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

User Guide

FPGA-IPUG-02079-1.2

August 2020

Disclaimers

Contents

Figures

Tables

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.

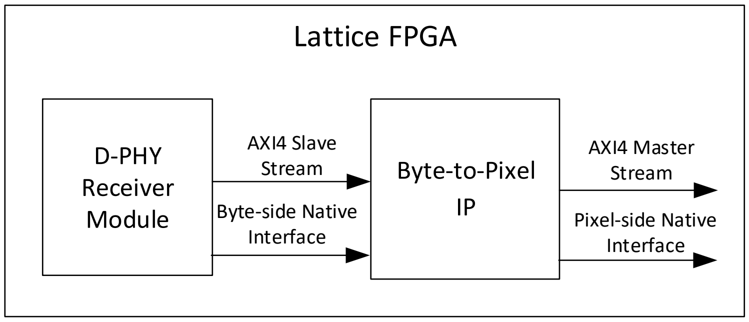


Figure 1.1. Byte-to-Pixel IP General Diagram

The Byte-to-Pixel Converter IP design is implemented in Verilog language. Lattice Radiant software Place and Route tool integrated with the Synplify Pro® synthesis tool is used for the implementation of the design. The design can be targeted to CrossLink™-NX family devices. When used on a different device, density, speed, or grade, performance and utilization may vary.

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
Resource Utilization	Targeted Devices	LIFCL-40, LIFCL-17, LFD2NX-40
	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
	Synthesis	Lattice Synthesis Engine (LSE)
		Synopsys® Synplify Pro for Lattice
	Simulation	For a list of supported simulators, see the Lattice Radiant Software 2.1 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 Slave, and AXI4 Device Master are used to synchronize the incoming D-PHY data bytes to the pixel clock domain.

