】

無

lvds_fid_type_e

【说明】

LVDS frame identification ID 類型。

【定义】

```
enum lvds_fid_type_e {
        LVDS_FID_NONE = 0,
        LVDS_FID_IN_SAV,
        LVDS_FID_BUTT,
};
```

【成員】

成員名稱	描述
LVDS_FID_NONE	不使用 frame identification id。

LVDS_FID_IN_SAV	FID 插入在 SAV 第 4 個字段中。

【芯片差异】

無

【注意事项】

無

【相关数据类型及接口】

無

lvds_fid_type_s

【说明】

LVDS frame identification ID 類型。

【定义】

```
struct lvds_fid_type_s {
        enum lvds_fid_type_e        fid;
};
```

【成員】

成員名稱	描述
fid	LVDS DOL 模式下的 frame identification 類型

【芯片差异】

無

【注意事项】

無

【相关数据类型及接口】

無

lvds_vsync_type_s

【说明】

LVDS WDR 同步參數。

【定义】

```
struct lvds_vsync_type_s {
        enum lvds_vsync_type_e  sync_type;
        unsigned short          hblank1;
        unsigned short          hblank2;
```

```
};
```

【芯片差异】

無

【注意事项】

- 當 sync_type 為 LVDS_VSYNC_HCONNECT 時，需要配置 hblank1 和 hblank2，表示長短曝間的消隱區長度。
- CV182x 只支援 DOL-2，所以 hblank2 不需要設定。

【相关数据类型及接口】

無

lvds_dev_attr_s

【说明】

LVDS/SubLVDS/HiSPi 設備屬性。

【定义】

```
struct lvds_dev_attr_s {
        enum wdr_mode_e                 wdr_mode;
        enum lvds_sync_mode_e           sync_mode;
        enum raw_data_type_e            raw_data_type;
        enum lvds_bit_endian            data_endian;
        enum lvds_bit_endian            sync_code_endian;
        short                           lane_id[MIPI_LANE_NUM+1];
        short
sync_code[MIPI_LANE_NUM][WDR_VC_NUM+1][SYNC_CODE_NUM];
        struct lvds_vsync_type_s        vsync_type;
        struct lvds_fid_type_s          fid_type;
        char                            pn_swap[MIPI_LANE_NUM+1];
  };
```

【成員】

成員名稱	描述
wdr_mode	WDR 模式
sync_mode	LVDS 同步模式
raw_data_type	傳輸的數據類型
data_endian	數據大小端模式

sync_code_endian	同步碼大小端模式
lane_id	發送端(sensor)和接收端(MIPI Rx) lane 的對應關係
sync_code	每个 Virtual Channel 有 4 个同步码，根据同步模式不同，分别表示 SOF/EOF/ SOL/EOL 的同步码或者 invalid SAV/invalid EAV/ valid SAV/valid EAV 的同步码。
vsync_type	vsync 类型，当 wdr_mod 为 DOL 模式并且 sync_mode 为 LVDS_SYNC_MODE_SAV 时，需要配置 vsync 的类型
fid_type	frame identification 类型
pn_swap	Positive/negative line 是否交換

【芯片差异】

無

【注意事项】

無

【相关数据类型及接口】

無

dphy_s

【说明】

MIPI RX DPHY 屬性。

【定义】

```
struct mipi_dev_attr_s {
        unsigned char       enable;
        unsigned char       hs_settle;
};
```

【成員】

成員名稱	描述
enable	開啟 MIPI RX DPHY 屬性設定
hs_settle	hs settle time

【芯片差异】

無

【注意事项】

無

【相关数据类型及接口】

無

mipi_demux_info_s

【说明】

MIPI CSI 使用 Virtual Channel 解多工的屬性設定。

【定义】

```
struct mipi_demux_info_s {
        unsigned int                    demux_en;
        unsigned char                   vc_mapping[MIPI_DEMUX_NUM];
};
```

【成員】

成員名稱	描述
demux_en	開啟 mipi virtual channel 解多工
vc_mapping	設定 ISP channel 與 mipi virtual channel 對應關係. 例如 vc_mapping = {0, 2, 3, 1}, ISP ch0 代表 vc=0; ch1 代表 vc=2; ch2 代表 vc=3; ch3 代表 vc=1.

