MIPI CSI-2 Intel® FPGA IP User Guide

Updated for Intel® Quartus® Prime Design Suite: **23.2.1**

IP Version: **1.0.0**

Send Feedback

783467
2023.09.28

Contents

1. About the MIPI CSI-2 Intel® FPGA IP

The MIPI Camera Serial Interface 2 (MIPI CSI-2) Intel® FPGA IP is a high-speed protocol IP for transmission of video images from image sensors to application processors. Among the use cases of MIPI CSI-2 Intel FPGA IP are camera modules and receiver modules in mobile phone engines. This IP core is designed to be compatible with the MIPI D-PHY Intel FPGA IP.

1.1. MIPI CSI-2 Intel FPGA IP Features

- Receiver protocol layer only
- Support for 1, 2, 4, and 8 D-PHY lanes
- Support for 1, 2, and 4 pixels in parallel
- Support for data formats YUV422 8-bit, RGB888, RGB565, RAW8, and RAW10
- Avalon® memory-mapped interface for memory access
- AMBA AXI4-Stream interface for video data streaming

1.2. Device Family Support for MIPI CSI-2 Intel FPGA IP

The following terms define IP support levels for Intel FPGA IP device families and state the level of support available in the current release:

- **Advance support**—the IP core is available for simulation and compilation for this device family. Timing models include initial engineering estimates of delays based on early post-layout information. The timing models are subject to change as silicon testing improves the correlation between the actual silicon and the timing models. You can use this IP core for system architecture and resource utilization studies, simulation, pinout, system latency assessments, basic timing assessments (pipeline budgeting), and I/O transfer strategy (data-path width, burst depth, I/O standards tradeoffs).
- **Preliminary support**—the IP core is verified with preliminary timing models for this device family. The IP core meets all functional requirements, but might still be undergoing timing analysis for the device family. It can be used in production designs with caution.
- **Final support**—the IP core is verified with final timing models for this device family. The IP core meets all functional and timing requirements for the device family and can be used in production designs.

Table 1. MIPI CSI-2 Intel FPGA IP Device Family Support

Device Family	Support Level
Intel Agilex® 5	Advanced (without simulation support)
Other device families	No support

1.3. Recommended Speed Grades

Table 2. Recommended Speed Grades

The following table shows the recommended speed grade when axi4s_clk is configured to run at certain clock frequencies. To calculate the required clock frequency, refer to the *Clock Requirements* section.

Clock Name	Clock Frequency	Recommended Speed Grades
axi4s_clk	>250	-5
	≤250	-6

Related Information

Clock Requirements on page 15

1.4. MIPI CSI-2 Intel FPGA IP Performance and Resource Utilization

Table 3. Performance and Resource Utilization

The following table shows the typical device resource utilization for selected configurations using the Intel Quartus® Prime Pro Edition software version 23.2.1. The number of ALMs and logic registers are rounded up to the nearest 50. The number stated for M20K memory blocks includes no rounding.

Configuration					ALMs	M20K Memory Blocks	Logic Registers
Data Lanes	PPI Bus Width	Pixel-in-Parallel	Buffer Depth	Data Type Supported			
1	16	1	4096	RGB888	1950	6	4750
1	16	4	1024	RGB888	2100	3	5300
2	16	1	4096	RGB888	2000	6	5000
2	16	4	1024	RGB888	2150	3	5500
4	16	1	4096	RGB888	2600	6	6250
4	16	4	1024	RGB888	2700	3	6750
8	16	1	4096	RGB888	4100	6	8950
8	16	4	1024	RGB888	4150	3	9350
2	16	4	1024	RGB565	2100	3	5300
4	16	4	1024	RGB565	2650	3	6500
8	16	4	1024	RGB565	4050	3	8950
2	16	4	1024	YUV422 8-bit	2100	2	5350
4	16	4	1024	YUV422 8-bit	2700	2	6650
8	16	4	1024	YUV422 8-bit	4200	2	9350
2	16	4	1024	RAW8	1950	1	4900
4	16	4	1024	RAW8	2550	1	6000
8	16	4	1024	RAW8	3900	1	8300
2	16	4	1024	RAW10	2100	2	5200
4	16	4	1024	RAW10	2700	2	6600
8	16	4	1024	RAW10	4300	2	9450

1.5. Release Information for MIPI CSI-2 Intel FPGA IP

Table 4. Release information for the MIPI CSI-2 Intel FPGA IP

Item	Description
IP Version	1.0.0
Intel Quartus Prime Pro Edition Version	23.2.1
Release Date	2023.08.11
Ordering Code	IP-MIPI-CSI-2

2. MIPI CSI-2 Intel FPGA IP Interface Overview

2.1. MIPI CSI-2 Intel FPGA IP Clocks

Table 5. MIPI CSI-2 Intel FPGA IP Clocks

Clock	Description
axi4s_clk	AXI4-Streaming video clock. This is used for AXI4-Streaming video processing pipeline as well as Control and Status registers.
rx_word_clk_hs_ck	High-Speed Receive Word Clock. This is used to synchronize signals in the high-speed receive clock domain.
rx_word_clk_hs_d<x> x represents lane index	High-Speed Receive Word Clock. This is used to synchronize signals in the high-speed receive clock domain.

2.2. MIPI CSI-2 Intel FPGA IP Resets

Table 6. MIPI CSI-2 Intel FPGA IP Resets

Reset	Associated Clock Domain	Description
axi4s_rst	axi4s_clk	Asserting this reset triggers a reset to all the blocks running at axi4s_clk clock domain.
rx_srst_n_ck	rx_word_clk_hs_ck	This signal is part of RX PHY-Protocol Interface (PPI) output signals which should be connected to MIPI D-PHY Intel FPGA IP.
rx_srst_n_d<x>	rx_word_clk_hs_d<x> x represents lane index	This signal is part of RX PPI output signals which should be connected to MIPI D-PHY Intel FPGA IP.