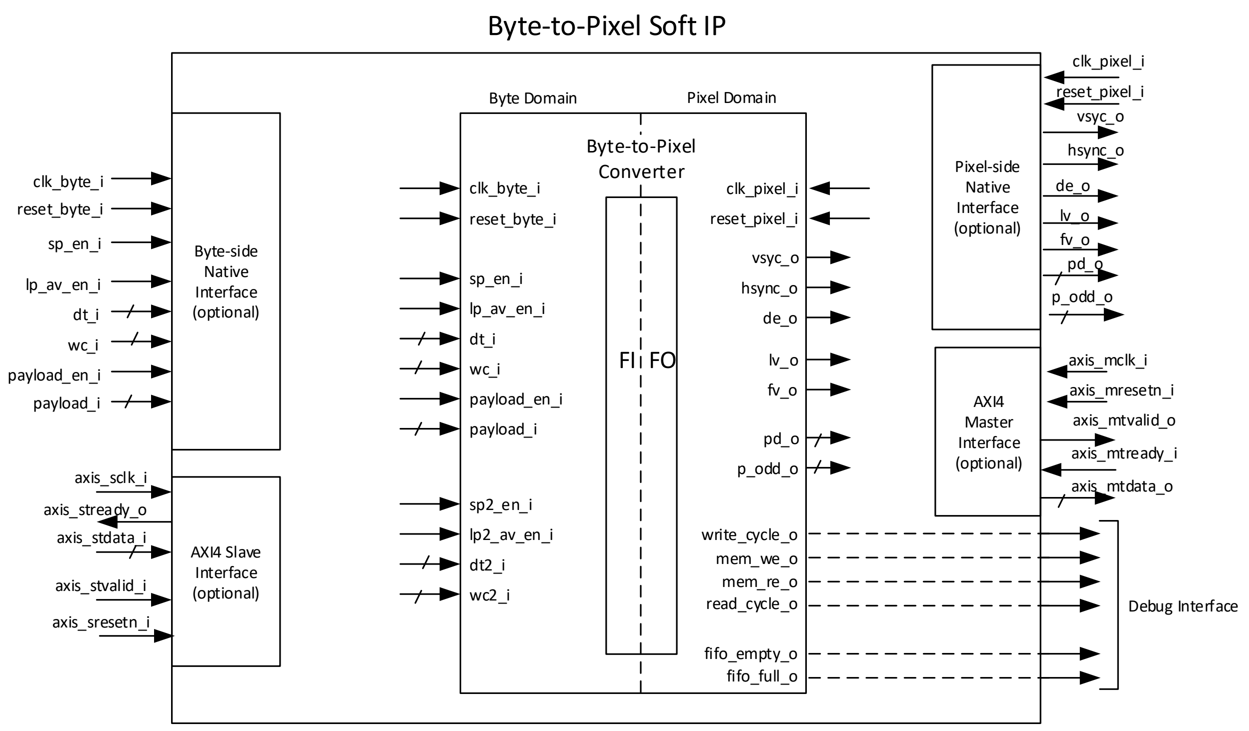


Figure 2.1. Byte-to-Pixel IP Functional Diagram

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

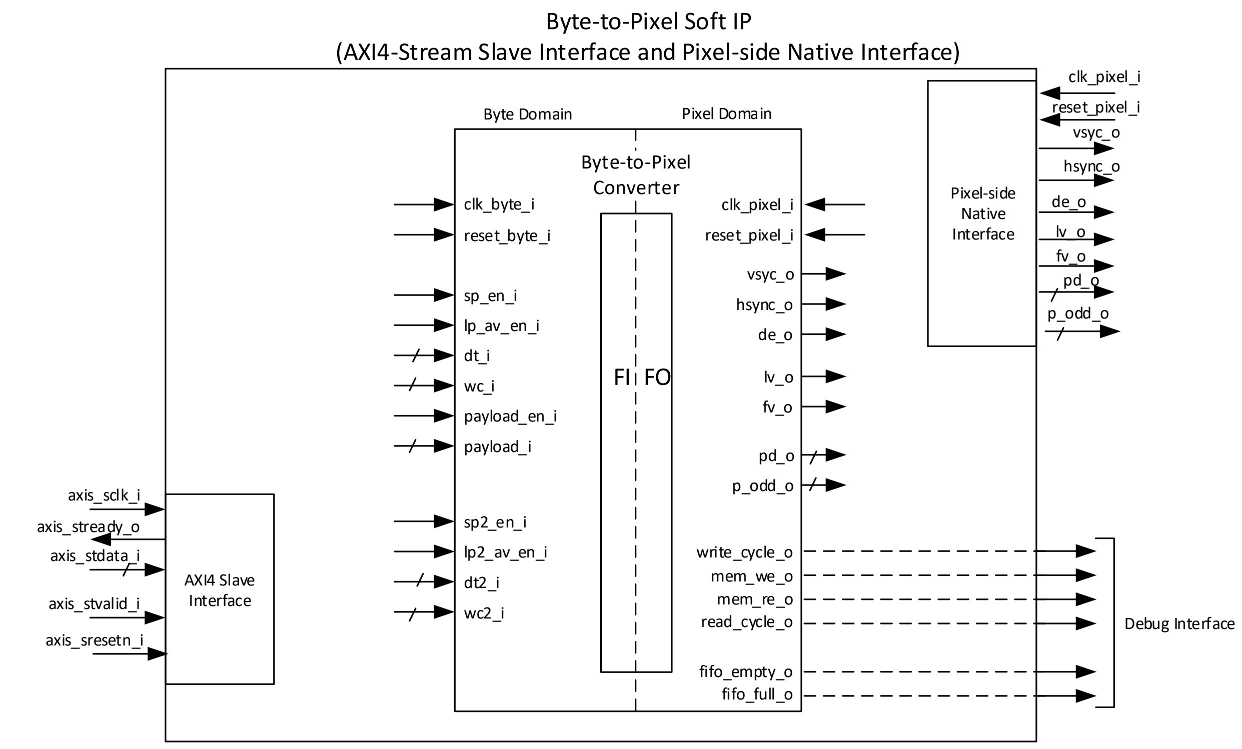


Figure 2.2. Case of AXI-Stream Slave and Pixel-side Native Interfaces

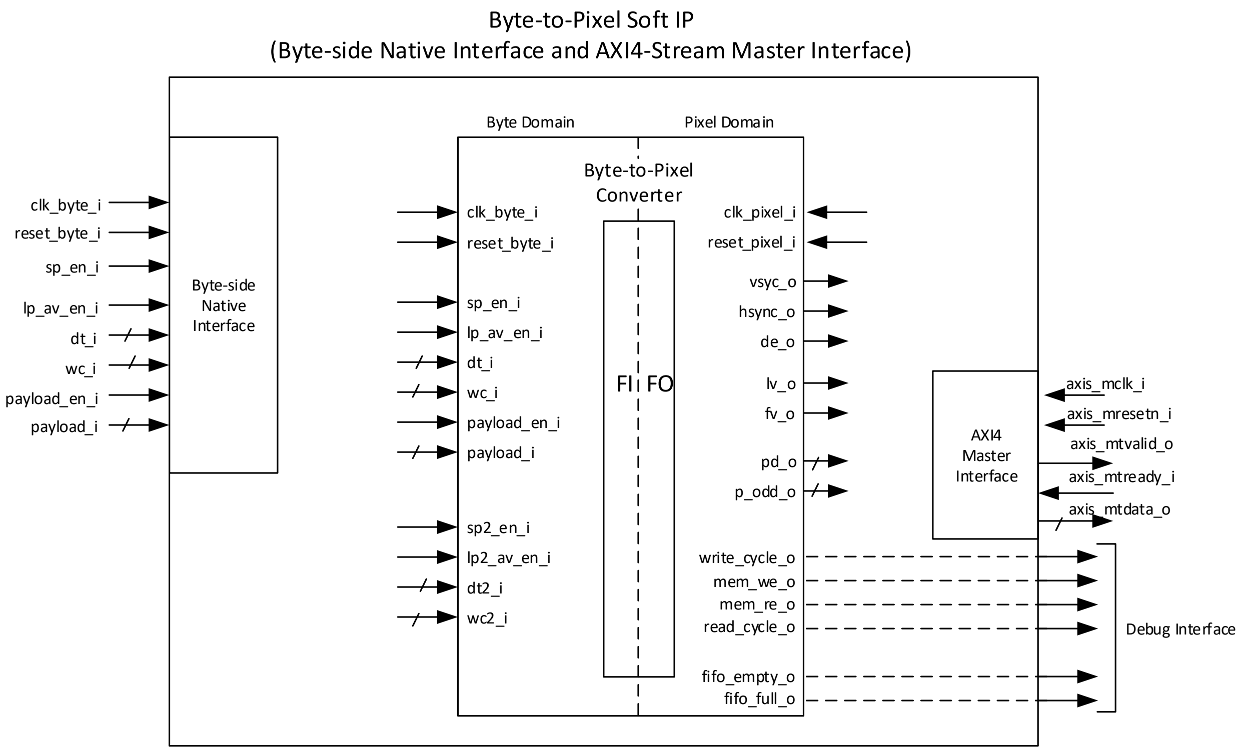


Figure 2.3. Case of Byte-side Native and AXI-Stream Master Interfaces

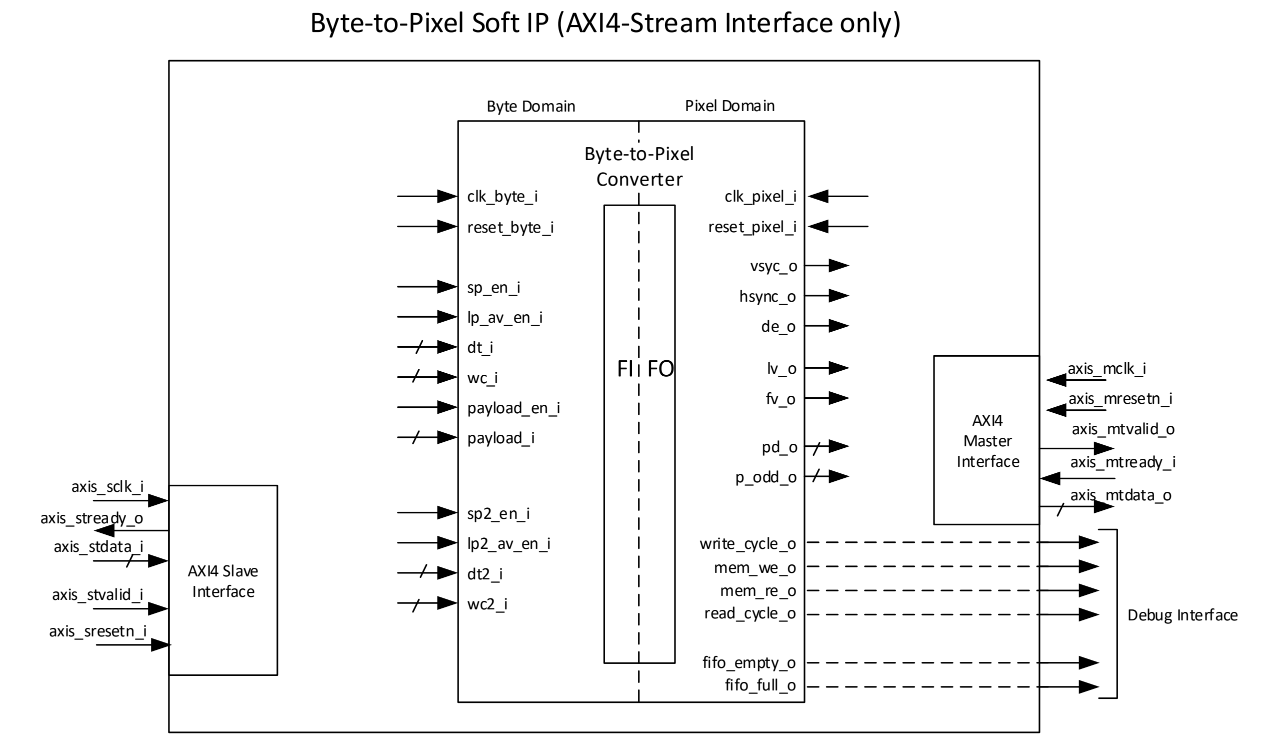


Figure 2.4. Case of AXI4-Stream Interface Only

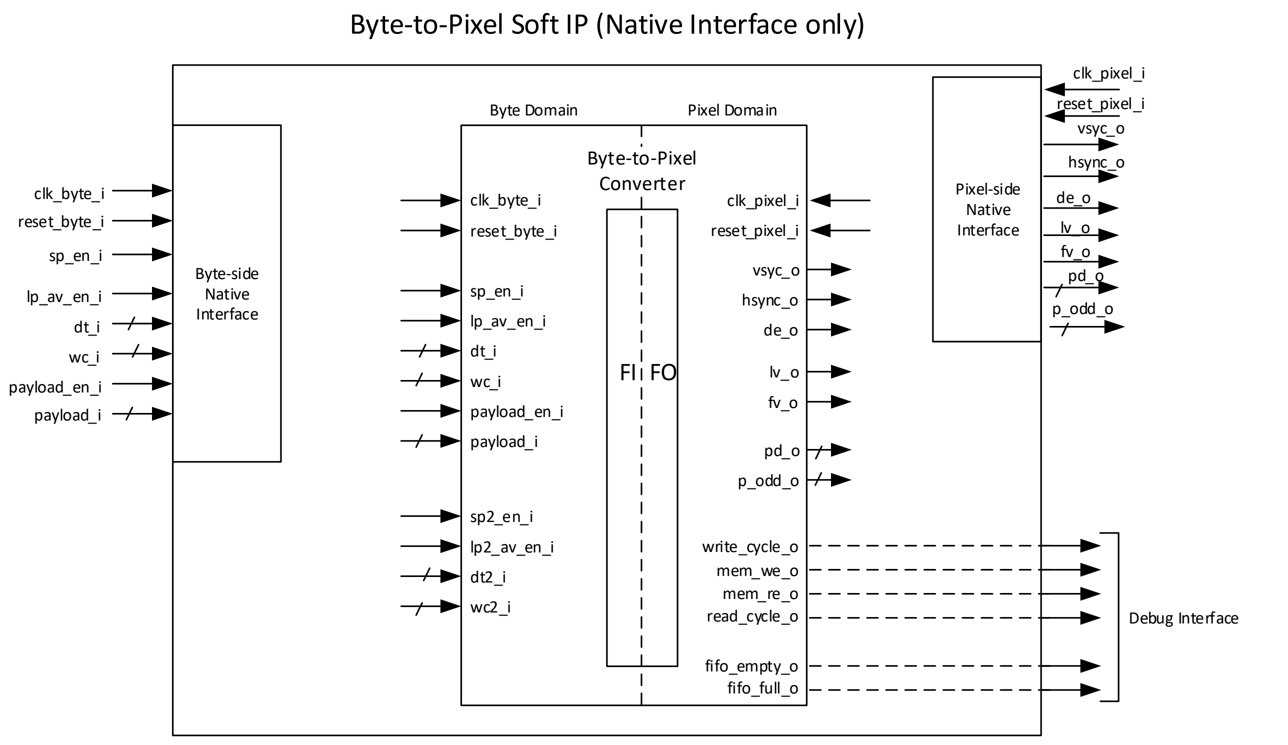


Figure 2.5. Case of Native Interface Only

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.
AXI4-Stream Slave Interface		
axis_sclk_i	in	AXI4-Stream Slave clock
axis_sresetn_i	in	Active low signal to reset the logic in the AXI4-Stream Slave domain
axis_svalid_i	in	AXI Stream Slave valid input signal
axis_sready_o	out	AXI Stream Slave 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 Slave 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 *Camera/Display Control Polarity* attribute is Positive. Otherwise, this is active Low.
hsync_o	out	HSYNC signal for DSI. Active High if *Camera/Display Control Polarity* attribute is Positive. Otherwise, this is active Low.
fv_o	out	Frame Valid signal for CSI-2. Active High if *Camera/Display Control Polarity* attribute is Positive. Otherwise, this is active Low.
lv_o	out	Line Valid signal for CSI-2. Active High if *Camera/Display Control Polarity* attribute is Positive. Otherwise, this is active Low.

Port Name	Direction	Description
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 Master 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 Master Stream valid output signal
axis_mready_i	in	AXI Master Stream ready input signal
axis_mdata_o[MASTER_DATA_W-1:0]	out	AXI Master Stream data**(See Notes at the end of this table)*
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 != 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
Receiver	RX Interface	DSI, CSI-2	CSI-2	—	—
	DSI Mode	Non-Burst Pulses, Non-Burst Events, Burst	Non-Burst Pulses	Grayed out unless *RX Interface* is *DSI*.	Values are selectable only when *RX Interface* is *DSI*.
	Number of RX Lanes	1, 2, 4	1	• 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	• Data Type • RX Interface • Number of RX Lanes • Number of Output Pixel Lanes	Use Table 2.3 and Table 2.4 for reference.
