NXP Semiconductors
Application Note

Document Number: AN5305
Rev. 0, 07/2016

MIPI–CSI2 Peripheral on i.MX6 MPUs

Contents

1. Introduction

1.1. Purpose

The purpose of this application note is to provide detailed information about the MIPI–CSI2 peripheral on the i.MX6 family of processors with a usage example and details.

1.2. Scope

This application note applies to these i.MX processors:

- i.MX6QuadPlus (i.MX6QP)
- i.MX6DualPlus (i.MX6DP)
- i.MX6Quad (i.MX6Q)
- i.MX6Dual (i.MX6D)
- i.MX6DualLite (i.MX6DL)
- i.MX6Solo (i.MX6S)

These devices are referred to by their abbreviated names throughout this document (shown above in parentheses).

1.3. Audience

This document is intended for those who:

- Need more information about the MIPI-CSI2 peripheral and its usage.
- Need to implement or debug a driver to capture still or moving images by the i.MX6 family processors using the MIPI-CSI2 interface.

1.4. Definitions, Acronyms, and Abbreviations

The definitions of the terms and acronyms used in this document are:

- IPU—image processing unit
- CSI—camera sensor interface
- MIPI®—mobile industry processor interface (a global organization that develops interface specifications for the mobile ecosystem including mobile-influenced industries)
- D-PHY—one of the physical-layer interfaces developed by MIPI®; designed to interface cameras and displays with low-power differential signals

1.5. References

- i.MX6 reference manuals
- i.MX6 datasheets
- *Chip Errata for the i.MX 6Dual/6Quad and i.MX 6DualPlus/6QuadPlus* (document IMX6DQCE)
- MIPI Alliance Standard for Camera Serial Interface 2 (CSI2)—MIPI Board Approved 11/29/2005
- MIPI Alliance Specification for Camera Serial Interface 2 (CSI-2)

2. Overview

The MIPI block has four data lanes (four differential pairs) on i.MX6QP, i.MX6Q, and i.MX6D, two data lanes on i.MX6DL and i.MX6S, and one clock differential pair in all processors.

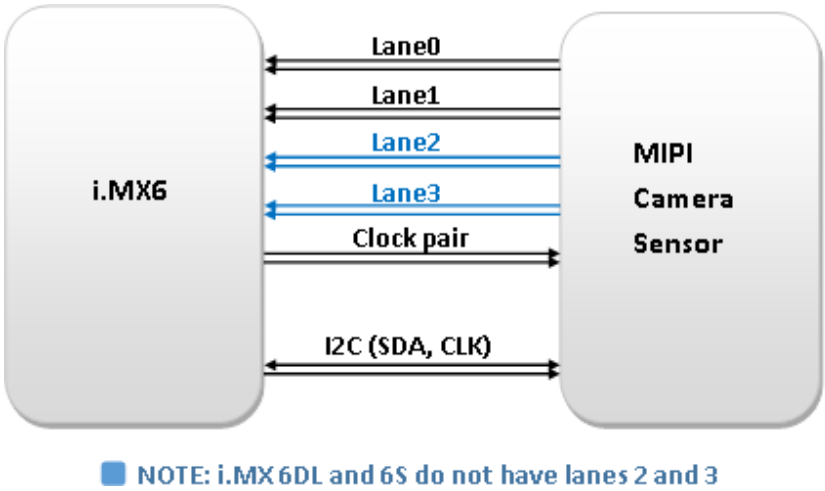

Figure 1. i.MX6 lanes

The number of lanes determines the bandwidth of the MIPI bus. Each lane can transfer up to 1000 Mb/s or up to 800 Mb/s when all four lanes are used. For more details, see Section 2.3.1 "Bandwidth".

The I^2C bus is required to configure most of the camera sensors.

2.1. Routing MIPI stream into CSI-2

The i.MX6 processors have one MIPI/CSI-2 input and two parallel input interfaces (parallel 0 and parallel 1; see Figure 2). The streams in the MIPI format pass through the MIPI/CSI receiver, the CSI/IPU gasket, and a mux.

On the i.MX6 ICs that have two IPUs, up to four streams can be received on the same MIPI bus. The CSI/IPU gasket can receive up to four different streams with different VCs (virtual channels) and route each stream to a specific CSI port (see Figure 2 and Figure 3). Each CSI port has a specific virtual channel number and this configuration can't be changed. For example, for i.MX6Q, VC0 is assigned to CSI0/IPU1, VC1 is assigned to CSI1/IPU1, and so on.

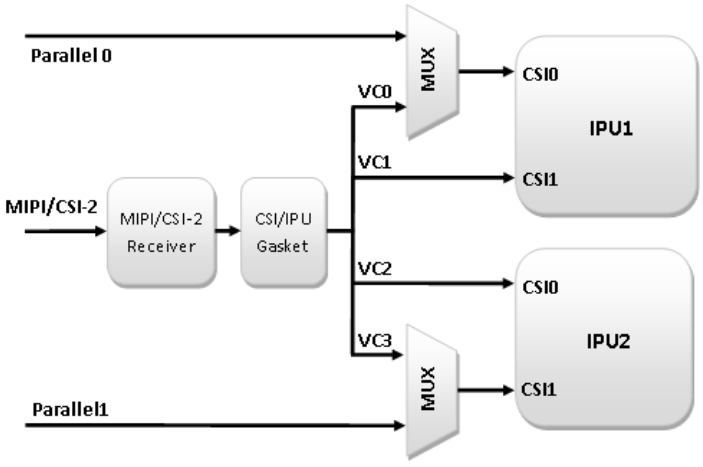

Figure 2. Input video routing for i.MX6QP, i.MX6Q, and i.MX6D

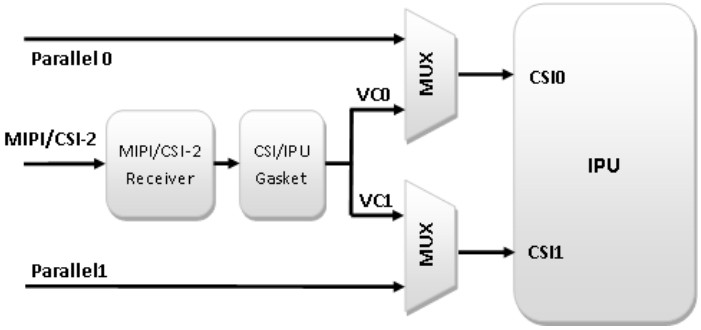

Figure 3. Input video routing for i.MX6DL and i.MX6S

2.2. MIPI signal

CSI-2 uses the MIPI standard for the D-PHY physical layer. This document provides an overview of the MIPI signal format. For more information about the MIPI specification, see MIPI Alliance Standard for Camera Serial Interface 2 documentation at mipi.org.

2.2.1. Lanes

CSI-2 is a lane-scalable specification. The applications that require more bandwidth than what is provided by one data lane or those trying to avoid high clock rates can expand the data path to two, three, or four lanes and obtain approximately linear increases in the peak bus bandwidth. The data stream is distributed between the lanes. This figure shows an example of a 4-lane transmission:

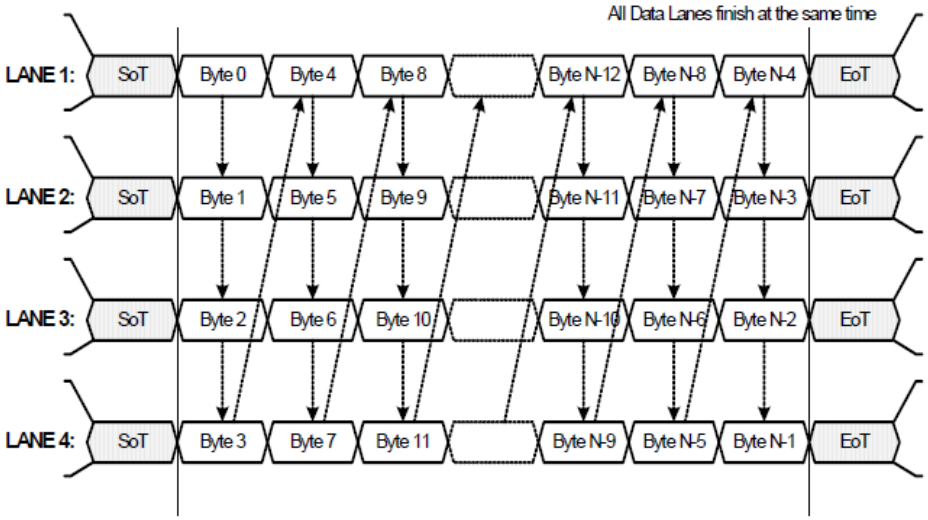

Figure 4. 4-lane data stream

2.2.2. Low-level protocol (LLP)

LLP is a byte-oriented, packet-based protocol which supports the transport of arbitrary data using the short and long packet formats.