2.3. MIPI CSI-2 Intel FPGA IP User Interfaces

2.3.1. Receiver PHY Protocol Interface (PPI) Signals

Table 7. Receiver PHY Protocol Interface (PPI) Signals

Signal	Width	Direction	Description
Clock Lane High Speed Receive. N=16			
rx_data_width_hs_ck	2	Output	Refer to *MIPI D-PHY specification version 2.5: Figure 95 and 97, 101 High Speed Receive on Slave Side, 8/16-bit Bus, LP Mode*.
rx_data_hs_ck	N	Input	
rx_valid_hs_ck	N/8	Input	
rx_active_hs_ck	1	Input	
rx_sync_hs_ck	1	Input	

continued...

Signal	Width	Direction	Description
rx_clk_active_hs_ck	1	Input	Refer to *MIPI D-PHY specification version 2.5: Figure 90, Example High Speed Clock Enable, LP Mode. Figure 121 High Speed Transmission and Reception with HS-Idle Function.*
rx_skew_cal_hs_ck	1	Input	Calibration. Implementation-specific how the system responds to this condition. Refer to *MIPI D-PHY specification version 2.5: Figure 119. Skew Calibration, LP Mode.*
rx_alternate_cal_hs_ck	1	Input	
rx_error_cal_hs_ck	1	Input	
Data Lane High Speed Receive. X=0-7; N=16			
rx_srst_n_d<X>	1	Input	Synchronous reset from Intel MIPI RX D-PHY.
rx_word_clk_hs_ d<X>	1	Input	Refer to *MIPI D-PHY specification version 2.5: Figure 95 and 97, 101 High Speed Receive on Slave Side, 8/16-bit Bus, LP Mode.*
rx_data_width_hs_ d<X>	2	Output	
rx_data_hs_ d<X>	N	Input	
rx_valid_hs_ d<X>	N/8	Input	
rx_active_hs_ d<X>	1	Input	
rx_sync_hs_ d<X>	1	Input	
rx_clk_active_hs_ d<X>	1	Input	Refer to *MIPI D-PHY specification version 2.5: Figure 90 Example High Speed Clock Enable, LP Mode. Figure 125 High Speed Transmission and Reception with HS-Idle Function.*
rx_skew_cal_hs_ d<X>	1	Input	Calibration. Implementation-specific how the system responds to this condition. Refer to *MIPI D-PHY specification version 2.5: Figure 119. Skew Calibration, LP Mode.*
rx_alternate_cal_hs_ d<X>	1	Input	
rx_error_cal_hs_ d<X>	1	Input	
Clock Lane Escape Mode Receive (LPDT = CCI Control Data Transmission, ULPS = Sleep Mode)			
rx_clk_esc_ck	1	Input	For low power data transmission (LPDT). Required for USL control data communication. Refer to *MIPI D-PHY specification version 2.5: Figure 107 Example Low-Power Data Reception, LP Mode.*
rx_lpdt_esc_ck	1	Input	
rx_data_esc_ck	8	Input	
rx_valid_esc_ck	1	Input	
rx_trigger_esc_ck	4	Input	For LPDT. Any bit received after a Trigger Command (i.e., Reset Trigger) and before the Lines go to the Stop state shall be ignored. Refer to *MIPI D-PHY specification version 2.5: Figure 110-112.*
rx_ulps_esc_ck	1	Input	For ULPS and used with ulps_active_not_ck. Refer to *MIPI D-PHY specification version 2.5: Figure 116 Example Data Lane ULPS Entry and Exit, LP Mode.*
Data Lane Escape Mode Receive (LPDT = CCI Control Data Transmission, ULPS = Sleep Mode). X=0-7			
rx_clk_esc_d<X>	1	Input	For LPDT. Required for USL control data communication. Refer to *MIPI D-PHY specification version 2.5: Figure 107 Example Low-Power Data Reception, LP Mode.*
rx_lpdt_esc_d<X>	1	Input	
rx_data_esc_d<X>	8	Input	
rx_valid_esc_d<X>	1	Input	
rx_trigger_esc_d<X>	4	Input	For LPDT. Any bit received after a Trigger Command (i.e., Reset Trigger) and before the Lines go to the Stop state shall be ignored. Refer to *MIPI D-PHY specification version 2.5: Figure 110-112.*
rx_ulps_esc_d<X>	1	Input	For ULPS and used with ulps_active_not_ck. Refer to *MIPI D-PHY specification version 2.5: Figure 116 Example Data Lane ULPS Entry and Exit, LP Mode.*

continued...

