



Byte-to-Pixel Converter IP Core - Lattice Radiant Software

User Guide

FPGA-IPUG-02079-1.7

April 2023

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults and associated risk the responsibility entirely of the Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

This document was created consistent with Lattice Semiconductor's inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice's inclusive language [FAQ 6878](#) for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Acronyms in This Document	5
1. Introduction	6
1.1. Quick Facts	6
1.2. Features	6
1.3. Conventions	7
1.3.1. Nomenclature	7
1.3.2. Signal Names	7
2. Functional Description	8
2.1. Overview	8
2.2. Signal Description	11
2.3. Attributes Summary	13
2.4. Modules Description	18
2.4.1. Byte-to-Pixel Converter	18
2.4.2. AXI4 Stream Subordinate	18
2.4.3. AXI4 Stream Manager	18
2.4.4. FIFO Implementation	19
2.4.5. Reset and Initialization	20
2.5. Timing Specifications	21
2.5.1. Input Timing	22
2.5.2. Output Timing	22
3. IP Generation, Simulation, and Validation	24
3.1. Generating the IP	24
3.2. Running Functional Simulation	25
3.3. Constraining the IP	27
3.4. IP Evaluation	27
4. Ordering Part Number	28
Appendix A. Resource Utilization	29
Appendix B. Limitations	31
References	32
Technical Support Assistance	33
Revision History	34

Figures

Figure 1.1. Byte-to-Pixel IP General Diagram.....	6
Figure 2.1. Byte-to-Pixel IP Functional Diagram	8
Figure 2.2. Case of AXI-Stream Subordinate and Pixel-side Native Interfaces	9
Figure 2.3. Case of Byte-side Native and AXI-Stream Manager Interfaces.....	9
Figure 2.4. Case of AXI4-Stream Interface Only.....	10
Figure 2.5. Case of Native Interface Only	10
Figure 2.6. Byte-to-Pixel IP FIFO Diagram	19
Figure 2.7. Timing Diagram between Inputs and Outputs for DSI	21
Figure 2.8. Timing Diagram between Inputs and Outputs for CSI-2	21
Figure 2.9. Input Timing Diagram for Byte-side	22
Figure 2.10. Output Timing Diagram from a DSI Input	22
Figure 2.11. Output Timing Diagram from a CSI-2 Input	22
Figure 2.12. AXI4 Stream Subordinate Interface Diagram.....	23
Figure 2.13. AXI4-Stream Manager Interface Diagram.....	23
Figure 3.1. Configure Block of Byte-to-Pixel Converter	24
Figure 3.2. Synthesizing the Design	25
Figure 3.3. Simulation Wizard.....	25
Figure 3.4. Adding and Reordering Source	26
Figure 3.5. Adding Constraint	27

Tables

Table 1.1. Quick Facts	6
Table 2.1. Byte-to-Pixel IP Ports.....	11
Table 2.2. Attributes Table	13
Table 2.3. Supported Configurations for DSI	15
Table 2.4. Supported Configuration for CSI-2.....	17
Table 2.5. Pixel and Byte Count Restriction.....	18
Table A.1. Device and Tool Tested.....	29
Table A.2. Resource Utilization	29

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
AXI	Advance eXtensible Interface
CSI-2	Camera Serial Interface-2
DSI	Display Serial Interface
FPGA	Field-Programmable Gate Array
RTL	Register Transfer Language

1. Introduction

Lattice Semiconductor Byte-to-Pixel Converter IP converts CSI-2/DSI standard based video payload packets from D-PHY Receiver Module output to pixel format. In addition, Byte-to-Pixel Converter IP generates camera/video control signals in the pixel domain, based on CSI-2 or DSI synchronization packets, as shown in [Figure 1.1](#). Byte-to-Pixel IP accepts CSI-2/DSI standard based video payload packets and generates Pixel Format output.

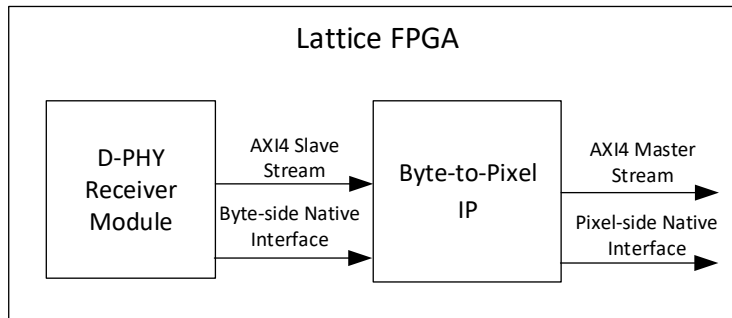


Figure 1.1. Byte-to-Pixel IP General Diagram

1.1. Quick Facts

[Table 1.1](#) presents a summary of the IP Core.

Table 1.1. Quick Facts

IP Requirements	Supported FPGA Family	CrossLink™-NX, Certus™-NX, CertusPro™-NX, Lattice Avant, MachXO5™-NX
Resource Utilization	Targeted Devices	LIFCL-40, LIFCL-17, LFD2NX-40, LFD2NX-17, LFCPNX-100, LAV-AT-500E, LFMXO5-25, LIFCL-33
	Supported User Interface	AXI4-Stream Interface
	Resources	See Table A.2 .
Design Tool Support	Lattice Implementation	IP Core v1.0.x – Lattice Radiant™ software 2.0 IP Core v1.1.x – Lattice Radiant software 2.1 IP Core v1.2.x - Lattice Radiant Software 2.2 or later
	Synthesis	Lattice Synthesis Engine (LSE) Synopsys® Synplify Pro® for Lattice
	Simulation	For a list of supported simulators, see the Lattice Radiant Software User Guide .

1.2. Features

The Byte-to-Pixel Converter IP supports:

- MIPI DSI compatible video formats
- MIPI CSI-2 compatible video formats
- 1-, 2-, or 4-lane inputs
- 8-bit (gear 8) or 16-bit (gear 16) inputs per lane
- 1, 2, or 4 output pixels per pixel clock cycle
- Burst mode, non-burst mode with sync events and non-burst mode with sync pulse

1.3. Conventions

1.3.1. Nomenclature

The nomenclature used in this document is based on Verilog HDL.

1.3.2. Signal Names

Signal Names that end with:

- `_n` are active low
- `_i` are input signals
- `_o` are output signals
- `_io` are bi-directional input/output signals

2. Functional Description

2.1. Overview

The purpose of the Byte-to-Pixel Converter IP is to convert D-PHY CSI-2/DSI standard based byte data stream to standard pixel data format. Thus, the Byte-to-Pixel Converter IP provides a bridge between CSI-2/DSI standard based input byte data stream and pixel format output data stream.

The AXI4-Stream of the Byte Domain Inputs/Outputs are used for receiving video payload packets.

Figure 2.1 shows the general block diagram of Byte-to-Pixel IP. FIFO, AXI4 Device Subordinate, and AXI4 Device Manager are used to synchronize the incoming D-PHY data bytes to the pixel clock domain.

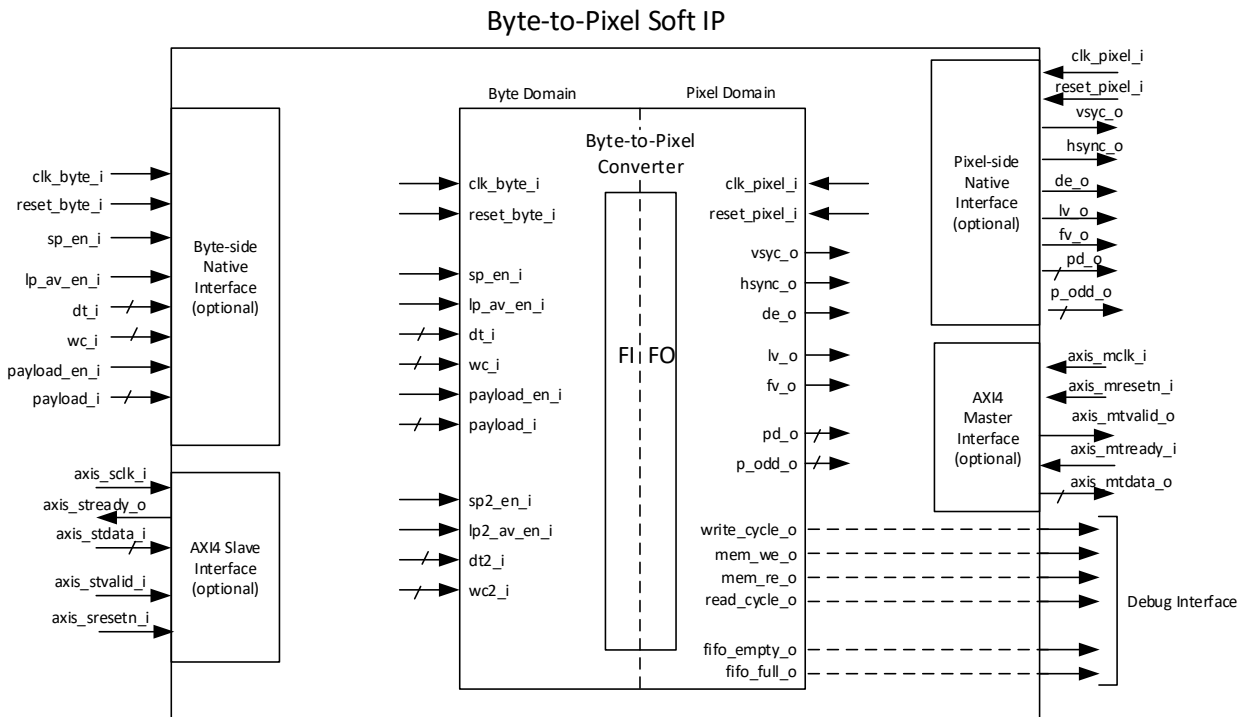


Figure 2.1. Byte-to-Pixel IP Functional Diagram

Various possible interface configurations are shown in Figure 2.2, Figure 2.3, Figure 2.4, and Figure 2.5.