Two packet structures are defined for the LLP communication: long packets and short packets. For each packet structure, the exit from the low-power state followed by the start of transmission (SoT) sequence indicates the start of a packet. The end of transmission (EoT) sequence followed by the low-power state indicates the end of a packet.

LLP features:

- Transport of arbitrary data (payload-independent)
- 8-bit word size
- Support for up to four interleaved virtual channels on the same link
- Special packets for the frame start, frame end, line start, and line end information
- Descriptor for the type, pixel depth, and format of the application-specific payload data
- 16-bit checksum code for error detection

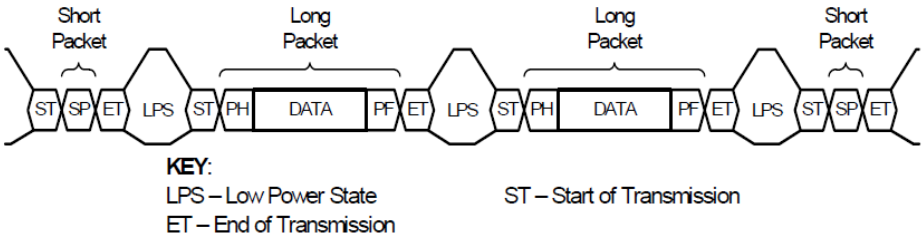

Figure 5. LLP format

In the above figure, PH represents the packet header and PF represents the packet footer.

2.2.2.1. Long packets

The following figure shows the structure of the LLP long packet. The long packet is identified by data types ranging from 0x10 to 0x37. See Table 1 for a description of data types. The long packet consists of three elements: a 32-bit packet header, an application-specific data payload with a variable number of 8-bit data words, and a 16-bit packet footer. The packet header is further composed of three elements: an 8-bit data identifier, a 16-bit word count field, and an 8-bit ECC. The packet footer has one element (a 16-bit checksum).

Figure 6. LLP long packet structure

Packet header = Data ID + Word count + ECC

Packet footer = 16-bit checksum

In Figure 6:

LPS—low-power state

SoT—start of transmission

Data ID—contains the virtual channel identifier and the data type information

WC—word count—the receiver uses the WC value to determine the “end of the packet”

ECC—8-bit ECC code for the packet header

Data—application-specific payload

EoT—end of transmission

2.2.2.2. Short packets

The following figure shows the structure of the LLP short packet. The short packet must be identified by data types ranging from 0x00 to 0x0F. See Table 1 for a description of data types. The short packet must contain only the packet header; the packet footer must not be present. The word-count field in the packet header must be replaced by the short-packet data field.

For frame synchronization data types, the short-packet data field must be the frame number. For the line synchronization data types, the short-packet data field must be the line number. See Table 1 for a description of the frame and line synchronization data types.

For the generic short-packet data types, the content of the short-packet data field must be user-defined.

The error correction code (ECC) byte allows for the single-bit errors to be corrected and for the 2-bit errors to be detected in the short packet.

Figure 7. LLP short packet structure

2.2.2.3. Data identifier and virtual channel

The data identifier byte contains the virtual channel identifier (VC) value and the data type (DT) value, as shown in Figure 8. The virtual channel identifier is contained in two MS bits of the data-identifier byte. The data type value is contained in six LS bits of the data-identifier byte.

The purpose of the virtual channel identifier is to provide separate channels for different data flows that are interleaved in the data stream.

The virtual channel identifier number is in the top two bits of the data-identifier byte. The receiver monitors the virtual channel identifier and de-multiplexes the interleaved video streams to their appropriate channel. A maximum of four data streams is supported; the valid channel identifiers range from 0 to 3. The virtual channel identifiers in the peripherals must be programmable to enable the host processor to control how the data streams are de-multiplexed.

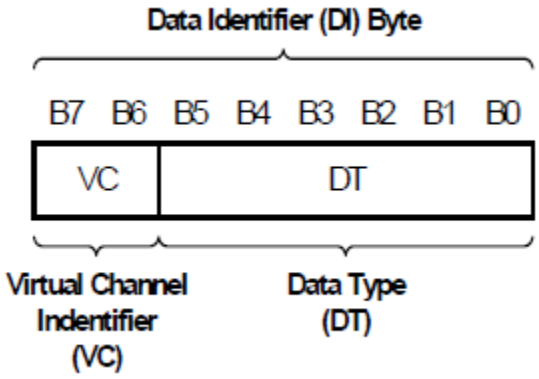

Figure 8. Data identifier byte structure

2.2.2.4. Data type

The data type value specifies the format and content of the payload data. A maximum of 64 data types is supported.

There are eight different data type classes, as shown in the following table. Within each class, there are up to eight different data type definitions. The first two classes denote the short-packet data types. The remaining six classes denote the long-packet data types.

Table 1. Data type classes

Data type	Description
0x00–0x07	Synchronization short-packet data types
0x08–0x0F	Generic short-packet data types
0x10–0x17	Generic long-packet data types
0x18–0x1F	YUV data
0x20–0x27	RGB data
0x28–0x2F	Raw data
0x30–0x37	User-defined byte-based data
0x38–0x3F	Reserved

For more information about the data types, see MIPI Alliance Specification for Camera Serial Interface 2 (CSI-2).

2.2.2.5. Interleaved video data streams

Multiple data streams with different formats can be transferred by the MIPI bus. Each video stream must have one virtual channel assigned, as shown in this figure:

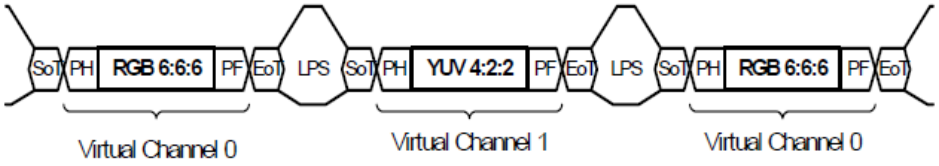

Figure 9. Multiple color format data stream

2.3. MIPI capabilities

This table lists the maximum data lanes, simultaneous streams, and bit rate for various i.MX devices:

Table 2. MIPI configuration limits for i.MX family members

—	i.MX6QP/i.MX6DP	i.MX6Q/i.MX6D	i.MX6DL/i.MX6S
Data lanes	4	4	2
Max. simultaneous streams	4	4	2
Max. bit rate	3200 Mb/s	3200 Mb/s	2000 Mb/s

2.3.1. Bandwidth

The MIPI operating frequency is set by selecting the MIPI-CSI2 clock source on the CCM (clock controller module). See the CCM section in the i.MX6 reference manual.

On i.MX6QP, i.MX6DP, i.MX6Q, and i.MX6D, the operating frequency is MIPI_PIXEL_CLK and its value can be changed without interfering with the other blocks.

On i.MX6DL and i.MX6S, the MIPI-CSI2 clock source is CCM_PIXEL_CLK and it is connected to the IPU clock.

The maximum MIPI-CSI2 frequency is 200 MHz.

The required minimum operating frequency of the interface is calculated as:

$$F = FH * FW * FPS * BI * DF$$

Where:

- FH—frame height (in pixels)
- FW—frame width (in pixels)
- FPS—frame rate (frames per second)
- DF—data format; defines the number of cycles needed to send a single pixel
- BI—blank interval; a 35 % (1.35) overhead provides a safe estimate for the minimum frequency

In case the video mode data is available, the minimum operating frequency can be also calculated as:

$$F = TFH * TFW * FPS * DF$$

where:

- TFH—total frame height = front porch + vsync + back porch
- TFW—total frame width = front porch + hsync + back porch
- FPS—frame rate (frames per second)
- DF—data format; defines the number of cycles needed to send a single pixel

The number of cycles needed to send a single pixel depends on the interface and the data format. The data bus is 16 bits wide. The data format examples are:

- YUV422—1 cycle/pixel
- RGB888—1.5 cycles/pixel

- Generic data—2 bytes/pixel

The maximum bandwidth of the interface is:

- 200 MHz for a 4-lane configuration (800 Mb/s/lane, 400 MB/s)
- 187.5 MHz for a 3-lane configuration (1000 Mb/s/lane, 375 MB/s)
- 125 MHz for a 2-lane configuration (1000 Mb/s/lane, 250 MB/s)
- 62.5 MHz for a 1-lane configuration (1000 Mb/s/lane, 125 MB/s)

When multiple virtual channels are used to stream more than one video stream at the same time, the total bandwidth is the sum of all streams.