Signal	Width	Direction	Description
Clock Lane Control			
direction_ck	1	Input	Indicates the current direction of Lane. 0 = Transmit mode. 1 = Receive mode.
force_rx_mode_ck	1	Output	For Transmit: Refer to *MIPI D-PHY specification version 2.5: Figure 86. Master PHY Enable, LP Mode*. For Receive: Refer to *MIPI D-PHY specification version 2.5: Figure 88. Example Slave PHY Enable, LP Mode*.
force_tx_stop_mode_ck	1	Output	
stop_state_ck	1	Input	
enable_ck	1	Output	
alp_mode_ck	1	Output	Alternate Low Power Mode Selection. 0 = LP mode. 1 = ALP mode.
tx_ulps_clk_ck	1	Output	Refer to *MIPI D-PHY specification version2.5: Figure 114. Example Clock Lane ULPS Entry and Exit, LP Mode*.
rx_ulps_clk_not_ck	1	Input	
ulps_active_not_ck	1	Input	
tx_hsidle_clk_hs_ck	1	Output	Refer to *MIPI D-PHY specification version 2.5: Figure 121. HS Transmission and Reception with HS-IDLE Function*.
tx_hsidle_clk_ready_hs_ck	1	Input	
Data Lane Control. X=0-7			
direction_d<X>	1	Input	Indicates the current direction of Lane. 0 = Transmit mode. 1 = Receive mode.
force_rx_mode_d<X>	1	Output	For Transmit: Refer to *MIPI D-PHY specification version 2.5: Figure 86. Master PHY Enable, LP Mode*. For Receive: Refer to *MIPI D-PHY specification version 2.5: Figure 88. Example Slave PHY Enable, LP Mode*.
force_tx_stop_mode_d<X>	1	Output	
stop_state_d<X>	1	Input	
enable_d<X>	1	Output	
alp_mode_d<X>	1	Output	Alternate Low Power Mode Selection. 0 = LP mode. 1 = ALP mode.
tx_ulps_clk_d<X>	1	Output	Refer to *D-PHY specification version 2.5: Figure 114. Example Clock Lane ULPS Entry and Exit, LP Mode*.
rx_ulps_clk_not_d<X>	1	Input	
ulps_active_not_d<X>	1	Input	
tx_hsidle_clk_hs_d<X>	1	Output	Refer to *MIPI D-PHY specification version 2.5: Figure 121. HS Transmission and Reception with HS-IDLE Function*.
tx_hsidle_clk_ready_hs_d<X>	1	Input	
Clock Lane Error Input			
i_err_sot_hs_ck	1	Input	Start-of-Transmission (SoT) error.
i_err_sot_sync_hs	1	Input	Start-of-Transmission Synchronization error.
i_err_esc_ck	1	Input	Escape entry error.
i_err_sync_ck	1	Input	Low-power data transmission synchronization error.
i_err_control_ck	1	Input	Control error.

continued...

Signal	Width	Direction	Description
i_err_contention_lp0_ck	1	Input	LP0 Contention error.
i_err_contention_lp1_ck	1	Input	LP1 Contention error.
Data Lane Error Input. N=0-7			
i_err_sot_hs_d<X>	1	Input	SoT error.
i_err_sot_sync_d<X>	1	Input	Start-of-Transmission Synchronization error.
i_err_esc_d<X>	1	Input	Escape entry error.
i_err_sync_d<X>	1	Input	Low-power data transmission synchronization error.
i_err_control_d<X>	1	Input	Control error.
i_err_contention_lp0_d<X>	1	Input	LP0 Contention error.
i_err_contention_lp1_d<X>	1	Input	LP1 Contention error.
Clock Lane Error Output			
o_err_sot_hs_ck	1	Input	SoT error.
o_err_sot_sync_hs	1	Input	Start-of-Transmission Synchronization error.
o_err_esc_ck	1	Input	Escape entry error.
o_err_sync_ck	1	Input	Low-power data transmission synchronization error.
o_err_control_ck	1	Input	Control error.
o_err_contention_lp0_ck	1	Input	LP0 Contention error.
o_err_contention_lp1_ck	1	Input	LP1 Contention error.
Data Lane Error Output. N=0-7			
o_err_sot_hs_d<X>	1	Input	SoT error.
o_err_sot_sync_d<X>	1	Input	Start-of-Transmission Synchronization error.
o_err_esc_d<X>	1	Input	Escape entry error.
o_err_sync_d<X>	1	Input	Low-power data transmission synchronization error.
o_err_control_d<X>	1	Input	Control error.
o_err_contention_lp0_d<X>	1	Input	LP0 Contention error.
o_err_contention_lp1_d<X>	1	Input	LP1 Contention error.

Related Information

Describing the MIPI D-PHY specification version 2.5

2.3.2. Avalon Memory-Mapped Interface Control Interface Signals

Table 8. Avalon Memory-Mapped Interface Control Interface Signals

Signal	Width	Direction	Description
control_address	10	Input	The Avalon memory-mapped interface agent port that provides access to internal control and status register, mainly for authentication messages transfer. This interface is expected to operate at Nios V processor clock domain. Because messages can be large (more than 4 Bytes), the IP transfers the message in burst mode with full handshaking mechanism.
control_write	1	Input	
control_byteenable	4	Input	

continued...

Preliminary Documentation – Subject to Change

Signal	Width	Direction	Description
control_writedata	32	Input	Write transfers always have a wait time of 0 cycle while read transfers have a wait time of 1 cycle. The addressing should be accessed as word addressing in the Platform Designer flow. For example, addressing of 4 in the Nios V software selects the address of 1 in the agent.
control_read	1	Input	
control_readdata	32	Output	
control_readdatavalid	1	Output	
control_waitrequest	1	Output	
control_irq	1	Output	Active high interrupt signal.

2.3.3. Receiver AXI4-Streaming Video Interface Signals

P = floor[((bits per color plane x number of color planes) + 7) / 8] x pixels in parallel x 8) where, for example:

bits per color plane = 8, number of color planes = 2 for YUV422 8-bit

bits per color plane = 6, number of color planes = 3 for RGB565

Note: The minimum value of P is 16. If the answer to the equation is less than 16, the default data width is 16.