【芯片差异】

無

【注意事项】

無

【相关数据类型及接口】

無

mipi_dev_attr_s

【说明】

MIPI-CSI 設備屬性。

【定义】

```
struct mipi_dev_attr_s {
```

```
    enum raw_data_type_e        raw_data_type;
    short                       lane_id[MIPI_LANE_NUM+1];
    enum mipi_wdr_mode_e        wdr_mode;
    short                       data_type[WDR_VC_NUM];
    char                        pn_swap[MIPI_LANE_NUM+1];
};
```

【成員】

成員名稱	描述
wdr_mode	WDR 模式
raw_data_type	傳輸的數據類型
lane_id	發送端(sensor)和接收端(MIPI Rx) lane 的對應關係
data_type	當 WDR 模式為 CVI_MIPI_WDR_MODE_DT 時，每個 WDR frames 對應的 data type
pn_swap	Positive/negative line 是否交換

【芯片差异】

無

【注意事项】

無

【相关数据类型及接口】

無

manaul_wdr_attr_s

【说明】

手動 WDR 模式參數。

【定义】

```
struct manaul_wdr_attr_s {
    unsigned int        manual_en;
    unsigned short      l2s_distance;
    unsigned short      lsef_length;
    unsigned int        discard_padding_lines;
    unsigned int        update;
};
```

【成員】

成員名稱	描述
manual_en	手動 WDR 開關
l2s_distance	一幀中首條長曝到首列短曝的距離, 單位為行數。
lsef_length	長曝/短曝的行數。
discard_padding_lines	Sensor 是否有把 padding 行當有效行輸出。
update	是否強制更新設定。 若否, 設定會等下一張的同步訊號來才更新。

【芯片差异】

無

【注意事项】

- HiSPi 輸入並且 sync_type 為 LVDS_VSYNC_SHARE 時，需打開手動 WDR 模式並設定參數。
- MIPI-CSI 輸入並且 wdr mode 為 CVI_MIPI_WDR_MODE_MANUAL 時，需打開手動 WDR 模式並設定參數。

【相关数据类型及接口】

無

ttl_pin_func_e

【说明】

TTL/BT 接口的配置功能.

【定义】

```
enum ttl_pin_func_e {
        TTL_PIN_FUNC_VS,
        TTL_PIN_FUNC_HS,
        TTL_PIN_FUNC_VDE,
        TTL_PIN_FUNC_HDE,
        TTL_PIN_FUNC_D0,
        TTL_PIN_FUNC_D1,
        TTL_PIN_FUNC_D2,
        TTL_PIN_FUNC_D3,
        TTL_PIN_FUNC_D4,
```

```
        TTL_PIN_FUNC_D5,
        TTL_PIN_FUNC_D6,
        TTL_PIN_FUNC_D7,
        TTL_PIN_FUNC_D8,
        TTL_PIN_FUNC_D9,
        TTL_PIN_FUNC_D10,
        TTL_PIN_FUNC_D11,
        TTL_PIN_FUNC_D12,
        TTL_PIN_FUNC_D13,
        TTL_PIN_FUNC_D14,
        TTL_PIN_FUNC_D15,
        TTL_PIN_FUNC_NUM,
};
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

無

ttl_src_e

【说明】

TTL/BT 接口輸入來源.

【定义】

```
enum ttl_src_e {
        TTL_VI_SRC_VI0 = 0,
        TTL_VI_SRC_VI1,
        TTL_VI_SRC_VI2,          /* BT demux */
        TTL_VI_SRC_NUM
};
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

参考表 1-1.

bt_demux_mode_e

【说明】

BT 解多工模式的通道數量.

【定义】

```
enum bt_demux_mode_e {
        BT_DEMUX_DISABLE = 0,
        BT_DEMUX_2,
        BT_DEMUX_3,
        BT_DEMUX_4,
};
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

無

bt_demux_sync_s

【说明】

BT 解多工模式的同步碼設定。

【定义】

```
struct bt_demux_sync_s {
        unsigned char       sav_vld;
        unsigned char       sav_blk;
        unsigned char       eav_vld;
        unsigned char       eav_blk;
};
```

【成員】

成員名稱	描述
sav_vld	有效資料區間的 SAV
sav_blk	空白區間的 SAV

eav_vld	有效資料區間的 EAV
eav_blk	空白區間的 EAV

【芯片差异】

無

【注意事项】

無

【相关数据类型及接口】

無

bt_demux_attr_s

【说明】

BT 解多工模式的屬性設定。

【定义】