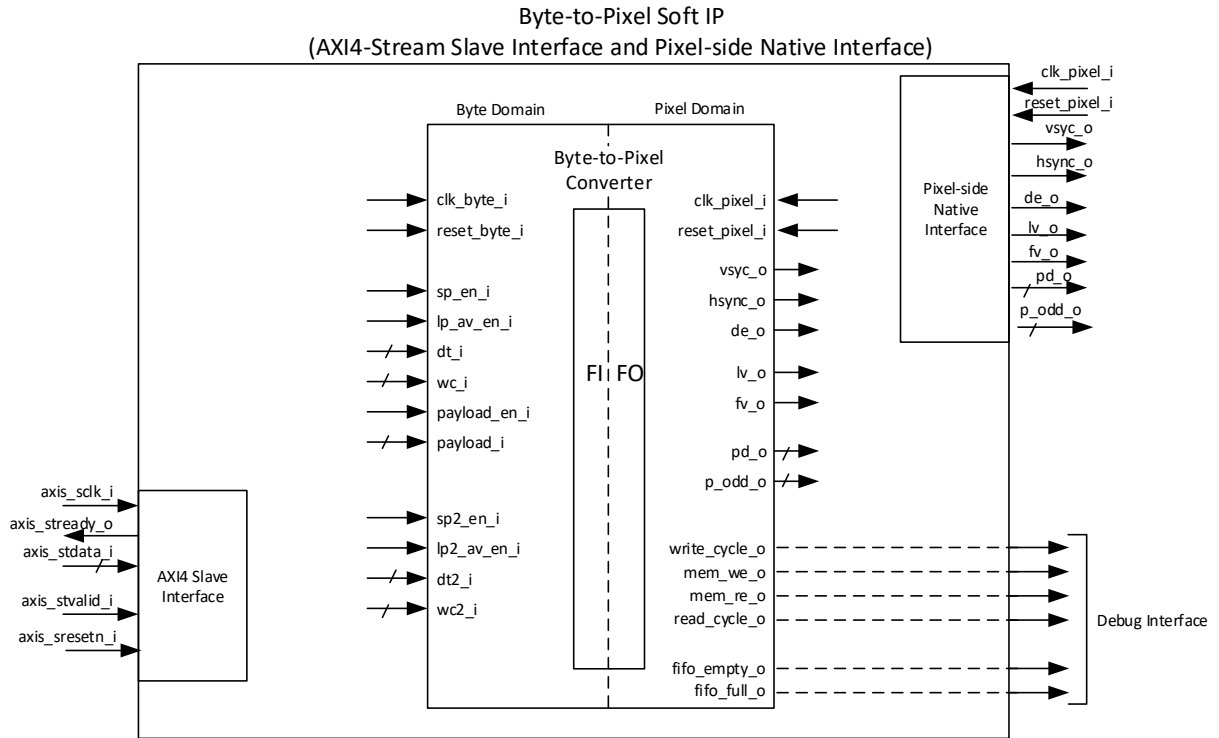


Figure 2.2. Case of AXI-Stream Subordinate and Pixel-side Native Interfaces

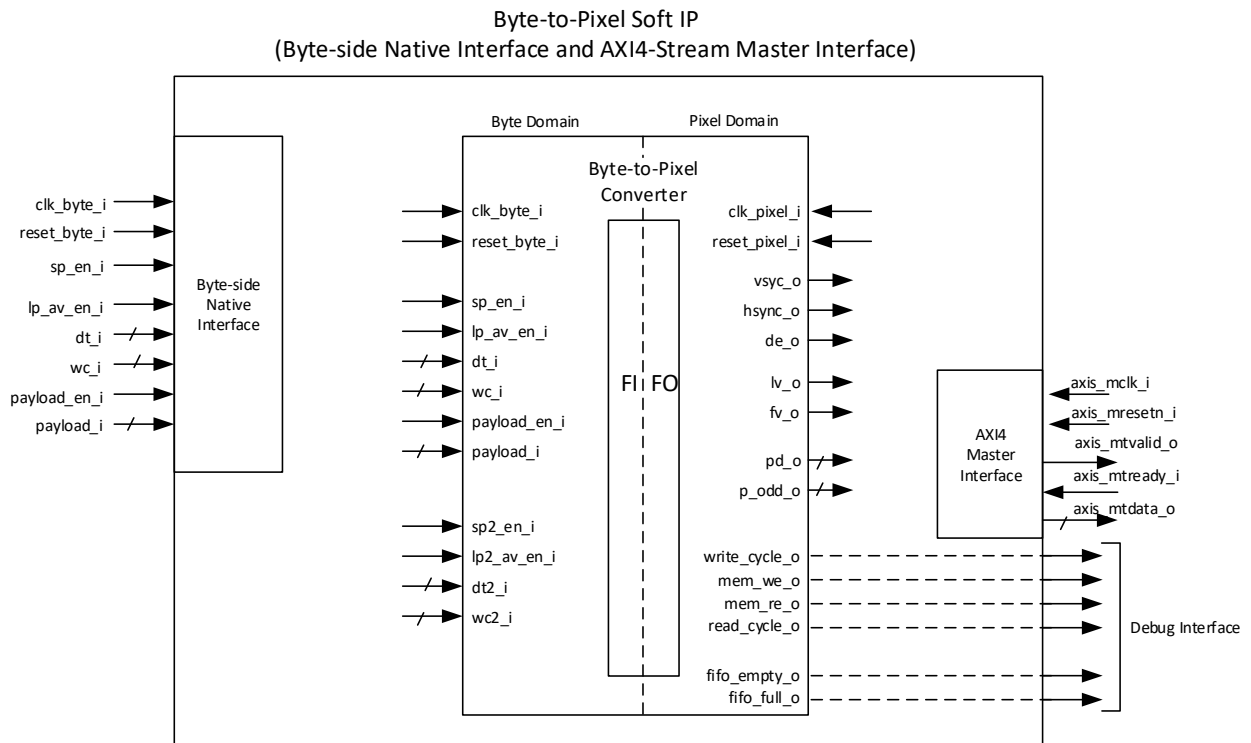


Figure 2.3. Case of Byte-side Native and AXI-Stream Manager Interfaces

Byte-to-Pixel Soft IP (AXI4-Stream Interface only)

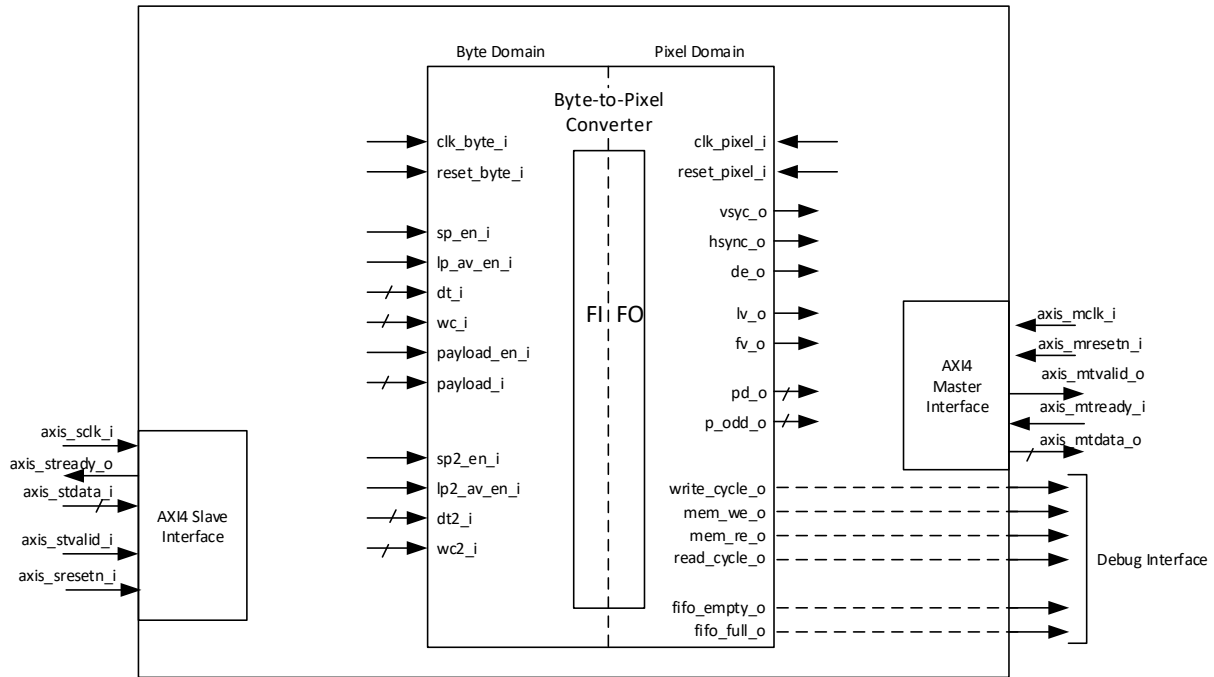


Figure 2.4. Case of AXI4-Stream Interface Only

Byte-to-Pixel Soft IP (Native Interface only)