	Byte Side Clock Frequency (MHz) [10–350]	10–350	10	—	—
	Enable AXI4-Stream Slave Interface	Checked, Not Checked	Not Checked	—	—
	Receiver Data Rate	—	Calculated	Grayed out. • Number of RX Lanes • RX Gear • Byte Side Clock Frequency	—
Transmitter	Number of Output Pixel Lanes	1, 2, 4	1	• Data Type • RX Interface • Number of RX Lanes • RX Gear	Use Table 2.3 and Table 2.4 for reference.
	Data Type	RGB565, RGB666, RGB666_LOOSE, RGB888, RAW8, RAW10, RAW12, 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	• RX Interface • Number of RX Lanes • RX Gear • Number of Output Pixel Lanes	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 *RX Interface* is *DSI*.	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 Pix Clock Cycles HSYNC is Active [3–1023]	3–1023	8	Grayed out unless *RX Interface* is *DSI*.	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.
	Pixel Side Clock Frequency (MHz) [10–500]	10–500	50	—	—
	Enable AXI4-Stream Master Interface	Checked, Not Checked	Not Checked	—	—
	Transmitter Data Rate	—	500	Grayed out • Pixel Data Width • Number of Output Pixels • Pixel Side Clock Frequency	—
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 *Byte Count Restriction* for the actual value being used on Table 2.5.
Debug	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
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

Number of Output Pixel Lanes	D-PHY Lanes	RX Gearing	Data Type
		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, RGB888
		16	RGB565, 10-bit, RAW12, RGB888
	2 lanes	8	RGB565, 8-bit, 10-bit, RAW12, RGB888
		16	RGB565, RGB888
	4 lanes	8	RGB565, 8-bit, 10-bit, RAW12, RGB888
		16	RGB565
2 output pixels	1 lane	8	RGB565
		16	RGB565, 8-bit
	2 lanes	8	RGB565, 8-bit, 10-bit
		16	RGB565, 8-bit, 10-bit, RAW12, RGB888
	4 lanes	8	RGB565, 8-bit, 10-bit, RAW12, RGB888
		16	RGB565, 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

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 Slave

In byte domain, AXI4-Stream Slave 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 Master

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

For DSI and CSI2 modes, the output data is concatenation of the following two signals {p_odd_o, pd_o} exactly in that order. As the p_odd_o is two bit signal, total width of AXI4 stream data (axis_mdata_o) is equal to 2 + pixel_width*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.