```
struct bt_demux_attr_s {
        signed char                     func[TTL_PIN_FUNC_NUM];
        unsigned short                  v_fp;
        unsigned short                  h_fp;
        unsigned short                  v_bp;
        unsigned short                  h_bp;
        enum bt_demux_mode_e                mode;
        unsigned char                   sync_code_part_A[3];    /* sync
code 0~2 */
        struct bt_demux_sync_s
sync_code_part_B[BT_DEMUX_NUM]; /* sync code 3 */
        char                            yc_exchg;
};
```

【成員】

成員名稱	描述
func	BT 接口對應輸入源 TTL_VI_SRC_VI2 的 lane number. Index 為 BT 邏輯功能, value 請參考表 1-1. 例如, func[TTL_PIN_FUNC_D0] = 5, 代表 BT 訊號的 D0 接到 VI2_D[5], 即 pad name VIVO_D5.

v_fp	垂直方向的 front porch
h_fp	水平方向的 front porch
v_bp	垂直方向的 back porch
h_bp	水平方向的 back porch
mode	BT 解多工模式的通道數量
sync_code_part_A	BT 解多工模式的 0~2 同步碼
sync_code_part_B	BT 解多工模式的同步碼 3
yc_exchg	BIT0~BIT3 分別代表 CH0~CH3 的 Y Cb/Cr bytes 順序互換. 1 為互換, 0 為不互換.

【芯片差异】

無

【注意事项】

無

【相关数据类型及接口】

無

ttl_dev_attr_s

【说明】

TTL/BT 接口的屬性設定。

【定义】

```
struct ttl_dev_attr_s {
        enum ttl_src_e          vi;
        signed char             func[TTL_PIN_FUNC_NUM];
        unsigned short          v_bp;
        unsigned short          h_bp;
};
```

【成員】

成員名稱	描述
vi	TTL/BT 接口的輸入來源, 允許值為 TTL_VI_SRC_VI0 或 TTL_VI_SRC_VI1
func	BT 接口對應輸入源的 lane number. Index 為 BT 邏輯功能, value 請參考表 1-1. 例如, vi = TTL_VI_SRC_VI1

	時, func[TTL_PIN_FUNC_D0] = 5, 代表 BT 訊號的 D0 接到 VI1_D[5], 即 pad name VIVO_D5.
v_bp	垂直方向的 back porch
h_bp	水平方向的 back porch

【芯片差异】

無

【注意事项】

無

【相关数据类型及接口】

無

combo_dev_attr_s

【说明】

combo 設備屬性，由於 MIPI Rx 能夠對接 CSI-2，sub-LVDS，HiSPi 等時序，所以將 MIPI Rx 稱為 combo 設備。

【定义】

```
struct combo_dev_attr_s {
        enum input_mode_e           input_mode;
        enum rx_mac_clk_e           mac_clk;
        struct mclk_pll_s           mclk;
        union {
                struct mipi_dev_attr_s   mipi_attr;
                struct lvds_dev_attr_s   lvds_attr;
                struct ttl_dev_attr_s    ttl_attr;
                struct bt_demux_attr_s   bt_demux_attr;
        };
        unsigned int                devno;
        struct img_size_s           img_size;
        struct manaul_wdr_attr_s    wdr_manu;
};
```

【成員】

成員名稱	描述
input_mode	輸入接口類型

mac_clk	MIPI RX MAC 時鐘設定
mclk	MIPI RX 輸出的 Sensor 參考時鐘設定
mipi_attr	如果 input_mode 配置為 INPUT_MODE_MIPI，則必須配置 mipi_attr
lvds_attr	如果 input_mode 配置為 INPUT_MODE_SUBLVDS/INPUT_MODE_HISPI，則必須配置 lvds_attr
devno	MIPI-Rx 設備號
img_size	輸入幀大小
wdr_manu	手動 WDR 屬性

【芯片差异】

無

【注意事项】

無

【相关数据类型及接口】

無

crop_top_s

【说明】

捨棄開頭的行資料。

【定义】

```
struct crop_top_s {
        unsigned int        devno;
        unsigned int        crop_top;
        unsigned int        update;
};
```

【成員】

成員名稱	描述
devno	MIPI-Rx 設備號
crop_top	開頭要捨棄的行數
update	是否強制更新設定。 若否，設定會等下一張的同步訊號來才更新。

【芯片差异】

無

【注意事项】

無

【相关数据类型及接口】

無

manual_wdr_s

【说明】

手動 WDR 模式設定。

【定义】

```
struct manual_wdr_s {
        unsigned int                devno;
        struct manaul_wdr_attr_s    attr;
};
```

【成員】

成員名稱	描述
devno	MIPI-Rx 設備號
attr	手動 WDR 屬性

【芯片差异】

無

【注意事项】

無

【相关数据类型及接口】

無

vsync_gen_s

【说明】

手動 WDR 模式設定。

【定义】