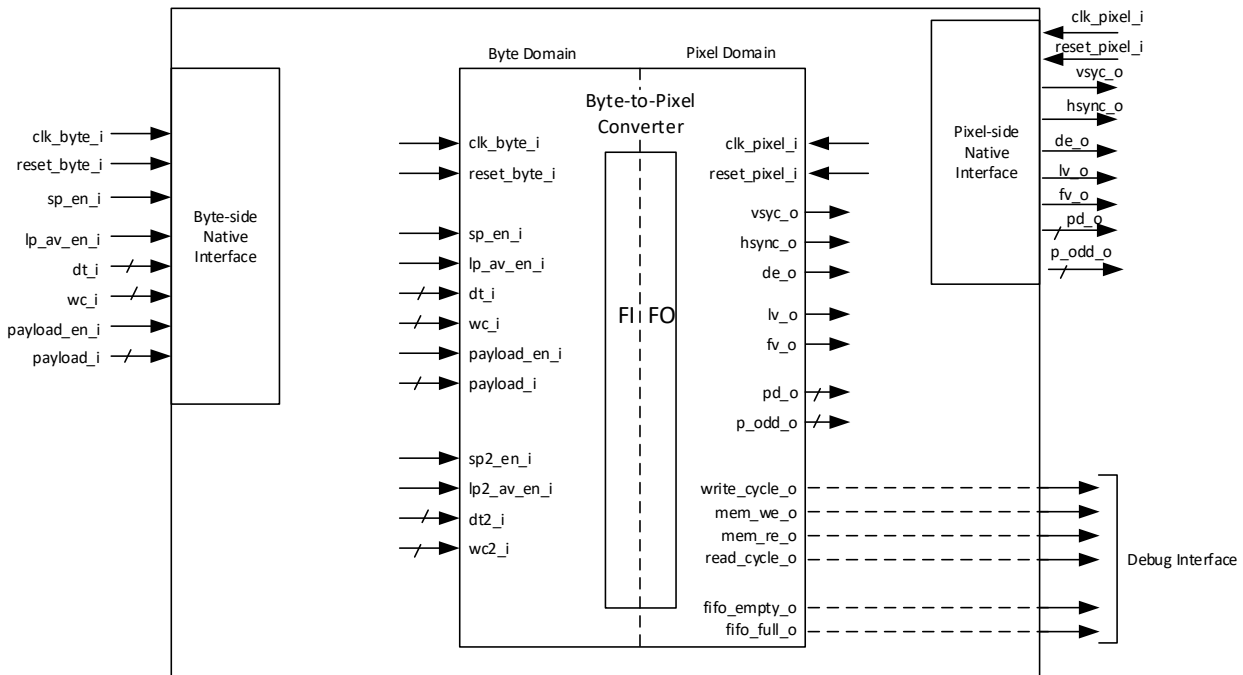


Figure 2.5. Case of Native Interface Only

2.2. Signal Description

Table 2.1 lists the top Input and Output signals and their descriptions for the Byte-to-Pixel IP.

Table 2.1. Byte-to-Pixel IP Ports

Port Name	Direction	Description
Byte Domain		
Native Interface		
clk_byte_i	In	Clock for the Rx byte clock domain logic
reset_byte_n_i	In	Active low signal to reset the logic in the clk_byte_i domain
sp_en_i	In	Active high pulse to indicate a valid short packet in the Rx side
lp_av_en_i	In	Active high pulse to indicate an active video long packet in the Rx side. The byte2pixel module prepares for the arrival of the video stream.
dt_i[5:0]	In	Data type field of the D-PHY Rx header packet
wc_i[15:0]	In	Word Count field of the D-PHY Rx header packet
payload_i[NUM_RX_LANE*RX_GEAR-1:0]	In	This is the active video data stream. The width of the data bus depends on the gearing and the number of D-PHY Rx lanes. See Table 2.2 for possible values for NUM_RX_LANE (Number of RX Lanes) and RX_GEAR (RX Gear).
payload_en_i	In	Active high payload valid indicator
sp2_en_i	In	This is valid only for gear 16, 4-lane configuration. Active high pulse to indicate a reception of a second valid short packet in the same byte clock cycle.
lp2_av_en_i	In	This is valid only for gear 16, 4-lane configuration. Active high pulse to indicate a second valid active video long packet in the same byte clock cycle.
dt2_i[5:0]	In	This is valid only for gear 16, 4-lane configuration. Data type field of the second D-PHY RX header packet.
wc2_i[15:0]	In	This is valid only for gear 16, 4-lane configuration. Word Count field of the second D-PHY RX header packet.
pixcnt_c_o[18:0]	Out	This is an internal net with critical path. It is ported out so that constraints can still be applied to this signal even on encrypted IP. This port may be left unconnected.
pix_out_cntr_o[15:0]	Out	
wc_pix_sync_o[15:0]	Out	
AXI4-Stream Subordinate Interface		
axis_sclk_i	In	AXI4-Stream Subordinate clock
axis_sresetn_i	In	Active low signal to reset the logic in the AXI4-Stream Subordinate domain
axis_svalid_i	In	AXI Stream Subordinate valid input signal
axis_sready_o	Out	AXI Stream Subordinate ready output signal. Currently, the value of the signal is always set to 1.
axis_sdata_i[SLAVE_DATA_W-1:0]	In	AXI Stream Subordinate data*(See Notes at the end of this table)
Pixel Domain		
Native Interface		
clk_pixel_i	In	Clock for the pixel domain logic
reset_pixel_n_i	In	Active low signal to reset the logic in the clk_pixel_i domain
vsync_o	Out	VSYNC signal for DSI. Active High if <i>Camera/Display Control Polarity</i> attribute is Positive. Otherwise, this is active Low.
hsync_o	Out	HSYNC signal for DSI. Active High if <i>Camera/Display Control Polarity</i> attribute is Positive. Otherwise, this is active Low.
fv_o	Out	Frame Valid signal for CSI-2. Active High if <i>Camera/Display Control Polarity</i> attribute is Positive. Otherwise, this is active

Port Name	Direction	Description
		Low.
lv_o	Out	Line Valid signal for CSI-2. Active High if <i>Camera/Display Control Polarity</i> attribute is Positive. Otherwise, this is active Low.
de_o	Out	Data Enable signal for DSI. Active high if Camera/Display Control Polarity attribute is Positive E. Otherwise, this is active low.
pd_o[PD_BUS_WIDTH*NUM_TX_CH-1:0]	Out	Pixel data output. The pixel_width may be 8, 10, 12, 18, 24, 36, 48, 72, or 96 bits. See Table 2.2 for possible values for PD_BUS_WIDTH (Data Type Width) and NUM_TX_CH (Number of Output Pixel Lanes).
p_odd_o[1:0]	Out	This signal is used to indicate the valid pixels for the last valid pixel data cycle in case of multiple pixel outputs per pixel clock cycle. This is a single bit signal for a 2-pixel output configuration, or a 2-bit bus for a 4-pixel output configuration. 00 – All pixels are valid 01 – Only the first pixel (LSB) is valid 10 – Only the lower two pixels in the lower bits are valid 11 – The last pixel (MSB) is not valid
AXI4-Stream Manager Interface		
axis_mclk_i	In	Clock for the pixel domain logic (input)
axis_mresetn_i	In	Active low signal to reset the logic in the axis_mclk_i domain
axis_mvalid_o	Out	AXI Manager Stream valid output signal
axis_mready_i	In	AXI Manager Stream ready input signal
axis_mdata_o[MASTER_DATA_W-1:0]	Out	AXI Manager Stream data (<i>See Notes at the end of this table</i>)
Debug Interface		
write_cycle_o[3:0]	Out	Payload data write cycle (debug only)
mem_we_o	Out	Payload data Write Enable, active High (debug only)
mem_re_o	Out	Payload data Read Enable, active High (debug only)
read_cycle_o[1:0]	Out	Pixel data read cycle (debug only)
fifo_empty_o	Out	Indicates FIFO empty condition (debug only).
fifo_full_o	Out	Indicates FIFO full condition (debug only).

***Notes:**

- $MASTER_DATA_W = p_odd_o + pd_o = 2 + (PD_BUS_WIDTH * NUM_TX_CH)$
- For the case when $NUM_RX_LANE * RX_GEAR \neq 64$:
 $SLAVE_DATA_W = dt_i + wc_i + payload_i = 6 + 16 + (NUM_RX_LANE * RX_GEAR)$
- For the case when $NUM_RX_LANE * RX_GEAR = 64$:
 $SLAVE_DATA_W = dt_i + wc_i + dt2_i + wc2_i + payload_i = 6 + 16 + 6 + 16 + 64 = 108$

2.3. Attributes Summary

Table 2.2 provides the list of user-selectable and compile time configurable attributes for the Byte-to-Pixel Converter IP. The attributes are specified using the Byte-to-Pixel Converter IP Configuration user interface in Lattice Radiant.

Table 2.2. Attributes Table

User Interface Config. Category	Attribute	Selectable Values	Defaults	Dependency On Other Attributes	Additional Requirements
General	Data Type	RGB565, RGB666, RGB666_LOOSE, RGB888, RAW8, RAW10, RAW12, RAW14, RAW16, YUV420_8, YUV420_8_CSPS, LEGACY_YUV420_8, YUV420_10, YUV420_10_CSPS, YUV422_8, YUV422_10, YCbCr422_16, YCbCr422_20_LOOSE, YCbCr422_24	RAW10	<ul style="list-style-type: none"> RX Interface Number of RX Lanes RX Gear Number of Output Pixel Lanes 	Use Table 2.3 and Table 2.4 for reference.
Byte Interface	RX Interface	DSI, CSI-2	CSI-2	—	—
	DSI Mode	Non-Burst Pulses, Non-Burst Events, Burst	Non-Burst Pulses	Grayed out unless <i>RX Interface</i> is <i>DSI</i> .	Values are selectable only when <i>RX Interface</i> is <i>DSI</i> .
	Number of RX Lanes	1, 2, 4	1	<ul style="list-style-type: none"> Data Type RX Interface RX Gear Number of Output Pixel Lanes 	Use Table 2.3 and Table 2.4 for reference.
	RX Gear	8, 16	8	<ul style="list-style-type: none"> Data Type RX Interface Number of RX Lanes Number of Output Pixel Lanes 	Use Table 2.3 and Table 2.4 for reference.
	Byte Clock Frequency (MHz) [10–350]	10–350	10	—	—
	Enable AXI4-Stream Receiver Interface	Checked, Not Checked	Not Checked	—	—
	Receiver Data Rate	—	Calculated	Grayed out. <ul style="list-style-type: none"> Number of RX Lanes RX Gear Byte Side Clock Frequency 	—
Pixel Interface	Number of Output Pixel Lanes	1, 2, 4	1	<ul style="list-style-type: none"> Data Type RX Interface Number of RX Lanes RX Gear 	Use Table 2.3 and Table 2.4 for reference.