Q = ceil(P / 8)

Table 9. Receiver AXI4-Streaming Video Interface Signals

Signal	Width	Direction	Description
axi4s_vid_out_0_tdata	P	Output	AXI4-Streaming data out.
axi4s_vid_out_0_tvalid	1	Output	AXI4-Streaming data valid.
axi4s_vid_out_0_tuser	Q	Output	Bit 0: AXI4-Streaming start of video frame. 0 = Not start of field 1 = Start of field Bit 1: AXI4-Streaming meta or data packet. 0 = Video packet 1 = Metapacket Bit Q-1:2: Unused.
axi4s_vid_out_0_tlast	1	Output	AXI4-Streaming end of packet.
axi4s_vid_out_0_tready	1	Input	AXI4-Streaming data ready.

3. MIPI CSI-2 Intel FPGA IP Parameters

Table 10. IP Core Parameters

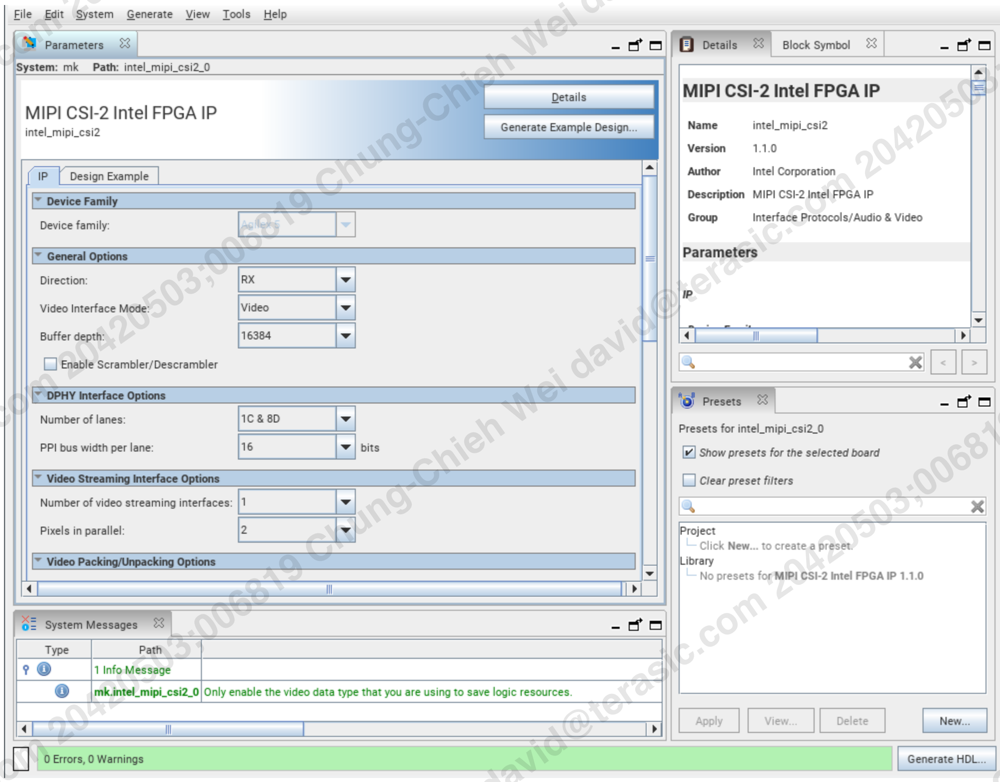
Parameter	Value	Description
Direction	RX	Select the port direction.
Buffer depth	128, 256, 512, 1024, 2048, 4096, 8192, 16384	Select the FIFO buffer depth. The unit is pixels of RX. The actual depth will be the selected value divided by PIXELS_IN_PARALLEL parameter.
Number of lanes	1C & 1D, 1C & 2D, 1C & 4D, 1C & 8D	Select the number of D-PHY lanes. C stands for clock lane while D is for Data lane.
PPI bus width per lane	16	Select the PHY Protocol Interface (PPI) bus width per lane.
Pixels in parallel	1, 2, 4	Select the number of pixels in parallel at the video streaming interface.
Support YUV422 8-bit	True, False	Turn on to support unpacking for YUV422 8-bit video data type.
Support RGB888	True, False	Turn on to support unpacking for the RGB888 video data type.
Support RGB565	True, False	Turn on to support unpacking for the RGB565 video data type.
Support RAW8	True, False	Turn on to support unpacking for the RAW8 video data type.
Support RAW10	True, False	Turn on to support unpacking for the RAW10 video data type.

4. Designing with the MIPI CSI-2 Intel FPGA IP

4.1. Generating the MIPI CSI-2 Intel FPGA IP

1. Create or open an Intel Quartus Prime project (`.qpf`) to include the IP variation.
2. In the IP catalog, navigate to **Interface Protocols ➤ Audio & Video**.
3. Locate **MIPI CSI-2 Intel FPGA IP**. You may type some or all the component's name in the IP Catalog search box to locate the IP. Double click on the IP.
4. The **New IP Variation** window appears.
5. Specify a top-level name for this IP. Do not include spaces in IP names or paths. The parameter editor saves the IP variation settings in a file named <your_ip>.ip.
6. Click **OK**. The parameter editor appears.

Figure 1. Parameter Editor



7. Set your preferred parameter values in the parameter editor. If there is any error, the **Parameterization Messages** tab at the bottom displays the error.

8. Click **Generate HDL**. The **Generation** dialog box appears.
9. Specify output file generation options, and click **Generate**. Intel Quartus Prime generates the synthesis and simulation files based on your selections.
10. Click **Finish** followed by **Yes** if prompted to add files representing the IP variation to your project.