```
struct vsync_gen_s {
        unsigned int                devno;
```

```
    unsigned int                distance_fp;
};
```

【成員】

成員名稱	描述
devno	MIPI-Rx 設備號
distance_fp	當 input_mode 為 INPUT_MODE_SUBLVDS 時，產生垂直同步訊號的時間點。

【芯片差异】

無

【注意事项】

- Sub-LVDS 不自帶垂直同步訊號，所以 MIPI-Rx 須自行生成送給 ISP。distance_fp 可調整垂直同步時間點以達到加大 front porch 的作用。

【相关数据类型及接口】

無

LANE_MAX_NUM

【说明】

一個 MIPI Tx 設備支援的最大 Lane 數。

【定义】

```
#define MIPI_LANE_NUM    4
```

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

無

output_mode_e

【说明】

MIPI Tx 輸出模式。

【定义】

```
enum output_mode_e
```

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

【芯片差异】

芯片类型	是否支持
CV183x	只支持 OUTPUT_MODE_DSI_VIDEO
CV182x	只支持 OUTPUT_MODE_DSI_VIDEO

【注意事项】

无。

【相关数据类型及接口】

無

video_mode_e

【说明】

MIPI Tx 视频模式。

【定义】

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

【芯片差异】

芯片类型	是否支持
CV183x	只支持 BURST_MODE
CV182x	只支持 BURST_MODE

【注意事项】

无。

【相关数据类型及接口】

無

output_format_e

【说明】

MIPI Tx 輸出格式。

【定义】

```
enum output_format_e {
        OUT_FORMAT_RGB_16_BIT               = 0x0,
        OUT_FORMAT_RGB_18_BIT               = 0x1,
        OUT_FORMAT_RGB_24_BIT               = 0x2,
        OUT_FORMAT_RGB_30_BIT               = 0x3,
        OUT_FORMAT_YUV420_8_BIT_NORMAL = 0x4,
        OUT_FORMAT_YUV420_8_BIT_LEGACY = 0x5,
        OUT_FORMAT_YUV422_8_BIT             = 0x6,

        OUT_FORMAT_BUTT
};
```

【芯片差异】

芯片类型	是否支持
CV183x	不支持 YUV420
CV182x	不支持 YUV420

【注意事项】

无。

【相关数据类型及接口】

無

sync_info_s

【说明】

MIPI Tx 設備同步信息。

【定义】

```
struct sync_info_s {
        unsigned short    vid_hsa_pixels;
        unsigned short    vid_hbp_pixels;
        unsigned short    vid_hfp_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;
    bool              vid_vsa_pos_polarity;
    bool              vid_hsa_pos_polarity;
};
```

【成員】

成員名稱	描述
vid_hsa_pixels	Horizontal sync-pluse 像素个数
vid_hbp_pixels	Horizontal back-porch 像素个数
vid_hfp_pixels	Horizontal front-porch 像素个数
vid_hline_pixels	Horizontal image active 像素个数
vid_vsa_lines	Vertical sync-pluse 行数
vid_hbp_pixels	Vertical back-porch 行数
vid_hbp_pixels	Vertical front-porch 行数
vid_active_pixels	Vertical image active 行数
edpi_cmd_size	写内存命令字节数。 video mode 时该值无效，command mode 时该值设为 hact。
vid_vsa_pos_polarity;	Horizontal sync-pluse polarity
vid_hsa_pos_polarity;	Vertical sync-pluse polarity

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

無

combo_dev_cfg_s

【说明】

MIPI Tx 設備屬性。

【定义】

```
struct combo_dev_cfg_s {
    unsigned int                devno;
    enum mipi_tx_lane_id        lane_id[LANE_MAX_NUM];
    enum output_mode_e          output_mode;
    enum video_mode_e           video_mode;
    enum output_format_e        output_format;
    struct sync_info_s          sync_info;
    unsigned int                phy_data_rate;
    unsigned int                pixel_clk;
    bool                        lane_pn_swap[LANE_MAX_NUM];
};
```

【成員】

成員名稱	描述
devno	MIPI Tx 设备号
lane_id	发送端(vo)和接收端(MIPI Tx) Lane的对应关系 未使用的 Lane 设置为-1。
output_mode	MIPI Tx 输出模式。
video_mode	MIPI Tx 视频模式。
output_format	MIPI Tx 输出格式。
sync_info	MIPI Tx 设备的同步信息。
phy_data_rate	MIPI Tx 输出速率，MIPI Tx 高速模式每个通道（lane）的速率范围的描述。
pixel_clk	像素时钟。单位为 KHz
lane_pn_swap	Lane 设置上是否 P/N 要互换

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

CVI_VIP_MIPI_TX_SET_DEV_CFG

cmd_info_s

【说明】

給 MIPI Tx 設備初始化信息。

【定义】