User Interface Config. Category	Attribute	Selectable Values	Defaults	Dependency On Other Attributes	Additional Requirements
	Camera/Display Control Polarity	Positive, Negative	Positive	—	—
	Number of HSYNC Pulses Inside VSYNC Active Region [3–1023]	3–1023	5	Grayed out unless <i>RX Interface</i> is <i>DSI</i> .	When the video mode is non-burst with sync events, this is used to determine the deassertion of the vsync_o signal. When non-burst with sync pulses, this is used only by the testbench for simulation; the actual vsync_o deassertion still depends on the reception of the vsync end packet.
	Number of Pixel Clock Cycles HSYNC is Active [3–1023]	3–1023	8	Grayed out unless <i>RX Interface</i> is <i>DSI</i> .	When the video mode is non-burst with sync events, this is used to determine the deassertion of the hsync_o signal. When non-burst with sync pulses, this is used only by the testbench for simulation; the actual hsync_o deassertion still depends on the reception of the hsync end packet.
	DSI Sync Packet Delay	5-1023	5	Grayed out unless <i>RX Interface</i> is <i>DSI</i> .	This is the number of Pixel Clock cycle to delay the assertion of the HSYNC and VSYNC signals.
	Pixel Clock Frequency (MHz) [10–500]	10–500	50	—	—
	Enable AXI4-Stream Transmitter Interface	Checked, Not Checked	Not Checked	—	—
	Transmitter Data Rate	—	500	Grayed out <ul style="list-style-type: none"> Pixel Data Width Number of Output Pixels Pixel Side Clock Frequency 	—

User Interface Config. Category	Attribute	Selectable Values	Defaults	Dependency On Other Attributes	Additional Requirements
FIFO	Manual Adjust	Checked, Not Checked	Not Checked	—	—
	Overflow/Underflow Threshold [1–65535]	1–65535	Calculated	Manual Adjust box is checked	—
	FIFO Depth [8–65536]	8–65536	Calculated	Manual Adjust box is checked	—
	Word Count [MIN_WC - 65535]	1–65535	5	Data Type	MIN_WC value depends on the Data Type. See column <i>Byte Count Restriction</i> for the actual value being used on Table 2.5 .
Miscellaneous	Enable Debug Ports	Checked, Not Checked	Not Checked	—	—

Table 2.3. Supported Configurations for DSI

Number of Output Pixel Lanes	D-PHY Lanes	RX Gearing	Data Type
1 output pixel	1 lane	8	YCbCr422 – 16 bit, YCbCr422 – 24 bit, YCbCr422 loosely packed – 20 bit, RGB565, RGB666, RGB666 loosely packed, RGB888
		16	YCbCr422 – 16 bit, YCbCr422 – 24 bit, YCbCr422 loosely packed – 20 bit, RGB565, RGB666, RGB666 loosely packed, RGB888
	2 lanes	8	YCbCr422 – 16 bit, YCbCr422 – 24 bit, YCbCr422 loosely packed – 20 bit, RGB565, RGB666, RGB666 loosely packed, RGB888
		16	YCbCr422 – 16 bit, YCbCr422 – 24 bit, YCbCr422 loosely packed – 20 bit, RGB565, RGB666, RGB666 loosely packed, RGB888
	4 lanes	8	YCbCr422 – 16 bit, YCbCr422 – 24 bit, YCbCr422 loosely packed – 20 bit, RGB565, RGB666, RGB666 loosely packed, RGB888

Number of Output Pixel Lanes	D-PHY Lanes	RX Gearing	Data Type
2 output pixels	1 lane	8	YCbCr422 – 16 bit, YCbCr422 – 24 bit, YCbCr422 loosely packed – 20 bit, RGB565
		16	YCbCr422 – 16 bit, YCbCr422 – 24 bit, YCbCr422 loosely packed – 20 bit, RGB565
	2 lanes	8	YCbCr422 – 16 bit, YCbCr422 – 24 bit, YCbCr422 loosely packed – 20 bit, RGB565
		16	YCbCr422 – 16 bit, YCbCr422 – 24 bit, YCbCr422 loosely packed – 20 bit, RGB565, RGB666, RGB666 loosely packed, RGB888
	4 lanes	8	YCbCr422 – 16 bit, YCbCr422 – 24 bit, YCbCr422 loosely packed – 20 bit, RGB565, RGB666, RGB666 loosely packed, RGB888
		16	YCbCr422 – 16 bit, YCbCr422 – 24 bit, YCbCr422 loosely packed – 20 bit, RGB565, RGB666, RGB666 loosely packed, RGB888
4 output pixels	4 lanes	16	YCbCr422 – 16 bit, YCbCr422 – 24 bit, YCbCr422 loosely packed – 20 bit, RGB565, RGB666, RGB666 loosely packed, RGB888

Table 2.4. Supported Configuration for CSI-2

Number of Output Pixel Lanes	D-PHY Lanes	RX Gearing	Data Type	
1 output pixel	1 lane	8	RGB565, 8-bit*, (See Notes below) 10-bit*, (See Notes below) RAW12, RAW14, RAW16, RGB888	
		16	RGB565, 10-bit, RAW12, RAW14, RAW16, RGB888	
	2 lanes	8	RGB565, 8-bit, 10-bit, RAW12, RAW14, RAW16, RGB888	
		16	RGB565, RAW12, RAW14, RAW16, RGB888	
	4 lanes	8	RGB565, 8-bit, 10-bit, RAW12, RAW14, RAW16 RGB888	
		16	RAW12, RAW14, RAW16, RGB565	
	2 output pixels	1 lane	8	RGB565, RAW12, RAW14, RAW16
			16	RGB565, 8-bit, RAW12, RAW14, RAW16
2 lanes		8	RGB565, 8-bit, 10-bit, RAW12, RAW14, RAW16	
		16	RGB565, 8-bit, 10-bit, RAW12, RAW14, RAW16, RGB888	
4 lanes		8	RGB565, 8-bit, 10-bit, RAW12, RAW14, RAW16 RGB888	
		16	RGB565, RAW12, RAW14, RAW16 RGB888	

***Notes:**

- Supported 8-bit CSI-2 data types are RAW8, YUV420 8-bit, Legacy YUV420 8-bit, YUV420 8-bit CSPS and YUV422 8-bit.
- Supported 10-bit CSI-2 data types are RAW10, YUV420 10-bit, YUV420 10-bit CSPS and YUV422 10-bit.

Table 2.5. Pixel and Byte Count Restriction

Data Type	Pixel Count Restriction	Byte Count Restriction
RGB565	multiple of 1	multiple of 2
RGB666	multiple of 4	multiple of 9
RGB666, Loosely packed	multiple of 1	multiple of 3
RGB888	multiple of 1	multiple of 3
RAW8	multiple of 1	multiple of 1
Legacy YUV420 8-bit	multiple of 2	multiple of 3
YUV420 8-bit	multiple of 2	multiple of 4
YUV422 8-bit	multiple of 2	multiple of 4
RAW10	multiple of 4	multiple of 5
YUV420 10-bit	multiple of 4	multiple of 10
YUV422 10-bit	multiple of 2	multiple of 5
RAW12	multiple of 2	multiple of 3
YCbCr422 16-bit	multiple of 2	multiple of 4
YCbCr422 24-bit	multiple of 4	multiple of 6
YCbCr422 Loosely packed 20-bit	multiple of 4	multiple of 6
RAW14	multiple of 4	multiple of 7
RAW16	multiple of 1	multiple of 2

2.4. Modules Description

2.4.1. Byte-to-Pixel Converter

Byte-to-Pixel Converter is FIFO based block, which provides conversion of D-PHY CSI-2 or DSI video payload packets to standard pixel format through generic CSI-2/DSI standard based input/output signaling.

2.4.2. AXI4 Stream Subordinate

In byte domain, AXI4-Stream Subordinate provides transmission of payload packets to Byte-to-Pixel Converter module. The `axis_sready_o` is always set to '1'. When `axis_svalid_i` asserted to 1, then `axis_sdata_o` receives the payload video stream.

2.4.3. AXI4 Stream Manager

In pixel domain, AXI4-Stream Manager provides transmission of data converted to pixel format.

For DSI and CSI2 modes, the output data is the concatenation of two signals {`p_odd_o`, `pd_o`} respectively in that order. As the `p_odd_o` is a two bit signal, the total width of AXI4 stream data (`axis_mdata_o`) is equal to $2 + \text{pixel_width} \times \text{number_of_pixels}$. Data valid (`axis_mvalid_o`) becomes active, when `de_o` signals becomes '1'.

2.4.4. FIFO Implementation

The FIFO depth is defined based on the design configuration, as shown in Table 2.2. The width of the FIFO is the lowest multiple of the output pixel data width that is greater than or equal to the input bus width. This determines the number of pixels that are grouped together and written to the same FIFO address. In order to avoid the FIFO full or empty states, the Data Safe Zone is defined by setting data Overflow/Underflow Threshold attribute, as shown in Table 2.2. The threshold value is used to trigger the start of reading from the FIFO.

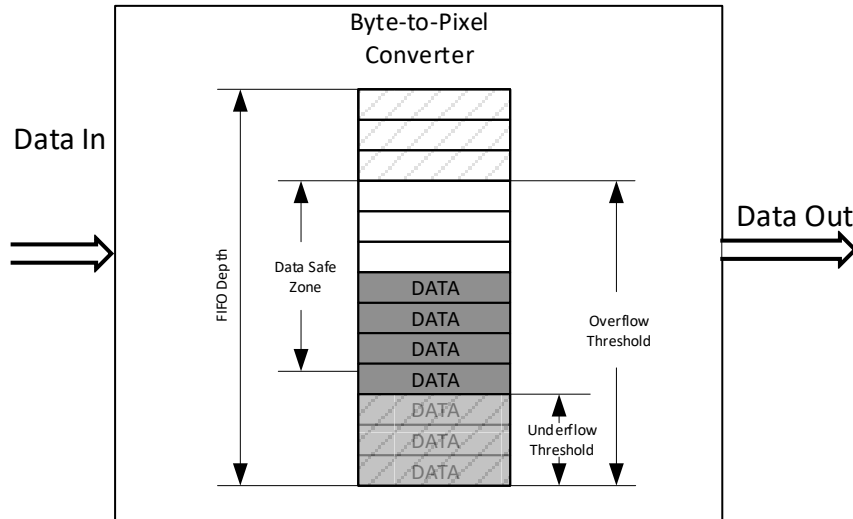


Figure 2.6. Byte-to-Pixel IP FIFO Diagram

The FIFO has the following three attributes: FIFO_Width, FIFO_Depth, and FIFO_Threshold.

The FIFO_Threshold is used to synchronize data flows between byte and pixel sides. Data transfer on pixel side is started after the FIFO_Threshold is reached.