The examples of single-stream interfaces are:

- 3.2-MP (mega-pixel) camera, 2-lane configuration, 15 fps, YUV422 format:

3.2 MP * 15 fps * 1 cycle/pixel * 1.35 blank interval = 64.8 MHz

- 6-MP camera, 4-lane configuration, 15 fps, RGB888 format:

6 MP * 15 fps * 1.5 cycle/pixel * 1.35 blank interval = 182.25 MHz

The examples of multiple virtual channel stream interfaces are:

- Camera 1—virtual channel 0: 1024x768, 4-lane configuration, 30 fps, YUV422 format:

1024 * 768 * 30 fps * 1 cycle/pixel * 1.35 blank interval = 31.85 MHz

- Camera 2—virtual channel 1: 1920x1080, 4-lane configuration, 30 fps, YUV422 format:

1920 * 1080 * 30 fps * 1 cycle/pixel * 1.35 blank interval = 83.98 MHz

Total bandwidth = 31.85 MHz + 83.98 MHz = 115.83 MHz

2.4. System overview

After passing through the MIPI/CSI-2 receiver, the MIPI/CSI-2 gasket, and a multiplexer (see Figure 2), the video signal is received by the CSI-2 block inside the IPU which is responsible for synchronizing and packing the video (or generic data) and sending it to other blocks (see Figure 10).

CSI-2 can send the video signal to three other blocks:

- SMFC—sensor multi FIFO controller; used as an interface between CSI-2 and IDMAC
- VDI—video de-interlace; used to de-interlace a video signal
- IC—image converter; used to process the image, perform the color space conversion, rotation, alpha blending, resizing, and combining

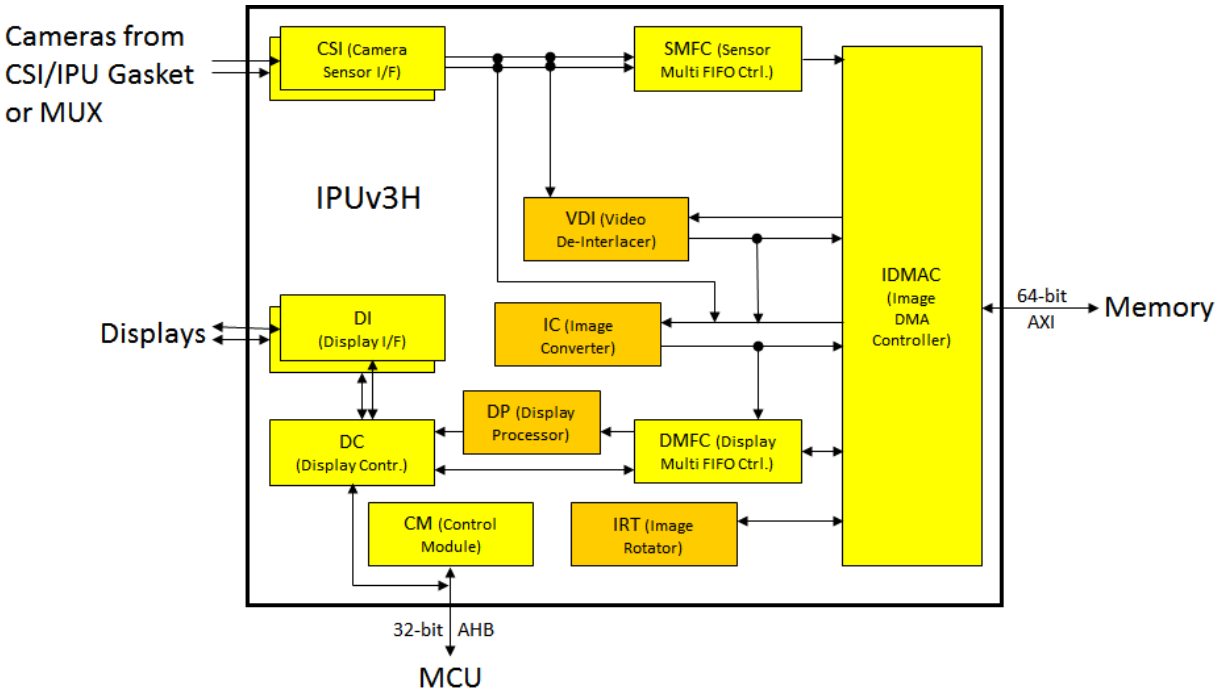

Figure 10. Video system block diagram

3. MIPI—Camera Serial Interface Host Controller and D-PHY Configuration Details

The CSI-2 MIPI interface is a digital core supplied with a multi-lane D-PHY that implements all protocol functions defined in the MIPI CSI-2 specification, providing an interface between the system and the MIPI CSI-2-compliant camera sensor.

The MIPI D-PHY macro interfaces with a CSI-2 controller. The CSI-2 controller is a digital core that implements all protocol functions defined in the MIPI CSI-2 specifications, providing an interface between the system and the MIPI D-PHY, allowing for a communication with a compliant MIPI camera sensor. The I/O block is responsible for interfacing with the analog physical world.

3.1. Block diagram and port configuration

The CSI-2/MIPI and D-PHY blocks are shown in this figure:

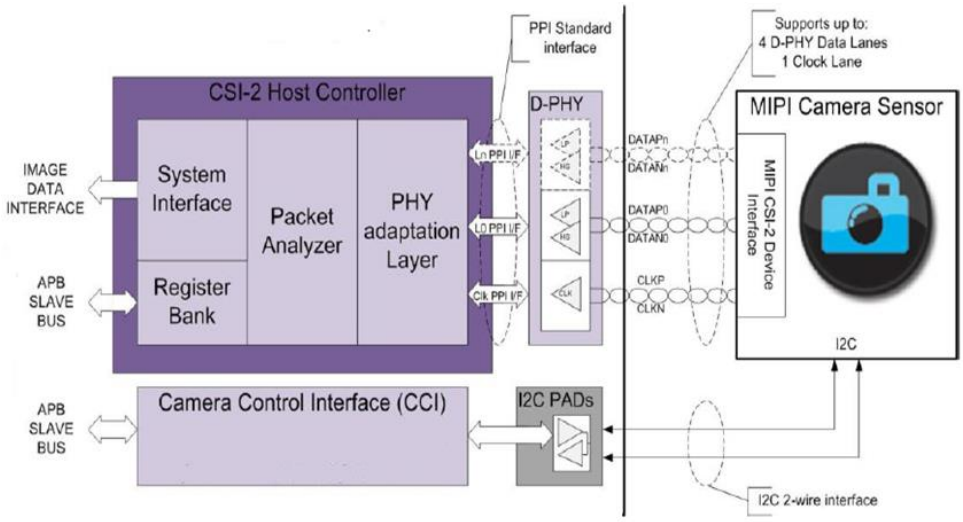

Figure 11. CSI-2/MIPI and D-PHY block diagram

The MIPI pins are not multiplexed with other peripherals and no port configuration is needed.

Table 3. MIPI signal and pad mode mapping

Signal	Pad mode	Description
CSI_CLK0_N	CSI_CLK0M	Negative clock
CSI_CLK0_P	CSI_CLK0P	Positive clock
CSI_DATA0_N	CSI_D0M	Lane 0, Data -
CSI_DATA0_P	CSI_D0P	Lane 0, Data +
CSI_DATA1_N	CSI_D1M	Lane 1, Data -
CSI_DATA1_P	CSI_D1P	Lane 1, Data +
CSI_DATA2_N	CSI_D2M	Lane 2, Data -
CSI_DATA2_P	CSI_D2P	Lane 2, Data +
CSI_DATA3_N	CSI_D3M	Lane 3, Data -
CSI_DATA3_P	CSI_D3P	Lane 4, Data +

3.2. D-PHY registers

A complete list of all D-PHY registers is shown in Appendix A—MIPI-CSI2 and D-PHY Registers.

3.3. MIPI-CSI2 and D-PHY configuration example sequence

To initialize the MIPI-CSI2 and D-PHY, use this sequence:

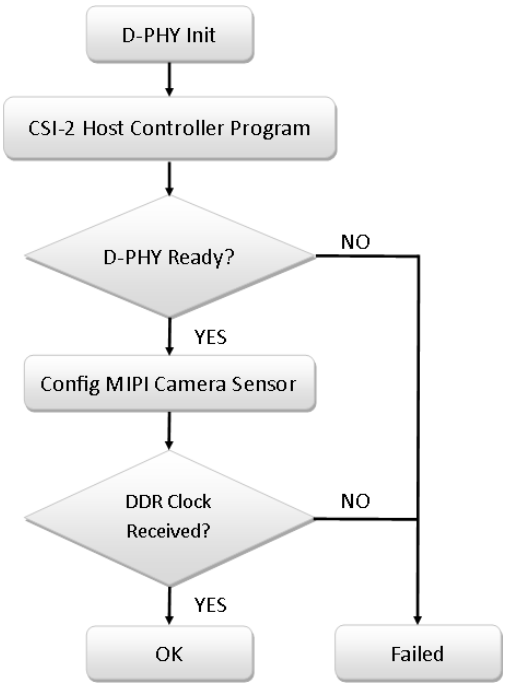

Figure 12. MIPI-CSI2 and D-PHY configuration sequence

1. If required, configure the MIPI camera sensor to have all Tx lanes in the LP-11 state (STOPSTATE).
2. The D-PHY specification states that the D-PHY master must be initialized in the LP-11 state (STOPSTATE). However, a CCI command may be required to switch the MIPI interface on.