```
struct cmd_info_s {
    unsigned int devno;
    unsigned short data_type;
    unsigned short cmd_size;
    unsigned char *cmd_data;
};
```

【成員】

成员名称	描述
devno	MIPI Tx 设备号
data_type	命令数据类型
cmd_size	命令数据字节数，范围：(0,128)。
cmd_data	命令数据地址指针，需要用户分配。

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

CVI_VIP_MIPI_TX_SET_CMD

get_cmd_info_t

【说明】

從 MIPI Tx 設備取回信息。

【定义】

```
struct get_cmd_info_s {
    unsigned int devno;
    unsigned short data_type;
    unsigned short data_param;
    unsigned short get_data_size;
```

```
    unsigned char *get_data;
} get_cmd_info_t;
```

【成員】

成员名称	描述
devno	MIPI Tx 设备号
data_type	命令数据类型
data_param	数据参数，低八比特为第一个参数，高八比特为第二个参数、不用时填 0
get_data_size	预期获取的数据字节数，范围：(0,4)。
get_data	获取到的数据存放地址指针，需要用户分配。

【芯片差异】

无。

【注意事项】

无。

【相关数据类型及接口】

CVI_VIP_MIPI_TX_GET_CMD

hs_settle_s

【说明】

MIPI Tx 設備 HS mode 下的 settle 信息。

【定义】

```
struct hs_settle_s {
    unsigned char       prepare;
    unsigned char       zero;
    unsigned char       trail;
};
```

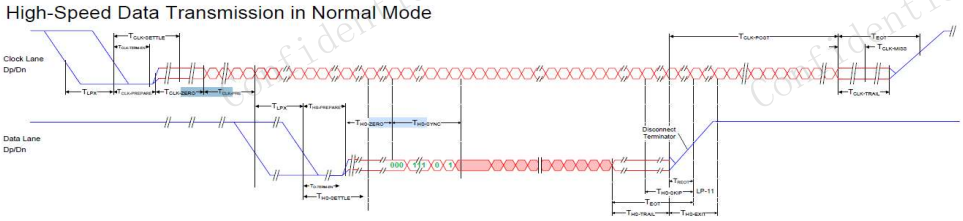
【成員】

成员名称	描述
prepare	為 LP->HS 後的前置準備 T 數
zero	在進入 HS 後，在輸出資料前為 0 的 T 數
trail	在 HS->LP 時，要結束 HS 前，資料結束後額外的 T 數

【芯片差异】

无。

【注意事项】



。

【相关数据类型及接口】

無

1.6. Proc 信息

1.6.1. MIPI_RX Proc 信息

MIPI_Rx 在正常工作下 proc 信息中的各種錯誤中斷計數應為 0。 若否，請檢查 MIPI_Rx 相關屬性是否配置正確。

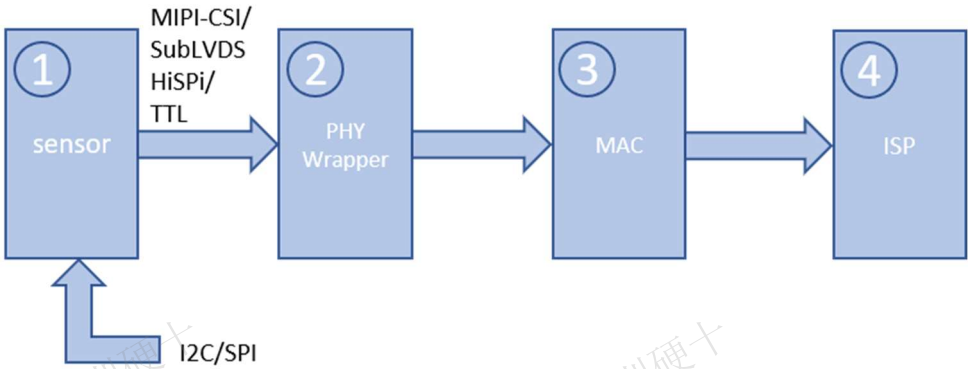
【調試信息】