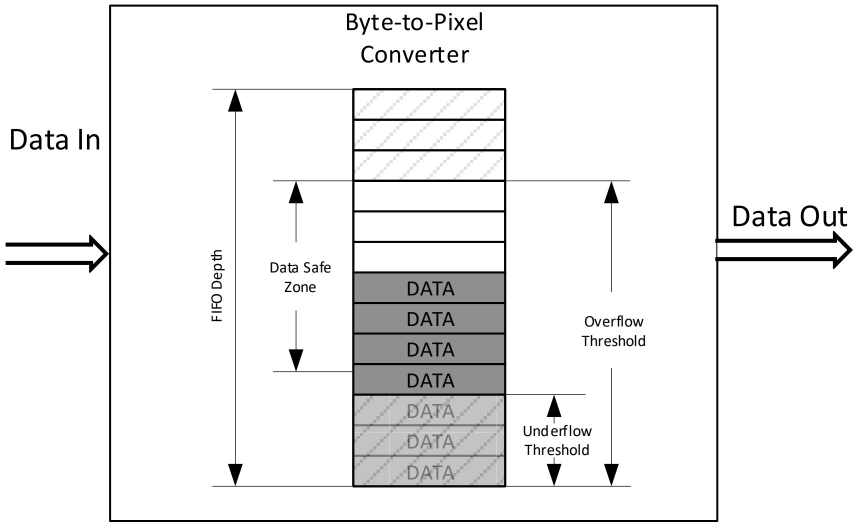


Figure 2.6. Byte-to-Pixel IP FIFO Diagram

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 the word width on byte side as Rx_width and the word width on pixel side as Tx_width. They are calculated as:

$$\text{Rx\_width} = \text{NUM\_Rx\_LANE} \times \text{Rx\_GEAR}$$
$$\text{Tx\_width} = \text{PD\_BUS\_WIDTH} \times \text{NUM\_Tx\_CH}$$

The FIFO Width depends on these two 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. They are calculated as:

$$\text{RX\_rate} = \text{RX\_width} \times \text{clk\_byte\_i}$$
$$\text{TX\_rate} = \text{TX\_width} \times \text{clk\_pixel\_i}$$

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

They 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.

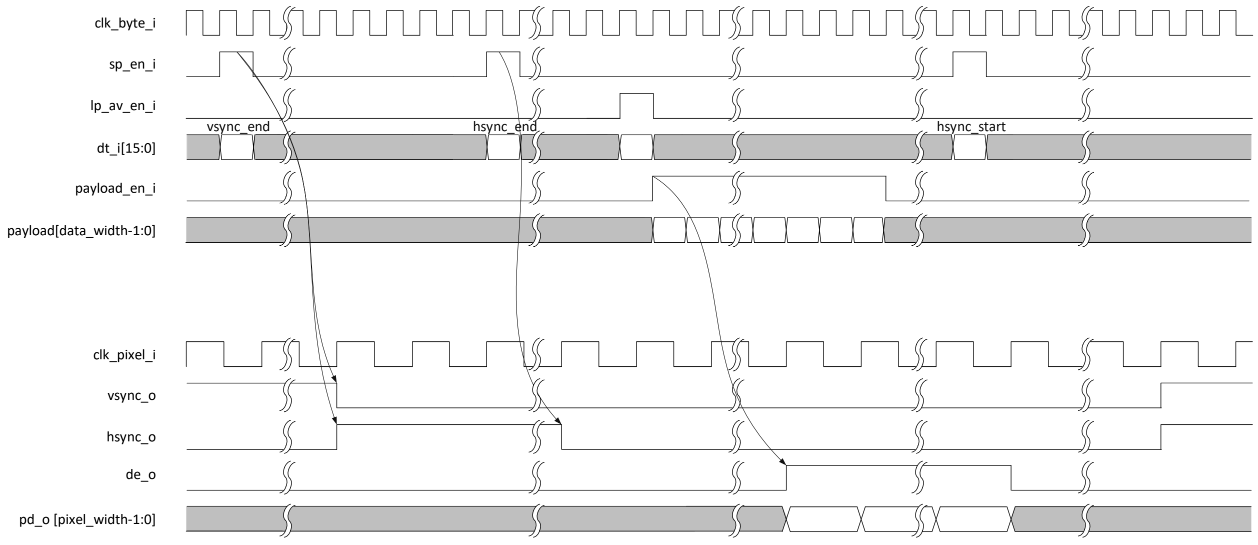


Figure 2.7. Timing Diagram between Inputs and Outputs for DSI

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. In both cases, an HSYNC end packet is expected next.

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

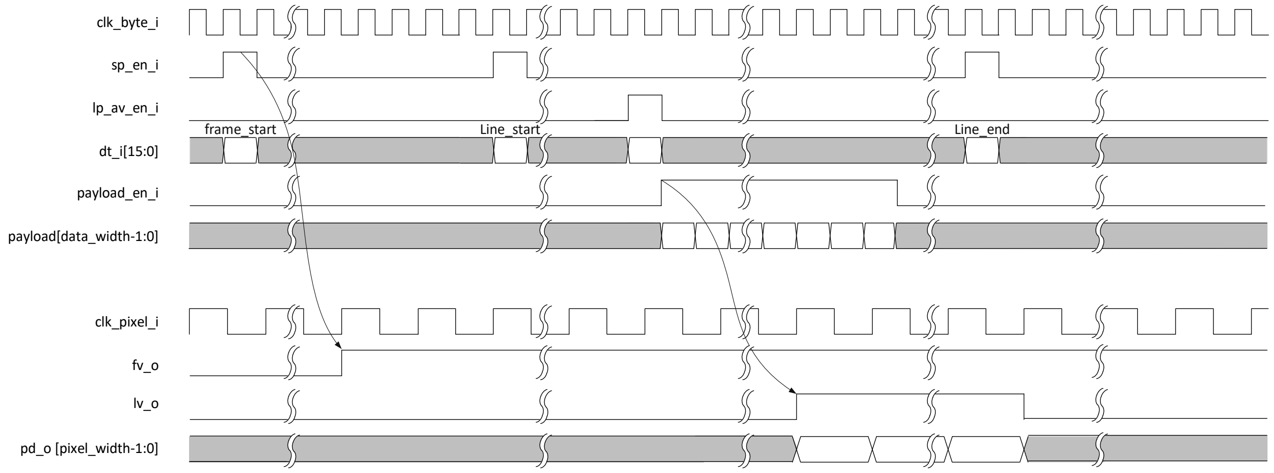


Figure 2.8. Timing Diagram between Inputs and Outputs for CSI-2

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.