Access the D-PHY programming interface to initialize and program the D-PHY according to the selected operating mode. See Section 3.5, “D-PHY test interface control”.

1. Program the CSI2 host controller registers according to the operating mode’s requirements:
   - Number of lanes (register N_LANES)
   - Deassert PHY shutdown (register PHY_SHUTDOWNZ)
   - Deassert PHY reset (register PHY_RSTZ)
   - Deassert CSI reset (register CSI2_RESETN)
   - (Optional) program data IDs for matching of error reporting (DATA_IDS_1 and DATA_IDS_2 registers)
   - (Optional) program the interrupt masks (MASK1 and MASK2 registers)
2. Read the PHY status register (PHY_STATE) to confirm that all data and clock lanes of the D-PHY are in the stop state, which means that they are ready to receive data.
3. Access the camera sensor using the CCI interface to initialize and configure the camera sensor to transmit a clock on the D-PHY clock lane.
4. Read the PHY status register (PHY_STATE) to confirm that the D-PHY is receiving a clock on the D-PHY clock lane.

3.4. MIPI D-PHY clock

The camera sensor (the sensor output differential clock) drives and controls the MIPI D-PHY clock.

The MIPI D-PHY clock must be calibrated to the actual clock range of the camera sensor's D-PHY clock and the calibrated value must be equal to or greater than the camera sensor clock. This frequency ranges from 80 MHz to 1000 MHz.

The MIPI D-PHY clock must be set according to a known value of the camera sensor's pixel clock. This must be a known value or a value measured with an oscilloscope during a high-speed burst.

To calculate the MIPI data rate, use these equations:

MIPI data rate = (MIPI clock * 2) * Number of lanes >= Pixel clock * Bits-per-pixel

MIPI clock = (Pixel clock * Bits-per-pixel) / (Number of lanes) / 2

For example, a video input of 720p, 59.94 fps, and YUV422 is calculated as follows:

Pixel clock = 1280 * 720 * 59.94 fps * 1 cycle/pixel * 1.35 blank interval = 74.57 MHz

Total MIPI data rate is 74.25 M * 16 bits = 1193 Mb/s.

The frame blank intervals and the interface packaging overhead were added as the 1.35 factor in the pixel clock equation above.

For a 2-lane interface:

MIPI clock = 1193 / 2 / 2 = 298.25 MHz

MIPI_CSI2_PHY_TST_CTRL1 setting = 298.25 MHz * 2 (DDR mode) = 596.5 MHz

According to Table 2, MIPI_CSI2_PHY_TST_CTRL1 = 0x2E.

For a 4-lane interface:

MIPI clock = 1193 / 4 / 2 = 149.12 MHz

MIPI_CSI2_PHY_TST_CTRL1 setting = 149.12 MHz * 2 (DDR mode) = 298.24 MHz

According to Table 2, MIPI_CSI2_PHY_TST_CTRL1 = 0x28.

The final clock result is multiplied by two because the clock mode is DDR. Check the corresponding MIPI_CSI_PHY_TST_CTRL1 register value in Table 2.

Section 3.5, "D-PHY test interface control" provides the steps to set the clock lane frequency.

If the clocks are not equal to or greater than the camera sensor clock, the MIPI_CSI error state register MIPI_CSI_ERR1 may indicate the "start of transmission error on data lane x" for the MIPI_CSI_ERR1 bits 0 to 3.

The recommended value for ref_clock is 27 MHz. This clock derives from PLL3PFD1 -> VIDEO_27M_CLK_ROOT. Besides MIPI, this clock is also used for the HDMI and VPU blocks. For more information, see the Systems Clock section in the i.MX6 reference manual.

This table shows the frequency range and the exact register value when ref_clock is set to 27 MHz:

Table 4. Register settings and resulting clock frequency

Frequency range	Register value	Exact value when ref_clock is 27 MHz
950–1000 MHz	0x74	999 MHz
900–950 MHz	0x54	972 MHz
850–900 MHz	0x34	900 MHz
800–850 MHz	0x14	849 MHz
750–800 MHz	0x32	783 MHz
700–750 MHz	0x12	750 MHz
650–700 MHz	0x30	699 MHz
600–650 MHz	0x10	648 MHz
550–600 MHz	0x2e	600 MHz
500–550 MHz	0x0e	549 MHz
450–500 MHz	0x2c	486 MHz
400–450 MHz	0x0c	450 MHz
360–400 MHz	0x4a	399 MHz
330–360 MHz	0x2a	360 MHz
300–330 MHz	0x08	330 MHz
270–300 MHz	0x28	300 MHz
250–270 MHz	0x08	270 MHz
240–250 MHz	0x46	249 MHz
210–240 MHz	0x26	240 MHz
200–210 MHz	0x06	210 MHz
180–200 MHz	0x44	180 MHz
160–180 MHz	0x04	180 MHz
150–160 MHz	0x04	159 MHz
140–150 MHz	0x42	150 MHz
125–140 MHz	0x22	135 MHz
110–125 MHz	0x02	123 MHz
100–110 MHz	0x40	108 MHz
90–100 MHz	0x20	99 MHz
80–90 MHz	0x00	90 MHz[1]

1. Default configuration

3.5. D-PHY test interface control

MIPI D-PHY has a procedure to test the high-speed RX clock of lane 0 “HS RX control of lane 0”. Besides the test, this procedure sets the D-PHY internal PLLs to correct frequencies. This is a required procedure.

Table 5. D-PHY test interface control

Step	Register -> bit field	Value	Comments
1	MIPI_CSI_PHY_TST_CTRL0 -> [1] phy_testclk	0	Clear testclk
2	MIPI_CSI_PHY_TST_CTRL0 -> [0] phy_testclr	1	Reset test interface
3	MIPI_CSI_PHY_TST_CTRL1	0x00	Clear testen, testdin, and testdout
4	MIPI_CSI_PHY_TST_CTRL0 -> [0] phy_testclr	0	—
5	MIPI_CSI_PHY_TST_CTRL0 -> [1] phy_testclk	1	—
6	MIPI_CSI_PHY_TST_CTRL1	0x44	Select the test number 0x44
7	MIPI_CSI_PHY_TST_CTRL1 -> [16] phy_testen	1	Enable the test
8	MIPI_CSI_PHY_TST_CTRL0 -> [1] phy_testclk	0	—
9	MIPI_CSI_PHY_TST_CTRL1 -> [16] phy_testen	0	—
10	MIPI_CSI_PHY_TST_CTRL1 -> [7:0] phy_testdin	xx	Write the clock value to testdin according to Table 4
11	MIPI_CSI_PHY_TST_CTRL0 -> [1] phy_testclk	1	—
12	MIPI_CSI_PHY_TST_CTRL0 -> [1] phy_testclk	0	—
13	MIPI_CSI_PHY_SHUTDOWNZ -> [0] PHY_SHUTDOWNZ	1	—
14	MIPI_CSI_DPHY_RSTZ -> [0] DPHY_RSTZ	1	—
15	MIPI_CSI_CSI2_RESETN -> [0] CSI2_RESETN	1	—

3.6. Note for 4-lane configuration

For a 4-lane configuration, there is an errata on the following part numbers and silicon revisions:

Table 6. Errata for 4-lane configuration

Example part numbers	Silicon revision
MCIMX6Q6AVT10AC	i.MX 6Quad, Revision 1.2
MCIMX6D6AVT10AD	i.MX 6Dual, Revision 1.3
MCIMX6QP6AVT1AA	i.MX 6QuadPlus, Revision 1.0
MCIMX6DP6AVT1AA	i.MX 6DualPlus, Revision 1.0

CRC errors can occur in the MIPI CSI-2 4-lane configuration. These errors occur during the inactive phase of the bus.

When using the 4-lane configuration with the long data packet video, the internal counter indicating the number of received payload data continues counting even after the long data packet ends until the next packet comes in. This causes the count overflow to produce a CRC error for the last packet received.

The CRC error only occurs when all of these conditions are met:

1. MIPI CSI-2 is configured to use four data lanes.
2. Vertical blanking before the frame end (FE) is >=0x40000/CSI_CLK0 period.
3. No line-start and line-end short packets occur during the frame.

The functionality of the received data is not impacted, only the CRC contains a wrong value. As a workaround, perform these steps:

1. Adjust the CSI transmit output timing to make sure the vertical blanking before the frame (FE) is < 0x40000/CSI_CLK0 period.
2. Make sure each line has both the line start (LS) and the line end (LE).
3. Ignore the CRC error if you confirm that the CRC error is due to the operating conditions described above.

For more information, see *Chip Errata for the i.MX 6Dual/6Quad and i.MX 6DualPlus/6QuadPlus* (document IMX6DQCE), section ERR009704 MIPI: CSI-2: CRC error produced in 4-lane configuration.

