UG0806
User Guide
MIPI CSI-2 Receiver Decoder For PolarFire

Microsemi
a Microchip company

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

50200806. 10.0 11/21

Contents

Figures

Tables

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the current publication.

1.1 Revision 10.0

The following is a summary of changes made in this revision.

- Updated Key Features, page 3
- Updated Figure 2, page 4.
- Updated Table 1, page 5
- Updated Table 2, page 6

1.2 Revision 9.0

The following is a summary of changes made in this revision.

- Updated Key Features, page 3
- Updated Table 4, page 8

1.3 Revision 8.0

The following is a summary of changes made in this revision.

- Added support for 8 lanes configuration for Raw-14, Raw-16 and RGB-888 Data types.
- Updated Figure 2, page 4.
- Updated section Key Features, page 3.
- Updated section mipi_csi2_rxdecoder, page 5.
- Updated Table 2, page 6 and Table 4, page 8.

1.4 Revision 7.0

The following is a summary of changes made in this revision.

- Added sub level sections Key Features, page 3 and Supported Families, page 3.
- Updated Table 4, page 8.
- Updated Figure 4, page 9 and Figure 5, page 9.
- Added sections License, page 10, Installation Instructions, page 11, and Resource Utilization, page 12.
- Core Support for Raw14, Raw16, and RGB888 data types for 1, 2, and 4 lanes were added.

1.5 Revision 6.0

The following is a summary of changes made in this revision.

- Updated Introduction, page 3.
- Updated Figure 2, page 4.
- Updated Table 2, page 6.
- Updated Table 4, page 8.

1.6 Revision 5.0

The following is a summary of changes made in this revision.

- Updated Introduction, page 3.
- Updated title for Figure 2, page 4.
- Updated Table 2, page 6 and Table 4, page 8.

1.7 Revision 4.0

Updated the document for Libero SoC v12.1.

1.8 Revision 3.0

The following is a summary of changes made in this revision.

- Support for RAW12 data type was added.
- Added frame_valid_o output signal in the IP, see Table 2, page 6.
- Added g_NUM_OF_PIXELS configuration parameter in Table 4, page 8.

1.9 Revision 2.0

Support for RAW10 data type was added.

1.10 Revision 1.0

The first publication of this document.

2 Introduction

MIPI CSI-2 is a standard specification defined by a Mobile Industry Processor Interface (MIPI) alliance. The Camera Serial Interface 2 (CSI-2) specification defines an interface between a peripheral device (camera) and a host processor (base-band, application engine). This user guide describes the MIPI CSI-2 receiver decoder for PolarFire (MIPI CSI-2 RxDecoder), which decodes the data from the sensor interface.

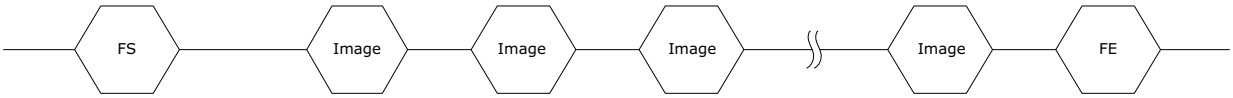
The IP core supports multi-lane (1, 2, 4, and 8 lanes) for Raw-8, Raw-10, Raw-12, Raw-14, Raw-16, and RGB-888 data types.

MIPI CSI-2 operates in two modes—high-speed mode and low-power mode. In high-speed mode, MIPI CSI-2 supports the transport of image data using short packet and long packet formats. Short packets provide frame synchronization and line synchronization information. Long packets provide pixel information. The sequence of transmitted packets is as follows.

1. Frame start (short packet)
2. Line start (optional)
3. Few image data packets (long packets)
4. Line end (optional)
5. Frame end (short packet)

One long packet is equivalent to one line of image data. The following illustration shows the video data stream.