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.

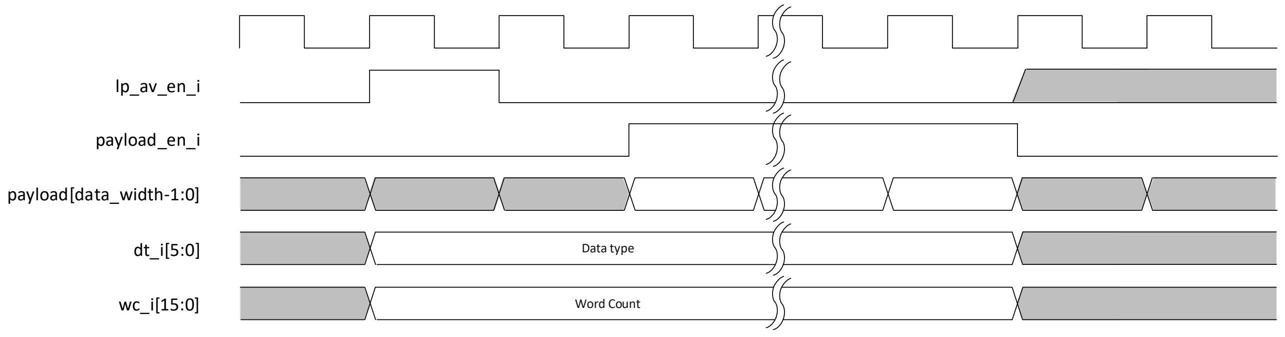


Figure 2.9. Input Timing Diagram for Byte-side

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.

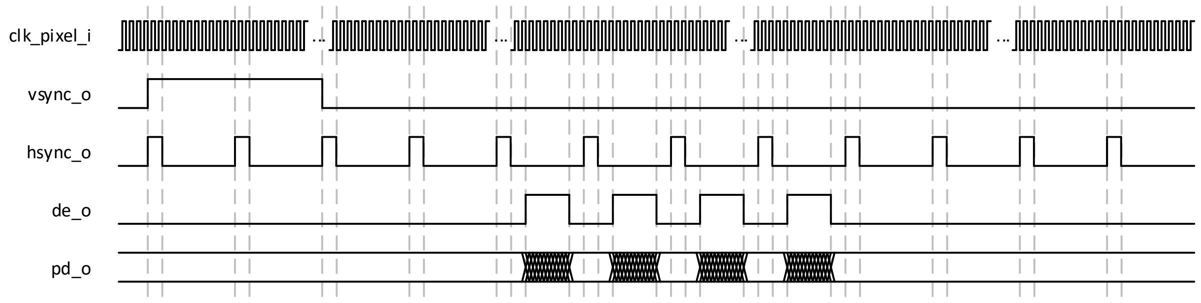


Figure 2.10. Output Timing Diagram from a DSI Input

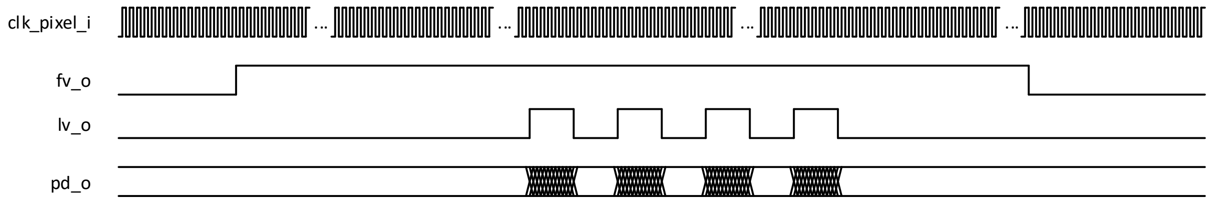


Figure 2.11. Output Timing Diagram from a CSI-2 Input

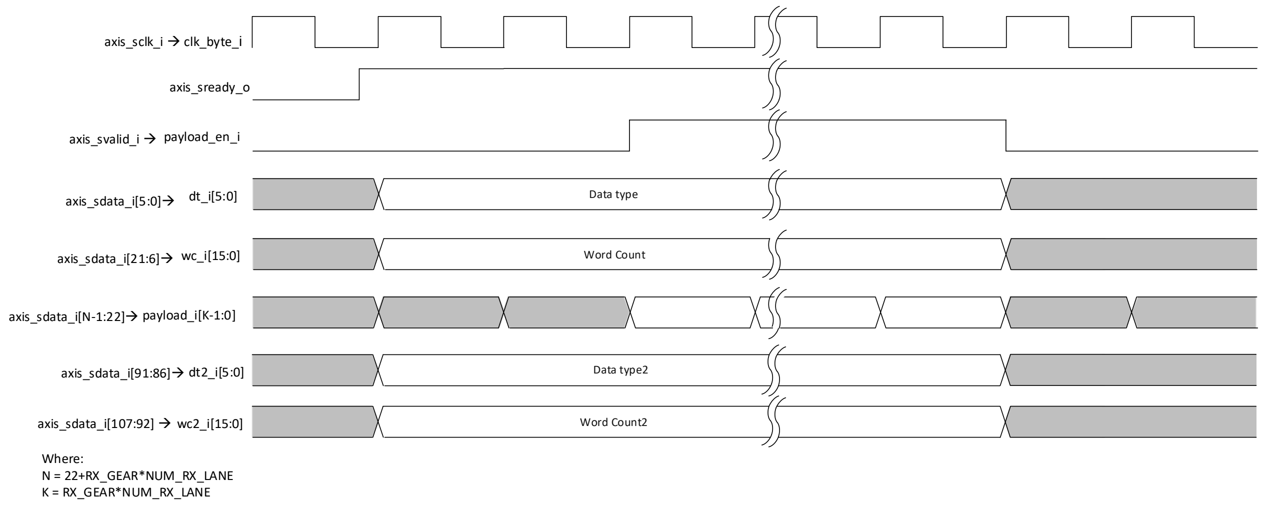

axis_sclk_i → clk_byte_i
axis_sready_o
axis_svalid_i → payload_en_i
axis_sdata_i[5:0]→ dt_i[5:0]
Data type
axis_sdata_i[21:6]→ wc_i[15:0]
Word Count
axis_sdata_i[N-1:22]→ payload_i[K-1:0]
axis_sdata_i[91:86]→ dt2_i[5:0]
Data type2
axis_sdata_i[107:92] → wc2_i[15:0]
Word Count2
Where:
N = 22+RX_GEAR*NUM_RX_LANE
K = RX_GEAR*NUM_RX_LANE