4. CSI-2/IPU Gasket Configuration Details

The CSI2/IPU gasket is a digital core that works as a gasket interface between the MIPI CSI-2 host controller and the IPU system (see Figure 2 and Figure 3). This facilitates the communication between a MIPI CSI-2-compliant camera sensor and IPU. The gasket's main functions are:

- To synchronize the CSI-2 input 32-bit data bus with the 16-bit data bus
- To separate the four virtual channels (up to four virtual channels of the MIPI CSI-2 host controller)

The main features of the CSI2IPU gasket are:

- Dynamically configurable pixel clock gating or non-gating for the IPU module
- Dynamically configurable RGB444 and YUV422 data format for the IPU module
- Up to four virtual channels of the MIPI CSI-2 host controller
- All data types of the MIPI Alliance Standard for Camera Serial Interface (CSI-2)
- A software reset to reset the program during the operation

4.1. Color formats and clock

The CSI2IPU gasket can receive these color formats: RGB888, RGB666, RGB565, RGB555, RGB444, YUV422, YUV420, RAW6, RAW7, RAW8, RAW10, RAW12, and RAW14.

In case of receiving the RGB444 or YUV422 formats, the RGB444_FM [3] and YUV422_8BIT_FM [2] bits of the CSI2IPU_SW_RST register must be configured properly. For all other color formats, these two bits are ignored.

The CLK_SEL [1] bit must be set according to the received clock mode (gated or non-gated clock mode). For more information about the clock gate mode, see the Camera Sensor Interface section in the i.MX6 reference manual.

4.2. Routing

The CSI2IPU gasket routes the video stream according to the VC (virtual channel) information embedded in the video stream. When the video stream passes through a multiplexer (Figure 2 and Figure 3), it must be configured to select the parallel input or the MIPI input (see Table 9).

After passing through the CSI-2/IPU gasket, the VC (virtual channel) information embedded in the stream is set to 0. The CSIx data identifier register (IPUx_CSIx_DI) contains only the data type information on bits 5 to 0. Bits 8 and 7 (reserved for VC) are 0x00.

4.3. CSI2/IPU gasket register

CSI2/IPU Gasket Software Reset (CSI2IPU_SW_RST):

Table 7. CSI2/IPU gasket register settings

Bits	Description
31–4	Reserved. Not used.
3 RGB444_FM	**rgb444 mode selection:** 0—{4'h0,r4b4g4} 1—{r4,1'b0,g4,2'b00,b4,1'b0}
2 YUV422_8BIT_FM	**YUV422 8-bit mode selection:** 0—YUYV 1—UYVY
1 CLK_SEL	**Clock mode selection:** 0—gated mode 1—non-gated mode
0 SW_RST	**Software reset:** 0—software reset disable 1—software reset enable

5. IPU and CSI-2 Configuration Details

5.1. Processing video stream in IPU

On the path from the camera to the memory or display (Figure 10), the IPU sub-blocks can carry out the image processing:

- CSI-2 (camera sensor interface)
  - Receives the image from the parallel or MIPI interfaces. Can crop the input image.
- IC (image converter)
  - Can perform color space conversion, resizing, and combining.
- VDIC (video de-interlacer or combiner)
  - Converts an interlaced video stream to a progressive order using a high-quality 3-field motion-adaptive filter and combines two progressive video/graphics planes.

- DP (display processor)
  - Can perform color space conversion, combining, color keying, and gamma correction on images before the display output.
- IRT (image rotator)
  - Can perform color space conversion, combining, and gamma correction on images before the display output.

For more information on using the blocks above to process the image, see the IPU section in the i.MX6 reference manual.

5.2. Single and double buffering

IPU contains a mechanism to automatically handle the single buffer or the double buffer to receive image frames. In the single-buffer mode, the received frames are stored at the same memory address. The memory address is defined in the CPMEM external memory buffer address parameter EBA0.

In the double-buffer mode, the frames are stored in the memory alternating the memory address according to the CPMEM external memory buffer address parameters EBA0 and EBA1.

In both modes, FSU (frame synchronization unit) updates the IPUx_CUR_BUF_x, IPUx_CH_BUF_RDY0, and IPUx_CH_BUF1_RDY registers to the new buffer address. NFACK (new frame acknowledge) is the switch point.

The example in Section 6, “Usage Example” shows how to implement the image capture using a single frame buffer.

5.3. Limitations

Table 8. IPU limitations

—	MIPI [6]	CSI-2	IC	VDIC	DP	IRT
Bandwidth	400 MB/s [5]	150 MP/s	Input—200 MP/s Output—100 MP/s	240 MP/s [2]	240 MP/s	100 MP/s or 120 MP/s [4]
Max. input size	—	8192 x 4096	4096 x 4096	968 x 1024 [1]	—	—
Max. output size	—	8192 x 4096	1024 x 1024	968 x 2048	—	—
Input color depths	6 to 24 bits per pixel	From 4 and up to 16 bits per value	8 bits per value	8 bits per value	8 bits per value	8 bits per value
Output color depths	6 to 24 bits per pixel	From 8 and up to 16 bits per value	8 bits per value	8 bits per value	8 bits per value	8 bits per value
Input color formats	Note 8	Note 7	YUV/RGB [3]	YUV422/YUV420	YUVA/RGBA	YUV/RGB [3]
Output color formats	Note 8	Note 7	YUV/RGB [3]	YUV422/YUV420	YUV/RGB	YUV/RGB [3]
Source	External MIPI sensor	External parallel sensor or MIPI	CSI-2 or memory	CSI-2 or memory	IC and/or memory	Memory
Destination	CSI-2	IC or SMFC	DP or memory	IC or memory	Display	Memory

1. May be a vertical stripe of a wider field; for example, 1920 pixels.
2. MP/s = Mega pixels per second.
3. All variances of YUV and RGB (YUV420, YUV422, RGB565, RGBA8888, and other).
4. Up to 120 MP/s when a single task is active; up to 100 MP/s when more than one task is active.
5. To calculate in MP/s, see Section 2.3.1, “Bandwidth”. The rate depends on the number of lanes, color depth, and format.
6. D-PHY + MIPI controller + MIPI/CSI gasket.
7. Bayer RGB, RGB444, RGB 555, RGB565, RGB888, YUV422, YUV444, gray scale, generic data.
8. Generic data, RGB888, RGB666, RGB565, RGB555, RGB444, YUV422 8 bits, YUV422 10 bits, YUV420 8 bits, YUV420 10 bits, RAW6, RAW7, RAW8, RAW10, RAW12, RAW14.

5.4. Virtual channels

The video streams are routed to the specific IPU/CSI-2 according to the virtual channels. The video source must provide the virtual channel information along with the video stream (see Section 2.2.2.3, “Data identifier and virtual channel”).

The routing is done by the CSI-2/IPU gasket, and, in cases where the CSI-2 can receive data either from the parallel interface or MIPI, the multiplexers must be set (Figure 2).

The general-purpose register 1 (IOMUXC_GPR1) bits 19 and 20 control the multiplexers.

Table 9. Video stream routing settings

Register	Bit field	Description
IOMUXC_GPR1	[19] MIPI_IPU1_MUX	MIPI sensor to IPU-1 mux control 0—enable MIPI to IPU1 CSI0—the virtual channel is fixed to 0. 1—enable parallel interface to IPU1 CSI0.
IOMUXC_GPR1	[20] MIPI_IPU2_MUX	MIPI sensor to IPU-2 mux control 0—enable MIPI to IPU2 CSI1—the virtual channel is fixed to 3. 1—enable parallel interface to IPU2 CSI1.

5.5. Data type configuration

The IPUx_CSIn_DI register contains only the data type for each IPU/CSI. The values must be set according to Table 1.

After passing through the CSI-2/IPU gasket, the virtual channel information from all streams is set to 0. The VC information from the IPUx_CSIn_DI register must be also set to 0.

The MIPI_CSI_DATA_IDS_1 and MIPI_CSI_DATA_IDS_2 registers are optional and needed only when the error reporting is required.

6. Usage Example

6.1. Use case example—image capture from sensor to memory

This use case shows how to capture an image from the sensor to the memory passing through Sensor 1 -> CSI0 -> MIPI -> SMFC -> dma_smfc_ch0 -> IDMAC and memory (see the blue path in the following figure).

The use case shows how to configure MIPI (the MIPI/CSI-2 module, MIPI D-PHY, and the CSI2IPU gasket), CSI-2, SMFC, and IDMAC.