Figure 1 • **Video Data Stream**



2.1 Key Features

- Supports Raw-8, Raw-10, Raw-12, Raw-14, Raw-16, and RGB-888 data types for 1, 2, 4, and 8 lanes
- Supports 4 pixels per pixel clock for 4 and 8 lanes mode
- Supports Native and AXI4 Stream Video Interface
- IP does not support transactions in Low power mode
- IP does not support Embedded/Virtual channel (ID) mode

2.2 Supported Families

- PolarFire® SoC
- PolarFire®

3 Hardware Implementation

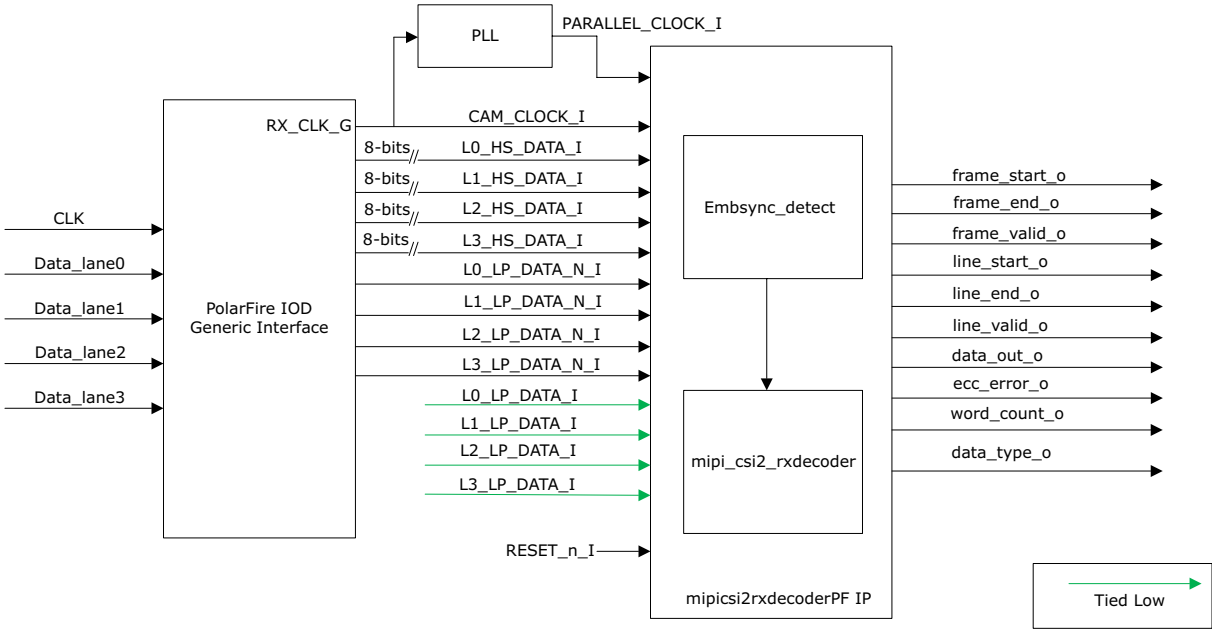
This section describes the hardware implementation details. The following illustration shows the MIPI CSI2 receiver solution that contains the MIPI CSI2 RxDecoder IP. This IP has to be used in conjunction with the PolarFire® MIPI IOD generic interface blocks and Phase-Locked Loop (PLL). The MIPI CSI2 RxDecoder IP is designed to work with the PolarFIre MIPI IOG blocks. Figure 2 shows the pin connection from the PolarFire IOG to the MIPI CSI2 RxDecoder IP. A PLL is required to generate the parallel clock (pixel clock). The input clock to the PLL will be from the RX_CLK_R output pin of the IOG. The PLL has to be configured to produce the parallel clock, based on the MIPI_bit_clk and the number of lanes used. The equation used to calculate the parallel clock is as follows.

$$CAM\_CLOCK\_I = (MIPI\_bit\_clk)/4$$

$$PARALLEL\_CLOCK = (CAM\_CLOCK\_I \times Num\_of\_Lanes \times 8)/(g\_DATAWIDTH \times g\_NUM\_OF\_PIXELS)$$

The following illustration shows the architecture of MIPI CSI-2 Rx for PolarFire.

Figure 2 • **Architecture of MIPI CSI-2 Rx Solution for 4 Lane Configuration**



The preceding figure shows the different modules in the MIPI CSI2 RxDecoder IP. When used in conjunction with the PolarFire IOD Generic and PLL, this IP can receive and decode the MIPI CSI2 packets to produce pixel data along with the valid signals.

3.1 Design Description

This section describes the different internal modules of the IP.

3.1.1 Embsync_detect

This module receives data from the PolarFire IOG and detects the embedded SYNC code in the received data of each lane. This module also aligns the data from each lane to the SYNC code and sends it to the mipi_csi2_rxdecoder module for decoding the packet.