Figure 2.12. AXI4 Stream Slave Interface Diagram

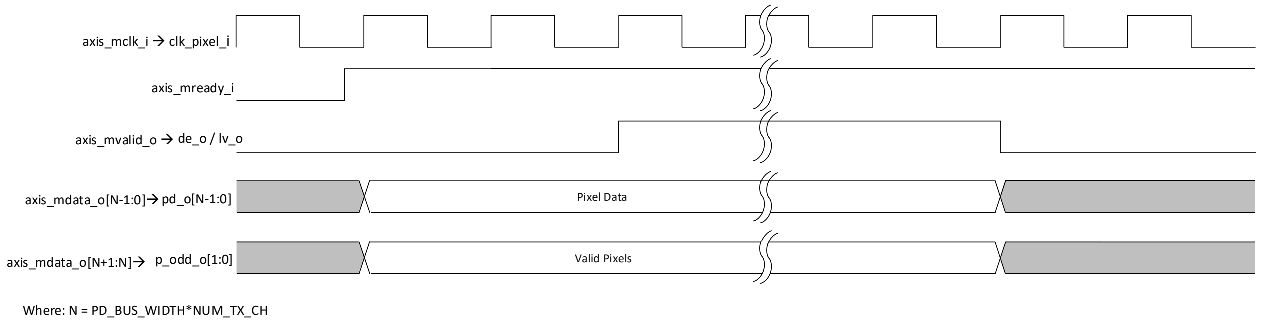

axis_mclk_i → clk_pixel_i
axis_mready_i
axis_mvalid_o → de_o / lv_o
axis_mdata_o[N-1:0]→ pd_o[N-1:0]
Pixel Data
axis_mdata_o[N+1:N]→ p_odd_o[1:0]
Valid Pixels
Where: N = PD_BUS_WIDTH*NUM_TX_CH

Figure 2.13. AXI4-Stream Master Interface Diagram

3. IP Generation and Evaluation

This chapter provides information on how to generate and synthesize Byte-to-Pixel Converter IP Core using Lattice Radiant software and how to run simulation. For more on Lattice Radiant software, refer to the Lattice Radiant Software 2.0 User Guide and relevant Lattice tutorials.

3.1. Licensing the IP

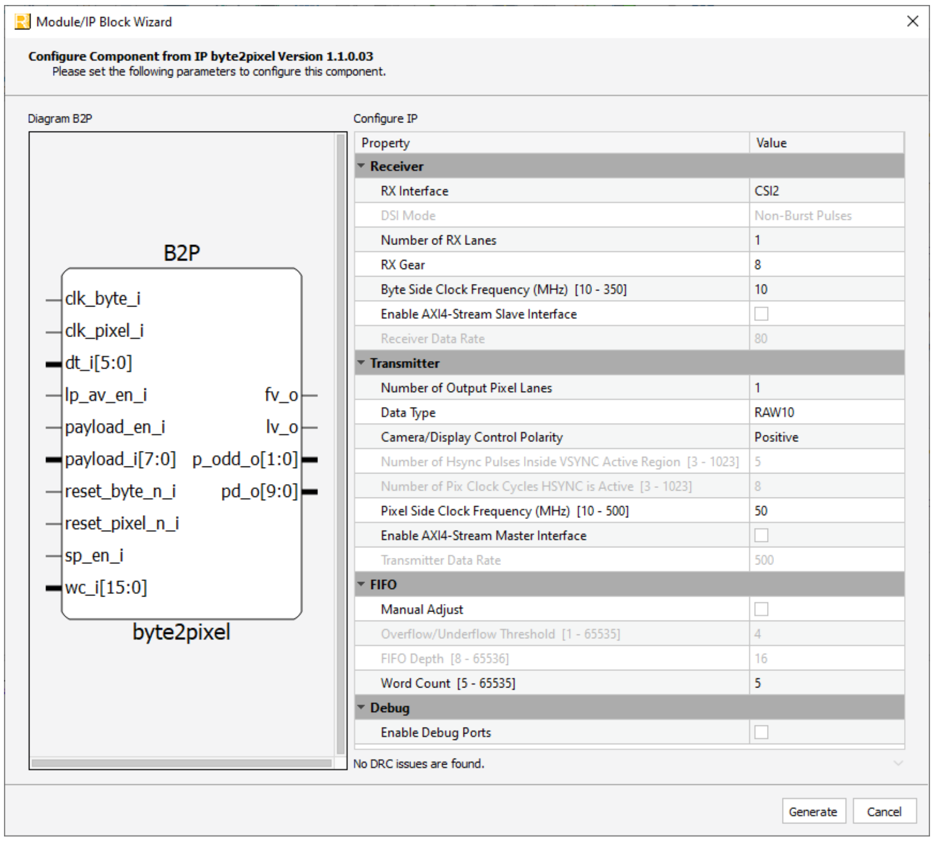
An IP core-specific license string is required enable full use of this IP core in a complete, top-level design. You can fully evaluate the IP core through functional simulation and implementation (synthesis, map, place and route) without an IP license string. This IP core supports Lattice's IP hardware evaluation capability, which makes it possible to create versions of the IP core, which operate in hardware for a limited time (approximately four hours) without requiring an IP license string. See Hardware Evaluation section for further details. However, a license string is required to enable timing simulation and to generate bitstream file that does not include the hardware evaluation timeout limitation.

3.2. Generation and Synthesis

Lattice Radiant software allows you 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.



Configure Component from IP byte2pixel Version 1.1.0.03
Please set the following parameters to configure this component.

Property	Value
Receiver	
RX Interface	CSI2
DSI Mode	Non-Burst Pulses
Number of RX Lanes	1
RX Gear	8
Byte Side Clock Frequency (MHz) [10 - 350]	10
Enable AXI4-Stream Slave Interface	☐
Receiver Data Rate	80
Transmitter	
Number of Output Pixel Lanes	1
Data Type	RAW10
Camera/Display Control Polarity	Positive
Number of Hsync Pulses Inside VSYNC Active Region [3 - 1023]	5
Number of Pix Clock Cycles HSYNC is Active [3 - 1023]	8
Pixel Side Clock Frequency (MHz) [10 - 500]	50
Enable AXI4-Stream Master Interface	☐
Transmitter Data Rate	500
FIFO	
Manual Adjust	☐
Overflow/Underflow Threshold [1 - 65535]	4
FIFO Depth [8 - 65536]	16
Word Count [5 - 65535]	5
Debug	
Enable Debug Ports	☐

No DRC issues are found.