```
Module: [MIPI_RX], Build Time[#1 SMP PREEMPT Thu Apr 29 11:18:57 CST 2021]

------------Combo DEV ATTR--------------
    Devno   WorkMode   DataType     WDRMode            LinkId          PN
Swap   SyncMode   DataEndian   SyncCodeEndian
        0       MIPI       RAW12         NONE   2, 3, 1, 4, 0   1, 1, 1, 1, 1
N/A            N/A                 N/A

------------MIPI info-------------------
 Devno EccErr CrcErr HdrErr WcErr fifofull     decode
      0       0       0       0       2          0      raw12
  Physical:         D0         D1         D2         D3         D4         D5
                     0          0          0          0          0          0
    Digital:         D0         D1         D2         D3      CK_HS    CK_ULPS
CK_STOP     CK_ERR     Deskew
                 hs_idle    hs_idle    hs_idle    hs_idle          1          0          0
0       idle
```

【調試信息分析】

- MIPI Rx 通過 PHY Wrapper 接收 Sensor 的 MIPI-CSI/SubLVDS/HiSPi/TTL 訊號，由 MAC 內對應的介面進行同步頭檢測與對齊。
- MAC 將各 Lane 的數據合併成 Pixel 數據，並將數據發送給後級的 ISP。
- PHY Wrapper 由 sensor 的 pixel clock 提供時鐘。 MAC 內的時鐘與後級的 ISP 相同。
- 若要操作數據的 Crop，可由後級的 ISP 來調試。

圖 1-5 數據示例圖



【參數說明】

參數		描述
MIPI DEV ATTR	Devno	MIPI 設備號
	WorkMode	MIPI 設備工作模式: MIPI/ SUBLVDS/HISPI/CMOS 等模式
	DataType	RAW8/RAW10/RAW12 等類型
	WDRMode	WDR 模式: • NONE • VC • DT • DOL • MANUAL
	LaneId	Lane id
	PN Swap	PN 訊號交換
LVDS DEV	Devno	MIPI 設備號
	WorkMode	MIPI 設備工作模式: MIPI/

ATTR		SUBLVDS/HISPI/CMOS 等模式
	DataType	RAW8/RAW10/RAW12 等類型
	WDRMode	WDR 模式: • NONE • 2To1 • 3To1 • DOL2To1 • DOL3To1
	LaneId	Lane id
	PN Swap	PN 訊號交換
	SyncMode	LVDS 的同步碼模式: • SOF • SAV
	DataEndian	Data 的比特位大小端模式
	SyncCodeEndian	同步碼的比特位大小端模式
MIPI Info (僅 MIPI 模式可見)	Devno	MIPI 設備號
	EccErr	ECC 錯誤的中斷計數
	CrcErr	CRC 錯誤的中斷計數
	HdrErr	HDR Flag 錯誤的中斷計數
	WcErr	Word Count 錯誤的中斷計數
	fifofull	Fifofull 的中斷計數
	Physical: D0	MIPIRX_PAD0 收到的資料
	Physical: D1	MIPIRX_PAD1 收到的資料
	Physical: D2	MIPIRX_PAD2 收到的資料
	Physical: D3	MIPIRX_PAD3 收到的資料
	Physical: D4	MIPIRX_PAD4 收到的資料
	Physical: D5	MIPIRX_PAD5 收到的資料
	Digital: D0	Sensor data lane 0 state
	Digital: D1	Sensor data lane 1 state
	Digital: D2	Sensor data lane 2 state
	Digital: D3	Sensor data lane 3 state
	CK_HS/CK_ULPS/ CK_STOP/CK_ERR	Sensor clock lane state
	Deskew	Deskew 結果

1.7.FAQ

1.7.1. Land id 如何配置

Land id 的配置對應 mipi_dev_attr_s 中的 short lane_id[MIPI_LANE_NUM+1]或者 lvds_dev_attr_s 中的 short lane_id[MIPI_LANE_NUM+1]，其中 lane_id 數組的索引號表示的是 Sensor 的 Lane ID，索引號 0 表示 sensor clock，索引號 1 表示 sensor lane 0。 land_id 數組的值表示的是 MIPI-Rx 的 Lane ID， 0 表示 MIPIRX1_PAD0，1 表示 MIPIRX1_PAD1。未使用的 lane 將其對應的 lane_id 配置為-1。
下面舉例說明，例如 MIPI 與 SENSOR 的引腳硬件連接如下表所示。

SENSOR 管腳	MIPI Lane 管腳
Clock Lane (index = 0)	MIPIRX1_PAD0 (value = 0)
Lane 0 (index = 1)	MIPIRX1_PAD1 (value = 1)
Lane 1 (index = 2)	MIPIRX1_PAD2 (value = 2)
Lane 2 (index = 3)	MIPIRX1_PAD3 (value = 3)
Lane 3 (index = 4)	MIPIRX1_PAD4 (value = 4)

MIPI 的最大 Lane 數加上 Clock 為 5，所以 lane_id 配置如下:

```
//索引 sensor_clk,    sensor_lane0,   sensor_lane1,   sensor_lane2,     sensor_lane3
//
land_id={MIPIRX1_PAD0  MIPIRX1_PAD1  MIPIRX1_PAD2  MIPIRX1_PAD3
MIPIRX1_PAD4 }
```

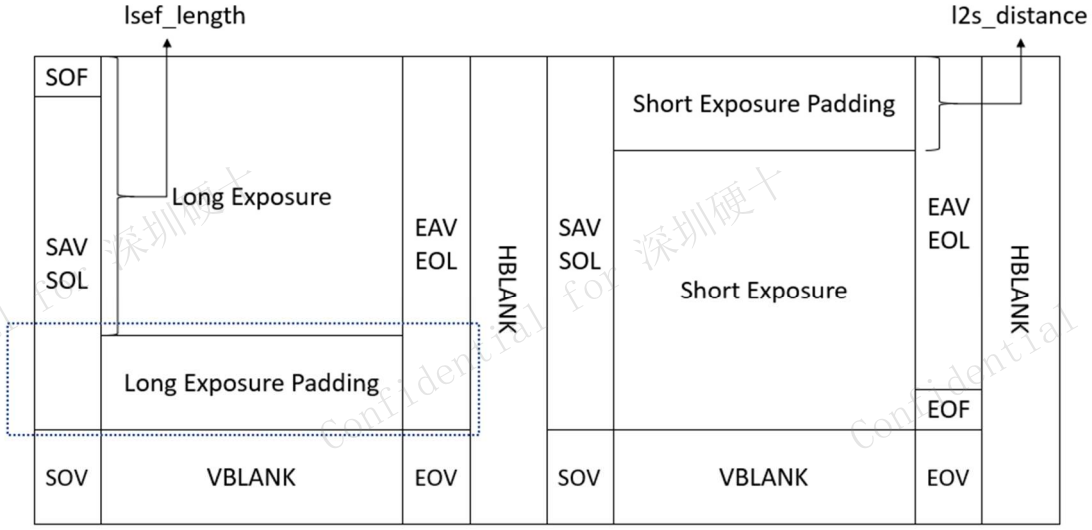
1.7.2. MIPI 頻率說明

使用以下公式計算 MIPI 每 Lane 最高頻率與 VI MAC 的工作頻率:
MAC_Freq * pixel_width = lane_num * MIPI_Freq * 2。
其中 MAC_Freq 為 VI MAC 的工作頻率，pixel_width 為像素位寬，lane_num 為 MIPI lane 個數，MIPI_Freq 為每條 lane 的工作頻率。
若 MAC clock 為 400M，pixel_width = 12，lane_num = 4，
可支持最快 MIPI_Freq = 400 * 12 / (4 * 2) = 600MHz。

1.7.3. Manual WDR 模式使用說明

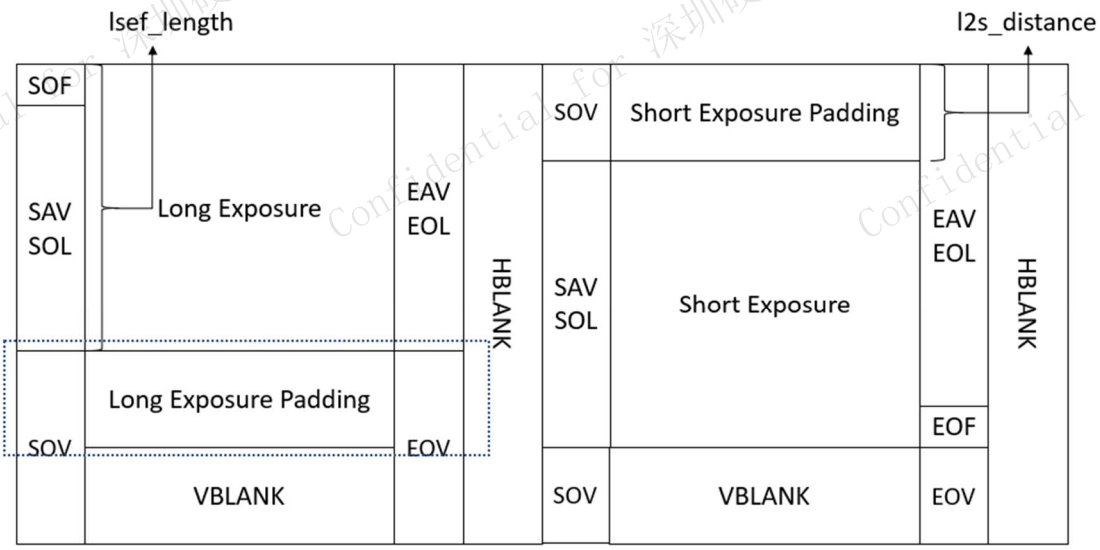
當手動 WDR 模式打開後，MIPI-Rx 會把收下來的資料以行為單位，遵遁以下規則分配給長曝幀與短曝幀。

- l2s_distance: 從第一行到第 l2s_distance 行都是長曝資料，第 l2s_distance+1 行開始長曝與短曝交錯分配。
- lsef_length: 第 lsef_length+1 行開始都是短曝幀資料。 直到下個垂直同步訊號為止。