3.1.2 mipi_csi2_rxdecoder

This module decodes the incoming short packets and long packets and generates the `frame_start_o, frame_end_o, frame_valid_o, line_start_o, line_end_o, word_count_o, line_valid_o, and data_out_o` outputs. Pixel data arrives between line start and line end signals. The short packet contains only the packet header and supports various data types. MIPI CSI-2 Receiver IP Core supports the following data types for short packets.

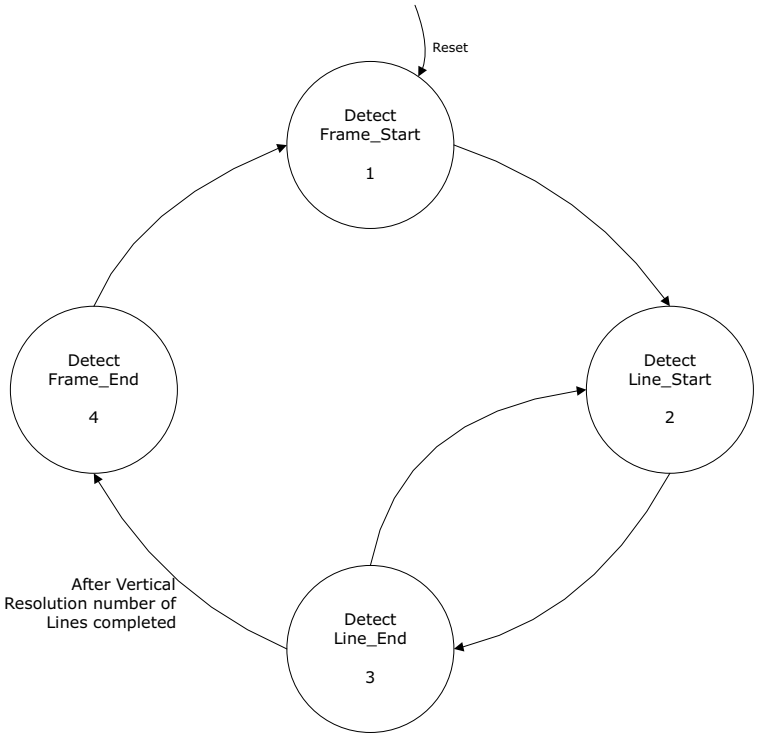
Table 1 • **Supported Short Packet Data Types**

Data Type	Description
0x00	Frame Start
0x01	Frame End

The long packet contains the image data. The length of the packet is determined by the horizontal resolution, to which the camera sensor is configured. This can be seen at the word_count_o output signal in bytes.

The following illustration shows the FSM implementation of decoder.

Figure 3 • **FSM Implementation of Decoder**



1. Frame Start: On receiving the frame start packet, generate the frame start pulse, and then wait for line start.
2. Line Start: On receiving the line start indication, generate the line start pulse.
3. Line End: On generating the line start pulse, store the pixel data, and then generate the line end pulse. Repeat Step 2 and 3 until the frame end packet is received.
4. Frame End: On receiving the frame end packet, generate the frame end pulse. Repeat the above steps for all frames.

The CAM_CLOCK_I must be configured to the image sensor frequency, to process the incoming data, regardless of Num_of_lanes_i configured to one lane, two lanes, or four lanes.

The IP supports Raw-8, Raw-10, Raw-12, Raw-14, Raw-16, and RGB-888 data types. One pixel per clock is received on data_out_o if the g_NUM_OF_PIXELS is set to one. If the g_NUM_OF_PIXELS is set to 4 then four pixels per clock are sent out and the parallel clock has to be configured 4 times lower than the normal case. The four pixels per clock configuration gives user the flexibility to run their design at higher resolutions and higher camera data rates, which makes it easier to meet design timings. To indicate valid image data, the line_valid_o output signal is sent. Whenever it is asserted high, output pixel data is valid.

3.2 Inputs and Outputs

The following table lists the input and output ports of the IP configuration parameters.