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, you can synthesize it by clicking **Synthesize Design** located on the top left corner of the screen, as shown in Figure 3.2.

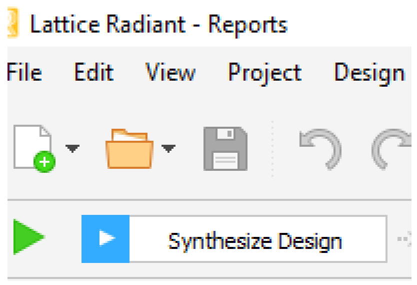
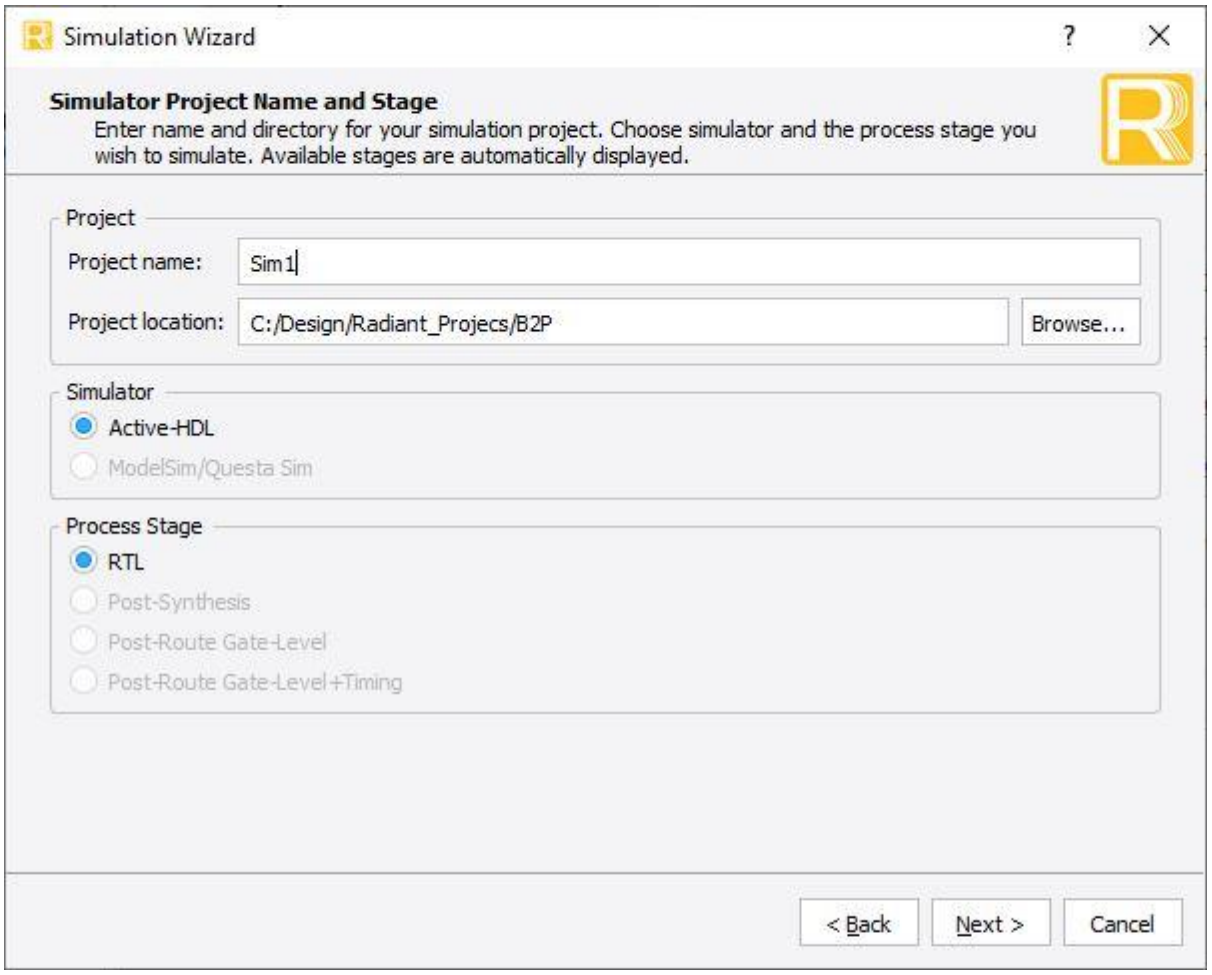


Figure 3.2. Synthesizing the Design

3.3. Functional Simulation

To run Verilog simulation:

1. Click the button located on the Toolbar to initiate the **Simulation Wizard** shown in Figure 3.3.



Simulator Project Name and Stage
Enter name and directory for your simulation project. Choose simulator and the process stage you wish to simulate. Available stages are automatically displayed.

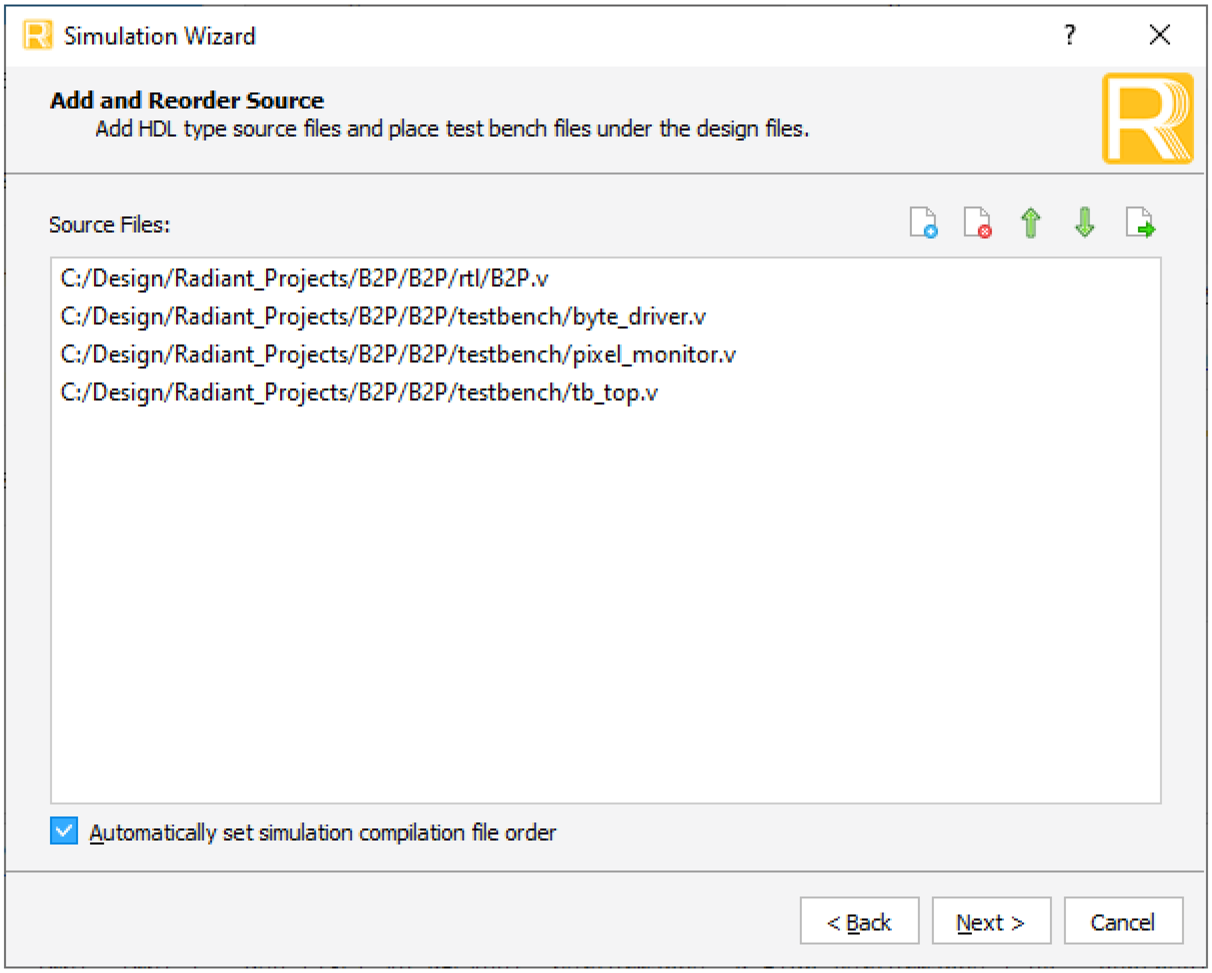
Project
Project name: Sim1
Project location: C:/Design/Radiant_Projecs/B2P