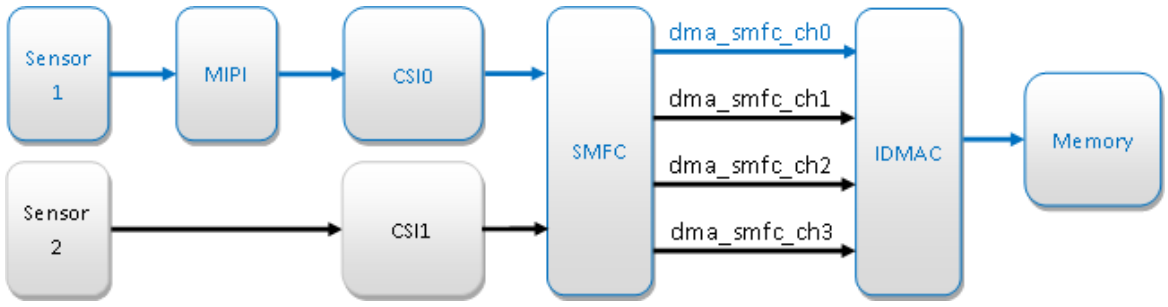

Figure 13. Sensor-to-memory image capture block diagram

The configuration in this example is as follows:

- Platform: i.MX6Q SabreSD
- Sensor: Omnivision OV5640
- Sensor output: 640x480 pixels, 15 fps, non-interleaved YUV422, 8 bits per color
- Number of lanes: 2

- Single frame buffer
- IPU 1/CSI 0/virtual channel 0

6.1.1. MIPI clock

- Configure the MIPI clock and set the video PLL (PLL5) to 596 MHz.

Table 10. MIPI clock PLL settings

Register	Bit field	Value	Comments
CCM_ANALOG_PLL_VIDEO	DIV_SELECT	0	—
CCM_ANALOG_PLL_VIDEO	ENABLE	1	—
CCM_ANALOG_PLL_VIDEO_NUM	—	0x00000000	—
CCM_ANALOG_PLL_VIDEO_DENOM	—	0x00000001	—
CCM_ANALOG_PLL_VIDEO	LOCK	1	Wait for PLL lock
CCM_ANALOG_PLL_VIDEO	BYPASS	0	Clear bypass

6.1.2. MIPI camera power and reset

On the i.MX6Q SabreSD board, the OV5640 camera power and reset pins are controlled by the i.MX GPIOs. Use this information to set these two pins:

- Power supply, pin25, GPIO_PORT6.

Table 11. GPIO configuration to control camera power

Register	Bit field	Value	Comments
IOMUXC_SW_MUX_CTL_PAD_NAND_WP_B	—	0x00000005	—
IOMUXC_SW_PAD_CTL_PAD_NAND_WP_B	—	0x0001B070	—
GPIO6_GDIR	bit [9]	1	Set direction GPIO_PORT6, 9, output
GPIO6_DR	bit [9]	1	Set level high level

- Reset, pin27, GPIO_PORT6.

Table 12. GPIO configuration to control camera reset

Register	Bit field	Value	Comments
IOMUXC_SW_MUX_CTL_PAD_NAND_READY	—	0x00000005	—
IOMUXC_SW_PAD_CTL_PAD_NAND_READY	—	0x0001B070	—
GPIO6_GDIR	Bit [10]	1	Set low level
GPIO6_DR	Bit [10]	0	Wait 1 ms
GPIO6_DR	Bit [10]	1	Set high level

6.1.3. MIPI/CSI-2 controller program

- Set the MIPI lanes to 2.

Table 13. GPIO configuration to control camera power/reset

Register	Bit field	Value	Comments
MIPI_CSI_N_LANES	—	0x00000001	—

6.1.4. D-PHY clock configuration

See the procedure in Section 3.5, "D-PHY test interface control". In this case, the testdin value is 0x14.

- Check if D-PHY is ready to receive data.

Table 14. Read D-PHY state

Register	Bit field	Value	Comments
MIPI_CSI_PHY_STATE	[9] phy_rxulpsclknot	—	Check the phy_rxulpsclknot for 100 ms
MIPI_CSI_PHY_STATE	[8] phy_rxclkactivehs	—	Check the phy_rxclkactivehs for 100 ms

6.1.5. MIPI2CSI gasket configuration

In this example, all bits from the unique MIPI2CSI gasket register remain in the default state: 0x00000000. There is no need to change them:

Table 15. Gasket configuration

Register	Bit field	Value	Comments
CSI2IPU_SW_RST	[3] RGB444_FM	0	—
CSI2IPU_SW_RST	[2] YUV422_8BIT_FM	0	—
CSI2IPU_SW_RST	[1] CLK_SEL	0	—
CSI2IPU_SW_RST	[0] SW_RST	0	—

6.1.6. IPU configuration—IDMAC

The IDMAC channel 0 connects CSI0 to the memory. Configure the IDMAC channel 0 as follows:

- Disable the IDMAC channel enable 1 register:

Table 16. Disabling IDMAC

Register	Bit field	Value	Comments
IPU1_IDMAC_CH_EN_1	—	0x00000000	—

- Configure the CPMEM parameters as follows:
  - Channel0 -> CSI to MEM channel 0

Table 17. Configuring CPMEM

CPMEM parameter	Value	Comments
External memory buffer 0 address (EBA0)	0x08000000	EBA0 channel 23/8 (0x40000000/8)
External memory buffer 1 address (EBA1)	0	EBA1 channel 23 (single buffer)
Frame width (FW)	0x0000027f	640–1
Frame height (FH)	0x000001df	480–1
Pixel format select (PFS)	0x01	Non-interleaved YUV422
Scan order (SO)	0	Scan order progressive
Stride line (SL)	640	Width
Mem U buffer offset (UBO)	0x4b000	Width x Height = 640 x 480
Number of pixels in the whole burst access (NPB)	15	—

- Single buffer selected for channel 0:

Table 18. Selecting single buffer

Register	Bit field	Value	Comments
IPU1_CH_DB_MODE_SEL0	—	0x00000000	—

- IDMAC channel enable 1 register:

Table 19. Enabling IDMAC channel

Register	Bit field	Value	Comments
IPU1_IDMAC_CH_EN_1	—	0x00000001	—

6.1.7. IPU configuration—SMFC

SMFC (sensor multi FIFO controller) must be configured to map the CSI0 to the DMASMFC channel 0 and set the burst size.

Table 20. SMFC mapping

Register	Bit field	Value	Comments
IPU1_SMFC_MAP	—	0x00000000	MAP_CH0 -> CSI0, ID=0 mapped to DMASMFC channel 0
IPU1_SMFC_BS	—	0x00000003	SMFC burst size register BURST0_SIZE -> Burst size of SMFCDMA channel 0. = 4) = 0x00000003 (4 – 1 = 3)
IPU1_CONF	—	0x00000001	CSI0 is enabled

6.1.8. CSI-2 configuration

Configure the CSI-2 general settings, sensor frame size, and output control as follows:

- Set the CSI-2 data source:

Table 21. CSI-2 general configuration

Register	Bit field	Value	Comments
IPU1_CONF	[28] CSI0_DATA_SOURCE	1	Configuration register; MIPI is connected to CSI0
IPU1_CSI0_DI	[0:7] CSI0_MIPI_DI0	0x1e	Set the MIPI_YUV pixel format
IPU1_CSI0_SENS_CONF	[24:26] CSI0_DATA_DEST	0x04	Destination is IDMAC via SMFC
IPU1_CSI0_SENS_CONF	[4:6] CSI0_SENS_PRTCL	0x00	Set the gated clock mode
IPU1_CSI0_SENS_CONF	[16:23] CSI0_DIV_RATIO	0x00	DIV ratio clock division ratio minus 1
IPU1_CSI0_SENS_CONF	[11:14] CSI0_DATA_WIDTH	0x01	Eight bits per color
IPU1_CSI0_SENS_CONF	[7] CSI0_PACK_TIGHT	0x00	Each component is written as a 16-bit word, where the MSB is written to bit #15
IPU1_CSI0_SENS_CONF	[10:8] CSI0_SENS_DATA_FORMAT	0x02	Set YUV422 (UYVY...)

- Set the sensor frame size:

Table 22. Setting sensor frame size

Register	Bit field	Value	Comments
IPU1_CSI0_SENS_FRM_SIZE	[27:16] CSI0_SENS_FRM_HEIGHT	0x01df	Height size = 480 – 1
IPU1_CSI0_SENS_FRM_SIZE	[12:0] CSI0_SENS_FRM_WIDTH	0x027f	Width size = 640 – 1
IPU1_CSI0_ACT_FRM_SIZE	[27:16] CSI0_ACT_FRM_HEIGHT	0x01df	Act Height size = 480 – 1
IPU1_CSI0_ACT_FRM_SIZE	[12:0] CSI0_ACT_FRM_WIDTH	0x027f	Act Width size = 640 – 1

- Set the CSI0 output control and configuration registers:

Table 23. CSI output settings

Register	Bit field	Value	Comments
IPU1_CSI0_OUT_FRM_CTRL	[31] CSI0_HORZ_DWNS	0	Horizontal downsizing disabled
IPU1_CSI0_OUT_FRM_CTRL	[30] CSI0_VERT_DWNS	0	Vertical downsizing disabled
IPU1_CSI0_OUT_FRM_CTRL	[16:28] CSI0_HSC	0	Vertical skip
IPU1_CSI0_OUT_FRM_CTRL	[0:11] CSI0_VSC	0	Horizontal skip
IPU1_CONF	[31] CSI_SEL	0	CSI0 is selected
IPU1_CONF	[0] CSI0_EN	1	CSI0 is enabled

6.1.9. General-purpose and IPU channels buffer registers

In the last IPU settings step, clear [28] IPU_CH_BUF0_RDY0_CLR and set [0] DMA_CH_BUF0_RDY_0.