Table 2 • **Input and Output Ports for Native Video Interface**

Signal Name	Direction	Width	Description
CAM_CLOCK_I	Input	1	Image sensor clock
PARALLEL_CLOCK_I	Input	1	Pixel clock
RESET_N_I	Input	1	Asynchronous active low reset signal
L0_HS_DATA_I	Input	8-bits	High speed input data from lane 1
L1_HS_DATA_I	Input	8-bits	High speed input data from lane 2
L2_HS_DATA_I	Input	8-bits	High speed input data from lane 3
L3_HS_DATA_I	Input	8-bits	High speed input data from lane 4
L4_HS_DATA_I	Input	8-bits	High speed input data from lane 5
L5_HS_DATA_I	Input	8-bits	High speed input data from lane 6
L6_HS_DATA_I	Input	8-bits	High speed input data from lane 7
L7_HS_DATA_I	Input	8-bits	High speed input data from lane 8
L0_LP_DATA_I	Input	1	Positive low power input data from lane one. Default value is 0 for PolarFire and PolarFire SoC.
L0_LP_DATA_N_I	Input	1	Negative low power input data from lane one
L1_LP_DATA_I	Input	1	Positive low power input data from lane two. Default value is 0 for PolarFire and PolarFire SoC.
L1_LP_DATA_N_I	Input	1	Negative low power input data from lane two
L2_LP_DATA_I	Input	1	Positive low power input data from lane three. Default value is 0 for PolarFire and PolarFire SoC.
L2_LP_DATA_N_I	Input	1	Negative low power input data from lane three
L3_LP_DATA_I	Input	1	Positive low power input data from lane four. Default value is 0 for PolarFire and PolarFire SoC.
L3_LP_DATA_N_I	Input	1	Negative low power input data from lane four
L4_LP_DATA_I	Input	1	Positive low power input data from lane five. Default value is 0 for PolarFire and PolarFire SoC.

Table 2 • Input and Output Ports for Native Video Interface *(continued)*

L4_LP_DATA_N_I	Input	1	Negative low power input data from lane five
L5_LP_DATA_I	Input	1	Positive low power input data from lane six. Default value is 0 for PolarFire and PolarFire SoC.
L5_LP_DATA_N_I	Input	1	Negative low power input data from lane six
L6_LP_DATA_I	Input	1	Positive low power input data from lane seven. Default value is 0 for PolarFire and PolarFire SoC.
L6_LP_DATA_N_I	Input	1	Negative low power input data from lane seven
L7_LP_DATA_I	Input	1	Positive low power input data from lane eight. Default value is 0 for PolarFire and PolarFire SoC.
L7_LP_DATA_N_I	Input	1	Negative low power input data from lane eight
data_out_o	Output	g_DATAWIDTH*g_NUM_OF_PIXELS-1: 0	8-bit, 10-bit, 12-bit, 14-bit, 16-bit, and RGB-888 (24-bit) with one pixel per clock. 32-bit, 40-bit, 48-bit, 56-bit, 64-bit, and 96-bit with four pixels per clock.
line_valid_o	Output	1	Data valid output. Asserted high when data_out_ois valid
frame_start_o	Output	1	Asserted high for one clock when frame start is detected in the incoming packets
frame_end_o	Output	1	Asserted high for one clock when frame end is detected in the incoming packets
frame_valid_o	Output	1	Asserted high for one clock for all active lines in a frame
line_start_o	Output	1	Asserted high for one clock when line start is detected in the incoming packets
line_end_o	Output	1	Asserted high for one clock when line end is detected in the incoming packets
word_count_o	Output	16-bits	Represents the pixel value in bytes
ecc_error_o	Output	1	Error signal which indicates ECC mismatch
data_type_o	Output	8-bits	Represents Data type of packet

3.3 AXI4 Stream Port

The following table lists the input and output ports of the AXI4 Stream Port.

Table 3 • **Ports for AXI4 Stream Video Interface**

Port Name	Type	Width	Description
RESET_N_I	Input	1bit	Active low asynchronous reset signal to design.
CLOCK_I	Input	1bit	System clock
TDATA_O	Output	g_NUM_OF_PIXELS*g_DATAWIDTH bit	Output Video Data
TVALID_O	Output	1bit	Output Line Valid
TLAST_O	Output	1bit	Output frame end signal
TUSER_O	Output	4bit	bit 0 = End of frame bit 1 = unused bit 2 = unused bit 3 = Frame Valid
TSTRB_O	Output	g_DATAWIDTH /8	Output Video Data strobe
TKEEP_O	Output	g_DATAWIDTH /8	Output Video Data Keep

3.4 Configuration Parameters

The following table lists the description of the configuration parameters used in the hardware implementation of the MIPI CSI-2 Rx Decoder block. They are generic parameters and can vary based on the application requirements.