Simulator
[✓ Active-HDL]
ModelSim/Questa Sim

Process Stage
[✓ RTL]
Post-Synthesis
Post-Route Gate-Level
Post-Route Gate-Level+Timing

Figure 3.3. Simulation Wizard

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



Add HDL type source files and place test bench files under the design files.

C:/Design/Radiant_Projects/B2P/B2P/rtl/B2P.v
C:/Design/Radiant_Projects/B2P/B2P/testbench/byte_driver.v
C:/Design/Radiant_Projects/B2P/B2P/testbench/pixel_monitor.v
C:/Design/Radiant_Projects/B2P/B2P/testbench/tb_top.v

✓ Automatically set simulation compilation file order

Figure 3.4. Adding and Reordering Source

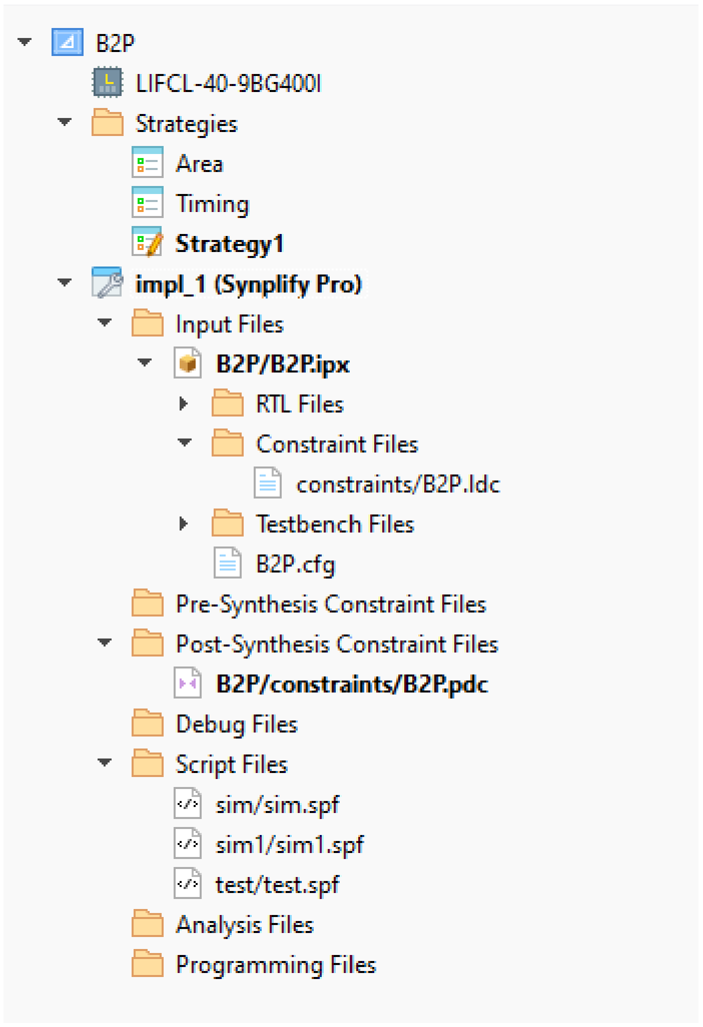
3. Add the *tb_top.v* file from the *testbench* directory.
4. Click **Next** to run the simulation.

3.3.1. Required Post-Synthesis Constraints

The Byte to Pixel Converter IP for Radiant has a default pre-synthesis constraints file *.ldc located in the constraints folder. For Map, Place and Route, a Post-Synthesis Constraints (*.pdc) file is needed as shown in Figure 3.5.

To properly constrain the byte to pixel design post-synthesis:

1. Uncomment *set_multi_cycle_path* from the *.ldc based on the synthesis tool being used, and then copy the constraints from the *.ldc file to the active *.pdc file of the Radiant project.
2. By default, the periods of the clocks are for 200 MHz designs. Adjust those based on the actual design to relax the timing.
3. Modify the paths based on the generated netlist.



```
B2P
    LIFCL-40-9BG400I
    Strategies
        Area
        Timing
        Strategy1
    impl_1 (Synplify Pro)
        Input Files
            B2P/B2P.ipx
                RTL Files
                Constraint Files
                    constraints/B2P.ldc
                Testbench Files
                B2P.cfg
        Pre-Synthesis Constraint Files
        Post-Synthesis Constraint Files
            B2P/constraints/B2P.pdc
        Debug Files
        Script Files
            sim/sim.spf
            sim1/sim1.spf
            test/test.spf
        Analysis Files
        Programming Files
```

Figure 3.5. Adding Constraint

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

Appendix A. Resource Utilization

Table A.1 and Table A.2 shows 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*.

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 2.1 User Guide.

Technical Support Assistance

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

Revision History

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

Section	Change Summary
Acronyms in This Document	Updated content.
Introduction	• Updated Table 1.1. • Updated Features section.
Functional Description	• 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	• 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	• 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