- 當 discard_padding_lines=1 時，1 到 l2s_distance 行分配方式為{長-ignore-長-ignore ...}，第 l2s_distance+1 到 lsef_length 行分配方式為{長-短-長-短 ...}。第 lsef_length+1 行到下個垂直同步訊號分配方式為{短-ignore-短-ignore ...}。



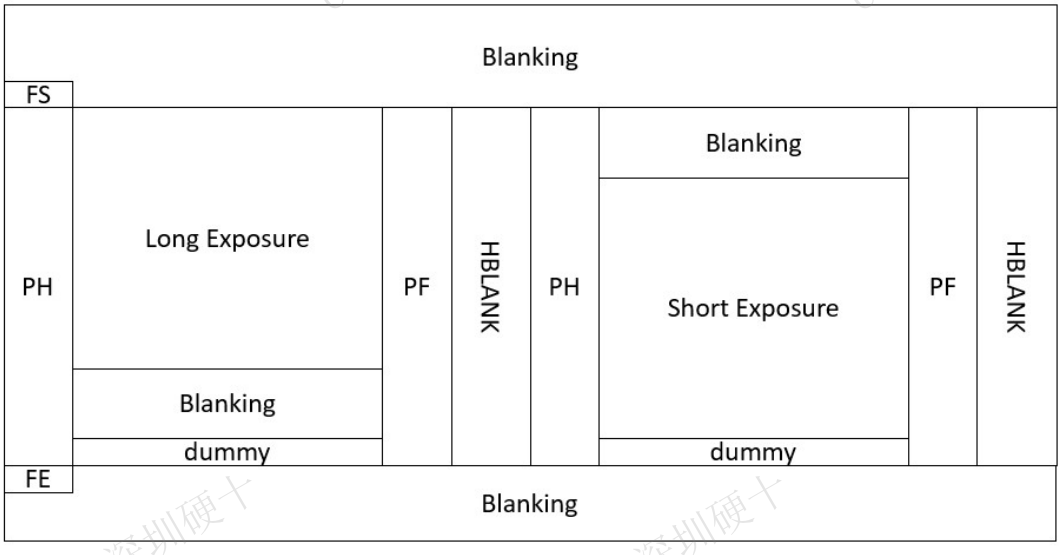
Padding data is sent as active lines, discard_padding_lines = 1

- 當 discard_padding_lines=0 時，1 到 l2s_distance 行分配方式為{長-長-長 ...}，第 l2s_distance+1 到 lsef_length 行分配方式為{長-短-長-短 ...}。第 lsef_length+1 行到下個垂直同步訊號分配方式為{短-短-短 ...}。



Padding data is sent as blanking lines, discard_padding_lines = 0

- MIPI-Rx 必須確保發送給 ISP 的長曝幀與短曝幀行數是一致的。
- 調整 sensor 短曝長度，有可能 l2s_distance 須要一起調整。
- 有些 sensor 可能會在送完短曝有效資料後帶 dummy 行。 這會造成長曝與短曝的行數不一致。 可將 l2s_distance 設成 0，lsef_length 設成最大值 0x1FFF，discard_padding_lines 為 1 即收下兩張帶有效與 dummy 資料的長短曝，再用 ISP crop 有效位置即可。



discard_padding_lines = 1
l2s_distance = 0
lsef_length = 1FFF