The FIFO_Width, FIFO_Depth and FIFO_Threshold depend on several attributes listed below:

- NUM_RX_LANE – Number of Rx Lanes (available on the user interface)
- RX_GEAR – Rx Gear (available on the user interface)
- clk_byte_i – Frequency on Byte side (not available on the user interface)
- NUM_TX_CH – Number of Output Pixel Lanes (available on the user interface)
- PD_BUS_WIDTH – Pixel Data Bus Width (not available on user interface, however it depends only on Data Type which is available on the user interface)
- clk_pixel_i – Frequency on Pixel side (not available on the user interface)
- WC – Word Count (number of bytes in a line transaction, not available on the user interface)

Some of the parameters in the list are not available in the user interface. As such, the FIFO parameters cannot be directly set through the user interface.

Denote Rx_width as the word width on byte side and Tx_width as the word width on pixel side. It can be calculated as:

$$\begin{aligned} \text{Rx_width} &= \text{NUM_Rx_LANE} \times \text{Rx_GEAR} \\ \text{Tx_width} &= \text{PD_BUS_WIDTH} \times \text{NUM_Tx_CH} \end{aligned}$$

The FIFO Width depends on both Rx_width and Tx_width values. It is the minimum multiple of Tx_width, which is not less than Rx_width. So,

$$\text{FIFO_Width} = \text{Tx_width} \times \text{CEIL}(\text{Rx_width} / \text{Tx_width})$$

Also, denote the Data Flow Speed on byte side as Rx_rate and the Data Flow Speed on pixel side as Tx_Rate. It can be calculated as:

$$\begin{aligned} \text{Rx_rate} &= \text{Rx_width} \times \text{clk_byte_i} \\ \text{Tx_rate} &= \text{Tx_width} \times \text{clk_pixel_i} \end{aligned}$$

The FIFO Depth (minimum value) and FIFO Threshold depend on the ratio (N_ratio) of the Data Flow Speed on the Byte side (RX_rate) and Data flow speed on the Pixel side (TX_rate). RX_rate should not be less than TX_rate.

The value of FIFO Depth and FIFO Threshold is doubled when using data types YUV420_8 and YUV420_10, this is to make the even lines data transaction fit within the FIFO.

$$N_ratio = RX_rate / TX_rate$$

FIFO Depth and FIFO Threshold can be calculated by the following formulas:

```
IF (TX_rate == RX_rate)
```

```
FIFO_Depth = 16;
```

```
FIFO_Threshold = 4 ELSE
```

```
FIFO_Depth = 8 × WC × (N_ratio-1)/(N_ratio × FIFO_Width) + 1;
```

```
FIFO_Threshold = FIFO_Depth -1;
```

2.4.5. Reset and Initialization

Active low reset with synchronous release is used in the design. This is the system reset input connected to the Byte-to-Pixel Converter. Follow this initialization and reset sequence:

1. Assert active low system reset for at least three clock cycles of the slower clock (pixel clock or byte clock).
2. Byte-to-Pixel Converter is ready to process data after reset.

2.5. Timing Specifications

This section contains operational timing diagrams applicable to the Byte-to-Pixel IP interfaces. Figure 2.7 shows the timing between input and output for DSI.

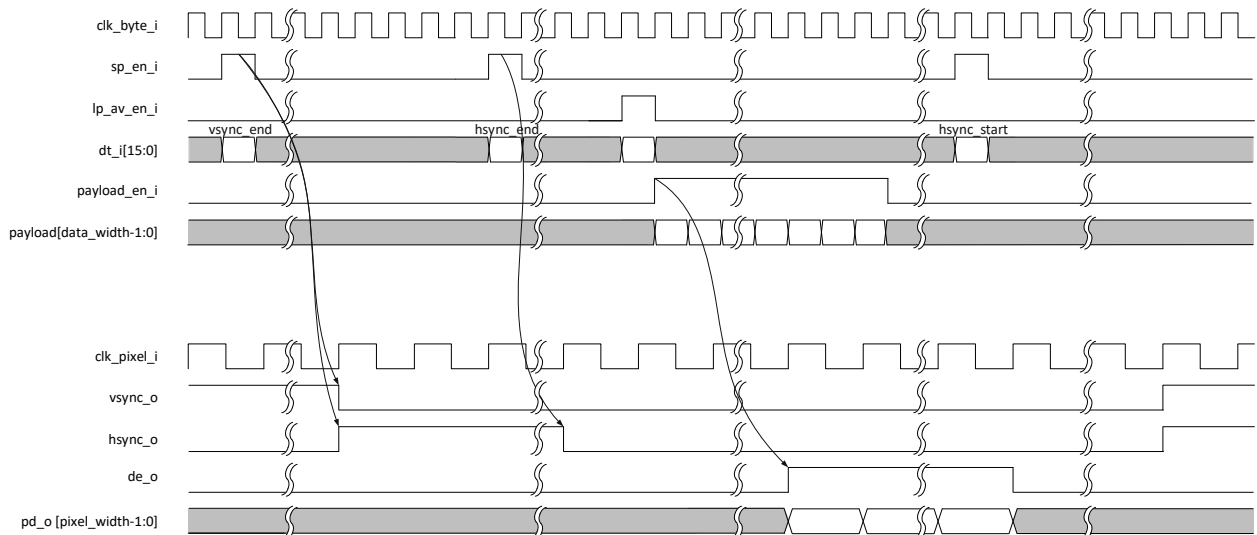


Figure 2.7. Timing Diagram between Inputs and Outputs for DSI

Reception of a VSYNC start packet triggers the assertion of both hsync_o and vsync_o signals. VSYNC end packets, on the other hand, trigger the deassertion of the vsync_o signal and the assertion of hsync_o signal.

Figure 2.8 shows the timing between input and output for CSI-2.

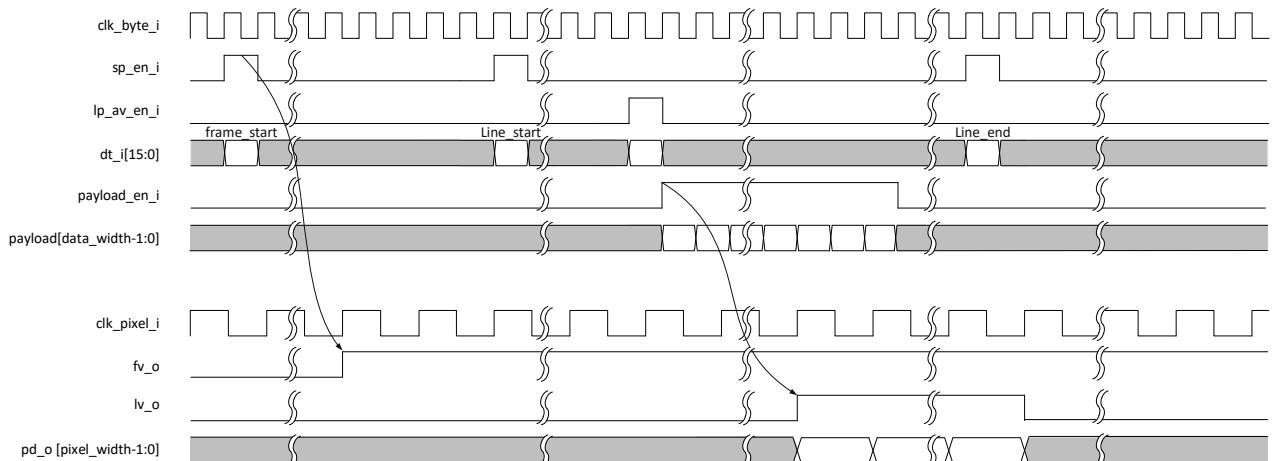


Figure 2.8. Timing Diagram between Inputs and Outputs for CSI-2

The behavior of the output synchronization signals (frame and line valid for CSI-2, and VSYNC and HSYNC for DSI) depend on the reception of the corresponding short packets. Due to the crossing of clock domains, pulse width and intervals between pulses may vary.

The pixel data is buffered and processed differently than the sync packets. Because of this, the pixel data might come out later than the sync signals. For the DSI protocol, the DSI Sync Packet Delay attribute may be increased to reduce the skew between the sync signals and the pixel data.

2.5.1. Input Timing

Figure 2.9 shows the timing diagram of the interface at the receiver side. The assertion of the payload_en_o with respect to the lp_av_en_i may vary depending on the gearing and number of lanes. The signals dt_i, vc_i and wc_i must be valid with the assertion of the lp_av_en_i.

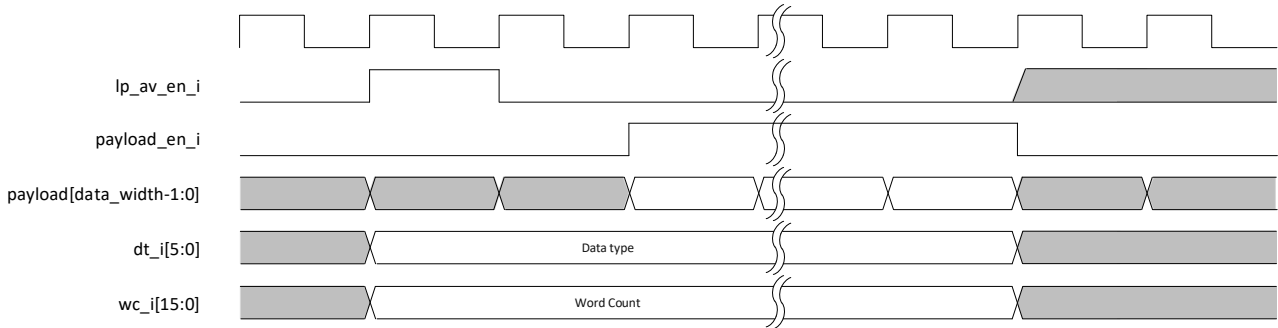


Figure 2.9. Input Timing Diagram for Byte-side

2.5.2. Output Timing

The output timing from a DSI input is shown in Figure 2.10 while the timing from a CSI-2 input is shown in Figure 2.11.