Table 4 • **Configuration Parameters**

Name	Description
Data Width	Input pixel data width. Supports 8-bits, 10-bits, 12-bits, 14-bits, 16-bits, and 24-bits (RGB 888)
Lane Width	Number of MIPI lanes. • Supports 1, 2, 4, and 8 lanes
Number of Pixels	The following options are available: 1: One pixel per clock 4: Four pixels per clock with pixel clock frequency reduced four times (available only in 4 lane or 8 lane mode).
Input Data Invert	The options to invert the incoming data are as follows: 0: does not invert the incoming data 1: inverts the incoming data
FIFO Size	Address Width of Byte2PixelConversion FIFO, Supported in Range: 8 to 13.
Video Interface	Native and AXI4 Stream Video Interface

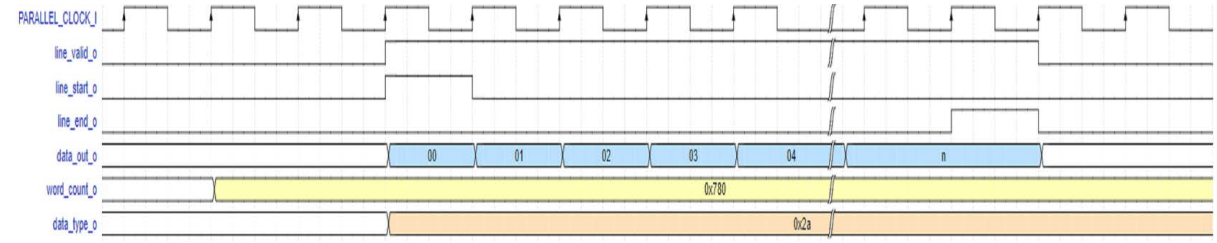
3.5 Timing Diagram

The following sections show the timing diagrams.

3.5.1 Long Packet

The following illustration shows the timing waveform of the long packet.

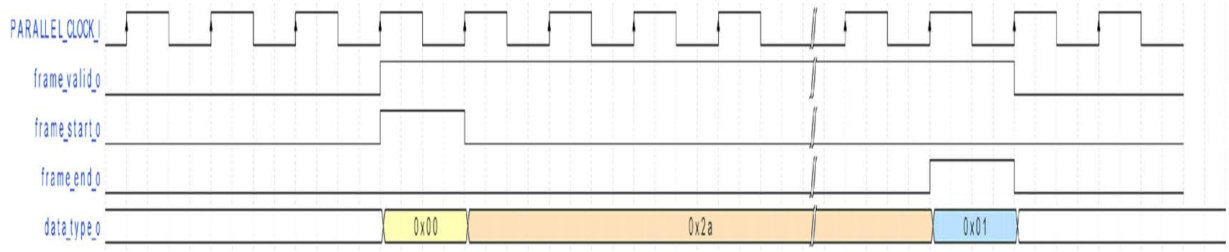
Figure 4 • **Timing Waveform of Long Packet**



3.5.2 Short Packet

The following illustration shows the timing waveform of the frame start packet.

Figure 5 • **Timing Waveform of Frame Start Packet**



4 License

MIPICSI2 RxDecoder IP clear RTL is license locked and the encrypted RTL is available for free.

4.1 Encrypted

Complete RTL code is provided for the core, allowing the core to be instantiated with the SmartDesign tool. Simulation, synthesis, and layout can be performed within Libero® System-on-Chip (SoC). The RTL code for the core is encrypted.

4.2 RTL

Complete RTL source code is provided for the core.

5 Installation Instructions

The core must be installed into Libero software. It is done automatically through the Catalog update function in Libero, or the CPZ file can be manually added using the **Add Core** catalog feature. Once the CPZ file is installed in Libero, the core can be configured, generated, and instantiated within SmartDesign for inclusion in the Libero project.

For further instructions on core installation, licensing, and general use, refer to the *Libero SoC Online Help*.

6 Resource Utilization

The following table shows the resource utilization of a sample MIPI CSI-2 Receiver Core implemented in a PolarFire FPGA (MPF300TS-1FCG1152I package) for RAW 10 and 4-lane configuration.

Table 5 • **Resource Utilization**

Element	Usage
DFFs	1327
4-input LUTs	1188
LSRAMs	12