Table 24. General-purpose registers

Register	Bit field	Value	Comments
IPU1_GPR	[28] IPU_CH_BUF0_RDY0_CLR	0	—
IPU1_CH_BUF0_RDY0	[0] DMA_CH_BUF0_RDY_0	1	—

6.2. Sensor setup

The OV5640 used in this example must be configured to match the MIPI/IPU settings (640x480@15, YUV422). It is configured by writing the sensor registers through the I^2C interface.

See the register values in the sensor datasheet or in its Linux OS driver.

7. Appendix A—MIPI-CSI2 and D-PHY Registers

7.1. D-PHY registers

- PHY shutdown control (MIPI_CSI_PHY_SHUTDOWNZ)

Table 25. PHY shutdown control register

Bits	Description
31–1	Reserved. Not used.
0 PHY_SHUTDOWNZ	Shutdown input. This line is used to place the complete module into power down. All analog blocks are in the power down mode and the digital logic is cleared. Active-low. Default value: 0

- PHY reset control (MIPI_CSI_DPHY_RSTZ)

Table 26. PHY reset control register

Bits	Description
31–1	Reserved. Not used.
0 DPHY_RSTZ	D-PHY reset output. Active-low. Default value: 0

- General settings for all blocks (MIPI_CSI_PHY_STATE)

Table 27. MIPI-CSI2 PHY state register

Bits	Description
31–12	Reserved. Not used.
11 bypass_2ecc_tst	Payload bypass test mode for double ECC errors Default value: 0
10 phy_stopstateclk	Clock lane is in the stop state. Default value: 0
9 phy_rxulpsclknot	Active-low. Indicates that the clock lane module entered the ultra-low power state. Default value: 0
8 phy_rxclkactivehs	Clock lane is actively receiving a DDR clock Default value: 0
7 phy_stopstatedata_3	Data lane 3 is in the stop state Default value: 0
6 phy_stopstatedata_2	Data lane 2 is in the stop state Default value: 0
5 phy_stopstatedata_1	Data lane 1 is in the stop state Default value: 0
4 phy_stopstatedata_0	Data lane 0 is in the stop state Default value: 0
3 phy_rxulpsesc_3	Lane module 3 entered the ultra-low power mode Default value: 0
2 phy_rxulpsesc_2	Lane module 2 entered the ultra-low power mode Default value: 0
1 phy_rxulpsesc_1	Lane module 1 entered the ultra-low power mode Default value: 0
0 phy_rxulpsesc_0	Lane module 0 entered the ultra-low power mode Default value: 0

- D-PHY test interface control 0 (MIPI_CSI_PHY_TST_CRTL0)

Table 28. D-PHY test interface control 0

Bits	Description
31–2	Reserved. Not used.
1 phy_testclk	The PHY test interface strobe signal which is used to clock the TESTDIN bus into the D-PHY. It controls the operation selection in conjunction with the TESTEN signal. Default value: 0
0 phy_testclr	PHY test interface clear. When active, it performs the vendor-specific interface initialization (active-high). Default value: 0

- D-PHY test interface control 1 (MIPI_CSI_PHY_TST_CTRL1)

Table 29. D-PHY test interface control 1

Bits	Description
31–17	Reserved. Not used.
16 phy_testen	PHY test interface operation selector: 1—configures the address write operation on the falling edge of TESTCLK 0—configures the data write operation on the rising edge of TESTCLK
15–8 phy_testdout	PHY output 8-bit data bus for read-back and internal probing functionalities Default value: 0
7–0 phy_testdin	PHY test interface input 8-bit data bus for internal register programming and test functionalities access Default value: 0

7.2. MIPI-CSI2 registers

- Controller version identification register (MIPI_CSI_VERSION)

Table 30. Controller version identification register

Bits	Description
31–0 Version	Version of the CSI-2 host controller Default value: CSI_VERSION_ID

- Number of active data lanes (MIPI_CSI_N_LANES)

Table 31. Number of active data lanes register

Bits	Description
31–2	Reserved. Not used.
1-0 N_LANES	Number of active data lanes Can be updated only when the PHY lane is in the stop state Default value: CSI_N_LANES 00—one data lane (lane 0) 01—two data lanes (lanes 0 and 1) 10—three data lanes (lanes 0, 1, and 2) 11—four data lanes (all lanes)

- CSI2 controller reset (MIPI_CSI_CSI2_RESETN)

Table 32. CSI2 controller reset register

Bits	Description
31–1	Reserved. Not used.
0 CSI2_RESETN	CSI-2 controller reset output. Active-low. Default value: 0

- Data IDs for which IDI reports line boundary matching errors (MIPI_CSI_DATA_IDS_1)

Table 33. MIPI-CSI2 data ID 1 register

Bits	Description
31–30 DI3_VC	Data ID 3 virtual channel Default value: 0
29–24 DI3_DT	Data ID 3 data type Default value: 0
23–22 DI2_VC	DATA ID 2 virtual channel Default value: 0
21–16 DI2_DT	DATA ID 2 data type Default value: 0
15–14 DI1_VC	Data ID 1 virtual channel Default value: 0
13–8 DI1_DT	Data ID 1 data type Default value: 0
7–6 DI0_VC	Data ID 0 virtual channel Default value: 0
5–0 DI0_DT	Data ID 0 data type Default value: 0

- Data IDs for which IDI reports line boundary matching errors (MIPI_CSI_DATA_IDS_2)

Table 34. MIPI-CSI2 data ID 2 register

Bits	Description
31–30 DI7_VC	Data ID 7 virtual channel Default value: 0
29–24 DI7_DT	Data ID 7 data type Default value: 0
23–22 DI6_VC	Data ID 6 virtual channel Default value: 0
21–16 DI6_DT	Data ID 6 data type Default value: 0
15–14 DI5_VC	Data ID 5 virtual channel Default value: 0
13–8 DI5_DT	Data ID 5 data type Default value: 0
7–6 DI4_VC	Data ID 4 virtual channel Default value: 0
5–0 DI4_DT	Data ID 4 data type Default value: 0

- Error state register 1 (MIPI_CSI_ERR1)

Table 35. Error state 1 register

Bits	Description
31–29	Reserved. Not used.
28 err_ecc_double	Header ECC contains two errors. Unrecoverable. Default value: 0
27 vc3_err_crc	Checksum error detected on virtual channel 3 Default value: 0
26 vc2_err_crc	Checksum error detected on virtual channel 2 Default value: 0
25 vc1_err_crc	Checksum error detected on virtual channel 1 Default value: 0
24 vc0_err_crc	Checksum error detected on virtual channel 0 Default value: 0
23 err_l_seq_di3	Error in the sequence of lines for vc3 and dt3 Default value: 0
22 err_l_seq_di2	Error in the sequence of lines for vc2 and dt2 Default value: 0
21 err_l_seq_di1	Error in the sequence of lines for vc1 and dt1 Default value: 0
20 err_l_seq_di0	Error in the sequence of lines for vc0 and dt0 Default value: 0
19 err_l_bndry_match_di3	Error matching the line start to the line end for vc3 and dt3 Default value: 0
18 err_l_bndry_match_di2	Error matching the line start to the line end for vc2 and dt2 Default value: 0
17 err_l_bndry_match_di1	Error matching the line start to the line end for vc1 and dt1 Default value: 0
16 err_l_bndry_match_di0	Error matching the line start to the line end for vc0 and dt0 Default value: 0
15 err_frame_data_vc3	The last received frame in virtual channel 3 had at least one CRC error Default value: 0
14 err_frame_data_vc2	The last received frame in virtual channel 2 had at least one CRC error Default value: 0

Table 35. Error state 1 register

Bits	Description
13 err_frame_data_vc1	The last received frame in virtual channel 1 had at least one CRC error Default value: 0
12 err_frame_data_vc0	The last received frame in virtual channel 0 had at least one CRC error Default value: 0
11 err_f_seq_vc3	Incorrect frame sequence detected in virtual channel 3 Default value: 0
10 err_f_seq_vc2	Incorrect frame sequence detected in virtual channel 2 Default value: 0
9 err_f_seq_vc1	Incorrect frame sequence detected in virtual channel 1 Default value: 0
8 err_f_seq_vc0	Incorrect frame sequence detected in virtual channel 0 Default value: 0
7 err_f_bndry_match_vc3	Error matching the frame start to the frame end for virtual channel 3 Default value: 0
6 err_f_bndry_match_vc2	Error matching the frame start to the frame end for virtual channel 2 Default value: 0
5 err_f_bndry_match_vc1	Error matching the frame start the frame end for virtual channel 1 Default value: 0
4 err_f_bndry_match_vc0	Error matching the frame start to the frame end for virtual channel 0 Default value: 0
3 phy_errsotsynchs_3	Start of transmission error on data lane 3 (no synchronization achieved) Default value: 0
2 phy_errsotsynchs_2	Start of transmission error on data lane 2 (no synchronization achieved) Default value: 0
1 phy_errsotsynchs_1	Start of transmission error on data lane 1 (no synchronization achieved) Default value: 0
0 phy_errsotsynchs_0	Start of transmission error on data lane 0 (no synchronization achieved) Default value: 0