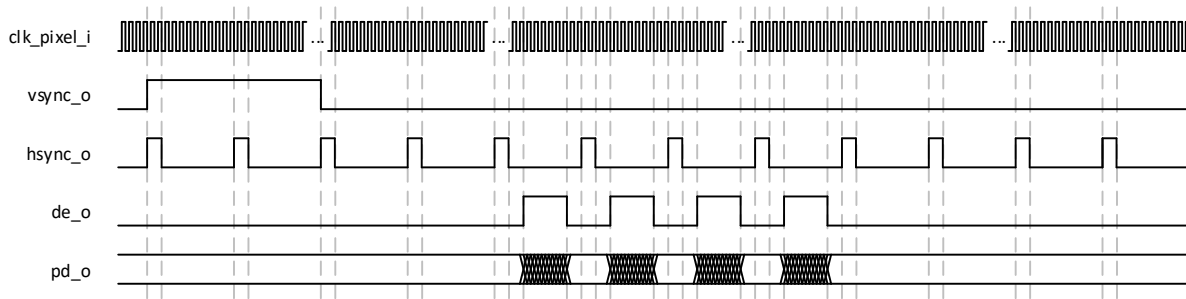


Figure 2.10. Output Timing Diagram from a DSI Input

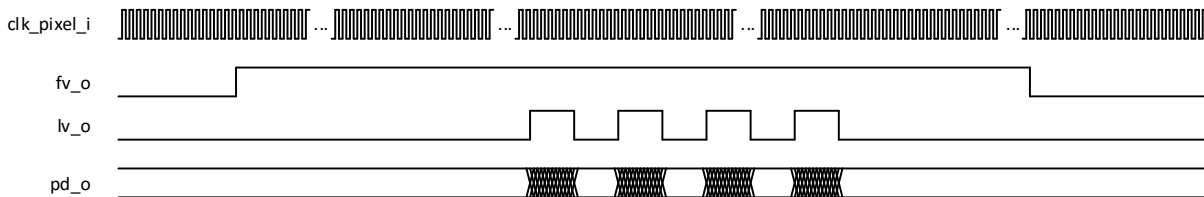


Figure 2.11. Output Timing Diagram from a CSI-2 Input

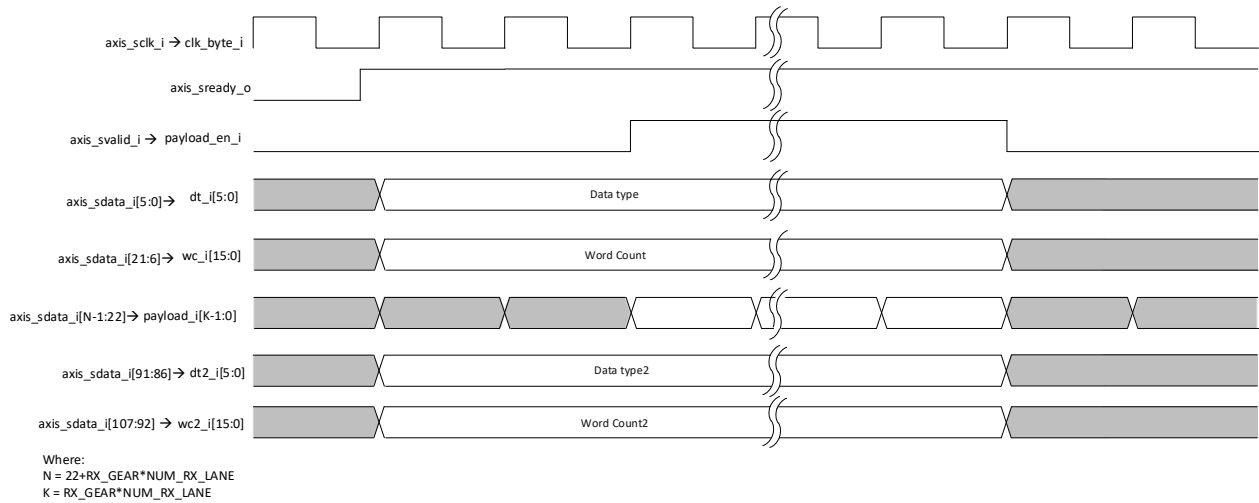


Figure 2.12. AXI4 Stream Subordinate Interface Diagram

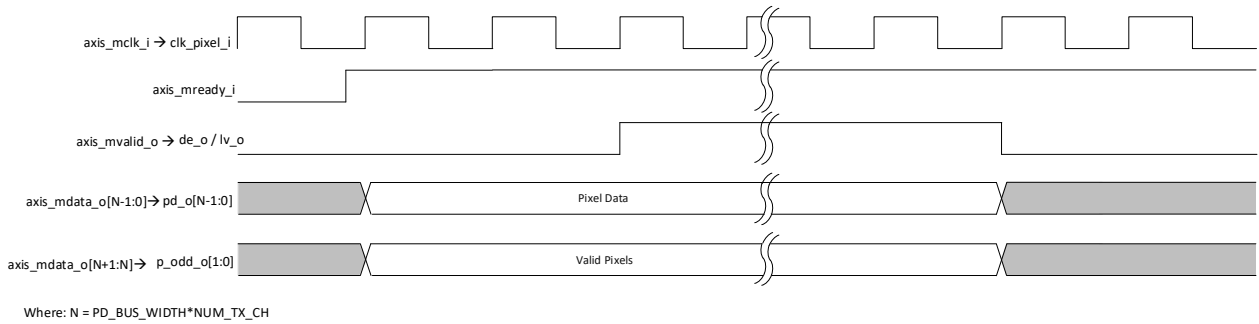


Figure 2.13. AXI4-Stream Manager Interface Diagram

3. IP Generation, Simulation, and Validation

This chapter provides information on how to generate, simulate, and validate Byte-to-Pixel Converter IP Core using Lattice Radiant software. For more on Lattice Radiant software, refer to the [Lattice Radiant Software User Guide](#) and relevant Lattice tutorials.

3.1. Generating the IP

Lattice Radiant software allows the user to generate and customize modules and IPs and integrate them into the device architecture. The procedure for generating Byte-to-Pixel Converter IP in Lattice Radiant software is described below.

To generate the Byte-to-Pixel Converter IP:

1. In the **Module/IP Block Wizard**, create a new Lattice Radiant software project for Byte-to-Pixel Converter module.
2. In the dialog box of the **Module/IP Block Wizard** window, configure the Byte-to-Pixel Converter module according to custom specifications using drop-down menus and check boxes. As a sample configuration, see [Figure 3.1](#). For configuration options, see [Table 2.2](#).

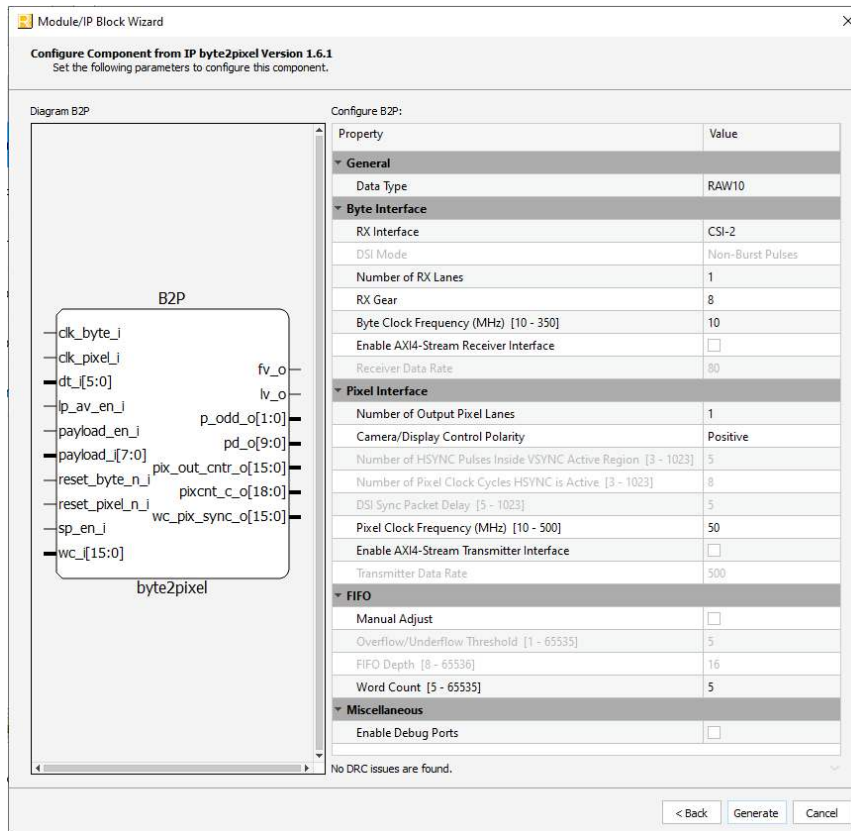


Figure 3.1. Configure Block of Byte-to-Pixel Converter

3. Click **Generate**. The **Check Generating Result** dialog box opens, showing design block messages and results.
4. Click the **Finish** button to generate the Verilog file.
5. After generating the design, it can be synthesized by clicking **Synthesize Design** located on the top left corner of the screen, as shown in [Figure 3.2](#).