4.2. MIPI CSI-2 Intel FPGA IP Generation Output Intel Quartus Prime Pro Edition

4.2.1. Files Generated for Intel FPGA IP Cores and Platform Designer Systems

Figure 2. Files Generated for IP Cores and Platform Designer Systems

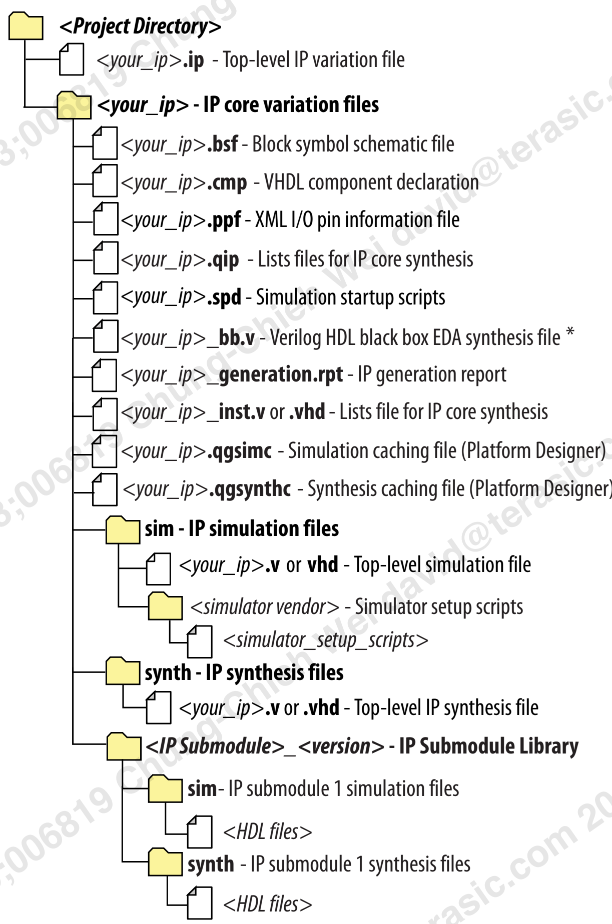


Table 11. IP Core and Platform Designer (Standard) Simulation Files

File Name	Description
<*my_ip*>.cmp	The VHDL Component Declaration (.cmp) file is a text file that contains local generic and port definitions that you can use in VHDL design files.
<*my_ip*>_generation.rpt	IP or Platform Designer generation log file. A summary of the messages during IP generation.
<*my_ip*>.qgsimc	Simulation caching file that compares the .qsys and .ip files with the current parameterization of the Platform Designer system and IP core. This comparison determines if Platform Designer can skip regeneration of the HDL.
<*my_ip*>.qgsynth	Synthesis caching file that compares the .qsys and .ip files with the current parameterization of the Platform Designer system and IP core. This comparison determines if Platform Designer can skip regeneration of the HDL.
<*my_ip*>.qip	Contains all the required information about the IP component to integrate and compile the IP component in the Intel Quartus Prime software.
<*my_ip*>.csv	Contains information about the upgrade status of the IP component.
<*my_ip*>.bsf	A Block Symbol File (.bsf) representation of the IP variation for use in Block Diagram Files (.bdf).
<*my_ip*<>.spd	Required input file for ip-make-simscript to generate simulation scripts for supported simulators. The .spd file contains a list of files generated for simulation, along with information about memories that you can initialize.
<*my_ip*>_bb.v	Use the Verilog black box (_bb.v) file as an empty module declaration for use as a black box.
<*my_ip*>_inst.v or _inst.vhd	HDL example instantiation template. Copy and paste the contents of this file into your HDL file to instantiate the IP variation.
<*my_ip*>.v <*my_ip*>.vhd	HDL files that instantiate each submodule or child IP core for synthesis or simulation.
mentor/	Contains a ModelSim® script msim_setup.tcl to set up and run a simulation.
aldec/	Contains a Riviera-PRO script rivierapro_setup.tcl to setup and run a simulation.
/synopsys/vcs /synopsys/vcsmx	Contains a shell script vcs_setup.sh to set up and run a VCS® simulation. Contains a shell script vcsmx_setup.sh and synopsys_ sim.setup file to set up and run a VCS MX® simulation.
/cadence	Contains a shell script ncsim_setup.sh and other setup files to set up and run an NCSIM simulation.
/xcelium	Contains a shell script xcelium_setup.sh and other setup files to set up and run a Xcelium simulation.
/common	Contains a set of Tcl files, <simulator>_files.tcl, which provide all design related simulation information required by a corresponding simulation script. The Tcl file contains designs from current system-level hierarchy, and references to sub-systems and IP components.
<*IP submodule*>/	For each generated IP submodule directory, Platform Designer generates /synth and /sim sub-directories.

4.3. MIPI CSI-2 Intel FPGA IP Systems Integration and Implementation

4.3.1. Clock Requirements

The derivation of D-PHY high speed word clock per lane is as follows:

- rx_word_clk_hs_d0 = line_rate / ppi_bus_width

Table 12. MIPI D-PHY Clock Frequency of Fabric Interface

PPI Bus Width (Bits)	Line Rate (Mbps)	rx_word_clk_hs_dX (MHz)
16	80	5
	500	31.25
	2500	156.25

The derivation of axi4_clk is as follows:

- axi4s_clk > rx_word_clk_hs_d0 * (lanes * ppi_bus_width) / (bits_per_pixel * pixels_in_parallel)

Table 13. AXI4-Streaming Video Clock (axi4s_clk) Examples

Video Data Type	Line Rate (Mbps)	PPI_Bus Width (Bits)	rx_word_clk_hs_dX (MHz)	Lanes	Pixels in Parallel	Bits per Pixel	axi4s_clk (MHz)
RAW10	800	16	50	2	1	10	>160
RGB888	1500	16	93.75	4	2	24	>125
YUV422 8-bit	1000	16	62.5	8	4	8	>250