- Error state register 2 (MIPI_CSI_ERR2)

Table 36. Error state 2 register

Bits	Description
31–24	Reserved. Not used.
23 err_l_seq_di7	Error in the sequence of lines for vc7 and dt7 Default value: 0
22 err_l_seq_di6	Error in the sequence of lines for vc6 and dt6 Default value: 0
21 err_l_seq_di5	Error in the sequence of lines for vc5 and dt5 Default value: 0
20 err_l_seq_di4	Error in the sequence of lines for vc4 and dt4 Default value: 0
19 err_l_bndry_match_di7	Error matching the line start to the line end for vc7 and dt7 Default value: 0
18 err_l_bndry_match_di6	Error matching the line start to the line end for vc6 and dt6 Default value: 0
17 err_l_bndry_match_di5	Error matching the line start to the line end for vc5 and dt5 Default value: 0
16 err_l_bndry_match_di4	Error matching the line start to the line end for vc4 and dt4 Default value: 0
15 err_id_vc3	Unrecognized or unimplemented data type detected in virtual channel 3 Default value: 0
14	Unrecognized or unimplemented data type detected in virtual channel 2

Table 36. Error state 2 register

Bits	Description
err_id_vc2	Default value: 0
13 err_id_vc1	Unrecognized or unimplemented data type detected in virtual channel 1 Default value: 0
12 err_id_vc0	Unrecognized or unimplemented data type detected in virtual channel 0 Default value: 0
11 vc3_err_ecc_corrected	Header error detected and corrected on virtual channel 3 Default value: 0
10 vc2_err_ecc_corrected	Header error detected and corrected on virtual channel 2 Default value: 0
9 vc1_err_ecc_corrected	Header error detected and corrected on virtual channel 1 Default value: 0
8 vc0_err_ecc_corrected	Header error detected and corrected on virtual channel 0 Default value: 0
7 phy_errsoths_3	Start of transmission error on data lane 3 (synchronization can still be achieved) Default value: 0
6 phy_errsoths_2	Start of transmission error on data lane 2 (synchronization can still be achieved) Default value: 0
5 phy_errsoths_1	Start of transmission error on data lane 1 (synchronization can still be achieved) Default value: 0
4 phy_errsoths_0	Start of transmission error on data lane 0 (synchronization can still be achieved) Default value: 0
3 phy_erresc_3	Escape entry error (ULPM) on data lane 3 Default value: 0
2 phy_erresc_2	Escape entry error (ULPM) on data lane 2 Default value: 0
1 phy_erresc_1	Escape entry error (ULPM) on data lane 1 Default value: 0
0 phy_erresc_0	Escape entry error (ULPM) on data lane 0 Default value: 0

- Masks for errors 1 (MIPI_CSI_MASK1)

Table 37. Error masks 1 register

Bits	Description
31–29	Reserved. Not used.
28 mask_err_ecc_double	Mask for err_ecc_double Default value: 0
27 mask_vc3_err_crc	Mask for vc3_err_crc Default value: 0
26 mask_vc2_err_crc	Mask for vc2_err_crc Default value: 0
25 mask_vc1_err_crc	Mask for vc1_err_crc Default value: 0
24 mask_vc0_err_crc	Mask for vc0_err_crc Default value: 0
23 mask_err_l_seq_di3	Mask for err_l_seq_di3 Default value: 0
22 mask_err_l_seq_di2	Mask for err_l_seq_di2 Default value: 0
21 mask_err_l_seq_di1	Mask for err_l_seq_di1 Default value: 0
20 mask_err_l_seq_di0	Mask for err_l_seq_di0 Default value: 0

Table 37. Error masks 1 register

Bits	Description
19 mask_err_l_bndry_match_di3	Mask for err_l_bndry_match_di3 Default value: 0
18 mask_err_l_bndry_match_di2	Mask for err_l_bndry_match_di2 Default value: 0
17 mask_err_l_bndry_match_di1	Mask for err_l_bndry_match_di1 Default value: 0
16 mask_err_l_bndry_match_di0	Mask for err_l_bndry_match_di0 Default value: 0
15 mask_err_frame_data_vc3	Mask for err_frame_data_vc3 Default value: 0
14 mask_err_frame_data_vc2	Mask for err_frame_data_vc2 Default value: 0
13 mask_err_frame_data_vc1	Mask for err_frame_data_vc1 Default value: 0
12 mask_err_frame_data_vc0	Mask for err_frame_data_vc0 Default value: 0
11 mask_err_f_seq_vc3	Mask for err_f_seq_vc3 Default value: 0
10 mask_err_f_seq_vc2	Mask for err_f_seq_vc2 Default value: 0
9 mask_err_f_seq_vc1	Mask for err_f_seq_vc1 Default value: 0
8 mask_err_f_seq_vc0	Mask for err_f_seq_vc0 Default value: 0
7 mask_err_f_bndry_match_vc3	Mask for err_f_bndry_match_vc3 Default value: 0
6 mask_err_f_bndry_match_vc2	Mask for err_f_bndry_match_vc2 Default value: 0
5 mask_err_f_bndry_match_vc1	Mask for err_f_bndry_match_vc1 Default value: 0
4 mask_err_f_bndry_match_vc0	Mask for err_f_bndry_match_vc0 Default value: 0
3 mask_phy_errsotsynchs_3	Mask for phy_errsotsynchs_3 Default value: 0
2 mask_phy_errsotsynchs_2	Mask for phy_errsotsynchs_2 Default value: 0
1 mask_phy_errsotsynchs_1	Mask for phy_errsotsynchs_1 Default value: 0
0 mask_phy_errsotsynchs_0	Mask for phy_errsotsynchs_0 Default value: 0

- Masks for errors 2 (MIPI_CSI_MASK2)

Table 38. Error masks 2 register

Bits	Description
31–24	Reserved. Not used.
23 mask_err_l_seq_di7	Mask for err_l_seq_di7 Default value: 0
22 mask_err_l_seq_di6	Mask for err_l_seq_di6 Default value: 0
21 mask_err_l_seq_di5	Mask for err_l_seq_di5 Default value: 0
20	Mask for err_l_seq_di4

Table 38. Error masks 2 register

Bits	Description
mask_err_l_seq_di4	Default value: 0
19 mask_err_l_bndry_match_di7	Mask for err_l_bndry_match_di7 Default value: 0
18 mask_err_l_bndry_match_di6	Mask for err_l_bndry_match_di6 Default value: 0
17 mask_err_l_bndry_match_di5	Mask for err_l_bndry_match_di5 Default value: 0
16 mask_err_l_bndry_match_di4	Mask for err_l_bndry_match_di4 Default value: 0
15 mask_err_id_vc3	Mask for err_id_vc3 Default value: 0
14 mask_err_id_vc2	Mask for err_id_vc2 Default value: 0
13 mask_err_id_vc1	Mask for err_id_vc1 Default value: 0
12 mask_err_id_vc0	Mask for err_id_vc0 Default value: 0
11 mask_vc3_err_ecc_corrected	Mask for vc3_err_ecc_corrected Default value: 0
10 mask_vc2_err_ecc_corrected	Mask for vc2_err_ecc_corrected Default value: 0
9 mask_vc1_err_ecc_corrected	Mask for vc1_err_ecc_corrected Default value: 0
8 mask_vc0_err_ecc_corrected	Mask for vc0_err_ecc_corrected Default value: 0
7 mask_phy_errsoths_3	Mask for phy_errsoths_3 Default value: 0
6 mask_phy_errsoths_2	Mask for phy_errsoths_2 Default value: 0
5 mask_phy_errsoths_1	Mask for phy_errsoths_1 Default value: 0
4 mask_phy_errsoths_0	Mask for phy_errsoths_0 Default value: 0
3 mask_phy_erresc_3	Mask for phy_erresc_3 Default value: 0
2 mask_phy_erresc_2	Mask for phy_erresc_2 Default value: 0
1 mask_phy_erresc_1	Mask for phy_erresc_1 Default value: 0
0 mask_phy_erresc_0	Mask for phy_erresc_0 Default value: 0

8. Revision History

This table summarizes the changes done to this document since the initial release:

Table 39. Revision history

Revision number	Date	Substantive changes
0	07/2016	Initial release.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number: AN5305
Rev. 0
07/2016