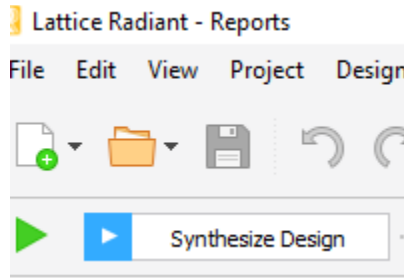



Figure 3.2. Synthesizing the Design

3.2. Running Functional Simulation

To run Verilog simulation:

1. Click the  button located on the **Toolbar** to initiate the **Simulation Wizard** shown in [Figure 3.3](#).

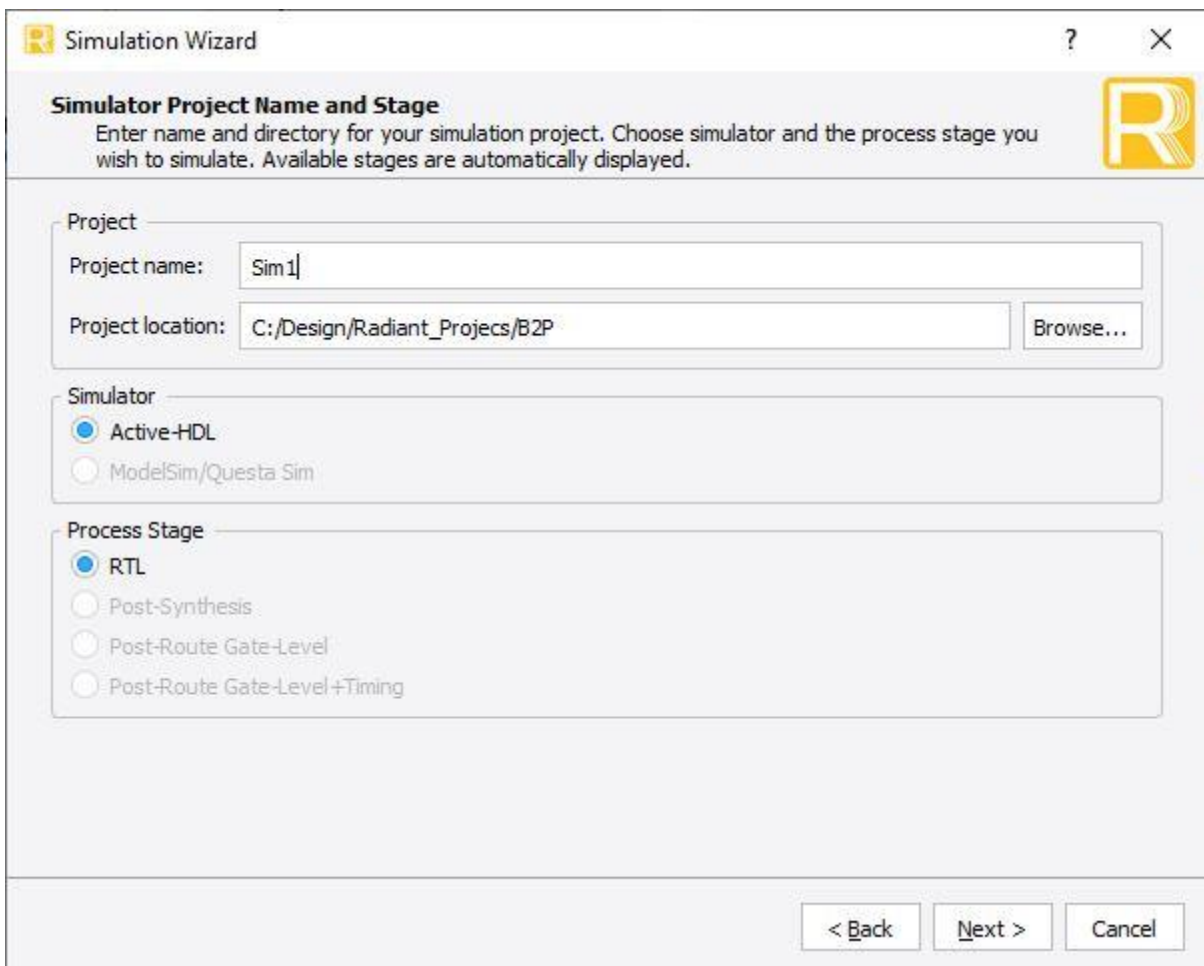


Figure 3.3. Simulation Wizard

2. Double-click **Next** to open the **Add and Reorder Source** window as shown in Figure 3.4.

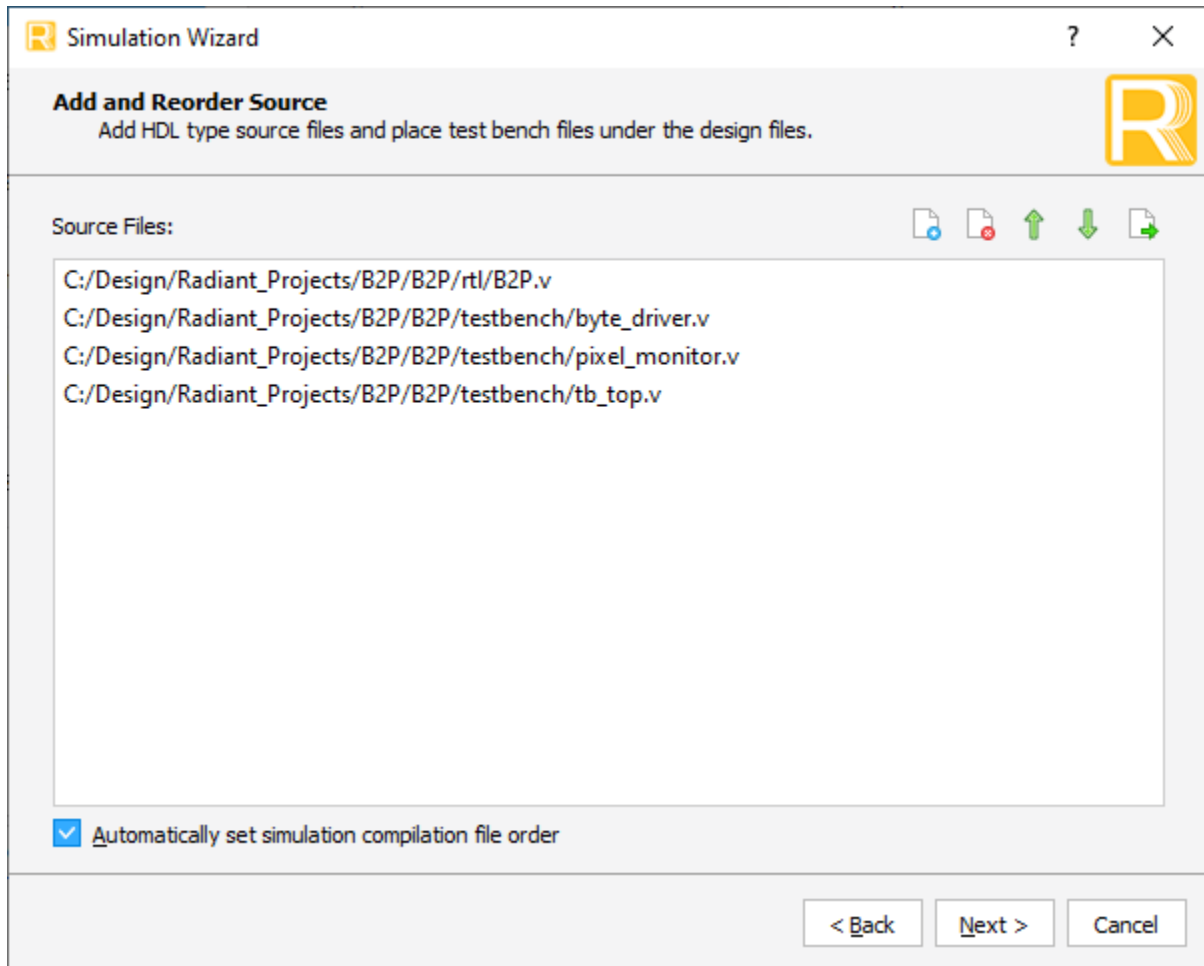


Figure 3.4. Adding and Reordering Source

3. Add the *tb_top.v* file from the *testbench* directory.
4. Click **Next** to run the simulation.

3.3. Constraining the IP

The Byte to Pixel Converter IP for Radiant has a default Pre-Synthesis Constraints file *.Idc located in <ip_instance_path>/constraints/<instance_name>.Idc that shows more details on how to constrain the IP. There is also a Post-Synthesis Constraints (*.pdc) file provided, located in <ip_instance_path>/eval/postsynconstraints.pdc as shown in Figure 3.5.

To properly constrain the byte to pixel design post synthesis:

1. By default, the periods of the clocks are for 200 MHz designs. Adjust those based on the actual design to relax the timing.
2. Modify the paths based on the generated netlist.

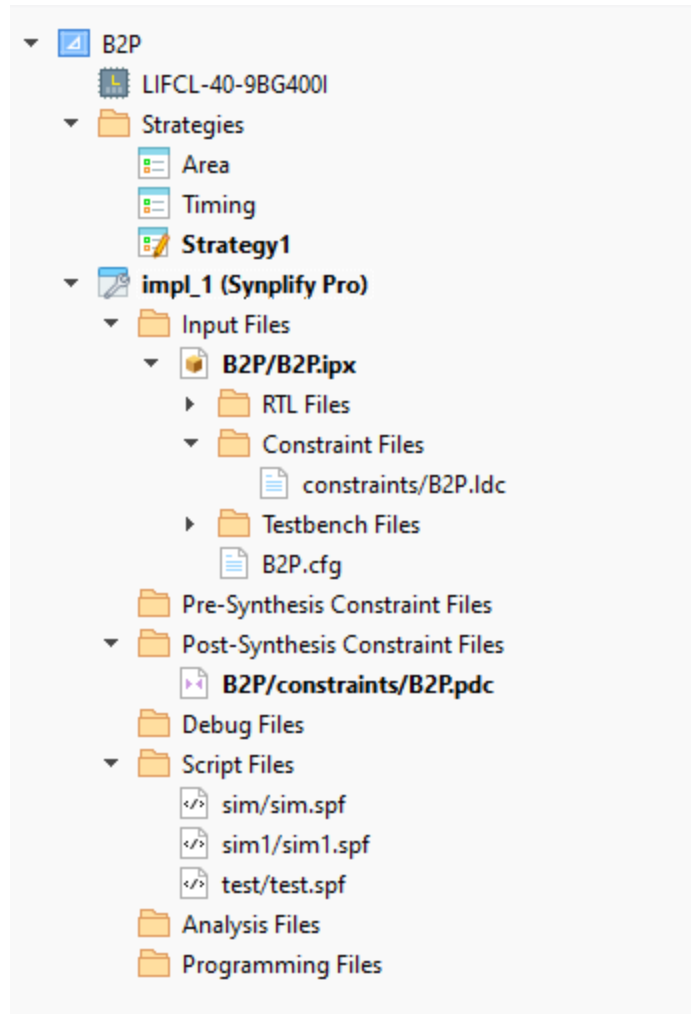


Figure 3.5. Adding Constraint

3.4. IP Evaluation

The IP Core supports Lattice’s IP evaluation capability when used with Lattice Avant. This makes it possible to create versions of the IP core that operate in hardware for a limited period (approximately four hours) without requiring the purchase of an IP license. The IP evaluation capability may be enabled/disabled in the **Strategies Dialog Box** which is disabled by default. To change this setting, go to Strategies > Strategy1 (active strategy) > Bitstream.