Related Information

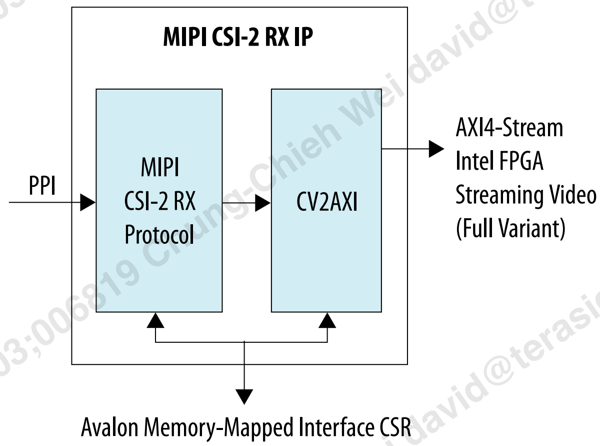
Recommended Speed Grades on page 4

4.3.2. Required Supporting IP

You must pair this IP with MIPI D-PHY Intel FPGA IP. For more information, refer to *MIPI CSI-2 Intel FPGA IP Design Example User Guide* (Intel RDC item #783603).

5. MIPI CSI-2 Intel FPGA IP Block Descriptions

Figure 3. The MIPI CSI-2 Intel FPGA IP Block Diagram



The MIPI CSI-2 RX Protocol component consists of multiple layers defined in the MIPI CSI-2 specification version 3.0, such as the lane management layer, low level protocol and byte to pixel conversion. The MIPI CSI-2 RX Protocol component receives 16-bit data per lane, with support for up to eight lanes, from the MIPI D-PHY Intel FPGA IP through the standard PHY Protocol Interface (PPI). The byte data received on the PPI is then processed by the low-level protocol module to extract the real video information. The extracted and formatted video is presented at its native clocked video interface.

The Clocked Video to AXI convertor (CV2AXI) component primarily converts the clocked video data to AXI4-Streaming video data, particularly the Intel FPGA streaming video protocol. The AXI4-Stream Intel FPGA streaming video packets always follow the pattern of one (and only one) image information metapacket followed by video packets and one (and only one) End-of-Field metapacket.

Each component has an Avalon memory-mapped interface register interface for control and status register access and will be given an address space. Example offset addresses from the system base address could be 0x0, 0x1 for MIPI CSI-2 RX Protocol component, CV2AXI component, respectively.

Table 14. Top Level Interfaces

Interface	Description
AXI4-Stream Intel FPGA Streaming Video (Full Variant)	Carries pure video packet in Intel FPGA streaming video format. Color/byte remapped and ready for processing.

continued...

Interface	Description
	Accompanied by Image Information Packet (IIP) and End of Field (EOP) packets.
PPI	PHY Protocol Interface. Refer to *MIPI D-PHY specification version 2.5 Annex A*.
Avalon memory-mapped interface CSR	Control and Status Register.

Figure 4. MIPI CSI-2 Intel FPGA IP RX Protocol

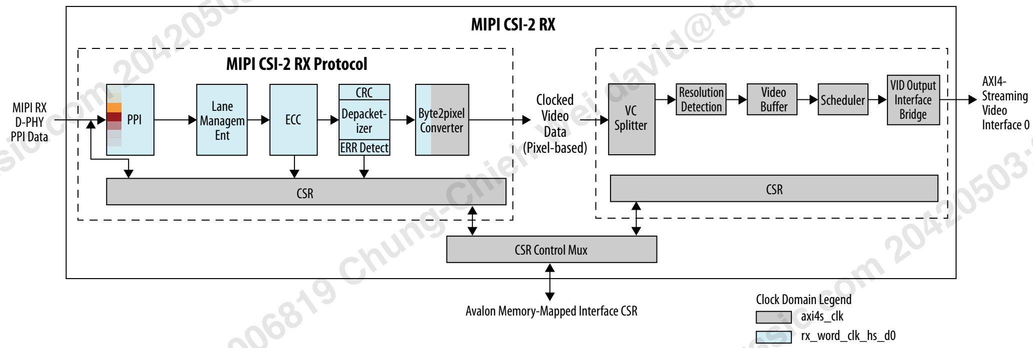


Table 15. MIPI CSI-2 Intel FPGA IP RX Protocol Functional Blocks

Functional Block	Description
RX PPI	• Synchronizes all D-PHY data lanes to a single clock domain (rx_word_clk_hs_d0). • Handles lanes de-skew if more than one data lane is in use. • For easier packet processing downstream, it generates start-of-packet (SOP) and end-of-packet (EOP) markers for each received packet.
Lane Management	• Performs lane merging function which collects incoming bytes from N number of D-PHY lanes and consolidates them into complete packets to pass onto subsequent low level protocol layers.
Error Correction Code (ECC) checking	• Generates a new ECC for the received packet header or short packet, computes the syndrome using the new ECC and the received ECC, decodes the syndrome to find if a single bit error has occurred, and if so, corrects it. • If more than one bit error has occurred, this means an uncorrectable error has occurred, and an error flag will be asserted in the Control and Status Register (CSR).
Depacketizer	• Extracts Packet Header, Packet Footer, HS-trail and all short packets • Generates important markers such as start of frame (SOF), interlaced, even field/line, data type, and Virtual Channel identifier for downstream modules.
CRC and Error checking	• Calculates the checksum of the payload data of every long packet to detect possible errors in transmission. • The calculated checksum will be compared against the received 16-bit CRC checksum from the packet footer and an error will be flagged if there is a mismatch. • The Error Detect submodule also detects frame synchronization error when a Frame End (FE) is not paired with a Frame Start (FS) on the same virtual channel.

continued...

Functional Block	Description
Byte-to-Pixel Converter	• The video payload data from the depacketizer is unpacked based on the video data type information. The pixel output is based on the requested number of pixels in parallel. • A dcfifo is used to transfer the byte-based video payload data from rx_word_clk_hs_d0 clock domain to axi4s_clk clock domain. It is important to ensure that the axi4s_clk should be selected such that the output bandwidth is greater than the input bandwidth.
CV2AXI	• Converts clocked video data from MIPI RX Protocol to AXI4-Stream video data. • Resolution Detection submodule retrieves the video attributes such as number of pixels per line, number of video lines per field, number of bits per symbol, color space, and interlaced/progressive at the CSR. • Stable bit will be generated once video line with consistent width and height is detected. • A video line buffer is present to accommodate throttling on output video interface. The buffer depth is GUI parameterizable. • The scheduler ensures that the outgoing AXI4-Stream video data adheres to Intel FPGA Streaming Video (Full variant) video protocol.
Control and Status Register	• MIPI RX protocol and CV2AXI submodules each has its own CSR. • A CSR Control MUX resides at the higher layer to multiplex the control operation between both CSR. • Base 0 represents MIPI RX Protocol CSR while base 1 represents CV2AXI's CSR.

Related Information

Describing the MIPI D-PHY specification version 2.5

6. Registers

Base address (**control_address[9:8]**) 0x0 represents MIPI CSI-2 RX Protocol Control and Status Register (CSR) while base 0x1 represents CV2AXI CSR.

Table 16. Register Access Codes

Code	Description
RW	Read and write.
RO	Read only.
RW1C	Read, write and clear. The user application writes 1 to the register bit(s) to invoke a defined instruction. The IP core clears the bit(s) upon executing instructions.

Table 17. CSI-2 RX IP Registers

Base	Address	Register	RW	Reset	Bit	Bit Name	Description
0x0	0x00	CORE_STATUS	RO	0x0	31:3	RESERVED	Reserved.
				LANE-1	2:0	MAXIMUM_LANES	Number of D-PHY lanes configured during IP generation.
	0x01 – 0x03	RESERVED	NA	NA	NA	NA	Reserved.
	0x04	GENERIC_SHORT_PACKET	RO	0x0	31:26	RESERVED	Reserved.
					25:8	DATA	16 bits payload data of the generic short packet.
					9:6	VIRTUAL_CHANNEL	Virtual channel identifier.
					5:0	DATA_TYPE	Generic short packet code.
	0x05 – 0x0F	RESERVED	NA	NA	NA	NA	Reserved.
	0x10	VC0_IMAGE_INFO	RO	0x0	31:6	RESERVED	Reserved.
					5:0	VIDEO_DATA_TYPE	Data type of current packet (0x18 and above) received on VC0.
	0x11	VC1_IMAGE_INFO	RO	0x0	31:6	RESERVED	Reserved.

continued...

Base	Address	Register	RW	Reset	Bit	Bit Name	Description
					5:0	VIDEO_DATA_TYPE	Data type of current packet (0x18 and above) received on VC1.
	0x12	VC2_IMAGE_INFO	RO	0x0	31:6	RESERVED	Reserved.
					5:0	VIDEO_DATA_TYPE	Data type of current packet (0x18 and above) received on VC2.
	0x13	VC3_IMAGE_INFO	RO	0x0	31:6	RESERVED	Reserved.
					5:0	VIDEO_DATA_TYPE	Data type of current packet (0x18 and above) received on VC3.
	0x14 – 0x1F	VC<X>_IMAGE_INFO	RO	0x0	31:6	RESERVED	Reserved.
					5:0	VIDEO_DATA_TYPE	Data type of current packet (0x18 and above) received on VC<X>. X: 4-15
	0x11 – 0x50	RESERVED	NA	NA	NA	NA	Reserved.
	0x51	DATA_TYPE_ERROR_IRQ_STATUS	RW1C	0x0	31:16	RESERVED	Reserved.
					15:1	VC<X>_DT_ERROR	Interrupt status registers for data type error indicator on VC1 through VC15.
					0	VC0_DT_ERROR	Interrupt status register. Asserts when a packet header is decoded with an unrecognized or not implemented ID on VC0.
	0x52	FRAME_SYNC_ERROR_IRQ_STATUS	RW1C	0x0	31:16	RESERVED	Reserved.
					15:1	VC<X>_FRAME_SYNC_ERROR	Interrupt status registers for frame sync

continued...

Base	Address	Register	RW	Reset	Bit	Bit Name	Description
							error indicator on VC1 through VC15.
					0	VC0_FRAME_SYNC_ERROR	Interrupt status register. Asserts when an FE is not paired with a FS on VC0.
	0x53	FRAME_DATA_ERROR_IRQ_STATUS	RW1C	0x0	31:16	RESERVED	Reserved.
					15:1	VC<X>_FRAME_DATA_ERROR	Interrupt status registers for frame CRC error indicator on VC1 through VC15.
					0	VC0_FRAME_DATA_ERROR	Interrupt status register. Asserts when the data payload received between FS and FE contains CRC errors on VC0.
	0x54	ECC_1BIT_ERROR_IRQ_STATUS	RW1C	0x0	31:16	RESERVED	Reserved.
					15:1	VC<X>_ECC_1BIT_ERROR	Interrupt status registers for ECC single bit error indicator on VC1 through VC15.
					0	VC0_ECC_1BIT_ERROR	Interrupt status register. Asserts when an ECC syndrome was computed and a single bit error in the packet header was detected and corrected on VC0.
	0x55	ECC_2BIT_ERROR_IRQ_STATUS	RW1C	0x0	31:16	RESERVED	Reserved.
					15:1	VC<X>_ECC_2BIT_ERROR	Interrupt status registers asserted

continued...