4. Ordering Part Number

The Ordering Part Number (OPN) for this IP Core are the following:

- BYTE-PIXEL-CNX-U – Byte to Pixel for CrossLink-NX – Single Design License
- BYTE-PIXEL-CNX-UT – Byte to Pixel for CrossLink-NX – Site License
- BYTE-PIXEL-CTNX-U – Byte to Pixel for Certus-NX – Single Design License
- BYTE-PIXEL-CTNX-UT – Byte to Pixel for Certus-NX – Site License
- BYTE-PIXEL-CPNX-U – Byte to Pixel Converter for CertusPro-NX – Single Design License
- BYTE-PIXEL-CPNX-UT – Byte to Pixel Converter for CertusPro-NX – Site License
- BYTE-PIXEL-AVE-U – Byte to Pixel Converter for Lattice Avant-E – Single Design License
- BYTE-PIXEL-AVE-UT – Byte to Pixel Converter for Lattice Avant-E – Site License
- BYTE-PIXEL-AVE-US – Byte to Pixel Converter for Lattice Avant-E – 1-Year Subscription License
- BYTE-PIXEL-XO5-U – Byte to Pixel Converter for MachXO5-NX – Single Design License
- BYTE-PIXEL-XO5-UT – Byte to Pixel Converter for MachXO5-NX – Site License
- BYTE-PIXEL-XO5-US – Byte to Pixel Converter for MachXO5-NX – 1-Year Subscription License

Appendix A. Resource Utilization

Table A.1, Table A.2, Table A.3, and Table A.4 show the details about the device, tools used and resource utilization for a certain IP configuration.

For more information on Lattice Radiant software, visit the Lattice web site at www.latticesemi.com/Products/DesignSoftwareAndIP.

Table A.1. Device and Tool Tested

	Value
Diamond Software Version	Radiant 2.1 Production Build
Device Used	LIFCL-40-BCG400
Performance Grade	9_High-Performance_1.0V
Synthesis Tool	Synplify Pro (R) Q-2020.03LR, Build 134R, May 8 2020

Table A.2. Resource Utilization

Device	LUTs	Register	sysMem EBRs	Programmable I/O
CSI2,RAW10,Byte Side Clock Frequency 100 MHz, Pixel Side Clock Frequency 80 MHz, Word Count 720	369	287	1	51
CSI2,RGB888,Byte Side Clock Frequency 150 MHz, Pixel Side Clock Frequency 100 MHz, Word Count 720	399	324	1	73
CSI2,RGB888, Number of RX Lanes 4, Byte Side Clock Frequency 50 MHz, Pixel Side Clock Frequency 160 MHz, Word Count 2050	495	363	2	75
CSI2,RGB888, Number of RX Lanes 4, Byte Side Clock Frequency 112.5 MHz, Pixel Side Clock Frequency 150 MHz, Word Count 3600	421	386	2	89
DSI,RGB666, Number of RX Lanes 1, Byte Side Clock Frequency 108 MHz, Pixel Side Clock Frequency 96 MHz, Word Count 2160	534	337	1	68
DSI,RGB666, Number of RX Lanes 2, Byte Side Clock Frequency 140.625 MHz, Pixel Side Clock Frequency 125 MHz, Number of Output Pixels 2, Word Count 2160	649	427	1	102

Note: The *distributed RAM* utilization is accounted for in the total LUT4 utilization. The actual LUT4 utilization is distribution among *logic*, *distributed RAM*, and *ripple logic*.

Table A.3. Lattice Avant Device and Tool Tested

	Value
Diamond Software Version	Radiant 2023.1 Production Build
Device Used	LAV-AT-500E-3LFG1156C
Performance Grade	3
Synthesis Tool	Synplify Pro (R) Q-2023.1LR, Build 151R, February 7 2023

Table A.4. Resource Utilization using Lattice Avant

Device	LUTs	Register	sysMem EBRs	Programmable I/O
CSI2,RAW10,Byte Clock Frequency 100 MHz, Pixel Clock Frequency 80 MHz, Word Count 720	545	304	1	19
CSI2,RGB888,Byte Clock Frequency 150 MHz, Pixel Clock Frequency 100 MHz, Word Count 720	589	351	1	19
CSI2,RGB888, Number of RX Lanes 4, Byte Clock Frequency 50 MHz, Pixel Clock Frequency 160 MHz, Word Count 2052	670	433	1	19
CSI2,RGB888, Number of RX Lanes 4, Byte Clock Frequency 112.5 MHz, Pixel Clock Frequency 150 MHz, Word Count 3600	616	406	1	19
DSI,RGB666, Number of RX Lanes 1, Byte Clock Frequency 108 MHz, Pixel Clock Frequency 96 MHz, Word Count 2160	587	392	1	22
DSI,RGB666, Number of RX Lanes 2, Byte Clock Frequency 140.625 MHz, Pixel Clock Frequency 125 MHz, Word Count 2160	631	371	1	22

Note: The *distributed RAM* utilization is accounted for in the total LUT4 utilization. The actual LUT4 utilization is distribution among *logic*, *distributed RAM*, and *ripple logic*.

Appendix B. Limitations

Post-Route Gate-Level simulation may fail on some configuration when using LSE but passing using Synplify Pro.

References

For complete information on Lattice Radiant Project-Based Environment, Design Flow, Implementation Flow and Tasks, as well as on the Simulation Flow, see the [Lattice Radiant software](#) user guide.

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Document Revision 1.7, Lattice Radiant SW version 2023.1, April 2023

Section	Change Summary
All	Updated for inclusive language.
Inclusive Language	Added this section.
Introduction	<ul style="list-style-type: none"> In Table 1.1. Quick Facts: Added MachXO5-NX in the Supported FPGA Family field. Added LFMXO5-25 and LIFCL-33 in the Targeted Devices field.
Functional Description	<ul style="list-style-type: none"> Updated Table 2.1. Byte-to-Pixel IP Ports. Updated Table 2.2. Attributes Table.
IP Generation, Simulation, and Validation:	Updated Figure 3.1. Configure Block of Byte-to-Pixel Converter .
Ordering Part Number	Added part numbers for MachXO5-NX.
Appendix A. Resource Utilization	<ul style="list-style-type: none"> Added Table A.3. Lattice Avant Device and Tool Tested. Added Table A.4. Resource Utilization using Lattice Avant.
Technical Support Assistance	Added reference link to the Lattice Answer Database.

Document Revision 1.6, Lattice Radiant SW version 2022.1, November 2022

Section	Change Summary
Introduction	<ul style="list-style-type: none"> In Table 1.1. Quick Facts: Added Lattice Avant in the Supported FPGA Family field. Added LAV-AT-500E in the Targeted Devices field.
IP Generation, Simulation, and Validation:	<ul style="list-style-type: none"> Revised the title from 'IP Generation and Evaluation' to 'IP Generation, Simulation, and Validation'. Deleted the section 'Licensing the IP'. Revised the title of section 3.1 from 'Generation and Synthesis' to 'Generating the IP'. Added Constraining the IP section. Added IP Evaluation section.
Ordering Part Number	Added part numbers for Lattice Avant-E

Document Revision 1.5, Lattice Radiant SW version 3.1, November 2021

Section	Change Summary
Functional Description	<ul style="list-style-type: none"> Added DSI Sync Packet Delay to Table 2.2. Attributes Table. Updated the descriptions of timing diagrams in the Timing Specifications section.

Document Revision 1.4, Lattice Radiant SW version 3.0, June 2021

Section	Change Summary
Introduction	Updated Table 1.1 to include CertusPro-NX support.
Ordering Part Number	Added part number for CertusPro-NX.

Document Revision 1.3, Lattice Radiant SW version 2.1, November 2020

Section	Change Summary
Introduction	Updated reference to the Lattice Radiant Software User Guide.
Functional Description	<ul style="list-style-type: none"> Added ports to Table 2.1. Byte-to-Pixel IP Ports. Added RAW14 and RAW16 selectable values to Table 2.2. Attributes Table. Added RAW12, RAW14, and RAW16 data types to Table 2.4. Supported Configuration for CSI-2. Added RAW14 and RAW16 data types to Table 2.5. Pixel and Byte Count Restriction. Updated FIFO Implementation section.
Core Generation, Simulation, and Validation	Updated reference to the Lattice Radiant Software User Guide
References	Updated reference to the Lattice Radiant Software User Guide

Document Revision 1.2, Lattice Radiant SW version 2.0, August 2020

Section	Change Summary
Acronyms in This Document	Updated content.
Introduction	<ul style="list-style-type: none"> Updated Table 1.1. Updated Features section.
Functional Description	<ul style="list-style-type: none"> Updated Table 2.1, Table 2.2, Table 2.3, Table 2.4, and Table 2.5. Updated FIFO Implementation section.
IP Generation and Evaluation	<ul style="list-style-type: none"> Updated Figure 3.1 and Figure 3.4. Added Required Post-Synthesis Constraints section.
Appendix A. Resource Utilization	Updated section content including Table A.1 and Table A.2.
Appendix B. Limitations	Added this section.

Document Revision 1.1, Lattice Radiant SW version 2.0, February 2020

Section	Change Summary
Introduction	Updated Table 1.1 to add LIFCL-17 as targeted device.
Functional Description	<ul style="list-style-type: none"> Updated descriptions for p_odd_o[1:0], axis_mdata_o[MASTER_DATA_W-1:0], and for axis_sdata_i[SLAVE_DATA_W-1:0] in Table 2.1. Byte-to-Pixel IP Ports. Updated Transmitter and FIFO attributes in Table 2.2. Attributes Table. Changed caption to Figure 2.6. Byte-to-Pixel IP FIFO Diagram. Updated formulas in FIFO Implementation section.

Document Revision 1.0, Lattice Radiant SW version 2.0, November 2019

Section	Change Summary
All	Changed document status from Preliminary to final.
Introduction	65536axUpdated Table 1.1. Quick Facts.
Functional Description	Updated Receiver and Transmitter attributes in Table 2.2. Attributes Table.
Ordering Part Number	Added this section.
Appendix A. Resource Utilization	Added this section.

Document Revision 0.80, Lattice Radiant SW version 2.0, October 2019

Section	Change Summary
All	Preliminary release



www.latticesemi.com