Base	Address	Register	RW	Reset	Bit	Bit Name	Description
							when an uncorrectable 2-bit ECC error was detected on VC1.
					0	VC0_ECC_2BIT_ERROR	Interrupt status register. Asserts when an ECC syndrome was computed and an uncorrectable 2-bit error was detected in the received packet header on VC0.
	0x56 – 0x6F	RESERVED	NA	NA	NA	NA	Reserved.
	0x70	CSI2_RX_INTERRUPT_MASK_ENABLE	RW	0x0	31:6	RESERVED	Reserved.
					5	IRQ_EN_ECC_2BIT_ERROR	Interrupt enable registers allow you to selectively generate an interrupt at the output port for each error bit in the registers 0x51 through 0x55 and 0x60. Set to 0 to disable the interrupt.
					4	IRQ_EN_ECC_1BIT_ERROR	
					3	IRQ_EN_FRAME_DATA_ERROR	
					2	IRQ_EN_FRAME_SYNC_ERROR	
					1	IRQ_EN_DATA_TYPE_ERROR	
					0	RESERVED	Reserved.
	0x71 – 0xFF	RESERVED	NA	NA	NA	NA	Reserved.
1	0x00 – 0x4F	RESERVED	NA	NA	NA	NA	Reserved.
	0x50	VIDEO_INTF0_STATUS	RO	0x0	31:2	RESERVED	Reserved.
					1	INTERLACED	Interlaced bit of video streaming interface 0. When asserted, the input video stream is interlaced. Otherwise, the input

continued...

Base	Address	Register	RW	Reset	Bit	Bit Name	Description
							video stream is progressive.
					0	RESERVED	Reserved.
	0x51 – 0x5F	RESERVED	NA	NA	NA	NA	Reserved.
	0x60	VIDEO_INTF0_WIDTH	RO	0x0	31:16	RESERVED	Reserved.
					15:0	WIDTH	Specifies the active video width (in pixel unit) for video streaming interface 0.
	0x61 – 0x6F	RESERVED	NA	NA	NA	NA	Reserved.
	0x70	VIDEO_INTF0_HEIGHT_F0	RO	0x0	31:16	RESERVED	Reserved.
					15:0	HEIGHT_F0	Specifies the active video height (in line unit) of progressive video or interlaced video field 0 for video streaming interface 0.
	0x71 – 0x7F	RESERVED	NA	NA	NA	NA	Reserved.
	0x80	VIDEO_INTF0_HEIGHT_F1	RO	0x0	31:16	RESERVED	Reserved.
					15:0	HEIGHT_F1	Specifies the active video height (in line unit) interlaced video field 1 for video streaming interface 0.
	0x81 – 0x8F	RESERVED	NA	NA	NA	NA	Reserved.
	0x90	VIDEO_INTF0_COLOR_PATTERN	RO	0x0	31:13	RESERVED	Reserved
				0xF	12:9	COLOR_SPACE	Specifies the color space for video streaming interface 0. 0: RGB. 1-4: Reserved. 5: RAW. 6: RGB565. 7: YUV.

continued...

Base	Address	Register	RW	Reset	Bit	Bit Name	Description
							8: YUV legacy. 9-15: Reserved.
				0x0	8:7	COSITE	Specifies the co-siting of the chroma samples for video streaming interface 0. 0: Reserved. 1: Top and center (ie. YUV420 8/10-bit Chroma Shifted Pixel Sampling). 2: Reserved. 3: Center and center (YUV420 8/10-bit).
				0x1	6:5	SUBSAMPLING	Specifies the chroma sampling for video streaming interface 0. 0: 420. 1: Reserved. 2: 422. 3: 444.
				0x0	4:0	BIT_WIDTH_MINUS_1	Specifies the bit width – 1 for video streaming interface 0. I.e., value of 5'd23 indicates bit width of 24.
	0x91 – 0xAF	RESERVED	NA	NA	NA	RESERVED	Reserved.
	0xB0	CORE_STATUS_IRQ_STATUS	RW1C	0x0	31:1	RESERVED	Reserved.
					0	VIDEO_INTF0_STABLE	Stable bit of video streaming interface 0. When asserted, the input video stream has had a consistent width and height.
	0xB1 – 0xB3	RESERVED	NA	NA	NA	RESERVED	Reserved.

continued...

Base	Address	Register	RW	Reset	Bit	Bit Name	Description
	0xB4	CV2AXI_INTERRUPT_MASK_ENABLE	RW	0x0	31:1	RESERVED	Reserved.
					0	IRQ_EN_CORE_STATUS	Interrupt enable registers allow you to selectively generate an interrupt at the output port for each error bit in the registers 0xB0. Set to 0 to disable the interrupt.
	0xB5 – 0xFF	RESERVED	NA	NA	NA	RESERVED	Reserved.

7. Document Revision History for MIPI CSI-2 Intel FPGA IP User Guide

Document Version	Intel Quartus Prime Version	IP Version	Changes
2023.09.28	23.2.1	1.0.0	• Changed the IP ordering code from "MIPI-CSI-2" to "IP-MIPI-CSI-2" to remain consistent with PSG naming rules. • Removed unsupported modes.
2023.08.11	23.2.1	1.0.0	Initial release.